Appendices

Appendix organization:

- Appendix A: Proofs for Theorem 1 and Theorem 2.
- Appendix B: Detailed Pseudo-code for `FedACD`.
- Appendix C: Time Complexity of `FedACD`.
- Appendix D: Experimental Results on More Metrics.
- Appendix E: Statistical Tests.
- Appendix F: Implementation Details.

A Proofs for Theorem 1 and Theorem 2

Let $\acute{A}$ represent the adjacency matrix corresponding to the causal skeleton in the ground truth, $r \in [0,1]$ denote the masking rate for each adjacency matrix $A_l^{c_k}$, t^{c_k} represent the number of causal edges removed on the adjacency matrix $A_l^{c_k}$ in Phase 1-1, and $\mathcal{E}_j^{c_k}$ ($j \in \{1,2,...,t^{c_k}\}$) denote the j-th causal edge removed on $A_l^{c_k}$. Further, $\mathcal{E}_j^{c_k}(a)$ and $\mathcal{E}_j^{c_k}(b)$ represent the two variables linked by this edge.

A.1 Proof for Theorem 1

Theorem 1. *If* $P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0) \geq \frac{(1-r)}{2}$, *then* $P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0 | \mathcal{A}_l^{c_k}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) \neq -1) \geq \frac{1}{2}$.

Proof. According to Assumption 2 in the main text, at client c_k, a larger value of $\hat{p}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b))$ implies that $\mathcal{E}_j^{c_k}$ is a causal edge that has been correctly removed from $A_l^{c_k}$. In other words, there is no causal connection between $\mathcal{E}_j^{c_k}(a)$ and $\mathcal{E}_j^{c_k}(b)$ in the ground truth, i.e.,

$$\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0. \tag{1}$$

Thus, $P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0)$ represents the probability of correctly removing a causal edge among all removed causal edges $\{\mathcal{E}_j^{c_k}\}_{j\in\{1,2,...,t^{c_k}\}}$ at client c_k. Based on Eq. (23), when $\{\hat{p}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b))\}_{j\in\{1,2,...,t^{c_k}\}}$ are sorted in descending order, if $\exists\ j_2 \in \{1,2,...,t^{c_k}\}$ such that $\mathcal{E}_{j_2}^{c_k}$ corresponds to $\hat{p}(\mathcal{E}_{j_2}^{c_k}(a), \mathcal{E}_{j_2}^{c_k}(b))$ that ranks among the top $P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0)$ of the sorted $\{\hat{p}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b))\}_{j\in\{1,2,...,t^{c_k}\}}$, then this $\mathcal{E}_{j_2}^{c_k}$ is correctly removed from $A_l^{c_k}$.

According to Eq. (23), Φ^{c_k} stores the indices of the masked $\mathcal{E}_j^{c_k}$ ($j \in \{1,2,...,t^{c_k}\}$) in $\mathcal{A}_l^{c_k}$. Therefore, $\{1,2,...,t^{c_k}\} \setminus \Phi^{c_k}$ represents the indices of the unmasked $\mathcal{E}_j^{c_k}$ ($j \in \{1,2,...,t^{c_k}\}$) in $\mathcal{A}_l^{c_k}$. We have:

$$\underset{k=1,2,...,m}{\{1,2,...,t^{c_k}\} \setminus \Phi^{c_k}} = \overset{\circ}{Top}_{\lfloor(1-r)*t^{c_k}\rfloor}(\underset{j\in\{1,2,...,t^{c_k}\}}{\{\hat{p}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b))\}}), \tag{2}$$

where $\overset{\circ}{Top}_{\lfloor(1-r)*t^{c_k}\rfloor}$ is used to obtain the indices corresponding to the top $\lfloor(1-r)*t^{c_k}\rfloor$ elements in a vector sorted in descending order.

(I) When $P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0) \geq 1-r$, we can obtain:

$$P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0) * t^{c_k} \geq (1-r) * t^{c_k}, \tag{3}$$

$$P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0) * t^{c_k} \geq \lfloor(1-r) * t^{c_k}\rfloor, and \tag{4}$$

$$P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0) * t^{c_k} \geq |\{1,2,...,t^{c_k}\} \setminus \Phi^{c_k}|. \tag{5}$$

Based on Formula (5), all the unmasked $\mathcal{E}_j^{c_k}$ ($j \in \{1,2,...,t^{c_k}\}$) are causal edges that are correctly removed from $A_l^{c_k}$, i.e.,

$$P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0 | \mathcal{A}_l^{c_k}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) \neq -1) = 1. \tag{6}$$

(II) When $\frac{(1-r)}{2} \leq P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0) < 1-r$, we can obtain:

$$\frac{(1-r)*t^{c_k}}{2} \leq P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0) * t^{c_k}, \tag{7}$$

$$\frac{\lfloor(1-r)*t^{c_k}\rfloor}{2} \leq P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0) * t^{c_k}, and \tag{8}$$

$$\frac{|\{1,2,...,t^{c_k}\} \setminus \Phi^{c_k}|}{2} \leq P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0) * t^{c_k}. \tag{9}$$

Based on Formula (9), more than half of the unmasked $\mathcal{E}_j^{c_k}$ ($j \in \{1,2,...,t^{c_k}\}$) are causal edges that are correctly removed from $A_l^{c_k}$, i.e.,

$$P(\acute{A}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) = 0 | \mathcal{A}_l^{c_k}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b)) \neq -1) \geq \frac{1}{2}. \tag{10}$$

Therefore, combining (I) and (II), Theorem 1 holds. □

A.2 Proof for Theorem 2

Theorem 2. *Given* $\forall X_{i_1}, X_{i_2} \in \mathcal{X}$, *in* $\{\mathcal{A}_l^{c_k}\}_{k\in\{1,2,...,m\}}$ *containing* m *matrices, if* $P(\mathcal{A}_l^{c_k}(i_1,i_2) = 0 | \acute{A}(i_1,i_2) = 0) > \frac{1-P(\mathcal{A}_l^{c_k}(i_1,i_2)=-1|\acute{A}(i_1,i_2)=0)}{2}$ *holds, the false causal edge between* X_{i_1} *and* X_{i_2} *can be correctly removed in Phase 1-3; if* $P(\mathcal{A}_l^{c_k}(i_1,i_2) = 1 | \acute{A}(i_1,i_2) = 1) \geq \frac{1-P(\mathcal{A}_l^{c_k}(i_1,i_2)=-1|\acute{A}(i_1,i_2)=1)}{2}$ *holds, the true causal edge between* X_{i_1} *and* X_{i_2} *does not be discarded in Phase 1-3.*

Proof. (I) When $P(\mathcal{A}_l^{c_k}(i_1,i_2) = 0 | \acute{A}(i_1,i_2) = 0) > \frac{1-P(\mathcal{A}_l^{c_k}(i_1,i_2)=-1|\acute{A}(i_1,i_2)=0)}{2}$, we have:

$$\begin{aligned} &m * P(\mathcal{A}_l^{c_k}(i_1,i_2) = 0 | \acute{A}(i_1,i_2) = 0) > \\ &\frac{m - m * P(\mathcal{A}_l^{c_k}(i_1,i_2) = -1 | \acute{A}(i_1,i_2) = 0)}{2}. \end{aligned} \tag{11}$$

