FedECE: Federated Estimation of Causal Effect Based on Causal Graphical Modelling

Yongsheng Zhao, Kui Yu*, Guodu Xiang, Xianjie Guo, and Fuyuan Cao

***Abstract*— Causal effect estimation as a basic task in causal inference has been widely studied in past decades. In recent years, preserving data privacy has gained significant attention due to increasing incidents of data abuse and data leakage, however, most existing methods do not consider the problem of protecting data privacy when calculating causal effects. Thus in this paper, we propose a FedECE (Federated Estimation of Causal Effect) framework for causal effect estimation in a federated setting using causal graphical modelling, which comprises two modules: a federated causal structure learning (FedCSL) module and a federated causal effect (FedCE) module. We first instantiate the FedECE framework with a basic FedECE algorithm, called FedECE-B. FedECE-B presents a layer-wise cooperative optimization strategy to learn a global skeleton by the consideration of preserving data privacy. In addition, a distributed optimal consensus strategy for V-structure identification is proposed to orient edges in the learned global skeleton. To tackle the CPDAG problem in the learned causal structure, FedECE-B presents a progressively integrated multiset strategy for federated causal effect computation. To further improve the computational efficiency and accuracy of FedECE-B, we also propose the FedECE-L and FedECE-O algorithms. The extensive experiments validate the effectiveness of the proposed methods.**

***Impact Statement*—Calculating the causal effect of a treatment variable on an outcome variable helps us understand how the world works and how events are generated. However, with the increasing number of data abuse and data leakage incidents in recent years, accurately computing causal effects while protecting users' data privacy has become a significant challenge. Our proposed method provides a new way to combine federated learning with causal effect computation to solve this problem. By using the FedCSL module, we can learn a global causal structure in a federated setting, and then we employ the FedCE module to perform federated causal effect computation based on this structure. This approach allows us to estimate accurate causal effects without compromising users' data privacy. We validated our algorithms on multiple datasets and achieved significant improvements in both accuracy and stability compared to current state-of-the-art methods. This approach is expected to play an important role in causal inference in social sciences, biomedicine, and more.**

***Index Terms*—Causal effect estimation, Federated learning, Causal structure learning, Privacy-preserving data.**

I. Introduction

CAUSAL effect estimation is the quantitative calculating causal effect of a treatment variable on an outcome variable, which has been widely applied to many fields, such as medicine [1], economics [2], and government decision-making [3]. For example, estimating the causal effect of regulatory genes on a disease from genetic data can help researchers to develop new drugs or make effective treatment plans [4]. Calculating the causal effect of advertising placement on product sales from sales data can support salesmen to produce reasonable product marketing strategies [5].

The core challenge in causal effect estimation is how to identify confounding variables. Randomized controlled trials (RCTs) are the gold standard for causal inference since they can effectively address the problem of confounding variables. However, RCTs are often infeasible in most cases due to ethical concerns, time constraints and other issues. Accordingly, calculating causal effects from observational data has become the mainstream research paradigm [6]. Data-driven causal effect estimation models are mainly divided into two categories: potential outcome models (POM) [7] and structural causal models (SCM) [6]. Based on the two models, many causal effect estimation methods have been proposed, such as the doubly robust learning [8] and counterfactual regression [9] methods based on POMs, as well as the IDA [4] and IDP [10] algorithms based on SCMs.

In recent years, the issue of data privacy protection has brought widespread attention due to an increasing number of data abuse and data leakage incidents. For example, in 2021, a data leakage at Facebook results in the exposure of personal details of over 500 million users. Due to data privacy protection issues, datasets are often isolated in various organizations or groups, making it hard to directly aggregate or share those datasets. For example, for the consideration of preserving patient privacy, it is not easy to collect patients' electronic medical records from different hospitals for intelligent medical data analysis. Federated learning [11] adopts the learning paradigm of "data stays, computation moves" to protect data privacy by sharing model parameters among different clients/groups for joint computations on a server without sharing or aggregating original data stored in different clients/groups. For example, researchers in hospitals can use the federated learning paradigm to co-train learning models for disease diagnosis without touching patients' medical records stored in each hospital.

However, the majority of existing methods for causal effect estimation usually require either aggregating data in different platforms or sharing original datasets, and there are only a few causal effect estimation methods taking data privacy protection

This work was supported by the National Science and Technology Major Project of China (2021ZD0111801) and the National Natural Science Foundation of China (under Grants 62376087 and 62176082). Corresponding author: Kui Yu.

Yongsheng Zhao, Kui Yu, Guodu Xiang, and Xianjie Guo are with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China (e-mail: yszhao@mail.hfut.edu.cn, yukui@hfut.edu.cn, xgd600600@mail.hfut.edu.cn, and xianjieguo@mail.hfut.edu.cn).

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

into consideration currently based on the potential outcome model, such as the work [12, 13]. Causal effect calculation based on POMs requires knowing the potential treatment variables of an outcome variable in advance, but it is a hard issue on high-dimensional data and it is unable to identify accurate confounding variables in high-dimensional datasets. In the federated setting, the isolated datasets in various organizations/groups enhance the difficulty of the problems.

In contrast, SCMs provide accurate definitions of confounding variables through directed acyclic graph (DAG), enabling this type of causal effect calculation methods to accurately identify confounding variables from high-dimensional datasets, such as the well-established IDA algorithm, serving as the basic algorithm for existing causal effect estimation methods based on SCMs. However, IDA and its variations do not take data privacy protection concerns into consideration. Thus calculating causal effects based on causal graphical modelling (SCMs) in a federated setting remains unexplored so far. Thus in this paper, we propose FedECE (Federated Estimation of Causal Effect), a novel federated causal effect estimation framework based on causal graphical modelling, and the main contributions are summarized below.

- We propose the FedECE framework for causal effect estimation in a federated setting by integrating causal structure learning and causal effect computation in a whole.
- We first instantiate the FedECE framework to a basic method, FedECE-B. In the FedECE-B algorithm, we propose a layer-wise cooperative optimization strategy for learning a causal skeleton in a federated setting. To identify suitable separation sets for orienting edges in the learned skeleton, we design a distributed optimal consensus mechanism. To address the multiset problem caused by CPDAG in a federated setting, we design a progressively integrated multiset strategy.
- To improve the computational efficiency and accuracy of FedECE-B, we propose the local FedECE (FedECE-L) and optimal FedECE (FedECE-O) algorithms. FedECE-L optimizes computational efficiency, while FedECE-O improves estimation accuracy.
- We conducted extensive experiments using synthetic, benchmark, IHDP and real datasets to validate the effectiveness of FedECE-B, FedECE-L and FedECE-O.

II. Related Work

Over the past decade, a large number of data-driven causal effect estimation methods have emerged, and most of this work has been performed directly on accessible local data sources, which can be broadly categorized into two types: the potential outcome model proposed by Robin and the structural causal model proposed by Pearl. Methods based on the potential outcome model mainly aim at controlling confounders by ensuring comparability or homogeneity between the treatment variable and the outcome variable, such as covariate balanced propensity score (CBPS) [14], inverse probability of treatment effect weighting (IPTW) [15], and double machine learning (DML) [16]. However, it is difficult for these methods to give specific criteria for confounder identifications.

In contrast, methods based on the structural causal model can graphically define confounders using backdoor criteria [6] or generalized backdoor criteria [17]. For example, Pearl et al. [6] argue that causal effects can be uniquely identified and estimated from observational data if the DAG is known. Additionally, Maathuis et al. [4] propose that causal effects can be estimated from observational data using the IDA algorithm in the absence of the DAG. Henckel et al. [18] and Witte et al. [19] further improve the IDA algorithm by proposing an optimal-IDA algorithm that minimizes the asymptotic variance of the computed causal effects. However, these work was not proposed for the federated setting.

Federated learning enables collaborative learning of a shared prediction model while keeping all original data decentralized at their local groups [11]. Recent work has also focused on federated causal effect estimation, but those methods are based on the potential outcome model. Xiong et al. [20] propose a federated inverse probability weighted (IPW) estimation method for calculating the average treatment effect (ATE) and the average treatment effect on the treated (ATT) for the entire study population. Vo et al. [12] propose a Bayesian approach to model potential outcomes as random functions that follow Gaussian processes distribution, which estimates the posterior distributions of causal effects to understand the uncertainty of causal estimands. Han et al. [21] propose to estimate causal effects for target populations through adaptive and optimal weighting of the source populations, considering the risk of negative transfer when the source and target populations are heterogeneous. Vo et al. [13] propose a causal inference framework based on adaptive kernel methods for estimating heterogeneous causal effects. Han et al. [22] use a multiply-robust, privacy-preserving approach and transfer learning to handle covariate shifts and mismatches in federated studies, optimizing ensemble weights for efficient and robust causal inference. Recently several algorithms have been proposed to learn causal structures in a federated setting [23–30], but they are not able to compute causal effects.

The work in this paper aims to establish a federated causal effect estimation method based on causal graphical modelling in a federated setting while considering data privacy.

III. Notation and Definitions

In this section the relevant definitions are presented and the notations used are summarized in Table I. Let $X_i \perp X_j \mid \boldsymbol{C}$ denote that two variables X_i and X_j are independent conditioning on a variable set $\boldsymbol{C}$ and $X_i \not\perp X_j \mid \boldsymbol{C}$ denote that X_i and X_j are dependent conditioning on $\boldsymbol{C}$. $\boldsymbol{C}$ is called a conditioning set and the size of a conditioning set is represented as ℓ.

Definition 3.1 (Separation set [31]). If $X_i \perp X_j | \boldsymbol{C}$ holds, $\boldsymbol{C}$ is called a separation set of X_i and X_j, denoted as $\boldsymbol{SepSet}(X_i, X_j)$.

Definition 3.2 (V-structure [32]) An unshielded triple in a local skeleton such as $X_i - X_k - X_j$, where X_i and X_j are not directly connected, forms a V-structure, if X_i and X_j are conditionally independent and X_k is not in $\boldsymbol{SepSet}(X_i, X_j)$.

For example, in Fig. 1(A), since $W_1 \perp W_2 \mid X_1$ holds, $\boldsymbol{SepSet}(W_1, W_2) = X_1$. In Fig. 1(B), since $W_1 \perp W_2 \mid X_1$ holds, and W_3 is not in the separation set $\{X_1\}$, then $W_1 - W_3 - W_2$ forms a V-structure $W_1 \rightarrow W_3 \leftarrow W_2$.

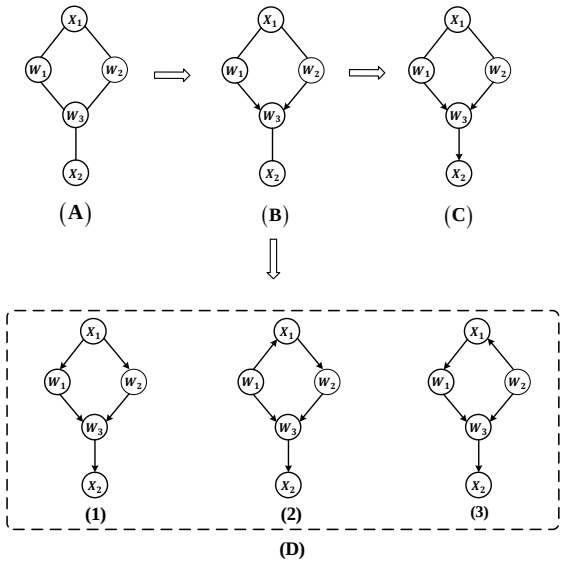

Fig. 1: Representations of a causal structure and its equivalence class. (A) A global skeleton containing five variables, (B) the PDAG of the skeleton in (A), (C) the CPDAG of the skeleton in (A), (D) Three DAGs belonging to the same equivalence class.

