CAUTION: KNOWING WHEN TO TRUST LLMS FOR ENSEMBLE CAUSAL DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Causal discovery from observational data remains challenging due to the fundamental limitations of purely statistical methods, such as statistical distinguishability within equivalence classes and sensitivity to finite sample sizes. While large language models (LLMs) offer a promising source of domain knowledge to complement statistical inference, existing LLM-augmented methods are vulnerable to LLM errors. A central challenge is therefore deciding when to trust LLM knowledge and when to trust statistical evidence. We propose `CauTion`, a trust-calibrated ensemble framework that addresses this challenge through annotation-free reliability estimation and selective arbitration. `CauTion` proceeds in three stages. First, an algorithm ensemble utilizes a consensus voting to resolve up to 96% of edges on which algorithms agree, achieving near-perfect accuracy on the filtered consensus edges. Second, a trust-calibrated arbitration mechanism estimates the relative reliability of the LLM and the algorithms via an annotation-free trust calibration procedure, which is then utilized to govern a trust-weighted voting process that restricts LLM arbitration exclusively to edges with unreliable algorithmic evidence. Third, a cycle repair step is applied to guarantee the final causal graph is validly acyclic. Experiments on six datasets demonstrate that `CauTion` consistently outperforms both data-centric and LLM-augmented baselines in SHD and F1, with strong robustness to LLM errors.

1 INTRODUCTION

Causal discovery serves as a cornerstone for scientific inquiry and robust artificial intelligence (Spirtes et al., 2000; Pearl, 2009; Pearl & Mackenzie, 2018; Yao et al., 2021). Existing methods, including constraint-based approaches (Spirtes et al., 2000; 1995), score-based methods (Chickering, 2002; Andrews et al., 2023; Lam et al., 2022), and continuous optimization (Zheng et al., 2018; Bello et al., 2022), have made substantial progress on causal recovery. However, they remain fundamentally limited: constraint-based methods rely on sufficient sample sizes and can only recover a Markov equivalence class (MEC) (Verma & Pearl, 2022); score-based and continuous optimization methods face the local minimum problem and are sensitive to the choice of score function under specific assumptions and hyperparameter choices.

Trained on vast corpora, LLMs encode rich prior knowledge about real-world mechanisms and can reason about causal relationships from variable semantics. LLM-augmented methods (Jiralerspong et al., 2024; Roy et al., 2025; Ban et al., 2025; Abdulaal et al., 2024; Peng et al., 2026b; Vashishtha et al., 2025; Takayama et al., 2025; Long et al., 2023) therefore use such semantic priors as an additional source of evidence to compensate for the limitations of purely data-centric causal discovery algorithms. Existing LLM-augmented causal discovery methods can be broadly grouped into two categories: (1) **Global methods** (Jiralerspong et al., 2024; Roy et al., 2025; Ban et al., 2025; Abdulaal et al., 2024; Peng et al., 2026b; Vashishtha et al., 2025; Takayama et al., 2025), which use LLMs to assess causal relations over all $O(n^2)$ variable pairs, and (2) the **Undirected-only method** (Long et al., 2023) which applies LLMs only to undirected edges in a pre-computed equivalence class. A central challenge in current LLM-augmented causal discovery is deciding when LLM knowledge should override statistical evidence when neither source is known to be reliable. Statistical algorithms may reflect finite-sample errors, different inductive biases, or ambiguity in observational

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

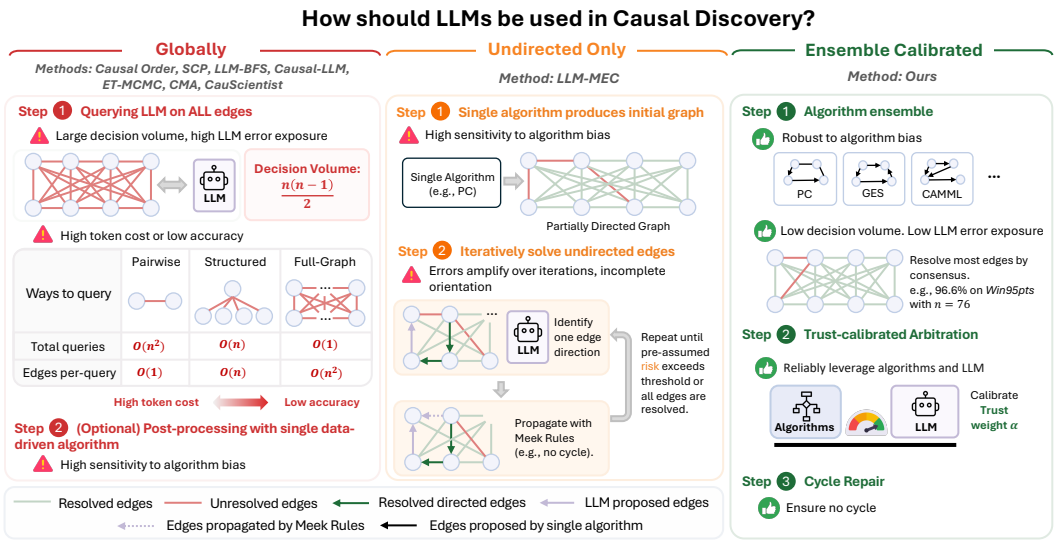


Figure 1: **The challenge of deciding when to trust LLM knowledge and data-centric algorithms in causal discovery. Global methods** expose $O(n^2)$ pairwise decisions to potentially unreliable LLM judgments, and incurs either high token cost or low accuracy due to high reasoning complexity per-query. **Undirected-only methods** restrict LLM intervention to unresolved orientations, but suffer from error amplification over iterations and incomplete orientation. In both settings, reliance on a single data-centric algorithm can further expose the final graph to algorithm-specific errors. On the other hand, `CauTion` addresses these limitations via algorithm ensemble and trust-calibrated arbitration, enabling reliable integration robust to both LLM and single algorithm errors.

evidence, while LLM judgments can introduce incorrect causal relations. Without access to the ground-truth graph, the relative reliability of these sources cannot be directly measured.

CauTion addresses this challenge through annotation-free reliability estimation and selective arbitration. It uses cross-algorithm consensus as proxy supervision, estimates the reliability of statistical and semantic evidence separately for edge existence and direction, and queries the LLM when its response can affect the calibrated voting decision. This design aims to exploit useful domain knowledge while limiting the damage caused by unreliable semantic judgments. `CauTion` operates in three stages. (1) *Algorithm ensemble* aggregates outputs from multiple causal discovery algorithms and resolves variable pairs on which all algorithms agree, thereby reducing individual algorithm bias and limiting LLM intervention to disputed cases. (2) *Trust-calibrated arbitration* estimates the reliability of the LLM and the algorithm ensemble, and resolves disputed edges through trust-weighted aggregation, querying the LLM only when algorithmic evidence is insufficiently reliable. (3) *Cycle repair* ensures that the final graph satisfies the DAG constraint. Together, these stages allow `CauTion` to use reliable algorithmic consensus, selectively invoke LLM domain knowledge for uncertain edges, and combine both sources through calibrated trust. Figure 1 shows the advantages of `CauTion` over existing LLM-augmented methods.

`CauTion` outperforms all baselines in SHD and F1 during experiments, with gains over other LLM-augmented baselines becoming more significant as graph scale grows. On the largest benchmark Win95pts (n=76), `CauTion` achieves an SHD (Structural Hamming Distance) of 28 compared to 69 for the second-best LLM-augmented baseline. Moreover, `CauTion` demonstrates strong robustness to LLM selection, maintaining superior performance across six LLMs of varying capability.

To summarize, our main contributions are:

- *Conceptually*, we formulate LLM-augmented causal discovery as a trust-calibrated arbitration problem. This perspective treats LLMs and data-centric algorithms as complementary but imperfect evidence sources, and emphasizes that LLM knowledge should be incorporated only after its reliability is calibrated against an ensemble of statistical causal discovery algorithms.

- *Methodologically*, we propose `CauTion`, a trust-calibrated ensemble framework for causal discovery. It combines cross-algorithm consensus, reliability-aware LLM arbitration, and DAG-preserving cycle repair to construct reliable and valid causal graphs.

- *Empirically*, we show that `CauTion` consistently improves causal graph recovery across benchmarks and graph scales. `CauTion` achieves stronger recovery performance while reducing unnecessary LLM queries, with gains becoming more pronounced as the number of variables increases.

2 Related Work

Data-driven Methods Purely data-driven causal discovery methods rely on the statistical properties of observational data and span a range of paradigms, including constraint-based methods (e.g., PC (Spirtes et al., 2000), FCI (Spirtes et al., 1995)), score-based methods (e.g., GES (Chickering, 2002), GRaSP (Lam et al., 2022), CaMML (Wallace et al., 1996)), and continuous optimization methods (e.g., NOTEARS (Zheng et al., 2018), DAGMA (Bello et al., 2022)). Constraint-based methods are sensitive to test errors under finite samples and can only recover a Markov equivalence class (MEC), leaving some edge directions unresolved. Score-based methods optimize a scoring function over an exponentially large graph space, relying on greedy or stochastic search heuristics that cannot guarantee global optimality. Moreover, they depend on distributional assumptions in score functions that may not hold in practice. Continuous optimization methods introduce non-convex objectives that are sensitive to hyperparameter choices and thresholding. Finally, these paradigms share a common limitation: without external domain knowledge, purely statistical methods cannot reliably resolve causal structure beyond what the data alone can identify.

LLM-Augmented Methods Existing LLM-augmented causal discovery methods apply LLMs without verifying their domain-specific reliability and reference on at most one data-driven CD algorithm. Therefore, most of them are sensitive to LLM errors and single algorithm bias. Based on how they involve LLMs in the discovery process, they can be broadly categorized into two categories: **global LLM methods** and **undirected only method**, as summarized in Figure 1 and Appendix A Table 2. (1) The **global LLM methods** query the LLM across all variable pairs. LLM-BFS (Jiralerspong et al., 2024) and Causal-LLM (Roy et al., 2025) use LLM alone to generate full graphs. ET-MCMC (Ban et al., 2025) provides more reliability by restricting queries to the edges that the LLM reports high confidence in and post-processing the results with a CD algorithm. CMA (Abdulaal et al., 2024) establishes an iterative feedback loop between LLM proposals and a deep structural causal model (Pawlowski et al., 2020). CauScientist (Peng et al., 2026b) validates each LLM-proposed modification with a BIC score. Most of these methods require the LLM to reason over $O(n^2)$ variable pairs in a single call, leading to degraded decision quality as graph size grows. While Causal Order (Vashishtha et al., 2025) and SCP (Takayama et al., 2025) avoid this by querying one relationship at a time, they incur $O(kn^2)$ and $O(n^2)$ total queries respectively, leading to high token costs. The high decision volume makes global LLM methods difficult to balance decision quality and computational cost. (2) The **undirected only method** LLM-MEC (Long et al., 2023) reduces decision volume by restricting LLM queries to undirected edges within a pre-computed MEC. However, its sequential edge resolution causes errors to accumulate: each orientation is immediately propagated via Meek rules, so a single LLM error can trigger a cascade of forced orientations across the remaining undirected edges. A detailed analysis of each method is shown in Appendix A.

3 Methods

We propose `CauTion`, a framework that reliably integrates calibrated LLM domain knowledge into an ensemble of statistical causal discovery algorithms. The full pipeline is illustrated in Figure 2 and summarized in Algorithm 5. We first formulate the problem setting in Section 3.1. We then introduce the three core components of `CauTion`: algorithm ensemble for extracting cross-algorithm consensus (Section 3.2), trust-calibrated arbitration for resolving disputed edges (Section 3.3), and cycle repair for enforcing the DAG constraint (Section 3.4).

