Supplementary Material for "Towards Privacy-Aware Causal Structure Learning in Federated Setting"

Jianli Huang†, Xianjie Guo†, Kui Yu*, Fuyuan Cao, and Jiye Liang

TABLE 1
Details of five benchmark Bayesian networks.

network	Number of variables	Number of edges	Maximum in/out-degree
alarm	37	46	4/5
insurance	27	52	3/7
win95pts	76	112	7/10
andes	223	338	6/12
pigs	441	592	2/39

S-1: Detailed experimental results.

S-1-1: Experiment settings.

We implement the typical federated setting where each client owns its local data and don't communicate with other clients. Then clients transmits/receives only model parameter to/from the central server.

Datasets.

The datasets used in the experiments include synthetic and real datasets as follows.

We assume that there are K data samples in a dataset and N clients exists, and N lies in $\{3, 5, 10, 15\}$. To introduce unevenness, we randomly assigned the data samples to each client while ensuring that each client contains at least $\frac{K}{N*2}$ data samples. This approach aims to ensure that the data distribution is not heavily skewed towards any specific client.

- **Synthetic datasets.** We conduct FedPC on linear and nonlinear datasets. For the linear synthetic datasets, we generated five continuous datasets using an open-source toolkit [1] with the number of variables to 10, 20, 50, 100 and 300 respectively. Each synthetic dataset contains 5000 continuous samples. The generative process employed a linear causal mechanism represented as follows:

$$y = \mathbf{X}W + \times E, \tag{1}$$

where $+\times$ denotes either addition or multiplication, $\mathbf{X}$ denotes the vector of causes, and E represents the noise variable accounting for all unobserved variables. For the nonlinear synthetic datasets, the causal mechanism used in the generative process is Gaussian Process (GP), and the mechanisms are represented as:

$$y = GP(\mathbf{X}) + \times E. \tag{2}$$

The proportion of noise in the mechanisms is set to 0.4. Gaussian noise was used in the generative process. In our experiments on nonlinear datasets, we utilized the Kernel-based Conditional Independence test (KCI-test) [2] instead of Fisher's Z Conditional Independence test. This was done to achieve better performance in detecting nonlinear dependencies in the data.

- **Benchmark Bayesian network (BN) datasets.** We use five benchmark BNs, alarm, insurance, win95pts, andes and pigs, to generate five discrete datasets, respectively. Each dataset contained 5000 samples. The details of the five benchmark BNs are presented in Table 1.
- **Real dataset.** We also compare FedPC with its rivals on a real biological dataset with 853 samples, Sachs [3]. Sachs is a protein signaling network expressing the level of different proteins and phospholipids in human cells. It is commonly viewed as a benchmark graphical model with 11 nodes (cell types) and 17 edges. Same as above, data samples are distributed unevenly across N clients ($N \in \{3, 5, 10, 15\}$), and the number of samples on each client is randomly generated and is at least $\frac{N}{K*2}$.

- † *represents equal contribution.*
- *Jianli Huang, Xianjie Guo, and Kui Yu are with the Intelligent Interconnected Systems Laboratory of Anhui Province and the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China (e-mail: janelee@mail.hfut.edu.cn, xianjieguo@mail.hfut.edu.cn, yukui@hfut.edu.cn). (*Corresponding author: Kui Yu).*
- *Fuyuan Cao and Jiye Liang are with the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China (e-mail: {cfy, ljy}@sxu.edu.cn).*

Metrics.

To evaluate the performance of FedPC, we use the following frequently used metrics in DAG learning.

- Reverse: Reverse is the number of edges with wrong directions in the DAG learnt by an algorithm against the true DAG.
- Extra: Extra is the number of extra edges in the DAG learnt by an algorithm against the true DAG.
- Miss: Miss is the number of missing edges in the DAG learnt by an algorithm against the true DAG.
- SHD (Structural Hamming Distance): The value of SHD is calculated by comparing the learnt causal structure with the true causal structure. Specifically, the value of SHD is the sum of undirected edges, reverse edges, missing edges and extra edges.
- TPR: TPR is also called recall, which refers to the probability that an actual positive will test positive. TPR is calculated as: $TPR = \frac{TP}{TP+FN}$
- FDR: FDR is the expected ratio of the number of false positive classifications to the total number of positive classifications: $FDR = \frac{FP}{TP+FP}$
- Precision: Precision denotes the number of true positives in the output divided by the number of features in the output of the algorithm. Precision is calculated as: $Precision = \frac{TP}{TP+FP}$

FedPC utilizes the PC algorithm to learn the skeleton and orient the undirected edges in the causal graph. Therefore, the output of FedPC is a CPDAG, which contains both directed and undirected edges. We employ the acyclic constraint technology proposed in reference [4] to transform CPDAGs returned by FedPC into DAGs, and then calculated reverse, extra, miss, shd, TPR, FDR and precision scores of these DAGs. In Tables 2-12, the symbol "-" denotes that an algorithm does not produce results on the corresponding datasets when the running time of the algorithm exceeded 72 hours or there is no enough memory space.

Baselines and rivals.

FedPC is compared with 8 rivals. The comparison methods are as follows:

(1) NOTEARS-Avg. We run the NOTEARS [5] algorithm at each client independently, then calculate the averaging results of SHD and F1 of all learnt DAGs as the final results.

(2) NOTEARS-ADMM. We run the NOTEARS-ADMM algorithm [6], and then calculate the SHD and F1 of the learnt DAG as the final results.

(3) FedDAG. We run the FedDAG [7] algorithm and then calculate the SHD and F1 of the learnt DAG as the final results.

(4) PC-Avg. We first run the PC algorithm at each client independently for obtaining N DAGs (N is the number of clients), and then calculate the averaging SHD values and F1 values of all learnt DAGs as the final results of PC-Avg.

(5) PC-Best. We first run the PC algorithm at each client independently to get N DAGs, and then select the DAG with the lowest SHD value as the final output.

(6) PC-All. We centralize all clients data to a single dataset and run the PC algorithm on it.

(7) FedPC-Simple-I. We run the PC algorithm at each client independently to learn the DAGs, then aggregate all learnt DAGs at the server by the strategy that if more than 30% (the same ratio as our method) of the learnt DAGs contain a directed edge between two variables, this edge is kept in the final DAG.

(8) FedPC-Simple-II. We run the PC algorithm to learn the skeletons independently at each client, then aggregate all learnt skeletons at the server by the strategy that if an undirected edge between two variables exists on more than 30% of the skeletons, this edge will be kept in the final skeleton. Then we take the intersection of the separation sets between two variables learnt from each client to learn v-structures. This is an ablation study of our proposed algorithm, by removing the layer-wise strategy of the FedOrien subroutine.

