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

Anonymous ACL submission

Abstract

Causal graph discovery can benefit from language knowledge, but direct LLM graph generation risks turning semantic priors into unaudited causal claims. We introduce **S2CRA**, a verifier-guided LLM proposal channel that uses an LLM only to rank legal atomic graph edits while a deterministic statistical verifier owns legality, scoring, acceptance, stopping, and trace logging. This converts language from a causal oracle into a replaceable search policy whose decisions can be replayed. On a leakage-safe observational BNRepository protocol, S2CRA improves oriented-edge recovery over GES (0.608 vs. 0.458) while using over $50\times$ fewer verifier score evaluations than strong matched atomic-search controls. Controlled text regimes, smaller-proposer checks, and alternative-score diagnostics show when language helps, harms, and remains contained by the same verifier contract. The result is a practical design point for safe LLM-assisted causal discovery: semantic proposals improve directed search, while final graph ownership stays with auditable verification.

1 Introduction

Causal graph discovery (CGD) asks for a directed graph of causal relations from data (Pearl, 2009; Spirtes et al., 2000). In this paper, the graph is a directed acyclic graph (DAG): nodes are variables and a directed edge $X \to Y$ means that X is a direct causal parent of Y in the modeled system. The task underpins scientific reasoning and decision support, yet it remains fragile. For a simple medical fragment such as *Smoking* $\to$ *Cancer* $\to$ *X-ray*, observational samples may reveal strong associations among all three variables while still leaving direction ambiguous. Different DAGs can imply the same conditional independences over observed data; this is *Markov equivalence*, a central obstacle for observational CGD.

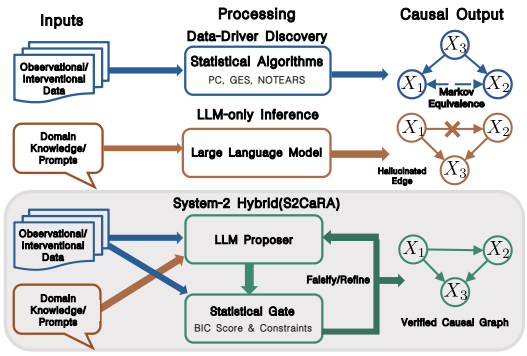

Figure 1: Conceptual contrast: statistical discovery respects observed data but faces ambiguity, while direct LLM inference offers semantic priors but can hallucinate. S2CRA restricts language to a proposal channel and uses data as a deterministic verifier.

Classical algorithms address this ambiguity in different ways. Peter–Clark (PC) and Fast Causal Inference (FCI) are constraint-based methods: they test conditional independences and orient edges when the tests justify doing so (Kalisch and B"uhlmann, 2007; Spirtes et al., 1995). Greedy Equivalence Search (GES) is a score-based search method: it searches over graph structures to optimize a decomposable data score (Chickering, 2002). NOTEARS casts DAG search as a continuous optimization problem with an explicit acyclicity constraint (Zheng et al., 2018). These methods are principled, but finite samples, Markov equivalence, and score misspecification still limit recovery.

Large language models (LLMs) sharpen this tension. Variable names and descriptions can provide useful semantic priors: in the running example, language makes *Smoking* $\to$ *Cancer* more plausible than the reverse. The same channel can also mislead, because names, descriptions, and benchmark artifacts may encode spurious cues. Direct LLM graph generation is therefore prompt-sensitive and hallucination-prone unless checked against data (Kıcıman et al., 2024; Ban et al., 2023; Jiralerspong

et al., 2024; Long et al., 2023; Zhou et al., 2024; Roy et al., 2025). Recent hybrid systems point to a more disciplined role for language: propose hypotheses that a verifier can accept or reject.

This motivates the central question of the paper: can an LLM serve as an auditable *proposal channel* inside a deterministic, edit-level search contract, while the verifier remains responsible for every graph change? In the main BN protocol, statistical initializers fix the starting graph; a deterministic enumerator creates legal atomic edit candidates; the model only ranks candidate ids; and the verifier accepts at most one hard-gate-valid, score-improving edit per iteration. Thus the model cannot invent variables, bypass acyclicity or in-degree checks, or directly write the final graph. The resulting edit trace is replayable and auditable, including rejected proposals and score deltas rather than only a final adjacency matrix. The key design move is to turn language from an untrusted causal oracle into a replaceable search proposer whose influence is measurable, stress-tested, and contained by data-derived gates.

We instantiate this idea in **S2CRA** and evaluate it under a **graph-first** protocol designed to isolate the proposal channel. Our main benchmark uses Bayesian networks (BNs), i.e., DAGs with local conditional probability tables; the source graphs are stored in Bayesian Interchange Format (BIF) files, a standard BN interchange format. The protocol uses anonymized variables, hidden dataset names, statistical initialization in the BN main track (LLM initialization disabled), and fixed hard/score gates with full trace logging. We evaluate with structural Hamming distance (SHD; the number of edge edits needed to match the target graph), Skeleton F1 (endpoint recovery after ignoring direction), and oriented-edge F1 (OE-F1; endpoint and direction recovery). Because the loop uses an observational score, our headline results are observational-only; intervention analyses are supplementary.

Our contributions are:

- **Verifier-owned LLM proposal channel:** atomic edits are proposed by language but accepted only when deterministic hard constraints and a fixed score gate agree, yielding a replayable accepted/rejected trace rather than an opaque LLM graph.
- **Matched controls and verifier-effort evidence:** non-LLM atomic-search policies under identical gates and budgets isolate proposal-policy effects; under the reported protocol, S2CRA uses 40.0 score evaluations per case versus 2,082+ for exhaustive/tabu controls.
- **Language-risk containment diagnostics:** fixed variable-text conditions, including misleading descriptions, proposal-level overlap, rejection categories, a smaller closed proposer, and BDeu-style score checks show when semantic priors help, when they hurt, and how the verifier contains them.

