HiBCD: Scalable Bayesian Causal Discovery for High-Dimensional Data

Anonymous authors
Paper under double-blind review

Abstract

Bayesian causal structure learning aims to infer causal graphs from observational data based on Bayesian posterior inference. However, the combinatorial nature of the graph space makes these methods computationally intractable for high-dimensional settings where the number of variables is large. To address this challenge, we propose HiBCD, a divide-and-conquer Bayesian causal discovery method for high-dimensional variables. The method first employs a hierarchical weighted sampling strategy to partition the high-dimensional variable set into several smaller, manageable subsets. Subsequently, for each subset, we infer local ancestral probabilities using a pre-trained meta estimator and derive a causal ordering. The meta estimator features a symmetry-aware encoder and a Bayesian scoring head and is trained via a variational objective to ensure robustness to latent confounders. Finally, these local orderings are iteratively aggregated into a global causal topology, from which the full causal graph can be efficiently recovered. Experiments demonstrate that HiBCD significantly outperforms previous Bayesian causal discovery methods on large-scale data in terms of structural accuracy, while maintaining highly scalable computational efficiency.

1 Introduction

Discovering causal relationships from observational data is a fundamental challenge central to scientific and artificial intelligence research. Traditional methods Spirtes & Glymour (1991); Zheng et al. (2018) often seek a single causal graph, but may require interventional data or face identifiability issues where observational data alone is insufficient to distinguish between competing structures. Bayesian Causal Discovery (BCD) Lorch et al. (2022); Dhir et al. (2025) offers a principled solution to this ambiguity by inferring a posterior distribution over the space of possible Directed Acyclic Graphs (DAGs). Instead of committing to a single structure, this approach quantifies the uncertainty associated with each potential causal link, offering greater flexibility.

However, the primary bottleneck for BCD methods is their significant computational complexity. The space of possible DAGs grows super-exponentially with the number of variables, making exhaustive search or sampling computationally intractable for high-dimensional problems. Consequently, most state-of-the-art BCD methods struggle to scale beyond a hundred variables Viinikka et al. (2020). This scalability ceiling stands in stark contrast to the demands of complex real-world systems, where high-dimensional modeling is often a prerequisite. For example, such demands arise in inferring gene regulatory networks Greenfield et al. (2010), mapping functional connectivity across hundreds of brain regions Glasser et al. (2016), and performing root cause analysis in extensive microservice architectures Ding et al. (2023); Gan et al. (2023). On the other hand, while some non-Bayesian methods have been developed specifically for large-scale discovery Dong et al. (2025); Gu & Zhou (2020), their scalability often comes at the cost of accuracy. Meanwhile, these methods typically rely on restrictive assumptions (e.g., linear causal relations), or employ oversimplified heuristics, limiting the ability to model the complex dynamics in real-world systems.

To bridge the gap between modeling flexibility and scalability, we propose HiBCD, a novel Hierarchical Bayesian Causal Discovery framework designed for high-dimensional settings. HiBCD is built on a divide-and-conquer paradigm that systematically deconstructs the high-dimensional problem. The core idea is to iteratively partition the full variable set into smaller, computationally affordable

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.

subsets, infer local causal knowledge within each subset, and then aggregate this local knowledge into a coherent global causal structure.

The implementation of this paradigm requires careful design to address a series of sub-challenges. First, to ensure the partitioning is both comprehensive and focuses on causally significant variables, our sampler employs a dual-pool strategy for coverage and an adaptive weighting mechanism that prioritizes variables based on their variance and emerging rank in the causal hierarchy. Second, to robustly infer local causal structure in the presence of unobserved confounders (i.e., variables outside the current subset), our Estimator is designed as a sophisticated meta-model for amortized variational inference. The Estimator is trained offline to predict local ancestral probabilities and derive a causal ordering for each subset, using a specialized symmetry-aware encoder and a Bayesian scoring head. Finally, to reconcile potentially conflicting evidence from different local analyses, our Aggregator uses a principled two-stage process that first establishes a global preference ranking and then iteratively refines it to construct the final global ordering.

The main contributions of this paper are threefold:

1. We propose HiBCD [1], a complete and scalable divide-and-conquer framework for Bayesian causal discovery, effectively bridging the gap between expressive, nonlinear modeling and high-dimensional applicability.
2. We design a novel and robust meta-model for amortized inference of local causal orderings. This Estimator, featuring a symmetry-aware architecture and trained with a variational objective, mitigates the effects of latent confounders and serves as a powerful, reusable component for local causal analysis.
3. We conduct extensive experiments on synthetic and real-world benchmarks, demonstrating that HiBCD significantly outperforms existing state-of-the-art Bayesian causal discovery algorithms in structural accuracy, while demonstrating remarkable scalability and efficiency in high-dimensional settings.

2 Background and Related Works

2.1 Bayesian causal structure learning

We consider the problem of learning causal relationships among random variables $\mathbf{X} = (X_1, \ldots, X_d)$ from n observations, represented by a data matrix $\mathcal{D} \in \mathbb{R}^{n \times d}$. Its j-th column, $\mathbf{x}_j = \mathcal{D}_{:,j}$, contains the observations of X_j. We assume the data is generated by a Structural Causal Model (SCM), defined by a Directed Acyclic Graph (DAG) $\mathcal{G} = (V, E) \in \mathbb{G}_d$, where $V = \{1, \ldots, d\}$, E is the directed edge set, and $\mathbb{G}_d$ is the space of DAGs over d variables. Each variable follows a function of its causal parents:

$$X_i = f_i(\mathbf{X}_{\mathrm{PA}_{\mathcal{G}}(i)},\ \varepsilon_i), \quad \text{for } i \in \{1, \ldots, d\}, \tag{1}$$

where $\mathrm{PA}_{\mathcal{G}}(i)$ denotes the set of parent nodes of i in $\mathcal{G}$, f_i is the causal mechanism, and ε_i is a random noise term independent of its causes.

Our objective is to recover the underlying causal graph $\mathcal{G}$ from the observational data $\mathcal{D}$ based on Bayesian posterior inference. Following much of the previous work Lorch et al. (2022); Dhir et al. (2025); Reizinger et al. (2025); Cüppers et al. (2024), we assume the model satisfies the Causal Markov Condition Pearl (2009) and the Independent Causal Mechanism (ICM) Janzing & Schölkopf (2010) assumption. Under these assumptions, the posterior probability of a specific DAG $\mathcal{G}$ given the data $\mathcal{D}$ can be computed via Bayes' theorem:

$$p(\mathcal{G} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \mathcal{G}) p(\mathcal{G})}{\sum_{\mathcal{G}' \in \mathbb{G}_d} p(\mathcal{D} \mid \mathcal{G}') p(\mathcal{G}')}, \tag{2}$$

where $p(\mathcal{G})$ is the prior over graph structures and $p(\mathcal{D} \mid \mathcal{G})$ is the marginal likelihood of the data given the graph $\mathcal{G}$.

The core challenge for Bayesian causal structure learning is the combinatorial explosion of the graph space. For d variables, the number of possible DAGs is super-exponential in d. Our work is precisely aimed at addressing this scalability bottleneck in high-dimensional settings.

[1] Source code: `https://anonymous.4open.science/r/HiBCD-475B`

2.2 Related works

Bayesian Causal Discovery (BCD). Traditional causal discovery methods typically rely on conditional independence tests or graph search to identify a single causal graph that best fits the observed data. However, such methods often require a large number of observations and suffer from identifiability issues. In contrast, Bayesian Causal Discovery methods model the uncertainty in causal structure explicitly by placing a prior over graphs and inferring a posterior distribution over possible DAGs that explain the data Dhir et al. (2025), thereby mitigating these challenges. Early approaches relied on MCMC sampling, which often struggles with efficiency and scalability. More recent methods leverage variational inference (VI) to improve efficiency. For instance, DiBS Lorch et al. (2021) and BayesDAG Annadani et al. (2023) use sophisticated VI techniques to handle expressive functional models, while amortized approaches like AVICI Lorch et al. (2022) and BCNP Dhir et al. (2025) train neural networks to generalize inference across datasets. However, the computational cost of these methods scales poorly with the number of variables. In general, BCD methods can only handle fewer than 100 variables Viinikka et al. (2020).

