FEDERATED CAUSAL DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

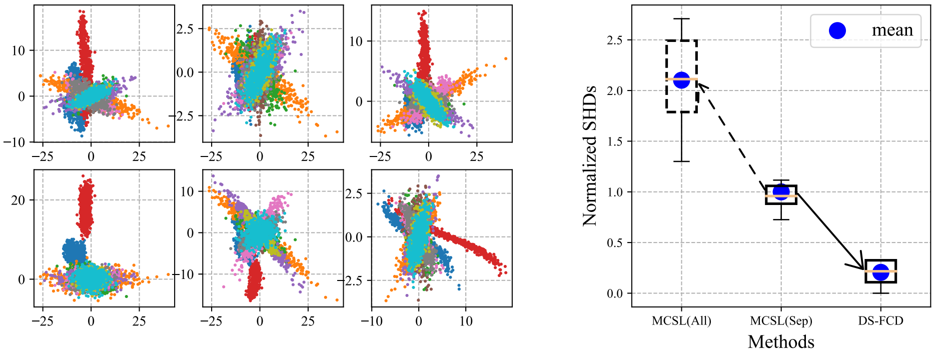
Causal discovery aims to learn a causal graph from observational data. To date, most causal discovery methods require data to be stored in a central server. However, data owners gradually refuse to share their personalized data to avoid privacy leakage, making this task more troublesome by cutting off the first step. A puzzle arises: *how do we infer causal relations from decentralized data?* In this paper, with the additive noise model assumption of data, we take the first step in developing a gradient-based learning framework named DAG-Shared Federated Causal Discovery (DS-FCD), which can learn the causal graph without directly touching local data and naturally handle the data heterogeneity. DS-FCD benefits from a two-level structure of each local model. The first level learns the causal graph and communicates with the server to get model information from other clients, while the second level approximates causal mechanisms and personally updates from its own data to accommodate the data heterogeneity. Moreover, DS-FCD formulates the overall learning task as a continuous optimization problem by taking advantage of an equality acyclicity constraint, which can be naturally solved by gradient descent methods. Extensive experiments on both synthetic and real-world datasets verify the efficacy of the proposed method.

1 INTRODUCTION

The discovery of causal relations among concerned variables is a fundamental and challenging problem in various fields, such as econometrics (Heckman, 2008), epidemiology (Greenland et al., 1999), and biological sciences (Imbens & Rubin, 2015). The requirement comes from the need of excavating the generation process behind data, guiding actions and policies, learning from the past (Pearl et al., 2016). To achieve this goal, a reliable way is to conduct randomized controlled (control) trials, which, however, may face difficulty or even be ethically forbidden in some cases (Resnik, 2008; Nardini, 2014). By leveraging the use of directed acyclic graphs (DAGs) to describe the cause-effect relations among variables, causal discovery (CD), which directly infers the causal relations from observational data by learning a DAG, brings a new solution for this problem and has received a great deal of attention (Peters et al., 2017; Glymour et al., 2019).

Various methods (Spirtes et al., 2001; Chickering, 2002; Shimizu et al., 2006; Zheng, 2020) for learning causal relations from purely observational data have been proposed over the recent decades. Regularly, (1) collecting data from various sources and then (2) designing a CD algorithm on all collected data are the common pipeline in this field. However, owing to the issue of data privacy, data owners gradually prefer not to share their personalized data with others (Kairouz et al., 2019). Naturally, the new predicament, *how do we infer causal relations from decentralized data?* has arisen. In statistical learning problems such as regression and classification, federated learning (FL) has been proposed to learn from locally stored data (McMahan et al., 2017). Inspired by these developments in FL, we aim to develop a federated causal discovery (FCD) framework that enables to learn DAG from decentralized data. Compared to FL in statistical learning, FCD, a **structural learning** task, has the following two main differences:

- **Learning objective difference.** FL aims to learn *an estimator* to fit the given conditional distribution while FCD tries to *find the underlying causal structure* and the *causal mechanism estimator* to fit with the joint distribution of observations.
- **Data heterogeneity difference.** FL usually cares about classification tasks where data heterogeneity is assumed by some specific distribution shift types such as label shift (the shift of $P(Y)$) (Lipton et al., 2018) or covariate shift (the shift of $P(X)$) (Reisizadeh et al., 2020), while FCD handles



(a) (b)

Figure 1: (a) Visualization of Non-IID data. (b) Normalized structural hamming distance (SHD)s (↓) of learned DAG, where MCSL(Sep) (Ng et al., 2019) separately trains model on local dataset while MCSL(All) trains one model on all data, which however is forbidden in FL.

a generative model where data heterogeneity means the joint distribution shift of all variables (as shown in Figure 1(a)), which would bring more challenges compared to the model design in the federated learning paradigm.

To overcome the aforementioned problem, we present DAG-Shared Federated Causal Discovery (DS-FCD), a gradient-based framework for learning the underlying causal graph from decentralized data, including the case of Non-Independent and Identically Distributed (Non-IID) data. (1) To alleviate the data leakage problem, DS-FCD inherits the merits of FL, which proposes to separately deploy a local model to each client and collaboratively learn a joint model at the server end. Instead of sharing raw data, DS-FCD exchanges model-info among clients and the server to achieve collaboration. (2) Taking into consideration of the first main difference between FCD and FL, a two-level structure consisting of causal graph learning (CGL) part and causal mechanism approximating (CMA) part respectively, is adopted as the local model. (3) Benefiting from this separated structure, the second difference between FL and FCD can naturally be handled by only sharing CGL parts of clients during FL and locally updating CMA to get with data heterogeneity. Moreover, we provide the identifiability conditions for learning the causal graph from decentralized data. Our contributions are summarized as follows:

- We introduce FCD, under the assumption that the underlying causal graph among different datasets remains invariant, while causal mechanisms are allowed to vary when it comes to the Non-IID setting. We also show the identifiability conditions of CD from decentralized data.
- We propose DS-FCD, which separately learns the causal mechanisms on local data and jointly learns the causal graph to elegantly handle data heterogeneity. Meanwhile, since 0 bits of raw data is shared but only the CGL parts of models, the privacy protection requirement is guaranteed and the communication pressure is quite low.
- We evaluate our proposed framework with data that follows a SEM with an additive noise structure on a variety of experimental settings, including simulated ablations and real dataset, against recent state-of-the-art algorithms for showing its superior performance and the ability to use one model for all settings.

2 Preliminaries

Additive Noise Models (ANM). A causal model is defined as a triple $\mathcal{M} = \langle \mathcal{X}, \mathcal{E}, \mathcal{F} \rangle$ where $\mathcal{E}$ is a set $\{\epsilon_1, \epsilon_2, \cdots, \epsilon_d\}$ of exogenous variables and $\mathcal{X} = \{X_1, X_2, \cdots, X_d\}$ is a set of endogenous variables. $\mathcal{F} = \{f_1, f_2, \cdots, f_d\}$ is a set of functions, where each f_i, called the causal mechanism of X_i, maps $\epsilon_i \cup \mathbf{PA}_i$ to X_i, i.e., $X_i = f_i(\mathbf{PA}_i, \epsilon_i)$, where the $\mathbf{PA}_i$ correspond to the direct parents of X_i. $\mathcal{M}$ can be leveraged to describe how nature assigns values to variables of interest (Pearl et al., 2016). Here, we narrow our focus to a commonly used model named ANM, which assumes

$$X_i = f_i(\mathbf{PA}_i) + \epsilon_i, \quad i = 1, 2, \cdots, d, \tag{1}$$

where ϵ_i is always taken as a random noise, which is independent of variables in $\mathbf{PA}_i$ and mutually independent with any ϵ_j for $i \neq j$.

Probabilistic Causal Graphical Models (PCGM). Let $X = (X_1, X_2, \cdots, X_d)$ be a vector that includes all variables in $\mathcal{X}$ with index set $\mathbb{V} := \{1, 2, \cdots, d\}$ and $P(X)$ be a marginal distribution induced from $\mathcal{M}$. A DAG $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ consists of a nodes set $\mathbb{V}$ and a edge set $\mathbb{E} \subseteq \mathbb{V}^2$. Every causal model $\mathcal{M}$ can be associated with a DAG $\mathcal{G}_{\mathcal{M}}$, in which each node i corresponds to a variable X_i and directed edges point from $\mathbf{PA}_i$ to X_i[1] for $i \in [d]$[2]. A PCGM is defined as a pair $\langle P(X), \mathcal{G}_{\mathcal{M}} \rangle$. Then $\mathcal{G}_{\mathcal{M}}$ is called the causal graph associated with $\mathcal{M}$ and $P(X)$ is Markovian to $\mathcal{G}_{\mathcal{M}}$. Thoughout this paper, we assume *Causal Sufficiency* (no hidden variable) (Spirtes et al., 2001) and then $P(X)$ can be factorized as

$$P(X) = \prod_{i=1}^{d} P(X_i | X_{pa_i}) \tag{2}$$

according to $\mathcal{G}_{\mathcal{M}}$ (Lauritzen, 1996). X_{pa_i} is the parental vector that includes all variables in $\mathbf{PA}_i$.

Characterizations of Acyclicity A DAG $\mathcal{G}$ with d nodes can be represented by a binary adjacency matrix $\boldsymbol{B} = [\boldsymbol{B}_{:,1} | \boldsymbol{B}_{:,2} | \cdots | \boldsymbol{B}_{:,d}]$ with $\boldsymbol{B}_{:,j} \in \{0, 1\}^d$. NOTEARS (Zheng et al., 2018) first formulates a sufficient and necessary condition for $\boldsymbol{B}$ representing a DAG by an equality constraint. The formulation is as follows:

$$\mathrm{Tr}[e^{\boldsymbol{B}}] - d = 0, \tag{3}$$

where $\mathrm{Tr}[\cdot]$ means the trace of a given matrix. $e^{(\cdot)}$, here, is the matrix exponential operation. To solve the non-linear model, Ng et al. (2019) proposes to use a mask $\boldsymbol{M}$, parameterized by a continuous proxy matrix $\boldsymbol{U}$, to approximate the adjacency matrix $\boldsymbol{B}$. To enforce the entries of $\boldsymbol{M}$ to approximate the binary form, i.e., 0 or 1. A two-dimensional version of Gumbel-Softmax (Jang et al., 2017) approach named Gumbel-Sigmoid is designed to reparameterize $\boldsymbol{U}$ and to ensure the differentiability of the model. Then, $\boldsymbol{M}$ can be obtained element-wisely by

$$\boldsymbol{M}_{ij} = \frac{1}{1 + \exp(-\log(\boldsymbol{U}_{ij} + \mathrm{Gumb}_{ij})/\tau)}, \tag{4}$$

where τ is the temperature, $\mathrm{Gumb}_{ij} = g^1_{ij} - g^0_{ij}$, g^1_{ij} and g^0_{ij} are two independent samples from $\mathrm{Gumbel}(0, 1)$. For simplicity but equivalence, g^1_{ij} and g^0_{ij} also can be sampled from $-\log(\log(a))$ with $a \sim \mathrm{Uniform}(0, 1)$. The proof can be found in Appendix D of (Ng et al., 2019). MCSL names Eq. (4) as Gumbel-Sigmoid w.r.t. $\boldsymbol{U}$ and temperature τ, which is written as $g_\tau(\boldsymbol{U})$. Then, the acyclicity constraint can be reformulated as

$$\mathrm{Tr}[e^{(g_\tau(\boldsymbol{U}))}] - d = 0. \tag{5}$$

3 Problem definition

Here, we first describe the property of decentralized data and the mechanisms of distribution shift among different clients if there exists data heterogeneity (Huang et al., 2020b; Mooij et al., 2020). Then, we define the problem, federated causal discovery, considered in this paper.

Decentralized Data and Probability distribution set. Let $\mathcal{C} = \{c_1, c_2, \cdots, c_m\}$ be the client set which includes m different clients and $\mathcal{S}$ be the only server. The data $\mathcal{D}^{c_k} \in \mathbb{R}^{n_{c_k} \times d}$, in which each observation $\mathcal{D}^{c_k}_i$ for $\forall i \in [n_{c_k}]$ independently sampled from its corresponding probability distribution $P^{c_k}(X)$, represents the personalized data owned by the client c_k, where n_{c_k} is the number of observations of $\mathcal{D}^{c_k}$. The dataset $\mathcal{D} = \{\mathcal{D}^{c_1}, \mathcal{D}^{c_2}, \cdots, \mathcal{D}^{c_m}\}$ is called a decentralized dataset and $P^{\mathcal{C}}(X) = \{P^{c_1}(X), P^{c_2}(X), \cdots, P^{c_m}(X)\}$ is defined as the decentralized probability distribution set. If $P^{c_{k_1}}(X) = P^{c_{k_2}}(X)$ for $\forall k_1, k_2 \in [m]$, then $\mathcal{D}$ is defined as an independent and identically distributed (IID) decentralized dataset throughout this paper. The Non-IID decentralized dataset is defined by assuming that there exists at least two clients, e.g., c_{k_1} and c_{k_2}, on which the local data are sampled from different distributions, i.e., $P^{c_{k_1}}(X) \neq P^{c_{k_2}}(X)$.