TABLE I: Summary of notations.

Notation	Meaning
$\mathcal{X} = \{X_1, ..., X_M\}$	The set of random variables in a dataset
X_i, X_j	A single variable in $\mathcal{X}$ $(i, j = 1, 2, ..., M)$
CN	Number of clients
$\boldsymbol{C}$	Conditioning set
$\boldsymbol{Z}$	A valid adjustment set
ℓ	The size of a separation set
$\mathcal{D}$	A direct acyclic graph over $\mathcal{X}$
$\mathcal{G}$	A completed partially directed acyclic graph over $\mathcal{X}$
$\mathcal{G}^C$	A completely undirected graph over $\mathcal{X}$
$\mathcal{G}^\ell$	A skeleton obtained at the ℓ-th layer
$\mathcal{G}^*$	A final skeleton obtained in Fedske
$X_i \perp X_j \mid \boldsymbol{C}$	X_i and X_j are conditionally independent given $\boldsymbol{C}$
$\boldsymbol{ne}(\mathcal{G}, X_i)$	The set of direct neighbors of X_i in $\mathcal{G}$
$\boldsymbol{pa}(\mathcal{G}, X_i)$	The set of parents of X_i in $\mathcal{G}$
$\boldsymbol{posspa}(\mathcal{G}, X_i)$	The set of possible parents of X_i in $\mathcal{G}$
$\boldsymbol{de}(\mathcal{G}, X_i)$	The set of descendants of X_i in $\mathcal{G}$
$\boldsymbol{possde}(\mathcal{G}, X_i)$	The set of possible descendants of X_i in $\mathcal{G}$
$\boldsymbol{cn}(\mathcal{G}, X_i, X_j)$	The set of variables excluding X_i on the correct causal path from X_i to X_j in $\mathcal{G}$
$\boldsymbol{posscn}(\mathcal{G}, X_i, X_j)$	The set of possible variables excluding X_i on the correct causal path from X_i to X_j in $\mathcal{G}$
$\boldsymbol{SepSet}(X_i, X_j)$	A separation set of X_i from X_j
$\langle X_i, X_k, X_j \rangle$	An unshielded triple $X_i - X_k - X_j$ in the skeleton
α	The significance level of the statistical test

Definition 3.3 (Meek's rules [33]). For a causal structure that has recognized all V-structures (i.e., partially directed acyclic graph (PDAG)), the following rules need to be satisfied when orienting the remaining undirected edges in it:

- Rule 1. No new V-structures are generated. Whenever there exists a directed edge $X_i \to X_k$ and X_i and X_j are not connected, orient $X_k - X_j$ as $X_k \to X_j$;
- Rule 2. Keep acyclicity. Whenever there exists a chain $X_i \to X_k \to X_j$, if X_i and X_j have edges connected, orient $X_i - X_j$ as $X_i \to X_j$;
- Rule 3. Whenever there are two chains $X_i - X_k \to X_j$ and $X_i - X_l \to X_j$ and X_k and X_l are not connected, orient $X_i - X_j$ as $X_i \to X_j$.

For example, applying the Meek's rules to the PDAG in Fig. 1(B), since it satisfies Rule 1, orient $W_3 - X_2$ as $W_3 \to X_2$, and the CPDAG in Fig. 1(C) is obtained.

Definition 3.4 (DAG, PDAG and CPDAG [34]). A directed graph contains only directed edges. A partially directed graph may contain both directed and undirected edges. A directed graph without directed cycles is a directed acyclic graph (DAG). A partially directed acyclic graph (PDAG) is a partially directed graph without directed cycles. If several DAGs have the same skeleton and V-structure, they belong to the same Markov equivalence class and are represented by a completely partially directed acyclic graph (CPDAG). The CPDAG is based on PDAG, which is formed by using the Meek's rules to orient all other-directed edges that can be oriented.

Fig. 1 illustrates the relationships between DAG, PDAG and CPDAG with a simple example. The difference between a PDAG and a CPDAG is that the CPDAG is formed by using the Meek's rules to orient all other-directed edges based on the PDAG. Fig. 1(B) represents a PDAG containing only one V-structure: $W_1 \to W_3 \leftarrow W_2$. Following Rule 1 of the Meek's rules, we infer a new directed edge: $W_3 \to X_2$, constructing a CPDAG, as shown in Fig. 1(C). In Fig. 1(D), three DAGs have the same independence relation: $W_1 \perp W_2 \mid X_1$, i.e., they belong to the same Markov equivalence class in Fig. 1(C).

Definition 3.5 (Forbidden set [18]). Given a CPDAG $\mathcal{G}$, a pair of variables (X_i, X_j) in $\mathcal{G}$, the forbidden set with respect to (X_i, X_j) and $\mathcal{G}$ is defined as $\boldsymbol{forb}(\mathcal{G}, X_i, X_j)$ = $\boldsymbol{possde}(\mathcal{G}, \boldsymbol{posscn}(\mathcal{G}, X_i, X_j)) \cup X_i$. Where $\boldsymbol{possde}(\mathcal{G}, \mathcal{X})$ denotes the set of possible descendants of each variable of $\mathcal{X}$ in $\mathcal{G}$, $\boldsymbol{posscn}(\mathcal{G}, X_i, X_j)$ denotes the set of possible variables excluding X_i on the correct causal path from X_i to X_j in $\mathcal{G}$.

In the example in Fig. 1(D)[(1)], since $\boldsymbol{posscn}(\mathcal{G}, X_1, X_2)$ = $\{W_1, W_2, W_3, X_2\}$, so $\boldsymbol{forb}(\mathcal{G}, X_1, X_2)$ = $\boldsymbol{possde}(\mathcal{G}, \boldsymbol{posscn}(\mathcal{G}, X_1, X_2))$ is $\{X_1, W_3, X_2\}$.

Definition 3.6 (Valid adjustment set [17]). Given a CPDAG $\mathcal{G}$ with a pair of variables (X_i, X_j) in $\mathcal{G}$ and a set of variables $\boldsymbol{Z}$ in $\mathcal{G}$. Then $\boldsymbol{Z}$ is a valid adjustment set relative to (X_i, X_j) in $\mathcal{G}$ if and only if the following three conditions hold: (i) every proper possibly causal path from X_i to X_j starts with a directed edge out of X_i, (ii) $\boldsymbol{Z} \cap \boldsymbol{forb}(\mathcal{G}, X_i, X_j) = \emptyset$, (iii) all non-causal paths from X_i to X_j are blocked by $\boldsymbol{Z}$.

Taking Fig. 1(D)[(1)] as an example, since $\boldsymbol{forb}(\mathcal{D}, X_1, X_2)$ = $\{X_1, W_3, X_2\}$ and the two paths from X_1 to X_2 are causal paths, thus the valid adjustment set can be $\emptyset$. For Fig. 1(D)[(2)], $\boldsymbol{forb}(\mathcal{D}, X_1, X_2)$ = $\{X_1, W_3, X_2\}$, and since W_1 blocks the non-causal path from X_1 to X_2 ($X_1 \leftarrow W_1 \to W_3 \to X_2$), the valid adjustment set is $\{W_1\}$. Similarly the valid adjustment set for Fig. 1(D)[(3)] is $\{W_2\}$.

Definition 3.7 (Casual effect [17]). The causal effect of variable X_i on variable X_j is a function of the probability distribution space from X_i to X_j, denoted as $f(X_j = x_j \mid do(X_i = x_i))$, abbreviated as $f(x_j \mid do(x_i))$. According to the backdoor criterion [6] and the generalized backdoor

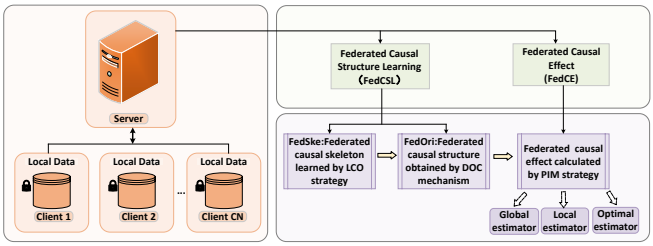

Fig. 2: The workflow of the FedECE framework.

criterion [17], $f(x_j \mid do(x_i))$ can be expressed as follows:

$$f(x_j \mid \mathrm{do}(x_i)) = \begin{cases} f(x_j \mid x_i) & \text{if } \boldsymbol{Z} = \emptyset \\ \int_z f(x_j \mid x_i, z) f(z) dz & \text{otherwise} \end{cases} \quad (1)$$

Assuming $\mathcal{X}$ is generated by a linear structural equation model (SEM) [6] with additive noise by $(\mathcal{G}, e)$, covariate adjustment allows researchers to estimate the causal effect by performing a multiple linear regression. In this setting, if $\boldsymbol{Z}$ is an valid adjustment set relative to some variables in $\mathcal{G}$ for X_i and X_j, the causal effect of X_i on X_j, denoted as θ_{ij}, is given by $f(X_j \mid x_i, z) = \gamma_0 + \gamma_i x_i + \gamma_z^T z$ (for some values $\gamma_0, \gamma_i \in \mathbb{R}$ and $\gamma_z \in \mathbb{R}^{|z|}$, where $|z|$ is the cardinality of the set z.). Here, γ_i represents the coefficient of X_i in the linear regression of X_j on X_i and $\boldsymbol{Z}$, and it is expressed as follows:

$$\theta_{ij} = \begin{cases} 0 & \text{if } X_j \in \boldsymbol{Z} \\ \gamma_i & \text{otherwise} \end{cases} \quad (2)$$

IV. Our Approach

In this section, we introduce the FedECE framework for causal effect estimation in a federated setting. This section is organized as follows: First, we present a new framework named as FedECE in Section IV-A. Next, in Section IV-B, we instantiate the framework with the FedECE-B algorithm. To address the computation and accuracy issues of FedECE-B, we further propose the FedECE-L and FedECE-O algorithms in Sections IV-C and IV-D, respectively.

In the federated learning paradigm, a group or an organization that generates datasets is often called a client, assuming that there are CN clients (labeled as Client 1, Client 2,..., Client CN) in the setup for this paper. A platform that aggregates model parameters sent by the clients is denoted as the server.

A. The FedECE framework

As shown in Fig. 2, our proposed FedECE framework consists of two main modules: a federated causal structure learning (FedCSL) module and a federated causal effect(FedCE) module.

Federated Causal Structure Learning. The FedCSL module learns a global causal structure in a federated setting. To achieve this without sharing the original datasets from each client, inspired by constraint-based causal structure algorithms, the module itself is divided into two submodules: a federated global skeleton (FedSke) learning submodule with the consideration of data privacy and a federated orientation (FedOri) submodule for orienting the edges in $\mathcal{G}^*$ learned in the FedSke submodule.

Federated Causal Effects Computation. Due to the structure learned by the constraint-based causal structure learning algorithm is usually a CPDAG with undirected edges, the existing causal effect estimation methods based on CPDAG often return a multiset of causal effect values. That is to say, a treatment variable may have multiple causal effects on the outcome variable. The FedCE module implements the calculation of multiset of federated causal effects of pairs of variables in a federated setting while protecting the data privacy of each client.

We instantiate the FedECE framework with three algorithms, FedECE-B, FedECE-L, and FedECE-O, as described in Section IV-B, Section IV-C, and Section IV-D respectively. These algorithms utilize the same federated causal structure learning method, with the key distinction lying in the federated causal effect computation component with three different estimators.

B. The FedECE-B algorithm

In this section, we propose a basic instantiation of the FedECE framework, called the FedECE-B algorithm (the pseudocode detail of the algorithm is given in the Supplementary Material). The details of the FedECE-B algorithm are discussed as follows.