3 HiBCD

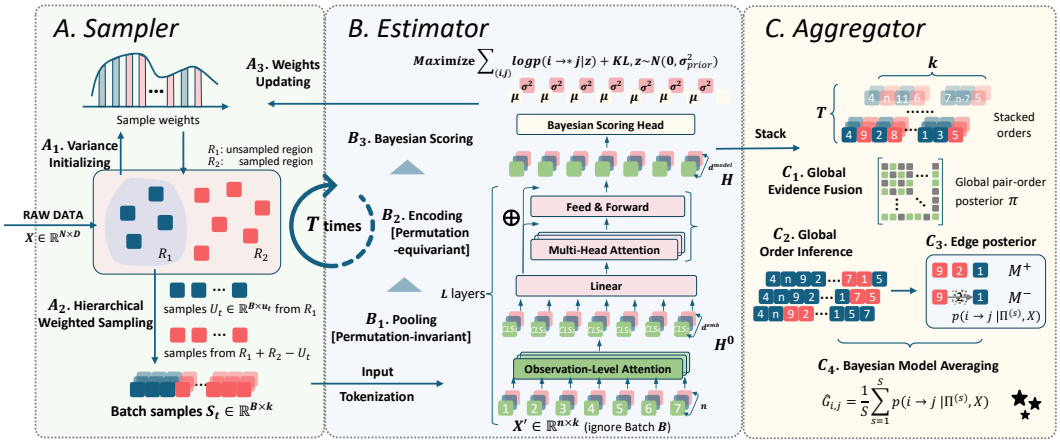


Figure 1: High-level design of HiBCD.

3.1 High-level architecture

Figure 1 shows the high-level architecture of HiBCD, a divide-and-conquer framework for Bayesian causal discovery in high-dimensional settings. Specifically, the framework operates through a pipeline of three core modules. First, a hierarchical weighted **Sampler** partitions the high-dimensional variable set into a series of smaller, computationally tractable subsets. Next, the **Estimator** utilizes a pre-trained meta-model for amortized variational inference to rapidly infer ancestral probabilities and derive a local causal ordering for each subset. Finally, the **Aggregator** synthesizes evidence from all local posterior distributions via Bayesian model averaging to construct a coherent global causal graph.

3.2 Sampler

To make the problem computationally tractable, we avoid operating on the full set of d variables at once. Instead, we iteratively sample smaller subsets of variables of size $k \ll d$, and the total process is repeated for T times. There are two main challenges to build such sampler: (i) we must ensure that the sampling process covers all variables; (ii) we must focus on more causally significant variables over time. To achieve this, we design a hierarchical weighted sampling scheme.

Dual-Pool Hierarchical Sampling. This strategy is designed to overcome challenge (i). Let $V = \{1, \ldots, d\}$ be the node index set corresponding to variables $\mathbf{X}$. For each sampling round $t \in \{1, \ldots, T\}$, let S_t be the set of k variables to be sampled. We define the pool of uncovered variables as $U_t = V \setminus \bigcup_{\ell<t} S_\ell$, and to ensure full coverage within the remaining $r_t = T - t + 1$ rounds, we must sample at least

$$\Delta_t = \max\left(0, |U_t| - k(r_t - 1)\right) \tag{3}$$

variables directly from U_t. The remaining $k - \Delta_t$ variables are then sampled from V, excluding those already selected in this round. Provided $Tk \geq d$, this strategy maintains $|U_t| \leq r_t k$, so all variables are included in at least one subset. For efficiency, both sampling steps are implemented using weighted reservoir sampling Efraimidis & Spirakis (2006).

Adaptive Weighting Mechanism. To overcome challenge (ii), we maintain a dynamically updated weight vector $\mathbf{w} \in \mathbb{R}^d$, where $w_i^{(t)}$ reflects the inferred importance of variable X_i after round t. We initialize $\mathbf{w}^{(0)}$ using the observed variances, assigning higher initial importance weights to variables with lower variance. The reason is, variables with lower observational variance are more likely to be exogenous causes, as their values are less influenced by other variables in the system Reisach et al. (2021; 2023). The initial weight $w_i^{(0)}$ for variable X_i is computed as:

$$w_i^{(0)} = 1 - \epsilon_{\text{samp}} \left(\frac{\mathrm{Var}(X_i) - \min_j \mathrm{Var}(X_j)}{\max_j \mathrm{Var}(X_j) - \min_j \mathrm{Var}(X_j) + \delta} - 0.5 \right), \tag{4}$$

where $\mathrm{Var}(X_i)$ is the sample variance of variable X_i, ϵ_{samp} controls the spread of the initial sampling weights, and δ is a small constant for numerical stability. After a local causal ordering is estimated for a subset in round t (detailed in § 3.3), we update the weights of the involved variables. The insight is that variables consistently appearing early in local causal orderings are more likely to be root or near-root nodes in the global graph. Such variables are crucial for explaining others and should be prioritized in subsequent sampling rounds. Let $\mathrm{rank}_t(j) \in \{1, \ldots, k\}$ be the rank of variable X_j in the local ordering estimated for its subset. We first compute a score $s_j^{(t)}$:

$$s_j^{(t)} = \frac{k - \mathrm{rank}_t(j)}{k - 1} - 0.5 \tag{5}$$

and then update the weight via an exponential moving average:

$$w_j^{(t)} = (1 - \eta) w_j^{(t-1)} + \eta s_j^{(t)}, \qquad j \in S_t, \tag{6}$$

where η is the update rate; weights outside S_t remain unchanged.

3.3 Estimator

Given a sampled subset S_t, the Estimator infers local ancestral relations and derives a causal ordering from them. We learn ancestral relations rather than direct edges because paths through variables outside S_t can induce dependencies among the observed variables. The meta-model consists of a symmetry-aware encoder and a pairwise Bayesian scoring head, allowing amortized inference on each subset. It is designed to be permutation-invariant over observations and permutation-equivariant over variables Maron et al. (2018); Zaheer et al. (2017); Lee et al. (2019).

Symmetry-Aware Encoder. Let $\mathcal{D}_t = \mathcal{D}_{:,S_t} \in \mathbb{R}^{n \times k}$, with columns arranged by increasing node index. Within this subsection, i, j denote local column positions; dependence of the network outputs on t is implicit. For each variable, a shared two-layer MLP $\phi : \mathbb{R} \to \mathbb{R}^{d_{\text{emb}}}$ maps its scalar observations $x_{a,j} = (\mathcal{D}_t)_{a,j}$, $a = 1, \ldots, n$, to embeddings $e_{a,j} = \phi(x_{a,j})$. The sequence $[\mathrm{CLS}, e_{1,j}, \ldots, e_{n,j}]$ passes through L_{obs} self-attention blocks without positional encodings. The normalized CLS output gives $h_j^{(0)} \in \mathbb{R}^{d_{\text{emb}}}$, invariant to the order of observations.

The initial variable representations form $H^{(0)} \in \mathbb{R}^{k \times d_{\text{emb}}}$. A stack of L_{var} self-attention blocks, also without positional encodings, processes these representations as a set. A final normalization and linear projection yield $H \in \mathbb{R}^{k \times d_{\text{model}}}$, whose rows h_i encode each variable and its relationships within the subset. This variable-level operation is permutation-equivariant.

