Appendices

Appendix organization:

- Appendix VII-A: Proof of Lemma 1
- Appendix VII-B: Proof of Lemma 2
- Appendix VII-C: Proof of Lemma 3
- Appendix VII-D: Proof of Lemma 4
- Appendix VII-E: Proof of Lemma 5
- Appendix VII-F: Proof of Lemma 6
- Appendix VII-G: Experiment Setting
- Appendix VII-H: Non-Partitioning Methods of Comparison
- Appendix VII-I: Ground Truth for Optimal Causal Partitioning
- Appendix VII-J: PC Algorithm v.s. CPA on Andes and Diabetes
- Appendix VII-K: Results on the Sensitivity of CPA to Average Degree
- Appendix VII-L: Results on Different Causal Functions and Noise Distributions
- Appendix VII-M: Results on Partitioning over Diabetes and Link

A. Proof of Lemma 1

Lemma 1. Let $G_{\mathcal{T}}$ be the true causal graph of the observed data D, G be a superstructure of $G_{\mathcal{T}}$ based on D, if $\mathbb{P}$ is a PARS over G, then $\mathbb{P}$ is also a causal partitioning over $G_{\mathcal{T}}$.

Proof. First, consider the d-separation of two non-adjacent variables v_i and v_j in V_1. As $V_1 = A \cup C$, we need to discuss the d-separation with respect to 1) $\forall v_i, v_j \in A$, 2) $\forall v_i, v_j \in C\backslash_A$, 3) $\forall v_i \in A, \forall v_j \in C\backslash_A$. We divide N into two parts $N_A \subset A$ and $N_B \subset B$. We can observe that

1) For $\forall v_i, v_j \in A$, then the neighbors set Z of v_i and v_j must be contained in A or A's neighbors. Because all A's neighbors are contained in $A \cup C$, then there is at least one set S, $S \subseteq Z \subset A \cup C = V_1$ such that v_i and v_j are d-separated by S;
2) For $\forall v_i, v_j \in C\backslash_A = M \cup N_B$, there are three cases (i) $v_i, v_j \in M$, (ii) $v_i \in M, v_j \in N_B$ and (iii) $v_i, v_j \in N_B$. If $v_i, v_j \in M$, then their d-separator S is contained in M's neighbors N where $N \subset C$, therefore v_i and v_j is d-separable in C; If $v_i \in M, v_j \in N_B$, then the neighbors set Z of v_i and v_j must be contained in C or B. So there is at least one set S, $S \subseteq Z \subset B \cup C = V_2$ such that v_i and v_j are d-separated by S; If $v_i, v_j \in N_B$, similarly, there is at least one set S, $S \subseteq Z \subset B \cup C = V_2$ such that v_i and v_j are d-separated by S;
3) For $\forall v_i \in A, \forall v_j \in C\backslash_A$, because A and N_B are separated by M, we need to consider only the case of $v_i \in A, v_j \in M$, we can see that their d-separator S is contained in $A \cup C$, therefore v_i and v_j is d-separable in $A \cup C = V_1$.

Similarly, we can prove the case of V_2 and then complete this proof. □

B. Proof of Lemma 2

Lemma 2. Given a connected undirected graph $G = \{V, E\}$ ($|V| \geq 2$) and its adjoint graph $G_{\mathcal{A}} = \{V_{\mathcal{A}}, E_{\mathcal{A}}\}$, if there is an edge-cut set $C_{\mathcal{A}}$ on $G_{\mathcal{A}}$ dividing $V_{\mathcal{A}}$ into $V_{\mathcal{A}} = \{A_{\mathcal{A}}, B_{\mathcal{A}}\}$ such that $A_{\mathcal{A}} \neq \emptyset$, $B_{\mathcal{A}} \neq \emptyset$ and any edge bridging $A_{\mathcal{A}}$ and $B_{\mathcal{A}}$ is contained in $C_{\mathcal{A}}$, then $V = \{A, M, B\}$ satisfying $A \cap M = B \cap M = \emptyset$ and either $A = \emptyset$ or $B = \emptyset$ or $\forall v_i \in A$ is nonadjacent to $\forall v_j \in B$, where A, M and B are obtained by following:

- $\forall e_k^* \in C_{\mathcal{A}}$ corresponds to two edges and one common vertex in G, denote all these common vertices by M;
- $\forall v_i^* \in A_{\mathcal{A}}\backslash_{V_{C_{\mathcal{A}}}}$ corresponds to an edge with two vertices in G, denote all these vertices disjoint to M by A, in which $V_{C_{\mathcal{A}}}$ denotes the vertices of $V_{\mathcal{A}}$ involved in $C_{\mathcal{A}}$;
- $\forall v_j^* \in B_{\mathcal{A}}\backslash_{V_{C_{\mathcal{A}}}}$ corresponds to an edge with two vertices in G, denote all these vertices disjoint to M by B.

Proof. We give a proof by contradiction. According to the generation process of M, A and B, we know $A \cap M = B \cap M = \emptyset$. If neither $A = \emptyset$ nor $B = \emptyset$, we need to prove that $\forall v_i \in A$ is nonadjacent to $\forall v_j \in B$.

1) Assume $\exists v_k \in A$ and $\exists v_k \in B$, then $\exists v_i^* \in A_{\mathcal{A}}\backslash_{V_{C_{\mathcal{A}}}}$ corresponds to an edge $e(i,k) \in E$ and $\exists v_j^* \in B_{\mathcal{A}}\backslash_{V_{C_{\mathcal{A}}}}$ corresponds to an edge $e(j,k) \in E$. We can see that $e(i,k)$ and $e(j,k)$ are connected by v_k, and further deduce that v_i^* and v_j^* are adjacent. Since $v_k \notin M$, the edge between v_i^* and v_j^* in $G_{\mathcal{A}}$ is not contained in $C_{\mathcal{A}}$, thus there is contradiction.

