S2CRA: Deterministic Verifier-Guided Proposal Channels for Auditable Causal Graph Discovery

Anonymous ACL submission

Abstract

Causal graph discovery remains brittle. Statistical methods are limited by equivalence classes and score misspecification, while direct LLM reasoning can contribute useful priors but is difficult to audit. We study a narrower design in which the LLM serves only as a proposal channel and a deterministic verifier controls the search. We present **S2CRA**, an atomic-edit loop that starts from a statistical initializer, lets the LLM propose one graph edit at a time, and accepts only edits that pass deterministic structural checks and improve a fixed score. The result is a fully replayable search trace. Under a leakage-safe observational-only BNRepository protocol, S2CRA improves oriented-edge recovery over classical baselines. Matched non-LLM controls and controlled text regimes further separate verifier effects from language effects. Overall, the system treats language as a bounded source of hypotheses rather than an unchecked graph generator.

1 Introduction

Causal discovery, the recovery of a directed acyclic graph (DAG) from data, underpins scientific reasoning and decision support (Pearl, 2009; Spirtes et al., 2000), yet it remains fragile: observational data identify graphs only up to Markov equivalence, finite samples reduce test power, and model assumptions are imperfect. Constraint- and score-based methods are therefore principled but limited (Kalisch and B"uhlmann, 2007; Chickering, 2002; Zheng et al., 2018).

Large language models (LLMs) make this tension more visible. Variable names and descriptions can provide useful priors (Kıcıman et al., 2024; Ban et al., 2023), but direct graph generation is prompt-sensitive and hallucination-prone unless checked against data (Jiralerspong et al., 2024; Long et al., 2023; Zhou et al., 2024; Roy et al., 2025). Recent hybrid systems therefore point to

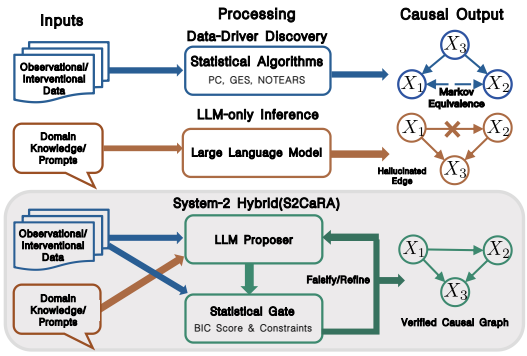

Figure 1: Conceptual contrast: statistical discovery respects observed data but faces ambiguity, while direct LLM inference offers semantic priors but can hallucinate. S2CRA restricts language to a proposal channel and uses data as a deterministic verifier.

a more disciplined role for language: propose hypotheses that a verifier can accept or reject.

This motivates a more specific question: can an LLM serve as an auditable *proposal channel* inside a deterministic, edit-level search contract, so that the language contribution is visible at the level of proposals rather than hidden inside free-form graph generation? The model may suggest an initial graph or one atomic edit at a time; the verifier deterministically enforces structural validity and requires a fixed score improvement. The resulting edit trace is replayable and auditable.

We instantiate this idea in **S2CRA** and evaluate it under a **graph-first** protocol designed to isolate the proposal channel: anonymized variables, hidden dataset names, statistical initialization in the BN main track (LLM initialization disabled), and fixed hard/score gates with full trace logging. Because the loop uses an observational score, our headline results are observational-only; intervention analyses are supplementary.

Our contributions are:

- **Edit-level verifier contract for LLM proposals:** atomic edits are proposed by the LLM but accepted only if they pass deterministic hard

constraints and improve a fixed score, with a replayable trace.

- **Matched proposer controls:** non-LLM atomic-search policies under identical gates and budgets, isolating proposer value beyond verifier-guided search.
- **Controlled text regimes:** fixed variable-text conditions (including misleading text) under the same verification, quantifying semantic influence and risk under leakage-safe evaluation.

Section 2 positions the contribution against nearby hybrids; Sections 3–5 then define the verifier-guided contract, the pre-specified protocol, and the evidence.

2 Related Work

Recent work on causal discovery with language models falls into three lines that are most relevant here: classical data-driven discovery, direct language-based inference, and hybrid systems that combine language with statistical verification. For our setting, the most important differences concern where language enters the pipeline, whether refinement is carried out through atomic edits, and whether candidate changes are accepted under a deterministic, auditable verifier.

Data-driven discovery. Classical causal discovery uses statistical tests or score-based search, including PC, FCI, and GES (Kalisch and B"uhlmann, 2007; Spirtes et al., 1995; Chickering, 2002), while continuous relaxations such as NOTEARS trade discrete search for differentiable optimization (Zheng et al., 2018). These methods are grounded in data but still struggle with finite samples, score misspecification, and Markov equivalence (Pearl, 2009; Spirtes et al., 2000). Amortized discovery models such as AVICI learn graph inference across tasks (Lorch et al., 2022).

Direct LLM inference. Another line of work uses language models as semantic priors, either for pairwise causal judgments from metadata (Kıcıman et al., 2024; Jin et al., 2024) or for graph-level proposal through prompting (Ban et al., 2023; Jiralerspong et al., 2024; Roy et al., 2025); see also recent surveys of LLM-assisted discovery (Wan et al., 2025). These methods are most promising when variable names or descriptions carry reliable domain knowledge, but their behavior is also sensitive to prompt design, metadata quality, and benchmark leakage. Several recent studies document these concerns, including hallucination, weak grounding, and fragility under anonymization or harder structural settings (Long et al., 2023; Zhou et al., 2024; Feng et al., 2025b; Chi et al., 2024; Chen et al., 2023).

Verified or co-refined hybrids. More recent systems place language models inside iterative pipelines, often together with debate, retrieval, tool use, or explicit verifier loops (Le et al., 2024; Lewis et al., 2020; Shen et al., 2024; Zhang et al., 2024; Khatibi et al., 2024; Abdulaal et al., 2024; Feng et al., 2025a; Peng et al., 2026). This line of work is the closest comparison point for S2CRA. Within this group, our focus is on a fixed edit-level contract shared across initialization and refinement, leakage-safe evaluation with anonymized and text-regime controls, and replayable traces that record both accepted and rejected edits under the same verifier. These design choices make the proposal–verification interface explicit and allow runs to be compared under a common protocol.

3 Methodology

We present **S2CRA**, a *deterministic verifier-guided proposal channel* for causal graph discovery: language may suggest hypotheses, but data-derived gates decide which edits survive. Figure 2 summarizes the overall workflow.

3.1 Problem Setup

Let $V = \{X_1, \ldots, X_d\}$ be a set of variables. We observe samples $\mathcal{D}_{\text{obs}}$ and, optionally, additional samples $\mathcal{D}_{\text{int}}$ collected under single-node hard interventions (Pearl, 2009). A candidate graph is a DAG $G = (V, E)$. Our goal is to recover a graph $\widehat{G}$ together with an auditable trace of how it was obtained.

The trace is a first-class output. It records the initializer choice, each proposed edit, deterministic rejection reasons, score deltas for accepted steps, and the stopping condition, enabling offline replay and inspection.

3.2 Unified Score Gate

We use a fixed lower-is-better score as the acceptance criterion:

$$\text{Score}(G) = S_{\text{data}}(G) + \lambda|E|,$$

where $S_{\text{data}}(G)$ is computed from the available tables and $\lambda \geq 0$ is an optional sparsity penalty. Figure 3 shows the two gates that define the contract.

Work	LLM init (graph)	Atomic edits	Deterministic accept/reject	Same score init/refine	Leakage-safe eval	Text-centric eval	Trace audit	Interv.-aware verifier
CMA (Abdulaal et al., 2024)	Y	–	P	–	–	–	–	–
Takayama et al. (Takayama et al., 2025)	Y	–	P	–	–	–	–	–
ALCM (Khatibi et al., 2024)	Y	Y	P	–	–	–	–	–
IRIS (Feng et al., 2025a)	Y	–	Y	–	–	–	P	–
Causal-LLM (Roy et al., 2025)	Y	–	P	–	–	–	–	–
CauScientist (Peng et al., 2026)	Y	Y	Y	P	–	–	P	P
S2CRA (ours)	(opt.)	Y	Y	Y	Y	Y	Y	(gated)

Table 1: Comparison of nearby hybrid systems along the properties most relevant to our protocol. "P" indicates a partial or non-fixed form.

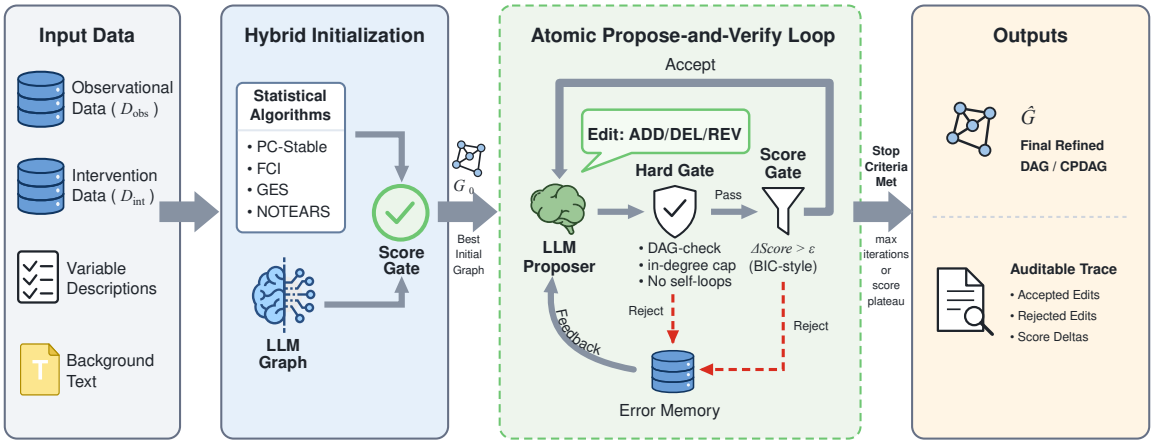

Figure 2: Overview of S2CRA. A candidate pool provides the starting graph, after which the LLM may propose only one atomic edit at a time. Every accepted change must pass a hard structural gate and a fixed score-improvement gate.

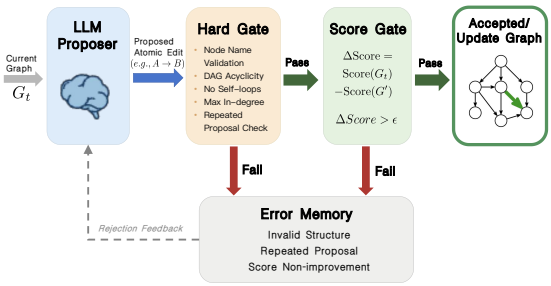

Figure 3: Verifier contract used by S2CRA. An edit must satisfy structural validity first; only then is it tested by the fixed score-improvement gate. Rejections are kept in memory for replay and prompt conditioning.

In the observational-only main track we instantiate the score gate with a BIC-style penalized likelihood (Schwarz, 1978) (plus the same edge penalty term). Intervention-aware scoring is treated as a secondary track; supplementary intervention analyses use sample concatenation as a pragmatic protocol rather than claiming a fully intervention-aware criterion.

