Bayesian Network Structure Discovery Using Large Language Models

Anonymous authors
Paper under double-blind review

Abstract

Understanding probabilistic relationships among variables is crucial for analyzing complex systems. Traditional structure learning methods often require extensive observational data and incur high computational costs. Recent studies have explored using large language models (LLMs) for structure learning, but most treat LLMs as auxiliary tools for pre-processing or post-processing, leaving the core learning process data-driven. In this work, we propose a unified framework for Bayesian network structure discovery that places LLMs at the center, supporting both data-free and data-aware settings. In the data-free case, we introduce **PromptBN** to query LLMs with metadata and efficiently uncover valid probabilistic relationships. When observational data are available, we introduce **ReActBN**, which integrates the ReAct reasoning paradigm with structure scores such as the Bayesian Information Criterion (BIC) for iterative refinement. Unlike prior methods that offload refinement to external algorithms, our framework maintains the LLM actively in the loop throughout the discovery process. Experiments demonstrate that our method significantly outperforms both existing LLM-based approaches and traditional data-driven algorithms, particularly in the low- or no-data scenario. Code will be publicly available upon publication.

1 Introduction

Structure discovery of probabilistic graphical models (PGMs) is key to understanding complex systems, as it reveals the underlying dependency structure among variables. Traditional approaches, including constraint-based algorithms (Spirtes et al., 2001) and score-based methods (Chickering, 2002b), typically require large amounts of observational data and intensive computation, limiting their practical use. Incorporating domain knowledge is another option, but it is typically done manually, which is labor-intensive and error-prone. This motivates the development of new methods to infer probabilistic structures more effectively and efficiently.

Recently, large language models (LLMs) trained on large-scale text corpora have demonstrated remarkable performance across many NLP benchmarks (Brown, 2020). Their strong ability to capture a broad range of dependencies, including implicit causal and statistical structures, makes LLMs a promising tool for structure learning. Several approaches attempt to use LLMs as sources of prior knowledge (Ban et al., 2023b; Vashishtha et al., 2023; Long et al., 2023a) or as post-hoc verifiers (Khatibi et al., 2024). In both cases, the core structure learning process is constrained to be tightly coupled with data-driven optimization techniques. Directly eliciting full probabilistic graphs from LLMs remains challenging, mainly due to two factors: (1) the combinatorial complexity of graph construction, making it difficult for LLMs to generate globally coherent structures, and (2) the lack of effective mechanisms to enforce structural constraints (e.g., acyclicity) and to incorporate observational data within prompt-based interactions.

In this paper, we address the core challenges and demonstrate the potential of LLMs as central agents for structure learning. We formulate probabilistic graph discovery as a prompting task, where an LLM is provided with variable metadata and queried to produce a directed acyclic graph (DAG) that captures probabilistic relationships. Our framework unifies data-free estimation and score-aware refinement, keeping the LLM in the loop throughout. In Phase-1 PromptBN, we introduce an effective prompting strategy tailored to elicit valid structures from the LLM without any observational data. When data are available in

Phase-2 ReActBN, we compute structure scores (e.g., Bayesian Information Criterion) offline and incorporate them into a ReAct-style agentic workflow Yao et al. (2023). This workflow enables the LLM to supervise and refine the initial graph, in contrast to prior methods that rely on external algorithms for post-hoc correction. Our key contributions are as follows:

- **A unified framework for structure discovery**: We propose a novel framework that performs both data-free graph estimation and score-aware refinement under the supervision of an LLM. This design enables flexible structure discovery with or without data, fully within the language model loop.
- **Bayesian network estimation via prompting**: We introduce PromptBN, which utilizes a meta-prompting mechanism to construct a full Bayesian network via a single-step LLM call.
- **ReAct-based search with observational data**: We introduce ReActBN, an agentic search method that drives the LLM to Reason-and-Act (ReAct) by leveraging structure scores like BIC and its internal knowledge, thereby refining the structure initially generated from PromptBN.
- **Effectiveness, efficiency, and generalization**: Our framework outperforms SOTA LLM-based algorithms and multiple traditional algorithms with $\mathcal{O}(1)$ complexity in LLM queries. Furthermore, evaluations of newer datasets highlight the generalizability of our framework, which does not require retraining or fine-tuning.

2 Related Work

Data-driven structure learning. Conventional structure learning methods infer Directed Acyclic Graphs (DAGs) primarily from observational data, broadly categorized into *constraint-based* and *score-based* approaches Glymour et al. (2019); Scutari et al. (2019). Constraint-based methods, such as the Peter-Clark algorithm Spirtes et al. (2001) and its variants, identify probabilistic structures by assessing conditional independencies. Score-based methods, including Hill Climbing Tsamardinos et al. (2006) and Greedy Equivalence Search (GES) (Chickering, 2002a), optimize predefined scoring functions (e.g., Bayesian Information Criterion). Although these methods are effective with sufficient data, their performance deteriorates significantly with limited samples Cui et al. (2022a). To alleviate data insufficiency issues, researchers incorporate prior expert knowledge such as structural constraints or topological ordering into learning frameworks (Borboudakis & Tsamardinos, 2012; Constantinou et al., 2023; Chen et al., 2016). However, eliciting this expert knowledge remains challenging and labor intensive.

LLM-guided structure discovery with data. Recent methods explore LLMs in structure discovery, primarily as a complement to conventional approaches. Several researchers use pairwise prompting to guide LLMs in uncovering causal relationships between pairs of nodes Choi et al. (2022); Kıcıman et al. (2023); Long et al. (2023b). Jiralerspong et al. (2024) propose a Breadth-First Search (BFS) approach, in which the LLM finds all connected nodes given the currently visited nodes. Pearson correlation coefficients are also included in the prompts as hints. Beyond the pairwise or BFS prompting, Ban et al. (2023b) introduced LLM-CD, treating the LLM as a structured knowledge retriever. By asking graph-level questions, it enables joint reasoning over multiple variables, reducing prompt complexity while improving globally consistency.

In addition, LLMs have been used to provide structural priors, establish topological orderings, or refine learned probabilistic graphs post hoc (Vashishtha et al., 2023; Long et al., 2023a; Khatibi et al., 2024). Some methods also use LLM-generated statements to iteratively guide data-driven approaches (Ban et al., 2023a; Chen et al., 2023) or to incorporate observational data into prompts under both pairwise and BFS paradigms for further improvement Susanti & Färber (2025).

LLM-guided structure discovery without data. LLMs have demonstrated expert-level performance across many domains. Compared to traditional statistical methods that rely solely on sampled observational data, LLMs show promising potential to infer the actual meaning of each variable from their internal knowledge. Cohrs et al. (2024) adopts a data-free approach that leverages the LLMs' potential in structural learning. This work mainly focuses on conditional independence (CI) testing, introducing an LLM-based