2) Assume $\exists v_k \in A$ and $\exists v_t \in B$ and they are adjacent, then $\exists v_i^* \in A_{\mathcal{A}}\backslash_{V_{C_{\mathcal{A}}}}$ corresponds to an edge $e(i,k) \in E$ and $\exists v_j^* \in B_{\mathcal{A}}\backslash_{V_{C_{\mathcal{A}}}}$ corresponds to an edge $e(j,t) \in E$. We can deduce that there is an edge $e(k,t) \in E$, and $e(k,t)$ correspond to a vertex in $G_{\mathcal{A}}$, denote it by $v_k^* \in V_{\mathcal{A}}$. As $V_{\mathcal{A}} = \{A_{\mathcal{A}}, B_{\mathcal{A}}\}$, without loss of generality, assume $v_k^* \in A_{\mathcal{A}}$, then v_k^* and v_j^* are adjacent. We can see that if the edge between v_k^* and v_j^* contained in $V_{C_{\mathcal{A}}}$, then $\exists v_t \in M$, which is contradictory to $B \cap M = \emptyset$.

We then complete this proof. □

C. Proof of Lemma 3

Lemma 3. For an arbitrary causal-cut C of $G = \{V, E\}$ with its corresponding edge-cut $C_{\mathcal{A}}$ of $G_{\mathcal{A}}$, if there is a causal-cut C^* obtained by adding an arbitrary vertex $v_i \in V\backslash_C$ to C and also correspond to an edge-cut $C_{\mathcal{A}}^*$ of $G_{\mathcal{A}}$, then

$$\phi(G) < \phi(G^*) \text{ and } \psi(G_{\mathcal{A}}) \leq \psi(G_{\mathcal{A}}^*) \tag{2}$$

where $\phi(\cdot)$ and $\psi(\cdot)$ follow Def. 5 and Def. 6, respectively. $\psi(G_{\mathcal{A}}) = \psi(G_{\mathcal{A}}^*)$ if and only if the added vertex is nonadjacent to C.

Proof. According to Def. 5 and Def. 6, we have

$$\begin{aligned} \phi(G) &= \frac{|C|}{min(|V_1|, |V_2|)}, \\ \phi(G^*) &= \frac{|C^*|}{min(|V_1^*|, |V_2^*|)}, \\ \psi(G_{\mathcal{A}}) &= \frac{|C_{\mathcal{A}}|}{min(|A_{\mathcal{A}}|, |B_{\mathcal{A}}|)}, \\ \psi(G_{\mathcal{A}}^*) &= \frac{|C_{\mathcal{A}}^*|}{min(|A_{\mathcal{A}}^*|, |B_{\mathcal{A}}^*|)}, \end{aligned} \tag{3}$$

where $A_{\mathcal{A}}^*$ and $B_{\mathcal{A}}^*$ are two subsets divided by $C_{\mathcal{A}}^*$, and V_1^* and V_2^* are two partitions w.r.t. C^*.

Since C^* is obtained by adding a vertex $\forall v^i \in V \backslash_C$ to C, then $|C^*| = |C| + 1$, and further have either $|A^*| = |A| - 1$ or $|B^*| = |B| - 1$. Consequently, $\phi(G) < \phi(G^*)$.

If the added vertex v_i is nonadjacent to C, then there is no new edge between v_i and any vertex in C, therefore $|C_{\mathcal{A}}| = |C^*_{\mathcal{A}}|$, $|A_{\mathcal{A}}| = |A^*_{\mathcal{A}}|$ and $|B_{\mathcal{A}}| = |B^*_{\mathcal{A}}|$, that is $\psi(G_{\mathcal{A}}) = \psi(G^*_{\mathcal{A}})$.

Else if v_i is adjacent to C, then there is at lest one new edge between v_i and one vertex of C, thus $|C_{\mathcal{A}}|+1 \leq |C^*_{\mathcal{A}}|$, $|A| \leq |A^*|$ and $|B| \leq |B^*|$. Consequently, $\psi(G_{\mathcal{A}}) < \psi(G^*_{\mathcal{A}})$.

We then complete this proof. □

D. Proof of Lemma 4

Lemma 4: Given the adjoint graph $G_{\mathcal{A}}$, let L be the Laplacian matrix of $G_{\mathcal{A}}$ with eigenvalue $\lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$, then $\psi_{min}(G_{\mathcal{A}}) \geq \frac{1}{2}\lambda_2$.

Proof [39]. Here first present the two well-known theorems, Courant-Fischer Formula and Rayleigh Quotient used in proof.

Courant-Fischer Formula. *Let A be an $n \times n$ symmetric matrix with eigenvalue $\lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$ and corresponding eigenvectors $v_1, ..., v_2$. Then*

$$\begin{aligned}
\lambda_1 &= \min_{\|x\|=1} x^T A x = \min_{x \neq 0} \frac{x^T A x}{x^T x}, \\
\lambda_2 &= \min_{\|x\|=1, x \perp\!\!\!\perp v_1} x^T A x = \min_{x \neq 0, x \perp\!\!\!\perp v_1} \frac{x^T A x}{x^T x}, \\
\lambda_n &= \lambda_{max} = \max_{\|x\|=1} x^T A x = \max_{x \neq 0} \frac{x^T A x}{x^T x},
\end{aligned} \tag{4}$$

Rayleigh Quotient. *Let $G = \{V, E\}$ be a graph and L be the Laplacian of G. Given the smallest eigenvalues $\lambda_1 = 0$ with its eigenvector $v_1 = \mathbf{1}$. Then by Courant-Fischer Formula, we have*

$$\begin{aligned}
\lambda_2 &= \min_{x \neq 0, x \perp\!\!\!\perp v_1} \frac{x^T A x}{x^T x} \\
&= \min_{x \neq 0, x \perp\!\!\!\perp \mathbf{1}} \frac{\sum_{i,j \in E} (x_i - x_j)^2}{\sum_{i \in V} x_i^2}
\end{aligned} \tag{5}$$

