Federated Causal Structure Learning with Missing Data

Jiaqi Shi[a], Xiaoling Huang[b,c,*], Xianjie Guo[d], Kui Yu[c], Chengxiang Hu[b], Peng Zhou[a]

[a]*School of Computer Science and Technology, Anhui University, 230601, Hefei, China*
[b]*School of Computer and Information Engineering, Chuzhou University, 239000, Chuzhou, China*
[c]*School of Computer Science and Information Engineering, Hefei University of Technology, 230601, Hefei, China*
[d]*School of Computer Science, Nanjing University of Posts and Telecommunications, 210023, Chuzhou, China*

Abstract

Federated causal structure learning (CSL) is an emerging research direction that aims to discover causal relationships from decentralized data across multiple clients, while preserving data privacy. Existing federated CSL algorithms primarily focus on complete datasets and often overlook data-quality issues, such as missing data, which are common in real-world scenarios. Moreover, client diversity can destabilize federated CSL, and this challenge is further worsened by missing data. To address these issues, we propose FedImpCSL, a novel federated CSL method, for effectively handling missing data. Our approach consists of two key components: (1) a local-to-global missing data imputation strategy that reconstructs imputed and accurate datasets from missing samples, and (2) a dynamic client weighting and weighted aggregation strategy to address inter-client differences, enhancing the CSL accuracy without utilizing each client's original data. We demonstrate the effectiveness of FedImpCSL through comprehensive experiments on various types of datasets, showing its superior performance over existing federated CSL methods in handling missing data scenarios.

Keywords: Causal structure learning, Federated learning, Missing data imputation, Client diversity

1. Introduction

Causal structure learning (CSL) aims to discover the causal relationships among multiple variables or features [1, 2] by utilizing conditional independence (CI) tests, score functions, or continuous optimization strategies, exemplified by methods such as PC [3] GES [4], and NOTEARS [5]. CSL plays a fundamental role in causal inference, machine learning, and diverse scientific domains. Given the sensitivity of decentralized data, such as healthcare records, and the potential for privacy breaches upon centralization, federated CSL has emerged as a solution to facilitate distributed CSL while preserving client privacy.

In recent years, several federated CSL methods have emerged. NOTEARS-ADMM [6] represents the initial method grounded in a distributed algorithm. Subsequently, RFCD [7], FedC^{2}SL [8], FedPC [9] and FedCSL [10] have been successively proposed. Specifically, FedPC introduces a strategy for layerwise aggregation based on the PC [11] within the Federated Learning (FL) setting [12], demonstrating remarkable effectiveness across various data types. The latest research, FedCSL [10], proposes a federated CSL method for executing local-to-global strategy and determining the weights of various clients in situations where sample allocation is uneven, facilitating weighted aggregation. Through this strategy, FedCSL resolves key scalability and accuracy limitations in federated CSL.

Challenge 1: Missing data issue. Missing data is a common phenomenon in the real world, particularly in scenarios such as market research and medical records, where crucial information often remains incomplete owing to various factors. Existing methods for handling missing data [13], such as mean imputation and ICkNNI (Incomplete-Case k Nearest Neighbors Imputation) [14], focus on a single dataset. These methods are often used in CSL with

*Corresponding author.
Email addresses: e23301276@stu.ahu.edu.cn (Jiaqi Shi), hxl@chzu.edu.cn (Xiaoling Huang), xianjieguo@njupt.edu.cn (Xianjie Guo), yukui@hfut.edu.cn (Kui Yu), chengxiang@chzu.edu.cn (Chengxiang Hu), zhoupeng@ahu.edu.cn (Peng Zhou)

missing data. Specifically, ICkNNI is first invoked to perform data imputation, and then a CSL method is performed on the imputed dataset. However, due to the unique structural characteristics of FL, the direct application of existing imputation methods to federated CSL cannot yield the expected results. This is because these methods are designed based on a single dataset and their applicability to the FL setting is not considered. Furthermore, none of the federated CSL algorithms mention how to handle missing data in federated CSL, which further increases the difficulty of such a task. Therefore, developing effective solutions for the problem of missing data in federated CSL is a significant challenge that must be solved.

Challenge 2: Client diversity issue. In the FL setting, data quality, such as the sample size and missing data rate, varies across clients [15]. This necessitates considering the impact of client diversity on federated CSL. Currently, only FedCSL [10] considers sample size as a weight to quantify the diversity among clients and then aggregates client skeletons based on these weights on the server side. However, When handling missing data, especially when the missing data rates vary across clients, relying solely on the sample size as the weighting criterion will fail to accurately reflect the actual contribution (that is, the weight, indicating the varying contributions of each client's learning model during server-side aggregation) of each client in the model learning process, thereby affecting the accuracy of client model aggregation on the server side. Consequently, in scenarios with missing data, devising an effective approach to evaluate the contribution of each client precisely is a crucial challenge for federated CSL.

To illustrate these issues, we conduct experiments on a benchmark Bayesian network insurance dataset using two state-of-the-art federated CSL algorithms: FedPC and FedCSL. In the experiments, we investigate two distinct scenarios: (1) each client has missing data at varying rates, with random missing data applied to the standard insurance dataset and a difference of 0.55 in missing data rates between clients, and (2) all clients possess complete data without any missing values.

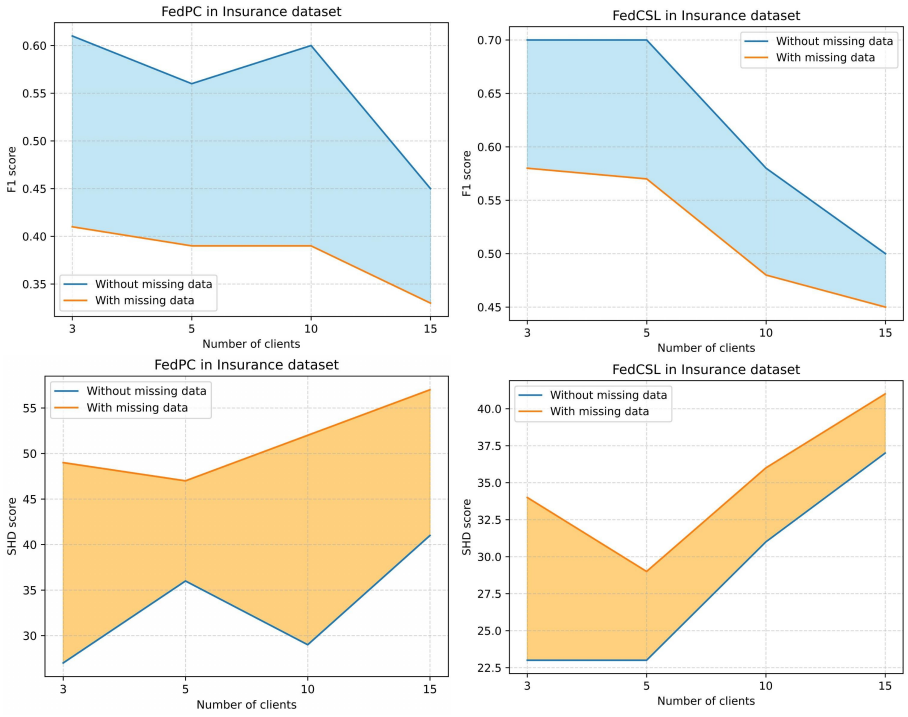
For the missing data scenario, we first perform data imputation independently for each client using the ICkNNI method before applying the FedPC and FedCSL algorithms. As shown in Fig. 1, The F1 scores of the FedPC and FedCSL algorithms differ by up to 32.8% and 18.6% when comparing scenarios with and without missing data, respectively. Additionally, the structural Hamming distances of the two algorithms are significantly different. The experimental results indicate that even when we impute missing data on each client before implementing the federated CSL methods, the performance of FedPC and FedCSL on clients with missing data remain notably lower than that on clients without missing data. This highlights the need to study missing data imputation methods within federated CSL.

Additionally, by comparing the difference in performance between FedPC and FedCSL (as indicated by the shaded area in Fig. 1), we observe that FedCSL utilizes the sample size as an indicator to assess the contribution of each client in the model learning process and employs it for weighted aggregation at the server side. This approach results in a better performance difference between the complete and missing data scenarios than FedPC. This highlights the significance of considering client diversity in FL. Moreover, the performance gap between FedPC and FedCSL in scenarios with and without missing data demonstrates that the missing data rate, along with the sample size, plays an important role in evaluating client contributions. Therefore, in federated CSL, it is not only necessary to design effective missing data imputation strategies but also to develop a method that comprehensively considers both sample size and missing data rate to assess client contributions, ensuring accurate CSL aggregation at the server side.

To address these issues, we propose a new federated CSL method called FedImpCSL, which considers missing data and client diversity. The specific contributions of this study are as follows:

(1) We design a novel local-to-global method for handling missing data in federated skeleton learning consisting of two algorithms: FedLocalImp (**Fed**erated **Local Imp**utation) and FedGlobalImp (**Fed**erated **Global Imp**utation). FedLocalImp employs local hybrid data imputation strategies based on the central sample and adaptive parameter adjustment. In each iteration, clients first use FedLocalImp to impute their original datasets with missing data and then learn their individual client skeletons based on these datasets. The server then performs a weighted aggregation of the client skeletons. This process is iteratively repeated until the aggregated skeleton on the server side ceases to change. Subsequently, in the FedGlobalImp algorithm, we design a global rule for handling missing data based on influential nodes in the dataset to help clients impute more accurate datasets and learn more reliable federated skeletons based on the accurate datasets.

(2) To evaluate client diversity comprehensively, we propose **Fed**erated **C**ontribution **A**ssessment **M**ethod (FedCAM), which considers both the sample size and missing data rate for federated skeleton learning and orientation. In each iteration of federated skeleton learning, each client calculates and sends two indicators (missing data rate



(a) FedPC with Insurance dataset

(b) FedCSL with Insurance dataset

Figure 1: F1 and SHD for scenarios with and without missing data across different numbers of clients. The missing range is set to 0.55, indicating a maximum missing data rate of 55%.

and sample size of its imputed dataset for the current iteration) to the server. The server then performs a weighted aggregation of client skeletons based on the two indicators received from each client in conjunction with the skeleton contribution scores of each client obtained from the previous iteration of aggregated skeletons. When the aggregated skeleton no longer changes, the server obtains accurate skeleton contribution scores for each client. Finally, the server utilizes these precise contribution indicators to achieve an accurate federated skeleton orientation.

(3) We conduct comprehensive experiments using various types of datasets, comparing FedImpCSL with state-of-the-art algorithms to demonstrate the effectiveness of our proposed method.

The study is structured as follows: Section 2 conducts a critical analysis of existing federated CSL and missing data imputation methods, as well as CSL for missing data, emphasizing their respective contributions and limitations. Section 3 provides notations and mathematical meanings employed in this paper. Section 4 proposes the FedImpCSL and describes the detailed for the federated CSL with missing data. Section 5 provides a step-by-step account of the experimental process to demonstrate the effectiveness of FedImpCSL.

2. Related Work

2.1. Federated causal structure learning

Extensive research has been conducted in the field of FL [16, 17] and federated CSL. Recently, NOTEARS-ADMM [6] has emerged as a pioneering method for federated CSL, which integrates the ADMM [18] distributed process with NOTEARS [5] to facilitate FL. However, this approach suffers from substantial limitations and impracticality. Similarly, RFCD [7] introduces the concept of regret value and provides an algorithm to calculate it. Meanwhile, FedDAG [19] addresses data heterogeneity by learning a federated Directed Acyclic Graph (DAG) and incorporating a two-level structure. Additionally, FedC^{2}SL [8] proposes a federated conditional independence test

protocol to perform global CI tests, although it encounters notable limitations in specific scenarios, such as those involving missing data.

To further fill the gap of federated CSL, the recently proposed FedPC method [9] utilizes a novel layer-wise aggregation strategy based on the PC algorithm for federated CSL. This layer-wise strategy enables each client to share and update its skeleton parameters learned at each layer of the FedPC algorithm and identify consistent separation sets at the server. However, due to limitations in identifying separation sets, it can impact the determination of edge direction during the orientation. Although FedPC provides a solution for CSL in a FL setting, it does not adequately handle issues of scalable and weighting during the aggregation process. To address these issues, the latest federated CSL method, FedCSL [10], proposes a local-to-global federation learning approach based on HITON-PC [20] and calculates weights for different clients. Specifically, FedCSL introduces sample size as a weighting factor for local-to-global CSL to improve accuracy, and proposes to calculate the percentage of each client's sample size using the p value from the Conditional Independence (CI) test. However, inaccuracies in the CI test can compromise the accuracy of sample size calculations, and relying solely on this weighting indicator is inadequate for the client diversity issue.

In summary, many approaches have been proposed for federated CSL, but the issue of missing data and client diversity has not received adequate attention so far. In this paper, we aim to develop novel iterative algorithms of federated CSL by considering both missing data and client diversity.

2.2. Missing data imputation methods and CSL for missing data

Most research on data quality issues has focused on dealing with missing data. The common approach for handling missing data involves either deleting or imputing the missing data, such as mean imputation, random forest imputation, ICkNNI, etc. [14]. Notably, ICkNNI facilitates the concurrent utilization of both complete and incomplete cases to fill in missing values. To address the CSL for missing data, some proposed approaches leverage imputation methods tailored for CSL to deal with missing data.

Several algorithms have been proposed for both global and local causal structure learning with missing data. Algorithms for learning the global causal structure, such as MVPC [21] and MICD [22], utilize missing data imputation techniques to address domain-specific challenges. For example, MICD employs multiple imputation in conjunction with constraint-based algorithms to infer the global causal structure from incomplete genetic data. Although these methods support causal learning in missing data environments, they exclusively focus on global structure and overlook local structure learning and high-precision interpolation. Furthermore, MissDAG [23] maximizes the expected likelihood of the observable portion of data within the expectation-maximization (EM) framework to perform causal discovery from incomplete observations. However, MissDAG inherits the time inefficiency issue of EM algorithm. To tackle these challenges, MissLCS [24] introduces an algorithm to learn local causal structures with missing data, which uses an iterative missing data imputation method to achieve more comprehensive and accurate results.

The aforementioned methods are designed for decentralized datasets, and do not adequately address the challenges posed by the federated learning (FL) setting, where multiple communications between clients and the server are required, and the server does not have direct access to original data from all clients. As a result, there are no effective methods that can handle CSL in the FL setting when missing data is present.