For a proposed edit that transforms G_t into G', we compute $\Delta = \text{Score}(G_t) - \text{Score}(G')$. The edit is accepted only if it passes the hard gate and $\Delta > \epsilon$. This makes acceptance deterministic once the proposal is fixed.

3.3 Initialization and Atomic Search

Initialization matters, so S2CRA scores a candidate pool and selects the best-scoring graph as G_0. The pool may contain statistical baselines and, optionally, an LLM proposal; in the BN main track we disable LLM initialization to keep G_0 deterministic and benchmark-safe.

Starting from G_0, the model may propose only one atomic operation at a time: ADD, DEL, or REV. The proposal can be conditioned on the current graph, current score, and a short memory of recent rejections. At the edge-set level,

$$\begin{aligned} \text{ADD}(u{\to}v) &: E' = E \cup \{(u,v)\}, \\ \text{DEL}(u{\to}v) &: E' = E \setminus \{(u,v)\}, \\ \text{REV}(u{\to}v) &: E' = (E \setminus \{(u,v)\}) \cup \{(v,u)\}. \end{aligned}$$

Before scoring, we enforce hard constraints: valid node names, no self-loops, acyclicity, and an optional max-in-degree cap. Only edits that satisfy

$$G' \text{ is acyclic} \quad \wedge \quad \max_{X_i \in V} \deg^-_{G'}(X_i) \le d_{\max}$$

Input: variables V, data table $\mathcal{D}_{\text{obs}}$, score gate $\text{Score}(\cdot)$, hard constraints ($d_{\max}$, acyclicity), iteration budget T, patience k.
Output: final graph $\widehat{G}$ and auditable trace Trace.
(1) Init: run statistical initializers; set $G_0 = \arg\min_G \text{Score}(G)$ over candidates.
(2) Loop: for t=1..T, propose atomic edit a_t; apply *HardGate*; accept iff $\text{Score}(G') < \text{Score}(G_{t-1}) - \epsilon$, else reject and log reason.
(3) Stop: early-stop if no accepted edit occurs for k steps; return best-scoring graph in the trace.

Figure 4: Pseudo-code for the verifier-guided atomic-edit contract (language proposes, data verifies).

reach the score gate. Rejected proposals are logged with reasons, which is why the method is auditable even when the LLM is imperfect.

3.4 Scope of Claims

The scope of the method is a verifier-guided proposal loop. Language supplies candidate graphs or edits, while acceptance is determined by a fixed edit-level verifier shared across the run. The resulting trace can be replayed and evaluated under the same parity-aware metrics used for the final graph. Figure 4 provides the compact algorithmic view of the contract.

4 Experiments

We evaluate S2CRA under a pre-specified protocol designed to isolate the *proposal channel*. We report (i) an observational-only BNRepository main track, (ii) matched non-LLM proposer controls under the same verifier contract, and (iii) a controlled text-regime study that changes only variable-side language under fixed verification. All numbers are prompting-only (no fine-tuning). Unless otherwise stated, S2CRA and LLM-ZERO use GPT-5.2 as the proposal model.

4.1 Benchmarks and Splits

Our main test bed is BNRepository/`bnlearn` built from BIF files (Elidan, 2001; Scutari, 2010). We evaluate four discrete networks: *Cancer*, *Asia*, *Child*, and *Alarm* (5, 8, 20, and 37 nodes, respectively). For each network we use observational data only, hide the dataset name, and evaluate three seeds. Appendix Table 6 lists the exact network sizes.

Protocol selection is separated from the main benchmark. We use a synthetic development suite of eight discrete DAGs spanning the same rough size range ($5/8/20/37$ nodes), with two seeds per graph, to choose the global protocol. Once selected, the protocol is transferred to the BN main track without retuning. The intervention setting is retained only as a secondary track and does not carry the main claim. Full protocol-selection and protocol-lock details are deferred to Appendix Tables 7 and 8.

4.2 Pre-specified Main Protocol

The main track uses a deterministic verifier contract. LLM initialization is disabled; the initial graph G_0 is selected from a statistical initializer pool. Refinement is then restricted to atomic edits (`ADD`/`DEL`/`REV`), filtered by the same hard structural gate and score-improvement gate throughout the run. Any gains attributable to language must occur under the same verifier contract shared with the matched non-LLM search baselines. Appendix Table 8 lists the fixed main-track settings.

4.3 Baselines and Evaluation

The main comparative matrix contains PC-Stable (Colombo and Maathuis, 2014), FCI (Spirtes et al., 1995), GES (Chickering, 2002), NOTEARS (Zheng et al., 2018), and our one-shot LLM-ZERO ablation, together with S2CRA on the same observational-only benchmark. To test whether the proposer itself is necessary, we additionally compare against four matched non-LLM search policies that share the same initializer pool, hard gate, score gate, move set, and budget: exhaustive best-improvement local search (Russell and Norvig, 2021), uniform random atomic search (Russell and Norvig, 2021), tabu search (Glover, 1986), and beam search (Russell and Norvig, 2021). These proposer controls separate the effect of the proposal policy from the effect of the verifier-guided search loop itself.

We report SHD, Skeleton F1, and Oriented-edge F1 (OE-F1). Oriented-edge F1 requires the correct endpoints and direction; Skeleton F1 evaluates the undirected skeleton, providing a parity view for partially oriented outputs. For constraint-based methods (PC/FCI), Skeleton F1 is computed on their skeleton view when available; oriented-edge metrics use only oriented edges.

To examine the language component directly, we include a controlled text-regime study. Under the same observational-only protocol we eval-

Method	SHD↓	Skel-F1↑	OE-F1↑
LLM-based			
S2CRA	**13.667**	**0.823**	**0.631**
LLM-zero	21.167	0.068	0.068
Classical baselines			
GES	14.583	0.477	0.458
PC-Stable	18.833	0.727	0.287
FCI	19.167	0.727	0.100
NOTEARS	23.667	0.300	0.149

Table 2: Observational-only BNRepository results on Cancer, Asia, Child, and Alarm.

Proposer	SHD↓	Skel-F1↑	OE-F1↑
S2CRA	13.667	**0.823**	0.631
Exhaustive	13.083	0.809	0.630
Tabu	**12.917**	0.809	**0.641**
Beam	13.583	0.809	0.620
Random	13.967 ± 0.217	0.774 ± 0.007	0.555 ± 0.013

Table 3: Matched proposer controls under the fixed verifier contract.

uate four text regimes: ANONYMIZED, RAW NAMES, RAW+DESC, and MISLEADING. Only the variable-side text changes; the verifier, data, and search budget remain fixed. This lets us measure whether language changes the proposal distribution, whether misleading text can induce bad proposals, and whether the verifier contains those errors.

Appendix A provides protocol details, prompt templates, structured output formats, and supplementary diagnostics.

5 Results and Analysis

We report results under the pre-specified observational-only BNRepository protocol (Section 4), designed to be leakage-safe (anonymized variables; hidden dataset names). Unless otherwise stated, BN numbers average over Cancer/Asia/Child/Alarm and three random seeds. **Headline.** Under this fixed protocol, S2CRA improves oriented-edge recovery over classical baselines (Table 2). We then isolate proposer effects with matched atomic-search controls under the same verifier contract (Tables 3–4) and probe language effects with controlled text regimes that vary only variable-side language (Table 5).

5.1 Observational-Only Main Results

Table 2 is the primary headline table. It reports BNRepository under the protocol fixed on a held-out development suite and then applied without retuning. This is the main evaluation view because the verifier used in the loop is observational. S2CRA achieves 0.631 oriented-edge F1 (Skel-F1 0.823), outperforming the strongest classical baseline (GES, 0.458) by 0.173. We report mean±std over complete reruns where applicable.

What improves (and what does not). Two points matter here. First, baselines that output partially oriented graphs can recover a reasonable skeleton but lag on oriented edges (e.g., PC-Stable/FCI in Table 2); S2CRA improves oriented-edge quality while maintaining a strong skeleton. Second, prompting without data-side verification is brittle under anonymization: LLM-ZERO fails under this protocol, motivating the separation between proposer and verifier.

Per-network view. Mean scores can hide heterogeneity across networks, so we report the full per-network breakdown in Appendix A.4.1. Performance varies across networks: relative to GES, S2CRA improves oriented-edge F1 on *Cancer* (+0.223), *Asia* (+0.126), and *Child* (+0.358), but trails slightly on *Alarm* (-0.016). Table 2 should therefore be read as a mean result over heterogeneous cases, not as uniform dominance on every network.

5.2 Proposer Necessity Under a Fixed Verifier

Table 3 compares S2CRA against matched non-LLM proposer baselines under the same initializer pool, hard gate, score gate, move set, and budget. This is the relevant control for the proposer, since the search policies share the same initializer pool, gates, move set, and budget. Exhaustive best-improvement matches the full method within rounding (0.630 vs. 0.631), and tabu search slightly exceeds it (0.641). Taken together, these comparisons suggest that the main distinction lies less in final anonymized accuracy than in the controlled contract and the associated verifier-effort profile.

Verifier effort. Final quality alone can hide a central practical constraint: how many score evaluations the verifier must perform. Table 4 reports trace-derived effort statistics under the same verifier contract. Even when matched search controls are close in final F1, they can require orders of magnitude more score evaluations than the shortlist-based proposer, which matters for scalability under fixed budgets. Concretely, S2CRA

Policy	ScoreEval/case↓	Acc./case↑	Score-rej./case↓
S2CRA	152.2	2.50	2.50
Exhaustive	2036.8	3.33	1.67
Tabu	2036.8	3.33	1.67
Beam	6789.9	3.33	0.00
Random	5.0	0.42 ± 0.21	4.58 ± 0.21

Table 4: Verifier effort under the same matched proposer/search budget.

Text regime	OE-F1↑	ΔF1	Acc./case	Score-rej./case
Anonymized	**0.631**	+0.000	2.50	2.50
Raw Names	0.612	-0.019	2.17	2.83
Raw+Desc	0.612	-0.019	2.17	2.83
Misleading	0.612	-0.019	2.17	2.83

Table 5: Controlled text-regime study under the same observational-only protocol.

performs about 152 score evaluations per case, compared to 2,037 for exhaustive/tabu and 6,790 for beam search under the same edit budget.

5.3 Text Regimes: Language Effects Under Fixed Verification

Table 5 reserves the compact main-text view for the controlled text-regime study. This experiment examines a language question under fixed statistical verification: do names and descriptions alter the proposal channel, and does the verifier contain bad priors when the text is misleading? The main-text view reports final quality together with proposal activity; richer diagnostics are deferred to the appendix. Under this controlled setup, anonymized variables perform best (0.631 F1), while adding variable names and short descriptions slightly *reduces* final oriented-edge F1 (0.612; $\Delta\text{F1} = -0.019$). The "Misleading" condition does not collapse performance under the deterministic verifier gates, suggesting that the fixed accept/reject rule can contain some semantic risk even when text is unhelpful. In this setting, the text-regime table is best read as evidence of containment: the verifier limits the effect of bad textual priors, but language does not improve graph recovery under leakage-safe evaluation.

5.4 Trajectory and Auditability Evidence

Beyond final metrics, the verifier-guided contract produces an auditable trajectory: which edits were proposed, which were rejected (and why), and how the verifier score evolved across accepted steps. Figure 5 summarizes this trajectory view, computed offline from the logged traces under the same protocol. The trajectory plots do not imply monotone improvement in downstream graph metrics; instead, they show where accepted edits changed the verifier score and where gains or regressions entered the trace.