Now, we let L be the Laplacian of $G_{\mathcal{A}}$. Given the smallest eigenvalue $\lambda_1 = 0$ and its eigenvector $\mathbf{v}_1 = \mathbf{1}$. Then, by the Rayleigh Quotient we have

$$\begin{aligned}
\lambda_2 &= \min_{x \neq 0, x \perp\!\!\!\perp \mathbf{v}_1} \frac{x^T A x}{x^T x} \\
&= \min_{x \neq 0, x \perp\!\!\!\perp \mathbf{1}} \frac{\sum_{i,j \in E} (x_i - x_j)^2}{\sum_{i \in V} x_i^2}.
\end{aligned} \tag{6}$$

We associate the vertices of $G_{\mathcal{A}}$ with a vector $x \in \{-1, 1\}^n$, where $n = |V_{\mathcal{A}}|$ and

$$x_i = \begin{cases} 1, i \in A_{\mathcal{A}} \\ -1, i \in B_{\mathcal{A}}. \end{cases} \tag{7}$$

Then, we can easily obtain

$$|C_{\mathcal{A}}| = \frac{1}{4} \sum_{i,j \in E_A} (x_i - x_j)^2, \tag{8}$$

and

$$|A_{\mathcal{A}}| \cdot |B_{\mathcal{A}}| = \frac{1}{4} \sum_{i<j} (x_i - x_j)^2. \tag{9}$$

Combining the two equations above, we have

$$\begin{aligned}
\psi_{min}(G_{\mathcal{A}}) &= \min_{C_{\mathcal{A}} \subset E_{\mathcal{A}}} \frac{|C_{\mathcal{A}}|}{min(|A_{\mathcal{A}}|, |B_{\mathcal{A}}|)} \\
&= \min_{x \in \{-1,1\}^n} \frac{\sum_{i,j \in E_{\mathcal{A}}} (x_i - x_j)^2}{\sum_{i<j} (x_i - x_j)^2}.
\end{aligned} \tag{10}$$

As the value of $\psi_{min}(G_A)$ depends only on the difference of $x_i - x_j$, we can relax Equ. (10) to

$$\min_{x \in \mathbb{R}^n} \frac{\sum_{i,j \in E_A} (x_i - x_j)^2}{\sum_{i<j} (x_i - x_j)^2}, \tag{11}$$

where $x \in \mathbb{R}^n$, $x \perp \mathbf{1}$, i.e., $\sum_{i=1}^n x_i = 0$, then

$$\sum_{i<j} (x_i - x_j)^2 = n \sum_{i=1}^n x_i^2. \tag{12}$$

By using the Rayleigh Quotient, we have

$$\begin{aligned}
\min_{x \in \mathbb{R}^n} \frac{\sum_{i,j \in E_A} (x_i - x_j)^2}{\sum_{i<j} (x_i - x_j)^2} &= \min_{x \in \mathbb{R}^n, x \perp \mathbf{1}} \frac{\sum_{i,j \in E_A} (x_i - x_j)^2}{n \sum_{i=1}^n x_i^2} \\
&= \min_{x \neq 0, x \perp\!\!\!\perp \mathbf{v}_1} \frac{x^T A x}{x^T x} = \frac{\lambda_2}{n},
\end{aligned} \tag{13}$$

where λ_2 is the second smallest eigenvalue of the Laplacian matrix of $G_{\mathcal{A}}$. Then, put all above together, we obtain

$$\begin{aligned}
&\psi_{min}(G_{\mathcal{A}}) \\
=&\frac{n}{2} \min_{C_{\mathcal{A}} \subset E_{\mathcal{A}}} \frac{|C_{\mathcal{A}}|}{min(|A_{\mathcal{A}}|, |B_{\mathcal{A}}|)} \\
\geq&\frac{n}{2} \min_{C_{\mathcal{A}} \subset E_{\mathcal{A}}} \frac{|C_{\mathcal{A}}|}{|A_{\mathcal{A}}| \cdot |B_{\mathcal{A}}|} \\
=&\frac{n}{2} \min_{x \in \{-1,1\}^n} \frac{\sum_{i,j \in E_A} (x_i - x_j)^2}{\sum_{i<j} (x_i - x_j)^2} \\
\geq&\frac{n}{2} \min_{x \in \mathbb{R}^n} \frac{\sum_{i,j \in E_A} (x_i - x_j)^2}{\sum_{i<j} (x_i - x_j)^2} \\
=&\frac{n}{2} \min_{x \in \mathbb{R}^n, x \perp \mathbf{1}} \frac{\sum_{i,j \in E_A} (x_i - x_j)^2}{n \sum_{i=1}^n x_i^2} \\
=&\frac{1}{2}\lambda_2.
\end{aligned} \tag{14}$$

Then complete this proof. □

E. Proof of Lemma 5

Lemma 5: The time complexity of Alg. 1 (i.e. CPA without subroutine) is $O(n^{2+k_{par}}T)$, and that of Alg. 2 (i.e. CPA with subroutine A_g) is $O(n^{2+k_{par}}T)$ plus $O(A_g)$ regarding m, T denotes the time complexity of the used CI test method, $n = |V|$, $m = \max(|V_1|, |V_2|)$.