Pairwise Bayesian Scoring Head. For each ordered pair $i \neq j$, we construct

$$F_{ij} = [h_i \parallel h_j \parallel (h_i - h_j) \parallel (h_i \odot h_j)], \tag{7}$$

where $\|$ denotes concatenation and $\odot$ denotes element-wise multiplication. The difference and product terms encode directional and symmetric interactions, respectively. An MLP with LayerNorm Ba et al. (2016) and GELU activations Hendrycks (2016) processes F_{ij}. Two linear output heads parameterize a factorized Gaussian posterior over latent ancestral scores $Z = (z_{ij})$:

$$q_\theta(Z \mid \mathcal{D}_t) = \prod_{i \neq j} \mathcal{N}(z_{ij} \mid \mu_{ij}, \sigma_{ij}^2). \tag{8}$$

Here θ denotes the network parameters. The variance head applies Softplus to its raw output and adds a small positive constant before returning $\log \sigma_{ij}^2$; the other head returns μ_{ij}. Self-pairs are excluded from training and local ranking.

Training with an Ancestral Variational Objective. For each training DAG $\mathcal{G}$, its transitive closure $\mathcal{T}(\mathcal{G})$ provides labels: $\mathcal{T}_{ij} = 1$ if $i \to^* j$, and 0 otherwise. We use a Probit-based variational surrogate for these labels, with a Gaussian prior $p(z_{ij}) = \mathcal{N}(0, \sigma_{\text{prior}}^2)$. Marginalizing the uncalibrated Probit probability under the Gaussian posterior gives $\mathbb{E}_{q_\theta}[\Phi(z_{ij}/\sigma_{\text{pair}})] = \Phi(\mu_{ij}/\sqrt{\sigma_{ij}^2 + \sigma_{\text{pair}}^2})$, where Φ is the standard normal CDF. During training, a learnable scale and bias calibrate the standardized score:

$$a_{ij} = s_{\text{cal}} \frac{\mu_{ij}}{\sqrt{\sigma_{ij}^2 + \sigma_{\text{pair}}^2}} + b_{\text{cal}}, \qquad \widetilde{p}_{ij} = \Phi(a_{ij}). \tag{9}$$

The positive and negative label probabilities are $\widetilde{p}_{ij}$ and $1 - \widetilde{p}_{ij} = \Phi(-a_{ij})$, respectively.

We subsample positive ancestral pairs and an equal number of negative pairs when available. Let $\mathcal{B}_+$ and $\mathcal{B}_-$ denote these pair sets pooled across a minibatch, and let $\mathcal{B} = \mathcal{B}_+ \cup \mathcal{B}_-$. Omitting dataset indices for readability, the minimized loss is

$$\begin{aligned} \mathcal{L} = & -\frac{1}{|\mathcal{B}_+|} \sum_{(i,j) \in \mathcal{B}_+} \log \Phi(a_{ij}) - \frac{1}{|\mathcal{B}_-|} \sum_{(i,j) \in \mathcal{B}_-} \log \Phi(-a_{ij}) \\ & + \frac{\beta_{\text{KL}}}{|\mathcal{B}|} \sum_{(i,j) \in \mathcal{B}} D_{\text{KL}}(q_\theta(z_{ij} \mid \mathcal{D}_t) \,\|\, p(z_{ij}))\,. \end{aligned} \tag{10}$$

Empty-set terms are taken as zero. The KL term has a closed-form Gaussian expression. This objective uses log marginal label probabilities rather than an expected log-likelihood, so it is a variational surrogate, not an ELBO for the observation matrix. It requires no Monte Carlo draws of the latent scores.

Local Order Inference. At inference, we use the final Probit noise level without the training calibration:

$$(P_{\text{anc}}^{(t)})_{ij} = \Phi\left(\frac{\mu_{ij}}{\sqrt{\sigma_{ij}^2 + \sigma_{\text{pair}}^2}} \right), \qquad i \neq j. \tag{11}$$

These are ancestral probabilities, which need not sum to one for opposite directions. Sorting variables by descending row-minus-column sums, $\sum_{j \neq i} [(P_{\text{anc}}^{(t)})_{ij} - (P_{\text{anc}}^{(t)})_{ji}]$, yields the local order used to update the Sampler. The probability matrix is passed to the Aggregator. Architecture dimensions and training hyperparameters are given in Appendix A.2.

3.4 AGGREGATOR

The local ancestral probabilities $\{P_{\text{anc}}^{(t)}\}_{t=1}^T$ leave two challenges for global reconstruction. First, evidence from different subsets is incomplete and may disagree. Second, ancestry can reflect indirect paths, so it does not by itself identify direct edges. The Aggregator addresses these challenges by combining local ordering evidence with Bayesian conditional models, then averaging edge probabilities across plausible global orders.

From local evidence to global orders. We first align each local matrix with the full variable set, writing $\widehat{P}^{(t)}_{\mathrm{anc}}$ with zeros outside $S_t \times S_t$. For each observed pair i, j, we average the available predictions:

$$M_{ij} = \frac{1}{N_{ij}} \sum_{t=1}^{T} (\widehat{P}^{(t)}_{\mathrm{anc}})_{ij}, \qquad N_{ij} = \sum_{t=1}^{T} \mathbb{I}(i, j \in S_t). \tag{12}$$

For unobserved pairs, we set $M_{ij} = 1/2$. To determine which direction the evidence favors, we compare opposing predictions through $\ell_{ij} = \mathrm{logit}(M_{ij}) - \mathrm{logit}(M_{ji})$. The corresponding pair-order probability π_{ij} expresses support for i preceding j, with $\pi_{ij} + \pi_{ji} = 1$. Combining these preferences gives

$$q_{\mathrm{ord}}(\Pi \mid \mathcal{D}) \propto \prod_{\Pi(i)<\Pi(j)} \pi_{ij}, \qquad \Pi \in \mathfrak{S}_d, \tag{13}$$

where $\Pi(i)$ is the rank of node i. Orders receive greater support when their pairwise relations agree with the local evidence. This distribution supplies the ordering evidence for global inference.

Bayesian inference of direct connections. An order restricts possible parents of j to $\mathrm{Pred}_\Pi(j) = \{i : \Pi(i) < \Pi(j)\}$. For a predictor set $\mathcal{S}_j \subseteq \mathrm{Pred}_\Pi(j)$, $\mathbf{\Phi}_{\mathcal{S}_j}$ encodes the observed predictors. We use a Gaussian observation model with unknown regression coefficients $\boldsymbol{\beta}_j$ and noise variance ν_j. A conjugate normal–inverse-gamma prior on $\vartheta_j = (\boldsymbol{\beta}_j, \nu_j)$ represents uncertainty in both quantities (Murphy, 2007, §6). Bayes' rule updates this prior using the observations, giving, with centering implicit,

$$\begin{aligned} \mathbf{x}_j \mid \mathbf{\Phi}_{\mathcal{S}_j}, \vartheta_j &\sim \mathcal{N}(\mathbf{\Phi}_{\mathcal{S}_j} \boldsymbol{\beta}_j, \nu_j I_n), \\ p(\vartheta_j \mid \mathbf{x}_j, \mathbf{\Phi}_{\mathcal{S}_j}) &= \frac{p(\mathbf{x}_j \mid \mathbf{\Phi}_{\mathcal{S}_j}, \vartheta_j)\, p(\vartheta_j)}{\int p(\mathbf{x}_j \mid \mathbf{\Phi}_{\mathcal{S}_j}, \vartheta_j)\, p(\vartheta_j)\, d\vartheta_j}. \end{aligned} \tag{14}$$

The posterior quantifies which coefficient and noise values the data support. Conjugacy keeps this distribution in the same family, making parameter integration analytic. The denominator is the marginal likelihood, which evaluates the model while accounting for parameter uncertainty. For a candidate edge, let $\mathcal{E}_{ij}$ denote the event that i contributes to predicting j after accounting for the other predictors. Its posterior probability $q_{\mathrm{H}}(\mathcal{E}_{ij} \mid \Pi, \mathcal{D})$ provides the edge support. These events induce the conditional graph distribution $q_{\mathrm{H}}(\mathcal{G} \mid \Pi, \mathcal{D})$, supported on structures consistent with Π.

