Supplementary Material for "Progressive Skeleton Learning for Effective Local-to-Global Causal Structure Learning"

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

CONTENTS

S-1: PROOF FOR THEOREM 1

Theorem 1. *Let L denote the number of iterations in Phase 2. In each iteration $i \in 1, 2, ..., L$, S^i denotes the constructed global skeleton, and $\mathcal{D}^i$ represents the set of sampled sub-datasets used. Assuming that Eq. (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 $i \in 1, 2, ..., L-1$.*
2) *The quality of the sub-datasets $\mathcal{D}^{i+1}$ is better than that of $\mathcal{D}^i$ for all $i \in 1, 2, ..., L-1$.*

Proof. Let $Acc(S^i)$ denote the accuracy of S^i, and $Qua(\mathcal{D}^i)$ denote the quality of $\mathcal{D}^i$. When $i = 1$, PCSL generates the 1-st batch of sub-datasets $\mathcal{D}^1$ by Bootstrap sampling. Based on the PC learning results obtained on $\mathcal{D}^1$, PCSL applies Criterion 1 to construct a global skeleton S^1 that corrects all *asymmetric edges* in S^0 for the first time, although some *asymmetric edges* may be incorrectly corrected. Subsequently, based on Bootstrap sampling and Eq. (10), PCSL uses the latest global skeleton S^1 (instead of S^0) as a reference to generate a batch of higher quality sub-datasets $\mathcal{D}^2$. Here, since the new batch of generated sub-datasets will be retained only if the condition "$DE(\mathcal{D}^{i+1}|S^i) > DE(\mathcal{D}^i|S^{i-1})$" in Fig. 4 holds, $Qua(\mathcal{D}^2) > Qua(\mathcal{D}^1)$. Similarly, based on the PC learning results obtained on $\mathcal{D}^2$ and Criterion 1, PCSL corrects each *asymmetric edge* in S^0 again for further improving the accuracy of the global skeleton and obtaining S^2. Since the core of calculating $DE(\mathcal{D}^i|S^{i-1})$ (Eq. (10)) is to calculate $Q(D_j^i, k, X_d)$ (Eq. (3)) with implicit structural accuracy information, the higher the quality of the sampled sub-datasets, the higher the correction accuracy of *asymmetric edges*. Here, since $DE(\mathcal{D}^2|S^1) > DE(\mathcal{D}^1|S^0)$ holds (i.e., $Qua(\mathcal{D}^2) > Qua(\mathcal{D}^1)$), $Acc(S^2) > Acc(S^1)$. Thus, when $i = 1$, Theorem 1 holds.

Assume that when $i = c$ ($c \in \{1, 2, ..., (L-2)\}$), Theorem 1 holds. Then, we can obtain $Acc(S^{c+1}) > Acc(S^c)$ and $Qua(\mathcal{D}^{c+1}) > Qua(\mathcal{D}^c)$. In the $(c+1)$-th iteration, based on Bootstrap sampling and Eq. (10), PCSL uses the latest global skeleton S^{c+1} as a reference to generate a new batch of sub-datasets $\mathcal{D}^{c+2}$. Here, since $c+1 \leq L-1 < L$ (i.e., $c+1 < L$), PCSL will execute the next iteration. In other words, $DE(\mathcal{D}^{c+2}|S^{c+1}) > DE(\mathcal{D}^{c+1}|S^c)$ holds, and $Qua(\mathcal{D}^{c+2}) > Qua(\mathcal{D}^{c+1})$. Subsequently, based on the PC learning results obtained on $\mathcal{D}^{c+2}$ and Criterion 1, PCSL corrects each *asymmetric edge* in S^0 again, and then constructs a new global skeleton S^{c+2}. Here, since $Qua(\mathcal{D}^{c+2}) > Qua(\mathcal{D}^{c+1})$, the correction accuracy of *asymmetric edges* in S^{c+2} is higher than that in S^{c+1}, and $Acc(S^{c+2}) > Acc(S^{c+1})$. Thus, when $i = c+1$, Theorem 1 also holds.

Based on mathematical induction, for $\forall\ i \in \{1, 2, ..., (L-$

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,xwu}@hfut.edu.cn).

Lin Liu and Jiuyong Li are with the UniSA STEM, University of South Australia, Adelaide 5095, Australia (e-mail: {Lin.Liu,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,cfy}@sxu.edu.cn).

* Corresponding author.

1)}, $Acc(S^{i+1}) > Acc(S^i)$ (*progressive skeleton learning*) and $Qua(\mathcal{D}^{i+1}) > Qua(\mathcal{D}^i)$ (*progressive data sampling*). *Summarizing:* Theorem 1 is true. (Q.E.D) □

S-2: Proof for Theorem 2

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 $\mathcal{A}^*$.*

Proof. We prove the theorem by contradiction, and we assume that there is a bidirectional edge $X_a \leftrightarrow X_b$ (i.e., both $X_a \rightarrow X_b$ and $X_a \leftarrow X_b$ exist) in $\mathcal{A}^*$, then $\mathcal{A}^*$ needs to satisfy:

$$\mathcal{A}^*(a, b) \geq 0.5 \; and \; \mathcal{A}^*(b, a) \geq 0.5. \tag{1}$$

Since $\mathcal{A}^* = (\sum_{j=1}^{N} \mathcal{A}_j)/N$, we have:

$$N * \mathcal{A}^*(a, b) = \sum_{j=1}^{N} \mathcal{A}_j(a, b), \; and \tag{2}$$

$$N * \mathcal{A}^*(b, a) = \sum_{j=1}^{N} \mathcal{A}_j(b, a). \tag{3}$$

Combining Eq. (2) and Eq. (3), we can obtain:

$$N * (\mathcal{A}^*(a, b) + \mathcal{A}^*(b, a)) = \sum_{j=1}^{N} (\mathcal{A}_j(a, b) + \mathcal{A}_j(b, a)). \tag{4}$$

Since $\mathcal{A}_j$ ($j \in [1, N]$) is a DAG, we have:

$$\mathcal{A}_j(a, b) + \mathcal{A}_j(b, a) = 0 \; or \; 1, \; and \tag{5}$$

$$\sum_{j=1}^{N} (\mathcal{A}_j(a, b) + \mathcal{A}_j(b, a)) \leq N. \tag{6}$$

Substitute Eq. (4) into the left side of Formula (6), thus,

$$N * (\mathcal{A}^*(a, b) + \mathcal{A}^*(b, a)) \leq N. \tag{7}$$

Simplify Formula (7), then:

$$\mathcal{A}^*(a, b) + \mathcal{A}^*(b, a) \leq 1. \tag{8}$$

Thus, the conditions of Formula (1) can be satisfied if and only if $\mathcal{A}^*(a, b) = 0.5$ and $\mathcal{A}^*(b, a) = 0.5$, i.e.,

$$\frac{1}{N} * \sum_{j=1}^{N} \mathcal{A}_j(a, b) = \frac{1}{2} \; and \; \frac{1}{N} * \sum_{j=1}^{N} \mathcal{A}_j(a, b) = \frac{1}{2}. \tag{9}$$

Or equivalently,

$$\sum_{j=1}^{N} \mathcal{A}_j(a, b) = \frac{N}{2} \; and \; \sum_{j=1}^{N} \mathcal{A}_j(a, b) = \frac{N}{2}. \tag{10}$$

Since N is an odd number, $N/2$ cannot be an integer. But $\mathcal{A}_j(a, b)$ ($j \in [1, N]; a, b \in [1, m]$) is an integer (0 or 1), thus two equations in (10) must not hold, i.e., the *assumption* that there is a bidirectional edge $X_a \leftrightarrow X_b$ in $\mathcal{A}^*$ does not hold and the theorem is proved. □

S-3: Implementation Details

All experiments were conducted on a computer with Intel Core i9-10900 3.70-GHz CPU, NVIDIA GeForce RTX 3060 GPU and 64-GB memory. The significance level for conditional independence tests is set to 0.01. For continuous-optimization-based DAG learning methods (i.e., NOTEARS, DAG-GNN and DAG-NoCurl), we adopt 0.3 as the threshold to prune the obtained DAGs [1]. Based on the sensitivity analysis of parameter N in [2], in our experiments, the parameter N for both BCSL and our method is set to 15. GSBN, PC-stable, F2SL-c/s[1], BCSL and our algorithm are implemented in MATLAB, GGSL is implemented in C/C++, and NOTEARS[2], DAG-GNN[3] and DAG-NoCurl[4] are implemented in PYTHON.

S-4: Experimental Results on More Metrics

Let TP be the number of true positives (edges in both the true structure and learned structure); FP the number of false positives (edges in the learned structure but not in the true causal structure; TN the number of true negatives (edges not in either the true or learned structure); and FN the number of false negatives (edges in the true structure but missing from the learned structure). We evaluate the performance of `PCSL` and its rivals using the following metrics.

- *False Discovery Rate (FDR)*. FDR is the ratio of false edges in the learned causal structure to the edges in the learned causal structure. That is, $FDR = \frac{FP}{TP+FP}$.
- *True Positive Rate (TPR)*. TPR is the ratio of correct edges in the learned causal structure to total edges in the true causal structure. That is, $TPR = \frac{FP}{TP+FN}$.
- *Time*. We report running time (in seconds) as the efficiency measure of different algorithms.