Since we have

$$\begin{aligned} &P(\mathcal{A}_l^{c_k}(i_1,i_2) = 0 | \acute{A}(i_1,i_2) = 0) \\ &= 1 - P(\mathcal{A}_l^{c_k}(i_1,i_2) = 1 | \acute{A}(i_1,i_2) = 0) \\ &- P(\mathcal{A}_l^{c_k}(i_1,i_2) = -1 | \acute{A}(i_1,i_2) = 0), \end{aligned} \tag{12}$$

Formula (11) can be transformed into:

$$\begin{aligned} &m - m * P(\mathcal{A}_l^{c_k}(i_1,i_2) = 1 | \acute{A}(i_1,i_2) = 0) \\ &- m * P(\mathcal{A}_l^{c_k}(i_1,i_2) = -1 | \acute{A}(i_1,i_2) = 0) > \\ &\frac{m - m * P(\mathcal{A}_l^{c_k}(i_1,i_2) = -1 | \acute{A}(i_1,i_2) = 0)}{2}. \end{aligned} \tag{13}$$

Simplify Formula (13), then:

$$m - m * P(\mathcal{A}_l^{c_k}(i_1, i_2) = -1 | \acute{A}(i_1, i_2) = 0) > 2m * P(\mathcal{A}_l^{c_k}(i_1, i_2) = 1 | \acute{A}(i_1, i_2) = 0). \quad (14)$$

Or equivalently,

$$m * P(\mathcal{A}_l^{c_k}(i_1, i_2) = 1 | \acute{A}(i_1, i_2) = 0) < \frac{m - m * P(\mathcal{A}_l^{c_k}(i_1, i_2) = -1 | \acute{A}(i_1, i_2) = 0)}{2}. \quad (15)$$

Under the condition of $\acute{A}(i_1, i_2) = 0$, according to Eq. (26), we can obtain:

$$|x| = m * P(\mathcal{A}_l^{c_k}(i_1, i_2) = -1 | \acute{A}(i_1, i_2) = 0). \quad (16)$$

Similarly, under the condition of $\acute{A}(i_1, i_2) = 0$, according to Eq. (27), we can obtain:

$$y = m * P(\mathcal{A}_l^{c_k}(i_1, i_2) = 1 | \acute{A}(i_1, i_2) = 0). \quad (17)$$

Substitute Eqs. (16)-(17) into Formula (15), thus,

$$y < \frac{m - |x|}{2}. \quad (18)$$

Finally, based on Eq. (25), when Formula (18) holds, $\mathcal{A}_l(i_1, i_2) = 0$. Since $\acute{A}(i_1, i_2) = 0$ holds, the false causal edge between X_{i_1} and X_{i_2} can be correctly removed in Phase 1-3.

(II) When $P(\mathcal{A}_l^{c_k}(i_1, i_2) = 1 | \acute{A}(i_1, i_2) = 1) \geq \frac{1 - P(\mathcal{A}_l^{c_k}(i_1, i_2) = -1 | \acute{A}(i_1, i_2) = 1)}{2}$, we have:

$$m * P(\mathcal{A}_l^{c_k}(i_1, i_2) = 1 | \acute{A}(i_1, i_2) = 1) \geq \frac{m - m * P(\mathcal{A}_l^{c_k}(i_1, i_2) = -1 | \acute{A}(i_1, i_2) = 1)}{2}. \quad (19)$$

Under the condition of $\acute{A}(i_1, i_2) = 1$, according to Eq. (26), we can obtain:

$$|x| = m * P(\mathcal{A}_l^{c_k}(i_1, i_2) = -1 | \acute{A}(i_1, i_2) = 1). \quad (20)$$

Similarly, under the condition of $\acute{A}(i_1, i_2) = 1$, according to Eq. (27), we can obtain:

$$y = m * P(\mathcal{A}_l^{c_k}(i_1, i_2) = 1 | \acute{A}(i_1, i_2) = 1). \quad (21)$$

Substitute Eqs. (20)-(21) into Formula (19), thus,

$$y \geq \frac{m - |x|}{2}. \quad (22)$$

Finally, based on Eq. (25), when Formula (22) holds, $\mathcal{A}_l(i_1, i_2) = 1$. Since $\acute{A}(i_1, i_2) = 1$ holds, the true causal edge between X_{i_1} and X_{i_2} does not be discarded in Phase 1-3.

Thus, combining (I) and (II), Theorem 2 holds. □

B Detailed Pseudo-code for FedACD

Our proposed FedACD consists of two phases: *Federated Causal Skeleton Learning* (FCSL) and *Federated Skeleton Orientation* (FSO), and their pseudo-codes are described in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 Federated Causal Skeleton Learning (FCSL)

Require: $\mathcal{D}^{\mathcal{C}} = \{\mathcal{D}^{c_1}, \mathcal{D}^{c_2}, ..., \mathcal{D}^{c_m}\}$: m local datasets held by m clients, and each dataset has the same variable space $\mathcal{X} = \{X_1, X_2, ..., X_d\}$; r: the masking rate
Ensure: $\mathcal{A}^*$: the final causal skeleton

