Personalized Federated Learning With a Graph

Fengwen Chen[1] , **Guodong Long**[1] , **Zonghan Wu**[1] , **Tianyi Zhou**[2,3] and **Jing Jiang**[1*]
[1]Australian Artificial Intelligence Institute, FEIT, University of Technology Sydney
[2]University of Washington, Seattle
[3]University of Maryland, College Park
{Fengwen.Chen, Zonghan.Wu-3}@student.uts.edu.au,
{Guodong.Long, Jing.Jiang }@uts.edu.au, tianyizh@uw.edu

Abstract

Knowledge sharing and model personalization are two key components in the conceptual framework of personalized federated learning (PFL). Existing PFL methods focus on proposing new model personalization mechanisms while simply implementing knowledge sharing by aggregating models from all clients, regardless of their relation graph. This paper aims to enhance the knowledge-sharing process in PFL by leveraging the graph-based structural information among clients. We propose a novel structured federated learning (SFL) framework to learn both the global and personalized models simultaneously using client-wise relation graphs and clients' private data. We cast SFL with graph into a novel optimization problem that can model the client-wise complex relations and graph-based structural topology by a unified framework. Moreover, in addition to using an existing relation graph, SFL could be expanded to learn the hidden relations among clients. Experiments on traffic and image benchmark datasets can demonstrate the effectiveness of the proposed method.

1 Introduction

Since Federated Learning (FL)[McMahan *et al.*, 2017] was first proposed in 2017, it has evolved into a new-generation collaborative machine learning framework with applications in a range of scenarios, including Google's Gboard on Android [McMahan *et al.*, 2017], Apple's siri [1], Computer Visions [Luo *et al.*, 2019; Jallepalli *et al.*, 2021; He *et al.*, 2021], Smart Cities [Zheng *et al.*, 2022], Finance [Long *et al.*, 2020] and Healthcare [Rieke *et al.*, 2020; Xu *et al.*, 2021; Long *et al.*, 2022]. Various FL tools and packages have been developed and open-sourced by hi-tech companies, such as Google[2], NVIDIA[3], Intel[4], Amazon [5], Baidu[6], and Webank[7].

The vanilla FL method, known as FedAvg [McMahan *et al.*, 2017], is derived from a distributed machine learning framework before it is applied to a large-scale mobile service system. In particular, it aims to train a single shared model at the server by aggregating the smartphones' local model, trained with its own data. Thus, the end-user's private data in each smartphone is not uploaded to the cloud server. FedAvg first proposed the non-IID challenge, a key feature of FL. To tackle the non-IID challenge in FL, some works [Li *et al.*, 2018; Reddi *et al.*, 2020] focus on training a single robust model at the server. However, other methods aim to learn multiple global or centric models [Ghosh *et al.*, 2020; Mansour *et al.*, 2020; Xie *et al.*, 2021], where each model serves a cluster of clients whose data distribution is the same or similar. More recently, personalized FL methods [Li *et al.*, 2021; Deng *et al.*, 2020; Tan *et al.*, 2022; Fallah *et al.*, 2020] are proposed to learn many client-specific personalized models using the global model as the component of knowledge sharing. Therefore, the objective of FL research has been changed from learning server-based models to client-specific models.

To learn client-specific personalized models in the FL setting, knowledge sharing and personalization of local models are two key components. However, existing personalized FL focuses on improving components of model personalization while implementing the component of knowledge sharing by simply aggregating all local models. This kind of implementation overlooks the graph relationship across clients with non-IID data. Moreover, if the degree of non-IID is very high, aggregation across all clients will produce a low-quality global model that will eventually impact the performance of model personalization. From an application perspective, graph relationship have many real-world applications, such as traffic sensors with road maps (Figure 1), devices in a smart home, mobile APPs using users' social networks, and fraud

[*] Corresponding Author

[1] https://machinelearning.apple.com/research/learning-with-privacy-at-scale

[2] https://www.tensorflow.org/federated

[3] https://nvidia.github.io/NVFlare

[4] https://github.com/intel/openfl

[5] https://aws.amazon.com/blogs/architecture/applying-federated-learning-for-ml-at-the-edge/

[6] https://github.com/PaddlePaddle/PaddleFL

[7] https://github.com/FederatedAI/FATE

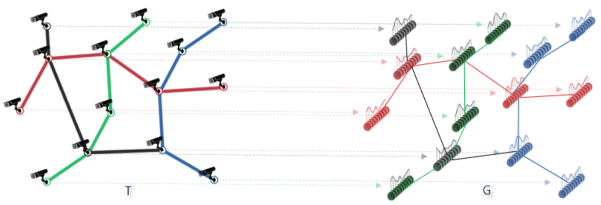

Figure 1: In a smart city, each devices deployed on the road can collect data and make real-time decision without waiting for the response of cloud servers. Each device needs to make intelligent decision based on the collected road conditions and nearby devices.

detection based on the mutual interactions of users. Adopting these types of relation graphs will enhance the performance of FL.