In all figures and tables, (↑) means the higher the better, (↓) means the lower the better, and the best results are highlighted in bold face.

Figures 1-2 report the quality of the causal structures learned by `PCSL` and its rivals in terms of FDR and TPR metrics. From the experimental results, we can see that `PCSL` always maintains a low FDR (as shown in Figure 1) and a high TPR (as shown in Figure 2) on almost all datasets.

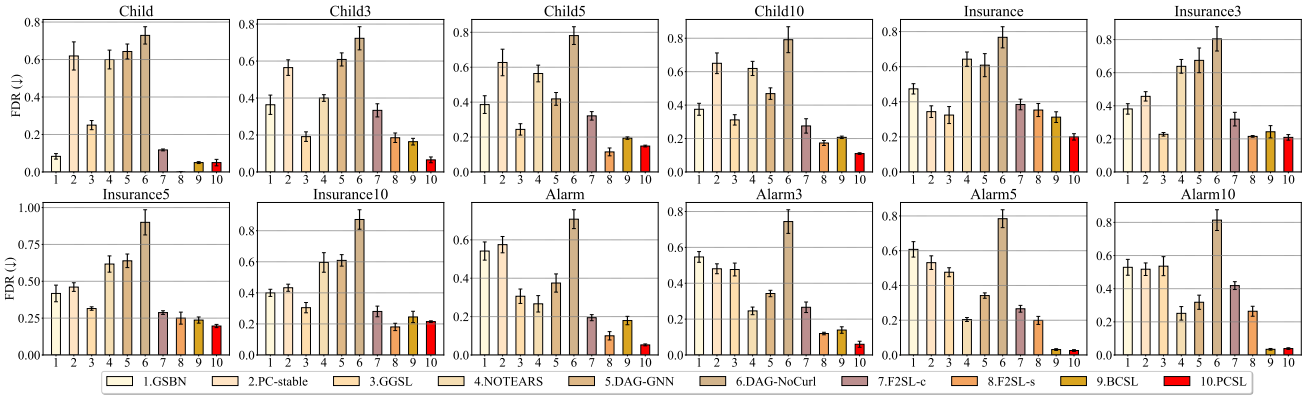
As a combinatorial-optimization-based global CSL method, PC-stable is competitive with our method in TPR, but significantly inferior to our method in FDR, especially on the Child, Child3, Child5, Child10, Alarm, Alarm3, Alarm5 and Alarm10 BNs. This indicates that PC-stable learns many extra edges. On almost all datasets, our method is significantly superior to GSBN on FDR and TPR 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. On some BNs (e.g., Insurance, Insurance3, Insurance5 and Insurance10), GGSL achieves a comparable performance against our method probably since they all use

[1]The source codes of GSBN, PC-stable, F2SL-c and F2SL-s are available at https://github.com/kuiy/CausalLearner.

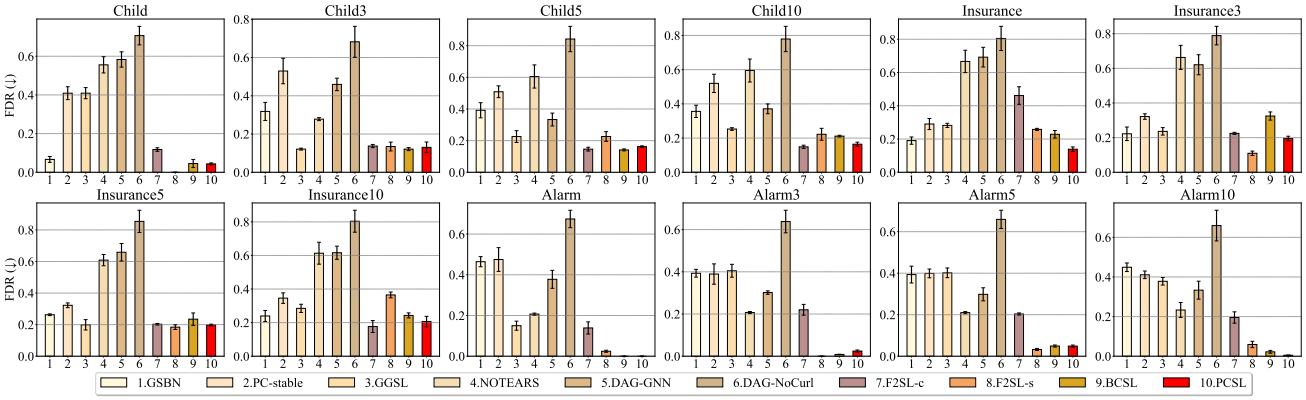
[2]The implementation is publicly available at https://github.com/xunzheng/notears.

[3]The code is available at https://github.com/fishmoon1234/DAG-GNN.

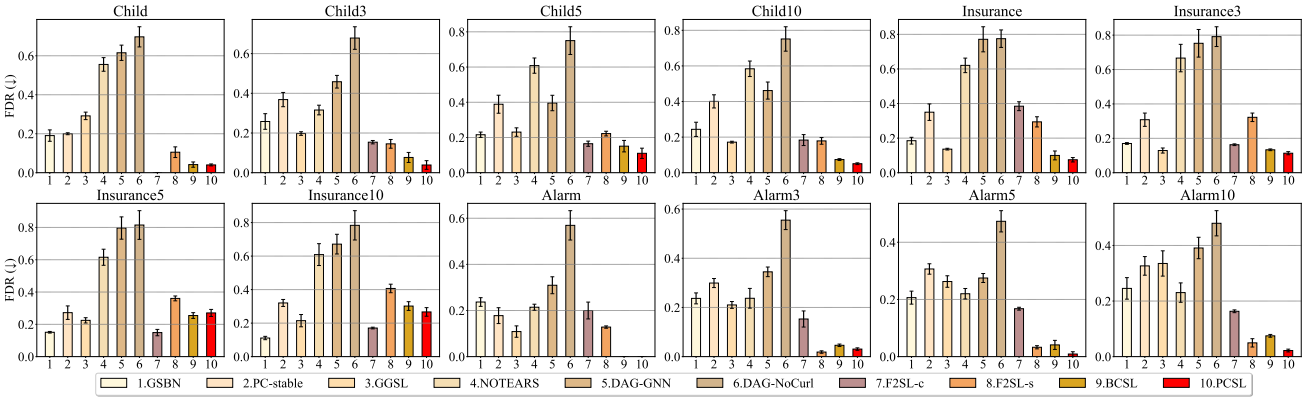
[4]The code is available at https://github.com/fishmoon1234/DAG-NoCurl.