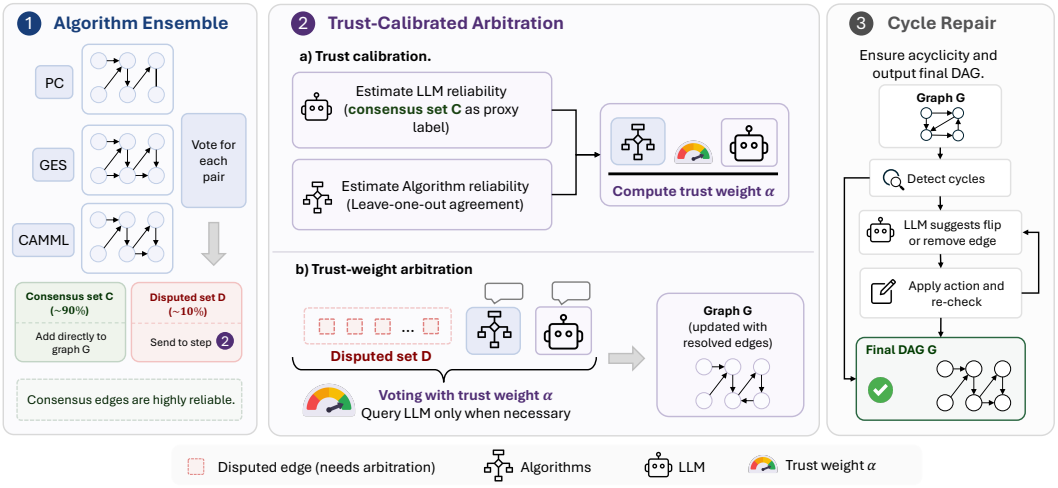


Figure 2: **Overview of `CauTion`.** 1) We first aggregate outputs from multiple causal discovery algorithms (PC, GES, CAMML) and partition variable pairs into consensus and disputed sets. Consensus edges are resolved directly, leaving disputed edges for further arbitration. 2) We use the consensus set as proxy labels to estimate the reliability of LLM, while using leave-one-out to estimate the algorithms. Then compute the calibrated trust weight, which is used for LLM arbitration. 3) A final cycle-repair step ensures acyclicity of the output graph.

3.1 Problem Setup

Let $\mathcal{V} = \{X_1, \ldots, X_n\}$ be a set of n variables, and let $\mathcal{A}$ denote a set of causal discovery algorithms. Each algorithm $A \in \mathcal{A}$ produces a partially directed graph over $\mathcal{V}$. For every unordered variable pair $e = \{i, j\}$, the output of algorithm A is mapped to one of four vote categories:

$$v_A(e) \in \{\text{fwd } (i \to j),\ \text{rev } (j \to i),\ \text{undir } (i - j),\ \text{none}\}$$

where `undir` indicates that the algorithm detected an undirected edge but could not orient it, and `none` indicates no edge. The goal is to aggregate these per-algorithm votes into a single directed acyclic graph (DAG).

3.2 Algorithm Ensemble

For each variable pair e, we accumulate weighted votes across algorithms. Undirected votes contribute equally to both directions:

$$w_{\text{fwd}}(e) = \textstyle\sum_{A \in \mathcal{A}} \big(\mathbf{1}[v_A = \text{fwd}] + 0.5 \cdot \mathbf{1}[v_A = \text{undir}]\big)$$

$$w_{\text{rev}}(e) = \textstyle\sum_{A \in \mathcal{A}} \big(\mathbf{1}[v_A = \text{rev}] + 0.5 \cdot \mathbf{1}[v_A = \text{undir}]\big) \quad w_{\text{none}}(e) = \textstyle\sum_{A \in \mathcal{A}} \mathbf{1}[v_A = \text{none}]$$

A pair $e = \{i, j\}$ is resolved by consensus if any of the following conditions holds: $w_{\text{fwd}}(e) = |\mathcal{A}|$, $w_{\text{rev}}(e) = |\mathcal{A}|$, or $w_{\text{none}}(e) = |\mathcal{A}|$. All remaining pairs form the disputed set $\mathcal{D}$. Only the disputed set is passed to the resolution pipeline in Section 3.3.1.

Consensus-based filtering serves a dual purpose: it reduces the number of pairs requiring LLM arbitration, and it yields high-confidence pseudo-ground-truth labels for downstream calibration. On the Win95pts dataset with $n = 76$, consensus agreement is achieved on the majority of variable pairs (96.6%), with near-perfect accuracy (99.6%)—confirming their suitability as proxy labels.

3.3 Trust-Calibrated Arbitration

3.3.1 Estimating Source Reliability

Before resolving disputed edges, we independently estimate the predictive accuracy of each algorithm and of the LLM directly from the data, requiring no external annotation. Consensus edges

serve as proxy ground-truth labels for LLM calibration, and leave-one-out agreement serves as proxy labels for algorithm calibration. The resulting estimates are used to compute per-task trust weights that govern how much each source is relied upon during voting.

We first estimate the target distribution $(p_{\text{exist}}, p_{\text{none}})$ over $\mathcal{D}$, i.e. the fraction of disputed pairs that involve a directed edge versus no edge. Using the vote weights defined in Section 3.2, the soft edge fraction for each disputed pair and the aggregate target distribution are:

$$p_{\text{exist}}^{(e)} = (w_{\text{fwd}}^{(e)} + w_{\text{rev}}^{(e)})/|\mathcal{A}|, \qquad p_{\text{exist}} = \frac{1}{|\mathcal{D}|}\sum_{e\in\mathcal{D}} p_{\text{exist}}^{(e)}, \qquad p_{\text{none}} = 1 - p_{\text{exist}}$$

This distribution is used to stratify LLM calibration samples and to reweight per-algorithm accuracy estimates to reflect the disputed-edge distribution, as described below.

All LLM queries share a three-stage pipeline: the LLM first infers a domain description from variable names, which is injected into all subsequent prompts; it then infers a semantic interpretation for each variable (e.g., `Dyspnoea [shortness of breath]`), which is appended inline to variable names in all edge-level queries. For each edge-level query, the LLM is presented with variable names and their interpretations. To mitigate position bias, direction queries are issued twice per pair—once in the $A \to B$ framing and once in $B \to A$—and only consistent answers are retained.

3.3.2 LLM Calibration

We estimate LLM accuracy on consensus edges, assuming that the LLM's domain knowledge is uniformly distributed across consensus and disputed pairs—an assumption that holds when both sets are drawn from the same domain and variable space. We treat edge existence and direction as two independent tasks, since they reflect distinct reasoning capabilities: existence requires knowing whether a causal mechanism operates between two variables, while direction requires knowing which variable is the cause and which is the effect. Separate accuracy estimates are therefore maintained for each task, for both the LLM and the algorithm ensemble. We sample $n_{\text{calib}} = 30$ consensus edges stratified by the target distribution:

$$n_{\text{calib,exist}} = \text{round}(n_{\text{calib}} \times p_{\text{exist}}), \qquad n_{\text{calib,none}} = n_{\text{calib}} - n_{\text{calib,exist}}$$

If the edge pool is smaller than $n_{\text{calib,exist}}$, the shortfall rolls over to the none pool.

Existence task. The LLM is queried for whether an edge exists and returns $\hat{e} \in \{\texttt{exist}, \texttt{none}\}$. We compute per-class accuracy conditioned on the ground-truth label d_e^*: $acc_{\text{cond,exist}}$ is the fraction of ground-truth edge pairs correctly identified as `exist`, and $acc_{\text{cond,none}}$ is the fraction of ground-truth none pairs correctly identified as `none`. The effective existence accuracy is reweighted to the target distribution to remove sampling bias:

$$acc_{\text{LLM,exist}} = p_{\text{exist}} \cdot acc_{\text{cond,exist}} + p_{\text{none}} \cdot acc_{\text{cond,none}} \tag{1}$$

Direction task. Each edge-class pair is queried twice to eliminate position bias, as described above; only pairs where both queries return the same direction are counted as consistent. The direction accuracy is then: $acc_{\text{LLM,dir}} = c_{\text{dir}}/n_{\text{dir}}$, where n_{dir} is the number of consistent pairs and c_{dir} is the subset where the consistent answer matches d_e^*.

3.3.3 Algorithm LOO Calibration

We estimate each algorithm's accuracy via leave-one-out (LOO) evaluation: for each pair e, algorithm A's prediction is compared against the majority vote of all remaining algorithms, which serves as a proxy label, with tied edges excluded. For existence, we compute the fraction of proxy-exist pairs and proxy-none pairs correctly predicted by A, then reweight by $(p_{\text{exist}}, p_{\text{none}})$ following Equation 1, yielding acc_{exist}^A. For direction, acc_{dir}^A is the fraction of pairs correctly oriented by A among those where A votes a definite direction, excluding undirected votes. To aggregate across algorithms, we use the mean restricted to algorithms at or above the median:

$$acc_{\text{algo,exist}} = \text{AboveMedianMean}_A\left(acc_{\text{exist}}^A\right), \qquad acc_{\text{algo,dir}} = \text{AboveMedianMean}_A\left(acc_{\text{dir}}^A\right)$$

Since poorly performing algorithms receive lower vote weights during aggregation, the ensemble accuracy is better reflected by the higher-performing half, rather than a mean pulled down by weak algorithms.

3.3.4 TRUST WEIGHTS

Given the calibrated accuracies $acc_{\text{LLM,exist}}$, $acc_{\text{LLM,dir}}$, $acc_{\text{algo,exist}}$, and $acc_{\text{algo,dir}}$, we compute two independent trust weights α_{exist} and α_{dir} that quantify the relative reliability of the LLM versus the algorithm ensemble on each task. Since both tasks reduce to binary decisions, random performance corresponds to an accuracy of 0.5. Trust weights are computed as a power-ratio of above-chance accuracies with exponent $\beta > 1$ (default $\beta = 3$), which sharpens the contrast between sources of similar accuracy:

$$\alpha_{\text{exist}} = \frac{\max(0,\ acc_{\text{LLM,exist}} - 0.5)^{\beta}}{\max(0,\ acc_{\text{LLM,exist}} - 0.5)^{\beta} + \max(0,\ acc_{\text{algo,exist}} - 0.5)^{\beta}},$$

$$\alpha_{\text{dir}} = \frac{\max(0,\ acc_{\text{LLM,dir}} - 0.5)^{\beta}}{\max(0,\ acc_{\text{LLM,dir}} - 0.5)^{\beta} + \max(0,\ acc_{\text{algo,dir}} - 0.5)^{\beta}}$$

A higher α indicates greater relative trust in the LLM. When both sources perform at or below chance, we fall back to $\alpha = 0.5$, treating both sources as equally unreliable.

3.3.5 CALIBRATED VOTING

For each disputed pair $e \in \mathcal{D}$, we first determine whether an edge exists, then determine its direction.

Existence vote. Algorithm votes are aggregated into a weighted existence score:

$$v_{\text{exist}}(e) = \frac{\sum_{A \in \mathcal{A}} acc^{A}_{\text{exist}} \cdot \mathbf{1}[v_A \in \{\text{fwd, rev, undir}\}]}{\sum_{A \in \mathcal{A}} acc^{A}_{\text{exist}}}, \qquad v_{\text{none}}(e) = 1 - v_{\text{exist}}$$

When the LLM is consulted and returns $\hat{e} \in \{\texttt{exist},\ \texttt{none}\}$, the combined score is:

$$q_{\text{exist}} = (1 - \alpha_{\text{exist}}) \cdot v_{\text{exist}} + \alpha_{\text{exist}} \cdot \mathbf{1}[\hat{e} = \texttt{exist}], \qquad q_{\text{none}} = 1 - q_{\text{exist}}$$

with decision $\arg\max(q_{\text{exist}},\ q_{\text{none}})$. In practice, the LLM is only queried when $m_{\text{exist}} = |v_{\text{exist}} - v_{\text{none}}| \le \alpha_{\text{exist}}/(1 - \alpha_{\text{exist}})$, since beyond this margin the algorithm vote is decisive regardless of the LLM response. This further reduces the number of LLM queries.

Direction vote. If an edge is determined to exist, direction is resolved using votes from algorithms that predict a definite direction ($v_A \in \{\text{fwd, rev}\}$):

$$v_{\text{fwd}}(e) = \frac{\sum_{A:\ v_A \in \{\text{fwd, rev}\}} acc^{A}_{\text{dir}} \cdot \mathbf{1}[v_A = \text{fwd}]}{\sum_{A:\ v_A \in \{\text{fwd, rev}\}} acc^{A}_{\text{dir}}}, \qquad v_{\text{rev}}(e) = 1 - v_{\text{fwd}}$$

The combined score incorporating the LLM response $\hat{d} \in \{\texttt{fwd},\ \texttt{rev}\}$ is:

$$q_{\text{fwd}} = (1 - \alpha_{\text{dir}}) \cdot v_{\text{fwd}} + \alpha_{\text{dir}} \cdot \mathbf{1}[\hat{d} = \texttt{fwd}], \qquad q_{\text{rev}} = 1 - q_{\text{fwd}}$$

with decision $\arg\max(q_{\text{fwd}},\ q_{\text{rev}})$. Similar to the existence vote, the LLM is queried only when the algorithm margin $m_{\text{dir}} = |v_{\text{fwd}} - v_{\text{rev}}|$ falls below the analogous threshold $\alpha_{\text{dir}}/(1 - \alpha_{\text{dir}})$. When queried, direction queries follow the double-framing procedure described above; if the two framings are inconsistent, the decision falls back to the algorithm vote alone.

