Bootstrap-based Causal Structure Learning

Xianjie Guo
Hefei University of Technology
Hefei, Anhui, China
xianjieguo@mail.hfut.edu.cn

Yujie Wang
Hefei University of Technology
Hefei, Anhui, China
yujiewang@mail.hfut.edu.cn

Xiaoling Huang
Hefei University of Technology
Hefei, Anhui, China
hxl@chzu.edu.cn

Shuai Yang
Hefei University of Technology
Hefei, Anhui, China
yangs@mail.hfut.edu.cn

Kui Yu*
Hefei University of Technology
Hefei, Anhui, China
yukui@hfut.edu.cn

ABSTRACT

Learning a causal structure from observational data is crucial for data scientists. Recent advances in causal structure learning (CSL) have focused on local-to-global learning, since the local-to-global CSL can be scaled to high-dimensional data. The local-to-global CSL algorithms first learn the local skeletons, then construct the global skeleton, and finally orient edges. In practice, the performance of local-to-global CSL mainly depends on the accuracy of the global skeleton. However, in many real-world settings, owing to inevitable data quality issues (e.g. noise and small sample), existing local-to-global CSL methods often yield many *asymmetric edges* (e.g., given an *asymmetric edge* containing variables A and B, the learned skeleton of A contains B, but the learned skeleton of B does not contain A), which make it difficult to construct a high quality global skeleton. To tackle this problem, this paper proposes a Bootstrap sampling based Causal Structure Learning (BCSL) algorithm. The novel contribution of BCSL is that it proposes an integrated global skeleton learning strategy that can construct more accurate global skeletons. Specifically, this strategy first utilizes the Bootstrap method to generate multiple sub-datasets, then learns the local skeleton of variables on each *asymmetric edge* on those sub-datasets, and finally designs a novel scoring function to estimate the learning results on all sub-datasets for correcting the *asymmetric edge*. Extensive experiments on both benchmark and real datasets verify the effectiveness of the proposed method.

CCS CONCEPTS

• **Computing methodologies** → **Causal reasoning and diagnostics**.

KEYWORDS

Bootstrap sampling, Causal structure learning, Directed acyclic graph, Local skeleton learning

*Corresponding author.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557249

ACM Reference Format:
Xianjie Guo, Yujie Wang, Xiaoling Huang, Shuai Yang, and Kui Yu. 2022. Bootstrap-based Causal Structure Learning. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557249

1 INTRODUCTION

Learning causal relationships between variables is an importanct goal in various disciplines, such as medicine, computer science and bioinformatics [4, 19, 24, 46]. A directed acyclic graph (DAG) or the structure of a Bayesian network (BN) [31] is one of major means used to represent causal relationships in complex systems, when a directed edge $X_i \rightarrow X_j$ in a DAG is interpreted as a direct cause (X_i) and a direct effect (X_j) relationship [3, 16, 45]. Therefore, estimating a DAG from observational data (called causal structure learning, CSL) is a critical step for inferring causal relationships between variables [13, 28, 32, 37].

In recent years, many CSL (i.e. DAG learning) methods have been proposed [22, 42], which can be mainly divided into global methods and local-to-global methods. Global methods, such as PC [35], GES [5] and NOTEARS [48], use conditional independence (CI) tests, or score functions [17, 39], or continuous optimization strategies [23, 27, 43, 44] to learn the causal structure of all variables. However, global CSL has been proven to be NP-hard [6] and its scalability has become a major problem. Particularly, when the number of variables in a dataset is large, most existing global CSL algorithms would suffer from the computational problem.

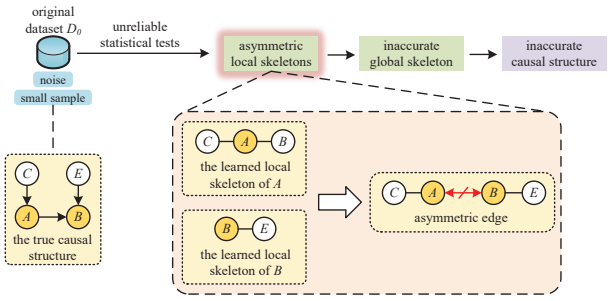


Figure 1: Problems faced by existing local-to-global causal structure learning methods.

To alleviate this problem, the local-to-global CSL methods have been designed, including GSBN [26], SLL+C/G [30] and GGSL [12],

which consist of three steps: 1) discovering the local skeleton of each variable in a dataset. A local skeleton often refers to the set of parents and children (PC) of a target variable in a DAG; 2) splicing each local skeleton into a global skeleton (undirected graph); 3) orienting the undirected edges in the global skeleton using independence tests [17, 36, 45, 47] or score-and-search strategies [5, 8, 18, 34].

Although existing local-to-global CSL methods have made landmark advances in both efficiency and accuracy, they still obtain unsatisfactory CSL performance due to inevitable data quality issues (e.g. noise and small sample). As shown in Fig. 1, data problems often make CI tests unreliable, further yielding some asymmetric local skeletons. For instance, in Fig. 1, we assume that the true causal structure behind the original data D_0 is $C \rightarrow A \rightarrow B \leftarrow E$. Theoretically, the learned local skeletons of A and B are symmetric, i.e., there is an edge between A and B in both the A's local skeleton ($C - A - B$) and the B's local skeleton ($A - B - E$). However, owing to data issues, the learned local skeletons of A and B might be asymmetric. As shown in Fig. 1, the learned local skeleton of A is $C - A - B$, but the learned local skeleton of B is $B - E$, which yields an *asymmetric edge* $A \nleftrightarrow B$. In practice, this situation is ubiquitous and seriously affects the construction of the global skeleton. As a result, the learned causal structure is quite different from the true causal structure behind the data.

Table 1: The number of *asymmetric edges* on different benchmark BNs. Here, Ee denotes the proportion of edges that actually exist in the true DAG among all the *asymmetric edges*; whereas $NotEe$ denotes the proportion of edges that do not actually exist in the true DAG among all the *asymmetric edges*. Clearly, $Ee + NotEe = 100\%$.

Bayesian network	Alarm	Alarm3	Alarm5	Alarm10
The total number of edges	46	149	265	570
Number of *asymmetric edges*	10	46	76	203
Asymmetric proportion	21.74%	30.87%	28.68%	35.61%
Ee	60.00%	45.65%	27.63%	33.99%
NotEe	40.00%	54.35%	72.37%	66.01%

To illustrate the universality of *asymmetric edges* learned by the existing local-to-global CSL algorithms, we perform experiments on four commonly used benchmark Bayesian networks (BNs), including Alarm, Alarm3, Alarm5 and Alarm10[1]. Specifically, we first utilize these four BNs to generate four synthetic datasets, each containing 500 samples. Then using these datasets, we run a classical local skeleton learning algorithm, HITON-PC [1], to learn the local skeleton of each variable. Finally, we record the number of *asymmetric edges* learned by HINTON-PC on these four datasets, and experimental results are reported in Table 1.

From Table 1, we can observe that the number of *asymmetric edges* in each dataset accounts for about 21% to 36% of the total edges. More importantly, both Ee (the proportion of edges that actually exist in the true DAG among all the *asymmetric edges*) and $NotEe$ (the proportion of edges that do not actually exist in the true DAG among all the *asymmetric edges*) float up and down 50%. The problem is that we do not have a suitable method to determine whether these *asymmetric edges* really exist in the true DAG. Existing local-to-global CSL methods usually adopt the following either correction methods: 1) all *asymmetric edges* are considered to exist in the global structure (e.g. the edge between A and B is kept in the final global skeleton in Figure 1), or 2) all *asymmetric edges* are removed from the global structure (e.g. the edge between A and B is removed in the final global skeleton in Figure 1). However, the first method may cause many false edges to be added if $NotEe > 0$; whereas, the second method may delete many true edges if $Ee > 0$.

