Progressive Skeleton Learning for Effective Local-to-Global Causal Structure Learning

Xianjie Guo, Kui Yu, *Member, IEEE*, Lin Liu, Jiuyong Li, *Member, IEEE*, Jiye Liang, *Senior Member, IEEE*, Fuyuan Cao, and Xindong Wu, *Fellow, IEEE*

***Abstract*—Causal structure learning (CSL) from observational data is a crucial objective in various machine learning applications. Recent advances in CSL have focused on local-to-global learning, which offers improved efficiency and accuracy. The local-to-global CSL algorithms first learn the local skeleton of each variable in a dataset, then construct the global skeleton by combining these local skeletons, and finally orient edges to infer causality. However, data quality issues such as noise and small samples often result in the presence of problematic *asymmetric edges* during global skeleton construction, hindering the creation of a high-quality global skeleton. To address this challenge, we propose a novel local-to-global CSL algorithm with a progressive enhancement strategy and make the following novel contributions: 1) To construct an accurate global skeleton, we design a novel strategy to iteratively correct *asymmetric edges* and progressively improve the accuracy of the global skeleton. 2) Based on the learned accurate global skeleton, we design an integrated global skeleton orientation strategy to infer the correct directions of edges for obtaining an accurate and reliable causal structure. Extensive experiments demonstrate that our method achieves better performance than the existing CSL methods.**

***Index Terms*—Asymmetric edges, local-to-global causal structure learning, progressive learning, skeleton learning.**

I. Introduction

UNCOVERING causal relations from observational data (also known as causal discovery) is important for a broad range of applications, such as Earth system science [1], the development of medical treatments [2] and biology [3]. Although controlled experiments can infer causal relations effectively, they cannot usually be undertaken due to prohibitive cost, ethical concerns, or impracticality. For example, to understand the impact of alcoholism, it would be necessary to force different individuals to drink. Fortunately, causal graphical modeling techniques, often based on Directed Acyclic Graphs (DAGs), can be used to represent causal relationships in complex systems. In a DAG, a directed edge $X_i \to X_j$ is interpreted as a direct cause (X_i) and direct effect (X_j) relationship [4], [5], [6], [7]. The process of learning causal structures (i.e. DAGs) from observational data has emerged as the primary approach for inferring causal relationships between variables [8], [9], [10], [11].

Existing methods for causal structure learning (CSL) [12], [13] can be mainly divided into global methods and local-to-global methods. Global methods, such as PC [14] and its variant algorithms [15], [16], [17]; GES [18] and its variant algorithms [19], [20]; NOTEARS [21] and its derivative algorithms [22], [23], [24], [25], use conditional independence (CI) tests, score-and-search strategies, and continuous optimization strategies to learn the causal structure over all variables in a dataset, respectively. However, global CSL is either an NP-hard problem [26] or requires the utilization of complex optimization techniques [21] (also intricate neural network models [22]), resulting in a significant decrease in its scalability. Particularly, when the number of variables in a dataset is large, most existing global CSL algorithms would suffer from the computational problem. In addition, when a global CSL algorithm makes mistakes in learning the edges (i.e., missing, adding or reversing edges), the errors are permanent in the structure, and escalated in the follow-up learning process, leading to unsatisfactory accuracy.

To alleviate these two issues (scalability and accuracy), local-to-global CSL methods have been designed, such as GSBN [27], SLL+C/G [28], GGSL [29] and F2SL-c/s [30], which usually consist of three steps: 1) learning the local skeleton of each variable independently. A local skeleton often refers to the set of parents and children (PC) of a target variable in a DAG; 2) constructing the global skeleton (undirected graph) by merging all local skeletons; 3) orienting edges in the global skeleton using independence tests [31], [32], [33] or score-and-search strategies [18], [34], [35].

Although existing local-to-global CSL methods have made landmark advances in both efficiency and accuracy, their performance is still unsatisfactory due to the inevitable data quality issues (e.g. noise and small sample). Specifically, the data

Received 2 May 2024; revised 27 August 2024; accepted 10 September 2024. Date of publication 17 September 2024; date of current version 13 November 2024. This work was supported in part by National Science and Technology Major Project under Grant 2020AAA0106100, in part by the National Natural Science Foundation of China under Grant 62376087, and in part by China Scholarship Council under Grant 202306690023. Recommended for acceptance by T. Weninger. *(Corresponding author: Kui Yu.)*

Xianjie Guo, Kui Yu, and Xindong Wu are with the Key Laboratory of Knowledge Engineering with Big Data of Ministry of Education, Hefei 230601, China, also with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China (e-mail: xianjieguo@mail.hfut.edu.cn; yukui@hfut.edu.cn; xwu@hfut.edu.cn).

Lin Liu and Jiuyong Li are with UniSA STEM, University of South Australia, Adelaide, SA 5095, Australia (e-mail: Lin.Liu@unisa.edu.au; Jiuyong.Li@unisa.edu.au).

Jiye Liang and Fuyuan Cao are with the School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China (e-mail: ljy@sxu.edu.cn; cfy@sxu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TKDE.2024.3461832, provided by the authors.

Digital Object Identifier 10.1109/TKDE.2024.3461832

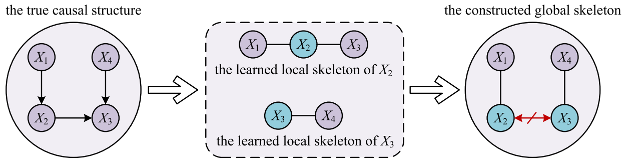
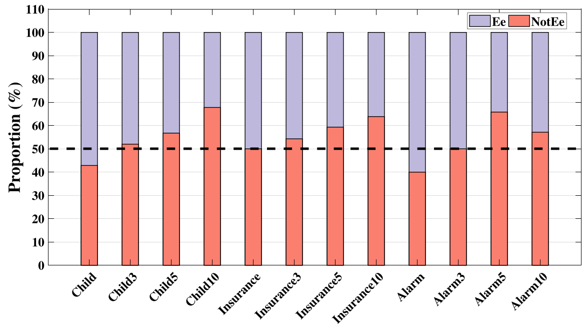

the true causal structure
X_1
X_4
X_2
X_3
X_1
X_2
X_3
the learned local skeleton of X_2
X_3
X_4
the learned local skeleton of X_3
the constructed global skeleton
X_1
X_4
X_2
X_3

Fig. 1. An example to illustrate why existing local-to-global CSL methods may encounter *asymmetric edges*.

Bayesian network	Child	Child3	Child5	Child10
The total number of edges	25	79	126	257
Number of asymmetric edges	7	25	37	90
Asymmetric proportion	28.00%	31.65%	29.37%	35.02%
Bayesian network	Insurance	Insurance3	Insurance5	Insurance10
The total number of edges	52	163	284	556
Number of asymmetric edges	14	35	59	130
Asymmetric proportion	26.92%	21.47%	20.77%	23.38%
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%

(a) The number of *asymmetric edges* learned on different datasets.



(b) Proportions of *asymmetric edges* that exist and do not exist in the true causal structure. Ee denotes the proportion of edges that actually exist in the true causal structure among all the *asymmetric edges*; whereas $NotEe$ denotes the proportion of edges that do not actually exist in the true causal structure. Clearly, $Ee + NotEe = 100\%$.

Fig. 2. The prevalence of *asymmetric edges* learned by the existing local-to-global causal structure learning algorithms.

problems often make CI tests unreliable, yielding some asymmetric local skeletons. For instance, in Fig. 1, we assume that the true causal structure behind the data is $X_1 \rightarrow X_2 \rightarrow X_3 \leftarrow X_4$. The true local skeletons of X_2 and X_3 are symmetric, i.e., there is an edge between X_2 and X_3 in both X_2's local skeleton ($X_1 - X_2 - X_3$) and X_3's local skeleton ($X_2 - X_3 - X_4$). However, owing to data issues, the learned local skeletons of X_2 and X_3 by existing methods might be asymmetric, as illustrated in Fig. 1, where the learned local skeleton of X_2 is $X_1 - X_2 - X_3$, but the learned local skeleton of X_3 is $X_3 - X_4$, which yields an *asymmetric edge* $X_2 \nleftrightarrow X_3$. In practice, this situation is ubiquitous and seriously affects the construction of the global skeleton.

To illustrate the prevalence of *asymmetric edges* learned by the existing local-to-global CSL algorithms, we perform experiments on several commonly used benchmark Bayesian networks (BNs), including Child, Child3, Child5, Child10, Insurance, Insurance3, Insurance5, Insurance10, Alarm, Alarm3, Alarm5 and Alarm10.[1] Specifically, we first utilize these 12 BNs to generate 12 benchmark datasets, each containing 500 samples. Then, using these datasets, we run a classical local skeleton learning algorithm, HITON-PC [36], to learn the local skeleton of each variable. Finally, we record the number of *asymmetric edges* learned by HINTON-PC on these 12 datasets, and the results are reported in Fig. 2(a).

From Fig. 2(a), we can observe that the number of *asymmetric edges* in each dataset accounts for about 21% to 36% of the total edges. Further, in Fig. 2(b), we also report the proportion of the *asymmetric edges* that exist and do not exist in the true causal structure. Intuitively, 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*) fluctuate around 50%.

In practice, the problem is that we do not have a suitable strategy to determine whether an *asymmetric edge* really exists in the true causal structure since we do not know the true structure. Existing local-to-global CSL algorithms usually adopt either of the following correction methods: 1) the final global skeleton retains all *asymmetric edges* (such that, e.g. in Fig. 1, X_2 and X_3 are considered adjacent in the constructed global skeleton), or 2) the final global skeleton removes all *asymmetric edges* (such that, e.g. in Fig. 1, X_2 and X_3 are considered nonadjacent in the constructed global skeleton). However, the first method will cause false edges to be added to the final global skeleton if $NotEe \neq 0$; whereas, the second method will delete true edges from the final global skeleton if $Ee \neq 0$. Further, from the empirical evidence shown in Fig. 2, both Ee and $NotEe$ are around 50%, so any of the above correction methods can at most fix around 50% of the wrongly learned edges due to the *asymmetric edge* issue.

To alleviate the problem of *asymmetric edges* more effectively, this paper proposes a novel local-to-global CSL algorithm with a progressive strategy, called PCSL (Progressive Causal Structure Learning by interleaving skeleton learning and data sampling), which can progressively improve the correction accuracy of *asymmetric edges*. Our contributions can be summarized as follows.

- We design a progressive global skeleton construction strategy to iteratively correct *asymmetric edges*. In each iteration, this strategy utilizes data sampling procedure combined with a novel weighted scoring function to correct *asymmetric edges* so that a more accurate global structure is learned, and in turn, a more accurate global skeleton guides the generation of higher quality sampled sub-datasets. As a result, this strategy achieves progressive enhancement by continuously improving the correction accuracy of *asymmetric edges*. We also theoretically analyze the effectiveness of this progressive strategy.
- Based on the best global skeleton and the best sampled sub-datasets, we design the integrated global skeleton orientation strategy, using the ideas of dispersion and aggregation, to learn an accurate causal structure.