This paper proposes a novel **structured federated learning (SFL)** that aims to leverage the relation graph among clients to enhance personalized FL. In particular, we design a fine-grained model aggregation mechanism to leverage each client's neighbors' local models. Specifically, a relation graph will be stored in the server, and then the client-centric model aggregation will be conducted along the relation graph's structure. To simplify implementation, we propose to use the Graph Convolutional Network (GCN) [Kipf and Welling, 2016] to implement the model aggregation function; therefore, the proposed solution is easy-to-implement by integrating FL and GCN. In addition, we formulate the problem in a unified optimization framework to include both personalized FL and graph-based model aggregation. **Contributions** of this paper are summarized as follows.

- We are the first to propose a new federated setting by considering a relation graph among clients. Moreover, the proposed GCN-based model aggregation mechanism is a new and easy-to-implement idea for FL;
- The research problem has been formulated to a unified optimization framework that can learn optimal personalized models while leveraging the graph;
- The proposed method has been expanded to learn the hidden relations among clients, and the conceptual framework can be extended to integrate with other model personalization techniques;
- Experiments on both the image and traffic datasets have demonstrated the effectiveness of the proposed method.

2 Related Work

2.1 Federated Learning with Non-IID

The vanilla FL method, FedAvg [McMahan *et al.*, 2017], has been suffering from the non-IID challenge where each client's local data distribution is varied [Kairouz *et al.*, 2021]. To tackle this challenge, [Li *et al.*, 2019a] proposed FedDANE by adapting the DANE to a federated setting. In particular, FedDANE is a federated Newton-type optimization method. [Li *et al.*, 2018] proposed FedProx for the generalization and re-parameterization of FedAvg. It adds a proximal term to clients' local objective functions by constraining the parameter-based distance between the global model and local model. [Reddi *et al.*, 2020] proposes to use adaptive learning rates to FL clients and [Jiang *et al.*, 2020] conduct attention-based adaptive weighting to aggregate clients' models. [Li *et al.*, 2019b] studies the convergence of the FedAvg on non-IID scenarios. In summary, the early-stage FL research focuses on learning a robust global model that usually assumes the non-IID scenario is relatively mild. In recent times, research on personalized FL (Sec 2.2) has attracted broad interest in tackling severe non-IID scenarios.

2.2 Personalized Federated Learning

This section will discuss two major FL solutions to tackle severe non-IID scenarios. **Group-wise PFL**, which is also named clustered FL, assumes the clients can be clustered to different groups with severe non-IID across inter-group clients and mild non-IID across intra-group clients. Hence, clustered FL can be categorized according to different clustering methods and distance measurements. Kmeans-based clustered FL [Xie *et al.*, 2021] and [Mansour *et al.*, 2020; Ghosh *et al.*, 2020] measured the distance using model parameters and accuracy respectively. Hierarchical clustering [Briggs *et al.*, 2020] has been applied to FL. Furthermore, [Ma *et al.*, 2022] proposed a general form to model the clustered FL problem into a bi-level optimization framework, then leveraged the important contributions among clients to form a weighted client-based clustered FL framework.

Client-wise PFL that usually assumes each client's data distribution is different from others; thus, each client should have a personalized model on their device. In general, a simple PFL method could train a global model in FedAvg, then conduct a few steps of fine-tuning on each client [Cheng *et al.*, 2021]. In this framework, knowledge sharing is model aggregation, and model personalization is local fine-tuning. Per-FedAvg [Fallah *et al.*, 2020] considered fine-tune as a regularization term on the learning objective function of the global model. Ditto [Li *et al.*, 2021] was proposed as a bi-level optimization framework for PFL while considering a regularization term to constrain the distance between the local model and global model. Investigations by [Shamsian *et al.*, 2021; Chen *et al.*, 2018] that aim to train a global hyper-network or meta-learner instead of a global model before sending it to clients for local optimization. SCAFFOLD [Karimireddy *et al.*, 2020] proposes to learn personalized control variate that correct the local model accordingly. Layer-wise personalization [Arivazhagan *et al.*, 2019; Liang *et al.*, 2020] and Representation-wise personalization [Tan *et al.*, 2021] are two simple but effective solution of PFL.