(a) 500 samples



(b) 1000 samples



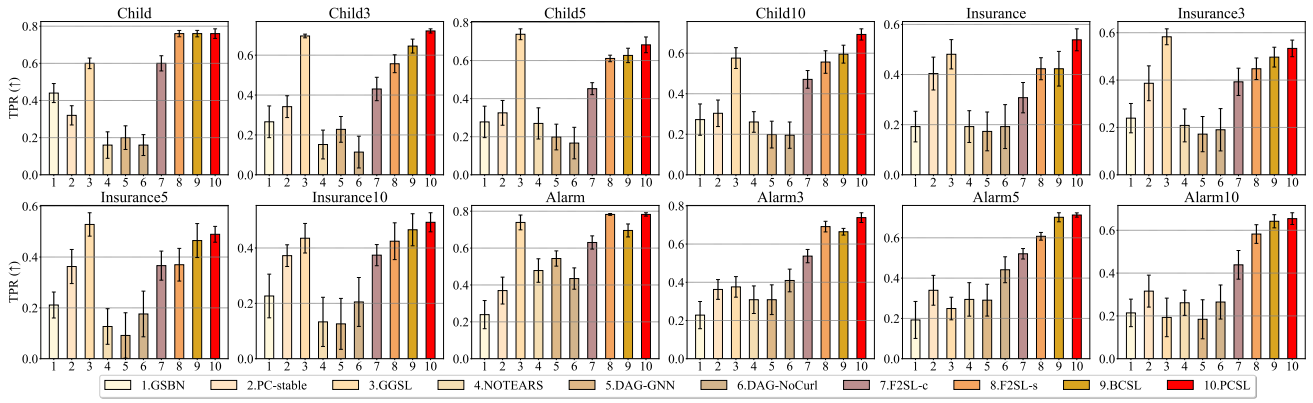
(c) 5000 samples

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

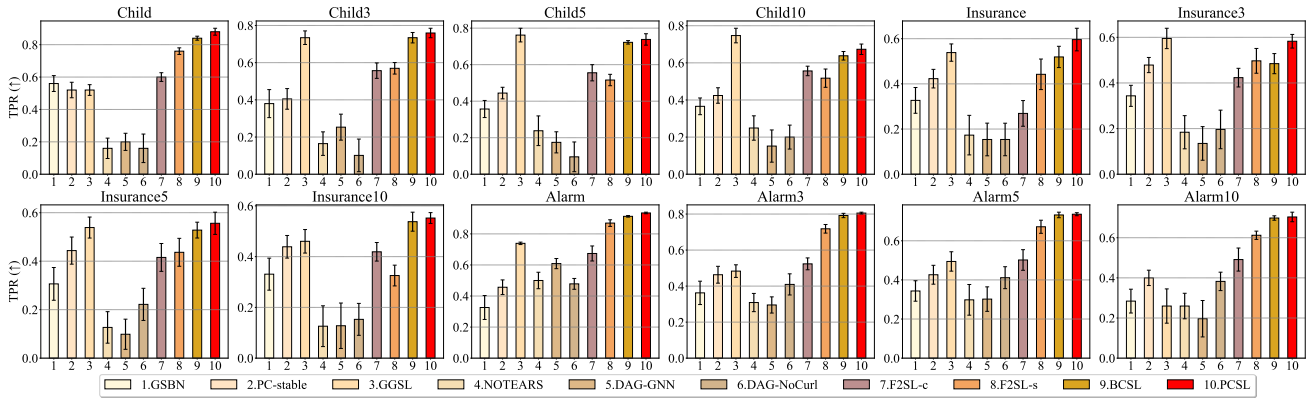
BDeu as a scoring function to orient the undirected edges. However, on most BNs, our method still outperforms GGSL, especially in terms of the FDR metric. On most datasets, the values of TPR of F2SL-c and F2SL-s are lower than those of other local-to-global CSL algorithms (i.e., GGSL, BCSL and PCSL), 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.

Table I provides the running time (CPU or GPU) for each algorithm on each dataset in the above experiments. We conclude from Table I that, PCSL is slower than most local-to-global CSL algorithms but faster than most global CSL algorithms. In practice, to achieve scalability in high-dimensional BNs (e.g., Child10, Insurance10 and Alarm10), continuous-optimization-based CSL methods (i.e., NOTEARS, DAG-GNN and DAG-NoCurl) must be accelerated using GPU.

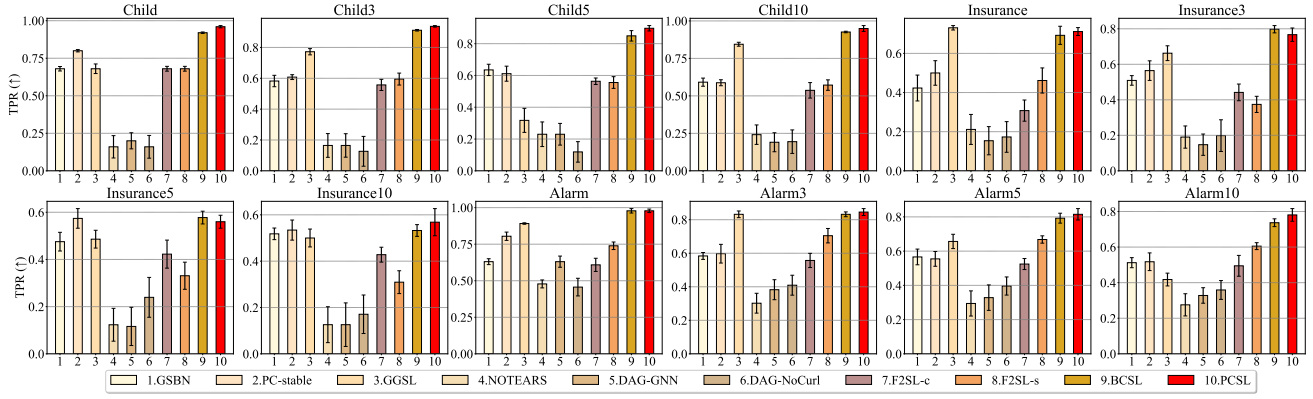
Note that, both PCSL and BCSL need to learn the local causal skeleton on multiple sampled sub-datasets for achieving higher accuracy of causal structure learning. Hence, the loss of time efficiency is unsurprising. Compared with BCSL, PCSL needs to iteratively correct *asymmetric edges* on multiple batches of sub-datasets, but the running time of PCSL is not much slower than that of BCSL since PCSL usually has



(a) 500 samples



(b) 1000 samples



(c) 5000 samples

Fig. 2. TPRs of PCSL and its rivals on all benchmark datasets with 500, 1,000 and 5,000 samples.

less than or equal to 7 iterations during global skeleton construction, see Section V-B2 for details. Although PCSL needs to repeatedly learn the local causal skeleton of variables on *asymmetric edges* on multiple batches of sub-datasets during the global skeleton construction, it is still faster than, the local-to-global CSL method, GGSL. Since DAG-NoCurl is designed to solve the resultant unconstrained optimization problem, it is more efficient than other continuous-optimization-based CSL methods (i.e., NOTEARS and DAG-GNN).

From Table I, we also find that the running time of an algorithm is not always positively correlated with the sample size of a dataset. For example, 1) the running time of NOTEARS and DAG-NoCurl on Alarm10 with 1,000 samples is less than that on Alarm10 with 500 samples since the running time of these two algorithms depends only on the number of iterations; 2) the running time of GGSL on Insurance10 with 1,000 samples is less than that on Insurance10 with 500 samples since GGSL randomly selects a variable as the initial variable each time, and its running time is very unstable.

Overall, since PCSL only needs to repeatedly learn the local skeleton of the variables on the *asymmetric edges* rather than the local skeleton of all variables, the time cost of PCSL is reasonable. Moreover, according to the analysis of the time complexity of PCSL in Section IV-D of the main text, if the sample size of a dataset is large, the number of *asymmetric edges* learned on this dataset is small, i.e., the

TABLE I
EXPERIMENT TIME ($lg(Time)$) OF EACH ALGORITHM.

#Sample	Network	GSBN	PC-stable	GGSL	NOTEARS	DAG-GNN	DAG-NoCurl	F2SL-c	F2SL-s	BCSL	PCSL
500	Child	0.018	0.024	0.316	0.928	1.982	0.212	**0.012**	0.021	0.085	0.313
	Child3	0.082	0.119	1.298	1.334	2.450	0.571	**0.066**	0.078	0.519	1.017
	Child5	0.174	0.299	1.722	2.052	2.025	0.869	**0.135**	0.182	0.755	1.197
	Child10	0.460	0.626	2.375	2.962	2.276	1.636	**0.370**	0.423	1.361	2.007
	Insurance	0.027	0.044	0.470	0.984	1.842	0.346	**0.018**	0.040	0.295	0.621
	Insurance3	**0.113**	0.216	2.048	1.852	1.897	0.936	0.118	0.177	0.693	1.402
	Insurance5	0.242	0.421	2.516	2.608	2.812	1.686	**0.223**	0.257	1.055	1.708
	Insurance10	0.579	0.831	3.469	3.363	3.143	2.395	**0.576**	0.610	1.592	2.271
	Alarm	**0.039**	0.114	0.869	1.303	2.288	0.633	0.171	0.185	0.286	0.424
	Alarm3	0.200	0.317	2.087	1.917	2.760	1.328	**0.162**	0.231	0.864	1.318
	Alarm5	0.404	0.544	3.000	2.608	2.929	1.998	**0.340**	0.366	1.217	1.702
	Alarm10	0.985	1.136	3.858	3.489	3.279	3.036	**0.742**	0.799	1.802	2.305
1000	Child	0.022	0.039	0.491	0.899	2.324	0.212	**0.014**	0.026	0.098	0.310
	Child3	0.104	0.211	1.294	1.427	2.756	0.633	**0.084**	0.123	0.454	0.838
	Child5	0.216	0.315	1.813	1.934	2.170	0.859	**0.214**	0.227	0.787	1.236
	Child10	0.546	0.662	2.438	2.749	2.423	1.559	**0.536**	0.577	1.320	1.805
	Insurance	0.030	0.067	0.664	0.939	2.015	0.318	**0.028**	0.045	0.359	0.724
	Insurance3	**0.144**	0.307	2.127	1.735	2.183	0.975	0.163	0.226	0.964	1.585
	Insurance5	**0.295**	0.520	2.648	2.332	3.110	1.617	0.394	0.411	1.276	1.993
	Insurance10	**0.665**	0.900	3.240	3.087	3.383	2.303	0.739	0.727	1.888	2.484
	Alarm	0.051	0.119	0.970	1.156	2.559	0.818	**0.050**	0.073	0.226	0.482
	Alarm3	**0.256**	0.372	2.067	1.951	3.015	1.324	0.264	0.270	0.798	1.250
	Alarm5	0.485	0.583	2.782	2.523	3.235	1.796	**0.477**	0.479	1.194	1.674
	Alarm10	1.098	1.136	3.821	3.255	3.548	2.729	1.062	**0.923**	1.839	2.343
5000	Child	**0.058**	0.153	0.962	1.127	2.969	0.231	0.074	0.068	1.156	1.659
	Child3	**0.264**	0.422	2.074	1.587	3.501	0.809	0.292	0.391	1.528	2.034
	Child5	**0.469**	0.568	2.176	1.936	2.822	1.048	0.586	0.623	1.257	1.760
	Child10	**0.854**	0.926	3.168	2.581	3.162	1.617	1.020	1.085	2.620	3.247
	Insurance	**0.083**	0.322	1.268	1.277	2.707	0.613	0.096	0.126	1.019	1.519
	Insurance3	**0.366**	0.800	2.661	1.684	2.787	1.150	0.476	0.495	1.980	2.714
	Insurance5	**0.591**	0.996	3.211	2.164	3.774	1.627	0.764	0.798	2.752	3.362
	Insurance10	**1.027**	1.317	3.803	2.634	4.091	2.197	1.291	1.337	3.127	3.838
	Alarm	**0.140**	0.315	1.472	1.489	3.256	0.758	0.161	0.193	0.484	0.966
	Alarm3	**0.511**	0.582	2.558	2.337	3.657	1.502	0.595	0.716	1.361	1.862
	Alarm5	**0.829**	0.842	3.116	2.604	3.900	1.980	0.950	1.024	1.995	2.475
	Alarm10	**1.357**	1.374	3.860	3.435	4.159	2.512	1.564	1.584	2.780	3.240

time complexity of Phase 2 of PCSL is low. Thus, when the sample size increases, the efficiency gap between PCSL and other algorithms decreases.