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

- Building upon the aforementioned novel strategies, we introduce PCSL, a highly effective and efficient local-to-global causal structure learning algorithm.
- Using benchmark BN datasets and real-world datasets, we have conducted extensive experiments to demonstrate the superiority of our proposed method.

II. Related Work

CSL methods are mainly divided into global methods [14], [18], [21] and local-to-global methods [27], [28], [29].

A. Global Causal Structure Learning

Global CSL methods formulate the CSL problem as a combinatorial optimization problem or continuous optimization problem. In the combinatorial optimization problem, existing global CSL methods are subdivided into two types: score-based and constraint-based approaches. Score-based algorithms, such as GES [18] and its variant algorithms [19], [20], 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 [37]. In contrast, constraint-based methods, such as PC [14] and its variant algorithms [15], [16], [17], adopt conditional independence (CI) tests to first assess whether there is an edge between two variables for learning an undirected graph, and then orient the edges [33].

To avoid the combinatorial constraint, Zheng et al. transformed global CSL problem to a continuous optimization problem, and proposed the NOTEARS [21] algorithm which formulates the acyclic constraint as a smooth term and solves 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. For example, DAG-GNN [22] reconstructs data using variational auto-encoder and uses an Evidence Lower Bound (ELBO) loss as its loss function. GAE [38] 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. Additionally, the authors of [24] studied the asymptotic role of the sparsity and DAG constraints in the general linear Gaussian case and other specific cases, and developed a likelihood-based structure learning method with continuous unconstrained optimization, called GOLEM [24]. Compared with GOLEM, DAG-NoCurl [25] is an efficient algorithm, since it is developed based on the graph Hodge theory [39] 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. In addition, if these global CSL methods learn false edges (e.g., missing edges, extra edges and reverse edges) in the early learning stage, the learning and orientation of the edges around these false edges will be seriously affected, resulting in more errors.

B. Local-to-Global Causal Structure Learning

To improve the efficiency and effectiveness of CSL, local-to-global CSL approaches are developed, which first learn the local structure of each variable independently instead of learning the global causal structure at once. In the past two decades, several local-to-global CSL methods have been proposed. For example, GSBN [27] first utilizes the GSMB [27] algorithm to learn the local skeleton of each variable, then constructs the global skeleton, and finally uses CI tests to orient edges. Compared with GSBN, MMHC [40] learns the local skeleton of each variable using the MMPC [41] algorithm and uses a score-and-search strategy to orient edges. SLL+C/G [28] first finds the local skeleton of each variable using a score-based local CSL algorithm (called SLL [28]), 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 [29] 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. Recently, Yu et al. point out that existing local skeleton learning algorithms adopted by the local-to-global methods are often computationally expensive, especially with a large-sized networks [30]. To further improve the efficiency of the local-to-global CSL algorithms, they linked feature selection methods to CSL, and proposed two efficient local-to-global CSL algorithms, F2SL-c and F2SL-s [30], which employ different orientation strategies (constraint-based or score-based).

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* (Definition 2). To resolve these *asymmetric edges*, existing local-to-global 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 alleviate the impact of data problems on the local-to-global CSL by data sampling technique combined with progressive learning strategy.

III. Preliminaries

A. Notations and Definitions

Let $\boldsymbol{V}$=$\{X_1, X_2, \ldots, X_m\}$ denote a set of random variables. The (global) causal structure over $\boldsymbol{V}$ is often represented using a causal graph. The most commonly used graphical representation of a causal structure over $\boldsymbol{V}$ is a causal directed acyclic graph (DAG) [42], denoted as $\mathbb{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{E} \subseteq \boldsymbol{V} \times \boldsymbol{V}$ is the set of directed edges each representing potential causal relationships between a pair of variables in $\boldsymbol{V}$, and $\mathbb{G}$ contains no cycles. Specifically, an edge $X_i \rightarrow X_j$ in a DAG represents that X_i is a direct cause (i.e. parent) of X_j and X_j is a direct effect (i.e.

child) of X_i. The (global) skeleton of a causal structure over $\boldsymbol{V}$, denoted as S in this paper, is represented as an undirected graph containing $\boldsymbol{V}$ and undirected edges between variables in $\boldsymbol{V}$. When using a DAG to represent a causal structure, causal structure learning (CSL) is to learn a DAG over a set of variables $\boldsymbol{V}$ from observational data. The conditional independence/dependence of two variables is defined as follows.

Definition 1 (Conditional Independence): Two variables X_i and X_j are conditionally independent given a variable set $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X_i, X_j\}$ if $P(X_i, X_j|\mathbf{Z}) = P(X_i|\mathbf{Z})P(X_j|\mathbf{Z})$ holds; otherwise, X_i and X_j are conditionally dependent given $\mathbf{Z}$.

Proposition 1: [42] In a causal DAG, if there is a direct 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 Z.

Proposition 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 subsets of the other variables. Proposition 1 is the rationale for all existing constraint-based local skeleton learning methods to learn the local skeleton of a variable.

To enhance the effectiveness and efficiency of global CSL approaches, local-to-global CSL approaches have been developed. However, as illustrated in Figs. 1 and 2, when these methods utilize learned local skeletons to construct a global skeleton, *asymmetric edges* arise, defined as follows.

Definition 2 (Asymmetric Edge): With a local skeleton learning algorithm, when learning the local skeletons of two variables X_d and X_f, if X_d's local skeleton contains X_f but X_f's local skeleton does not contain X_d (or vice versa), we say that an asymmetric edge $X_d \nleftrightarrow X_f$ is formed between X_d and X_f.

B. Bootstrap Method

In this paper, we propose to use the Bootstrap method (also called Bootstrapping) [43] to overcome the quality issue of the original dataset. Thus, in this section, we provide the necessary background of the Bootstrap method.

Bootstrapping is a sampling method often used in the field of machine learning. Given an original dataset D_{orig}, the process of sampling through the Bootstrap method for generating a sub-dataset D_j is as follows:

- Randomly select an instance from D_{orig} each time and put it into D_j, and then put the instance back into the original dataset D_{orig}, so that the instance may still be sampled in the next sampling.
- Repeat the above procedure n times to create a sub-dataset D_j that contains n instances, where n is equal to the number of instances in D_{orig}.

According to the sampling process of the Bootstrap method, we can get Proposition 2.

Proposition 2 ([43]): Given an original dataset D_{orig} with n instances for generating a sampled dataset D_j by the Bootstrap method, if $n \to \infty$, roughly 36.8% of the instances in D_{orig} do not appear in D_j.

In this paper, we use D to denote a single dataset with m variables and n samples, and $\mathcal{D}$ to represent a set/batch of datasets with the same dimensionality and number of samples.

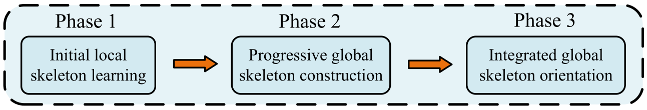

Phase 1
Initial local skeleton learning
Phase 2
Progressive global skeleton construction
Phase 3
Integrated global skeleton orientation

Fig. 3. The framework of our proposed PCSL method.

IV. The Proposed PCSL Algorithm

To alleviate the adverse effects of *asymmetric edges* (Definition 2) on the local-to-global causal structure learning methods, in this section, we present our proposed PCSL method for local-to-global causal structure learning.

As shown in Fig. 3, PCSL consists of three phases. First, in Phase 1, PCSL discovers the initial local skeleton (i.e. parents and children (PC) set) of each variable in a dataset. Then, based on the local skeletons obtained in Phase 1, Phase 2 applies a progressive enhancement method to continuously correct *asymmetric edges* for constructing the best global skeleton. Finally, Phase 3 uses an integrated strategy to orient the undirected edges in the global skeleton, and obtains the best causal structure. The novel contribution of PCSL is the progressive global skeleton construction strategy (i.e. Phase 2) and the integrated global skeleton orientation strategy (i.e. Phase 3). In the following, we describe the three phases in detail in Section IV-A, IV-B, and IV-C, respectively.

A. Initial Local Skeleton Learning

Learning the local skeleton of each variable independently can make PCSL not only scalable to high-dimensional data (please refer to the time complexity analysis in Section S-10 of the Supplementary Material), but also avoid cascading errors encountered in structure learning as much as possible.

Given an original dataset D_{orig} with the variable set $\boldsymbol{V}=\{X_1, X_2, \ldots, X_m\}$ and n samples, in Phase 1, PCSL uses an existing PC (Parent-Child) learning algorithm to discover the PC set of each variable on D_{orig}. In our implementation, we employ HITON-PC [36], one widely used PC learning algorithm, for this phase, and the rationale of the HITON-PC algorithm has been presented in Proposition 1.

Let $\boldsymbol{PC}(X_d)$ denote the learned PC set of variable X_d ($d \in [1, m]$), and at the end of Phase 1, we obtain the local skeleton of all variables, i.e., $\boldsymbol{PC}(X_1), \boldsymbol{PC}(X_2), \ldots, \boldsymbol{PC}(X_m)$. According to Proposition 1, a PC learning algorithm can discover the true PC set of a target variable theoretically. Hence in theory, there is the following property.

Property 1 (Symmetry): If X_d is in the learned PC set of X_f, X_f must be in the learned PC set of X_d.

However, as shown in Fig. 2(a), the HITON-PC algorithm (or other existing local skeleton learning methods) often yields some *asymmetric edges* (Definition 2) due to data quality issues (e.g., data insufficiency for CI tests and noises). To determine whether each *asymmetric edge* exists in the true global skeleton, we design a progressive global skeleton construction strategy as follows.

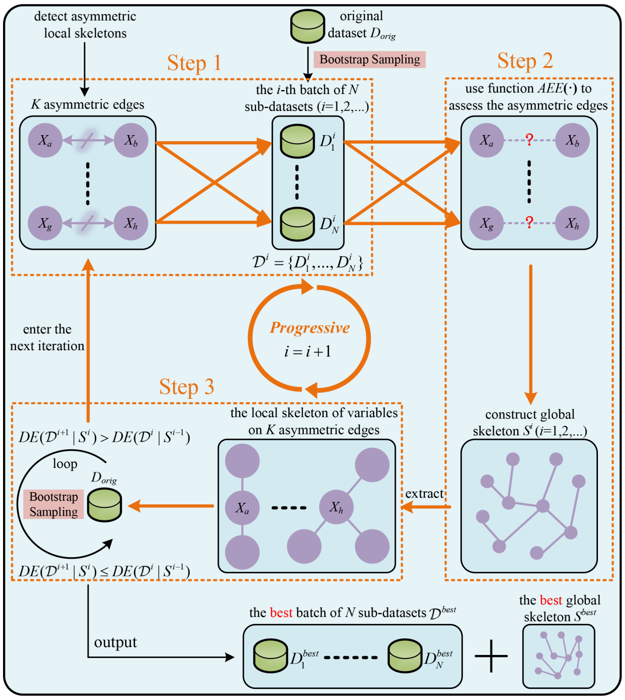

detect asymmetric local skeletons
original dataset D_{orig}
Bootstrap Sampling
Step 1
Step 2
K asymmetric edges
the i-th batch of N sub-datasets (i=1,2,...)
use function $AEE(\cdot)$ to assess the asymmetric edges
X_a
X_b
X_g
X_h
D_1^i
D_N^i
?
$\mathcal{D}^i = \{D_1^i, \ldots, D_N^i\}$
enter the next iteration
Progressive
$i = i + 1$
Step 3
the local skeleton of variables on K asymmetric edges
construct global skeleton S^i (i=1,2,...)
$DE(\mathcal{D}^{i+1} \mid S^i) > DE(\mathcal{D}^i \mid S^{i-1})$
loop
D_{orig}
Bootstrap Sampling
extract
$DE(\mathcal{D}^{i+1} \mid S^i) \leq DE(\mathcal{D}^i \mid S^{i-1})$
the best batch of N sub-datasets $\mathcal{D}^{best}$
the best global skeleton S^{best}
output
D_1^{best}
D_N^{best}
+

Fig. 4. The progressive global skeleton construction strategy of PCSL.