1) The FedSke submodule: Given that current constraint-based structure learning algorithms are mostly designed for a single dataset, a simple strategy to apply this type of algorithms to federated learning is to learn a causal structure at each client and subsequently aggregate the learned structures at the server. However, this approach poses a potential challenge[1]: the varying qualities of data, such as small-sized samples, among different clients may result in learned structures with significantly different qualities. Directly aggregating these structures may not yield a satisfactory result.

Motivated by the layer-wise idea of constraint-based causal structure learning algorithms, we propose a federated global skeleton learning (FedSke) submodule with the consideration of data privacy. In the FedSke submodule, we design a layer-wise cooperative optimization (LCO) strategy which enables each client to share and update its skeleton parameters learned at each layer of the FedSke submodule at the server without sharing their original data. The LCO strategy learns the federated global skeleton process as shown in Fig. 3 and the basic steps of the strategy are as follows.

Step 1 (Skeleton initialization). When $\ell = 0$ (conditioning set size is 0), the initial skeleton on each client is a completely undirected graph $\mathcal{G}^C$.

Step 2 (Layer-wise iterative skeleton updates). At the ℓ-th layer (i.e., conditioning set of size ℓ), each client uses its local dataset to update the existence of edges between variables in $\mathcal{G}^{(\ell-1)}$ ($\mathcal{G}^{(\ell-1)}$ denotes the skeleton obtained at the $(\ell-1)$-th layer; when $\ell = 0$, $\mathcal{G}^{(\ell-1)} = \mathcal{G}^C$) by conducting the CI

[1]We simulate varying data quality across different clients in our experimental environment through setting different clients with different sample sizes. In addition, we assume that the sample size information on each client is private and cannot be accessed.

test independently. Specifically, given the value of ℓ, the set of neighbors of each variable under the current graph $\mathcal{G}^{(\ell-1)}$ is first obtained, and then for X_i and its each neighbor X_j, if there exists a subset $\boldsymbol{C} \subseteq \boldsymbol{ne}(\mathcal{G}^{(\ell-1)}, X_i) \setminus \{X_j\}$ (or $\boldsymbol{C} \subseteq \boldsymbol{ne}(\mathcal{G}^{C}, X_i) \setminus \{X_j\}$ if ℓ = 0) with $|\boldsymbol{C}| = \ell$, $X_i \perp X_j \mid \boldsymbol{C}$ holds (i.e., X_i and X_j are conditionally independent given $\boldsymbol{C}$), X_j is removed from the neighbor set of X_i. After all variables have been tested, $\mathcal{G}^{(\ell-1)}$ is updated with the new neighbor sets of those variables. The updated skeleton on Client cn is represented as $\mathcal{G}^{\ell}_{cn}$ ($cn \in \{1, 2, ..., CN\}$), and $\mathcal{G}^{\ell}_{cn}$ is a adjacency matrix of size $M \times M$. $\mathcal{G}^{\ell}_{cn}(i, j)$ = 0 means that there is no edge between X_i and X_j; otherwise, there is an edge. This is shown in Eq. (3):

$$\mathcal{G}^{\ell}_{cn}(i,j) = \begin{cases} 0 & \text{if } X_i \perp X_j \mid \boldsymbol{C} \text{ holds} \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

Then all clients send the skeletons $\mathcal{G}^{\ell}_1, \mathcal{G}^{\ell}_2, ..., \mathcal{G}^{\ell}_{CN}$ learned at the ℓ-th layer to the server simultaneously.

Step 3 (Cooperative optimization of the skeleton). The server aggregates all skeletons sent by the clients, and based on Eq. (4), we can obtain the total score matrix T^{ℓ} of $M \times M$ skeleton at the ℓ-th layer, which reflects the consensus degree among the clients regarding the connectivity of the variables in the skeleton.

$$T^{\ell}(i,j) = \sum_{cn=1}^{CN} \mathcal{G}^{\ell}_{cn}(i,j) \tag{4}$$

In Eq. (4), $T^{\ell}(i, j)$ is an element in the total score matrix T^{ℓ}, which represents the total number of clients that consider the existence of an undirected edge between X_i and X_j. For example, if there are 10 clients, $T^{\ell}(i, j)$ = 5 indicates that 5 clients believe there is an undirected edge between X_i and X_j.

Next, based on the total score matrix T^{ℓ}, the final skeleton $\mathcal{G}^{\ell}$ for the ℓ-th layer is constructed, which is also an $M \times M$ adjacency matrix. In Eq. (5), when the number of clients that believe that there is an edge between X_i and X_j is less than the given threshold β, there is no edge between X_i and X_j in the consensus skeleton $\mathcal{G}^{\ell}$, i.e., $\mathcal{G}^{\ell}(i, j)$ = 0; otherwise, $\mathcal{G}^{\ell}(i, j)$ = 1 indicates the presence of an edge in the consensus skeleton.

$$\mathcal{G}^{\ell}(i,j) = \begin{cases} 0 & \text{if } T^{\ell}(i,j) < \beta \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

Step 4 (Obtain the federated global skeleton). Set ℓ to $\ell + 1$ and send the aggregated skeleton $\mathcal{G}^{\ell}$ back to the clients as the initial skeleton of the $(\ell + 1)$-th layer. Steps 2 and 3 are repeated for a new iteration, continuing until the value of ℓ is bigger than the maximum number of direct neighbors that a variable has in the ℓ-th skeletons learned by all clients. We record the final skeleton as $\mathcal{G}^{*}$.

In Fig. 3, we show a simple example of federated skeleton learning using the LCO strategy in the FedSke submodule. Consider a scenario with 30 clients. Initially, a completely undirected graph $\mathcal{G}^{C}$ with five variables is constructed at the central server. Subsequently, the server sends $\mathcal{G}^{C}$ to each client, where clients perform global updates to the skeleton

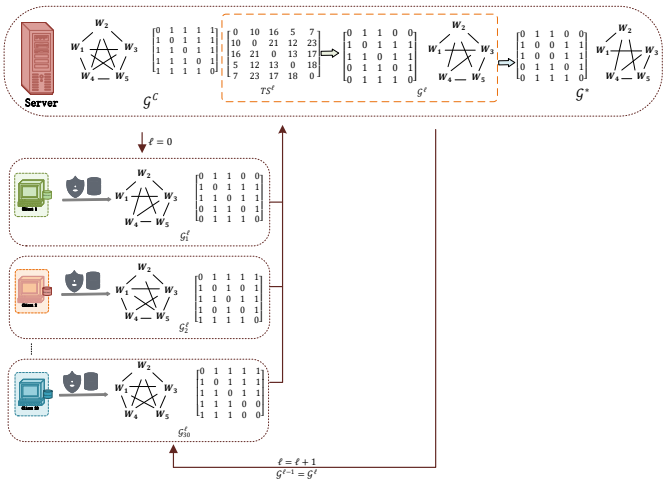

Fig. 3: An example of the LCO strategy.

using their local datasets. At $\ell = 0$, Client 1 takes W_1 as the target variable, and its neighbors are W_2, W_3, W_4 and W_5. Assuming the CI test results show that W_1 is conditionally independent of W_4 and W_5 respectively under the empty set condition, but the edges between W_1 and W_4, W_1 and W_5 are not deleted immediately until all conditional independence tests are completed. In this round, 4 independent relationships are found: $W_1 \perp W_4 \mid \emptyset$, $W_4 \perp W_1 \mid \emptyset$, $W_1 \perp W_5 \mid \emptyset$ and $W_5 \perp W_1 \mid \emptyset$. Thus, at Client 1, the edge W_1 and W_4 and the edge W_1 and W_5 are considered independent under the empty set condition and are removed. This produces an adjacency matrix $\mathcal{G}^{0}_1$ of size 5×5. Each client similarly updates the global skeleton, obtaining $\mathcal{G}^{0}_1$, $\mathcal{G}^{0}_2$, ... , $\mathcal{G}^{0}_{30}$. These 30 skeletons learned at the 0-th layer are sent to the server, which aggregates the adjacency matrices to form a score matrix T^{0} of size 5×5. For example, $T^{0}(2, 4) = T^{0}(4, 2) = 12$ indicates that 12 clients consider W_2 and W_4 not independent. If the threshold β is set to 9, and since $12 > 9$, it is concluded that W_2 and W_4 have an edge in the consensus skeleton $\mathcal{G}^{0}$. Similarly, for $T^{0}(1, 4) = T^{0}(4, 1) = 5$, as $5 < 9$, it is concluded that W_1 and W_4 do not have an edge in $\mathcal{G}^{0}$. Set $\mathcal{G}^{0}$ as the initial skeleton at the 1-th layer and send it to each client for the next round of updating. This process continues until $\ell = 3$ (i.e., the maximum number of direct neighbors in the learned 3-th layer skeletons across all clients is 3). At this point the value of ℓ is not smaller than the maximum number of direct neighbors in the ℓ-th layer skeleton, the entire skeleton learning phase is terminated, yielding the global skeleton $\mathcal{G}^{*}$.

2) The FedOri submodule: For any unshielded triple $\langle X_i, X_k, X_j \rangle$ in the $\mathcal{G}^{*}$, the triple can be identified as a V-structure $X_i \rightarrow X_k \leftarrow X_j$ if there exists a conditioning set (separation set) $\boldsymbol{C}$ in $\mathcal{G}^{*}$ such that $X_i \perp X_j \mid \boldsymbol{C}$ and $X_k \notin \boldsymbol{C}$ holds, to complete the orientation of the triple $\langle X_i, X_k, X_j \rangle$ in $\mathcal{G}^{*}$. Therefore, accurate separation set identification is key to identify V-structures. But there are two inconsistencies in learning separation set in the federated setting:

First, the problem of within-layer inconsistency. At the same ℓ-th layer, different clients may have different separation sets for the unshielded triple $\langle X_i, X_k, X_j \rangle$, due to the potential data

quality problem of clients. For example, the separation sets learned at the ℓ-th layer for the unshielded triple $\langle X_i, X_k, X_j\rangle$ on different clients are $\boldsymbol{C}_1^\ell, \boldsymbol{C}_2^\ell, ..., \boldsymbol{C}_{CN}^\ell$ ($\boldsymbol{C}_{CN}^\ell$ denotes the separation set learned at the ℓ-th layer on client CN), which exhibit the inconsistency problem.

Second, the problem of between-layer inconsistency. Since $\mathcal{G}^*$ learned by the LCO strategy is an aggregated skeleton, the separation set for unshielded triple $\langle X_i, X_k, X_j\rangle$ in $\mathcal{G}^*$ may be different from those obtained at the clients in the FedSke phase. Specifically, assuming that the skeleton updating stops when ℓ = 3. At this time, X_i and X_j are conditionally independent, and the separation set of the unshielded triple $\langle X_i, X_k, X_j\rangle$ is $\boldsymbol{C}^3$. But during the iterative updating of the skeleton at ℓ = 0, 1, 2, there are cases where a few clients consider that $X_i \perp X_j \mid \boldsymbol{C}^\ell$ holds ($\boldsymbol{C}^\ell$ obtained at the ℓ-th layer) , obtaining $\boldsymbol{C}^0$, $\boldsymbol{C}^1$ and $\boldsymbol{C}^2$. The four separation sets $\boldsymbol{C}^0$, $\boldsymbol{C}^1$, $\boldsymbol{C}^2$ and $\boldsymbol{C}^3$ are completely different, so how do we choose the most accurate separation set from them?

As each client does not share its raw data with the server, the FedOri submodule faces a challenge in directly computing the separation set for any non-adjacent variables in $\mathcal{G}^*$ at the server and it is crucial to tackle the aforementioned inconsistency problems. Thus, motivated by [26], we propose a federated orientation (FedOri) submodule for orient the edges in the learned global skeleton $\mathcal{G}^*$ in FedSke submodule. In this submodule, we design a distributed optimal consensus (DOC) mechanism to identify consistent separation sets in a federated setting for learning V-structures in $\mathcal{G}^*$ across clients. The main steps of the DOC mechanism are outlined as follows.