Bayesian model averaging and graph recovery. Different orders admit different parents, so edge evidence must be combined across these alternatives. With $q_{\mathrm{H}}(\Pi \mid \mathcal{D})$ denoting the order distribution represented by global inference, Bayesian model averaging gives

$$q_{\mathrm{H}}(\mathcal{G} \mid \mathcal{D}) = \sum_{\Pi \in \mathfrak{S}_d} q_{\mathrm{H}}(\mathcal{G} \mid \Pi, \mathcal{D}) q_{\mathrm{H}}(\Pi \mid \mathcal{D}). \tag{15}$$

The parameter integrals above are analytic, but enumerating all $d!$ global orders is infeasible. We therefore use Monte Carlo inference to retain R orders $\{\Pi^{(r)}\}_{r=1}^{R}$ and associated conditional probabilities $s_{ij}^{(r)}$ for the edge events $\mathcal{E}_{ij}$. Their average, $\overline{G}_{ij} = R^{-1} \sum_{r=1}^{R} s_{ij}^{(r)}$, integrates edge evidence across the retained states.

Even when individual structures are acyclic, their averaged edge probabilities can support conflicting directions. Sorting nodes by ascending mean rank $\overline{r}_i = R^{-1} \sum_{r=1}^{R} \Pi^{(r)}(i)$ gives a consensus order $\Pi^\star$. We retain only connections consistent with this order:

$$\widehat{G}_{ij} = \overline{G}_{ij}\, \mathbb{I}[\Pi^\star(i) < \Pi^\star(j)], \qquad \widehat{A}_{ij} = \mathbb{I}\Big[\widehat{G}_{ij} > \tau_{\mathrm{edge}}\Big]. \tag{16}$$

Here $\tau_{\text{edge}} \in [0, 1]$ is the edge-probability threshold. Both matrices have zero diagonals, and the shared order guarantees that $\widehat{A}$ is acyclic. Appendix A.2 gives the conditional models and inference settings.

4 Evaluation

4.1 Experiment setup

Implementation. HiBCD is implemented in Python 3.10 with PyTorch PyTorch (2025) and evaluated on a 56-core CPU server. Implementation details and experiment configurations are provided in Appendix A.

Datasets. We evaluate HiBCD on a large synthetic corpus of linear and nonlinear Additive Noise Models (ANMs) Shimizu et al. (2006); Hoyer et al. (2008) and three real-world Bnlearn networks, MUNIN, ANDES, and PATHFINDER Scutari; 2010). Synthetic data generation and experiment settings are detailed in Appendix A, and real-world dataset descriptions are provided in Appendix D.

Baselines. We compare against two amortized Bayesian causal discovery methods, BCNP Dhir et al. (2025) and AVICI Lorch et al. (2022), and the optimization-based DrBO Duong et al. (2025). Baseline training, inference, and hyperparameter settings are summarized in Appendix B.

Metrics. We report graph recovery metrics (SHD, TPR, FDR, FPR, and F1), causal order quality metrics (Reverse Pairs and Kendall's Tau), and total inference time. In Tables 1 and 2, we report mean $\pm$ sample standard deviation at each graph size: five linear and three nonlinear runs for HiBCD, and five runs per setting for AVICI. Metric definitions are given in Appendix A.5.

4.2 Results on synthetic datasets

Tables 1 and 2 report the structure learning performance of all competing methods on synthetic linear and nonlinear datasets with varying graph sizes. Overall, HiBCD achieves the lowest SHD and highest F1 from $d = 100$ onward in linear settings and across all four evaluated graph sizes in nonlinear settings.

Table 1: Performance on synthetic linear ER graphs.

Nodes (d)	Method	SHD↓	TPR↑	FDR↓	F1↑
50	BCNP	242.925 ± 12.616	0.185 ± 0.033	0.813 ± 0.033	0.186 ± 0.033
	AVICI	121.800 ± 13.828	0.165 ± 0.061	0.143 ± 0.052	0.272 ± 0.092
	DrBO	23.400 ± 23.050	0.959 ± 0.029	0.103 ± 0.086	0.926 ± 0.060
	HiBCD (Ours)	40.400 ± 3.286	0.844 ± 0.023	0.136 ± 0.027	0.854 ± 0.013
100	BCNP	505.484 ± 17.828	0.148 ± 0.021	0.849 ± 0.021	0.150 ± 0.021
	AVICI	284.000 ± 25.179	0.089 ± 0.024	0.158 ± 0.112	0.159 ± 0.038
	DrBO	296.600 ± 86.668	0.742 ± 0.076	0.480 ± 0.113	0.609 ± 0.098
	HiBCD (Ours)	119.400 ± 17.799	0.828 ± 0.034	0.211 ± 0.026	0.808 ± 0.029
200	BCNP	1085.755 ± 26.970	0.109 ± 0.013	0.894 ± 0.012	0.108 ± 0.012
	AVICI	588.600 ± 14.943	0.032 ± 0.007	0.277 ± 0.133	0.061 ± 0.013
	DrBO	1186.400 ± 119.222	0.537 ± 0.042	0.737 ± 0.030	0.353 ± 0.034
	HiBCD (Ours)	227.600 ± 36.322	0.820 ± 0.024	0.206 ± 0.025	0.807 ± 0.023
300	BCNP	1680.813 ± 31.684	0.072 ± 0.008	0.929 ± 0.008	0.071 ± 0.008
	AVICI	899.400 ± 30.443	0.012 ± 0.007	0.353 ± 0.164	0.024 ± 0.013
	DrBO	—	—	—	—
	HiBCD (Ours)	417.800 ± 97.873	0.816 ± 0.027	0.261 ± 0.062	0.775 ± 0.046

—: No result obtained within a practical runtime for this setting.

Linear Setting. HiBCD maintains strong recovery accuracy as graph size increases, achieving the lowest SHD and highest F1 from $d = 100$ onward (Table 1). At $d = 200$, its SHD of 227.6 is 61.3% lower than AVICI's 588.6, while its F1 of 0.807 exceeds DrBO's 0.353 by 128.6%. This gain combines higher recall with fewer false discoveries: HiBCD achieves a TPR of 0.820 and an FDR of

0.206, compared with 0.537 and 0.737 for DrBO. At $d = 300$, HiBCD retains an F1 of 0.775 and a TPR of 0.816, recovering over 80% of true edges even as graph size grows sixfold from $d = 50$. BCNP and AVICI recover only 7.2% and 1.2% of true edges at this scale, highlighting HiBCD's advantage in preserving causal structure in high dimensions.

Non-Linear Setting. HiBCD achieves the lowest SHD and highest TPR and F1 across all four graph sizes (Table 2). At $d = 100$, HiBCD attains a TPR of 0.669, 4.3× AVICI's 0.157, with an FDR increase of only 3.4 percentage points (0.110 to 0.144). It thus substantially expands causal-edge discovery while retaining 85.6% precision. At $d = 200$, HiBCD reduces SHD by 33.7% relative to AVICI (550.0 to 364.4) and improves F1 from 0.172 to 0.634. At $d = 300$, it still achieves an F1 of 0.503, 4.7× AVICI's 0.106, and recovers 40.9% of true edges. These consistent gains show that HiBCD recovers substantially more of the underlying nonlinear structure across the evaluated scales.

Table 2: Performance on synthetic nonlinear ER graphs.