B. *Progressive Global Skeleton Construction*

In this section, we focus on describing the progressive global skeleton construction strategy of PCSL (see Fig. 4 for details). Using the local skeletons (i.e. PC sets) obtained at Phase 1, Phase 2 is to construct the global skeleton by splicing all local skeletons.

Specifically, if local skeletons between variables are symmetric, such as $X_d \in \boldsymbol{PC}(X_f)$ and $X_f \in \boldsymbol{PC}(X_d)$ (or $X_d \notin \boldsymbol{PC}(X_f)$ and $X_f \notin \boldsymbol{PC}(X_d)$), we believe that there is (not) an edge between X_d and X_f; otherwise, an *asymmetric edge* $X_d \nleftrightarrow X_f$ is formed between X_d and X_f. According to the learned PC set of each variable in $\boldsymbol{V}$, PCSL records all *asymmetric edges*, and we use K to denote the total number of *asymmetric edges*. To deal with these *asymmetric edges*, we design a progressive global skeleton construction strategy with the following three iterative steps, illustrated in Fig. 4.

Step 1: Relearn the local skeleton of variables on each asymmetric edge **.** In this step, we first generate a batch of sub-datasets by sampling from the original dataset D_{orig} and then learn the local skeletons of the variables on each *asymmetric edge* again on each of the sampled datasets.

In reality, the available (original) datasets often contain a limited number of samples and may include some noise, leading to that the local-to-global CSL algorithms often find some extra edges and lose some true edges during global skeleton construction. To overcome the quality limitation of the original dataset, the Bootstrap method [43] (see Section III-B for details) is a good choice to achieve effective CSL.

Based on Bootstrap, the original dataset D_{orig} containing n samples and m variables can be sampled into multiple batch sub-datasets. Let the number of batches be L, and the number of datasets in each batch of sub-datasets be N. Further, the i-th batch of sub-datasets is marked as $\mathcal{D}^i = \{D_1^i, D_2^i, \ldots, D_N^i\}$ ($i = 1, 2, \ldots, L$). Following the Bootstrap method, each sub-dataset in $\mathcal{D}^i$ contains n samples and m variables. Note that the existing Bootstrap-based methods usually need to generate hundreds of sub-datasets (i.e., N is large) to train a large number of models for achieving stable performance, and this is a very time-consuming process. In contrast, on each batch of sub-datasets,[2] our method only needs a small N to achieve reliable performance by adopting a novel weighting strategy, see Step 2 for details).

According to Proposition 2, for causal structure learning, generating sub-datasets through the Bootstrap method has the following advantages:

- 36.8% of the samples between D_{orig} and a sampled sub-dataset are different. The sample difference increases the diversity of the causal structures learned from different sub-datasets to a certain extent.
- Bootstrap keeps a sampled sub-dataset have the same sample size as the original dataset D_{orig}. Therefore, the sampled sub-datasets will not significantly reduce the reliability of the statistical tests, such as conditional independence (CI) tests. Specifically, to perform a reliable CI test between variables X_d and X_f conditioning on a variable set $\boldsymbol{S}$ ($\boldsymbol{S} \subset \boldsymbol{V} \setminus \{X_d, X_f\}$), the average number of samples per cell of the contingency table of $\{X_d, X_f\} \cup \boldsymbol{S}$ must be at least t [44]:

$$\frac{n}{C_{X_d} \times C_{X_f} \times C_{\boldsymbol{S}}} \geq t, \tag{1}$$

where n denotes the number of samples in a dataset, and t is a constant; given a discrete dataset, C_{X_d}, C_{X_f} and $C_{\boldsymbol{S}}$ denote the number of categories of values that X_d, X_f and the variables in $\boldsymbol{S}$ (jointly) take, respectively. Compared with D_{orig}, ($C_{X_d} \times C_{X_f} \times C_{\boldsymbol{S}}$) in a sampled sub-dataset often remains unchanged.

Subsequently, given the datasets in $\mathcal{D}^i$ (initially $i = 1$), PCSL relearns the local skeleton (i.e., PC set) of variables on each *asymmetric edge* again. For example, as shown in Fig. 4, "$X_a \nleftrightarrow X_b$" is an *asymmetric edge*, thus PCSL needs to discover the local skeleton of X_a and the local skeleton of X_b again on all sub-datasets (i.e., $D_1^i, D_2^i, \ldots, D_N^i$).

Step 2: Correct each asymmetric edge **.** We design a scoring function to determine whether an *asymmetric edge* should be in the global skeleton, and then construct a more accurate global skeleton by excluding those erroneous *asymmetric edges*. The global skeleton constructed by correcting the *asymmetric edges* on the i-th batch of sub-datasets (i.e., $\mathcal{D}^i$) is marked as S^i. In particular, S^0 denotes the initial local skeleton set of all variables learned on D_{orig}.

For each *asymmetric edge*, through combining the local skeleton learning results from all sub-datasets at Step 1, PCSL designs

[2] Even the total number of sub-datasets ($N \times L$) required for our method would not exceed 105, since N is set to 15 and $L \leq 7$ in our experiments, see Section V for details.

a scoring function, AEE (Asymmetric Edge Evaluation), to determine whether this edge exists.

Specifically, given the k-th *asymmetric edge* ($k = 1, 2, \ldots, K$) containing variables X_d and X_f ($d, f \in [1, m]$ and $d \neq f$), the score of the k-th *asymmetric edge* on the j-th sub-dataset is formalized as

$$\underset{\substack{j=1,2,\ldots,N\\k=1,2,\ldots,K}}{AEE(j,k)} = \begin{cases} 1+1=2 & if\ X_f \in \boldsymbol{PC}(X_d)\ and\ X_d \in \boldsymbol{PC}(X_f) \\ -1-1=-2 & if\ X_f \notin \boldsymbol{PC}(X_d)\ and\ X_d \notin \boldsymbol{PC}(X_f) \\ 1-1=0 & if\ X_f \in \boldsymbol{PC}(X_d)\ and\ X_d \notin \boldsymbol{PC}(X_f) \\ -1+1=0 & if\ X_f \notin \boldsymbol{PC}(X_d)\ and\ X_d \in \boldsymbol{PC}(X_f). \end{cases} \tag{2}$$

That is, if the learned local skeleton of X_d contains X_f, then $AEE(j,k)$ plus one point; otherwise $AEE(j,k)$ minus one point. Similarly, if the learned local skeleton of X_f contains X_d, then $AEE(j,k)$ plus one point; otherwise $AEE(j,k)$ minus one point.

However, due to the randomness of the sampling, the quality of the sampled sub-datasets in $\mathcal{D}^i$ is different. For example, for the PC learning of X_d, the quality of the learned PC set using D_2^i may be higher than that using D_1^i. Further, the parameter N in our method usually takes a small value for maintaining efficiency. Therefore, to obtain stable performance with a small N, we should adjust (weight) the score of an *asymmetric edge* on each sub-dataset according to the quality of this sub-dataset. For a variable X_d on the k-th *asymmetric edge*, our method uses the F1 score [45] of the PC set of X_d learned on the generated sub-dataset D_j^i to measure the reliability of the result of PC learning for X_d on D_j^i, i.e., the higher the value of F1 score, the higher the reliability of PC learning for X_d on D_j^i. Unfortunately, the true PC set of D_j^i is unknown. Instead, we regard the PC set of X_d in S^{i-1}, the global skeleton obtained at the end of iteration i-1, as a reference for the calculation of F1 score, and the reasons for this are as follows.

From Fig. 4, it is not difficult to see that Phase 2 of PCSL is a progressive iterative process. Before the end of Step 2 of the i-th iteration, S^{i-1} is the latest global skeleton, and the accuracy of S^{i-1} is higher than that of S^{i-2} (A detailed analysis will be provided at the end of this subsection). Thus, we measure the reliability (stability) of PC learning result for X_d on D_j^i by comparing the PC set of X_d learned on D_j^i with the PC set of X_d in S^{i-1}. Let $Q(D_j^i, k, X_d)$ denote the reliability of PC learning result for X_d (on the k-th *asymmetric edge*) on D_j^i. Based on the definition of F1 score, $Q(D_j^i, k, X_d)$ is formalized as

$$\begin{aligned} Q(D_j^i,k,X_d) &= \frac{2*\frac{TP}{TP+FP}*\frac{TP}{TP+FN}}{\frac{TP}{TP+FP}+\frac{TP}{TP+FN}} \\ &= \frac{2*\frac{|\boldsymbol{PC}(X_d|D_j^i)\cap\boldsymbol{PC}(X_d|S^{i-1})|}{|\boldsymbol{PC}(X_d|D_j^i)|}*\frac{|\boldsymbol{PC}(X_d|D_j^i)\cap\boldsymbol{PC}(X_d|S^{i-1})|}{|\boldsymbol{PC}(X_d|S^{i-1})|}}{\frac{|\boldsymbol{PC}(X_d|D_j^i)\cap\boldsymbol{PC}(X_d|S^{i-1})|}{|\boldsymbol{PC}(X_d|D_j^i)|}+\frac{|\boldsymbol{PC}(X_d|D_j^i)\cap\boldsymbol{PC}(X_d|S^{i-1})|}{|\boldsymbol{PC}(X_d|S^{i-1})|}}, \end{aligned} \tag{3}$$

where TP, FP, and FN denote the number of true positives, false positives, and false negatives, respectively; further, $|\cdot|$ denotes the size of a set, and $\boldsymbol{PC}(X_d|D_j^i)$ and $\boldsymbol{PC}(X_d|S^{i-1})$ ($i \in [1, L], j \in [1, N], d \in [1, m]$) represent the PC set of X_d learned from D_j^i and the PC set of X_d in the skeleton S^{i-1}, respectively. Here we regard the PC set of X_d in S^{i-1} as a reference PC set of X_d. In particular, when $i = 1$, we take the PC sets learned on D_{orig} as the reference PC sets. On the one hand, we are eager to generate the sub-datasets that deviate from the original data distribution for alleviating the quality issue of D_{orig}; on the other hand, we should not generate sub-datasets that deviate too much from the original data distribution.