Accordingly, a question naturally arises: can we tackle the limitation of data quality by utilizing data sampling technique [10] combined with an ensemble learning strategy [2, 15, 21, 49] and learn a more accurate causal structure? To this end, this paper proposes a novel Bootstrap [10] based causal structure learning (BCSL) algorithm for tackling the problem of *asymmetric edges*. Our contributions can be summarized as follows.

- The novel contribution of the BCSL algorithm lies in that we propose a new integrated global skeleton learning strategy that can reasonably correct the *asymmetric edges*. Specifically, we first use the Bootstrap method to sample N sub-datasets from the original dataset, and then learn the local skeleton of variables on each *asymmetric edge* again on N sub-datasets, finally design a novel scoring function to determine whether each *asymmetric edge* exists or not according to the learning results on N sub-datasets.
- Using ten benchmark BN datasets and a real-world dataset, we have conducted extensive experiments to compare BCSL with nine well-established and state-of-the-art CSL algorithms to demonstrate the effectiveness of BCSL.

2 RELATED WORK

In recent years, many CSL methods have been proposed, and they are mainly divided into global methods [5, 35, 48] and local-to-global methods [12, 26, 30].

2.1 Global causal structure learning

Global CSL methods have formulated the CSL problem as combinatorial optimization problem [5] and continuous optimization problem [48]. In the combinatorial optimization problem, existing global CSL methods are subdivided into two types: score-based and constraint-based approaches [13]. Score-based algorithms, such as GES [5] and bnlearn [25], generally use a scoring function to measure the goodness of fit of different graphs over data, and then use a search procedure to find the best graph [9]. In contrast, constraint-based methods, such as PC [35] and PC-stable [7], adopt conditional independence (CI) tests to first assess whether there is an edge between two variables, and then orient the edges [36].

To avoid the combinatorial constraint, recently, Zheng et al. transfer global CSL problem to a continuous optimization problem, and proposed the NOTEARS [48] algorithm which formulates the acyclic constraint as a smooth term and solve the problem using gradient-based numerical methods. NOTEARS is specifically developed for linear structures, and has been extended to handle nonlinear cases via neural networks [23, 27, 29, 43, 44]. To name a few, DAG-GNN [43] reconstructs data using variational auto-encoder and uses an Evidence Lower Bound (ELBO) loss as its loss function.

[1]Those benchmark BNs are publicly available at http://www.bnlearn.com/bnrepository/

GAE [29] abandons the variational part in DAG-GNN, instead, it takes graph auto-encoder as its generative model and adopts least square loss. Different from previous methods, aiming at leveraging all the parameters of the neural network in representing the weighted adjacency matrix, GraN-DAG [23] uses path products of the weights of its multilayer perceptrons (MLP) generative model to represent the matrix coefficients. Ng et al. study the asymptotic role of the sparsity and DAG constraints in the general linear Gaussian case and other specific cases, and develop a likelihood-based structure learning method with continuous unconstrained optimization, called GOLEM [27]. Compared to GOLEM, DAG-NoCurl [44] is an efficient algorithm, since it is developed based on the graph Hodge theory [20] and can solve the resultant unconstrained optimization problem in the DAG space.

However, these global CSL algorithms attempt to learn an entire causal structure at once, and they would face computational issues when the number of variables is large.

2.2 Local-to-global causal structure learning

To improve the efficiency of CSL, the local-to-global CSL approaches are developed, which first learn the local skeleton of each variable in a dataset independently. Then, those approaches construct a global skeleton by splicing these learned local skeletons, and finally orient the undirected edges in the global skeleton using independence tests [17, 36, 45, 47] or score-and-search strategies[5, 8, 18, 34].

In the past two decades, several local-to-global CSL methods have been proposed. For example, GSBN [26] first utilizes the GSMB [26] algorithm to learn the local skeleton of each variable, then constructs the global skeleton, and finally uses CI tests to orient edges. Compared to GSBN, MMHC [41] learns the local skeleton of each variable using the MMPC [40] algorithm and uses a score-and-search strategy to orient edges. SLL+C/G [30] first finds the local skeleton of each variable using a score-based local CSL algorithm (called SLL [30]), then constructs the global skeleton by combining all local skeletons, and finally SLL+C uses CI tests to orient edges in the global skeleton whereas SLL+G employs a score-and-search strategy to orient edges. Instead of finding the local skeleton of each variable in advance, the GGSL algorithm [12] first randomly selects a variable and learns the local causal structure around the variable, then gradually expands the learned structure until the entire causal structure is learned.

However, in many real-world settings, due to data issues (e.g. noise and small sample), existing local-to-global CSL methods may produce many *asymmetric edges*. To resolve these *asymmetric edges*, existing methods either assume that all *asymmetric edges* exist in the global skeleton or do not exist. In practice, the solution above may result in the loss of many true edges or the addition of many false edges in the constructed global skeleton, further leading to unsatisfactory CSL performance. In this paper, we weaken the impact of data problems on causal structure learning by data sampling technique combined with ensemble learning strategy.

3 PROPOSED BCSL APPROACH

3.1 Algorithm Overviews

Let $\boldsymbol{V}$=$\{X_1, X_2, ..., X_m\}$ denote a set of random variables, and $D^{n\times m}$ denote the input data matrix with m variables and n samples. $\mathbb{P}$ represents a joint probability distribution over $\boldsymbol{V}$, and $\mathbb{G}$ is a DAG over $\boldsymbol{V}$. In a DAG, X_1 is a parent of X_2 and X_2 is a child of X_1 if there exists a directed edge from X_1 to X_2.

In the section, we propose the BCSL approach to local-to-global causal structure learning (CSL) with three steps shown in Fig. 2. In Step 1, BCSL first discovers the local skeleton (i.e. PC set) of each variable in a dataset. Then, based on the local skeletons obtained in Step 1, Step 2 of BCSL uses an integrated global skeleton learning strategy to resolve *asymmetric edges* for constructing the best global skeleton. Finally, Step 3 employs a score-and-search strategy to orient the undirected edges in the global skeleton, and obtains a Complete Partially DAG (CPDAG), i.e., global causal structure. In the following section, we provide the details of the three steps.

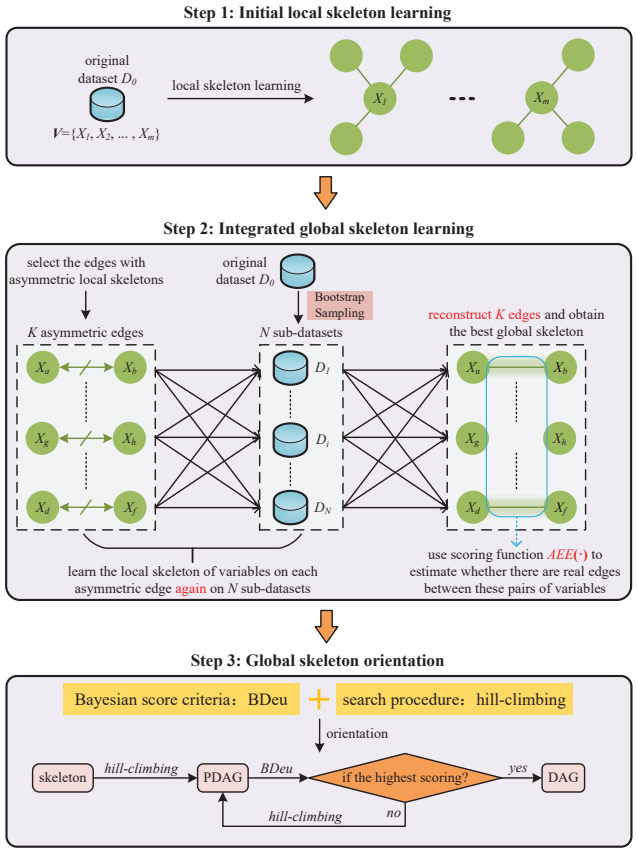


Figure 2: The framework of BCSL.