5.5 Secondary Diagnostics

Per-network breakdowns, trace-level trajectory analyses, rerun stability, collider-oriented breakdowns, and cost summaries remain important. We report extended diagnostics, auxiliary tracks, and earlier protocol-variant evidence in Appendix A.6.

6 Discussion

Verification as a system property. A deterministic verifier is the main stabilizing component in our setting: prompting-only inference degrades sharply under anonymization, while verifier-gated search remains stable (Table 2). The resulting output includes a replayable search trace in addition to the final graph, since every proposal and verifier decision is logged.

Matched search controls. Our proposer-control baselines show that, under the same initializer pool and deterministic verifier gates, strong non-LLM atomic search policies can approach the full method on anonymized BNRepository. Concretely, exhaustive best-improvement is essentially tied with S2CRA, and tabu search slightly exceeds it under the same edit budget (Table 3). These results put more weight on the controlled contract and the verifier-effort comparison in Table 4.

Effort–quality trade-off. Matched controls can be competitive in final F1 while requiring far more score evaluations (Table 4). This matters when verification is expensive, e.g., when each score evaluation requires refitting local conditionals or recomputing sufficient statistics. In such regimes, a proposal channel that concentrates evaluation on a small shortlist can improve practical throughput even when final quality is similar.

Language as a controlled prior. Text regimes show that variable-side language can change outcomes even when verification is fixed. In our controlled fixture, anonymized variables perform best; adding names/descriptions slightly reduces mean F1, while misleading text does not collapse performance under the gates (Table 5). In this leakage-safe setting, the verifier mostly *limits* harm from

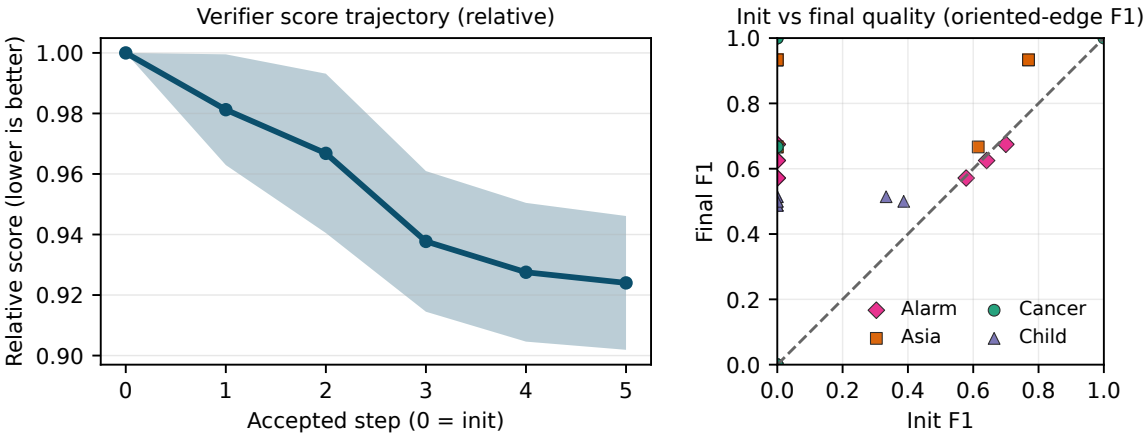

Figure 5: Offline trajectory evidence under the pre-specified observational-only protocol. Left: relative verifier score over accepted steps. Right: initial vs final oriented-edge F1 per case (points above the diagonal indicate improvement over the selected statistical initializer).

semantic priors rather than turning language into consistent gains.

When should language help? Our main track is deliberately leakage-safe and anonymized; in this regime, surface semantics can be misaligned with numerical evidence and the verifier acts mainly as a safety filter. We expect clearer gains in text-rich regimes where variable descriptions provide reliable causal cues, provided evaluation avoids metadata shortcuts. The controlled text-regime setup provides a template to study such regimes without conflating language with benchmark leakage.

Why not just run GES? GES remains a strong score-based baseline in our comparisons, but it does not address the paper's target question: how a language-conditioned proposal channel behaves under a deterministic verifier contract. Under the same observational-only benchmark, S2CRA improves oriented-edge F1 over GES (Table 2), but the more important comparison for "is language necessary" is against the matched search controls in Table 3.

7 Limitations

Proposer advantage is modest under anonymized settings. On the BN main track with anonymized variables and hidden dataset identities, the gap between the full method and matched non-LLM search controls is small. This motivates conservative claims and further work on improving the proposal policy or identifying regimes where language provides clearer leverage.

Intervention handling remains pragmatic. The main evaluation track is observational-only. When intervention samples are analyzed, we use a pragmatic concatenation protocol rather than a fully intervention-aware scoring criterion; stronger intervention-aware verification and matched baselines (e.g., GIES (Hauser and B"uhlmann, 2012)) remain future work.

Benchmark leakage remains a real threat. Even with anonymization and hidden dataset names, language-conditioned discovery can still exploit metadata or prompt artifacts if the benchmark is not carefully designed (Zhou et al., 2024; Feng et al., 2025b). For this reason, we include explicit text-regime controls instead of relying on a single prompt setting.

Scale remains limited. Our BN main track tops out at 37 nodes; extending the contract to larger graphs and richer variable types is an important next step.

Ethics Statement

Reliable causal discovery can influence downstream reasoning and decisions in high-stakes domains. S2CRA reduces one class of risk by making acceptance deterministic and auditable, but it does not turn causal discovery into a solved problem. Any graph inferred from limited or biased data should be treated as a hypothesis to review, not as an automatic decision rule.

8 Conclusion

We presented **S2CRA** as a verifier-guided proposal channel for causal graph discovery. The central idea is narrow by design: language may suggest hypotheses and atomic edits, but a determin-

istic verifier decides what survives, and the full search is logged for replay.

On BNRepository under a fixed observational-only protocol, S2CRA improves over classical baselines. Matched proposer controls show that strong non-LLM search can approach the full method in anonymized settings, but often at substantially higher verifier effort. Text regimes show that variable-side language can shift outcomes under fixed verification, yet does not reliably improve recovery under leakage-safe evaluation; deterministic gates can nonetheless contain some misleading priors.

References

Ahmed Abdulaal, Adamos Hadjivasiliou, Nina Montana-Brown, Tiantian He, Ayodeji Ijishakin, Ivana Drobnjak, Daniel C. Castro, and Daniel C. Alexander. 2024. Causal modelling agents: Causal graph discovery through synergising metadata- and data-driven reasoning. In *The Twelfth International Conference on Learning Representations*.

Taiyu Ban, Lyvzhou Chen, Xiangyu Wang, and Huanhuan Chen. 2023. From query tools to causal architects: Harnessing large language models for advanced causal discovery from data. *Preprint*, arXiv:2306.16902.

Lyuzhou Chen, Taiyu Ban, Xiangyu Wang, Derui Lyu, and Huanhuan Chen. 2023. Mitigating prior errors in causal structure learning: Towards LLM driven prior knowledge. *Preprint*, arXiv:2306.07032.

Haoang Chi, He Li, Wenjing Yang, Feng Liu, Long Lan, Xiaoguang Ren, Tongliang Liu, and Bo Han. 2024. Unveiling causal reasoning in large language models: Reality or mirage? In *Advances in Neural Information Processing Systems*.

David Maxwell Chickering. 2002. Optimal structure identification with greedy search. In *Journal of Machine Learning Research*, volume 3, pages 507–554.

Diego Colombo and Marloes H. Maathuis. 2014. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(1):3741–3782.

Gal Elidan. 2001. Bayesian network repository. https://www.cse.huji.ac.il/~galel/Repository/.

Tao Feng, Lizhen Qu, Niket Tandon, and Gholamreza Haffari. 2025a. IRIS: An iterative and integrated framework for verifiable causal discovery in the absence of tabular data. *Preprint*, arXiv:2510.09217.

Tao Feng, Lizhen Qu, Niket Tandon, Zhuang Li, Xiaoxi Kang, and Gholamreza Haffari. 2025b. On the reliability of large language models for causal discovery. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9565–9590, Vienna, Austria. Association for Computational Linguistics.

Fred Glover. 1986. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549.

Alain Hauser and Peter B"uhlmann. 2012. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13(1):2409–2464.

Zhijing Jin, Jiarui Liu, Zhiheng Lyu, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona T. Diab, and Bernhard Schölkopf. 2024. Can large language models infer causation from correlation? In *The Twelfth International Conference on Learning Representations*.

Thomas Jiralerspong, Xiaoyin Chen, Yash More, Vedant Shah, and Yoshua Bengio. 2024. Efficient causal graph discovery using large language models. In *ICLR 2024 Workshop: How Far Are We From AGI*.

Markus Kalisch and Peter B"uhlmann. 2007. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8:613–636.

Elahe Khatibi, Mahyar Abbasian, Zhongqi Yang, Iman Azimi, and Amir M. Rahmani. 2024. ALCM: Autonomous LLM-augmented causal discovery framework. *Preprint*, arXiv:2405.01744.

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. 2024. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*.

Hao Duong Le, Xin Xia, and Chen Zhang. 2024. Multi-agent causal discovery using large language models. *Preprint*, arXiv:2407.15073.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K"uttler, Mike Lewis, Wen-tau Yih, Tim Rockt"aschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*.

Stephanie Long, Alexandre Piché, Valentina Zantedeschi, Tibor Schuster, and Alexandre Drouin. 2023. Causal discovery with language models as imperfect experts. In *ICML 2023 Workshop: SPIGM (Structured Probabilistic Inference & Generative Modeling)*.

Lars Lorch, Scott Sussex, Jonas Rothfuss, Andreas Krause, and Bernhard Sch"olkopf. 2022. Amortized inference for causal structure learning. In *Advances in Neural Information Processing Systems*, volume 35.

Judea Pearl. 2009. *Causality: Models, Reasoning, and Inference*, 2 edition. Cambridge University Press, Cambridge, UK.

Bo Peng, Sirui Chen, Lei Xu, and Chaochao Lu. 2026. Causcientist: Teaching LLMs to respect data for causal discovery. *Preprint*, arXiv:2601.13614.

Amartya Roy, N Devharish, Shreya Ganguly, and Kripabandhu Ghosh. 2025. Causal-LLM: A unified one-shot framework for prompt- and data-driven causal graph discovery. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 8259–8279, Suzhou, China. Association for Computational Linguistics.

Stuart Russell and Peter Norvig. 2021. *Artificial Intelligence: A Modern Approach*, 4 edition. Pearson.

Gideon Schwarz. 1978. Estimating the dimension of a model. *The Annals of Statistics*, pages 461–464.

Marco Scutari. 2010. Learning bayesian networks with the `bnlearn` R package. *Journal of Statistical Software*, 35:1–22.

ChengAo Shen, Zhengzhang Chen, Dongsheng Luo, Dongkuan Xu, Haifeng Chen, and Jingchao Ni. 2024. Exploring multi-modal integration with tool-augmented LLM agents for precise causal discovery. *Preprint*, arXiv:2412.13667.