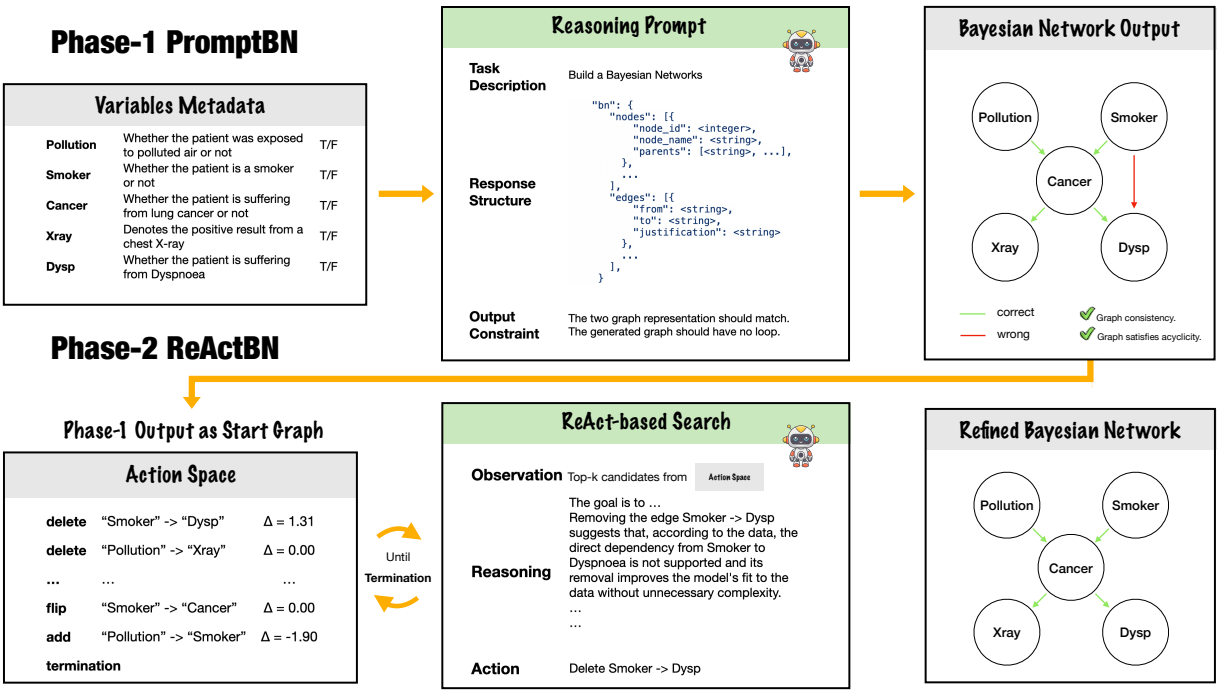


Figure 1: Overview of the proposed unified prompting framework for Bayesian network structure discovery using LLMs. The framework supports both data-free and data-driven settings through a two-phase process: Phase 1 induces the structure without observational data, while Phase 2 refines the estimated structure by incorporating data when available.

CI method that is later integrated into the PC framework for global causal discovery. It remains computationally intensive and limited to small datasets. By contrast, our approach directly constructs a causal graph by extracting implicit causal knowledge from pre-trained LLMs through a structured query prompt, with only $O(1)$ complexity. Moreover, Babakov et al. (2024) investigate the scalability of structure discovery with multiple LLM "experts". Instead, we introduce a novel unified framework that effectively integrates data-aware reasoning.

3 Problem Definition

A Bayesian network $\mathcal{G}$ is a directed acyclic graph (DAG) consisting of nodes $\mathcal{V}$ and edges $\mathcal{E}$, i.e., $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The task of structure learning for Bayesian networks typically refers to learning the structure of a DAG from the data $\mathcal{D}$. For example, score-based structure learning can be formulated as a constrained optimization problem:

$$\begin{aligned} \mathcal{G}^* &= \arg\max_{\mathcal{G} \in G} \text{Score}(\mathcal{G}, \mathcal{D}) \\ \text{s.t. } & \mathcal{G} \in \text{DAG} \end{aligned} \tag{1}$$

where $\mathcal{G}^*$ is the optimal network structure; G is the search space of all possible DAGs; $\text{Score}(\mathcal{G}, \mathcal{D})$ is a scoring function that evaluates how well the structure $\mathcal{G}$ fits the data $\mathcal{D}$.

4 Proposed Approach

We have developed **Prompt2BN**, a two-phase framework for LLM-assisted Bayesian network structures discovery.

- Phase-1 **PromptBN**: A single-step prompting approach that elicits Bayesian network structures directly from variable metadata, relying exclusively on the LLM's internal knowledge base.
- Phase-2 **ReActBN**: A ReAct-inspired agent system that integrates score-based evaluation with LLM reasoning, where the LLM supervises and selects optimal actions from top-k candidates identified through scoring mechanisms.

An overview of the proposed method is shown in Figure 1, and pseudocode for the framework is provided in the Appendix.

4.1 PromptBN: Bayesian Network Estimation via Prompting

In Phase 1, we formulate structure learning as a single-query inference task:

$$\mathcal{G} = \text{LLM-Query}(\textit{metadata}(\mathcal{V}); \textit{knowledge}(\mathcal{M})),$$

where LLM-Query$(\cdot)$ denotes a single query to the selected model $\mathcal{M}$, leveraging its implicit internal knowledge. Here, $\mathcal{V}$ represents the set of variables, and the *metadata* of each variable include its name, description of the variable, and distribution of the variable. An example input is illustrated in the architecture diagram shown in Figure 1.

Our approach employs meta-prompting Zhang et al. (2023), where the prompt defines not only the task objective but also the reasoning protocol, response structure, and output constraints. Specifically, PromptBN requires the model to produce a valid Directed Acyclic Graph (DAG) while articulating its reasoning process.

To ensure robustness, we require the model to output the graph structure in two complementary representations: a node-centric format that specifies parent nodes for each variable, and an edge-centric format that enumerates all directed edges as "from-to" pairs.

Following generation, we apply dual validation procedures. First, *Structural Consistency Validation* confirms that parent–child relationships are identical across both representations. Second, *DAG Constraint Validation* verifies acyclicity. If the output fails either criterion, we prompt the LLM to regenerate the structure until validation succeeds or the maximum retry threshold is reached.

4.2 ReActBN: ReAct-based Score-Aware Search

Phase 2 refines the initial Bayesian network generated by incorporating observational data. Inspired by the ReAct paradigm (Reason + Act), ReActBN implements an iterative search framework in which the LLM actively supervises the structure-learning process using structural scores as feedback, through alternating cycles of reasoning and action selection.

Unlike traditional score-based algorithms such as Hill Climbing or Greedy Equivalence Search, which greedily select the single highest-scoring move, ReActBN presents the LLM with the top-k candidate structures at each reasoning iteration. These candidates are derived through exhaustive enumeration of all valid neighboring structures, each annotated with its structure score and score differential. The scoring function itself is fully configurable to accommodate different statistical criteria.

At each iteration, the LLM receives comprehensive contextual information: variable metadata, the current graph structure, the current score, and the set of candidate actions with their respective scores. The LLM then outputs a recommended action (or termination decision), along with explicit reasoning and a confidence estimate. To encourage exploration and prevent cycling, we incorporate a tabu search mechanism that prohibits revisiting previously explored structures.

We maintain strict enforcement of acyclicity constraints throughout the search process, ensuring all proposed modifications preserve the DAG property. ReActBN is designed to be scalable to arbitrary dataset sizes and model-agnostic. By synergistically combining statistical evidence with domain-informed reasoning, this approach provides a flexible and interpretable framework for data-driven structural refinement, particularly valuable when working with limited or noisy observational data.

5 Experiment

In this section, we validate the effectiveness of our approach. We first describe the datasets, evaluation metrics, and experimental setup (Section 5.1). We then compare our method against state-of-the-art approaches in both data-free and data-driven settings (Section 5.2), and demonstrate its generalization capability on recent datasets (Section 5.3). Through qualitative analysis, we highlight improvements over existing data-free methods (Section 5.4). Finally, we conduct ablation studies (Section 5.6) and analyze computational efficiency (Section 5.5).