3. Notations and Mathematical Meanings

Let $X = \{x_1, x_2, ..., x_d\}$ be a set of d variables (i.e. nodes) under consideration, $C = \{c_1, c_2, ..., c_m\}$ be a set of m different clients, and $D_{original} = \{D_{c_1}, D_{c_2}, ..., D_{c_m}\}$ be the client original incomplete datasets. For each client c_k ($k \in \{1, 2, ..., m\}$), $D^l_{c_k} \in \mathbb{R}^{n^l_{c_k} \times d}$ denotes imputed dataset in the l-th iteration, where $n^l_{c_k}$ is the number of samples in $D^l_{c_k}$ ($l \in \{1, 2, ..., p\}$, p denotes the total number of iterations). Each sample in $D^l_{c_k}$ is defined as $sam^{c_k}_j$ ($j \in \{1, 2, ..., n^l_{c_k}\}$). We define $w^l_{n_{c_k}}$ as the normalization of sample size of client c_k in the l-th iteration using the imputed dataset $D^l_{c_k}$, and $w_{miss_{c_k}}$ as the missing data rate of client c_k. In the skeleton learning of CSL, $skele^l_{c_k}$ denotes the skeleton of client c_k in the l-th iteration and G^l represents the federated skeleton from the server in the l-th iteration.

The definitions and assumptions involved in this paper are as follows:

Definition 1 *(Central sample).* A central sample is a representative sample calculated the arithmetic mean of the same node values across different samples in a dataset. The equation for calculating the central sample $Central\_Sam^l_{c_k}$ of the imputed dataset, for any client $D^l_{c_k} \in \mathbb{R}^{n_{c_k} \times d}$ in the l-th iteration is as follows:

$$Central\_Sam^l_{c_k} = (\frac{1}{n^l_{c_k}}\sum_{j=1}^{n^l_{c_k}} x_1^j, \frac{1}{n^l_{c_k}}\sum_{j=1}^{n^l_{c_k}} x_2^j, ..., \frac{1}{n^l_{c_k}}\sum_{j=1}^{n^l_{c_k}} x_d^j) \tag{1}$$

where $\frac{1}{n^l_{c_k}}\sum_{j=1}^{n^l_{c_k}} x_i^j$ represents the sum of values of the node x_i ($i \in \{1, 2, ..., d\}$) in $D^l_{c_k}$ across n_{c_k} samples. We denote $Central\_Sam\_set^l = \{Central\_Sam^l_{c_1}, ..., Central\_Sam^l_{c_m}\}$ as the set of central samples from all clients in the l-th iteration.

Definition 2 (Client skeleton contribution scores). Let the skeleton learned by client c^l_k in l-th iteration be $skele^l_{c_k}$ and the aggregated federated skeleton for this iteration be G^l. The contribution score of client c^l_k skeleton $skele^l_{c_k}$ to the federated skeleton G^l in l-th iteration, denoted as $H^l_{c_k}$, is calculated as follows:

$$H^l_{c_k} = \begin{cases} \frac{Num\_EE(c_k)}{\sum_{i=1}^{m} Num\_EE(c_i)} & \text{if } Num\_EE(c_k) \neq 0 \\ 1 & \text{if } Num\_EE(c_k) = 0 \end{cases} \tag{2}$$

where $Num\_EE(c_k)$ denotes the number of error edges by counting both the extra edges and miss edges of $skele^l_{c_k}$ comparing to G^l, and $\sum_{i=1}^{m} Num\_EE(c_i)$ denotes the sum of error edge counts across m, which is used for normalization to calculate the relative error.

The set of skeleton contribution scores from m clients in l-th iteration is denoted as $H\_set^l = \{H^l_{c_1}, H^l_{c_2}, ..., H^l_{c_m}\}$

Proposition 1. For original datasets, if a sample misses data for several highly influential variables (nodes), then the data quality of that sample is considered low. Influential nodes are defined based on degree centrality in graph theory.

Definition 3 (Incident edges). For a node $x \in X$ in an undirected graph $G = (X, E)$, the incident edges of x are the set of all edges where at least one endpoint is x. Formally, let $Inc(x)$ denote the set of incident edges of node x, then $Inc(x) = \{e \in E | x \text{ is an endpoint of } e\}$.

Definition 4 (Degree centrality). For the aggregated skeleton G^p, the degree centrality cd_{x_i} of the node x_i ($i \in \{1, 2, ..., d\}$) is calculated using the following equation:

$$cd_{x_i} = \frac{|Inc(x_i)|}{d-1}, (i = \{1, 2, ..., d\}) \tag{3}$$

where $|Inc(x_i)|$ represents the degree of node x_i (i.e., the number of edges directly connected to node x_i). The denominator $d - 1$ is used for normalization, ensuring that the degree centrality values range between $(0, 1)$. Let the set of degree centralities of all nodes in skeleton G^p be denoted as $cd_x\_set = \{cd_{x_1}, cd_{x_2}, ..., cd_{x_d}\}$.

Definition 5 (Influential nodes). For the node x_i, the higher the degree of x_i, the greater its degree centrality, typically indicating a greater importance of x_i. We define $X_{influence}$ as a set of influential variables, which is the set of nodes corresponding to the set cd_x_set in descending order.

Assumption 1 (Invariant Causal DAG [25]). All local datasets are uniformly sampled from the same causal DAG G, and the probability distribution of samples for the same variable space can differ across different clients.

4. Proposed FedImpCSL Approach

To address the limitations of federated CSL with missing data, as shown in Fig. 2, we propose FedImpCSL for hybrid CSL in the FL setting as follows:

Phase 1: Federated skeleton learning with missing data.

Phase 1-1: Each client initially executes the FedLocalImp algorithm to locally impute data into the original incomplete dataset. Subsequently, independent skeleton learning is conducted on the imputed dataset using traditional CSL algorithms. After learning, clients send the learned skeletons and related information (such as the sample size and missing data rate) to the server.

Phase 1-2: The server evaluates the contribution of each client using the FedCAM algorithm. This evaluation uses both the interaction information sent by clients and the evaluation values of each client from the previous iteration on the server side. The server then performs a weighted aggregation of the client skeletons to form the federated skeleton and calculates the evaluation values for the next iteration of clients based on the FedCAM algorithm.

Iteration process: Phases 1-1 and 1-2 are iteratively repeated until the aggregated skeleton on the server side no longer changes. At this point, the server obtains accurate H_{best} scores for the contribution of each client skeleton.

Phase 1-3: Our designed FedGlobalImp algorithm uses accurate client skeleton contribution scores H_{best} to impute the client missing datasets globally by introducing influential nodes. After imputation, a precise and imputed dataset, D_{best}, is formed. Based on D_{best} and H_{best}, an accurate federated skeleton, G_{best}, is learned .

Phase 2: Federated skeleton orientation. Based on the precise dataset D_{best} and federated skeleton G_{best} obtained in Phase 1, a scoring algorithm is used to generate oriented scores for each client. These scores are then sent to the server. The server uses the accurate client skeleton contribution scores H_{best} to perform a weighted aggregation of each client's causal structure, ultimately obtaining the oriented federated skeleton, also known as the federated DAG.

Sections 4.1 and 4.2 will provide more detailed explanations of these two phases.

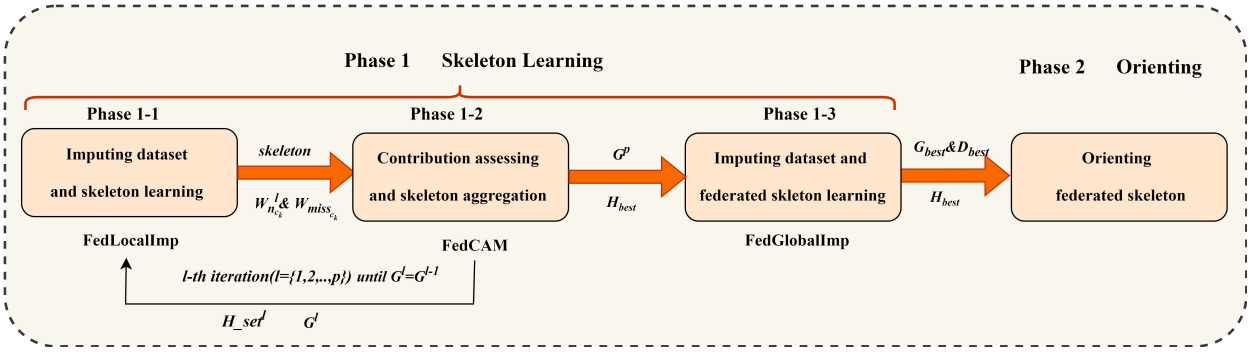


Figure 2: The framework of FedImpCSL.

4.1. Federated skeleton learning with missing data (Phase 1)

As shown in Fig. 3, this section provides a detailed explanation of the federated skeleton learning process in the presence of missing data. This process is primarily supported by three core algorithms: FedLocalImp for imputing missing data on the client side, FedCAM for evaluating the contribution of each client participating in the weighted aggregation, and FedGlobalImp for imputing missing data from a global perspective across clients. The following is a detailed elaboration of these algorithms:

4.1.1. FedLocalImp algorithm for local missing data imputation (Algorithm 1)

As discussed in Section 1, the existing missing data imputation methods cannot be directly applied to the FL setting. These methods are designed for single datasets, whereas the advantage of FL lies in the ability of the server to synthesize information from multiple clients, to enhance the imputation accuracy in the presence of missing data. Therefore, it is necessary to adapt existing imputation methods such as ICkNNI to the FL setting. Methods such as ICkNNI employs K-nearest neighbors for imputation; however, when the client data is low-quality, the imputation results may be biased, thereby affecting the serve's overall evaluation of the client data. To address this issue, we propose a local missing data imputation algorithm, FedLocalImp, in the FL setting. This algorithm fully considers the differences in data quality across clients and achieves more accurate missing data imputation through collaboration between clients and the server, thereby satisfying the need to handle local missing data in the FL setting.

The specific process of the FedLocalImp algorithm is shown in Algorithm 1. The core idea is as follows. In the *l*-th iteration of skeleton learning, because the server cannot obtain information from each client, each client deletes samples with a high missing data rate and directly uses ICkNNI for imputation to form an imputed dataset. On this imputed dataset, each client executes a skeleton learning algorithm [3] to learn its skeleton and then sends both its skeleton and the central sample based on the imputed dataset to the server. After entering Phase 1-2, the server performs a weighted

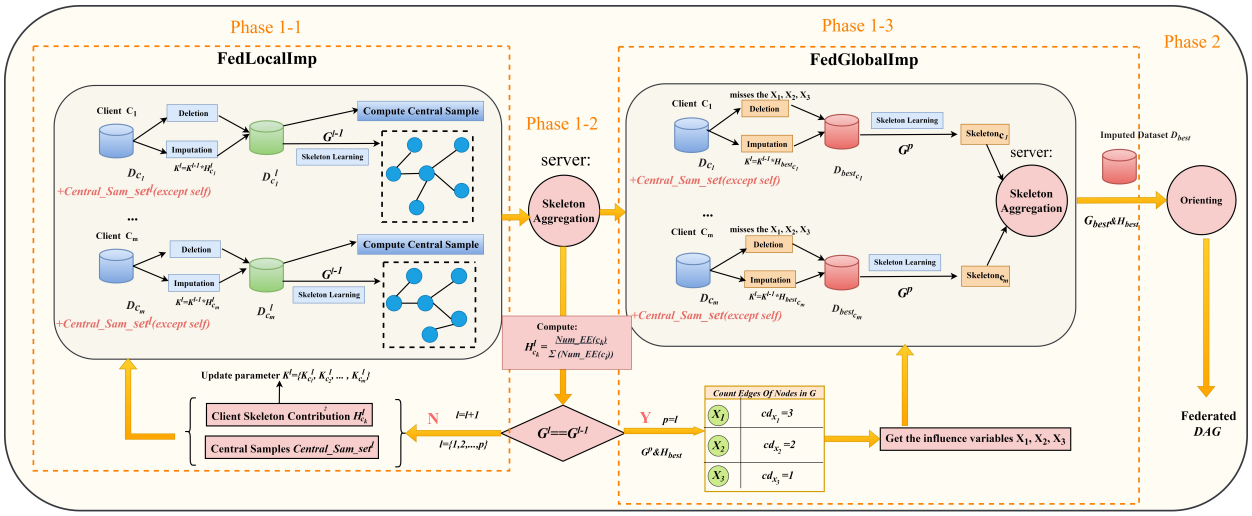


Figure 3: The flowchart of FedLocalImp and FedGlobalImp for tackling incomplete datasets.

aggregation of the skeletons based on the contribution of each client to form the federated skeleton for this iteration and sends the set of central samples $Central\_Sam\_set = \{Central\_Sam^l_{c_1}, Central\_Sam^l_{c_2}, ..., Central\_Sam^l_{c_m}\}$ from all clients to each client. In subsequent iterations (from the 2nd to the p-th), each client first constructs an enhanced dataset based on its original dataset and the central samples received from the server. The K parameter in the current execution of the ICkNNI method is dynamically adjusted based on the contributions of clients from the previous iteration. The ICkNNI method with a dynamic K value is then executed on the enhanced dataset to achieve client data imputation.

FedLocalImp comprises the following two key steps:

Step 1: Each client deletes samples with high missing data rates based on the original dataset (see Algorithm 1 in lines 6-11). Since the number of missing nodes varies among different samples within a client's dataset, we use the number of missing nodes per sample as the criterion. A sample is deleted if the number of missing nodes exceeds half of the total number of nodes d. And a threshold that has shown good accuracy across all datasets in experiments.

Although the deletion strategy in Step 1 can alleviate data quality issues to some extent, directly applying the ICkNNI method to impute missing data in the remaining samples would result in different samples with missing data being imputed to the same values. To improve the accuracy of ICkNNI, we propose the following two strategies:

Strategy 1: Enhance the data quality of each client in the l-th iteration by utilizing the set of central samples $Central\_Sam\_set = \{Central\_Sam^l_{c_1}, Central\_Sam^l_{c_2}, ..., Central\_Sam^l_{c_m}\}$ sent by the server. In each iteration, clients compute their own central samples based on imputed datasets using Eq. (1). This process inherently avoids raw data exposure. The central samples exclusively encodes causal structure learning consistency and global imputation accuracy, rather than replicating individual clients' raw data distributions. Then, through communication between server and clients, the exchange of central samples from different clients is facilitated. During iteration l, each client constructs its enhanced dataset $D^l_{c_k} = D_{c_k} \cup Central\_Sam\_set^{l-1}$, where $Central\_Sam\_set^{l-1}$ denotes the central sample set aggregated from all clients' imputed datasets at $(l-1)$-th iteration (see Definition 1). The imputed dataset is then generated through the ICkNNI algorithm with adaptive K value. Using these central samples from other clients can adjust each client's data distribution in the l-th iteration, enhancing the accuracy of ICkNNI.