Assumption 1 **(Invariant DAG)** *For $\forall c_k$, $P^{c_k}(X) \in P^{\mathcal{C}}(X)$ admits the product factorization of Eq. (2) relative to the same DAG $\mathcal{G}$.*

[1] In this paper, $\mathcal{G}$ is only defined over the endogenous variables.

[2] For simplicity, we use $[d] = \{1, 2, \cdots, d\}$ to represent the set of all integers from 1 to d.

With Assumption 1, it is easy to conclude that distribution shift across $P^{c_k}(X)$ comes from the change of causal mechanisms $\mathcal{F}$ or distribution shift of the exogenous variables $\mathcal{E}$. That is to say, if $P(X^{c_{k_1}}) \neq P(X^{c_{k_2}})$, then, at least one of the following cases occurs. (1) $\exists i \in [d]$, $P^{c_{k_1}}(X_i|X_{pa_i}) \neq P^{c_{k_2}}(X_i|X_{pa_i})$, i.e., $f_i^{c_{k_1}} \neq f_i^{c_{k_2}}$ (2) $\exists i \in [d]$, $P^{c_{k_1}}(\epsilon_i) \neq P^{c_{k_2}}(\epsilon_i)$.

Federated Causal Discovery. Given the decentralized dataset $\mathcal{D}$ consisting of data from m clients while the corresponding $P^{\mathcal{C}}(X)$ satisfies Assumption 1, the aim of federated causal discovery is to identify the underlying DAG $\mathcal{G}$ from $\mathcal{D}$.

4 Methodology

To solve the aforementioned problem, we formulate a continuous score-based method named DAG-shared federated causal discovery (DS-FCD). Firstly, we define an objective function that guides all models from different clients to federally learn the underlying causal graph $\mathcal{G}$ (or adjacency matrix $\boldsymbol{B}$), and at the same time also learn personalized causal mechanisms for each client. As shown in Figure 2, for each client c_k, the local model consists of a causal graph learning (CGL) part and a causal mechanisms approximation (CMA) part. The CGL part is parameterized by a matrix $\boldsymbol{U}^{c_k} \in \mathbb{R}^{d \times d}$, which would be exactly the same for all clients finally[3]. To leverage every entry of $\boldsymbol{U}^{c_k}$ for approximating the binary entry of adjacency matrix. A Gumbel-Sigmoid method (Jang et al., 2017; Ng et al., 2019) represented as $g_\tau(\boldsymbol{U}^{c_k})$, is further leveraged to transform $\boldsymbol{U}^{c_k}$ to a differentiable approximation of the adjacency matrix. The causal mechanisms $f_1^{c_k}, f_2^{c_k}, \cdots, f_d^{c_k}$ are parameterized by d sub-networks, each of which has d inputs and one output. In the learning process, the CGL parts (specifically $\boldsymbol{U}^{c_k}$) of participating clients are shared with the server $\mathcal{S}$. Then, the processed information is broadcast to each client for self-updating its own matrix. The details of our method are demonstrated in the following subsections.

4.1 The Overall Learning objective

Now we present the overall learning objective of FCD as the following optimization problem:

$$\begin{aligned} \arg\max_{\boldsymbol{\Phi}, \boldsymbol{U}} \quad & \sum_{k=1}^{m} \mathfrak{S}^{c_k}(\mathcal{D}^{c_k}, \boldsymbol{\Phi}^{c_k}, \boldsymbol{U}) \\ \text{subject to} \quad & g_\tau(\boldsymbol{U}) \in \textbf{DAGs} \quad \Leftrightarrow \quad h(\boldsymbol{U}) = \operatorname{Tr}[e^{(g_\tau(\boldsymbol{U}))}] - d = 0, \end{aligned} \tag{6}$$

where $\boldsymbol{\Phi}^{c_k} := \{\boldsymbol{\Phi}_1^{c_k}, \boldsymbol{\Phi}_2^{c_k}, \cdots, \boldsymbol{\Phi}_d^{c_k}\}$ represents the CMA part of the model on c_k. $\mathfrak{S}^{c_k}(\cdot)$ is the scoring function for evaluating the fitness of local model of client c_k and observations $\mathcal{D}^{c_k}$. For score-based causal discovery, selecting a proper score function such as BIC score (Schwarz, 1978) or Generalized score (Huang et al., 2018) for the corresponding data generation model can guarantee to identify up the underlying ground-truth causal graph $\mathcal{G}$ because $\mathcal{G}$ is supposed to have the maximal score over Eq. (6). However, the global minimum is hard to reach by using gradient descent method due to the non-convexity of $h(\boldsymbol{U})$.

In this paper, the likelihood part leverages the distribution of $P(\mathcal{E})$, i.e., $P(\mathcal{E}|\mathcal{F}^{c_k}, \mathcal{G})$. According to Eq. (1), we have $\epsilon_i = X_i - f_i(\mathbf{PA}_i)$. To get ϵ_i, the first step is to select the parental set $\mathbf{PA}_i$ for X_i. This can be achieved by $\boldsymbol{B}[:, i] \circ X$, where $\circ$ means the element-wise product. In our paper, for client c_k, we predict the noise by $\epsilon_i = X_i - \boldsymbol{\Phi}_i(g_\tau(\boldsymbol{U})[:, i] \circ X)$, where $g_\tau(\boldsymbol{U})$ is to approximate $\boldsymbol{B}$ and $\boldsymbol{\Phi}_i(\cdot)$ is parameterized by a neural network to approximate f_i. The specific formulation of $\mathfrak{S}^{c_k}$ would depend on the assumption of noise distribution.

4.2 DAG-shared learning

As suggested in NOTEARS (Zheng et al., 2018), the hard-constraint optimization problem can be addressed by an Augmented Lagrangian method (ALM) to get an approximate solution. Similar to penalty methods, ALM transforms a constrained optimization problem by a series of unconstrained sub-problems and adds a penalty term to the objective function. ALM also introduces a Lagrangian

[3] Since CGL parts of different clients may not be the same during training, we index them.

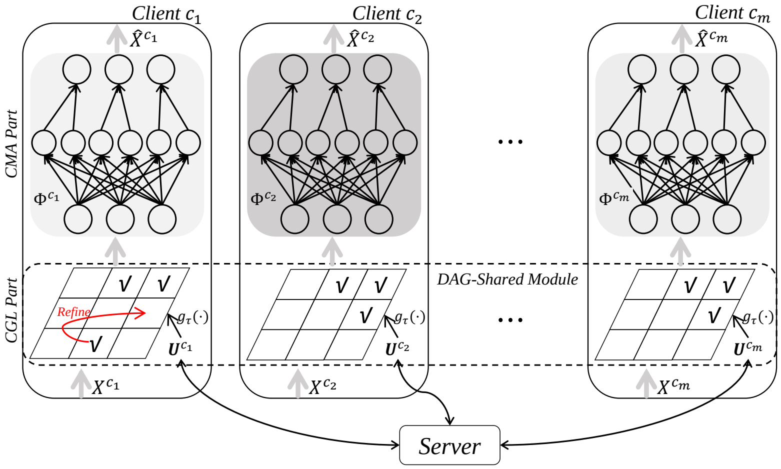


Figure 2: An overview of DS-FCD. Each solid-line box includes CD on each local client. For client c_k, the CGL part includes a continuous proxy U^{c_k} and $g_\tau(\cdot)$, the Gumbel-Sigmoid function, which maps U^{c_k} to approximate the binary causal graph. The CMA part uses Φ^{c_k}, a neural network, to approximate the causal mechanisms. X^{c_k} represents observations on c_k and $\hat{X}^{c_k}$ is the predicted data. X^{c_k} firstly goes through the CGL part to select the parental variables and then the CMA part to get $\hat{X}^{c_k}$. The server coordinates the FL procedures by leveraging $\boldsymbol{U}$ among clients.

multiplier term to avoid ill-conditioning by preventing the coefficient of penalty term from going too large. To solve Eq. (6), the sub-problem can be written as

$$\arg\max_{\boldsymbol{\Phi},\boldsymbol{U}} \sum_{k=1}^{m} \mathfrak{S}^{c_k}\left(\boldsymbol{\Phi}^{c_k}, D_i^{c_k}, g_\tau(\boldsymbol{U})\right) - \alpha_t h(\boldsymbol{U}) - \frac{\rho_t}{2} h(\boldsymbol{U})^2, \tag{7}$$

where α_t and ρ_t are the Lagrangian multiplier and penalty parameter of the t-th sub-problem, respectively. These parameters are updated after the sub-problem is solved. Since neural networks are adopted to fit the causal mechanisms in our work, there is no closed form for Eq. (7). We solve it approximately via Adam (Kingma & Ba, 2015). The method is described in Algorithms 1 and 2.

Algorithm 1 DAG-Shared Federated Causal Discovery (DS-FCD)

```
Inputs: $\mathcal{D}, \mathcal{C}$, Parameter-list = $\{\alpha_{init}, \rho_{init}, h_{tol}, it_{max}, \rho_{max}, \beta, \gamma, r\}$.
Output: $\mathbb{E}g_\tau(\boldsymbol{U}_t), \boldsymbol{\Phi}_t$
$t \leftarrow 1, \alpha_t \leftarrow \alpha_{init}, \rho_t \leftarrow \rho_{init}$        ▷ Parameter Initializing
while $t \leq it_{max}$ and $h(\boldsymbol{U}_t) \geq h_{tol}$ and $\rho \leq \rho_{max}$ do
    $U_{t+1}, \Phi_{t+1} \leftarrow$ SPS($\mathcal{D}, \mathcal{C}, \alpha_t, \rho_t, it_{in}, it_{fl}, r$)        ▷ Sub-problem Solving
    $\alpha_{t+1} \leftarrow \alpha_t + \rho_t \mathbb{E}[h(\boldsymbol{U}_{t+1})], \quad t \leftarrow t+1$        ▷ Coefficients Updating
    if $\mathbb{E}[h(\boldsymbol{U}_{t+1})] > \gamma \mathbb{E}[h(\boldsymbol{U}_t)]$ then
        $\rho_{t+1} = \beta \rho_t$
    else
        $\rho_{t+1} = \rho_t$
    end if
end while
```

Each sub-problem as Eq. (7) is solved mainly by distributing the computation across all clients. Since data is prevented from sharing among clients and the server, each client owns its personalized model, which is only trained on its personalized data. The server communicates with clients by exchanging the parameters information of models and coordinates the joint learning task. To achieve so, our method alternately updates the server and clients in each communication round.

Client Update. For each model of client c_k, there are two main parts, named CGL part parameterized by $\boldsymbol{U}^{c_k}$ and CMA part parameterized by $\boldsymbol{\Phi}^{c_k}$, respectively. Essentially, the joint objective in Eq. (7) guides the learning process. In the self-updating as described in Algorithm 2, client c_k

Algorithm 2 Sub-Problem Solver (SPS) for DS-FCD

```
Input: $\mathcal{D}$, $\mathcal{C}$, Parameter-list = $\{\alpha_t, \rho_t, it_{in}, it_{fl}, r\}$.
Output: $\boldsymbol{U}_{new}, \boldsymbol{\Phi}^{it_{in}}$
Define $\mathrm{SP}^{c_k} = \mathfrak{S}^{c_k} - \alpha_t h(\boldsymbol{U}) - \frac{\rho_t}{2} h(\boldsymbol{U})^2$
for $i$ in $(1, 2, \cdots, it_{in})$ do
    for each client $c_k$ do
        $\boldsymbol{U}^{i,c_k}, \boldsymbol{\Phi}^{i,c_k} \leftarrow \arg\max_{\boldsymbol{\Phi}^{c_k}, \boldsymbol{U}} \mathrm{SP}^{c_k}$    ▷ Self-updating
    end for
    if $i$ ($\%\ it_{fl}) = 0$ or $i = it_{in}$ then
        $\mathbb{U} \leftarrow \mathrm{Agg}(r, \mathcal{C})$ ▷ Model aggregating: select $r$ clients and collect their $\boldsymbol{U}$s into $\mathbb{U}$, send $\mathbb{U}$ to server
        $\boldsymbol{U}_{new} \leftarrow \mathrm{Avg}(\mathbb{U})$    ▷ Server Updating: average $\boldsymbol{U} \in \mathbb{U}$
        $\mathcal{C} \leftarrow \mathrm{BD}(\boldsymbol{U}_{new})$    ▷ Broadcasting: distribute $U_{new}$ to all clients
        for each client $c_k$ do
            $\boldsymbol{U}^{i,c_k} \leftarrow \boldsymbol{U}_{new}$    ▷ Clients Updating
        end for
    end if
end for
```

makes it_{fl} local gradient-based parameter updates to maximize its personalized score defined as $\mathrm{SP}^{c_k} = \mathfrak{S}^{c_k} - \alpha_t h(\boldsymbol{U}) - \frac{\rho_t}{2} h(\boldsymbol{U})^2$. The other *Clients Update* procedure is a federally update of $\boldsymbol{U}^{c_k}$. After receiving $\boldsymbol{U}_{new}$ from the server $\mathcal{S}$, each client directly replaces its $\boldsymbol{U}^{c_k}$ by $\boldsymbol{U}_{new}$.

