Leveraging Large Language Models for Causal Discovery: a Constraint-based, Argumentation-driven Approach

Zihao Li[1] **Fabrizio Russo**[1]

[1]Department of Computing,
Imperial College London, UK,
zihao.li24@imperial.ac.uk, fabrizio@imperial.ac.uk

Abstract

Causal discovery seeks to uncover causal relations from data, typically represented as causal graphs, and is essential for predicting the effects of interventions. While expert knowledge is required to construct principled causal graphs, many statistical methods have been proposed to leverage observational data with varying formal guarantees. Causal Assumption-based Argumentation (ABA) is a framework that uses symbolic reasoning to ensure correspondence between input constraints and output graphs, while offering a principled way to combine data and expertise. We explore the use of large language models (LLMs) as imperfect experts, eliciting semantic structural constraints from variable names and descriptions and integrating them with statistical evidence through Causal ABA. We propose ABAPC-LLM as a principled and conservative approach to hybrid data- and LLM-driven causal discovery. Additionally, we introduce an evaluation protocol to mitigate memorisation bias when assessing LLMs for causal discovery and show competitive performance on novel semantically grounded random benchmarks, as well as on standard small- and medium-sized benchmarks.

1 INTRODUCTION

Understanding the causal structure that governs observational data is central to explanation, prediction, and decision-making and underpins reasoning about interventions (Pearl, 2009; Peters et al., 2017; Spirtes et al., 2000). Although interventional/experimental data can help resolve causal directions, it is costly and often unavailable; accordingly, a rich literature has focused on extracting causal graphs from observational data only. Constraint- and score-based methods infer (equivalence classes of) directed acyclic graphs (DAGs) from statistical regularities, but reliability hinges on assumptions and sample size (Spirtes et al., 2000), motivating the use of domain knowledge alongside data.

Formulating such background knowledge is labour-intensive: domain experts must identify relevant variables, specify admissible orientations, and articulate forbidden relations. Existing pipelines either encode these judgements as hard structural constraints, fixing or forbidding edges before discovery (Meek, 1995), or as Bayesian priors that weight the search over graph structures (Cooper and Herskovits, 1992; Heckerman et al., 1995). Both hard constraints and priors can encode expertise, but offer limited mechanisms to diagnose and resolve conflicts with different sources of evidence such as observational data.

Causal Assumption-Based Argumentation (Causal ABA) (Russo et al., 2024) reframes discovery as an ABA framework (Bondarenko et al., 1997; Cyras et al., 2017) where candidate causal edges are defeasible assumptions, and rules for acyclicity and d-separation (Pearl, 2009) induce attacks between incompatible constraints. Solving it yields DAGs supported by the strongest jointly defensible subset of data- and expert-derived constraints, with provenance via the accepted/defeated assumptions. Crucially, this framework allows to integrate difference sources of knowledge without treating it as ground truth, and to expose conflicts within the input constraints transparently.

Large language models (LLMs) provide an attractive, yet imperfect, source of knowledge. They encode broad semantic and scientific information and can perform structured reasoning when prompted appropriately (Bommasani et al., 2021; Kojima et al., 2022). To leverage this information, one needs to rely on semantically meaningful variable metadata (names and descriptions rather than anonymised identifiers such as $X_1, \ldots, X_n$), providing semantic signal unavailable to data-only discovery pipelines. Yet, LLMs can hallucinate plausible-sounding but unsupported causal links (Ji et al., 2023), which goes against the rigour that causal discovery demands. Additionally, their training on web-scale corpora, where many benchmark causal graphs are publicly available,

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:3631–3667.

makes it difficult to discern whether suggested dependences stem from genuine reasoning or the verbatim retrieval of the memorised causal graph in its entirety.

These tensions raise two research questions: how can we exploit the knowledge embedded in LLMs without sacrificing rigour and transparency, and how can we assess LLMs' contributions to causal discovery while mitigating memorisation bias? We address these questions as follows:

- We design a robust **LLM integration pipeline** that elicits high-precision directional constraints, consensus-filters them, and integrates them defeasibly with data-derived conditional-independence evidence within Causal ABA.
- We develop a **synthetic evaluation protocol** that grounds random DAGs in the `CauseNet` knowledge graph (Heindorf et al., 2020) to better assess LLM generalisation beyond memorised benchmarks.
- We show competitive performance on standard benchmarks and on our synthetic protocol.

A key design choice is conservatism: semantic information from LLMs is used to orient (not introduce) edges, and statistical evidence can overrule LLM-derived constraints through defeasible reasoning.

2 RELATED WORK

We build on three strands of literature: causal discovery from observational data, the integration of expert knowledge into causal discovery, and the use of LLMs for knowledge elicitation and causal discovery.

Causal Discovery. Causal discovery algorithms aim to reconstruct causal graphs from observational data, often under the assumptions of acyclicity and faithfulness: that the causal structure is a DAG and that all and only the conditional independencies implied by the DAG are present in the data (Spirtes et al., 2000). A rich literature (see e.g. (Glymour et al., 2019; Vowels et al., 2022; Zanga et al., 2022)) has proposed various algorithms with different assumptions, guarantees, and trade-offs between statistical and computational efficiency. Throughout, we assume causal sufficiency (no unobserved confounders) (Spirtes et al., 2000).

Constraint-based algorithms such as Peter-Clark (PC) (Spirtes et al., 2000) and its improved version, Majority-PC (MPC) (Colombo and Maathuis, 2014), infer Markov equivalence classes (MEC, the upper bound of what can be discovered from observational data alone (Verma and Pearl, 1990)) through conditional independence (CI) tests, whereas score-based procedures such as greedy equivalence search (Chickering, 2002) or its faster counterpart Fast Greedy Search (FGS) (Ramsey et al., 2017) trade combinatorial search for statistical fit criteria. We use MPC to derive an efficient number of CI constraints, but our approach is agnostic to the underlying testing procedure. We also compare against representative score-based (FGS), continuous-optimisation (NOTEARS-MLP (Zheng et al., 2020)[1]), and order-based baselines (GRaSP (Lam et al., 2022), BOSS (Andrews et al., 2023)).

Hybrid approaches tighten the shortcomings of single types of methods by combining constraint- and score-based approaches (Tsamardinos et al., 2006) while logic-based approaches embed logical inference in the process e.g. via SAT-based encodings (Hyttinen et al., 2014). Causal ABA fits into the logical constraint-based literature using ABA (Bondarenko et al., 1997) to guarantee that every inferred orientation is backed by an explicit chain of assumptions and rules (Russo et al., 2024). Our work complements these efforts by integrating structured assumptions from LLMs.

Integrating Expert Knowledge. Prior knowledge has long been recognised as essential to trim the search space and enforce domain restrictions in causal discovery (Cooper and Herskovits, 1992; Heckerman et al., 1995). Conventional approaches require experts to enumerate admissible edges, ancestral relations, or forbidden paths before running an algorithm (Meek, 1995). (Hyttinen et al., 2014) incorporate domain constraints during search or as post-processing filters, including answer-set-programming (ASP, (Gebser et al., 2019)) and weighted-constraint formulations that can encode background knowledge and resolve inconsistent constraints during optimisation. By embedding (LLM- or human-sourced) statements into Causal ABA, we can integrate expert knowledge defeasibly and expose conflicts with data-derived constraints. This is complementary to logic programming approaches: LLM- or human-derived statements are represented as assumptions whose acceptance, rejection, and attacks can be inspected, giving provenance for why a semantic constraint survived or was defeated rather than only enforcing it as ordinary hard background knowledge.

Recent work has moved toward interactive integration of expert knowledge during discovery. This includes approaches that request human input on uncertain edges or constraints as the search progresses (Cao and Liu, 2024; Kitson and Constantinou, 2025) and expert-in-the-loop procedures that iteratively refine a learned graph with domain feedback (Ankan and Textor, 2025). These methods depend on ongoing human interaction or repeated model refitting. In contrast, our pipeline elicits structured constraints once (from LLMs or experts) and integrates them defeasibly within Causal ABA, allowing conflicts with data-derived CI evidence to be resolved transparently without iterative expert sessions.

LLMs for Causal Discovery. LLMs exhibit strong few-shot generalisation (Brown et al., 2020) and can be prompted to elicit causal constraints from variable metadata (Kıcıman et al., 2024; Zhao et al., 2023). For causal discovery, this ability enables LLMs to act as experts and *translate metadata into candidate structural constraints*.

[1]Without claiming its fitness for causal discovery; e.g. see cautionary discussion in (Reisach et al., 2021).

A growing literature leverages LLMs in three complementary roles (Wan et al., 2025): (i) querying them to orient edges or assemble full causal graphs (Jiralerspong et al., 2024; Kıcıman et al., 2024; Vashishtha et al., 2025); (ii) using them as assistants that critique, refine, or document algorithmic outputs via refinement loops or natural-language analytics (Abdulaal et al., 2024; Gkountouras et al., 2025; Khatibi et al., 2024; Liu et al., 2024); and (iii) coupling textual constraints with statistical search through hard (Chen et al., 2025) or soft constraints (Ban et al., 2025), or even by replacing CI tests with conversational judgments within PC (Cohrs et al., 2023). This literature also highlights that LLM and human judgements are noisy imperfect-expert signals rather than ground truth (Vashishtha et al., 2025). In our approach, MPC provides a set of CI statements from data, while the consensus-filtered LLM constraints are encoded as additional required/forbidden arrow facts and enforced within Causal ABA only when they do not contradict strong statistical counter-evidence (§4). As in (Jiralerspong et al., 2024; Kıcıman et al., 2024; Vashishtha et al., 2025), we solicit direct causal relations, yet we consolidate the queries into a single prompt for improved efficiency and context. Additionally, we adapt the consensus refinement strategy of (Cohrs et al., 2023) and benchmark against the BFS-style LLM-only method of (Jiralerspong et al., 2024), which iteratively expands a graph via pairwise queries. Compared to directly enforcing text-derived constraints, our contribution is using argumentation to expose and resolve conflicts, providing a principled integration with data and traceable justifications for accepted and rejected orientations.

LLM-agentic systems have also been proposed for interactive graph discovery, where agents plan queries and update graphs over multiple steps (Havrilla et al., 2025). Our method is non-agentic and avoids iterative prompting loops; instead, we prioritise high-precision, consensus-filtered constraints that can be justified and, if necessary, overturned by the argumentation-based consistency checks.

Despite this promise, LLM responses remain fallible. Benchmarks reveal gaps between memorised answers and genuine causal reasoning (Jin et al., 2023; Wan et al., 2025; Zečević et al., 2023), and repeated studies caution against using LLMs as sole decision-makers for causal discovery (Wu et al., 2025). These limitations motivate treating elicited knowledge defeasibly rather than as ground truth and designing evaluation protocols that mitigate memorisation bias. Our empirical study therefore treats LLMs as an input source to the Causal ABA integration mechanism.

3 BACKGROUND

Let $G = (\mathbf{V}, E)$ be a graph with nodes $\mathbf{V}$ and edges $E \subseteq \mathbf{V} \times \mathbf{V}$. Edges can be directed or undirected (for $u, v \in \mathbf{V}$, if both $(u, v) \in E$ and $(v, u) \in E$, then u and v are connected by an undirected edge); the *skeleton* replaces every directed edge with an undirected one. A node's degree is the number of edges linked to it. A *path* $p = x_1 \dots x_n$ is a sequence of adjacent nodes; it is *simple* when all nodes are distinct. It is *directed* if $(x_i, x_{i+1}) \in E$ for all i, and cyclic if it contains a repeated node, i.e. $x_i = x_j$ for some $i < j$; a directed cycle is a cyclic directed path. A *directed acyclic graph (DAG)* has only directed edges and no cycles and will serve as our causal graph formalism. A DAG is ascribed a causal semantics by interpreting nodes as random variables and edges as direct causal effects (Pearl, 2009). A Bayesian Network (BN) is a pair $(G, \mathcal{P})$ where $\mathcal{P}$ is a joint distribution over $\mathbf{V}$ that satisfies the Markov condition relative to G: every variable is independent of its non-descendants given its parents. For pairwise disjoint sets $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}$ we let $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})$ indicate that $\mathbf{X}$ and $\mathbf{Y}$ are *independent* given the conditioning set $\mathbf{Z}$; $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \emptyset)$ is simply written as $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y})$ and singleton sets $\{x\}$ with $x \in \mathbf{V}$ are denoted by x (e.g., $(\{x\} \perp\!\!\!\perp \{y\} \mid \emptyset)$ is written as $(x \perp\!\!\!\perp y)$). The link between DAGs and CI is captured by the d-separation criterion (Pearl, 2009).

Causal ABA (Russo et al., 2024) is a framework that combines computational argumentation (Dung, 1995; Toni, 2014) with causal reasoning. An ABA framework (ABAF) (Bondarenko et al., 1997) is a tuple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, {}^{-} \rangle$ where $\mathcal{L}$ is a formal *language*, $\mathcal{R}$ is a set of *inference rules* of the form $head \leftarrow body$ where $head \in \mathcal{L}$ and $body$ is a finite (possibly empty) set of elements from $\mathcal{L}$, $\mathcal{A} \subseteq \mathcal{L}$ is a non-empty set of *assumptions*, and ${}^{-}$ is a total mapping from $\mathcal{A}$ into $\mathcal{L}$ (*contrary*). In Causal ABA, the language $\mathcal{L}$ includes statements about graphical (causal) relationships (e.g. (x, y)), paths and independencies; the rules $\mathcal{R}$ encode the principles of acyclicity and d-separation, and the assumptions $\mathcal{A}$ represent candidate causal relations and/or independencies that can be accepted or rejected based on the evidence and their relations. An assumption $a \in \mathcal{A}$ *attacks* an assumption $b \in \mathcal{A}$ if and only if a can be used to derive the contrary of b, i.e., $\bar{b}$, using the rules in $\mathcal{R}$.

An ABAF is evaluated using argumentation *semantics* (Baroni et al., 2011), which determine which sets of assumptions (called *extensions*) can be collectively accepted based on their ability to cohexist (conflict-freeness) and defend against external attacks (see e.g. (Toni, 2014) for formal definitions and examples). A set of assumptions $S \subseteq \mathcal{A}$ will produce no solution if it contains conflicts, i.e., if there exists $a, b \in S$ such that a attacks b. Causal ABA employs *stable semantics* to guarantee a one-to-one correspondence between the accepted assumptions and the d-separation relations induced by the output DAG (Russo et al., 2024). This makes inconsistencies explicit: incompatible constraints attack each other, and only jointly defensible sets are retained. A bodyless rule ($head \leftarrow$) is called a *fact* and represents an unconditionally true statement in the ABAF i.e., it is by default included in every extension. In the implementation, the abstract rules, assumptions, and facts are *grounded* into a ASP program over the variables and candidate paths before stable extensions are computed; this grounding step is the main source of the scalability trade-offs discussed below.

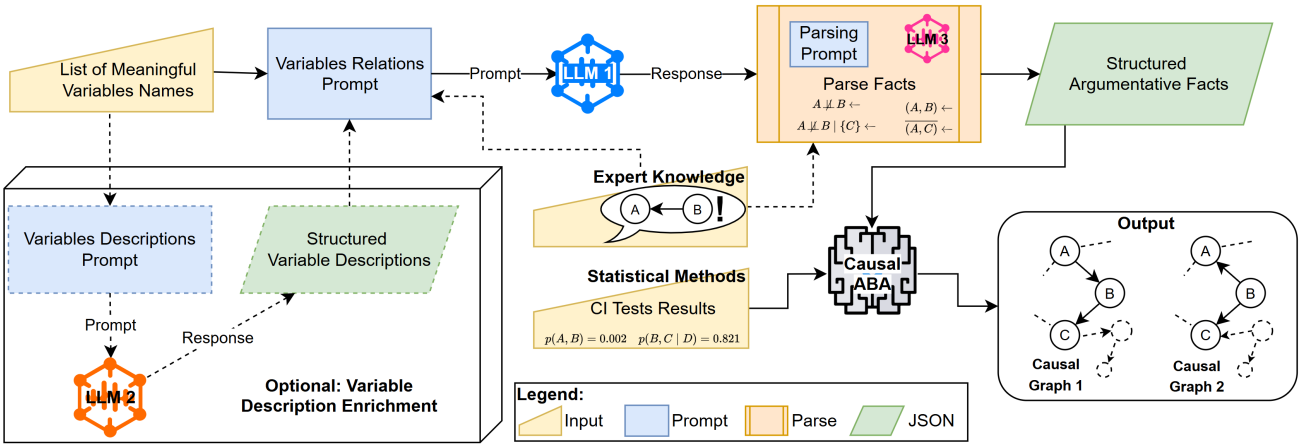
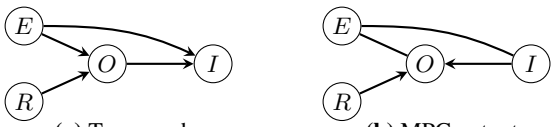


Figure 1: LLM Integration Pipeline: Given a set of variables with their names and (possibly) descriptions, we prompt an LLM to generate pairwise causal statements. These statements are then parsed into structured assumptions for Causal ABA, which combines them with data-derived independence or arrow constraints and background knowledge to infer a set of causal graphs. Expert knowledge can be injected in the LLM prompts or as defeasible facts. Variable descriptions and LLM parsing are optional components of the pipeline (dashed arrows) but enhance the quality of the generated assumptions. Detailed prompts and parsing rules are provided in App. A.

In (Russo et al., 2024) ABAPC is proposed as an instantiation of Causal ABA using MPC as the statistical engine to derive CI constraints from data. The CI constraints from MPC are ranked based on a transform of their associated p-values, and the algorithm iteratively attempts to construct stable extensions by removing the least *credible* facts (according to the ranking) until one is found. In this way, ABAPC treats CI outcomes as defeasible and selects a high-confidence subset that is jointly consistent with the d-separations of one or multiple DAGs.

Example 3.1 *Consider the simple four-variable DAG in Fig. 2(a): Education (E) and Race (R) influence Occupation (O), which in turn influences Income (I); Education also directly affects Income ($E \to I$). From synthetic data*



(a) True graph. **(b)** MPC output.

Figure 2: Toy example graphs (true DAG vs. MPC output).

sampled from this ground-truth DAG we obtain 23 CI tests using MPC. These test results, including $E \perp\!\!\!\perp R$ (correct) and $E \perp\!\!\!\perp R \mid O$ (wrong wrt the d-separations in the ground truth DAG), become facts in Causal ABA. Conditioning on the collider O should render E and R dependent, yet PC-style procedures treat both statements as equally trustworthy. When MPC consumes these tests, it returns the partially oriented DAG in Fig. 2(b). Causal ABA ingests the same CI statements, but reasons about their joint consistency. Each test result is ranked by credibility, and the framework searches for a stable extension. Including both independence statements yields no stable extension: the d-separation rules make the statements inconsistent, because $E \perp\!\!\!\perp R \mid O$ can hold together with $E \perp\!\!\!\perp R$ only if R and E are disconnected from the other variables—contradicting many of the other CI tests. After progressive exclusions, the stable extension contains only the correct independence statement, yielding the ground-truth DAG.

This toy example highlights the key advantage of ABAPC over PC-style procedures: it can discard a small number of low-credibility CI outcomes that are globally inconsistent, preventing cascading orientation errors. Having described our statistical and symbolic engines, we turn to our proposed method to integrate LLM-derived knowledge with data.

4 LLM-AUGMENTED CAUSAL ABA

Our methodology integrates semantic knowledge from LLMs and statistical test results with the symbolic rigour of Causal ABA. The core idea is to use an LLM as a proxy domain expert to generate high-precision structural constraints, which are then encoded as facts within the argumentative framework to prune the vast search space of possible causal graphs. This hybrid approach, depicted in Fig. 1, aims to enhance both the efficiency and the accuracy of the discovery process. The pipeline consists of a method to elicit structured constraints from LLMs (left), and the formal integration of these constraints into Causal ABA (right). We refer to the resulting LLM-augmented framework as **ABAPC-LLM**. Human expert knowledge is included as optional, and not currently part of the experiments, but can be readily integrated. Importantly, the LLM never sees the

Algorithm 1 ABAPC-LLM

Input: $\mathcal{I}, \alpha, |\mathbf{V}| = d$; metadata M, LLM L, repetitions $r = 5$, MPC skeleton $\mathcal{C}$

1: $\mathbf{T} \leftarrow$ Lines 1–8 of (Russo et al., 2024, Alg. 1) on $(\mathcal{I}, \alpha, d)$ ▷ CI facts sorted by strength
2: $\mathbf{R} \leftarrow [\,], \mathbf{F} \leftarrow [\,]$
3: **for** $i = 1, \ldots, r$ **do**
4: $\quad (R_i, F_i) \leftarrow L(M)$ ▷ required and forbidden arrows
5: **end for**
6: $\mathbf{R} \leftarrow \{(x, y) : (x, y) \in \cap_i R_i \ \wedge \ x - y \in \mathcal{C}\}$
7: $\mathbf{F} \leftarrow \{\, \overline{(x, y)} : (x, y) \in \cap_i F_i \,\}$
8: $\mathbf{T} \leftarrow \mathbf{T} + [\mathbf{R}, \mathbf{F}]$ ▷ add unanimous compatible LLM facts
9: $G \leftarrow$ Lines 9–21 of Causal ABA, with `causalaba` grounded over $\mathcal{C}$ and $\mathbf{T}$
10: **return** G

observational samples: it is queried only on variable names/-descriptions. Algorithm 1 is written as a modification of the original Causal ABA algorithm: the CI-fact construction, relaxation loop, and final graph scoring are unchanged except that the grounded ABAF receives the additional unanimous LLM facts and uses the reduced MPC skeleton.

4.1 CONSTRAINT ELICITATION PIPELINE

The quality of LLM-derived knowledge is highly dependent on the elicitation process (Anthropic, 2024). We propose a robust pipeline to transform unstructured metadata (in the form of variable names and descriptions) into conservative and precise causal constraints.

Eliciting Structural Constraints. The process begins with a set of variables, each with a meaningful name and an optional longer description. We developed a detailed prompt that instructs a primary LLM (LLM 1 in Fig. 1) to act as a domain expert, asking for the generation of two lists containing the following types of information based on established knowledge or logical consistency:

- **Required Directions**: Causal relationships ((x, y) with $x, y \in \mathbf{V}$) that the causal graph should contain.
- **Forbidden Directions**: Causal relationships $\overline{(x, y)}$ that the causal graph should not contain.

We use these as directional constraints (edge orientations) rather than as claims that an adjacency exists in the data. The LLM is instructed to structure its response and provide brief justifications for each constraint. The complete prompt is provided in App. A.1. We then apply a schema-guided extraction stage (LLM 3 in Fig. 1, using (567-labs, 2025)) to parse and validate the free-form output into structured required/forbidden arrow lists; details are in App. A.2.

Consensus for High Precision. To mitigate the stochasticity of LLM outputs and adhere to the high-precision requirement of causal discovery, we adapt the majority voting idea from (Cohrs et al., 2023) and introduce a consensus-based filtering step. We query the primary LLM five times independently with the same prompt. The final constraint sets consist only of the intersection of the required and forbidden arrow sets across all five parsed responses. This conservative approach ensures that only the most consistently suggested constraints are passed to Causal ABA, significantly reducing the risk of incorporating spurious constraints (see App. A.3 for details). Of course, consensus does not guarantee correctness; it is a pragmatic variance-reduction step aimed at improving precision.