3.4 CYCLE REPAIR

The voting procedure above produces a directed graph but does not guarantee a valid DAG. We therefore apply a cycle repair step to further process the graph. We first detect cycles in the produced graph. If cycles are detected, for each edge e on a detected cycle, we compute the weighted existence and direction margins:

$$m_{\text{exist}}(e) = |v_{\text{exist}} - v_{\text{none}}|, \qquad m_{\text{dir}}(e) = |v_{\text{fwd}} - v_{\text{rev}}|$$

An edge e is designated as a *repair candidate* if it is algorithmically uncertain in either sense:

$$\text{candidate}(e) = \mathbf{1}\big[m_{\text{exist}}(e) < \theta_{\text{exist}}\big] \ \vee\ \mathbf{1}\big[m_{\text{dir}}(e) < \theta_{\text{dir}}\big]$$

where $\theta_{\text{exist}} = \alpha_{\text{exist}}/(1-\alpha_{\text{exist}})$ and $\theta_{\text{dir}} = \alpha_{\text{dir}}/(1-\alpha_{\text{dir}})$ are the same thresholds used in Section 3.3.5.

Repair candidates are then presented to the LLM together with the cycle path and variable interpretations. For each candidate, the LLM may choose from one of two actions: FLIP (reverse the edge direction, available when $m_{\text{dir}}(e) < \theta_{\text{dir}}$) or REMOVE (delete the edge, available when $m_{\text{exist}}(e) < \theta_{\text{exist}}$). Each selected action is applied immediately, and the action is removed from the *available action* set. New cycle check is performed, and if cycles remain, the procedure repeats until all cycles have been removed or there is no *available action* left. If all *available actions* are exhausted and cycles persist, edges on the remaining cycle are removed in ascending order of v_{exist} until the graph is acyclic.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets We evaluate on six datasets from the bnlearn repository (Scutari, 2010)[1]: Cancer (n=5, $|E|$=4) (Korb & Nicholson, 2010), Insurance (n=27, $|E|$=52) (Binder et al., 1997), Water (n=32, $|E|$=66) (Jensen et al., 1989), Alarm (n=37, $|E|$=46) (Beinlich et al., 1989), Barley (n=48, $|E|$=84) (Kristensen & Rasmussen, 2002), and Win95pts (n=76, $|E|$=112) (Heckerman et al., 1995), spanning a range of scales and edge densities. For each dataset, we sample five versions of observations, with 5000 samples in each version.

Metrics Three metrics were used. **Structural Hamming Distance (SHD)** counts the minimum number of edge insertions, deletions, and reversals required to transform the predicted graph into the ground truth, with lower values indicating better performance. **F1 score** measures the harmonic mean of precision and recall over directed edges. **Structural Intervention Distance (SID)** (Peters & Bühlmann, 2015) quantifies structural errors in intervention inference. It yields a single exact value for DAG outputs and lower and upper bounds for CPDAG outputs.

Baselines We compare against three categories of data-centric methods, including **constraint-based** PC (Spirtes et al., 2000) and FCI (Spirtes et al., 1995), **score-based** GES (Chickering, 2002), GRaSP (Lam et al., 2022), and CAMML (Wallace et al., 1996), and **continuous optimization-based** NOTEARS-MLP (Zheng et al., 2020) and DAGMA (Bello et al., 2022). We compare four LLM-augmented methods, including **global** methods LLM-BFS (Jiralerspong et al., 2024), SCP (Takayama et al., 2025), ET-MCMC (Ban et al., 2025), and **undirected-only** method LLM-MEC (Long et al., 2023). We additionally include CORR-LLM, a simple baseline that prompts the LLM with pairwise Pearson correlations to generate the full causal graph in a single query.

LLM backend and majority vote algorithms We use Claude-Sonnet-4.6 as the main LLM backend for all LLM-agumented methods including ours. We use PC, GES and CAMML as the algorithm ensemble for all experiments, selected for their efficiency (Table 10), wide adoption in the literature, and diversity in underlying search strategies.

4.2 PERFORMANCE

Table 1 reports results across six benchmarks. **`CauTion` achieves the best or tied-best mean SHD and F1 on all six benchmarks.** Among the reported exact SID values, `CauTion` also achieves the best or tied-best result on five benchmarks. Water is the exception: LLM-BFS achieves the lowest SID at 374, slightly below CauTion's 377, while CauTion achieves the best SHD and F1.

`CauTion` outperforms all statistical baselines on all metrics. It mitigate algorithm bias by aggregating signals from multiple algorithms, and applies reliable LLM knowledge to compensate for what statistical algorithms cannot resolve alone. **Among LLM-augmented baselines, existing methods scale poorly to larger graphs.** In **global LLM methods**, CORR-LLM, LLM-BFS, ET-MCMC performs well on small networks, but degrades sharply on larger graphs. Though SCP shows higher performance compared to the other three global methods, its token cost is 100 times larger

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

Table 1: Performance across six datasets (SHD ↓, SID ↓; F1 in % ↑). Best values are in **bold**. "—" indicates that the exact SID value is unavailable. Results are averaged across 5 runs (Wu et al., 2025; Peng et al., 2026a). `CauTion` achieves the best or tied-best SHD and F1 on all datasets.

Method	Cancer (n=5)			Insurance (n=27)			Water (n=32)			Alarm (n=37)			Barley (n=48)			Win95pts (n=76)		
	SHD	F1	SID	SHD	F1	SID	SHD	F1	SID	SHD	F1	SID	SHD	F1	SID	SHD	F1	SID
Constraint-based																		
PC	2	50.0	10	25	63.7	—	55	46.1	—	14	85.9	—	56	49.4	—	72	56.6	—
FCI	4	66.7	10	31	70.0	—	59	46.8	—	33	71.8	—	67	60.2	—	101	58.5	—
Score-based																		
GES	**0**	**100.0**	**0**	22	72.9	—	47	50.2	—	8	91.2	—	54	64.0	—	40	80.8	—
GRaSP	2	80.0	—	41	63.3	—	52	48.3	—	13	87.1	—	86	45.4	—	58	73.4	—
CAMML	2	50.3	9	27	60.5	340	55	31.8	567	9	85.8	96	77	34.7	1144	66	67.1	358
Continuous optimization																		
NOTEARS-MLP	3	26.7	10	50	33.8	535	63	9.4	539	52	30.6	427	84	19.3	1191	142	23.3	1010
DAGMA	4	0.0	10	51	27.0	587	58	20.9	522	33	58.0	252	82	5.6	1233	103	43.9	674
LLM-augmented																		
CORR-LLM	**0**	**100.0**	**0**	23	74.7	375	42	53.5	404	16	80.2	110	78	46.1	881	78	58.0	588
LLM-BFS	**0**	**100.0**	**0**	25	73.3	326	54	51.8	**374**	44	49.4	268	121	28.1	1023	147	23.2	1017
SCP	1	70.0	6	20	72.3	—	53	47.3	—	10	87.2	—	57	47.0	—	69	58.7	—
ET-MCMC	0	91.4	1	21	73.7	285	52	44.5	525	10	86.5	63	90	29.0	1218	95	60.7	385
LLM-MEC	2	50.0	10	24	64.1	—	54	44.7	—	13	84.9	—	61	44.3	—	72	56.6	—
Ours	**0**	**100.0**	**0**	**13**	**85.7**	**199**	**39**	**59.2**	377	**5**	**93.8**	**38**	**43**	**68.4**	**743**	**28**	**84.2**	**343**

than ours (Table 10). Though the **undirected only method** LLM-MEC shows more stable behavior and low token cost than global methods, its performance is constrained by its reliance on a single upstream algorithm (PC). Moreover, it incurs 8 × higher run time than `CauTion` due to large computing complexity for MECs. `CauTion` addresses these failure modes by restricting LLM queries to disputed edges and downweighting LLM influence when its estimated domain accuracy is low, limiting exposure to unreliable LLM responses. The gains over LLM-augmented baselines become substantially more pronounced as graph scale grows, consistent with the framework design: larger graphs yield more edges, amplifying the value of consensus filtering and selective LLM integration.

To evaluate performance beyond published benchmarks, we additionally test `CauTion` on Airport, a newly constructed synthetic Bayesian network with 15 variables and 23 directed edges. We compare `CauTion` against the same baselines as in the main experiments. `CauTion` achieves the best or tied-best performance across all three metrics on this network, with construction details and full results provided in Appendix C.8.

4.3 Ablation Study

We evaluate the contribution of each component in our pipeline by comparing the following ablated strategies: *Consensus only*, which accepts consensus edges and discards all disputed edges; *LLM Only*, which predicts all variable pairs using the LLM alone; *Majority Vote*, which resolves each disputed edge by selecting the option with the most votes across algorithms; *Cons + LLM*, which resolves all disputed edges using LLM responses alone.

Shown in Figure 3, several trends emerge. First, selecting *Consensus only* is not enough for causal discovery, leaving many edges unresolved. *Majority Vote* outperforms individual algorithms, confirming that **ensemble aggregation mitigates single algorithm bias**. However, without LLM domain knowledge, it remains constrained by the inherent limitations of purely statistical inference. On the other hand, *LLM Only* achieves competitive performance on small graphs utilizing domain knowledge but degrades sharply at larger scale graph. *Cons + LLM* demonstrates that **combining algorithmic consensus with LLM knowledge substantially reduces LLM error exposure**. However, its suboptimal performance on Alarm, and Win95pts compared to *Majority Vote* suggests that LLM reliability on disputed edges is overestimated. `CauTion` outperforms both *Majority Vote* and

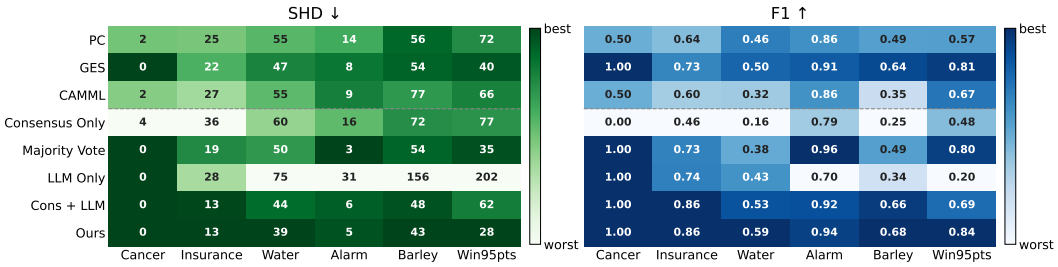


Figure 3: Ablation study with various edge decision strategies. Color is normalized column-wise. The three algorithms used in ensemble aggregation are included above the dashed line for reference.

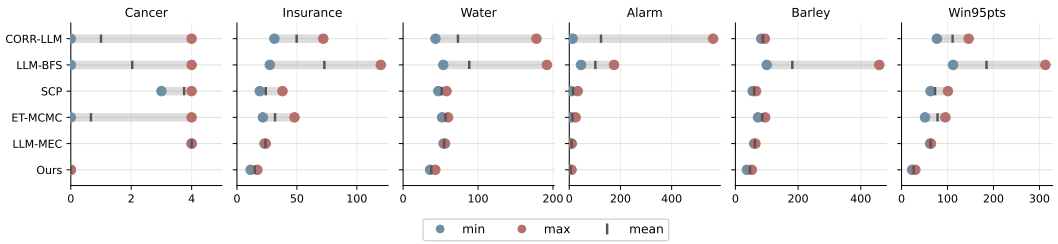


Figure 4: Cross-model robustness of LLM-augmented methods in terms of SHD. Each horizontal bar spans the minimum to maximum SHD achieved across six LLM backends, where shorter bars indicate greater robustness to backend selection.

Cons + LLM on disputed edges, demonstrating that **calibrated trust weighting effectively balances the respective strengths of the algorithm ensemble and LLM domain knowledge**.