Nodes (d)	Method	SHD↓	TPR↑	FDR↓	F1↑
50	BCNP	215.301 ± 13.539	0.266 ± 0.041	0.724 ± 0.042	0.271 ± 0.041
	AVICI	97.400 ± 13.849	0.331 ± 0.060	0.059 ± 0.042	0.487 ± 0.070
	DrBO	285.000 ± 40.577	0.407 ± 0.108	0.778 ± 0.062	0.288 ± 0.078
	HiBCD (Ours)	58.800 ± 12.538	0.724 ± 0.061	0.133 ± 0.042	0.788 ± 0.043
100	BCNP	471.048 ± 18.598	0.206 ± 0.025	0.790 ± 0.025	0.208 ± 0.025
	AVICI	263.400 ± 23.352	0.157 ± 0.022	0.110 ± 0.036	0.267 ± 0.031
	DrBO	1111.400 ± 23.212	0.237 ± 0.031	0.924 ± 0.007	0.115 ± 0.012
	HiBCD (Ours)	132.000 ± 21.272	0.669 ± 0.036	0.144 ± 0.052	0.750 ± 0.033
200	BCNP	1071.672 ± 25.452	0.106 ± 0.013	0.894 ± 0.013	0.106 ± 0.013
	AVICI	550.000 ± 17.507	0.095 ± 0.012	0.109 ± 0.074	0.172 ± 0.020
	DrBO	3186.600 ± 189.447	0.192 ± 0.024	0.959 ± 0.003	0.067 ± 0.006
	HiBCD (Ours)	364.400 ± 49.702	0.541 ± 0.057	0.235 ± 0.049	0.634 ± 0.056
300	BCNP	1613.356 ± 32.214	0.087 ± 0.010	0.910 ± 0.010	0.088 ± 0.010
	AVICI	862.800 ± 32.368	0.057 ± 0.007	0.144 ± 0.041	0.106 ± 0.012
	DrBO	—	—	—	—
	HiBCD (Ours)	707.200 ± 43.574	0.409 ± 0.019	0.339 ± 0.069	0.503 ± 0.020

—: No result obtained within a practical runtime for this setting.

Computational Efficiency. We further evaluate computational efficiency by analyzing the runtime breakdown across varying graph sizes, as shown in Figure 2. HiBCD completes the causal discovery task for $d = 100$ in just 8.6 seconds and for $d = 200$ in 57.6 seconds. The average inference time per sample decreases significantly as the scale increases, dropping to a negligible $0.0042s$ at $d = 200$. This trend confirms the effectiveness of our amortized inference design, where local ordering inference becomes computationally inexpensive via batching, serving as the cornerstone of HiBCD's high-dimensional scalability.

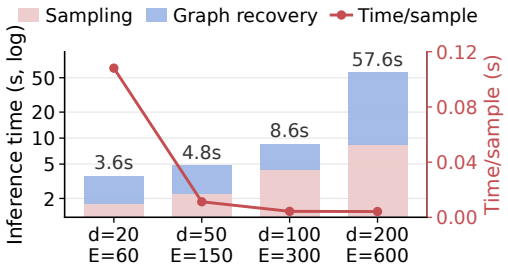


Figure 2: Efficiency of HiBCD.

4.3 Deep Dive

Ways for Encoding Observations. We compare three ways to summarize the i.i.d. observations of each variable before node-level relational reasoning: Direct Linear Mapping, Statistical Pooling, and Attention Pooling with a $[CLS]$ token; detailed descriptions are deferred to Appendix C.1. As shown in Figure 3, Direct Linear Mapping converges quickly but overfits to sample permutations, while Statistical Pooling loses too much distributional information. The Attention Pooling design achieves the best generalization, steadily reaching a Kendall's Tau of ≈ 0.8 on large-scale datasets.

Impact of Sampling Strategy. To validate Module 1, Figure 4 reports coverage rate (solid lines) and reverse-pair errors (dashed lines) as T increases. The Dual-Pool sampler reaches 100% coverage even with a small budget (e.g., $T = 30$), while Random sampling needs many more rounds. Adaptive weighting also yields fewer reverse pairs; removing variance-based initialization or rank-based updates slows convergence, showing that coverage control and feedback-driven reweighting are both important for stitching reliable global orders.

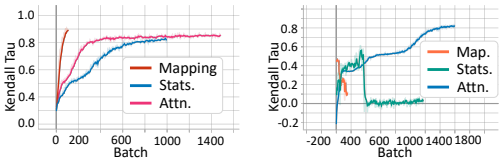


Figure 3: Kendall tau of encoding designs.

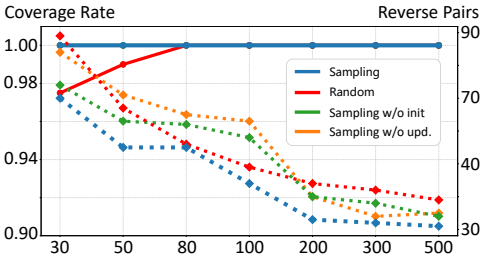


Figure 4: Impact of sampling strategy.

Impact of Sampling Size k. We further vary the local subset size k in Appendix C.2. The results show that the default $k = 20$ provides the most stable SHD profile across linear and nonlinear settings. Across both linear and nonlinear settings, suggesting a moderate subset size better balances local estimator reliability and global coverage.

4.4 Results on Real-World Benchmarks

The radar charts in Figure 5 present our evaluation results on the MUNIN and PATHFINDER real-world benchmarks. HiBCD demonstrates the most robust and balanced performance across all five metrics (F1, Precision, Recall, Accuracy, and Specificity), where Accuracy is defined as $(1-\text{SHD}/d^2)$ and Specificity as $(1-\text{FPR})$. While baselines perform well in specific metrics, HiBCD exceeds them across the board with superior overall balance. Further real-world evaluations and visualizations are detailed in Appendix D.



(a) MUNIN. (b) PATHFINDER.

Figure 5: Results on real-world benchmarks.

5 Conclusion

We introduced HiBCD, a scalable divide-and-conquer framework for Bayesian causal discovery in high-dimensional settings. By combining hierarchical weighted sampling, amortized inference of local causal orderings, and global aggregation, HiBCD reconstructs large causal graphs from local evidence. Experiments on synthetic and real-world benchmarks demonstrate its effectiveness and scalability, with the lowest SHD and highest F1 in almost all evaluated linear and nonlinear settings. These results support the use of reusable local inference and global structural aggregation for large-scale Bayesian causal discovery.

AI USE STATEMENT

We used AI tools to assist with code development by delegating routine tasks, such as environment setup and figure plotting, as well as to support mathematical calculations. All AI-generated outputs were carefully reviewed and verified by the authors. AI tools were also used to help identify potentially relevant literature, while the relevance and appropriateness of all cited references were manually assessed by the authors. In addition, AI was used to improve the language and presentation of the manuscript. Specifically, the initial draft was written entirely by the authors, after which AI tools were used for language refinement and polishing. The authors take full responsibility for the accuracy, integrity, and final content of this work.

REPRODUCIBILITY STATEMENT

Our code is available at `https://anonymous.4open.science/r/HiBCD-475B`. To support reproducibility, Appendix A provides data-generation procedures, model and experiment configurations, and evaluation metrics, while Appendix B documents baseline implementations and settings.

REFERENCES

Stig Kjær Andersen, Steen Andreassen, B Falck, Finn V Jensen, U Kjærulff, A Rosenfalck, AR Sørensen, and M Woldbye. Munin: an expert emg assistant. In *Proceedings of the III International Conference on Computerised and Single Fiber EMG, 5-8 June 1988, Cyprus*, pp. 52, 1988.

Yashas Annadani, Nick Pawlowski, Joel Jennings, Stefan Bauer, Cheng Zhang, and Wenbo Gong. Bayesdag: Gradient-based posterior inference for causal discovery. *Advances in Neural Information Processing Systems*, 36:1738–1763, 2023.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Cristina Conati, Abigail S Gertner, Kurt VanLehn, and Marek J Druzdzel. On-line student modeling for coached problem solving using bayesian networks. In *User Modeling: Proceedings of the Sixth International Conference UM97 Chia Laguna, Sardinia, Italy June 2–5 1997*, pp. 231–242. Springer, 1997.