Implementation details.

All experiments were conducted on a computer with Intel Core i9-10900F 2.80-GHz CPU and 32-GB memory. The significance level for CI tests is set to 0.01. For PC, NOTEARS, and NOTEARS-ADMM, we used the source codes provided by their authors. NOTEARS-Avg and NOTEARS-ADMM use 0.3 as the threshold to prune edges in a DAG and FedDAG uses 0.5 as the threshold, those are the same as the original paper.

S-1-2: Experiment results on benchmark data.

Structural errors.

Table 2 to 11 show the reverse, extra, miss, SHD values of FedPC and its rivals using five benchmark BN datasets and five continuous synthetic datasets, respectively.

- Generally, we can see that FedPC achieves a lower SHD than its rivals, which indicates the superiority of our method. The reason is as follows: The excellent performance of FedPC relies on correctly learnt skeleton structures, because the layer-wise aggregation strategy enables FedPC to reduce the number of miss edges. And simultaneously the consistent separation set identification strategy further lessens the number of reverse edges.
- For all datasets, NOTEARS-Avg and NOTEARS-ADMM have similar performance. But we notice that NOTEARS-Avg and NOTEARS-ADMM perform good at reverse and extra edges on almost all datasets. The explanation is that the size of the learnt DAGs is much smaller than others, leading that NOTEARS-Avg and NOTEARS-ADMM miss many true edges (i.e. having higher miss values and higher SHD values). So the final DAGs exist a small amount of reverse and extra edges.
- It can be seen that the quality of the DAGs learnt by FedPC-Simple-I is competitive with that learnt by and FedPC-Simple-II. Inaccurate orientation leads that FedPC-Simple-I and FedPC-Simple-II are inferior in reverse. FedPC-Simple-I orients edges by simply using a voting scheme to determine the orientation. FedPC-Simple-II directly employs the intersection-rule to construct the separation set. As a result, FedPC-Simple-II avoids learning many correctly oriented edges in the DAG.

- The DAGs learnt by PC-Avg and PC-Best have more extra and reverse edges than those learnt by FedPC, which indicates that FedPC finds more accurate skeletons than PC-Avg and PC-Best. An explanation might be that PC-Avg and PC-Best do not exchange information between clients while FedPC exchanges its learnt skeleton with other clients at each layer. This verifies the effectiveness of FedPC and indicates that exchanging information during the skeleton learning process is a key to learning an accurate DAG, as is the case for FedPC.

Structural correctness.

Through the metrics of structural correctness, i.e. TPR, FDR and precision, Table 2 to 11 report the quality of DAG learnt by different algorithms on five benchmark BN datasets and five continuous synthetic datasets, respectively. We find that on most datasets, FedPC not only achieves fewer structural errors than its rivals, bust also achieves more structural correctness than other algorithms on almost all datasets.

- Compared with its rivals, FedPC obtains highest values of TPR on most benchmark datasets and most synthetic datasets. Specifically, for TPR metric, our method achieves clear improvements of approximately 58% more than NOTEARS-ADMM on alarm with 5000 samples when the number of clients is 3, 50% more than NOTEARS-Avg on insurance with 5000 samples when the number of clients is 10, 10% more than FedPC-Simple-I, PC-Best, PC-Avg on win95pts with 5000 samples when the number of clients is 15. The reason is that FedPC adopts the layer-wise skeleton learning strategy to construct more accurate skeleton, that is, some missed edges are restored and some extra edges are removed.
- For precision metric, our method achieves clear improvements on high-dimensional data, the explanation may be that FedPC gets more correct edges in the learnt DAG. And the novel method for separation sets employed by our method, which can recover some missed true directed edges, thus FedPC achieves a higher precision value.
- FedPC obtains lowest values of FDR on most benchmark datasets and most synthetic datasets. FDR measures the number of false positive items indicating that there exiting an edge in the predicted DAG but not in the true DAG. FedPC, employing a novel strategy to find out the true separation set, removes some redundant false directed edges.
- PC-Avg and PC-Best have little improvement on TPR, FDR and precision can be explained that PC-Avg and PC-Best do not exchange information between clients, they learns more wrong edges because of insufficient data samples. But the layer-wise aggregation strategy makes FedPC exchange enough information between clients for accurate DAG learning. This further verifies the effectiveness of the two strategies of FedPC.
- In addition, we find that the performance of continuous optimization methods (such as NOTEARS-Avg and NOTEARS-ADMM) is generally worse than that of combinatorial optimization methods (such as PC-Avg, PC-Best, FedPC-Simple-I, FedPC-Simple-II) on most benchmark datasets in terms of TPR, FDR and precision. This is because that the causal structures learnt by continuous optimization methods contain many extra edges and reverse edges. More specifically, the TPR is greatly reduced and the FDR is significantly promoted.
- FedPC-Simple-I and FedPC-Simple-II performs worse compared with PC-Avg and PC-Best. In particular, FedPC-Simple-I and FedPC-Simple-II is not suitable to datasets with large in/out-degrees,such as Pigs. The explanation is that FedPC-Simple-I and FedPC-Simple-II lack a efficient way to identify the correct separation set when orienting edges. And because of the decreased scale of data samples, CI tests become inaccurate, leading the performance of PC-Avg and PC-Best .

S-1-3: Experiment results on nonlinear data.

In order to evaluate the performance of FedPC on nonlinear datasets, we conducted additional experiments using synthetic data generated with nonlinear causal mechanism. Specifically, we used Kernel-based Conditional Independence (KCI) tests [2] instead of Fisher's Z Conditional Independence test to achieve ideal performance in the nonlinear setting. The KCI tests have been shown to be effective in detecting nonlinear causal relationships in previous studies.

We evaluated the performance of FedPC on the nonlinear datasets using metrics such as Structural Hamming Distance (SHD), True Positive Rate (TPR), precision, and F1 score. Due to the slow running time of the KCI test, we conducted experiments on causal graphs with 10 and 20 nodes only. To provide a comprehensive comparison, we also included FedDAG and NOTEARS-ADMM in the comparison. Both FedDAG and NOTEARS-ADMM have been shown to support nonlinear relationships [6] [7], making them suitable for comparison with FedPC in the nonlinear setting.

The results of the experiments are shown in Table 13. As can be seen from the table, FedPC achieved the lowest SHD and highest F1 score among all three algorithms, demonstrating its effectiveness in handling nonlinear datasets.

The experiments show that FedPC is not only effective in handling linear datasets, but also performs well on nonlinear datasets, making it a versatile algorithm for federated learning.