1: Initialize $l = 0$. *// the size of the conditioning set*
2: **for** k=1 to m; i_1, i_2=1 to d **do**
3: **if** $i_1 \neq i_2$ **then**
4: $A_l^{c_k}(i_1, i_2) = 1$ *// a completely undirected graph*
5: **else**
6: $A_l^{c_k}(i_1, i_2) = 0$
7: **end if**
8: **end for**
9: **repeat**
10: **for** k=1 to m; i_1, i_2=1 to d; $A_l^{c_k}(i_1, i_2) \neq 0$ **do**
11: **if** $\exists \mathbf{Z} \subseteq CN_{i_1}^{c_k}$ with $|\mathbf{Z}| = l$ s.t. $X_{i_1} \perp\!\!\!\perp X_{i_2} | \mathbf{Z}$ **then**
12: $A_l^{c_k}(i_1, i_2) = A_l^{c_k}(i_2, i_1) = 0$ *// remove edges*
13: $\mathcal{E}_j^{c_k}(a) = i_1$ and $\mathcal{E}_j^{c_k}(b) = i_2$ ($j \in \{1, 2, ..., t^{c_k}\}$)
14: **end if**
15: **end for**
16: Record the maximum p-value $\hat{p}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b))$ returned by all CI tests that render $\mathcal{E}_j^{c_k}(a)$ and $\mathcal{E}_j^{c_k}(b)$ conditionally independent on local dataset $\mathcal{D}^{c_k}$.
17: **for** k=1 to m **do**
18: $\Phi^{c_k} = \overset{\circ}{Bottom}_{\lceil r * t^{c_k} \rceil}(\{\hat{p}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b))\}_{j \in \{1,2,...,t^{c_k}\}})$
19: **for** i_1=1 to d; i_2=1 to d **do**
20: **if** $\exists \psi \in \Phi^{c_k}$ s.t. $\mathcal{E}_\psi^{c_k}(a) = i_1 \wedge \mathcal{E}_\psi^{c_k}(b) = i_2$ or $\mathcal{E}_\psi^{c_k}(a) = i_2 \wedge \mathcal{E}_\psi^{c_k}(b) = i_1$ **then**
21: $\mathcal{A}_l^{c_k}(i_1, i_2) = -1$
22: **else**
23: $\mathcal{A}_l^{c_k}(i_1, i_2) = A_l^{c_k}(i_1, i_2)$
24: **end if**
25: **end for**
26: **end for**
27: **for** i_1=1 to d; i_2=1 to d **do**
28: $x = \sum_{k=1}^{m} \mathcal{A}_l^{c_k}(i_1, i_2)\ subject\ to\ \mathcal{A}_l^{c_k}(i_1, i_2) = -1$
29: $y = \sum_{k=1}^{m} \mathcal{A}_l^{c_k}(i_1, i_2)\ subject\ to\ \mathcal{A}_l^{c_k}(i_1, i_2) = 1$
30: **if** $y \geq \frac{m - |x|}{2}$ **then**
31: $\mathcal{A}_l(i_1, i_2) = 1$
32: **else**
33: $\mathcal{A}_l(i_1, i_2) = 0$
34: **end if**
35: **end for**
36: $l = l + 1$ *// increase the size of the conditioning set*
37: **until** $\max_{i_1=1}^{d} \left(\sum_{i_2=1}^{d} \mathcal{A}_{l-1}(i_1, i_2) \right) < l$
38: $\mathcal{A}^* = \mathcal{A}_{l-1}$
39: **return** $\mathcal{A}^*$

B.1 Detailed Pseudo-code for FCSL

Initially, the size of the conditioning set for conditional independence (CI) tests is set to 0 (Line 1). At Lines 2-8, FCSL first constructs a complete undirected graph over $\mathcal{X}$ at each client c_k, and then we obtain m causal skeletons and their corresponding adjacency matrices $\{A_l^{c_k}\}_{k \in \{1,2,...,m\}}$.

At Lines 10-15, for any two variables X_{i_1} and X_{i_2} ($X_{i_1}, X_{i_2} \in \mathcal{X}$), FCSL utilizes CI tests to determine whether they are conditionally independent given the conditioning set $\boldsymbol{Z}$ ($\boldsymbol{Z} \subseteq CN_{i_1}^{c_k}$ or $\boldsymbol{Z} \subseteq CN_{i_2}^{c_k}$) with $|\boldsymbol{Z}| = l$. If $X_{i_1} \perp\!\!\!\perp X_{i_2} | \boldsymbol{Z}$ holds, the causal edge between X_{i_1} and X_{i_2} is removed (Line 12). Let t^{c_k} represent the number of causal edges removed on the adjacency matrix $A_l^{c_k}$ during Lines 10-15, and $\mathcal{E}_j^{c_k}$ ($j \in \{1, 2, ..., t^{c_k}\}$) denote the j-th causal edge removed on $A_l^{c_k}$. Further, $\mathcal{E}_j^{c_k}(a)$ and $\mathcal{E}_j^{c_k}(b)$ represent the two variables linked by this edge. After performing the above operations at each client, $A_l^{c_k}(i_1, i_2) = 1$ indicates that on the local dataset $\mathcal{D}^{c_k}$, X_{i_1} and X_{i_2} are conditionally dependent given any $\boldsymbol{Z}$ ($\boldsymbol{Z} \subseteq CN_{i_1}^{c_k}$ or $\boldsymbol{Z} \subseteq CN_{i_2}^{c_k}$) with $|\boldsymbol{Z}| = l$. On the contrary, $A_l^{c_k}(i_1, i_2) = 0$ means that there exists a $\boldsymbol{Z}$ ($\boldsymbol{Z} \subseteq CN_{i_1}^{c_k}$ or $\boldsymbol{Z} \subseteq CN_{i_2}^{c_k}$) with $|\boldsymbol{Z}| = l$ that makes X_{i_1} and X_{i_2} conditionally independent. Since the causal skeleton is an undirected graph, $A_l^{c_k}(i_1, i_2) = A_l^{c_k}(i_2, i_1)$ holds for $\forall i_1, i_2, k, l$.

In Line 11, there might be multiple conditioning sets that can make two variables $\mathcal{E}_j^{c_k}(a)$ and $\mathcal{E}_j^{c_k}(b)$ conditionally independent. However, the CI test with a larger p-value is considered more reliable [Ramsey, 2016]. Therefore, in Line 16, FCSL only records the test results with the maximum p-value among all CI tests that indicate the conditional independence of two variables, and denotes the maximum p-value as $\hat{p}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b))$. According to Assumption 2 in the main text, the smaller $\hat{p}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b))$, the more likely outcome is that $\mathcal{E}_j^{c_k}$ is a mistakenly deleted causal edge. Conversely, a larger value of $\hat{p}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b))$ implies that there is no causal connection between $\mathcal{E}_j^{c_k}(a)$ and $\mathcal{E}_j^{c_k}(b)$ in the ground truth. Thus, in Lines 17-26, FCSL masks the causal edges that have been removed from $A_l^{c_k}$ with low $\hat{p}(\cdot, \cdot)$. In Line 18, we use a one-dimensional vector $\boldsymbol{\Phi}^{c_k}$ to store the indices of the edges that need to be masked in $A_l^{c_k}$, and then we have:

$$\underset{k=1,2,...,m}{\boldsymbol{\Phi}^{c_k}} = \mathring{Bottom}_{\lceil r * t^{c_k} \rceil}(\{\hat{p}(\mathcal{E}_j^{c_k}(a), \mathcal{E}_j^{c_k}(b))\}), \quad (23)$$
$$j \in \{1,2,...,t^{c_k}\}$$

where $\mathring{Bottom}_{\lceil r * t^{c_k} \rceil}$ is used to obtain the indices corresponding to the bottom $\lceil r * t^{c_k} \rceil$ elements in a vector sorted in descending order. We use $\mathcal{A}_l^{c_k}$ to denote the adjacency matrix after masking $A_l^{c_k}$:

$$\underset{\substack{k=1,2,...,m \\ i_1,i_2=1,2,...,d}}{\mathcal{A}_l^{c_k}(i_1, i_2)} = \begin{cases} \text{-1} & \begin{array}{l} if\ \exists \psi \in \boldsymbol{\Phi}^{c_k}\ such\ that \\ \mathcal{E}_\psi^{c_k}(a) = i_1 \wedge \mathcal{E}_\psi^{c_k}(b) = i_2 \\ or\ \mathcal{E}_\psi^{c_k}(a) = i_2 \wedge \mathcal{E}_\psi^{c_k}(b) = i_1 \end{array} \\ A_l^{c_k}(i_1, i_2) & otherwise. \end{cases} \quad (24)$$