Strategy 2: An adaptive K value strategy based on feedback from the server is employed to dynamically adjust the parameter K used by each client when executing ICkNNI. Here, K represents the number of global neighbors, which is a crucial parameter in ICkNNI. In the FL setting, significant differences in data quality among clients may lead to the inclusion of less similar neighbor samples during data imputation when using a fixed K parameter, thereby affecting the accuracy of data imputation. To address this issue, we design a strategy to dynamically adjust the K value based on the client's skeleton contribution scores, following the principle that "better data quality

leads to a larger K value for the number of nearest neighbors during data imputation; conversely, a smaller K value is used". The data quality of a client is quantitatively assessed using the skeleton contribution scores $H^l_{c_k}$ ($0<H^l_{c_k}<1$) (see Definition 2) provided by the server. Based on this, we design the adaptive K calculation equation according to $H\_set^l = \{H^l_{c_1}, H^l_{c_2}, ..., H^l_{c_m}\}$:

$$K^l_{c_k} = \begin{cases} K_{initial} & \text{if } l = 1 \\ K^{l-1}_{c_k} * H^l_{c_k} & \text{if } l \neq 1 \end{cases} \tag{4}$$

where $K_{initial}$ represents an initial K value (often set to 9 [24]), all clients start with the same initial K. $K^l_{c_k}$ indicates the adjusted K value for client c_k in the l-th iteration, based on its skeleton contribution score $H^l_{c_k}$. Thus, when $H^l_{c_k} = 1$, it indicates that client c_k possesses the highest data quality and has significantly contributed to the skeleton learning in the previous iteration. In such cases, the client maintains its K value unchanged while executing ICkNNI. Conversely, lower $H^l_{c_k}$ value indicates lower data quality for the respective client. Correspondingly, $K^l_{c_k}$ is proportionally reduced based on the $H^l_{c_k}$, thereby decreasing the number of nearest neighbors in the data imputation process to enhance the accuracy of data imputation.

Algorithm 1 FedLocalImp

Input Missing datasets from m Clients: $D_{original} = \{D_{c_1}, D_{c_2}, ..., D_{c_m}\}$; number of variable d; number of global nearest neighbors K

Output Imputed dataset from m Clients: $D^p = \{D^p_{c_1}, D^p_{c_2}, ..., D^p_{c_m}\}$; federated skeleton G^p

$G \leftarrow$full connectivity matrix;
$H\_set^1 = \{H^1_{c_1}, ..., H^1_{c_m}\} = \{1, ..., 1\}, l = 1$; /*Let the initial values of H_set and l be 1*/
while $G^l \neq G^{l-1}$ **do**
    /*Client side*/
    for $k = 1$ to m **do**
        for $j = 1$ to $n^l_{c_k}$ **do**
            if $MissNum > \frac{d}{2}$ **then** /*$MissNum$ represents the number of missing data in a sample*/
                delete sam_j in D_{c_k};
                $DelNum^l_{c_k} = DelNum^l_{c_k} + 1$; /*$DelNum$ represents the number of deletion samples in dataset of c_k*/
            end if
        end for
        if $H^l_{c_k} \neq 1$ **then**
            $K^l_{c_k} = K^{l-1}_{c_k} * H^l_{c_k}$;
        end if
        $D^l_{c_k} = ICkNNI(D_{c_k} + Central\_Sam\_set^L, K^l_{c_k})$;
        $skele^l_{c_k} = Skeleton\_learning\_PC(D^l_{c_k}, G^{l-1})$; /*skeleton learning method utilizes Peter-Clark (PC)*/
        compute $Central\_Sam^l_{c_k} = (\frac{1}{n^l_{c_k}} \sum_{j=1}^{n^l_{c_k}} X^j_1, ..., \frac{1}{n^l_{c_k}} \sum_{j=1}^{n^l_{c_k}} X^j_d)$
        Server$\Leftarrow Central_Sam^l_{c_k}$; $skele^l_{c_k}$; each weight indicator (sample size and missing data rate)
    end for
    /*Server side*/
    $l = l + 1$;
    compute $H\_set^l = \{H^l_{c_1}, ..., H^l_{c_m}\}$;
    $G^l = Weighted\_Aggregate(Skele\_set^l)$; /*$Skele\_set^l$ represents the set of $skele^l_{c_k}$ from client* /
    Each Client $\Leftarrow Central\_Sam\_set^l$ (except self); G^l; H_set^l
end while
$p = l$;
return imputed dataset D^p, federated skeleton G^p

Step 2: On the dataset with high-missing samples deleted in Step 1, each client independently executes ICkNNI based on the central sample set (Strategy 1) and adaptive K value (Strategy 2) to impute the missing

data (see Algorithm 1 in lines 3-25). Specifically, each client first receives the central sample set from the server and adds it to its local dataset, thereby constructing an enhanced client dataset. Then, ICkNNI with the adaptive K value strategy is executed on the enhanced client dataset to impute the local missing data.

In summary, during each iteration, each client sequentially executes Step 1 and Step 2 and performs skeleton learning based on its imputed dataset. Subsequently, the client sends relevant information (including sample size, missing data rate, central samples, etc.) and its skeleton to the server. The server then performs weighted aggregation of the skeletons based on client contributions. After generating the federated skeleton in the l-th iteration, the server feeds back the central sample set $Central\_Sam\_set^l$ along with the corresponding client skeleton contribution score $H^l_{c_k}$ to the respective clients. This process iterates until the federated skeleton on the server no longer changes.

4.1.2. FedCAM algorithm for client contribution assessment within weighted aggregation(Algorithm 2)

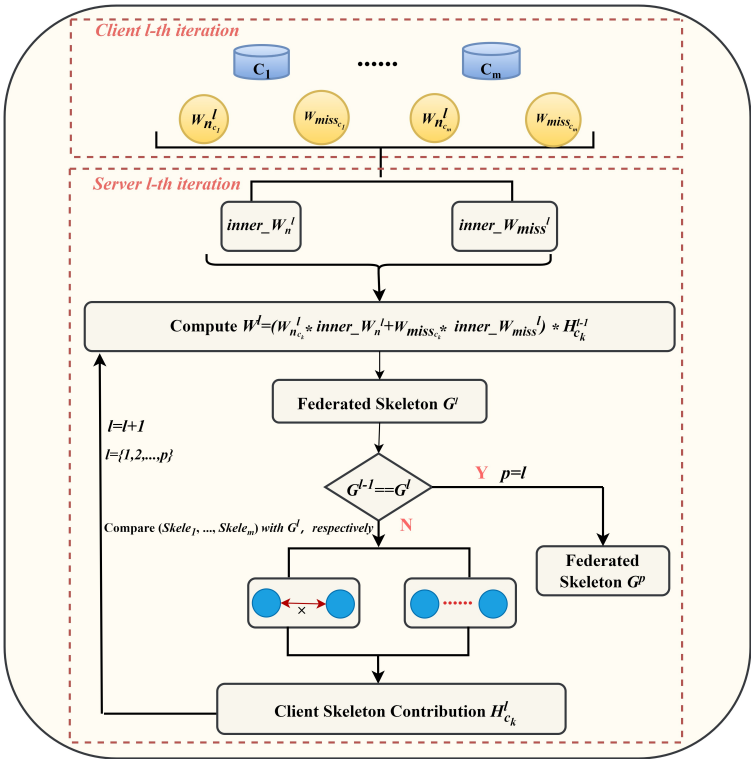


Figure 4: The client contribution assessment within weighted aggregation algorithm FedCAM of FedImpCSL.

To address the issue of client diversity with missing data, we propose the FedCAM algorithm, which evaluates client contributions by considering both the sample size and missing data rate, as shown in Fig. 4. In this algorithm, the client contribution is determined by the sample size, missing data rate, and skeleton contribution scores of the client. The core idea of the algorithm operates as follows. In each iteration after Phase 1-1, the server evaluates the overall contribution of each client through weighted aggregation, considering the missing data rate and the sample size of the imputed dataset in the l-th iteration (current iteration), combined with the previous iteration skeleton contribution scores H_set^{l-1} (initial contribution score of each client is 1). After obtaining the aggregated skeleton for the current iteration, the client skeleton contribution scores H_set^{l-1} are calculated by comparing the aggregated skeleton with the client skeleton and stored for use in subsequent iterations. Therefore, the key to the FedCAM algorithm lies in balancing the relationship between the sample size, missing data rate, and client skeleton contribution scores to better assist the server in performing a weighted aggregation of skeletons.

When measuring the impact of sample size and missing data rate on the contribution of FedCAM, traditional standard deviation methods fail to measure the intrinsic relationships between multiple indicators, particularly when

there are significant differences in the sample size distribution and missing data rate distribution among different clients. Therefore, we introduce entropy weights [26] to determine the weights between these two indicators. Entropy weight is an important weight calculation method in information theory and is particularly suitable for the objective evaluation and importance allocation of various independent indicators in multi-attribute decision-making scenarios. Given that the sample size and missing data rate are independent indicators, they satisfy the prerequisite assumptions for applying the entropy weight. To address this, we propose a weight aggregation strategy based on the entropy weight to more accurately reflect the relative importance of each indicator in a comprehensive evaluation.

Specifically, the FedCAM algorithm consists of the following two steps:

Step1: The server aggregates clients' sample sizes and missing data rates using the entropy weight strategy (see Algorithm 2 in lines 5-14). As shown in Fig. 4, this process details how entropy weights are used to calculate the proportional relationship between sample size and missing data rate during aggregation.

Let $F^l_{c_k j}$ as the value of client c_k under the j-th weight indicator in the l-th iteration, where $j \in \{1, 2\}$, corresponds to sample size and missing data rate, respectively.

First, calculate the proportion of client c_k under the j-th indicator using Eq. (5), denoted as $Q^l_{c_k j}$.

$$Q^l_{c_k j} = \frac{F^l_{c_k j}}{\sum_{k=1}^{m} F^l_{c_k j}}, (k \in \{1, 2, 3, ..., m\}; j \in \{1, 2\}) \tag{5}$$

Next, according to Eq. (6), calculate the entropy e^l_j for each weight indicator using $Q^l_{c_k j}$.

$$e^l_j = -\frac{1}{ln(m)} \sum_{k=1}^{m} Q^l_{c_k j} * ln(Q^l_{c_k j}), (j \in \{1, 2\}) \tag{6}$$

Then calculate the coefficient of variation for each indicator using Eq. (7).

$$CVar^l_j = 1 - e^l_j, (0 <= e^l_j < 1) \tag{7}$$

Finally, according to Eq. (8), determine the weights $inner\_W^l$ in the l-th iteration between the two indicators sample size in the l-th iteration and missing data rate. We define $inner\_w^l_n$ as the weight of sample size and $inner\_w^l_{miss}$ as the weight of missing data rate.

$$inner\_W^l = \frac{CVar^l_j}{\sum_{j=1}^{IndCount} CVar^l_j}, (j \in \{1, 2\}, IndCount = 2) \tag{8}$$

Step2: Perform the weighted aggregation to determine the overall contribution $Score^l$ for the current (l-th) iteration (see Algorithm 2 in lines 15-25). This aggregation considers the combined weight indicators of sample size and missing data rate from Step 1, along with the client skeleton contribution scores H_set^{l-1} from previous iteration. $Score^l$ is utilized to determine whether to retain edges between aggregated skeleton nodes. Each element in $Score^l$, denoted as $Score\_x_u x^l_v$ ($u \in \{1, 2, ..., d\}, v \in \{1, 2, ..., d\}$), is defined as follows:

$$Score\_x_u x^l_v = \sum_{k=1}^{m} [(w^l_{n_{c_k}} * inner\_w^l_n + w_{miss_{c_k}} * inner\_w^l_{miss}) * skele^l_{c_k}(x_u, x_v) * H^{l-1}_{c_k}] \tag{9}$$

where $W^l_{n_{c_k}}$ and $W_{miss_{c_k}}$ are the normalized sample size and missing data rate, respectively. $skele^l_{c_k}$ represents the edge result between nodes x_v and x_u in the skeleton learned by client c_k in l-th iteration. By comparing $Score\_x_v x_u$ with the threshold $ratio * m$ (where m is the number of clients, and ratio is set to 0.3 [9]), decide whether to retain the edge between node x_v and node x_u is made. If $Score\_x_v x_u{}^l >= ratio * m$, the edge is retained; otherwise, it is deleted.

In each iteration, each client performs Step 1 and Step 2 to calculate the overall contribution by integrating the client's sample size for the current iteration, missing data rate, and the client skeleton contribution scores from previous iteration. This process is used to aggregate the server's federated skeleton for the current iteration and calculate the client skeleton contribution scores for the next iteration. This process iterates continuously until the skeleton obtained

after weighted aggregation remains unchanged. Ultimately, the algorithm provides a reliable aggregated skeleton and accurate client skeleton contribution scores for Phase 1-3.

It is important to note that the server receives only the missing data rate and the sample size after imputation from each client, without involving any original data privacy. Additionally, since the clients adopt a combination of deletion and imputation methods to process their original datasets in each iteration, the sample sizes of these clients' imputed datasets may vary. Therefore, clients need to send the updated sample sizes to the server after each iteration.

Algorithm 2 FedCAM

Input Missing datasets from m Clients: $D_{original} = \{D_{c_1}, D_{c_2}, ..., D_{c_m}\}$; number of variable d; number of global nearest neighbors K

Output G^p

$G \leftarrow$full connectivity matrix;
$H\_set^l = \{H_{c_1}^1, ..., H_{c_m}^1\} = \{1, ..., 1\}, l = 1$; /*Let the initial values of H_set and l be 1*/
while $G^l \neq G^{-1}$ **do**
    /*Client side*/
    for $k = 1$ to m **do**
        $W_{miss_{c_k}} = missing\_rate(D_{c_k})$;
        $D_{c_k}^l, DelNum_{c_k}^l, n_{c_k}^l = FedLocalImp(D_{c_k})$; /*$DelNum$ represents the number of deletion samples in dataset of c_k*/
        $W_{n_{c_k}}^l = n_{c_k}^l - DelNum_{c_k}^l$;
        $skele_{c_k}^l = Skeleton\_learning\_PC(D_{c_k}^l, G^{l-1})$; /*Skeleton learning method utilizes Peter-Clark (PC)*/
        Server $\Leftarrow skele_{c_k}^l; w_{n_{c_k}}^l; w_{miss_{c_k}}$
    end for
    /*Server side*/
    $compute\ e_j^l = -\frac{1}{ln(m)} \sum_{k=1}^{m} Q_{ij} * ln(Q_{c_k j}), (j = \{1, 2\})$;
    $compute\ inner\_W^l = \frac{CVar_j}{\sum_{j=1}^{m} CVar_j}$;
    $compute\ Score\_x_u x_v^l = \sum_{i=1}^{m} [(w_{n_{c_i}^l}^l * inner\_w_n^l + w_{miss_{c_i}} * inner\_w_{miss}^l) * skele_{c_i}^l(x_u, x_v) * H_{c_i}^{l-1}], (u, v \in \{1, 2, ..., d\})$;
    if $Score\_x_u x_v^l < ratio * m$ **then**
        Delete $e(x_u, x_v)$ from G^l; /*$e(x_u, x_v)$ represents the edge between x_u and x_v*/
    end if
    for *Each Client* **do**
        if $Num\_EE(c_k) \neq 0$ **then**
            $compute\ H_{c_k}^l = \frac{Num\_EE(c_k)}{\sum_{i=1}^{n} Num\_EE(c_i)}$;
        else
            $H_{c_k}^l = 1$;
        end if
    end for
    $l = l + 1$;
end while
$p = l$
return G^p

4.1.3. FedGlobalImp algorithm for global missing data imputation (Algorithm 3)

As shown in Fig. 3, through multiple iterations of Phases 1-1 and 1-2, an accurate client skeleton contribution score H_{best} can be ultimately determined. In Phase1-1, FedLocalImp does not fully consider other factors (such as sample deletion conditions) when handling missing data. In particular, after the aggregated skeleton stabilizes, not all samples with fewer missing nodes should be retained. This is because some samples, despite having few missing nodes, may lack influential nodes (see Definition 5). Consequently, this can lead to inaccuracies in the information regarding the influential nodes in the client datasets generated during the $(p+1)$-th iteration, thereby affecting the accuracy of the targeted results.