S-1-4: Experiment results on real data.

To compare the performance of our proposed framework on a real dataset, we consider a real bioinformatics dataset Sachs. Sachs is a protein signaling network expressing the level of different proteins and phospholipids in human cells. It is commonly viewed as a benchmark graphical model with 11 nodes and 17 edges. In our experiments, we adopt the observational data with 853 samples.

Among all methods in the experiments, as shown in Table 12, the performance of FedPC-Simple-I, FedPC-Simple-II, PC-Avg and PC-Best are competitive, but FedPC achieves

TABLE 2
Details of benchmark Bayesian network: alarm

Number of clients	Algorithm	Reverse(↓)	Extra(↓)	Miss(↓)	SHD(↓)	TPR (↑)	FDR(↓)	Precision(↑)
3	PC-Avg	9	9.667	4	22.667	0.717	0.361	0.639
	PC-Best	9	9	**3**	21	0.739	0.346	0.654
	FedPC-Simple-I	5	**0**	4	9	0.804	0.119	0.881
	FedPC-Simple-II	8	**0**	4	12	0.739	0.190	0.810
	NOTEARS-Avg	**3**	4	36	43	0.152	0.500	0.500
	NOTEARS-ADMM	5	17	28	50	0.283	0.629	0.371
	FedDAG	46	46	0	92	**1.000**	0.667	0.333
	FedPC	**3**	**0**	**3**	**6**	0.870	**0.070**	**0.930**
5	PC-Avg	9.4	13.8	7.2	30.4	0.639	0.430	0.570
	PC-Best	7	10	**6**	23	0.717	0.340	0.660
	FedPC-Simple-I	7	**4**	**6**	**17**	0.717	**0.250**	**0.750**
	FedPC-Simple-II	14	**4**	**6**	24	0.565	0.409	0.591
	NOTEARS-Avg	**3**	**4**	37	44	0.130	0.538	0.462
	NOTEARS-ADMM	4	13	27	44	0.326	0.531	0.469
	FedDAG	46	46	0	92	**1.000**	0.667	0.333
	FedPC	7	7	**6**	20	0.717	0.298	0.702
10	PC-Avg	10.3	26.3	8.2	44.8	0.598	0.535	0.465
	PC-Best	10	9	7	26	0.630	0.396	0.604
	FedPC-Simple-I	11	**7**	9	27	0.565	0.409	0.591
	FedPC-Simple-II	15	**7**	9	31	0.478	0.500	0.500
	NOTEARS-Avg	5	5	36	46	0.109	0.667	0.333
	NOTEARS-ADMM	**4**	10	28	42	0.304	0.500	0.500
	FedDAG	46	46	0	92	**1.000**	0.667	0.333
	FedPC	10	**7**	**6**	**23**	0.652	**0.362**	**0.638**
15	PC-Avg	11.2	49.133	9.933	70.267	0.541	0.685	0.315
	PC-Best	10	24	**7**	41	0.630	0.540	0.460
	FedPC-Simple-I	10	7	9	26	0.587	0.386	0.614
	FedPC-Simple-II	13	7	9	29	0.522	0.455	0.545
	NOTEARS-Avg	5	**4**	36	45	0.109	0.643	0.357
	NOTEARS-ADMM	**4**	12	29	45	0.283	0.552	0.448
	FedDAG	46	46	0	92	**1.000**	0.667	0.333
	FedPC	6	7	9	**22**	0.674	**0.295**	**0.705**

TABLE 3
Details of benchmark Bayesian network: insurance

Number of clients	Algorithm	Reverse(↓)	Extra(↓)	Miss(↓)	SHD(↓)	TPR (↑)	FDR(↓)	Precision(↑)
3	PC-Avg	7.333	6.333	17.667	31.333	0.519	0.335	0.665
	PC-Best	9	4	**16**	29	0.519	0.325	0.675
	FedPC-Simple-I	6	**0**	19	25	0.519	0.182	0.818
	FedPC-Simple-II	8	**0**	19	27	0.481	0.242	0.758
	NOTEARS-Avg	**3**	2	44	49	0.096	0.500	0.500
	NOTEARS-ADMM	6	31	35	72	0.212	0.771	0.229
	FedDAG	52	52	0	104	**1.000**	0.667	0.333
	FedPC	6	**0**	18	**24**	0.538	**0.176**	**0.824**
5	PC-Avg	7.6	12.8	19.2	39.6	0.485	0.445	0.555
	PC-Best	6	10	**19**	35	0.519	0.372	0.628
	FedPC-Simple-I	5	2	21	28	0.500	0.212	0.788
	FedPC-Simple-II	11	**1**	21	33	0.385	0.375	0.625
	NOTEARS-Avg	**3**	2	44	49	0.096	0.500	0.500
	NOTEARS-ADMM	5	27	35	67	0.231	0.727	0.273
	FedDAG	52	52	0	104	**1.000**	0.667	0.333
	FedPC	**3**	2	21	**26**	0.538	**0.152**	**0.848**
10	PC-Avg	10.5	26.3	18.9	55.7	0.435	0.594	0.406
	PC-Best	6	11	**19**	36	0.519	0.386	0.614
	FedPC-Simple-I	8	8	22	38	0.423	0.421	0.579
	FedPC-Simple-II	9	7	22	38	0.404	0.432	0.568
	NOTEARS-Avg	**3**	**2**	44	49	0.096	0.500	0.500
	NOTEARS-ADMM	5	20	39	64	0.154	0.758	0.242
	FedDAG	52	52	0	104	**1.000**	0.667	0.333
	FedPC	4	7	22	**33**	0.500	**0.297**	**0.703**
15	PC-Avg	12.867	47.133	17.867	77.867	0.409	0.728	0.272
	PC-Best	12	19	**19**	50	0.404	0.596	0.404
	FedPC-Simple-I	11	13	21	45	0.385	0.545	0.455
	FedPC-Simple-II	11	8	23	42	0.346	0.514	0.486
	NOTEARS-Avg	**3**	**1**	44	48	0.096	0.444	0.556
	NOTEARS-ADMM	5	16	39	60	0.154	0.724	0.276
	FedDAG	52	52	0	104	**1.000**	0.667	0.333
	FedPC	7	9	22	**38**	0.442	**0.410**	**0.590**