After obtaining all the masked matrices $\{\mathcal{A}_l^{c_k}\}_{k \in \{1,2,...,m\}}$, the FL server aggregates and updates these matrices at Lines 27-35. Since a value of "-1" in a cell of the masked matrices indicates that the corresponding learning result is unreliable, such masked causal relationships are invalid during aggregation. In other words, each element in $\mathcal{A}_l^{c_k}$ only participates in voting if its value is not "-1". Let the aggregated adjacency matrix be $\mathcal{A}_l$, at Lines 30-34, we have:

$$\underset{i_1,i_2=1,2,...,d}{\mathcal{A}_l(i_1, i_2)} = \begin{cases} 1 & if\ y \geq \frac{m-|x|}{2} \\ 0 & otherwise. \end{cases}, \quad (25)$$

where x represents the number of adjacency matrices in $\{\mathcal{A}_l^{c_k}\}_{k \in \{1,2,...,m\}}$ where the causal relationship between X_{i_1} and X_{i_2} is masked (Line 28), i.e.,

$$x = \sum_{k=1}^{m} \mathcal{A}_l^{c_k}(i_1, i_2)\ subject\ to\ \mathcal{A}_l^{c_k}(i_1, i_2) = -1, \quad (26)$$

and y denotes the number of adjacency matrices in $\{\mathcal{A}_l^{c_k}\}_{k \in \{1,2,...,m\}}$ where there exists a causal edge between X_{i_1} and X_{i_2} (Line 29), i.e.,

$$y = \sum_{k=1}^{m} \mathcal{A}_l^{c_k}(i_1, i_2)\ subject\ to\ \mathcal{A}_l^{c_k}(i_1, i_2) = 1. \quad (27)$$

After obtaining the newly aggregated causal skeleton $\mathcal{A}_l$, l is incremented by 1 (Line 36). Then, at Line 37, the FL server determines whether $\mathcal{A}_{l-1}$ (i.e., the newly aggregated causal skeleton) has converged. Specifically, the FL server checks whether there are variables in $\mathcal{A}_{l-1}$ whose number of causal neighbors is greater than or equal to l. If such variables exist, it means that new CI tests, where the size of the conditioning set is equal to l, can be performed to remove possible erroneous causal edges further; otherwise, it means that $\mathcal{A}_{l-1}$ has converged and no more CI test is needed. The convergence condition is formalized as:

$$\max_{i_1=1}^{d} \left(\sum_{i_2=1}^{d} \mathcal{A}_{l-1}(i_1, i_2) \right) < l. \quad (28)$$

If Eq. (28) does not hold, the FL server sends $\mathcal{A}_{l-1}$ as the new initial causal skeleton to each client to repeat Lines 9-37; otherwise, the optimal causal skeleton $\mathcal{A}_{l-1}$ is returned.

B.2 Detailed Pseudo-code for FSO

After obtaining the final causal skeleton $\mathcal{A}^*$, FSO orients the undirected edges in $\mathcal{A}^*$ at each client for learning m causal

Algorithm 2 Federated Skeleton Orientation (FSO)

Require: $\mathcal{D}^{\mathcal{C}} = \{\mathcal{D}^{c_1}, \mathcal{D}^{c_2}, ..., \mathcal{D}^{c_m}\}$: m local datasets held by m clients; $\mathcal{A}^*$: the final causal skeleton
Ensure: $\mathcal{G}^*$: the final causal DAG
1: **for** k=1 to m **do**
2: **if** $\mathcal{D}^{c_k}$ is a discrete dataset **then**
3: $\mathcal{G}^{c_k} \xleftarrow[\mathcal{D}^{c_k}]{hill-climbing\ search\ +\ BDeu\ score} \mathcal{A}^*$
4: **else**
5: $\mathcal{G}^{c_k} \xleftarrow[\mathcal{D}^{c_k}]{hill-climbing\ search\ +\ BIC\ score} \mathcal{A}^*$
6: **end if**
7: **end for**
8: $\mathcal{G}^* = \mathcal{G}^{c_1} \oplus \mathcal{G}^{c_2} \oplus \cdots \oplus \mathcal{G}^{c_m}$
9: **for** i_1=1 to d; i_2=1 to $(i_1 - 1)$ **do**
10: **if** $\mathcal{G}^*(i_1, i_2) > \mathcal{G}^*(i_2, i_1)$ **then**
11: $\mathcal{G}^*(i_1, i_2) = 1, \mathcal{G}^*(i_2, i_1) = 0$
12: **else if** $\mathcal{G}^*(i_1, i_2) = \mathcal{G}^*(i_2, i_1) = 0$ **then**
13: $\mathcal{G}^*(i_1, i_2) = \mathcal{G}^*(i_2, i_1) = 0$
14: **else**
15: $\mathcal{G}^*(i_1, i_2) = 0, \mathcal{G}^*(i_2, i_1) = 1$
16: **end if**
17: **end for**
18: **return** $\mathcal{G}^*$

DAGs (Lines 1-7) and then aggregates all causal DAGs at the FL server to produce the final causal DAG $\mathcal{G}^*$ (Lines 8-17).

Specifically, the FL server first sends $\mathcal{A}^*$ to each client, then the score-and-search strategy is adopted to greedily orient the undirected edges in $\mathcal{A}^*$ for obtaining a causal DAG with the highest score at each client. Let $\mathcal{G}^{c_k}$ denote the causal DAG learned at client c_k, and "$\mathcal{G}^{c_k}(i_1, i_2) = 1$" denotes that there is an edge from X_{i_1} to X_{i_2} in $\mathcal{G}^{c_k}$. For discrete datasets, at Line 3, FSO utilizes a Bayesian score, BDeu [Scutari, 2016], and a search procedure, hill-climbing [Gámez *et al.*, 2011], to implement the above score-and-search process. Here, the BDeu score for the causal DAG $\mathcal{G}^{c_k}$ learned on client dataset $\mathcal{D}^{c_k}$ is defined as:

$$\mathrm{BDeu}(\mathcal{G}^{c_k}, \mathcal{D}^{c_k}) = \ln P(\mathcal{D}^{c_k}|\beta, \mathcal{G}^{c_k}) + \ln P(\beta, \mathcal{G}^{c_k}), \quad (29)$$

where $P(\mathcal{D}^{c_k}|\beta, \mathcal{G}^{c_k})$ denotes the probability of the local dataset $\mathcal{D}^{c_k}$ given the equivalent sample size parameter β and the causal DAG $\mathcal{G}^{c_k}$. It measures how well the causal DAG predicts the observed discrete data. $P(\beta, \mathcal{G}^{c_k})$ represents the prior probability of β. It serves as a regularization term and influences the strength of prior beliefs about the density of the causal DAG.