4.4 Robustness to LLM Model Selection

We further evaluate robustness to LLM backend selection. We test six LLMs with varying levels of capability: Claude-Sonnet-4.6, GPT-5.2, Llama-3.3-70B-Instruct, DeepSeek-V3.2, Qwen3-8B, and Llama-3.1-8B-Instruct, covering both strong and weak models to test sensitivity to LLM errors. Figure 4 reports the SHD range across models for each LLM-augmented method and dataset. **`CauTion` achieves the lowest SHD on all six datasets with low performance fluctuation across LLM backends, demonstrating robustness to LLM errors.** Among **global LLM methods**, fully LLM-driven methods (CORR-LLM, LLM-BFS) exhibit severe sensitivity on large graphs. CORR-LLM reaches SHD 562 on Alarm for the worst backend (Llama-3.1-8B-Instruct) compared to 13 for the best. LLM-BFS similarly ranges from 100 to 459 on Barley, with Qwen3-8B achieving 100 and DeepSeek-V3.2 collapses to 459, indicating that LLM methods alone are fragile in practice causal discovery. SCP and ET-MCMC are more stable than CORR-LLM and LLM-BFS, but still exhibit substantially higher cross-model variance and consistently worse performance than `CauTion`. **The undirected-only method** LLM-MEC shows little cross-model variance, but remains bottlenecked by its single upstream algorithm (PC). More detailed comparisons in terms of SHD and F1 are provided in Appendix C.6 (Figures 6–7), where x-axes are zoomed in per dataset to better highlight cross-model variance. Figures 8 and 9 further provide radar plots showing the performance of all backbend models across datasets and methods.

5 Conclusion

In conclusion, we present `CauTion`, a trust-calibrated ensemble framework for LLM-augmented causal discovery. By combining cross-algorithm consensus filtering, annotation-free reliability estimation, and calibrated trust-weighted voting, `CauTion` addresses two fundamental limitations of existing approaches: the unverified reliability of LLM domain knowledge and the vulnerability of single-algorithm methods to algorithm-specific bias. Experiments across six benchmarks of varying scale demonstrate that `CauTion` consistently outperforms all baselines in SHD and F1.

AI USE STATEMENT

In this work, we used generative AI tools for polish writing. We have not used generative AI tools for retrieval and discovery, research ideation or execution, draft sections of the paper, generating synthetic datasets, and providing mathematical claims are not applicable to this work. We have reviewed all AI-assisted work. We reviewed the AI-edited text for clarity and coherence and revised it where necessary. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we describe the `CauTion` method in Section 3 and provide the full algorithmic pseudocode in Appendix B. The LLM prompt templates and query settings are provided in Appendix B.1. Section 4.1 specifies the benchmarks, sample sizes, LLM backend, and ensemble algorithms, while Appendices C.1–C.3 detail data generation, evaluation metrics, and baseline implementations. Appendix C.4 reports results across independently sampled dataset versions, together with the hardware setup, runtime, and token usage. An anonymized implementation is available at `https://anonymous.4open.science/r/CauTion-37C6`.

REFERENCES

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

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 *Advances in Neural Information Processing Systems*, volume 36, pp. 63945–63956, 2023.

Ankur Ankan and Johannes Textor. pgmpy: A python toolkit for bayesian networks. *Journal of Machine Learning Research*, 25(265):1–8, 2024. URL `http://jmlr.org/papers/v25/23-0487.html`.

Taiyu Ban, Lyuzhou Chen, Derui Lyu, Xiangyu Wang, Qinrui Zhu, Qiang Tu, and Huanhuan Chen. Integrating large language model for improved causal discovery. *IEEE Transactions on Artificial Intelligence*, 6(11):3030–3042, 2025. doi: 10.1109/TAI.2025.3560927.

Ingo A Beinlich, Henri Jacques Suermondt, R Martin Chavez, and Gregory F Cooper. The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In *AIME 89: Second European Conference on Artificial Intelligence in Medicine, London, August 29th–31st 1989. Proceedings*, pp. 247–256. Springer, 1989.

Kevin Bello, Bryon Aragam, and Pradeep Ravikumar. DAGMA: Learning DAGs via M-matrices and a Log-Determinant Acyclicity Characterization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 8226–8239, 2022.

John Binder, Daphne Koller, Stuart Russell, and Keiji Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29(2):213–244, 1997.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3:507–554, 2002.

David Heckerman, John S Breese, and Koos Rommelse. Decision-theoretic troubleshooting. *Communications of the ACM*, 38(3):49–57, 1995.

FV Jensen, U Kjærulff, KG Olesen, and J Pedersen. An expert system for control of waste water treatment—a pilot project. Technical report, Technical report, Judex Datasystemer A/S, Aalborg, 1989. In Danish, 1989.

Thomas Jiralerspong, Xiaoyin Chen, Yash More, Vedant Shah, and Yoshua Bengio. Efficient causal graph discovery using large language models. In *ICLR 2024 Workshop: How Far Are We From AGI*, 2024. URL https://openreview.net/forum?id=5RBUTx75yr.

Kevin B Korb and Ann E Nicholson. *Bayesian Artificial Intelligence*. CRC Press, 2nd edition, 2010.

Kristian Kristensen and Ilse A Rasmussen. The use of a bayesian network in the design of a decision support system for growing malting barley without use of pesticides. *Computers and Electronics in Agriculture*, 33(3):197–217, 2002.

Wai-Yin Lam, Bryan Andrews, and Joseph Ramsey. Greedy relaxations of the sparsest permutation algorithm. In *Uncertainty in Artificial Intelligence*, pp. 1052–1062. PMLR, 2022.

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

Nick Pawlowski, Daniel Coelho de Castro, and Ben Glocker. Deep structural causal models for tractable counterfactual inference. *Advances in neural information processing systems*, 33:857–869, 2020.

Judea Pearl. *Causality*. Cambridge University Press, 2009.

Judea Pearl and Dana Mackenzie. The book of why. 2018.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Bo Peng, Sirui Chen, Jiaguo Tian, Yu Qiao, and Chaochao Lu. Causcale: Neural causal discovery at scale. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026a.

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

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

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

Marco Scutari. Learning bayesian networks with the bnlearn r package. *Journal of statistical software*, 35:1–22, 2010.

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

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

Masayuki Takayama, Tadahisa Okuda, Thong Pham, Tatsuyoshi Ikenoue, Shingo Fukuma, Shohei Shimizu, and Akiyoshi Sannai. Integrating large language models in causal discovery: A statistical causal approach. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Reh1S8rxfh.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9juyeCqL0u.

Thomas S Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Probabilistic and causal inference: The works of Judea Pearl*, pp. 221–236. 2022.

Chris S Wallace, Kevin B Korb, and Honghua Dai. Causal discovery via mml. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 516–524, 1996.

Menghua Wu, Yujia Bao, Regina Barzilay, and Tommi Jaakkola. Sample, estimate, aggregate: A recipe for causal discovery foundation models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=h434zx5SX0.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *International Conference on Learning Representations*, volume 2024, pp. 23650–23678, 2024.

Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. A survey on causal inference. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(5):1–46, 2021.

Keli Zhang, Shengyu Zhu, Marcus Kalander, Ignavier Ng, Junjian Ye, Zhitang Chen, and Lujia Pan. gcastle: A python toolbox for causal discovery. *arXiv preprint arXiv:2111.15155*, 2021.

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

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*, pp. 3414–3425. PMLR, 2020.

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. *Journal of Machine Learning Research*, 25(60):1–8, 2024.

A Related Work Details

A.1 Data-driven Baselines

A.1.1 Constraint-Based Methods

PC (Spirtes et al., 2000). PC is a landmark constraint-based algorithm that recovers causal structure via two phases: skeleton discovery through conditional independence (CI) tests, followed by orientation of v-structures and propagation via Meek rules. PC outputs a Completed Partially Directed Acyclic Graph (CPDAG) representing the MEC of the true DAG, leaving edges whose directions cannot be determined from observational data unoriented. It is sensitive to CI test errors under finite samples during skeleton discovery phase, and its errors propagate through the orientation phase.

FCI (Spirtes et al., 1995). FCI extends PC to handle latent confounders and selection bias, outputting Partial Ancestral Graph (PAG) rather than CPDAG. FCI introduces additional edge marks (circle, tail, arrowhead) that encode uncertainty about ancestral relationships. It inherits PC's sensitivity to CI test errors and error propagation.

A.2 Score-Based Methods

Unless otherwise noted, methods in this category search over equivalence classes and output a CPDAG, leaving some edge directions unresolved from observational data alone.

GES (Chickering, 2002). GES performs a two-phase greedy search over the space of CPDAGs: a forward phase that adds edges to maximize a decomposable score (e.g., BIC, BDeu), followed by a backward phase that removes edges. Its performance is sensitive to the choice of score function (usually BIC or BDeu) and susceptible to local optima due to its greedy nature.

GRaSP (Lam et al., 2022). GRaSP is a permutation-based method that leverages the Sparsest Permutation (SP) principle, searching for the causal ordering that yields the sparsest DAG under a given score. It improves upon earlier SP-based methods in scalability but remains computationally expensive on dense graphs.

CAMML (Wallace et al., 1996). CAMML is a Bayesian score-based method that uses the Minimum Message Length (MML) principle to search over DAGs via MCMC. Unlike the above methods, CAMML outputs a fully oriented DAG rather than a CPDAG. However, its MCMC search scales poorly to large graphs: the DAG space grows super-exponentially with the number of variables, making it difficult to achieve adequate mixing and convergence in practice.

A.3 Continuous Optimization Methods

NOTEARS-MLP (Zheng et al., 2020). NOTEARS-MLP extends the original NOTEARS framework (Zheng et al., 2018) to non-linear relationships by parameterizing each variable's conditional distribution with a multilayer perceptron (MLP). The acyclicity constraint is enforced via a smooth algebraic characterization, converting the combinatorial graph search into a continuous optimization problem. While this allows gradient-based optimization, the resulting objective is highly non-convex due to both the non-linear acyclicity constraint and the MLP parameterization, making the optimization prone to local optima and sensitive to initialization. In practice, the method requires careful tuning of the regularization strength and the thresholding applied to continuous edge weights to recover a discrete graph structure.

DAGMA (Bello et al., 2022). DAGMA replaces the NOTEARS acyclicity constraint with a log-determinant-based characterization that avoids numerical instability in the original formulation. While more stable than NOTEARS in optimization, DAGMA shares the same fundamental limitations: non-convex objectives, hyperparameter sensitivity, and reliance on thresholding to obtain a discrete graph from continuous edge weights.

Method	Decision Volume	LLM Error Exposure	Algorithmic Bias Exposure	Uncertainty Quantification	DAG
Category I: Global LLM Methods					
LLM-BFS	$O(n^2)$	High	None	✗	✓
Causal-LLM	$O(n^2)$	High	None	✗	✓
Causal Order	$O(kn^2)$	High	High	✗	✓
SCP	$O(n^2)$	High	High	✗	✗
ET-MCMC	$O(n^2)$	High	High	✗	✓
CMA	$O(Tn^2)$	High	High	✗	✗
CauScientist	$O(Tn^2)$	Medium	High	✗	✓
Category II: Undirected Only Methods					
LLM-MEC	Undirected only	Medium	High	✗	✗
Category III: Ensemble Calibrated (Ours)					
`CauTion` (Ours)	Disputed only	Low	Low	✓	✓

Table 2: Comparison of LLM-integrated causal discovery methods. *Decision Volume*: the total number of edges that the LLM is required to analyze during causal structure learning. *LLM Error Exposure*: how much the final graph is affected by incorrect LLM predictions. *Algorithmic Bias Exposure*: degree to which the final graph is subject to errors from a single statistical algorithm. None indicates no statistical algorithm is used. *Uncertainty Quantification*: whether LLM reliability is explicitly estimated from data. *DAG*: whether the output graph is guaranteed to be a DAG.

A.4 LLM-Augmented Baselines

LLM-BFS (Jiralerspong et al., 2024). LLM-BFS constructs the causal graph via a Breadth-First Search (BFS) over the variable space, querying the LLM over all variable pairs with a total decision volume of $O(n^2)$. The procedure begins by asking the LLM to identify root variables—those not caused by any other variable in the system. Starting from these roots, the BFS iteratively expands each node by prompting the LLM to select all variables directly caused by it, visiting each node exactly once, resulting in $O(n)$ calls of $O(n)$ per-call complexity. Since all edges are determined solely by LLM responses without any statistical grounding, errors in root node identification or downstream variable selection propagate through the entire traversal without any correction mechanism.

Causal-LLM (Roy et al., 2025). Causal-LLM offers two operating modes depending on data availability. When variable metadata is accessible, it uses a prompt-based approach to generate the full causal graph in a single LLM call via in-context learning, with a total decision volume of $O(n^2)$, achieved through $O(1)$ calls of $O(n^2)$ per-call complexity. When metadata is absent, it falls back to a purely data-driven approach using LLM representations. The two modes operate independently. Asking the LLM to jointly reason over all variable pairs in a single prompt places a heavy burden on the model's context window, leading to degraded decision quality on larger graphs with no statistical safeguard against erroneous outputs.