TABLE 4
Details of benchmark Bayesian network: win95pts

Number of clients	Algorithm	Reverse(↓)	Extra(↓)	Miss(↓)	SHD(↓)	TPR (↑)	FDR(↓)	Precision(↑)
3	PC-Avg	4.333	2.333	59.667	66.333	0.429	0.123	0.877
	PC-Best	5	1	54	60	0.473	0.102	0.898
	FedPC-Simple-I	5	6	**46**	57	0.545	0.153	0.847
	FedPC-Simple-II	5	6	**46**	57	0.545	0.153	0.847
	NOTEARS-Avg	**2**	**0**	110	112	0.000	1.000	0.000
	NOTEARS-ADMM	4	5	94	103	0.125	0.391	0.609
	FedDAG	-	-	-	-	-	-	-
	FedPC	**2**	5	**46**	**53**	**0.571**	**0.099**	**0.901**
5	PC-Avg	5.2	3.4	67.2	75.8	0.354	0.182	0.818
	PC-Best	5	2	**59**	**66**	**0.429**	0.127	0.873
	FedPC-Simple-I	2	3	62	67	**0.429**	0.094	0.906
	FedPC-Simple-II	4	3	62	69	0.411	0.132	0.868
	NOTEARS-Avg	**0**	**0**	111	111	0.009	**0.000**	**1.000**
	NOTEARS-ADMM	**0**	3	100	103	0.107	0.200	0.800
	FedDAG	-	-	-	-	-	-	-
	FedPC	6	4	62	72	0.393	0.185	0.815
10	PC-Avg	3.5	4.5	74.6	82.6	0.303	0.190	0.810
	PC-Best	4	6	**65**	75	0.384	0.189	0.811
	FedPC-Simple-I	3	4	66	73	0.384	0.140	0.860
	FedPC-Simple-II	7	4	66	77	0.348	0.220	0.780
	NOTEARS-Avg	**1**	**0**	111	112	0.000	1.000	0.000
	NOTEARS-ADMM	**1**	2	102	105	0.080	0.250	0.750
	FedDAG	-	-	-	-	-	-	-
	FedPC	3	4	**65**	**72**	**0.393**	**0.137**	**0.863**
15	PC-Avg	3.067	5.133	80	88.2	0.258	0.218	0.782
	PC-Best	2	3	75	80	0.313	**0.125**	**0.875**
	FedPC-Simple-I	3	3	74	80	0.313	0.146	0.854
	FedPC-Simple-II	8	3	74	85	0.268	0.268	0.732
	NOTEARS-Avg	**1**	**0**	111	112	0.000	1.000	0.000
	NOTEARS-ADMM	**1**	2	105	108	0.054	0.333	0.667
	FedDAG	-	-	-	-	-	-	-
	FedPC	4	3	**63**	**70**	**0.402**	0.135	0.865

TABLE 5
Details of benchmark Bayesian network: andes

Number of clients	Algorithm	Reverse(↓)	Extra(↓)	Miss(↓)	SHD(↓)	TPR (↑)	FDR(↓)	Precision(↑)
3	PC-Avg	2.333	10	115.333	127.667	0.652	0.053	0.947
	PC-Best	**1**	7	117	125	0.651	**0.035**	**0.965**
	FedPC-Simple-I	3	28	99	130	0.698	0.116	0.884
	FedPC-Simple-II	2	28	99	129	0.701	0.112	0.888
	NOTEARS-Avg	3	**0**	325	328	0.030	0.231	0.769
	NOTEARS-ADMM	30	2	284	316	0.071	0.571	0.429
	FedDAG	-	-	-	-	-	-	-
	FedPC	2	23	**98**	**123**	**0.704**	0.095	0.905
5	PC-Avg	1.8	10.8	127.2	139.8	0.618	0.057	0.943
	PC-Best	1	8	117	126	**0.651**	0.039	0.961
	FedPC-Simple-I	**0**	1	119	**120**	0.648	**0.005**	**0.995**
	FedPC-Simple-II	1	1	119	121	0.645	0.009	0.991
	NOTEARS-Avg	3	**0**	326	329	0.027	0.250	0.750
	NOTEARS-ADMM	23	2	292	317	0.068	0.521	0.479
	FedDAG	-	-	-	-	-	-	-
	FedPC	4	1	**115**	**120**	0.648	0.022	0.978
10	PC-Avg	4.8	22.2	152.3	179.3	0.535	0.128	0.872
	PC-Best	3	17	140	160	0.577	0.093	0.907
	FedPC-Simple-I	**1**	2	142	145	0.577	**0.015**	**0.985**
	FedPC-Simple-II	4	2	142	148	0.568	0.030	0.970
	NOTEARS-Avg	5	**0**	327	332	0.018	0.455	0.545
	NOTEARS-ADMM	20	1	307	328	0.033	0.656	0.344
	FedDAG	-	-	-	-	-	-	-
	FedPC	5	3	**134**	**142**	**0.589**	0.039	0.961
15	PC-Avg	6.933	32.133	158.4	197.467	0.511	0.185	0.815
	PC-Best	3	29	**147**	179	**0.556**	0.145	0.855
	FedPC-Simple-I	**1**	3	166	**170**	0.506	**0.023**	**0.977**
	FedPC-Simple-II	**1**	3	166	**170**	0.506	**0.023**	**0.977**
	NOTEARS-Avg	4	**0**	327	331	0.021	0.364	0.636
	NOTEARS-ADMM	13	**0**	315	328	0.030	0.565	0.435
	FedDAG	-	-	-	-	-	-	-
	FedPC	4	3	164	171	0.503	0.040	0.960