2.3 Learning with Structural Information

Learning on structural data with GCNs [Kipf and Welling, 2016] or graph neural networks [Wu *et al.*, 2020] are ubiquitous in many fields for tasks such as , node classification [Pan *et al.*, 2016a], link prediction[Wang *et al.*, 2017b], node clustering[Wang *et al.*, 2017a], and graph classification [Pan *et al.*, 2016b]. GCNs capture relationships between concepts (also called nodes) in a graph using the k-hop aggregation mechanism. Thus, a weighted hop in GCN can capture a more complex relationship in the graph [Chen *et al.*, 2019;

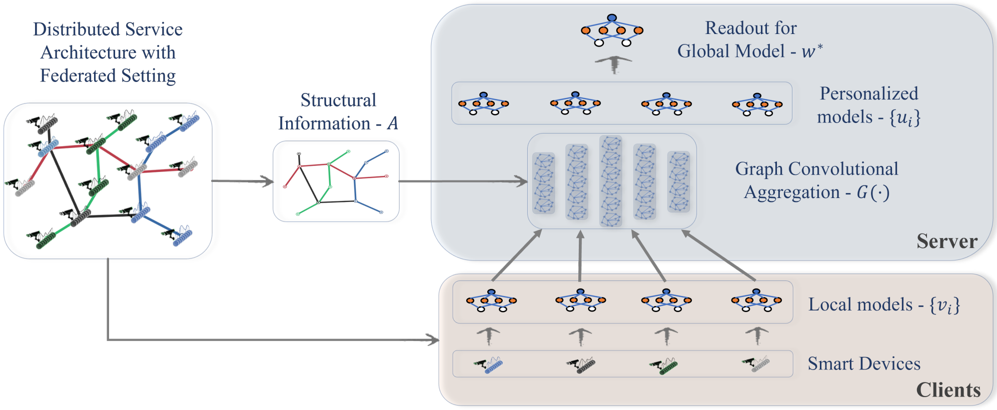
Distributed Service Architecture with Federated Setting
Structural Information - A
Readout for Global Model - w^*
Personalized models - $\{u_i\}$
Graph Convolutional Aggregation - $G(\cdot)$
Server
Local models - $\{v_i\}$
Smart Devices
Clients

Figure 2: The overview of structured federated learning (SFL). A GCN module is used in the server to generate personalized client-specific models $\{u_i\}$ by aggregating the collected local model parameters $\{v_i\}$ according to the graph structure A of all clients.

Zhang *et al.*, 2022]. Recently, GraphFL[Wang *et al.*, 2020] has been proposed to explore the graph learning problem in a federated setting. In particular, each client trains a graph neural networks (e.g., GCNs) for learning the structured data locally, then shares knowledge via model aggregation at the server.

3 Problem Formulation

Given N participants in an FL system, each one has a local dataset D_i which is drawn from a distribution P_i. Given the non-IID setting, we usually assume all P_i are distinct from each other. An adjacency matrix $A \in \{0,1\}^{N\times N}$ represents the topological relationship across participants. In general, a FL system is to solve below optimal objective.

$$\min_{w} G(F_1(w), ...F_N(w)) \tag{1}$$

where $F_i(W)$ is the supervised loss of the i-th client that has a locally stored dataset D_i, and all clients using the same global model M parameterized by w. The $G(.)$ is a function that aggregates the local objectives. For example, in FedAvg [McMahan *et al.*, 2017], $G(.)$ is a weighted average of local lossess using the size of local dataset, ie.e., $\sum |D_i|/\sum_j |D_j|$.

In general, a personalized FL system is usually modeled as a bi-level optimization problem as below.

$$\begin{aligned} \min_{\{v_1...v_N\}} \quad & h_i(v_i; w^*) := F_i(v_i) + \lambda R(v_i, w^*) \\ s.t. \quad & w^* \in \arg\min_{w} G(F_1(w), ..., F_N(w)) \end{aligned} \tag{2}$$

where each client has a unique personalized model M_i parameterized by v_i, and w^* is an optimal global model to minimize the loss as mentioned in the Eq. 1. R is the regularization term to control model updates on clients, for example, [Li *et al.*, 2021] propose a L2 term $\frac{1}{2}||v_i - w^*||^2$ to constraint the local updating won't be far away to the global model.

To find the optimal solution for the loss Eq. 2, existing personalized FL methods will take various forms, such as fine-tuning [Cheng *et al.*, 2021], meta-training [Fallah *et al.*, 2020], and partial parameter-sharing [Liang *et al.*, 2020]. Our proposed structured federated learning is a new solution to leverage both structural information and model parameters for personalized FL.

4 Structured Federated Learning

Our proposed structured FL will formulate the below bi-level optimization problem.

$$\begin{aligned} \min_{v_{1:N};u_{1:N}} \quad & \sum_{i=1}^{N} (F_i(v_i) + \lambda \mathcal{R}(v_i, w^*, u_i)) + \gamma \mathcal{G}(A, A^*) \\ s.t. \quad & w^* \in \arg\min_{w} G(F_1(w), ..., F_N(w)) \\ & A^*_{i,j} = S(u_i, u_j) \end{aligned} \tag{3}$$

where the A is the adjacent matrix of the pre-defined graph among clients, and the A^* is a similarity matrix among clients using their paired distance $S(u_i, u_j)$ which is the euclidean distance of two client-specific models' parameters.