Server Update. After it_{fl} local updates, the server $\mathcal{S}$ randomly chooses r clients to collect their $\boldsymbol{U}$s to the set $\mathbb{U}$. Then, $\boldsymbol{U}$s in $\mathbb{U}$ are averaged to get $\boldsymbol{U}_{new}$, which is then broadcast to clients. Notice that with assuming that data distribution across clients is IID, $\boldsymbol{\Phi}^{c_k}$ of the chosen r clients can also be collected and averaged to update clients' local models, which is named as All-Shared FCD (AS-FCD) in this paper. It is worth noting that AS-FCD can further enhance the performance in the IID case but introduce some additional communication costs.

4.3 Thresholding

As illustrated in the previous works (Zheng et al., 2018; Ng et al., 2019), the solution of ALM just satisfies the numerical precision of the constraint. This is because we set h_{tol} and it_{max} maximally but not infinite coefficients of penalty terms to formulate the last sub-problem. Therefore, some entries of the output $\boldsymbol{M} = \mathbb{E}g_{\tau}(\boldsymbol{U})$ will be near but not exactly 0 or 1. To alleviate this issue, ℓ_1 sparsity is added to the objective function. Moreover, after obtaining $\boldsymbol{M}$, we use a hard threshold 0.5 to prune the dense graph. If $\mathcal{G}(\boldsymbol{M})$ is still not a DAG, we take iterative thresholding to cut off the edge with the minimum weight until the graph is acyclic.

4.4 Privacy and Costs Discussion

Privacy issues of DS-FCD. The strongest motivation of FL is to avoid *personalized raw data leakage*. To achieve this, DS-FCD proposes to exchange the parameters for modelling the causal graph. Here, we argue that the information leakage of local data is rather limited. The server, receiving parameters with client index, may infer some data property. However, according to the data generation model (1), the distribution of local data is decided by (1) causal graph, (2) noise types/strengths and (3) causal mechanisms. The gradient information of the shared matrix is decided by (1) the type of learning objective and (2) model architecture, which are agnostic to the server. Especially for the network part, clients may choose different networks to make the inference more complex. Furthermore, one may leverage some advanced methods (Wei et al., 2020b) for easing this issue, but this is beyond the main scope of our study. Moreover, if the causal graph is also private information for clients, this problem can be easily solved by selecting a client to serve as the proxy server. For the proxy server, it needs to play two roles, including training its own model and taking the server's duties. Then, in the communication round, other clients communicate with the proxy server instead of a real server.

Communication cost. Since DS-FCD requires exchanging parameters between the server and clients, additional communication costs are raised. In our method, however, we argue that DS-FCD only brings rather small additional communication pressures. For the case of d variables, a single communication only exchanges a $d \times d$ matrix twice (sending and receiving). For the IID setting, which assumes that local data are sampled from the same distribution, one can also transmit the neural network together to further improve the performance since causal mechanisms are also shared among clients. The trade-off between performance and communication costs can also be controlled by r in Algorithm2, i.e., enlarging or reducing r. Surprisingly, we find that reducing r does not harm the performance severely (see Table 17 in Appendix A.5 for detailed results).

5 Experimental Results

In this section, we study the empirical performances of DS-FCD on both synthetic and real-world data. More detailed ablation experiments also can be found in Appendix A.5.

Baselines and Metrics. We compare our method with various baselines including some continuous search methods, named NOTEARS (Zheng et al., 2018), NOTEARS-MLP(Zheng et al., 2020), DAG-GNN (Yu et al., 2019) and MCSL (Ng et al., 2019), and also two traditional combinatorial search methods named PC (Spirtes et al., 2001) and GES (Chickering, 2002). We provide two training ways for these compared methods. The first way is using all data to train only one model, which, however, is not permitted in FCD since the ban of data sharing in our setting. For the IID data, the results on this setting can be an *approximate upper bound* of our method but unobtainable. The second one is separately training each local model over its personalized data, of which the performances reported are the average results of all clients. We report two metrics named Structural Hamming Distance (SHD) and True Positive Rate (TPR) averaged over 10 random repetitions to evaluate the discrepancies between estimated DAG and the ground-truth graph $\mathcal{G}$. Notice that PC and GES can only reach the completed partially DAG (CPDAG, or MEC) at most, which shares the same Skelton with the ground-truth DAG $\mathcal{G}$. When we evaluate SHD, we just ignore the direction of undirected edges learned by PC and GES. That is to say, these two methods can get SHD 0 if they can identify the CPDAG. The implementation details of all methods are detailed in Appendix A.3.

5.1 Synthetic data

The synthetic data we consider here is generated from Gaussian ANM (Model (1)). The score function used in all experiments can be seen in Appendix A.1. Two random graph models named Erdős-Rényi (ER) and Scale-Free (SF) are adopted to generate causal graph $\mathcal{G}$. And then, for each node V_i corresponding to X_i in $\mathcal{G}$, we sample a function from the given function sets to simulate f_i. Finally, data are generated according to a specific sampling method. In the following experiments, we take 10 clients and each with 600 observations throughout this paper. According to Assumption 1, data across all clients share the same causal graph for both IID and Non-IID settings.

5.1.1 IID setting

For the IID setting, all data are generated by an ANM and divided into 10 pieces. Each f_i is sampled from a Gaussian Process (GP) with RBF kernel of bandwidth one (See Table 14 and Table 15 in Appendix A.5 for results of other functions.) and noises are sampled from one zero-mean Gaussian distribution with fixed variance. We consider graphs of d nodes and $2d$ expected edges.

Experimental results are reported in Table 1with nodes 10 and 40. Since all local data are IID, here, we also provide another effective training method named AS-FCD, in which the CMA parts are also shared among clients. In all settings, AS-FCD shows a better performance than DS-FCD due to that more model information are shared during training. While DS-FCD can also show a consistent advantage over other methods. When separately training local models, all models suffer from data scarcity. Therefore, we can observe that both DS-FCD and AS-FCD perform better than other methods in the fashion of separate training. NOTEARS and DAG-GNN, as continuous search methods, obtain unsatisfactory results due to the weak model capacity and improper model assumption. While BIC score of GES gets a linear-Gaussian likelihood, which is incapable to deal with non-linear data[4]. With the number of nodes increasing, DS-FCD still shows better results than the closely-related baseline method MCSL. However, NOTEAES-MLP can show a comparable result with DS-FCD owing to the advantage over MCSL.

[4] Please find the ablation experiment with linear data and more discussions of the experimental results in Section A.4 of the Appendix

Table 1: Results on nonlinear ANM with GP (IID).

		ER2 with 10 nodes		SF2 with 10 nodes		ER2 with 40 nodes		SF2 with 40 nodes	
		SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑
All data	PC	15.3 ± 2.6	0.37 ± 0.10	14.1 ± 4.3	0.44 ± 0.20	84.9 ± 13.4	0.40 ± 0.08	95.0 ± 10.4	0.36 ± 0.07
	GES	13.0 ± 3.9	0.50 ± 0.18	9.6 ± 4.4	0.71 ± 0.17	59.0 ± 9.8	0.53 ± 0.08	73.8 ± 11.9	0.47 ± 0.10
	NOTEARS	16.5 ± 2.0	0.05 ± 0.04	14.5 ± 1.1	0.09 ± 0.07	71.2 ± 7.2	0.08 ± 0.03	70.8 ± 2.3	0.07 ± 0.03
	N-S-MLP	8.1 ± 3.8	0.56 ± 0.17	8.3 ± 2.8	0.51 ± 0.16	45.3 ± 6.8	0.43 ± 0.08	49.2 ± 7.7	0.39 ± 0.09
	DAG-GNN	16.2 ± 2.1	0.07 ± 0.06	15.2 ± 0.8	0.05 ± 0.05	73.0 ± 7.7	0.06 ± 0.03	72.4 ± 1.6	0.05 ± 0.02
	MCSL	1.9 ± 1.5	*0.90 ± 0.08*	*1.6 ± 1.2*	*0.91 ± 0.07*	*25.4 ± 13.1*	*0.68 ± 0.14*	31.6 ± 10.0	*0.59 ± 0.13*
Sep data	PC	14.1 ± 2.4	0.31 ± 0.06	13.6 ± 2.7	0.30 ± 0.10	83.8 ± 7.4	0.24 ± 0.03	86.1 ± 4.6	0.23 ± 0.04
	GES	12.7 ± 2.7	0.37 ± 0.09	12.7 ± 2.4	0.33 ± 0.11	71.0 ± 6.7	0.29 ± 0.03	73.2 ± 4.4	0.29 ± 0.05
	NOTEARS	16.5 ± 2.0	0.06 ± 0.04	14.6 ± 1.0	0.09 ± 0.06	71.1 ± 7.3	0.08 ± 0.03	70.7 ± 2.0	0.07 ± 0.03
	N-S-MLP	8.5 ± 2.9	0.56 ± 0.13	8.7 ± 2.9	0.53 ± 0.16	51.0 ± 6.9	0.41 ± 0.06	53.6 ± 5.5	0.39 ± 0.08
	DAG-GNN	15.7 ± 2.3	0.11 ± 0.05	14.5 ± 1.0	0.10 ± 0.06	71.5 ± 7.5	0.08 ± 0.02	70.8 ± 1.8	0.07 ± 0.02
	MCSL	7.1 ± 3.2	0.83 ± 0.08	6.9 ± 2.8	0.84 ± 0.08	77.3 ± 19.8	0.64 ± 0.11	72.9 ± 16.4	**0.58 ± 0.13**
	DS-FCD	**2.4 ± 2.0**	**0.86 ± 0.13**	**2.7 ± 2.2**	**0.86 ± 0.13**	**36.5 ± 12.1**	**0.65 ± 0.15**	**46.4 ± 10.4**	0.57 ± 0.13
	AS-FCD	**1.8 ± 2.0**	**0.89 ± 0.12**	**2.5 ± 2.7**	**0.85 ± 0.15**	**30.0 ± 12.3**	**0.74 ± 0.15**	**31.5 ± 10.0**	**0.59 ± 0.13**

5.1.2 Non-IID setting

As defined in Section3, the Non-IID property of data across clients come from the changes of causal mechanisms or the shift of noise distributions. To simulate the Non-IID data, we firstly generate a DAG shared by all clients and then decide the types of causal mechanisms $f_i^{c_k}$ and noises ϵ_i for $i \in [d]$ for each client c_k. In our experiments, We allow that f^{c_k} can be linear or non-linear for each client. If being linear, f^{c_k} here is a weighted adjacency matrix with coefficients sampled from Uniform $([-2.0, -0.5] \cup [0.5, 2.0])$, with equal probability. If being non-linear, $f_i^{c_k}$ is independently sampled from GP, GP-add, MLP or MIM functions (Yuan, 2011), randomly. Then, a fixed zero-mean Gaussian noise is set to each client with a randomly sampled variance from $\{0.8, 1\}$.

We can see that the conclusion of experimental results on the Non-IID setting is rather similar to that of the IID. As can be read from Table 2, DS-FCD always shows the best performances across all settings. If taking all data together to train one model using other methods, we can see that data heterogeneity would put great trouble to all compared methods while DS-FCD plays pretty well. Moreover, DS-FCD shows consistent good results with different numbers of observations on each client (see Table 16). NOTEARS takes second place at the setting of 40 nodes because there are some linear data among clients, which is also the reason that DS-FCD shows lower SHDs on Non-IID data in Table 2 than Table 1. Compared with Non-linear models, NOTEARS easily fits well with even fewer linear data.

Table 2: Results on ANM (Non-IID).