Step 1 (Unshielded triple identification). The server identifies all unshielded triples in $\mathcal{G}^*$ and then sends each triple and its direct neighbors to each client for learning the separation set. Here we use the unshielded triple $\langle X_i, X_k, X_j\rangle$ in $\mathcal{G}^*$ as an example, which represents the local skeleton $X_i - X_k - X_j$, where X_i and X_j are not directly adjacent to each other, but X_k is the direct neighbors of X_i and X_j. The server sends the triple $\langle X_i, X_k, X_j\rangle$ and $\boldsymbol{ne}(\mathcal{G}^*, X_i)$ to the clients.

Step 2 (Distributed separation set learning). If $X_i \perp X_j \mid \boldsymbol{C}$ holds using the local dataset on a client, the client sends the separation set $\boldsymbol{C}$ and the p-value (as shown in Eq. (6)) obtained from the CI test to the server.

$$p\text{−value} = \mathrm{P}(|z| \le z_{\frac{\alpha}{2}} \mid H_0) \tag{6}$$

where z denotes, under the null hypothesis H_0 of independence, for the linear Gaussian model, the transformation of partial correlation into value obeying a standard normal distribution through Fisher's Z transformation and $z_{\frac{\alpha}{2}}$ is the critical value of the standard normal distribution. The p-value denotes the probability of accepting the null hypothesis of conditional independence between two variables.

Fig. 4 uses a simple example to illustrate the distributed separation set identifying at each layer. Suppose there are 30 clients in Fig. 4. For an unshielded triple $\langle X_i, X_k, X_j\rangle$ in $\mathcal{G}^*$, at $\ell = 0$, assuming that the set $\{\emptyset\}$ corresponds to a maximum number of votes of 10 and it is unique, $C^0 = \emptyset$ is chosen as the separation set at $\ell = 0$. Similarly, at $\ell = 1$, assuming that the set $\{W_3\}$ with the maximum number of votes of 10 is chosen as the separation set for this layer, $C^1 = \{W_3\}$. At $\ell = 3$,

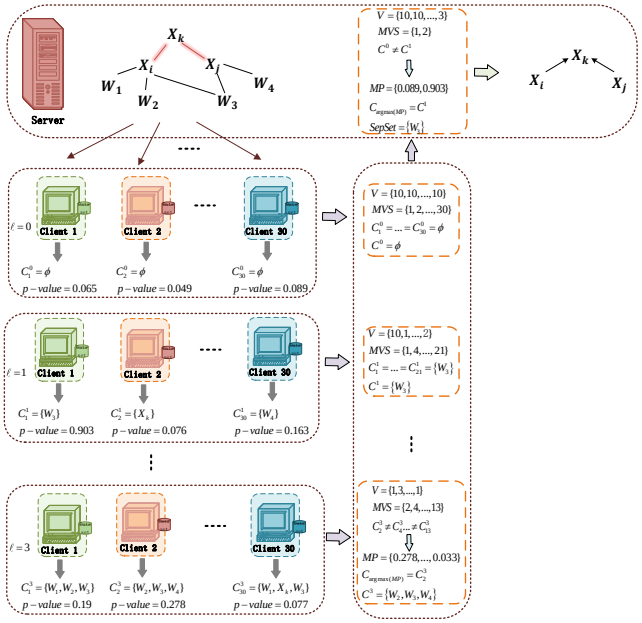

Fig. 4: An example of the DOC mechanism.

assuming that there are four sets with the maximum votes of 3, so we select the separation set according to p-values. We assume that C_2^3 has the highest p-value of 0.278 (note that the p-values obtained from the CI tests given in this example are all assumed), so $C^3 = \{W_2, W_3, W_4\}$ is selected as the separation set for this layer.

Step 3 (Accurate separation set identification). After Step 2, the server aggregates the separation sets and p-values sent by the clients to get $CS = \{C_1, C_2, ..., C_q\}$ and the corresponding p-value set $ps = \{p_1, p_2, ..., p_q\}$. We count each separation set in CS to obtain the count set $V = \{v_1, v_2, ..., v_q\}$ to identify the index corresponding to the separation set with the largest number of votes in CS, and we label this index set as MVS:

$$MVS = \arg\max(V) = \{s_1, ..., s_t\} \tag{7}$$

If all $s_T (T \in \{1, ..., t\})$ in $MVS = \{s_1, ..., s_t\}$ point to the same separation set in CS, i.e., $C_{s_1} = ... = C_{s_t}$ holds, then the separation set with the maximum number of votes is unique, and the set is the optimal separation set. However, if there are multiple different separation sets with the same maximum votes, further filtering is required. To obtain the optimal decision, we consider the set of p-value corresponding to the set $\{C_{s_1}, ..., C_{s_t}\}$ to get $MP = \{p_{s_1}, ..., p_{s_t}\}$. Then we select the separation set corresponding to the maximum value in MP as the final consistent separation set, as shown in Eq. (8):

$$\boldsymbol{SepSet} = \begin{cases} C_{s_i} & \text{if } \forall s_i, s_j \in MVS : C_{s_i} = C_{s_j} \\ C_{\arg\max(MP)} & \text{otherwise} \end{cases} \tag{8}$$

For example, in Fig. 3, for C^0, C^1, C^2 and C^3, although the number of votes for C^0 is the same as that for C^1, the

p-value of $\{W_3\}$ is bigger than that of C^0, so we finally select $\{W_3\}$ as the optimal consistent separation set for $\langle X_i, X_k, X_j\rangle$. That is, ***SepSet***$X_i, X_j) = \{W_3\}$. Then based on the obtained consistent separation set, the server determines whether $\langle X_i, X_k, X_j\rangle$ is a V-structure. For example, for the unshielded triple $\langle X_i, X_k, X_j\rangle$, if $X_k \notin$ ***SepSet***(X_i, X_j) holds, then we consider $\langle X_i, X_k, X_j\rangle$ as the V-structure and orient it as $X_i \rightarrow X_k \leftarrow X_j$.

Step 4 (Orient remaining edges). Specifically, for the PDAG that has recognized all V-structures, the Meek's rules is used to orient the remaining undirected edges as many as possible. Since this part of the processing does not involve the clients' data, in the federated setting, based on the learned V-structures, we complete the federated orientation propagation of the remaining undirected edges at the server, and finally learn CPDAG $\hat{\mathcal{G}}$.

The DOC mechanism improves upon the method proposed by Huang et al. [26] by introducing a dual-selection process. It first prioritizes the separation set with the most votes, enhancing the robustness of the selection process. If multiple sets receive the same number of votes, DOC then chooses the set with the largest p-value. This two-step approach increases the likelihood of selecting a separation set that accurately represents the data distribution, thereby improving the reliability and precision of causal discovery in a federated setting.

3) The FedCE module: In a federated setting, it is more challenging to compute causal effects between variables from CPDAG while protecting data privacy of each client. In addition, due to the existence of equivalence class, existing causal effect estimation methods based on a CPDAG often return a multiset of causal effects. In a federated setting, due to different data quality of different clients, there may be different multisets in each client for a pair of variables. So how to deal with those federated multisets is also a challenge.

To tackle these problems, we propose a federated causal effect (FedCE) module with a novel progressive integrated multiset (PIM) strategy for causal effect calculation in the federated setting (FedECE-B uses the global estimator to calculate the causal effect). The basic idea of the PIM strategy consists of the following steps.

Step 1 (Identification of valid DAGs). At the server, all the undirected edges in the learned CPDAG $\hat{\mathcal{G}}$ are oriented to obtain different valid DAGs = $\{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_K\}$, and then each valid DAG $\mathcal{D}_k$ $(k \in 1, 2, ..., K)$ is sent to each client in turn.

Step 2 (Distributed causal effect computation). Using the local dataset, for a pair of treatment variable X_i and outcome variable X_j in $\mathcal{D}_k$, each client computes the causal effect of X_i on X_j, $\theta_n^k = \gamma_{x_i|pa(\mathcal{D}_k,x_i)}$ on $\mathcal{D}_k$ using the backdoor criterion and sends the value to the server.

Step 3 (Causal effect aggregation based on DAG). The server averages the CN values of $\theta_1^k, \theta_2^k, ..., \theta_{CN}^k$ obtained from the clients. $\theta^k = \frac{1}{CN}\sum_{cn=1}^{CN}\theta_{cn}^k$. We consider θ^k to be the consistent causal effect of X_i on X_j corresponding to DAG $\mathcal{D}_k$ in the federated setting.

Step 4 (Causal effect aggregation based on DAGs). Set k to $k+1$ and the server continues to send valid DAG to the client for the calculation of relevant causal effects. Executing Step 2 and Step 3 until all DAGs are traversed. This allows us to obtain the final consistent causal effect multiset θ of X_i on X_j:

$$\begin{aligned}\theta &= \{\theta^1, \theta^2, \ldots, \theta^k\} \\ &= \{\bar{\gamma}_{x_i|pa(\mathcal{D}_1,x_i)}, \bar{\gamma}_{x_i|pa(\mathcal{D}_2,x_i)}, \cdots, \bar{\gamma}_{x_i|pa(\mathcal{D}_k,x_i)}\}\end{aligned} \tag{9}$$

where θ is the multiset of causal effects calculated based on $\hat{\mathcal{G}}$ in the federated setting.

Note that the multiset is similar to a set, with the only difference being that in a multiset, the multiplicity of elements is important. In a multiset, the multiplicity of an element indicates how many times that element is repeated in the multiset and different elements may have different multiplicities. For example, the sets {a,a} and {a} are equal, but the multisets {a,a} and {a} are not equal because the multiplicity is different.

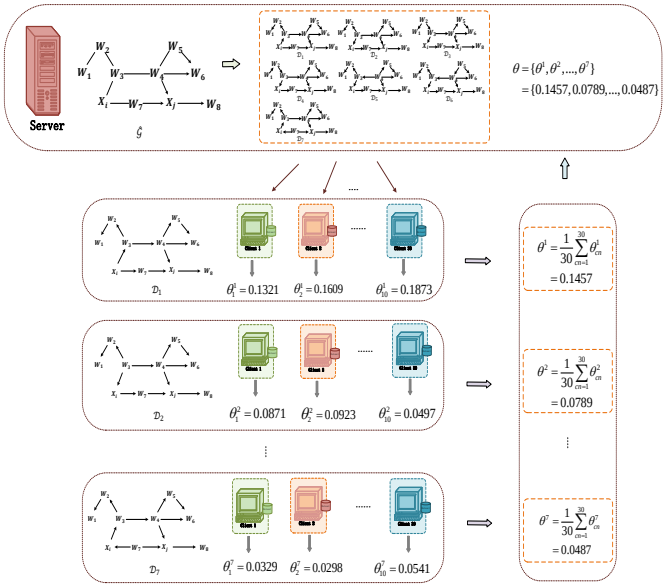

Fig. 5: An example of federated global causal effects estimation, where CPDAG $\hat{\mathcal{G}}$ is assumed to be the causal structure learned by FedCSL.