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. This distinction matters because graph revision and graph acceptance are different responsibilities. S2CRA assigns revision priority to the LLM ranker and acceptance authority to the verifier, so every accepted graph change passes through the same deterministic contract.

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 re-

cent 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, 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. The novelty is the locked proposal–verification contract that makes the language channel swappable, stress-testable, and comparable against matched non-LLM proposer policies while preserving deterministic acceptance.

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. Every graph emitted by the system must use exactly the variables in V; node names are validated by exact string match against this provided set. This rule is important for language models, because a fluent but renamed or invented variable would otherwise be a silent graph error.

Contract item	Deterministic rule
State	Current DAG $G_t = (V, E_t)$, verifier score, recent rejected edits
Legal edit	One `ADD`, `DEL`, or `REV` over valid nodes in V
Hard gate	Reject invalid node names, self-loops, cycles, duplicate edges, and in-degree violations
Score gate	Accept only if $\text{Score}(G_t) - \text{Score}(G') > \epsilon$
Memory	Log accepted edits, rejected edits, rejection reason, score delta, and ranked proposal ids
Stopping	Stop at budget T or after patience k with no accepted edit
Replay	Final graph is reproducible from initializer, ordered edits, gates, and score family

Table 1: Verifier contract used by S2CRA. The LLM controls only which legal edit candidates are proposed; all accept/reject decisions are deterministic.

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. This contract defines what the LLM is allowed to control: it may rank candidate edit ids, but it never changes the variable set, the score function, the hard constraints, the stopping rule, or the accept/reject decision.

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. In the main BN experiments, S_{data} is a Bayesian information criterion (BIC)-style local likelihood score for discrete conditional probability tables: lower values prefer graphs that fit the data while penalizing unnecessary parameters. Figure 3 shows the two gates that define the contract.

In the observational-only main track we instantiate the score gate with the same 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. We also support alternative discrete BN scores for robustness checks, including

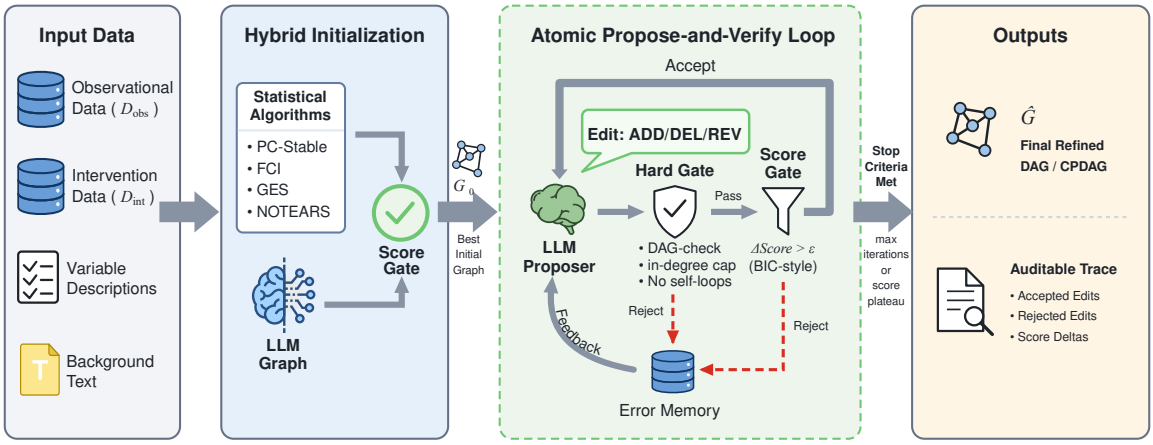

Figure 2: Overview of S2CRA. A candidate pool provides the starting graph, after which the LLM ranks a deterministic shortlist of legal atomic edit candidates. The verifier scores the proposed batch and accepts at most one edit per iteration, subject to a hard structural gate and a fixed score-improvement gate.

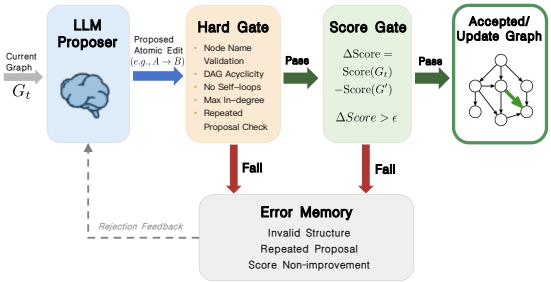

Figure 3: Verifier contract used by S2CRA. An edit must satisfy structural validity first; only then is it tested by the fixed score-improvement gate. Rejections are kept in memory for replay and prompt conditioning.

Bayesian Dirichlet equivalent uniform (BDeu) scoring, but the main accept/reject contract fixes one score family before the run.

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. The locked BN main track uses PC, FCI, GES, and NOTEARS as the initializer pool; the in-degree cap, edge penalty, iteration budget, patience, shortlist size, and ranked-candidate budget are fixed in Table 2.