To address this issue, we propose a global imputation algorithm called FedGlobalImp, based on the client skeleton contribution scores H_{best}. Specifically, in Phase1-1, we replace FedLocalImp with FedGlobalImp to impute missing data, and then proceed to Phase 1-2 based on the client skeleton contribution scores H_{best}, ultimately obtaining an accurate federated skeleton and imputed client datasets.

Next, we elaborate on FedGlobalImp. The key innovation of this algorithm is to redefine the conditions for deleting samples from the client datasets. Specifically, the condition for deleting samples no longer solely depends on the number of missing nodes but also considers whether the sample contains influential nodes (see Definition 5) to jointly determine whether a sample should be deleted. That is, if a sample lacking these influential nodes, it will be deleted directly, regardless of the number of missing nodes.

As shown in Algorithm 3, the hybrid imputation rules of FedGlobalImp mainly include the following three steps:

Step1: Based on the p-th aggregated skeleton G^p, determine the set of influential nodes $X_{influence}$ (see Algorithm 3 in lines 2-6). The specific procedure is as follows: first, calculate the degree correlation set $cd_{x\_}set$ (see Definition 4) for each node on the aggregated skeleton G^p. Then, compare the degrees $cd_{x_i}(i \in \{1, 2, ..., d\})$ for all nodes, and sort them in descending order to obtain the node set X. The set of influential nodes $X_{influence}$ comprises the nodes that correspond to the top $\frac{d}{2}$ elements in the set $cd_{x\_}set$.

Step2: Update the hybrid estimation rules based on the influential node set and implement the corresponding strategy (see Algorithm 3 in lines 8-17). The specific rule is: if a sample lacks nodes from the set $X_{influence}$, and the number of these missing nodes exceeds half of the $X_{influence}$, then that sample will be directly deleted, even if no other nodes are missing in terms of quantity.

Algorithm 3 FedGlobalImp

Input Missing datasets from m Clients: $D_{original} = \{D_{c_1}, D_{c_2}, ..., D_{c_m}\}$; number of variable d; number of global nearest neighbors $K^p_{c_k}$; federated skeleton G^p; central samples $Central\_Sam^p_{c_k}$; accurate client skeleton contribution H_{best}

Output Complete datasets from m Clients: $D_{best} = \{D_{best_{c_1}}, D_{best_{c_2}}, ..., D_{best_{c_m}}\}$, federated skeleton G_{best}

1: /*Server side*/
2: **for** $i = 1$ to d **do**
3: compute the correlated edges cd_{x_i} of x_i in G^p
4: **end for**
5: Descending order the set $cd_{x\_}set$ and calculates the influence variables set $X_{influence}$ corresponding to the $cd_{x\_}set$;
6: Each Client$\Leftarrow X_{influence}$;
7: /*Client side*/
8: **for** $k = 1$ to m **do**
9: **for** $j = 1$ to $Sam^p_{c_k}$ **do**
10: **if** $MissNum > \frac{\|X_{influence}\|}{2}$ **then** /*$MissNum$ represents the number of missing data in a sample*/
11: Delete sam_j in D_{c_k};
12: **end if**
13: **end for**
14: $D_{best_{c_k}} = ICkNNI(D_{c_k} + Central\_Sam\_set^p, K^p_{c_k} * H_{best_{c_k}})$;
15: $skele_{c_k} = Skeleton\_learning\_PC(D_{best_{c_k}}, G^l)$; /*Skeleton learning method utilizes Peter-Clark (PC)*/
16: **end for**
17: $G_{best} = Weighted\_Aggregate(Skele\_set)$; /*$Skele\_set$ represents the set of $skele_{c_k}$ from client*/
18: **return** complete datasets D_{best}, federated skeleton G_{best}

After executing Step 1 and Step 2, for the remaining samples, we adopt an adaptive K value based ICkNN method using H_{best} to impute the missing data, which is the final imputation of missing data on their original datasets. Subsequently, based on the imputed datasets, the final federated skeleton learning is conducted, yielding an accurate federated skeleton G_{best} and the accurate client datasets D_{best}. These results will lay the foundation for the federated skeleton orientation in Phase 2.

4.2. Federated skeleton orientation (Phase 2)

In the second phase of FedImpCSL, we orient the federated skeleton obtained in Phase 1. Existing federated CSL algorithms such as FedPC conduct orientation by identifying a consistent separation set on the client side. However, this approach suffers from critical limitations in conditional independence (CI) tests and inaccuracies in the aggregated skeleton, leading to reduced accuracy in the orientation phase. To improve this process, we ensure that the two phases of skeleton learning and orientation are independent of each other and designed a novel orientation strategy.

To enhance the accuracy of the federated skeleton orientation, we introduce the Tabu hill-climbing algorithm [27, 4]. This algorithm is characterized by its ability to search from any DAG and does not rely on the aggregation of separation sets. In our implementation, the Tabu hill-climbing algorithm is applied separately to each client's local dataset to compute the orientation scores. Specifically, based on the accurate client dataset $D_{best} = \{D_{best_{c_1}}, D_{best_{c_2}}, ..., D_{best_{c_m}}\}$, the aggregated skeleton G_{best} is locally oriented. After determining the causal structure for each client, the clients communicate with the server once to upload their orientation results. Finally, the server performs a weighted aggregation of all the client causal structures to generate the final federated DAG.

The specific steps for the federated skeleton orientation phase are as follows:

Step1: Weighted aggregation of client DAGs. In Phase 1, we use FedCAM to calculate precise contribution scores for each client, including sample size $W_{n_{c_k}}^{p+1}$, missing data rate $W_{miss_{c_k}}$, and client skeleton contribution H_{best}. We also determine the proportional relationship between sample size and missing data rate in the weight allocation, denoted as $inner\_W_n^{p+1}$ and $inner\_W_{miss}^{p+1}$. Based on these parameters, the server calculates the weighted aggregation of each client's DAG matrix using the following equation:

$$DAG(x_u, x_v) = \sum_{k=1}^{m} [(W_{n_{c_k}}^{p+1} * inner\_W_n^{p+1} + W_{miss_{c_k}} * inner\_W_{miss}^{p+1}) * DAG_{c_k}(x_u, x_v) * H_{best_{c_k}}] \tag{10}$$

In Eq. (10), $DAG_{c_k}(x_u, x_v)$ denotes the DAG result between x_u and x_v obtained after orientation at each client.

Step 2: Determining the direction of the federated DAG based on the aggregated DAG matrix. Inspired by the FedCSL algorithm, we establish our orientation rules. In this process, the server directly compares the aggregated data for each pair of $DAG(x_u, x_v)$ and $DAG(x_v, x_u)$. If $DAG(x_u, x_v) > DAG(x_v, x_u)$, the direction between nodes is set as $x_u \rightarrow x_v$; otherwise, it is set as $x_v \rightarrow x_u$. Through this series of numerical comparisons and analysis steps, we can accurately determine the final direction of the federated DAG.

5. Experiments

In this section, we compare the performance of the proposed method, FedImpCSL, with six baseline methods across six benchmark Bayesian networks (BNs), a real-world dataset, a high-dimensional synthetic dataset and Non-IID synthetic datasets. We design a comprehensive series of experiments to demonstrate the effectiveness of FedImpCSL.

5.1. Experiment settings

5.1.1. Datasets

We utilize seven datasets to illustrate the effectiveness of our method, encompassing both the benchmark datasets and real-world dataset. These datasets are detailed in Table 1. Each client is assigned 5000 samples, with varying sample size and missing data rate set to simulate real-world scenarios.

Benchmark Bayesian network (BN) datasets. We employ six benchmark datasets; Child, Insurance, Alarm, Hepar2, Andes and Pigs. These datasets belong to different size classes based on the number of nodes, each containing 5000 observations. The specifics of these six benchmark BNs are presented in Table 1.

Table 1: Summary of Benchmark BNs

Network	Num. Vars	Num. Edges	Data Size
Sachs	11	17	5000
Child	20	25	5000
Insurance	27	52	5000
Alarm	37	46	5000
Hepar2	70	123	5000
Andes	223	338	5000
Pigs	441	592	5000

Real-world dataset. We also incorporate the real-world dataset Sachs [28] into our experiments. The results demonstrate that our method achieves high levels of all metrics on this dataset. Sachs represents a protein signaling network, and is a biological dataset that indicates the levels of different proteins and phospholipids in human cells. It serves as a benchmark graphical model, consisting of 11 nodes and 17 edges.

High-dimensional synthetic dataset. We utilize high-dimensional dataset (>1000 nodes) to compare FedImpCSL with the best baseline method, FedCSL, to verify the scalability of our proposed method. The synthetic dataset is generated using the BN toolbox [1], and it is constructed based on specific causal structures and the corresponding conditional probability tables (CPTs).

Non-IID synthetic datasets. We employ Non-IID datasets to compare FedImpCSL with the state-of-the-art baseline method to validate its practical applicability in real-world federated settings. Specifically, by randomly assigning each client distinct linear models, structural equation model (SEM) types, and noise intensities, we introduce distributional heterogeneity across clients while maintaining a consistent DAG structure. This design effectively simulates complex Non-IID data scenarios in federated learning. Furthermore, we control the degree of statistical heterogeneity by adjusting Dirichlet parameter α (e.g., $\alpha = 0.5, 0.8, 1.0$). The code implementing this data generation methodology is publicly available for reproducibility [2]. Under the Assumption 1 (shown in Section 3), federated causal structure learning approaches remains effective even with Non-IID data [29]. This assumption posits that while the data distributions may vary across clients (Non-IID), they share the same underlying causal relationships. This ensures compatibility with causal sufficiency requirement, as distributional differences do not violate the invariance of the causal DAG.

5.1.2. Evaluation metrics

We use two metrics to evaluate the experimental results: the F1 score, SHD (Structural Hamming Distance).

SHD. Specifically, the SHD score is calculated as the sum of undirected edges, reversed edges, missing edges, and extra edges, where a smaller SHD signifies better performance.

F1 score. The F1 score is a composite evaluation metric computed as $F1 = 2 * Recall * Precision/(Recall + Precision)$, where a larger F1 indicates better performance. Precision is defined as the number of correctly predicted arrows in the output divided by the number of edges in the algorithm's output, while Recall is the number of correctly predicted arrows in the output divided by the number of true arrows in the true causal structure.

5.1.3. Baselines

We compare the performance of FedImpCSL with five federated causal learning methods: NOTEARS-ADMM [6], RFCD [7], FedPC [9], FedC^{2}SL [8], and FedCSL [10]. The implementation details are as follows: FedPC, FedMMHC, FedCSL and our proposed FedImpCSL approach are implemented in *MATLAB*, while, NOTEARS-ADMM, RFCD, FedC^{2}SL are implemented in *Python* [30]. Since all five methods lack native support for missing data, we adopt the ICkNNI imputation method to impute in the missing data, with each client processed independently. Additionally, FedCSL is the only method based on the HITON-PC algorithm, which takes a hybrid approach to causal structure learning from local to global. Therefore, we implement a hybrid federated CSL method, FedMMHC, based on the MMHC algorithm for comparison, using the same imputation method to process the missing data.

[1] https://www.cs.ubc.ca/ murphyk/Bayes/junk/bnsoft.html

[2] https://github.com/ErdunGAO/FedDAG/tree/main/datasets

NOTEARS-ADMM. Apply the ADMM distributed approach to NOTEARS for parameter exchange.

RFCD. Exchange regret values to identify the best DAG.

FedPC. Aggregate skeletons iteratively utilizing a voting mechanism based on the global constraint algorithm PC.

FedC^{2}SL. Perform CI tests on the server side, and proposes a global CI test.

FedCSL. Add sample size as a weight indicator to achieve federated CSL from local to global based on the HITON-PC algorithm.

FedMMHC. Each client learns a complete DAG based on MMHC and sends it to the server for aggregation.

5.2. Results on BN datasets

In this section, we report the experimental results of our method compared with six baselines on the standard BN datasets by F1 and SHD as evaluation metrics.

Tables 2 and 3 summarize the F1 and SHD values obtained with BNs utilizing 5000 data samples. In our experimental results, we use '-' to denote that a method does not generate results for the corresponding BN due to the total running times exceeding one day. Based on our analysis, we make the following observation.

Table 2 presents the F1 value for six BNs. The F1 values of FedImpCSL are higher than the best baseline non-iterative method, FedCSL and FedMMHC, for 84% of the datasets. Otherwise, the rest of the F1 values are nearly comparable with these two methods. The special situations attributable to the method utilizes a local-to-global casual structure learning. Furthermore, it significantly outperforms iterative algorithms such as RFCD, FedC^{2}SL, FedPC and NOTEARS-ADMM, especially in medium-sized networks like the Child and Alarm datasets. For the large network Pigs (441 nodes), FedImpCSL performs better than the other six algorithms. From the perspective of the number of clients and the range of missing data rates, the F1 value always performs best when the number of clients increases to 15 and the largest range of missing data rates is set to 0.55. In the condition of the 800 dataset with 15 clients, the evaluation metrics of FedCSL drop significantly due to its limitations under localized learning, and our method also suffers losses in such a small sample setting, but it is far better than FedCSL, as detailed in Fig. 5.

Table 3 displays the SHD values across six BNs. Notably, some SHD values among the datasets exhibit minimal differences, and the FedImpCSL algorithm demonstrates superior performance compared to iterative algorithms such as RFCD, FedC^{2}SL, FedPC and Notears-ADMM. The SHD values of the FedImpCSL are lower than those of the best baseline non-iterative method, FedCSL, when using a 75% dataset. Especially, the SHD values perform exceptionally well on the large network, Pigs (441 nodes) dataset. Its optimal performance has reached 0, indicating minimal structural errors in the DAG. From the perspective of the number of clients and the range of missing data rates, the SHD values consistently perform best when the number of clients increases to 15 and the largest range of missing data rates is set to 0.55.

These observational results can be attributed to several factors:

(1) FedCSL which is based on the local-to-global learning, performs poorly compared to FedImpCSL. It is almost inapplicable in cases of missing data, and its sample size weighting method is inaccurate because it relies on the CI test. Conversely, the FedImpCSL method learns globally, offering good inclusiveness for small sample size and complex node relationships. Additionally, the introduction of weight indicators is more comprehensive and both the weight indicators and missing treatment approaches are more objective.