For continuous datasets, at Line 5, FSO uses an information-theoretic score, BIC [Watanabe, 2013], to calculate the fitting score between $\mathcal{G}^{c_k}$ and $\mathcal{D}^{c_k}$. The BIC score for $\mathcal{G}^{c_k}$ learned on $\mathcal{D}^{c_k}$ is defined as:

$$\mathrm{BIC}(\mathcal{G}^{c_k}, \mathcal{D}^{c_k}) = -2 \cdot \ln(\hat{L}(\mathcal{G}^{c_k}, \mathcal{D}^{c_k})) + \mu \cdot \ln(n_{c_k}), \quad (30)$$

where $\hat{L}(\mathcal{G}^{c_k}, \mathcal{D}^{c_k})$ denotes the ability of the causal DAG to explain the observed continuous dataset $\mathcal{D}^{c_k}$. A higher likelihood indicates a better fit. μ represents the average density of $\mathcal{G}^{c_k}$. The penalty term "$\mu \cdot \ln(n_{c_k})$" discourages overly dense causal DAGs, favoring sparser causal DAGs to reduce the risk of overfitting.

Subsequently, at Line 8, FSO sends all causal DAGs $\{\mathcal{G}^{c_1}, \mathcal{G}^{c_2}, \cdots, \mathcal{G}^{c_m}\}$ back to the FL server for computing the aggregated causal DAG $\mathcal{G}^*$ as:

$$\mathcal{G}^* = \mathcal{G}^{c_1} \oplus \mathcal{G}^{c_2} \oplus \cdots \oplus \mathcal{G}^{c_m}, \quad (31)$$

where $\oplus$ represents the element-wise addition of matrices. Finally, in Lines 9-17, FSO compares the elements at corresponding positions on the diagonal of matrix $\mathcal{G}^*$ for obtaining the final causal DAG. Specifically, if $\mathcal{G}^*(i_1, i_2) > \mathcal{G}^*(i_2, i_1)$, then there exists a directed edge from X_{i_1} to X_{i_2}. If $\mathcal{G}^*(i_1, i_2) \leq \mathcal{G}^*(i_2, i_1)$ and $\mathcal{G}^*(i_2, i_1) \neq 0$, there exists a directed edge from X_{i_2} to X_{i_1}; Otherwise, there is no edge between X_{i_1} and X_{i_2}. To summarize, we have:

$$\begin{cases} \mathcal{G}^*(i_1, i_2) = 1 \wedge \mathcal{G}^*(i_2, i_1) = 0 & if\ \mathcal{G}^*(i_1, i_2) > \mathcal{G}^*(i_2, i_1) \\ \mathcal{G}^*(i_1, i_2) = \mathcal{G}^*(i_2, i_1) = 0 & if\ \mathcal{G}^*(i_1, i_2) = \mathcal{G}^*(i_2, i_1) = 0 \\ \mathcal{G}^*(i_1, i_2) = 0 \wedge \mathcal{G}^*(i_2, i_1) = 1 & \text{otherwise,} \end{cases} \quad (32)$$

where $i_1 = 1, 2, \cdots, d$ and $i_2 = 1, 2, \cdots, (i_1 - 1)$.

C Time Complexity of FedACD

In Phase 2, at each client, FedACD performs the score-and-search strategy on the given adjacency matrix $\mathcal{A}^*$ corresponding to the final causal skeleton rather than on an empty graph. It means that during the search process, FedACD does not need to perform adding causal edges and removing causal edges operations, but only needs to perform reversing causal edges operations to achieve the highest score, that is, the entire search space is very small.

Therefore, the time complexity of FedACD mainly lies in Phase 1, and the computational cost of this phase is measured via the number of CI (conditional independence) tests. Let p denote the maximum size of any conditioning set for CI tests. In Phase 1-4, according to Formula (28), when $l = p$, the whole Phase 1 converges. Thus, for Phase 1-1, on a single dataset (a local dataset held by a client), the total time complexity of all iterations in the causal skeleton learning process is denoted as follows.

$$\frac{d^2(d-1)^{p-1}}{(p-1)!}. \quad (33)$$

Consequently, the time complexity of Phase 1-1 at all m local datasets is

$$\frac{md^2(d-1)^{p-1}}{(p-1)!}. \quad (34)$$

In the subsequent Phase 1-2, Phase 1-3 and Phase 1-4, the FedACD algorithm does not conduct any additional CI tests. Overall, the computational complexity of FedACD is $(md^2(d-1)^{p-1})/(p-1)!$ CI tests.

D Experimental Results on More Metrics

D.1 More Evaluation Metrics

Let TP be the number of true positives (edges in both the true causal DAG and learned causal DAG); FP the number of false positives (edges in the learned causal DAG but not in the true causal DAG; TN the number of true negatives (edges not in either the true or learned causal DAG); and FN the number of false negatives (edges in the true causal DAG but missing from the learned causal DAG). To further evaluate the performance of FedACD in comparison to its rivals, we employ five new metrics, False Discovery Rate (FDR), True Positive Rate (TPR), Reverse, Miss and Extra, as follows.

- *False Discovery Rate (FDR)*. FDR is the ratio of false edges in the learned causal DAG to the edges in the learned causal DAG. That is, $FDR = \frac{FP}{TP+FP}$.
- *True Positive Rate (TPR)*. TPR is the ratio of correct edges in the learned causal DAG to total edges in the true causal DAG. That is, $TPR = \frac{TP}{TP+FN}$.
- *Reverse*. The number of edges with wrong directions according to the true causal DAG.
- *Miss*. The number of missing edges in the causal DAG learned by the algorithm against the true causal DAG.
- *Extra*. The number of extra edges in the learned causal DAG.

In all figures, (↑) means the higher the better, and (↓) means the lower the better.

D.2 Experimental Results on More Metrics for Benchmark Bayesian Network Dataset

In this section, we report the experimental results of FedACD and its baselines on benchmark BN datasets in terms of FDR, TPR, Reverse, Miss and Extra metrics.

From Fig. 1, we observe that in most cases, our method achieves higher TPR (True Positive Rate) and lower FDR (False Discovery Rate), Reverse, Miss, and Extra values compared to the baseline algorithms, which further validates the superiority of our method. In comparison to the best baseline FedPC, as the number of clients increases, FedPC exhibits a significant decline in performance, whereas our method remains remarkably stable. The Miss values achieved by GS-FedDAG and AS-FedDAG are generally higher than those achieved by other algorithms since the causal DAGs they learn are too sparse.