Fig. 5 shows a specific application of the global estimator of the federated causal effect. In this example, we assume that $\hat{\mathcal{G}}$ in Fig. 5 is the CPDAG learned by FedCSL. When calculating θ, the set of possible causal effects of X_i on X_j, for the CPDAG $\hat{\mathcal{G}}$ in Fig. 5, we first exhaust all valid DAGs in the equivalence class of $\hat{\mathcal{G}}$. Since $\hat{\mathcal{G}}$ contains 6 undirected edges $W_1 - W_2$, $W_2 - W_3$, $W_3 - W_4$, $W_4 - W_5$, $X_i - W_3$ and $X_i - W_7$, there are 64 possible ways to direct these edges, but some of these lead to graphs that are not in the equivalence class of $\hat{\mathcal{G}}$. For example, the configuration $W_1 \leftarrow W_2$, $W_2 \leftarrow W_3$, $W_3 \leftarrow W_4$, $W_4 \rightarrow W_5$, $X_i \rightarrow W_3$, and $X_i \leftarrow W_7$ is invalid because it creates a new V-structure $X_i \rightarrow W_3 \leftarrow W_4$ that is incompatible with the equivalence class represented by $\hat{\mathcal{G}}$. Excluding such these invalid configurations, seven DAGs remain that in the equivalence class of $\hat{\mathcal{G}}$ (see $\mathcal{D}_1$, ..., $\mathcal{D}_7$ in Fig. 5). Then for each k $(k = 1, \ldots, 7)$, we use Steps 21-30 of Algorithm 1 (presented in the Supplementary Material)

to compute the causal effect θ^k of X_i on X_j in a federated setting, yielding

$$\begin{aligned}\theta &= \{\theta^1, \theta^2, \ldots, \theta^7\} \\ &= \{\bar{\gamma}_{x_i|\emptyset}, \bar{\gamma}_{x_i|W_3}, \cdots, \bar{\gamma}_{x_i|W_7}\} \\ &= \{0.1457, 0.0789, \ldots, 0.0487\}\end{aligned} \tag{10}$$

where we assume that $\{0.1457, 0.0789, 0.0789, 0.0789, 0.0789, 0.0789, 0.0487\}$ is the specific causal effect value of X_i on X_j computed for the seven valid DAGs in a federated setting.

C. The FedECE-L algorithm

The FedECE-B performs well when the number of variables is relatively small (e.g., less than about 10). However, as the number of variables in an equivalence class increases, it quickly becomes infeasible to enumerate all valid DAGs of this equivalence class. Thus we further improve this method and give the FedECE-L algorithm. FedECE-L uses the local estimator to calculate the causal effect.

In FedECE-B, we find that the key to computing the causal effect of X_i on X_j relying on the parent set of X_i. Then instead of exhaustively enumerating all DAGs from an equivalence class, it is only necessary to locally identify a possible parent set $\boldsymbol{posspa}(X_i)$ of X_i in a CPDAG. Thus to tackle the computational problem of FedECE-B, we develop an efficient FedECE-L algorithm (the pseudocode of FedECE-L is given in Algorithm 2 in the Supplementary Material).

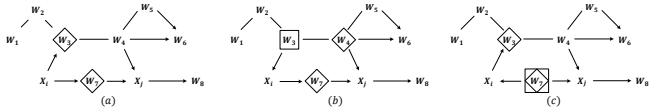

Fig. 6: The diamonds $\Diamond$ indicate $\boldsymbol{O}(\hat{\mathcal{G}}, X_i, X_j)$, while the boxes $\square$ indicate $\boldsymbol{posspa}(\hat{\mathcal{G}}, X_i)$.

The federated causal structure learning module of the algorithm is the same as that of FedECE-B algorithm. The difference between FedECE-B and FedECE-L is that FedECE-L replaces the exhaustively DAG enumerating step with a much simpler step which only checks if $\mathcal{G}_{\boldsymbol{posspa}\to X_s}$ is locally valid, meaning that $\mathcal{G}_{\boldsymbol{posspa}\to X_s}$ does not contain an additional V-structure with X_s as a collider. $\mathcal{G}_{\boldsymbol{posspa}\to X_s}$ denotes the graph that is obtained by changing all undirected edges $X_k - X_s$ with $X_k \in \boldsymbol{posspa}(\mathcal{G}, X_s)$ into directed edges $X_k \to X_s$, and all undirected edges $X_k - X_s$ with $X_k \in \boldsymbol{ne}(\mathcal{G}, X_s) \backslash \boldsymbol{posspa}(\mathcal{G}, X_s)$ into directed edges $X_k \leftarrow X_s$. In the example in Fig. 6(a) to (c), $\hat{\mathcal{G}}_{\boldsymbol{posspa}\to X_i}$ is locally valid for $\boldsymbol{posspa} = \emptyset$, $\{W_3\}$ and $\{W_7\}$ respectively, and it is not locally valid for $\boldsymbol{posspa} = \{W_3, W_7\}$.

For each valid parent set, we compute the causal effects of X_i on X_j on each client. We then compute the multiset of the causal effects of X_i on X_j by taking the average of the elements $\gamma_{x_i|\boldsymbol{posspa}}$ on all clients. For example, in Fig. 6(a) to (c), by computing the causal effects of X_i on X_j using three different parent sets, assuming that the final values of the causal effect of X_i on X_j are 0.1457, 0.0789, and 0.0487 respectively, the final multiset of causal effects are shown as follows.

$$\begin{aligned}\theta_L &= \{\bar{\gamma}_{x_i|\emptyset}, \bar{\gamma}_{x_i|W_3}, \bar{\gamma}_{x_i|W_7}\} \\ &= \{0.1457, 0.0789, 0.0487\}\end{aligned} \tag{11}$$

For example, in Eq. (10), we assume $\theta = \{0.1457, 0.0789, 0.0789, 0.0789, 0.0789, 0.0789, 0.0487\}$ while $\theta_L = \{0.1457, 0.0789, 0.0487\}$ in Eq. (11). Due to $pa(\mathcal{D}_1, x_i) = \emptyset$, $pa(\mathcal{D}_2, x_i) = pa(\mathcal{D}_3, x_i) = pa(\mathcal{D}_4, x_i) = pa(\mathcal{D}_5, x_i) = pa(\mathcal{D}_6, x_i) = \{W_3\}$ and $pa(\mathcal{D}_7, x_i) = \{W_7\}$, thus $\mathcal{D}_2, \ldots, \mathcal{D}_6$ correspond to the same value of causal effect, 0.0789, so the multiplicity (the multiplicity of a multiset is the number of times a particular element occurs in the multiset) of 0.0789 in θ is 5. Similarly, $\mathcal{D}_1$ and $\mathcal{D}_7$ correspond to causal effect values of 0.1457 and 0.0487, respectively, and the multiplicity in θ is 1. The causal effect values in the multiset θ_L are 0.1457, 0.0789 and 0.0487, and the multiplicities are all 1. Furthermore, the unique values in both θ and θ_L are 0.1457, 0.0789 and 0.0487. Therefore, it can be concluded that for the CPDAG in Fig. 5, the multisets of the causal effects of FedECE-B and FedECE-L have the same unique values, but their multiplicities may be different.

D. The FedECE-O algorithm

Learning a valid adjustment set is key to causal effect estimation. From FedECE-B and FedECE-L, the parent set of a treatment variable is crucial for the federated causal effect calculation because the parent set is the valid adjustment set. However, from the definition of valid adjustment set given in Section III, we can see that the parent set is not the unique valid adjustment set. And in theory, the causal effect of X_i on X_j computed with different valid adjustment sets should be consistent. However, in practice, due to data quality issues, for X_i and X_j, different valid adjustment sets may produce different causal effects of X_i on X_j. So if the multiple valid adjustment sets are available for X_i and X_j, which one should be used for causal effect estimation?

Recent studies [18] [19] propose the concept of O-set from the asymptotic variance perspective and prove that the causal effect results obtained by estimating using O-set as the valid adjustment set have the smallest asymptotic variance, when the underlying causal model conforms to the linear assumption. The graph criterion of O-set is shown in Eq. (12). Let X_i and X_j be disjoint variable sets in CPDAG, the O-set is defined as follows:

$$\boldsymbol{O}(X_i, X_j) = \boldsymbol{pa}(\boldsymbol{posscn}(X_i, X_j)) \setminus \boldsymbol{forb}(X_i, X_j) \tag{12}$$

To improve the accuracy problem of FedECE-B, we propose FedECE-O (the pseudocode of FedECE-O is given in Algorithm 3 in the Supplementary Material). In FedECE-O, we introduce the O-set to replace the parent set of FedECE-L as the valid adjustment set for the federated causal effect estimation. For comparison in Fig. 6, the boxs show the adjustment sets in FedECE-L, i.e., $\boldsymbol{posspa}(\hat{\mathcal{G}}, X_i)$, and the diamonds show the O-set in FedECE-O. In (a)-(c), $\boldsymbol{O}(\hat{\mathcal{G}}, X_i, X_j) = \{W_3, W_7\}$, $\{W_4, W_7\}$, $\{W_3, W_7\}$, and $\boldsymbol{posspa}(\hat{\mathcal{G}}, X_i) = \emptyset$, $\{W_3\}$, $\{W_7\}$, where the former improves efficiency.

After valid adjustment set identification, there is also a little difference between FedECE-L and FedECE-O when the client utilizes the local dataset for causal effect computation: while FedECE-L only checks if $X_j \notin \boldsymbol{posspa}(X_i)$ holds, FedECE-O further checks if $X_j \in \boldsymbol{possde}(X_i)$ holds. These two conditions ensure that the considered adjustment set is a valid adjustment set. Since the learning of the parent set in FedECE-L only needs to focus on the information about the neighborhoods of X_i, therefore we set $X_j \notin \boldsymbol{posspa}(X_i)$ as the local identification condition to complete the validity adjustment; while since the learning of the O-set in FedECE-O focuses on the entire causal path between X_i to X_j, we set $X_j \in \boldsymbol{possde}(X_i)$, a non-local identification condition, to complete the validity adjustment. Note that this additional checking approach in Algorithm 3 makes it less robust than Algorithm 2. This is because if the CPDAG is estimated well, the non-local identification tends to perform better than the local identification because it requires more information about the CPDAG; whereas if the CPDAG is not estimated accurately, the non-local identification is more likely than the local identification to use the incorrect information from the estimated CPDAG, which can amplify errors due to erroneous edges in the estimation of causal effects.

V. Experiments

In this section, we conduct experiments to verify the effectiveness of the FedECE framework. We first conduct simulation performance studies for the three algorithms proposed in this paper, FedECE-B, FedECE-L, and FedECE-O, in Section V-A, and then evaluate the performance of FedECE-L and FedECE-O with the baseline algorithms on synthetic dataset, BN dataset, IHDP dataset, and real dataset in Section V-B.

All experiments were conducted on a computer with Intel(R) Core(TM) i7-8700 3.20 GHZ CPU and processor 16-GB memory. All statistical independence tests are performed under the significance level $\alpha = 0.01$.

A. Evaluation of FedECE-B, FedECE-L and FedECE-O

We evaluate FedECE-B, FedECE-L and FedECE-O, in terms of multiplicity, runtime, and accuracy.

1) Multiplicity Analysis: To demonstrate the performance of FedECE-B, FedECE-L, and FedECE-O in terms of multiplicity, we generate a random causal weighted DAG $\mathcal{D}$ with the number of variables $p = 8$ and the number of expected neighbors $EN = 3$, and generate the corresponding CPDAG $\mathcal{G}$. We also randomly select a treatment variable X_i and an outcome variable X_j on $\mathcal{G}$ to compute the causal effects (Note that the DAG with its unique true causal effect is simulated for convenience only. Conceptually, we draw directly from the space of CPDAGs, which is why we consider the whole multiset of possible effects to be the truth). We generate 100 datasets of size 2000 from this DAG $\mathcal{D}$. The generative process employs a linear causal mechanism represented as follows:

$$\mathcal{X} \leftarrow B^T \mathcal{X} + e \tag{13}$$

where $e = (e_1, ..., e_M)$ is a continuous random vector of jointly independent error variables with mean 0, B is a weight matrix of $M \times M$, and $\leftarrow$ in Eq. (13) emphasizes a generative mechanism.