Thus, when $i = 1$, the greater the difference between the distribution of the generated sub-datasets and that of D_{orig}, the lower the reliability of the PC learning results on these sub-datasets. Further, our method assumes that the symmetric edges learned are correct, and in fact, the majority edges learned are symmetric. Therefore, the better the quality of PC learning on D_j^i is, the more symmetric edges should be kept in the PC learned from D_j^i. It means if $\boldsymbol{PC}(X_d|D_j^i)$ and $\boldsymbol{PC}(X_d|D_{orig})$ are more similar, i.e., $Q(D_j^i, k, X_d)$ is larger, the reliability of PC learning result for X_d on D_j^i is higher.

When $Q(D_j^i, k, X_d)$ is considered, the score of the k-th *asymmetric edge* on the j-th sub-dataset can be reformulated as (4) shown at the bottom of this next page. Finally, based on (4), we can obtain the total score of the k-th *asymmetric edge* on N sub-datasets as

$$\underset{k=1,2,\ldots,K}{AEE(:,k)} = \sum_{j=1}^{N} AEE(j,k). \tag{5}$$

According to (5), we propose the following Criterion 1 to determine whether an *asymmetric edge* should be in the final global skeleton.

Criterion 1: if $AEE(:,k) > 0$, the k-th *asymmetric edge* will be retained in the global skeleton; otherwise, this edge will be removed.

By applying Criterion 1, PCSL can assess whether K *asymmetric edges* exist in the underlying causal structure behind the original dataset D_{orig} and construct a more accurate global skeleton S^i.

However, in rare cases, $AEE(:,k)$ 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 $AEE(:,k) = 0$, PCSL introduces a weight factor w to enlarge the influence of the reliability of learning results (i.e., $Q(D_j^i, k, X_d)$) on the score $AEE(\cdot)$ and have reliability score $\hat{Q}(D_j^i, k, X_d)$ as follows:

$$\begin{aligned} &\hat{Q}(D_j^i,k,X_d) \\ &= \begin{cases} Q(D_j^i,k,X_d)*w & if\ Q(D_j^i,k,X_d) > Q(D_j^i,k,X_f) \\ Q(D_j^i,k,X_d) & otherwise \end{cases} \end{aligned} \tag{6}$$

$$\hat{Q}(D_j^i,k,X_f) = \begin{cases} Q(D_j^i,k,X_f)*w & if\ Q(D_j^i,k,X_f) > Q(D_j^i,k,X_d) \\ Q(D_j^i,k,X_f) & otherwise \end{cases} \tag{7}$$

In (6) and (7), "$w > 1.0$" means that the score gap of two variables on an *asymmetric edge* is further enlarged. Specifically, given a sub-dataset D_j^i and the k-th *asymmetric edge* containing X_d and X_f, in (6), the reliability of PC learning result for X_d on D_j^i will be further enlarged if the reliability of PC learning result for X_d on D_j^i is higher than that for X_f on D_j^i. Similarly, in (7), the reliability of PC learning result for X_f on D_j^i will be further enlarged if the reliability of PC learning result for X_f on D_j^i is higher than that for X_d on D_j^i. The weight factor w is initially set to 1.0. When $AEE(\cdot) = 0$, w will be automatically enlarged by any multiple (such as 1.5 times) for making $AEE(\cdot) = 0$ no longer hold. Then, by combining (6) and (7) with (4), $AEE(j,k)$ can be reformulated as (8) shown at the bottom of this page.

Step 3: Generate higher quality sub-datasets guided by the newly learned skeleton: After obtaining a more accurate global skeleton, the reference skeleton used in (3) should be replaced with the newly learned skeleton, since compared with S^{i-1} ($i \in [1,L]$), S^i is much closer to the true global skeleton. Thus, the reliability of PC learning results calculated by (3) is more faithful when S^i is considered the reference skeleton rather than S^{i-1}. This step aims to generate a batch of sub-datasets with higher quality based on the current most accurate skeleton S^i.

To this end, PCSL designs a scoring function, DE (Dataset Evaluation), to calculate the average quality of each batch of sub-datasets. First, based on (3), we can calculate the quality of a single sub-dataset. Specifically, given a sub-dataset D_j^i, the quality of D_j^i can be expressed as the average reliability of all PC learning results (for all variables on all *asymmetric edges*) obtained on D_j^i. Let $AvgQ(D_j^i)$ denote the average reliability of all PC learning results on D_j^i, and it is formalized as

$$\underset{i=1,2,\ldots;j=1,2,\ldots,N}{AvgQ(D_j^i)} = \frac{\sum_{k=1}^{K}(Q(D_j^i,k,X_d)+Q(D_j^i,k,X_f))}{2*K}, \tag{9}$$

where X_d and X_f ($d, f \in [1,m]$ and $d \neq f$) are the two variables on the k-th *asymmetric edge* ($k = 1,2,\ldots,K$).

Since the calculation of $Q(D_j^i,k,X_d)$ in (3) and $AvgQ(D_j^i)$ in (9) takes S^{i-1} as the reference skeleton, we let $DE(\mathcal{D}^i|S^{i-1})$ denote the average quality of all sub-datasets in $\mathcal{D}^i = \{D_1^i, D_2^i, \ldots, D_N^i\}$ with S^{i-1} as the reference skeleton, and its formulation is as follows:

$$\begin{aligned}\underset{i=1,2,\ldots}{DE(\mathcal{D}^i|S^{i-1})} &= \frac{\sum_{j=1}^{N} AvgQ(D_j^i)}{N} \\ &= \frac{\sum_{k=1}^{K}\sum_{j=1}^{N}(Q(D_j^i,k,X_d)+Q(D_j^i,k,X_f))}{2*K*N}.\end{aligned} \tag{10}$$

$$\underset{\substack{j=1,2,\ldots,N\\k=1,2,\ldots,K}}{AEE(j,k)} = \begin{cases} \begin{aligned}&1*Q(D_j^i,k,X_d)\\&+1*Q(D_j^i,k,X_f).\end{aligned} & if\ X_f \in \boldsymbol{PC}(X_d) \wedge X_d \in \boldsymbol{PC}(X_f) \\ \begin{aligned}&(-1)*Q(D_j^i,k,X_d)\\&+(-1)*Q(D_j^i,k,X_f).\end{aligned} & if\ X_f \notin \boldsymbol{PC}(X_d) \wedge X_d \notin \boldsymbol{PC}(X_f) \\ \begin{aligned}&1*Q(D_j^i,k,X_d)\\&+(-1)*Q(D_j^i,k,X_f).\end{aligned} & if\ X_f \in \boldsymbol{PC}(X_d) \wedge X_d \notin \boldsymbol{PC}(X_f) \\ \begin{aligned}&(-1)*Q(D_j^i,k,X_d)\\&+1*Q(D_j^i,k,X_f).\end{aligned} & if\ X_f \notin \boldsymbol{PC}(X_d) \wedge X_d \in \boldsymbol{PC}(X_f). \end{cases} \tag{4}$$

$$\underset{\substack{j=1,2,\ldots,N\\k=1,2,\ldots,K}}{AEE(j,k)} =$$

$$\begin{cases} 1*Q(D_j^i,k,X_d)*w+1*Q(D_j^i,k,X_f) & if\ X_f \in \boldsymbol{PC}(X_d),\ X_d \in \boldsymbol{PC}(X_f),\ and\ Q(D_j^i,k,d) > Q(D_j^i,k,f) \\ 1*Q(D_j^i,k,X_d)+1*Q(D_j^i,k,X_f) & if\ X_f \in \boldsymbol{PC}(X_d),\ X_d \in \boldsymbol{PC}(X_f),\ and\ Q(D_j^i,k,d) = Q(D_j^i,k,f) \\ 1*Q(D_j^i,k,X_d)+1*Q(D_j^i,k,X_f)*w & if\ X_f \in \boldsymbol{PC}(X_d),\ X_d \in \boldsymbol{PC}(X_f),\ and\ Q(D_j^i,k,d) < Q(D_j^i,k,f) \\ (-1)*Q(D_j^i,k,X_d)*w+(-1)*Q(D_j^i,k,X_f) & if\ X_f \notin \boldsymbol{PC}(X_d),\ X_d \notin \boldsymbol{PC}(X_f),\ and\ Q(D_j^i,k,d) > Q(D_j^i,k,f) \\ (-1)*Q(D_j^i,k,X_d)+(-1)*Q(D_j^i,k,X_f) & if\ X_f \notin \boldsymbol{PC}(X_d),\ X_d \notin \boldsymbol{PC}(X_f),\ and\ Q(D_j^i,k,d) = Q(D_j^i,k,f) \\ (-1)*Q(D_j^i,k,X_d)+(-1)*Q(D_j^i,k,X_f)*w & if\ X_f \notin \boldsymbol{PC}(X_d),\ X_d \notin \boldsymbol{PC}(X_f),\ and\ Q(D_j^i,k,d) < Q(D_j^i,k,f) \\ 1*Q(D_j^i,k,X_d)*w+(-1)*Q(D_j^i,k,X_f) & if\ X_f \in \boldsymbol{PC}(X_d),\ X_d \notin \boldsymbol{PC}(X_f),\ and\ Q(D_j^i,k,d) > Q(D_j^i,k,f) \\ 1*Q(D_j^i,k,X_d)+(-1)*Q(D_j^i,k,X_f) & if\ X_f \in \boldsymbol{PC}(X_d),\ X_d \notin \boldsymbol{PC}(X_f),\ and\ Q(D_j^i,k,d) = Q(D_j^i,k,f) \\ 1*Q(D_j^i,k,X_d)+(-1)*Q(D_j^i,k,X_f)*w & if\ X_f \in \boldsymbol{PC}(X_d),\ X_d \notin \boldsymbol{PC}(X_f),\ and\ Q(D_j^i,k,d) < Q(D_j^i,k,f) \\ (-1)*Q(D_j^i,k,X_d)*w+1*Q(D_j^i,k,X_f) & if\ X_f \notin \boldsymbol{PC}(X_d),\ X_d \in \boldsymbol{PC}(X_f),\ and\ Q(D_j^i,k,d) > Q(D_j^i,k,f) \\ (-1)*Q(D_j^i,k,X_d)+1*Q(D_j^i,k,X_f) & if\ X_f \notin \boldsymbol{PC}(X_d),\ X_d \in \boldsymbol{PC}(X_f),\ and\ Q(D_j^i,k,d) = Q(D_j^i,k,f) \\ (-1)*Q(D_j^i,k,X_d)+1*Q(D_j^i,k,X_f)*w & if\ X_f \notin \boldsymbol{PC}(X_d),\ X_d \in \boldsymbol{PC}(X_f),\ and\ Q(D_j^i,k,d) < Q(D_j^i,k,f). \end{cases} \tag{8}$$

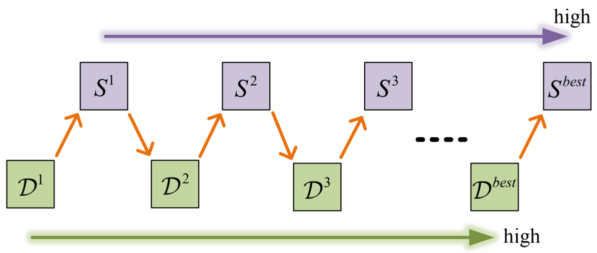


Fig. 5. Progressive data sampling and progressive skeleton learning.