Proof. As shown in Alg. 1, CPA has two main parts, first construct the superstructure G based on the given variable set V ($|V| = n$), then achieve edge-cut set on the adjoint graph $G_{\mathcal{A}}$. For the first part, we need to check CIs between every two variables and thus require at most $C_{n-2}^0 + \cdots + C_{n-2}^{k_{par}}$ times CI tests, where k_{par} denotes the maximum order of CI test for constructing the superstructure for partitioning. Generally, k_{par} = 1∼2 as suggested by the previous works [21], [22]. There are C_n^2 pairs of variables in total. Therefore, the time complexity

of constructing the superstructure is $O(C_n^2 \sum_{i=0}^{k_{par}} C_{n-2}^i T)$ where T is the complexity of the used CI test. For the second part, the time consumed on finding the edge-cut set is mostly spent on calculating the eigenvector, whose time complexity is known as $O(n^3)$ [39]. Therefore, the time complexity of CPA (without causal discovery subroutine) is $O(C_n^2 \sum_{i=0}^{k_{par}} C_{n-2}^i T + n^3)$. For T exceeds $O(n)$ (even simply using Spearman's correlation takes $O(n\log(n))$), then $O(C_n^2 \sum_{i=0}^{k_{par}} C_{n-2}^i T + n^3)$ turns to be $O(C_n^2 \sum_{i=0}^{k_{par}} C_{n-2}^i T)$ and further to be $O(n^{2+k_{par}} T)$. On the other hand, suppose we use the PC algorithm as the subroutine A_g for CPA, the time complexity of PC is up to $O(C_n^2 \sum_{i=0}^{k_{PC}} C_{n-2}^i T)$ and further to be $O(n^{2+k_{PC}} T)$, where k_{PC} denotes the maximum order of CI test used in PC. Generally, $k_{PC} = 2{\sim}4$ can get a good result. Note that, if we do not limit the value of k_{PC}, then the time complexity of PC would be $O(n^2 2^{n-2} T)$ [8]. As shown in Alg. 2, if CPA returns two smaller subsets V_1 and V_2, the time complexity of solving the two subsets V_1 and V_2 is $O(m^{2+k_{PC}} T)$, where $m = max(|V_1|, |V_2|)$. Then the time complexity of CPA+PC is $O((n^{2+k_{par}} + m^{2+k_{PC}})T)$, that is $O(n^{2+k_{par}} T)$ plus $O(A_g)$ regarding m. □

F. Proof of Lemma 6

Lemma 6: Given a variable set $V = \{v_1, ..., v_n\}$ following a causal graph $G_{\mathcal{T}}$, and the causal faithfulness and Markov assumptions, if V_1 and V_2 are the two subsets returned by Alg. 1, i.e. $\{V_1, V_2\} = \text{CPA}(V)$, then $\forall v_i, v_j \in V$ are d-separable in $V \iff \forall v_i, v_j \in V$ are d-separable either in V_1 or V_2, or exist in different subsets.

Proof. We first go back to Def. 1 on causal partitioning, which indicates that any adjacent variables cannot be separated during causal partitioning, and any non-adjacent variables are either separated during causal partitioning or d-separable in at least one subset. That is, if CPA returns causal partitioning, then the conclusion ($\forall v_i, v_j \in V$ are d-separable in $V \iff \forall v_i, v_j \in V$ are d-separable either in V_1 or V_2, or V_1 V_2 exist in different subsets) holds.

Second, according to lemma 1, let G be a superstructure of D, if $\mathbb{P}$ is a PARS (Partitioning of superstructure) over G, then $\mathbb{P}$ is also a causal partitioning over $G_{\mathcal{T}}$. That is, if CPA returns a PARS, then the conclusion still holds.

Third, according to Lemma 2, the edge-cut adjoint graph corresponds to two sets $V_1 = \{A \cup C\}, V_2 = \{B \cup C\}$ in superstructure G satisfying $V = \{A, M, B\}$ i) $\forall v_i \in A$ and $\forall v_j \in B$ are non-adjacent, $M = V \backslash_{A,B}$ and ; ii) Let N be the neighbor set of M, and $C = \{M, N\}$. That is $V = \{V_1, V_2\}$ forms a PARS (please refer to Def. 3). Therefore, the conclusion holds. □

G. Experiment Setting

Experiments are conducted on four causal networks *Alarm*, *Andes*, *Diabetes* and *Link* that can be downloaded at http://www.bnlearn.com/bnrepository/, in which the *Link* (724 nodes) is the second highest dimensional network in this repository, and we do not use the highest dimensional dataset *Munin* (1041 nodes) for the reason that SADA cannot return the result in considerable time. On the other side, because there are not large-scale causal inference problems with ground truth, simulated data on synthetic and real-world structures are used in almost all causal partitioning works, including SADA [27], CAPA [22] and Dsep-CP [24]. The data generation process follows SADA, which is a linear equation model: $x = \sum a_i Pa_i + b_i e$, Pa_i is the ith parent of x, Pa_i and the noise e follow $U(-0.5, 0.5)$, the coefficients a_i and b_i follow $U(0.5, 1)$. More information of its parameters can be found in these related works and our provided code. Each of the experiments except Experiment *C* is conducted on the same planform, therefore it is fair to compare their efficiency. Because most of the counterparts in Experiment *C* use GPU, we run CPA on Intel i9-13900K 3.00 GHz, 128GB RAM, others are on Xeon E5-2678 v3 2.5GHz/GeForce RTX 3090 24GB, 264GB RAM.