For each dataset, we use α = 0.01 on the number of clients CN = 5 to estimate the multiset of possible causal effects. We then aggregate these 100 estimates and construct a density plot. The true possible causal effects are 0 and 0.356, as shown by the vertical lines in Fig. 7, where the height of each line indicates the relative frequency of the given value in the multiset. Fig. 7 shows the smoothed density curves for the causal effect estimations returned by FedECE-B, FedECE-L, and FedECE-O. The higher a particular value is on the density curve, the more often that value occurs in the data. For example, in Fig. 7, the highest height of the vertical coordinate of FedECE-B at horizontal coordinate 0 indicates that FedECE-B achieves the highest multiplicity at an effect value of 0. Thus, a peak in the density plot indicates a high multiplicity of that particular value.

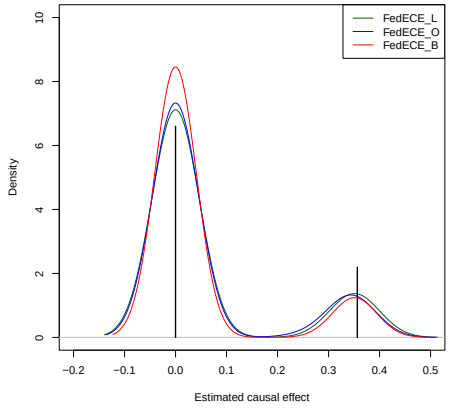

Fig. 7: Density plot in the style of Maathuis et al. [4]. Shown are density curves for estimated possible causal effects returned by FedECE-B(red), FedECE-L(green), and FedECE-O(blue). The true possible causal effects are 0 and 0.356 (vertical lines; heights indicate relative frequency of values)

We find that all three methods pick up the set of possible causal effects quite reliably. The FedECE-B algorithm captures the multiplicities better than the other two methods, while there is little difference in multiplicity performance between FedECE-L and FedECE-O. The reason that FedECE-B outperforms the others in multiplicity is mainly that the multiplicity of the element θ^k (i.e., the causal effect of X_i on X_j computed on a valid DAG $\mathcal{D}_k$) in the multiset θ corresponds to the number of DAGs in the equivalence class, whereas the multiplicity of the element θ^k in θ_L and θ_O corresponds to the number of valid adjustment sets in FedECE-L and FedECE-O. Since each adjustment set corresponds to at least one valid DAG in the equivalence class, this results in FedECE-L and FedECE-O losing the multiplicity of some of the values compared to FedECE-B. Specifically, in the example of Fig. 5, the valid parent set $\{W_3\}$ corresponds to valid DAG $\mathcal{D}_2, \ldots, \mathcal{D}_6$, and thus the multiplicity of $\gamma_{x_i|W_3}$ is 5 in FedECE-B, while in Fig. 6 the multiplicity of $\gamma_{x_i|W_3}$ is

1 in FedECE-L.

TABLE II: Mean runtime in seconds of FedECE-B, FedECE-L and FedECE-O over 10 replicates. A value NA means that at least one of the 10 replicates took more than 48 hours to compute, so that the computation was aborted.

	p = 5	p = 10	p = 15	p = 30	p = 50	p = 100
FedECE-B	0.4148	43.4882	NA	NA	NA	NA
FedECE-L	0.3182	0.4461	0.9109	7.4489	34.1472	43.8953
FedECE-O	0.3735	0.4762	0.9471	7.3960	34.4555	44.4461

2) Runtime Analysis: This section considers the runtime of the FedECE-B, FedECE-L and FedECE-O. Table II shows the mean runtime in seconds of FedECE-B, FedECE-L and FedECE-O over 10 replicates with EN = 2, S = 1000, CN = 5, and p denoting the number of variables. A value NA means that at least one of the 10 replicates took more than 48 hours, so that the computation was aborted. As we analyzed earlier, due to the relatively more time-consuming enumeration method of FedECE-B, causal structures with 15 variables or more cannot be handled reliably by FedECE-B, making FedECE-B suitable for datasets with a small number of variables. FedECE-L and FedECE-O perform similarly in terms of runtime.

3) Accuracy Analysis: In this section, we carry out a simulation study to compare the accuracy performance of FedECE-L and FedECE-O. Since the issue of excessive runtime for FedECE-B when dealing with more than 10 variables, in this section, we only compare FedECE-L and FedECE-O. The purpose of the experiments is to reflect the differences between the estimated causal effect values and the true causal effect values of the two algorithms for X_i on X_j based on the estimated CPDAG. Since the result obtained by the algorithms is a set of values, we use the Hausdorff distance to calculate the distance between the estimated set $\hat{\theta}$ and the true set θ^*:

$$H(\hat{\theta}, \theta^*) = \max\{\sup_{u\in\hat{\theta}} \inf_{v\in\theta^*} |u-v|, \sup_{v\in\theta^*} \inf_{u\in\hat{\theta}} |u-v|\} \quad (14)$$

Based on the Hausdorff distance, we primarily employ the mean squared error (MSE) to compare the estimated multisets of causal effects with the true multisets of causal effects:

$$\mathrm{MSE}(\{\hat{\theta}\}_{i=1}^N, \{\theta^*\}_{i=1}^N) = \frac{1}{N}\sum_{i=1}^N (H(\hat{\theta}, \theta^*))^2 \quad (15)$$

We consider the comparisons for different number of variables $p \in \{10, 20, 50, 100\}$ and the expected number of neighbors per variable EN = 2 and the sample dataset size S = 5000. In each setting, we generate 250 different synthetic graphs. For each graph, we randomly select a DAG $\mathcal{D}$ with p variables, generate the corresponding CPDAG $\mathcal{G}$, and randomly choose two variables (X_i, X_j) to compute causal effects. Then for each DAG, the following is repeated 5 times: a dataset with S observations is generated from a linear causal model based on $\mathcal{D}$ where the non-zero coefficients are randomly selected from a uniform distribution on [-1,-0.1] ∪ [0.1,1]. We then use FedECE-L and FedECE-O separately on the sample dataset to obtain two estimated multisets of causal effects. We calculate the MSE values between the estimated values and the true values, i.e., the squared error between the estimated effect set and the true effect set, by averaging over 5 replications. Specifically, we compute the relative MSE (RMSE = $\mathrm{MSE}_{FedECE-O}/\mathrm{MSE}_{FedECE-L}$) to compare the MSE value between the two algorithms. An RMSE of less than one indicates that FedECE-O is more accurate than FedECE-L in estimating causal effects.

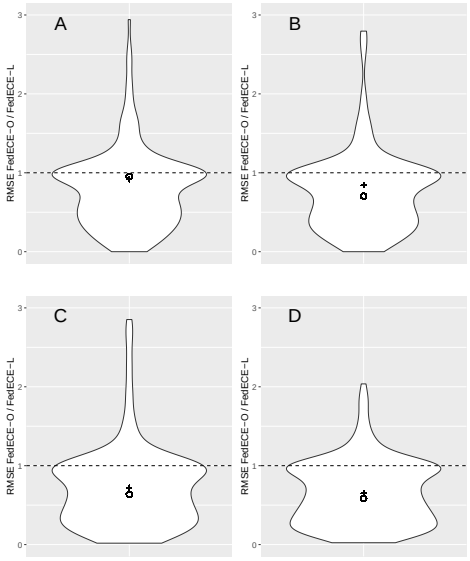

Fig. 8: Violin plot in the style of Witte et al. [19]. FedECE-L and FedECE-O are applied to the estimated CPDAG $\hat{\mathcal{G}}$. The dot marks the geometric means and the plus signs the medians.

TABLE III: Geometric means and medians of the RMSEs ($\mathrm{MSE}_{FedECE-O}/\mathrm{MSE}_{FedECE-L}$) over 250 repetitions with different numbers of variables (p).

	p = 10	p = 20	p = 50	p = 100
Geometic mean	0.9541	0.7049	0.6364	0.5854
Median	0.9151	0.8424	0.7216	0.6513

Fig. 8 shows the violin plots of RMSEs for $p \in$ {10,20,50,100} with 250 different DAGs. The accompanying geometric mean and median provide a concise summary of the central tendency within each RMSE distribution. Meanwhile, Table III summarizes the geometric means and medians for all scenarios with varying numbers of variables (i.e., the values of $\mathrm{MSE}_{FedECE-O}/\mathrm{MSE}_{FedECE-L}$). Just as discussed above, a geometric mean or a median of the RMSEs less than one indicates that FedECE-O is more accurate than FedECE-L in estimating the multiset of causal effects. In Table III, in most cases, FedECE-O outperforms FedECE-L in terms of the MSE metric.

B. Comparing with existing baselines

In this section, we compare the causal effect values estimated by FedECE-B, FedECE-L and FedECE-O. To evaluate the performance of FedECE-B, FedECE-L and FedECE-O with rivals, we use the Mean Absolute Error (MAE), a frequently used metric in causal effect estimation [35]. The MAE between the estimated $\hat{\theta}$ and ground-truth θ^* in this paper is calculated based on the Hausdorff distance.

$$\mathrm{MAE}(\{\hat{\theta}\}_{i=1}^{N}, \{\theta^*\}_{i=1}^{N}) = \frac{1}{N}\sum_{i=1}^{N} H(\hat{\theta}, \theta^*) \tag{16}$$

The comparison methods are as follows:

1) IDA-Avg: We first run the IDA algorithm at each client for obtaining CN causal effect multisets (CN is the number of clients), and then compute the averaging MAE values of all the computed causal effect multisets as the final result of IDA-Avg.
2) IDA-Best: we first run the IDA algorithm independently at each client to get CN causal effect multisets, and then select the causal effect multiset with the lowest MAE value as the final output.
3) FedECE$_{min}$: we run the FedCSL module to learn a unique federated causal structure, and then compute the causal effect multiset based on this structure. When aggregating CN effect values at the server, we choose the minimum value as the round-wise consistent causal effect value.
4) FedECE$_{max}$: we run the FedCSL module to learn a unique federated causal structure, and then compute the causal effect multiset based on this structure. When aggregating CN effect values at the server, we choose the maximum value as the round-wise consistent causal effect value.
5) FedECE$_{vote}$: we run the PCstable algorithm [36] (the clients' updating mechanism in our designed FedCSL module is similar to the PCstable algorithm) at each client to learn the CPDAG, and then we aggregate all learned CPDAGs at the server by the strategy that if the number of the learned CPDAGs containing a directed edge between two variables is more than a given threshold, this edge is kept in the final CPDAG. Finally, we calculate the multiset of causal effects using the PIM method based on the learned CPDAG.
6) FedCI: we run the FedCI [12] algorithm and compute the MAE between the estimated causal effect values and the true effect values.
7) CausalRFF: we run the CausalRFF [13] algorithm and then compute the MAE between the estimated causal effect values and the true effect values.

Since FedCI and CausalRFF require knowing potential causal relationships for a treatment variable and an outcome variable in a dataset in advance, they are not applicable to all the datasets in this experiment. Thus we only compare FedCI and CausalRFF with our two algorithms using the IHDP dataset.

As Section IV-B3 discussed, computing causal effects based on a causal structure requires the identification of valid adjustment set. FedECE-L uses a parent set as a valid adjustment set and FedECE-O uses an O-set as a valid adjustment set. Then for ease presentation, we use IDA-Avg^L, IDA-Best^L, FedECE$^L_{min}$, FedECE$^L_{max}$, and FedECE$^L_{vote}$ to denote that those methods use a parent set as a valid adjustment set, while IDA-Avg^O, IDA-Best^O, FedECE$^O_{min}$, FedECE$^O_{max}$ and Fe2dECE$^O_{vote}$ to denote that the methods use an O-set as a valid adjustment set.