3.2 Detailed Descriptions

3.2.1 Step 1: Initial Local Skeleton Learning. Learning a local skeleton for each variable makes BCSL scalable to high-dimensional data. Given an original dataset $D_0^{n\times m}$ with the variable set $\boldsymbol{V}$=$\{X_1, X_2, ..., X_m\}$, in Step 1 of Fig. 2, BCSL aims to learn the local skeleton (i.e. PC set) of each variable in $\boldsymbol{V}$ using an existing PC (Parent-Child) learning algorithm. In our implementation, we employ HITON-PC [1], one of the best PC learning algorithms for this step. The following Proposition 3.1 is the rationale of the HITON-PC algorithm.

Proposition 3.1 ([31]). *In a DAG, if there is an edge between variables X_i and X_j, $\forall \boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X_i, X_j\}$, X_i and X_j are conditionally dependent given $\boldsymbol{Z}$.*

Proposition 3.1 states that if X_i is a parent or a child of X_j, X_i and X_j are not conditionally independent conditioning on any variable subsets. In other words, if X_i in the learned PC set of X_j, X_j must be in the learned PC set of X_i.

According to Proposition 3.1, the HITON-PC algorithm can discover the true PC set of a target variable theoretically by leveraging conditional independence (CI) tests. Let $\boldsymbol{PC}(X_i)$ represents the learned PC set of variable X_i, at the end of the step, we obtain the local skeleton of all variables, i.e., $\boldsymbol{PC}(X_1), \boldsymbol{PC}(X_2), \cdots, \boldsymbol{PC}(X_m)$.

However, as shown in Table 1, HITON-PC (or other existing PC learning methods) often yields some *asymmetric edges* due to data quality issues. For instance, $X_i \in \boldsymbol{PC}(X_j)$ but $X_j \notin \boldsymbol{PC}(X_i)$. In this case, when constructing the global skeleton, there is an *asymmetric edge* $X_i \nleftrightarrow X_j$. According to the learned PC set of each variable in $\boldsymbol{V}$, BCSL records all *asymmetric edges*, and we let the number of *asymmetric edges* be K. To determine whether each *asymmetric edge* really exists in the true global skeleton, we design an integrated global skeleton learning strategy as follows.

3.2.2 Step 2: Integrated Global Skeleton Learning. Using the learned local skeletons (i.e. PC sets) at Step 1, Step 2 is to construct the global skeleton by splicing all local skeletons. If local skeletons between variables are symmetric, such as $X_i \in \boldsymbol{PC}(X_j)$ and $X_j \in \boldsymbol{PC}(X_i)$ (or $X_i \notin \boldsymbol{PC}(X_j)$ and $X_j \notin \boldsymbol{PC}(X_i)$), we believe that there is an (or no) edge between X_i and X_j. To deal with the *asymmetric edges*, in BCSL, we design the integrated global skeleton learning strategy with the following two novel sub-steps (Steps 2-1 and 2-2) for obtaining a best global skeleton.

Step 2-1: Learn the local skeletons of variables on each *asymmetric edge* again on all sampled datasets. First, based on Bootstrap method [10], the original dataset $D_0^{n\times m}$ is sampled into N sub-datasets $(D_1, D_2, ..., D_N)$. As a common technology in ensemble learning algorithms [2, 15, 21, 49], Bootstrap is a sampling method commonly used in the field of machine learning. Given an original dataset $D_0^{n\times m}$, the process of generating a sub-dataset $D_1^{n\times m}$ through Bootstrap method is as follows:

- Randomly select a sample from $D_0^{n\times m}$ each time and put it into $D_1^{n\times m}$, and then put the sample back into the original dataset $D_0^{n\times m}$, so that the sample may still be sampled in the next sampling.
- Repeat the above procedure n times to obtain a sub-dataset $D_1^{n\times m}$ containing n samples.

Proposition 3.2 ([10]). *Given an original dataset $D_0^{n\times m}$, and generating a sub-dataset $D_1^{n\times m}$ using Bootstrap method, if $n \to \infty$, approximately 36.8% of the samples in $D_0^{n\times m}$ do not appear in $D_1^{n\times m}$.*

According to Proposition 3.2, for causal structure learning, generating sub-datasets through Bootstrap method has the following advantages:

- Bootstrap method can keep the same sample size for each sub-dataset, i.e., $D_0^{n\times m}$ and $D_1^{n\times m}$ have the same sample size.
- $D_1^{n\times m}$ has 36.8% samples that are different from $D_0^{n\times m}$. The sample difference increases the diversity of the casual structures learned from different sub-datasets.

Then, on each sub-dataset, BCSL learns the local skeleton (i.e. PC set) of variables on each *asymmetric edge* again. For example, as shown in Fig. 2, "$X_a \nleftrightarrow X_b$" is an *asymmetric edge*, thus BCSL needs to discover the local skeleton (i.e. PC set) of X_a and the local skeleton of X_b again on all sub-datasets.

Step 2-2: use a scoring function to determine whether each *asymmetric edge* exists in the global skeleton. For each *asymmetric edge*, through combining the local skeleton learning results from all sub-datasets at Step 2-1, BCSL designs a scoring function, *AEE* (Asymmetric Edge Evaluation), to determine whether this edge really exists. The detailed design process of scoring function $AEE(\cdot)$ is as follows:

Given the k-th *asymmetric edge* ($k = 1, 2, ..., K$) containing variables X_g ($g = 1, 2, ..., m$) and X_h ($h = 1, 2, ..., m$), we use a scoring function $score(g, h, j, k)$ to record whether the learned local skeleton of X_g contains X_h on the j-th sub-dataset ($j = 1, 2, ..., N$). If the learned local skeleton of X_g contains X_h, then $score(g, h, j, k) = 1$; otherwise $score(g, h, j, k) = -1$. Similarly, we also record whether the learned local skeleton of X_h contains X_g by $score(h, g, j, k)$. Then, the score of the k-th *asymmetric edge* on the j-th sub-dataset is formalized as

$$\underset{j=1,2,...,N;k=1,2,...,K}{AEE(j,k)} = score(g, h, j, k) + score(h, g, j, k). \quad (1)$$

Based on Eq. (1), we can obtain the total score of the k-th *asymmetric edge* on all sub-datasets as

$$\underset{k=1,2,...,K}{AEE(:,k)} = \sum_{j=1}^{N} AEE(j, k). \quad (2)$$

Finally, we set that if $AEE(:, k) > 0$, the k-th *asymmetric edge* will be retained in the global skeleton; otherwise, this edge will be removed.