		ER2 with 10 nodes		SF2 with 10 nodes		ER2 with 40 nodes		SF2 with 40 nodes	
		SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑
All data	PC	22.3 ± 4.2	0.41 ± 0.11	21.0 ± 3.6	0.41 ± 0.12	151.9 ± 14.2	0.27 ± 0.08	152.5 ± 5.4	0.26 ± 0.04
	GES	26.4 ± 6.2	0.53 ± 0.14	25.4 ± 4.6	0.54 ± 0.13	NaN	NaN	NaN	NaN
	NOTEARS	20.4 ± 4.1	0.49 ± 0.14	18.7 ± 3.3	0.45 ± 0.11	164.8 ± 47.4	0.39 ± 0.07	178.1 ± 33.0	0.40 ± 0.10
	N-S-MLP	22.8 ± 5.0	0.87 ± 0.07	24.7 ± 3.3	0.88 ± 0.07	344.4 ± 71.9	*0.92 ± 0.08*	325.0 ± 50.2	*0.85 ± 0.08*
	DAG-GNN	21.2 ± 6.0	0.39 ± 0.11	16.6 ± 3.0	0.48 ± 0.18	146.6 ± 41.6	0.29 ± 0.08	168.2 ± 34.2	0.31 ± 0.09
	MCSL	19.4 ± 4.4	0.75 ± 0.19	19.0 ± 4.0	0.81 ± 0.14	118.6 ± 18.1	0.68 ± 0.11	126.9 ± 16.5	0.59 ± 0.12
Sep data	PC	12.5 ± 2.7	0.45 ± 0.07	11.0 ± 2.1	0.49 ± 0.07	65.7 ± 11.0	0.43 ± 0.06	73.7 ± 5.5	0.36 ± 0.05
	GES	12.9 ± 2.6	0.58 ± 0.07	10.3 ± 2.8	0.60 ± 0.09	68.2 ± 20.8	0.65 ± 0.09	77.2 ± 13.8	0.60 ± 0.07
	NOTEARS	7.6 ± 2.6	0.60 ± 0.11	7.6 ± 1.8	0.58 ± 0.09	**34.9 ± 12.7**	0.63 ± 0.11	**43.4 ± 8.4**	0.53 ± 0.10
	N-S-MLP	**5.2 ± 1.4**	**0.80 ± 0.05**	**6.1 ± 1.6**	**0.76 ± 0.05**	46.0 ± 10.2	**0.73 ± 0.08**	56.0 ± 9.5	**0.66 ± 0.09**
	DAG-GNN	8.2 ± 2.9	0.67 ± 0.12	8.4 ± 2.1	0.67 ± 0.09	45.7 ± 13.5	0.64 ± 0.11	52.7 ± 8.4	0.60 ± 0.11
	MCSL	9.2 ± 1.8	0.72 ± 0.06	8.9 ± 2.0	0.71 ± 0.08	76.1 ± 13.7	0.53 ± 0.09	78.1 ± 6.3	0.47 ± 0.07
	DS-FCD	**1.9 ± 1.6**	**0.99 ± 0.02**	**2.6 ± 1.3**	**0.93 ± 0.07**	**24.3 ± 10.2**	**0.86 ± 0.09**	**33.9 ± 10.9**	**0.73 ± 0.09**

5.2 Real data

We consider a real public dataset named **fMRI Hippocampus** (Poldrack et al., 2015) to discover the causal relations among six brain regions. This dataset records signals from six separate brain regions in the resting state of one person in 84 successive days and the anatomical structure provides 7 edges as the ground truth graph (see Figure A.5 in Appendix A.5). Herein, we separately select 500 records in each of 10 days (see Figure 7 for the normalized data distribution in Appendix A.5), which can be regarded as different local data. It is worth noting that though this data does not have a real data privacy problem, we can use this dataset to evaluate the learning accuracy of our method. Here, in Table 3 we show part of the experimental results while others lie in Table 18 (Appendix A.5). AS-FCD shows the best performance over all criterion while DS-FCD also performs better than most of the other methods.

Table 3: Empirical results on **fMRI Hippocampus** dataset (Part 1).

	All data			Separate data			DS-FCD	AS-FCD
	PC	NOTEARS	MCSL	PC	NOTEARS	MCSL		
SHD ↓	9.0 ± 0.0	*5.0 ± 0.0*	9.0 ± 0.6	8.7 ± 1.3	8.0 ± 1.9	8.3 ± 1.7	**6.4 ± 0.9**	**5.0 ± 0.0**
NNZ	11.0 ± 0.0	4.0 ± 0.0	12.0 ± 0.6	7.6 ± 1.3	5.4 ± 1.5	9.0 ± 1.7	6.8 ± 0.6	5.0 ± 0.0
TPR ↑	0.43 ± 0.00	0.29 ± 0.00	*0.44 ± 0.04*	0.26 ± 0.11	0.19 ± 0.18	**0.35 ± 0.15**	0.27 ± 0.12	**0.29 ± 0.00**
FDR ↓	0.73 ± 0.00	*0.50 ± 0.00*	0.74 ± 0.03	0.76 ± 0.10	0.78 ± 0.19	0.73 ± 0.11	**0.72 ± 0.11**	**0.60 ± 0.00**

s

6 Related work

Two mainstreams named constraint-based and score-based methods push the development of causal discovery. Constraint-based methods, including PC and fast causal inference (FCI) (Spirtes et al., 2001), take conditional independence constraints induced from the observed distribution to decide the graph skeleton and part of the directions. Another branch of methods (Chickering, 2002) define a score function, which evaluate the fitness between the distribution and graph, and identify the graph $\mathcal{G}$ with the highest score after searching the DAG space. To avoid solving the combinatorial optimization problem, NOTEARS (Zheng et al., 2018) introduces an equivalent acyclicity constraint and formulates a fully continuous optimization for searching the graph. Following this work, many works leverages this constraint to non-linear case (Zheng et al., 2020; Lachapelle et al., 2020; Zhu et al., 2020; Wang et al., 2021), interventional data (Brouillard et al., 2020), time-series data (Pamfil et al., 2020), and unmeasured confounding (Bhattacharya et al., 2021). DAG-NoCurl (Yu et al., 2021) and NOFEARS (Wei et al., 2020a) focus on the optimization aspect.

The second line of related work is on the Overlapping Datasets (OD) (Danks et al., 2009; Tillman & Spirtes, 2011; Triantafillou & Tsamardinos, 2015; Huang et al., 2020a) problem in causal discovery. OD assumes each dataset owns observations of partial variables and targets learning the integrated DAG from multiple datasets. In these works, data from different sites need to be put together on a central server.

The last line is on federated learning (Yang et al., 2019; Kairouz et al., 2019), which provides the joint training paradigm to learn from decentralized data while avoiding sharing raw data during the learning process. FedAvg (McMahan et al., 2017) first formulates and names federated learning. FedProx (Li et al., 2020) studies the Non-IID case and provides the convergence analysis results. SCAFFOLD leverages variance reduction by correcting client-shift to enhance the training efficiency. Besides these fundamental problems in FL itself, this novel learning way has been widely co-operated with or applied to many real-world tasks such as healthcare (Sheller et al., 2020), recommendation system (Yang et al., 2020), and smart transport (Samarakoon et al., 2019).

7 Conclusion

Learning causal structures from decentralized data brings huge challenges to traditional causal discovery methods. In this context, we have introduced the first federated causal discovery method called DS-FCD, which uses a two-level structure of each local model. During the learning process, each client tries to learn an adjacency matrix to approximate the causal graph and neural networks to approximate the causal mechanisms. The matrix parts of participating clients are aggregated and processed by the server and then broadcast to each client for updating its personalized matrix. The overall problem is formulated as a continuous optimization problem and solved by gradient descent methods. Structural identifiability conditions are provided and extensive experiments on various data sets show the effectiveness of our DS-FCD.

8 Ethics Statement

In this paper, we propose DS-FCD to enable causal discovery from decentralized data. We first formulate the federated causal learning problem and provide DS-FCD to solve this problem, which is proved to be efficient by various experiments. In future, owing to privacy protection, it would be hard to collect personalized data and brings great challenges to causal discovery. DS-FCD just paves a novel way for learning causal relationships while avoiding data sharing.

9 Reproducibility Statement

For the sake of reproducibility of our proposed, we make the following efforts: (**i**) In Appendix A.3, we clearly state the implementations of compared methods in this paper and all hyper-parameters of our model in all settings. (**ii**) At last, we will open-source the source codes, trained models, detailed training logs upon acceptance.

References

Rohit Bhattacharya, Tushar Nagarajan, Daniel Malinsky, and Ilya Shpitser. Differentiable causal discovery under unmeasured confounding. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. *Advances in Neural Information Processing Systems*, 2020.

Peter Bühlmann, Jonas Peters, and Jan Ernest. Cam: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556, 2014.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(Nov):507–554, 2002.

David Danks, Clark Glymour, and Robert Tillman. Integrating locally learned causal structures with overlapping variables. 2009.

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

Sander Greenland, Judea Pearl, and James M Robins. Causal diagrams for epidemiologic research. *Epidemiology*, pp. 37–48, 1999.

James J Heckman. Econometric causality. *International statistical review*, 76(1):1–27, 2008.

Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *Advances in Neural Information Processing Systems*, 2008.

Biwei Huang, Kun Zhang, Yizhu Lin, Bernhard Schölkopf, and Clark Glymour. Generalized score functions for causal discovery. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

Biwei Huang, Kun Zhang, Mingming Gong, and Clark Glymour. Causal discovery from multiple data sets with non-identical variable sets. volume 34, pp. 10153–10161, 2020a.

Biwei Huang, Kun Zhang, Jiji Zhang, Joseph Ramsey, Ruben Sanchez-Romero, Clark Glymour, and Bernhard Schölkopf. Causal discovery from heterogeneous/nonstationary data. *Journal of Machine Learning Research*, 21:1–53, 2020b.

Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *International Conference on Learning Representations*, 2017.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Marcus Kaiser and Maksim Sipos. Unsuitability of notears for causal graph discovery. *arXiv preprint arXiv:2104.05441*, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. Gradient-based neural dag learning. In *International Conference on Learning Representations*, 2020.

Steffen Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Conference on Machine Learning and Systems*, 2020.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pp. 3122–3130. PMLR, 2018.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 2017.

Joris M Mooij, Sara Magliacane, and Tom Claassen. Joint causal inference from multiple contexts. *Journal of Machine Learning Research*, 21:1–108, 2020.

Yongchan Na and Jihoon Yang. Distributed bayesian network structure learning. pp. 1607–1611. IEEE, 2010.

Cecilia Nardini. The ethics of clinical trials. *Ecancermedicalscience*, 8, 2014.

Ignavier Ng, Zhuangyan Fang, Shengyu Zhu, Zhitang Chen, and Jun Wang. Masked gradient-based causal structure learning. *arXiv preprint arXiv:1910.08527*, 2019.

Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

Jonas Peters and Peter Bühlmann. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 2014.

Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Identifiability of causal graphs using functional models. *Conference on Uncertainty in Artificial Intelligence*, 2011.

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

Russell A Poldrack, Timothy O Laumann, Oluwasanmi Koyejo, Brenda Gregory, Ashleigh Hover, Mei-Yen Chen, Krzysztof J Gorgolewski, Jeffrey Luci, Sung Jun Joo, Ryan L Boyd, et al. Long-term neural and physiological phenotyping of a single human. *Nature communications*, 6(1): 1–15, 2015.

Amirhossein Reisizadeh, Farzan Farnia, Ramtin Pedarsani, and Ali Jadbabaie. Robust federated learning: The case of affine distribution shifts. *arXiv preprint arXiv:2006.08907*, 2020.

David B Resnik. Randomized controlled trials in environmental health research: ethical issues. *Journal of environmental health*, 70(6):28, 2008.

Sumudu Samarakoon, Mehdi Bennis, Walid Saad, and Mérouane Debbah. Distributed federated learning for ultra-reliable low-latency vehicular communications. *IEEE Transactions on Communications*, 68(2):1146–1159, 2019.

Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pp. 461–464, 1978.

Micah J Sheller, Brandon Edwards, G Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R Colen, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):1–12, 2020.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

Peter Spirtes, Clark Glymour, Richard Scheines, et al. *Causation, Prediction, and Search*, volume 1. The MIT Press, 2001.

Robert Tillman and Peter Spirtes. Learning equivalence classes of acyclic models with latent and selection variables from multiple datasets with overlapping variables. 2011.

Sofia Triantafillou and Ioannis Tsamardinos. Constraint-based causal discovery from multiple interventions over overlapping variable sets. *Journal of Machine Learning Research*, 16(1):2147–2205, 2015.

Xiaoqiang Wang, Yali Du, Shengyu Zhu, Liangjun Ke, Zhitang Chen, Jianye Hao, and Jun Wang. Ordering-based causal discovery with reinforcement learning. *International Joint Conference on Artificial Intelligence*, 2021.

Dennis Wei, Tian Gao, and Yue Yu. Dags with no fears: A closer look at continuous optimization for learning bayesian networks. *Advances in Neural Information Processing Systems*, 2020a.

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020b.

Liu Yang, Ben Tan, Vincent W Zheng, Kai Chen, and Qiang Yang. Federated recommendation systems. pp. 225–239. Springer, 2020.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. DAG-GNN: DAG structure learning with graph neural networks. *International Conference on Machine Learning*, 2019.

Yue Yu, Tian Gao, Naiyu Yin, and Qiang Ji. Dags with no curl: An efficient DAG structure learning approach. *International Conference on Machine Learning*, 2021.

Ming Yuan. On the identifiability of additive index models. *Statistica Sinica*, pp. 1901–1911, 2011.

Kun Zhang and Aapo Hyvarinen. On the identifiability of the post-nonlinear causal model. *Conference on Uncertainty in Artificial Intelligence*, 2009.

Xun Zheng. *Learning DAGs with Continuous Optimization*. PhD thesis, Carnegie Mellon University, 2020.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. DAGs with NO TEARS: Continuous Optimization for Structure Learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Xun Zheng, Chen Dan, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Learning sparse nonparametric DAGs. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Shengyu Zhu, Ignavier Ng, and Zhitang Chen. Causal discovery with reinforcement learning. In *International Conference on Learning Representations*, 2020.

A Appendix

A.1 Score function in this paper

Throughout all experiments in this paper, we assume the noise type are Gaussian with equal variance for each local distribution. And, the overall score function utilized in this paper is as follows,

$$\mathfrak{S}^{c_k}(\mathcal{D}^{c_k}, \mathbf{\Phi}^{c_k}, \boldsymbol{U}) = -\frac{1}{2n_k}\sum_{i=1}^{n_k}\sum_{j}^{d} \|\mathcal{D}^{c_k}_{ij} - \mathbf{\Phi}^{c_k}_j(g_\tau(\boldsymbol{U}_{j,:}) \circ \mathcal{D}^{c_k}_i)\|_2^2 - \lambda\|g_\tau(\boldsymbol{U})\|_1 \tag{8}$$

In our score function, we take the negative Least Squares loss and a sparsity term, which corresponds to the model complexity penalty in the BIC score (Schwarz, 1978).