4.2 INTEGRATION INTO CAUSAL ABA

ABAPC-LLM combines CI evidence from data with semantic arrow constraints elicited from LLMs: a set of required and forbidden arrow facts derived from variable metadata and added to the ABAF alongside the MPC-derived CI facts (line 6-8 of Alg. 1). CI statements are handled within Causal ABA as weighted constraints and are progressively relaxed until the program admits a stable extension (and thus a DAG). In the current implementation, a consensus LLM direction that enters the framework is treated as a hard directional fact, subject to compatibility with the reduced skeleton: an LLM orientation can be blocked when the corresponding adjacency has been removed from the search space because of a CI test. To treat LLM constraints as weak constraints and make them defeasible not only against direct CI facts, they would need to be calibrated against CI-test confidence. Deriving such calibrated semantic weights, allowing low-trust semantic constraints to be dropped or deferred when conflicts arise, is left to future work.

Data-Driven Skeleton Reduction. To improve the scalability of Causal ABA we perform an initial, one-off *skeleton reduction step*: we run the MPC testing procedure to obtain an initial search skeleton, and then ground edge and path rules only over the adjacencies that survive that skeleton search. This follows causal theory and the strategy employed by both PC and Causal ABA, with the difference that, during the subsequent iterative relaxation of lower-confidence CI constraints, these edges are not reintroduced. MPC still runs normally and records the relevant CI tests; the approximation is only in the Causal ABA grounding. Unlike full ABAPC, which reintroduces edge/path rules after relaxing an independence fact, our optimised version does not, so the full correspondence guarantee (Prop. 5.18 in (Russo et al., 2024)) of the unoptimised framework does not apply unchanged and final graphs can differ when recovery would require omitted paths. This optimisation is applied to both ABAPC and ABAPC-LLM. Overall, ABAPC-LLM inherits the CI-testing cost of MPC but adds overhead from solving the ABAF and from a one-off set of LLM calls per dataset; empirical runtimes are discussed in §6.2.

LLM Constraints Enforcement. We then add the consensus-filtered constraints to Causal ABA as facts:

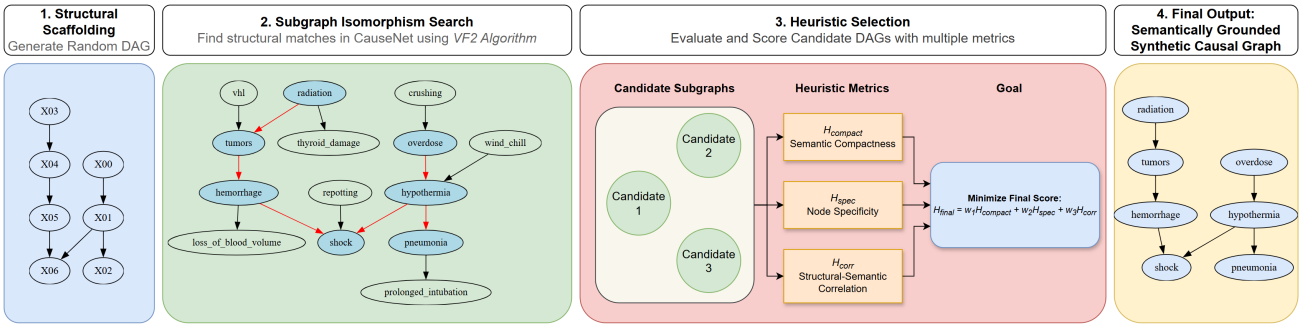


Figure 3: `CauseNet` synthetic benchmark protocol: 1. generate a random DAG, 2. ground it via induced sub-graph isomorphism in `CauseNet`, 3. rank candidate mappings with a three-term heuristic, 4. yielding a semantically grounded DAG with named variables for LLM elicitation.

- A **required arrow** (encoded as $(x, y) \leftarrow$) is enforced only if the undirected edge $x - y$ remains in the reduced skeleton, so semantic knowledge can orient data-supported adjacencies but cannot introduce new ones (yielding conservative, sparse graphs).
- A **forbidden arrow** (encoded as $\overline{(x, y)} \leftarrow$) restricts the program from orienting an edge in a certain direction.

This two-stage process leverages statistical evidence to define the initial search space and employs semantic knowledge to further guide the discovery with information that is often orthogonal to what can be inferred from data. LLMs orient surviving adjacencies only; they cannot add edges. This helps avoiding spurious semantic adjacencies, but limits the gains when CI tests wrongly delete an edge (see §6.2).

5 CAUSENET SYNTHETIC DAGS

A fundamental challenge in evaluating LLMs for causal discovery is the risk of data leakage (Balloccu et al., 2024; Dong et al., 2024). Standard benchmarks, such as `Asia` (Lauritzen and Spiegelhalter, 1988) or `Sachs` (Sachs et al., 2005) in the `bnlearn` repository (Scutari, 2010), are widely published and likely part of LLMs' pre-training corpora. Consequently, high performance on these benchmarks may reflect non-generalisable memorisation. Our proposed protocol, illustrated in Fig. 3, embeds randomly generated DAG structures within `CauseNet`, a large-scale knowledge graph of cause-and-effect relationships extracted from web text (Heindorf et al., 2020). It produces semantically grounded ground-truth DAGs with named variables; we then instantiate each DAG as a BN and sample observational data for structure-learning experiments. The pipeline consists of three main stages, as follows.

1. Structural Scaffolding. We begin by generating a structural "scaffold". To ensure structural diversity in our benchmarks, we first define a DAG topology using one of three methods: generating Erdős-Rényi (ER) (Erdös and Rényi, 1959), Scale-Free (SF) (Barabási and Albert, 1999) graphs, or directly constructing a DAG from a randomly populated lower triangular (LT) adjacency matrix. More details, including generation schema and examples, are in App. B.1.

2. Semantic Grounding via Isomorphism. To assign meaningful concepts to the nodes of the random DAG, we use `CauseNet`. We search for an **induced sub-graph isomorphism** from the random DAG to the `CauseNet` graph. Formally, given a random DAG $H = (V_H, E_H)$ and `CauseNet` $G = (V_G, E_G)$, we find an injective function $f : V_H \to V_G$ such that for any pair of nodes $u, v \in V_H$, an edge $(u, v) \in E_H$ exists if and only if an edge $(f(u), f(v)) \in E_G$ exists. This strategy ensures that the selected sub-graph from `CauseNet` has the same structure as our random DAG. For this search, we employ the VF2 algorithm of (Cordella et al., 2004).

3. Heuristic-Guided Candidate Selection. A single random DAG can have thousands or even millions of isomorphic matches within `CauseNet`. To find the most plausible causal scenario, we score and rank all candidate sub-graphs using a heuristic that combines three distinct cost terms into a single objective function to be minimised:

- **Semantic Compactness:** Measures the thematic coherence of the concepts in a candidate graph. Using pre-computed vector embeddings for all concepts, we calculate the average cosine distance of each concept's vector from their geometric centroid. A low cost indicates that the concepts form a tight semantic cluster (e.g., `virus`, `fever`, `fatigue`).
- **Node Specificity:** Penalises candidate graphs that include overly general or ambiguous "hub" nodes (e.g., `illness`, `problem`). The cost is the average logarithm of each node's degree in the full `CauseNet` graph, favouring sub-graphs composed of more specific, well-defined concepts.
- **Structural-Semantic Correlation:** Induces faithfulness of the graph's topology to the semantic relationships between its concepts. We compute the rank correlation (Spearman, 1904) between all-pairs shortest-path distances in the graph and the corresponding pairwise cosine distances between semantic embeddings. Lower cost (inverse of correlation) means that concepts directly connected in the graph are also semantically very close, ensuring holistic consistency.

The selected candidate is the graph that minimises a weighted sum of these three heuristic cost terms. This pipeline produces semantically grounded causal graphs that serve as a robust foundation for evaluating causal discovery involving LLMs. We make our synthetic DAG generator publicly available as open-source software and provide more details in App. B.2.

Why Semantic Grounding Is Necessary. Since LLM elicitation relies on variable semantics, anonymised variables (e.g., $X_1, \ldots, X_n$ instead of `virus,...fatigue`) would remove the signal and reduce the model to an uninformed oracle. We therefore ground randomly generated structures in real concepts via `CauseNet` to obtain semantically meaningful *yet structurally diverse* graphs; while individual CauseNet facts may appear in pre-training corpora, composing them into random, previously unseen subgraphs makes verbatim retrieval of an entire target DAG implausible and helps distinguish memorisation from generalisation. Our protocol does not claim that individual `CauseNet` relations are unseen by the LLMs. Instead, it mitigates recall of complete public benchmark graphs by randomising topology and composing many local semantic relations into novel DAGs. Accordingly, stronger results on standard `bnlearn` networks should be interpreted cautiously, since they are consistent with possible benchmark memorisation, whereas `CauseNet` is intended to reduce full-graph leakage.

Sampling Observational Data. For structure-learning experiments, we instantiate each grounded DAG as a BN and sample observational data from it. We randomly initialise CPTs since we evaluate structure recovery rather than effect sizes; if needed, CPT parameters can be chosen to better reflect the semantic grounding (details in App. B.3).

6 EXPERIMENTAL EVALUATION

We now present an empirical evaluation of our LLM-augmented Causal ABA framework. Our experiments aim at assessing the accuracy and robustness of the inferred causal graphs, particularly in comparison to established causal discovery algorithms and under varying conditions of LLM input quality. Our goal is to use semantic constraints as additional information to orient edges within the MEC; we therefore evaluate the final output against the ground-truth DAG. Additionally, we investigate the interaction between semantic and statistical constraints, and how this interplay affects the discovery process.

6.1 EXPERIMENTAL SETUP

Data. We evaluate our approach on both standard causal discovery benchmarks and our novel synthetic datasets derived from `CauseNet`. For each dataset, we generate 5000 observational samples and repeat the experiment 50 times with different random seeds. LLM-derived constraints are generated once per dataset from the metadata and reused across all 50 data repetitions. We use standard pseudo-real BNs from the `bnlearn` repository (Scutari, 2010), specifically: *cancer, earthquake, survey, asia, sachs* and *child*. Our synthetic datasets consist of 54 random DAGs with number of nodes $|\mathbf{V}| \in \{5, 10, 15\}$, generated using the protocol described in §5. Detailed statistics of all datasets are provided in App. B, and the generated synthetic BNs and full experimental code are available in our GitHub repository.

Metadata Preparation and Prompting. We generated variable descriptions for causal graphs using an LLM (LLM 2 in Fig. 1), providing it with the ground-truth graph as a basis for defining each variable's role. The prompt was carefully designed to ensure the resulting descriptions were semantically consistent with the underlying causal structure without explicitly stating any of the causal links. The full prompt is provided in App. A.4. Because these synthetic descriptions are LLM-generated from a ground-truth graph-level description, they should be viewed as a controlled upper-information condition rather than realistic documentation. Names-only ablations in the appendix (Tabs. 7 and 8) provide a lower-information stress test; noisy, human-written, and obfuscated metadata are important future robustness tests. We then used these variable names and descriptions as input to our constraint elicitation pipeline (§4) to generate the LLM-derived constraints (LLM 1 in Fig. 1). We use two relatively small but mainstream LLMs, Google's `gemini-2.5-flash` and OpenAI `gpt-5-mini`, with the same prompts and five-run unanimity rule; `gemini-2.5-flash-lite` is used for structured output extraction (LLM 3 in Fig. 1). We leveraged Anthropic's prompt improver (Anthropic, 2024) to iteratively refine the prompts before running both primary models. We evaluate `gemini-2.5-flash` and `gpt-5-mini` under the same consensus protocol; coverage counts are reported in App. A.3. We use the same variable names and descriptions for all LLM-based methods.

Metrics. We measure the accuracy of the estimated causal graphs using several metrics: Structural Hamming Distance (Tsamardinos et al., 2006, SHD), Structural Intervention Distance (Peters and Bühlmann, 2015, SID), precision, recall and F1-score. SHD quantifies the number of edge modifications needed to transform the estimated graph into the true graph, while SID measures the correctness of estimated causal effects under interventions. The F1-score is the harmonic mean of precision and recall with respect to the classification of arrows in the estimated graph against the ground-truth DAG. SID is only used for the `bnlearn` datasets, as it requires plausible BN parameters that are not available for the `CauseNet` graphs. Details in App. C.1.

Baselines. We compare ABAPC-LLM against several baselines: ABAPC (Russo et al., 2024), MPC (Colombo and Maathuis, 2014), FGS (Ramsey et al., 2017), NOTEARS-MLP (Zheng et al., 2020), GRaSP (Lam et al., 2022), and BOSS (Andrews et al., 2023). ABAPC and MPC share our

Table 1: Structural metrics on `CauseNet` synthetic datasets, grouped by node count $|\mathbf{V}|$. Among repeated-run methods, bold marks the best mean and statistically tied methods under BH-corrected Welch tests within each metric/column (Tabs. 10-11, App. C.4.3). Significance levels for the p-values against the next non-bold method in the ranking are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$. LLM-BFS is shown from one saved run and is excluded from Welch tests and bolding.

	Method	SHD ↓	Precision ↑	Recall ↑	F1 ↑
$\|\mathbf{V}\| = 5$	Random	6.886 ± 1.478	0.266 ± 0.145	0.384 ± 0.230	0.307 ± 0.171
	FGS	5.684 ± 1.079	0.273 ± 0.198	0.181 ± 0.133	0.209 ± 0.151
	NOTEARS-MLP	5.332 ± 0.008	0.001 ± 0.008	0.000 ± 0.002	0.001 ± 0.003
	GRaSP	3.590 ± 1.404	0.765 ± 0.257	0.346 ± 0.238	0.597 ± 0.125
	BOSS	3.529 ± 1.305	0.778 ± 0.224	0.356 ± 0.226	0.606 ± 0.117
	MPC	3.294 ± 0.643	0.859 ± 0.101	0.371 ± 0.117	0.602 ± 0.060
	MPC-LLM	1.500 ± 0.318	$\mathbf{0.958 \pm 0.031}$ ***	0.710 ± 0.050	0.795 ± 0.037
	ABAPC	2.025 ± 0.549	0.712 ± 0.087	0.620 ± 0.092	0.671 ± 0.084
	ABAPC-LLM	$\mathbf{1.230 \pm 0.369}$ ***	0.882 ± 0.046	$\mathbf{0.763 \pm 0.062}$ ***	$\mathbf{0.812 \pm 0.049}$ *
	LLM-BFS	2.667	0.842	0.559	0.628
$\|\mathbf{V}\| = 10$	Random	28.247 ± 6.281	0.141 ± 0.069	0.302 ± 0.177	0.183 ± 0.091
	FGS	16.362 ± 1.747	0.173 ± 0.114	0.099 ± 0.066	0.123 ± 0.078
	NOTEARS-MLP	12.642 ± 0.122	0.026 ± 0.119	0.002 ± 0.011	0.004 ± 0.021
	GRaSP	8.148 ± 2.752	0.819 ± 0.257	0.389 ± 0.187	0.537 ± 0.176
	BOSS	7.391 ± 2.627	$\mathbf{0.880 \pm 0.197}$ ***	0.439 ± 0.184	0.579 ± 0.173
	MPC	7.055 ± 1.338	0.854 ± 0.107	0.453 ± 0.101	0.579 ± 0.098
	MPC-LLM	6.206 ± 1.023	0.856 ± 0.088	0.511 ± 0.078	0.627 ± 0.077
	ABAPC	6.166 ± 1.442	0.724 ± 0.127	0.535 ± 0.105	0.612 ± 0.110
	ABAPC-LLM	$\mathbf{5.626 \pm 1.176}$ ***	0.788 ± 0.089	$\mathbf{0.565 \pm 0.086}$ ***	$\mathbf{0.653 \pm 0.084}$ ***
	LLM-BFS	13.278	0.505	0.436	0.439
$\|\mathbf{V}\| = 15$	Random	63.766 ± 19.364	0.082 ± 0.036	0.299 ± 0.171	0.122 ± 0.053
	FGS	24.303 ± 2.112	0.097 ± 0.077	0.060 ± 0.047	0.074 ± 0.057
	NOTEARS-MLP	16.982 ± 0.180	0.024 ± 0.121	0.001 ± 0.008	0.002 ± 0.014
	GRaSP	10.422 ± 3.014	0.857 ± 0.217	0.413 ± 0.153	0.544 ± 0.158
	BOSS	9.994 ± 3.010	0.880 ± 0.191	0.431 ± 0.156	0.563 ± 0.162
	MPC	8.809 ± 1.190	$\mathbf{0.936 \pm 0.071}$ ***	0.479 ± 0.069	0.621 ± 0.072
	MPC-LLM	7.874 ± 1.038	$\mathbf{0.929 \pm 0.061}$ ***	0.532 ± 0.061	$\mathbf{0.667 \pm 0.058}$ *
	ABAPC	7.902 ± 1.537	0.762 ± 0.100	0.546 ± 0.087	0.632 ± 0.090
	ABAPC-LLM	$\mathbf{7.550 \pm 1.281}$ **	0.789 ± 0.077	$\mathbf{0.565 \pm 0.074}$ ***	0.654 ± 0.072
	LLM-BFS	18.833	0.438	0.351	0.356

CI-testing setup; comparing ABAPC-LLM to ABAPC isolates the contribution of semantic arrow constraints, while comparing ABAPC to MPC isolates the contribution of the argumentative layer. MPC-LLM injects the same consensus LLM constraints used by ABAPC-LLM as hard background knowledge into MPC through direct background knowledge (Zheng et al., 2024), without the Causal ABA layer, separating direct hard-constraint use in MPC from argumentative integration using Causal ABA. The choice of CI testing method is consistent across MPC/-LLM and ABAPC/-LLM: Wilks G^2 test (Agresti, 2018), with significance level $\alpha = 0.01$. Within ABAPC/-LLM, MPC is only used to compute (an efficient number of) CI tests and their associated p-values. The remaining baselines span score-based, continuous optimisation, and permutation-based search used with standard hyperparameters. We additionally include an LLM-only baseline (Jiralerspong et al., 2024) (LLM-BFS; metadata-only), which isolates LLM performance without CI tests regardless of the method of integration.[2] Finally, we include a random graph generation baseline for a lower bound on performance. Details in App. C.2.

[2]This was run only once per dataset due to API costs.

6.2 RESULTS AND ANALYSIS

We first report structural recovery, then discuss runtimes and the semantic–statistical constraint interaction. Unless explicitly stated otherwise, the LLM-based main-text results use `gemini-2.5-flash`; the matched-datasets comparison (Tab. 9, App. C.4.1) shows that the differences between using the two sets of constraints are often not significant even though `gpt-5-mini` constraint quality is lower.

Structural Learning Performance. Table 1 compares DAG reconstruction on `CauseNet`. ABAPC-LLM has the lowest SHD and highest recall for every node count, and the best F1 at $|\mathbf{V}| = \{5, 10\}$. The $|\mathbf{V}| = 15$ F1 exception is about the integration method: adding the same LLM constraints raises MPC F1 by 0.046, but ABAPC F1 by only 0.022. The graph-type breakdown in App. C.3.1 (Tab. 3) shows that this shift is concentrated in LT graphs, where MPC-LLM gains 0.09 F1 and reduces SHD by 1.91 over MPC, while ABAPC-LLM gains 0.035 F1 and reduces SHD by 0.60 over ABAPC. This matches the integration policy in §4: ABAPC-LLM only uses LLM-required arrows when the reduced skeleton permits them, while MPC-LLM passes them directly as MPC background knowledge. In the $|\mathbf{V}| = 15$ LT cases, 14.4%

of true LLM-required arrow opportunities are blocked by initially wrong CI facts; because blocked arrows are not regrounded after later relaxation, conservative arbitration can leave useful semantic directions unused. Overall, this conservatism preserves the best SHD and recall on average, but gives MPC-LLM an F1 advantage where the direct constraints are especially useful. A significance level ablation (Tab. 4, App. C.3.1) is compatible with the same interaction story: at $\alpha = 0.05$, the looser CI threshold gives MPC-LLM a larger recall boost, making it the top F1 method; ABAPC-LLM still improves over ABAPC, but its conservative arbitration converts less of the additional semantic signal into F1. On `bnlearn`, ABAPC-LLM is also competitive across most structural metrics (Tab. 5, App. C.3.2).

LLM Constraints Quality and Interaction with Data. To better understand how our LLM-augmented pipeline improves structural accuracy, we inspect the quality of the LLM constraints entering Causal ABA in relation to the data-driven ones. For each run we compute: (i) an aggregate F1-score of the consensus-filtered required/forbidden arrow constraints against the ground-truth DAG; (ii) the F1-score of the CI statements ultimately retained by ABAPC (data-only, after progressive relaxation), again against ground-truth; and (iii) the resulting change in final graph F1 after adding the semantic constraints (ΔF1). We then bin both constraint-quality F1 into low/mid/high ranges ([0,0.33), [0.33,0.66), [0.66,1]) and report the mean ΔF1 in each cell.

Fig. 4 shows synergy when the semantic signal is reliable: high-LLM cells are consistently positive. Conversely, low-quality LLM constraints can hurt even with strong CI evidence (Low LLM/High CI: mean $\Delta\text{F1} = -0.122$). Although *required* arrows only orient adjacencies surviving skeleton reduction (§4), LLM constraints are enforced: conflicts are repaired by relaxing additional CI statements, which can weaken the statistical signal. The conflict is often not a single CI fact directly opposing an LLM arrow: orientations are supported by rules across multiple CI tests. In the current implementation, an LLM direction can be defeated when its adjacency has been removed, but it is not calibrated against a composite CI signal supporting the opposite orientation. This failure mode is most visible with variable descriptions: they increase constraint quality and coverage in many cases (Tabs. 7 and 8; App. C.4), but can also introduce misleading context. With names only, negative cells are much weaker (minimum mean $\Delta\text{F1} = -0.013$; Fig. 7a), supporting the interpretation that richer metadata can both help and mislead. On the `bnlearn` benchmarks, the interaction heatmap is almost entirely non-negative and has larger gains in high-LLM cells (e.g., 0.287 for High LLM/High CI; Fig. 6b). This is consistent with the higher apparent quality of LLM-derived constraints on public benchmark networks, but should be interpreted cautiously because those networks are likely represented in pre-training data. See App. C.4 for the metadata and consensus ablations, and App. C.4.2 for the per-size, per-type, and per-dataset heatmaps.

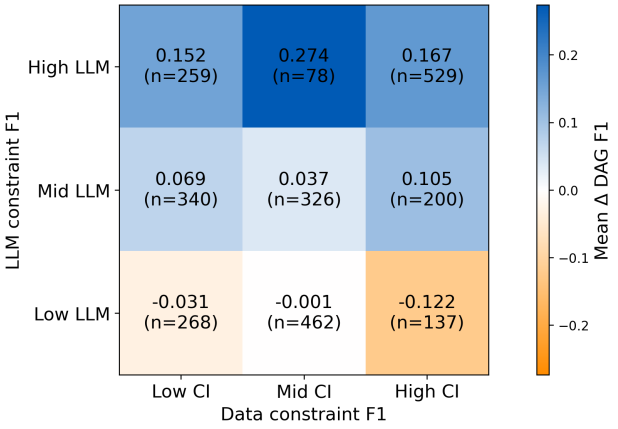


Figure 4: Heatmap of the quality of LLM-derived and data-derived constraints in relation to changes in final graph reconstruction accuracy after integration (ΔF1 against the true DAG; in brackets the number of constraints n).