Causal Order (Vashishtha et al., 2025). Causal Order infers causal ordering priors by querying the LLM with variable triplets, with a total decision volume of $O(kn^2)$ achieved through $O(kn^2)$ calls of $O(1)$ per-call complexity, where k is the number of triplets per variable pair. The inferred ordering is then used to guide a downstream statistical algorithm. The high query complexity limits its scalability to small graphs only (maximum n=24 in the original paper). Since the ordering prior influences all downstream edge decisions, both LLM error exposure and algorithmic bias exposure cover the full graph: errors in the LLM-inferred ordering propagate into the downstream algorithm, whose own biases further affect all edges without any calibration to verify whether the LLM's ordering judgments are reliable.

SCP (Takayama et al., 2025). SCP first runs a statistical algorithm (e.g., PC) to obtain an initial graph, then provides the result as context to the LLM via "Statistical Causal Prompting," querying it on each of the $O(n^2)$ variable pairs to produce a probability matrix, which is further processed by a downstream statistical algorithm. The total decision volume is $O(n^2)$, achieved through $O(n^2)$ calls of $O(1)$ per-call complexity. Since the LLM is queried over all variable pairs, LLM error exposure

covers the entire graph. Furthermore, both the upstream and downstream statistical algorithms are single algorithms, making the final graph susceptible to the individual algorithm limitations and failure modes.

ET-MCMC (Ban et al., 2025). ET-MCMC first queries the LLM on all $O(n^2)$ variable pairs to assess confidence, then uses only the high-confidence subset as soft ancestral constraints for a downstream score-based method. The total decision volume is $O(n^2)$, achieved through $O(n^2)$ calls of $O(1)$ per-call complexity. Although LLM queries are restricted to high-confidence pairs with greater reliability, empirical studies have shown that LLMs tend to be overconfident (Xiong et al., 2024). LLM errors can still affect any edge in the final graph unchecked. The final graph is further subject to the individual limitations and biases of the single downstream algorithm.

CMA (Abdulaal et al., 2024). CMA combines LLM metadata-based reasoning with Deep Structural Causal Models (DSCMs) in an iterative agentic loop. The LLM proposes graph modifications in both global and local phases, with the DSCM providing data-driven feedback at each step. The total decision volume is $O(Tn^2)$ across T iterations, as the LLM operates over the full graph at each round. Since all edges are subject to LLM-proposed modifications at every iteration, LLM error exposure covers the entire graph. Moreover, the method is subject to the limitations and assumptions of the single DSCM used for validation in each iteration.

CauScientist (Peng et al., 2026b). CauScientist operates in three stages: hybrid initialization selects the better of a data-driven baseline graph and an LLM-proposed graph based on BIC score; iterative refinement where the LLM proposes atomic edge modifications given the current full graph as context, evaluated by a BIC-based verifier; and an error memory that prevents the LLM from repeating previously rejected proposals. The total decision volume is $O(Tn^2)$ across T iterations, achieved through T calls each of $O(n^2)$ per-call complexity as the full graph is provided as context at every step. LLM error exposure is partially mitigated by the BIC verifier, but BIC-based validation provides only a local structural criterion that may not be reliable under finite samples.

LLM-MEC (Long et al., 2023). LLM-MEC restricts LLM queries to the undirected edges within a pre-computed MEC, iteratively orienting edges with minimum risk of eliminating the true graph from the MEC. The total decision volume is $|\mathcal{U}|$, the number of undirected edges in the MEC, achieved through $|\mathcal{U}|$ calls of $O(1)$ per-call complexity. Since only undirected edges are queried, LLM error exposure is limited to these edges. However, the method relies on a single upstream algorithm (e.g., PC) to compute the initial MEC, making the entire graph vulnerable to the individual limitations of that algorithm. Beyond this, the method resolves edges sequentially: each orientation decision is immediately propagated via Meek rules to infer additional edge directions before the next edge is queried. This means a single LLM error can trigger a cascade of forced orientations, amplifying the impact of individual mistakes across the remaining undirected edges. Moreover, the LLM error rate used in the risk computation is derived from human assumption or LLM token probabilities rather than empirical calibration, which can deviate substantially from the true error rate. Finally, the iteration stops when the cumulative risk of excluding the true graph exceeds a threshold, so the output is not guaranteed to be a fully oriented DAG—some edges may remain undirected in the final graph.

B METHOD DETAILS

Algorithm 5 summarizes the complete `CauTion` pipeline, which proceeds in three stages: consensus-based pre-filtering, accuracy calibration, and calibrated voting with cycle repair.

Require: Dataset $\mathcal{X}$, variable names $\mathcal{V}$, algorithm set $\mathcal{A}$, LLM backend, calibration budget n_{calib}, trust exponent β
Ensure: DAG G over $\mathcal{V}$

// Stage 1: Consensus Pre-Filtering
Initialize $G \leftarrow (\mathcal{V}, \emptyset)$
for each $A \in \mathcal{A}$ **do**
    Run A on $\mathcal{X} \rightarrow$ vote matrix $\{v_A(e)\}$ for all pairs e
end for
Partition all pairs into $\mathcal{C}$ (consensus) and $\mathcal{D}$ (disputed) ▷ §3.2
Add unanimously supported directed edges from $\mathcal{C}$ to G

// Stage 2: Accuracy Calibration
Estimate target distribution $(p_{\text{exist}},\ p_{\text{none}})$ from $\mathcal{D}$
Infer domain context and variable semantics via LLM ▷ §B.1
Sample up to n_{calib} pairs from $\mathcal{C}$ proportional to $(p_{\text{exist}}, p_{\text{none}})$
Query LLM on samples $\rightarrow$ $acc_{\text{LLM,exist}}$, $acc_{\text{LLM,dir}}$ ▷ §3.3.1
Compute LOO accuracies acc^{A}_{exist} and acc^{A}_{dir} for each $A \in \mathcal{A}$ on $\mathcal{C} \cup \mathcal{D}$ ▷ §3.3.3
Aggregate $\rightarrow$ $acc_{\text{algo,exist}}$, $acc_{\text{algo,dir}}$
Compute trust weights α_{exist}, α_{dir} from accuracies §3.3.4

// Stage 3: Calibrated Voting on Disputed Pairs
Compute algorithm vote scores and margins for all pairs ▷ §3.3.5
for each $e \in \mathcal{D}$ **do**
    $q_{\text{exist}} \leftarrow v_{\text{exist}}(e)$
    if $m_{\text{exist}}(e) \leq \theta_{\text{exist}}$ **then**
        Query LLM $\rightarrow \hat{e} \in \{\texttt{exist}, \texttt{none}\}$
        $q_{\text{exist}} \leftarrow (1 - \alpha_{\text{exist}}) \cdot v_{\text{exist}}(e) + \alpha_{\text{exist}} \cdot \mathbf{1}[\hat{e} = \texttt{exist}]$
    end if
    $q_{\text{none}} \leftarrow 1 - q_{\text{exist}}$
    if $q_{\text{none}} \geq q_{\text{exist}}$ **then**
        continue ▷ No edge added
    end if
    $q_{\text{fwd}} \leftarrow v_{\text{fwd}}(e)$
    if $m_{\text{dir}}(e) < \theta_{\text{dir}}$ **then**
        Query LLM with both variable orders; map answers to the same order
        if both answers agree on $d_{\text{LLM}} \in \{\texttt{fwd}, \texttt{rev}\}$ **then**
            $q_{\text{fwd}} \leftarrow \alpha_{\text{dir}} \cdot \mathbf{1}[d_{\text{LLM}} = \texttt{fwd}] + (1 - \alpha_{\text{dir}}) \cdot v_{\text{fwd}}(e)$
        end if
    end if
    $q_{\text{rev}} \leftarrow 1 - q_{\text{fwd}}$; $\hat{d} \leftarrow \arg\max_{d \in \{\texttt{fwd},\texttt{rev}\}} q_d$
    Add directed edge to G according to $\hat{d}$
end for

// Stage 4: Cycle Repair
$\mathcal{T} \leftarrow \emptyset$
while G contains a directed cycle $\mathcal{C}_{\text{cyc}}$ **do**
    $\mathcal{S} \leftarrow$ eligible actions on $\mathcal{C}_{\text{cyc}}$ excluding $\mathcal{T}$ ▷ §3.4
    if $\mathcal{S} \neq \emptyset$ **then**
        Query LLM to select one action $a \in \mathcal{S}$
        Apply a to G; $\mathcal{T} \leftarrow \mathcal{T} \cup \{a\}$
    else
        Remove edge with lowest v_{exist} from $\mathcal{C}_{\text{cyc}}$ ▷ Fallback
    end if
end while
return G

Figure 5: `CauTion`: Calibrated LLM-Augmented Causal Discovery

B.1 LLM PROMPTING

All LLM queries follow a shared prompting structure detailed below. Placeholders in {braces} denote runtime-substituted values; their contents are described in Table 3. *Italic* annotations in the prompts are explanatory comments and are not part of the actual prompt text.

Table 3: Placeholder variables used across all prompts.

Placeholder	Description
{domain_name}	Domain label (e.g. "Alarm")
{domain_context}	Domain description inferred by the LLM
{variables}	Variable list with categorical values and meanings
{N}	Number of variables
{consensus_directed_edges}	Directed edges agreed upon by all algorithms
{pairs}	Candidate variable pairs to be queried
{cycle}	Detected directed cycle path (e.g. $A \to B \to C \to A$)
{candidates}	Uncertain edges on the cycle with their algorithmic margins

We use four categories of prompts. **Domain inference (P1)** is called once at the start to infer a domain label and description from variable names, which is injected into the system prompt (P2) of all subsequent calls. **Variable interpretation (P3)** is called once before before querying disputed edges to assign a natural-language meaning to each variable; during calibration, consensus edges are withheld to prevent label leakage. **Edge-level queries (P4-5)** cover the the existence and direction queries issued during calibration and disputed-edge resolution; direction queries are issued twice per pair in opposite framings to eliminate position bias, and only consistent answers are retained. **Cycle repair** (P6) is called when a directed cycle is detected in the assembled graph, presenting uncertain cycle edges to the LLM for correction via FLIP or REMOVE actions.

LLM calls use a temperature of 0.0 to ensure deterministic outputs. Edge existence and direction queries are batched into groups of at most 10 variable pairs per call to keep prompts within a manageable length.

P1. DOMAIN INFERENCE PROMPT

System

```
You are analyzing a system to support causal reasoning about its
variables.
```

User

```
Here are variables from an unknown system:
{variables} (one line per variable: " - NAME: val1, val2, ...")
What is this system about?
```

P2. SHARED SYSTEM PROMPT

System

```
You are an expert in causal inference and the {domain_name} domain.
Domain context: {domain_context}
Your task is to assess the most plausible causal relationship between
pairs of variables using your domain knowledge alone.
```

P3. VARIABLE INTERPRETATION PROMPT

User

```
Before analyzing causal directions, build an initial understanding of
what each variable in this dataset likely represents.

Variables ({N} total):
{variables} (with optional value list)

=== Relationships agreed on by ALL algorithms ===
{consensus_directed_edges} (e.g. " VAR1 -> VAR2")

For each variable, describe what it likely measures or represents.
Rate your confidence from 0 to 10 (10 = completely certain, 0 = no
idea).

Output valid JSON only:
{
"variable_interpretations": {
"VAR_NAME": {
"meaning": "...",
"confidence": 8
},
...
}
}
```

P4. EDGE EXISTENCE QUERY PROMPT

User

```
For each variable pair below, determine whether a direct causal
relationship exists between them, using your domain knowledge only.

Definitions:
"EDGE" = a direct causal link exists between the two variables (in
either direction)
"NONE" = no direct causal link exists between them

Variable meanings:
{variables} (e.g. " VAR1: meaning [values: val1, val2]")

Output a single JSON object. Each key is a pair number; each value
has "reasoning" (brief explanation) and "existence" ("EDGE" or
"NONE").
Example: {"1": {"reasoning": "...", "existence": "EDGE"},
          "2": {"reasoning": "...", "existence": "NONE"}}

--- Variable pairs ---
{pairs} (e.g. " "1": VAR1 <-> VAR2")
```