1) Results on synthetic data: We conduct a simulation study by combing four synthetic datasets with $p \in \{10, 20, 50, 100\}$ and $CN \in \{5, 8, 10, 15\}$. We evaluate EedECE-B, FedECE-L and FedECE-O and the rivals on 250 synthetic graphs generated for each number of variables. As we generate data for each DAG 5 times, we have a total of N = 1250 runs. We use the MAE as the measure and an algorithm with a smaller MAE value indicates that the algorithm achieves more accurate causal effects.

TABLE IV: Comparison of the MAE values of FedECE-B, FedECE-L and FedECE-O with the ten baseline methods with CN = 5 and $p \in \{10, 20, 50, 100\}$. A value NA means that the calculation took more than 48 hours, so the calculation was terminated.

Method	p = 10	p = 20	p = 50	p = 100
IDA-Avg^L	0.0882	0.0855	0.0918	0.0955
IDA-Best^L	0.0716	0.0747	0.0671	0.0685
IDA-Avg^O	0.0954	0.0864	0.0927	0.1021
IDA-Best^O	0.0767	0.0762	0.0653	0.0684
FedECE$^L_{min}$	0.0788	0.0708	0.0719	0.0852
FedECE$^L_{max}$	0.0764	0.0704	0.0655	0.0852
FedECE$^L_{vote}$	0.0707	0.0666	0.0635	0.0684
FedECE$^O_{min}$	0.0760	0.0572	0.0546	0.0626
FedECE$^O_{max}$	0.0745	0.0583	0.0518	0.0614
FedECE$^O_{vote}$	0.0625	0.0662	0.0680	0.0587
FedECE-B	**0.0474**	NA	NA	NA
FedECE-L	**0.0474**	0.0382	0.0316	0.0337
FedECE-O	0.0540	**0.0357**	**0.0293**	**0.0296**

Table IV shows the MAE values of FedECE-B, FedECE-L and FedECE-O with the rivals using the synthetic datasets with $p \in \{10, 20, 50, 100\}$ and CN = 5 (Due to space constraints, please see the Supplementary Material for the complete MAE analysis for $CN \in \{5, 8, 10, 15\}$). Among them, the first part of the configuration in Table IV involves variants of existing causal effect estimation algorithms based on graphical structures adapted to single-source datasets (i.e., modifications of the IDA algorithm to handle federated settings). The second part comprises self-comparison algorithms designed according to the proposed algorithms in this paper to validate the effectiveness of the FedECE-B, FedECE-L, and FedECE-O algorithms.

The smaller value of MAE indicates the better performance of an algorithm. We can see that FedECE-L and FedECE-O outperform all rivals at $p \in \{20, 50, 100\}$ (FedECE-B cannot be handled reliably on datasets with more than 15 variables due to the use of the global estimator). FedECE-L outperforms

IDA-Avg^L and IDA-Best^L since IDA-Avg^L and IDA-Best^L do not exchange information between clients and the server during causal structure learning and causal effect estimation. This further validates the effectiveness of the LCO strategy, DOC mechanism, and PIM strategy proposed in this paper for causal effect calculations in a federated setting.

FedECE-L is superior to FedECE$^L_{min}$, FedECE$^L_{max}$ and FedECE$^L_{vote}$ on the MAE metric. FedECE$^L_{min}$ and FedECE$^L_{max}$ use the minimum and maximum value for causal effect aggregation at each round respectively, which can be sensitive to the outliers of causal effects. FedECE-L adopts the mean value which helps reduce the impact of the outliers of causal effects, and the experimental results also validate this conclusion. FedECE$^L_{vote}$ directly applies the PCstable algorithm to each client, aggregates the learned CPDAGs to get the final CPDAG, and then adopts the PIM method to calculate causal effects based on this CPDAG. Since the CPDAGs learned from FedECE$^L_{vote}$ are not correct, thus FedECE$^L_{vote}$ does not get an accurate adjustment set from the learned CPDAG, leading to unsatisfactory results.

The comparison of FedECE-O with those baseline algorithms is consistent with the results for FedECE-L. And we find that, when $p \in \{20, 50, 100\}$, the MAE value of FedECE-O is lower than that of FedECE-L, which further confirms our conclusion in Section V-A3 that FedECE-O can provide more accurate causal effects than FedECE-L as the number of variables increases as the estimated CPDAG is accurate.

2) Results on benchmark datasets: **Magic-niab:** Magic-niab is a linear Gaussian DAG with 44 variables and 66 edges. We select the variables $G257$ and MTL as the treatment and outcome variables, respectively. We sample the data 50 times, with each dataset containing 5000 samples, and then compare the performance of FedECE-B, FedECE-L and FedECE-O with the first five baseline algorithms with the number of clients $CN \in \{5, 8, 10, 15\}$. Table V shows that FedECE-O outperforms most of the baselines in terms of accuracy and demonstrates stable performance across all four client number scenarios.

According to Section IV-B3, causal effect calculation based on the O-set is more sensitive to the accuracy of the learned CPDAG, but if the learned CPDAG is accurate, the computed causal effect values become precise. FedECE-L outperforms its rivals significantly and demonstrates stable performance across different number of clients $CN \in \{5, 8, 10, 15\}$.

In addition, comparing FedECE-L with FedECE-O, we find that the causal effect values computed by FedECE-O are more accurate than FedECE-L at $CN \in \{5, 8, 15\}$.

Magic-irri: Magic-irri is a linear Gaussian DAG with 64 variables and 102 edges. We select the variables $G3964$ and $CHALK$ as the treatment and outcome variables, respectively. We sample the data 50 times, with each dataset containing 5000 samples, and then compare the performance of FedECE-B, FedECE-L and FedECE-O with the first five baseline algorithms with the number of clients $CN \in \{5, 8, 10, 15\}$. Table VI shows that FedECE-L and FedECE-O outperform almost all baseline algorithms on $CN \in \{5, 8, 10, 15\}$.

3) Results on the IHDP dataset: The Infant Health and Development Program (IHDP) dataset collected from a randomized study that investigated the causal effect of home visits by specialists on future cognitive test scores [37]. There are 25 pretreatment variables and 747 infants, including 139 treated (having home visits by specialists) and 608 controls. Two potential outcomes for the treatment (with or without a specialist visit) of each child are generated using the NPCI package [38]. The experimental settings for the FedCI [12] algorithm and the CausalRFF [13] algorithm are adopted from the source codes provided in the original paper. Considering the small sample size of this dataset, the experiments are conducted with the number of clients, $CN \in \{5, 8, 10\}$.

Table VII shows that both FedECE-L and FedECE-O are better than almost all rivals in most cases. The possible reason for FedECE-L and FedECE$^L_{vote}$ having consistent results is that both algorithms learn the same valid parent set of the

TABLE V: Comparison of the MAE values of FedECE-B, FedECE-L and FedECE-O with the ten classes of baseline methods on the magic-niab dataset with $CN \in \{5, 8, 10, 15\}$. A value NA means that the calculation took more than 48 hours, so the calculation was terminated.

Method	$CN = 5$	$CN = 8$	$CN = 10$	$CN = 15$
IDA-Avg^L	0.0300	0.0370	0.0414	0.0501
IDA-Best^L	0.0230	0.0290	0.0671	0.0302
IDA-Avg^O	0.0271	0.0295	0.0310	0.0348
IDA-Best^O	0.0247	0.0264	0.0271	0.0268
FedECE$^L_{min}$	0.0484	0.0695	0.0918	0.1339
FedECE$^L_{max}$	0.0310	0.0422	0.0512	0.0910
FedECE$^L_{vote}$	0.0265	0.0490	0.0400	0.0313
FedECE$^O_{min}$	0.0307	0.0393	0.0490	0.0928
FedECE$^O_{max}$	0.0388	0.0475	0.0579	0.0921
FedECE$^O_{vote}$	0.0278	0.0279	0.0246	0.0170
FedECE-B	NA	NA	NA	NA
FedECE-L	0.0180	0.0191	**0.0197**	0.0177
FedECE-O	**0.0171**	**0.0170**	0.0199	**0.0148**

TABLE VI: Comparison of the MAE values of FedECE-B, FedECE-L and FedECE-O with the ten baseline methods on the magic-irri dataset with $CN \in \{5, 8, 10, 15\}$. A value NA means that the calculation took more than 48 hours, so the calculation was terminated.

Method	$CN = 5$	$CN = 8$	$CN = 10$	$CN = 15$
IDA-Avg^L	0.3163	0.4099	0.4534	0.5457
IDA-Best^L	0.2462	0.3172	0.3494	0.3869
IDA-Avg^O	0.3873	0.7918	0.9862	1.3148
IDA-Best^O	0.2786	0.3529	0.4931	0.7736
FedECE$^L_{min}$	0.4366	0.7050	0.8682	1.3509
FedECE$^L_{max}$	0.5158	0.7754	0.8663	1.2049
FedECE$^L_{vote}$	0.2106	0.2771	0.2667	0.2754
FedECE$^O_{min}$	0.4067	0.6645	0.8731	1.4370
FedECE$^O_{max}$	0.4555	0.7283	0.8843	1.3383
FedECE$^O_{vote}$	0.1317	0.1220	0.2641	0.6980
FedECE-B	NA	NA	NA	NA
FedECE-L	0.1332	0.1231	**0.1237**	**0.1302**
FedECE-O	**0.1148**	**0.1089**	0.1613	0.4989

TABLE VII: Comparison of the MAE values of FedECE-B, FedECE-L and FedECE-O with the twelve baseline methods on the IHDP dataset with $CN \in \{5, 8, 10\}$. A value NA means that the calculation took more than 48 hours, so the calculation was terminated.

Method	$CN = 5$	$CN = 8$	$CN = 10$
IDA-Avg^L	0.3449	0.4369	0.5837
IDA-Best^L	0.1545	0.2910	0.3593
IDA-Avg^O	0.3504	0.5701	0.6588
IDA-Best^O	0.2879	0.3297	0.2648
FedECE$^L_{min}$	0.5851	1.1338	1.4964
FedECE$^L_{max}$	0.5032	0.5890	0.8966
FedECE$^L_{vote}$	**0.0921**	**0.1143**	0.1408
FedECE$^O_{min}$	0.5356	1.4300	1.2504
FedECE$^O_{max}$	0.3780	0.5738	0.8988
FedECE$^O_{vote}$	0.1935	0.1365	0.1085
FedCI	0.4249	0.3217	0.1774
CausalRFF	0.4275	0.4260	0.6701
FedECE-B	NA	NA	NA
FedECE-L	**0.0921**	**0.1143**	0.1408
FedECE-O	0.1042	0.1453	**0.0957**

treatment variable, leading to the same results.

FedECE-L and FedECE-O outperform FedCI and CausalRFF since FedCI and CausalRFF cannot identify accurate confounders (i.e., the valid adjustment set of the treatment variable) in a federated setting. In contrast, the CPDAG learned by the FedCSL module allows FedECE-L and FedECE-O to identify of accurate confounders, leading to accurate causal effects.

4) Results on real data: In this section, FedECE-B, FedECE-L and FedECE-O are applied on the synthetic gene expression dataset from the DREAM4 in *silico* challenge [39]. Here we use the 4-th $Size10$ dataset which is a small network containing 10 gene variables. Fig. 9 shows the true gene regulation network which is constructed based on the networks of living organism. In the experiment, we use only the observational data in each dataset including 61 observations and normalize the dataset.

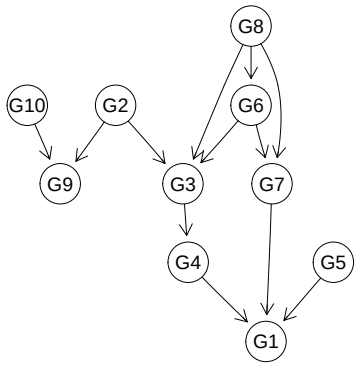

Fig. 9: The gene regulation network from the DREAM4 dataset.