Runtime. ABAPC-LLM is slower than MPC due to solving the ABAF, but our skeleton-reduction optimisation makes both ABAPC and ABAPC-LLM tractable with less than 30 variables: excluding external API latency (one-off per dataset), runtimes are in seconds up to 15 nodes on synthetic benchmarks and up to 20 variables on `bnlearn`. ABAPC-LLM has comparable runtime to ABAPC on small datasets (5 nodes) but reduces solve time by pruning the search space on bigger ones (Tab. 6, App. C.3.3). The bottleneck is grounding and solving the Causal ABAF: CI literals grow with ordered variable pairs and conditioning sets, path rules grow rapidly with the number of simple paths, and stable-extension reasoning is NP-complete in the grounded ABAF size. Skeleton reduction keeps the evaluated cases tractable, but larger dense skeletons remain a failure mode.

7 CONCLUSION

We introduced a hybrid causal discovery framework that injects LLM-derived semantic knowledge into Causal ABA, combining auditable symbolic conflict resolution with rich and unstructured semantic information. A dedicated elicitation pipeline extracts conservative structural constraints from natural-language metadata, while our `CauseNet`-grounded synthetic benchmarks provide a setting where memorisation is unlikely and improvements in structural, interventional and precision–recall metrics become apparent. Overall, we advocate using LLMs as auditable sources of causal directions, mediated by formal consistency checks.

Our findings show that consensus-filtered semantic constraints can boost structural accuracy complementing statistical evidence, motivating several avenues for future work. In particular, we plan to mine knowledge from scientific corpora, extend the dialogue with LLMs to reason about potential unobserved confounders, and endow Causal ABA with adaptive weighting schemes that calibrate LLM constraints against data-driven ones.

Acknowledgements

Russo was supported by the ERC under the ERC-POC programme (grant number 101189053).

References

567-labs. Instructor: Structured outputs for LLMs. GitHub repository, 2025.

Ahmed Abdulaal, adamos hadjivasiliou, Nina Montana-Brown, Tiantian He, Ayodeji Ijishakin, Ivana Drobnjak, Daniel C. Castro, and Daniel C. Alexander. Causal modelling agents: Causal graph discovery through synergising metadata- and data-driven reasoning. In *Proc. of ICLR 2024*, 2024. OpenReview.

Alan Agresti. *An Introduction to Categorical Data Analysis*. Wiley, 3 edition, 2018.

Bryan Andrews, Joseph Ramsey, Ruben Sanchez Romero, Jazmin Camchong, and Erich Kummerfeld. Fast scalable and accurate discovery of DAGs using the best order score search and grow–shrink trees. In *Proc. of NeurIPS 2023*, 2023. NeurIPS proceedings.

Ankur Ankan and Johannes Textor. Expert-In-The-Loop Causal Discovery: Iterative Model Refinement Using Expert Knowledge. In *Proc. of UAI 2025*, pages 172–183, 2025. PMLR.

Anthropic. Prompt improver, 2024. Anthropic website.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondřej Dušek. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. In *Proc. of EACL 2024*, pages 67–93, 2024. doi: 10.18653/V1/2024.EACL-LONG.5.

T. Ban, L. Chen, D. Lyu, X. Wang, Q. Zhu, Q. Tu, and H. Chen. Integrating large language model for improved causal discovery. *IEEE Trans. on Art. Int.*, 6(11):3030–3042, 2025. doi: 10.1109/tai.2025.3560927.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. doi: 10.1126/science.286.5439.509.

Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *Knowl. Eng. Rev.*, 26(4):365–410, 2011. doi: 10.1017/S0269888911000166.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B Methodol.*, 57(1): 289–300, 1995. JSTOR.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Sanjeev Arora, Sydney von Arx, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Andrei Bondarenko, Phan Minh Dung, Robert A. Kowalski, and Francesca Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artif. Intell.*, 93:63–101, 1997. doi: 10.1016/S0004-3702(97)00015-5.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proc. of NeurIPS 2020*, 2020. NeurIPS proceedings.

Jin Cao and Renxiong Liu. Causal discovery with interactive human inputs. In *Proc. of CSCE 2024*, pages 241–256, 2024. doi: 10.1007/978-3-031-85628-0-18.

Lyuzhou Chen, Taiyu Ban, Xiangyu Wang, Derui Lyu, and Huanhuan Chen. Mitigating prior errors in causal structure learning: A resilient approach via bayesian networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(12):10990–11002, 2025. doi: 10.1109/TPAMI.2025.3594755.

David Maxwell Chickering. Optimal structure identification with greedy search. *J. Mach. Learn. Res.*, 3:507–554, 2002. JMLR.

Kai-Hendrik Cohrs, Emiliano Diaz, Vasileios Sitokonstantinou, Gherardo Varando, and Gustau Camps-Valls. Large language models for constrained-based causal discovery. In *Proc. of AAAI Workshop on Large Language Models and Causality 2024*, 2023. OpenReview.

Diego Colombo and Marloes H. Maathuis. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.*, 15(1):3741–3782, 2014. JMLR.

Gregory F. Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Mach. Learn.*, 9(4):309–347, 1992. doi: 10.1007/BF00994110.

L.P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub)graph isomorphism algorithm for matching large graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(10): 1367–1372, 2004. doi: 10.1109/TPAMI.2004.75.

Kristijonas Cyras, Xiuyi Fan, Claudia Schulz, and Francesca Toni. Assumption-based argumentation: Disputes, explanations, preferences. *FLAP*, 4(8), 2017. FLAP.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of ACL 2024*, pages 12039–12050, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.716.

Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2): 321–358, 1995. doi: 10.1016/0004-3702(94)00041-X.

P Erdös and A Rényi. On random graphs i. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Multi-shot ASP solving with clingo. *Theory Pract. Log. Program.*, 19(1):27–82, 2019. doi: 10.1017/S1471068418000054.

John Gkountouras, Matthias Lindemann, Phillip Lippe, Efstratios Gavves, and Ivan Titov. Language agents meet causality – bridging llms and causal world models. In *Proc. of ICLR 2025*, 2025. OpenReview.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Front. Genet.*, 10:524, 2019. doi: 10.3389/fgene.2019.00524.

Alexander Havrilla, David Alvarez-Melis, and Nicolo Fusi. IGDA: Interactive graph discovery through large language model agents. In *Proc. of the ICLR 2025 Workshop on Reasoning and Planning for Large Language Models*, 2025. OpenReview.

David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Mach. Learn.*, 20(3):197–243, 1995. doi: 10.1023/A:1022623210503.

Stefan Heindorf, Yan Scholten, Henning Wachsmuth, Axel-Cyrille Ngonga Ngomo, and Martin Potthast. Causenet: Towards a causality graph extracted from the web. In *Proc. of CIKM 2020*, pages 3023–3030, 2020. doi: 10.1145/3340531.3412763.

Antti Hyttinen, Frederick Eberhardt, and Matti Järvisalo. Constraint-based causal discovery: Conflict resolution with answer set programming. In *Proc. of UAI 2014*, pages 340–349, 2014.

Ziwei Ji, Nanyun Lee, Tianyi Yu, Qihui He, Ruibo Zhang, Xiang Ren, et al. A survey on hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):1–38, 2023. doi: 10.1145/357173.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. CLadder: Assessing causal reasoning in language models. In *Proc. of NeurIPS 2023*, 2023. OpenReview.

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

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

Emre Kıcıman, Robert Ness, Amit Sharma, and Chenhao Tan. Causal reasoning and large language models: Opening a new frontier for causality. *Trans. Mach. Learn. Res.*, 2024. OpenReview.

Neville K. Kitson and Anthony C. Constantinou. Causal discovery using dynamically requested knowledge. *Knowl. Based Syst.*, 314:113185, 2025. doi: 10.1016/j.knosys.2025.113185.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Proc. of NeurIPS 2022*, 2022. NeurIPS proceedings.

Wai-Yin Lam, Bryan Andrews, and Joseph Ramsey. Greedy relaxations of the sparsest permutation algorithm. In *Proc. of UAI 2022*, 2022. PMLR.

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *J. R. Stat. Soc. Ser. B Methodol.*, 50(2):157–194, 1988. doi: 10.1111/j.2517-6161.1988.tb01721.x.

Chenxi Liu, Yongqiang Chen, Tongliang Liu, Mingming Gong, James Cheng, Bo Han, and Kun Zhang. Discovery of the hidden world with large language models. In *Proc. of NeurIPS 2024*, 2024. NeurIPS proceedings.

Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proc. of UAI 1995*, pages 403–410, 1995. doi: 10.5555/2074158.2074204.

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

J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.

Jonas Peters and Peter Bühlmann. Structural intervention distance for evaluating causal graphs. *Neural Comput.*, 27(3):771–799, 2015. doi: 10.1162/NECO_a_00708.

Joseph D. Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *Int. J. Data Sci. Anal.*, 3(2):121–129, 2017. doi: 10.1007/s41060-016-0032-z.

Alexander G. Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated DAG! causal discovery benchmarks may be easy to game. In *Proc. of NeurIPS 2021*, pages 27772–27784, 2021. NeurIPS proceedings.

Fabrizio Russo, Anna Rapberger, and Francesca Toni. Argumentative causal discovery. In *Proc. of KR 2024*, 2024. doi: 10.24963/KR.2024/88.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005. doi: 10.1126/science.1105809.

Marco Scutari. Learning bayesian networks with the bnlearn R package. *J. Stat. Softw.*, 35(3):1–22, 2010. doi: 10.18637/jss.v035.i03.

C. Spearman. The proof and measurement of association between two things. *Am. J. Psychol.*, 15(1):72–101, 1904. JSTOR.

D. J. Spiegelhalter and R. G. Cowell. Learning in probabilistic expert systems. In *Bayesian Statistics 4*, pp. 447–466, 1992. doi: 10.1093/oso/9780198522669.003.0025.

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

Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. Let Me Speak Freely? A Study On The Impact Of Format Restrictions On Large Language Model Performance. In *Proc. of EMNLP 2024*, pages 1218–1236, 2024. ACL Anthology.

Francesca Toni. A tutorial on assumption-based argumentation. *Argument Comput.*, 5(1):89–117, 2014. doi: 10.1080/19462166.2013.869878.

Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Mach. Learn.*, 65(1):31–78, 2006. doi: 10.1007/s10994-006-6889-7.

Aniket Vashishtha, Abbavaram Gowtham Reddy, Abhinav Kumar, Saketh Bachu, Vineeth N. Balasubramanian, and Amit Sharma. Causal order: The key to leveraging imperfect experts in causal inference. In *Proc. of ICLR 2025*, 2025. OpenReview.

Thomas Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Proc. of UAI 1990*, pages 255–270, 1990. doi: 10.5555/647233.719736.

Matthew J. Vowels, Necati Cihan Camgoz, and Richard Bowden. D'ya Like DAGs? A Survey on Structure Learning and Causal Discovery. *ACM Comput. Surv.*, 55(4):1–35, 2022. doi: 10.1145/3527154.

Guangya Wan, Yunsheng Lu, Yuqi Wu, Mengxuan Hu, and Sheng Li. Large language models for causal discovery: Current landscape and future directions. In *Proc. of IJCAI 2025*, pages 10687–10695, 2025. doi: 10.24963/ijcai.2025/1186.

Xingyu Wu, Kui Yu, Jibin Wu, and Kay Chen Tan. LLM Cannot Discover Causality, and Should Be Restricted to Non-Decisional Support in Causal Discovery. *arXiv preprint arXiv:2506.00844*, 2025.

Alessio Zanga, Elif Ozkirimli, and Fabio Stella. A survey on causal discovery: Theory and practice. *Int. J. Approx. Reason.*, 151:101–129, 2022. doi: 10.1016/j.ijar.2022.09.004.

Matej Zečević, Moritz Willig, Devendra Singh Dhami, and Kristian Kersting. Causal Parrots: Large Language Models May Talk Causality But Are Not Causal. *Trans. Mach. Learn. Res.*, 2023. OpenReview.

Zirui Zhao, Wee Sun Lee, and David Hsu. Large language models as commonsense knowledge for large-scale task planning. In *Proc. of NeurIPS 2023*, 2023. NeurIPS proceedings.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with NO TEARS: continuous imization for structure learning. In *Proc. of NeurIPS 2018*, pages 9492–9503, 2018. NeurIPS proceedings.

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Learning sparse nonparametric dags. In *Proc. of AISTATS 2020*, volume 108 of *PMLR*, pages 3414–3425, 2020. PMLR.

Yujia Zheng, Biwei Huang, Wei Chen, Joseph Ramsey, Mingming Gong, Ruichu Cai, Shohei Shimizu, Peter Spirtes, and Kun Zhang. Causal-learn: Causal discovery in python. *J. Mach. Learn. Res.*, 25(60):1–8, 2024. JMLR.

Leveraging Large Language Models for Causal Discovery: a Constraint-based, Argumentation-driven Approach (Supplementary Material)

Zihao Li[1] **Fabrizio Russo**[1]

[1]Department of Computing,

Imperial College London, UK,

zihao.li24@imperial.ac.uk, fabrizio@imperial.ac.uk

In this supplementary material we provide details omitted from the main text due to space limitations. In particular, we describe the prompts used for metadata enrichment and constraint elicitation (App. A), the consensus strategy for stabilising LLM outputs, the generation and semantic grounding of the synthetic CauseNet datasets (App. B), and the full experimental protocol, metrics, baselines, ablations, and additional quantitative analyses (App. C).

A PROMPTING DETAILS

This section covers the prompts used for the different tasks in which LLMs are involved, see Fig. 1 in the main text for the pipeline overview. The following prompts were built with various iterations of Anthropic's prompt improver tool (Anthropic, 2024).

A.1 LLM CONSTRAINTS ELICITATION PROMPT

Below, we provide the main prompt used in our proposed LLM elicitation pipeline. Note that this prompt significantly differs from prior work on LLMs and causal discovery, e.g. (Jiralerspong et al., 2024; Kıcıman et al., 2024), since it involves a single call for all variables in the dataset. This is enabled by the much longer context window of modern LLMs (e.g., over 1 million tokens for `gemini-2.5-flash`). This single-call approach is not only more computationally efficient but also allows the LLM to reason holistically about the entire system of variables, potentially capturing higher-order interactions that pairwise queries might miss.

```
You will be analyzing a set of causal variables to determine the required and forbidden
    causal directions between them. Your goal is to achieve very high precision in
    identifying these relationships.

<causal_variables>
{CAUSAL_VARIABLES}
</causal_variables>

Your task is to analyze these causal variables and generate two sets:
1. **Required directions**: Causal relationships that must exist based on logical
    necessity, temporal ordering, or fundamental causal principles
2. **Forbidden directions**: Causal relationships that cannot exist due to logical
    impossibility, temporal constraints, or definitional contradictions

Here are the key principles to follow:

**Required directions** should include:
- Relationships where one variable definitionally or logically must cause another
- Temporal precedence relationships (causes must precede effects)
```

```
- Relationships where the causal mechanism is well-established and unavoidable

**Forbidden directions** should include:
- Relationships that would violate temporal ordering (effects cannot cause their own
    causes)
- Relationships that are logically contradictory or definitionally impossible
- Relationships where the direction would violate established causal mechanisms

**Important guidelines:**
- Only include relationships where you have very high confidence
- When in doubt about a relationship, do not include it in either set
- Consider both direct and indirect causal pathways
- Be precise about the direction of causality (A → B is different from B → A)

Format your final answer as follows:

**Required Directions:**
- [Variable A] → [Variable B]: [Brief justification]
- [Continue for all required directions]

**Forbidden Directions:**
- [Variable C] → [Variable D]: [Brief justification]
- [Continue for all forbidden directions]

Your final answer should only include the Required Directions and Forbidden Directions
    sections with their respective causal relationships and justifications. Aim for very
    high precision - only include relationships where you are highly confident in the
    causal direction requirement or prohibition.
```

A.2 SCHEMA-GUIDED EXTRACTION OF CONSTRAINTS

To ensure reliability and avoid constraining the LLM's reasoning process with rigid formatting requirements (Tam et al., 2024), we decouple constraint generation from structured output extraction. The primary LLM produces a semi-structured natural-language answer with required/forbidden directions and brief rationales; we then use a schema-enforcing tool (567-labs, 2025) with a lightweight model (`gemini-2.5-flash-lite`, LLM 3 in Fig. 1) to map this text into typed required/forbidden arrow lists. The extracted constraints are validated and normalised (e.g., variable-name matching and de-duplication) before downstream consensus filtering and integration.

A.3 LLM CONSENSUS DETAILS

The consensus mechanism is designed to generate high-precision causal constraints by aggregating the outputs from multiple independent LLM queries. This is due to the inherent stochasticity of LLMs, and inspired by the majority voting strategy proposed by (Cohrs et al., 2023). Both `gemini-2.5-flash` and `gpt-5-mini` use the same elicitation prompt and the same five-run unanimity rule.

For each causal discovery problem, the LLM is queried five times, yielding five distinct sets of constraints. Let R_i and F_i represent the set of "required" and "forbidden" arrows, respectively, from run $i \in \{1, \ldots, 5\}$. The final high-confidence consensus sets are obtained by taking the intersection across the five runs: $R_{\text{consensus}} = \bigcap_{i=1}^{5} R_i$ and $F_{\text{consensus}} = \bigcap_{i=1}^{5} F_i$. This guarantees that an arrow appears in the final constraints only if it was proposed unanimously across all independent evaluations. The repetition count of five was chosen empirically to balance precision and recall, with enough repeats to filter out stochastic or low-confidence suggestions while avoiding excessive computational cost. The conservative intersection rule substantially improves the precision of the resulting constraints, as empirically validated in Appendix C.4. Across the 54 synthetic datasets, `gemini-2.5-flash` yields 476 unanimous constraints and non-empty consensus sets for 53 datasets, while `gpt-5-mini` yields 129 unanimous constraints and non-empty consensus sets for 43 datasets.

A.4 METADATA ENRICHMENT PROMPT (ADDING VARIABLES' DESCRIPTIONS)

Below, we provide the prompt used for eliciting the variable descriptions before interrogating the main LLM (LLM 1 in Fig. 1) about causal relations amongst variables. The following prompt is related to LLM 2 in Fig. 1, which we deem an optional block as its usefulness may depend on context. Ablations on the importance of variable descriptions are in §4. As noted in the main text, for synthetic data, this prompt receives a graph-level description generated from the ground-truth graph, making the resulting metadata a controlled high-information condition rather than a realistic documentation setting. Names-only, noisy, human-written, and obfuscated metadata are separate robustness regimes.

```
You are tasked with generating descriptions for causal variables in a randomly generated
    causal graph used for synthetic dataset evaluations of causal discovery algorithms.
    Your goal is to create meaningful descriptions for each variable that accurately
    reflect their role in the graph without revealing their relationships with other
    variables.

Here is the description of the randomly generated causal graph:'

<causal_graph>
{CAUSAL_GRAPH_DESCRIPTION}
</causal_graph>

To complete this task, follow these steps:

1. Carefully read and analyze the causal graph description.

2. For each variable mentioned in the graph, create a description that:
   a) Precisely describes its meaning within the context of the graph
   b) Does NOT spoil its relationships with other variables, either explicitly or
       implicitly
   c) Maintains a consistent context or scenario across all variable descriptions

3. If a variable has different contextual meanings in its relationships with other
    variables, provide a general description that could encompass these different meanings
     without revealing the specific relationships.

4. After generating all variable descriptions, assess the quality of the random causal
    graph based on how well the variables' meanings align with each other and how
    realistic the overall scenario is.

5. Present your output in the following format:

<title>
[Provide a concise title for the causal graph]
</title>

<variable_descriptions>
[List each variable and its description]
</variable_descriptions>

<graph_quality_assessment>
[Provide your assessment of the graph quality, including how well the variables' meanings
    align and how realistic the overall scenario is]
</graph_quality_assessment>

Remember, your primary goal is to create meaningful descriptions without revealing any
    causal relationships. Be creative in developing a consistent context that could
    plausibly connect all the variables.
```

B GRAPH AND DATA GENERATION DETAILS

The generation of our synthetic datasets is a two-stage process designed to create structurally diverse and semantically plausible causal graphs. First, we generate a structural scaffold (a DAG) using one of three different topology generation methods (§B.1). Second, we ground this abstract structure in real-world concepts by finding a matching sub-graph within the `CauseNet` knowledge graph, guided by specific heuristics (§B.2). We provide an open-source UI implementing this pipeline at https://clarg-group.github.io/CauseNet-graph-generator/, with the full code included in our repository. In §B.3, we describe the schema used to generate the synthetic datasets and sample observational data from the grounded DAGs, while in §B.4 we provide details on the `bnlearn` datasets used in the experiments within this supplementary material.

B.1 STRUCTURAL SCAFFOLDING AND GRAPH TYPES

We generate the initial DAG topologies using three distinct methods to ensure a variety of graph structures:

- **Erdős-Rényi (ER) and Scale-Free (SF):** These two methods are adapted from the well-established NOTEARS codebase (Zheng et al., 2018). The ER model (Erdös and Rényi, 1959) produces graphs with a random, uniform edge distribution, while the SF model (Barabási and Albert, 1999) generates graphs with power-law degree distributions, characterised by a few high-degree "hub" nodes.
- **Lower Triangle (LT):** This custom method constructs a connected DAG by directly populating a lower triangular adjacency matrix. The algorithm proceeds in two stages. First, to guarantee connectivity, it ensures every node from 1 to $n-1$ has at least one parent by randomly adding a single incoming edge for each from a node with a lower index. Second, it distributes the remaining specified number of edges by randomly placing them in the unoccupied positions of the lower triangle. This two-step process ensures both connectivity and the desired edge density.

While the ER and SF implementations from NOTEARS can produce graphs with a slightly varied number of nodes and edges, the LT method allows for precise control over these parameters.

B.2 SEMANTIC GROUNDING AND HEURISTICS

Once a structural scaffold is generated, we imbue it with semantic meaning by finding an isomorphic sub-graph within the `CauseNet` knowledge graph (Heindorf et al., 2020). Since multiple isomorphic sub-graphs can exist, we employ one of three heuristics to select the most suitable candidate match.

- **`none`:** This baseline heuristic serves as a control. It performs no optimisation and simply selects the first isomorphic sub-graph returned by the search algorithm.
- **`degrees`:** This heuristic aims to find the most specific and least central concepts. It selects the candidate sub-graph that minimises the sum of the in-degrees and out-degrees of all its nodes within the larger `CauseNet` graph. The intuition is that nodes with lower total degrees represent more niche concepts.
- **`semantics`:** This heuristic, as detailed in the main text, selects the candidate sub-graph that minimises a weighted score over three metrics. For a given candidate subgraph with concept (node) set $\mathcal{S}_c$, the metrics are:
  - **Semantic Compactness ($H_{compact}$):** Measures thematic coherence by calculating the average cosine distance of each concept's vector embedding from their geometric centroid:

    $$H_{compact}(\mathcal{S}_c) = \text{avg}_{v \in \mathcal{S}_c} \left(d_{\cos}(\text{emb}(v), \boldsymbol{\mu}_{\mathcal{S}_c}) \right)$$

    where $\text{emb}(v)$ is the vector embedding of concept v, $\boldsymbol{\mu}_{\mathcal{S}_c}$ is the centroid vector of all concept embeddings in the set $\mathcal{S}_c$, and $d_{\cos}$ is the cosine distance.
  - **Node Specificity (H_{spec}):** Penalises overly general "hub" nodes using the average log-degree of each node in the full `CauseNet` graph:

    $$H_{spec}(\mathcal{S}_c) = \text{avg}_{v \in \mathcal{S}_c} \left(\log(\deg(v) + 1) \right)$$

    where $\deg(v)$ is the degree (sum of in- and out-degrees) of node v in the full `CauseNet` graph. The logarithm dampens the penalty for extremely high-degree nodes.