Joscha Cüppers, Sascha Xu, Ahmed Musa, and Jilles Vreeken. Causal discovery from event sequences by local cause-effect attribution. In *NeurIPS*, 2024.

Anish Dhir, Matthew Ashman, James Requeima, and Mark van der Wilk. A meta-learning approach to bayesian causal discovery. In *ICLR*. OpenReview.net, 2025.

Ruomeng Ding, Chaoyun Zhang, Lu Wang, Yong Xu, Minghua Ma, Xiaomin Wu, Meng Zhang, Qingjun Chen, Xin Gao, Xuedong Gao, et al. Tracediag: Adaptive, interpretable, and efficient root cause analysis on large-scale microservice systems. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 1762–1773, 2023.

Shuyu Dong, Michele Sebag, Kento Uemura, Akito Fujii, Shuang Chang, Yusuke Koyanagi, and Koji Maruhashi. Dcilp: A distributed approach for large-scale causal structure learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16345–16353, 2025.

Bao Duong, Sunil Gupta, and Thin Nguyen. Causal discovery via bayesian optimization. *arXiv preprint arXiv:2501.14997*, 2025.

Pavlos S Efraimidis and Paul G Spirakis. Weighted random sampling with a reservoir. *Information processing letters*, 97(5):181–185, 2006.

Yu Gan, Guiyang Liu, Xin Zhang, Qi Zhou, Jiesheng Wu, and Jiangwei Jiang. Sleuth: A trace-based root cause analysis system for large-scale microservices with graph neural networks. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 4*, pp. 324–337, 2023.

Matthew F Glasser, Timothy S Coalson, Emma C Robinson, Carl D Hacker, John Harwell, Essa Yacoub, Kamil Ugurbil, Jesper Andersson, Christian F Beckmann, Mark Jenkinson, et al. A multi-modal parcellation of human cerebral cortex. *Nature*, 536(7615):171–178, 2016.

Alex Greenfield, Aviv Madar, Harry Ostrer, and Richard Bonneau. Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PloS one*, 5(10): e13397, 2010.

Jiaying Gu and Qing Zhou. Learning big gaussian bayesian networks: Partition, estimation and fusion. *J. Mach. Learn. Res.*, 21:158:1–158:31, 2020.

David Earl Heckerman, Eric J Horvitz, and Bharat N Nathwani. Toward normative expert systems: Part i the pathfinder project. *Methods of information in medicine*, 31(02):90–105, 1992.

D Hendrycks. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 21, 2008.

Dominik Janzing and Bernhard Schölkopf. Causal inference using the algorithmic markov condition. *IEEE Trans. Inf. Theory*, 56(10):5168–5194, 2010.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Lars Lorch, Jonas Rothfuss, Bernhard Schölkopf, and Andreas Krause. Dibs: Differentiable bayesian structure learning. *Advances in Neural Information Processing Systems*, 34:24111–24123, 2021.

Lars Lorch, Scott Sussex, Jonas Rothfuss, Andreas Krause, and Bernhard Schölkopf. Amortized inference for causal structure learning. In *NeurIPS*, 2022.

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. *arXiv preprint arXiv:1812.09902*, 2018.

Kevin P. Murphy. Conjugate Bayesian analysis of the Gaussian distribution, 2007. URL `https://www.cs.ubc.ca/~murphyk/Papers/bayesGauss.pdf`.

Judea Pearl. Causal inference in statistics: An overview. 2009.

PyTorch. A Deep Learning Framework. `https://pytorch.org/`, 2025. Accessed Apr 2025.

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

Alexander G. Reisach, Myriam Tami, Christof Seiler, Antoine Chambaz, and Sebastian Weichwald. A scale-invariant sorting criterion to find a causal order in additive noise models. In *NeurIPS*, 2023.

Patrik Reizinger, Siyuan Guo, Ferenc Huszár, Bernhard Schölkopf, and Wieland Brendel. Identifiable exchangeable mechanisms for causal structure and representation learning. In *ICLR*. OpenReview.net, 2025.

Marco Scutari. Very large networks (100–1000 nodes) — bayesian network repository. `https://www.bnlearn.com/bnrepository/discrete-verylarge.html`. Accessed: 2026-01-29.

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

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72, 1991.

Jussi Viinikka, Antti Hyttinen, Johan Pensar, and Mikko Koivisto. Towards scalable bayesian learning of causal dags. *Advances in Neural Information Processing Systems*, 33:6584–6594, 2020.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

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

A HiBCD IMPLEMENTATION DETAILS

A.1 DATA GENERATION

Overview. We evaluate our method on synthetic datasets generated from randomly sampled Erdős–Rényi (ER) Directed Acyclic Graphs (DAGs). Both linear and nonlinear main experiments use graph sizes $d \in \{50, 100, 200, 300\}$. The nominal edge count is $3d$; realized HiBCD edge counts vary across generated graphs. The data generation process follows two distinct Structural Equation Models (SEMs):

Linear SEM. In the linear setting, the value of each node X_j is determined by a linear combination of its parents plus additive noise: $X_j = \sum_{i \in \mathrm{PA}_{\mathcal{G}}(j)} W_{ij} X_i + \varepsilon_j$. The edge weights W_{ij} are sampled uniformly from the range $[0.5, 2.0]$ (and $[-2.0, -0.5]$ to include negative associations). The noise terms ε_j follow a standard Gaussian distribution $\mathcal{N}(0, 1)$.

Nonlinear SEM. In the nonlinear setting, we model the dependencies using Multi-Layer Perceptrons (MLP). Each variable is generated as $X_j = f_j(\mathbf{X}_{\mathrm{PA}_{\mathcal{G}}(j)}) + \varepsilon_j$, where f_j is a randomly initialized MLP with one hidden layer and sigmoid activation functions. The noise ε_j remains Gaussian. This setting is designed to test the model's capability in capturing complex functional relationships.

Number of Observations. All linear and nonlinear main experiments use $n = 400$ observations. The full configurations are summarized in Table 3.

A.2 MAIN EXPERIMENT CONFIGURATIONS

Estimator Architecture. The estimator uses attention pooling over observations followed by self-attention over variables. A shared MLP with dimensions $1 \to 128 \to 128$ and a SiLU activation embeds each scalar observation. For each variable, a learnable CLS token is prepended to its observation embeddings. The sequence passes through $L_{\mathrm{obs}} = 2$ attention blocks without positional encodings; each block uses four attention heads, residual connections, LayerNorm, and a GELU feed-forward network. The normalized CLS output is the variable embedding, with $d_{\mathrm{emb}} = 128$. The variable encoder then applies $L_{\mathrm{var}} = 4$ self-attention blocks with $n_{\mathrm{heads}} = 4$ and $d_{\mathrm{model}} = 512$, followed by LayerNorm and a linear projection. The pairwise head concatenates $h_i, h_j, h_i - h_j, h_i \odot h_j$ into a 2048-dimensional feature vector. LayerNorm and two GELU-activated hidden layers of width 1024 feed separate scalar mean and variance heads. The variance is made positive with Softplus, yielding the Gaussian parameters in Eq. (8).

Training Hyperparameters. The estimator is trained on $k = 20$ variables with $n = 400$ observations per dataset for 1000 epochs, using a batch size of 256. We use AdamW with a peak learning rate of 5×10^{-4} and weight decay 10^{-4}, with linear learning-rate warmup followed by cosine decay. For the ancestral objective in Eq. (10), the prior variance is $\sigma^2_{\mathrm{prior}} = 0.25$, and β_{KL} is warmed up from zero to 0.1. The Probit noise scale starts at $\sigma_{\mathrm{pair}} = 0.30$ and decreases linearly to 0.20 during the final third of training. Inference uses $\sigma_{\mathrm{pair}} = 0.20$ in Eq. (11).