H. Details of Non-Partitioning Methods

The implementation details of the non-partitioning baselines are listed below:

- GES. We use the code from the GitHub repository https://github.com/py-why/causal-learn. For reducing runtime, we restrict the maximum number of parents for each node when searching the graph, which is set to 1 in Andes.
- LiNGAM. We use the code from the GitHub repository https://github.com/huawei-noah/trustworthyAI/blob/master/gcastle/castle/algorithms/lingam/ica_lingam.py. The threshold is set to 0.3 as default.
- Direct-LiNGAM. We use the code from the GitHub repository https://github.com/huawei-noah/trustworthyAI/blob/master/gcastle/castle/algorithms/lingam/direct_lingam.py. The threshold is set to 0.3 as default.
- NOTEARS. We use the code from the GitHub repository https://github.com/xunzheng/notears/blob/master/notears/linear.py. The threshold is set to 0.3 as default.
- GOLEM. We use the code available at the GitHub repository https://github.com/ignavierng/golem. We follow the initialization scheme that optimizes GOLEM-NV objective from the solution returned by GOLEM-EV objective.
- DAG-GNN. We use the code available at the GitHub repository https://github.com/fishmoon1234/DAG-GNN.
- GOLEM+CIR. As for CIR, we use the code available at the GitHub repository https://github.com/xyw5vplus1/CIR. The CI coefficient is set to 1.0.
- DAG-GNN+CIR. As for CIR, we use the code available at the GitHub repository https://github.com/xyw5vplus1/CIR. The CI coefficient is set to 1.0.
- GOLEM+CIR. We add a CI regularization term CIR on GOLEM. The code is transferred from the GitHub repository https://github.com/xyw5vplus1/CIR. The CI coefficient is set to 1.0.
- DAG-GNN+CIC/CIR. We separately add a CI hard constraint CIC and a regularization term CIR on DAG-GNN. The code is sourced from the GitHub repository available at https://github.com/xyw5vplus1/CIR. The CI coefficient is fixed at 1.0.
- SCORE. We use the code available at the GitHub repository https://github.com/paulrolland1307/SCORE. CAM

pruning is used after estimating the causal order. Potential causes are selected if the reported p-value is lower or equal to 0.001.
- NoGAM. We utilize the code implemented in the package dodiscover https://github.com/py-why/dodiscover. The pruning process employed is identical to that of SCORE.

I. Ground Truth for Optimal Causal Partitioning

In section IV-B, the ground truth for optimal causal partitioning over the actual skeletons is required for evaluating the performance of partitioning. We know the exhaustive method has a complexity of $O(2^n)$ to searching all permutations for a causal partitioning $\mathbb{P}$ meets min $\frac{|V_1 \cap V_2|}{min(|V_1|,|V_2|)}$, so such a strategy is not feasible. Alternatively we use a "spectral clustering + traversal search" strategy to achieve this goal, although the complexity is still very high. Concretely, 1) the graph spectral clustering is first applied to divide node sets V into two disjoint parts A, B, where A and B are separated by the edge-cut set; 2) note that we can move the cluster center by changing the cut-position corresponding to the elements in the eigenvector v_2 of the second smallest eigenvalue of Laplacian matrix of ground truth graph. Therefore, we sort the elements of v_2, and iteratively change the cut-position, there are $|V|$-2 positions we can choose; 3) for each position, we exhaustively choose C such that C is a vertex-cut set, and C is connected to edge-cut set, then expand C with its one-hop neighbors; 4) we calculate the corresponding cut-ratio, and choose the minimum one as the ground truth although which may not be globally optimal.

J. PC Algorithm v.s. CPA on Andes and Diabetes

The results on *Andes* is presented in Fig. 7 (a)∼(b), PC and CPA become more accurate as the sample size increases. On the other hand, the accuracy of the CI test also increases with the increasing sample size. Fig. 7 (d)∼(e) shows how the F1 score and SHD of CPA and PC change with different maximum size of conditional set. We can see that increasing the maximum size of conditional set does not always increase their accuracy, and their F1 and SHD tend to be stable or even worse when the maximum conditional set $|Z| \geq 3$. But it is shown that the elapsed time continues to rise with increasing $|Z|$ as well as increasing samples, which are shown in Fig.3 (c)&(f). Here, we do not present the results w.r.t. the case of $|Z| \leq 5$ for PC does not finish the work even the runtime is more than 100 hours. Another set of experiment is on *Diabetes* with its results shown in Fig. 8. Note that, maximum conditional set $|Z| \leq 1$ is a special case where PC uses only 1-order CI tests while CPA uses 2-order CI tests for constructing superstructure and also 1-order CI tests for its PC subroutine.

K. Results on the Sensitivity of CPA to Average Degree

The results on the Sensitivity of CPA to Average Degree are presented in Fig. 9. These results show the superior performance of our method.

L. Results on Different Causal Functions and Noise Distributions

The results on three causal functions and five noise distributions are presented in Fig. 10, Fig. 11 and Fig. 12, respectively. These results show the superior performance of our method.

M. Results on Partitioning over Diabetes and Link

The results on partitioning over *Diabetes* and *Link* are presented in Fig. 14 and Fig. 15. It is clear that CPA obtains significantly better cut-set than the other methods. The vertices in the cut-set C of CPA are distributed in the middle of the graph, while the vertices in C of other methods are scattered over the graph due to their different heuristic search strategies. These results are similar to those presented in Fig. 4. Here, we also presented a high-resolution figure of results on *Andes* in Fig. 13.