– **Structural-Semantic Correlation (H_{corr}):** Ensures alignment between graph topology and semantic relationships. The score to be minimised is based on the Spearman's rank correlation:

$$H_{corr}(\mathcal{S}_c) = 1 - \text{SpearmanCorr}(d_{\text{graph}}, d_{\text{sem}})$$

where d_{graph} is the set of all pairwise shortest-path distances between nodes in the candidate graph, and d_{sem} is the corresponding set of pairwise cosine distances between their semantic embeddings. A high correlation (low score) indicates that structurally close nodes are also semantically close.

The final score to be minimised is a weighted sum $H_{final} = w_1 H_{compact} + w_2 H_{spec} + w_3 H_{corr}$, where the weights w_i are customisable. In the generation code, the `semantics` heuristic defaults to equal weights $(w_1, w_2, w_3) = (1, 1, 1)$.

Additionally, our implementation supports user-defined heuristics, allowing researchers to specify custom scoring functions to rank candidate sub-graphs according to their own criteria or domain-specific preferences.

B.3 CAUSENET DATASETS GENERATION SCHEMA

The full suite of 54 synthetic datasets was generated by creating one graph for each unique combination of the following parameters:

- **Number of Nodes:** {5, 10, 15}
- **Graph Density:** Two levels of edge counts for each node size:
  - **5 nodes:** 5 and 7 edges
  - **10 nodes:** 10 and 15 edges
  - **15 nodes:** 15 and 22 edges
- **Graph Type:** {ER, SF, LT}
- **Heuristic:** {none, degrees, semantics}

Some examples of the generated graphs are shown in Fig. 5.

B.3.1 CPT and Data Generation

After a semantically grounded DAG is finalised, we use the `pyAgrum` library to construct the corresponding Bayesian Network. For each variable in the BN, the domain size is set to 2, making all variables binary. The Conditional Probability Tables (CPTs) for the BN are randomly generated during the initialisation process. The full collection of the 54 generated BNs in BIFXML and PNG format is available in the project's public repository, allowing for full reproducibility of our experiments.

The randomly initialised CPTs are not intended to reflect realistic effect sizes or context-specific independencies. Their sole purpose is to generate observational data that respects the sampled graph structure. Our evaluation targets structural recovery (presence and orientation of edges), not the calibration of causal effect magnitudes. Mismatches between semantic strength and statistical effect size are therefore expected and mirror low-signal regimes encountered in practice.

Table 2: Details of the `bnlearn` benchmark networks.

Dataset Name	$\lvert\mathbf{V}\rvert$	$\lvert E\rvert$	$\lvert E\rvert/\lvert\mathbf{V}\rvert$
CANCER	5	4	0.8
EARTHQUAKE	5	4	0.8
SURVEY	6	6	1
ASIA	8	8	1
SACHS	11	17	1.55
CHILD	20	25	1.25

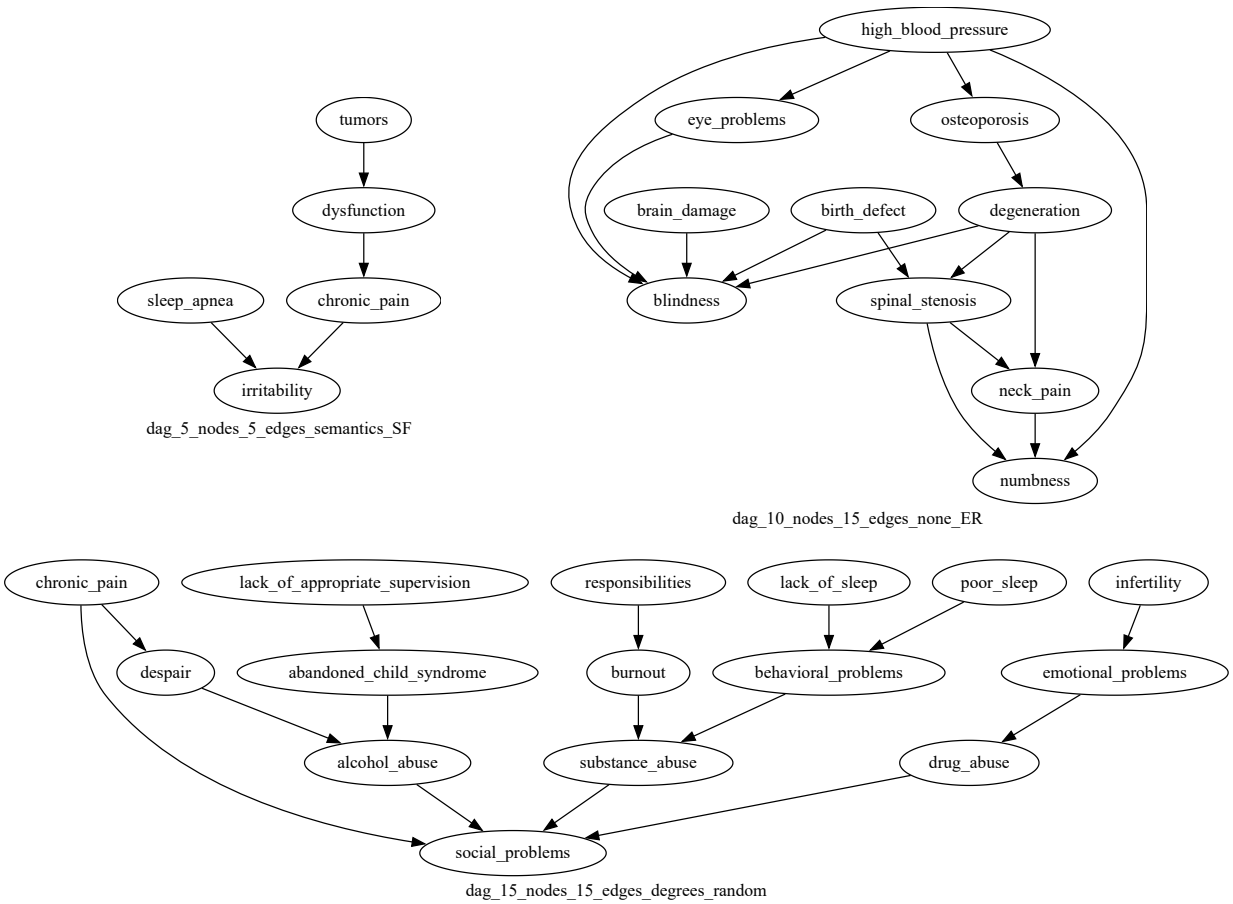


Figure 5: Examples of semantically grounded DAGs from our synthetic data pipeline, generated with different structural methods (ER, SF, LT) and semantic heuristics (`none`, `degrees`, `semantics`).

B.4 BNLEARN DATASETS GENERATION

For the complementary benchmark evaluation (see §6), we use six discrete Bayesian networks from the `bnlearn` repository (Scutari, 2010).[1] The repository is widely used in causal-discovery and Bayesian-network structure-learning work, and collects standard toy, expert-designed, and experimentally derived networks. Specifically, we use `asia`, `cancer`, `earthquake`, `survey`, `sachs`, and `child` (Table 2). This set includes small diagnostic or illustrative networks, the Sachs protein-signalling network (Sachs et al., 2005), and the Child diagnosis network (Spiegelhalter and Cowell, 1992); the Asia network traces back to the classic expert-system example of Lauritzen and Spiegelhalter (1988). For each network, we load the BIF structure and associated conditional probability tables from the repository and sample 5000 observations with 50 different random seeds to measure variance and confidence intervals.

C DETAILS ON EXPERIMENTS

We provide here additional details on the experimental setup, metrics, and results that were omitted from the main text due to space constraints.

C.1 METRICS DEFINITIONS

We evaluate the performance of causal discovery algorithms using standard metrics that capture different aspects of structural accuracy. Let $G_{true} = (V, E_{true})$ be the ground truth and $G_{est} = (V, E_{est})$ the estimated DAG.

[1] `https://www.bnlearn.com/bnrepository/`

- **Structural Hamming Distance (SHD):** The SHD (Tsamardinos et al., 2006) measures the structural difference between two graphs. It is defined as the number of edge operations (additions, deletions, or reversals) required to transform G_{est} into G_{true}. A lower SHD indicates a better structural match.
- **Structural Intervention Distance (SID):** The SID (Peters and Bühlmann, 2015) is a metric that quantifies the difference in the interventional distributions implied by two causal graphs. It counts the number of pairs (i, j) for which the causal effect of intervening on node i on node j is incorrectly predicted by G_{est} compared to G_{true}. A lower SID indicates a better match in terms of causal predictions.
- **Precision, Recall, and F1-Score:** These metrics evaluate the accuracy of the learned directed edge set. Let TP (True Positives) be the number of true arrows recovered with the correct orientation, FP (False Positives) be the number of predicted arrows not present in the true directed edge set, and FN (False Negatives) be the number of true arrows not recovered with the correct orientation.
  - **Precision** is the fraction of correctly identified arrows among all arrows in the estimated graph: Precision $= \frac{\text{TP}}{\text{TP+FP}}$.
  - **Recall** is the fraction of true arrows that were correctly identified: Recall $= \frac{\text{TP}}{\text{TP+FN}}$.
  - **F1-Score** is the harmonic mean of Precision and Recall: $F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$.

C.2 BASELINES

We used the following nine baselines with respective implementations (see §6 for context):

- A Random baseline (RND) as in (Russo et al., 2024), by sampling 10 random ER DAGs with the same number of nodes d as the ground truth. For each draw, the number of edges is sampled as $S \sim \text{Unif}\{d, \ldots, d(d-1)/2\}$; conditional on $S = s$, the undirected skeleton is sampled uniformly from all s-edge subsets of the $\binom{d}{2}$ possible unordered pairs, and then oriented acyclically by a uniformly random node ordering, i.e. for order π, $\{i, j\}$ is oriented as $i \to j$ iff $\pi(i) > \pi(j)$.
- Fast Greedy Equivalence Search[2] (FGS) (Ramsey et al., 2017) is a score-based causal discovery algorithm. It is a fast implementation of GES (Chickering, 2002): graphs are evaluated using the Bayesian Information Criterion (BIC) as edges are greedily inserted and removed in forward and backward phases.
- NOTEARS-MLP[3] (NT) (Zheng et al., 2020) learns a non-linear SEM via continuous optimisation. Having a Multi-Layer Perceptron (MLP) at its core, this method should adapt to different functional dependencies among the variables. The optimisation is carried out via augmented Lagrangian with a continuous formulation of acyclicity (Zheng et al., 2018).
- GRaSP[4] (Lam et al., 2022) is a permutation-based method that searches over variable orderings as a scalable relaxation of the sparsest permutation criterion. For each candidate permutation it constructs a subgraph-minimal DAG by selecting parents only among preceding variables (using a grow–shrink strategy) and scores the projected DAG via BIC; it then greedily explores neighbouring permutations via *tuck* operations (a generalisation of local transpositions) to improve sparsity/score.
- BOSS[5] (Andrews et al., 2023) is a permutation-based score search that greedily traverses variable orderings, projects each ordering to a subgraph-minimal DAG using grow–shrink, and returns the MEC of the highest-scoring projected DAG (BIC). To scale to larger graphs, it uses grow–shrink trees to cache intermediate computations across permutations, avoiding redundant scoring.
- Majority-PC[6] (MPC) (Colombo and Maathuis, 2014) is a constraint-based causal discovery algorithm. It uses CI tests and graphical rules based on d-separation to extract a CPDAG from the data. MPC is an improved version of the original Peter-Clark (PC) algorithm (Spirtes et al., 2000) that renders it order-independent while maintaining soundness and completeness with infinite data. This constitutes the statistical engine underlying ABAPC, isolating the impact of Causal ABA.

[2] https://github.com/bd2kccd/py-causal

[3] https://github.com/xunzheng/notears

[4] https://github.com/py-why/causal-learn/blob/main/causallearn/search/PermutationBased/GRaSP.py

[5] https://github.com/py-why/causal-learn/blob/main/causallearn/search/PermutationBased/BOSS.py

[6] https://github.com/briziorusso/ArgCausalDisco/blob/llm-uai2026-freeze/cd_algorithms/PC.py

- MPC-LLM[7] runs MPC with the same consensus required/forbidden arrow constraints supplied through the hard background-knowledge functionality of `causal-learn` (Zheng et al., 2024). This enforces the specified directions and prohibitions in the returned graph, with no subsequent arbitration or relaxation step.
- ABAPC[8] (ABAPC) (Russo et al., 2024) is an instantiation of Causal ABA, a logic- and constraint-based causal discovery algorithm that uses an Assumption-Based Argumentation framework to resolve conflicts and ambiguities within observational data. The method uses the CI tests from the MPC algorithm as inputs into the Causal ABA (hence ABAPC) and excludes a progressively larger number of low-confidence tests to find a stable extension of the ABAF that contains a DAG which is guaranteed to imply the d-separations supported by the retained tests. This is the non-LLM variant of our proposed method, allowing us to isolate the impact of LLM-derived constraints.
- LLM-BFS[9] (BFS) (Jiralerspong et al., 2024) is a breadth-first search-based approach that leverages LLMs for causal discovery. It explores the graph structure by iteratively expanding the search space and incorporating LLM-generated insights to build the causal graph, only leveraging variable names and descriptions. This method is the LLM-only baseline, allowing us to isolate the impact of data-driven constraints.

C.3 ADDITIONAL DETAILS ON EXPERIMENTS AND RESULTS

The main text reports the most compact `CauseNet` DAG view. This appendix gives the complementary breakdowns in the order they are referenced in the main text: additional `CauseNet` structure tables (§C.3.1), `bnlearn` results (§C.3.2), runtime (§C.3.3), LLM-constraint ablations (§C.4), LLM-source comparisons (§C.4.1), and interaction heatmaps (§C.4.2). Statistical-test tables are collected at the end because they support multiple preceding results (§C.4.3).

C.3.1 Additional CauseNet Results

Table 3 refines the main `CauseNet` result by crossing node count with graph type. The average columns show that ABAPC-LLM has the best marginal SHD and F1 for $|\mathbf{V}| = 5$ and $|\mathbf{V}| = 10$, and the best marginal SHD for $|\mathbf{V}| = 15$; the only node-count marginal exception is $|\mathbf{V}| = 15$ F1, where MPC-LLM is higher. Averaging also over graph type and size, ABAPC-LLM remains best overall for both SHD and F1. The individual cells make clear where direct hard constraints are most competitive: MPC-LLM is strong for F1 on larger LT graphs and some $|\mathbf{V}| = 15$ settings, matching the mechanism discussed in the main text. It uses LLM-required arrows directly, whereas ABAPC-LLM only orients them after skeleton compatibility and conflict resolution. Table 4 compares the headline metrics under the main $\alpha = 0.01$ CI threshold and the archived $\alpha = 0.05$ run. Moving to $\alpha = 0.05$ changes the CI-derived skeleton and the downstream compatibility checks faced by semantic constraints. The result is a useful stress test for the integration policy: MPC-LLM becomes the top F1 method at all node counts, while ABAPC-LLM still improves over ABAPC but by a smaller margin, consistent with direct MPC background knowledge extracting more of the usable LLM signal. Specifically, going from $\alpha = 0.01$ to $\alpha = 0.05$, mean relaxed facts increase from 209 to 270 at $|\mathbf{V}| = 10$ and from 255 to 360 at $|\mathbf{V}| = 15$, while retained-CI F1 falls from 0.519 to 0.477 and from 0.505 to 0.455, respectively. A less conservative significance threshold $\alpha = 0.05$ leaves a denser CI-derived skeleton, so more true adjacencies remain available for the final graph. At the same time, the retained CI evidence is less clean: ABAPC-LLM has to drop more CI facts before a stable extension exists, and those retained facts match the ground truth less well. This weakens the benefit of its conservative semantic integration, while MPC-LLM still applies the same LLM constraints directly as hard background knowledge.

[7] `https://github.com/briziorusso/ArgCausalDisco/blob/llm-uai2026-freeze/cd_algorithms/models.py`

[8] `https://github.com/briziorusso/ArgCausalDisco/blob/llm-uai2026-freeze/abapc.py`

[9] `https://github.com/superkaiba/causal-llm-bfs`

Table 3: Structural metrics on `CauseNet` by graph type and $|\mathbf{V}|$, with marginal averages. Among repeated-run methods, bold marks the best mean and statistically tied methods under BH-corrected Welch tests within each metric/column. Significance levels for the p-values against the next non-bold method in the ranking are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$. LLM-BFS is shown from one saved run and is excluded from Welch tests and bolding. Tests: Table 12; tied bolding: Table 11.

		Method	ER	SF	LT	Average
$\|\mathbf{V}\| = 5$	SHD ↓	Random	6.717 ± 1.517	7.133 ± 1.453	6.807 ± 1.465	6.886 ± 1.478
		FGS	6.570 ± 0.972	4.580 ± 0.907	5.903 ± 1.358	5.684 ± 1.079
		NOTEARS-MLP	6.000	3.997 ± 0.023	6.000	5.332 ± 0.008
		GRaSP	3.750 ± 1.553	2.910 ± 1.038	4.110 ± 1.620	3.590 ± 1.404
		BOSS	3.843 ± 1.515	2.740 ± 1.008	4.003 ± 1.392	3.529 ± 1.305
		MPC	2.727 ± 0.658	3.130 ± 0.468	4.027 ± 0.802	3.294 ± 0.643
		MPC-LLM	1.680 ± 0.305	1.603 ± 0.305	$\mathbf{1.217 \pm 0.345}$***	1.500 ± 0.318
		ABAPC	1.515 ± 0.542	1.850 ± 0.383	2.709 ± 0.722	2.025 ± 0.549
		ABAPC-LLM	$\mathbf{1.054 \pm 0.372}$***	$\mathbf{1.302 \pm 0.278}$**	$\mathbf{1.335 \pm 0.457}$***	$\mathbf{1.230 \pm 0.369}$***
		LLM-BFS	2.667	2.000	3.333	2.667
	F1 ↑	Random	0.337 ± 0.178	0.252 ± 0.163	0.332 ± 0.170	0.307 ± 0.171
		FGS	0.175 ± 0.140	0.197 ± 0.157	0.257 ± 0.157	0.209 ± 0.151
		NOTEARS-MLP	0.000	0.002 ± 0.010	0.000	0.001 ± 0.003
		GRaSP	0.612 ± 0.167	0.577 ± 0.042	0.603 ± 0.167	0.597 ± 0.125
		BOSS	0.605 ± 0.152	0.593 ± 0.042	0.620 ± 0.157	0.606 ± 0.117
		MPC	0.692 ± 0.063	0.615 ± 0.007	0.500 ± 0.110	0.602 ± 0.060
		MPC-LLM	0.805 ± 0.025	$\mathbf{0.697 \pm 0.055}$***	$\mathbf{0.883 \pm 0.030}$**	0.795 ± 0.037
		ABAPC	0.766 ± 0.078	0.632 ± 0.075	0.617 ± 0.098	0.671 ± 0.084
		ABAPC-LLM	$\mathbf{0.843 \pm 0.044}$***	$\mathbf{0.734 \pm 0.046}$***	0.858 ± 0.057	$\mathbf{0.812 \pm 0.049}$*
		LLM-BFS	0.653	0.630	0.601	0.628
$\|\mathbf{V}\| = 10$	SHD ↓	Random	27.053 ± 5.917	28.720 ± 6.325	28.967 ± 6.600	28.247 ± 6.281
		FGS	16.350 ± 1.835	15.923 ± 1.612	16.813 ± 1.795	16.362 ± 1.747
		NOTEARS-MLP	12.477 ± 0.137	12.963 ± 0.147	12.487 ± 0.082	12.642 ± 0.122
		GRaSP	7.913 ± 2.340	9.513 ± 2.852	7.017 ± 3.065	8.148 ± 2.752
		BOSS	7.143 ± 2.267	8.530 ± 2.622	6.500 ± 2.993	7.391 ± 2.627
		MPC	7.188 ± 1.275	6.883 ± 1.527	7.093 ± 1.212	7.055 ± 1.338
		MPC-LLM	5.488 ± 0.772	$\mathbf{6.112 \pm 1.198}$	7.017 ± 1.098	6.206 ± 1.023
		ABAPC	5.720 ± 1.161	6.221 ± 1.493	6.557 ± 1.671	6.166 ± 1.442
		ABAPC-LLM	$\mathbf{5.061 \pm 0.791}$**	$\mathbf{5.711 \pm 1.302}$*	$\mathbf{6.105 \pm 1.437}$*	$\mathbf{5.626 \pm 1.176}$***
		LLM-BFS	9.500	17.167	13.167	13.278
	F1 ↑	Random	0.192 ± 0.098	0.183 ± 0.095	0.175 ± 0.078	0.183 ± 0.091
		FGS	0.127 ± 0.087	0.122 ± 0.072	0.120 ± 0.077	0.123 ± 0.078
		NOTEARS-MLP	0.002 ± 0.022	0.007 ± 0.028	0.003 ± 0.013	0.004 ± 0.021
		GRaSP	0.542 ± 0.162	0.467 ± 0.160	0.602 ± 0.205	0.537 ± 0.176
		BOSS	0.588 ± 0.160	0.512 ± 0.167	$\mathbf{0.638 \pm 0.192}$***	0.579 ± 0.173
		MPC	0.560 ± 0.100	0.620 ± 0.098	0.558 ± 0.097	0.579 ± 0.098
		MPC-LLM	$\mathbf{0.663 \pm 0.057}$***	0.657 ± 0.092	0.560 ± 0.083	0.627 ± 0.077
		ABAPC	0.616 ± 0.094	0.643 ± 0.101	0.575 ± 0.135	0.612 ± 0.110
		ABAPC-LLM	$\mathbf{0.656 \pm 0.057}$***	$\mathbf{0.681 \pm 0.086}$**	$\mathbf{0.623 \pm 0.111}$*	$\mathbf{0.653 \pm 0.084}$***
		LLM-BFS	0.448	0.383	0.497	0.439
$\|\mathbf{V}\| = 15$	SHD ↓	Random	62.333 ± 18.323	64.417 ± 20.652	64.547 ± 19.117	63.766 ± 19.364
		FGS	26.243 ± 2.217	20.207 ± 1.870	26.460 ± 2.250	24.303 ± 2.112
		NOTEARS-MLP	18.483 ± 0.090	13.977 ± 0.265	18.487 ± 0.185	16.982 ± 0.180
		GRaSP	11.150 ± 2.942	9.317 ± 2.747	10.800 ± 3.353	10.422 ± 3.014
		BOSS	10.947 ± 3.300	9.067 ± 2.800	9.970 ± 2.930	9.994 ± 3.010
		MPC	9.673 ± 1.293	8.340 ± 0.955	8.413 ± 1.322	8.809 ± 1.190
		MPC-LLM	8.815 ± 1.132	8.307 ± 0.978	$\mathbf{6.502 \pm 1.003}$**	7.874 ± 1.038
		ABAPC	8.843 ± 1.696	$\mathbf{7.103 \pm 1.144}$***	7.761 ± 1.770	7.902 ± 1.537
		ABAPC-LLM	$\mathbf{8.259 \pm 1.245}$*	$\mathbf{7.226 \pm 1.151}$***	7.165 ± 1.447	$\mathbf{7.550 \pm 1.281}$**
		LLM-BFS	18.000	21.500	17.000	18.833
	F1 ↑	Random	0.130 ± 0.055	0.102 ± 0.050	0.133 ± 0.055	0.122 ± 0.053
		FGS	0.085 ± 0.060	0.042 ± 0.055	0.095 ± 0.055	0.074 ± 0.057
		NOTEARS-MLP	0.000 ± 0.008	0.005 ± 0.022	0.000 ± 0.013	0.002 ± 0.014
		GRaSP	0.540 ± 0.150	0.507 ± 0.168	0.585 ± 0.157	0.544 ± 0.158
		BOSS	0.555 ± 0.160	0.515 ± 0.177	0.618 ± 0.148	0.563 ± 0.162
		MPC	0.622 ± 0.075	0.557 ± 0.073	0.685 ± 0.067	0.621 ± 0.072
		MPC-LLM	$\mathbf{0.677 \pm 0.062}$**	0.548 ± 0.072	$\mathbf{0.775 \pm 0.042}$***	$\mathbf{0.667 \pm 0.058}$*
		ABAPC	0.619 ± 0.098	$\mathbf{0.601 \pm 0.080}$***	0.677 ± 0.092	0.632 ± 0.090
		ABAPC-LLM	0.660 ± 0.068	$\mathbf{0.590 \pm 0.077}$***	0.712 ± 0.070	0.654 ± 0.072
		LLM-BFS	0.317	0.339	0.411	0.356
Avg.	SHD ↓	Random	32.034 ± 8.586	33.423 ± 9.477	33.440 ± 9.061	32.966 ± 9.041
		FGS	16.388 ± 1.674	13.570 ± 1.463	16.392 ± 1.801	15.450 ± 1.646
		NOTEARS-MLP	12.320 ± 0.076	10.312 ± 0.145	12.324 ± 0.089	11.652 ± 0.103
		GRaSP	7.604 ± 2.278	7.247 ± 2.212	7.309 ± 2.679	7.387 ± 2.390
		BOSS	7.311 ± 2.361	6.779 ± 2.143	6.824 ± 2.438	6.971 ± 2.314
		MPC	6.529 ± 1.076	6.118 ± 0.983	6.511 ± 1.112	6.386 ± 1.057
		MPC-LLM	5.328 ± 0.736	5.341 ± 0.827	$\mathbf{4.912 \pm 0.816}$***	5.193 ± 0.793
		ABAPC	5.359 ± 1.133	5.058 ± 1.007	5.676 ± 1.387	5.364 ± 1.176
		ABAPC-LLM	$\mathbf{4.791 \pm 0.803}$**	$\mathbf{4.747 \pm 0.910}$*	$\mathbf{4.868 \pm 1.114}$***	$\mathbf{4.802 \pm 0.942}$***
		LLM-BFS	10.056	13.556	11.167	11.593
	F1 ↑	Random	0.219 ± 0.111	0.179 ± 0.103	0.213 ± 0.101	0.204 ± 0.105
		FGS	0.129 ± 0.096	0.120 ± 0.094	0.157 ± 0.096	0.135 ± 0.095
		NOTEARS-MLP	0.001 ± 0.010	0.004 ± 0.020	0.001 ± 0.009	0.002 ± 0.013
		GRaSP	0.564 ± 0.159	0.517 ± 0.123	0.597 ± 0.176	0.559 ± 0.153
		BOSS	0.583 ± 0.157	0.540 ± 0.128	0.626 ± 0.166	0.583 ± 0.150
		MPC	0.624 ± 0.079	0.597 ± 0.059	0.581 ± 0.091	0.601 ± 0.077
		MPC-LLM	$\mathbf{0.715 \pm 0.048}$***	0.634 ± 0.073	$\mathbf{0.739 \pm 0.052}$***	0.696 ± 0.057
		ABAPC	0.667 ± 0.090	0.625 ± 0.085	0.623 ± 0.108	0.638 ± 0.095
		ABAPC-LLM	$\mathbf{0.720 \pm 0.056}$***	$\mathbf{0.668 \pm 0.070}$***	$\mathbf{0.731 \pm 0.079}$***	$\mathbf{0.706 \pm 0.069}$*
		LLM-BFS	0.473	0.451	0.503	0.475