TABLE 6
Details of benchmark Bayesian network: pigs

Number of clients	Algorithm	Reverse(↓)	Extra(↓)	Miss(↓)	SHD(↓)	TPR (↑)	FDR(↓)	Precision(↑)
3	PC-Avg	59	6.333	**0**	65.333	0.900	0.109	0.891
	PC-Best	1	8	**0**	**9**	0.998	**0.015**	**0.985**
	FedPC-Simple-I	50	13	**0**	63	0.916	0.104	0.896
	FedPC-Simple-II	127	13	**0**	140	0.785	0.231	0.769
	NOTEARS-Avg	59	**0**	379	438	0.260	0.277	0.723
	NOTEARS-ADMM	180	22	60	262	0.595	0.365	0.635
	FedDAG	-	-	-	-	-	-	-
	FedPC	**0**	10	**0**	10	**1.000**	0.017	0.983
5	PC-Avg	193.6	821	**0**	1014.6	0.673	0.499	0.501
	PC-Best	**3**	4	**0**	**7**	**0.995**	**0.012**	**0.988**
	FedPC-Simple-I	162	3	**0**	165	0.726	0.277	0.723
	FedPC-Simple-II	342	3	**0**	345	0.422	0.580	0.420
	NOTEARS-Avg	24	**0**	491	515	0.130	0.238	0.762
	NOTEARS-ADMM	181	19	72	272	0.573	0.371	0.629
	FedDAG	-	-	-	-	-	-	-
	FedPC	16	6	**0**	22	0.973	0.037	0.963
10	PC-Avg	257.8	584.5	**0**	842.3	0.565	0.594	0.406
	PC-Best	229	175	**0**	404	0.613	0.527	0.473
	FedPC-Simple-I	236	1	**0**	237	0.601	0.400	0.600
	FedPC-Simple-II	334	1	**0**	335	0.436	0.565	0.435
	NOTEARS-Avg	30	**0**	453	483	0.184	0.216	0.784
	NOTEARS-ADMM	174	13	161	348	0.434	0.421	0.579
	FedDAG	-	-	-	-	-	-	-
	FedPC	**7**	1	**0**	**8**	**0.988**	**0.013**	**0.987**
15	PC-Avg	304.933	1752.53	**0**	2057.47	0.485	0.742	0.258
	PC-Best	213	200	**0**	413	0.640	0.521	0.479
	FedPC-Simple-I	280	129	**0**	409	0.527	0.567	0.433
	FedPC-Simple-II	323	129	**0**	452	0.454	0.627	0.373
	NOTEARS-Avg	**23**	**0**	484	507	0.144	**0.213**	**0.787**
	NOTEARS-ADMM	143	2	238	383	0.356	0.407	0.593
	FedDAG	-	-	-	-	-	-	-
	FedPC	120	135	**0**	**255**	**0.797**	0.351	0.649

TABLE 7
Details of continuous datasets: 10 nodes

Number of clients	Algorithm	Reverse(↓)	Extra(↓)	Miss(↓)	SHD(↓)	TPR (↑)	FDR(↓)	Precision(↑)
3	PC-Avg	2.333	0.667	3.667	6.667	0.600	0.246	0.754
	PC-Best	**0**	1	5	6	0.667	0.091	0.909
	FedPC-Simple-I	1	1	**3**	**5**	**0.733**	0.154	0.846
	FedPC-Simple-II	4	4	**3**	11	0.533	0.500	0.500
	NOTEARS-Avg	3	1	9	13	0.200	0.571	0.429
	NOTEARS-ADMM	3	9	8	20	0.267	0.750	0.250
	FedDAG	5	2	8	15	0.133	0.778	0.222
	FedPC	**0**	**0**	5	**5**	0.667	**0.000**	**1.000**
5	PC-Avg	1.8	**0**	4.4	6.2	0.587	0.171	0.829
	PC-Best	**0**	**0**	4	4	0.733	0.000	1.000
	FedPC-Simple-I	1	**0**	3	4	0.733	0.083	0.917
	FedPC-Simple-II	6	**0**	**2**	8	0.467	0.462	0.538
	NOTEARS-Avg	2	1	11	14	0.133	0.600	0.400
	NOTEARS-ADMM	3	7	8	18	0.267	0.714	0.286
	FedDAG	4	2	9	15	0.133	0.750	0.250
	FedPC	**0**	**0**	3	**3**	**0.800**	**0.000**	**1.000**
10	PC-Avg	0.8	0.2	5.8	6.8	0.560	0.118	0.882
	PC-Best	**0**	**0**	4	4	0.733	**0.000**	**1.000**
	FedPC-Simple-I	**0**	**0**	3	**3**	**0.800**	**0.000**	**1.000**
	FedPC-Simple-II	6	5	**2**	13	0.467	0.611	0.389
	NOTEARS-Avg	1	**0**	12	13	0.133	0.333	0.667
	NOTEARS-ADMM	3	3	10	16	0.133	0.750	0.250
	FedDAG	2	0	8	10	0.333	0.286	0.714
	FedPC	**0**	**0**	3	**3**	**0.800**	**0.000**	**1.000**
15	PC-Avg	0.733	0.267	7.067	8.067	0.480	0.131	0.869
	PC-Best	**0**	**0**	4	4	0.733	0.000	1.000
	FedPC-Simple-I	1	**0**	6	7	0.533	0.111	0.889
	FedPC-Simple-II	5	2	**1**	8	0.600	0.438	0.563
	NOTEARS-Avg	1	**0**	12	13	0.133	0.333	0.667
	NOTEARS-ADMM	4	5	9	18	0.133	0.818	0.182
	FedDAG	3	0	8	11	0.267	0.429	0.571
	FedPC	**0**	**0**	3	**3**	**0.800**	**0.000**	**1.000**

TABLE 8
Details of continuous datasets: 20 nodes

Number of clients	Algorithm	Reverse(↓)	Extra(↓)	Miss(↓)	SHD(↓)	TPR (↑)	FDR(↓)	Precision(↑)
3	PC-Avg	1.667	3.333	11.333	16.333	0.519	0.262	0.738
	PC-Best	2	3	10	15	0.556	0.250	0.750
	FedPC-Simple-I	**1**	10	10	21	**0.593**	0.407	0.593
	FedPC-Simple-II	4	19	**9**	32	0.519	0.622	0.378
	NOTEARS-Avg	2	4	18	24	0.259	0.462	0.538
	NOTEARS-ADMM	6	35	16	57	0.185	0.891	0.109
	FedDAG	27	29	0	56	1.000	0.675	0.325
	FedPC	**1**	**0**	13	**14**	0.481	**0.071**	**0.929**
5	PC-Avg	2.2	2.6	12.8	17.6	0.444	0.276	0.724
	PC-Best	1	**0**	13	14	0.481	0.071	0.929
	FedPC-Simple-I	2	3	11	16	0.519	0.263	0.737
	FedPC-Simple-II	4	5	**8**	17	**0.556**	0.375	0.625
	NOTEARS-Avg	**0**	3	25	28	0.074	0.600	0.400
	NOTEARS-ADMM	4	35	18	57	0.185	0.886	0.114
	FedDAG	27	29	0	56	1.000	0.675	0.325
	FedPC	1	**0**	11	**12**	**0.556**	**0.063**	**0.938**
10	PC-Avg	2.9	1.3	12.4	16.6	0.433	0.260	0.740
	PC-Best	2	**0**	11	13	0.519	0.125	0.875
	FedPC-Simple-I	2	1	10	13	0.556	0.167	0.833
	FedPC-Simple-II	2	16	**8**	26	**0.630**	0.514	0.486
	NOTEARS-Avg	**0**	**0**	25	25	0.074	**0.000**	**1.000**
	NOTEARS-ADMM	3	33	19	55	0.185	0.878	0.122
	FedDAG	26	28	0	54	1.000	0.667	0.333
	FedPC	**0**	**0**	11	**11**	0.593	**0.000**	**1.000**
15	PC-Avg	2	1.667	13.667	17.333	0.420	0.235	0.765
	PC-Best	1	1	12	14	0.519	0.125	0.875
	FedPC-Simple-I	1	1	11	13	0.556	0.118	0.882
	FedPC-Simple-II	2	16	**9**	27	**0.593**	0.529	0.471
	NOTEARS-Avg	**0**	**0**	25	25	0.074	**0.000**	**1.000**
	NOTEARS-ADMM	3	29	21	53	0.111	0.914	0.086
	FedDAG	26	28	0	54	1.000	0.667	0.333
	FedPC	1	**0**	10	**11**	**0.593**	0.059	0.941