However, due to the randomness of Bootstrap method, the quality of sub-datasets obtained by sampling is different each time. Therefore, we should adjust (increase or decrease) the score of an *asymmetric edge* on each sub-dataset according to the quality of this sub-dataset. In other words, the reliability of the learning results obtained on the sub-dataset with low data quality is lower, and we should weaken the score on this sub-dataset. In our method, BCSL uses F1 score[14] to evaluate the quality of the generated sub-dataset $D_j^{n\times m}$ by comparing the local skeleton of a variable learned on $D_0^{n\times m}$ with that learned on $D_j^{n\times m}$. Specifically, we take the local skeleton of a variable learned on $D_0^{n\times m}$ as the standard, and then calculate the F1 score of the local skeleton of the variable learned on $D_j^{n\times m}$. We think that the higher value of F1 score indicates the higher quality of the sub-dataset $D_j^{n\times m}$. BCSL utilizes a weight matrix W^k (with 2 rows and N columns) to store the F1 scores which are calculated on all sub-datasets for the two variables X_g and X_h on the k-th *asymmetric edge*, and $W^k(1, j)$ and $W^k(2, j)$ denote the F1 score calculated on the j-th sub-dataset for variable X_g and variable X_h, respectively. For instance, let $N = 6$, if

"$X_g \nleftrightarrow X_h$" is the third *asymmetric edge* and

$$W^3 = \begin{bmatrix} 0.6 & 0.7 & 0.9 & 1.0 & 0.8 & 0.8 \\ 0.5 & 0.9 & 0.6 & \mathbf{0.3} & 0.7 & 0.5 \end{bmatrix}, \tag{3}$$

$W^3(2,4) = 0.3$ denotes that the F1 score calculated on $D_4^{n\times m}$ for variable X_h on "$X_g \nleftrightarrow X_h$" is 0.3. Similarly, we also arrange $score(g,h,:,k)$ and $score(h,g,:,k)$ into a matrix with 2 rows and N columns, and obtain

$$\underset{k=1,\ldots,K}{score^*(k)} = \begin{bmatrix} score(g,h,1,k) & \cdots & score(g,h,N,k) \\ score(h,g,1,k) & \cdots & score(h,g,N,k) \end{bmatrix}. \tag{4}$$

Thus, if the quality of the generated sub-datasets is considered, the total score of the k-th *asymmetric edge* on all sub-datasets can be reformulated as

$$\underset{k=1,\ldots,K}{AEE(:,k)} = \sum_{i=1}^{2}\sum_{j=1}^{N}[score^*(k) \odot W^k]_{ij}. \tag{5}$$

Here, $\odot$ is the Hadamard product symbol, and denotes the multiplication of the corresponding position elements of two matrices with the same dimension.

In rare cases, $AEE(\cdot)$ may be equal to 0, which makes it difficult to determine whether an *asymmetric edge* exists in the global skeleton. To further avoid the case that $AEE(\cdot) = 0$, BCSL introduces a weight factor w to enlarge the influence of the weight matrix W^k on the total score $AEE(\cdot)$. The weight matrix W^k processed by the weight factor w is marked as $\hat{W^k}$, and the formulation is as follows:

$$\hat{W^k}(1,j) = \begin{cases} W^k(1,j) \times w & if \quad W^k(1,j) > W^k(2,j) \\ W^k(1,j) & otherwise \end{cases} \tag{6}$$

$$\hat{W^k}(2,j) = \begin{cases} W^k(2,j) \times w & if \quad W^k(2,j) > W^k(1,j) \\ W^k(2,j) & otherwise \end{cases} \tag{7}$$

The weight factor w is initially set to 1.0. When $AEE(\cdot) = 0$, we only need to enlarge w by any multiple (for example, 1.5 times or 2 times). "$w > 1.0$" means that the score gap of two variables on an *asymmetric edge* is further enlarged. *Finally, we define the total score of the k-th asymmetric edge on all sub-datasets as:*

$$\underset{k=1,\ldots,K}{AEE(:,k)} = \sum_{i=1}^{2}\sum_{j=1}^{N}[score^*(k) \odot \hat{W^k}]_{ij}. \tag{8}$$

Through employing scoring function $AEE(\cdot)$ based on Eq. (8), BCSL can estimate whether K *asymmetric edges* exist in the underlying causal structure behind the original dataset $D_0^{n\times m}$, and finally constructs a best global skeleton.

3.2.3 Step 3: Global Skeleton Orientation. Based on the global skeleton obtained in Step 2, BCSL uses a Bayesian score criteria, BDeu [17], and a search procedure, hill-climbing [11] to greedily orient the undirected edges in the global skeleton. Here, the BDeu score for DAG $\mathbb{G}$ learned on dataset $D^{n\times m}$ is defined as

$$BDeu(\mathbb{G}, D^{n\times m}) = \log P(\mathbb{G}) + \sum_{i=1}^{m}\sum_{l=1}^{q_i}\Big[\log\frac{\Gamma(\frac{H'}{q_i})}{\Gamma(H_{il}+\frac{H'}{q_i})} + \sum_{u=1}^{r_i}\log\frac{\Gamma(H_{ilu}+\frac{H'}{r_i q_i})}{\Gamma(\frac{H'}{r_i q_i})}\Big], \tag{9}$$

where Γ is the Gamma function, i is the index over the m variables, l is the index over the q_i combinations of values of the parents of the variable X_i, and u is the index over the r_i possible values (states) of X_i; further, H_{ilu} is the number of instances on $D^{n\times m}$ where X_i has the u^{th} value, and its parents have the l^{th} combination of values, and $H_{il} = \sum_{u=1}^{r_i} H_{ilu}$ denotes the total number of instances on $D^{n\times m}$ where the parents of X_i have the l^{th} combination of values; H' is the equivalent sample size (ESS, also sometimes known as the imaginary sample size, ISS) and expresses our confidence in the prior parameters; $P(\mathbb{G})$ is the prior probability of a particular graph structure which is generally assumed to be the same for all graphs and so can be ignored.

By alternately performing the search procedure and the scoring criteria, finally, BCSL achieves a global causal structure (i.e. CPDAG) with the highest scoring.

4 EXPERIMENTS

In this section, we present a comprehensive set of experiments to demonstrate the effectiveness of the proposed BCSL method, and this section is organized as follows. Section 4.1 gives the experimental settings. Section 4.2 and Section 4.3 summarize and discuss the experimental results on benchmark data and real data, respectively. Finally, we analyze the sensitivity of parameter N (number of sampling) of BCSL in Section 4.4.

4.1 Experiment Setting

4.1.1 Comparison Methods. We compare BCSL against two local-to-global CSL methods, GSBN [26] and GGSL [12], three well-established global CSL methods, PC [35], GES [5] and PC-stable [7], and four state-of-the-art global CSL methods, NOTEARS [48], DAG-GNN [43], GOLEM [27] and DAG-NoCurl [44].

4.1.2 Evaluation Metrics. We evaluate the performance of BCSL and its nine rivals from two aspects: structure error and structure correctness. The *SHD* (Structural Hamming Distance) and *Ar_F1* metrics as shown below are used to measure structure error and structure correctness, respectively.

- *SHD*: the number of total error edges, containing undirected edges, reverse edges, missing edges and extra edges. The smaller value of SHD is better.
- *Ar_F1*=$\frac{2*Ar\_Precision*Ar\_Recall}{Ar\_Precision+Ar\_Recall}$. The Ar_Precision metric denotes the number of correctly predicted arrowheads in the output divided by the number of edges in the output of an algorithm, while the Ar_Recall metric represents the number of correctly predicted arrowheads in the output divided by the number of true arrowheads in a test DAG. Compared to SHD, Ar_F1 not only considers erroneous edges, but also correct edges. The larger value of *Ar_F1* is better.

In Table 3, the symbol "-" denotes that an algorithm does not produce results due to out of memory. Similarly, in Figs. 3 and 4, *the values of SHD and Ar_F1 are less than 0, which means that memory is exceeded.* In addition, in all figures and tables, (↑) means the higher the better, (↓) means the lower the better, and the best results are highlighted in bold face.

4.1.3 Implementation Details. All experiments were conducted on a computer with Inter Core i9-10900 3.70-GHz CPU and 64-GB memory. PC, GSBN, PC-stable and our algorithm are implemented in MATLAB, GGSL is implemented in C++, and GES, NOTEARS, DAG-GNN, GOLEM and DAG-NoCurl are implemented in PYTHON. The

significance level for CI tests is set to 0.01, the parameter N of BCSL is set to 15, and the weight factor w of BCSL is initialized to 1.0.