Peter Spirtes, Clark Glymour, and Richard Scheines. 2000. *Causation, Prediction, and Search*, 2 edition. MIT Press, Cambridge, MA.

Peter Spirtes, Christopher Meek, and Thomas Richardson. 1995. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 499–506.

Masayuki Takayama, Tadahisa Okuda, Thong Pham, Tatsuyoshi Ikenoue, Shingo Fukuma, Shohei Shimizu, and Akiyoshi Sannai. 2025. Integrating large language models in causal discovery: A statistical causal approach. *Transactions on Machine Learning Research*. Also available as arXiv:2402.01454.

Guangya Wan, Yunsheng Lu, Yuqi Wu, Mengxuan Hu, and Sheng Li. 2025. Large language models for causal discovery: Current landscape and future directions. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25), Survey Track*.

Yuzhe Zhang, Yipeng Zhang, Yidong Gan, Lina Yao, and Chen Wang. 2024. Causal graph discovery with retrieval-augmented generation based large language models. *Preprint*, arXiv:2402.15301.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. 2018. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*.

Yu Zhou, Xingyu Wu, Beicheng Huang, Jibin Wu, Liang Feng, and Kay Chen Tan. 2024. Causalbench: A comprehensive benchmark for causal learning capability of LLMs. *Preprint*, arXiv:2404.06349.

A Appendix Roadmap and Reproducibility

This appendix collects (i) reproducibility details and evidence lineage, (ii) prompt templates and structured output formats, and (iii) extended experimental evidence and diagnostics that do not fit the main-paper budget. To prevent accidental mixing across protocols, the main-paper tables are generated from one archived observational-only trace set; the appendix separates same-protocol diagnostics from earlier exploratory variants.

A.1 Roadmap and Evidence Scope

Main-paper evidence. Tables 2–5 in the main paper are generated from a single archived observational-only trace set, with protocol selection performed on a separate synthetic development suite. This is the only evidence chain used for the main headline claim.

Additional appendix evidence (earlier variants). For completeness, we also include additional tables/figures from earlier protocol variants, which explored broader settings (e.g., intervention-inclusive tracks, additional audits, auxiliary benchmarks, and case studies). These supplementary results are provided as supporting context and should not be mixed with the main headline protocol.

Navigation.

- Appendix A.2: evidence lineage and reproduction modes.
- Appendix A.3 and Appendix A.4: protocol selection, proposer tuning, and additional diagnostics under the fixed observational-only BN-Repository protocol.
- Appendix A.5: prompts, structured schemas, and text-regime fixtures.
- Appendix A.6: historical/auxiliary evidence (not used for the main claim).

A.2 Reproducibility Details

Released logs and lineage. We organize the evidence chain as: *development-grid evaluations* → *fixed protocol selection* → *observational-only suite* → *archived traces* → *paper tables/figures*. Each evaluation stores structured metrics together

with the configuration used (model, text regime, move policy, gate settings, and a code version identifier). Main-text tables/figures are rendered from these archived traces to avoid transcription errors.

Protocol lock (BNRepository). The main claim uses observational-only BNRepository under a fixed verifier contract. Protocol selection is performed on a separate synthetic development suite, after which the chosen configuration is transferred to the BN main track with zero retuning. At runtime, the accept/reject loop accesses only the current graph, the data-derived verifier score, and hard structural constraints; ground-truth graphs are used strictly for offline evaluation metrics.

BNRepository network statistics. Table 6 summarizes the four discrete networks used in the observational-only main track.

Network	Nodes d	Edges $\|E\|$
Cancer	5	4
Asia	8	8
Child	20	25
Alarm	37	46

Table 6: BNRepository network statistics used in the observational-only main track.

Reproducibility modes. We support two modes. **Offline regeneration** re-renders paper tables/figures from the saved logs without any new model calls. **End-to-end reruns** reproduce the full procedure by re-running the development-suite grid, selecting a protocol, executing the observational-only suite, and then exporting paper tables/figures (which requires model access for S2CRA and LLM-zero). A step-by-step reproduction checklist is provided in the supplementary materials.

Trace record format. For each BN case, the per-run results include the full S2CRA search trace. The trace records (i) which initializer provided the starting hypothesis, (ii) accepted and rejected edits with rejection categories, (iii) per-step score deltas, and (iv) the stopping reason. These records enable offline replay and trajectory plots without rerunning any LLM calls. Concretely, the per-case records include the initializer choice, score history, accepted edits (with score deltas), rejected proposals (with deterministic rejection reasons), and a terminal stopping reason.

A.3 Protocol Selection and Proposer Tuning (Supplementary)

A.3.1 Synthetic Dev-Suite Protocol Ranking

Table 7 reports the complete protocol ranking on the synthetic development suite. This split is intentional: protocol selection is performed off-benchmark, and the chosen configuration is transferred to the BNRepository main track with zero retuning.

A.3.2 Fixed Main-Track Protocol

Table 8 records the locked observational-only configuration used for every main-text result. It makes the transfer boundary explicit: development-suite selection happens once, after which the BNRepository benchmark is executed without retuning.

Component	Setting
Benchmark	BNRepository (Cancer/Asia/Child/Alarm); seeds=42,43,44
Data regime	Observational only; dataset identity hidden
Text regime	Anonymized variables (main track)
Initializer	Statistical pool: PC, FCI, GES; choose G_0 by verifier score
Move set	`ADD`/`DEL`/`REV`
Hard gate	Acyclic + max in-degree ($d_{\max} = 3$)
Score gate	BIC + $\lambda\|E\|$ ($\lambda = 2.0$)
Stopping	Budget $T = 5$; patience $k = 5$; $\epsilon = 1 \times 10^{-6}$
Proposer	Shortlist-ranker (size=36; rank-k=36; hints=union-of-initializers); no-repeat
Trace	Log initializer, proposals, reject reasons, score deltas, stop reason

Table 8: Fixed protocol summary for the observational-only main track.

A.3.3 Proposer-Tuning Screen (Asia + Child)

We summarize the proposer variants screened on `asia` and `child` (seeds 42/43/44), under the fixed observational-only verifier contract. These results document the proposer iteration that informed the final configuration; they are supplementary and are not part of the main headline table.

A.4 Additional Diagnostics (Observational-Only Main Protocol)

This section collects same-protocol diagnostics that support the observational-only headline table while remaining outside the main-paper page budget.

Rank	$d_{\max}$	λ	Budget	Dev F1↑	Dev SHD↓	Dir-Acc↑
1	**3**	**2**	**5**	**0.660**	**12.812**	**0.757**
2	2	2	5	0.658	12.688	0.772
3	2	5	5	0.656	12.750	0.772
4	2	0	5	0.652	12.625	0.765
5	3	5	3	0.652	12.938	0.754
6	3	0	3	0.649	12.750	0.756
7	3	0	5	0.648	13.062	0.744
8	3	5	5	0.647	12.938	0.746
9	2	2	3	0.646	12.750	0.759
10	2	5	3	0.646	12.750	0.759
11	2	0	3	0.638	12.812	0.756
12	3	2	3	0.633	12.938	0.740

Table 7: Synthetic development-suite protocol ranking. The selected configuration is the rank-1 entry. Selection rule: higher oriented-edge F1 -> lower SHD -> lower budget -> lower d_max.

Variant	Comb. F1↑	Asia	Child	Acc./case	ScoreEval/case	Repeat-rej./case	MaxInDeg-rej./case
Free proposer (top-1) (20260311_phase1_gpt52)	0.554	0.718	0.390	0.17	4.7	0.00	0.33
Free proposer (top-3) (20260309_0940)	0.532	0.667	0.398	0.33	14.7	0.00	0.33
Free proposer (top-3) (20260311_phase1_gpt52)	0.534	0.667	0.402	0.50	14.7	0.00	0.17
Free proposer (top-3) + no-repeat (20260309_0940)	0.571	0.747	0.394	0.83	14.7	0.17	0.17
Free proposer (top-3) + no-repeat (20260311_phase1_gpt52)	0.534	0.667	0.402	0.50	14.5	0.17	0.00
Free proposer (top-5) + no-repeat + oracle (20260311_phase1_gpt52)	0.549	0.700	0.398	0.50	23.7	0.33	1.00
Free proposer (top-3) + structure (20260311_phase2_gpt52)	0.530	0.667	0.394	0.17	14.0	0.33	0.00
Free proposer (top-3) + structure + hints (20260309_0950)	0.556	0.718	0.394	0.33	14.8	0.17	0.00
Free proposer (top-3) + structure + hints (20260309_1004)	0.556	0.684	0.429	2.17	15.0	0.00	0.00
Free proposer (top-3) + structure + hints (20260311_phase2_gpt52)	0.536	0.639	0.434	2.00	14.8	0.00	0.00
Free proposer (3 calls) + structure + hints (20260309_1004)	0.521	0.654	0.388	1.83	8.7	0.33	0.00
Free proposer (3 calls) + structure + hints (20260311_phase2_gpt52)	0.503	0.621	0.384	2.00	8.5	0.50	0.00
Shortlist-ranker (24→5) (20260309_1017)	0.614	0.717	0.512	2.83	25.0	0.00	0.00
Shortlist-ranker (24→5) (20260311_phase3_gpt52)	0.607	0.684	0.531	2.67	25.0	0.00	0.00
Shortlist-ranker (36→8) (20260309)	0.676	0.794	0.559	3.17	40.0	0.00	0.00
Shortlist-ranker (36→8) (20260311_phase3_gpt52)	0.642	0.745	0.538	3.00	40.0	0.00	0.00
Shortlist-ranker (36→16; v2) (20260311_phase4_gpt52)	0.648	0.727	0.569	3.00	71.0	0.00	0.00
Shortlist-ranker (36→36; v2) (20260311_phase4b_gpt52)	0.681	0.844	0.519	3.33	179.0	0.00	0.00
Shortlist-ranker (36→36; v2 add/rev) (20260311_phase4c_gpt52)	**0.681**	**0.844**	**0.519**	**3.33**	**177.7**	**0.00**	**0.00**
Reference (non-LLM upper bound under the same contract)							
Exhaustive (atomic)	0.696	0.844	0.547	3.50	1044.0	0.00	0.00

Table 9: Proposer-tuning screen on BNRepository (asia+child, seeds 42/43/44) under the fixed obs-only verifier contract. The final selected configuration is bolded. Acc./case counts accepted edits; ScoreEval/case counts verifier score evaluations; batch rejects record deterministic pre-score rejections (e.g., repeats).

A.4.1 Per-Network Breakdown

Table 10 reports the per-network oriented-edge F1 values with SHD in parentheses under the fixed observational-only protocol. It makes the heterogeneity behind the main mean explicit and shows where the aggregate gain is concentrated.