S-5: 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 this section, 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.

Specifically, we first set the number of datasets in each batch of sub-datasets to 3, i.e., $N = 3$. Then, we run PCSL on the Child with 500 samples, and record the sub-datasets generated in the first batch and the final batch. Finally, we visualize the distribution of these sub-datasets and the original dataset in Fig. 3, where (a) shows the distribution of the original dataset, (b)-(d) and (f)-(h) show the distributions of the sub-datasets generated in the first batch and the final batch, respectively. Further, to quantify the difference in distributions between datasets, we use the maximum mean discrepancy (MMD) [3] to measure the distribution divergence, and the measurement results are shown in Fig. 3(e), where the smaller the value of MMD(·,·), the smaller the distribution difference between two datasets.

From Fig. 3, we can see that the distribution of the first batch of generated sub-datasets differs greatly from that of the original dataset. In contrast, the distribution difference between the final batch of generated sub-datasets and the original dataset is small. As described in Eq. (3), PCSL aims to generate the sub-datasets that deviate from the original data distribution for alleviating the quality issue of the original dataset, but avoids generating the sub-datasets that deviate too much from the original data distribution. Thus, based on the results in Fig. 3, PCSL can improve the quality of the sub-datasets by the progressive strategy.

S-6: STATISTICAL TESTS FOR EXPERIMENTAL RESULTS

In this section, we employ the Friedman test [4] and Nemenyi test [4] to evaluate whether PCSL demonstrates statistically significant superiority over other methods across 12 benchmark datasets.

We begin with the Friedman test [4], conducted at a 0.05 significance level. The null hypothesis posits that all algorithms perform equivalently across all datasets, implying equal average rankings. Table II summarizes the average rankings of PCSL and the baseline algorithms for various metrics, based on experimental results from all benchmark datasets.

TABLE II
THE AVERAGE RANKINGS OF PCSL AND THE BASELINES ON THE BENCHMARK DATASETS USING SHD, AR_F1, FDR AND TPR METRICS.

Algorithm		GSBN	PC-stable	GGSL	NOTEARS	DAG-GNN	DAG-NoCurl	F2SL-c	F2SL-s	BCSL	PCSL
Avg rank	SHD	6.07	5.11	4.64	7.94	8.46	10	5.21	4.17	2.28	1.13
	Ar_F1	6.33	5.96	4.03	8.06	8.61	9.58	4.92	3.92	2.38	1.22
	FDR	5.81	7.24	5.11	7.06	7.94	10	4.14	3.19	2.78	1.74
	TPR	6.67	5.29	3.72	8.67	8.97	8.39	5.36	4.18	2.42	1.33

Notably, the null hypothesis is rejected for all metrics, indicating significant performance differences among the algorithms. Furthermore, PCSL consistently outperforms the baselines across all metrics. (Note: In Table II, lower ranking values indicate superior performance.)

To further elucidate the significant differences between PCSL and the baselines, we conduct the Nemenyi test [4]. This test stipulates that the performance of two algorithms is significantly different if their corresponding average rankings differ by at least one critical difference (CD). The CD for the Nemenyi test is calculated as follows (i.e., Eq. (11)).

$$\mathrm{CD} = q_{\alpha,\theta}\sqrt{\frac{\theta(\theta+1)}{6\eta}}, \tag{11}$$

where α is the significance level, θ is the number of comparison algorithms, and η denotes the number of datasets. In our experiments, $\theta = 10$, $q_{\alpha=0.05,\theta=10} = 3.164$ at significance level $\alpha = 0.05$. For the benchmark datasets, $\eta = 12 * 3 = 36$ (twelve benchmark datasets with three types of sample sizes), and thus CD ≈ 2.26.

Figures 4(a)-4(d) present the CD diagrams for four different metrics. In each diagram, the average ranking of each algorithm is plotted along the axis, with lower rankings positioned to the right. The results reveal:

- For the SHD, Ar_F1, and TPR metrics: PCSL achieves comparable performance to BCSL and significantly outperforms other baselines.
- For the FDR metric: PCSL significantly outperforms F2SL-c, GGSL, GSBN, NOTEARS, PC-stable, DAG-GNN, and DAG-NoCurl, while achieving comparable performance to the remaining baselines.

Notably, PCSL consistently achieves the lowest ranking value across all metrics, underscoring its robust performance.

S-7: SENSITIVITY ANALYSIS OF PARAMETER r

To evaluate the robustness of our PCSL algorithm with respect to its parameter r, we conducted a comprehensive sensitivity analysis across three benchmark BN datasets: Child, Insurance, and Alarm. The experimental procedure was as follows:

1. Data Generation: For each of the three datasets, we generated multiple batches of data, each containing 5,000 samples. This approach ensures a diverse and representative set of instances for our analysis.