A.2 Structure identifiability

Besides exploring effective causal discovery methods, identifiability conditions of causal model (Spirtes et al., 2001) are also important. In general, unique identification of the ground truth DAG is impossible from purely observational data without some specific assumptions. However, accompanying some specific data generation assumptions, the causal graph can be identified (Peters et al., 2011; Peters & Bühlmann, 2014; Zhang & Hyvarinen, 2009; Shimizu et al., 2006; Hoyer et al., 2008). We first give the definition of identifiability in the decentralized setting.

Definition 1 *Consider a decentralized distribution set $P^{\mathcal{C}}(X)$ satisfying Assumption 1. Then, $\mathcal{G}$ is said to be identifiable if $P^{\mathcal{C}}(X)$ cannot be induced from any other DAG.*

Condition 1 (Cond. 19 in (Peters & Bühlmann, 2014)) *The triple $(f_j, P(X_i), P(\epsilon_j))$ does not solve the following differential equation for all x_i, x_j with $v''(x_j - f(x_i))f'(x_i) \neq 0$:*

$$\xi''' = \xi''\left(-\frac{\nu'''f'}{\nu''} + \frac{f''}{f'}\right) - 2\nu''f''f' + \nu'f''' + \frac{\nu'\nu'''f''f'}{\nu''} - \frac{\nu'\left(f''\right)^2}{f'}.$$

Here, $f := f_j$ and $\xi := logP(X_i)$, and $v := logP(\epsilon_j)$ are the logarithms of the strictly positive densities.

Definition 2 (Restricted ANM. Def. 27 in (Peters & Bühlmann, 2014)) *Consider an ANM with d variables. This SEM is called restricted ANM if for all $j \in \mathbb{V}, i \in \mathbf{PA}_j$ and all sets $\mathbb{S} \subseteq \mathbb{V}$ with $\mathbf{PA}_j\backslash\{i\} \subseteq \mathbb{S} \subseteq \mathbf{PA}_j\backslash\{i,j\}$, there is an $x_{\mathbb{S}}$ with $P(x_{\mathbb{S}}) > 0$, s.t. the tripe*

$$\left(f_j(x_{\mathbf{PA}_j\backslash\{i\}}, \underbrace{\cdot}_{X_i}), P\left(X_i \mid X_{\mathbb{S}} = x_{\mathbb{S}}\right), P\left(\epsilon_j\right)\right)$$

satisfies Condition1. Here, the under-brace indicates the input component of f_j for variable X_i. In particular, we require the noise variables to have non-vanishing densities and the functions f_j to be continuous and three times continuously differentiable.

Condition 2 (Causal Minimality) *Given the joint distribution $P(X)$, $P(X)$ is Markov to a DAG $\mathcal{G}$ but not Markov to any subgraph of $\mathcal{G}$.*

Assumption 2 *Let a distribution $P(X)$ with $X = (X_1, X_2, \cdots, X_d)$ be induced from a restricted ANM with graph $\mathcal{G}$, and $P(X)$ satisfies Causal Minimality w.r.t $\mathcal{G}$.*

Assumption 3 *Let $P^{\mathcal{C}}(X)$ satisfy Assumption 1. At least one distribution $P^{c_k}(X) \in P^{\mathcal{C}}(X)$ meets Assumption 2 and the other distributions are faithful to $\mathcal{G}$.*

Proposition 1 *Given $P^{\mathcal{C}}(X)$ satisfying Assumption 3, and then, $\mathcal{G}$ can be identified up from $P^{\mathcal{C}}(X)$.*

Remark 1 *If $P^{\mathcal{C}}(X)$ satisfies Assumption 1, then, each $P^{c_k}(X) \in P^{\mathcal{C}}(X)$ is Markov relative to $\mathcal{G}$.*

Proof of Prop.1. From Remark 1, we have $P^{c_k}(X) \in P^{\mathcal{C}}(X)$ for $\forall c_k$, is Markov with $\mathcal{G}$. For each $c_k \in \mathcal{C}$ with $P^{c_k}(X)$ does not satisfy Assumption 2, the Completed Partially DAG (CPDAG) $\hat{\mathcal{G}}$ (Pearl, 2009), which represents the CPDAG induced by $\mathcal{G}$, can be identified (Spirtes et al., 2001). (1) That also says that these distributions can be induced from any DAG induced from $\mathcal{M}(\mathcal{G})$, including $\mathcal{G}$ definitely. Notice that skeleton($\hat{\mathcal{G}}$) = Skeleton($\mathcal{G}$) and any $X_i \leftarrow X_j$ in $\hat{\mathcal{G}}$ is also existed in $\mathcal{G}$. Then, for those c_k with with $P^{c_k}(X)$ satisfying Assumption 2, $\mathcal{G}$ can be identified. (2) That is to say, distributions satisfying Assumption 2 can only be induced from $\mathcal{G}$. Then, two kinds of graph, $\hat{\mathcal{G}}$ and $\mathcal{G}$, are obtained. Therefore, $\mathcal{G}$ can be easily identified. With (1) and (2), $P^{c_k}(X) \in P^{\mathcal{C}}(X)$ for $\forall c_k$ can only be induced by $\mathcal{G}$. Then, $\mathcal{G}$ is said to be identifiable ■

A.3 Implementations

The comparing causal discovery methods used in this paper all have available implementations, listed below:

- PC and MCSL: Codes are available at gCastle `https://github.com/huawei-noah/trustworthyAI/tree/master/gcastle`. The first author of MCSL added the implementation in this package.
- NOTEARS and NOTEARS-MLP: Codes are available at the first author's GitHub repository `https://github.com/xunzheng/notears`
- DAG-GNN: Codes are available at the author's GitHub repository `https://github.com/fishmoon1234/DAG-GNN`
- GES: an implementation of GES is available at `https://github.com/juangamella/ges`
- CAM: the codes of CAM is available at CRAN R package repository `https://cran.r-project.org/src/contrib/Archive/CAM/`

Our implementation is highly based on the existing Tool-chain named gCastle, which includes many gradient-based causal discovery methods.

A.3.1 Hyper-parameters setting

In all experiments, there is no extra hyper-parameter to adjust for PC (with Fisher-z test and p-value 0.01) and GES (BIC score). For NOTEARS, NOTEARS-MLP and DAG-GNN, we use the default hyper-parameters provided in their papers/codes. For MCSL, the hyper-parameters need to be modified are ρ_{init} and β. Specifically, if experimental settings (10 variables and 20 variables) are the same as those in their paper, we just take all the recommended hyper-parameters. For settings not implemented in their paper (40 variables exactly), we have two kinds of implementations. The first one is taking a linear interpolation for choosing the hyper-parameters. The second one is taking the same parameters as ours. We find that the second choice always works better. In our experiment, we report the experimental results done in the second way. Notice that CAM pruning is also introduced to improve the performance of MCSL, which however can not guarantee a better result in our settings. For simplicity and fairness, we just take the direct outputs of MCSL.

Similar to MCSL (Ng et al., 2019) and GraN-DAG (Lachapelle et al., 2020), we implement several experiments on simulated data with known causal graphs to search for the hyper-parameters and then use these hyper-parameters for all the simulated experiments. Specifically, we use seeds $1 \sim 10$ to generate the simulated data to search for the best combination of hyperparameters while all our experimental results reported in this paper are all conducted using seeds $2021 \sim 2030$.

A.3.2 Hyper-parameters in real-data setting

Most CSL methods have hyper-parameters, more or less, which need to be decided prior to learning. Moreover, NN-based methods are especially sensitive to the selection of hyper-parameters. For instance, Gran-DAG (Lachapelle et al., 2020) defines a really large hyper-parameters space for searching the optimal combination, which even uses different learning rates for the first subproblem and the other subproblems. MCSL and DS-FCD are sensitive to the selection of ρ_{init} and β when constructing and solving the subproblem. As pointed out in (Kairouz et al., 2019), NOTEARS focus more on optimizing the scoring term in the early stage and pays more attention to approximate DAG in the late stage. If NOTEARS cannot find a graph near $\mathcal{G}$ in the early stage, then, it would lead to a worse result.

To alleviate this problem, one may choose to (1) enlarge the learning rate or take more steps when solving the first few subproblems as Gran-DAG; (2) reduce the value of coefficient ρ_{init} to let the optimizer pay more attention to the scoring term in the early stages as MCSL. The other trick we find when dealing with real data is increasing ℓ_1. This mostly results from that real data may not fit well with the data generation assumptions in most papers. Therefore, we choose to conduct a grid search to find the best combination of $\rho_{init}, \beta, \ell_1$ for causal discovery on real data.

In the practice of causal discovery, it is impossible to have $\mathcal{G}$ to select the hyper-parameters. One common approach is trying multiple hyper-parameter combinations and keeping the one yielding the best score evaluated on a validation set (Koller & Friedman, 2009; Ng et al., 2019; Lachapelle et al., 2020). However, the direct use of this method may not work for some algorithms, such as MCSL, NOTEARS-MLP, and DS-FCD. This mainly lies in the similar explanations of the property of the traditional solution of AL. In the late stage of optimization, the optimizer focuses heavily on finding *a DAG* by enlarging the penalty coefficient ρ. Then, the learning of causal mechanisms would be nearly ignored. To address this problem, we firstly report the DAG directly learned by a combination of hyper-parameters. And then, we replace the parameters part for describing the causal graph with the learned DAG. Afterwards, we just take the score without DAG constraint to optimize the causal mechanism approximation part (which may not be the same name in the other algorithms). Finally, the validation set is taken to evaluate the learned model. The final hyper-parameters used on the real dataset in our paper is as follows:

Table 4: The hyper-parameters used on real data.

Para	ρ_{init}	β	λ_{ℓ_1}
Value	0.008	2	0.3

A.3.3 Model Parameters

The CGL part in each local model is parameterized by a $d \times d$ matrix named $\boldsymbol{U}$ and the Gumbel-Sigmoid approach is leveraged for approximating the binary form. Each entry in $\boldsymbol{U}$ is initialized as 0. The temperature τ is set to 0.2 for all settings. Then, for the causal mechanism approximation part, we use 4 dense layers with 16 variables in each hidden layer. All weights in the Network are initialized using the Xavier uniform initialization.

A.3.4 Training Parameters

Our AS-FCD and DS-FCD reach this point and are implemented with the following hyper-parameters. We take Adam (Kingma & Ba, 2015) with learning rate 3×10^{-2} and all the observational data $\mathcal{D}^{c_k}$ on each client are used for computing the gradient. And the detailed parameters used in Algorithms 1 and 2 are listed in Table 5. Notice that as illustrated in MCSL (Ng et al., 2019), the performance of the algorithm is affected by the initial value of ρ_{init} and the choice of β. Since a small initial of ρ_{init} and β would result in a rather long training time. As said in (Kaiser & Sipos, 2021), MLE plays an important role in the early stage of training and highly affects the final results. Therefore, carefully picking a proper combination of ρ_{init} and β will lead to a better result. In our method, we tune these two parameters via the same scale of experiment with seeds $1 \sim 10$. For each variable scale and training type, the parameters are adjusted once and are applied to all other experiments with the same variable scale. We find the combinations of the following parameters in

Table 5: The hyper-parameters used on simulated data in this paper.

Para	α_{init}	h_{tol}	it_{max}	it_{inner}	it_{fl}	γ	ρ_{max}	λ_{ℓ_1}
Value	0	1×10^{-10}	25	1000	200	0.25	1×10^{14}	0.01

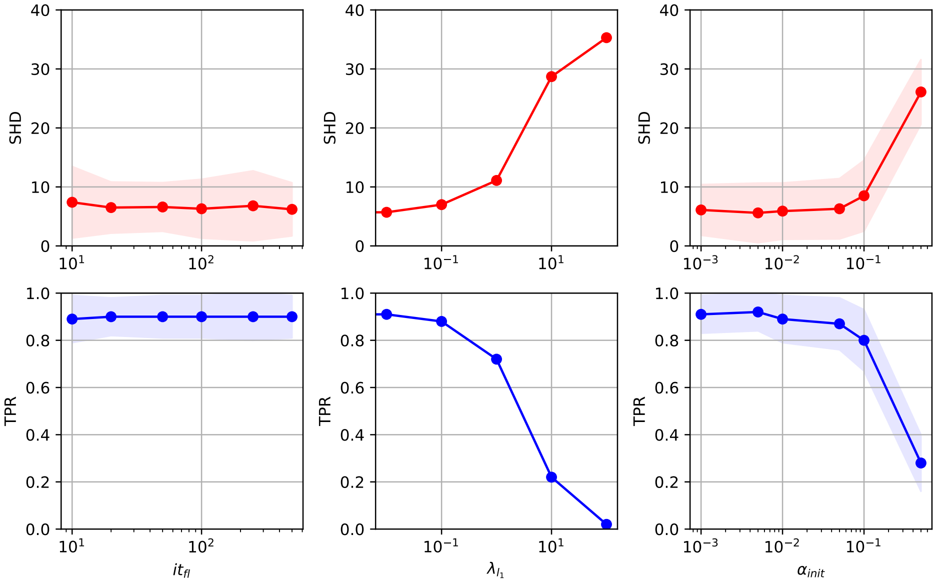


Figure 3: The sensitivity analysis of hyper-parameters

Table 6 work well in our method. Our method also adopts a ℓ_1 sparsity term on $g_\tau(\boldsymbol{U})$, where the sparsity coefficient λ_{ℓ_1} is chosen as 0.01 for all settings.