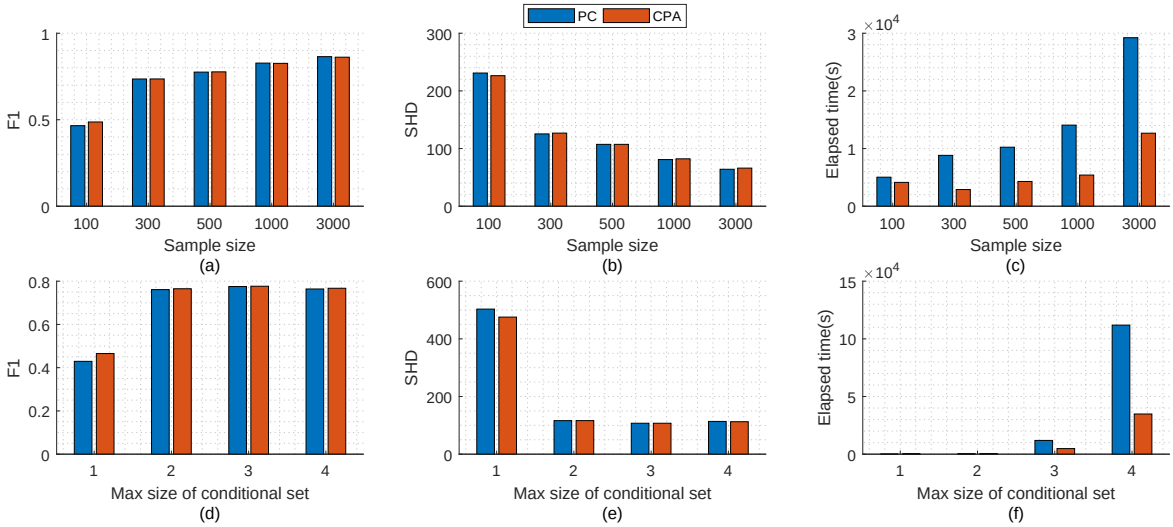

Fig. 7. Causal discovery performance on *Andes*. (a)~(c): PC and CPA with the CI test fixing the maximum size of conditional set $|Z| \leq 3$ under different sample sizes = {100, 300, 500, 1000, 3000}; (d)~(f): their performance under 500 samples when fixing $|Z| \leq \{1, 2, 3, 4\}$, respectively.

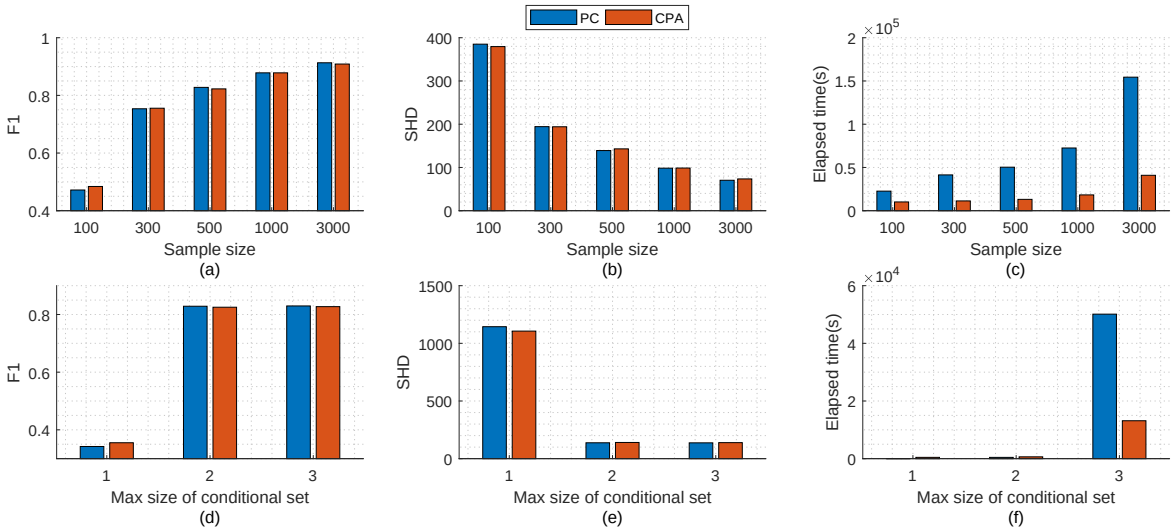

Fig. 8. Causal discovery performance on *Diabetes*. (a)~(c): PC and CPA with the CI test fixing the maximum size of conditional set $|Z| \leq 3$ under different sample sizes = {100, 300, 500, 1000, 3000}; (d)~(f): their performance under 500 samples when fixing $|Z| \leq \{1, 2, 3\}$, respectively.

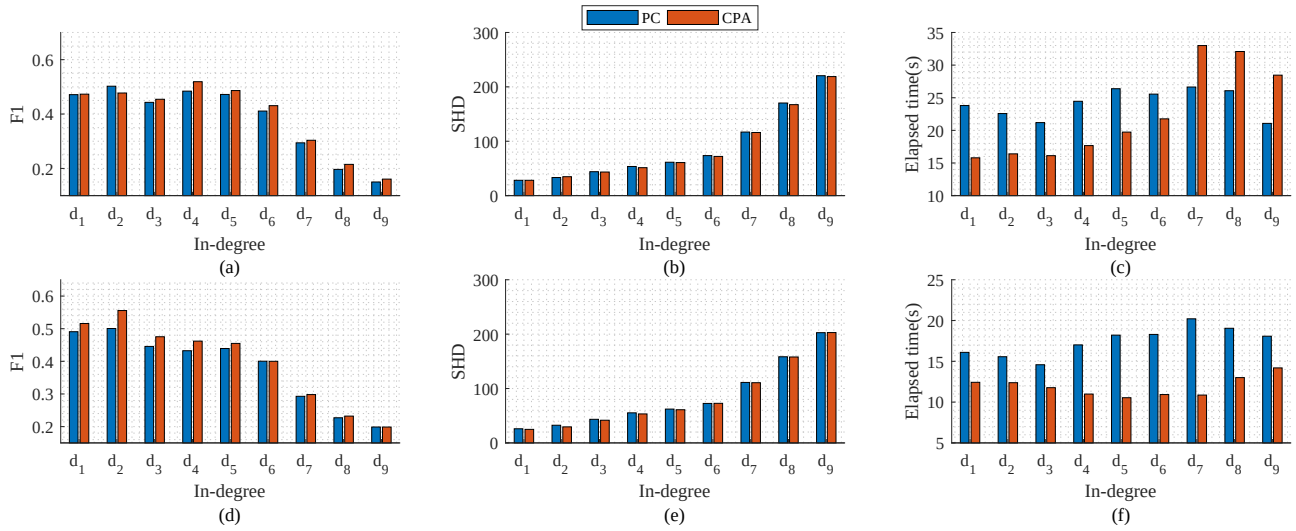

Fig. 9. Performance results of PC and CPA on simulated causal structures with different in-degrees: $d_1 = (1, 1)$, $d_2 = (1, 2)$, $d_3 = (1, 3)$, $d_4 = (2, 1)$, $d_5 = (2, 2)$, $d_6 = (2, 3)$, $d_7 = (3, 3)$, $d_8 = (4, 4)$, $d_9 = (5, 5)$, where $d_i = (a_i, b_i)$ means each node (except for the first 5 nodes) has an 80% probability of having a_i parents and a 20% probability of having b_i parents. There are two ways to choose parents: (a)~(c): the first is to randomly select from the nodes generated before the target node; (d)~(f): the second is to randomly select from the 5 latest nodes generated just before the target node.