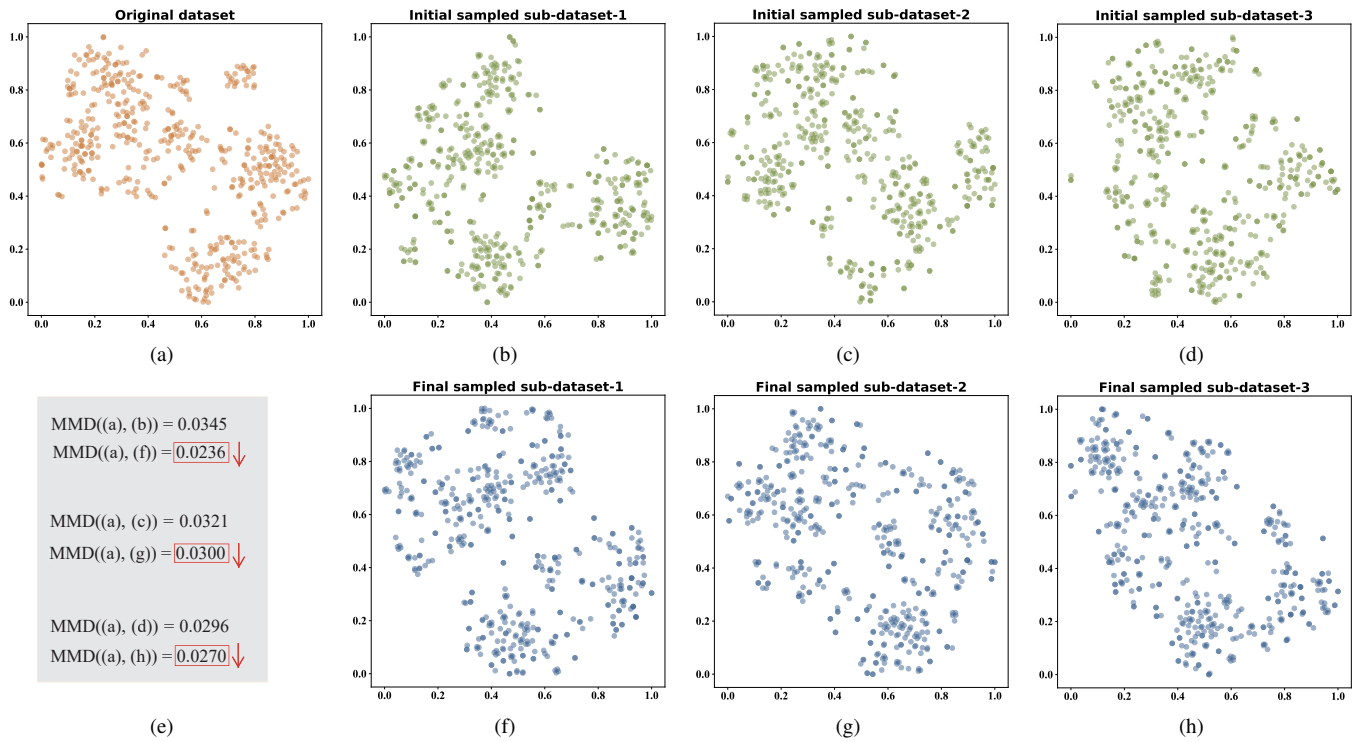
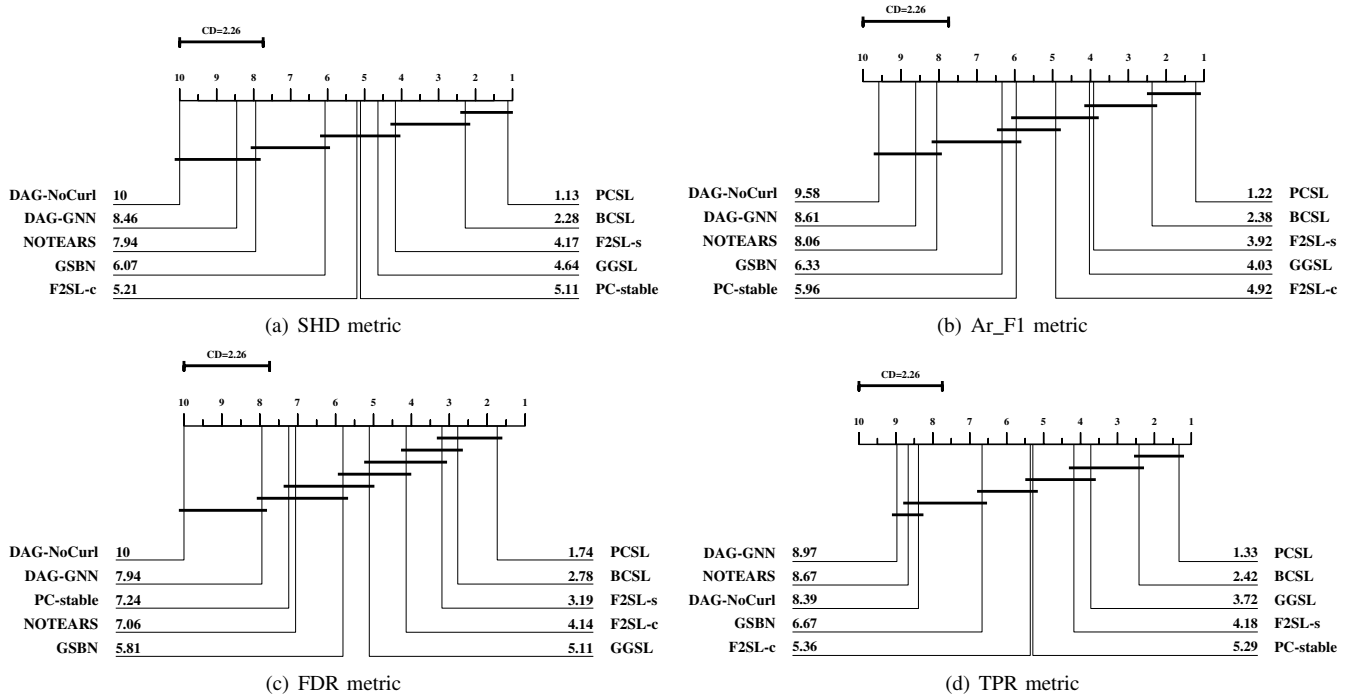


Fig. 3. Visualization of the distribution of the original dataset and the sampled sub-datasets.



(a) SHD metric (b) Ar_F1 metric

(c) FDR metric (d) TPR metric

Fig. 4. Crucial difference diagrams from the Nemenyi test for all algorithms across 12 benchmark datasets.

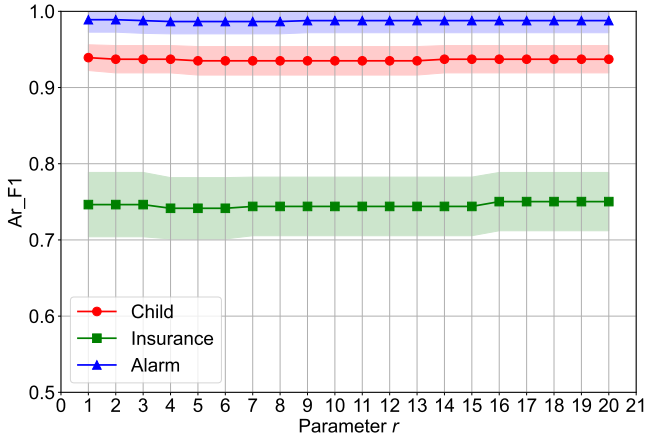


Fig. 5. Sensitivity analysis for parameter r of PCSL.

2. Algorithm Execution: We ran our PCSL algorithm on each dataset, systematically varying the parameter r from 1 to 20. This range was chosen to cover a wide spectrum of possible values for r.

3. Performance Metric: For each value of r, we calculated the Ar_F1 metric, which serves as our primary metric for assessing the algorithm's performance.

4. Visualization: The results of this analysis are presented in Figure 5, which illustrates the relationship between the parameter r and the Ar_F1 metric for each dataset.

As evident from Figure 5, a striking observation emerges: the performance of our PCSL algorithm demonstrates remarkable stability across all three datasets, regardless of the value chosen for parameter r. This consistency is manifested by the nearly horizontal lines in the graph, indicating minimal fluctuation in the Ar_F1 metric as r varies.

This finding has several important implications:

- The PCSL algorithm exhibits strong robustness to changes in the parameter r. This characteristic is highly desirable, as it suggests that the algorithm's performance is not overly sensitive to precise parameter tuning.
- The consistent performance across different datasets (i.e., Child, Insurance, and Alarm) indicates that the algorithm's stability is not dataset-specific, but rather a general property of the method.
- The low sensitivity to r simplifies the application of the PCSL algorithm in practice. Users can choose from a wide range of r values without significantly impacting the algorithm's effectiveness.

In conclusion, this sensitivity analysis provides strong evidence for the robustness and reliability of our PCSL algorithm. The minimal impact of parameter r on performance across diverse datasets underscores the algorithm's potential for broad applicability in various domains without the need for fine-tuned parameter adjustments.

S-8: Detailed Pseudo-code for PCSL

The pseudo-code of the PCSL algorithm is detailed in Algorithm 1, and PCSL comprises the following three phases:

- Phase 1: Initial local skeleton learning (**Lines 1-9**)
- Phase 2: Progressive global skeleton construction (**Lines 10-32**)
  - Step 1: Relearn the local skeleton of variables on each *asymmetric edge* (**Lines 14-18**).
  - Step 2: Correct each *asymmetric edge* (**Lines 19-23**).
  - Step 3: Generate higher quality sub-datasets guided by the newly learned skeleton (**Lines 25-31**).

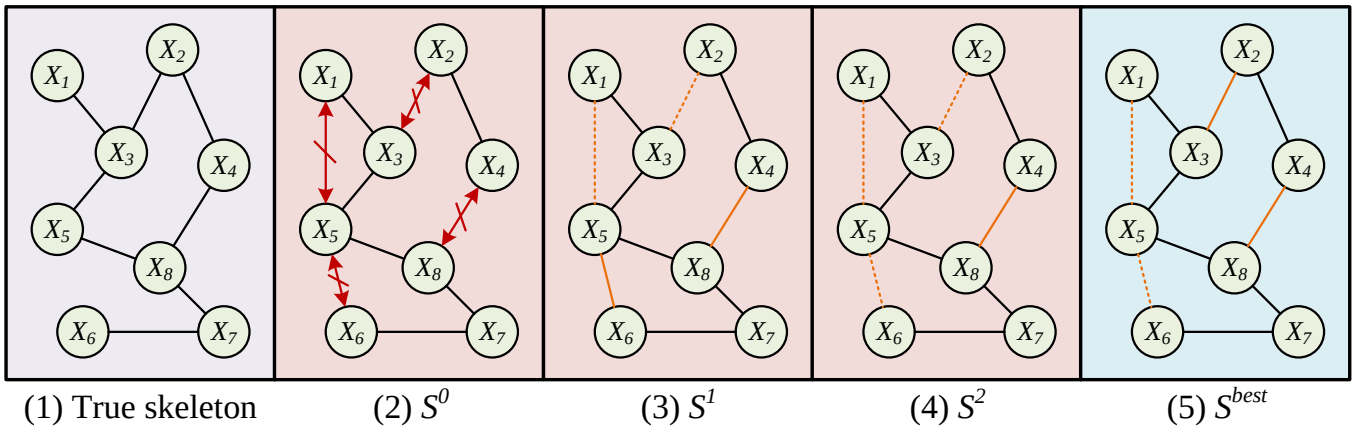


Fig. 6. An example of tracing the progressive learning strategy of PCSL. Sub-figure (1) shows the true global causal skeleton and Sub-figures (2)-(5) show the evolution of the global skeleton during the iterative process.

- Phase 3: Integrated global skeleton orientation (**Lines 33-49**)
  - Step 1: Orient the global skeleton independently on each sub-dataset (**Lines 34-36**).
  - Step 2: Merge all adjacency matrices to form the final causal structure (**Lines 37-48**).