Table 6: The combinations of ρ_{init} and β on simulated data in our method.

	10 nodes		20 nodes		40 nodes	
	ρ_{init}	β	ρ_{init}	β	ρ_{init}	β
AS-FCD	6×10^{-3}	10	1×10^{-5}	20	1×10^{-11}	120
DS-FCD	6×10^{-3}	10	6×10^{-5}	20	1×10^{-11}	120

A.3.5 Sensitivity Analysis of Hyper-parameters

Here, we show the sensitivity analysis of it_{fl}, α_{init}, and λ_{l_1}. From the experimental results in Figure 3, we find that our method is relatively robust to it_{fl}. That is to say, the it_{fl} can be reduced to alleviate the pressure of communication costs while the performance can be well kept. λ_{l_1} is the coefficient of l_1 sparsity, which will affect the final results. Because we have no sparsity information of the underlying causal graph, we set $\lambda_{l_1} = 0.01$ in all settings. When dealing with real data, we recommend the audiences adjust this parameter by using our parameter-tuning method provided in the Section A.3.2. The results of α_{init} are exactly as expected. As discussed before, our method tries to maximize the likelihood term of the total loss in the early stages, which is important to find the final ground-truth DAG. If setting a relatively large α_{init}, the early learning stages would be affected. Therefore, we recommend directly taking α_{init} as 0 in all settings.

A.4 More discussions on the experimental results

Here, we give the detailed discussions on the experimental results in the paper. First of all, PC and GES can only reach the CPDAG (or MEC) at most, which shares the same Skelton with the ground-truth DAG. When we evaluate SHD, we just ignore the direction of undirected edges learned by PC and GES. That is to say, these two methods can get SHD if they can identify the CPDAG. Therefore, the final results are not caused by unfair comparison. For PC, the independence test is leveraged to decode the (conditional) independence from the data distribution. Therefore, the accuracy would be affected by (1) the amount of the observations and (2) the effectiveness of ”the non-parametric kernel

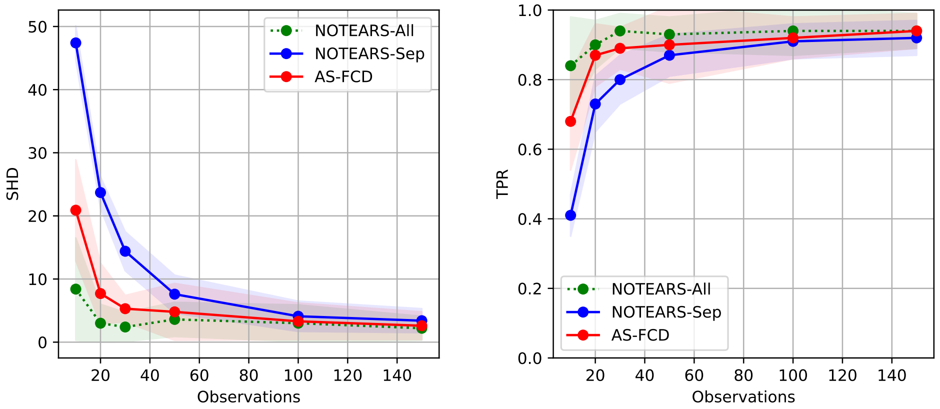


Figure 4: Comparisons with NOTEARS on linear data (IID).

independence test" method. GES leverages greedy search with BIC score. However, the likelihood part of BIC in GES is Linear Gaussian, which is unsuitable for data generated by the Non-linear model. NOTEARS is a linear model but the causal mechanisms are non-linear. The reason will be the unfitness between data and model. Therefore, the comparisons with GES and NOTEARS on linear IID data are implemented in the next section. DAG-GNN is also a non-linear model. However, the non-linear assumption of DAG-GNN is not the same as the data generation model assumed in our paper. The second reason comes from its "mechanisms approximation" modules are compulsory to share some parameters. Both NOTEARS-MLP and MCSL have their own advantages. Please refer to Tables 14 and 15, you will find that NOTEARS-MLP performs better when the non-linear functions are MIM and MLP while MCSL works better on GP and GP-add models.

A.5 Supplementary Experimental details

Results on Linear model As aforementioned for the IID case, the BIC score of GES takes the Gaussian likelihood and NOTEARS is a linear model. Therefore, for fair comparison, here, we also provide the linear version of our method. Since linear data are parameterized with an adjacency matrix, we can directly take the adjacency matrix as our model instead of a CGL part and a CMA part. During training, the matrix are communicated and averaged by the server to coordinate the joint learning procedures. All experiments are implemented with 50 observations on each client.

Table 7: Results on the linear model (IID).

		IID-GP				Non-IID			
		ER2 with 10 nodes		ER2 with 20 nodes		ER2 with 10 nodes		ER2 with 12 nodes	
		SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑
All	GES	11.0 ± 7.8	0.70 ± 0.21	8.5 ± 6.1	0.73 ± 0.17	25.0 ± 24.8	0.78 ± 0.20	33.2 ± 17.6	0.69 ± 0.16
	NOTEARS	2.2 ± 3.0	0.90 ± 0.13	1.8 ± 1.9	0.89 ± 0.10	3.6 ± 2.7	0.93 ± 0.05	10.2 ± 8.1	0.80 ± 0.15
Sep	GES	15.8 ± 4.5	0.50 ± 0.14	12.6 ± 3.8	0.54 ± 0.13	35.4 ± 10.3	0.60 ± 0.14	39.1 ± 6.6	0.52 ± 0.09
	NOTEARS	4.3 ± 1.9	0.85 ± 0.08	3.6 ± 2.1	0.83 ± 0.10	7.6 ± 3.0	0.87 ± 0.06	14.4 ± 6.5	0.76 ± 0.11
	AS-FCD	1.8 ± 1.7	0.91 ± 0.10	2.4 ± 2.4	0.86 ± 0.14	4.8 ± 4.5	0.90 ± 0.11	10.4 ± 7.0	0.79 ± 0.14

From Table 7, we find that our method can consistently show its advantage on the linear case. If you consider that why GES with all data still performs worse than our method, this results from the searching method and the credit should be of NOTEARS (our baseline method in the above experiments). Furthermore, we also added an ablation experiment that considers the effect of data number on the performance. The details are shown in Figure 4. We can see that our AS-FCD consistently performs well in the linear case.

Model Mis-specification Here, we add the experiments of model mis-specification, where a Post-Nonlinear model (PNL) is taken. The data of PNL model is generated according to $X_i =$

Table 8: Results on PNL with GP (IID).

		ER2 with 10 nodes		SF2 with 10 nodes	
		SHD ↓	TPR ↑	SHD ↓	TPR ↑
All data	PC	13.3 ± 3.5	0.43 ± 0.13	13.3 ± 4.4	0.50 ± 0.07
	NOTEARS	17.5 ± 2.1	0.00 ± 0.00	16.0 ± 0.0	0.00 ± 0.00
	MCSL	17.6 ± 2.4	0.01 ± 0.02	16.2 ± 0.6	0.01 ± 0.02
Sep data	PC	13.9 ± 3.0	0.34 ± 0.08	13.9 ± 1.4	0.34 ± 0.08
	NOTEARS	17.5 ± 2.1	0.00 ± 0.00	16.0 ± 0.0	0.00 ± 0.00
	MCSL	17.8 ± 2.1	0.01 ± 0.01	16.3 ± 0.5	0.01 ± 0.01
	AS-FCD	17.5 ± 2.1	0.00 ± 0.00	16.0 ± 0.0	0.00 ± 0.00
	DS-FCD	17.5 ± 2.1	0.00 ± 0.00	15.9 ± 0.3	0.01 ± 0.02

$\sigma(f_i(X_{Pa_i}) + Laplace(0, \epsilon_i))$, where function f_i is independently sampled from a Gaussian process with bandwidth one, $\epsilon_i \sim \mathcal{U}[0, 1]$ and $\sigma(\cdot)$ is the Sigmoid function. $\mathcal{U}[0, 1]$ means the uniform distribution from 0 to 1. The additional experimental results are shown in Table 8.

DS-FCD, carrying the ANM assumption of data, tries to maximize the likelihood of noise distribution. It is the same as NOTEARS and MCSL. Therefore, the model misspecification would hugely harm the performance of these methods.

Dense Graph Our method is also implemented on some denser graphs. Experimental results in Table 9 and Table 10.

Table 9: Results on nonlinear ANM with dense graphs (IID).

		ER4 with 10 nodes		SF4 with 10 nodes		ER4 with 20 nodes		SF4 with 20 nodes	
		SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑
All data	PC	27.3 ± 3.2	0.29 ± 0.07	18.9 ± 4.9	0.37 ± 0.16	68.2 ± 9.5	0.23 ± 0.06	60.2 ± 9.3	0.30 ± 0.08
	NOTEARS	34.3 ± 1.7	0.03 ± 0.02	22.7 ± 1.3	0.05 ± 0.05	71.8 ± 7.2	0.03 ± 0.01	62.8 ± 0.9	0.02 ± 0.01
	MCSL	15.5 ± 5.9	0.57 ± 0.15	4.5 ± 3.1	0.83 ± 0.11	33.8 ± 10.4	0.55 ± 0.11	19.8 ± 7.5	0.69 ± 0.11
Sep data	PC	31.5 ± 2.1	0.14 ± 0.03	20.4 ± 0.58	0.21 ± 0.03	68.7 ± 8.1	0.13 ± 0.03	60.9 ± 2.8	0.15 ± 0.02
	NOTEARS	34.3 ± 1.8	0.03 ± 0.01	22.7 ± 1.0	0.06 ± 0.04	70.1 ± 6.9	0.03 ± 0.01	62.3 ± 0.56	0.03 ± 0.01
	MCSL	15.8 ± 3.3	0.61 ± 0.09	8.3 ± 4.3	0.78 ± 0.11	49.3 ± 11.8	0.63 ± 0.10	39.7 ± 5.6	0.73 ± 0.07
	DS-FCD	16.9 ± 4.9	0.53 ± 0.12	5.4 ± 3.0	0.78 ± 0.12	35.4 ± 10.9	0.53 ± 0.11	20.7 ± 5.1	0.69 ± 0.08
	AS-FCD	17.4 ± 4.8	0.53 ± 0.12	5.5 ± 2.8	0.79 ± 0.11	40.7 ± 4.8	0.57 ± 0.10	24.1 ± 5.8	0.71 ± 0.09

Table 10: Results on nonlinear ANM with dense graphs (Non-IID).

		ER4 with 10 nodes		SF4 with 10 nodes		ER4 with 20 nodes		SF4 with 20 nodes	
		SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑
Sep data	PC	29.3 ± 1.3	0.23 ± 0.03	20.3 ± 2.1	0.31 ± 0.06	71.9 ± 8.1	0.19 ± 0.03	62.7 ± 2.8	0.22 ± 0.03
	NOTEARS	20.5 ± 2.6	0.45 ± 0.08	12.2 ± 2.9	0.54 ± 0.11	43.2 ± 7.0	0.49 ± 0.08	39.4 ± 6.8	0.47 ± 0.10
	MCSL	20.0 ± 3.2	0.52 ± 0.07	13.7 ± 2.2	0.65 ± 0.07	65.1 ± 7.7	0.33 ± 0.05	59.4 ± 5.3	0.31 ± 0.05
	DS-FCD	8.5 ± 3.7	0.84 ± 0.09	4.5 ± 2.0	0.93 ± 0.07	40.7 ± 14.5	0.74 ± 0.07	39.9 ± 10.8	0.68 ± 0.07

From the above experimental results, we can see that our method shows consistently better performance over other methods on the denser graph setting. For the IID case, both AS-DAG and DS-DAG obtain the nearly low SHD as MCSL trained on all data and far better than all methods trained on separated data. For the Non-IID case, our DS-FCD still shows the best performance. Compared to NOTEARS in 20 variables case, DS-FCD shows similar SHD results but much better TPR result. Therefore, how to reduce the false discovery rate of DS-FCD would be an interesting thing.

Voting method There is another interesting research line (Na & Yang, 2010), which also try to learn DAG from decentralized data. We add a DAG combination method proposed in (Na & Yang, 2010), which proposes to vote for each entry of the adjacency matrix to get the final DAG.

From the experimental results in Table 11, we can find that For PC and NOTEARS, the combining method seems to contribute little improvement. This is because the reported DAGs local clients are

Table 11: Comparison with the voting method.