Table 4: Significance threshold ablation on `CauseNet`. The main runs use $\alpha = 0.01$; the comparison uses $\alpha = 0.05$ runs. Among repeated-run methods, bold marks the best mean and statistically tied methods under BH-corrected Welch tests within each metric/column. Significance levels for the p-values against the next non-bold method in the ranking are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$. LLM-BFS is shown from one saved run and is excluded from Welch tests and bolding. Tests: Table 13; tied bolding: Table 11.

α	Metric	Method	$\lvert\mathbf{V}\rvert = 5$	$\lvert\mathbf{V}\rvert = 10$	$\lvert\mathbf{V}\rvert = 15$
$\alpha = 0.01$	SHD ↓	Random	6.886 ± 1.478	28.247 ± 6.281	63.766 ± 19.364
		FGS	5.684 ± 1.079	16.362 ± 1.747	24.303 ± 2.112
		NOTEARS-MLP	5.332 ± 0.008	12.642 ± 0.122	16.982 ± 0.180
		GRaSP	3.590 ± 1.404	8.148 ± 2.752	10.422 ± 3.014
		BOSS	3.529 ± 1.305	7.391 ± 2.627	9.994 ± 3.010
		MPC	3.294 ± 0.643	7.055 ± 1.338	8.809 ± 1.190
		MPC-LLM	1.500 ± 0.318	6.206 ± 1.023	7.874 ± 1.038
		ABAPC	2.025 ± 0.549	6.166 ± 1.442	7.902 ± 1.537
		ABAPC-LLM	$\mathbf{1.230 \pm 0.369}$***	$\mathbf{5.626 \pm 1.176}$***	$\mathbf{7.550 \pm 1.281}$**
		LLM-BFS	2.667	13.278	18.833
	F1 ↑	Random	0.307 ± 0.171	0.183 ± 0.091	0.122 ± 0.053
		FGS	0.209 ± 0.151	0.123 ± 0.078	0.074 ± 0.057
		NOTEARS-MLP	0.001 ± 0.003	0.004 ± 0.021	0.002 ± 0.014
		GRaSP	0.597 ± 0.125	0.537 ± 0.176	0.544 ± 0.158
		BOSS	0.606 ± 0.117	0.579 ± 0.173	0.563 ± 0.162
		MPC	0.602 ± 0.060	0.579 ± 0.098	0.621 ± 0.072
		MPC-LLM	0.795 ± 0.037	0.627 ± 0.077	$\mathbf{0.667 \pm 0.058}$*
		ABAPC	0.671 ± 0.084	0.612 ± 0.110	0.632 ± 0.090
		ABAPC-LLM	$\mathbf{0.812 \pm 0.049}$*	$\mathbf{0.653 \pm 0.084}$***	0.654 ± 0.072
		LLM-BFS	0.628	0.439	0.356
$\alpha = 0.05$	SHD ↓	Random	6.886 ± 1.478	28.247 ± 6.281	63.766 ± 19.364
		FGS	5.684 ± 1.079	16.362 ± 1.747	24.303 ± 2.112
		NOTEARS-MLP	5.332 ± 0.008	12.642 ± 0.122	16.982 ± 0.180
		GRaSP	3.590 ± 1.404	8.148 ± 2.752	10.422 ± 3.014
		BOSS	3.529 ± 1.305	7.391 ± 2.627	9.994 ± 3.010
		MPC	2.913 ± 0.868	5.267 ± 1.791	7.289 ± 1.971
		MPC-LLM	$\mathbf{1.170 \pm 0.583}$***	$\mathbf{4.531 \pm 1.516}$***	$\mathbf{6.439 \pm 1.769}$***
		ABAPC	1.894 ± 0.695	5.833 ± 1.583	7.893 ± 1.882
		ABAPC-LLM	$\mathbf{1.233 \pm 0.514}$***	5.414 ± 1.381	7.504 ± 1.670
		LLM-BFS	2.667	13.278	18.833
	F1 ↑	Random	0.307 ± 0.171	0.183 ± 0.091	0.122 ± 0.053
		FGS	0.209 ± 0.151	0.123 ± 0.078	0.074 ± 0.057
		NOTEARS-MLP	0.001 ± 0.003	0.004 ± 0.021	0.002 ± 0.014
		GRaSP	0.597 ± 0.125	0.537 ± 0.176	0.544 ± 0.158
		BOSS	0.606 ± 0.117	0.579 ± 0.173	0.563 ± 0.162
		MPC	0.653 ± 0.110	0.691 ± 0.131	0.684 ± 0.106
		MPC-LLM	$\mathbf{0.850 \pm 0.073}$***	$\mathbf{0.734 \pm 0.103}$***	$\mathbf{0.721 \pm 0.092}$***
		ABAPC	0.695 ± 0.099	0.636 ± 0.111	0.640 ± 0.103
		ABAPC-LLM	0.817 ± 0.064	0.672 ± 0.093	0.663 ± 0.088
		LLM-BFS	0.628	0.439	0.356

C.3.2 Performance on `bnlearn` Benchmarks

Table 5 reports the full structural metric table for the standard `bnlearn` benchmarks, with $|\mathbf{V}|$ and $|E|$ included in the dataset labels. These benchmarks are useful for comparability with the causal-discovery literature, but they are public and likely present in LLM pre-training data; the results should therefore be interpreted as a complementary pseudo-real benchmark rather than as the main evidence for generalisation.

The pattern is nevertheless informative. ABAPC-LLM is strongest or statistically tied on the larger and more difficult `sachs` and `child` networks, and remains tied with the best group on the easy `earthquake` network where several methods nearly recover the graph. MPC-LLM is best on `asia` and `survey`, and tied with ABAPC-LLM on `cancer`; this mirrors the `CauseNet` story that direct LLM background knowledge can help when the semantic constraints are highly reliable. The benefit of the argumentative layer is clearest where purely hard constraints would otherwise over-commit or where many statistical and semantic facts must be reconciled.

The LLM-only baseline achieves very high recall on several of these public networks, but at lower precision; this is useful as a diagnostic, not as a deployable strategy. It suggests that LLM-derived constraints can contain real signal, although maybe confounded by memorisation, while also reinforcing the need for a data-driven and conflict-aware layer before accepting a graph.

Table 5: Structural metrics on `bnlearn` benchmarks; dataset labels report $|\mathbf{V}|$ and $|E|$. Among repeated-run methods, bold marks the best mean and statistically tied methods under BH-corrected Welch tests within each metric/column. Significance levels for the p-values against the next non-bold method in the ranking are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$. LLM-BFS is shown from one saved run and is excluded from Welch tests and bolding. Tests: Tables 14, 15, and 16; tied bolding: Table 17.

Dataset	Method	SHD ↓	SID ↓	Precision ↑	Recall ↑	F1 ↑
`cancer` $\lvert\mathbf{V}\rvert = 5$ $\lvert E\rvert = 4$	Random	5.600 ± 1.650	11.260 ± 4.400	0.200 ± 0.220	0.200 ± 0.220	0.360 ± 0.160
	FGS	2.320 ± 0.470	11.280 ± 1.880	0.450 ± 0.080	0.420 ± 0.120	0.430 ± 0.100
	NOTEARS-MLP	3.980 ± 0.140	9.980 ± 0.620	0.250 ± 0.500	0.000 ± 0.040	0.400
	GRaSP	3.800 ± 0.760	9.740 ± 1.880	0.580 ± 0.370	0.050 ± 0.190	0.660 ± 0.280
	BOSS	3.600 ± 1.110	9.160 ± 2.620	0.860 ± 0.240	0.100 ± 0.280	0.770 ± 0.300
	MPC	3.420 ± 1.400	8.520 ± 3.520	0.940 ± 0.180	0.150 ± 0.350	**0.940 ± 0.180** **
	MPC-LLM	**0.680 ± 0.590** ***	**0.780 ± 1.000** ***	**0.990 ± 0.050** .	**0.840 ± 0.130** ***	**0.900 ± 0.080** **
	ABAPC	2.106 ± 1.148	8.070 ± 4.491	0.556 ± 0.272	0.478 ± 0.276	0.604 ± 0.179
	ABAPC-LLM	**0.680 ± 0.587** ***	**0.960 ± 1.324** ***	**0.996 ± 0.028** .	**0.835 ± 0.148** ***	**0.901 ± 0.091** **
	LLM-BFS	4.000	0.000	0.500	1.000	0.667
`earthquake` $\lvert\mathbf{V}\rvert = 5$ $\lvert E\rvert = 4$	Random	5.700 ± 1.420	10.560 ± 3.550	0.170 ± 0.160	0.170 ± 0.160	0.290 ± 0.100
	FGS	2.620 ± 0.640	9.940 ± 0.240	0.440 ± 0.060	0.500	0.470 ± 0.030
	NOTEARS-MLP	2.120 ± 1.170	9.760 ± 5.380	0.500 ± 0.270	0.500 ± 0.280	0.500 ± 0.270
	GRaSP	**0.160 ± 0.790** ***	**0.400 ± 1.980**	**1.000** ***	**0.960 ± 0.200**	**1.000** ***
	BOSS	**0.400 ± 1.210** ***	1.000 ± 3.030	**1.000** ***	0.900 ± 0.300	**1.000** ***
	MPC	**0.040 ± 0.200** ***	**0.000** *	**0.990 ± 0.040** ***	**1.000** *	**1.000 ± 0.020** ***
	MPC-LLM	**0.040 ± 0.200** ***	**0.000** *	**0.990 ± 0.040** ***	**1.000** *	**1.000 ± 0.020** ***
	ABAPC	**0.100 ± 0.463** ***	**0.160 ± 1.131** .	**0.982 ± 0.080** ***	**0.990 ± 0.071** *	**0.986 ± 0.073** ***
	ABAPC-LLM	**0.080 ± 0.340** ***	**0.080 ± 0.566** .	**0.987 ± 0.052** ***	**0.995 ± 0.035** *	**0.991 ± 0.041** ***
	LLM-BFS	4.000	0.000	0.500	1.000	0.667
`survey` $\lvert\mathbf{V}\rvert = 6$ $\lvert E\rvert = 6$	Random	7.980 ± 1.770	17.540 ± 6.060	0.240 ± 0.160	0.240 ± 0.160	0.280 ± 0.140
	FGS	6.680 ± 1.000	14.260 ± 3.680	0.400 ± 0.160	0.210 ± 0.080	0.270 ± 0.100
	NOTEARS-MLP	6.000	17.000	–	0.000	–
	GRaSP	5.920 ± 0.400	16.920 ± 0.400	**1.000** ***	0.010 ± 0.070	0.500
	BOSS	5.880 ± 0.480	16.880 ± 0.480	**1.000** ***	0.020 ± 0.080	0.500
	MPC	4.620 ± 1.880	14.120 ± 4.040	0.830 ± 0.140	0.230 ± 0.310	0.680 ± 0.140
	MPC-LLM	**2.060 ± 1.040** ***	**6.320 ± 4.180** ***	0.890 ± 0.120	**0.670 ± 0.160** ***	**0.750 ± 0.140** *
	ABAPC	3.675 ± 1.027	15.860 ± 4.087	0.518 ± 0.153	0.394 ± 0.165	0.453 ± 0.145
	ABAPC-LLM	3.297 ± 0.981	13.477 ± 3.868	0.612 ± 0.165	0.457 ± 0.155	0.517 ± 0.147
	LLM-BFS	6.000	0.000	0.500	1.000	0.667
`asia` $\lvert\mathbf{V}\rvert = 8$ $\lvert E\rvert = 8$	Random	12.660 ± 1.810	30.360 ± 7.940	0.140 ± 0.110	0.140 ± 0.110	0.190 ± 0.080
	FGS	11.240 ± 0.770	38.160 ± 2.780	0.250 ± 0.050	0.240 ± 0.030	0.250 ± 0.040
	NOTEARS-MLP	7.180 ± 1.260	32.000 ± 8.000	0.310 ± 0.200	0.230 ± 0.160	0.300 ± 0.150
	GRaSP	6.800 ± 1.500	17.080 ± 6.220	0.670 ± 0.250	0.360 ± 0.130	0.450 ± 0.110
	BOSS	6.220 ± 1.000	24.740 ± 6.740	0.600 ± 0.350	0.240 ± 0.140	0.400 ± 0.110
	MPC	6.000	23.040 ± 0.830	**0.980 ± 0.080** ***	0.250 ± 0.020	0.400 ± 0.020
	MPC-LLM	**2.840 ± 0.470** ***	**13.720 ± 0.540** ***	**0.990 ± 0.030** ***	**0.650 ± 0.050** ***	**0.780 ± 0.040** ***
	ABAPC	5.047 ± 1.028	25.730 ± 6.848	0.684 ± 0.207	0.372 ± 0.122	0.477 ± 0.149
	ABAPC-LLM	3.580 ± 0.710	15.500 ± 2.308	**0.992 ± 0.040** ***	0.557 ± 0.082	0.710 ± 0.068
	LLM-BFS	5.000	0.000	0.615	1.000	0.762
`sachs` $\lvert\mathbf{V}\rvert = 11$ $\lvert E\rvert = 17$	Random	26.020 ± 2.490	50.400 ± 5.870	0.160 ± 0.080	0.160 ± 0.080	0.170 ± 0.080
	FGS	23.820 ± 1.640	48.700 ± 3.850	0.140 ± 0.040	0.130 ± 0.040	0.140 ± 0.030
	NOTEARS-MLP	13.580 ± 1.200	52.360 ± 6.610	0.600 ± 0.190	0.200 ± 0.070	0.300 ± 0.100
	GRaSP	16.420 ± 1.110	52.320 ± 1.870	0.550 ± 0.130	0.030 ± 0.070	0.220 ± 0.050
	BOSS	17.000	53.000	–	0.000	–
	MPC	16.000 ± 2.740	50.560 ± 6.680	0.730 ± 0.010	0.060 ± 0.160	0.590 ± 0.030
	MPC-LLM	13.280 ± 1.970	47.840 ± 5.910	0.740 ± 0.020	0.220 ± 0.120	0.320 ± 0.100
	ABAPC	5.244 ± 1.684	26.212 ± 9.472	0.721 ± 0.101	0.693 ± 0.100	0.706 ± 0.100
	ABAPC-LLM	**4.460 ± 1.264** **	**21.056 ± 6.983** **	**0.759 ± 0.063** *	**0.739 ± 0.075** **	**0.748 ± 0.069** *
	LLM-BFS	20.000	50.000	0.312	0.294	0.303
`child` $\lvert\mathbf{V}\rvert = 20$ $\lvert E\rvert = 25$	Random	45.260 ± 2.350	304.400 ± 22.620	0.070 ± 0.040	0.070 ± 0.040	0.080 ± 0.040
	FGS	57.000 ± 4.110	271.700 ± 28.450	0.110 ± 0.040	0.190 ± 0.070	0.140 ± 0.050
	NOTEARS-MLP	20.660 ± 2.400	220.300 ± 29.260	0.450 ± 0.100	0.370 ± 0.080	0.400 ± 0.080
	GRaSP	16.320 ± 3.390	232.400 ± 36.750	0.880 ± 0.160	0.370 ± 0.130	0.510 ± 0.140
	BOSS	14.220 ± 2.360	210.500 ± 28.260	**0.960 ± 0.080** ***	0.440 ± 0.090	0.600 ± 0.100
	MPC	9.160 ± 4.130	164.300 ± 63.590	0.840 ± 0.070	0.630 ± 0.170	0.710 ± 0.120
	MPC-LLM	**5.060 ± 1.800** ***	87.820 ± 26.340	0.880 ± 0.050	**0.800 ± 0.070** ***	0.830 ± 0.050
	ABAPC	**5.372 ± 3.250** ***	**84.833 ± 45.226**	0.851 ± 0.089	**0.786 ± 0.129** ***	0.815 ± 0.105
	ABAPC-LLM	**4.417 ± 2.009** ***	**72.653 ± 24.836** **	0.894 ± 0.050	**0.824 ± 0.080** ***	**0.855 ± 0.052** *
	LLM-BFS	56.000	114.0	0.242	0.600	0.345

Table 6: Runtime in seconds on `CauseNet` synthetic datasets (mean±std; LLM API latency excluded).

Method	$\lvert\mathbf{V}\rvert = 5$	$\lvert\mathbf{V}\rvert = 10$	$\lvert\mathbf{V}\rvert = 15$
Random	0.00	0.00	0.00
FGS	0.11 ± 0.03	0.14 ± 0.04	0.15 ± 0.04
NOTEARS-MLP	0.19 ± 0.10	0.64 ± 0.25	1.20 ± 0.77
GRaSP	1.10 ± 0.27	5.45 ± 2.88	14.06 ± 5.69
BOSS	1.29 ± 0.32	7.22 ± 3.41	18.63 ± 7.64
MPC	0.01	0.04 ± 0.03	0.05 ± 0.03
MPC-LLM	0.01	0.04 ± 0.03	0.05 ± 0.03
ABAPC	0.05 ± 0.03	1.96 ± 2.88	4.42 ± 4.06
ABAPC-LLM	0.06 ± 0.05	1.81 ± 2.54	4.33 ± 3.97

C.3.3 Runtime

Table 6 reports wall-clock runtimes excluding LLM API latency, which is incurred once per dataset when the LLM-derived constraints are generated. The relevant comparison is therefore the solver/search cost after those constraints are available. ABAPC and ABAPC-LLM are slower than MPC because they ground and solve an argumentative program, but the skeleton-reduction step keeps the evaluated networks tractable. The LLM constraints can also reduce the effective search space, so ABAPC-LLM is not uniformly slower than ABAPC despite adding semantic facts.

C.4 LLM CONSTRAINTS EVALUATION

In this section we evaluate the LLM-derived required and forbidden arrows before they are integrated with CI tests, compare single-run averages with the five-run unanimity consensus used in the main experiments, and test whether variable descriptions help beyond variable names. These analyses explain when the semantic layer helps structural learning and when it can become a source of conflict.

Table 7 summarises LLM-derived constraint quality for average single runs and consensus filters, with and without variable descriptions. The description effect depends strongly on the metadata regime. On `bnlearn`, descriptions are valuable because some variable names are cryptic (e.g., Survey A, S, E, O, R, T and Sachs protein abbreviations); both forbidden and required constraint F1 improve in the average-run setting. On `CauseNet`, variable names are already meaningful, so descriptions add less and can sometimes introduce misleading context. The consensus filter is deliberately conservative: it keeps fewer constraints and often lowers recall, but raises reliability when a constraint survives unanimity. This is the intended operating point for ABAPC-LLM, because false semantic facts are costly once they enter the conflict-resolution. Table 8 repeats the same analysis after removing side-specific zero-constraint cases. This separates the quality of emitted constraints from consensus coverage. When unanimity emits a non-empty constraint set, precision is often very high, especially for forbidden constraints.