S-9: Tracing the Progressive Strategy of PCSL

In this section, we illustrate the progressive learning strategy of the PCSL algorithm using a small network example, as shown in Figure 6. Figure 6(1) displays the true global causal skeleton, while Figures 6(2)-(5) demonstrate the evolution of the skeleton during PCSL's progressive optimization process.

Initially, the HITON-PC algorithm is applied to the original dataset D_{orig} to learn the local causal skeleton for each variable, resulting in the global skeleton S^0.

However, according to Definition 2, we can identify four *asymmetric edges* (i.e., "$X_1 \nleftrightarrow X_5$", "$X_5 \nleftrightarrow X_6$", "$X_2 \nleftrightarrow X_3$", and "$X_4 \nleftrightarrow X_8$") in S^0. PCSL then employs the Bootstrap method to resample D_{orig}, generating a batch of new sub-datasets $\mathcal{D}^1$. PCSL re-learns the local causal skeletons for each variable involved in the four *asymmetric edges* on each sub-dataset in $\mathcal{D}^1$. Finally, using Equation (5) and Criterion 1 from the main text, PCSL repairs the *asymmetric edges* in S^0 to obtain a new skeleton S^1.

Compared to S^0, S^1 more closely approximates the true skeleton. Subsequently, based on Equation (10) from the main text, PCSL uses the Bootstrap method to generate a batch of higher-quality sub-datasets $\mathcal{D}^2$ (i.e., $DE(\mathcal{D}^2|S^1) > DE(\mathcal{D}^1|S^0)$). PCSL then re-learns the local causal skeletons for each variable involved in the four *asymmetric edges* on each sub-dataset in $\mathcal{D}^2$. Again, using Equation (5) and Criterion 1, PCSL repairs the *asymmetric edges* in S^1 to obtain a new skeleton S^2.

S^2 is even closer to the true skeleton than S^1. At this point, the PCSL algorithm has successfully removed the edges between X_1 and X_5, and between X_5 and X_6. However, it has erroneously deleted the edge between X_2 and X_3.

Next, using S^2 as a reference and based on Equation (10), PCSL generates another batch of higher-quality sub-datasets $\mathcal{D}^3$ through the Bootstrap method (i.e., $DE(\mathcal{D}^3|S^2) > DE(\mathcal{D}^2|S^1)$). PCSL re-learns the local causal skeletons for each variable involved in the four *asymmetric edges* on each sub-dataset in $\mathcal{D}^3$. Finally, using Equation (5) and Criterion 1, PCSL repairs the *asymmetric edges* in S^2 to obtain the final skeleton S^{best}.

At this point, PCSL can no longer generate sub-datasets of higher quality than $\mathcal{D}^3$ within the tolerance range. Therefore, the progressive learning strategy terminates. This example demonstrates how PCSL iteratively refines the causal skeleton, progressively improving its accuracy through multiple rounds of learning on increasingly higher-quality datasets.

S-10: Time Complexity of PCSL

Phase 3 of PCSL performs the score-and-search strategy on the given best global skeleton S^{best} rather than on an empty graph. It means that during the search process, PCSL does not need to perform *adding edges* and *removing edges* operations, but only needs to perform *reversing edges* operation to achieve the highest score, that is, the entire search space is very small. Thus, the time complexity of PCSL mainly lies in Phase 1 and Phase 2, and the computational cost of these two phases is measured via the number of CI (conditional independence) tests. Let p denote the largest size of the learned PC (Parent-Child) set of any variable in a dataset. For Phase 1 in PCSL, the time complexity of the PC learning process of any variable is $O(2^p m)$ CI tests, and thus the time complexity of Phase 1 is $O(2^p m^2)$ CI tests. In Phase 2, let the number of *asymmetric edges* with respect to the original dataset be K and the number of iterations in Phase 2 (or the number of batches of the sampled sub-datasets) be L, the time complexity of each iteration is $O(2KN2^p m)$ CI tests. Thus, the time complexity of Phase 2 is $O(2KNL2^p m)$ CI tests. Normally, $L \leq 7$, and N is always set to 15 (see Section S-3 in the Supplementary Material for details). Let $v = max\{2K, m\}$, then the overall time complexity of PCSL is $O(2^p mv)$ CI tests.

S-11: Convergence Analysis of PCSL

In this section, we present a comprehensive analysis of the convergence properties of PCSL. We begin with a theoretical

Algorithm 1: Progressive Causal Structure Learning

Input: D_{orig}: an original dataset with the variable set $\boldsymbol{V}$=$\{X_1, X_2, ..., X_m\}$ and n samples; N: the number of datasets in each batch of sub-datasets; r: tolerance

Output: $\mathbb{G}^*$: the final causal structure

{*Phase 1: Initial local skeleton learning*}
for *d=1 to m* **do**
  $\boldsymbol{PC}(X_d)$=HITON-PC(D_{orig}, X_d)
end
for *d=1 to m; f=1 to (d-1)* **do**
  if $(X_d \in \boldsymbol{PC}(X_f) \wedge X_f \notin \boldsymbol{PC}(X_d)) \vee (X_d \notin \boldsymbol{PC}(X_f) \wedge X_f \in \boldsymbol{PC}(X_d))$ **then**
    Record an *asymmetric edge* $X_d \nleftrightarrow X_f$
  end
end
{*Phase 2: Progressive global skeleton construction*}
$i = 1$ /*batch index of sub-datasets*/
Use Bootstrapping to generate the i-th batch of sub-datasets $\mathcal{D}^i = \{D_1^i, D_2^i, ..., D_N^i\}$;
while $r > 0$ **do**
  for *each* $X_d \nleftrightarrow X_f$ **do**
    for *j=1 to* N **do**
      $\boldsymbol{PC}(X_d)$=HITON-PC(D_j^i, X_d)
      $\boldsymbol{PC}(X_f)$=HITON-PC(D_j^i, X_f)
    end
    if $AEE(:, k) > 0$ **then**
      $S^i(d, f) = S^i(f, d) = 1$
    else
      $S^i(d, f) = S^i(f, d) = 0$
    end
  end
  Generate the $(i+1)$-th batch of sub-datasets $\mathcal{D}^{i+1}$
  if $DE(\mathcal{D}^{i+1}|S^i) > DE(\mathcal{D}^i|S^{i-1})$ **then**
    $S^{best} = S^i$; $\mathcal{D}^{best} = \mathcal{D}^{i+1}$
    $i = i + 1$
  else
    $r = r - 1$
  end
end
{*Phase 3: Integrated global skeleton orientation*}
for *j=1 to* N **do**
  $\mathcal{A}_j \xleftarrow{D_j^{best}} S^{best}$ /*greedy search and scoring*/
end
Let $\mathcal{A}^* = zeros(m, m)$ /*an empty graph*/
$\mathcal{A}^* = \dfrac{\mathcal{A}_1 \oplus \mathcal{A}_2 \oplus \cdots \oplus \mathcal{A}_N}{N}$ /*integration*/
for *a=1 to m* **do**
  for *b=1 to m* **do**
    if $\mathcal{A}^*(a, b) \geq 0.5$ **then**
      $\mathcal{A}^*(a, b) = 1$ /*$X_a \rightarrow X_b$*/
    else
      $\mathcal{A}^*(a, b) = 0$ /*$X_a \nrightarrow X_b$*/
    end
  end
end
$\mathbb{G}^* = acyclic\_constraint(\mathcal{A}^*)$;
return $\mathbb{G}^*$

examination of the convergence in Phase 2 of PCSL, followed by empirical validation through extensive experiments.

Theoretical Convergence Analysis: Theorem 3 establishes the theoretical foundation for the convergence of PCSL in Phase 2. Specifically, it demonstrates that as the number of iterations L approaches infinity, the probability of Phase 2 in PCSL converging approaches 1. This theoretical result provides a strong basis for the reliability and stability of our method.

Theorem 3 (Convergence of Phase 2 in PCSL)**.** *Let $\mathcal{P}^i$ be the probability of successfully generating a batch of sub-datasets $\mathcal{D}^{i+1}$ with higher quality than the current batch $\mathcal{D}^i$ in the i-th iteration (i.e., $DE(\mathcal{D}^{i+1}|S^i) > DE(\mathcal{D}^i|S^{i-1})$ holds in Step 3 of Phase 2). Assume that $\mathcal{P}^i$ decreases with each iteration, i.e., $\mathcal{P}^i > \mathcal{P}^{i+1}$ for $\forall i \in \{1, 2, ..., L-1\}$, where L ($L > r$) is the total number of iterations. This assumption is based on the increasing difficulty of generating higher quality sub-datasets as the quality of $\mathcal{D}^i$ improves with each iteration*[5].

Then, as L approaches infinity, the probability of Phase 2 in PCSL terminating after L iterations approaches 1, i.e., the probability of Phase 2 in PCSL converging approaches 1.

Proof. First, let's clearly define our terms:

- "Failing" or "failure" in an iteration means PCSL is unable to generate a batch of higher quality sub-datasets than the previous one in that iteration.
- "Success" means successfully generating a batch of higher quality sub-datasets in an iteration.