Based on the DAG in Fig. 8, a treatment X_i and an outcome X_j are randomly selected. Considering that the size of the observational dataset is only 61, we conduct experiments for the first five methods with $CN \in \{3, 5\}$ on the DREAM4 dataset. The MAE results are shown in Table VIII. It can be observed that FedECE-O significantly outperforms the first five baseline algorithms, while FedECE-L, except for the number of clients $CN = 5$, is better than the other rivals. In addition, in the comparison between FedECE-L and FedECE-O, the latter achieves a favorable advantage. This may be due to the highest accuracy of the CPDAG learned by FedCSL, resulting in results in accurate O-set for FedECE-O. The possible reason why the MAE values of FedECE$^L_{vote}$ and FedECE$^O_{vote}$ do not change is that there may be missing edges or misdirected edges in the causal paths between X_i and X_j during the CPDAG learning process of FedECE$_{vote}$, which further demonstrates that the effectiveness of the skeleton learning and skeleton orientation strategy designed in the FedCSL module.

TABLE VIII: Comparison of the MAE values of FedECE-B, FedECE-L and FedECE-O with the ten baseline methods on the DREAM4 dataset with $CN \in \{3, 5\}$.

Method	$CN = 3$	$CN = 5$
IDA-Avg^L	0.3999	0.2370
IDA-Best^L	0.1479	0.1239
IDA-Avg^O	0.3999	0.3693
IDA-Best^O	0.3999	0.2509
FedECE$^L_{min}$	0.1751	0.1239
FedECE$^L_{max}$	0.2038	0.2543
FedECE$^L_{vote}$	0.5259	0.5259
FedECE$^O_{min}$	0.1814	0.2509
FedECE$^O_{max}$	0.1206	0.1272
FedECE$^O_{vote}$	0.3999	0.3999
FedECE-B	0.0440	0.0717
FedECE-L	0.0440	0.0717
FedECE-O	**0.0420**	**0.0554**

VI. Discussion

In this paper, we integrate the federated causal structure learning and the federated causal effect calculation as a unified framework, and propose three methods for a federated causal estimation. Although our methods achieve promising results, the following directions deserve further exploration.

More Complex Settings. FedECE currently handles causal effect estimation under a single intervention (i.e., the causal effect of a treatment variable on the outcome variable). Future research should explore algorithms capable of managing joint interventions [40] to capture more complex causal relationships. Additionally, in practical settings, as multiple privacy-preserving datasets may contain hidden variables [41] [42], it is worth extending FedECE to handle the complex case.

Reduced boundary range. The bounded causal effects computed by FedECE exhibit a potentially wide range, limiting their ability to precisely indicate the exact causal effect values. A key future improvement lies in narrowing the range of these bounded causal effects [43] [44]. This not

only reduces the uncertainty for causal relationship between variables but also enhances the precision of the boundary estimation of causal effects.

Computation cost. The FedCSL module in FedECE learns a global causal structure. However, this type of global structure learning approach is inefficient when dealing with high-dimensional data. A future development direction can integrate less computationally expensive causal structure learning algorithms [27] [28] [29] [30] into the FedECE framework.

Federated Optimization. The FedECE framework employs federated averaging for model aggregation. However, this straightforward averaging approach may not effectively capture the unique information from each client, potentially impacting the performance of the global model. An important direction for future is to explore the use of other federated optimization techniques such as federated matched averaging [45], to better integrate models from diverse clients.

REFERENCES

[1] S. Powers, J. Qian, K. Jung, A. Schuler, N. H. Shah, T. Hastie, and R. Tibshirani, "Some methods for heterogeneous treatment effect estimation in high dimensions," *Stat. Med.*, vol. 37, no. 11, pp. 1767–1787, 2018.

[2] A. Finkelstein and N. Hendren, "Welfare analysis meets causal inference," *J. Econ. Perspect.*, vol. 34, no. 4, pp. 146–167, 2020.

[3] D. P. Green and H. L. Kern, "Modeling heterogeneous treatment effects in survey experiments with bayesian additive regression trees," *Public Opin. Q.*, vol. 76, no. 3, pp. 491–511, 2012.

[4] M. H. Maathuis, M. Kalisch, and P. Bhlmann, "Estimating high-dimensional intervention effects from observational data," *Ann. Stat.*, vol. 37, no. 6A, pp. 3133–3164, 2009.

[5] M. A. Darrat, G. B. Wilcox, V. Funches, and M. A. Darrat, "Toward an understanding of causality between advertising and sales: New evidence from a multivariate cointegrated system," *J. Advert.*, vol. 45, no. 1, pp. 62–71, 2016.

[6] J. Pearl, *Causality*. Cambridge university press, 2009.

[7] D. B. Rubin, "Bayesian inference for causality: The importance of randomization," in *Proc. Soc. Stat. Sect. Am. Stat. Assoc.* American Statistical Association Alexandria, VA, 1975, p. 239.

[8] D. O. Scharfstein, A. Rotnitzky, and J. M. Robins, "Adjusting for nonignorable drop-out using semiparametric nonresponse models," *J. Am. Stat. Assoc.*, vol. 94, no. 448, pp. 1096–1120, 1999.

[9] U. Shalit, F. D. Johansson, and D. Sontag, "Estimating individual treatment effect: generalization bounds and algorithms," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2017, pp. 3076–3085.

[10] A. Jaber, J. Zhang, and E. Bareinboim, "Causal identification under markov equivalence: Completeness results," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2019, pp. 2981–2989.

[11] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[12] T. V. Vo, Y. Lee, T. N. Hoang, and T.-Y. Leong, "Bayesian federated estimation of causal effects from observational data," in *Proc. Conf. Uncertain. Artif. Intell.* PMLR, 2022, pp. 2024–2034.

[13] T. V. Vo, A. Bhattacharyya, Y. Lee, and T.-Y. Leong, "An adaptive kernel approach to federated learning of heterogeneous causal effects," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 24 459–24 473.

[14] K. Imai and M. Ratkovic, "Covariate balancing propensity score," *J. R. Stat. Soc. Ser. B*, vol. 76, no. 1, pp. 243–263, 2014.

[15] K. Hirano, G. W. Imbens, and G. Ridder, "Efficient estimation of average treatment effects using the estimated propensity score," *Econometrica*, vol. 71, no. 4, pp. 1161–1189, 2003.

[16] V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins, "Double/debiased machine learning for treatment and structural parameters," *Econometrics J.*, vol. 21, no. 1, pp. C1–C68, 2018.

[17] E. Perković, M. Kalisch, and M. H. Maathuis, "Interpreting and using cpdags with background knowledge," *arXiv preprint arXiv:1707.02171*, 2017.

[18] L. Henckel, E. Perković, and M. H. Maathuis, "Graphical criteria for efficient total effect estimation via adjustment in causal linear models," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 84, no. 2, pp. 579–599, 2022.

[19] J. Witte, L. Henckel, M. H. Maathuis, and V. Didelez, "On efficient adjustment in causal graphs," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 9956–10 000, 2020.

[20] R. Xiong, A. Koenecke, M. Powell, Z. Shen, J. T. Vogelstein, and S. Athey, "Federated causal inference in heterogeneous observational data," *Stat. Med.*, vol. 42, no. 24, pp. 4418–4439, 2023.

[21] L. Han, Y. Li, B. Niknam, and J. R. Zubizarreta, "Privacy-preserving, communication-efficient, and target-flexible hospital quality measurement," *Ann. Appl. Stat.*, vol. 18, no. 2, pp. 1337–1359, 2024.

[22] L. Han, Z. Shen, and J. Zubizarreta, "Multiply robust federated estimation of targeted average treatment effects," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2023, pp. 70 453–70 482.

[23] E. Gao, J. Chen, L. Shen, T. Liu, M. Gong, and H. Bondell, "Feddag: Federated dag structure learning," *Trans. Mach. Learn. Res.*, 2023.

[24] I. Ng and K. Zhang, "Towards federated bayesian network structure learning with continuous optimization," in *Proc. Int. Conf. Artif. Intell. Stat.* PMLR, 2022, pp. 8095–8111.

[25] Z. Wang, P. Ma, and S. Wang, "Towards practical federated causal structure learning," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*. Springer, 2023, pp. 351–367.

[26] J. Huang, X. Guo, K. Yu, F. Cao, and J. Liang, "Towards privacy-aware causal structure learning in federated setting," *IEEE Trans. Big Data*, vol. 9, no. 6, pp. 1525–1535, 2023.

[27] O. Mian, D. Kaltenpoth, M. Kamp, and J. Vreeken, "Nothing but regretsprivacy-preserving federated causal discovery," in *Proc. Int. Conf. Artif. Intell. Stat.* PMLR, 2023, pp. 8263–8278.

[28] D. Yang, X. He, J. Wang, G. Yu, C. Domeniconi, and J. Zhang, "Federated causality learning with explainable adaptive optimization," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 16 308–16 315.

[29] Q. Ye, A. A. Amini, and Q. Zhou, "Federated learning of generalized linear causal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 01, pp. 1–14, 2024.

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

[31] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

[32] G. Xiang, H. Wang, K. Yu, X. Guo, F. Cao, and Y. Song, "Bootstrap-based layer-wise refining for causal structure learning," *IEEE Trans. Artif. Intell.*, vol. 5, no. 6, pp. 2708–2722, 2024.

[33] C. Meek, "Causal inference and causal explanation with background knowledge," *arXiv preprint arXiv:1302.4972*, 2013.

[34] P. Spirtes, C. N. Glymour, and R. Scheines, *Causation, prediction, and search*. MIT press, 2000.

[35] L. Cheng, R. Guo, R. Moraffah, P. Sheth, K. S. Candan, and H. Liu, "Evaluation methods and measures for causal learning algorithms," *IEEE Trans. Artif. Intell.*, vol. 3, no. 6, pp. 924–943, 2022.

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

[37] J. L. Hill, "Bayesian nonparametric modeling for causal inference," *J. Comput. Graph. Stat.*, vol. 20, no. 1, pp. 217–240, 2011.

[38] V. Dorie, "Npci: Non-parametrics for causal inference, 2016," *URL https://github. com/vdorie/npci*, 2016.

[39] D. Marbach, T. Schaffter, C. Mattiussi, and D. Floreano, "Generating realistic in silico gene networks for performance assessment of reverse engineering methods," *J. Comput. Biol.*, vol. 16, no. 2, pp. 229–239, 2009.

[40] P. Nandy, M. H. Maathuis, and T. S. Richardson, "Estimating the effect of joint interventions from observational data in sparse high-dimensional settings," *Ann. Stat.*, vol. 45, no. 2, pp. 647–674, 2017.

[41] D. Cheng, J. Li, L. Liu, J. Liu, K. Yu, and T. D. Le, "Causal query in observational data with hidden variables," in *Proc. Eur. Conf. Artif. Intell.* IOS Press, 2020, pp. 2551–2558.

[42] D. Cheng, J. Li, L. Liu, K. Yu, T. D. Le, and J. Liu, "Toward unique and unbiased causal effect estimation from data with hidden variables," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 6108–6120, 2022.

[43] Y. Liu, Z. Fang, Y. He, and Z. Geng, "Collapsible ida: Collapsing parental sets for locally estimating possible causal effects," in *Proc. Conf. Uncertainty Artif. Intell.* PMLR, 2020, pp. 290–299.

[44] R. Guo and E. Perkovic, "Minimal enumeration of all possible total effects in a markov equivalence class," in *Proc. Int. Conf. Artif. Intell. Stat.* PMLR, 2021, pp. 2395–2403.

[45] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Proc. Int. Conf. Learn. Represent.*, 2020.