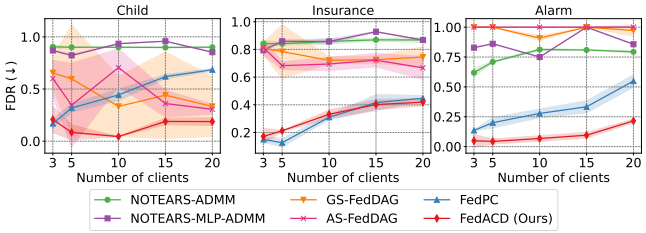
D.3 Experimental Results on More Metrics for Synthetic Non-IID Datasets

In this section, we present the experimental results of FedACD and its baselines on synthetic Non-IID datasets in terms of FDR, TPR, Reverse, Miss and Extra metrics. As shown in Fig. 2, our method significantly outperforms all other baseline algorithms in terms of FDR and TPR metrics. Regarding the Miss metric, our method achieves a comparable performance against FedPC, and significantly outperforms the other baselines regardless of the number of clients. AS-FedDAG is specifically designed to handle Non-IID data, hence achieving good performance in this set of experiments. However, due to the regularization term in its employed loss function, many true causal edges might be discarded, limiting its performance. In contrast to AS-FedDAG, NOTEARS-ADMM tends to learn causal DAGs that are generally dense, leading to high Extra values achieved by this algorithm. The orientation rules designed by FedPC might not be well-suited for the federated causal discovery scenario, resulting in significantly higher Reverse values compared to other algorithms.

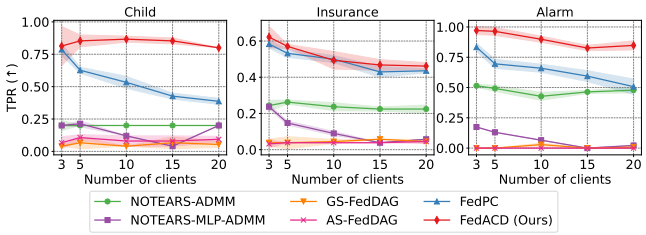
These experimental results further demonstrate the substantial superiority of our method, not only on benchmark Bayesian network datasets but also on synthetic Non-IID datasets, highlighting the outstanding performance of the FedACD algorithm in federated causal discovery tasks under privacy-preserving scenarios.

D.4 Experimental Results on More Metrics for Real-World Datasets

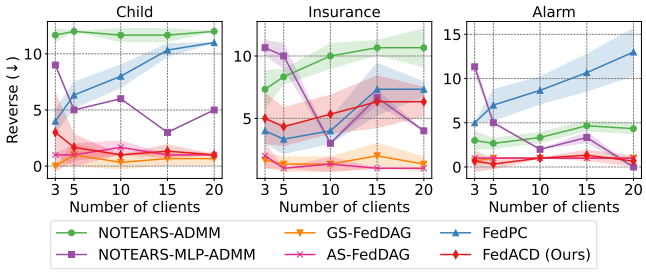
In this section, we present the experimental results of FedACD and its baselines on real-world datasets in terms of FDR, TPR, Reverse, Miss and Extra metrics. The experimental results are shown in Fig. 3, and we can see that on the REGED dataset with 20 nodes, our method outperforms all other baseline algorithms significantly in terms of FDR metric. On the REGED dataset with 8 nodes, the low Miss values achieved by our method indicate that the FedACD algorithm can identify nearly all true causal edges. AS-FedDAG has achieved promising performance on real-world datasets, indicating that the REGED dataset might follow a Non-IID, as AS-FedDAG is specifically designed for Non-IID data.



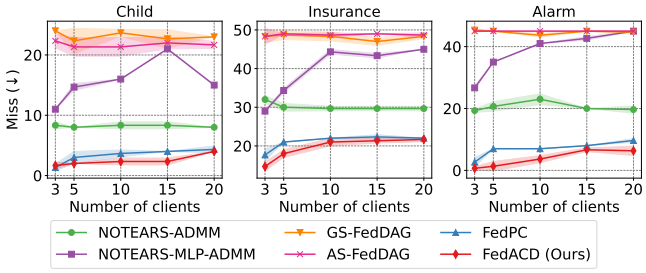
(a) FDR metric



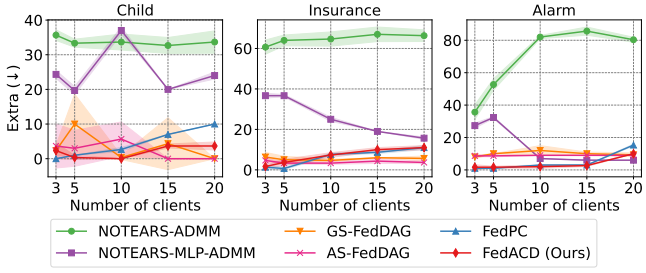
(b) TPR metric



(c) Reverse metric

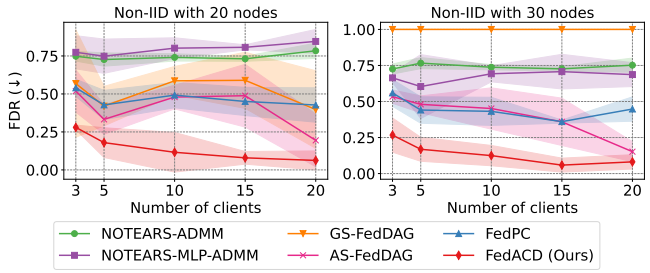


(d) Miss metric

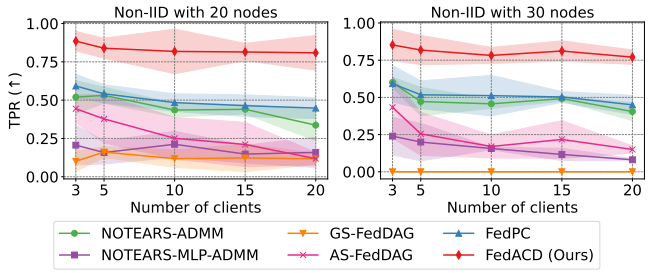


(e) Extra metric

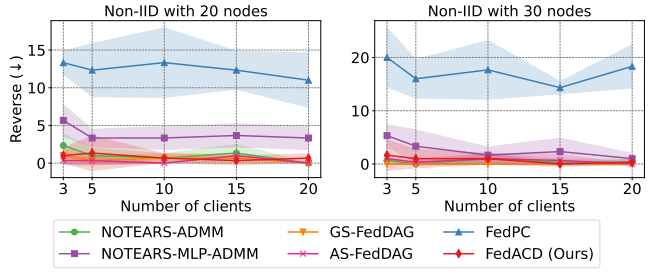
Figure 1: Experimental results on benchmark BN datasets. There are 5,000 samples in total, distributed evenly across {3, 5, 10, 15, 20} clients. We show the performance of all methods in five metrics (FDR, TPR, Reverse, Miss and Extra from top to bottom).