Most real applications assume that the neighbors in a graph share similar behaviors which indicate similar data distribution and intelligent models. To implement this assumption, a distance-based penalty term $G(A, A^*)$ has been proposed to force neighbors in a graph that have similar model parameters. In particular, G is the distance between the graph's adjacent matrix A and model parameter-based affinity matrix $A^* = S(u_i, u_j)$. To minimize this penalty term G, we will update u using the below equation.

$$u_i^* \in \arg\min_{u_i} \sum_{j \in \mathcal{N}(i)} A_{j,i} S(u_i, u_j) \tag{4}$$

In many real applications, the adjacency matrix A may not always exist. Thus, it needs to be learned. For this case, we can formulate the optimization problem as below.

$$\begin{aligned} \min_{v_{1:N};u_{1:N};A} \quad & \sum_{i=1}^{N} (F_i(v_i) + \lambda \mathcal{R}(v_i, w^*, u_i)) + \gamma \mathcal{G}(A, A^*) \\ s.t. \quad & w^* \in \arg\min_{w} G(F_1(w), ..., F_N(w)) \\ & A^*_{i,j} = S(u_i, u_j) \\ & \|A_i\| \leq K \end{aligned} \tag{5}$$

where $\mathcal{R}(.) = R(v_i, w^*) + R(v_i, u_i^*)$ is an abbreviation of two regularization terms. $\mathcal{G}(.)$ is to constraint the distance between the learned graph A and its sparse representation A^*

which is sparse and able to preserve important proximity relationship among clients. Specifically, we can add use a constant K to constraint each node's L0 norm $\|A_i\|$. There are various ways to measure the proximity between two clients: for example, distance of model parameters

4.1 Optimization

To solve the optimization problem in Eq. 3, we conduct the below steps. First, we update the v_i^* by solving the local loss $F_i(v_i)$ with two regularization terms: distance between local model and gradient-based aggregated global model $R(v_i, w^*)$, and distance between local model and graph-based aggregated personalized model $R(v_i, u_i)$. Then, we conduct model aggregation at the server to update w and $\{u_i\}_i^N$. In particular, we can use a Graph Convolution Network (GCN) [Kipf and Welling, 2016] to implement the graph-based model aggregation by constructing the graph: N clients represent the node in the graph, a pre-defined adjacency matrix A, and each node's attribute u_i is initialized by its local model v_i. The GCN module will automatically update each node's model parameters u_i by the GCN aggregator. It will satisfy the second constraint in Eq. 2. Moreover, the global model w will be updated by aggregating all personalized models u_i which is to satisfy the first constraint in Eq. 3. This gradient-based model aggregation across all clients is equivalent to the read-out operator in the GCN.

To solve the optimization problem in Eq. 5, we can refine the local update step (line 7, Algorithm 1) by adding the graph regularization term $G(A)$. Then, we will add a structure-learning step in the aforementioned optimization steps for Eq. 3. In particular, we will design a graph encoder to minimize three regularization terms of Eq. 5, as below.

$$\min_{A} \sum_{i=1}^{N} (\lambda[R(v_i, w^*) + R(v_i, u_i)] + \gamma \mathcal{G}(A)) \tag{6}$$

We can construct the relation graph A using the learned representation of nodes. We can also define a fully connected graph with weighted edges. The GCN will not only learn representation but also learn the structure by adjusting the weights of edges.

4.2 Algorithm

The aforementioned optimization procedure has been implemented in Algorithm 1. The optimization goal will be achieved iterative through multiple communication rounds between the server and clients. In each communication round, we will have two steps to solve the bi-level optimization problem. First, we update the local model v_i by conducting local model training with supervised loss and regularization terms. Second, we conduct model aggregation at the server using GCN. In the case that A is not present, we will add an optional step for structure learning.

Algorithm 1 Structural Federated Learning - Server.

```
1:  Initialize $\lambda_0, \eta, A, \{v_i^{(0)}\}_{i=1}^N \leftarrow v$
2:  for each communication round t = 0, 1, ..., T do
3:      $\lambda = 1[t > 0] \times \lambda_0$
4:      Local updating:
5:      for each client i = 1, 2, ...., N in parallel do
6:          Update $v_i$ for $s$ local steps:
7:          $v_i^{(t)} \leftarrow v_i^{(t)} - \eta\nabla\left(F_i(v_i^{(t)}) + \lambda[R(v_i^{(t)}, w^{(t)}) + R(v_i^{(t)}, u_i^{(t)})]\right)$
8:          $v_i^{(t+1)} \leftarrow v_i^{(t)}$
9:      end for
10:     Structure-based aggregating:
11:     $\{u_i^{(t+1)}\}_{i=1}^N \leftarrow \{v_i^{(t)}\}_{i=1}^N$
12:     Update $u_i^{(t+1)}$ for m steps of $GCN(A,\ \{u_i^{(t+1)}\}_{i=1}^N)$
13:     $w^{(t+1)} \leftarrow GCN\_readout(\{u_i^{(t+1)}\}_{i=1}^N)$
14:     (Optional) Structure learning:
15:     $A \leftarrow Structure\_learn(\{v_i^{(t+1)}, u_i^{(t+1)}, w^{(t+1)}\}_{i=1}^N)$.
16: end for
```