5.1 Evaluation Datasets and Metrics

Datasets. We evaluate on ten datasets for Bayesian network structure learning. Specifically, seven datasets, including Asia, Cancer, Earthquake, Child, Insurance, Alarm, and Hailfinder, are widely used benchmarks in structure learning, from Bnlearn (Scutari, 2010). The remaining three datasets, blockchain, consequence-Covid (referred to covid hereafter), and disputed3, are from BnRep (Leonelli, 2025). We retrieve the metadata of all datasets, including variable names, descriptions, and distributional properties, from the corresponding released packages, except for Insurance, whose description is taken from (Long et al., 2023a). For the Earthquake dataset, we primarily report qualitative results by visualizing the learned structure, as no reported results are available for quantitative comparison. We provide a summary of all dataset statistics in Table 1.

Table 1: Statistics of the datasets used for method validation. † covid is short for consequenceCovid.

Dataset	Asia	Cancer	Earthquake	Child	Insurance	Alarm	Hailfinder	blockchain	covid†	disputed3
#Nodes	8	5	5	20	27	37	56	12	15	27
#Edges	8	4	4	25	52	46	66	13	34	34

Evaluation metrics. We evaluate both the effectiveness and efficiency of structure learning algorithms. For effectiveness, we use two well-established metrics: ***Structural Hamming Distance (SHD)*** and ***Normalized Hamming Distance (NHD)***. SHD measures the difference between two graph structures by counting the number of edge insertions, deletions, or reversals needed to transform one graph into the other. NHD is a variant of SHD that offers a scale-invariant measure of structural difference by dividing the Hamming distance by the total number of possible edges. Following (Khatibi et al., 2024), for a graph with N nodes, NHD is calculated as $\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{N^2}\cdot\mathbf{1}(E_{ij}\neq\hat{E}_{ij})$, where E_{ij} and $\hat{E}_{ij}$ denote the presence or absence of an edge from node i to node j in the ground truth and predicted graphs, respectively. $\mathbf{1}(\cdot)$ is the indicator function, which returns 1 if $E_{ij}\neq\hat{E}_{ij}$ and 0 otherwise.

For efficiency measurement, as our work goes beyond traditional data-driven algorithms and explores the use of LLMs, we propose ***Query Complexity***, which quantifies the minimum number of API calls required for a given graph size. Specifically, let n denote the size of the input graph. The query complexity can be expressed using big-O notation as $Q(n)=\mathcal{O}(f(n))$, where $f(n)$ characterizes the growth rate of API calls as n increases.

Experimental settings and hyperparameters. To mitigate the inherent uncertainty of LLMs, we report statistics over five valid runs for each configuration. For clarity, we present the mean results in the main paper, while variance statistics are provided in the Appendix. For experiments involving data, we use 100 samples following Cui et al. (2022a), unless otherwise specified. We used the OpenAI model `o3-2025-04-16` (referred to as o3 hereafter; accessed via the OpenAI API Achiam et al. (2023)) for PromptBN and `gpt-4.1-2025-04-14` for ReActBN. For the reported results of statistical algorithms, we run PC-stable and HC using the Python pgmpy packageAnkan & Textor (2024) and report GES results from Susanti & Färber (2025). In the PC-stable and HC search, we use the default hyperparameters in pgmpy package. For ReActBN, the hyperparameters are listed in Table 2.

Table 2: Summary of key hyperparameters.

Hyperparameter	**Meaning**	**Value**
top-k	the candidate action space size	10
max_iter	the maximum number of iterations	20
tabu_length	the size of the tabu list	100
scoring	the type of structure score	BIC

Table 3: Quantitative comparison to SOTA methods.

Category	**# Sample**	**Algorithm**	**Asia**		**Cancer**		**Child**		**Insurance**		**Alarm**		**Hailfinder**	
			SHD↓	NHD↓	SHD↓	NHD↓	SHD↓	NHD↓	SHD↓	NHD↓	SHD↓	NHD↓	SHD↓	NHD↓
LLM	0	PairwisePrompt[1]	–	290	–	220	–	–	–	–	–	–	–	–
LLM	0	BFSPrompt[1]	–	40	–	150	–	–	–	–	–	–	–	–
LLM	0	Scalability (GPT-4)[2]	–	–	–	–	–	–	52	–	60	–	**68**	–
LLM	0	**PromptBN (Ours)**	**0.0**	**0.0**	**0.6**	**24**	21.6	44.5	**35.6**	43.9	41.8	30.5	76.8	185
Data	100	PC-Stable	9	156	4	160	20	65	51	73	46	37	76	24
Data	100	HC	8	125	4	160	20	68	54	67	50	40	72	24
Data	100	RAI-BF	–	–	–	–	30.4	–	54.9	–	48.4	–	–	–
Data	100	rI^{eB}	–	–	–	–	21.6	–	48.9	–	44.5	–	88.0†	–
Data	100	rBF_{chi2}	–	–	–	–	24.2	–	50.1	–	42.7	–	98.3†	–
Data	Unknown	GES[2]	–	280	–	440	–	–	–	–	–	–	–	–
LLM+Data	Unknown	PairwisePrompt+Data[2]	–	160	–	180	–	–	–	–	–	–	–	–
LLM+Data	Unknown	BFSPrompt+Data[2]	–	30	–	100	–	–	–	–	–	–	–	–
LLM+Data	100	PromptBN+HC	8.0	94	0.0	0	18.2	30	46.6	55	43.6	28	**63.8**	**18**
LLM+Data	100	**ReActBN (Ours)**	**6.4**	**75**	**0.0**	**0**	**18.0**	**28**	**40.2**	**49**	**35.4**	**24**	75.0	23

Note. The evaluation is conducted on seven Bayesian network datasets from the bnlearn repository. Different categories of methods are compared, including LLM-based approaches (top block) and data-driven approaches (bottom block). Results are reported as SHD and NHD (values scaled by $\times 10^{-3}$), where smaller values indicate better performance. [1] refers to the performance reported in Susanti & Färber (2025); [2] *Scalability* refers to Babakov et al. (2024); † indicates that Hailfinder results are based on 500 samples Cui et al. (2022a); "–" denotes unavailable values. Best results are highlighted in bold.

5.2 Quantitative Comparison to SOTA Methods

We evaluate the effectiveness of our method on 7 Bnlearn datasets and report the results in Table 3.

Quantitative comparisons to LLM-based methods. First, we demonstrate the enhancements over existing LLM-based methods. We evaluate three alternative prompting strategies: pairwise prompting, BFS prompting Susanti & Färber (2025), and Scalability (GPT-4) Babakov et al. (2024). In Table 3, we can see that our method stands out compared to other LLM-based approaches across 4 out of 5 datasets. For instance, when looking at the Asia dataset, our PromptBN manages to achieve perfect structure recovery, with both the Structural Hamming Distance (SHD) and Normalized Hamming Distance (NHD) equals to 0.0. In contrast, BFSPrompt shows an NHD of 0.04, while PairwisePrompt shows an NHD of 0.29. Furthermore, even on larger datasets like Insurance, our PromptBN consistently performs well, with an SHD of 35.6, far better than Scalability(GPT-4)'s SHD of 52. These findings really emphasize the effectiveness of our new prompting strategy.

Quantitative comparison with data-driven methods. Next, we focus on the setting where observational data are available. We compare against three lines of data-driven baselines: classical structure learning algorithms (PC-Stable (Colombo & Maathuis, 2013), Hill Climbing (HC) (Tsamardinos et al., 2006), and Greedy Equivalence Search (GES) (Chickering, 2002a)), approaches tailored for low-data regimes (RAI-BF, rI^{eB}, and rBF_{χ^2} from Cui et al. (2022b)), and hybrid methods that integrate LLMs with data-driven

Table 4: Evaluation on newer Bayesian network datasets.