Then, through the Bootstrap method, PCSL iteratively samples the sub-datasets from D_{orig} until a batch of the sub-datasets makes $DE(\mathcal{D}^{i+1}|S^i) > DE(\mathcal{D}^i|S^{i-1})$ hold. We believe that the average data quality of $\mathcal{D}^{i+1}$ is higher than that of $\mathcal{D}^i$. Subsequently, PCSL selects a batch of sub-datasets with higher average quality as the next batch of sub-datasets, i.e., the (i+1)-th batch of sub-datasets $\mathcal{D}^{i+1} = \{D_1^{i+1}, D_2^{i+1}, \ldots, D_N^{i+1}\}$.

When a batch of sub-datasets with higher average quality cannot be generated, Phase 2 of PCSL will terminate. In our experiments, if the newly sampled sub-datasets satisfy $DE(\mathcal{D}^{i+1}|S^i) \leq DE(\mathcal{D}^i|S^{i-1})$ for r consecutive times,[3] we believe that both $\mathcal{D}^i$ and S^i are already the best, and PCSL outputs the best batch of N sub-datasets $\mathcal{D}^{best}$ and the best global skeleton S^{best}.

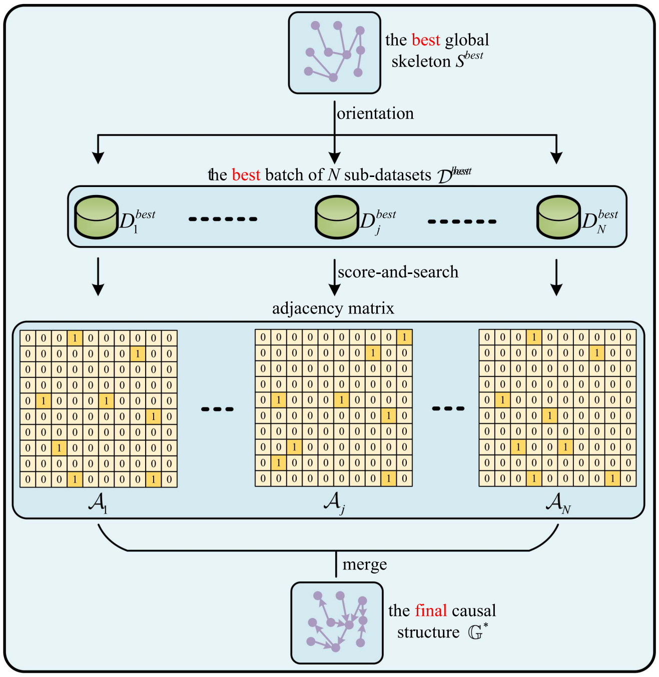
Implement Step 1 to Step 3 as a progressive iterative process: As shown in Fig. 4, after obtaining a batch of higher quality sub-datasets $\mathcal{D}^{i+1}$, it will be re-entered into Step 1 as a new batch of datasets to further improve the correction accuracy of *asymmetric edges*.

With the iteration of Steps 1 to 3, the performance of PCSL (in global skeleton learning) improves progressively by interleavingly increasing the quality of the sampled sub-datasets and learned skeletons, as indicated in Fig. 5. In the following, we will give an effectiveness analysis of this progressive strategy.

Theorem 1: Let L denote the number of iterations in Phase 2. In each iteration $i \in 1, 2, \ldots, L$, S^i denotes the constructed global skeleton, and $\mathcal{D}^i$ represents the set of sampled sub-datasets used. Assuming that (3) accurately measures the quality of each sub-dataset, the progressive strategy employed by PCSL is effective, satisfying the following properties:

1) The accuracy of the global skeleton S^{i+1} is higher than that of S^i for all $i \in 1, 2, \ldots, L-1$.
2) The quality of the sub-datasets $\mathcal{D}^{i+1}$ is better than that of $\mathcal{D}^i$ for all $i \in 1, 2, \ldots, L-1$.

The proof of Theorem 1 is given in Section S-1in the Supplementary Material. According to Theorem 1, a more accurate global skeleton supervises PCSL to continuously generate the sampled datasets with higher quality, and meanwhile, a batch of higher quality sub-datasets supervises PCSL to continuously construct more accurate global skeletons. Thus, Phase 2 of PCSL progressively improves the quality of $\mathcal{D}^i$, and enhances the correction accuracy of *asymmetric edges* for obtaining a more accurate global skeleton S^i. As the proportion of *asymmetric edges* in each iteration becomes lower and lower, Phase 2 will tend to converge.


the best global skeleton S^{best}
orientation
the best batch of N sub-datasets $\mathcal{D}^{best}$
D_1^{best}
D_j^{best}
D_N^{best}
score-and-search
adjacency matrix
merge
the final causal structure $\mathbb{G}^*$

$$\mathcal{A}_1 = \begin{bmatrix} 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&1&0 \end{bmatrix} \quad \cdots \quad \mathcal{A}_j = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0 \end{bmatrix} \quad \cdots \quad \mathcal{A}_N = \begin{bmatrix} 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&1&0 \end{bmatrix}$$

Fig. 6. The integrated global skeleton orientation strategy of PCSL.

C. Integrated Global Skeleton Orientation

After having obtained S^{best} and $\mathcal{D}^{best}$, PCSL moves to Phase 3 to orient the edges in S^{best}. As shown in Fig. 6, the integrated global skeleton orientation strategy consists of the following two steps.

Step 1: Orient the global skeleton independently on each sub-dataset: On each sub-dataset D_j^{best} ($j \in [1, N]$) separately, PCSL uses a Bayesian score criteria, BDeu [46], and a search procedure, hill-climbing [40] to greedily orient the undirected edges in S^{best}. Here, the BDeu score for the graph structure $\mathbb{G}_j$ learned on dataset D_j^{best} is defined as (11):

$$BDeu(\mathbb{G}_j, D_j^{best}) = \log P(\mathbb{G}_j) + \sum_{i=1}^{m}\sum_{l=1}^{q_i}\left[\log\frac{\Gamma(\frac{H'}{q_i})}{\Gamma(H_{il}+\frac{H'}{q_i})} + \sum_{u=1}^{r_i}\log\frac{\Gamma(H_{ilu}+\frac{H'}{r_i q_i})}{\Gamma(\frac{H'}{r_i q_i})}\right], \tag{11}$$

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 variable X_i, and u is the index of the r_i possible values (states) of X_i; further, H_{ilu} is the number of instances in D_j^{best} where X_i has the u-th value, and its parents have the l-th combination of values, and $H_{il} = \sum_{u=1}^{r_i} H_{ilu}$; H' is the equivalent sample size (ESS, also sometimes known as the imaginary sample size, ISS) representing the confidence level in the prior parameters; $P(\mathbb{G}_j)$ is the prior probability

[3] We utilize a specified constant r to limit the times of tolerance and its value is set to 3in our experiments. The sensitivity of parameter r is analyzed in Section S-7 of the Supplementary Material.

Algorithm 1: Integration of Causal Structures.

```
   Input: $\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_N$: $N$ adjacency matrices
   Output: $\mathbb{G}^*$: the final causal structure
1  Let $\mathcal{A}^* = zeros(m, m)$ /*an empty graph*/;
2  $\mathcal{A}^* = \dfrac{\mathcal{A}_1 \oplus \mathcal{A}_2 \oplus \cdots \oplus \mathcal{A}_N}{N}$ /*integration*/;
3  for $a$=1 to $m$ do
4      for $b$=1 to $m$ do
5          if $\mathcal{A}^*(a, b) \geq 0.5$ then
6              $\mathcal{A}^*(a, b) = 1$ /*$X_a \rightarrow X_b$*/;
7          else
8              $\mathcal{A}^*(a, b) = 0$ /*$X_a \nrightarrow X_b$*/;
9          end
10     end
11 end
12 $\mathbb{G}^* = acyclic\_constraint(\mathcal{A}^*)$;
13 return $\mathbb{G}^*$
```

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, PCSL gets a global causal structure with the highest scoring on D_j^{best}. To facilitate the merger of all the causal structures $\mathbb{G}_1$, $\mathbb{G}_2, \ldots, \mathbb{G}_N$ with their edges oriented as described above, we use adjacency matrices to represent them. Let $\mathcal{A}_j$ denote the adjacency matrix of $\mathbb{G}_j$, and "$\mathcal{A}_j(5, 2) = 1$" denotes that there is an edge from X_5 to X_2 in $\mathbb{G}_j$.

Step 2: Merge all adjacency matrices to form the final causal structure: In this step, PCSL aims to integrate all adjacency matrices (i.e., $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_N$) learned in Step 1 for obtaining the final causal structure (marked as $\mathbb{G}^*$), and Algorithm 1 describes the details of Step 2.

1) *Initialization (Line 1):* Create an $m \times m$ zero matrix $\mathcal{A}^*$ (an empty graph). Here, m is the number of variables.
2) *Integration Process (Line 2):* Perform element-wise addition (denoted by $\oplus$) of all N adjacency matrices, then divide by N to obtain the average. This step synthesizes information from all learned causal structures.
3) *Edge Determination (Lines 3–11):* Iterate through each element $\mathcal{A}^*(a, b)$ of $\mathcal{A}^*$. If $\mathcal{A}^*(a, b) \geq 0.5$, add the edge $X_a \rightarrow X_b$ to the final graph (set $\mathcal{A}^*(a, b) = 1$); otherwise, do not add this edge (set $\mathcal{A}^*(a, b) = 0$). This process determines the existence of each edge based on a majority voting principle.
4) *Acyclicity Constraint (Line 12):* Call the function $acyclic\_constraint(\mathcal{A}^*)$ to ensure the final graph structure is acyclic. This step may remove some edges to eliminate cycles.
5) *Return Result (Line 13):* Return the final acyclic causal structure $\mathbb{G}^*$.

Note that the causal structure corresponding to $\mathcal{A}^*$, obtained immediately after executing Lines 3-11, may contain bidirectional edges. To address this, we propose Theorem 2, which demonstrates that if N is an odd number, there will be no bidirectional edges in $\mathcal{A}^*$ after completing Lines 3-11.

Theorem 2: Let $\mathcal{A}_j$ ($j \in [1, N]$) be an adjacency matrix used to represent a causal structure (DAG), $\mathcal{A}^* = (\sum_{j=1}^{N} \mathcal{A}_j)/N$ and $\mathcal{A}^*(a, b) \geq 0.5$ ($a, b \in [1, m]$) means that there is an edge from X_a to X_b. If N is an odd number, then there are no bidirectional edges in $\mathcal{A}^*$.

The proof of Theorem 2 is given in Section S-2 in the Supplementary Material. In our experiments, we set N to an odd number for avoiding the occurrence of bidirectional edges. But even then, there may be directed circles in the merged adjacency matrix $\mathcal{A}^*$. Thus, at Line 12, PCSL imposes acyclic constraints [47] on $\mathcal{A}^*$ to ensure the acyclicity of $\mathbb{G}^*$. Due to space limit, the time complexity of PCSL is analyzed in Section S-10 of the Supplementary Material.

D. Difference With BCSL

As a Bootstrap-sampling-based local-to-global CSL algorithm, it is necessary to state the main difference between PCSL and BCSL, another Bootstrap-sampling-based method presented in our conference paper [48]. Compared with BCSL, the strengths and weakness of PCSL are as follows.