		IID-GP				Non-IID			
		ER2 with 10 nodes		ER2 with 20 nodes		ER2 with 10 nodes		ER2 with 12 nodes	
		SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑
Sep data	PC	14.1 ± 2.4	0.31 ± 0.06	32.7 ± 6.5	0.28 ± 0.07	12.5 ± 2.7	0.45 ± 0.07	28.5 ± 6.3	0.44 ± 0.07
	NOTEARS	16.5 ± 2.0	0.06 ± 0.04	31.7 ± 6.0	0.11 ± 0.04	7.6 ± 2.6	0.60 ± 0.11	15.0 ± 3.1	0.62 ± 0.09
	MCSL	7.1 ± 3.2	0.83 ± 0.08	24.8 ± 5.5	0.88 ± 0.07	9.2 ± 1.8	0.72 ± 0.06	23.3 ± 5.8	0.56 ± 0.08
Voting	PC	13.3 ± 3.0	0.27 ± 0.11	29.7 ± 5.9	0.22 ± 0.05	11.4 ± 3.4	0.36 ± 0.13	25.5 ± 6.8	0.29 ± 0.13
	NOTEARS	15.6 ± 2.2	0.11 ± 0.06	32.6 ± 6.2	0.09 ± 0.05	7.8 ± 4.0	0.56 ± 0.20	18.4 ± 11.6	0.49 ± 0.30
	MCSL	8.0 ± 3.1	0.85 ± 0.16	18.1 ± 7.8	0.88 ± 0.06	6.9 ± 2.2	0.71 ± 0.13	10.1 ± 4.6	0.79 ± 0.09
	DS-FCD	2.4 ± 2.0	0.86 ± 0.12	6.2 ± 4.0	0.85 ± 0.10	1.9 ± 1.6	0.99 ± 0.02	6.2 ± 4.7	0.89 ± 0.09
	AS-FCD	1.8 ± 2.0	0.89 ± 0.12	5.0 ± 4.2	0.88 ± 0.11	Nan	Nan	Nan	Nan

too bad to get a good result. For MCSL, this combing method works really well for improving the performance. The reason is easy to be inferred from the results. For MCSL, DAGs reported by local clients are of bad SHDs but good TPR, which means that the False Discovery Rates (FDRs) are high. While the combing method can further reduce the FDRs and keep the TPRs still good. Then, SHD can be further reduced. Luckily, our DS-FCD still shows the best performances in all settings.

CAM Here, we add one more identifiable baseline named causal additive model (CAM) (Bühlmann et al., 2014), which also serves as a baseline in MCSL (Ng et al., 2019), GraNDAG (Lachapelle et al., 2020), and DAG-GAN (Yu et al., 2019).

Table 12: Comparisons with CAM on nonlinear ANM (IID-GP).

		ER2 with 10 nodes		SF2 with 10 nodes		ER2 with 20 nodes		SF2 with 20 nodes	
		SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑
All data	CAM	9.5 ± 2.9	0.87 ± 0.09	9.1 ± 3.1	0.84 ± 0.10	21.4 ± 4.7	0.77,± 0.08	26.6 ± 6.1	0.75 ± 0.07
Sep data	CAM	11.8 ± 2.6	0.40 ± 0.10	11.1 ± 1.5	0.38 ± 0.11	24.3 ± 5.8	0.40 ± 0.07	26.8 ± 2.0	0.36 ± 0.06
	DS-FCD	2.4 ± 2.0	0.86 ± 0.12	2.7 ± 2.2	0.86 ± 0.13	6.2 ± 4.0	0.85 ± 0.10	14.7 ± 7.0	0.80 ± 0.11
	AS-FCD	1.8 ± 2.0	0.89 ± 0.12	2.5 ± 2.7	0.85 ± 0.15	5.0 ± 4.2	0.88 ± 0.11	7.8 ± 5.5	0.80 ± 0.14

Table 13: Comparisons with CAM on nonlinear ANM (Non-IID).

		ER2 with 10 nodes		SF2 with 10 nodes		ER2 with 20 nodes		SF2 with 20 nodes	
		SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑
All data	CAM	31.9 ± 4.8	0.39 ± 0.15	31.8 ± 4.4	0.31 ± 0.17	104.6 ± 15.4	0.46,± 0.15	116.9 ± 13.8	0.35 ± 0.07
Sep data	CAM	18.0 ± 1.7	0.52 ± 0.04	17.8 ± 2.1	0.51 ± 0.3	47.5 ± 9.2	0.52 ± 0.04	53.0 ± 6.1	0.50 ± 0.03
	DS-FCD	1.9 ± 1.6	0.99 ± 0.02	2.6 ± 1.3	0.93 ± 0.07	6.2 ± 4.7	0.89 ± 0.09	11.5 ± 6.7	0.81 ± 0.14

From result in Table 12 and 13, we can see that our methods always show an advantage over CAM. CAM also assumes a non-linear additive noise model for data generation. However, CAM limits the non-linear function to be additive. In normal ANM, $X_i = f_i(X_{pa_i}) + \epsilon_i$ while CAM assumes $X_i = \sum_{j \in X(pa_i)} f_{i \leftarrow j}(X_j) + \epsilon_i$, which limits the capacity of its model. From the above experimental results, we can see that our methods show consistent advantages over CAM.

Table 14: Results on nonlinear ANM with different functions (IID, 10 nodes, ER2).

		GP		MIM		MLP		GP-add	
		SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑
All data	PC	15.3 ± 2.6	0.37 ± 0.10	11.0 ± 4.9	0.60 ± 0.16	11.8 ± 4.3	0.61 ± 0.14	14.0 ± 4.7	0.49 ± 0.16
	GES	13.0 ± 3.9	0.50 ± 0.18	9.6 ± 4.4	0.71 ± 0.17	15.8 ± 6.0	0.63 ± 0.14	14.4 ± 4.9	0.57 ± 0.17
	DAG-GNN	16.2 ± 2.1	0.07 ± 0.06	13.7 ± 2.4	0.26 ± 0.10	18.2 ± 3.3	0.36 ± 0.12	13.3 ± 2.3	0.24 ± 0.10
	NOTEARS	16.5 ± 2.0	0.05 ± 0.04	12.1 ± 3.2	0.34 ± 0.13	13.3 ± 3.4	0.35 ± 0.15	13.4 ± 2.2	0.23 ± 0.09
	N-S-MLP	8.1 ± 3.8	0.56 ± 0.17	1.6 ± 1.3	0.95 ± 0.06	*5.6 ± 1.3*	*0.81 ± 0.11*	6.8 ± 4.0	0.65 ± 0.16
	MCSL	*1.9 ± 1.5*	*0.90 ± 0.08*	*0.7 ± 1.2*	*0.97 ± 0.06*	12.7 ± 3.6	0.58 ± 0.24	*1.9 ± 1.7*	*0.91 ± 0.07*
Sep data	PC	14.1 ± 2.4	0.31 ± 0.06	11.1 ± 3.6	0.48 ± 0.14	13.2 ± 3.6	0.42 ± 0.09	13.5 ± 3.2	0.37 ± 0.12
	GES	12.7 ± 2.7	0.37 ± 0.09	10.6 ± 3.3	0.54 ± 0.12	14.6 ± 4.6	0.50 ± 0.13	12.0 ± 2.6	0.48 ± 0.08
	DAG-GNN	15.7 ± 2.3	0.11 ± 0.05	11.7 ± 3.3	0.37 ± 0.12	17.7 ± 3.6	0.39 ± 0.11	13.0 ± 2.0	0.26 ± 0.10
	NOTEARS	16.5 ± 2.0	0.06 ± 0.04	12.3 ± 3.0	0.33 ± 0.12	13.4 ± 3.4	0.35 ± 0.14	13.3 ± 2.3	0.24 ± 0.09
	N-S-MLP	8.5 ± 2.9	0.56 ± 0.13	2.8 ± 1.5	**0.93 ± 0.06**	**6.4 ± 1.3**	**0.81 ± 0.11**	7.4 ± 2.9	0.67 ± 0.13
	MCSL	7.1 ± 3.2	0.83 ± 0.08	4.4 ± 2.1	**0.91 ± 0.06**	13.4 ± 3.9	0.57 ± 0.21	6.5 ± 3.5	0.84 ± 0.07
	DS-FCD	**2.4 ± 2.0**	**0.86 ± 0.12**	**2.1 ± 1.4**	0.91 ± 0.07	11.1 ± 3.1	0.57 ± 0.20	**2.6 ± 1.6**	**0.87 ± 0.09**
	AS-FCD	**1.8 ± 2.0**	**0.89 ± 0.12**	**1.7 ± 1.6**	0.91 ± 0.08	**10.5 ± 3.5**	**0.59 ± 0.22**	**2.4 ± 1.6**	**0.87 ± 0.08**

Table 15: Results on nonlinear ANM with different functions (IID, 20 nodes, ER2).

		GP		MIM		MLP		GP-add	
		SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑
All data	PC	32.7 ± 9.4	0.48 ± 0.13	22.8 ± 5.8	0.60 ± 0.15	33.7 ± 12.3	0.50 ± 0.13	35.2 ± 8.0	0.50 ± 0.09
	GES	27.1 ± 8.5	0.56 ± 0.11	21.5 ± 6.1	0.78 ± 0.09	44.9 ± 12.5	0.65 ± 0.11	41.7 ± 11.6	0.66 ± 0.08
	DAG-GNN	32.5 ± 6.8	0.10 ± 0.08	26.7 ± 7.4	0.26 ± 0.13	32.1 ± 10.4	0.38 ± 0.08	27.2 ± 2.4	0.24 ± 0.08
	NOTEARS	31.8 ± 6.0	0.11 ± 0.04	25.6 ± 6.1	0.29 ± 0.08	25.3 ± 8.0	0.40 ± 0.09	25.6 ± 3.9	0.28 ± 0.06
	N-S-MLP	18.2 ± 4.5	0.52 ± 0.10	4.1 ± 2.0	0.95 ± 0.04	*8.0 ± 3.9*	*0.86 ± 0.07*	12.6 ± 2.2	0.70 ± 0.06
	MCSL	*4.6 ± 4.6*	*0.90 ± 0.13*	*1.7 ± 1.6*	*0.97 ± 0.04*	18.1 ± 6.6	0.72 ± 0.14	*3.1 ± 1.9*	*0.92 ± 0.05*
Sep data	PC	32.7 ± 6.5	0.28 ± 0.07	24.4 ± 5.6	0.46 ± 0.11	30.6 ± 8.0	0.41 ± 0.09	29.5 ± 5.6	0.42 ± 0.10
	GES	28.6 ± 5.5	0.34 ± 0.06	20.5 ± 3.7	0.61 ± 0.06	34.4 ± 11.3	0.52 ± 0.09	29.3 ± 5.5	0.51 ± 0.07
	DAG-GNN	31.7 ± 6.1	0.12 ± 0.04	26.8 ± 5.8	0.26 ± 0.06	34.1 ± 9.7	0.46 ± 0.07	26.5 ± 4.0	0.27 ± 0.05
	NOTEARS	31.7 ± 6.0	0.11 ± 0.04	25.7 ± 5.9	0.29 ± 0.07	25.4 ± 7.4	0.42 ± 0.07	25.6 ± 3.8	0.29 ± 0.06
	N-S-MLP	19.5 ± 4.7	0.52 ± 0.07	**6.5 ± 1.9**	**0.92 ± 0.03**	**16.1 ± 8.6**	**0.86 ± 0.07**	16.2 ± 3.3	0.70 ± 0.07
	MCSL	24.8 ± 5.5	**0.88 ± 0.07**	20.4 ± 3.8	0.91 ± 0.05	30.2 ± 5.1	0.67 ± 0.12	16.2 ± 5.3	**0.87 ± 0.05**
	DS-FCD	**6.2 ± 4.0**	0.85 ± 0.10	8.5 ± 2.8	**0.93 ± 0.05**	21.4 ± 7.9	0.71 ± 0.14	**8.1 ± 3.2**	0.85 ± 0.05
	AS-FCD	**5.0 ± 4.2**	**0.88 ± 0.11**	**3.3 ± 2.5**	0.92 ± 0.07	**20.1 ± 8.3**	**0.72 ± 0.14**	**5.6 ± 2.8**	**0.86 ± 0.06**

Table 16: Results on Non-IID setting with the different number of observations, (20nodes, ER2).