Starting from G_0, the proposal channel ranks a deterministic shortlist of legal atomic candidates, each drawn from: ADD, DEL, or REV. In the reported protocol this is a shortlist-ranker rather than free-form graph generation: a deterministic enumerator creates candidate edit ids, and the LLM ranks only those ids. The proposal can be conditioned on the current graph, current score, initializer-derived hints, and a short memory of recent rejections. This design is deliberately restrictive. If a proposal names a variable outside V, repeats a blocked move, creates a cycle, exceeds the in-degree cap, or fails to improve the fixed score, it is rejected by code before it can affect the graph. 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: exact valid node names, valid source/target fields, no self-loops, acyclicity, and an optional max-in-degree cap. Only edits that satisfy

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

reach the score gate. Within each ranked batch, the verifier evaluates the legal candidates and applies the score gate to the best-scoring candidate; therefore at most one edit can be accepted in an iteration. Rejected proposals are logged with reasons such as invalid node, cycle, max-in-degree violation, repeated edit, or no score improvement. The trace also stores proposal order, accepted operation, score delta, current graph, and stopping

reason, which is why the method is auditable even when the LLM is imperfect. The same fields are used by the matched non-LLM proposer controls, so the comparison changes only the proposal policy and not the verifier.

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. Our claims therefore concern graph recovery, verifier effort, and auditability under this contract. The contract also separates two notions that are easy to conflate: accuracy of the final graph and effort required from the verifier. The latter is measured directly from the trace by counting score evaluations, accepted edits, and score-gate rejections under the shared budget.

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, (iii) a controlled text-regime study that changes only variable-side language under fixed verification, and (iv) same-protocol diagnostics for smaller proposer and score robustness. 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 Bayesian Interchange Format (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 8 lists the exact network sizes. The small graph scope is intentional for the main paper: it enables exhaustive trace audit, matched proposer comparisons under identical verifier budgets, and controlled text perturbations.

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. These development graphs are generated separately from BNRepository and are used only for protocol selection, not for reporting main-test metrics. They are separate from BNRepository test graphs: no BNRepository structure, network name, or test-case data is used when choosing the locked protocol. They use anonymous variables, a fixed topological order, bounded in-degree, randomly sampled discrete arities, and Dirichlet-sampled local conditional tables; each sample seed produces an observational table from that synthetic BN. 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. Table 2 lists the main protocol, and Appendix Table 9 gives the development-grid selection details. The reported evidence covers all four networks and seeds 42/43/44 for every required stage, and all main tables are derived from the same locked-protocol trace set.

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, NOTEARS; 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=8; hints=union-of-initializers); no-repeat
Trace	Log initializer, proposals, reject reasons, score deltas, stop reason

Table 2: Fixed protocol summary for the observational-only main track.

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 then uses a shortlist-ranker over atomic edit candidates (`ADD`/`DEL`/`REV`); the verifier scores the proposed batch and can accept at most one hard-gate-valid, score-improving edit per iteration. Any gains attributable to language occur under the same verifier contract shared with the matched non-LLM search baselines.

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. PC-Stable and FCI may output partially oriented equivalence-class graphs, whereas GES and NOTEARS return directed candidates under score or optimization assumptions. 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. Formally, for predicted edge set $\widehat{E}$ and target edge set $E^\star$, precision is $|\widehat{E} \cap E^\star|/|\widehat{E}|$, recall is $|\widehat{E} \cap E^\star|/|E^\star|$, and F1 is their harmonic mean; OE-F1 uses directed edges, while Skeleton F1 first removes edge orientation. SHD counts additions, deletions, and reversals needed to transform the prediction into the target graph.

To examine the language component directly, we include a controlled text-regime study. Under the same observational-only protocol we evaluate 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 measures how language changes final quality and verifier activity, how misleading text shifts proposals, and how the verifier contains those errors. We additionally export proposal-overlap, accepted-edit overlap, rejection categories, and ranked-candidate shifts from the same traces. For robustness, we rerun one full condition under BDeu-style scoring and rescore archived accepted edits and final graphs under BDeu without changing the original accept/reject decisions. For proposer dependence, we run a closed smaller-proposer control with GPT-5.4-mini under the same verifier contract, keeping model-family sensitivity visible without changing the verifier.

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

Method	SHD↓	Skel-F1↑	OE-F1↑
LLM-based			
S2CRA	**14.194** ± 0.048	**0.779** ± 0.004	**0.608** ± 0.003
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 3: Observational-only BNRepository results on Cancer, Asia, Child, and Alarm.

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 gives the strongest mean oriented-edge recovery in the main comparison table (Table 3). We then isolate proposer effects with matched atomic-search controls under the same verifier contract (Tables 5–6) and probe language effects with controlled text regimes that vary only variable-side language (Table 7). Same-protocol appendix diagnostics extend this evidence chain.

5.1 Observational-Only Main Results

Table 3 is the primary headline table: BNRepository under the protocol fixed on a held-out development suite and applied without retuning. S2CRA achieves 0.608 oriented-edge F1 (Skel-F1 0.779), above the strongest classical baseline in the mean table (GES, 0.458) by 0.150. This is a material directed-edge recovery gain under a locked verifier protocol. The comparison is most favorable on oriented recovery: PC-Stable and FCI have competitive skeleton scores but leave many directions unresolved, whereas S2CRA returns a directed graph through the fixed edit contract.

Method	Cancer	Asia	Child	Alarm
LLM-based				
S2CRA	**0.619 (2.3)**	0.711 (4.1)	**0.553 (16.8)**	0.551 (33.6)
Matched atomic-search controls				
Exhaustive	0.508 (3.0)	**0.889 (1.7)**	0.494 (19.0)	0.525 (34.3)
Tabu	0.508 (3.0)	**0.889 (1.7)**	0.494 (19.0)	0.525 (34.3)
Beam	0.508 (3.0)	**0.889 (1.7)**	0.457 (20.3)	0.525 (34.3)
Random	0.484 (2.8)	0.700 (4.0)	0.407 (20.0)	0.528 (34.1)
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)

Cells show oriented-edge F1 with SHD in parentheses. S2CRA aggregates three end-to-end reruns; deterministic controls are single-run unless otherwise noted. Best-per-column is bold with darker shading; second-best is lightly shaded.