(2) Other methods such as NOTEARS-ADMM and RFCD are sensitive to user-defined thresholds, resulting in their performance not being outstanding. Different clients may have different edge pruning thresholds, and it is challenging to select an appropriate threshold to prune false directed edges. Moreover, inappropriate thresholds may prune correct edges or retain incorrect edges. In contrast, FedImpCSL does not require the introduce of additional ADMM update rules and conducts the communication process with the server during the weighted iteration of the aggregation skeleton to obtain the skeleton and the complete dataset, making it more efficient.

(3) Compared to RFCD, whose calculation of local regret values may be affected by the imputed data, the server calculates their worst-case regrets and sends the algorithm to each client, without considering the differences that exist among client (i.e., the weighting problem). FedImpCSL provides more robust solution in considering the aggregation and updating of missing data methods on the server side.

(4) FedC^{2}SL aggregates the CI tests of each client, and which makes it difficult to identify problems within the CI tests of each client itself. It further leads to issues with the global skeleton and separation set, ultimately resulting in incorrect causal results. Moreover, its proposed federated conditional independence testing protocol requires clients

to send some statistics for global CI testing. However, this process may be affected by the incompletion data and also fails to account for the diversity among clients.

(5) In contrast to the FedPC algorithm, it performs averagely. The FedImpCSL demonstrates improved dynamic convergence over static convergence, and the dynamic weighted aggregation method exhibits even more significantly enhancements in terms of performance and other aspects when addressing missing data.

Table 2: Structure learning results of F1 on the benchmark datasets. There are 5,000 samples in total, allocated unevenly across {3, 5, 10, 15} clients. And there are three different situation about missing data rate range, shown in Section 4.5 detailed.

		range 0				range 0.15				range 0.55			
Dataset	Method	3	5	10	15	3	5	10	15	3	5	10	15
Child	Notears-ADMM	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
	FEDC2SL	0.32	0.32	0.34	0.32	0.35	0.34	0.35	0.35	0.32	0.38	0.32	0.33
	RFCD	0.37	0.32	0.32	0.35	0.38	0.38	0.34	0.38	0.41	0.34	0.31	0.32
	FEDPC	0.44	0.45	0.35	0.41	0.64	0.54	0.45	0.29	0.36	0.51	0.34	0.31
	FEDMMHC	0.84	0.61	0.55	0.52	0.82	0.76	0.61	0.48	0.55	0.52	0.44	0.45
	FedCSL	0.83	0.76	**0.88**	0.79	0.87	0.85	0.89	0.79	0.89	**0.91**	0.87	0.76
	FedImpCSL	**0.94**	**0.9**	0.87	**0.83**	**0.91**	**0.87**	**0.91**	**0.81**	**0.9**	0.9	**0.92**	**0.81**
Insurance	Notears-ADMM	0.06	0.08	0.11	0.18	0.08	0.09	0.12	0.12	0.2	0.11	0.12	0.11
	FEDC2SL	0.36	0.35	0.33	0.34	0.35	0.35	0.36	0.33	0.33	0.34	0.36	0.37
	RFCD	0.39	0.35	0.39	0.36	0.38	0.35	0.36	0.33	0.37	0.36	0.35	0.35
	FEDPC	0.41	0.45	0.42	0.42	0.38	0.4	0.37	0.4	0.41	0.39	0.39	0.33
	FEDMMHC	0.6	0.45	0.46	0.52	0.52	0.53	0.42	0.43	0.5	0.46	0.41	0.35
	FedCSL	**0.66**	0.52	0.55	**0.58**	0.57	**0.59**	0.55	**0.53**	0.58	0.57	0.48	0.45
	FedImpCSL	0.64	**0.6**	**0.57**	0.57	**0.65**	0.54	**0.58**	0.46	**0.67**	**0.61**	**0.59**	**0.5**
Alarm	Notears-ADMM	0.15	0.18	0.21	0.2	0.22	0.2	0.21	0.23	0.15	0.18	0.24	0.23
	FEDC2SL	0.36	0.34	0.35	0.36	0.37	0.38	0.37	0.34	0.38	0.38	0.36	0.34
	RFCD	0.4	0.4	0.38	0.37	0.37	0.37	0.41	0.4	0.36	0.37	0.44	0.41
	FEDPC	0.48	0.55	0.39	0.4	0.39	0.46	0.37	0.39	0.42	0.36	0.38	0.4
	FEDMMHC	0.5	0.41	0.41	0.33	0.37	0.39	0.41	0.3	0.39	0.41	0.38	0.25
	FedCSL	0.57	0.53	0.42	0.56	0.55	**0.62**	0.53	0.38	0.58	0.47	0.45	0.36
	FedImpCSL	**0.69**	**0.57**	**0.49**	**0.62**	**0.7**	0.51	**0.52**	**0.53**	**0.63**	**0.51**	**0.48**	**0.52**
Hepar 2	Notears-ADMM	0.14	0.14	0.13	0.14	0.14	0.15	0.13	0.1	0.14	0.14	0.13	0.1
	FEDC2SL	0.38	0.38	0.36	0.35	0.38	0.37	0.36	0.34	0.39	0.39	0.38	0.35
	RFCD	0.37	0.36	0.37	0.33	0.37	0.37	0.37	0.36	0.36	0.34	0.35	0.33
	FEDPC	0.44	0.42	0.31	0.22	0.39	0.38	0.29	0.26	0.42	0.38	0.32	0.21
	FEDMMHC	0.43	0.39	0.27	0.15	0.47	0.41	0.3	0.16	0.41	0.37	0.19	0.17
	FedCSL	**0.6**	0.48	0.36	0.33	**0.63**	0.44	0.36	0.33	**0.54**	0.48	0.42	0.31
	FedImpCSL	0.56	**0.51**	**0.53**	**0.44**	0.51	**0.5**	**0.37**	**0.4**	0.51	**0.56**	**0.57**	**0.42**
Andes	Notears-ADMM	0.14	0.14	0.13	0.14	0.14	0.13	0.1	0.12	0.13	0.14	0.12	0.1
	FEDC2SL	0.34	0.34	0.33	0.32	0.35	0.34	0.33	0.33	0.33	0.33	0.32	0.28
	RFCD	0.35	0.34	0.32	0.33	0.34	0.34	0.29	0.3	0.34	0.33	0.33	0.3
	FEDPC	0.71	0.68	0.65	0.61	0.71	0.7	0.69	0.6	0.68	0.7	0.67	0.6
	FEDMMHC	0.49	0.38	0.26	0.18	0.51	0.39	0.26	0.19	0.44	0.34	0.27	0.17
	FedCSL	**0.74**	**0.76**	0.71	0.7	**0.78**	**0.74**	0.72	0.66	**0.75**	**0.76**	0.7	0.64
	FedImpCSL	0.7	0.65	**0.74**	**0.71**	0.58	0.61	**0.77**	**0.73**	0.63	0.55	**0.79**	**0.68**
Pigs	Notears-ADMM	0.35	0.35	0.35	0.34	0.34	0.35	0.35	0.33	0.35	0.34	0.33	0.3
	FEDC2SL	-	-	-	-	-	-	-	-	-	-	-	-
	RFCD	-	-	-	-	-	-	-	-	-	-	-	-
	FEDPC	0.81	0.8	0.7	0.61	0.8	0.76	0.7	0.72	0.75	0.7	0.65	0.6
	FEDMMHC	0.93	0.85	0.66	0.58	0.95	0.85	0.69	0.57	0.73	0.7	0.64	0.54
	FedCSL	0.98	0.98	0.95	0.9	0.99	0.98	0.96	0.9	0.97	0.97	0.96	0.89
	FedImpCSL	**0.99**	**0.99**	**1**	**1**	**0.99**	**1**	**1**	**1**	**0.99**	**1**	**1**	**0.97**

In particular, the results of the FedImpCSL algorithm and other baseline algorithms are shown in Fig. 6. Utilizing the Nemenyi test [31], a Critical Difference (CD) value is employed to compare the difference between the average rankings of each algorithms. Computation of CD is shown in Eq. (11).

$$CD = q_{\alpha,r}\sqrt{\frac{r(r+1)}{6N}} \tag{11}$$

In Eq. (11), α is the significance level, r denotes the number of comparison methods, N is the number of datasets. We set α=0.05, r =7 and N=6. As shown in Fig. 6, FedImpCSL is the only method that achieves the lowest rank across

Table 3: Structure learning results of SHD on the benchmark datasets. There are 5,000 samples in total, allocated unevenly across {3, 5, 10, 15} clients. And there are three different situation about missing data rate range, shown in Section 4.5 detailed.

		range 0				range 0.15				range 0.55			
Dataset	Method	3	5	10	15	3	5	10	15	3	5	10	15
Child	Notears-ADMM	7	6	6	7	5	5	6	7	5	6	7	7
	FEDC2SL	12	12	12	12	10	13	13	15	10	9	11	13
	RFCD	10	8	8	15	9	11	12	13	17	11	13	10
	FEDPC	19	19	26	25	12	14	20	27	28	18	27	29
	FEDMMHC	8	31	36	20	9	15	29	44	37	45	53	44
	FedCSL	6	8	4	9	4	5	4	9	4	**3**	5	12
	FedImpCSL	**2**	**4**	**6**	**7**	**3**	**3**	**6**	**8**	**4**	4	**3**	**9**
Insurance	Notears-ADMM	50	50	51	55	50	52	52	58	52	52	60	63
	FEDC2SL	35	**31**	38	42	38	34	38	32	37	38	41	40
	RFCD	40	39	41	42	30	32	37	37	36	41	40	39
	FEDPC	42	41	45	44	43	43	49	43	49	47	52	57
	FEDMMHC	51	80	72	38	58	60	81	67	73	78	89	78
	FedCSL	**26**	34	33	**33**	31	**31**	34	**35**	34	**29**	36	41
	FedImpCSL	28	34	**15**	37	**28**	32	**31**	38	**24**	31	**32**	**36**
Alarm	Notears-ADMM	45	46	46	47	42	46	51	46	42	48	52	51
	FEDC2SL	29	33	30	37	37	32	34	32	33	37	36	33
	RFCD	47	51	38	41	43	42	39	46	51	43	41	47
	FEDPC	56	42	71	65	59	48	68	66	56	62	77	59
	FEDMMHC	57	95	95	94	68	101	108	93	93	104	103	122
	FedCSL	28	32	40	29	30	**25**	33	46	28	**35**	36	50
	FedImpCSL	**22**	**29**	**34**	**35**	**23**	31	**30**	**37**	**25**	37	**33**	**34**
Hepar 2	Notears-ADMM	123	116	120	125	122	123	135	136	118	125	136	142
	FEDC2SL	91	96	100	97	97	95	101	100	108	99	105	106
	RFCD	154	143	156	150	145	146	155	152	151	143	155	156
	FEDPC	99	89	99	106	105	93	13	103	98	93	97	107
	FEDMMHC	139	172	284	349	132	167	250	394	169	197	314	355
	FedCSL	**69**	84	96	99	**67**	87	96	98	**76**	**85**	90	100
	FedImpCSL	77	**83**	**84**	**88**	83	**81**	**95**	**91**	80	81	**72**	**90**
Andes	Notears-ADMM	442	512	502	542	468	485	552	854	551	562	436	663
	FEDC2SL	164	152	163	166	164	178	165	157	172	156	182	188
	RFCD	143	146	156	158	156	162	168	186	163	168	188	196
	FEDPC	181	164	174	144	175	158	160	192	202	167	170	196
	FEDMMHC	489	723	1102	1521	546	689	1103	1455	563	819	1136	1640
	FedCSL	**150**	**152**	157	163	**133**	**148**	151	174	**141**	**135**	159	185
	FedImpCSL	207	235	**154**	**156**	247	249	**140**	**151**	216	276	**124**	**169**
Pigs	Notears-ADMM	286	320	318	326	310	293	326	382	336	441	422	578
	FEDC2SL	-	-	-	-	-	-	-	-	-	-	-	-
	RFCD	-	-	-	-	-	-	-	-	-	-	-	-
	FEDPC	207	181	257	325	451	362	423	661	362	487	462	556
	FEDMMHC	73	199	584	825	47	195	527	837	437	502	662	992
	FedCSL	12	19	48	107	5	13	34	108	25	25	43	122
	FedImpCSL	**5**	**1**	**0**	**0**	**6**	**0**	**0**	**0**	**2**	**0**	**0**	**46**

different observational data. Specifically, as the size of the clients-number increases, the rank value of FedImpCSL consistently remains at 1.

5.3. Results on a real-world dataset

In this section, we compare the FedImpCSL method with six federated causal learning methods on the real-world dataset, Sachs, with an experimental sample size set to 5000. As shown in Figs. 7 and 8, FedImpCSL is able to maintain superior performance across various missing data rate settings and numbers of clients. In particular, even under the condition of 15 clients, the F1 and SHD metrics remain superior when compared to the best baseline non-iterative algorithm FedCSL and other methods.

5.4. Results on a high-dimensional synthetic dataset

To verify the scalability of FedImpCSL on high-dimensional synthetic dataset, we compare FedImpCSL with the best baseline method, FedCSL, which emphasizes scalability. Experiments are conducted on a high-dimensional

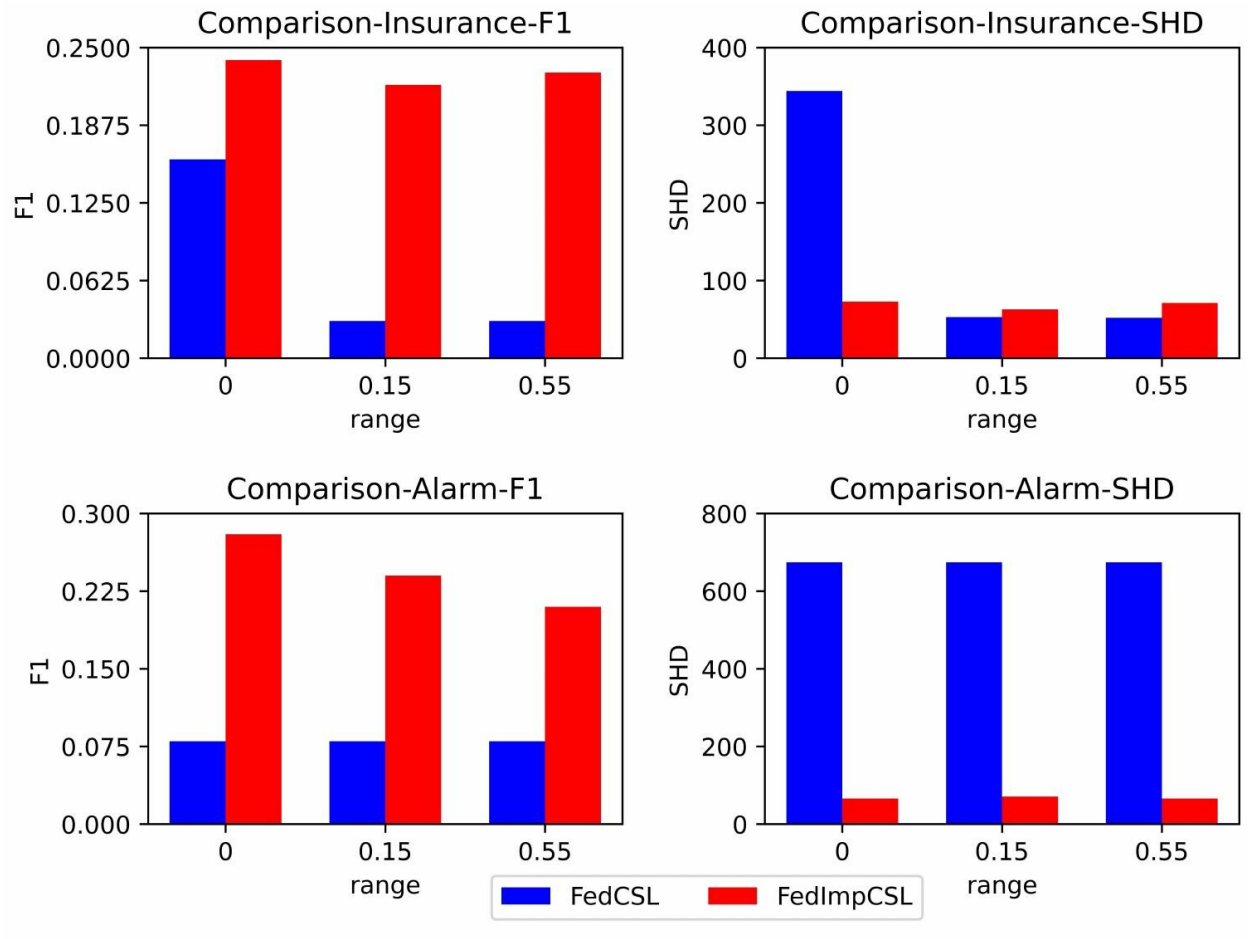


Figure 5: Comparing results of FedCSL and FedImpCSL on Insurance and alarm dataset with small(800) samples.