A.4.2 Worked Examples (BNRepository)

How to read the trace excerpts. Each case stores (i) the selected initializer, (ii) accepted edits (with score deltas), (iii) rejected edits with deterministic reject reasons, and (iv) the stopping condition. Below we show ten short per-case excerpts from the archived traces, selected deterministically to cover: large improvements, modest

Method	Cancer	Asia	Child	Alarm
LLM-based				
S2CRA	**0.556** (2.7)	0.844 (2.3)	0.501 (18.7)	0.624 (31.0)
LLM-zero	0.274 (5.3)	0.000 (8.0)	0.000 (25.3)	0.000 (46.0)
Matched atomic-search controls				
Exhaustive	0.508 (3.0)	0.844 (2.3)	**0.547** (17.0)	0.619 (30.0)
Tabu	0.508 (3.0)	**0.889** (1.7)	**0.547** (17.0)	0.619 (30.0)
Beam	0.508 (3.0)	**0.889** (1.7)	0.457 (20.3)	0.627 (29.3)
Random	0.471 (2.9)	0.700 (4.0)	0.407 (20.0)	**0.640** (29.0)
Classical baselines				
GES	0.333 (2.7)	0.718 (3.7)	0.143 (23.0)	**0.640** (29.0)
PC-Stable	0.373 (4.3)	0.111 (8.0)	0.424 (21.7)	0.239 (41.3)
FCI	0.000 (4.0)	0.000 (8.0)	0.402 (18.7)	0.000 (46.0)
NOTEARS	0.000 (4.0)	0.148 (7.7)	0.227 (31.7)	0.222 (51.3)

Table 10: Per-network BNRepository breakdown under the fixed observational-only protocol. Cells show oriented-edge F1, with SHD in parentheses.

gains with verifier rejections, containment cases (no accepted edits when the initializer is already optimal), and counterexamples (a severe failure plus mild regressions or wasted proposals). The goal is to reflect the range of behaviors seen in the audited runs rather than to showcase only best-case trajectories.

Large improvements.

Cancer (n=5; seed=44) (dF1=+0.67) **LARGE IMPROVEMENT**

Init: `fci` (OE-F1=0.000, SHD=4) **Final:** OE-F1=0.667, SHD=2
Edges: 0→2; **Acc/Rej:** 2/3 **Stop:** `max-iterations`

`summary: edges 0->2; acc: ADD=2 DEL=0 REV=0; rej(top2): boundary: no clear gain=2, score: no improve=1.`

Proposer channel.

```
> first accepted edit: it=1 ADD V3->V4; dScore=+14.43; ACCEPT (score)
> proposal policy: mode=shortlist-rank; candidates=20; scored=20
> init hints: hint_edges=4; e.g., V1->V3, V2->V3, V4->V3
> shortlist (top-3): #4 ADD V3->V4 (selected), #3 ADD V1->V3, #6 ADD V2->V3
```

Accepted edits (excerpt).

```
+ it=1 ADD V3->V4          +14.43
+ it=2 ADD V3->V5           +5.29
```

Rejected edits (excerpt).

```
- it=3 REV V3->V4    boundary: no clear gain    +0.00
- it=4 REV V3->V5    boundary: no clear gain    +0.00
- it=5 ADD V1->V3    score: no improve          -2.67
```

Child (n=20; seed=43) **IMPROVES (dF1=+0.18)**

Init: `fci` (OE-F1=0.333, SHD=20) **Final:** OE-F1=0.514, SHD=17
Edges: 5→10; **Acc/Rej:** 5/0 **Stop:** `max-iterations`

`summary: edges 5->10; acc: ADD=5 DEL=0 REV=0; rej(top2): none.`

Proposer channel.

```
> first accepted edit: it=1 ADD V12->V19; dScore=+136.29; ACCEPT (score)
> proposal policy: mode=shortlist-rank; candidates=36; scored=36
> init hints: hint_edges=12; e.g., V12->V16, V12->V17, V14->V12
> shortlist (top-3): #5 ADD V12->V19 (selected), #1 ADD V14->V12, #10 ADD V15->V12
```

Accepted edits (excerpt).

```
+ it=1 ADD V12->V19         +136.29
+ it=2 ADD V19->V5          +129.20
+ it=3 ADD V5->V18          +105.05
+ it=4 ADD V12->V15          +79.49
```

Asia (n=8; seed=42) **IMPROVES (dF1=+0.16)**

Init: `ges` (OE-F1=0.769, SHD=3) **Final:** OE-F1=0.933, SHD=1
Edges: 5→7; **Acc/Rej:** 2/3 **Stop:** `max-iterations`

`summary: edges 5->7; acc: ADD=2 DEL=0 REV=0; rej(top2): score: no improve=2, boundary: no clear gain=1.`

Proposer channel.

```
> first accepted edit: it=1 ADD V3->V5; dScore=+47.18; ACCEPT (score)
> proposal policy: mode=shortlist-rank; candidates=36; scored=36
> init hints: hint_edges=0
> shortlist (top-3): #21 ADD V3->V5 (selected), #22 ADD V5->V6, #14 ADD V3->V6
```

Accepted edits (excerpt).

```
+ it=1 ADD V3->V5           +47.18
+ it=2 ADD V3->V4           +13.02
```

Rejected edits (excerpt).

```
- it=3 REV V3->V4      boundary:  +0.00
                       no clear
                       gain
- it=4 ADD V1->V3      score:  no  -7.50
                       improve
- it=5 ADD V3->V1      score:  no  -7.50
                       improve
```

Asia (n=8; seed=44) **IMPROVES (dF1=+0.16)**

Init: `ges` (OE-F1=0.769, SHD=3) **Final:** OE-F1=0.933, SHD=1

Edges: 5→7; **Acc/Rej:** 2/3 **Stop:** `max-iterations`

`summary: edges 5->7; acc: ADD=2 DEL=0 REV=0; rej(top2): score: no improve=2, boundary: no clear gain=1.`

Proposer channel.

```
> first accepted edit:  it=1 ADD
V3->V5; dScore=+54.24; ACCEPT
(score)
> proposal policy:
mode=shortlist-rank; candidates=36;
scored=36
> init hints:  hint_edges=0
> shortlist (top-3):  #21 ADD V3->V5
(selected), #36 REV V4->V6, #30 REV
V2->V6
```

Accepted edits (excerpt).

```
+ it=1 ADD V3->V5                +54.24
+ it=2 ADD V3->V4                +23.96
```

Rejected edits (excerpt).

```
- it=3 REV V3->V4      boundary:  +0.00
                       no clear
                       gain
- it=4 ADD V3->V1      score:  no  -6.24
                       improve
- it=5 ADD V1->V3      score:  no  -6.24
                       improve
```

Containment and modest gains.

Cancer (n=5; seed=42) **CONTAINMENT (acc=0; dF1=+0.00)**

Init: `ges` (OE-F1=1.000, SHD=0) **Final:** OE-F1=1.000, SHD=0

Edges: 4→4; **Acc/Rej:** 0/5 **Stop:** `early-stop (patience)`

`summary: edges 4->4; acc: ADD=0 DEL=0 REV=0; rej(top2): score: no improve=5.`

Rejected edits (excerpt).

```
- it=1 REV V1->V3      score:  no  -3.03
                       improve
- it=2 REV V2->V3      score:  no  -3.03
                       improve
- it=3 ADD V2->V1      score:  no  -8.37
                       improve
- it=4 ADD V1->V2      score:  no  -8.37
                       improve
```

Child (n=20; seed=42) **MODEST GAIN (rej=2; dF1=+0.03)**

Init: `pc` (OE-F1=0.462, SHD=21) **Final:** OE-F1=0.488, SHD=21

Edges: 14→16; **Acc/Rej:** 3/2 **Stop:** `max-iterations`

`summary: edges 14->16; acc: ADD=2 DEL=0 REV=1; rej(top2): score: no improve=2.`

Proposer channel.

```
> first accepted edit:  it=1 REV
V8->V3; dScore=+213.64; ACCEPT
(score)
> proposal policy:
mode=shortlist-rank; candidates=36;
scored=36
> init hints:  hint_edges=8; e.g.,
V12->V16, V12->V17, V12->V19
> shortlist (top-3):  #34 REV V8->V3
(selected), #4 ADD V1->V12, #36 ADD
V10->V5
```

Accepted edits (excerpt).

```
+ it=1 REV V8->V3                +213.64
+ it=2 ADD V10->V4               +104.51
+ it=5 ADD V13->V18               +20.42
```

Rejected edits (excerpt).

```
- it=3 ADD V10->V13    score:  no  -1.97
                       improve
- it=4 ADD V10->V18    score:  no  -4.47
                       improve
```

Failures and regressions.

Cancer (n=5; seed=43) **FAILS (dF1=-0.33)**

Init: `pc` (OE-F1=0.333, SHD=4) **Final:** OE-F1=0.000, SHD=6

Edges: 2→2; **Acc/Rej:** 1/4 **Stop:** `max-iterations`

`summary: edges 2->2; acc: ADD=0 DEL=0 REV=1; rej(top2): score: no improve=3, boundary: no clear gain=1.`

Proposer channel.

```
> first accepted edit:  it=1 REV
V2->V3; dScore=+5.25; ACCEPT (score)
> proposal policy:
mode=shortlist-rank; candidates=18;
scored=18
> init hints:  hint_edges=0
> shortlist (top-3):  #13 REV V2->V3
(selected), #3 ADD V1->V3, #8 ADD
V5->V3
```

Accepted edits (excerpt).

```
+ it=1 REV V2->V3                 +5.25
```

Rejected edits (excerpt).

```
- it=2 REV V4->V3      boundary:   +0.00
                       no clear
                       gain
- it=3 ADD V1->V4      score:  no  -2.32
                       improve
- it=4 ADD V4->V1      score:  no  -2.32
                       improve
- it=5 REV V3->V2      score:  no  -5.25
                       improve
```

Alarm (n=37; seed=43) (dF1=-0.03) **SLIGHT REGRESSION**

Init: ges (OE-F1=0.700, SHD=24) **Final:** OE-F1=0.675, SHD=27
Edges: 34→37; **Acc/Rej:** 5/0 **Stop:** max-iterations

summary: edges 34->37; acc: ADD=3 DEL=0 REV=2; rej(top2): none.

Proposer channel.

> first accepted edit: it=1 REV V16->V33; dScore=+372.95; ACCEPT (score)
> proposal policy: mode=shortlist-rank; candidates=36; scored=36
> init hints: hint_edges=4; e.g., V25->V32, V27->V30, V29->V30
> shortlist (top-3): #27 REV V16->V33 (selected), #19 REV V17->V26, #1 ADD V10->V1

Accepted edits (excerpt).

```
+ it=1 REV V16->V33          +372.95
+ it=2 REV V17->V26          +157.88
+ it=3 ADD V1->V6             +21.42
+ it=4 ADD V1->V5             +13.36
```

Alarm (n=37; seed=44) (dF1=-0.02) **SLIGHT REGRESSION**

Init: ges (OE-F1=0.641, SHD=28) **Final:** OE-F1=0.625, SHD=30
Edges: 32→34; **Acc/Rej:** 2/3 **Stop:** max-iterations

summary: edges 32->34; acc: ADD=2 DEL=0 REV=0; rej(top2): score: no improve=3.

Proposer channel.

> first accepted edit: it=1 ADD V1->V6; dScore=+21.75; ACCEPT (score)
> proposal policy: mode=shortlist-rank; candidates=36; scored=36
> init hints: hint_edges=8; e.g., V27->V30, V29->V30, V34->V15
> shortlist (top-3): #34 ADD V1->V6 (selected), #36 ADD V1->V8, #32 ADD V1->V4

Accepted edits (excerpt).