Category	Algorithm	blockchain		covid		disputed3	
		SHD	NHD	SHD	NHD	SHD	NHD
LLM	**PromptBN (Ours)**	15.2	100	45.0	225	15.6	18
Data	PC-Stable	13	104	36	160	35	69
Data	HC	13.2	94	**32.6**	**156**	30.8	51
LLM+Data	**PromptBN**+HC	12.2	85	33.2	**156**	15.4	19
LLM+Data	**ReActBN (Ours)**	**11**	**74**	32.8	164	**12.2**	**15**

Note. Results are reported as SHD and NHD (values scaled by $\times 10^{-3}$), where smaller values indicate better performance. Best results are highlighted in bold.

algorithms. Specifically, PromptBN+HC refers to an HC algorithm with an initial graph estimated by PromptBN. This set of comparisons demonstrates the competitive performance of our ReActBN and its potential to improve data efficiency. LLM-CD Ban et al. (2023b) adopts a hybrid paradigm that combines priors derived from LLM and data as well. However, their evaluation was based on various sample sizes (250˘1000 per dataset). Hence, a direct numerical comparison is not appropriate.

ReActBN consistently delivers strong performance across both small and medium Bayesian networks, especially in data-scarce scenarios. The results are shown in Table 3. ReActBN achieves exact structure recovery on Cancer ($SHD = 0$, $NHD = 0$) and maintains competitive accuracy on larger networks. On Child, ReActBN achieves an NHD of 0.028 and an SHD of 18.0, outperforming PC-Stable(0.065) and HC(0.065) in NHD, RAI-BF(30.4), rI^{eB}(21.6) and rBF_{chi2}(24.2) in SHD. Another example is the Hailfinder dataset. ReActBN achieves an SHD of 75, comparable with the traditional algorithms and surpassing specialized data-efficient baselines such as rI^{eB}(88.0) and rBF_{χ^2}(98.3).

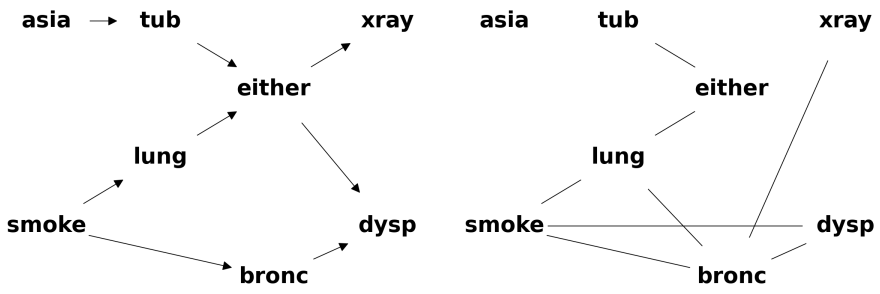
5.3 Generalization on Newer Datasets

To assess the generalization capability of our method, we evaluate it on 3 recently introduced Bayesian networks: blockchain, covid, and disputed3. These datasets differ from the widely used benchmarks and are less likely to be reflected in the training distribution of existing LLMs. Although they may not strictly be zero-shot, they serve as a strong proxy for out-of-distribution generalization. As no LLM-based baselines have reported results on these datasets, we compare our method against two standard statistical algorithms: PC-Stable and Hill Climbing (HC), using 100 data samples. The results are reported in Table 4. ReActBN achieves the $SHD = 11$ and $NHD = 74$ on blockchain, and $SHD = 12.2$, $NHD = 15$ on disputed3, lower than those of both PC-Stable and HC. On the blockchain dataset, our method achieves performance comparable to that of traditional algorithms.

Despite operating in a data-free setting, our method achieves competitive performance. For example, on the disputed3 dataset, PromptBN achieves an NHD of 0.018, substantially better than both PC-Stable (0.069) and HC (0.051). When given data, PromptBN+HC further improves performance, and our method, ReActBN, achieves the best results on two of the three datasets: blockchain and disputed3. On covid, our method performs competitively, though HC achieves a slightly lower SHD. Notably, ReActBN achieves an SHD of 12.2, reducing the error by more than half compared to HC (30.8) and PC-Stable (35).

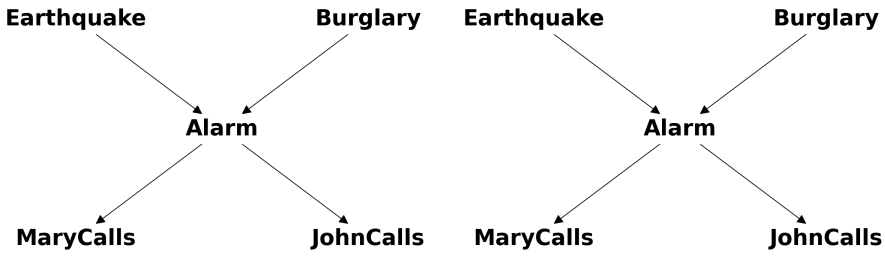
5.4 Qualitative Comparison to LLM-based Data-Free Method ChatPC

We also provide a comparison with chatPC (Cohrs et al., 2024), which reports only qualitative results. Following them, we visualize the structures learned by chatPC and ours in Figure 2. ChatPC performed constraint-based structure learning on four small datasets: burglary (also known as earthquake), cancer,



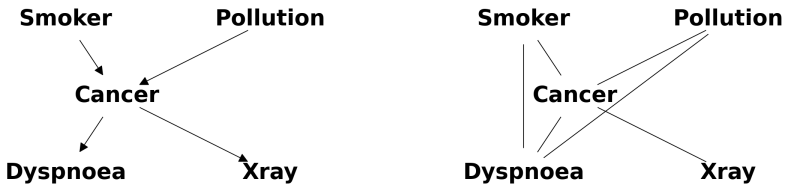
(i) Ours / GT (ii) chatPC

(a) Asia



(i) Ours / GT (ii) chatPC

(b) Earthquake



(i) Ours / GT (ii) chatPC

(c) Cancer

Figure 2: Qualitative comparison to chatPC across different datasets. Each subfigure shows results for a specific dataset: (a) Asia, (b) Earthquake, and (c) Cancer.

asia, and nao-dk-med. Due to the lack of semantic meaning and distributional properties, we are not able to apply our method to nao-dk-med dataset.

5.5 Efficiency Analysis

In terms of efficiency, we compare the query complexity of our approach with that of Pairwise prompting, BFS prompting, LLM-CD, Scalability, and chatPC. Results are shown in Table 5.

ChatPC replaces CI tests in PC with LLM queries. For each pair of variables and each conditioning set, it queries the LLM once, resulting in a query complexity of $\mathcal{O}(N^3)$. Pairwise prompting scales as $\mathcal{O}(N^2)$ since

Table 5: Comparison of query complexity across LLM-guided structure discovery algorithms.

Algorithm	PromptBN	ReActBN	Scalability	Pairwise	BFS	LLM−CD	chatPC
Query Complexity	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(E)\approx\mathcal{O}(1)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N)$	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$

each ordered pair is queried. BFS prompting reduces the complexity to $\mathcal{O}(N)$ by querying only once per visited node. LLM-CD consists of three prompt-driven stages, and the complexity is $\mathcal{O}(N^2)$ in the theoretical worst case.

Our method and *scalability* are the only works having query complexity of $\mathcal{O}(1)$. Both PromptBN and ReActBN stages have a linear query complexity due to the design of our method. Scalability adopts a multi-expert elicitation framework, where a facilitator LLM coordinates several "LLM experts" to independently reason about causal relations and aggregate their responses through majority voting. Let E denote the number of experts used, and the total query complexity is $\mathcal{O}(E) \approx \mathcal{O}(1)$ with respect to the number of variables N. Therefore, *Scalability* achieves constant query complexity per graph, comparable to our methods in efficiency but differing in design philosophy by leveraging multiple parallel experts rather than a single structured reasoning process. In contrast, both PromptBN and ReActBN achieve a query complexity of $\mathcal{O}(1)$ through a single or a constant number of LLM queries.