Let E_L be the event that PCSL has not converged after L iterations. We need to prove that

$$\lim_{L \to \infty} P(E_L) = 0. \tag{12}$$

Let X be a random variable representing the number of failures (as defined above) in L iterations. For PCSL to not converge after L iterations, it must fail at most $(r-1)$ times in these L iterations. If it fails r times, PCSL will terminate and thus converge. Therefore, E_L is equivalent to the event $X < r$. The probability of failing exactly k times in L iterations is given by the binomial probability:

$$P(X = k) = \binom{L}{k} (1-p)^k p^{L-k}, \tag{13}$$

where $p = \min_{i=1}^{L} \mathcal{P}^i$ (we use the minimum probability to get a lower bound on the success probability). Therefore, the probability of not converging after L iterations is:

$$\begin{aligned} P(E_L) &= P(X < r) \\ &= \sum_{k=0}^{r-1} P(X = k) \\ &= \sum_{k=0}^{r-1} \binom{L}{k} (1-p)^k p^{L-k}. \end{aligned} \tag{14}$$

[5]Note that although the accuracy of the global skeleton S^i also improves with each iteration, this improvement in S^i accuracy only leads to a more accurate measurement of dataset quality but does not make it easier to generate a batch of sub-datasets with higher quality.

This is equivalent to the cumulative probability of having fewer than r failures in L trials of a Bernoulli process with failure probability $(1-p)$. By the law of large numbers, as $L \to \infty$, the proportion of failures converges in probability to $(1-p)$:

$$\frac{X}{L} \xrightarrow{P} (1-p). \tag{15}$$

This means that for any $\epsilon > 0$, we have:

$$\lim_{L\to\infty} P\left(\left|\frac{X}{L} - (1-p)\right| < \epsilon\right) = 1. \tag{16}$$

Choose $\epsilon = (1-p) - \frac{r-1}{L}$ for large enough L such that

$$\frac{r-1}{L} < (1-p). \tag{17}$$

This choice is possible because r is fixed and $p < 1$, so for sufficiently large L, $\frac{r-1}{L}$ will be arbitrarily close to 0 and thus less than $(1-p)$. With this choice of ϵ, we have:

$$\lim_{L\to\infty} P\left(\frac{X}{L} > \frac{r-1}{L}\right) = 1. \tag{18}$$

This is equivalent to:

$$\lim_{L\to\infty} P(X > r-1) = 1. \tag{19}$$

Therefore:

$$\lim_{L\to\infty} P(X < r) = \lim_{L\to\infty} (1 - P(X > r-1)) = 0. \tag{20}$$

Due to $P(X < r) = P(E_L)$, we can obtain:

$$\lim_{L\to\infty} P(E_L) = \lim_{L\to\infty} P(X < r) = 0. \tag{21}$$

Thus, as the number of iterations L approaches infinity, the probability of PCSL not converging (i.e., failing fewer than r times) approaches 0, or equivalently, the probability of convergence approaches 1. □

Empirical Convergence and Consistency Analysis: To further validate the practical convergence of Phase 2 in PCSL and analyze its consistency, we conducted a series of experiments using two benchmark BN datasets: Alarm and Child. Our experimental procedure was as follows:

- For each dataset, we generated multiple batches of data with sample sizes ranging from 300 to 15,000.
- We ran PCSL on each dataset, recording two key metrics: a) Ar_F1 metric, representing the accuracy of the learned causal structure. b) The number of iterations required in Phase 2 of PCSL.

The results of these experiments are presented in Figures 7 and 8.

Consistency Analysis: Figure 7 illustrates the relationship between sample size and Ar_F1 metric. The results demonstrate that as the sample size increases, the accuracy of the causal structure learned by PCSL consistently improves across both datasets. Specifically, we observe that the Ar_F1 metric steadily approaches 1, indicating that the learned structure increasingly approximates the true causal structure. This trend reflects the consistency of our method under finite sample conditions, highlighting its ability to recover the true causal relationships given sufficient data.

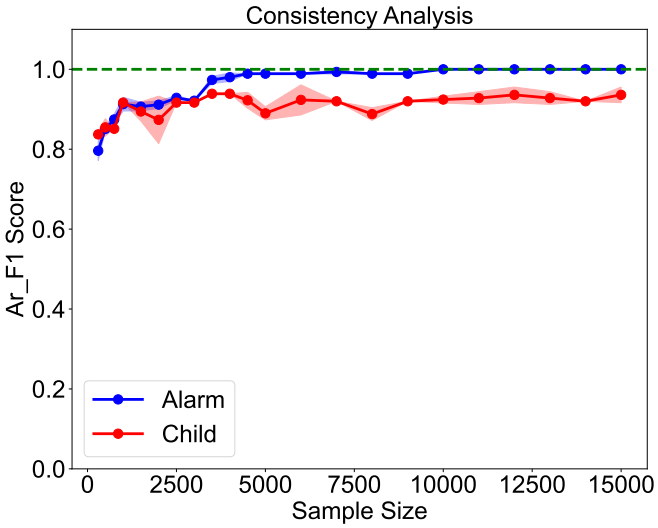


Fig. 7. Consistency Analysis of PCSL on Alarm and Child datasets.

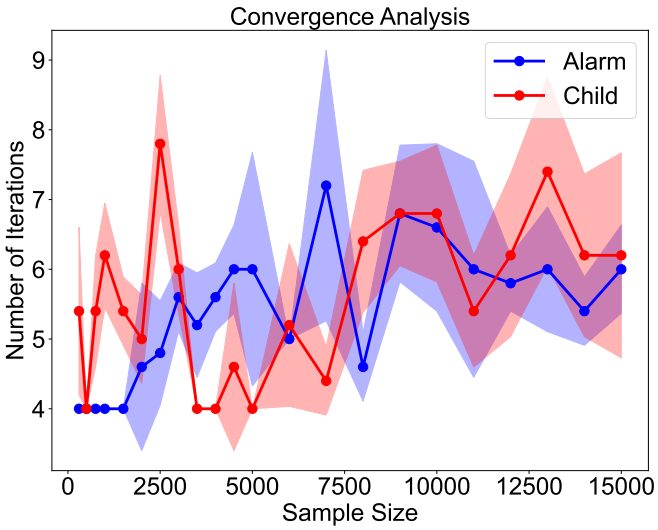


Fig. 8. Convergence analysis of Phase 2 in PCSL on Alarm and Child datasets.

Convergence Analysis: Figure 8 depicts the relationship between sample size and the number of iterations in Phase 2 of PCSL. Notably, we observe that regardless of the sample size, the average number of iterations in Phase 2 remains consistently below 7.5. This finding empirically demonstrates the rapid convergence of our method in practical applications.

Furthermore, our experimental results align with the implications of Theorem 3, which suggests that the convergence speed of Phase 2 in PCSL is independent of sample size. This theoretical prediction is corroborated by our empirical observations in Figure 8, where we see no significant correlation between sample size and the number of iterations required for convergence.

In conclusion, our theoretical analysis, supported by comprehensive empirical evidence, establishes PCSL as a consistent and rapidly converging method for causal structure learning.

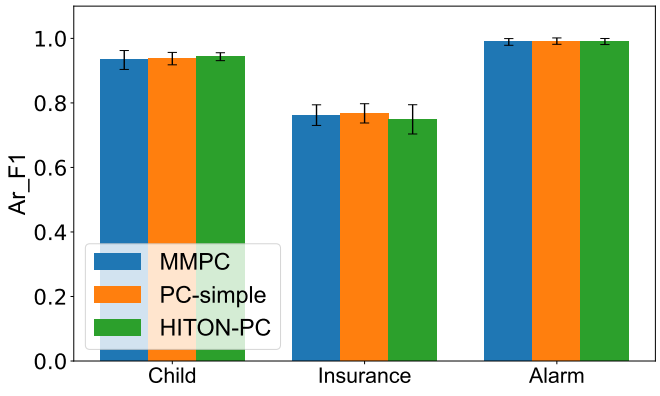


Fig. 9. Ar_F1s of PCSL with different initial skeleton learning algorithms across three benchmark datasets.

S-12: Sensitivity Analysis of PCSL to Initial Skeleton Accuracy

In this section, we analyze our algorithm's sensitivity to the accuracy of the initial skeleton through a series of experiments. This analysis aims to demonstrate the robustness of PCSL to different initial conditions, addressing concerns about the algorithm's dependence on the quality of the initial skeleton.

Specifically, we conducted our experiments using three benchmark Bayesian networks (BNs): Child, Insurance, and Alarm[6]. For each BN, we generated multiple datasets, each containing 5,000 samples. This sample size was chosen to ensure reliable results while maintaining computational feasibility. To assess the impact of different initial skeletons on PCSL's performance, we employed three distinct local causal skeleton learning algorithms in Phase 1 of PCSL: MMPC [5], PC-simple [6] and HITON-PC [7]. For each dataset and each local causal skeleton learning algorithm, we executed the entire PCSL to learn the causal structure. Then, we evaluated the accuracy of the learned structures using the widely-adopted Ar_F1 metric.