TABLE 9
Details of continuous datasets: 50 nodes

Number of clients	Algorithm	Reverse(↓)	Extra(↓)	Miss(↓)	SHD(↓)	TPR (↑)	FDR(↓)	Precision(↑)
3	PC-Avg	6	18.667	28	52.667	0.521	0.399	0.601
	PC-Best	6	13	24	43	0.577	0.317	0.683
	FedPC-Simple-I	6	50	**16**	72	0.690	0.533	0.467
	FedPC-Simple-II	7	76	13	96	**0.718**	0.619	0.381
	NOTEARS-Avg	9	6	50	65	0.169	0.556	0.444
	NOTEARS-ADMM	14	29	32	75	0.352	0.632	0.368
	FedDAG	71	72	0	143	1.000	0.668	0.332
	FedPC	**2**	**3**	32	**37**	0.521	**0.119**	**0.881**
5	PC-Avg	3.8	12	28.4	44.2	0.546	0.283	0.717
	PC-Best	3	4	25	32	0.606	0.140	0.860
	FedPC-Simple-I	2	7	19	**28**	**0.704**	0.153	0.847
	FedPC-Simple-II	6	33	**15**	54	**0.704**	0.438	0.562
	NOTEARS-Avg	4	**1**	56	61	0.155	0.313	0.688
	NOTEARS-ADMM	12	43	36	91	0.324	0.705	0.295
	FedDAG	71	72	0	143	1.000	0.668	0.332
	FedPC	**0**	3	25	**28**	0.648	**0.061**	**0.939**
10	PC-Avg	3.6	6.5	31	41.1	0.513	0.216	0.784
	PC-Best	1	3	28	32	0.592	0.087	0.913
	FedPC-Simple-I	1	6	25	32	0.634	0.135	0.865
	FedPC-Simple-II	6	44	**13**	63	**0.732**	0.490	0.510
	NOTEARS-Avg	5	**0**	56	61	0.141	0.333	0.667
	NOTEARS-ADMM	9	49	44	102	0.254	0.763	0.237
	FedDAG	71	72	0	143	1.000	0.668	0.332
	FedPC	**0**	4	22	**26**	0.690	**0.075**	**0.925**
15	PC-Avg	3.333	5.733	33.733	42.8	0.478	0.209	0.791
	PC-Best	4	3	30	37	0.521	0.159	0.841
	FedPC-Simple-I	1	5	30	36	0.563	0.130	0.870
	FedPC-Simple-II	4	51	**13**	68	**0.761**	0.505	0.495
	NOTEARS-Avg	5	**1**	58	64	0.113	0.429	0.571
	NOTEARS-ADMM	8	40	46	94	0.239	0.738	0.262
	FedDAG	71	72	0	143	1.000	0.668	0.332
	FedPC	**0**	7	21	**28**	0.704	**0.123**	**0.877**

TABLE 10
Details of continuous datasets: 100 nodes

Number of clients	Algorithm	Reverse(↓)	Extra(↓)	Miss(↓)	SHD(↓)	TPR (↑)	FDR(↓)	Precision(↑)
3	PC-Avg	7.667	38.667	63.333	109.667	0.551	0.339	0.661
	PC-Best	6	36	53	95	0.627	0.298	0.702
	FedPC-Simple-I	6	94	48	148	0.658	0.490	0.510
	FedPC-Simple-II	7	218	**46**	271	**0.665**	0.682	0.318
	NOTEARS-Avg	14	16	99	129	0.285	0.400	0.600
	NOTEARS-ADMM	19	26	102	147	0.234	0.549	0.451
	FedDAG	-	-	-	-	-	-	-
	FedPC	**5**	**9**	80	**94**	0.462	**0.161**	**0.839**
5	PC-Avg	7.6	28	68.6	104.2	0.518	0.298	0.702
	PC-Best	10	19	57	86	0.576	0.242	0.758
	FedPC-Simple-I	**4**	17	57	78	0.614	0.178	0.822
	FedPC-Simple-II	10	70	**47**	127	**0.639**	0.442	0.558
	NOTEARS-Avg	13	11	100	124	0.291	0.343	0.657
	NOTEARS-ADMM	20	19	109	148	0.184	0.574	0.426
	FedDAG	-	-	-	-	-	-	-
	FedPC	7	**8**	61	**76**	0.570	**0.143**	**0.857**
10	PC-Avg	**6.8**	18.3	70.8	95.9	0.509	0.237	0.763
	PC-Best	7	11	66	84	0.538	**0.175**	**0.825**
	FedPC-Simple-I	7	16	54	77	0.614	0.192	0.808
	FedPC-Simple-II	7	113	**44**	164	**0.677**	0.529	0.471
	NOTEARS-Avg	13	**9**	102	124	0.272	0.338	0.662
	NOTEARS-ADMM	19	12	115	146	0.152	0.564	0.436
	FedDAG	-	-	-	-	-	-	-
	FedPC	10	11	54	**75**	0.595	0.183	0.817
15	PC-Avg	6.533	16.6	78.6	101.733	0.461	0.239	0.761
	PC-Best	6	**9**	75	90	0.487	**0.163**	**0.837**
	FedPC-Simple-I	**5**	13	71	89	0.519	0.180	0.820
	FedPC-Simple-II	13	100	**46**	159	**0.627**	0.533	0.467
	NOTEARS-Avg	12	10	107	129	0.253	0.355	0.645
	NOTEARS-ADMM	16	**9**	122	147	0.127	0.556	0.444
	FedDAG	-	-	-	-	-	-	-
	FedPC	9	15	57	**81**	0.582	0.207	0.793