Table 6: Ablation study under different observational data sample sizes.

Category	**# Sample**	**Algorithm**	**Asia**		**Cancer**		**Child**		**Insurance**		**Alarm**		**Hailfinder**		**Blockchain**		**Covid**		**Disputed3**	
			SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD
Data	100	HC	8	125	4	160	20	68	54	67	50	40	72	24	13.20	94	**32.6**	**156**	30.80	51
LLM+Data	100	PromptBN+HC	8.0	94	0.0	0	18.2	30	46.6	55	43.6	28	**63.8**	**18**	12.20	85	33.20	**156**	15.40	19
LLM+Data	100	**ReActBN (Ours)**	**6.4**	**75**	**0.0**	**0**	**18.0**	**28**	**40.2**	**49**	**35.4**	**24**	75.0	23	**11**	**74**	32.8	164	**12.2**	**15**
Data	250	HC	3	94	4	160	17	48	49	63	38	37	76	25	13	90	30	133	44	78
LLM+Data	250	PromptBN+HC	**1.0**	**16**	**0**	**0**	13.2	26	37.6	46	32.8	**26**	**65.8**	**19**	**9.60**	79	**28.40**	**140**	11.80	13
LLM+Data	250	**ReActBN (Ours)**	**1.0**	**16**	**0**	**0**	**12.6**	28	**32.4**	**39**	**32.2**	**25**	69.6	22	10.20	**75**	30.80	157	**9.80**	**12**

Note. Results are reported using Structural Hamming Distance (SHD) and Normalized Hamming Distance (NHD), with values scaled by $\times 10^{-3}$. Smaller values indicate better performance. The best results in each block are highlighted in **bold**.

Table 7: Ablation study on the use of different LLMs with our PromptBN.

Model	**Asia**		**Cancer**		**Child**		**Insurance**		**Alarm**		**Hailfinder**		**blockchain**		**covid**		**disputed3**	
	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD
o3-pro	0.0	0	1.0	40	22.2	45	37.6	41	44.8	33	79.6	26	12.8	85	44.2	226	9.8	10
o3	0.0	0	0.6	24	21.6	44	35.6	43	41.8	30	76.8	57	15.2	45	45.0	224	15.6	18
o4-mini	0.0	0	0.0	0	22.8	50	43.2	53	46.8	35	81.8	55	13.2	42	42.8	234	19.6	24
gpt-4.1	0.0	0	0.0	0	18.2	23	39.6	45	37.6	27	71.2	48	16.0	46	46.8	241	20.0	24
gpt-4o	0.0	0	0.0	0	22.0	40	46.2	54	32.0	21	64.0	20	14.4	94	44.0	229	26.4	34
Deepseek-r1	0.0	0	0.0	0	18.2	38	48.8	37	48.8	37	65.4	21	13.6	44	44.2	237	5.2	7
Deepseek-v3	0.0	0	0.0	0	21.2	46	UnP	UnP	UnP	UnP	65.8	21	16.2	41	41.4	207	28.2	33
gemini-2.5-pro	0.0	0	0.0	0	18.4	31	33.0	38	14.2	11	71.8	23	9.6	67	41.8	228	9.4	9
gemini-2.0-flash	0.0	0	0.0	0	21.2	45	47.4	64	40.4	29	54.4	17	14.0	94	39.8	192	28.0	38

Note. Results are reported using SHD and NHD (values scaled by $\times 10^{-3}$), where smaller values indicate better performance. "UnP" (Unparsable) denotes cases where the LLM fails to strictly follow the output formatting requirements specified in the prompt, resulting in a response that cannot be parsed into a valid structure.

5.6 Ablation Study

In this section, we conduct ablation studies and address two key questions as follows.

5.6.1 Q1: How does the sample size of observational data affect performance?

We investigate how access to different amounts of observational data influences the performance of our framework. Following widely adopted evaluation protocols Cui et al. (2022a), we compare our approach under two data regimes: 100 and 250 samples. For each setting, we evaluate three variants: (1) HC as a purely data-driven baseline, (2) PromptBN+HC, which uses an LLM to initialize HC, and (3) ReActBN, which further leverages LLM reasoning to guide the score-based selection.

As shown in Table 6, increasing the sample size to 250 leads to further performance improvements across all methods. Notably, our proposed ReActBN consistently outperforms both HC and PromptBN+HC. For example, on Child, ReActBN achieves an SHD of 12.6, compared to 13.2 for PromptBN+HC and 17 for HC. As the amount of data increases, the performance of conventional data-driven methods improves accordingly, thereby narrowing the gap between them and our ReActBN. This trend is expected, as the reliance on LLM priors naturally decreases when sufficient data are available for reliable structure estimation.

5.6.2 Q2: How does the choice of LLM affect structure learning performance?

We examine how different language models impact the effectiveness of our proposed PromptBN. To this end, we evaluate PromptBN across a diverse set of LLMs, including proprietary models from OpenAI and Google, and open-source models from Deepseek. The tested models range from chat-oriented ones (e.g., DeepSeek-v3, GPT-4o, GPT-4.1) to reasoning-focused models (e.g., DeepSeek-r1, OpenAI o-series), and general-purpose language models (e.g., Gemini-2.5-Pro and Gemini-2.0-Flash). The results are presented in Table 7. Interestingly, more advanced models with stronger general-purpose reasoning capabilities do not always achieve better performance in structure learning. For instance, o3-pro, the strongest reasoning model from OpenAI, outperforms other models on most LLM benchmarking datasets. However, in our experiments on bayesian network structure discovery, o3-pro performs worse than GPT-4o, the oldest model evaluated (o3-pro has an SHD of 44.8, while GPT-4o achieves 32.0). Similarly, Google's best model, Gemini-2.5-Pro, performs worse than Gemini-2.0-Flash on Hailfinder (with SHD of 71.8 vs. 54.4).

6 Conclusion

In this work, we propose a unified framework for Bayesian network structure discovery using LLMs. The framework consists of a single-step prompting module that generates an initial graph from variable metadata and a refinement phase that integrates LLM reasoning with score-based search. Our method achieves strong performance across established benchmarks, surpassing existing LLM-based approaches and several traditional data-driven algorithms. In particular, it accurately recovers exact structures on small networks and maintains high accuracy on medium-sized graphs, even with limited data. Furthermore, by framing structure learning as an inference task, our approach enhances computational efficiency.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Ankur Ankan and Johannes Textor. pgmpy: A Python Toolkit for Bayesian Networks. *Journal Of Machine Learning Research*, (25):1–8, 2024.

Nikolay Babakov, Ehud Reiter, and Alberto Bugarin. Scalability of bayesian network structure elicitation with large language models: a novel methodology and comparative analysis. *arXiv preprint arXiv:2407.09311*, 2024.

Taiyu Ban, Lyuzhou Chen, Derui Lyu, Xiangyu Wang, and Huanhuan Chen. Causal structure learning supervised by large language model. *arXiv preprint arXiv:2311.11689*, 2023a.

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

Giorgos Borboudakis and Ioannis Tsamardinos. Incorporating causal prior knowledge as path-constraints in bayesian networks and maximal ancestral graphs. *arXiv preprint arXiv:1206.6390*, 2012.

Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Eunice Yuh-Jie Chen, Yujia Shen, Arthur Choi, and Adnan Darwiche. Learning bayesian networks with ancestral constraints. *Advances in Neural Information Processing Systems*, 29, 2016.

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

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

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

Kristy Choi, Chris Cundy, Sanjari Srivastava, and Stefano Ermon. Lmpriors: Pre-trained language models as task-specific priors. *arXiv preprint arXiv:2210.12530*, 2022.