5 Experiment

This section discusses the experimental analysis of the proposed method. In section 5.1, we choose a graph-based benchmark dataset on traffic forecasting and an FL-based benchmark dataset on image classification. In section 5.2, we introduce the baseline and experimental settings. Then, we compare the proposed SFL with other baselines and also perform the ablation study to verify this observation in section 5.3. Further analysis is provided with visualization in section 5.4 and compatibility analysis in section **??**. All implementation codes are available on Github[8].

5.1 Datasets

The traffic datasets are ideal for validating our hypothesis, as the data comes with a natural topological structure and per-user data with non-IID distribution, all collected in the real world. We used four traffic datasets, METR-LA, PEMS-BAY, PEMS-D4, and PEMS-D8 to observe the performance of the SFL in different real-world scenarios. We apply the same data pre-processing procedures as described in [Wu *et al.*, 2019]. All the readings are arranged in units of 5-minutes. The adjacency matrix is generated based on Gaussian kernel [Shuman *et al.*, 2013]. We also apply Z-score normalization to the inputs and separate the training-set, validation-set, and test-set in a 70% 20% and 10% ratio. The evaluation metrics we use for the datasets include mean absolute error (MAE), root mean squared error(RMSE), and mean absolute percentage error (MAPE).

For the image datasets, we applied the same train/test splits as in the work [9]. We artificially partitioned the CIFAR-10 with parameter $k(shards)$ to control the level of non-IID data. The whole dataset is sorted according to the label and then split into $n \times k$ shards equally, and each of n clients is

[8] https://github.com/dawenzi098/SFL-Structural-Federated-Learning

[9] Learning multiple layers of features from tiny images

	METR-LA			PEMS-BAY			PEMS-D4			PEMS-D8		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
FedAvg	7.03	21.63	10.81	3.62	10.65	7.26	44.96	30.03	59.97	36.76	21.04	49.14
FedAtt	6.89	23.54	10.55	3.26	5.50	6.41	45.53	30.15	60.68	35.80	23.27	47.75
SFL	**5.22**	**16.55**	8.98	**2.96**	7.62	**5.95**	45.86	56.31	**59.00**	**32.95**	**20.98**	**46.03**
SFL*	5.26	16.77	**8.95**	3.02	**7.42**	6.04	**40.75**	**31.06**	59.45	35.82	34.68	47.82
STGCN	4.59	12.70	9.40	4.59	12.70	9.40	25.15	-	38.29	18.88	-	27.87
Graph WaveNet	3.53	10.01	7.37	1.95	4.63	4.52	18.71	13.45	30.04	14.39	9.4	23.03

Table 1: Performance of traffic forecasting in federated setting

assigned k shards. In short, the smaller the $shards$ is, the more serious are the non-IID data issues.

5.2 Baselines and Experiment Settings

We compare our method with four representative federated-learning frameworks including the standard FedAvg [McMahan *et al.*, 2017] and three other personalization federated frameworks, FedAtt [Ji *et al.*, 2019], FedProx [Li *et al.*, 2020] and Scaffold [Karimireddy *et al.*, 2020]. In addition, we also implement two fine-tune-based methods FedPer[Arivazhagan *et al.*, 2019] and LG-FedAvg[Liang *et al.*, 2020]. During the client model selection, to focus more attention on the impact of introducing structural information during the server aggregation process, we choose simple and fixed client models for all frameworks to shield the influence of client model architecture. We use pure RNN for traffic prediction tasks with 64 hidden layer sizes. For CIFAR-10, we use ResNet9 as the base model for all evaluated methods, For a fair comparison, without any additional statement, all reported results are based on the same training set as follows. We employ SGD with the same learning rate as the optimizer for all training operations, use 128 for batch size, and the number of total communication rounded to 20. It is worth mentioning that higher capacity models and larger communication rounds can always produce a higher performance on any of those datasets. As such, the goal of our experiment is to compare the relative performances of these frameworks with the same basic models, rather than comparing specific values.