1) Strengths: Although both BCSL and PCSL make use of Bootstrap sampling, they are two very different algorithms. The key difference between BCSL and our method lies in the novel and effective progressive learning strategy of PCSL. Specifically, in the global skeleton learning phase, BCSL only samples one single batch of N sub-datasets for the correction of *asymmetric edges*. Although BCSL evaluates the quality of each dataset in this batch of N sub-datasets for adjusting the score of an *asymmetric edge* on each dataset, the average quality of this batch of sub-datasets is constant. Due to the randomness of Bootstrap sampling, BCSL may generate a batch of sub-datasets with low average quality to correct *asymmetric edges*, which seriously deteriorates the performance of BCSL. Whereas in this paper, our proposed method designs a progressive strategy to continuously reform the average quality of each batch of sub-datasets for obtaining a more and more accurate global skeleton. Thus, our method compensates for the performance reduction caused by the randomness of sampling, and achieves more stable performance than BCSL.

In addition, our method improves the deficiencies of BCSL in the global skeleton orientation phase. Specifically, BCSL still orients the global skeleton on the original dataset D_{orig} with data quality issues, leading to unsatisfactory orientation results. In contrast, our method continues to use the best batch of sub-datasets $\mathcal{D}^{best}$ obtained in the skeleton construction phase for the orientation phase, and designs an integrated strategy to achieve a more stable orientation, thus overcoming the impact of data quality problems on the orientation phase.

2) Weakness: Since the global skeleton construction phase of PCSL is an iterative process, PCSL needs to generate multiple batches of sub-datasets, and learn the local skeleton of variables on each *asymmetric edge* multiple times on those sub-datasets. Therefore, the time complexity of PCSL is higher than that of BCSL. Despite this, as shown by the experiments in Section S-4 of the Supplementary Material, the running time of PCSL is not much slower than that of BCSL, and it remains a highly practical algorithm and outperforms numerous existing CSL algorithms across a range of dimensional datasets in terms of speed.

TABLE I
SUMMARY OF BENCHMARK BNS

Network	Child	Child3	Insurance	Insurance3	Alarm	Alarm3
#Variables	20	60	27	81	37	111
#Edges	25	79	52	163	46	149
Network	Child5	Child10	Insurance5	Insurance10	Alarm5	Alarm10
#Variables	100	200	135	270	185	370
#Edges	126	257	284	556	265	570

V. EXPERIMENTS

A. Experiment Setting

1) Comparison Methods: We compare PCSL with five representative local-to-global CSL algorithms, including GSBN [27], GGSL [29], F2SL-c [30], F2SL-s [30] and the previously proposed BCSL [48], and four state-of-the-art global CSL algorithms, including PC-stable [15], NOTEARS [21], DAG-GNN [22] and DAG-NoCurl [25].

2) Evaluation Metrics: We use the Structural Hamming Distance (SHD) and Ar_F1 metrics [48] to evaluate the discovered causal structures. We have also done evaluation with other metrics, please refer to Section S-4 in the Supplementary Material. In all figures and tables, (↑) means the higher the better, (↓) means the lower the better, and the best results are highlighted in bold face.

3) Implementation Details: Implementation details of the PCSL algorithm and the baselines are provided in Section S-3 of the Supplementary Material.

B. Benchmark Datasets

In Section V-B1, we first evaluate our method and its rivals on 12 benchmark BNs, using the datasets provided in existing work [40]. Each BN contains three datasets with 500, 1,000 and 5,000 data samples, respectively. The details of these 12 benchmark BNs are presented in Table I. Then, in Section V-B2, we conduct more experiments on the benchmark datasets to further verify the effectiveness of the progressive strategy adopted by PCSL.

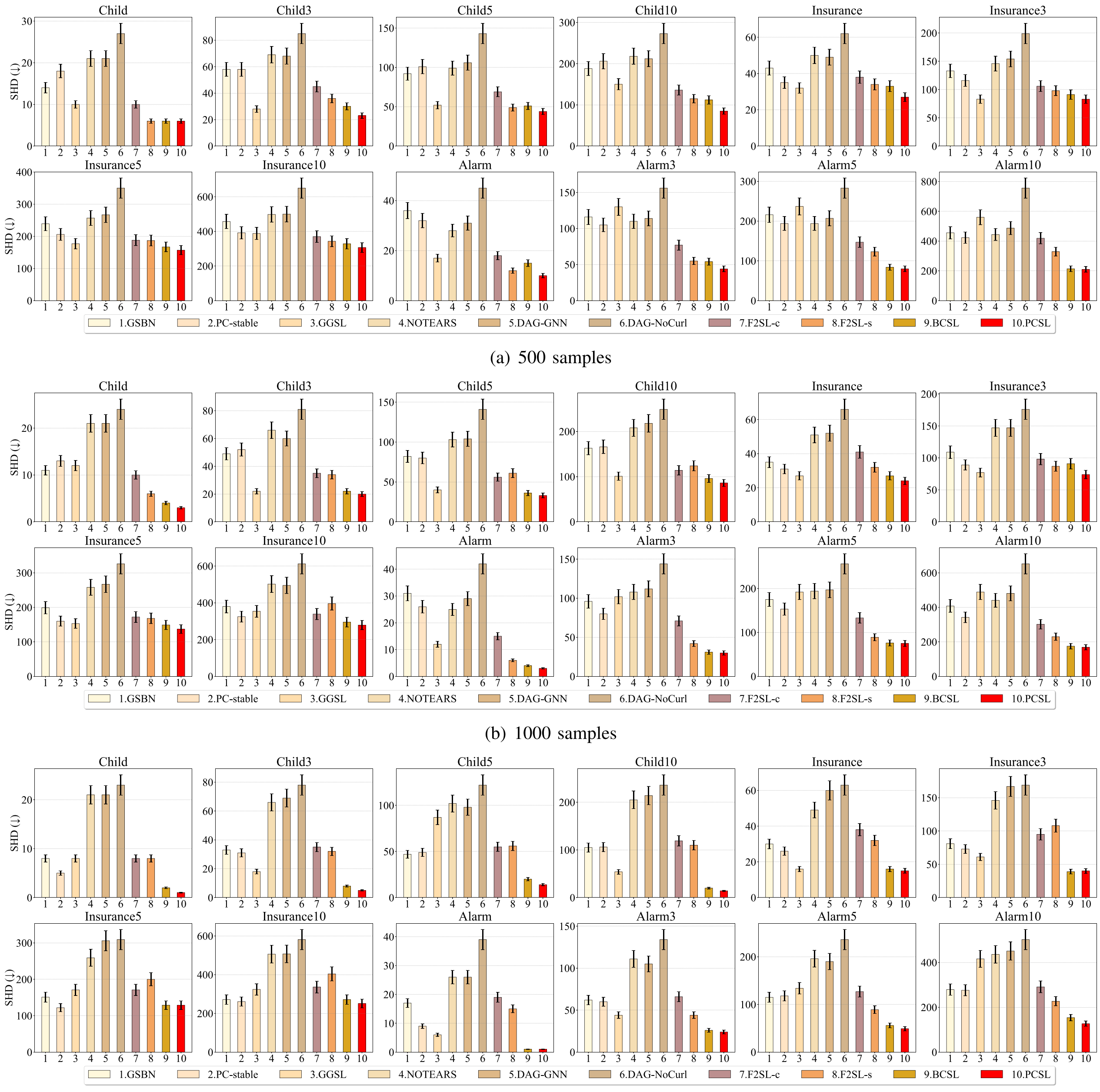
1) Performance Comparison: Figs. 7 and 8 report the quality of the causal structures learned by PCSL and its rivals in terms of SHD and Ar_F1 metrics. Specifically, from Fig. 7, we can see that for almost all benchmark datasets with 500, 1,000 and 5,000 samples, the PCSL algorithm achieves a lower SHD value than the other algorithms, indicating the superiority of our method. The reason is as follows: first, the progressive global skeleton construction strategy of PCSL reduces the number of missing edges and extra edges by constructing an accurate global skeleton; second, the integrated global skeleton orientation strategy of PCSL reduces the number of reverse edges by combining the learning results on the best batch of sampled sub-datasets. Fig. 8 shows that on almost all benchmark datasets with 500, 1,000 and 5,000 samples, PCSL not only achieves fewer structural errors than its rivals in terms of SHD metric, but also discovers more edges correctly than other algorithms based on the Ar_F1 metric.

(a) Comparison with global CSL algorithms: From Figs. 7 and 8, we can observe that no matter which metric is used, our method significantly outperforms the continuous-optimization-based global CSL methods (i.e., NOTEARS, DAG-GNN and DAG-NoCurl) on all datasets. The possible reasons for the poor performance of the continuous-optimization-based global CSL methods on the benchmark datasets are as follows. NOTEARS is specially designed for linear cases instead of non-linear cases. In addition, the convex optimization approach adapted by NOTEARS may fall into local optimal solution, leading to a higher value of SHD compared to PCSL. Although DAG-GNN and DAG-NoCurl can be applied to non-linear cases by adopting different types of neural network models, loss functions and representations of adjacency matrix, their performance is still poor due to the strong theoretical assumptions [25].

PC-stable, as a combinatorial-optimization-based global CSL method, demonstrates competitiveness with our method on datasets with larger sample sizes (e.g., 5,000 samples), particularly evident in the Insurance5 and Insurance10 BNs. However, the performance gap between PC-stable and our method widens significantly when dealing with small sample datasets (e.g., 500 samples). This amplification is attributed to the heightened susceptibility of global CSL methods to cascading errors in such scenarios.

(b) Comparison with local-to-global CSL algorithms: Based on the experimental results in Figs. 7 and 8, we make the following observations. 1) On almost all datasets, our method is significantly superior to GSBN regardless of the SHD and Ar_F1 metrics, since compared with the true structure, the size of the local skeletons learned by GSBN is much small, so the causal structures learned by GSBN misses many true edges. 2) On some BNs (e.g., Insurance, Insurance3, Insurance5 and Insurance10), GGSL achieves a comparable performance against our method probably since they all use BDeu as a scoring function to orient the undirected edges. However, on most BNs, our method still outperforms GGSL, especially in terms of the SHD metric. 3) On many datasets, the values of Ar_F1 of F2SL-c and F2SL-s are lower than those of other local-to-global CSL algorithms (i.e., GGSL, BCSL and PCSL), since both F2SL-c and F2SL-s employ a mutual-information-based feature selection method to learn the local skeleton of a target variable, and this mutual-information-based method focuses on discovering the correlation between variables rather than causality, resulting in that the learned local skeletons may lose many true edges. 4) On most datasets, our method achieves better performance than BCSL, although they all utilize the same algorithm to learn the local skeleton of each variable, which indicates that the progressive global skeleton construction strategy and the integrated global skeleton orientation strategy of our method are effective. We also note that as the sample size decreases, the performance gap between our method and BCSL is further widened. This is because the quality of the sampled sub-datasets obtained from the small sample datasets is unstable (or poor), whereas our method can obtain the best batch of the sampled sub-datasets through the progressive strategy, thus avoiding the drawbacks caused by the unstable quality of the sampled datasets.