Sampler and Aggregator Configurations. All synthetic main experiments use $k = 20$, initial sampling-weight spread $\epsilon_{\mathrm{samp}} = 0.30$, and update rate $\eta = 0.10$. Following Section 3.4, the Aggregator combines local ancestral probabilities into global ordering evidence and averages conditional edge probabilities over R retained orders. The averaged probabilities are projected onto a consensus order and thresholded at τ_{edge}, as in Eq. (16). Table 3 lists the observation count n, sampled-subset budget T, and retained-order count R for the main experiments.

A.3 ABLATION AND EFFICIENCY CONFIGURATIONS

Ablation Settings. All ablation studies are consistently evaluated on the representative scale of $d = 50, |E| = 150$, and $n = 1000$. The baseline sampled subgraphs T are set to 200 for linear models and 800 for nonlinear models. When isolating specific components (e.g., candidate pruning, parent-set MCMC, and varying order samples), we keep the remaining settings fixed within each ablation. These ablations retain their original configurations, separately from the main experiments in Table 3.

Table 3: Main synthetic experiment configurations. T is the number of sampled variable subsets, and R is the number of retained global orders used for averaging.

Setting	**Nodes** (d)	**Observations** (n)	**Subsets** (T)	**Orders** (R)
Linear	50	400	200	8
	100	400	1000	4
	200	400	2000	8
	300	400	4500	8
Nonlinear	50	400	200	128
	100	400	2000	128
	200	400	4000	128
	300	400	9000	128

Efficiency Settings. The efficiency evaluation retains the original per-order Aggregator configuration and estimator dimensions, with the number of order samples reduced to $R = 1$ across all settings. It does not use the nonlinear joint-chain budget in Table 3.

A.4 Real-world dataset configurations

Real-world benchmarks require specific tuning regarding parent scoring metrics and prior regularization due to domain-specific complexities. The configurations for MUNIN, ANDES, and PATHFINDER datasets are summarized in Table 4.

Table 4: Aggregator configurations for real-world datasets. T_{BF} represents the Bayes factor temperature, γ is the edge prior cap, and τ_{edge} is the final edge-probability threshold.

Dataset	**Nodes** (d)	**Subgraphs** (T)	T_{BF}	γ	τ_{edge}
MUNIN	186	2000	1.5	0.5	0.805
ANDES	223	2000	2.0	0.2	> 0.999
PATHFINDER	109	1000	1.5	0.5	0.327

A.5 Evaluation metrics

We evaluate recovered graphs against the ground-truth adjacency matrix using Structural Hamming Distance (SHD), True Positive Rate (TPR), False Discovery Rate (FDR), False Positive Rate (FPR), and F1-score. Let TP, FP, FN, and TN denote the number of true positives, false positives, false negatives, and true negatives, respectively. Then $\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, $\mathrm{FDR} = \mathrm{FP}/(\mathrm{TP}+\mathrm{FP})$, $\mathrm{FPR} = \mathrm{FP}/(\mathrm{FP}+\mathrm{TN})$, and F1 is the harmonic mean of precision and recall. SHD counts the minimum number of edge additions, deletions, and reversals required to transform the estimated graph into the ground truth.

For causal order evaluation, Reverse Pairs counts ground-truth directed edges that violate the estimated order, and Kendall's Tau measures rank agreement between the estimated order and a ground-truth topological order. Runtime is reported as total wall-clock inference time unless otherwise specified.

For the HiBCD results in Table 1, refinement is disabled. At each graph size, all metrics are summarized over the same five experimental runs using the mean and sample standard deviation. The nonlinear results in Table 2 summarize three experimental runs per graph size in the same way.

B Baseline implementation details

In this section, we provide detailed configurations and experimental protocols for the baseline methods: BCNP, DrBO, and AVICI.

B.1 BCNP CONFIGURATION AND TRAINING

B.1.1 MODEL ARCHITECTURE AND STRUCTURAL INTERFACE

The BCNP baseline employs a Transformer-based architecture consisting of $L = 4$ encoder and decoder layers with $n_{\text{heads}} = 8$ attention heads and an embedding depth of 1. The hidden dimensionality d_{model} and feed-forward dimensionality d_{ff} are specified in Table 5.

B.1.2 DATA GENERATION AND EXPERIMENTAL PROTOCOL

For linear data settings, the training process utilizes $100,000$ unique causal graphs, whereas non-linear settings utilize $200,000$ unique graphs. Each graph is associated with $1,000$ observational samples. The evaluation follows a protocol where one model instance per experimental setting is used for zero-shot inference on 25 independently generated test datasets. Each test set consists of $1,000$ samples, and the reported results are the arithmetic mean across these 25 runs.

B.1.3 OPTIMIZATION AND HYPERPARAMETER CONFIGURATION

The **Adam with Weight Decay optimizer** is used for a training duration of 10 epochs. The base learning rate is set to 1×10^{-4} with a weight decay coefficient of 1×10^{-5}. A linear learning rate warmup is applied during the first 10% of total iterations. Additionally, gradient norm clipping is implemented with a threshold of 1.0.

Table 5: Hyperparameter settings and data generation scale for BCNP experiments. All configurations use 1,000 testing samples.

Nodes	**Edges**	**Training Samples**		**Batch Size**	d_{model}	d_{ff}
		Linear	**Non-Linear**			
50	150	100K	200K	4	1024	2048
100	300	100K	200K	2	1024	2048
200	600	100K	200K	2	1024	2048
300	900	100K	200K	2	1024	2048

B.2 DRBO CONFIGURATIONS

B.2.1 MODEL ARCHITECTURE AND SCORING METHODOLOGY

For linear settings, the framework utilizes a linear regression model assuming equal noise variance. In non-linear scenarios, a Gaussian Process (GP) regressor is employed, utilizing an alpha value of 1.0 and a heteroscedastic noise assumption to allow for varying noise levels across nodes. The scoring is governed by the Bayesian Information Criterion (BIC). The underlying neural components, where applicable, consist of hidden layers with 64 units and a dropout rate of 0.1, optimized with a learning rate of 0.1. The search process maintains a replay buffer of 1,024 and performs 10 gradient steps per iteration.

B.2.2 EXPERIMENTAL PROTOCOL AND DATA GENERATION

The evaluation covers synthetic datasets generated via the Erdős-Rényi model across graph scales of 50, 100, 200, and 300 nodes, maintaining a constant average degree of 3. Each dataset consists of 500 observational samples with functional relationships defined as either linear or non-linear, supplemented by Gaussian noise. For settings with reported results, three independent runs are performed on unique datasets generated from scratch. The final reported performance metrics represent the arithmetic mean across these three independent repeats. At 300 nodes, no result was obtained within a practical runtime, as indicated in Tables 1 and 2.

B.2.3 HYPERPARAMETER CONFIGURATIONS AND OPTIMIZATION

The optimization and search parameters follow the requirements of the specific regressor and graph scale. For linear data, the search is conducted over 4,000 evaluations using a batch size of 32, with structural refinement performed via a linear pruning strategy. The non-linear (GP) configuration employs a search budget of 10,000 evaluations. For the reported nonlinear results, the batch size is 64 for graphs with 50 or 100 nodes, and 16 for graphs with 200 nodes to prevent memory errors. Final structures are refined through Conditional Independence Test (CIT) pruning.

B.3 AVICI CONFIGURATIONS

The AVICI baseline utilizes the pre-trained *scm-v0/checkpoint_0300000* model, which is built upon a Transformer-based architecture. The model is characterized by a hidden dimension of 128 and a feed-forward dimensionality of 512. The configuration includes 16 layers and 8 attention heads, with each head utilizing a key size of 32. The model was originally trained with diagonal masking to enforce structural priors, and it does not incorporate an explicit directed acyclic graph (DAG) penalty in its loss function. For each experimental configuration, the model executes inference on five independently generated datasets, each containing 500 observations. We retain the model's built-in standardization and predict edges with probability greater than 0.5, with diagonal entries set to zero. Results are reported as mean $\pm$ sample standard deviation over the five runs.