Table 7: LLM-derived constraint quality by dataset, consensus strategy, and description use. Bold compares Average vs. Consensus within each setting; green marks the row best; arrows show the description effect.

Metric	**bnlearn**				**CauseNet**			
	Average		**Consensus**		**Average**		**Consensus**	
	w/o desc	**with desc**	**w/o desc**	**with desc**	**w/o desc**	**with desc**	**w/o desc**	**with desc**
Forbidden constraints								
# constraints	19.80 ± 13.90	25.47 ± 24.39	8.00 ± 6.60	8.17 ± 1.72	17.59 ± 13.02	20.64 ± 16.60	4.85 ± 4.83	4.07 ± 3.42
Precision	$\mathbf{0.927 \pm 0.186}$	$\mathbf{0.973 \pm 0.041}\uparrow$	0.833 ± 0.408	$0.948 \pm 0.085\uparrow$	$\mathbf{0.952 \pm 0.132}$	$\mathbf{0.942 \pm 0.155}\downarrow$	0.855 ± 0.339	$0.877 \pm 0.317\uparrow$
Recall	$\mathbf{0.409 \pm 0.234}$	$\mathbf{0.472 \pm 0.219}\uparrow$	0.241 ± 0.240	$0.275 \pm 0.216\uparrow$	$\mathbf{0.293 \pm 0.216}$	$\mathbf{0.312 \pm 0.221}\uparrow$	0.123 ± 0.164	$0.115 \pm 0.168\downarrow$
F1	$\mathbf{0.533 \pm 0.260}$	$\mathbf{0.606 \pm 0.224}\uparrow$	0.338 ± 0.316	$0.392 \pm 0.281\uparrow$	$\mathbf{0.409 \pm 0.232}$	$\mathbf{0.431 \pm 0.234}\uparrow$	0.189 ± 0.216	$0.174 \pm 0.218\downarrow$
Required constraints								
# constraints	8.27 ± 6.97	10.80 ± 11.71	4.50 ± 3.73	3.67 ± 4.13	12.57 ± 9.60	14.56 ± 11.52	4.11 ± 3.29	4.74 ± 3.33
Precision	$\mathbf{0.468 \pm 0.288}$	$\mathbf{0.597 \pm 0.235}\uparrow$	0.375 ± 0.327	$0.505 \pm 0.422\uparrow$	0.487 ± 0.249	$0.464 \pm 0.230\downarrow$	$\mathbf{0.568 \pm 0.353}$	$\mathbf{0.514 \pm 0.335}\downarrow$
Recall	$\mathbf{0.445 \pm 0.352}$	$\mathbf{0.611 \pm 0.328}\uparrow$	0.303 ± 0.388	$0.281 \pm 0.375\downarrow$	$\mathbf{0.486 \pm 0.255}$	$\mathbf{0.516 \pm 0.242}\uparrow$	0.244 ± 0.210	$0.278 \pm 0.244\uparrow$
F1	$\mathbf{0.401 \pm 0.240}$	$\mathbf{0.555 \pm 0.232}\uparrow$	0.284 ± 0.284	$0.321 \pm 0.337\uparrow$	$\mathbf{0.450 \pm 0.206}$	$\mathbf{0.453 \pm 0.190}\uparrow$	0.314 ± 0.228	$0.334 \pm 0.249\uparrow$

Table 8: LLM-derived constraint quality excluding zero-constraint cases. Formatting follows Table 7: bold compares Average vs. Consensus within each setting; green marks the row best; arrows show the description effect.

Metric	bnlearn				CauseNet			
	Average		Consensus		Average		Consensus	
	w/o desc	with desc	w/o desc	with desc	w/o desc	with desc	w/o desc	with desc
Forbidden constraints								
# constraints	20.48 ± 13.63	25.47 ± 24.39	9.60 ± 5.94	8.17 ± 1.72	17.86 ± 12.94	20.79 ± 16.57	5.57 ± 4.78	4.58 ± 3.29
Precision	0.959 ± 0.062	$\mathbf{0.973 \pm 0.041}\uparrow$	$\mathbf{1.000}$	$0.948 \pm 0.085\downarrow$	0.966 ± 0.061	$0.949 \pm 0.132\downarrow$	$\mathbf{0.982 \pm 0.069}$	$\mathbf{0.986 \pm 0.053}\uparrow$
Recall	$\mathbf{0.423 \pm 0.225}$	$\mathbf{0.472 \pm 0.219}\uparrow$	0.289 ± 0.234	$0.275 \pm 0.216\downarrow$	$\mathbf{0.297 \pm 0.215}$	$\mathbf{0.314 \pm 0.220}\uparrow$	0.141 ± 0.168	$0.129 \pm 0.172\downarrow$
F1	$\mathbf{0.551 \pm 0.244}$	$\mathbf{0.606 \pm 0.224}\uparrow$	0.405 ± 0.300	$0.392 \pm 0.281\downarrow$	$\mathbf{0.415 \pm 0.228}$	$\mathbf{0.434 \pm 0.231}\uparrow$	0.217 ± 0.218	$0.196 \pm 0.222\downarrow$
Required constraints								
# constraints	9.92 ± 6.45	11.17 ± 11.74	6.75 ± 1.71	5.50 ± 3.87	12.66 ± 9.57	14.66 ± 11.50	4.53 ± 3.17	5.02 ± 3.22
Precision	0.562 ± 0.213	$0.618 \pm 0.210\uparrow$	$\mathbf{0.562 \pm 0.195}$	$\mathbf{0.757 \pm 0.206}\uparrow$	0.491 ± 0.247	$0.468 \pm 0.227\downarrow$	$\mathbf{0.626 \pm 0.317}$	$\mathbf{0.544 \pm 0.320}\downarrow$
Recall	$\mathbf{0.535 \pm 0.316}$	$\mathbf{0.632 \pm 0.312}\uparrow$	0.454 ± 0.399	$0.422 \pm 0.394\downarrow$	$\mathbf{0.490 \pm 0.252}$	$\mathbf{0.520 \pm 0.239}\uparrow$	0.269 ± 0.204	$0.294 \pm 0.241\uparrow$
F1	$\mathbf{0.482 \pm 0.172}$	$\mathbf{0.574 \pm 0.211}\uparrow$	0.426 ± 0.233	$0.481 \pm 0.294\uparrow$	$\mathbf{0.454 \pm 0.203}$	$\mathbf{0.457 \pm 0.187}\uparrow$	0.346 ± 0.214	$0.354 \pm 0.242\uparrow$

C.4.1 Gemini and GPT-5-mini Constraint Sources

We report `Gemini-2.5-flash` and `GPT-5-mini` under the same prompting, parsing, and five-run consensus protocol. `Gemini-2.5-flash` is used for the broader ablation figures because most constraint-quality and metadata ablations were run with `Gemini-2.5-flash` and because it gives broader non-empty consensus coverage on the synthetic datasets: 476 unanimous constraints and non-empty consensus sets on 53/54 datasets, compared with 129 unanimous constraints and non-empty consensus sets on 43/54 datasets for `GPT-5-mini`. Table 9 reports the matched structural comparison on the 43 synthetic datasets where both `Gemini-2.5-flash` and `GPT-5-mini` consensus runs are available. ABAPC-LLM is stable across the two LLM sources, while MPC-LLM remains competitive mainly where direct hard constraints help. `GPT-5-mini` produces more zero-F1 consensus sets than `Gemini-2.5-flash` (33.3% vs. 1.9%) and fewer high-quality sets (3.7% vs. 13.0%).

Table 9: Structural metrics on the 43 matched `CauseNet` datasets with Gemini and GPT-5-mini constraints, grouped by $|\mathbf{V}|$. LLM-constraint variants are labelled by source. Among repeated-run methods, bold marks the best mean and statistically tied methods under BH-corrected Welch tests within each metric/column. Significance levels for the p-values against the next non-bold method in the ranking are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$. LLM-BFS is shown from one saved run and is excluded from Welch tests and bolding. Tests: Table 18; tied bolding: Table 11.

	Method	SHD ↓	Precision ↑	Recall ↑	F1 ↑
$\|\mathbf{V}\| = 5$	Random	6.857 ± 1.470	0.273 ± 0.145	0.385 ± 0.225	0.311 ± 0.168
	FGS	5.610 ± 1.101	0.295 ± 0.199	0.192 ± 0.130	0.224 ± 0.149
	NOTEARS-MLP	5.467	0.000	0.000	0.000
	GRaSP	3.677 ± 1.465	0.737 ± 0.266	0.347 ± 0.242	0.591 ± 0.137
	BOSS	3.633 ± 1.367	0.746 ± 0.232	0.355 ± 0.229	0.601 ± 0.127
	MPC	3.395 ± 0.643	0.836 ± 0.107	0.367 ± 0.111	0.592 ± 0.065
	MPC-LLM (Gemini)	1.328 ± 0.331	$\mathbf{0.952 \pm 0.027}$***	0.753 ± 0.053	$\mathbf{0.822 \pm 0.041}$***
	MPC-LLM (GPT)	2.213 ± 0.479	0.835 ± 0.088	0.589 ± 0.089	0.713 ± 0.083
	ABAPC	2.127 ± 0.566	0.702 ± 0.084	0.609 ± 0.090	0.663 ± 0.083
	ABAPC-LLM (Gemini)	$\mathbf{1.186 \pm 0.375}$*	0.902 ± 0.043	$\mathbf{0.778 \pm 0.063}$*	$\mathbf{0.828 \pm 0.048}$***
	ABAPC-LLM (GPT)	1.604 ± 0.546	0.807 ± 0.090	0.702 ± 0.095	0.755 ± 0.084
	LLM-BFS (Gemini)	2.800	0.877	0.558	0.631
$\|\mathbf{V}\| = 10$	Random	28.076 ± 6.230	0.143 ± 0.071	0.300 ± 0.176	0.184 ± 0.091
	FGS	16.127 ± 1.747	0.189 ± 0.122	0.107 ± 0.069	0.132 ± 0.081
	NOTEARS-MLP	12.687 ± 0.126	0.029 ± 0.123	0.002 ± 0.011	0.005 ± 0.022
	GRaSP	8.466 ± 2.772	0.805 ± 0.274	0.366 ± 0.184	0.519 ± 0.174
	BOSS	7.560 ± 2.604	$\mathbf{0.884 \pm 0.201}$***	0.426 ± 0.180	0.569 ± 0.171
	MPC	7.173 ± 1.314	0.836 ± 0.109	0.446 ± 0.098	0.574 ± 0.096
	MPC-LLM (Gemini)	6.197 ± 1.095	0.843 ± 0.093	0.519 ± 0.081	0.634 ± 0.084
	MPC-LLM (GPT)	6.630 ± 1.231	0.860 ± 0.095	0.492 ± 0.092	0.621 ± 0.084
	ABAPC	6.345 ± 1.449	0.723 ± 0.126	0.526 ± 0.102	0.605 ± 0.107
	ABAPC-LLM (Gemini)	$\mathbf{5.984 \pm 1.257}$.	0.776 ± 0.096	0.542 ± 0.090	0.633 ± 0.089
	ABAPC-LLM (GPT)	$\mathbf{5.755 \pm 1.417}$***	0.787 ± 0.114	$\mathbf{0.569 \pm 0.101}$***	$\mathbf{0.655 \pm 0.102}$***
	LLM-BFS (Gemini)	12.857	0.489	0.432	0.446
$\|\mathbf{V}\| = 15$	Random	63.989 ± 19.393	0.083 ± 0.034	0.302 ± 0.171	0.123 ± 0.052
	FGS	24.747 ± 2.130	0.095 ± 0.074	0.059 ± 0.046	0.073 ± 0.056
	NOTEARS-MLP	17.197 ± 0.177	0.024 ± 0.121	0.001 ± 0.008	0.001 ± 0.014
	GRaSP	10.230 ± 3.184	0.854 ± 0.217	0.436 ± 0.161	0.564 ± 0.163
	BOSS	9.653 ± 3.086	0.892 ± 0.176	0.460 ± 0.161	0.589 ± 0.165
	MPC	8.677 ± 1.207	$\mathbf{0.934 \pm 0.068}$***	0.497 ± 0.069	0.640 ± 0.070
	MPC-LLM (Gemini)	$\mathbf{7.616 \pm 1.052}$.	$\mathbf{0.934 \pm 0.049}$***	0.557 ± 0.060	$\mathbf{0.691 \pm 0.055}$***
	MPC-LLM (GPT)	8.059 ± 1.217	$\mathbf{0.941 \pm 0.064}$***	0.532 ± 0.070	0.671 ± 0.068
	ABAPC	7.863 ± 1.602	0.770 ± 0.101	0.554 ± 0.089	0.641 ± 0.092
	ABAPC-LLM (Gemini)	$\mathbf{7.508 \pm 1.292}$*	0.799 ± 0.072	$\mathbf{0.573 \pm 0.073}$**	0.663 ± 0.070
	ABAPC-LLM (GPT)	$\mathbf{7.575 \pm 1.513}$*	0.790 ± 0.093	$\mathbf{0.570 \pm 0.086}$*	0.658 ± 0.087
	LLM-BFS (Gemini)	16.429	0.448	0.341	0.357

C.4.2 Interaction Between Constraint Quality and Structural Performance

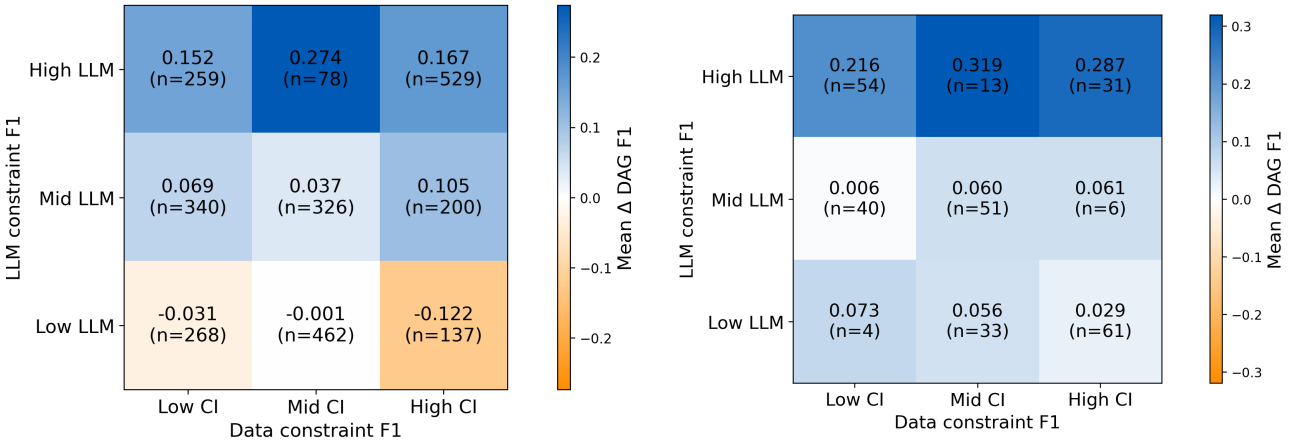
The interaction heatmaps ask whether semantic constraints help because they are independently accurate, because the CI evidence is accurate, or because both signals are accurate together. Each cell bins runs by LLM-constraints F1 and retained CI-constraints F1 and reports the mean change in final DAG F1 after adding semantic constraints to ABAPC.

Figure 6 compares `CauseNet` and `bnlearn` with descriptions. On `CauseNet`, the highest LLM-quality row is positive across all CI bins, while the low-LLM/high-CI cell is negative. This is the central failure mode of the current integration policy: misleading semantic constraints can still force the solver to relax otherwise useful statistical facts. On `bnlearn`, the cells are non-negative and the high-LLM row is strongly positive, which is consistent with the much cleaner LLM constraints obtained on these public benchmark networks.

Figure 7 repeats the comparison with names only. The `CauseNet` heatmap becomes less extreme: gains remain positive in the high-LLM row and the most negative cell is close to zero. This supports the interpretation from Table 7: descriptions add useful information in many cases, but they can also add misleading context when variable names already carry strong semantic signal. For `bnlearn`, names-only LLM-derived constraints remain useful but are less informative for datasets with terse labels, especially `survey` (single-letter variables A, S, E, O, R, T) and `sachs` (protein-signalling abbreviations such as `Akt`, `Mek`, and `PKC`).

Figures 8 and 9 provide the same `CauseNet` analysis split by node count and graph type. They show that the qualitative pattern is not an artefact of one graph family: high-quality semantic constraints tend to help across sizes and ER/SF/LT topologies, while low-quality semantic constraints are risky precisely when the data-derived signal is already informative.

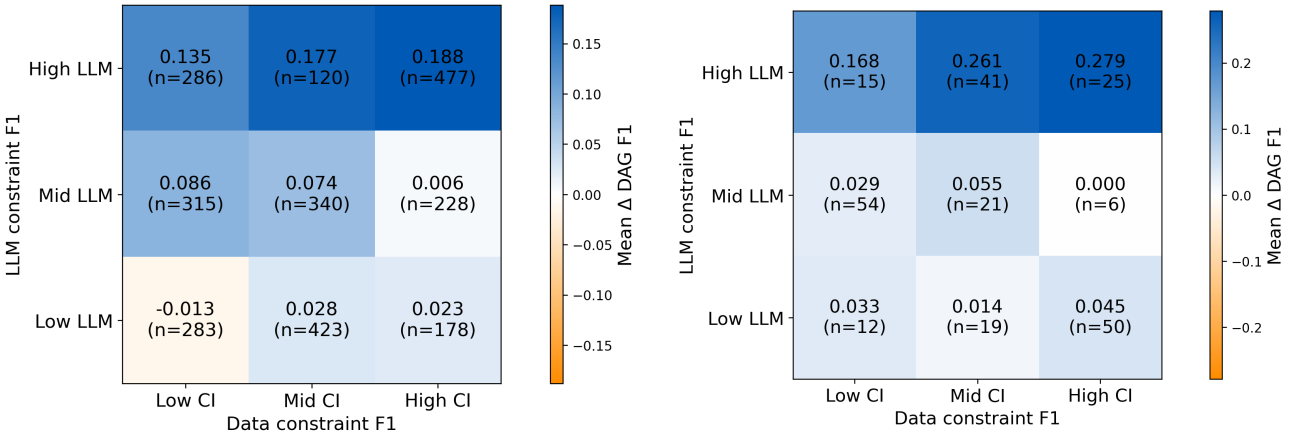
Figures 10 and 11 split the `bnlearn` heatmap by dataset. The global-bin version is useful for comparing absolute LLM-constraint quality across datasets; the within-dataset quantile version is useful for seeing whether better LLM-derived constraints help inside each benchmark even when a dataset occupies only a narrow global quality range. The public benchmarks concentrate in fewer bins than `CauseNet`, again consistent with smaller graphs and possible benchmark familiarity.



(a) `CauseNet` synthetic datasets.

(b) `bnlearn` benchmarks.

Figure 6: With variable descriptions: interaction between LLM-constraint F1 and retained-CI F1. Cell values are mean Δ DAG F1 after adding semantic constraints to ABAPC; counts are in parentheses.



(a) `CauseNet` synthetic datasets (names only).

(b) `bnlearn` benchmarks (names only).

Figure 7: Without variable descriptions: interaction between LLM-constraint F1 and retained-CI F1. Names-only metadata weakens the negative `CauseNet` cells but also removes useful contextual information for some `bnlearn` labels.

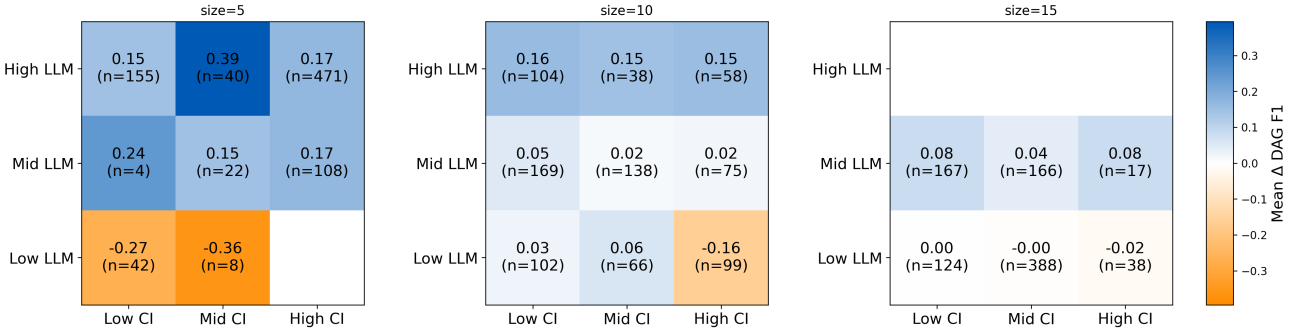


Figure 8: `CauseNet`: interaction heatmaps split by node count, using the same global bins as Fig. 4.

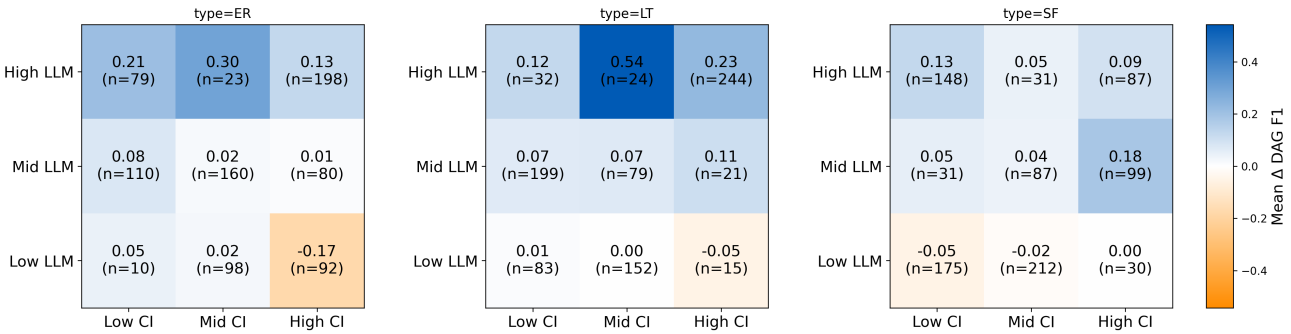


Figure 9: `CauseNet`: interaction heatmaps split by graph type, using the same global bins as Fig. 4.