4.2 Benchmark Data

In this section, we evaluate our method and its rivals on ten benchmark BNs, using the datasets provided from existing work [41]. Each BN contains three datasets with 500, 1,000 and 5,000 data instances, respectively. The details of the ten benchmark BNs are presented in Table 2[2]. In addition, we compare BIC (Bayesian information criterion) scores [38] of each algorithm on these datasets.

Table 2: Summary of benchmark BNs

Network	Num. Vars	Num. Edges	Max In/out-Degree	Min/Max |PCset|	Variable Domain
Child	20	25	2/7	1/8	2-6
Alarm	37	46	4/5	1/6	2-4
Child3	60	79	3/7	1/8	2-6
Alarm3	111	149	4/5	1/6	2-4
Insurance5	135	284	5/8	1/10	2-5
Alarm5	185	265	4/6	1/8	2-4
Insurance10	270	556	5/8	1/11	2-5
Alarm10	370	570	4/7	1/9	2-4
Pigs	441	592	2/39	1/41	3-3
Gene	801	972	4/10	0/11	3-5

4.2.1 Structure errors. From Fig. 3, we can see that on almost all benchmark datasets with 500, 1,000 and 5,000 samples, our proposed BCSL algorithm achieves a lower SHD than its rivals, which indicates the superiority of BCSL. This is because that employing the integrated global skeleton learning strategy is helpful to reduce the number of missing edges and extra edges simultaneously, and effectively alleviates the problem of asymmetry between local skeletons. As a result, BCSL obtains a best global skeleton, further reducing the number of incorrect directed edges.

Compared with the BCSL algorithm, the size of the local skeletons learned by GSBN is much small, leading to GSBN misses many true edges (i.e. having higher SHD values). In most benchmark BNs, the performance of the PC, GSBN and PC-stable algorithms is close since they all employ the constraint-based methods to construct the skeleton and orient edges. However, when the dimensionalities of BNs become higher (such as Pigs and Gene), the SHD gap between GSBN, PC and PC-stable becomes larger. With the increase of the number of nodes in BNs, the performance of GES is significantly reduced. For example, on Gene with 1,000 and 5,000 samples, the SHD value of GES is significantly higher than that of other algorithms. GGSL achieves a comparable performance against BCSL since they all use BDeu as a scoring function to orient the undirected edges.

NOTEARS often achieves a much larger number of extra edges and DAG-GNN often achieves a much larger number of missing edges than the other algorithms, leading to that they obtain the inaccurate global skeletons and the causal structures with poor quality. GOLEM and DAG-NoCurl are designed with strong theoretical assumptions, thus they achieve higher SHD values than the other algorithms on most datasets.

[2]Those benchmark BNs are publicly available at http://www.bnlearn.com/bnrepository/.

4.2.2 Structural correctness. Fig. 4 reports the quality of causal structures learned by BCSL and its rivals in terms of the Ar_F1 metric. We find that BCSL not only achieves less structure errors than its rivals in terms of SHD, bust also is superior to the other nine algorithms on Ar_F1 on almost all datasets, especially on Child and Alarm10. Compared with its rivals, BCSL obtains high values of Ar_Precision and Ar_Recall (i.e. high value of Ar_F1) on most datasets, since BCSL adopts the integrated global skeleton learning strategy to construct more accurate global skeleton, that is, some missed edges are restored and some extra edges are removed.

In addition, we find that the performance of continuous optimization methods (such as NOTEARS, DAG-GNN, GOLEM and DAG-NoCurl) is generally worse than that of combinatorial optimization methods (such as PC, GSBN, PC-stable, GGSL and BCSL) on most benchmark datasets in terms of SHD and Ar_F1. This is because that the causal structures learned by continuous optimization methods contain many extra edges and reverse edges.

In summary, our method is obviously superior to other algorithms on both sparse network (such as Child, Alarm, Child3, Alarm3, Alarm5, Alarm10 and Gene) and dense network (such as Insurance5 and Pigs).

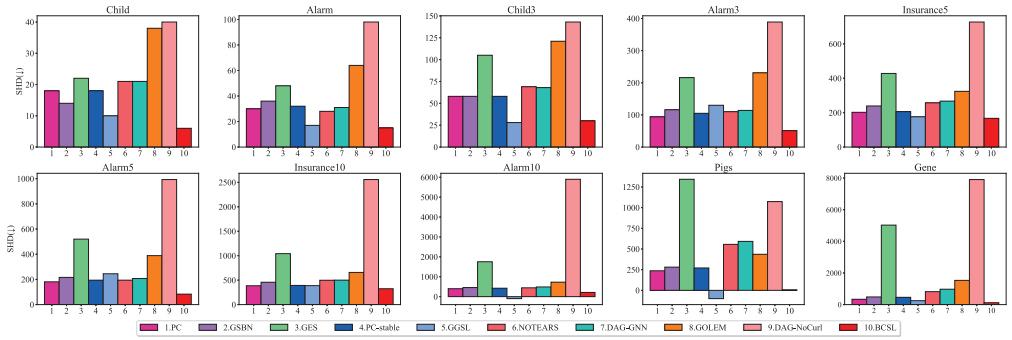
4.2.3 The BIC score of each algorithm. We are unable to reasonably evaluate how good the learned causal structures are as probability models, if only using the structure evaluation metrics, SHD and Ar_F1. In this section, to evaluate the proximity of the learned causal structure to the real probability distribution, we compare the BIC (Bayesian information criterion) score [38] of each algorithm shown in Table 3. The higher the BIC score, the higher the fitting degree between a learned causal structure and a dataset. As a commonly used information-theoretic score, the BIC score can avoid overfitting by balancing the goodness of fit with the dimensionality of the learned structure given the limited data.

From Table 3, we can see that BCSL improves the learning scores by a significant margin over other algorithms on all datasets with 500, 1,000 and 5,000 samples. GGSL achieves a comparable performance against BCSL on some datasets (such as Child3 and Gene) since it employs the score-based method to learn the local causal structures and the same scoring function (BDeu score) as BCSL to orient edges. Although existing neural CSL approaches (NOTEARS, DAG-GNN, GOLEM and DAG-NoCurl) have used different types of neural network models, loss functions and representations of adjacency matrix to improve their performance, but they still achieve lower BIC scores since the causal structures learned by these algorithms contain many extra edges and reverse edges.

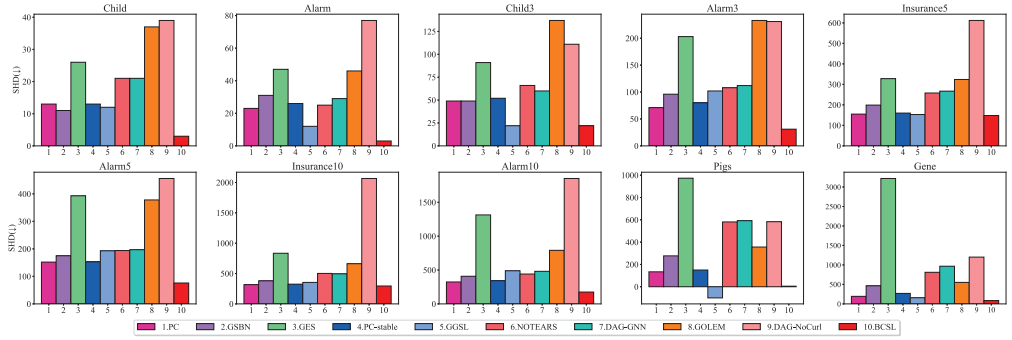
4.3 Real Data

The interpretability of biology data is of significance. Here we apply BCSL to a bioinformatics dataset [33] for the discovery of a protein signaling network based on expression levels of proteins and phospholipids. This is a widely used dataset for research on graphical models, with experimental annotations accepted by the biological research community.