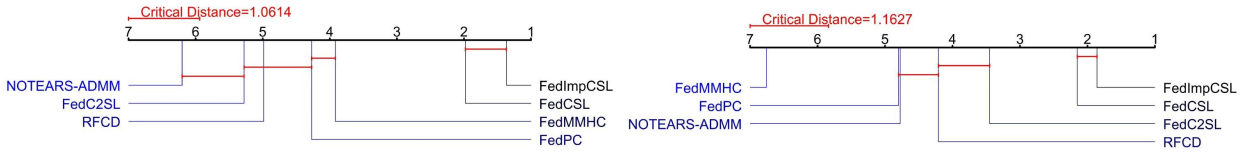


Figure 6: Crucial difference diagram of the Nemenyi test for F1 and SHD of federated causal structure learning algorithms

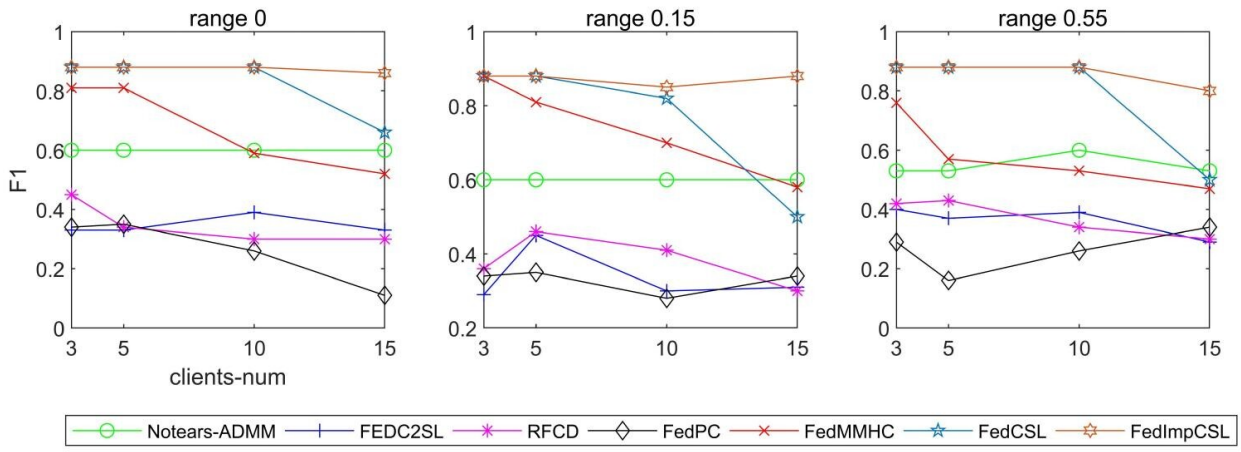


Figure 7: Comparing results on real-world dataset Sachs with 5000 samples under *F1*.

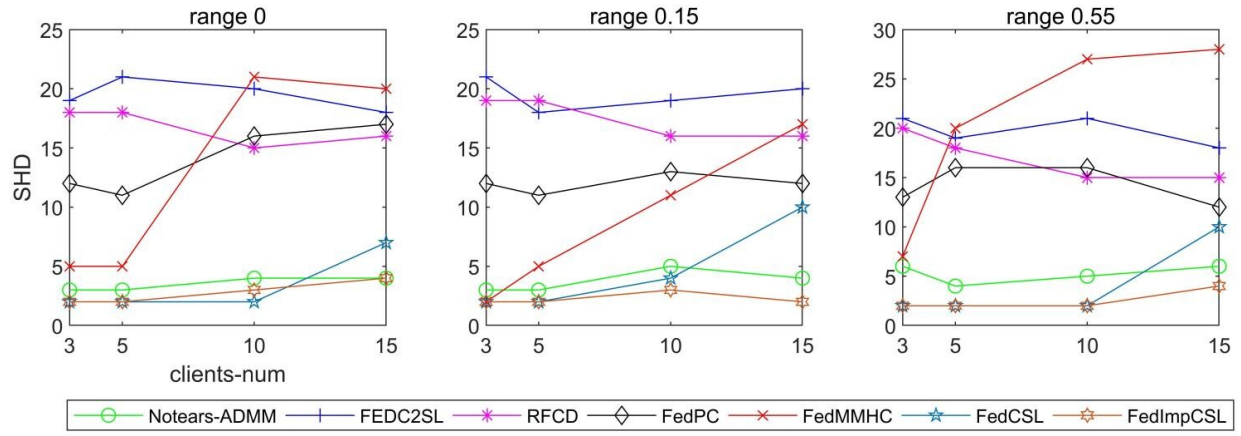


Figure 8: Comparing results on real-world dataset Sachs with 5000 samples under *SHD*.

synthetic dataset comprising 1010 nodes. The experimental setups include a sample size of 5000, the missing data rate is set to 0.15, and the number of clients are set to 3, 5, 10, and 15 for comparative purposes. As shown in Fig. 9, for both F1 and SHD values, when dealing with missing data in high-dimensional synthetic dataset, FedImpCSL outperforms the best baseline, FedCSL, across various client settings. This indicates the scalability of the FedImpCSL method in handling missing data in high-dimensional synthetic dataset.

5.5. Results on Non-IID synthetic datasets

In real-world federated settings, clients often exhibit heterogeneous data distributions. To validate the usability and robustness of FedImpCSL under Non-IID data conditions, we compare it with the state-of-the-art baseline method, FedCSL, which emphasizes scalability. Experiments are performed on Non-IID datasets consisting of 46 nodes. The experimental setup includes a sample size of 5,000, a missing data rate of 0.15, and varying client counts (3, 5, 10, and 15) for comparative analysis. Notably, to simulate usage across different distributional scenarios, we configure three distinct Dirichlet parameters: 0.5, 0.8, and 1. A smaller Dirichlet parameter indicates a greater degree of data distribution heterogeneity.

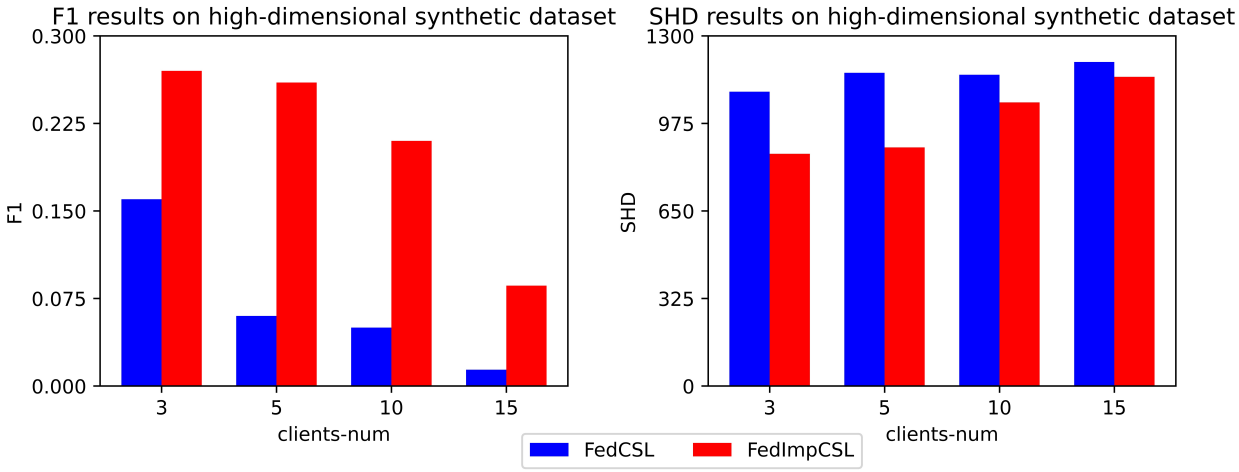


Figure 9: Comparing results of FedCSL and FedImpCSL on high-dimensional synthetic dataset.

As shown in Fig. 10, FedImpCSL outperforms FedCSL, in most cases across F1 and SHD metrics when performing federated causal structure learning on Non-IID datasets within a federated missing data environment. Notably, FedImpCSL achieves superior performance under a Dirichlet distribution parameter α=1 when operating with 10 or 15 clients. These findings not only validate the method's applicability to real-world federated causal learning scenarios but also demonstrate its capacity to achieve robust learning outcomes under heterogeneous data distributions.

5.6. Ablation study

To verify the effectiveness of the proposed algorithms, named FedLocalImp outlined in Algorithm 1, FedCAM outlined in Algorithm 2 and FedGlobalImp outlined in Algorithm 3, we conduct ablation experiments.

5.6.1. Missing algorithm with federated learning (FedLocalImp and FedGlobalImp)

We first validate the effectiveness of the FedLocalImp and FedGlobalImp (Algorithm 1 and Algorithm 3), named the overall procedure as FedLocal-GlobalImp. Specifically, we develop a variant of this method, denoted as FedLocal-GlobalImp_ICkNNI, which performs missing data processing at each client only once, utilizing a combination of ICkNNI imputation and deletion methods. For the ablation experiments, we use the Insurance dataset with the missing data rate range set at 15% and the number of clients set 3, 5, 10 and 15. FedLocal-GlobalImp (ours) is compared with FedLocal-GlobalImp_ICkNNI for the comparison experiments. The results are shown in Fig. 11. We observe that FedLocal-GlobalImp achieves higher F1 scores and lower SHD values than FedLocal-GlobalImp_ICkNNI on the benchmark BN dataset, demonstrating the effectiveness of our designed missing data processing method. In particular, the scenario where different clients hold samples with non-uniform missing data rates is specifically described in Section 5.7.

We evaluate the effectiveness of FedLocal-GlobalImp through a more intuitive experiment. We use the PCP metrics [24] for this purpose. PCP represents the percentage of correct prediction, as depicted in Eq. (12), and this metrics can accurately compare different missing data imputation methods. To validate the FedLocal-GlobalImp method for every client in the federation execution process, we conduct two groups of experiments where the number of samples held by the client is 500 and 1000, respectively, and the missing data rate is set in three situations: 0.05, 0.15 and 0.25. From the four BNs datasets shown in Fig. 12 and Fig. 13, we can observe the following results:

$$PCP = 100 \times \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions} \tag{12}$$

(1) The FedLocal-GlobalImp algorithm performs better compared to the ICkNNI imputation across different datasets, sample numbers and missing data rates.

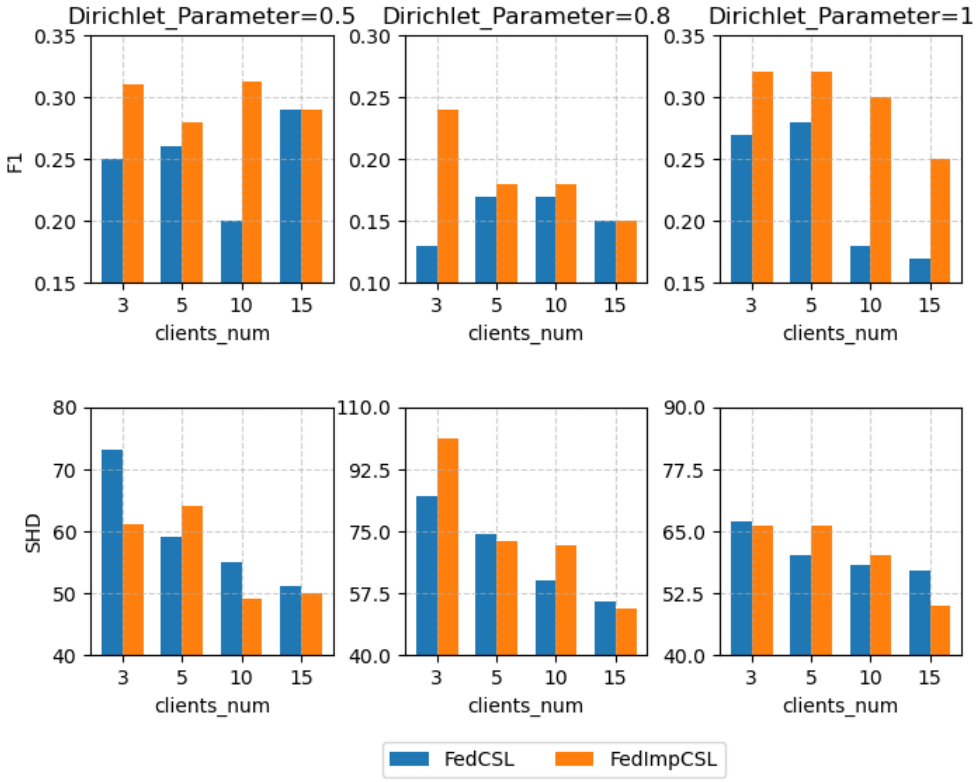


Figure 10: Comparing results of FedCSL and FedImpCSL on Non-IID datasets.

(2) The FedLocal-GlobalImp algorithm maintains a good performance in various scenarios, such as when dealing with a small sample number and a high missing data rate.