Table 4: Per-network breakdown on BNRepository under the fixed observational-only protocol.

What the main gain measures. The main improvement is directed recovery under a protocol in which the model cannot directly write the graph; partially oriented outputs can look reasonable under skeleton metrics while leaving directions unresolved. Matched search policies close part of the gap under the same verifier, so the claim is a high-value proposal-channel result judged with verifier effort and trace auditability.

Per-network view. Mean scores can hide heterogeneity across networks, so Table 4 reports the full per-network breakdown in the main text. Performance varies across networks: relative to GES, S2CRA is higher in oriented-edge F1 on *Cancer* (+0.286), *Asia* (+0.066), and *Child* (+0.396), but trails on *Alarm* (-0.099). Table 3 should therefore be read as a mean result over heterogeneous cases, with *Alarm* left visible as a stress case rather than hidden by the aggregate.

5.2 Proposer Necessity Under a Fixed Verifier

Table 5 compares S2CRA against matched non-LLM proposer baselines under the same initializer pool, hard gate, score gate, move set, and budget. Exhaustive best-improvement and tabu search approach the full method and slightly exceed it on skeleton F1, but remain lower in oriented-edge F1 (0.604 vs. 0.608). This is the intended stress test: strong non-LLM search under the same verifier makes the comparison fair and prevents the language policy from receiving credit for the verifier itself. The distinction is therefore precise and positive: S2CRA reaches comparable directed graph quality while using a much smaller verifier workload and producing an auditable language-conditioned proposal trace.

Proposer	SHD↓	Skel-F1↑	OE-F1↑
S2CRA	**14.194** ± 0.048	0.779 ± 0.004	**0.608** ± 0.003
Exhaustive	14.500	**0.788**	0.604
Tabu	14.500	**0.788**	0.604
Beam	14.833	**0.788**	0.595
Random	15.217 ± 0.286	0.749 ± 0.007	0.530 ± 0.021

Table 5: Matched proposer controls under the fixed verifier contract.

Policy	ScoreEval/case↓	Acc./case↑	Score-rej./case↓
S2CRA	40.0	2.44 ± 0.17	2.56 ± 0.17
Exhaustive	2082.3	3.33	1.67
Tabu	2082.3	3.33	1.67
Beam	6943.2	3.33	0.00
Random	5.0	0.47 ± 0.16	4.53 ± 0.16

Table 6: Verifier effort under the same matched proposer/search budget.

Verifier effort. Table 6 reports trace-derived effort: S2CRA performs 40.0 score evaluations per case, compared to 2,082 for exhaustive/tabu and 6,943 for beam search under the same edit budget. Thus exhaustive and tabu spend about $52\times$ more score evaluations per case, while beam spends about $173\times$ more. Random search is cheap (5.0 evaluations per case) but much less effective (0.530 F1), showing that the result is not simply a consequence of evaluating fewer candidates. This is the strongest efficiency result: the LLM channel concentrates verifier work on high-value edits while preserving the same hard and score gates.

Audit trajectory. Archived traces record the initializer, accepted edits, rejected proposals, score deltas, and stopping reason. Appendix Figure 4 shows why we report both final metrics and replayable traces: score-improving edits need not monotonically improve offline graph quality.

5.3 Text Regimes: Language Effects Under Fixed Verification

The text-regime study in Table 7 tests whether variable-side names/descriptions alter the proposal channel under fixed statistical verification. Under this controlled setup, anonymized variables reach 0.594 F1, raw variable names increase final oriented-edge F1 to 0.618, and adding short descriptions increases it to 0.656. The "Misleading" condition reduces performance to 0.554 but does not collapse the deterministic verifier gates, suggesting that the fixed accept/reject rule can contain

Text regime	OE-F1↑	ΔF1	Acc./case	Score-rej./case
Anonymized	0.594	+0.000	2.42	2.58
Raw Names	0.618	+0.024	2.67	2.33
Raw+Desc	**0.656**	+0.062	2.83	2.17
Misleading	0.554	-0.039	2.75	2.25

Table 7: Controlled text-regime study under the same observational-only protocol.

some semantic risk even when text is unhelpful. The table is best read as containment and sensitivity evidence: language can improve or harm the proposal channel, yet all edits remain subject to the same replayable verifier contract. The exported diagnostics show zero proposal-overlap and zero accepted-edit overlap across non-anonymized regimes, and most rejected candidates fail the score gate, so harmful text shifts proposals but does not bypass verification.

5.4 Secondary Diagnostics

Appendix A.4 gives trace trajectories and same-protocol boundary checks. The smaller closed proposer remains comparable to the main proposer (0.620 vs. 0.608 OE-F1), BDeu-style rescoring gives final-direction agreement of at least 0.833 and accepted-edit agreement of 0.727–0.909 with zero robustness errors, and proposal-level text diagnostics report overlap and rejection-category shifts. Earlier protocol variants and auxiliary benchmarks are separated in Appendix A.6 and are not used for the headline claim.

6 Discussion

Verification as a system property. The evaluated object is a deterministic proposal-verification system: the model ranks candidate edits, while the verifier owns legality, scoring, acceptance, and stopping. This separation is why the result is both useful and auditable: the final graph is a replayable sequence of verifier decisions, not a fluent free-form graph answer. The contribution is a safe LLM-assisted causal discovery interface with verifier-owned graph changes.

Matched search controls and effort. The matched controls make proposal policy the object being tested: all methods share the initializer pool, atomic move set, hard gate, score gate, budget, and metrics. Exhaustive and tabu approach S2CRA and win some cells, but remain lower in mean oriented-edge F1 while using about $52\times$ more score evaluations; beam uses about $173\times$ more. The intended design point is proposal efficiency, directed recovery, and traceability under the same verifier.