5.3 Comparison Analysis

The performance of SFL in a traffic forecasting task compared with other baselines is provided in Table 1. We use SFL* to denote the SFL with structure learning enabled. In this table, we report the average MAE, MAPE, and RMSE across all the clients for 60 minutes (12-time steps) ahead of prediction. The full result can be explored in three parts. First, for METR-LA and PEMS-BAY there is a 25% and 18% performance improvement in terms of MAE. Since the two datasets have relatively more nodes and complex structural information (edges) as stated in Table 1, the use of a graph convolutional network to introduce sufficient structural information into the server aggregation process could significantly improve the performance of the FL system. Even compared with privacy non-preserved, the overall performance of our proposed method is still very competitive. Second, the PEMS-D4 provides us with a very practical scenario where the structural information is missing and the SFL cannot directly benefit from this lack of structural information. In this case, the results prove that our structure self-learning module can learn in the absence of information, thus bringing a greater than 10% performance gain. Finally, the PEMS-D8 dataset provides the performance of SFL with a worst-case scenario where clients are few and far between. Here, the relationships are fragile. The results confirm that the performance lower-bound of SFL remains slightly better than the traditional methods due to the natural data distribution skew. This trait of SFL was carefully examined and analyzed in the next set of experiments.

We then ran experiments on the image-based CIFAR-10 dataset to further validate SFL's ability to deal with the non-IID data. Table 2 and 3 demonstrates SFL's superior performance in different levels of distribution skew for non-IID. Note that the larger value of $shards$ indicates that the data is distributed more evenly across clients. For CIFAR-10, with the mimics of an severe non-IID data environment (shards are 2 and 5), the traditional FL algorithms are not functional. Our SFL performs significantly better than other algorithms, both from the best 5% and worst 5% due to its unique aggregation mechanism. As the data distribution tends to iid (shards are 10 and 20), the performance of the traditional algorithms increases to the normal level while our SFL still maintains a very competitive performance.

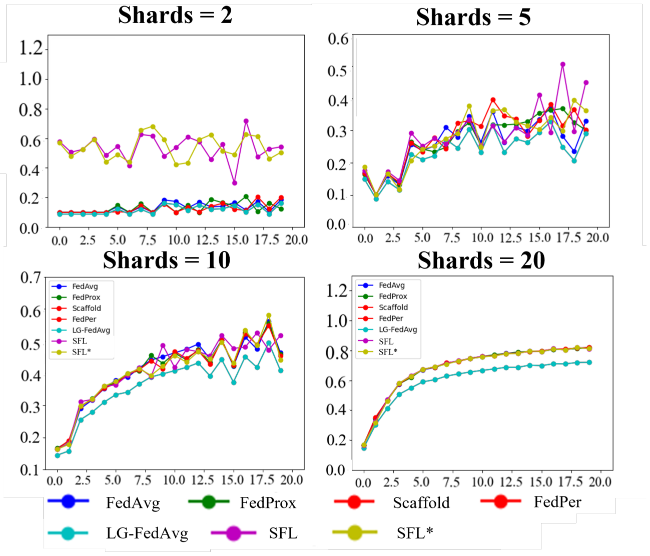

Figure 3: Visualization of Convergence

	Shards = 2			Shards = 5		
	Mean Acc	Best 5%	Worst 5%	Mean Acc	Best 5%	Worst 5%
FedAvg	18.55 ± 21.74	73.20 ± 10.93	0.00 ± 0.00	32.95 ± 17.61	67.40 ± 3.98	2.20 ± 1.47
FedAtt	10.08 ± 24.46	90.00± 20.00	0.00 ± 0.00	28.25 ± 6.02	52.80 ± 0.75	1.40 ± 0.80
FedProx	12.49 ± 21.99	74.20 ± 19.65	0.00 ± 0.00	30.11 ± 14.85	57.40 ± 1.85	4.00 ± 1.90
Scaffold	20.20 ± 26.73	90.40 ± 1.85	0.00 ± 0.00	30.16 ± 13.66	57.40 ± 5.57	2.40 ± 2.87
FedPer	20.24 ± 18.52	78.30 ± 14.26	0.00 ± 0.00	34.59 ± 18.26	69.25 ± 4.68	4.29 ± 1.24
LG-FedAvg	16.73 ± 22.01	67.31 ± 12.68	0.00 ± 0.00	31.75 ± 14.35	67.24 ± 3.53	2.73 ± 1.95
SFL	**54.25 ± 21.72**	**100.00 ± 0.00**	**6.2 ± 2.22**	**45.03 ± 15.66**	**75.20 ± 4.26**	9.20 ± 5.53
SFL*	50.54 ± 29.52	100.00 ± 0.00	0.00 ± 0.00	36.18 ± 12.74	62.60 ± 1.02	**12.20 ± 2.64**