TABLE 11
Details of continuous datasets: 300 nodes

Number of clients	Algorithm	Reverse(↓)	Extra(↓)	Miss(↓)	SHD(↓)	TPR (↑)	FDR(↓)	Precision(↑)
3	PC-Avg	14	139.333	159	312.333	0.586	0.343	0.657
	PC-Best	14	47	168	229	0.565	0.205	0.795
	FedPC-Simple-I	6	393	116	515	0.708	0.574	0.426
	FedPC-Simple-II	**0**	2427	**88**	2515	**0.789**	0.880	0.120
	NOTEARS-Avg	23	**5**	311	339	0.203	0.248	0.752
	NOTEARS-ADMM	74	64	226	364	0.282	0.539	0.461
	FedDAG	-	-	-	-	-	-	-
	FedPC	4	9	187	**200**	0.543	**0.054**	**0.946**
5	PC-Avg	12.2	109.4	160.4	282	0.587	0.316	0.684
	PC-Best	11	90	130	231	0.663	0.267	0.733
	FedPC-Simple-I	**5**	44	128	177	**0.682**	0.147	0.853
	FedPC-Simple-II	28	353	**109**	490	0.672	0.576	0.424
	NOTEARS-Avg	23	**10**	312	345	0.199	0.284	0.716
	NOTEARS-ADMM	50	40	253	343	0.275	0.439	0.561
	FedDAG	-	-	-	-	-	-	-
	FedPC	6	26	144	**176**	0.641	**0.107**	**0.893**
10	PC-Avg	13.5	67.4	175.4	256.3	0.548	0.260	0.740
	PC-Best	11	51	162	224	0.586	0.202	0.798
	FedPC-Simple-I	**3**	30	139	**172**	0.660	**0.107**	**0.893**
	FedPC-Simple-II	18	756	**97**	871	**0.725**	0.719	0.281
	NOTEARS-Avg	23	**6**	321	350	0.179	0.279	0.721
	NOTEARS-ADMM	45	25	283	353	0.215	0.438	0.563
	FedDAG	-	-	-	-	-	-	-
	FedPC	4	31	140	175	0.656	0.113	0.887
15	PC-Avg	15.667	56.733	185.333	257.733	0.519	0.250	0.750
	PC-Best	12	49	166	227	0.574	0.203	0.797
	FedPC-Simple-I	**8**	21	162	191	0.593	**0.105**	**0.895**
	FedPC-Simple-II	32	375	**110**	517	**0.660**	0.596	0.404
	NOTEARS-Avg	24	**6**	324	354	0.167	0.300	0.700
	NOTEARS-ADMM	42	21	291	354	0.203	0.426	0.574
	FedDAG	-	-	-	-	-	-	-
	FedPC	10	40	133	**183**	0.658	0.154	0.846

the best performance with highest precision values and lowest FDR vlaues. It means that our method gets a more accurate DAG and identifies more correct directed edges than its rivals. Continuous optimization approaches, i.e. NOTEARS-Avg and NOTEARS-ADMM, achieve a high TPR, it also learns many extra edges. In addition, we also observe that the performance of continuous optimization approaches is comparable to that of other comparison methods on real data, whereas generally worse than that of traditional methods on benchmark data.

S-2: An illustrative example of the privacy protection strategy employed by FedPC

As shown in Fig. 1, we firstly sort the first letters of all variables on each client in alphabetical order to establish a correspondence between variables and identifiers. However, this correspondence may encounter instances of ambiguity, exemplified by the case of "Theatre" and "Theft". Because both words share identical first and second letters, sorting based solely on the initial letter fails to determine a unique and unambiguous mapping between variables and identifiers. Then we sort the remaining letters of each variable until all letters have been sorted. The same feature space among clients ensures that the resulting correspondence between variables and identifiers remains consistent. Furthermore, this method effectively safeguards against the server's access to the semantic information associated with each variable, thereby mitigating potential risks of privacy leakage.

References

[1] D. Kalainathan, O. Goudet, and R. Dutta, "Causal discovery toolbox: Uncovering causal relationships in python," *Journal of Machine Learning Research*, vol. 21, pp. 37:1–37:5, 2020.
[2] K. Zhang, J. Peters, D. Janzing, and B. Schölkopf, "Kernel-based conditional independence test and application in causal discovery," *arXiv preprint arXiv:1202.3775*, 2012.
[3] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan, "Causal protein-signaling networks derived from multiparameter single-cell data," *Science*, vol. 308, no. 5721, pp. 523–529, 2005.
[4] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman, *Causation, prediction, and search*. MIT press, 2000.
[5] X. Zheng, B. Aragam, P. K. Ravikumar, and E. P. Xing, "DAGs with no tears: Continuous optimization for structure learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
[6] I. Ng and K. Zhang, "Towards federated bayesian network structure learning with continuous optimization," in *Proceedings of International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 8095–8111.
[7] E. Gao, J. Chen, L. Shen, T. Liu, M. Gong, and H. Bondell, "FedDAG: Federated DAG structure learning," *Transactions on Machine Learning Research*, 2023. [Online]. Available: https://openreview.net/forum?id=MzWgBjZ6Le

TABLE 12
Details of the real dataset

Number of clients	Algorithm	Reverse(↓)	Extra(↓)	Miss(↓)	SHD(↓)	TPR (↑)	FDR(↓)	Precision(↑)
3	PC-Avg	4.333	**0**	9.667	14	0.176	0.595	0.405
	PC-Best	4	**0**	9	**13**	0.235	**0.500**	**0.500**
	FedPC-Simple-I	5	**0**	9	14	0.176	0.625	0.375
	FedPC-Simple-II	**3**	8	8	19	**0.353**	0.647	0.353
	NOTEARS-Avg	7	5	**4**	16	**0.353**	0.667	0.333
	NOTEARS-ADMM	7	10	**4**	21	**0.353**	0.739	0.261
	FedDAG	6	10	5	21	0.353	0.727	0.273
	FedPC	4	**0**	9	**13**	0.235	**0.500**	**0.500**
5	PC-Avg	3.8	0.2	10.4	14.4	0.165	0.556	0.444
	PC-Best	4	**0**	10	14	0.176	0.571	0.429
	FedPC-Simple-I	5	**0**	9	14	0.176	0.625	0.375
	FedPC-Simple-II	5	4	9	18	0.176	0.750	0.250
	NOTEARS-Avg	7	6	5	18	0.294	0.722	0.278
	NOTEARS-ADMM	8	10	**2**	20	**0.412**	0.720	0.280
	FedDAG	7	10	4	21	0.353	0.739	0.261
	FedPC	**2**	**0**	9	**11**	0.353	**0.250**	**0.750**
10	PC-Avg	2	0	11.7	13.7	0.194	0.355	0.645
	PC-Best	**0**	**0**	12	12	0.294	**0.000**	**1.000**
	FedPC-Simple-I	3	**0**	10	13	0.235	0.429	0.571
	FedPC-Simple-II	3	10	8	21	0.353	0.684	0.316
	NOTEARS-Avg	7	9	4	20	0.353	0.727	0.273
	NOTEARS-ADMM	8	13	**3**	24	0.353	0.778	0.222
	FedDAG	7	12	2	21	0.471	0.704	0.296
	FedPC	**0**	**0**	10	**10**	**0.412**	**0.000**	**1.000**
15	PC-Avg	1.533	0.133	12.133	13.8	0.196	0.313	0.687
	PC-Best	**0**	1	12	13	0.294	**0.167**	**0.833**
	FedPC-Simple-I	2	**0**	12	14	0.176	0.400	0.600
	FedPC-Simple-II	5	6	8	19	0.235	0.733	0.267
	NOTEARS-Avg	7	12	**3**	22	0.412	0.731	0.269
	NOTEARS-ADMM	6	15	**3**	24	**0.471**	0.724	0.276
	FedDAG	6	14	4	24	0.412	0.741	0.259
	FedPC	2	**0**	10	**12**	0.294	0.286	0.714