Language as a controlled prior. The text-regime results show that language is active but bounded. Variable names and descriptions change the proposal distribution, and in the controlled text-regime runs raw+description fixtures improve mean oriented-edge F1 relative to anonymization. Misleading descriptions harm the proposal channel, but the deterministic gates continue to reject non-improving or boundary-violating edits. This is the desired safety shape: useful text can steer candidate selection, harmful text remains visible in rejection records, and neither case bypasses verification.

Why not just run GES or exhaustive search? GES and exhaustive/tabu are the strongest comparisons for classical search and verifier-matched policy value. S2CRA targets the setting where semantic hints prioritize a few legal edits while every decision remains replayable and score-gated.

7 Conclusion

We presented **S2CRA**, a verifier-guided LLM proposal channel for causal graph discovery. The method turns language into a replaceable ranker over legal edit ids while a deterministic verifier owns scoring, acceptance, and trace logging. Across the locked BNRepository protocol, S2CRA improves directed recovery over GES and strong matched search controls at far lower verifier effort, with text-regime and score-robustness diagnostics showing how semantic risk remains visible rather than hidden. This establishes a concrete design point for safe LLM-assisted causal discovery: useful semantic proposals, verifier-owned graph changes, and replayable evidence for every accepted edge.

Limitations

The main claim is bounded to four small discrete BN graphs and an observational-only verifier. This scope is deliberate: it enables trace-level auditing, exact matched controls, and controlled text diagnostics before scaling the same contract to larger graph families. Intervention-aware discovery is a natural extension; the present intervention analyses use sample concatenation rather than a fully intervention-aware score or GIES-style verifier

(Hauser and B"uhlmann, 2012). Benchmark leakage remains a general risk for LLM-based causal discovery, addressed here through anonymized variables, hidden dataset names, and text-regime perturbations (Zhou et al., 2024; Feng et al., 2025b). The result is therefore a bounded design-point claim rather than broad accuracy dominance. The verifier is score-specific by design: BDeu diagnostics support the same trace behavior under one alternative discrete BN score, while score dependence remains explicit and other score families can be locked before deployment. Closed proposer dependence is treated as model-family sensitivity: GPT-5.2 and GPT-5.4-mini test a closed-proposer boundary, and the proposer remains a replaceable ranker over explicit edit ids.

Ethical Considerations

Causal graphs inferred from limited or biased data should be treated as hypotheses for expert review, not automatic decision rules. S2CRA is designed for auditability: every accepted graph change is tied to a legal edit id, a score decision, and a replayable trace. This does not remove the need for domain validation, but it makes the role of language-visible proposals and statistical verification explicit.

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 Evidence Details

This appendix is organized as a verification map for the main paper. It first states which evidence belongs to the headline observational-only claim, then records the locked protocol and trace structure, then gives same-protocol diagnostics, and only then lists historical or auxiliary evidence. The ordering is intentional: a reader should be able to audit the main claim without mixing it with earlier protocol variants.

A.1 Roadmap and Evidence Scope

Main-paper evidence boundary. Tables 3–7 are derived from one observational-only BNRepository trace set, after protocol selection on a separate synthetic development suite. This chain supports the headline claim: a deterministic verifier-guided proposal channel reaches comparable directed graph quality at much lower verifier effort, with replayable traces and controlled language-risk diagnostics. No intervention-inclusive result, historical ablation, or auxiliary case study is used to make that headline claim.

Why the gains are meaningful. The main evidence should be read along two coupled axes: directed recovery and verifier effort. The oriented-edge gain over GES shows that the locked proposal-verification loop recovers more directed structure than the strongest classical mean baseline in the main table, while the effort table shows that this happens with orders-of-magnitude fewer verifier score evaluations than strong matched atomic-search controls. The appendix diagnostics then

show why the result is more than a single final F1 number: the trace records expose proposal shifts, rejection categories, accepted edits, score deltas, smaller-proposer behavior, and alternative-score agreement under the same contract.

Same-protocol appendix diagnostics. Appendix A.3 and Appendix A.4 are still part of the locked observational-only evidence chain. They document how the protocol was selected, how the proposer interface was screened, what the per-case traces look like, how smaller-proposer and alternative-score checks behave, and how proposal-level text diagnostics shift under fixed verification. These diagnostics explain the main result; they do not introduce a second headline.

Historical and auxiliary evidence. Appendix A.6 preserves earlier protocol variants and auxiliary analyses for continuity with the project history. Those results are useful for context and boundary analysis, but they should not be averaged with or used to reinterpret the locked observational-only benchmark.

Navigation.

- To verify the headline protocol: Appendix A.2 and Table 2.
- To check protocol selection and proposer tuning: Appendix A.3.
- To inspect trace-level behavior, smaller proposer, score robustness, and text diagnostics: Appendix A.4.
- To audit the LLM interface and structured outputs: Appendix A.5.
- To see earlier non-headline evidence: Appendix A.6.

A.2 Evidence Chain and Trace Structure

Evidence lineage. The evidence chain is: *development-grid evaluations* $\rightarrow$ *fixed protocol selection* $\rightarrow$ *observational-only BNRepository suite* $\rightarrow$ *structured traces* $\rightarrow$ *paper tables/figures*. Each evaluation stores structured metrics together with the configuration used: model label, text regime, move policy, hard-gate settings, score family, and budget. Main-text tables and appendix diagnostics are derived from the same structured trace records to avoid manual transcription. Those records also contain proposal rankings, accepted edits, rejected proposals, score deltas, and final graphs.

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. For each graph size, the development-suite generator samples a DAG over anonymous variables in topological order subject to a maximum in-degree, samples variable arities in $\{2, 3, 4\}$, samples local conditional probability tables from symmetric Dirichlet distributions, and produces 5,000 observational rows for each sample seed. 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 8 summarizes the four discrete networks used in the observational-only main track.