Table 2: Performance comparisons on severe non-IID scenarios with CIFAR-10

	Shards = 10			Shards = 20		
	Mean Acc	Best 5%	Worst 5%	Mean Acc	Best 5%	Worst 5%
FedAvg	46.33 ± 11.69	69.40 ± 5.00	20.40 ± 5.54	81.80 ± 4.38	89.60 ± 1.36	**71.20 ± 1.94**
FedAtt	40.00 ± 8.94	52.00 ± 4.00	12.16 ± 1.40	76.09 ± 6.31	82.00 ± 2.76	44.02 ± 1.02
FedProx	45.85 ± 11.55	68.20 ± 1.94	21.00 ± 3.22	81.94 ± 4.64	89.40 ± 1.96	69.40 ± 2.33
Scaffold	45.49 ± 11.36	67.40 ± 2.24	21.60 ± 2.87	**82.00 ± 4.38**	**90.00 ± 1.10**	70.80 ± 2.99
FedPer	46.65 ± 10.71	72.73 ± 3.71	**31.95 ± 4.24**	81.71 ± 3.71	88.95 ± 1.23	69.98 ± 1.95
LG-FedAvg	45.63 ± 12.22	65.13 ± 3.21	26.23 ± 3.17	80.33 ± 3.20	87.03 ± 1.30	69.26 ± 1.73
SFL	**51.79 ± 14.04**	**78.80 ± 2.56**	23.00 ± 4.90	81.70 ± 4.70	89.60 ± 0.80	69.60 ± 1.62
SFL*	44.20 ± 11.85	67.40 ± 2.24	20.00 ± 3.85	81.25 ± 4.78	89.20 ± 0.98	68.40 ± 2.50

Table 3: Performance comparisons on relatively mild non-IID scenarios with CIFAR-10

5.4 Visualization

Fig. 3 illustrates the convergence process of SFL in different non-IID scenarios. Under the severe conditions where shards=2, there is only a small overlap in the client data distributions, which results in serious parameter conflicts during the server aggregation process, with all algorithms failing to converge. Using SFL can reduce the client parameters' conflict, thus producing a better result. When shards=5 or 10, SFL still has an obvious advantage on both convergence and robustness. When shards=20, most FL methods perform similarly because the data distribution has nearly become an IID scenarios.

Fig. 4 visualizes the comparison of the pre-defined relation graph and the learned graph by SFL. Twenty-five clients were selected to visualize their partial pre-defined structural information (top half) and the learned graph (bottom half). For image classification tasks on MNIST and CIFA-10, when sharks = 5, the learned adjacent connection approaches pre-defined adjacent values. In particular, the learned graph on traffic data is relatively ”comprehensive” than the pre-defined graph because the pre-defined graphs of PEMS-LA and PEMS-BAY are purely constructed by a road connection relationship. In contrast, the learned graph includes the long-dependence on non-connected roads. This visualization demonstrates that SFL not only learns knowledge from simple pre-defined graphs but also learns comprehensive hidden relationships among clients. Moreover, the SFL also can be combined with the existing PFL methods(e.g. LG and PER) to further improve the performance. The detailed experiment results can be found in the Appendix.

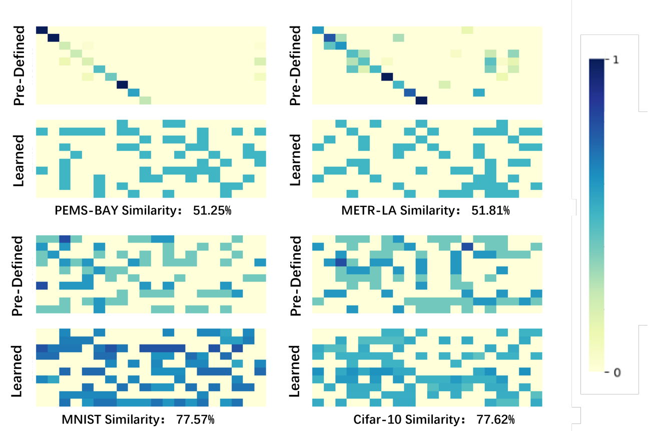

Figure 4: Visualizing adjacent matrix comparison between pre-defined graph and learned graph on four datasets

6 Conclusion

This paper is the first work to study graph-guided personalized FL. A GCN-based model aggregation mechanism has been adopted to facilitate implementation. SFL can not only leverage the graph structure and also discover comprehensive hidden relationships amongst clients. Experiments on GNN-based traffic dataset and FL-based image benchmark datasets have demonstrated the effectiveness of SFL.

References

[Arivazhagan *et al.*, 2019] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[Briggs *et al.*, 2020] Christopher Briggs, Zhong Fan, and Peter Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2020.

[Chen *et al.*, 2018] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

[Chen *et al.*, 2019] Fengwen Chen, Shirui Pan, Jing Jiang, Huan Huo, and Guodong Long. Dagcn: dual attention graph convolutional networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[Cheng *et al.*, 2021] Gary Cheng, Karan Chadha, and John Duchi. Fine-tuning is fine in federated learning. *arXiv preprint arXiv:2108.07313*, 2021.

[Deng *et al.*, 2020] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

[Fallah *et al.*, 2020] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020.

[Ghosh *et al.*, 2020] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020.

[He *et al.*, 2021] Chaoyang He, Alay Dilipbhai Shah, Zhenheng Tang, Di Fan1Adarshan Naiynar Sivashunmugam, Keerti Bhogaraju, Mita Shimpi, Li Shen, Xiaowen Chu, Mahdi Soltanolkotabi, and Salman Avestimehr. Fedcv: A federated learning framework for diverse computer vision tasks. *arXiv preprint arXiv:2111.11066*, 2021.