P5. EDGE DIRECTION QUERY PROMPT

User

```
A direct causal edge is KNOWN to exist for each pair below.  Your task
is ONLY to determine its direction.  You MUST output exactly "FWD" or
"REV" for every pair --- no other values.

Definitions:
"FWD" = the first-listed variable directly causes the second
"REV" = the second-listed variable directly causes the first

Variable meanings:
{variables} (e.g. " VAR1: meaning [values: val1, val2]")

Output a single JSON object.  Each key is a pair number; each value
has "reasoning" (brief explanation) and "direction" ("FWD" or "REV").
Example:  {"1":  {"reasoning":  "...", "direction":  "FWD"},
           "2":  {"reasoning":  "...", "direction":  "REV"}}

--- Pairs to orient ---
{pairs} (e.g. " "1": VAR1 <-> VAR2 (FWD = VAR1 -> VAR2 — REV = VAR2 -> VAR1)")
```

P6. CYCLE REPAIR PROMPT

User

```
A directed cycle was detected in the causal graph:
  {cycle} (e.g. "VAR1 -> VAR2 -> VAR3 -> VAR1")

Causal graphs must be acyclic (DAG). The cycle above needs to be
corrected.

Variable meanings:
{variables} (meanings for variables involved in the cycle)

The following candidate edges have uncertain existence or direction
(based on algorithm confidence).  Review them and specify corrections
for those you think are incorrect:
{candidates}
(e.g. " "1": VAR1 -> VAR2 (existence uncertain; margin=0.12) [actions: FLIP/REMOVE]")

Actions:
"FLIP"   -- reverse the direction of the edge
"REMOVE" -- delete this causal relationship entirely
(only for candidates marked [actions:  FLIP/REMOVE])

Output ONLY a JSON object with a "reasoning" field and a "corrections"
field.  "corrections" maps candidate number to action ("FLIP" or
"REMOVE"); include only the edges you want to correct.
Example:  {"reasoning":  "...", "corrections":  {"2":  "FLIP"}}
```

C EXPERIMENT DETAILS

C.1 DATASET DETAILS

The six main benchmarks are standard discrete Bayesian networks drawn from the *bnlearn* repository (Scutari, 2010). Each dataset is generated by forward-sampling from the ground-truth network specified by its BIF file, using the `pgmpy` library (Ankan & Textor, 2024) (`BayesianModelSampling.forward_sample`). We additionally evaluate on a newly constructed synthetic network, *Airport*, in Appendix C.8.

C.2 METRICS

We evaluate the performance of all causal discovery methods using the following structural metrics.

Structural Hamming Distance (SHD). SHD counts the minimum number of edge insertions, deletions, and reversals needed to transform the predicted graph into the ground truth.

Precision, Recall, and F1 Score. Delegated to `sklearn.metrics.f1_score` and related functions (Pedregosa et al., 2011), applied to the flattened adjacency matrices. Precision and Recall are computed at the edge level over the full $n \times n$ binary matrix; reversed edges thus count simultaneously as a false positive and a false negative.

Structural Interventional Distance (SID). We use the official R implementation of SID (Peters & Bühlmann, 2015), which returns an exact value for DAG estimates and lower and upper bounds for CPDAG estimates. We report exact SID values as scalars and distinct lower and upper bounds as intervals $[L, U]$. Coincident bounds are also displayed as a single value. SID measures how useful the predicted graph actually is for causal inference. Concretely, the algorithm iterates over all ordered variable pairs (i, j) and checks whether the predicted graph yields the same causal effect of i on j as the ground-truth graph. Each mismatch is counted as one inference error. SID is the total number of such mismatches across all variable pairs.

C.3 Baseline Implementation Detail

C.3.1 Data-driven Baselines

PC (Spirtes et al., 2000), FCI (Spirtes et al., 1995), GES (Chickering, 2002), and GRaSP (Lam et al., 2022) are implemented via `causal-learn`[2] (Zheng et al., 2024), using the chi-squared independence test and BDeu score for discrete data. CAMML (Wallace et al., 1996) is implemented via the official BI-CaMML Java implementation[3].

NOTEARS-MLP (Zheng et al., 2020) is implemented via the `gCastle` package[4] (Zhang et al., 2021), with z-score standardization applied to the input data. DAGMA (Bello et al., 2022) is implemented via the official `dagma` Python package[5].

C.3.2 LLM-Augmented Baselines

CORR-LLM. A zero-shot single-call baseline designed and implemented by us. The Pearson correlation matrix of the observations is passed to the prompt alongside variable names. The LLM is asked to output a an `edges` list. Post-processing: up to 3 retries if the returned graph contains cycles. If cycles persist after all candidate actions are exhausted, edges on the remaining cycle are randomly removed until the graph becomes acyclic.

LLM-BFS (Jiralerspong et al., 2024). We use the official `causal-llm-bfs` repository.

SCP (Takayama et al., 2025). We use the official repository for SCP.

ET-MCMC (Ban et al., 2025). We use the official repository for ET-MCMC.

LLM-MEC (Long et al., 2023). We use the official repository for LLM-MEC. Since not all LLMs support token-level log-probabilities, we handle two cases: for models that support logprobs, we follow the original implementation and extract $\log P(\mathrm{A})$ and $\log P(\mathrm{B})$ directly from the top token candidates; for models that do not support logprobs, we fall back to parsing the text response and assigning a fixed probability of $(0.85, 0.15)$ to the chosen and unchosen options, respectively.

[2] `https://causal-learn.readthedocs.io/en/latest/`

[3] `https://bayesian-intelligence.com/software/BI-CaMML-Quickstart-Guide-1.4/`

[4] `https://gcastle-test.readthedocs.io/en/latest/getting_started.html`

[5] `https://dagma.readthedocs.io/en/latest/`

C.4 Performance Detail

C.4.1 Full Results with Standard Deviations

We report mean ± sample standard deviation across 5 independently sampled dataset versions. F1, precision, and recall are percentages. For DAG outputs, SID is reported as mean ± sample standard deviation. For CPDAG outputs, SID is characterized by lower and upper bounds corresponding to the minimum and maximum SID values among DAGs in the represented Markov equivalence class. For outputs with distinct SID bounds, we report $[\bar{L} \pm s_L, \bar{U} \pm s_U]$, with the mean and sample standard deviation computed separately for each bound across the five dataset versions.

Table 4: Full results on **Cancer** ($n = 5$, $|E| = 4$).

Method	SHD	F1 (%)	Prec (%)	Rec (%)	SID
Constraint-based					
PC	2.0±0.0	50.0±0.0	50.0±0.0	50.0±0.0	10.0±0.0
FCI	4.0±0.0	66.7±0.0	50.0±0.0	**100.0±0.0**	10.0±0.0
Score-based					
GES	**0.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	0.0±0.0
GRaSP	2.4±2.2	80.0±18.3	70.0±27.4	**100.0±0.0**	[3.0±2.7, 9.0±8.2]
CAMML	2.2±0.4	50.3±4.5	51.3±9.6	50.0±0.0	9.0±1.7
Continuous optimization					
NOTEARS-MLP	3.2±0.4	26.7±14.9	40.0±22.4	20.0±11.2	10.2±0.4
DAGMA	4.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	10.0±0.0
LLM-augmented					
CORR-LLM	**0.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	0.0±0.0
LLM-BFS	**0.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	0.0±0.0
SCP	1.2±0.4	70.0±11.2	70.0±11.2	70.0±11.2	6.0±2.2
ET-MCMC	0.4±0.9	91.4±19.2	93.3±14.9	90.0±22.4	1.2±2.7
LLM-MEC	2.0±0.0	50.0±0.0	50.0±0.0	50.0±0.0	10.0±0.0
Ours	**0.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	0.0±0.0

Table 5: Full results on **Insurance** ($n = 27$, $|E| = 52$).

Method	SHD	F1 (%)	Prec (%)	Rec (%)	SID
Constraint-based					
PC	24.8±1.5	63.7±3.1	71.7±4.1	57.3±2.5	[398.6±15.1, 399.6±15.1]
FCI	31.2±2.8	70.0±1.7	69.3±3.5	70.8±1.6	[350.2±21.3, 426.6±20.8]
Score-based					
GES	22.2±5.4	72.9±6.3	76.4±7.1	69.6±5.7	[256.6±28.7, 333.2±30.9]
GRaSP	40.6±10.0	63.3±6.4	59.4±10.1	68.5±3.5	[285.6±33.9, 474.4±86.6]
CAMML	27.4±6.9	60.5±10.6	65.6±11.8	56.2±9.7	340.4±26.5
Continuous optimization					
NOTEARS-MLP	50.2±5.2	33.8±6.6	40.8±8.1	28.8±5.6	535.2±54.3
DAGMA	51.0±3.8	27.0±4.6	36.5±8.2	21.5±3.2	587.4±31.5
LLM-augmented					
CORR-LLM	22.8±3.6	74.7±4.4	85.9±2.7	66.2±5.5	374.8±57.7
LLM-BFS	25.2±3.1	73.3±3.5	80.8±4.2	67.3±4.9	326.2±62.4
SCP	20.4±1.7	72.3±2.4	82.0±3.7	64.6±1.7	[365.8±25.2, 397.8±20.1]
ET-MCMC	21.0±5.3	73.6±6.3	76.4±8.1	71.2±4.9	285.0±24.7
LLM-MEC	24.2±1.1	64.1±2.8	72.7±3.4	57.3±2.5	[398.6±15.1, 399.6±15.1]
Ours	**13.0±1.7**	**85.7±1.6**	**96.7±3.5**	**76.9±1.4**	198.6±12.4

Table 6: Full results on **Water** ($n = 32$, $|E| = 66$).

Method	SHD	F1 (%)	Prec (%)	Rec (%)	SID
Constraint-based					
PC	54.6±3.0	46.1±3.6	61.5±5.6	37.0±3.0	[411.4±38.9, 531.8±43.1]
FCI	59.2±3.3	46.8±2.2	57.2±4.7	39.7±1.3	[386.8±12.7, 523.0±20.4]
Score-based					
GES	47.0±2.5	50.2±2.8	69.1±3.6	39.4±2.4	[423.0±15.1, 481.0±15.1]
GRaSP	51.6±1.3	48.3±2.0	63.3±2.3	39.1±2.2	[415.4±28.6, 523.2±31.8]
CAMML	55.0±4.7	31.8±7.2	45.0±10.3	24.5±5.6	566.8±32.8
Continuous optimization					
NOTEARS-MLP	63.2±1.1	9.4±2.8	20.7±5.1	6.1±1.9	539.4±21.2
DAGMA	58.2±0.8	20.9±2.1	52.4±4.4	13.0±1.4	521.6±8.1
LLM-augmented					
CORR-LLM	42.0±0.0	53.5±0.5	**99.2±1.7**	36.7±0.7	403.8±2.7
LLM-BFS	54.0±17.0	51.8±14.2	64.2±19.5	43.6±11.9	374.4±24.0
SCP	52.6±4.1	47.4±3.2	65.2±7.0	37.3±2.3	[405.4±26.1, 512.2±47.0]
ET-MCMC	52.0±4.2	44.5±4.5	55.9±5.0	37.0±4.2	525.0±39.0
LLM-MEC	54.0±3.9	44.7±7.2	62.9±6.8	34.9±6.8	[416.2±37.1, 519.6±38.6]
Ours	**39.0±4.3**	**59.2±6.2**	90.6±9.1	**43.9±4.8**	377.0±55.7

Table 7: Full results on **ALARM** ($n = 37$, $|E| = 46$).

Method	SHD	F1 (%)	Prec (%)	Rec (%)	SID
Constraint-based					
PC	13.6±5.4	85.9±5.1	82.4±8.0	90.0±1.9	[44.8±12.6, 133.2±64.3]
FCI	32.8±6.6	71.8±4.4	59.9±5.6	90.0±1.9	[39.8±14.3, 170.4±82.4]
Score-based					
GES	7.8±3.0	91.2±3.9	88.6±4.9	93.9±2.8	[28.8±15.8, 77.6±21.4]
GRaSP	12.8±6.3	87.1±5.7	81.2±9.2	**94.3±1.9**	[31.8±21.8, 125.0±69.3]
CAMML	8.8±3.4	85.8±6.0	85.5±7.0	86.1±5.0	95.8±51.6
Continuous optimization					
NOTEARS-MLP	52.2±4.2	30.6±3.8	27.3±4.1	34.8±3.4	427.0±56.6
DAGMA	33.4±6.9	58.0±7.9	56.6±8.4	59.6±7.8	251.6±26.4
LLM-augmented					
CORR-LLM	16.4±2.3	80.2±4.3	80.9±1.8	79.6±7.0	110.4±36.2
LLM-BFS	44.2±3.8	49.4±2.7	48.2±4.8	50.9±1.9	268.0±18.4
SCP	10.4±2.3	87.2±2.8	87.2±4.2	87.4±2.8	[77.0±20.4, 115.0±28.6]
ET-MCMC	10.2±4.6	86.5±6.1	83.8±8.3	89.6±3.6	63.4±45.0
LLM-MEC	13.0±4.8	84.9±6.8	82.7±8.3	87.4±5.4	[71.0±48.4, 133.2±64.3]
Ours	**4.6±1.1**	**93.8±1.4**	**95.5±2.2**	92.2±1.2	37.8±17.0

Table 8: Full results on **Barley** ($n = 48$, $|E| = 84$).