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

Table 8: BNRepository network statistics used in the observational-only main track.

Reproducibility modes. There are two intended reproduction modes. **Trace-level regeneration** reconstructs paper tables and figures from stored trace records without any model calls; this is the right mode for auditing reported numbers. **End-to-end reruns** reproduce the full procedure by rerunning the development-suite grid, selecting a protocol, executing the observational-only suite, and exporting tables/figures; this requires model access for S2CRA and LLM-ZERO. The paper's numerical claims should be checked against trace records first, then rerun only if model access and stochastic variation are part of the audit.

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 re-running 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. For the observational-only BN main track, the proposal payload is the ranked shortlist-id format in Appendix A.5.3; free graph initialization and free-form edit prompts are documented only as reusable interfaces and are not used to generate the main BN results.

A.3 Protocol Selection and Proposer Tuning (Supplementary)

A.3.1 Synthetic Dev-Suite Protocol Ranking

Table 9 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 2 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.

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.

A.4.1 Per-Network Breakdown

Table 4 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 gains with verifier rejections, containment cases (no accepted edits when the initializer is already optimal), and stress cases with regressions or low-yield 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:   +0.00
                     no clear
                     gain
- it=4 REV V3->V5    boundary:   +0.00
                     no clear
                     gain
- it=5 ADD V1->V3    score: no   -2.67
                     improve
```

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.

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 9: Synthetic development-suite protocol ranking. The selected configuration is highlighted. Selection rule: higher oriented-edge F1 → lower SHD → lower budget → lower $d_{\max}$.

Work	LLM graph init	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; BN main)	N	Y	Y	Y	Y	Y	Y	(gated)

Table 10: Supplementary comparison of nearby hybrid systems along the properties most relevant to our protocol. "P" indicates a partial or non-fixed form. The BN main protocol disables LLM graph initialization and uses the model only as a bounded edit-candidate ranker.

```
> 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.

Variant	Comb. F1↑	Asia	Child	Acc./case	ScoreEval/case	Repeat-rej./case	MaxInDeg-rej./case
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 11: Proposer-tuning screen on BNRepository (asia+child, seeds 42/43/44) under the fixed obs-only verifier contract. Acc./case counts accepted edits; ScoreEval/case counts verifier score evaluations; batch rejects record deterministic pre-score rejections (e.g., repeats).

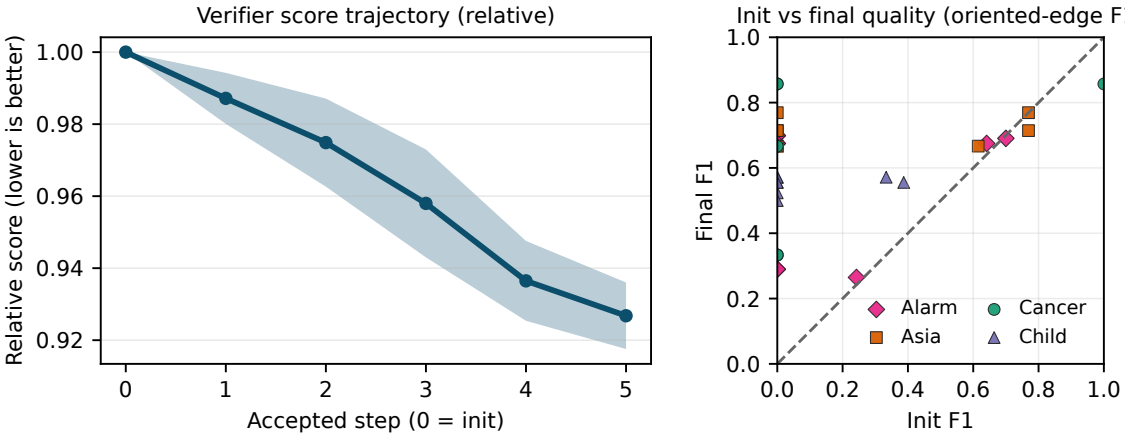

Figure 4: Offline trajectory evidence under the pre-specified observational-only protocol. Left: relative verifier score over accepted steps. Right: initial vs final oriented-edge F1 per case (points above the diagonal indicate improvement over the selected statistical initializer).

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

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: no clear gain	+0.00
- it=4 ADD V3->V1	score: no improve	-6.24
- it=5 ADD V1->V3	score: no improve	-6.24

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

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

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

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

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).

```
+ 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).

Proposer	SHD↓	OE-F1↑	ScoreEval/case↓
Main	14.194 ± 0.048	0.608 ± 0.003	40.0
gpt54-mini	14.500	0.620	40.0

Table 12: Smaller-proposer control under the same verifier contract.

Stage	Condition	Run score	Rescore	Edit-agree↑	Final-dir-agree↑	Errors
Text	Anonymized	BIC+edge	BDeu	0.909	0.917	0
Text	Raw Names	BIC+edge	BDeu	0.891	0.917	0
Text	Raw+Desc	BIC+edge	BDeu	0.909	0.917	0
Text	Misleading	BIC+edge	BDeu	0.727	0.833	0
Rerun	BDeu rerun	BDeu	BDeu	1.000	1.000	0
Small	gpt54-mini	BIC+edge	BDeu	0.868	0.917	0

Table 13: Offline score-robustness rescore of final graphs and accepted edits.