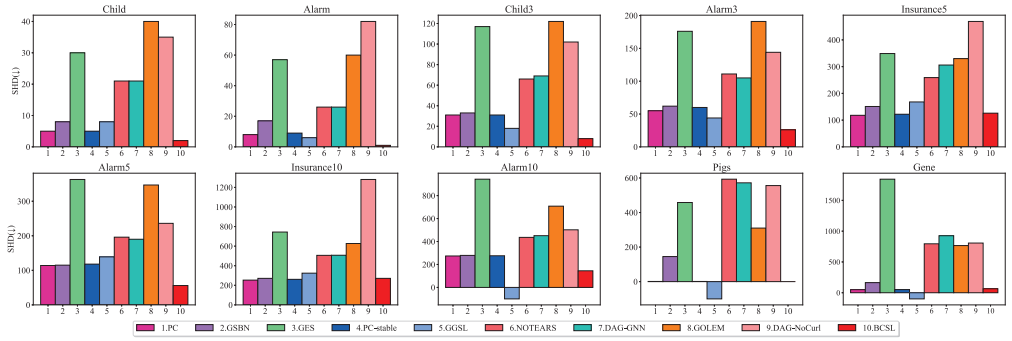
The ground truth graph contains 11 nodes and 17 edges. In the experiments, we use n =7466 observed samples for training. Among all methods in the experiments, BCSL achieves the best



(a) 500 samples



(b) 1000 samples



(c) 5000 samples

Figure 3: SHDs of BCSL and its nine rivals on all benchmark datasets with 500, 1,000 and 5,000 samples. Note that *the values of SHD and Ar_F1 are less than 0, which means that memory is exceeded.*

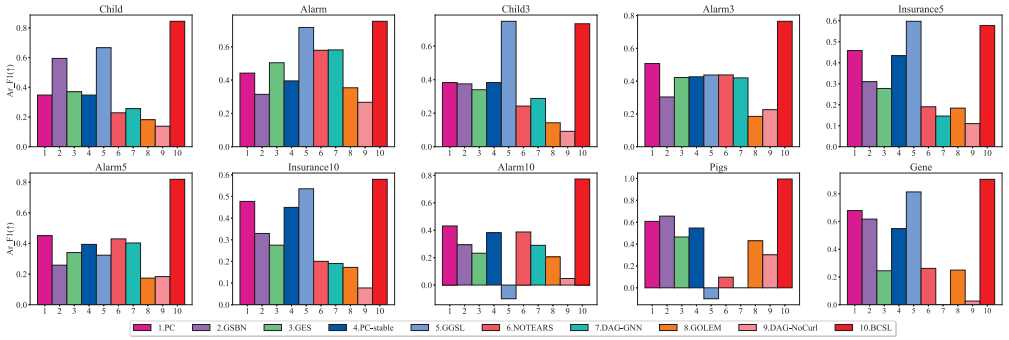
performance with SHD 10. GSBN has an SHD 14, GGSL has an SHD 16, and NOTEARS, DAG-GNN and GOLEM all have an SHD 18.

The performance of other methods are much worse perhaps due to strong theoretical assumptions. The results are summarized in Table 4. Besides SHD, we also incorporate the results of Ar_F1, Ar_Precision and Ar_Recall, and we find that BCSL achieves higher values of Ar_F1 and Ar_Precision than its rivals. Although GES achieves a high Ar_Recall, it also learns many extra edges. In addition, we also observe that the performance of continuous optimization approaches is comparable to that of the traditional methods on real data, whereas generally worse than that of traditional methods on benchmark data.

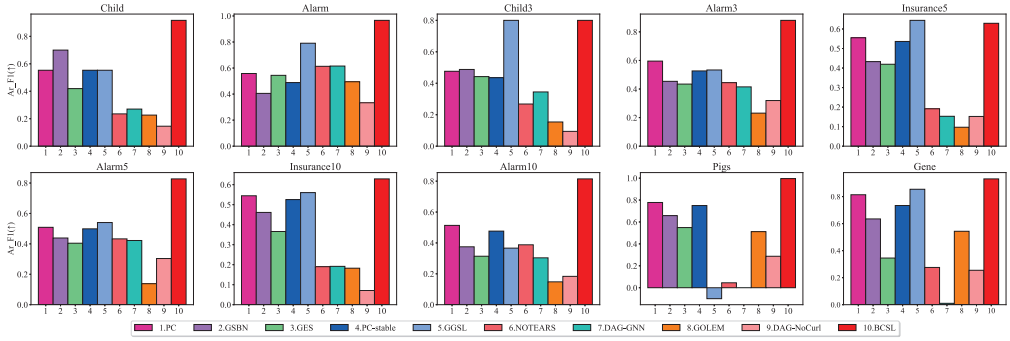
4.4 Parameter sensitivity analysis

In step 2-1 of BCSL (see Section 3.2.2 for details), we need to determine the number of sub-datasets (i.e. N) generated by Bootstrap sampling in advance. In this section, we study the sensitivity of the parameter N in our proposed BCSL method.

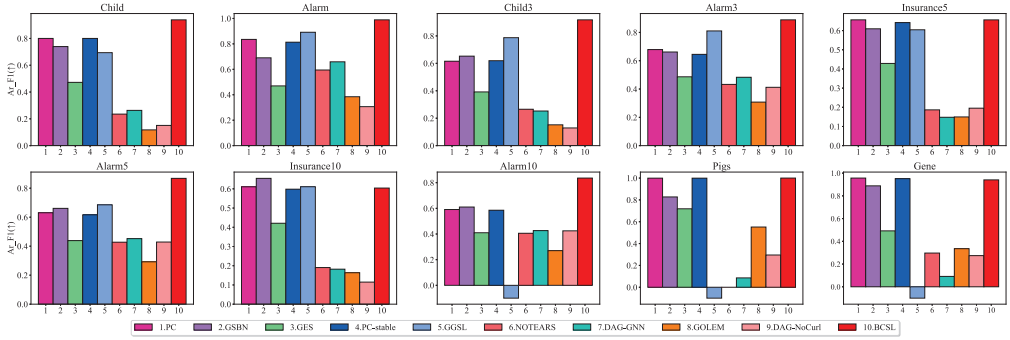
Specifically, on 10 benchmark BN datasets with 500, 1,000 and 5,000 samples, we perform BCSL by varying the value of parameter N from 5 to 100, and the experimental results are plotted in Fig. 5.



(a) 500 samples



(b) 1000 samples



(c) 5000 samples

Figure 4: Ar_F1s of BCSL and its nine rivals on all benchmark datasets with 500, 1,000 and 5,000 samples. Note that *the values of SHD and Ar_F1 are less than 0, which means that memory is exceeded.*

In Fig. 5, (a), (b) and (c) show the variation curve of SHD, and (d), (e) and (f) show the variation curve of Ar_F1. From the experimental results, we have the following three observations.

(1) On the large-sized networks (such as Pigs and Gene), the influence of N on the experimental results is relatively small. In contrast, on the small-sized networks (such as Child and Alarm), the parameter N has a relatively large effect on the experimental results.
(2) For most benchmark BNs, with the increase of the sample size, the influence of N on the experimental results becomes smaller.
(3) When $N \geqslant 15$, the fluctuation of SHD and A_F1 metrics will be very small on almost all benchmark BNs.

Thus, in our experiments, the parameter N of BCSL is set to 15 on all benchmark BN datasets.