Method	SHD	F1 (%)	Prec (%)	Rec (%)	SID
Constraint-based					
PC	55.8±3.6	49.4±3.3	57.3±4.1	43.6±3.6	[1012.2±115.6, 1087.6±68.2]
FCI	67.4±6.5	60.2±3.0	58.1±4.1	62.6±2.0	[740.8±55.0, 1110.0±47.3]
Score-based					
GES	54.0±0.7	64.0±0.4	66.8±0.7	61.4±0.7	[665.0±0.0, 1015.8±20.0]
GRaSP	85.6±5.8	45.4±4.7	45.2±4.1	45.7±5.7	[856.8±106.7, 1213.0±92.1]
CAMML	77.2±10.2	34.7±9.7	38.7±10.6	31.4±8.9	1144.4±73.8
Continuous optimization					
NOTEARS-MLP	84.2±4.0	19.3±4.2	37.3±11.2	13.1±2.7	1190.6±67.9
DAGMA	82.2±1.3	5.6±2.9	30.1±13.7	3.1±1.6	1233.4±18.6
LLM-augmented					
CORR-LLM	78.2±5.4	46.1±4.1	54.0±4.2	40.2±4.2	881.2±62.5
LLM-BFS	120.8±10.4	28.1±5.1	28.2±1.5	29.0±8.9	1022.6±168.0
SCP	56.8±2.3	47.0±3.6	54.9±3.7	41.2±3.6	[1069.8±114.6, 1132.6±72.6]
ET-MCMC	90.2±7.0	29.0±2.7	31.0±4.0	27.4±2.1	1217.8±69.3
LLM-MEC	61.2±7.4	44.3±8.3	60.7±5.8	35.7±9.9	[1019.4±104.8, 1094.2±67.7]
Ours	**42.8±5.0**	**68.4±3.7**	**74.8±7.0**	**63.3±4.3**	743.4±101.6

Table 9: Full results on **Win95pts** ($n = 76$, $|E| = 112$).

Method	SHD	F1 (%)	Prec (%)	Rec (%)	SID
Constraint-based					
PC	72.0±5.2	56.7±3.0	64.8±4.3	50.4±2.2	[819.8±52.9, 976.6±76.8]
FCI	100.6±3.0	58.5±1.0	54.4±1.2	63.2±1.0	[477.2±11.4, 967.2±65.4]
Score-based					
GES	39.8±2.6	80.8±1.5	78.1±1.3	**83.6±2.1**	[179.4±17.7, 230.2±12.7]
GRaSP	57.8±16.1	73.4±6.6	70.4±8.0	76.8±5.1	[320.0±134.9, 418.2±166.1]
CAMML	65.6±11.4	67.1±5.3	63.9±6.5	70.7±4.8	358.2±46.2
Continuous optimization					
NOTEARS-MLP	141.6±12.1	23.3±4.6	21.6±4.3	25.4±5.0	1009.6±78.7
DAGMA	103.2±7.2	43.9±4.2	44.5±4.0	43.4±4.6	674.4±68.8
LLM-augmented					
CORR-LLM	77.6±7.5	58.0±3.7	70.4±4.3	49.3±3.6	587.8±51.0
LLM-BFS	146.6±8.5	23.2±3.3	27.3±4.2	20.2±2.8	1017.0±54.9
SCP	69.0±5.5	58.7±3.4	67.7±4.8	51.8±2.5	[773.4±58.4, 922.0±85.8]
ET-MCMC	94.6±10.5	60.7±4.5	52.0±4.4	72.9±4.7	384.8±76.0
LLM-MEC	71.8±5.5	56.6±2.9	64.9±4.4	50.2±2.0	[823.8±46.6, 975.0±78.6]
Ours	**28.4±5.5**	**84.2±2.6**	**92.7±2.4**	77.1±3.7	342.6±39.2

C.4.2 Run Time and Token Usage

All experiments are conducted on a server with an AMD EPYC 9654 96-Core Processor and 14GB RAM. All methods are run single-threaded on CPU. Reported runtimes are approximate wall-clock times measured on Win95pts. Token usage is reported for LLM-augmented methods only, measured with Claude-Sonnet-4.6 as the underlying LLM.

Table 10: Runtime and token usage comparison on Win95pts (n=76). Time is measured in wall-clock seconds. Token usage is reported for LLM-augmented methods only, measured using Claude-Sonnet-4.6. "—" indicates no LLM queries are made.

Method	**Time (s)**	**Token Usage**
PC	4	—
FCI	4	—
GES	164	—
GRaSP	33,454	—
CAMML	4,077	—
NOTEARS-MLP	2,232	—
DAGMA	16,655	—
LLM-BFS	637	2,971,861
SCP	25,116	3,029,612
ET-MCMC	224,485	12,895
LLM-MEC	35,857	1,659
LLM	41	26,964
Ours	4387	26,267

C.5 Details on Ablation Study

We detail the implementation of each variant strategy in our ablation study.

Consensus Only. Disputed edges are unconditionally discarded. The predicted graph contains exactly the edges on which all algorithms agree.

LLM Only. The LLM is queried independently over every variable with no consensus filtering and no algorithmic guidance. Domain context and variable interpretations are inferred in the same way as *Ours*. For each variable X_i ($i = 1, \ldots, n$), a single prompt asks *"which variables does X_i directly cause?"* and the LLM returns a list of effect variables. The predicted graph is the union of all returned out-edges across the n calls.

Majority Vote. All edges are resolved by majority vote: the option (fwd, rev, or none) receiving the most votes from the three algorithms is accepted, with ties broken in favor of none.

Cons + LLM. The consensus edges are kept as-is. The LLM is queried on disputed edges using the same bidirectional protocol as *Ours* (Section 3.3.1), producing identical LLM outputs. The LLM's answer is accepted directly without trust-weighted combination with algorithm votes.

C.6 Additional Results on Model Selection Sensitivity

We provide more detailed cross-model robustness results in Figures 6–7 and radar plots in Figures 8–9. Figures 6 and 7 show the SHD and F1 ranges across six LLMs for each method and dataset, where each dataset uses an independently scaled x-axis to make cross-model variance more distinguishable. Figures 8 and 9 show radar plots summarizing the performance of all LLMs across datasets and methods on SHD and F1 respectively.

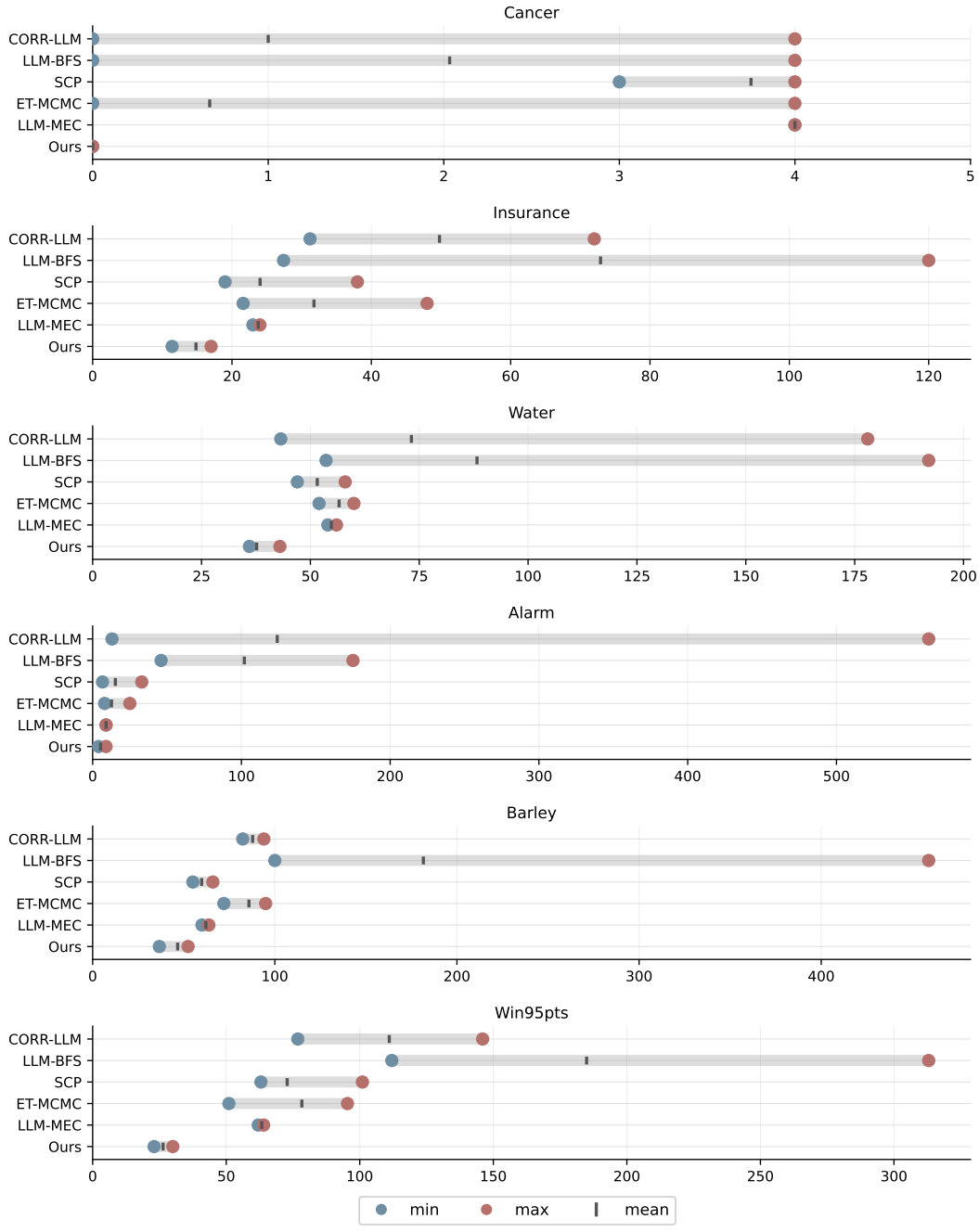


Figure 6: Cross-model robustness of LLM-augmented methods across datasets (SHD). Each horizontal bar spans the minimum to maximum SHD achieved across six LLMs, where shorter bars indicate greater robustness to backend selection.