(c) Running Time: In Section S-4 of the Supplementary Material, we provide the running time for each algorithm on each dataset in the above experiments.



(a) 500 samples

(b) 1000 samples

(c) 5000 samples

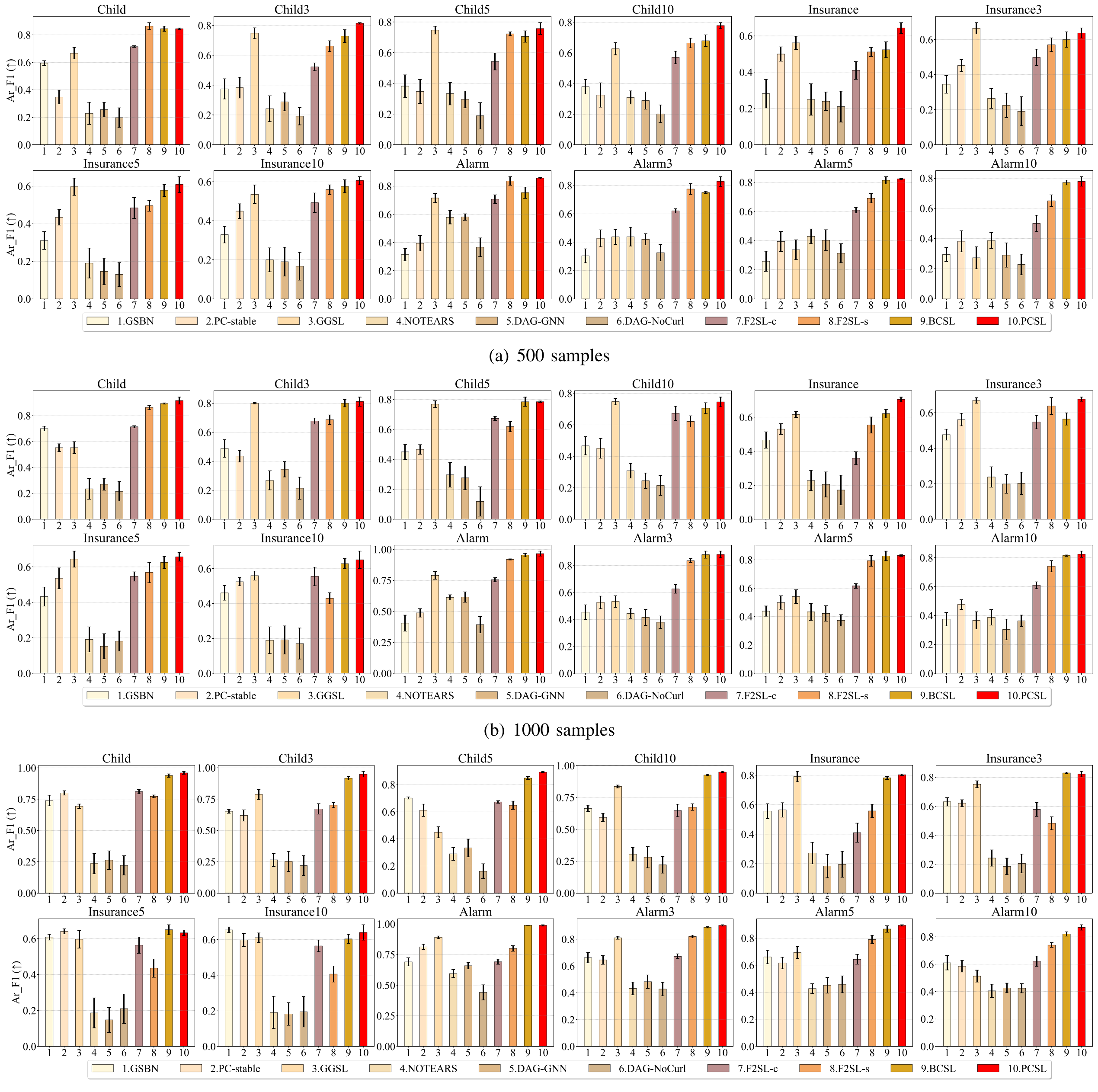
Fig. 7. SHDs of PCSL and its rivals on all benchmark datasets with 500, 1,000 and 5,000 samples.

2) Effectiveness of the Progressive Strategy: As described in Theorem 1, the progressive strategy of PCSL is theoretically effective. In Section S-14 of the Supplementary Material, we use the experimental results on the benchmark datasets to further verify the actual effectiveness of the progressive strategy of PCSL.

3) Quality Assessment of Sampled Sub-Datasets: In our method, it is crucial to generate a batch of high-quality sampled sub-datasets. The constraint of "$DE(\mathcal{D}^{i+1}|S^{i}) > DE(\mathcal{D}^{i}|S^{i-1})$" in Fig. 4 can only theoretically promote PCSL to generate higher quality sub-datasets. In Section S-5 of the Supplementary Material, we visualize the distribution of the sampled sub-datasets for showing whether the quality of the sub-datasets will be improved by the progressive strategy.

C. Applications

1) Estimation of Protein Signaling Network: The interpretability of biology data is of significance. Here we apply PCSL to a bioinformatics dataset, Sachs [3], for the discovery of a protein signaling network based on the expression levels of proteins and phospholipids. Sachs is a widely used dataset for research on graphical models, with experimental annotations accepted by the biological research community.



(a) 500 samples

(b) 1000 samples

(c) 5000 samples

Fig. 8. Ar_F1s of PCSL and its rivals on all benchmark datasets with 500, 1,000 and 5,000 samples.

The ground truth graph contains 11 nodes and 17 edges. In the experiments, we use 853 observed samples for training. Among all methods in the experiments, PCSL achieves the best performance with an SHD of 13. GSBN, PC-stable, F2SL-c and BCSL all have an SHD of 14, GGSL has an SHD of 15, and DAG-NoCurl has an SHD of 17, and the detailed experimental results are summarized in Table II. Besides SHD, we also report the results of Ar_F1, False Discovery Rate (FDR) and True Positive Rate (TPR), and we find that PCSL achieves higher values of Ar_F1 and TPR than its rivals. Although GGSL achieves a low FDR, the causal structure learned by GGSL misses many true edges, leading to a low TPR. In addition, we also observe that the performance of continuous optimization approaches is

TABLE II
RESULTS ON THE PROTEIN SIGNALING NETWORK: COMPARISON OF THE PREDICTED GRAPHS WITH RESPECT TO THE GROUND TRUTH

Method	SHD (↓)	Ar_F1 (↑)	FDR (↓)	TPR (↑)
GSBN	14	0.261	0.500	0.176
PC-stable	14	0.261	0.500	0.176
GGSL	15	0.211	**0.000**	0.118
NOTEARS	15	0.200	0.333	0.118
DAG-GNN	16	0.100	0.667	0.059
DAG-NoCurl	17	0.261	0.500	0.176
F2SL-c	14	0.286	0.250	0.176
F2SL-s	16	0.095	0.750	0.059
BCSL	14	0.261	0.500	0.176
PCSL	**13**	**0.348**	0.333	**0.235**

(↓ means That the lower, the better while ↑ denotes the higher, the Better.)

TABLE III
EXAMPLES OF EXTRACTED EDGES WITH HIGH CONFIDENCE

Domain	Causal relation
Company	organization/has/agent ⟹ worker
	organization/hired/person ⟹ subpart/of
	organization/terminated/person ⟹ organization/has/person
Sports	agent/competes/with/agent ⟹ team/won/trophy
	organization/hired/person ⟹ organization/has/person
	located/at ⟹ location/contains/location
City	location/contains/location ⟹ geopolitical/location/contains/city
	located/at ⟹ location/contains/location
	person/died/in/city ⟹ person/died/in/location
Athlete	team/coach ⟹ organization/has/person
	athlete/beat/athlete ⟹ athlete/wins/award/trophy/tournament
	top/member/of/organization ⟹ person/leads/organization
Person	person/leads/organization ⟹ ceo/of
	has/spouse ⟹ wife/of
	person/born/in/location ⟹ person/graduated/school

(The two sides of ⟹ indicate the cause and effect of a relation, respectively.)

comparable to that of the traditional methods on the real data, but worse than that of traditional methods on the benchmark data.

On the whole, by adopting the progressive global skeleton construction and integrated global skeleton orientation strategies, our proposed method not only achieves a good performance on the benchmark datasets, but also obtains the best results on the real dataset.

2) Extraction of Causality in KB: Recently, Yu et al. proposed a new causal inference task over the relations defined in a knowledge base (KB) schema [22]. The task aims at learning a causal structure, where the nodes are relations and the edges indicate whether one relation suggests another. For example, the relation "person/born/in/location" may imply "person/graduated/school", since people usually choose to attend local schools.

In our experiments, we construct a new dataset from the triples in NELL-One [49], a large-scale operational knowledge system that continuously extracts structured knowledge from web corpora. NELL-One contains approximately 68k entities and 358 relations. In the constructed dataset, each sample corresponds to an entity and each variable corresponds to a relation in this knowledge base. Each sample has on average 2.11 relations (i.e. 2.11 non-zero entries in each row).

Table III gives some causal relationships learned by our method with highest confidence scores. In Table III, we list the causal relations learned from the five domains, and for each effect relation on the right-hand side, we show the highest ranked relations within the same domain.

The causal relationships learned by our method can have a practical value, since NELL-One is a small sample dataset that contains a lot of noise, and the causal structure learned by our method can help filter out the noise in NELL-One, making it suitable for few-shot knowledge graph completion. We plan to conduct a comprehensive study with field experts to systematically evaluate the learned causal relationships.

VI. CONCLUSION

Researchers are increasingly focusing on the local-to-global approach due to its effectiveness and efficiency in the field of causal structure learning. However, existing local-to-global CSL methods encounter many *asymmetric edges* during the global skeleton construction phase, which seriously deteriorates the performance of those methods. To correct those *asymmetric edges*, we propose a novel local-to-global CSL algorithm, called PCSL, with two novel strategies. Specifically, to construct an accurate global skeleton, we design a progressive strategy to iteratively correct *asymmetric edges* and continuously improve the accuracy of the global skeleton. Subsequently, based on the learned accurate global skeleton, we design an integrated global skeleton orientation strategy to obtain an accurate causal structure. Experiments have shown that the proposed PCSL method outperforms nine state-of-the-art CSL algorithms. In addition, the two strategies we proposed can also be integrated into existing local-to-global CSL algorithms. Therefore, in future, we could consider designing PCSL as a unified framework to improve the performance of existing local-to-global CSL algorithms.

REFERENCES

[1] J. Runge et al., "Inferring causation from time series in Earth system sciences," *Nat. Commun.*, vol. 10, no. 1, pp. 1–13, 2019.

[2] J. G. Richens, C. M. Lee, and S. Johri, "Improving the accuracy of medical diagnosis with causal machine learning," *Nat. Commun.*, vol. 11, no. 1, pp. 1–9, 2020.

[3] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan, "Causal protein-signaling networks derived from multiparameter single-cell data," *Sci.*, vol. 308, no. 5721, pp. 523–529, 2005.