TABLE 13
Details of the results on nonlinear datasets

Nonlinear datasets	Number of clients	Algorithms	SHD(↓)	TPR (↑)	Precision(↑)	F1(↑)
10 nodes	3	FedDAG	5	0.818	0.692	0.750
		NOTEARS-MLP-ADMM	27	0.091	0.040	0.056
		FedPC-KBT	**4**	**0.909**	**0.714**	**0.800**
	5	FedDAG	4	0.636	**0.875**	0.737
		NOTEARS-MLP-ADMM	15	0.182	0.182	0.182
		FedPC-KBT	**2**	**0.909**	0.833	**0.870**
	10	FedDAG	7	0.364	0.667	0.471
		NOTEARS-MLP-ADMM	14	0.182	0.222	0.200
		FedPC-KBT	**2**	**0.818**	**0.818**	**0.818**
	15	FedDAG	7	0.364	0.667	0.471
		NOTEARS-MLP-ADMM	14	0.091	0.111	0.100
		FedPC-KBT	**3**	**0.727**	**0.889**	**0.800**
20 nodes	3	FedDAG	24	0.143	0.500	0.222
		NOTEARS-MLP-ADMM	30	0.286	0.348	0.314
		FedPC-KBT	**11**	**0.786**	**0.667**	**0.721**
	5	FedDAG	26	0.071	0.333	0.118
		NOTEARS-MLP-ADMM	36	0.250	0.250	0.250
		FedPC-KBT	**4**	**0.893**	**0.862**	**0.877**
	10	FedDAG	27	0.036	1.000	0.069
		NOTEARS-MLP-ADMM	31	0.071	0.250	0.111
		FedPC-KBT	**7**	**0.786**	**0.815**	**0.800**
	15	FedDAG	27	0.036	0.500	0.067
		NOTEARS-MLP-ADMM	30	0.071	0.222	0.108
		FedPC-KBT	**13**	**0.571**	**0.696**	**0.628**

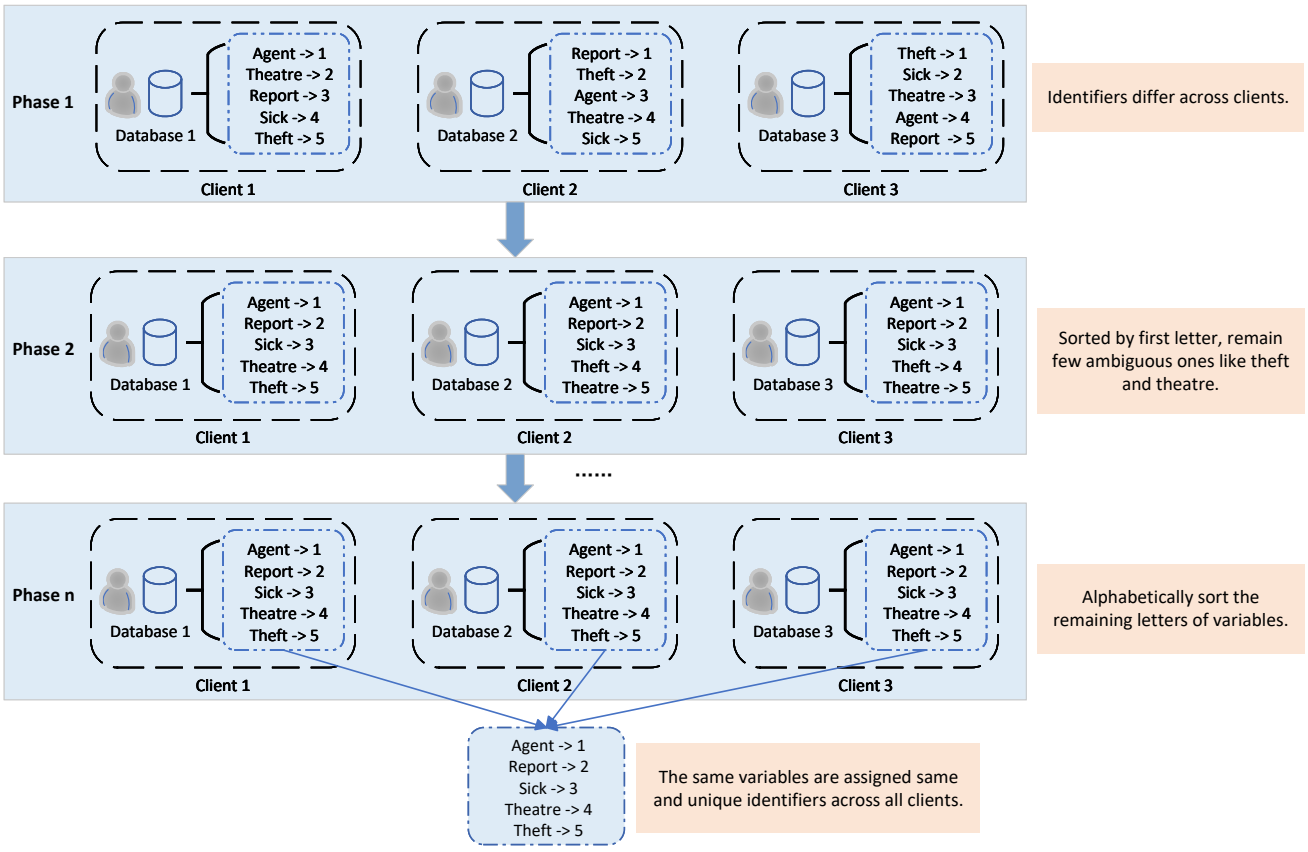

Fig. 1. An easily implementable privacy protection strategy in the FedPC algorithm.