Kai-Hendrik Cohrs, Gherardo Varando, Emiliano Diaz, Vasileios Sitokonstantinou, and Gustau Camps-Valls. Large language models for constrained-based causal discovery. *arXiv preprint arXiv:2406.07378*, 2024.

Diego Colombo and Marloes H. Maathuis. Order-independent constraint-based causal structure learning, 2013. URL `https://arxiv.org/abs/1211.3295`.

Anthony C Constantinou, Zhigao Guo, and Neville K Kitson. The impact of prior knowledge on causal structure learning. *Knowledge and Information Systems*, 65(8):3385–3434, 2023.

Zijun Cui, Naiyu Yin, Yuru Wang, and Qiang Ji. Empirical bayesian approaches for robust constraint-based causal discovery under insufficient data. *arXiv preprint arXiv:2206.08448*, 2022a.

Zijun Cui, Naiyu Yin, Yuru Wang, and Qiang Ji. Empirical bayesian approaches for robust constraint-based causal discovery under insufficient data. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 4850–4856. International Joint Conferences on Artificial Intelligence Organization, July 2022b. doi: 10.24963/ijcai.2022/672. URL `https://doi.org/10.24963/ijcai.2022/672`. Main Track.

Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Thomas Jiralerspong, Xiaoyin Chen, Yash More, Vedant Shah, and Yoshua Bengio. Efficient causal graph discovery using large language models. *arXiv preprint arXiv:2402.01207*, 2024.

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. *arXiv preprint arXiv:2305.00050*, 2023.

Manuele Leonelli. bnRep: A repository of Bayesian networks from the academic literature. *Neurocomputing*, 624:129502, 2025.

Stephanie Long, Alexandre Piché, Valentina Zantedeschi, Tibor Schuster, and Alexandre Drouin. Causal discovery with language models as imperfect experts. *arXiv preprint arXiv:2307.02390*, 2023a.

Stephanie Long, Tibor Schuster, and Alexandre Piché. Can large language models build causal graphs? *arXiv preprint arXiv:2303.05279*, 2023b.

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

Marco Scutari, Catharina Elisabeth Graafland, and José Manuel Gutiérrez. Who learns better bayesian network structures: Accuracy and speed of structure learning algorithms. *International Journal of Approximate Reasoning*, 115:235–253, 2019.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2001.

Yuni Susanti and Michael Färber. Can llms leverage observational data? towards data-driven causal discovery with llms. *arXiv preprint arXiv:2504.10936*, 2025.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65:31–78, 2006.

Aniket Vashishtha, Abbavaram Gowtham Reddy, Abhinav Kumar, Saketh Bachu, Vineeth N Balasubramanian, and Amit Sharma. Causal inference using llm-guided discovery. *arXiv preprint arXiv:2310.15117*, 2023.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Yifan Zhang, Yang Yuan, and Andrew Chi-Chih Yao. Meta prompting for ai systems. *arXiv preprint arXiv:2311.11482*, 2023.

A Pseudo Code and Examples

A.1 PromptBN

Pseudo-code of the PromptBN (Phase-1 in our framework) is in Algorithm 1.

Algorithm 1 PromptBN: Bayesian Network Estimation via Prompting

0: **Input:** Variable metadata table $\mathcal{I} = metadata(\mathcal{V})$, where $\mathcal{V}$ is the set of variables; LLM model $\mathcal{M}$; retry limit N
0: **Output:** Bayesian network $\mathcal{G}$
0: Construct meta-prompt $\mathcal{P}$ including:
0: Task definition: instruct the LLM to infer a Bayesian network from $\mathcal{I}$
0: Variable schema: include each variable's name, description, and distribution
0: Output format: specify both node-centric and edge-centric representations
0: Reasoning protocol: instruct the LLM to reason over variable metadata
0: Response constraints: require strict formatting, completeness, and acyclicity
0: retry_count $\leftarrow 1$
0: **repeat**
0: $\mathcal{O}_{raw} \leftarrow$ LLM-Query$(\mathcal{I}, \mathcal{P}; \mathcal{M})$
0: Parse $\mathcal{O}_{raw}$ to obtain both:
0: $\mathcal{G}_{node}$: node-centric representation
0: $\mathcal{G}_{edge}$: edge-centric representation
0: retry_count $\leftarrow$ retry_count +1
0: **until** (consistent$(\mathcal{G}_{node}, \mathcal{G}_{edge})$ **and** DAG$(\mathcal{G}_{edge})$) **or** retry_count $> N$
0: **if** retry_count $> N$ **then**
0: **Raise error or halt:** Valid output not achieved within N retries.
0: **else**
0: **return** Bayesian network $\mathcal{G} \leftarrow \mathcal{G}_{edge}$
0: **end if**
=0

A.2 Input and Output Examples

An example of variable metadata $\mathcal{I}$ is shown in Table 8.

Table 8: Example input: $\mathcal{I}_{\text{Asia}} = metadata(\mathcal{V}_{\text{Asia}})$

node	var_name	var_description	var_distribution
1	asia	visit to Asia	a two-level factor with levels *yes* and *no*.
2	tub	tuberculosis	a two-level factor with levels *yes* and *no*.
3	smoke	smoking	a two-level factor with levels *yes* and *no*.
4	lung	lung cancer	a two-level factor with levels *yes* and *no*.
5	bronc	bronchitis	a two-level factor with levels *yes* and *no*.
6	either	tuberculosis versus lung cancer/bronchitis	a two-level factor with levels *yes* and *no*.
7	xray	chest X-ray	a two-level factor with levels *yes* and *no*.
8	dysp	dyspnoea	a two-level factor with levels *yes* and *no*.

The following is an example an LLM response for Asia dataset, illustrating the defined structure query.

```
"bn": {
    "nodes": [
        ...,
        {
```

```
            "node_id": 8,
            "node_name": "dysp",
            "parents": ["bronc", "either"],
            "description": "dyspnoea",
            "distribution": "a two-level factor with levels yes and no.",
            "conditional_probability_table": "P(dysp|bronc, either)"
        }
    ],
    "edges": [
        ...,
        {
            "from": "bronc",
            "to": "dysp",
            "justification": "Bronchitis can cause dyspnoea."
        },
        {
            "from": "either",
            "to": "dysp",
            "justification": "Either tuberculosis or lung cancer can cause dyspnoea."
        }
    ],
    "network_summary": "The Bayesian Network models the relationships between visiting Asia, smoking, and respiratory
    diseases such as tuberculosis, lung cancer, and bronchitis. It captures how these factors influence symptoms like
    dyspnoea and diagnostic results like chest X-rays."
}
```


A.3 ReActBN

Pseudo-code of the ReActBN (Phase-2 in our framework) is in Algorithm 2.

B Experiments

B.1 Variance Statistics

In all our experiments, we report the mean and standard deviation with 5 valid runs under the same settings. To complement the results in the main article, we report the **standard deviation** in the following tables to provide a clearer view of the stability and robustness of our method:

- Table 9 reports the quantitative comparison with the SOTA methods, supplementing Table 3 in the main article.
- Table 10 reports the evaluation on newer Bayesian network datasets for the generalization evaluation, supplementing Table 4 in the main paper.
- Table 11 reports the ablation study in different sample sizes, supplementing Table 6 in the main article.
- Table 12 reports the ablation study with different LLMs, supplementing Table 7 in the main article.