5 CONCLUSION

In this paper, we first show that existing local-to-global CSL methods learn many *asymmetric edges* during the global skeleton construction phase. Then, we design a novel integrated global skeleton learning strategy to deal with the problem of *asymmetric edges*. Finally, based on this strategy, we propose the BCSL method to learn

Table 3: Learning BIC scores for different causal structure learning algorithms on different datasets

#Sample	Network	PC	GSBN	GES	PC-stable	GGSL	NOTEARS	DAG-GNN	GOLEM	DAG-NoCurl	BCSL
500	Child	-17126.13	-7157.42	-11871.12	-17126.13	-6818.25	-7434.11	-9416.23	-7501.08	-13966.54	**-6660.14**
	Alarm	-7749.98	-7111.18	-31342.93	-7895.60	-7268.30	-7641.32	-14717.05	-9019.57	-30500.04	**-6193.17**
	Child3	-95391.17	-22264.25	-62097.57	-95391.17	-21094.46	-23406.10	-26122.99	-23840.51	-284292.69	**-20524.77**
	Alarm3	-29764.12	-22065.98	-112485.67	-43659.48	-25909.82	-22764.42	-23312.90	-28138.89	-96596419.39	**-20795.35**
	Insurance5	-69649.69	-48848.03	-651104.89	-64975.10	-46684.98	-50765.96	-51841.46	-53668.85	-43007178.97	**-44782.63**
	Alarm5	-72632.90	-37585.42	-348376.11	-75512.80	-53322.08	-39736.90	-39650.32	-47650.98	-45783.04	**-35179.68**
	Insurance10	-134218.70	-96994.05	-32323647.14	-124676.50	-106275.37	-102485.58	-103351.77	-106601.95	-56045725.56	**-90561.04**
	Alarm10	-115601.63	-74389.27	-200561073.49	-118882.53	-	-77549.73	-81193.40	-90065.27	-569541457.84	**-69220.62**
	Pigs	-215632.86	-200784.12	-89772641.37	-236789.18	-	-217172.24	-229766.65	-193792.61	-288571.04	**-181294.69**
	Gene	-318754.52	-261953.07	-306535.16	-331220.36	-252344.40	-291898.26	-321726.68	-301678.21	-2965350.17	**-242951.41**
1000	Child	-19184.43	-13609.48	-42434.19	-19184.43	-12886.44	-14812.84	-14984.35	-15226.71	-17850.08	**-12679.66**
	Alarm	-11761.58	-12806.19	-47340.05	-12773.70	-13459.85	-12224.10	-47636.07	-15126.84	-62811.76	**-10863.40**
	Child3	-84242.01	-41916.01	-61444.10	-161872.91	-40055.84	-46516.78	-46323.37	-46327.11	-112588.40	**-39774.90**
	Alarm3	-38954.12	-40286.81	-86653.44	-39310.73	-43578.02	-45034.98	-45935.51	-51857.72	-60067.28	**-38399.79**
	Insurance5	-97337.08	-91695.78	-451821.99	-94375.61	-89973.08	-98589.66	-100922.76	-105953.59	-1697342.56	**-84233.33**
	Alarm5	-67930.14	-69038.10	-211754.12	-68237.99	-77208.56	-76893.24	-76574.09	-90494.69	-1643380.47	**-64973.07**
	Insurance10	-190420.98	-183756.50	-2873632.40	-191916.47	-207067.20	-196635.30	-195132.57	-197207.79	-15471176.67	**-167954.90**
	Alarm10	-138212.16	-137284.95	-12205684.77	-142747.41	-171752.19	-152146.15	-157045.74	-180916.02	-174765.89	**-128268.30**
	Pigs	-385678.97	-393163.23	-4806476.87	-389176.89	-	-436163.99	-455894.32	-364977.16	-408404.24	**-348484.86**
	Gene	-499024.49	-511325.58	-1771793755.57	-517541.21	-471147.00	-575523.96	-635810.84	-597121.67	-682104.32	**-463194.22**
5000	Child	-62738.14	-62978.58	-77437.75	-62738.14	-65670.54	-73748.94	-72967.94	-75449.73	-73518.40	**-61746.14**
	Alarm	-49288.62	-52922.32	-160831.02	-49439.83	-48784.59	-58598.72	-58386.28	-66946.32	-86952.88	**-48777.28**
	Child3	-198055.73	-197034.98	-374244.35	-198025.38	-192114.84	-227468.27	-225140.79	-225740.64	-218429.26	**-189172.16**
	Alarm3	-180275.86	-180419.48	-401687.85	-183077.17	-182965.87	-218952.60	-207282.74	-219265.50	-196666.01	**-176155.05**
	Insurance5	-390174.49	-404798.20	-3274020.49	-395408.03	-448108.43	-467543.34	-466804.73	-466424.25	-499922.64	**-383047.90**
	Alarm5	-310183.69	-310160.90	-889463.18	-310540.42	-336318.03	-373133.36	-363502.07	-383958.90	-335712.48	**-298738.43**
	Insurance10	-789515.31	-800858.50	-9471924.23	-796845.48	-866841.94	-939225.10	-931651.19	-943201.41	-2069982.03	**-777071.81**
	Alarm10	-615260.78	-615356.93	-36998846.00	-618783.86	-	-737458.40	-702486.54	-794689.13	-670961.04	**-589827.95**
	Pigs	-1674401.02	-1813881.97	-1835053.56	-1674401.02	-	-2186094.18	-2100834.83	-1750940.27	-1916146.16	**-1674260.41**
	Gene	-2216702.44	-2303722.11	-35335084.76	-2223339.88	-	-2825816.71	-3076358.62	-2759648.51	-2590280.49	**-2214672.05**

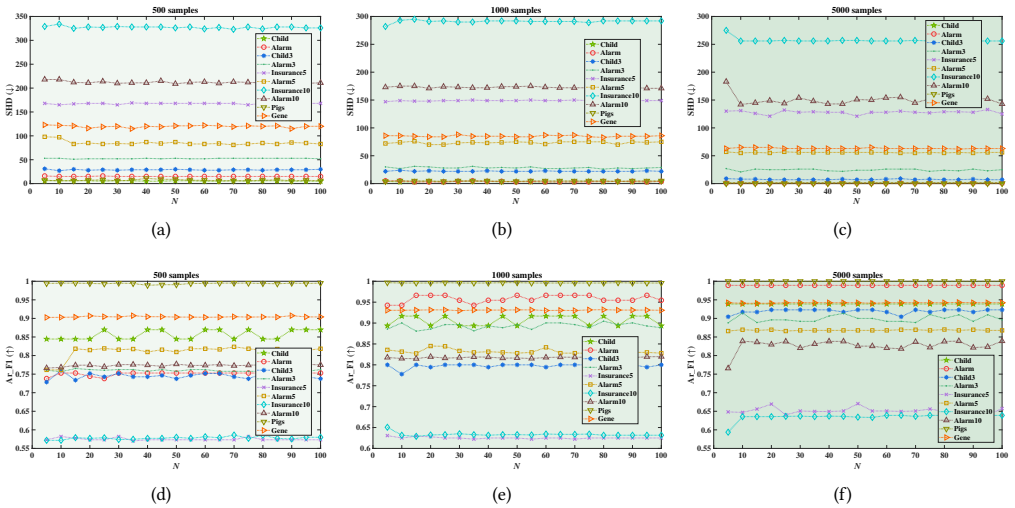


Figure 5: Sensitivity analysis of parameters N of BCSL on different benchmark BN datasets.

Table 4: Results on protein signaling network: comparison of the predicted graphs with respect to the ground truth.