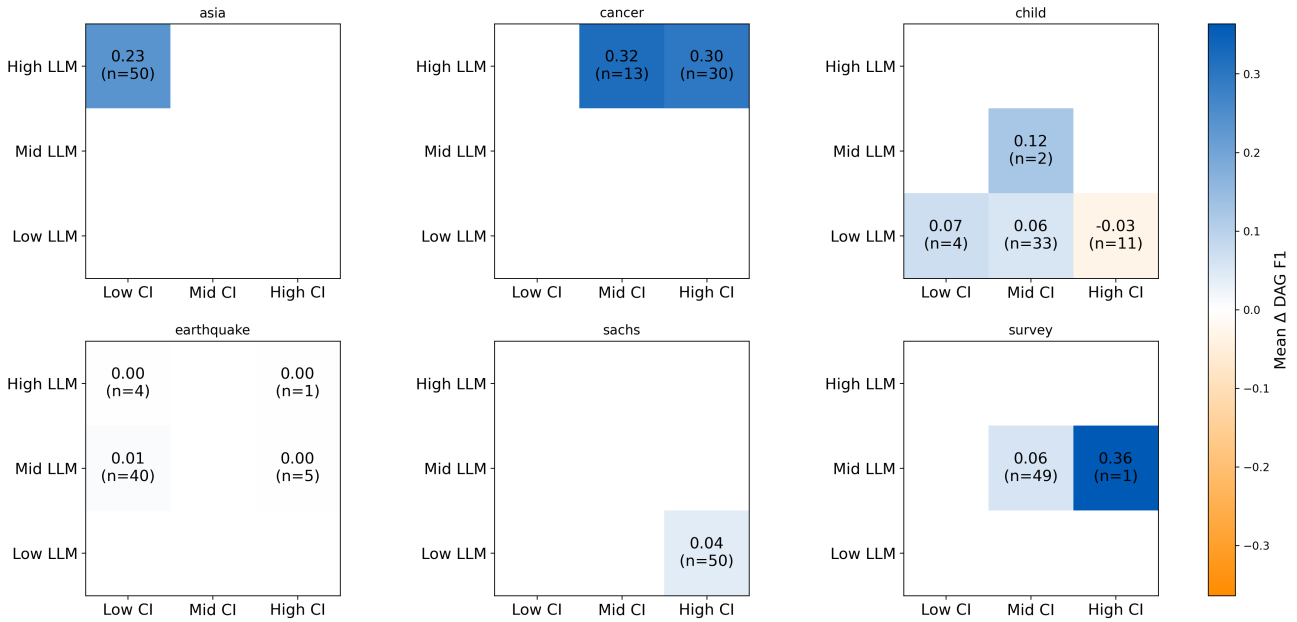


Figure 10: `bnlearn`: per-dataset interaction heatmaps using the same global bins as Fig. 6b, allowing absolute LLM-constraint-quality comparisons across benchmarks.

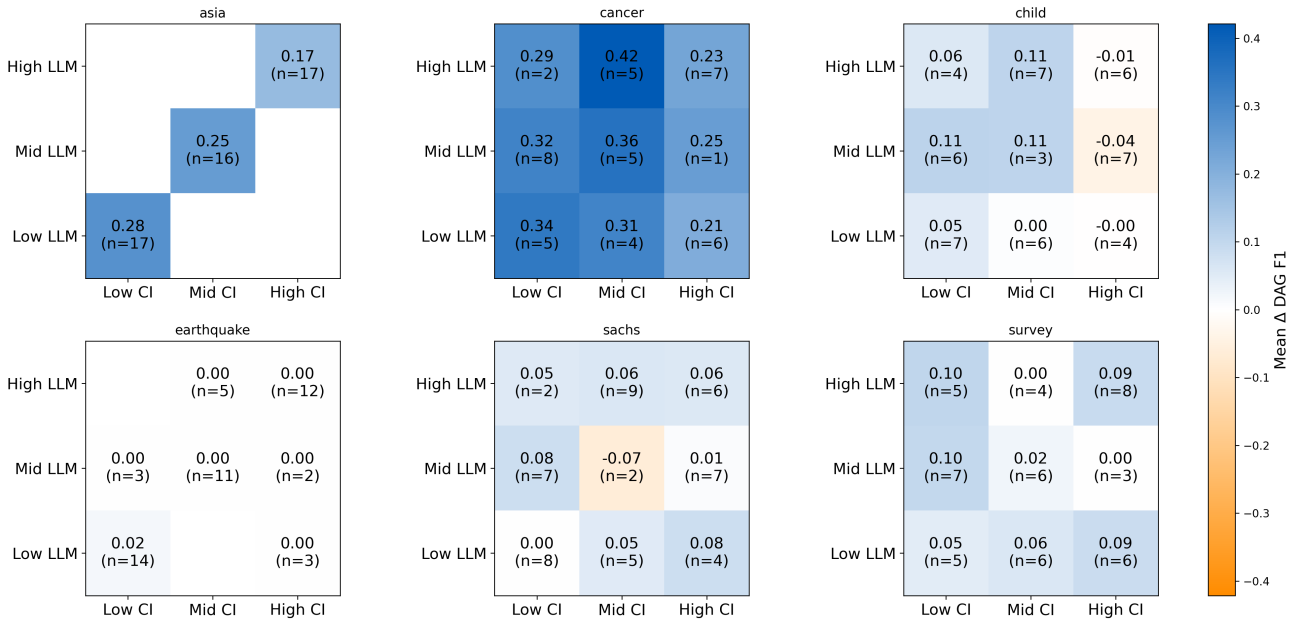


Figure 11: `bnlearn`: per-dataset interaction heatmaps using within-dataset quantile bins, highlighting relative variation inside each benchmark.

Table 10: Welch tests for Table 1. Each row compares a bolded top entry with the next non-bold method in that column. Significance levels for both reported p-values are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$.

Group	Metric	Methods	Means±Std	t	p-value	p_{BH}
$\lvert\mathbf{V}\rvert = 5$	SHD	ABAPC-LLM vs MPC-LLM	1.230 ± 0.369 vs 1.500 ± 0.318	-5.60	2.19e-8***	2.19e-8***
$\lvert\mathbf{V}\rvert = 5$	Precision	MPC-LLM vs ABAPC-LLM	0.958 ± 0.031 vs 0.882 ± 0.046	12.94	2.57e-38***	2.57e-38***
$\lvert\mathbf{V}\rvert = 5$	Recall	ABAPC-LLM vs MPC-LLM	0.763 ± 0.062 vs 0.710 ± 0.050	5.12	3.13e-7***	3.13e-7***
$\lvert\mathbf{V}\rvert = 5$	F1	ABAPC-LLM vs MPC-LLM	0.812 ± 0.049 vs 0.795 ± 0.037	1.98	0.048*	0.048*
$\lvert\mathbf{V}\rvert = 10$	SHD	ABAPC-LLM vs ABAPC	5.626 ± 1.176 vs 6.166 ± 1.442	-5.03	4.89e-7***	4.89e-7***
$\lvert\mathbf{V}\rvert = 10$	Precision	BOSS vs MPC-LLM	0.880 ± 0.197 vs 0.856 ± 0.088	4.34	1.40e-5***	1.40e-5***
$\lvert\mathbf{V}\rvert = 10$	Recall	ABAPC-LLM vs ABAPC	0.565 ± 0.086 vs 0.535 ± 0.105	5.40	6.63e-8***	6.63e-8***
$\lvert\mathbf{V}\rvert = 10$	F1	ABAPC-LLM vs MPC-LLM	0.653 ± 0.084 vs 0.627 ± 0.077	4.38	1.17e-5***	1.17e-5***
$\lvert\mathbf{V}\rvert = 15$	SHD	ABAPC-LLM vs MPC-LLM	7.550 ± 1.281 vs 7.874 ± 1.038	-2.94	0.003**	0.003**
$\lvert\mathbf{V}\rvert = 15$	Precision	MPC vs BOSS	0.936 ± 0.071 vs 0.880 ± 0.191	18.74	2.28e-78***	4.55e-78***
$\lvert\mathbf{V}\rvert = 15$	Precision	MPC-LLM vs BOSS	0.929 ± 0.061 vs 0.880 ± 0.191	16.57	1.14e-61***	1.14e-61***
$\lvert\mathbf{V}\rvert = 15$	Recall	ABAPC-LLM vs ABAPC	0.565 ± 0.074 vs 0.546 ± 0.087	4.42	9.95e-6***	9.95e-6***
$\lvert\mathbf{V}\rvert = 15$	F1	MPC-LLM vs ABAPC-LLM	0.667 ± 0.058 vs 0.654 ± 0.072	2.52	0.012*	0.012*

C.4.3 Statistical Test Results

The following tables report Welch tests used by the result tables throughout the paper. There are two kinds of test summaries. Indicator-test tables compare a bold top entry with the next non-bold method in the same metric/group ranking; these rows explain the significance symbols shown in the result tables. Tied-bolding tables list the non-significant best-vs-comparator tests that justify bolding methods other than the numeric best. In all cases, p-values are Benjamini–Hochberg (Benjamini and Hochberg, 1995) corrected within the corresponding metric/group family.

For `CauseNet`, Table 10 gives the indicator tests for the main node-count table, while Table 11 collects the tied-bolding tests for the structural `CauseNet` tables and the structural ablations. These two tables should be read together: the first explains the significance symbols against the next non-bold method, and the second explains why multiple top methods can be bolded when they are not significantly different. Table 12 reports the corresponding indicator tests for the `CauseNet` graph-type/node-count breakdown, including its marginal averages. Table 13 gives the indicator tests for the `CauseNet` α threshold ablation. For `bnlearn`, Tables 14–16 split the indicator tests by dataset blocks to keep the tables readable, and Table 17 gives the tied-bolding tests for the same benchmark table. Finally, Table 18 gives the indicator tests for the LLM-source ablation.

Table 11: Non-significant best-vs-comparator tests that justify additional bolded methods in Tables 1, 3, 4, and 9. Significance levels for both reported p-values are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$.

Table	Group	Metric	Methods	Means±Std	t	p-value	p_{BH}
1	$\lvert\mathbf{V}\rvert = 15$	Precision	MPC vs MPC-LLM	0.936 ± 0.071 vs 0.929 ± 0.061	1.91	0.056.	0.056.
3	SF $\lvert\mathbf{V}\rvert = 5$	F1	ABAPC-LLM vs MPC-LLM	0.734 ± 0.046 vs 0.697 ± 0.055	1.81	0.071.	0.071.
3	LT $\lvert\mathbf{V}\rvert = 5$	SHD	MPC-LLM vs ABAPC-LLM	1.217 ± 0.345 vs 1.335 ± 0.457	−1.20	0.231ns	0.231ns
3	ER $\lvert\mathbf{V}\rvert = 10$	F1	MPC-LLM vs ABAPC-LLM	0.663 ± 0.057 vs 0.656 ± 0.057	0.57	0.570ns	0.570ns
3	SF $\lvert\mathbf{V}\rvert = 10$	SHD	ABAPC-LLM vs MPC-LLM	5.711 ± 1.302 vs 6.112 ± 1.198	−1.95	0.051.	0.051.
3	LT $\lvert\mathbf{V}\rvert = 10$	F1	BOSS vs ABAPC-LLM	0.638 ± 0.192 vs 0.623 ± 0.111	1.67	0.095.	0.095.
3	SF $\lvert\mathbf{V}\rvert = 15$	SHD	ABAPC vs ABAPC-LLM	7.103 ± 1.144 vs 7.226 ± 1.151	−1.71	0.087.	0.087.
3	SF $\lvert\mathbf{V}\rvert = 15$	F1	ABAPC vs ABAPC-LLM	0.601 ± 0.080 vs 0.590 ± 0.077	1.84	0.065.	0.065.
3	ER Avg.	F1	ABAPC-LLM vs MPC-LLM	0.720 ± 0.056 vs 0.715 ± 0.048	0.70	0.481ns	0.481ns
3	LT Avg.	SHD	ABAPC-LLM vs MPC-LLM	4.868 ± 1.114 vs 4.912 ± 0.816	−0.27	0.784ns	0.784ns
3	LT Avg.	F1	MPC-LLM vs ABAPC-LLM	0.739 ± 0.052 vs 0.731 ± 0.079	1.14	0.253ns	0.253ns
4	$\alpha = 0.05$ $\lvert\mathbf{V}\rvert = 5$	SHD	MPC-LLM vs ABAPC-LLM	1.170 ± 0.583 vs 1.233 ± 0.514	−1.45	0.146ns	0.146ns
9	$\lvert\mathbf{V}\rvert = 5$	F1	ABAPC-LLM (Gemini) vs MPC-LLM (Gemini)	0.828 ± 0.048 vs 0.822 ± 0.041	0.67	0.506ns	0.506ns
9	$\lvert\mathbf{V}\rvert = 10$	SHD	ABAPC-LLM (GPT) vs ABAPC-LLM (Gemini)	5.755 ± 1.417 vs 5.984 ± 1.257	−1.95	0.051.	0.051.
9	$\lvert\mathbf{V}\rvert = 15$	SHD	ABAPC-LLM (Gemini) vs MPC-LLM (Gemini)	7.508 ± 1.292 vs 7.616 ± 1.052	−0.84	0.400ns	0.444ns
9	$\lvert\mathbf{V}\rvert = 15$	SHD	ABAPC-LLM (Gemini) vs ABAPC-LLM (GPT)	7.508 ± 1.292 vs 7.575 ± 1.513	−0.52	0.604ns	0.604ns
9	$\lvert\mathbf{V}\rvert = 15$	Precision	MPC-LLM (GPT) vs MPC	0.941 ± 0.064 vs 0.934 ± 0.068	1.67	0.094.	0.094.
9	$\lvert\mathbf{V}\rvert = 15$	Precision	MPC-LLM (GPT) vs MPC-LLM (Gemini)	0.941 ± 0.064 vs 0.934 ± 0.049	1.69	0.092.	0.094.
9	$\lvert\mathbf{V}\rvert = 15$	Recall	ABAPC-LLM (Gemini) vs ABAPC-LLM (GPT)	0.573 ± 0.073 vs 0.570 ± 0.086	0.78	0.435ns	0.435ns

Table 12: Welch tests for Table 3, including marginal averages. Significance levels for both reported p-values are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$.

Group	Metric	Methods	Means±Std	t	p-value	p_{BH}
ER, $\lvert\mathbf{V}\rvert = 5$	SHD	ABAPC-LLM vs ABAPC	1.054 ± 0.372 vs 1.515 ± 0.542	-7.91	2.63e-15***	2.63e-15***
ER, $\lvert\mathbf{V}\rvert = 5$	F1	ABAPC-LLM vs MPC-LLM	0.843 ± 0.044 vs 0.805 ± 0.025	3.94	8.25e-5***	8.25e-5***
SF, $\lvert\mathbf{V}\rvert = 5$	SHD	ABAPC-LLM vs MPC-LLM	1.302 ± 0.278 vs 1.603 ± 0.305	-3.06	0.002**	0.002**
SF, $\lvert\mathbf{V}\rvert = 5$	F1	ABAPC-LLM vs ABAPC	0.734 ± 0.046 vs 0.632 ± 0.075	7.05	1.79e-12***	3.59e-12***
SF, $\lvert\mathbf{V}\rvert = 5$	F1	MPC-LLM vs ABAPC	0.697 ± 0.055 vs 0.632 ± 0.075	3.91	9.07e-5***	9.07e-5***
LT, $\lvert\mathbf{V}\rvert = 5$	SHD	MPC-LLM vs ABAPC	1.217 ± 0.345 vs 2.709 ± 0.722	-13.57	5.71e-42***	1.14e-41***
LT, $\lvert\mathbf{V}\rvert = 5$	SHD	ABAPC-LLM vs ABAPC	1.335 ± 0.457 vs 2.709 ± 0.722	-13.03	8.06e-39***	8.06e-39***
LT, $\lvert\mathbf{V}\rvert = 5$	F1	MPC-LLM vs ABAPC-LLM	0.883 ± 0.030 vs 0.858 ± 0.057	2.65	0.008**	0.008**
Avg., $\lvert\mathbf{V}\rvert = 5$	SHD	ABAPC-LLM vs MPC-LLM	1.230 ± 0.369 vs 1.500 ± 0.318	-5.60	2.19e-8***	2.19e-8***
Avg., $\lvert\mathbf{V}\rvert = 5$	F1	ABAPC-LLM vs MPC-LLM	0.812 ± 0.049 vs 0.795 ± 0.037	1.98	0.048*	0.048*
ER, $\lvert\mathbf{V}\rvert = 10$	SHD	ABAPC-LLM vs MPC-LLM	5.061 ± 0.791 vs 5.488 ± 0.772	-3.28	0.001**	0.001**
ER, $\lvert\mathbf{V}\rvert = 10$	F1	MPC-LLM vs ABAPC	0.663 ± 0.057 vs 0.616 ± 0.094	4.20	2.63e-5***	5.25e-5***
ER, $\lvert\mathbf{V}\rvert = 10$	F1	ABAPC-LLM vs ABAPC	0.656 ± 0.057 vs 0.616 ± 0.094	3.77	1.65e-4***	1.65e-4***
SF, $\lvert\mathbf{V}\rvert = 10$	SHD	ABAPC-LLM vs ABAPC	5.711 ± 1.302 vs 6.221 ± 1.493	-2.43	0.015*	0.030*
SF, $\lvert\mathbf{V}\rvert = 10$	SHD	MPC-LLM vs ABAPC	6.112 ± 1.198 vs 6.221 ± 1.493	-0.50	0.618ns	0.618ns
SF, $\lvert\mathbf{V}\rvert = 10$	F1	ABAPC-LLM vs MPC-LLM	0.681 ± 0.086 vs 0.657 ± 0.092	2.69	0.007**	0.007**
LT, $\lvert\mathbf{V}\rvert = 10$	SHD	ABAPC-LLM vs BOSS	6.105 ± 1.437 vs 6.500 ± 2.993	-2.34	0.019*	0.019*
LT, $\lvert\mathbf{V}\rvert = 10$	F1	BOSS vs GRaSP	0.638 ± 0.192 vs 0.602 ± 0.205	5.88	4.15e-9***	8.30e-9***
LT, $\lvert\mathbf{V}\rvert = 10$	F1	ABAPC-LLM vs GRaSP	0.623 ± 0.111 vs 0.602 ± 0.205	2.33	0.020*	0.020*
Avg., $\lvert\mathbf{V}\rvert = 10$	SHD	ABAPC-LLM vs ABAPC	5.626 ± 1.176 vs 6.166 ± 1.442	-5.03	4.89e-7***	4.89e-7***
Avg., $\lvert\mathbf{V}\rvert = 10$	F1	ABAPC-LLM vs MPC-LLM	0.653 ± 0.084 vs 0.627 ± 0.077	4.38	1.17e-5***	1.17e-5***
ER, $\lvert\mathbf{V}\rvert = 15$	SHD	ABAPC-LLM vs MPC-LLM	8.259 ± 1.245 vs 8.815 ± 1.132	-2.53	0.011*	0.011*
ER, $\lvert\mathbf{V}\rvert = 15$	F1	MPC-LLM vs ABAPC-LLM	0.677 ± 0.062 vs 0.660 ± 0.068	3.26	0.001**	0.001**
SF, $\lvert\mathbf{V}\rvert = 15$	SHD	ABAPC vs MPC-LLM	7.103 ± 1.144 vs 8.307 ± 0.978	-12.22	2.55e-34***	5.11e-34***
SF, $\lvert\mathbf{V}\rvert = 15$	SHD	ABAPC-LLM vs MPC-LLM	7.226 ± 1.151 vs 8.307 ± 0.978	-11.20	4.12e-29***	4.12e-29***
SF, $\lvert\mathbf{V}\rvert = 15$	F1	ABAPC vs MPC	0.601 ± 0.080 vs 0.557 ± 0.073	6.04	1.59e-9***	3.17e-9***
SF, $\lvert\mathbf{V}\rvert = 15$	F1	ABAPC-LLM vs MPC	0.590 ± 0.077 vs 0.557 ± 0.073	4.60	4.25e-6***	4.25e-6***
LT, $\lvert\mathbf{V}\rvert = 15$	SHD	MPC-LLM vs ABAPC-LLM	6.502 ± 1.003 vs 7.165 ± 1.447	-2.90	0.004**	0.004**
LT, $\lvert\mathbf{V}\rvert = 15$	F1	MPC-LLM vs ABAPC-LLM	0.775 ± 0.042 vs 0.712 ± 0.070	12.24	1.93e-34***	1.93e-34***
Avg., $\lvert\mathbf{V}\rvert = 15$	SHD	ABAPC-LLM vs MPC-LLM	7.550 ± 1.281 vs 7.874 ± 1.038	-2.94	0.003**	0.003**
Avg., $\lvert\mathbf{V}\rvert = 15$	F1	MPC-LLM vs ABAPC-LLM	0.667 ± 0.058 vs 0.654 ± 0.072	2.52	0.012*	0.012*
ER, Avg.	SHD	ABAPC-LLM vs MPC-LLM	4.791 ± 0.803 vs 5.328 ± 0.736	-3.27	0.001**	0.001**
ER, Avg.	F1	ABAPC-LLM vs ABAPC	0.720 ± 0.056 vs 0.667 ± 0.090	8.73	2.50e-18***	5.01e-18***
ER, Avg.	F1	MPC-LLM vs ABAPC	0.715 ± 0.048 vs 0.667 ± 0.090	7.96	1.72e-15***	1.72e-15***
SF, Avg.	SHD	ABAPC-LLM vs ABAPC	4.747 ± 0.910 vs 5.058 ± 1.007	-2.26	0.024*	0.024*
SF, Avg.	F1	ABAPC-LLM vs MPC-LLM	0.668 ± 0.070 vs 0.634 ± 0.073	4.31	1.62e-5***	1.62e-5***
LT, Avg.	SHD	ABAPC-LLM vs ABAPC	4.868 ± 1.114 vs 5.676 ± 1.387	-5.05	4.31e-7***	8.62e-7***
LT, Avg.	SHD	MPC-LLM vs ABAPC	4.912 ± 0.816 vs 5.676 ± 1.387	-4.88	1.06e-6***	1.06e-6***
LT, Avg.	F1	MPC-LLM vs BOSS	0.739 ± 0.052 vs 0.626 ± 0.166	17.32	3.32e-67***	3.32e-67***
LT, Avg.	F1	ABAPC-LLM vs BOSS	0.731 ± 0.079 vs 0.626 ± 0.166	17.83	4.11e-71***	8.21e-71***
Avg.	SHD	ABAPC-LLM vs MPC-LLM	4.802 ± 0.942 vs 5.193 ± 0.793	-4.38	1.18e-5***	1.18e-5***
Avg.	F1	ABAPC-LLM vs MPC-LLM	0.706 ± 0.069 vs 0.696 ± 0.057	2.42	0.015*	0.015*

Table 13: Welch tests for Table 4. Significance levels for both reported p-values are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$.