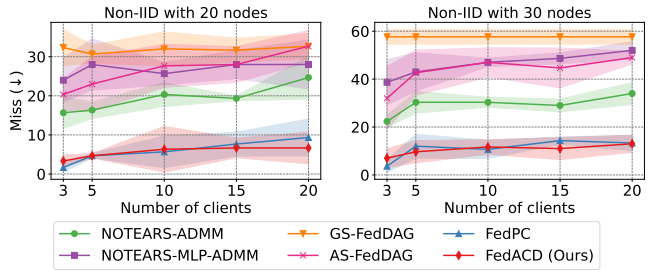
(a) FDR metric



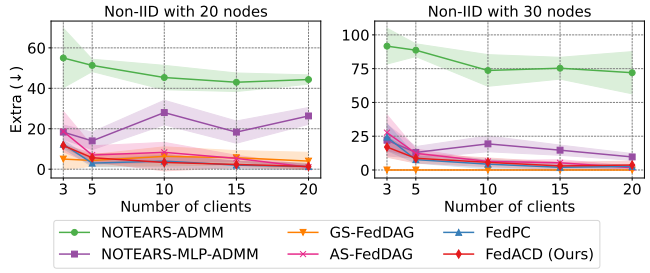
(b) TPR metric



(c) Reverse metric

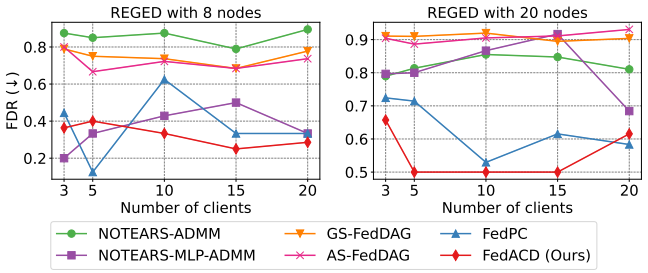


(d) Miss metric

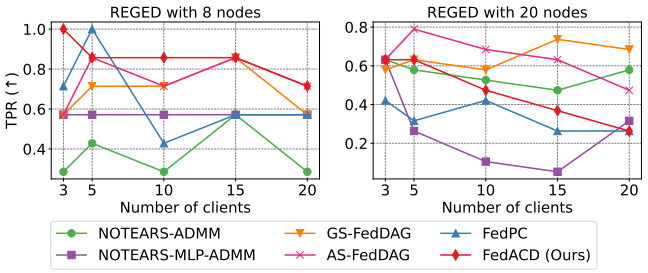


(e) Extra metric

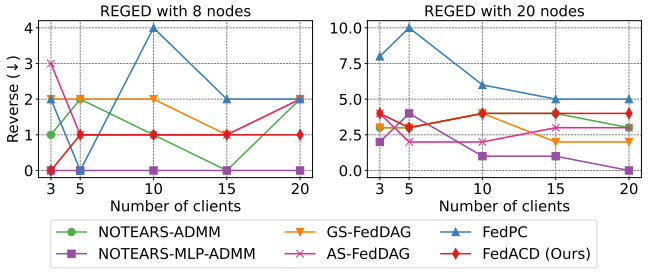
Figure 2: Experimental results on synthetic Non-IID datasets. There are 5,000 samples in total, distributed evenly across {3, 5, 10, 15, 20} clients. We show the performance of all methods in five metrics (FDR, TPR, Reverse, Miss and Extra from top to bottom).



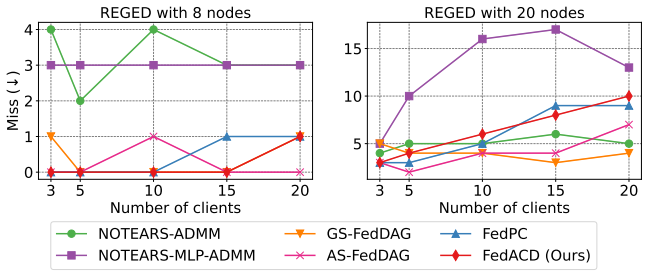
(a) FDR metric



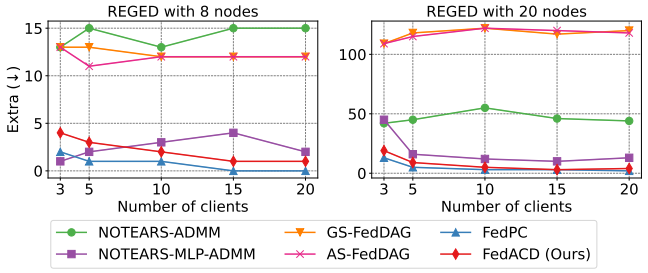
(b) TPR metric



(c) Reverse metric



(d) Miss metric



(e) Extra metric

Figure 3: Experimental results on real-world datasets. There are 1,000 samples in total, distributed evenly across {3, 5, 10, 15, 20} clients. We show the performance of all methods in five metrics (FDR, TPR, Reverse, Miss and Extra from top to bottom).

E Statistical Tests

In this section, we adopt the Friedman test and Nemenyi test [Demšar, 2006] to verify whether FedACD is significantly better than other methods.

We first perform the Friedman test [Demšar, 2006] at the 0.05 significance level under the null hypothesis which states that the performance of all algorithms is the same on all datasets (i.e., the average rankings of all algorithms are equivalent). For benchmark BN datasets, the average rankings of FedACD and the baselines when using different metrics are summarized in Table 1; For synthetic Non-IID datasets, the average rankings of FedACD and the baselines when using different metrics are summarized in Table 2; For real-world datasets, the average rankings of FedACD and the baselines when using different metrics are summarized in Table 3. Tables 1-3 show that regardless of the type of dataset, the null hypothesis is rejected on these two metrics (i.e. SHD and F1 score). We also note that no matter which metric, FedACD always performs better than the baselines on all datasets. (In Tables 1-3, the lower ranking value is better.)

Table 1: The average rankings of FedACD and the baselines on benchmark BN datasets using SHD and F1 metrics.

Algorithm	Avg rank	
	SHD	F1
NOTEARS-ADMM	5.87	3.47
NOTEARS-MLP-ADMM	4.73	4.13
GS-FedDAG	3.9	5.4
AS-FedDAG	3.37	5
FedPC	1.97	1.87
FedACD (Ours)	**1.17**	**1.13**

Table 2: The average rankings of FedACD and the baselines on synthetic Non-IID datasets using SHD and F1 metrics.

Algorithm	Avg rank	
	SHD	F1
NOTEARS-ADMM	6	3.5
NOTEARS-MLP-ADMM	5	5.3
GS-FedDAG	3.8	5.7
AS-FedDAG	3.2	3.5
FedPC	2	2
FedACD (Ours)	**1**	**1**

Table 3: The average rankings of FedACD and the baselines on real-world datasets using SHD and F1 metrics.

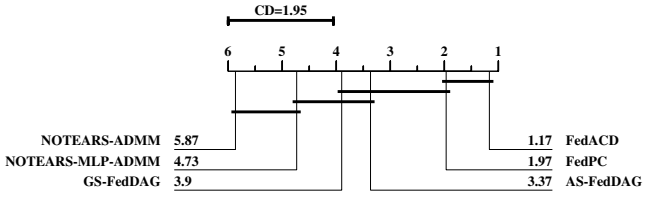
Algorithm	Avg rank	
	SHD	F1
NOTEARS-ADMM	4.9	4.6
NOTEARS-MLP-ADMM	3	3.45
GS-FedDAG	5.15	4.95
AS-FedDAG	4.85	4.6
FedPC	**1.55**	2.1
FedACD (Ours)	**1.55**	**1.3**

To further analyze the significant difference between FedACD and the baselines, we perform the Nemenyi test [Demšar, 2006], which states that the performance levels of two algorithms are significantly different if the corresponding average rankings differ by at least one critical difference (CD). The CD for the Nemenyi test is calculated as follows (i.e., Eq. (35)).