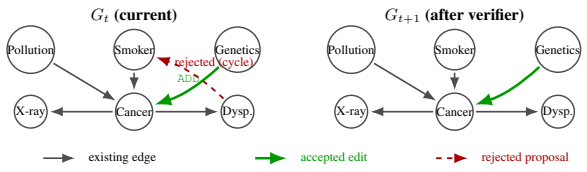

Figure 6: Illustrative atomic-edit step under the propose-and-verify contract. The proposer may suggest edits, but the verifier deterministically accepts only hard-gate-valid, score-improving moves; rejected proposals are logged for replay.

```
+ it=1 ADD V1->V6             +21.75
+ it=2 ADD V1->V14             +1.06
```

Rejected edits (excerpt).

```
- it=3 ADD V1->V23    score:  no  -2.00
                      improve
- it=4 ADD V10->V23   score:  no  -2.00
                      improve
- it=5 ADD V11->V23   score:  no  -2.00
                      improve
```

Alarm (n=37; seed=42) WASTED PROPOSALS (rej=4; dF1=-0.01)

Init: ges (OE-F1=0.578, SHD=35) **Final:** OE-F1=0.571, SHD=36
Edges: 37→38; **Acc/Rej:** 1/4 **Stop:** max-iterations

summary: edges 37->38; acc: ADD=1 DEL=0 REV=0; rej(top2): score: no improve=4.

Proposer channel.

> first accepted edit: it=1 ADD V16->V31; dScore=+7.75; ACCEPT (score)
> proposal policy: mode=shortlist-rank; candidates=36; scored=36
> init hints: hint_edges=14; e.g., V11->V12, V13->V23, V24->V23
> shortlist (top-3): #5 ADD V16->V31 (selected), #1 ADD V23->V13, #4 ADD V23->V24

Accepted edits (excerpt).

```
+ it=1 ADD V16->V31            +7.75
```

Rejected edits (excerpt).

```
- it=2 ADD V1->V14    score:  no  -2.00
                      improve
- it=3 ADD V10->V14   score:  no  -2.00
                      improve
- it=4 ADD V11->V14   score:  no  -2.00
                      improve
- it=5 ADD V12->V14   score:  no  -2.00
                      improve
```

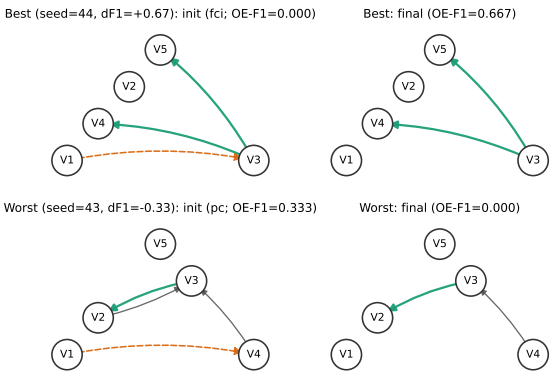

Figure 7: Real case visualization from an archived BN-Repository trace (Cancer network; anonymized variables). Gray edges denote the initializer graph. Green edges denote accepted edits (i.e., edges in the final graph but not in the initializer). Red dashed edges show one rejected proposal, illustrating deterministic containment by the verifier gates.

A.5 Prompt Templates and Structured Schemas

All prompts are designed to be **benchmark-safe**: they support variable anonymization and avoid dataset-name leakage.

A.5.1 Graph Initialization Prompt (Optional)

Prompt (template).

```
System: You are a causal discovery
scientist. Propose a DAG as a
hypothesis.
Verifier: Your proposal will be
verified against data.

User: Variables V = {X_1, ..., X_d} and
optional text hints.
Output: JSON object matching the
output schema below.
```

Inputs.

- Variable list (optionally anonymized): $V = \{X_1, \ldots, X_d\}$.
- Optional text hints (may be empty or sanitized).

Output schema (minimal).

Field	Type	Req	Constraints / Notes
`nodes`	list of strings	Y	Must match the provided variable set V.
`edges`	list of (src, dst) pairs	Y	Directed pairs; must form a DAG (no cycles, no self-loops).
`rationale`	string	N	Brief justification for auditability (optional).

A.5.2 Atomic Edit Proposal Prompt (Free Batch / Multi-call)

Prompt (template).

```
System: You are a causal refinement
agent. Propose atomic edit(s).
Verifier: The verifier will select
among candidates under hard
constraints and a fixed score gate.

User: Current graph (nodes, edges),
current score, and error-memory
summary.
Output: JSON object matching the
output schema below.
```

Output schema (single edit).

Field	Type	Req	Constraints / Notes
`operation`	string	Y	One of `ADD`, `DEL`, `REV`, `STOP`.
`source`	string	C	Required unless `STOP`; must be a valid node token.
`target`	string	C	Required unless `STOP`; must be a valid node token.
`justification`	string	N	Short reason (optional).
`confidence`	float	N	Optional confidence in $[0, 1]$.

Output schema (ranked batch).

Field	Type	Req	Constraints / Notes
`edits`	list of objects	Y	Each item follows the single-edit schema above.

A.5.3 Shortlist-Ranker Prompt (Hint-Diverse Candidate List)

Prompt (template).

```
System: You are a causal refinement
agent. Rank candidate edits.
Constraint: You must choose among
candidate ids shown in the shortlist.

User: Current graph (nodes, edges),
current score, error-memory summary,
optional text, and a numbered
shortlist produced by a deterministic
legal-edit enumerator.
Output: JSON object matching the
output schema below.
```

Output schema (ranked ids).

Field	Type	Req	Constraints / Notes
`ranked_ids`	list of ints	Y	Distinct ids, best-first; only ids present in the short-list.
`reasons`	dict	N	Optional short reasons keyed by id (string key).

A.5.4 Text-Regime Fixtures (ACL-Relevance Controls)

Regimes. We use four fixed text regimes that modify *only* variable text while keeping dataset identity hidden: `Anonymized`, `RawNames`, `RawPlusDesc`, and `MisleadingDesc`. All descriptions are hand-written fixtures tracked as versioned artifacts to avoid runtime contamination.

Fixture path. The regimes are declared in `config/bn_text_regimes.json`.

Example fixture. Table 11 shows an excerpt of the hand-written short vs misleading descriptions used to vary only variable-side language while keeping dataset identity hidden.

Variable	Short description	Misleading description
`Smoker`	smoking status	diagnostic finding produced by X-ray screening
`Cancer`	lung cancer status	preventive behavior that reduces respiratory risk
`Xray`	X-ray test result	medical intervention that directly changes whether cancer exists

Table 11: Example text fixtures used in the controlled regime study (Cancer network).

A.6 Extended Experimental Evidence (Earlier Variants)

This section aggregates additional experiments, diagnostics, and case studies produced during earlier iterations. They are included to maximize auditability and provide a complete supplementary record.

A.6.1 BNRepository: Legacy Main Table (Hard-Gate Protocol)

Model	Method	SHD ↓	Precision ↑	Recall ↑	F1 ↑	Direction-Acc ↑
GPT-5.2	**S2CRA (hard-gate)**	14.333	0.748	0.532	0.607	0.858
GPT-5.2	PC-Stable	18.833	0.516	0.220	0.287	0.525
GPT-5.2	FCI	19.167	0.250	0.063	0.100	0.250
GPT-5.2	GES (BDeu)	14.583	0.671	0.389	0.458	0.724
GPT-5.2	NOTEARS	23.667	0.332	0.107	0.149	0.416
GPT-5.2	LLM-zero	20.917	0.021	0.021	0.021	0.083
Gemini	**S2CRA (hard-gate)**	14.333	0.788	0.522	0.611	0.907
Gemini	PC-Stable	18.833	0.516	0.220	0.287	0.525
Gemini	FCI	19.167	0.250	0.063	0.100	0.250
Gemini	GES (BDeu)	14.583	0.671	0.389	0.458	0.724
Gemini	NOTEARS	23.667	0.332	0.107	0.149	0.416
Gemini	LLM-zero	27.583	0.303	0.294	0.293	0.780

Table 12: BNRepository main table (hard-gate protocol). Values are averaged over the four BN networks and three random seeds.

A.6.2 BNRepository: Hybrid Initialization and Refinement Deltas

To make hybrid initialization auditable, we summarize (i) which baseline graph was selected as the initial hypothesis and (ii) how much the refinement loop improves structure metrics from init to final.

Model	Init-source counts	$P(\Delta F1 > 0)$	$P(\Delta F1 < 0)$	mean $\Delta F1$	mean ΔSHD
Gemini	ges=7, pc=2, fci=2, notears=1	0.333	0.167	0.102	0.083
GPT-5.2	ges=7, pc=5, fci=0, notears=0	0.167	0.250	0.009	-0.083

Table 13: Hybrid init audit on BNRepository: init-source distribution and refinement deltas. Positive ΔSHD indicates improvement (lower SHD).

A.6.3 BNRepository: Legacy Trajectory View (Main Track)

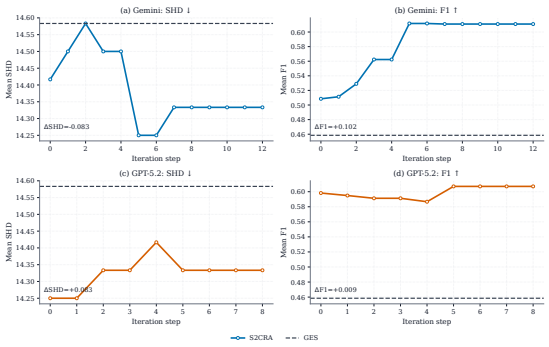

Figure 8: Trajectory evidence on BNRepository (earlier protocol variant). The plot summarizes how final structure metrics evolve relative to the selected initializer across cases.

A.6.4 BNRepository: Per-Network Breakdown

To preempt reviewer concerns that mean scores could be dominated by a single network, we report per-network metrics for three key directed methods. Each cell reports directed F1, with SHD in parentheses.

Model	Network	**S2CRA**	GES (BDeu)	LLM-zero
Gemini	Cancer	0.71 (2.0)	0.33 (2.7)	0.44 (5.0)
Gemini	Asia	0.75 (3.3)	0.72 (3.7)	0.42 (9.3)
Gemini	Child	0.35 (23.0)	0.14 (23.0)	0.01 (45.3)
Gemini	Alarm	0.64 (29.0)	0.64 (29.0)	0.30 (50.7)
GPT-5.2	Cancer	0.68 (2.3)	0.33 (2.7)	0.08 (4.7)
GPT-5.2	Asia	0.68 (4.3)	0.72 (3.7)	0.00 (8.0)
GPT-5.2	Child	0.42 (21.7)	0.14 (23.0)	0.00 (25.0)
GPT-5.2	Alarm	0.64 (29.0)	0.64 (29.0)	0.00 (46.0)

Table 14: BNRepository per-network breakdown (directed F1 with SHD in parentheses). Values are averaged over three seeds for each network.

A.6.5 BNRepository: Run-to-Run Stability (GPT-5.2)

Setting	Runs	SHD ↓	Precision ↑	Recall ↑	F1 ↑	Direction-Acc ↑	Skel-F1 ↑
Obs-only	3	16.306 ± 0.210	0.828 ± 0.014	0.420 ± 0.006	0.527 ± 0.004	0.888 ± 0.010	0.732 ± 0.001
Main (obs+interv)	3	14.528 ± 0.210	0.718 ± 0.030	0.518 ± 0.012	0.589 ± 0.017	0.830 ± 0.024	0.755 ± 0.009

Table 15: Run-to-run stability on BNRepository for S2CRA with GPT-5.2. We rerun the full benchmark three times with identical datasets, seeds, and fixed hyperparameters; values are mean $\pm$ std across reruns.