Algorithm 2 ReActBN: ReAct-based Score-Aware Search

Require: Initial Bayesian network $\mathcal{G}_0$; variable metadata $\mathcal{I}$; observational data $\mathcal{D}$
Require: Hyperparameters: top-k k; scoring method $\mathcal{S}$; LLM model $\mathcal{M}$; tabu list length L; maximum iterations T
Ensure: Refined Bayesian network $\mathcal{G}^*$

$\mathcal{G} \leftarrow \mathcal{G}_0$; $s \leftarrow \mathcal{S}(\mathcal{G})$; $TL \leftarrow \emptyset$; $t \leftarrow 1$
while $t \leq T$ **do**
  $\mathcal{A} \leftarrow \emptyset$
  for all ordered pairs of nodes (u, v) **do**
    for all actions $a \in \{\text{ADD}, \text{REMOVE}, \text{FLIP}\}$ **do**
      if a is applicable to (u, v) and applying a keeps the graph acyclic **then**
        $\mathcal{N} \leftarrow \mathcal{G} + a(u, v)$
        if $\mathcal{N} \notin TL$ **then**
          $\mathcal{A} \leftarrow \mathcal{A} \cup \{a(u, v)\}$
          $s_{a(u,v)} \leftarrow \mathcal{S}(\mathcal{N})$
          $\Delta_{a(u,v)} \leftarrow s_{a(u,v)} - s$
        end if
      end if
    end for
  end for
  $\mathcal{A}_{\text{cand}} \leftarrow$ top-k elements of $\mathcal{A}$ with the largest $\Delta_{a(u,v)}$
  Build prompt $\mathcal{P}$ using $(\mathcal{I}, \mathcal{G}, s, \mathcal{A}_{\text{cand}})$
  $idx \leftarrow \text{LLMQuery}(\mathcal{P}; \mathcal{M})$ $\{idx \in \{-1, 0, \ldots, k-1\}\}$
  if $idx = -1$ **then**
    break
  else
    $a^\star \leftarrow \mathcal{A}_{\text{cand}}[idx]$
    $\mathcal{G}_{\text{new}} \leftarrow \mathcal{G} + a^\star$
    if not acyclic$(\mathcal{G}_{\text{new}})$ **then**
      continue
    end if
    $TL \leftarrow TL \cup \{\mathcal{G}_{\text{new}}\}$
    if $|TL| > L$ **then**
      remove oldest element from TL
    end if
    $\mathcal{G} \leftarrow \mathcal{G}_{\text{new}}$
    $s \leftarrow \mathcal{S}(\mathcal{G})$
  end if
  $t \leftarrow t + 1$
end while
return $\mathcal{G}^* \leftarrow \mathcal{G}$ =0

Table 9: Quantitative comparison to SOTA methods.

Category	#Sample	Algorithm	Asia		Cancer		Child		Insurance		Alarm		Hailfinder	
			SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD
LLM	0	Claude+/100k[1]	4/7	–	1/0	–	32/38	–	49/49	–	53/59	–	–	–
LLM	0	GPT-3.5-turbo[1]	3	–	2	–	36	–	54	–	46	–	–	–
LLM	0	GPT-4[1]	3	–	1	–	**21**	–	48	–	**40**	–	–	–
LLM	0	PairwisePrompt[2]	–	290	–	220	–	–	–	–	–	–	–	–
LLM	0	BFSPrompt[2]	–	40	–	150	–	–	–	–	–	–	–	–
LLM	0	**PromptBN (Ours)**	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0 \pm 0}$	$\mathbf{0.60 \pm 0.55}$	$\mathbf{24 \pm 22}$	21.60 ± 5.73	44 ± 12	$\mathbf{35.60 \pm 3.78}$	$\mathbf{44 \pm 4}$	41.80 ± 5.22	31 ± 4	76.80 ± 6.72	25 ± 3
Data	100	PC-Stable	9	156	4	160	20	65	51	73	46	37	76	24
Data	100	HC	8	125	4	160	20	68	54	67	50	40	72	24
Data	100	RAI-BF	–	–	–	–	30.4 ± 3.7	–	54.9 ± 3.6	–	48.4 ± 5.8	–	–	–
Data	100	rI^{eB}	–	–	–	–	21.6 ± 2.1	–	48.9 ± 1.3	–	44.5 ± 2.2	–	$88.0 \pm 2.0^{\dagger}$	–
Data	100	rBF_{chi2}	–	–	–	–	24.2 ± 2.3	–	50.1 ± 2.9	–	42.7 ± 2.3	–	$98.3 \pm 1.5^{\dagger}$	–
Data	–	GES[2]	–	280	–	440	–	–	–	–	–	–	–	–
LLM+Data	–	PairwisePrompt+Data[2]	–	160	–	180	–	–	–	–	–	–	–	–
LLM+Data	–	BFSPrompt+Data[2]	–	30	–	100	–	–	–	–	–	–	–	–
LLM+Data	100	**PromptBN**+HC	8.00 ± 0.00	94 ± 0	0.00 ± 0.00	0 ± 0	18.20 ± 1.10	30 ± 3	46.60 ± 2.41	55 ± 6	43.60 ± 0.55	28 ± 1	$\mathbf{63.80 \pm 0.45}$	$\mathbf{18 \pm 0}$
LLM+Data	100	**ReActBN(Ours)**	$\mathbf{6.40 \pm 2.19}$	$\mathbf{75 \pm 26}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0 \pm 0}$	$\mathbf{18.00 \pm 1.58}$	$\mathbf{28 \pm 4}$	$\mathbf{40.20 \pm 4.49}$	$\mathbf{49 \pm 4}$	$\mathbf{35.40 \pm 3.44}$	$\mathbf{24 \pm 1}$	75.00 ± 3.87	23 ± 2

Note. The evaluation is conducted on seven Bayesian network datasets from the bnlearn repository. Different categories of methods are compared, including LLM-based approaches (top block) and data-driven approaches (bottom block). Results are reported as SHD and NHD (values scaled by $\times 10^{-3}$), where smaller values indicate better performance. [1] denotes the method using three-stage prompting from Ban et al. (2023b); [2] refers to the performance reported in Susanti & Färber (2025); † indicates that Hailfinder results are based on 500 samples, following Cui et al. (2022a). "–" denotes values that are unavailable. Best results are highlighted in bold. **This table is supplementing Table 3 in the main article with std.**

Table 10: Evaluation on newer Bayesian network datasets.

Category	# Sample	Algorithm	blockchain		covid		disputed3	
			SHD	NHD	SHD	NHD	SHD	NHD
LLM	0	**PromptBN (Ours)**	15.20 ± 6.54	100 ± 40	45.00 ± 2.83	225 ± 17	15.60 ± 2.19	18 ± 4
Data	100	PC-Stable	13.00 ± 0.00	104 ± 0	36.00 ± 0.00	160 ± 0	35.00 ± 0.00	69 ± 0
Data	100	HC	13.20 ± 1.92	94 ± 0	$\mathbf{32.60 \pm 1.52}$	$\mathbf{156 \pm 0}$	30.80 ± 2.28	51 ± 0
LLM+Data	100	PromptBN+HC	12.20 ± 1.64	85 ± 22	33.20 ± 2.49	156 ± 8	15.40 ± 4.34	19 ± 6
LLM+Data	100	**ReActBN (Ours)**	$\mathbf{11.00 \pm 0.00}$	$\mathbf{74 \pm 6}$	32.80 ± 2.49	164 ± 14	$\mathbf{12.20 \pm 2.86}$	$\mathbf{15 \pm 5}$

Note. Results are reported as SHD and NHD (values scaled by $\times 10^{-3}$), where smaller values indicate better performance. Best results are highlighted in bold. **This table is supplementing Table 4 in the main article with std.**

Table 11: Ablation study under different sample sizes.