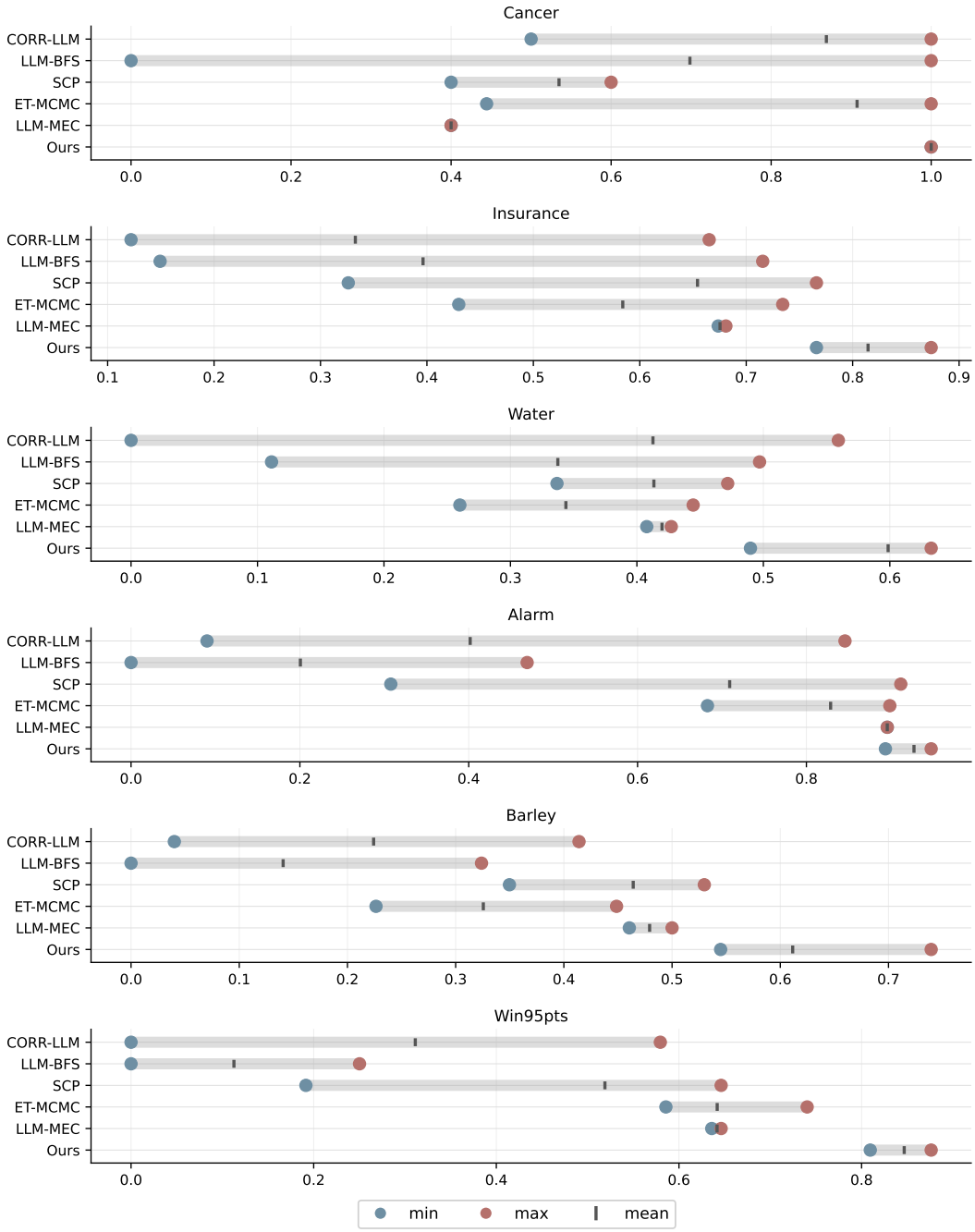


Figure 7: Cross-model robustness of LLM-augmented methods across datasets (F1). Each horizontal bar spans the minimum to maximum F1 achieved across six LLMs.

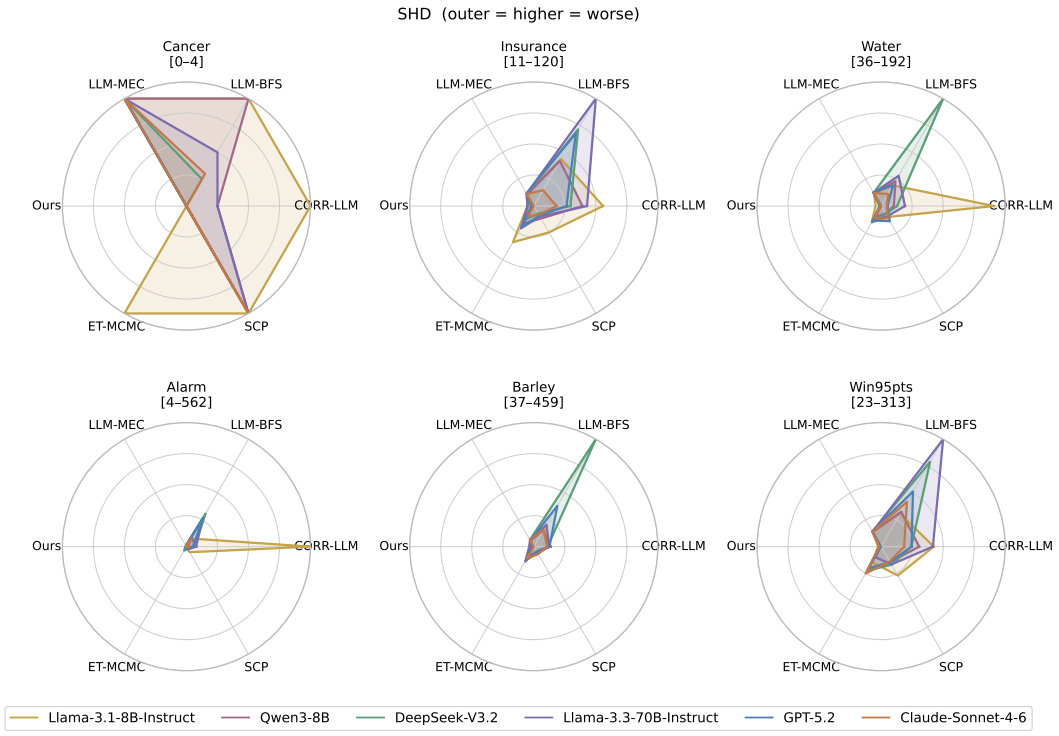


Figure 8: Cross-model radar of LLM-augmented methods across datasets (SHD). For each dataset sub-figure, the axis is scaled to the observed range across all models and methods for that dataset.

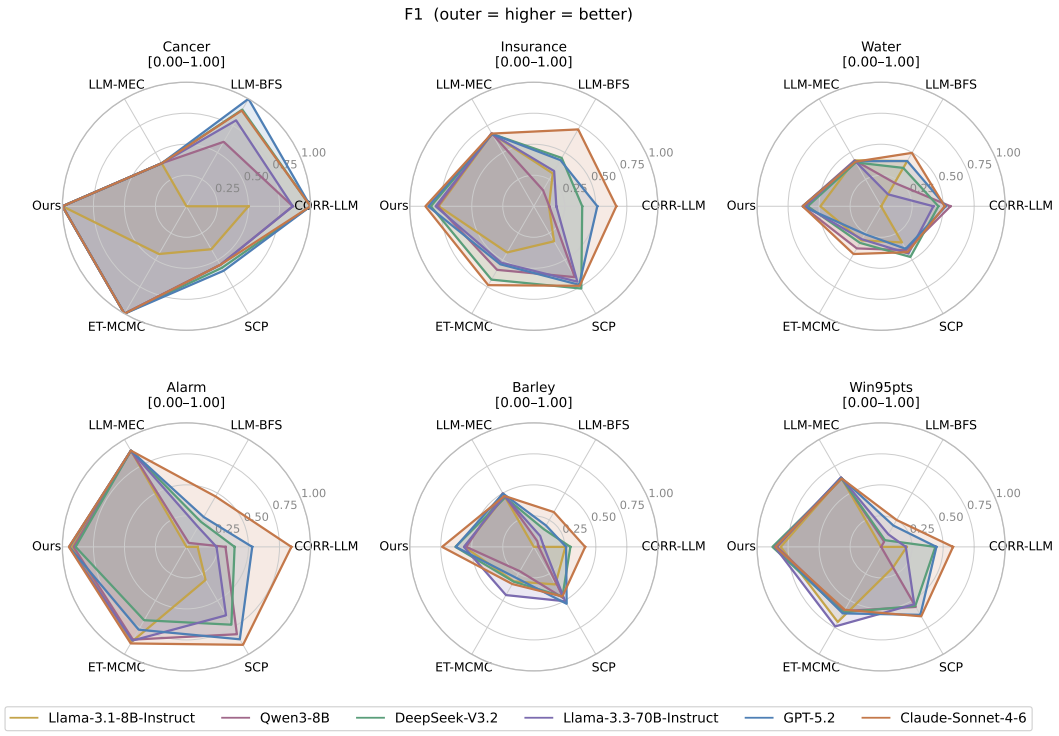


Figure 9: Cross-model radar of LLM-augmented methods across datasets (F1).

C.7 Consensus Filtering Rate and Accuracy

Table 11 reports the consensus filtering statistics across all six datasets. On larger graphs, consensus filtering resolves over 90% of variable pairs with near-perfect accuracy, confirming the suitability of consensus edges as proxy ground truth for downstream calibration. The lower filtering rate on Cancer (n=5) is expected given its small number of variable pairs.

Table 11: Consensus filtering statistics across datasets. Filter rate denotes the fraction of variable pairs resolved by consensus. Cons. Acc. denotes the fraction of consensus-resolved edges correctly classified.

Dataset	Nodes	Total Pairs	Disputed	Filtered	Filter Rate	Cons. Acc.
Cancer	5	10	7	3	30.0%	100.0%
Insurance	27	351	26	325	92.6%	97.8%
Alarm	37	666	23	643	96.5%	99.8%
Water	32	496	31	465	93.8%	92.7%
Barley	48	1,128	70	1,058	93.8%	98.2%
Win95pts	76	2,850	96	2,754	96.6%	99.6%

C.8 Evaluation on a Newly Constructed Graph

Beyond published benchmarks, we independently constructed *Airport*, a synthetic Bayesian network with 15 variables and 23 directed edges. Figure 10 shows its ground-truth causal structure. The network combines three airport-related processes, with ambient temperature and holiday peaks as exogenous drivers. The weather branch links adverse weather, de-icing demand, and de-icing queue time. The air-traffic-control branch links flow restrictions, inbound flight delay, and flight cancellation. The terminal and ramp branch links scheduled flight volume to passenger load, which affects security queues, baggage handling, and boarding. Security queues affect boarding time; baggage handling and boarding both affect aircraft turnaround time. Turnaround time and de-icing queues, in turn, affect departure delay.

Each variable has three states (low, average, and high). The construction procedure fits discrete conditional probability tables to a discretized linear-Gaussian structural model. Table 12 reports `CauTion` results on this sampled dataset. `CauTion` achieves the best SHD and F1 and the tied-best SID on this newly constructed graph, indicating its robustness on non-public benchmarks.

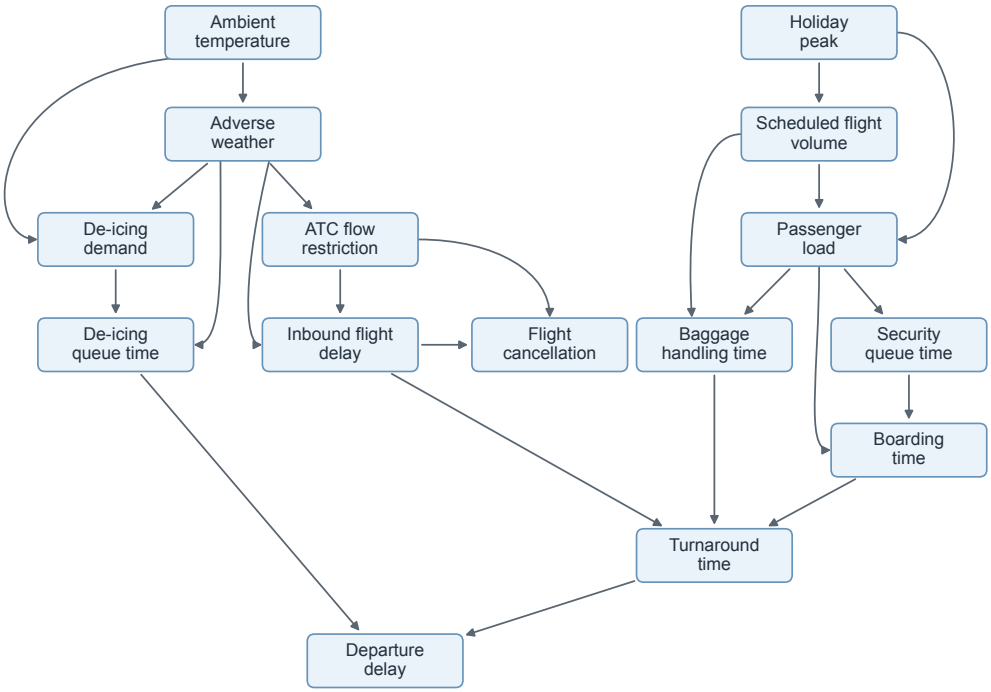


Figure 10: Ground-truth causal graph of Airport. Arrows indicate direct causal relationships. ATC denotes air traffic control.

Table 12: Results on **Airport** ($n = 15$, $|E| = 23$). F1 is reported in percent. SID is reported as an exact value for DAG outputs or as $[L, U]$ for distinct lower and upper bounds.

Method	**SHD** ↓	**F1 (%)** ↑	**SID** ↓
PC	21	67.7	$[8, 89]$
FCI	22	66.7	$[8, 102]$
GES	13	75.0	$[17, 70]$
GRaSP	20	59.6	$[50, 93]$
CAMML	5	78.3	38
NOTEARS-MLP	8	68.1	51
DAGMA	6	77.3	54
CORR-LLM	5	89.4	8
LLM-BFS	11	78.4	15
SCP	7	86.3	$[8, 44]$
ET-MCMC	5	78.3	37
LLM-MEC	16	73.3	$[8, 75]$
Ours	**2**	**95.7**	8