A.6.6 BNRepository: Additional Parity Tables

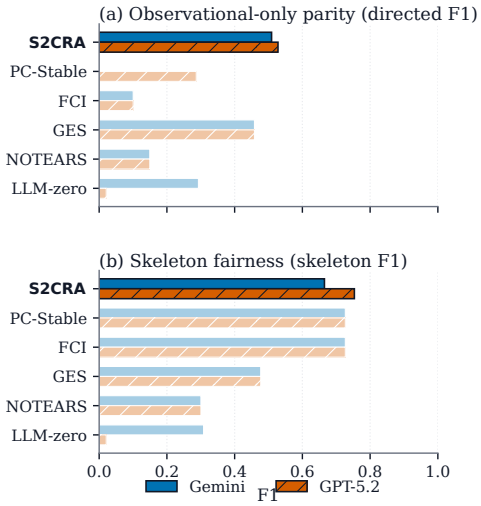

Figure 9: BNRepository parity views (earlier protocol variant). (a) Observational-only parity (directed F1) across methods. (b) CPDAG-friendly skeleton F1, giving partial credit to partially oriented outputs (PC/FCI).

Method	SHD ↓	F1 ↑	Dir-Acc ↑
GPT-5.2			
S2CRA (obs-only)	**16.500**	**0.528**	**0.878**
PC-Stable	18.833	0.287	0.525
FCI	19.167	0.100	0.250
GES (BDeu)	14.583	0.458	0.724
NOTEARS	23.667	0.149	0.416
LLM-zero	20.917	0.021	0.083
Gemini			
S2CRA (obs-only)	**14.750**	**0.509**	**0.775**
PC-Stable	20.750	0.000	0.000
FCI	19.167	0.100	0.250
GES (BDeu)	14.583	0.458	0.724
NOTEARS	23.667	0.149	0.416
LLM-zero	27.583	0.293	0.780

Table 16: Observational-only parity across methods on BNRepository. S2CRA uses the dedicated observational-only run; baselines are observational-only by default.

Model	Setting	SHD ↓	Precision ↑	Recall ↑	F1 ↑
Gemini	Main (with interventions)	14.333	0.788	0.522	0.611
Gemini	Observational-only	14.750	0.711	0.420	0.509
GPT-5.2	Main (with interventions)	14.333	0.748	0.532	0.607
GPT-5.2	Observational-only	16.500	0.813	0.424	0.528

Table 17: Observational-only fairness check for S2CRA on BNRepository (averaged over networks and seeds).

Model	Method	Skeleton SHD ↓	Skeleton F1 ↑
Gemini	S2CRA	11.500	0.665
Gemini	GES	13.083	0.477
Gemini	NOTEARS	17.167	0.300
Gemini	FCI	10.583	0.727
Gemini	PC-Stable	10.583	0.727
GPT-5.2	S2CRA	9.333	0.754
GPT-5.2	GES	13.083	0.477
GPT-5.2	NOTEARS	17.167	0.300
GPT-5.2	FCI	10.583	0.727
GPT-5.2	PC-Stable	10.583	0.727

Table 18: Skeleton-aware BN metrics (orientation-agnostic, CPDAG-friendly supplementary view).

A.6.7 BNRepository: Full Ablation Table

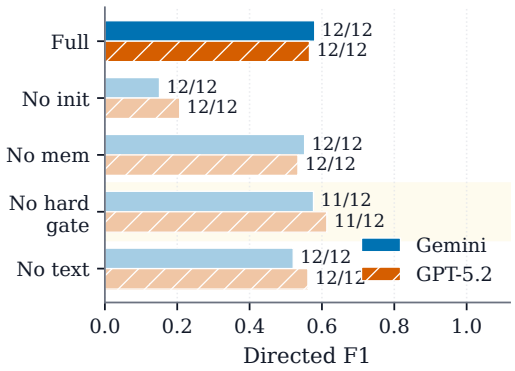

Figure 10: BNRepository ablations (earlier protocol variant). Bars show directed F1 under variants that remove or change key components.

Model	Variant	Success	SHD ↓	Precision ↑	Recall ↑	F1 ↑
GPT-5.2	**Full**	12/12	14.750	0.707	0.505	0.565
GPT-5.2	No hybrid init	12/12	22.333	0.385	0.149	0.207
GPT-5.2	No memory	12/12	15.000	0.671	0.477	0.534
GPT-5.2	No hard gate	11/12	12.636	0.772	0.549	0.614
GPT-5.2	No text hints	12/12	14.667	0.755	0.481	0.561
Gemini	**Full**	12/12	14.667	0.784	0.491	0.580
Gemini	No hybrid init	12/12	22.917	0.298	0.112	0.151
Gemini	No memory	12/12	14.500	0.715	0.470	0.552
Gemini	No hard gate	11/12	13.545	0.767	0.486	0.577
Gemini	No text hints	12/12	15.083	0.670	0.449	0.521

Table 19: BNRepository ablations (S2CRA). Removing the hard gate is diagnostic only; the hard-gate full variant remains the primary protocol.

A.6.8 BNRepository: Score-Robustness Tables

Verifier	SHD ↓	F1 ↑	DirAcc ↑
BDeu (T=2)	14.583	0.529	0.824
BIC (T=2)	15.583	**0.850**	**0.950**

Table 20: Score-gate robustness on BNRepository for Gemini. We compare two verifier score families at the same iteration budget (2). BDeu-style metrics are computed via offline trace truncation (no re-runs).

Verifier	SHD ↓	F1 ↑	DirAcc ↑
BDeu (T=2)	14.333	0.591	0.816
BIC (T=2)	22.083	**0.713**	**0.851**

Table 21: Score-gate robustness on BNRepository for GPT-5.2. We compare two verifier score families at the same iteration budget (2). BDeu-style metrics are computed via offline trace truncation (no re-runs).

A.6.9 BNRepository: Loop Audit Table

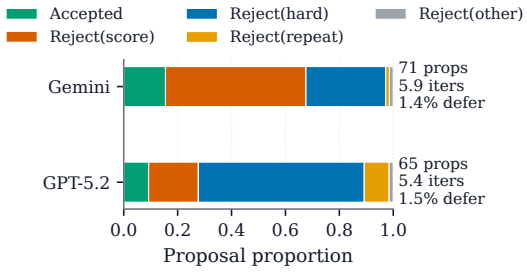

Figure 11: Gate behavior on BNRepository (earlier protocol variant). Stacked bars show the proportions of accepted edits and mutually exclusive rejection categories per proposal.

Model	Avg. iters	Proposed	Accepted	Reject(score)	Reject(hard)	Reject(repeat)	Defer
Gemini	5.92	71	11 (15.5%)	37 (52.1%)	21 (29.6%)	1 (1.4%)	1 (1.4%)
GPT-5.2	5.42	65	6 (9.2%)	12 (18.5%)	40 (61.5%)	6 (9.2%)	1 (1.5%)

Table 22: S2CRA loop audit statistics on BNRepository main runs. Counts are aggregated across networks and seeds; percentages are per proposed edit.

A.6.10 BNRepository: Loop Cost Summary

We summarize the S2CRA search cost using trace-derived quantities (iterations, proposals, accept rate) computed directly from archived traces.

Model	Iters	Props	Acc. rate	Acc. edits
Gemini	5.92	5.92	0.145	0.92
GPT-5.2	5.42	5.42	0.089	0.50

Table 23: Verifier-loop cost summary on BNRepository (from logged traces). Iterations/proposals/accept rate are computed from per-case traces.

A.6.11 BNRepository: Leakage Audit (Prompt Hygiene)

To make leakage mitigation concrete, we run a small diagnostic that toggles whether the LLM sees original variable names and/or the benchmark network name. This audit is intentionally small and does not affect the main BN claims; it serves as a sanity check that benchmark-safe prompt hygiene is not hiding an essential shortcut.

Setting	Var	Name	SHD ↓	F1 ↑	DirAcc ↑
masked+hidden	Y	Y	1.0	0.89	1.00
masked+shown	Y	N	1.0	0.89	1.00
unmasked+hidden	N	Y	0.0	1.00	1.00
unmasked+shown	N	N	0.0	1.00	1.00

Table 24: Leakage audit (prompt hygiene) on BNRepository. We toggle whether the LLM sees original variable names and/or the benchmark network name. Reported results use GPT-5.2 and dataset(s) cancer (n=1). This diagnostic is not used for the main BN table; the main protocol uses anonymized variables and hides dataset names.

A.6.12 BNRepository: Hard-Gate Hyperparameter Sensitivity

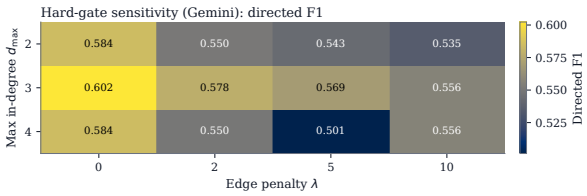

Figure 12: Hard-gate sensitivity on BNRepository (earlier protocol variant). Cells show directed F1 for each $(d_{\max}, \lambda)$ pair under the fixed protocol, averaged over four networks and three random seeds.

A.6.13 BNRepository: Intervention Micro-Analysis

Network	Main F1 ↑	Obs-only F1 ↑	ΔF1
Cancer	0.709	0.333	0.376
Asia	0.747	0.751	-0.004
Child	0.348	0.425	-0.078
Alarm	0.640	0.527	0.113

Table 25: Within-S2CRA intervention effect on BNRepository for Gemini (directed F1). Main uses observational+interventional tables; obs-only rerun removes intervention tables.

Network	Main F1 ↑	Obs-only F1 ↑	ΔF1
Cancer	0.679	0.667	0.013
Asia	0.685	0.718	-0.033
Child	0.424	0.390	0.034
Alarm	0.640	0.337	0.303

Table 26: Within-S2CRA intervention effect on BNRepository for GPT-5.2 (directed F1). Main uses observational+interventional tables; obs-only rerun removes intervention tables.

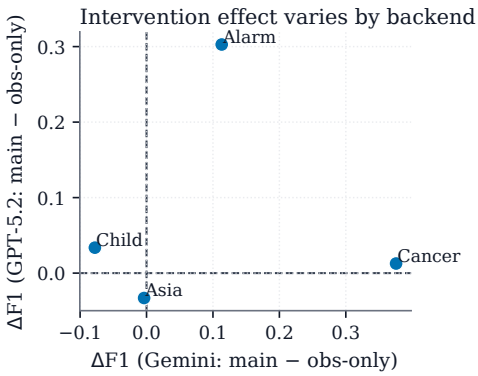

Figure 13: Per-network intervention effect for S2CRA (earlier protocol variant): each point is one BN network, with x-axis showing ΔF1 (Gemini: main − observational-only) and y-axis showing ΔF1 (GPT-5.2: main − observational-only).