		n =100		n =300		n =600		n =900	
		SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑
All data	PC	55.5 ± 8.5	0.21 ± 0.06	57.3 ± 5.7	0.29 ± 0.07	60.4 ± 9.8	0.32 ± 0.11	62.4 ± 6.6	0.29 ± 0.10
	GES	82.8 ± 13.7	0.38 ± 0.12	96.4 ± 14.9	0.48 ± 0.08	102.9 ± 13.6	0.51 ± 0.08	106.3 ± 14.3	0.50 ± 0.11
	DAG-GNN	61.8 ± 14.7	0.39 ± 0.07	56.8 ± 9.7	0.37 ± 0.08	57.7 ± 12.0	0.38 ± 0.08	57.9 ± 12.1	0.32 ± 0.08
	NOTEARS	58.7 ± 12.8	0.41 ± 0.12	57.6 ± 10.2	0.44 ± 0.06	57.3 ± 12.9	0.43 ± 0.08	59.4 ± 10.3	0.39 ± 0.10
	N-S-MLP	111.2 ± 14.4	0.92 ± 0.10	101.0 ± 16.8	0.92 ± 0.05	100.8 ± 14.7	0.90 ± 0.10	97.6 ± 14.8	0.90 ± 0.07
	MCSL	49.0 ± 8.1	0.62 ± 0.06	54.0 ± 10.0	0.70 ± 0.10	53.8 ± 9.6	0.73 ± 0.10	57.6 ± 11.6	0.73 ± 0.08
Sep data	PC	31.2 ± 5.7	0.30 ± 0.05	29.0 ± 5.9	0.39 ± 0.06	28.5 ± 6.3	0.44 ± 0.07	27.9 ± 6.6	0.47 ± 0.08
	GES	35.1 ± 8.3	0.48 ± 0.10	31.6 ± 9.8	0.57 ± 0.08	30.0 ± 8.0	0.62 ± 0.06	30.5 ± 10.7	0.64 ± 0.07
	DAG-GNN	29.9 ± 7.2	0.66 ± 0.09	20.3 ± 5.0	0.67 ± 0.09	18.5 ± 4.9	0.67 ± 0.09	18.0 ± 5.2	0.66 ± 0.11
	NOTEARS	**16.3 ± 3.4**	0.61 ± 0.08	**15.5 ± 3.2**	0.60 ± 0.08	15.0 ± 3.1	0.62 ± 0.09	15.2 ± 2.9	0.61 ± 0.09
	N-S-MLP	68.0 ± 5.4	**0.80 ± 0.04**	22.6 ± 3.3	**0.79 ± 0.06**	**12.7 ± 2.6**	**0.80 ± 0.05**	**11.8 ± 2.8**	**0.80 ± 0.05**
	MCSL	32.8,± 5.4	0.49 ± 0.08	26.4 ± 5.5	0.53 ± 0.09	23.3 ± 5.8	0.56 ± 0.08	23.1 ± 6.5	0.56 ± 0.07
	DS-FCD	**11.6 ± 5.6**	**0.83 ± 0.11**	**7.1 ± 6.1**	**0.90 ± 0.12**	**6.2 ± 4.7**	**0.89 ± 0.09**	**6.0 ± 5.5**	**0.91 ± 0.11**

Table 17: Results on randomly selecting models-info of partial clients (Non-IID, 20nodes, ER2).

		IID				NON-IID			
		ER2 with 10 nodes		ER2 with 20 nodes		ER2 with 20 nodes		ER2 with 10 nodes	
		SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑	SHD ↓	TPR ↑
$\frac{r}{m}$	10%	3.8 ± 2.4	0.78 ± 0.14	8.6 ± 4.8	0.77 ± 0.13	3.8 ± 1.4	0.93 ± 0.05	8.5 ± 5.4	0.89 ± 0.07
	20%	3.2 ± 2.0	0.81 ± 0.12	6.7 ± 4.8	0.82 ± 0.13	2.5 ± 2.1	0.97 ± 0.04	8.2 ± 5.4	0.87 ± 0.09
	50%	2.9 ± 1.8	0.83 ± 0.11	5.8 ± 4.4	0.85 ± 0.12	1.8 ± 1.4	0.99 ± 0.02	6.3 ± 5.1	0.89 ± 0.10
	80%	2.7 ± 1.9	0.84 ± 0.12	6.0 ± 3.9	0.86 ± 0.10	1.8 ± 1.3	0.99 ± 0.02	5.9 ± 4.1	0.90 ± 0.08
	100%	2.4 ± 2.0	0.86 ± 0.12	6.2 ± 4.0	0.85 ± 0.10	1.9 ± 1.6	0.99 ± 0.02	6.2 ± 4.7	0.89 ± 0.09

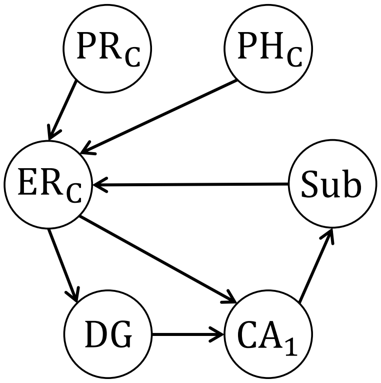


Figure 5: Anatomical causal-effect relationships of **fMRI Hippocampus** dataset

Table 18: Empirical results on **fMRI Hippocampus** dataset (Part 2).

	All data			Separate data			DS-FCD	AS-FCD
	GES	N-S-MLP	DAG-GNN	GES	N-S-MLP	DAG-GNN		
SHD ↓	8.0 ± 0.0	9.0 ± 0.0	5.4 ± 0.5	8.3 ± 1.2	11.3 ± 1.0	8.2 ± 1.9	**6.4 ± 0.9**	**5.0 ± 0.0**
NNZ	11.0 ± 0.0	12.0 ± 0.0	3.3 ± 0.8	8.5 ± 1.1	14.4 ± 0.8	5.7 ± 1.4	6.8 ± 0.6	5.0 ± 0.0
TPR ↑	0.43 ± 0.00	0.43 ± 0.00	0.23 ± 0.07	0.31 ± 0.17	**0.44 ± 0.10**	0.17 ± 0.18	0.27 ± 0.12	**0.29 ± 0.00**
FDR ↓	0.73 ± 0.00	0.75 ± 0.00	0.52 ± 0.09	0.75 ± 0.12	0.78 ± 0.05	0.80 ± 0.18	**0.72 ± 0.11**	**0.60 ± 0.00**

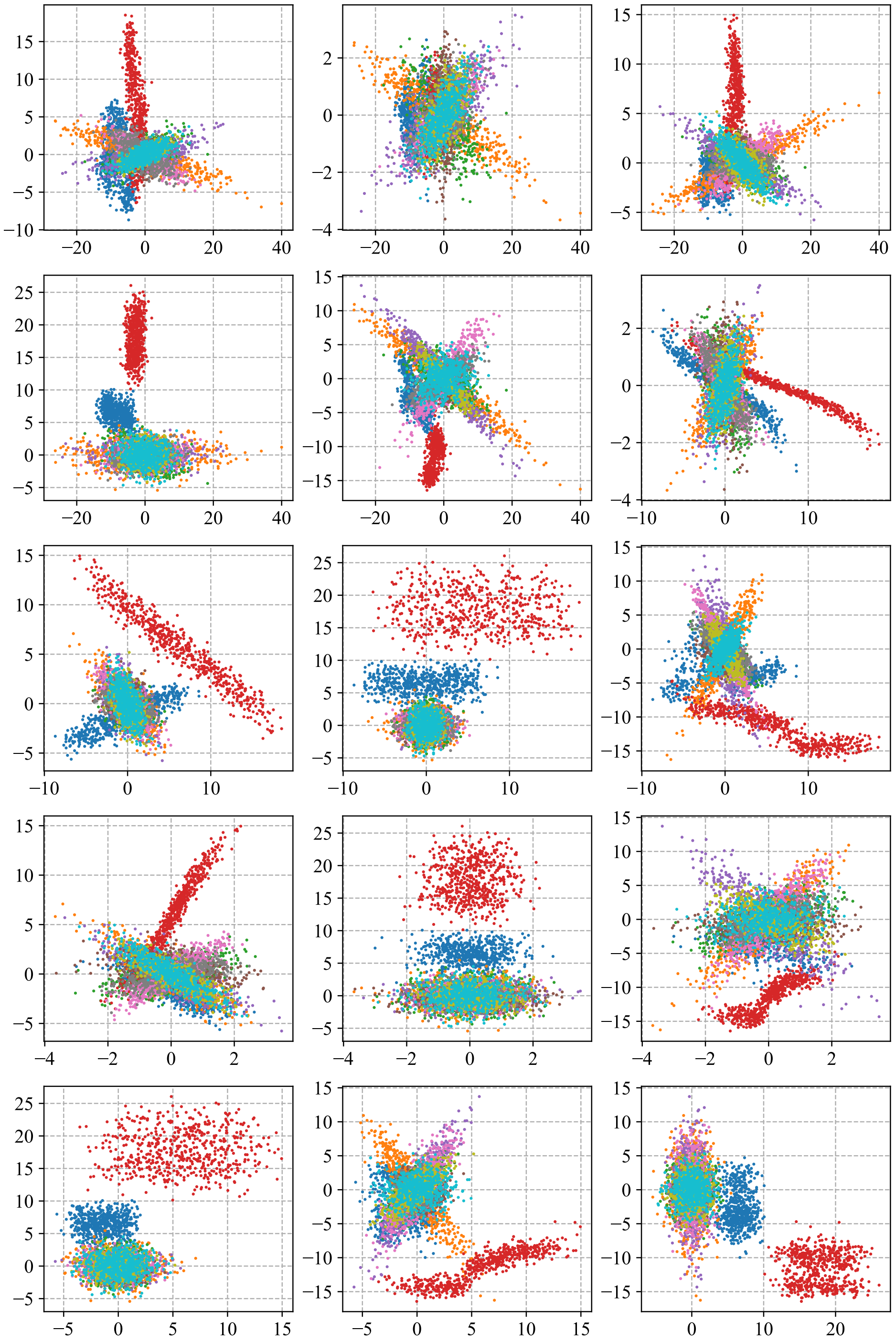


Figure 6: The visualization of simulated Non-IID data with 10 variables, where 6 variables are randomly selected and two of them are chosen for one subfigure.

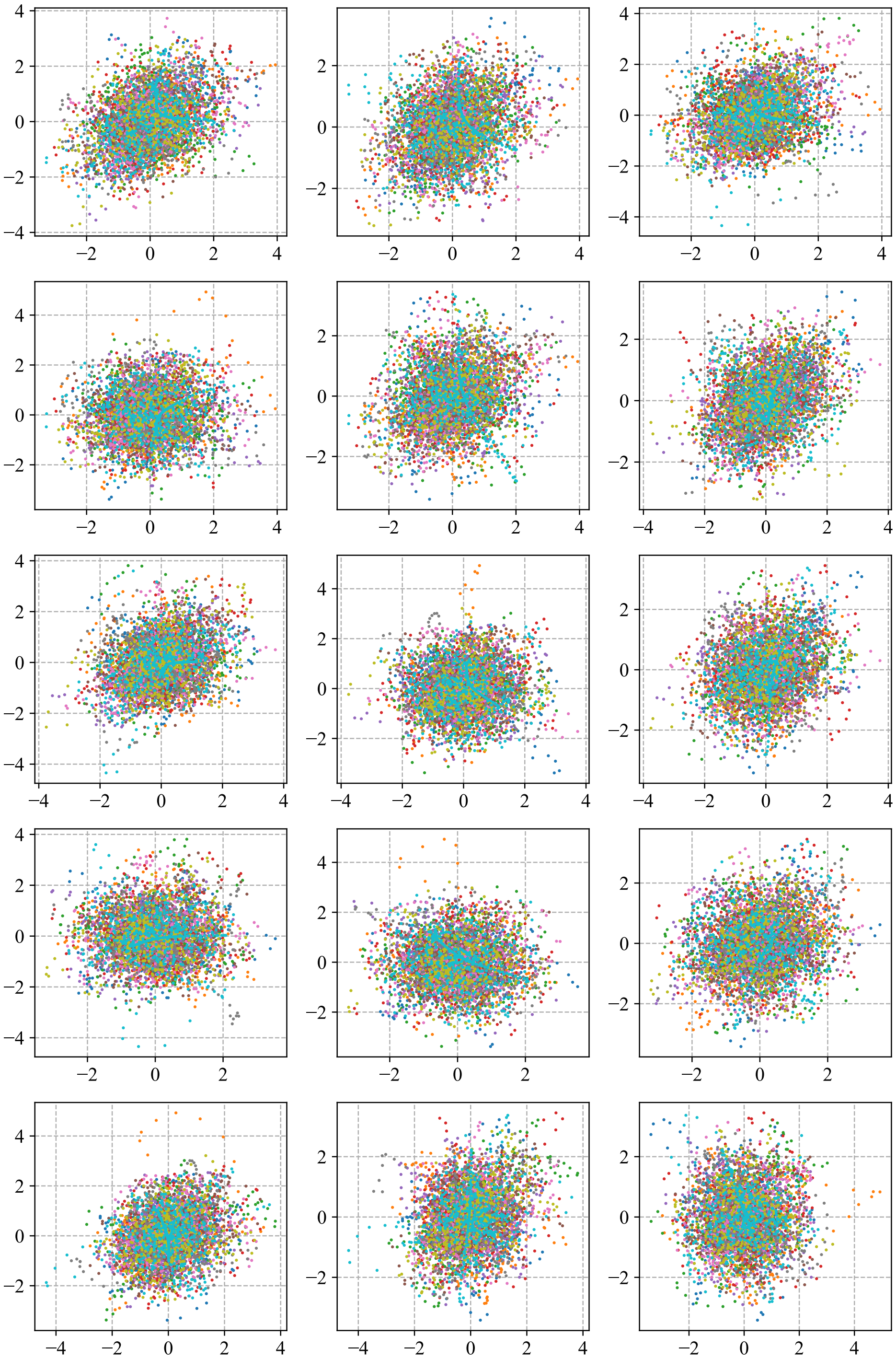


Figure 7: Normalized distribution of real data used in this paper.
the