5.6.2. Client contribution assessment within weighted aggregation algorithm (FedCAM)

In this section, we primarily validate the effectiveness of the client contribution assessment method within weighted aggregation (i.e. FedCAM Algorithm 2). This is because the effectiveness of the weighted aggregation approach can be easily verified by comparing it to the aggregation skeleton without weight operation in FedPC. Specifically, the experiments are conducted by comparing the weight updating mechanism using weighted aggregation called FedCAM_no_weight_update, which does not conduct the iterative dynamic updating method. Using the Insurance dataset with the missing data rate range set 15% and the number of clients set 3, 5, 10, 15, the results presented in Fig. 14 demonstrate that the performance of the iterative updating method is significantly improved with the weighted aggregation.

5.7. Missing data rates partition

In this section, we verify the extensiveness of the range of missing data rates for clients. We report the experimental results on six BNs datasets, with the number of clients set to 15. We mainly discuss three cases: when the client missing data rates are the same, when the extreme difference is 0.15, and when the extreme difference is 0.55. The minimum missing data rate is 1% and the maximum reaches 55%, with 15% being the common scenario. Details for the other cases involving fifteen clients can be seen in Tables 1 and 2.

As shown in Fig. 15, we can observe that FedImpCSL is able to maintain better performance in all three situations and across all datasets. Especially in Alarm and Insurance, the best baseline, FedCSL, exhibits good performance

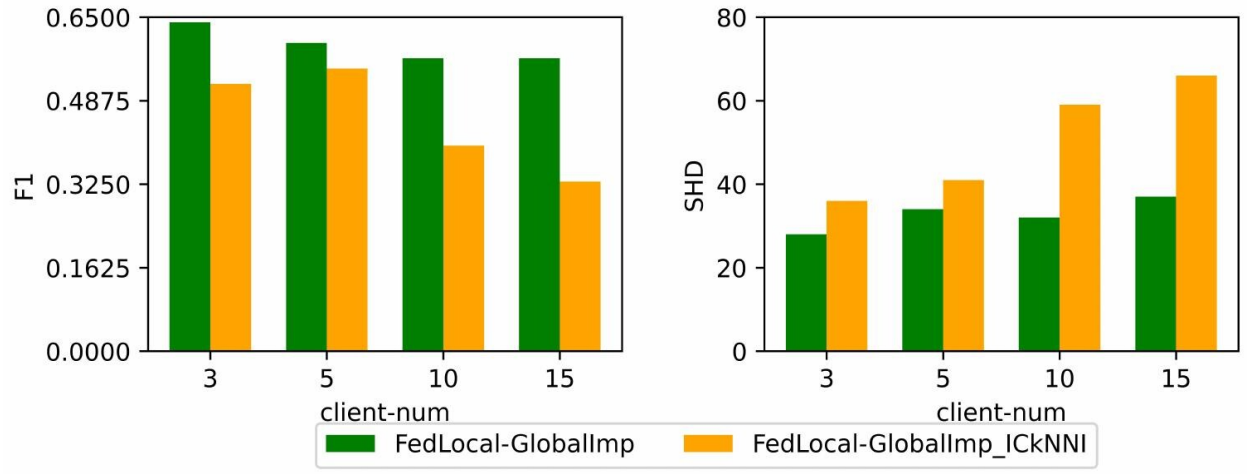


Figure 11: Comparing results of FedLocal-GlobalImp(ours) and FedLocal-GlobalImp_ICkNNI on Insurance dataset with 5000 samples under *F1* and *SHD*.

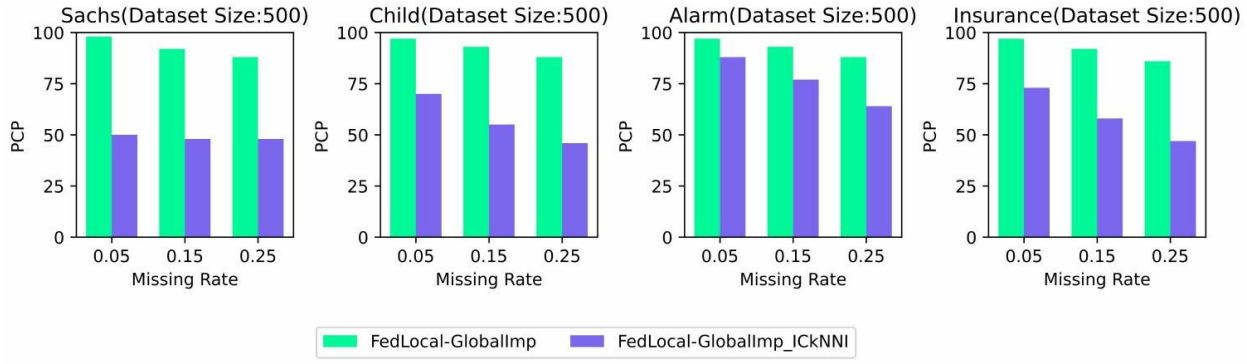


Figure 12: Comparing results of FedLocal-GlobalImp(ours) and ICkNNI on four datasets with 500 samples under *PCP*.

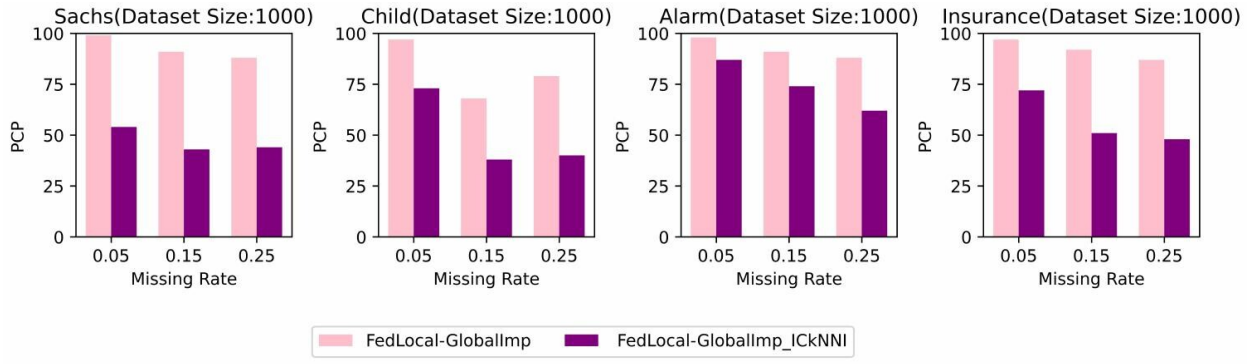


Figure 13: Comparing results of FedLocal-GlobalImp(ours) and ICkNNI on four datasets with 1000 samples under *PCP*.

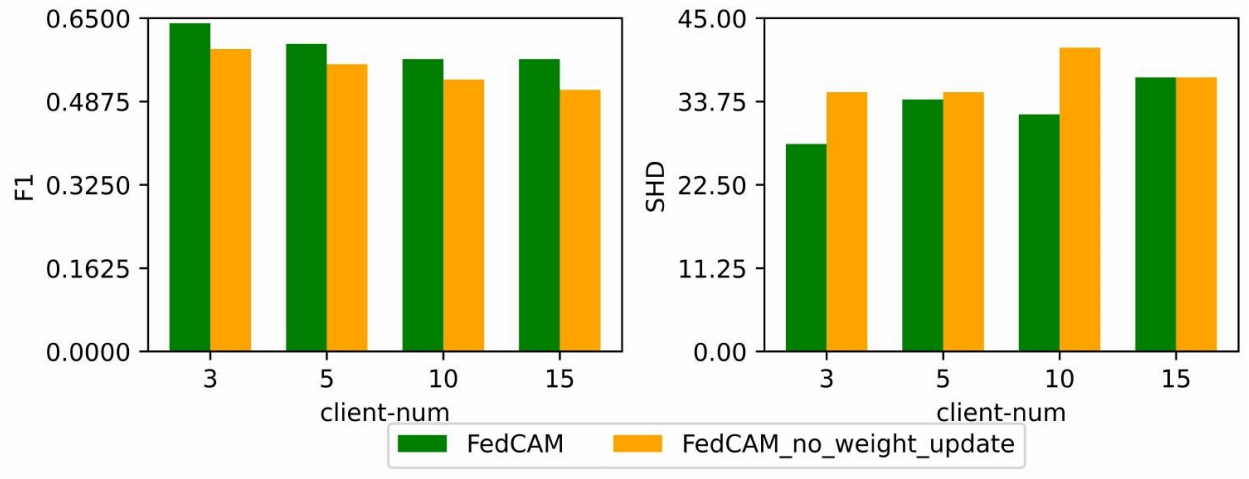


Figure 14: Comparing results of FedCAM(ours) and FedCAM_no_weight_update on Insurance dataset with 5000 samples under *F1* and *SHD*.

when the missing data rate range is set to 0 and 0.15. However, FedCSL cannot achieve a better value compared to FedImpCSL when the missing data rate range increases to 0.55. Similarly, other algorithms like FedMMHC and FedC^{2}SL face the same issue. In summary, the FedImpCSL algorithm not only performs better compared to six algorithms, regardless of the number of clients, but is also unaffected by the range of missing data rates among different clients.

The reasons for this observation are as follows:

(1) In the weighted aggregation stage, FedImpCSL considers the missing data as one of the weight indicators to measure the difference in data quality of the client through the missing data. This approach aims to mitigate such differences during weighted aggregation and obtain a more robust skeleton.

(2) For FedLocalImp, since each client executes a method to address missing data in the skeleton stage, FedLocalImp develops a tailored method for handling each client's unique missing data on the server side. This is achieved through the client skeleton contribution scores.

(3) For FedGlobalImp, the missing method is further optimized based on a reliable skeleton. This optimization enhances the accuracy of data imputation and reduces the lack of universality stemming from variations in data quality among clients.

5.8. Convergence analysis of FedImpCSL

This section investigates the convergence of FedImpCSL in both static and dynamic scenarios.

5.8.1. Static scenario analysis

FedImpCSL converges when the aggregation skeleton reaches uniformity. This uniformity is achieved when the skeleton of each individual client remains unchanged, and the client weights in the server-weighted aggregation also reach a stable value. Below, we conduct a theoretical convergence analysis for these two aspects respectively.

Theoretical Analysis 1: Convergence of individual client skeleton

Constraint-based causal structure learning methods, such as the PC algorithm, perform CI (conditional independence) tests by gradually expanding the conditioning set, with their convergence relying on the monotonic growth property of the conditioning set size. Specifically, the PC algorithm [9, 32] starts from a completely undirected graph and continuously performs edge pruning operations when the size of the conditioning set k satisfies $k \leq threshold$. After $k > k_{max}$, the edge set in the graph remains stable, indicating the convergence of the skeleton. Although the original PC algorithm lacks a rigorous proof of convergence, FedPC [9] achieves experimentally verified convergence by defining a conditioning set growth threshold τ (terminating iteration when $\Delta|E(l)| < \tau$, where $\Delta|E(l)|$ represents

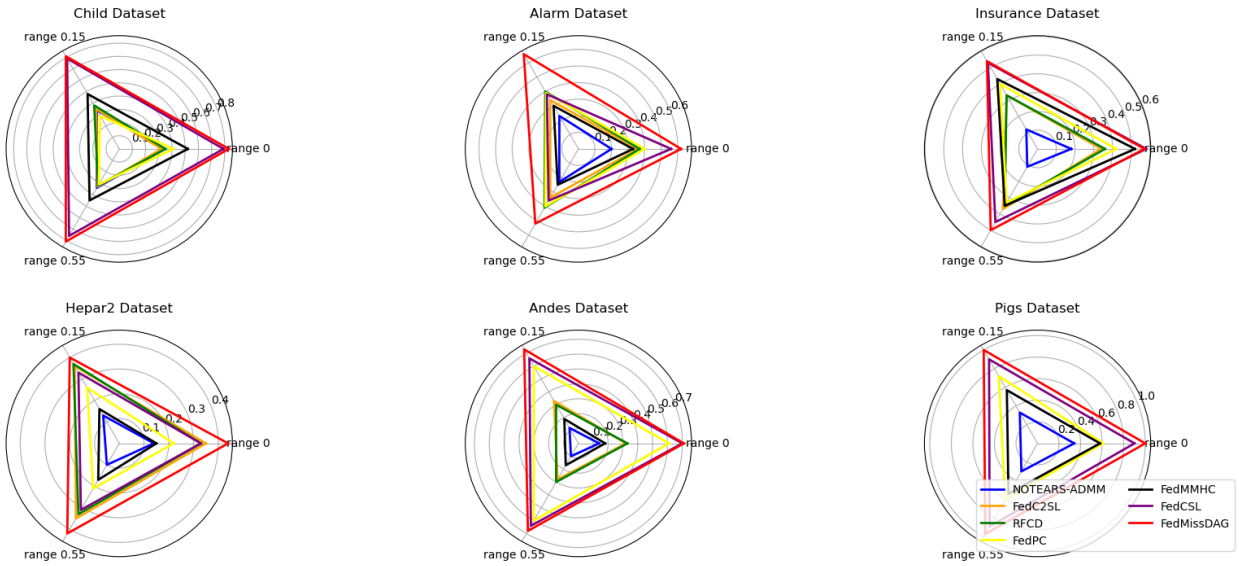


Figure 15: Comparison of federated structure casual learning with different missing data rates range.

the change in the number of edges at l-th iteration). FedImpCSL inherits this mechanism and guarantees the eventual convergence of the local skeletons for each client through the monotonic growth property of $k > k_{max}$ in federated iterations.

Theoretical Analysis 2: Convergence of client weights in server-weighted aggregation

As presented in Section 4.1.2, the calculation of weight $W^l_{c_k}$ is shown below.

$$W^l_{c_k} = \left(W^l_{n_{c_k}} * \alpha + W_{miss_{c_k}} * \beta\right) * H^{l-1}_{c_k} \tag{13}$$

where $W^l_{n_{c_k}}$ and $W_{miss_{c_k}}$ denote of c_k at l-th iteration and missing data rate of the client c_k respectively, and α and β are weighting coefficients.

For each client c_k, the missing data rate $W_{miss_{c_k}}$ remains constant over multiple iterations, and the sample size $W^l_{n_{c_k}}$ converges as the data after imputing missing values approaches the true data after multiple estimations. Thus, FedImpCSL ensures that the sequence of the client weights $\{W^l_{n_{c_k}}\}$ satisfies Eq. (14), suggesting that the change is stable and each client's contribution is guaranteed to converge to a stable point.

$$\sum_{l=1}^{\infty} \left| W^l_{c_k} - W_{c_k} \right| < \infty \tag{14}$$

Experimental verification

The previous two subsections present a theoretical convergence analysis of our proposed method. In addition to the theoretical analyses, we experimentally demonstrate the convergence of FedImpCSL on BN and synthetic datasets. As shown in Fig. 16(a), under the setting of 10 clients and a missing data rate of 0.15, the number of aggregated skeleton edges eventually stabilized. Moreover, Fig. 16(c) further confirms the convergence of the real-world Sachs dataset with 10 client weights. A comparison of the two subgraphs shows that the client weights are consistent with the convergence step of FedImpCSL, thereby providing further evidence for the convergence of the algorithm.