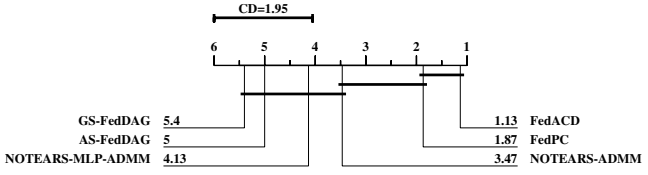
$$\mathrm{CD} = q_{\alpha,\theta}\sqrt{\frac{\theta(\theta+1)}{6\eta}}, \tag{35}$$

where α is the significance level, θ is the number of comparison algorithms, and η denotes the number of datasets with different numbers of clients. In our experiments, $\theta = 6$, $q_{\alpha=0.05,\theta=6} = 2.85$ at significance level $\alpha = 0.05$. When using benchmark BN datasets, $\eta = 3 * 5 = 15$ (three benchmark BN datasets across $\{3, 5, 10, 15, 20\}$ clients), and thus $\mathrm{CD} = 1.95$; when using synthetic Non-IID datasets, $\eta = 2 * 5 = 10$ (two synthetic Non-IID datasets across $\{3, 5, 10, 15, 20\}$ clients), and thus $\mathrm{CD} = 2.38$; when using real-world datasets, $\eta = 2 * 5 = 10$ (two real-world datasets across $\{3, 5, 10, 15, 20\}$ clients), and thus $\mathrm{CD} = 2.38$.

Figs. 4-6 provide the CD diagrams on three different types of datasets, respectively. In each CD diagram, the average ranking of each algorithm is marked along the axis (lower rankings to the right). On benchmark BN datasets, whether using SHD or F1 metrics, we observe that FedACD significantly outperforms NOTEARS-ADMM, NOTEARS-MLP-ADMM, GS-FedDAG and AS-FedDAG, and FedACD achieves a comparable performance against FedPC. On synthetic Non-IID datasets, when using SHD metric, we observe that FedACD significantly outperforms NOTEARS-ADMM, NOTEARS-MLP-ADMM and GS-FedDAG, and FedACD achieves a comparable performance against AS-FedDAG and FedPC; when using F1 metric, we observe that FedACD achieves a comparable performance against FedPC, and FedACD significantly outperforms the other baselines. On real-world datasets, whether using SHD or F1 metrics, we observe that FedACD achieves a comparable performance

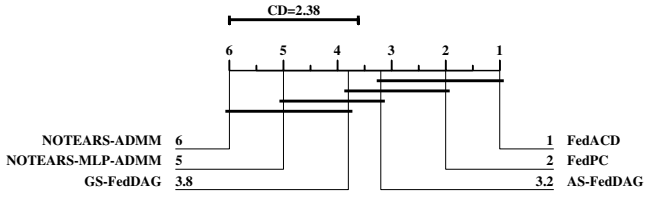


(a) SHD metric

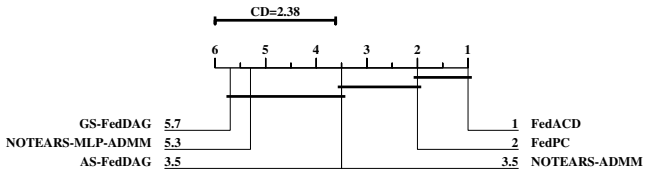


(b) F1 metric

Figure 4: Crucial difference diagram of the Nemenyi test for SHD and F1 metrics on benchmark BN datasets.

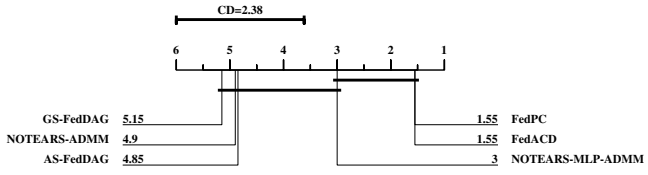


(a) SHD metric

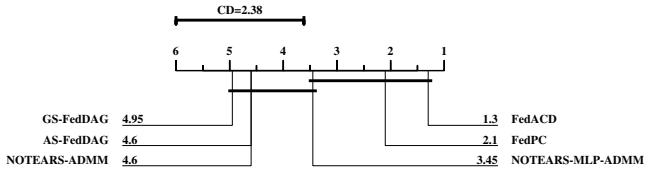


(b) F1 metric

Figure 5: Crucial difference diagram of the Nemenyi test for SHD and F1 metrics on synthetic Non-IID datasets.



(a) SHD metric



(b) F1 metric

Figure 6: Crucial difference diagram of the Nemenyi test for SHD and F1 metrics on real-world datasets.

against FedPC and NOTEARS-MLP-ADMM, and FedACD significantly outperforms the other baselines. Additionally, on all datasets, FedACD is the only algorithm that achieves the lowest ranking value whether using SHD or F1 metrics.

F Implementation Details

All experiments were conducted on a computer with an Intel Core i9-10900 3.70-GHz CPU, NVIDIA GeForce RTX 3060 GPU, and 32 GB memory. For the NOTEARS-ADMM[1], NOTEARS-MLP-ADMM[2], GS-FedDAG[3], AS-FedDAG[4] and FedPC[5] algorithms, we used the source codes provided by their authors. Among them, FedPC and our method are implemented in MATLAB, while the other algorithms are implemented in Python.

[1]https://github.com/ignavierng/notears-admm.

[2]https://github.com/ignavierng/notears-admm.

[3]https://github.com/ErdunGAO/FedDAG.

[4]https://github.com/ErdunGAO/FedDAG.

[5]https://github.com/Xianjie-Guo/FedPC.

The significance level for conditional independence (CI) tests is set to 0.01. NOTEARS-ADMM and NOTEARS-MLP-ADMM use 0.3 as the threshold to prune edges in a causal structure represented by an adjacency matrix, whereas GS-FedDAG and AS-FedDAG use 0.5 as the threshold. These are the same as the original paper. For all three types of data, the masking rate of FedACD is set to 0.6, as detailed in the sensitivity analysis in Section 5.6 of the main text.

References

[Demšar, 2006] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

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

[Ramsey, 2016] Joseph Ramsey. Improving accuracy and scalability of the PC algorithm by maximizing p-value. *arXiv preprint arXiv:1610.00378*, 2016.

[Scutari, 2016] Marco Scutari. An empirical-Bayes score for discrete Bayesian networks. In *Conference on Probabilistic Graphical Models*, pages 438–448. PMLR, 2016.

[Watanabe, 2013] Sumio Watanabe. A widely applicable Bayesian information criterion. *Journal of Machine Learning Research*, 14(1):867–897, 2013.