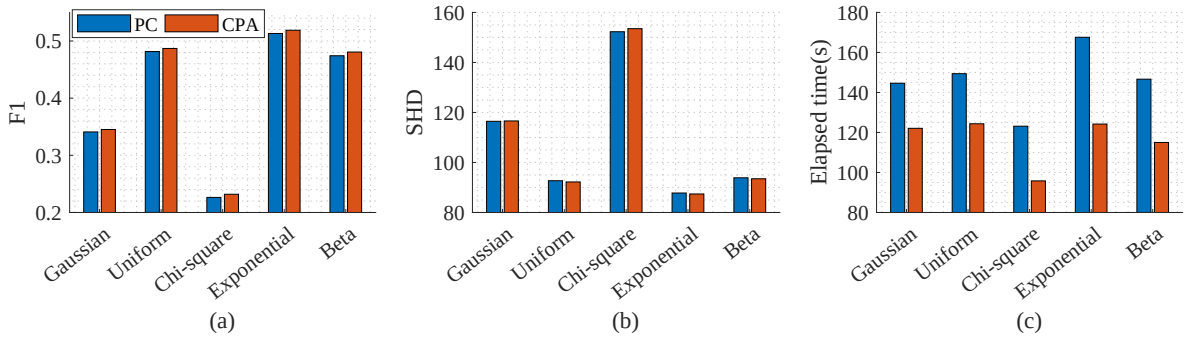

Fig. 10. Performance on simulation of causal function $x = \sum a_i Pa_i + b_i e_j$ (Linear additive noise).

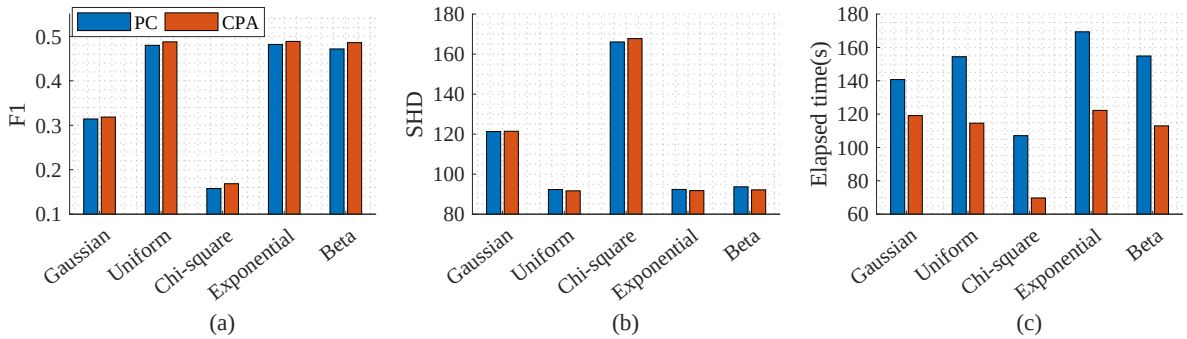

Fig. 11. Performance on simulation of causal function $x = sin(\sum a_i Pa_i) + b_i e_j$ (Nonlinear additive noise).

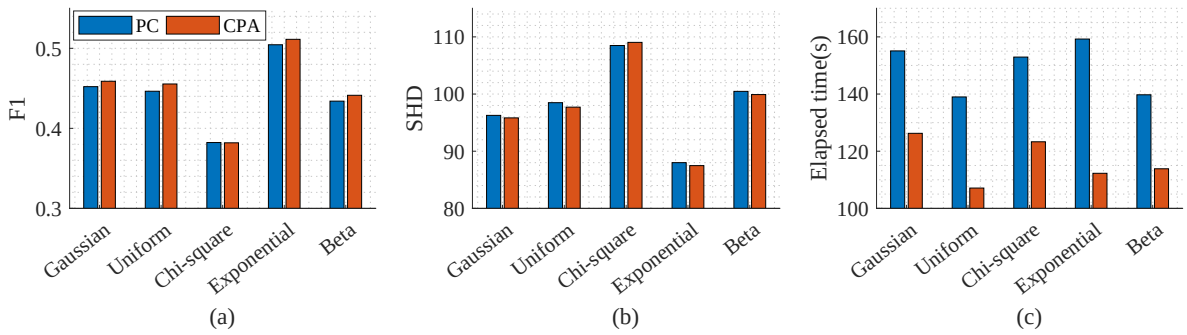

Fig. 12. Performance on simulation of causal function $x = sin(\sum a_i Pa_i + b_i e_j)$ (Post-nonlinear).

Fig. 13. *Andes* partitioned by CPA, SADA, CAPA and Dsep-CP respectively.