Category	# Sample	Algorithm	Asia		Cancer		Child		Insurance		Alarm	
			SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD
Data	100	HC	8	125	4	160	20	68	54	67	50	40
LLM+Data	100	PromptBN+HC	8.00 ± 0.00	94 ± 0	0.00 ± 0.00	0 ± 0	18.20 ± 1.10	30 ± 3	46.60 ± 2.41	55 ± 6	43.60 ± 0.55	28 ± 1
LLM+Data	100	**ReActBN (Ours)**	$\mathbf{6.40 \pm 2.19}$	$\mathbf{75 \pm 26}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0 \pm 0}$	$\mathbf{18.00 \pm 1.58}$	$\mathbf{28 \pm 4}$	$\mathbf{40.20 \pm 4.49}$	$\mathbf{49 \pm 4}$	$\mathbf{35.40 \pm 3.44}$	$\mathbf{24 \pm 1}$
Data	250	HC	3	94	4	160	17	48	49	63	38	37
LLM+Data	250	PromptBN+HC	$\mathbf{1.00 \pm 0.00}$	$\mathbf{16 \pm 0}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0 \pm 0}$	13.20 ± 1.10	26 ± 4	37.60 ± 3.36	46 ± 5	32.80 ± 3.63	26 ± 3
LLM+Data	250	**ReActBN (Ours)**	$\mathbf{1.00 \pm 0.00}$	$\mathbf{16 \pm 0}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0 \pm 0}$	$\mathbf{12.60 \pm 1.34}$	28 ± 6	$\mathbf{32.40 \pm 4.16}$	$\mathbf{39 \pm 2}$	$\mathbf{32.20 \pm 2.49}$	$\mathbf{25 \pm 3}$

Category	# Sample	Algorithm	Hailfinder		blockchain		covid		disputed3	
			SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD
Data	100	HC	72	24	13.20	94	**32.6**	**156**	30.80	51
LLM+Data	100	PromptBN+HC	$\mathbf{63.80 \pm 0.45}$	$\mathbf{18 \pm 0}$	12.20 ± 1.64	85 ± 22	33.20 ± 2.49	156 ± 8	15.40 ± 4.34	19 ± 6
LLM+Data	100	**ReActBN (Ours)**	75.00 ± 3.87	23 ± 2	$\mathbf{11.00 \pm 0.00}$	$\mathbf{74 \pm 6}$	$\mathbf{32.80 \pm 2.49}$	164 ± 14	$\mathbf{12.20 \pm 2.86}$	$\mathbf{15 \pm 5}$
Data	250	HC	76	25	13	90	30	133	44	78
LLM+Data	250	PromptBN+HC	$\mathbf{65.80 \pm 0.84}$	$\mathbf{19 \pm 1}$	$\mathbf{9.60 \pm 1.34}$	79 ± 6	$\mathbf{28.40 \pm 1.95}$	$\mathbf{140 \pm 4}$	11.80 ± 7.36	13 ± 6
LLM+Data	250	**ReActBN (Ours)**	69.60 ± 1.52	22 ± 1	10.20 ± 1.64	75 ± 3	30.80 ± 1.30	157 ± 14	$\mathbf{9.80 \pm 5.85}$	$\mathbf{12 \pm 7}$

Note. Results are reported as SHD and NHD (values scaled by $\times 10^{-3}$), where smaller values indicate better performance. Best results are highlighted in bold in each block. **This table is supplementing Table 6 in the main article with std.**

Table 12: Ablation study on different LLMs with our PromptBN.

Model	Asia		Cancer		Child		Insurance		Alarm	
	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD
o3-pro	0.00 ± 0.00	0 ± 0	1.00 ± 0.00	40 ± 0	22.20 ± 4.87	45 ± 11	37.60 ± 2.61	41 ± 5	44.80 ± 5.31	33 ± 4
o3	0.00 ± 0.00	0 ± 0	0.60 ± 0.55	24 ± 22	21.60 ± 5.73	44 ± 12	35.60 ± 3.78	44 ± 4	41.80 ± 5.22	31 ± 4
o4-mini	0.00 ± 0.00	0 ± 0	0.00 ± 0.00	0 ± 0	22.80 ± 5.07	50 ± 11	43.20 ± 4.32	53 ± 7	46.80 ± 8.17	35 ± 6
gpt-4.1	0.00 ± 0.00	0 ± 0	0.00 ± 0.00	0 ± 0	18.20 ± 1.64	23 ± 4	39.60 ± 2.73	45 ± 2	37.60 ± 2.97	27 ± 2
gpt-4o	0.00 ± 0.00	0 ± 0	0.00 ± 0.00	0 ± 0	22.00 ± 0.00	40 ± 0	46.20 ± 0.44	54 ± 2	32.00 ± 6.12	21 ± 3
Deepseek-r1	0.00 ± 0.00	0 ± 0	0.00 ± 0.00	0 ± 0	18.20 ± 4.09	38 ± 13	35.40 ± 6.99	42 ± 8	48.80 ± 7.60	37 ± 9
Deepseek-v3	0.00 ± 0.00	0 ± 0	0.00 ± 0.00	0 ± 0	21.20 ± 0.45	46 ± 2	UnP	UnP	UnP	UnP
gemini-2.5-pro	0.00 ± 0.00	0 ± 0	0.00 ± 0.00	0 ± 0	18.40 ± 3.72	31 ± 11	33.00 ± 3.74	38 ± 4	14.20 ± 6.26	11 ± 4
gemini-2.0-flash	0.00 ± 0.00	0 ± 0	0.00 ± 0.00	0 ± 0	21.20 ± 1.92	45 ± 6	47.40 ± 3.72	64 ± 6	40.40 ± 3.29	29 ± 2

Model	Hailfinder		blockchain		consequenceCovid		disputed3	
	SHD	NHD	SHD	NHD	SHD	NHD	SHD	NHD
o3-pro	79.60 ± 11.68	26 ± 4	12.80 ± 4.03	85 ± 20	44.20 ± 2.78	226 ± 11	9.80 ± 2.78	10 ± 2
o3	76.80 ± 6.72	25 ± 3	15.20 ± 6.54	100 ± 40	45.00 ± 2.83	225 ± 17	15.60 ± 2.19	18 ± 4
o4-mini	81.80 ± 10.76	27 ± 4	13.20 ± 1.92	92 ± 13	42.80 ± 3.03	234 ± 13	19.60 ± 5.27	24 ± 8
gpt-4.1	64.00 ± 6.78	20 ± 2	16.00 ± 0.00	111 ± 0	46.80 ± 1.48	242 ± 7	20.00 ± 4.90	25 ± 6
gpt-4o	64.00 ± 0.00	20 ± 0	14.40 ± 0.55	107 ± 4	44.00 ± 1.23	229 ± 7	26.40 ± 1.67	34 ± 3
Deepseek-r1	65.40 ± 6.31	21 ± 2	13.60 ± 3.72	94 ± 26	44.20 ± 5.17	237 ± 24	5.20 ± 2.17	7 ± 3
Deepseek-v3	UnP	UnP	16.20 ± 1.34	118 ± 7	41.40 ± 2.30	207 ± 17	28.20 ± 2.05	33 ± 3
gemini-2.5-pro	71.80 ± 7.66	23 ± 2	9.60 ± 2.61	67 ± 18	41.80 ± 2.28	228 ± 9	9.40 ± 3.72	9 ± 4
gemini-2.0-flash	54.40 ± 2.70	17 ± 1	14.0 ± 2.74	94 ± 14	39.80 ± 1.48	192 ± 13	28.00 ± 5.52	38 ± 9

Note. Results are reported as SHD and NHD (values scaled by $\times 10^{-3}$), where smaller values indicate better performance. "UnP" (UnParsable) refers to an error that occurs when the LLM fails to strictly follow the output formatting requirements specified in the prompt, resulting in a response that cannot be successfully preprocessed or parsed into a valid structure. **This table is supplementing Table 7 in the main article with std.**