Method	SHD (↓)	Ar_F1 (↑)	Ar_Precision (↑)	Ar_Recall (↑)
PC	39	0.2540	0.1739	0.4706
GSBN	14	0.3571	0.4545	0.2941
GES	29	0.4364	0.3158	**0.7059**
PC-stable	39	0.2540	0.1739	0.4706
GGSL	16	0.2857	0.3636	0.2353
NOTEARS	18	0.2143	0.2727	0.1765
DAG-GNN	18	0.1600	0.2500	0.1176
GOLEM	18	0.2143	0.2727	0.1765
DAG-NoCurl	23	0.1818	0.1875	0.1765
BCSL	**10**	**0.5517**	**0.6667**	0.4706

a more accurate causal structure. Experiments have shown that the proposed BCSL outperforms two existing local-to-global CSL algorithms and seven global CSL algorithms in terms of the quality of causal structure learning. In addition, our proposed integrated global skeleton learning strategy can also be integrated to most existing local-to-global CSL algorithms. Therefore, in future, we could consider designing BCSL as a unified framework to improve the performance of most existing local-to-global CSL algorithms.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China under Grant 2021ZD0111801, in part by the National Natural Science Foundation of China under Grant 61876206, and in part by the Key Project of the Natural Science Foundation of Educational Commission of Anhui Province under Grant KJ2021A1065

REFERENCES

[1] Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. 2010. Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: algorithms and empirical evaluation. *Journal of Machine Learning Research* 11, 1 (2010).

[2] Leo Breiman. 1996. Bagging predictors. *Machine learning* 24, 2 (1996), 123–140.

[3] Ruichu Cai, Jie Qiao, Kun Zhang, Zhenjie Zhang, and Zhifeng Hao. 2019. Causal discovery with cascade nonlinear additive noise models. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 1609–1615.

[4] Rui Chen, Sanjeeb Dash, and Tian Gao. 2021. Integer Programming for Causal Structure Learning in the Presence of Latent Variables. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Vol. 139. 1550–1560.

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

[6] Max Chickering, David Heckerman, and Chris Meek. 2004. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research* 5 (2004).

[7] Diego Colombo, Marloes H Maathuis, et al. 2014. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.* 15, 1 (2014), 3741–3782.

[8] James Cussens. 2011. Bayesian network learning with cutting planes. In *UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011*. 153–160.

[9] Cassio P de Campos, Mauro Scanagatta, Giorgio Corani, and Marco Zaffalon. 2018. Entropy-based pruning for learning Bayesian networks using BIC. *Artificial Intelligence* 260 (2018), 42–50.

[10] Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

[11] José A Gámez, Juan L Mateo, and José M Puerta. 2011. Learning Bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood. *Data Mining and Knowledge Discovery* 22, 1 (2011), 106–148.

[12] Tian Gao, Kshitij Fadnis, and Murray Campbell. 2017. Local-to-global Bayesian network structure learning. In *International Conference on Machine Learning*. PMLR, 1193–1202.

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

[14] Xianjie Guo, Kui Yu, Fuyuan Cao, Peipei Li, and Hao Wang. 2022. Error-Aware Markov Blanket Learning for Causal Feature Selection. *Information Sciences* (2022).

[15] Lars Kai Hansen and Peter Salamon. 1990. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence* 12, 10 (1990), 993–1001.

[16] Yue He, Peng Cui, Zheyan Shen, Renzhe Xu, Furui Liu, and Yong Jiang. 2021. Daring: Differentiable causal discovery with residual independence. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 596–605.

[17] David Heckerman, Dan Geiger, and David M Chickering. 1995. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning* 20, 3 (1995), 197–243.

[18] Biwei Huang, Kun Zhang, Yizhu Lin, Bernhard Schölkopf, and Clark Glymour. 2018. Generalized score functions for causal discovery. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1551–1560.

[19] Johannes Huegle, Christopher Hagedorn, Michael Perscheid, and Hasso Plattner. 2021. MPCSL-A Modular Pipeline for Causal Structure Learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3068–3076.

[20] Xiaoye Jiang, Lek-Heng Lim, Yuan Yao, and Yinyu Ye. 2011. Statistical ranking and combinatorial Hodge theory. *Mathematical Programming* 127, 1 (2011), 203–244.

[21] ZHANG Jiang-She. 2011. A survey of selective ensemble learning algorithms. *Chinese Journal of Computers* 34, 8 (2011), 1399–1410.

[22] Neville K Kitson, Anthony C Constantinou, Zhigao Guo, Yang Liu, and Kiattikun Chobtham. 2021. A survey of Bayesian Network structure learning. *arXiv preprint arXiv:2109.11415* (2021).

[23] Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. 2020. Gradient-based neural dag learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[24] Honghao Li, Vincent Cabeli, Nadir Sella, and Hervé Isambert. 2019. Constraint-based Causal Structure Learning with Consistent Separating Sets. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.

[25] Dimitris Margaritis. 2003. *Learning Bayesian network model structure from data*. Technical Report. Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science.

[26] Dimitris Margaritis and Sebastian Thrun. 1999. Bayesian network induction via local neighborhoods. In *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*. 505–511.

[27] Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. 2020. On the role of sparsity and dag constraints for learning linear dags. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

[28] Ignavier Ng, Yujia Zheng, Jiji Zhang, and Kun Zhang. 2021. Reliable Causal Discovery with Improved Exact Search and Weaker Assumptions. *Advances in Neural Information Processing Systems* 34 (2021).

[29] Ignavier Ng, Shengyu Zhu, Zhitang Chen, and Zhuangyan Fang. 2019. A graph autoencoder approach to causal structure learning. *arXiv preprint arXiv:1911.07420* (2019).

[30] Teppo Niinimaki and Pekka Parviainen. 2012. Local structure discovery in Bayesian networks. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*. 634–643.

[31] Judea Pearl. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann.

[32] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

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

[34] Tomi Silander and Petri Myllymaki. 2006. A simple approach for finding the globally optimal Bayesian network structure. In *UAI '06, Proceedings of the 22nd Conference in Uncertainty in Artificial Intelligence, Cambridge, MA, USA, July 13-16, 2006*. AUAI Press.

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

[36] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. 2000. *Causation, prediction, and search*. MIT press.

[37] Scott Sussex, Caroline Uhler, and Andreas Krause. 2021. Near-Optimal Multi-Perturbation Experimental Design for Causal Structure Learning. *Advances in Neural Information Processing Systems* 34 (2021).

[38] Joe Suzuki. 1993. A construction of Bayesian networks from databases based on an MDL principle. In *Uncertainty in Artificial Intelligence*. Elsevier, 266–273.

[39] Marc Teyssier and Daphne Koller. 2012. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. *arXiv preprint arXiv:1207.1429* (2012).

[40] Ioannis Tsamardinos, Constantin F Aliferis, and Alexander Statnikov. 2003. Time and sample efficient discovery of Markov blankets and direct causal relations. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 673–678.

[41] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. 2006. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine learning* 65, 1 (2006), 31–78.

[42] Matthew J Vowels, Necati Cihan Camgoz, and Richard Bowden. 2021. D'ya like dags? a survey on structure learning and causal discovery. *arXiv preprint arXiv:2103.02582* (2021).

[43] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. 2019. Dag-gnn: Dag structure learning with graph neural networks. In *International Conference on Machine Learning*. PMLR, 7154–7163.

[44] Yue Yu, Tian Gao, Naiyu Yin, and Qiang Ji. 2021. DAGs with No Curl: An Efficient DAG Structure Learning Approach. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Vol. 139. 12156–12166.

[45] Hao Zhang, Kun Zhang, Shuigeng Zhou, Jihong Guan, and Ji Zhang. 2021. Testing Independence Between Linear Combinations for Causal Discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6538–6546.

[46] Hao Zhang, Shuigeng Zhou, Chuanxu Yan, Jihong Guan, and Xin Wang. 2019. Recursively learning causal structures using regression-based conditional independence test. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3108–3115.

[47] Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2011. Kernel-based Conditional Independence Test and Application in Causal Discovery. In *UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011*. AUAI Press, 804–813.

[48] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P Xing. 2018. Dags with no tears: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. 9492–9503.

[49] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. 2002. Ensembling neural networks: many could be better than all. *Artificial intelligence* 137, 1-2 (2002), 239–263.