The experimental results are presented in Figure 9, and we can observe that regardless of which local causal skeleton learning algorithm is employed, the accuracy of the final causal structure achieved by PCSL remains consistently high. These findings provide compelling evidence that PCSL is capable of overcoming potential limitations or biases introduced by the initial skeleton learning phase. The progressive learning strategy allows PCSL to iteratively refine the causal structure, converging towards a high-quality solution.

S-13: Detailed Analysis of the Causes of *Asymmetric Edges*

In this section, we provide an in-depth analysis of the factors contributing to the formation of *asymmetric edges* in local-to-global causal structure learning. We focus on two main causes: limited sample sizes, and violations of causal assumptions.

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

Limited Sample Sizes: When the original dataset has a small sample size, the conditional independence tests performed in learning each variable's local causal skeleton may become unreliable, leading to Type II errors (missing true causal neighbors of the target variable) in conditional independence tests. Specifically, to perform a reliable conditional independence 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 [8]:

$$\frac{n}{C_{X_d} \times C_{X_f} \times C_{\boldsymbol{S}}} \geq t, \tag{22}$$

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. When the original dataset has a small sample size, the condition in Eq. (22) may not be met, leading to the direct omission of true causal neighbors of the target variable, resulting in *asymmetric edges*. Inspired by existing works [9], [10] demonstrating the practicality of Bootstrapping in small sample scenarios, we employ Bootstrapping to address the *asymmetric edge* problem caused by small sample datasets.

Violations of Causal Assumptions: It is important to clarify that our proposed PCSL algorithm is built upon the Faithfulness and causal Markov assumptions [11], which are common foundations for most existing causal structure learning algorithms. However, in practical datasets (both simulated and real-world), these assumptions may not hold perfectly due to limited sample sizes. This can lead to both Type I errors (learning extra false causal neighbors for the target variable) and Type II errors (missing true causal neighbors of the target variable) in conditional independence tests, resulting in *asymmetric edges*. To address this issue, we employ Bootstrapping to modify the distribution of generated sub-datasets (aiming to better satisfy the Faithfulness and causal Markov assumptions) and combine this with ensemble learning principles to correct deficiencies in the original dataset, thereby resolving the problem of *asymmetric edges*.

S-14: Effectiveness of the Progressive Strategy

As described in Theorem 1, the progressive strategy of PCSL is theoretically effective. In this section, we use the experimental results on the benchmark datasets to further verify the effectiveness of the progressive strategy of PCSL. Specifically, we record the correction accuracy[7] of *asymmetric edges* in each iteration during the progressive learning, as well as the quality of the sampled sub-datasets in each iteration during the progressive learning. As shown in Figure 10, we reports the trends in the correction accuracy of *asymmetric edges* and the quality of the sampled sub-datasets as PCSL learns the causal structures on the benchmark datasets. In Figure 10, *Qsd* (marked red) and *Caae* (marked green) in the legend represent the quality of the sampled sub-datasets and

[7]The proportion of correctly preserved or removed *asymmetric edges* to the total number of *asymmetric edges*.

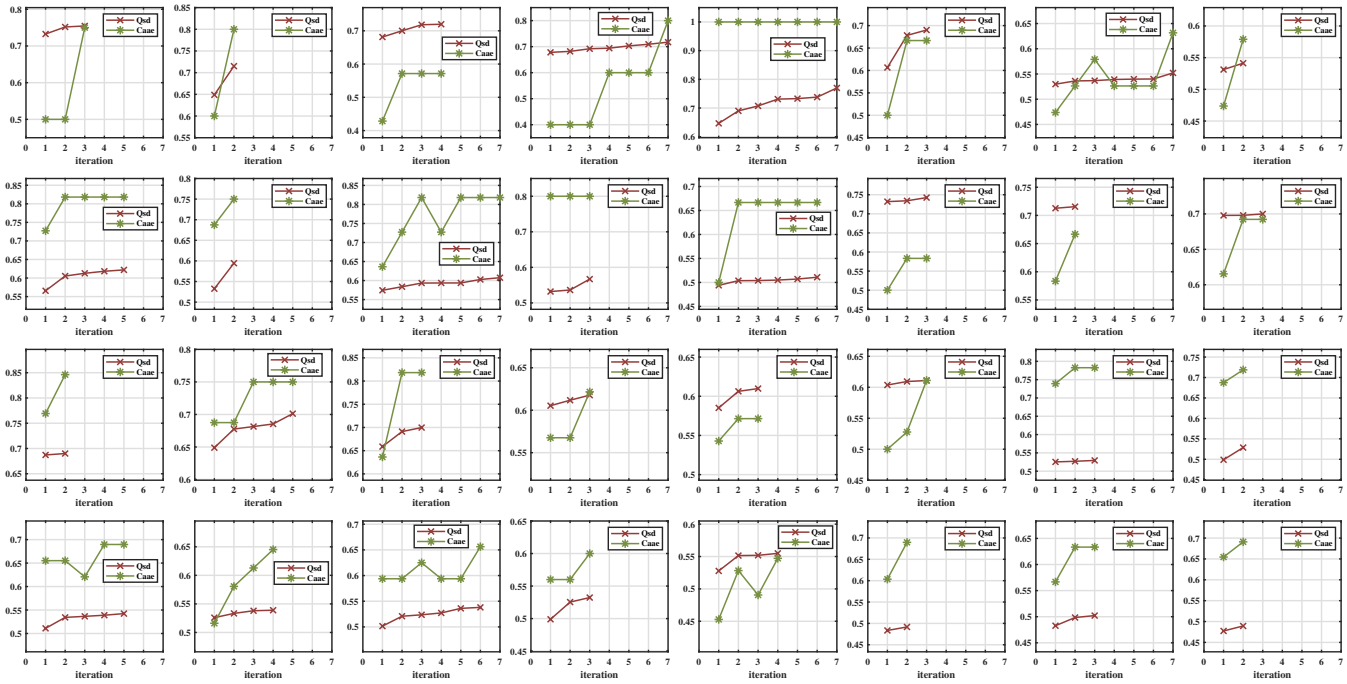


Fig. 10. Verification of the effectiveness of the progressive strategy on the benchmark datasets. *Qsd* and *Caae* in the legend represent the quality of the sampled sub-datasets and the correction accuracy of *asymmetric edges*, respectively. The horizontal axis denotes the number of iterations of PCSL in the progressive global skeleton construction phase.

the correction accuracy of *asymmetric edges*, respectively, and the horizontal axis denotes the number of iterations of PCSL in the progressive global skeleton construction phase. For the convenience of observation, we only record the experimental results with more than 2 iterations. From Figure 10, we see that:

- On all datasets, the number of iterations of the progressive learning process is less than or equal to 7, which indicates that the time complexity of PCSL is very acceptable in practice.
- Based on the constraint of condition "$DE(\mathcal{D}^{i+1}|S^i) > DE(\mathcal{D}^i|S^{i-1})$" in Step 3 of Phase 2, the quality of each batch of the sampled sub-datasets increases monotonically with the increase of iterations.
- Although the correction accuracy of *asymmetric edges* occasionally decreases with the increase of iterations, the overall trend in the correction accuracy of *asymmetric edges* is upward. Moreover, the final correction accuracy of *asymmetric edges* on each dataset is greater than 50%.

REFERENCES

[1] I. Ng, A. Ghassami, and K. Zhang, "On the role of sparsity and DAG constraints for learning linear DAGs," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 943–17 954, 2020.

[2] X. Guo, Y. Wang, X. Huang, S. Yang, and K. Yu, "Bootstrap-based causal structure learning," in *Proceedings of ACM International Conference on Information & Knowledge Management*, 2022, pp. 656–665.

[3] S. Yang, Y. Zhang, H. Wang, P. Li, and X. Hu, "Representation learning via serial robust autoencoder for domain adaptation," *Expert Systems with Applications*, vol. 160, p. 113635, 2020.

[4] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[5] I. Tsamardinos, C. F. Aliferis, and A. Statnikov, "Time and sample efficient discovery of Markov blankets and direct causal relations," in *Proceedings of ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2003, pp. 673–678.

[6] J. Li, L. Liu, T. D. Le, J. Li, L. Liu, and T. D. Le, "Local causal discovery with a simple pc algorithm," *Practical Approaches to Causal Relationship Exploration*, pp. 9–21, 2015.

[7] 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." *Journal of Machine Learning Research*, vol. 11, no. 1, 2010.

[8] S. Yaramakala and D. Margaritis, "Speculative Markov blanket discovery for optimal feature selection," in *IEEE International Conference on Data Mining*. IEEE, 2005, pp. 4–pp.

[9] E. Amalnerkar, T. H. Lee, and W. Lim, "Reliability analysis using bootstrap information criterion for small sample size response functions," *Structural and Multidisciplinary Optimization*, vol. 62, pp. 2901–2913, 2020.

[10] A. K. Dwivedi, I. Mallawaarachchi, and L. A. Alvarado, "Analysis of small sample size studies using nonparametric bootstrap test with pooled resampling method," *Statistics in medicine*, vol. 36, no. 14, pp. 2187–2205, 2017.

[11] J. Pearl, *Causality*. Cambridge University Press, 2009.