A.6.14 Supporting Graph Benchmark: SynthRisk

We report structural recovery on the mechanism-grounded SynthRisk benchmark (100 synthetic SCM scenarios). Across both LLM backends, the verifier-guided agent matches the LLM-only baseline in edge-F1 while improving SHD, consistent with the claim that a closed-loop verifier can prune hallucinated edges without collapsing recall.

Model	Method	n	SHD ↓	Precision ↑	Recall ↑	F1 ↑
Gemini	LLM-only	100	4.52	0.727	0.708	0.712
Gemini	S2CRA (agent)	100	**4.16**	0.769	0.676	0.712
Gemini	PC-Stable	100	9.27	0.191	0.143	0.157
GPT-5.2	LLM-only	100	5.04	0.682	0.761	0.710
GPT-5.2	S2CRA (agent)	100	**4.43**	0.765	0.690	0.710
GPT-5.2	PC-Stable	100	9.27	0.191	0.143	0.157

Table 27: Graph discovery on SynthRisk (100 scenarios). The agent improves SHD over LLM-only while maintaining comparable F1.

A.6.15 Supporting Tracks: Hard Cases and Tübingen

Hard Cases (adversarial traps). Table 28 reports success rates on 4 adversarial cases. Across both LLM backends, the agent improves over LLM-only prompting under our rubric, suggesting that deterministic gates reduce trap-triggered hallucinations.

Model	LLM-only ↑	S2CRA (agent) ↑	PC-Stable ↑
Gemini	0.50	0.75	1.00
GPT-5.2	0.25	0.75	1.00

Table 28: Hard Cases (4 adversarial trap cases). Values are success rates under our task rubric.

Tübingen Cause–Effect Pairs (bivariate direction). Table 29 reports direction accuracy on 108 Tübingen cause–effect pairs under a randomized A/B swap protocol. Providing the sanitized pair description is a major information channel: removing text causes the agent to abstain, while LLM-only defaults to a direction prior.

Setting	Model	Method	Dir. Acc. ↑	Weighted Acc. ↑	Abstention ↓
Text	Gemini	LLM-only	0.630	0.600	0.019
Text	Gemini	S2CRA (agent)	**0.667**	**0.615**	0.000
Text	Gemini	LiNGAM	0.398	0.351	0.324
Text	GPT-5.2	LLM-only	0.620	0.568	0.056
Text	GPT-5.2	S2CRA (agent)	**0.639**	**0.564**	0.028
Text	GPT-5.2	LiNGAM	0.398	0.351	0.324
No text	Gemini	LLM-only	0.500	0.477	0.000
No text	Gemini	S2CRA (agent)	0.000	0.000	1.000
No text	Gemini	LiNGAM	0.398	0.351	0.324
No text	GPT-5.2	LLM-only	0.407	0.428	0.250
No text	GPT-5.2	S2CRA (agent)	0.000	0.000	1.000
No text	GPT-5.2	LiNGAM	0.398	0.351	0.324

Table 29: Tübingen cause–effect pairs (108). "No text" removes pair descriptions, isolating numerical-only direction evidence.

A.6.16 Case Study: FinCausal QA Extractive Auditing

Why include a text case study? Although the main claim centers causal graph discovery, we include one minimal text case study to demonstrate a practical auditability principle: *structured outputs with explicit failure-mode counters* (e.g., empty or out-of-context answers) are easier to validate at scale than free-form generations.

Extractive QA protocol. For FinCausal QA, the system must output an exact substring of the provided context. Under this protocol, both backends achieve **empty=0** and **overflow=0** on the labeled dev splits (answers are mapped back to exact substrings). Table 30 summarizes EM and token-level F1.

Split (dev)	Gemini EM ↑	Gemini F1 ↑	GPT-5.2 EM ↑	GPT-5.2 F1 ↑
FinCausal 2025 EN	0.596	0.876	0.247	0.847
FinCausal 2025 ES	0.417	0.854	0.166	0.819
FinCausal 2026 EN	0.623	0.881	0.241	0.827
FinCausal 2026 ES	0.452	0.877	0.126	0.832

Table 30: FinCausal QA (labeled dev splits). Extractive QA with strict substring enforcement.

Structured output record (auditability). This case study uses the same principle as our graph-edit protocol: constrain outputs so that correctness checks become explicit and scalable. Each prediction record includes an extracted substring plus failure-mode counters (e.g., `empty`, `overflow`) that can be audited without subjective judging. Below is a truncated example of the saved record format (illustrative).

```
{ "answer": "...",
  "span": [123,145],
  "empty": 0, "overflow": 0 }
```

Worked examples (correct vs incorrect). To make the evaluation concrete, we include a small set of exact-match and mismatch examples from the saved prediction logs. All examples satisfy the substring constraint, so mismatches reflect reasoning errors rather than extraction overflow.

FinCausal QA — **EXACT MATCH (2025 EN)**

`meta: id=2356 | type=cause | refine=0 | in-context=True | forced-empty=False`

Context: — Missed opportunities due to failure to mobilise resources efficiently.
Q: What was the reason for the missed opportunities?
Gold: `failure to mobilise resources efficiently`
Pred: `failure to mobilise resources efficiently`

FinCausal QA — **MISMATCH (2025 EN)**

`meta: id=874 | type=effect | refine=0 | in-context=True | forced-empty=False`

Context: 2 Ms Ginman was unable to attend one meeting due to illness.
Q: What was the consequence of Ms Ginman's illness?
Gold: `Ms Ginman was unable to attend one meeting`
Pred: `was unable to attend one meeting`

FinCausal QA — **EXACT MATCH (2025 EN)**

`meta: id=1804 | type=unknown | refine=0 | in-context=True | forced-empty=False`

Context: As a result of changes to coal, iron ore, and steel price expectations, revised production volumes, and improvements to net working capital, there is an impairment revers...
Q: What accounts for the existence of an impairment reversal of US$12 million recorded in the financial statement...
Gold: `changes to coal, iron ore, and steel price expectations, revised production volumes, and improvements to net w...`
Pred: `changes to coal, iron ore, and steel price expectations, revised production volumes, and improvements to net w...`

FinCausal QA — **MISMATCH (2025 EN)**

`meta: id=5459 | type=unknown | refine=0 | in-context=True | forced-empty=False`

Context: We set out in 2017 to embed a culture of safer behaviour in the business, encouraging all our colleagues to recognise the importance of moving towards Zero Harm. This led...
Q: What does the further 25 per cent reduction in the frequency of employee lost-time injuries across the Group s...
Gold: `We set out in 2017 to embed a culture of safer behaviour in the business, encouraging all our colleagues to re...`
Pred: `We set out in 2017 to embed a culture of safer behaviour in the business, encouraging all our colleagues to re...`

FinCausal QA **EXACT MATCH (2025 EN)**
`meta: id=5171.2.b | type=effect | refine=0 | in-context=True | forced-empty=False`

Context: Farmer confidence has returned during the year as a result of the increase in farmgate milk prices and improved revenues from beef and sheep leading to a recovery in farm...
Q: What have a number of factors, including the devaluation of sterling, resulted in?
Gold: `in the short-term UK livestock and dairy prices have responded positively`
Pred: `in the short-term UK livestock and dairy prices have responded positively`

FinCausal QA **MISMATCH (2025 EN)**
`meta: id=5010 | type=effect | refine=0 | in-context=True | forced-empty=False`

Context: Scope 3 emissions account for 11,265 tCO 2e, which is a significant increase from 2015 due to including new reporting metrics such as tenant consumption and home office c...
Q: What did the inclusion of the new reporting metrics like tenant consumption and home office consumption lead t...
Gold: `Scope 3 emissions account for 11,265 tCO 2e, which is a significant increase from 2015`
Pred: `a significant increase from 2015`

FinCausal QA **EXACT MATCH (2025 ES)**
`meta: id=37 | type=unknown | refine=0 | in-context=True | forced-empty=False`

Context: Gracias a esta medida, hemos mejorado la densidad de dichos envíos en un 1,5% respecto al año anterior.
Q: £Qué ha permitido esta medida?
Gold: `hemos mejorado la densidad de dichos envíos en un 1,5% respecto al año anterior`
Pred: `hemos mejorado la densidad de dichos envíos en un 1,5% respecto al año anterior`

FinCausal QA **MISMATCH (2025 ES)**
`meta: id=100000210 | type=unknown | refine=0 | in-context=True | forced-empty=False`

Context: La eficiencia operativa mejoró un 10 % gracias a las medidas implementadas en todos los países.
Q: £Gracias a qué ha mejorado la eficiencia operativa?
Gold: `a las medidas implementadas en todos los países`
Pred: `las medidas implementadas en todos los países`

FinCausal QA **EXACT MATCH (2025 ES)**
`meta: id=347 | type=unknown | refine=0 | in-context=True | forced-empty=False`

Context: Riesgo operacional: se define como el riesgo de sufrir pérdidas debido a la inadecuación o a fallos de los procesos, las personas y los sistemas internos, o bien a causa...
Q: £A qué se debe el riesgo operacional, o riesgo de sufrir pérdidas?
Gold: `a la inadecuación o a fallos de los procesos, las personas y los sistemas internos, o bien a causa de aconteci...`
Pred: `a la inadecuación o a fallos de los procesos, las personas y los sistemas internos, o bien a causa de aconteci...`

FinCausal QA **MISMATCH (2025 ES)**
`meta: id=2734 | type=unknown | refine=0 | in-context=True | forced-empty=False`

Context: Y Línea Directa, la filial aseguradora de Automóvil y Hogar, ha tenido el mejor margen asegurador del sector del automóvil, un 12,1%, gracias a un ratio de siniestralidad...

Q: £Gracias a qué Línea Directa ha obtenido el mejor margen asegurador del sector del automóvil?

Gold: `a un ratio de siniestralidad imbatible, de un 68,6%, y un ratio de gastos operacionales de un 19,3%`

Pred: `gracias a un ratio de siniestralidad imbatible, de un 68,6%, y un ratio de gastos operacionales de un 19,3%`

982

FinCausal QA — **EXACT MATCH (2025 ES)**

`meta: id=7 | type=cause | refine=0 | in-context=True | forced-empty=False`

Context: Convivir con el peso de la deuda: mejora la del sector privado Entre el conjunto de desequilibrios financieros acumulados por la economía española, el excesivo endeudamie...

Q: £Por qué razón tiene especial relevancia el excesivo endeudamiento de las empresas y del sector público?

Gold: `por cuanto supone un freno a la recuperación de la inversión y por consiguiente a la recuperación del empleo`

Pred: `por cuanto supone un freno a la recuperación de la inversión y por consiguiente a la recuperación del empleo`

983

FinCausal QA — **MISMATCH (2025 ES)**

`meta: id=258 | type=cause | refine=0 | in-context=True | forced-empty=False`

Context: Durante el ejercicio 2016 ha continuado la mejora en el margen de intereses sobre los activos totales medios, como consecuencia de varios factores, entre los que destacan...

Q: £Cuáles han sido las causas de que durante el ejercicio 2016 haya continuado la mejora en el margen de interes...

Gold: `varios factores, entre los que destacan el incremento de los márgenes de clientes (por los menores costes de f...`

Pred: `como consecuencia de varios factores, entre los que destacan el incremento de los márgenes de clientes (por lo...`

984