C MORE SETTINGS AND RESULTS

C.1 ENCODING DESIGNS FOR OBSERVATIONS

We evaluate three observation encoders for the meta-estimator. **Direct Linear Mapping** treats the n observations of each variable as a fixed-dimensional vector and maps it directly to the embedding space, i.e., $\mathbb{R}^n \to \mathbb{R}^{d_{\text{emb}}}$. This option is simple but ignores the permutation invariance of i.i.d. samples. **Statistical Pooling** first applies an MLP to each observation and then aggregates summary statistics such as the mean and standard deviation of the resulting embeddings. This design is permutation-invariant but can discard fine-grained distributional information. **Attention Pooling with a $[CLS]$ token** prepends a learnable summary token to the observation embeddings and applies self-attention without positional encoding; the $[CLS]$ output is used as the variable representation. This is the design adopted in our estimator, as it offers a flexible balance between permutation invariance and expressive distributional summarization.

C.2 SENSITIVITY TO SAMPLING SIZE k

Figure 6 reports the SHD obtained when varying the local subset size k in the sampler. We compare $k \in \{10, 20, 30\}$ on the three larger synthetic graph scales under both linear and nonlinear settings, while keeping the remaining evaluation protocol aligned with the main experiments. Overall, $k = 20$ gives the most stable behavior. A smaller subset size ($k = 10$) often lacks enough local context for reliable ordering and increases SHD, especially on large graphs. A larger subset size ($k = 30$) can improve several linear cases, but it is less consistent in nonlinear settings, where the local estimator and Aggregator face more complex functional dependencies. These results support using a moderate subset size as the default trade-off between local inference quality and global sampling coverage.

D REAL-WORLD DATASET DESCRIPTIONS AND RESULTS

D.1 DATASET DESCRIPTIONS

We evaluate the performance of our method on three standard real-world Bayesian networks commonly used in causal discovery tasks: MUNIN, ANDES, and PATHFINDER. These networks cover diverse domains including medicine and education, varying in size and structural complexity.

MUNIN. The MUNIN (Muscle and Nerve Inference Network) dataset originates from the field of neuromuscular medicine Andersen et al. (1988). It is designed as an expert electromyography (EMG) assistant to diagnose disorders in the peripheral nerve and muscle systems. The causal structure of

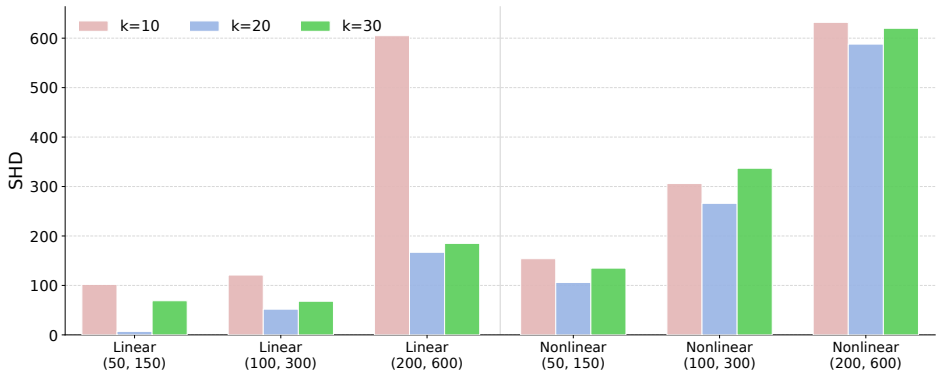


Figure 6: Impact of sampling size k on SHD.

the network is hierarchical, modeling the physiological mechanism where pathological states (e.g., nerve lesions) cause changes in anatomical structures, which in turn manifest as observable EMG signal features. In our experiments, we utilize a standard subset of the network containing 186 nodes and 273 edges. The structure is characterized by a "multi-to-many" mapping, where a single disorder may trigger multiple parameter abnormalities, and a specific EMG finding may result from various pathological causes, introducing complex V-structures and confounding factors.

ANDES. The ANDES network is a component of an Intelligent Tutoring System (ITS) for physics, developed for student modeling and cognitive assessment Conati et al. (1997). Its primary function is to infer a student's mastery of physics concepts based on their problem-solving actions. The causal logic flows from the student's knowledge of rules and axioms (Causes) to their specific derivation steps and interface actions (Effects). We employ a subset of this network consisting of 223 nodes and 338 edges. Structurally, ANDES differs significantly from medical diagnostic networks; it features deep causal chains representing multi-step reasoning and possesses a convergent structure, where multiple preconditions (e.g., a known rule and a specific fact) jointly determine a resultant action.

PATHFINDER. PATHFINDER is a normative expert system developed for hematopathology to assist in the diagnosis of lymph node diseases Heckerman et al. (1992). It models the causal dependencies between approximately 60 distinct diseases (including benign and malignant variations) and over 100 morphological features observed in tissue sections. The network we use comprises 109 nodes and 195 edges. While it has fewer nodes compared to ANDES, PATHFINDER is characterized by high parametric complexity due to the non-binary, multi-state nature of its variables. Its structure generally follows a "Disease $\rightarrow$ Symptom" flow but incorporates complex dependencies among the morphological features themselves, distinguishing it from simple Naive Bayes models.

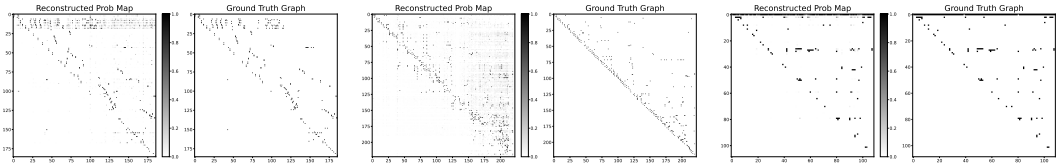
D.2 Results on ANDES

We further evaluated HiBCD on the ANDES dataset, which consists of 223 nodes and 338 edges. As summarized in the quantitative results, HiBCD achieves the best overall performance on ANDES with an F1-score of 0.4375, improving substantially over the strongest baseline AVICI (F1=0.1548), DrBO (F1=0.1199), and BCNP (F1=0.0171).

A key advantage of HiBCD in this setting is its improved balance between sensitivity and precision. HiBCD achieves a Recall of 0.4142 and a Precision of 0.4636, while AVICI is more conservative, with a much lower Recall of 0.1065. DrBO recovers more edges than AVICI but remains substantially less accurate, with an F1-score of 0.1199 and a higher SHD. BCNP fails to recover meaningful structure in this large-scale setting.

D.3 Visualization of causal discovery

Figure 7 visualizes the edge posterior probability matrices recovered by HiBCD and the ground-truth DAGs for MUNIN, ANDES, and PATHFINDER. Each entry (i, j) denotes the posterior probability that a directed edge $i \rightarrow j$ exists, with darker entries indicating higher probabilities and white entries indicating probabilities close to zero. Overall, the high-probability regions align well with the nonzero



(a) MUNIN (186 nodes) (b) ANDES (223 nodes) (c) PATHFINDER (109 nodes)

Figure 7: Visualization of recovered causal structures. Comparison of the adjacency matrices recovered by HiBCD against the ground truth across three real-world datasets. The visual alignment shows that HiBCD recovers meaningful sparse causal structure across networks of varying sizes.

entries of the ground-truth adjacency matrices, especially for MUNIN and PATHFINDER, consistent with their strong quantitative results. For the larger ANDES graph, HiBCD still captures a meaningful subset of the global topology and local dependencies, outperforming the baseline methods while remaining a challenging large-scale recovery setting.