```
- 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
```

A.4.3 Smaller-Proposer and Score-Robustness Diagnostics

Table 12 reports the closed smaller-proposer control under the same verifier contract, replacing only the proposal model while keeping the initializer, shortlist policy, gates, and budget fixed. Table 13 summarizes an offline BDeu-style rescore of final graphs and accepted edits from the archived traces; the dedicated BDeu rerun is shown separately from the BIC-style traces that are rescored offline. These diagnostics should be read as boundary checks rather than new headline claims. The smaller-proposer row tests whether the contract remains stable when the closed proposer is reduced, keeping model-family sensitivity auditable. The BDeu rows test one alternative discrete BN score; agreement rates indicate that many accepted edits and final directions remain aligned under that score.

A.4.4 Proposal-Level Text Diagnostics

Table 14 complements the main text-regime table by reporting proposal-overlap and rejection-category shifts relative to the anonymized condition. These diagnostics are computed from the same observational-only trace bundle as Table 7. The overlap columns compare proposed and accepted edit identities against the anonymized run. Low overlap therefore means that language changed what the proposer asked the verifier to in-

Regime	Prop-overlap	Acc-overlap	Top reject reasons
Anonymized	1.000	1.000	score_not_improve:30, bound-ary_no_clear_gain:1
Misleading	0.000	0.000	score_not_improve:23, bound-ary_no_clear_gain:4
Raw Names	0.000	0.000	score_not_improve:28
Raw+Desc	0.000	0.000	score_not_improve:25, bound-ary_no_clear_gain:1

Table 14: Proposal-level text diagnostics across controlled text regimes.

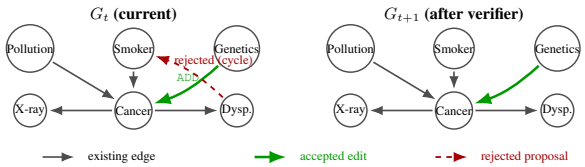

Figure 5: Illustrative atomic-edit step under the propose-and-verify contract. The proposer may suggest edits, but the verifier deterministically accepts only hard-gate-valid, score-improving moves; rejected proposals are logged for replay.

spect, not that the verifier changed its acceptance rule. The rejection summaries then show where the deterministic gates contained those shifted proposals.

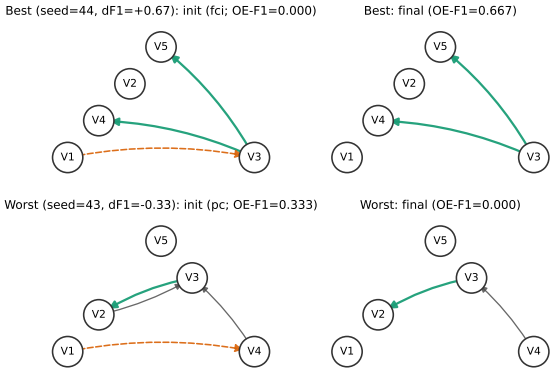

Figure 6: Real case visualization from an archived BN-Repository trace (Cancer network; anonymized variables). Gray edges denote the initializer graph. Green edges denote accepted edits (i.e., edges in the final graph but not in the initializer). Red dashed edges show one rejected proposal, illustrating 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. The BN main results use the shortlist-ranker prompt in Appendix A.5.3; the initialization and free-batch edit templates below document the broader implementation interface and are not part of the observational-only headline protocol.

A.5.1 Graph Initialization Prompt (Optional; Not Used in BN Main)

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; Not Used in BN Main)

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 (BN Main Proposal Interface)

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 shortlist.
`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 `conf ig/bn_text_regimes.json`.

Example fixture. Table 15 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 15: 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, but they are not part of the May observational-only headline protocol.

A.6.1 BNRepository: Earlier Variant 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 16: BNRepository earlier variant main table (hard-gate protocol). Values are averaged over the four BN networks and three random seeds; this table is supplementary and not part of the locked observational-only headline protocol.

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 structure metrics change 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 17: Hybrid init audit on BNRepository: init-source distribution and refinement deltas. Positive ΔSHD indicates improvement (lower SHD).

A.6.3 BNRepository: Earlier Variant Trajectory View

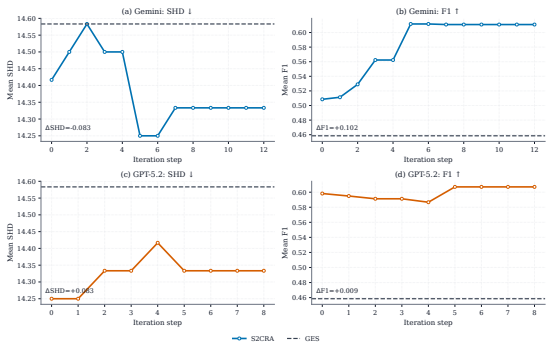

Figure 7: Trajectory evidence on BNRepository (earlier protocol variant). The plot summarizes how final structure metrics evolve relative to the selected initializer across cases.

A.6.4 BNRepository: Per-Network Breakdown

To avoid hiding aggregate effects, 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 18: 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
Earlier intervention-inclusive	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 19: Earlier-variant run-to-run stability on BNRepository for S2CRA with GPT-5.2. We rerun the benchmark three times with identical datasets, seeds, and fixed hyperparameters; values are mean $\pm$ std across reruns.