Group	Metric	Methods	Means±Std	t	p-value	p_{BH}
$\alpha = 0.01, \lvert\mathbf{V}\rvert = 5$	SHD	ABAPC-LLM vs MPC-LLM	1.230 ± 0.369 vs 1.500 ± 0.318	-5.60	2.19e-8***	2.19e-8***
$\alpha = 0.01, \lvert\mathbf{V}\rvert = 5$	F1	ABAPC-LLM vs MPC-LLM	0.812 ± 0.049 vs 0.795 ± 0.037	1.98	0.048*	0.048*
$\alpha = 0.01, \lvert\mathbf{V}\rvert = 10$	SHD	ABAPC-LLM vs ABAPC	5.626 ± 1.176 vs 6.166 ± 1.442	-5.03	4.89e-7***	4.89e-7***
$\alpha = 0.01, \lvert\mathbf{V}\rvert = 10$	F1	ABAPC-LLM vs MPC-LLM	0.653 ± 0.084 vs 0.627 ± 0.077	4.38	1.17e-5***	1.17e-5***
$\alpha = 0.01, \lvert\mathbf{V}\rvert = 15$	SHD	ABAPC-LLM vs MPC-LLM	7.550 ± 1.281 vs 7.874 ± 1.038	-2.94	0.003**	0.003**
$\alpha = 0.01, \lvert\mathbf{V}\rvert = 15$	F1	MPC-LLM vs ABAPC-LLM	0.667 ± 0.058 vs 0.654 ± 0.072	2.52	0.012*	0.012*
$\alpha = 0.05, \lvert\mathbf{V}\rvert = 5$	SHD	MPC-LLM vs ABAPC	1.170 ± 0.583 vs 1.894 ± 0.695	-16.90	4.28e-64***	8.56e-64***
$\alpha = 0.05, \lvert\mathbf{V}\rvert = 5$	SHD	ABAPC-LLM vs ABAPC	1.233 ± 0.514 vs 1.894 ± 0.695	-14.12	2.85e-45***	2.85e-45***
$\alpha = 0.05, \lvert\mathbf{V}\rvert = 5$	F1	MPC-LLM vs ABAPC-LLM	0.850 ± 0.073 vs 0.817 ± 0.064	4.79	1.70e-6***	1.70e-6***
$\alpha = 0.05, \lvert\mathbf{V}\rvert = 10$	SHD	MPC-LLM vs MPC	4.531 ± 1.516 vs 5.267 ± 1.791	-8.48	2.23e-17***	2.23e-17***
$\alpha = 0.05, \lvert\mathbf{V}\rvert = 10$	F1	MPC-LLM vs MPC	0.734 ± 0.103 vs 0.691 ± 0.131	9.40	5.46e-21***	5.46e-21***
$\alpha = 0.05, \lvert\mathbf{V}\rvert = 15$	SHD	MPC-LLM vs MPC	6.439 ± 1.769 vs 7.289 ± 1.971	-7.84	4.63e-15***	4.63e-15***
$\alpha = 0.05, \lvert\mathbf{V}\rvert = 15$	F1	MPC-LLM vs MPC	0.721 ± 0.092 vs 0.684 ± 0.106	8.67	4.19e-18***	4.19e-18***

Table 14: Welch tests for the `earthquake` rows in Table 5. Significance levels for both reported p-values are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$.

Group	Metric	Methods	Means±Std	t	p-value	p_{BH}
earthquake	SHD	MPC vs NOTEARS-MLP	0.040 ± 0.200 vs 2.120 ± 1.170	-12.39	2.92e-35***	8.77e-35***
earthquake	SHD	MPC-LLM vs NOTEARS-MLP	0.040 ± 0.200 vs 2.120 ± 1.170	-12.39	2.92e-35***	8.77e-35***
earthquake	SHD	ABAPC-LLM vs NOTEARS-MLP	0.080 ± 0.340 vs 2.120 ± 1.170	-11.84	2.48e-32***	4.97e-32***
earthquake	SHD	ABAPC vs NOTEARS-MLP	0.100 ± 0.463 vs 2.120 ± 1.170	-11.35	7.25e-30***	1.09e-29***
earthquake	SHD	GRaSP vs NOTEARS-MLP	0.160 ± 0.790 vs 2.120 ± 1.170	-9.82	9.49e-23***	1.14e-22***
earthquake	SHD	BOSS vs NOTEARS-MLP	0.400 ± 1.210 vs 2.120 ± 1.170	-7.23	4.98e-13***	4.98e-13***
earthquake	Precision	GRaSP vs NOTEARS-MLP	1.000 vs 0.500 ± 0.270	13.09	3.54e-39***	1.06e-38***
earthquake	Precision	BOSS vs NOTEARS-MLP	1.000 vs 0.500 ± 0.270	13.09	3.54e-39***	1.06e-38***
earthquake	Precision	MPC vs NOTEARS-MLP	0.990 ± 0.040 vs 0.500 ± 0.270	12.69	6.37e-37***	9.56e-37***
earthquake	Precision	MPC-LLM vs NOTEARS-MLP	0.990 ± 0.040 vs 0.500 ± 0.270	12.69	6.37e-37***	9.56e-37***
earthquake	Precision	ABAPC-LLM vs NOTEARS-MLP	0.987 ± 0.052 vs 0.500 ± 0.270	12.52	5.71e-36***	6.85e-36***
earthquake	Precision	ABAPC vs NOTEARS-MLP	0.982 ± 0.080 vs 0.500 ± 0.270	12.10	1.02e-33***	1.02e-33***
earthquake	Recall	MPC vs BOSS	1.000 vs 0.900 ± 0.300	2.36	0.018*	0.044*
earthquake	Recall	MPC-LLM vs BOSS	1.000 vs 0.900 ± 0.300	2.36	0.018*	0.044*
earthquake	Recall	ABAPC-LLM vs BOSS	0.995 ± 0.035 vs 0.900 ± 0.300	2.22	0.026*	0.044*
earthquake	Recall	ABAPC vs BOSS	0.990 ± 0.071 vs 0.900 ± 0.300	2.06	0.039*	0.049*
earthquake	Recall	GRaSP vs BOSS	0.960 ± 0.200 vs 0.900 ± 0.300	1.18	0.239ns	0.239ns
earthquake	F1	GRaSP vs NOTEARS-MLP	1.000 vs 0.500 ± 0.270	13.09	3.54e-39***	8.49e-39***
earthquake	F1	BOSS vs NOTEARS-MLP	1.000 vs 0.500 ± 0.270	13.09	3.54e-39***	8.49e-39***
earthquake	F1	MPC vs NOTEARS-MLP	1.000 ± 0.020 vs 0.500 ± 0.270	13.06	5.66e-39***	8.49e-39***
earthquake	F1	MPC-LLM vs NOTEARS-MLP	1.000 ± 0.020 vs 0.500 ± 0.270	13.06	5.66e-39***	8.49e-39***
earthquake	F1	ABAPC-LLM vs NOTEARS-MLP	0.991 ± 0.041 vs 0.500 ± 0.270	12.70	5.84e-37***	7.00e-37***
earthquake	F1	ABAPC vs NOTEARS-MLP	0.986 ± 0.073 vs 0.500 ± 0.270	12.27	1.30e-34***	1.30e-34***
earthquake	SID	MPC vs BOSS	0.000 vs 1.000 ± 3.030	-2.33	0.020*	0.049*
earthquake	SID	MPC-LLM vs BOSS	0.000 vs 1.000 ± 3.030	-2.33	0.020*	0.049*
earthquake	SID	ABAPC-LLM vs BOSS	0.080 ± 0.566 vs 1.000 ± 3.030	-2.11	0.035*	0.058.
earthquake	SID	ABAPC vs BOSS	0.160 ± 1.131 vs 1.000 ± 3.030	-1.84	0.066.	0.083.
earthquake	SID	GRaSP vs BOSS	0.400 ± 1.980 vs 1.000 ± 3.030	-1.17	0.241ns	0.241ns

Table 15: Welch tests for Table 5, excluding the `earthquake` and `child` rows. Significance levels for both reported p-values are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$.

Group	Metric	Methods	Means±Std	t	p-value	p_{BH}
asia	SHD	MPC-LLM vs ABAPC-LLM	2.840 ± 0.470 vs 3.580 ± 0.710	-6.15	7.89e-10***	7.89e-10***
asia	Precision	ABAPC-LLM vs ABAPC	0.992 ± 0.040 vs 0.684 ± 0.207	10.31	6.65e-25***	9.97e-25***
asia	Precision	MPC-LLM vs ABAPC	0.990 ± 0.030 vs 0.684 ± 0.207	10.32	5.96e-25***	9.97e-25***
asia	Precision	MPC vs ABAPC	0.980 ± 0.080 vs 0.684 ± 0.207	9.41	5.11e-21***	5.11e-21***
asia	Recall	MPC-LLM vs ABAPC-LLM	0.650 ± 0.050 vs 0.557 ± 0.082	6.83	8.49e-12***	8.49e-12***
asia	F1	MPC-LLM vs ABAPC-LLM	0.780 ± 0.040 vs 0.710 ± 0.068	6.22	5.00e-10***	5.00e-10***
asia	SID	MPC-LLM vs ABAPC-LLM	13.720 ± 0.540 vs 15.500 ± 2.308	-5.31	1.10e-7***	1.10e-7***
cancer	SHD	MPC-LLM vs ABAPC	0.680 ± 0.590 vs 2.106 ± 1.148	-7.81	5.65e-15***	5.65e-15***
cancer	SHD	ABAPC-LLM vs ABAPC	0.680 ± 0.587 vs 2.106 ± 1.148	-7.82	5.28e-15***	5.65e-15***
cancer	Precision	ABAPC-LLM vs MPC	0.996 ± 0.028 vs 0.940 ± 0.180	2.17	0.030*	0.058.
cancer	Precision	MPC-LLM vs MPC	0.990 ± 0.050 vs 0.940 ± 0.180	1.89	0.058.	0.058.
cancer	Recall	MPC-LLM vs ABAPC	0.840 ± 0.130 vs 0.478 ± 0.276	8.38	5.51e-17***	1.10e-16***
cancer	Recall	ABAPC-LLM vs ABAPC	0.835 ± 0.148 vs 0.478 ± 0.276	8.04	8.72e-16***	8.72e-16***
cancer	F1	MPC vs BOSS	0.940 ± 0.180 vs 0.770 ± 0.300	3.44	5.91e-4***	0.002**
cancer	F1	ABAPC-LLM vs BOSS	0.901 ± 0.091 vs 0.770 ± 0.300	2.95	0.003**	0.003**
cancer	F1	MPC-LLM vs BOSS	0.900 ± 0.080 vs 0.770 ± 0.300	2.96	0.003**	0.003**
cancer	SID	MPC-LLM vs ABAPC	0.780 ± 1.000 vs 8.070 ± 4.491	-11.20	3.96e-29***	7.93e-29***
cancer	SID	ABAPC-LLM vs ABAPC	0.960 ± 1.324 vs 8.070 ± 4.491	-10.74	6.86e-27***	6.86e-27***
survey	SHD	MPC-LLM vs ABAPC-LLM	2.060 ± 1.040 vs 3.297 ± 0.981	-6.12	9.51e-10***	9.51e-10***
survey	Precision	GRaSP vs MPC-LLM	1.000 vs 0.890 ± 0.120	6.48	9.06e-11***	9.06e-11***
survey	Precision	BOSS vs MPC-LLM	1.000 vs 0.890 ± 0.120	6.48	9.06e-11***	9.06e-11***
survey	Recall	MPC-LLM vs ABAPC-LLM	0.670 ± 0.160 vs 0.457 ± 0.155	6.76	1.38e-11***	1.38e-11***
survey	F1	MPC-LLM vs MPC	0.750 ± 0.140 vs 0.680 ± 0.140	2.50	0.012*	0.012*
survey	SID	MPC-LLM vs ABAPC-LLM	6.320 ± 4.180 vs 13.477 ± 3.868	-8.89	6.36e-19***	6.36e-19***
sachs	SHD	ABAPC-LLM vs ABAPC	4.460 ± 1.264 vs 5.244 ± 1.684	-2.63	0.008**	0.008**
sachs	Precision	ABAPC-LLM vs MPC-LLM	0.759 ± 0.063 vs 0.740 ± 0.020	2.01	0.044*	0.044*
sachs	Recall	ABAPC-LLM vs ABAPC	0.739 ± 0.075 vs 0.693 ± 0.100	2.61	0.009**	0.009**
sachs	F1	ABAPC-LLM vs ABAPC	0.748 ± 0.069 vs 0.706 ± 0.100	2.47	0.014*	0.014*
sachs	SID	ABAPC-LLM vs ABAPC	21.056 ± 6.983 vs 26.212 ± 9.472	-3.10	0.002**	0.002**

Table 16: Welch tests for the `child` rows in Table 5. Significance levels for both reported p-values are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$.

Group	Metric	Methods	Means±Std	t	p-value	p_{BH}
child	SHD	ABAPC-LLM vs MPC	4.417 ± 2.009 vs 9.160 ± 4.130	-7.30	2.82e-13***	8.45e-13***
child	SHD	MPC-LLM vs MPC	5.060 ± 1.800 vs 9.160 ± 4.130	-6.44	1.23e-10***	1.85e-10***
child	SHD	ABAPC vs MPC	5.372 ± 3.250 vs 9.160 ± 4.130	-5.10	3.46e-7***	3.46e-7***
child	Precision	BOSS vs ABAPC-LLM	0.960 ± 0.080 vs 0.894 ± 0.050	4.94	7.67e-7***	7.67e-7***
child	Recall	ABAPC-LLM vs MPC	0.824 ± 0.080 vs 0.630 ± 0.170	7.31	2.59e-13***	7.76e-13***
child	Recall	MPC-LLM vs MPC	0.800 ± 0.070 vs 0.630 ± 0.170	6.54	6.22e-11***	9.32e-11***
child	Recall	ABAPC vs MPC	0.786 ± 0.129 vs 0.630 ± 0.170	5.17	2.38e-7***	2.38e-7***
child	F1	ABAPC-LLM vs MPC-LLM	0.855 ± 0.052 vs 0.830 ± 0.050	2.48	0.013*	0.013*
child	SID	ABAPC-LLM vs MPC-LLM	72.653 ± 24.836 vs 87.820 ± 26.340	-2.96	0.003**	0.006**
child	SID	ABAPC vs MPC-LLM	84.833 ± 45.226 vs 87.820 ± 26.340	-0.40	0.687ns	0.687ns

Table 17: Non-significant best-vs-comparator tests that justify additional bolded methods in Table 5. Significance levels for both reported p-values are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$.

Group	Metric	Methods	Means±Std	t	p-value	p_{BH}
asia	Precision	ABAPC-LLM vs MPC	0.992 ± 0.040 vs 0.980 ± 0.080	0.95	0.342ns	0.391ns
asia	Precision	ABAPC-LLM vs MPC-LLM	0.992 ± 0.040 vs 0.990 ± 0.030	0.28	0.776ns	0.776ns
cancer	SHD	MPC-LLM vs ABAPC-LLM	0.680 ± 0.590 vs 0.680 ± 0.587	0.00	1.000ns	1.000ns
cancer	Precision	ABAPC-LLM vs MPC-LLM	0.996 ± 0.028 vs 0.990 ± 0.050	0.74	0.460ns	0.460ns
cancer	Recall	MPC-LLM vs ABAPC-LLM	0.840 ± 0.130 vs 0.835 ± 0.148	0.18	0.858ns	0.858ns
cancer	F1	MPC vs MPC-LLM	0.940 ± 0.180 vs 0.900 ± 0.080	1.44	0.151ns	0.167ns
cancer	F1	MPC vs ABAPC-LLM	0.940 ± 0.180 vs 0.901 ± 0.091	1.38	0.167ns	0.167ns
cancer	SID	MPC-LLM vs ABAPC-LLM	0.780 ± 1.000 vs 0.960 ± 1.324	−0.77	0.443ns	0.443ns
earthquake	SHD	MPC vs GRaSP	0.040 ± 0.200 vs 0.160 ± 0.790	−1.04	0.298ns	0.476ns
earthquake	SHD	MPC vs BOSS	0.040 ± 0.200 vs 0.400 ± 1.210	−2.08	0.038*	0.076.
earthquake	SHD	MPC vs MPC-LLM	0.040 ± 0.200 vs 0.040 ± 0.200	0.00	1.000ns	1.000ns
earthquake	SHD	MPC vs ABAPC	0.040 ± 0.200 vs 0.100 ± 0.463	−0.84	0.400ns	0.534ns
earthquake	SHD	MPC vs ABAPC-LLM	0.040 ± 0.200 vs 0.080 ± 0.340	−0.72	0.474ns	0.541ns
earthquake	Precision	GRaSP vs BOSS	1.000 vs 1.000	0.00	1.000ns	1.000ns
earthquake	Precision	GRaSP vs MPC	1.000 vs 0.990 ± 0.040	1.77	0.077.	0.105ns
earthquake	Precision	GRaSP vs MPC-LLM	1.000 vs 0.990 ± 0.040	1.77	0.077.	0.105ns
earthquake	Precision	GRaSP vs ABAPC	1.000 vs 0.982 ± 0.080	1.59	0.112ns	0.128ns
earthquake	Precision	GRaSP vs ABAPC-LLM	1.000 vs 0.987 ± 0.052	1.76	0.079.	0.105ns
earthquake	Recall	MPC vs GRaSP	1.000 vs 0.960 ± 0.200	1.41	0.157ns	0.252ns
earthquake	Recall	MPC vs MPC-LLM	1.000 vs 1.000	0.00	1.000ns	1.000ns
earthquake	Recall	MPC vs ABAPC	1.000 vs 0.990 ± 0.071	1.00	0.317ns	0.363ns
earthquake	Recall	MPC vs ABAPC-LLM	1.000 vs 0.995 ± 0.035	1.00	0.317ns	0.363ns
earthquake	F1	GRaSP vs BOSS	1.000 vs 1.000	0.00	1.000ns	1.000ns
earthquake	F1	GRaSP vs MPC	1.000 vs 1.000 ± 0.020	0.00	1.000ns	1.000ns
earthquake	F1	GRaSP vs MPC-LLM	1.000 vs 1.000 ± 0.020	0.00	1.000ns	1.000ns
earthquake	F1	GRaSP vs ABAPC	1.000 vs 0.986 ± 0.073	1.39	0.164ns	0.263ns
earthquake	F1	GRaSP vs ABAPC-LLM	1.000 vs 0.991 ± 0.041	1.63	0.104ns	0.208ns
earthquake	SID	MPC vs GRaSP	0.000 vs 0.400 ± 1.980	−1.43	0.153ns	0.245ns
earthquake	SID	MPC vs MPC-LLM	0.000 vs 0.000	0.00	1.000ns	1.000ns
earthquake	SID	MPC vs ABAPC	0.000 vs 0.160 ± 1.131	−1.00	0.317ns	0.363ns
earthquake	SID	MPC vs ABAPC-LLM	0.000 vs 0.080 ± 0.566	−1.00	0.317ns	0.363ns
survey	Precision	GRaSP vs BOSS	1.000 vs 1.000	0.00	1.000ns	1.000ns
child	SHD	ABAPC-LLM vs MPC-LLM	4.417 ± 2.009 vs 5.060 ± 1.800	−1.69	0.092.	0.092.
child	SHD	ABAPC-LLM vs ABAPC	4.417 ± 2.009 vs 5.372 ± 3.250	−1.77	0.077.	0.088.
child	Recall	ABAPC-LLM vs MPC-LLM	0.824 ± 0.080 vs 0.800 ± 0.070	1.61	0.107ns	0.107ns
child	Recall	ABAPC-LLM vs ABAPC	0.824 ± 0.080 vs 0.786 ± 0.129	1.78	0.075.	0.085.
child	SID	ABAPC-LLM vs ABAPC	72.653 ± 24.836 vs 84.833 ± 45.226	−1.67	0.095.	0.095.

Table 18: Welch tests for Table 9. Significance levels for both reported p-values are: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, . $p < 0.1$.

Group	Metric	Methods	Means±Std	t	p-value	p_{BH}
$\lvert\mathbf{V}\rvert = 5$	SHD	ABAPC-LLM (Gemini) vs MPC-LLM (Gemini)	1.186 ± 0.375 vs 1.328 ± 0.331	-2.57	0.010*	0.010*
$\lvert\mathbf{V}\rvert = 5$	Precision	MPC-LLM (Gemini) vs ABAPC-LLM (Gemini)	0.952 ± 0.027 vs 0.902 ± 0.043	7.39	1.50e-13***	1.50e-13***
$\lvert\mathbf{V}\rvert = 5$	Recall	ABAPC-LLM (Gemini) vs MPC-LLM (Gemini)	0.778 ± 0.063 vs 0.753 ± 0.053	2.20	0.028*	0.028*
$\lvert\mathbf{V}\rvert = 5$	F1	ABAPC-LLM (Gemini) vs ABAPC-LLM (GPT)	0.828 ± 0.048 vs 0.755 ± 0.084	7.82	5.42e-15***	1.08e-14***
$\lvert\mathbf{V}\rvert = 5$	F1	MPC-LLM (Gemini) vs ABAPC-LLM (GPT)	0.822 ± 0.041 vs 0.755 ± 0.084	6.56	5.37e-11***	5.37e-11***
$\lvert\mathbf{V}\rvert = 10$	SHD	ABAPC-LLM (GPT) vs MPC-LLM (Gemini)	5.755 ± 1.417 vs 6.197 ± 1.095	-3.63	2.81e-4***	5.62e-4***
$\lvert\mathbf{V}\rvert = 10$	SHD	ABAPC-LLM (Gemini) vs MPC-LLM (Gemini)	5.984 ± 1.257 vs 6.197 ± 1.095	-1.86	0.063.	0.063.
$\lvert\mathbf{V}\rvert = 10$	Precision	BOSS vs MPC-LLM (GPT)	0.884 ± 0.201 vs 0.860 ± 0.095	3.80	1.43e-4***	1.43e-4***
$\lvert\mathbf{V}\rvert = 10$	Recall	ABAPC-LLM (GPT) vs ABAPC-LLM (Gemini)	0.569 ± 0.101 vs 0.542 ± 0.090	4.71	2.50e-6***	2.50e-6***
$\lvert\mathbf{V}\rvert = 10$	F1	ABAPC-LLM (GPT) vs MPC-LLM (Gemini)	0.655 ± 0.102 vs 0.634 ± 0.084	3.59	3.27e-4***	3.27e-4***
$\lvert\mathbf{V}\rvert = 15$	SHD	ABAPC-LLM (Gemini) vs ABAPC	7.508 ± 1.292 vs 7.863 ± 1.602	-2.67	0.007**	0.022*
$\lvert\mathbf{V}\rvert = 15$	SHD	ABAPC-LLM (GPT) vs ABAPC	7.575 ± 1.513 vs 7.863 ± 1.602	-2.14	0.033*	0.049*
$\lvert\mathbf{V}\rvert = 15$	SHD	MPC-LLM (Gemini) vs ABAPC	7.616 ± 1.052 vs 7.863 ± 1.602	-1.85	0.064.	0.064.
$\lvert\mathbf{V}\rvert = 15$	Precision	MPC-LLM (GPT) vs BOSS	0.941 ± 0.064 vs 0.892 ± 0.176	14.36	9.39e-47***	2.82e-46***
$\lvert\mathbf{V}\rvert = 15$	Precision	MPC-LLM (Gemini) vs BOSS	0.934 ± 0.049 vs 0.892 ± 0.176	12.27	1.33e-34***	1.99e-34***
$\lvert\mathbf{V}\rvert = 15$	Precision	MPC vs BOSS	0.934 ± 0.068 vs 0.892 ± 0.176	12.14	6.40e-34***	6.40e-34***
$\lvert\mathbf{V}\rvert = 15$	Recall	ABAPC-LLM (Gemini) vs MPC-LLM (Gemini)	0.573 ± 0.073 vs 0.557 ± 0.060	2.96	0.003**	0.006**
$\lvert\mathbf{V}\rvert = 15$	Recall	ABAPC-LLM (GPT) vs MPC-LLM (Gemini)	0.570 ± 0.086 vs 0.557 ± 0.060	2.35	0.019*	0.019*
$\lvert\mathbf{V}\rvert = 15$	F1	MPC-LLM (Gemini) vs MPC-LLM (GPT)	0.691 ± 0.055 vs 0.671 ± 0.068	3.71	2.09e-4***	2.09e-4***