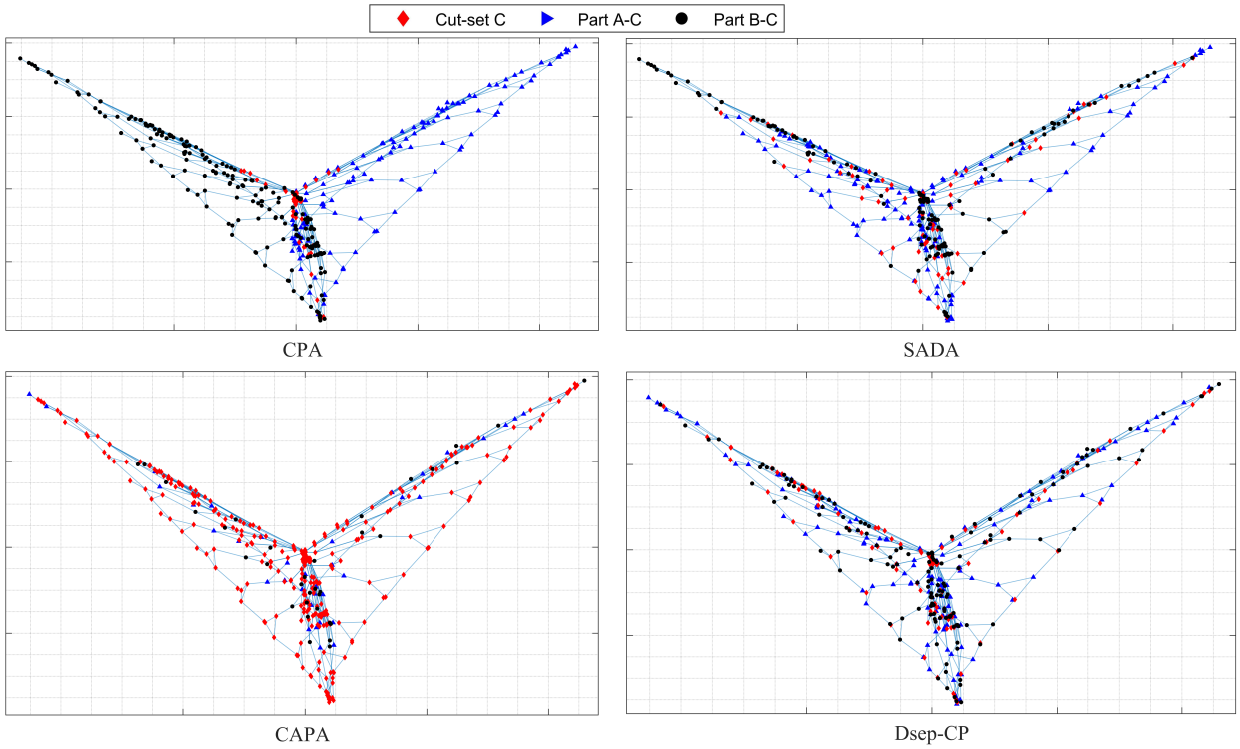

Fig. 14. *Diabetes* partitioned by CPA, SADA, CAPA and Dsep-CP respectively.

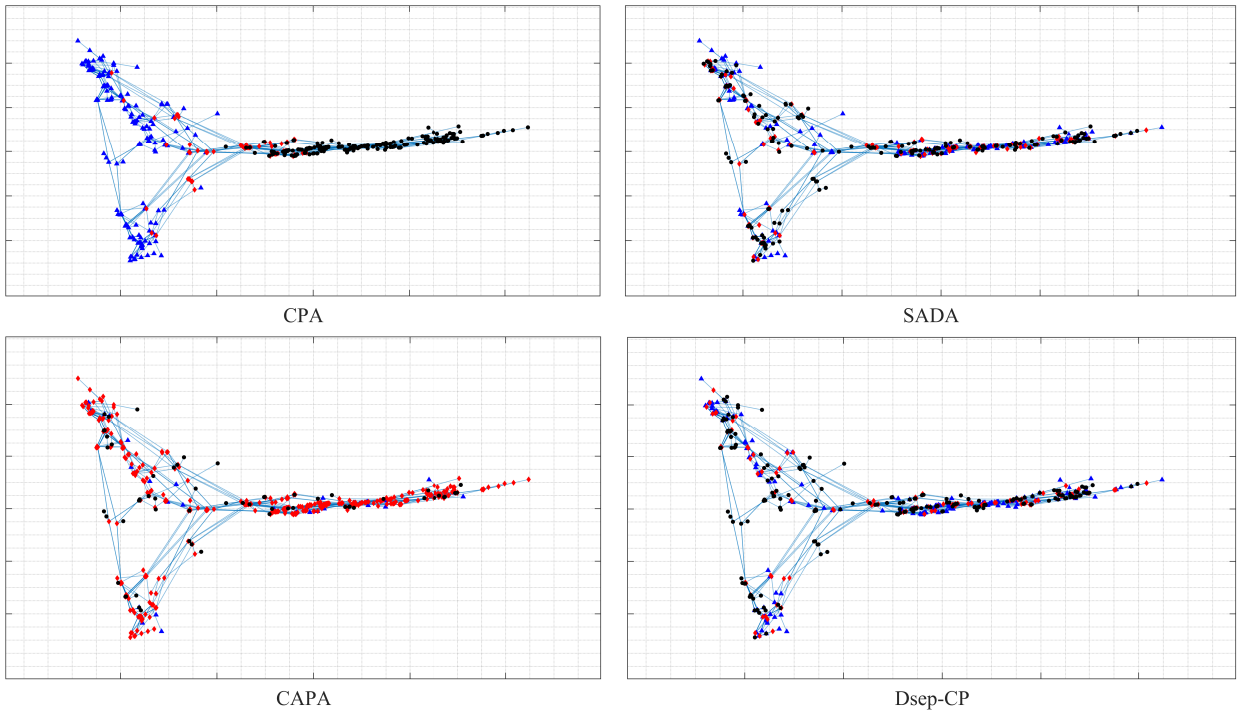

Fig. 15. *Link* partitioned by CPA, SADA, CAPA and Dsep-CP respectively.