5.8.2. Dynamic scenario analysis

To further illustrate the scalability of FedImpCSL across diverse scenarios, this section analyzes its convergence under dynamic scenario. Practically, client data distributions often evolve over time due to factors like sample size and missing data rate. To rigorously validate the convergence of FedImpCSL in dynamic scenario, we present a

comprehensive theoretical analysis combining the empirical experiments. FedImpCSL guarantees convergence in dynamic environments through adaptive weight updates and a historical constraint.

Theoretical Analysis 1: Adaptive weight update

In real-world federated learning scenarios, client data characteristics (e.g., sample size, missing data rate) may evolve across iterations. When client c_k experiences distribution shifts, such as sample size changing from $W^l_{n_{c_k}}$ to $W^{l+1}_{n_{c_k}}$ or missing data rate varying from $W^l_{miss_{c_k}}$ to $W^{l+1}_{miss_{c_k}}$, the server dynamically adjusts aggregation weights accordingly Eq. (13).

This adaptive mechanism eliminates the need for causal structure re-initialization during data distribution shifts.

Theoretical Analysis 2: Historical weight constraint

FedImpCSL maintains convergence through the historical constraint $H^{l-1}_{c_k}$ ($0<H^{l-1}_{c_k} \leq 1$). The essence of convergence is that the model's update magnitude (i.e., the step size) approaches zero. This constraint stabilizes aggregation weight sums across iterations, preventing abrupt changes even as client data evolves. The global causal structure updates from a fully connected graph via:

$$DAG^{l+1} = \mathrm{DAG}^l - \sum_{k=1}^{m} skele^{l+1}_{c_k}(\nabla_{\mathrm{data}}) * W^{l+1}_{c_k}(\nabla_{\mathrm{data}}) \tag{15}$$

where $W^{l+1}_{c_k}(\nabla_{\mathrm{data}})$ follows Eq. (15). The constraint ensures $W(\nabla_{\mathrm{data}})$ stays between constants b and a, avoiding sudden fluctuations.

Consequently, $skele^{l+1}_{c_k}(\nabla_{\mathrm{data}}) * W^{l+1}_{c_k}(\nabla_{\mathrm{data}})$ remains bounded, making the total update magnitude $\sum_{k=1}^{m} skele^{l+1}_{c_k}(\nabla_{\mathrm{data}}) * W^{l+1}_{c_k}(\nabla_{\mathrm{data}})$ diminish over iterations. While minor fluctuations may occur, the overall update magnitude diminishes, ensuring eventual convergence to a stable causal structure.

Experimental verification

We validate FedImpCSL's convergence under dynamic sample size conditions through experimental simulation. Notably, sample size adjustments directly influence the missing data rate, as reduced sample size proportionally increase missingness percentages. The experiment simulates a scenario in which client sample size fluctuate dynamically. Specifically, in each iteration, each client randomly increases or decreases its sample size by 10%-30% (or combines both operations) before server aggregation.

Using a setup of 10 clients and a 15% missing data rate, we track the number of skeletal edges in the aggregated causal graph across iterations. As shown in Fig. 16(b), despite dynamic sample size adjustments, the number of edges in the aggregated causal graph may fluctuate initially but stabilizes progressively. This stabilization provides empirical evidence for algorithmic convergence, even with sample size fluctuations within the 10%-30% range.

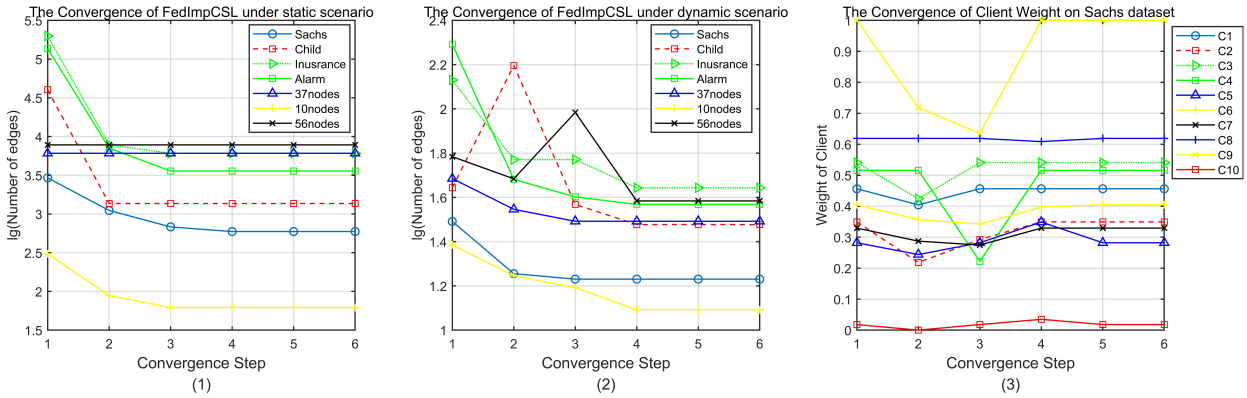


Figure 16: The convergence of FedImpCSL.

5.9. Analysis of the deletion ratio for FedLocalImp and FedGlobalImp methods

For FedLocalImp (see Line 7 in Algorithm 1) and FedGlobalImp (see Line 10 in Algorithm 3), a threshold ratio of 50% is adopted to determine whether a sample should be deleted or retained from the dataset of each client. An

excessively, low ratio may lead to the deletion of a large number of samples in the dataset, causing a sudden drop in the sample size and widening the gap between the modified and original datasets. Conversely, a ratio that is too high may retain non-informative samples, thereby diminishing the accuracy of data imputation.

To further analyze the influence of the threshold, we conduct experiments with ratios ranging from 10% to 100% on seven datasets, which includes four benchmark BN datasets (Sachs, Child, Insurance, Alarm) and three synthetic datasets (10nodes, 37nodes, 56nodes). The accuracy of data imputation is assessed using the PCP values calculated according to Eq. (12). The PCP values obtained at different ratios for different datasets are presented in Fig. 17.

As shown in Fig. 17, as the deletion ratio for FedLocalImp increases, the PCP values show an upward trend and generally stabilizes at 50%. Similarly, for FedGlobalImp, when the deletion ratio reaches 50% and 80%, the PCP values are relatively higher compared with other ratios, with the peaks of each dataset beng more pronounced at 50% across all datasets. These experimental results indicate that setting the threshold is set to 50% yields higher imputation accuracies for both FedLocalImp and FedGlobalImp.

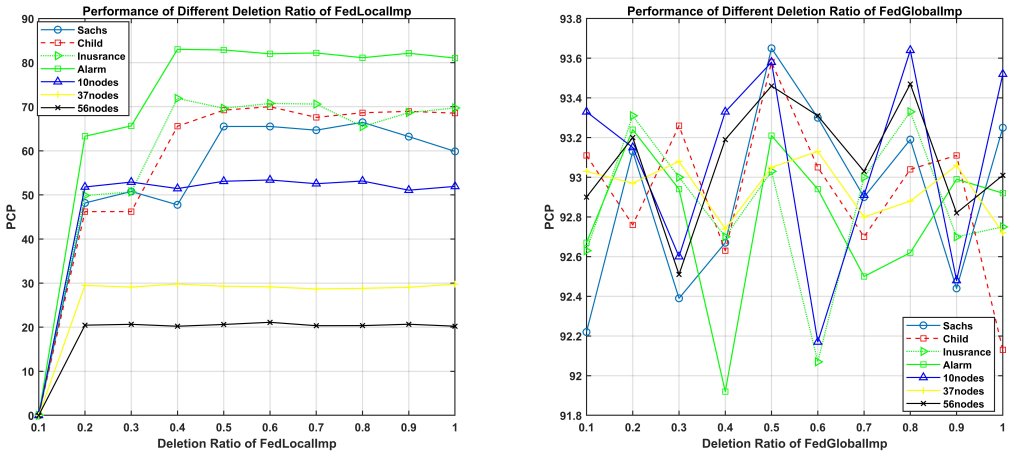


Figure 17: Performance of different deletion ratio for the FedLocalImp and the FedGlobalImp method.

6. Conclusions

In this study, we propose a novel processing method based on federated CSL in scenarios with missing data. Additionally, we incorporate a contribution assessment algorithm within the weighted aggregation to enhance accuracy. Broadly, both these important mechanisms are designed to effectively address two non-negligible problems in the real world: missing data and client diversity. Compared to the six state-of-the-art methods, our method achieve good performance across various datasets, missing data rates, and sample sizes. The applicability of our approach under heterogeneous and unstructured data such as text, images, and multi-modal data is currently uncertain; therefore, future work will focus on federated CSL in scenarios with missing data, specifically targeting heterogeneous data and mixed data structures. In the future, we intend to evaluate the proposed method using a variety of unstructured data (e.g., text, images, multi-modal data) to further assess its robustness and effectiveness. Simultaneously, we will explore the integration of federated CSL with deep representations, aiming to develop a novel method that can directly and efficiently process and analyze unstructured data.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62376087, 62506174), in part by the Anhui Province Natural Science Foundation of Educational Commission (2022AH051099, 2023AH051600, 2023AH040216), in part by the Research Startup Fund of Chuzhou University (2024qd10).

References

[1] N. K. Kitson, A. C. Constantinou, Z. Guo, Y. Liu, K. Chobtham, A survey of bayesian network structure learning, Artificial Intelligence Review 56 (8) (2023) 8721–8814.

[2] C. Glymour, R. Scheines, P. Spirtes, Discovering causal structure: Artificial intelligence, philosophy of science, and statistical modeling, Academic Press, New York, 2014.

[3] C. Schmidt, J. Huegle, M. Uflacker, Order-independent constraint-based causal structure learning for gaussian distribution models using gpus, in: Proceedings of the 30th International Conference on Scientific and Statistical Database Management, 2018, pp. 1–10.

[4] Chickering, D. Maxwell, Optimal structure identification with greedy search, Journal of Machine Learning Research 3 (2002) 507–554.

[5] X. Zheng, B. Aragam, P. K. Ravikumar, E. P. Xing, Dags with no tears: Continuous optimization for structure learning, Advances in Neural Information Processing Systems 31 (2018) 9492–9503.

[6] I. Ng, K. Zhang, Towards federated bayesian network structure learning with continuous optimization, in: International Conference on Artificial Intelligence and Statistics, 2022, pp. 8095–8111.

[7] O. Mian, D. Kaltenpoth, M. Kamp, J. Vreeken, Nothing but regrets-privacy-preserving federated causal discovery, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2023, pp. 8263–8278.

[8] Z. Wang, P. Ma, S. Wang, Towards practical federated causal structure learning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2023, pp. 351–367.

[9] J. Huang, X. Guo, K. Yu, F. Cao, J. Liang, Towards privacy-aware causal structure learning in federated setting, IEEE Transactions on Big Data 9 (6) (2023) 1525–1535.

[10] X. Guo, K. Yu, L. Liu, J. Li, Fedcsl: A scalable and accurate approach to federated causal structure learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 12235–12243.

[11] T. Burr, Causation, prediction, and search, Technometrics 45 (2003) 272–273.

[12] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, Y. Gao, A survey on federated learning, Knowledge-Based Systems 216 (2021) 106775.

[13] Y. Sun, J. Li, Y. Xu, T. Zhang, X. Wang, Deep learning versus conventional methods for missing data imputation: A review and comparative study, Expert Systems with Applications 227 (2023) 120201.

[14] J. Van Hulse, T. M. Khoshgoftaar, Incomplete-case nearest neighbor imputation in software measurement data, Information Sciences 259 (2014) 596–610.

[15] T. Nishio, R. Yonetani, Client selection for federated learning with heterogeneous resources in mobile edge, in: ICC 2019 IEEE International Conference on Communications, IEEE, 2019, pp. 1–7.

[16] S. Moreno-Álvarez, M. E. Paoletti, A. J. Sanchez-Fernandez, J. A. Rico-Gallego, L. Han, J. M. Haut, Federated learning meets remote sensing, Expert Systems with Applications 255 (2024) 124583.

[17] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Transactions on Intelligent Systems and Technology 10 (2) (2019) 1–19.

[18] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends® in Machine Learning 3 (1) (2011) 1–122.

[19] E. Gao, J. Chen, L. Shen, T. Liu, M. Gong, H. Bondell, Feddag: Federated dag structure learning, Transactions on Machine Learning Research 2023 (2023).

[20] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, X. D. Koutsoukos, Local causal and markov blanket induction for causal discovery and feature selection for classification part I: algorithms and empirical evaluation, Journal of Machine Learning Research 11 (1) (2010) 171–234.

[21] R. Tu, C. Zhang, P. Ackermann, K. Mohan, H. Kjellström, K. Zhang, Causal discovery in the presence of missing data, in: the 22nd International Conference on Artificial Intelligence and Statistics, 2019, pp. 1762–1770.

[22] R. Foraita, J. Friemel, K. GÃijnther, T. Behrens, J. Bullerdiek, R. Nimzyk, W. Ahrens, V. Didelez, Causal discovery of gene regulation with incomplete data, Journal of the Royal Statistical Society Series A: Statistics in Society 183 (2020) 1747–1775.

[23] E. Gao, I. Ng, M. Gong, L. Shen, W. Huang, T. Liu, K. Zhang, H. Bondell, Missdag: Causal discovery in the presence of missing data with continuous additive noise models, Advances in Neural Information Processing Systems 35 (2022) 5024–5038.

[24] S. Sheng, X. Guo, K. Yu, X. Wu, Local causal structure learning with missing data, Expert Systems with Applications 238 (2024) 121831.

[25] E. Gao, J. Chen, L. Shen, T. Liu, M. Gong, H. D. Bondell, Feddag: Federated DAG structure learning, Transactions on Machine Learning Research 2023 (2023) 1–36.

[26] J. A. Aslam, E. Yilmaz, V. Pavlu, The maximum entropy method for analyzing retrieval measures, in: Proceedings of The 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2005, pp. 27–34.

[27] J. A. Gámez, J. L. Mateo, J. M. Puerta, Learning bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood, Data Mining and Knowledge Discovery 22 (2011) 106–148.

[28] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, G. P. Nolan, Causal protein-signaling networks derived from multiparameter single-cell data, Science 308 (5721) (2005) 523–529.

[29] X. Guo, K. Yu, H. Wang, L. Cui, H. Yu, X. Li, Sample quality heterogeneity-aware federated causal discovery through adaptive variable space selection, Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence 2024 (2024) 4071–4079.

[30] D. Kalainathan, O. Goudet, R. Dutta, Causal discovery toolbox: Uncovering causal relationships in python, Journal of Machine Learning Research 21 (37) (2020) 1–5.

[31] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[32] H. Li, V. Cabeli, N. Sella, H. Isambert, Constraint-based causal structure learning with consistent separating sets, Advances in neural information processing systems 32 (2019) 14257–14266.