[4] Y. He, P. Cui, Z. Shen, R. Xu, F. Liu, and Y. Jiang, "Daring: Differentiable causal discovery with residual independence," in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2021, pp. 596–605.

[5] X. Guo, K. Yu, F. Cao, P. Li, and H. Wang, "Error-aware Markov blanket learning for causal feature selection," *Inf. Sci.*, vol. 589, pp. 849–877, 2022.

[6] I. Ng, Y. Zheng, J. Zhang, and K. Zhang, "Reliable causal discovery with improved exact search and weaker assumptions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 20308–20320.

[7] X. Huang, X. Guo, Y. Li, and K. Yu, "A novel data enhancement approach to DAG learning with small data samples," *Appl. Intell.*, vol. 53, no. 22, pp. 27 589–27 607, 2023.

[8] R. Chen, S. Dash, and T. Gao, "Integer programming for causal structure learning in the presence of latent variables," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 1550–1560.

[9] Z. Fang, Y. Liu, Z. Geng, S. Zhu, and Y. He, "A local method for identifying causal relations under Markov equivalence," *Artif. Intell.*, vol. 305, 2022, Art. no. 103669.

[10] X. Guo, K. Yu, L. Liu, and J. Li, "FedCSL: A scalable and accurate approach to federated causal structure learning," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 12 235–12 243.

[11] K. Yu et al., "Causality-based feature selection: Methods and evaluations," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–36, 2020.

[12] N. K. Kitson, A. C. Constantinou, Z. Guo, Y. Liu, and K. Chobtham, "A survey of Bayesian network structure learning," *Artif. Intell. Rev.*, vol. 56, pp. 8721–8814, 2023.

[13] M. J. Vowels, N. C. Camgoz, and R. Bowden, "D'ya like DAGs? A survey on structure learning and causal discovery," *ACM Comput. Surv.*, vol. 55, pp. 1–36, 2021.

[14] P. Spirtes and C. Glymour, "An algorithm for fast recovery of sparse causal graphs," *Social Sci. Comput. Rev.*, vol. 9, no. 1, pp. 62–72, 1991.

[15] D. Colombo et al., "Order-independent constraint-based causal structure learning," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3741–3782, 2014.

[16] H. Li, V. Cabeli, N. Sella, and H. Isambert, "Constraint-based causal structure learning with consistent separating sets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14257–14266.

[17] A. Sondhi and A. Shojaie, "The reduced PC-algorithm: Improved causal structure learning in large random networks," *J. Mach. Learn. Res.*, vol. 20, no. 164, pp. 1–31, 2019.

[18] D. M. Chickering, "Optimal structure identification with greedy search," *J. Mach. Learn. Res.*, vol. 3, no. Nov, pp. 507–554, 2002.

[19] J. Ramsey, M. Glymour, R. Sanchez-Romero, and C. Glymour, “A million variables and more: The fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images,” *Int. J. Data Sci. Analytics*, vol. 3, no. 2, pp. 121–129, 2017.

[20] M. Chickering, “Statistically efficient greedy equivalence search,” in *Proc. Conf. Uncertainty Artif. Intell.*, PMLR, 2020, pp. 241–249.

[21] X. Zheng, B. Aragam, P. K. Ravikumar, and E. P. Xing, “DAGs with no tears: Continuous optimization for structure learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9492–9503.

[22] Y. Yu, J. Chen, T. Gao, and M. Yu, “DAG-GNN: DAG structure learning with graph neural networks,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 7154–7163.

[23] S. Lachapelle, P. Brouillard, T. Deleu, and S. Lacoste-Julien, “Gradient-based neural DAG learning,” in *Proc. Int. Conf. Learn. Representations*, 2020.

[24] I. Ng, A. Ghassami, and K. Zhang, “On the role of sparsity and DAG constraints for learning linear DAGs,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 17 943–17 954.

[25] Y. Yu, T. Gao, N. Yin, and Q. Ji, “DAGs with no curl: An efficient DAG structure learning approach,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 12 156–12 166.

[26] M. Chickering, D. Heckerman, and C. Meek, “Large-sample learning of Bayesian networks is NP-hard,” *J. Mach. Learn. Res.*, vol. 5, pp. 1287–1330, 2004.

[27] D. Margaritis and S. Thrun, “Bayesian network induction via local neighborhoods,” in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 505–511.

[28] T. Niinimaki and P. Parviainen, “Local structure discovery in Bayesian networks,” in *Proc. Conf. Uncertainty Artif. Intell.*, 2012, pp. 634–643.

[29] T. Gao, K. Fadnis, and M. Campbell, “Local-to-global Bayesian network structure learning,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2017, pp. 1193–1202.

[30] K. Yu, Z. Ling, L. Liu, P. Li, H. Wang, and J. Li, “Feature selection for efficient local-to-global Bayesian network structure learning,” *ACM Trans. Knowl. Discov. Data*, vol. 18, no. 2, pp. 1–27, 2023.

[31] H. Zhang, K. Zhang, S. Zhou, J. Guan, and J. Zhang, “Testing independence between linear combinations for causal discovery,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 6538–6546.

[32] K. Zhang, J. Peters, D. Janzing, and B. Schölkopf, “Kernel-based conditional independence test and application in causal discovery,” in *Proc. Conf. Uncertainty Artif. Intell.*, 2011, pp. 804–813.

[33] P. Spirtes, C. N. Glymour, and R. Scheines, *Causation, Prediction, and Search*. Cambridge, MA, USA: MIT Press, 2000.

[34] B. Huang, K. Zhang, Y. Lin, B. Schölkopf, and C. Glymour, “Generalized score functions for causal discovery,” in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2018, pp. 1551–1560.

[35] T. Silander and P. Myllymaki, “A simple approach for finding the globally optimal Bayesian network structure,” in *Proc. Conf. Uncertainty Artif. Intell.*, 2006, pp. 445–452.

[36] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos, “Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation,” *J. Mach. Learn. Res.*, vol. 11, no. 1, pp. 171–234, 2010.

[37] C. P. de Campos, M. Scanagatta, G. Corani, and M. Zaffalon, “Entropy-based pruning for learning Bayesian networks using BIC,” *Artif. Intell.*, vol. 260, pp. 42–50, 2018.

[38] I. Ng, S. Zhu, Z. Chen, and Z. Fang, “A graph autoencoder approach to causal structure learning,” 2019, *arXiv: 1911.07420*.

[39] X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye, “Statistical ranking and combinatorial Hodge theory,” *Math. Program.*, vol. 127, no. 1, pp. 203–244, 2011.

[40] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, “The max-min hill-climbing Bayesian network structure learning algorithm,” *Mach. Learn.*, vol. 65, no. 1, pp. 31–78, 2006.

[41] I. Tsamardinos, C. F. Aliferis, and A. Statnikov, “Time and sample efficient discovery of Markov blankets and direct causal relations,” in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2003, pp. 673–678.

[42] J. Pearl, *Causality*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[43] T. Hesterberg, “Bootstrap,” *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 3, no. 6, pp. 497–526, 2011.

[44] S. Yaramakala and D. Margaritis, “Speculative Markov blanket discovery for optimal feature selection,” in *Proc. IEEE Int. Conf. Data Mining*, 2005, pp. 4–pp.

[45] X. Guo, K. Yu, L. Liu, F. Cao, and J. Li, “Causal feature selection with dual correction,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 938–951, Jan. 2024.

[46] M. Scutari, “An empirical-Bayes score for discrete Bayesian networks,” in *Proc. Conf. Probabilistic Graphical Models*, PMLR, 2016, pp. 438–448.

[47] M. Luttermann, M. Wienöbst, and M. Liskiewicz, “Practical algorithms for orientations of partially directed graphical models,” in *Proc. Mach. Learn. Res.*, vol. 231, pp. 1–20, 2023.

[48] X. Guo, Y. Wang, X. Huang, S. Yang, and K. Yu, “Bootstrap-based causal structure learning,” in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 656–665.

[49] W. Xiong, M. Yu, S. Chang, X. Guo, and W. Y. Wang, “One-shot relational learning for knowledge graphs,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1980–1990.

Xianjie Guo received the BS degree from Anhui Normal University, Wuhu, China, in 2018 and the MS degree from the Hefei University of Technology, Hefei, China, in 2021. From 2023 to 2024, he was a visiting PhD student with Nanyang Technological University, Singapore, where he focused on research in federated causal discovery. He is currently working toward the PhD degree with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China. His current research interests include causal discovery and federated learning.

Kui Yu (Member, IEEE) received the PhD degree in computer science from the Hefei University of Technology, Hefei, China, in 2013. From 2015 to 2018, he was a research fellow of computer science with the University of South Australia, Adelaide, SA, Australia. From 2013 to 2015, he was a postdoctoral fellow with the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada. He is currently a professor with the School of Computer Science and Information Engineering, Hefei University of Technology. His main research interests include causal discovery and machine learning.

Lin Liu received the BS and MS degrees in electronic engineering from Xidian University, Xi’an, China, in 1991 and 1994, respectively, and the PhD degree in computer systems engineering from the University of South Australia, Adelaide, SA, Australia. She is currently a professor with the University of South Australia, Adelaide, SA, Australia. Her research interests include data mining, machine learning, causal inference, and bioinformatics.

Jiuyong Li (Member, IEEE) received the PhD degree in computer science from Griffith University, Brisbane, QLD, Australia, in 2002. He is currently a professor with the University of South Australia, Adelaide, Australia. His main research interests include data mining, causal discovery and inference, and bioinformatics. His research work has been supported by eight Australian Research Council Discovery projects and many industry and government projects.

Jiye Liang (Senior Member, IEEE) received the MS and PhD degrees from Xi'an Jiaotong University, Xi'an, China, in 1990 and 2001, respectively. He is currently a professor with the School of Computer and Information Technology, Shanxi University, Taiyuan, China, where he is also the director of the Key Laboratory of Computational Intelligence and Chinese Information Processing of the Ministry of Education. He has authored more than 200 journal papers in his research fields. His current research interests include computational intelligence, granular computing, data mining, and knowledge discovery.

Fuyuan Cao received the MS and PhD degrees in computer science from Shanxi University, Taiyuan, China, in 2004 and 2010, respectively. He is currently a professor with the school of computer and information technology, Shanxi University, China. His current research interests include machine learning and clustering analysis.

Xindong Wu (Fellow, IEEE) received the PhD degree in artificial intelligence from the University of Edinburgh, Edinburgh, Scotland, U.K., in 1993. He is currently a Chang Jiang scholar with the School of Computer Science and Information Engineering, Hefei University of Technology, China. His research interests include data mining, Big Data analytics, and knowledge-based systems. He is also a steering committee chair of the IEEE International Conference on Data Mining, an editor-in-chief of the *Knowledge and Information Systems* (Springer), and the series editor-in-chief of the Springer Book Series on *Advanced Information and Knowledge Processing*. From 2005 to 2008, he was an editor-in-chief of *IEEE Transactions on Knowledge and Data Engineeing*. He was also a program committee chair/co-chair of the 2003 IEEE International Conference on Data Mining (ICDM'03), 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-07), and 19th ACM Conference on Information and Knowledge Management (CIKM2010). He is also a fellow AAAS (American Association for the Advancement of Science).