[Jallepalli *et al.*, 2021] Deepthi Jallepalli, Navya Chennagiri Ravikumar, Poojitha Vurtur Badarinath, Shravya Uchil, and Mahima Agumbe Suresh. Federated learning for object detection in autonomous vehicles. In *2021 IEEE Seventh International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 107–114. IEEE, 2021.

[Ji *et al.*, 2019] Shaoxiong Ji, Shirui Pan, Guodong Long, Xue Li, Jing Jiang, and Zi Huang. Learning private neural language modeling with attentive aggregation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[Jiang *et al.*, 2020] Jing Jiang, Shaoxiong Ji, and Guodong Long. Decentralized knowledge acquisition for mobile internet applications. *World Wide Web*, 23(5):2653–2669, 2020.

[Kairouz *et al.*, 2021] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[Karimireddy *et al.*, 2020] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Li *et al.*, 2018] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[Li *et al.*, 2019a] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smithy. Feddane: A federated newton-type method. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 1227–1231. IEEE, 2019.

[Li *et al.*, 2019b] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[Li *et al.*, 2020] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[Li *et al.*, 2021] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

[Liang *et al.*, 2020] Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

[Long *et al.*, 2020] Guodong Long, Yue Tan, Jing Jiang, and Chengqi Zhang. Federated learning for open banking. In *Federated learning*, pages 240–254. Springer, 2020.

[Long *et al.*, 2022] Guodong Long, Tao Shen, Yue Tan, Leah Gerrard, Allison Clarke, and Jing Jiang. Federated learning for privacy-preserving open innovation future on digital health. In *Humanity Driven AI*, pages 113–133. Springer, 2022.

[Luo *et al.*, 2019] Jiahuan Luo, Xueyang Wu, Yun Luo, Anbu Huang, Yunfeng Huang, Yang Liu, and Qiang Yang.

Real-world image datasets for federated learning. *arXiv preprint arXiv:1910.11089*, 2019.

[Ma *et al.*, 2022] Jie Ma, Guodong Long, Tianyi Zhou, Jing Jiang, and Chengqi Zhang. On the convergence of clustered federated learning. *arXiv preprint arXiv:2202.06187*, 2022.

[Mansour *et al.*, 2020] Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[Pan *et al.*, 2016a] Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang. Tri-party deep network representation. *Network*, 11(9):12, 2016.

[Pan *et al.*, 2016b] Shirui Pan, Jia Wu, Xingquan Zhuy, Chengqi Zhang, and Philip S Yuz. Joint structure feature exploration and regularization for multi-task graph classification. In *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*, pages 1474–1475. IEEE, 2016.

[Reddi *et al.*, 2020] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[Rieke *et al.*, 2020] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

[Shamsian *et al.*, 2021] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning*, pages 9489–9502. PMLR, 2021.

[Shuman *et al.*, 2013] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

[Tan *et al.*, 2021] Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning over heterogeneous devices. *arXiv preprint arXiv:2105.00243*, 2021.

[Tan *et al.*, 2022] Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Toward personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Wang *et al.*, 2017a] Chun Wang, Shirui Pan, Guodong Long, Xingquan Zhu, and Jing Jiang. Mgae: Marginalized graph autoencoder for graph clustering. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 889–898, 2017.

[Wang *et al.*, 2017b] Zhitao Wang, Chengyao Chen, and Wenjie Li. Predictive network representation learning for link prediction. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 969–972, 2017.

[Wang *et al.*, 2020] Binghui Wang, Ang Li, Hai Li, and Yiran Chen. Graphfl: A federated learning framework for semi-supervised node classification on graphs. *arXiv preprint arXiv:2012.04187*, 2020.

[Wu *et al.*, 2019] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*, 2019.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[Xie *et al.*, 2021] Ming Xie, Guodong Long, Tao Shen, Tianyi Zhou, Xianzhi Wang, Jing Jiang, and Chengqi Zhang. Multi-center federated learning. *arXiv preprint arXiv:2108.08647*, 2021.

[Xu *et al.*, 2021] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, 5(1):1–19, 2021.

[Zhang *et al.*, 2022] Yifei Zhang, Hao Zhu, Ziqiao Meng, Piotr Koniusz, and Irwin King. Graph-adaptive rectified linear unit for graph neural networks. In *Proceedings of the ACM Web Conference 2022*, pages 1331–1339, 2022.

[Zheng *et al.*, 2022] Zhaohua Zheng, Yize Zhou, Yilong Sun, Zhang Wang, Boyi Liu, and Keqiu Li. Applications of federated learning in smart cities: recent advances, taxonomy, and open challenges. *Connection Science*, 34(1):1–28, 2022.