A.6.6 BNRepository: Additional Parity Tables

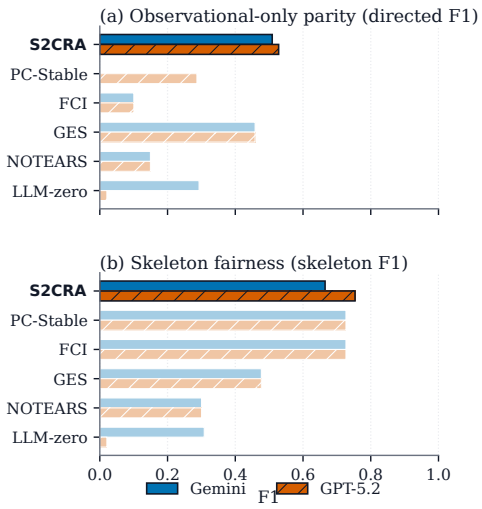

Figure 8: BNRepository parity views (earlier protocol variant). (a) Observational-only parity (directed F1) across methods. (b) CPDAG-friendly skeleton F1, giving partial credit to partially oriented outputs (PC/FCI).

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

Model	Earlier-variant setting	SHD ↓	Precision ↑	Recall ↑	F1 ↑
Gemini	Intervention-inclusive	14.333	0.788	0.522	0.611
Gemini	Observational-only	14.750	0.711	0.420	0.509
GPT-5.2	Intervention-inclusive	14.333	0.748	0.532	0.607
GPT-5.2	Observational-only	16.500	0.813	0.424	0.528

Table 21: Earlier-variant observational-only fairness check for S2CRA on BNRepository (averaged over networks and seeds). This supplementary table is separate from the locked observational-only headline protocol.

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

A.6.7 BNRepository: Full Ablation Table

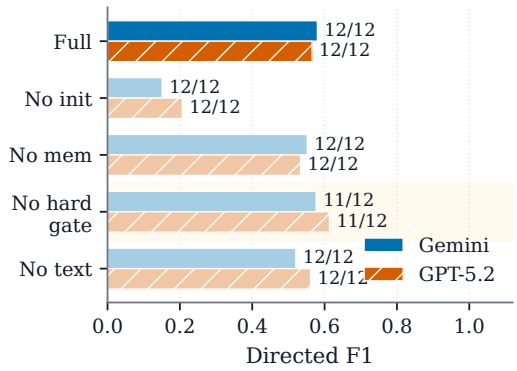

Figure 9: BNRepository ablations (earlier protocol variant). Bars show directed F1 under variants that remove or change key components.

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

A.6.8 BNRepository: Earlier Variant 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 24: Earlier-variant 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 25: Earlier-variant 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

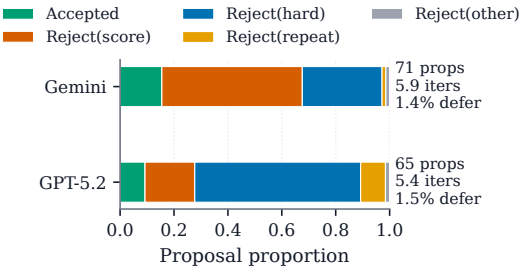

Figure 10: Gate behavior on BNRepository (earlier protocol variant). Stacked bars show the proportions of accepted edits and mutually exclusive rejection categories per proposal.

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 26: 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 27: 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 focused diagnostic that toggles whether the LLM sees original variable names and/or the benchmark network name. This audit is scoped as a sanity check and does not affect the main BN claims; it tests whether benchmark-safe prompt hygiene is 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 28: 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

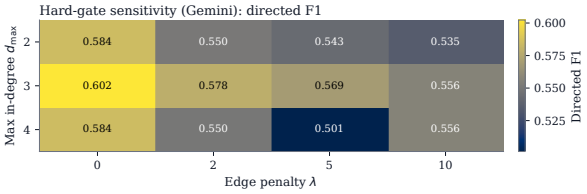

Figure 11: Hard-gate sensitivity on BNRepository (earlier protocol variant). Cells show directed F1 for each $(d_{\max}, \lambda)$ pair under the fixed protocol, averaged over four networks and three random seeds.

A.6.13 BNRepository: Intervention Micro-Analysis

Network	Intervention-inclusive 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 29: Within-S2CRA intervention effect on BNRepository for Gemini in an earlier protocol variant (directed F1). The intervention-inclusive run uses observational+interventional tables; the obs-only rerun removes intervention tables.

Network	Intervention-inclusive 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 30: Within-S2CRA intervention effect on BNRepository for GPT-5.2 in an earlier protocol variant (directed F1). The intervention-inclusive run uses observational+interventional tables; the obs-only rerun removes intervention tables.

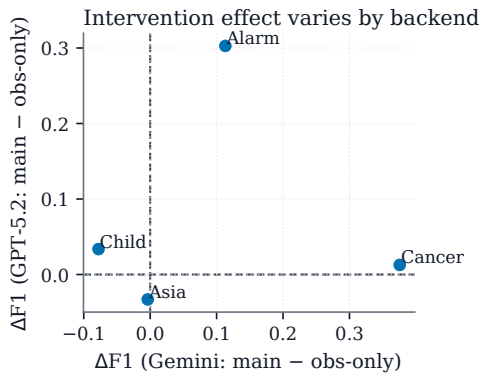

Figure 12: Per-network intervention effect for S2CRA (earlier protocol variant): each point is one BN network, with x-axis showing ΔF1 (Gemini: main − observational-only) and y-axis showing ΔF1 (GPT-5.2: main − observational-only).

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 yielding lower SHD, a supplementary pattern consistent with using a closed-loop verifier to 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 31: Graph discovery on SynthRisk (100 scenarios). The agent has lower SHD than LLM-only while maintaining comparable F1.

A.6.15 Supporting Tracks: Hard Cases and Tübingen

Hard Cases (adversarial traps). Table 32 reports success rates on 4 adversarial cases. Across both LLM backends, the agent has higher success rates than LLM-only prompting under this rubric, suggesting that deterministic gates can reduce trap-triggered hallucinations in this auxiliary setting.

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

Table 32: Hard Cases (4 adversarial trap cases). Values are success rates under our task rubric.

Tübingen Cause–Effect Pairs (bivariate direction). Table 33 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 33: 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 34 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 34: 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%`

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`

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...`