STRUCTURED FEDERATED AGGREGATION FOR PERSONALIZING ON-DEVICE INTELLIGENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalizing on-device intelligence with privacy-preserving is an emerging requirement for the Mobile Internet and many other service areas. The recent development of federated learning is to embody personalization by tackling statistical heterogeneity across devices. However, these methods ignore the structural information between clients which can indicate a similar behavior pattern or decision logic among clients who are connected to each other in a graph. For example, the traffic condition is very similar to its adjacent blocks. Motivated by this assumption, we propose structured federated learning(SFL) to update each device's personalized model by leveraging its neighbors' local model. This problem has been formulated to a new optimization problem to integrate the prediction loss, federated aggregation, and structured aggregation into a unified framework. Moreover, it could be further enhanced by adding the structure learning component to learn the relation graph in the same optimization framework. The effectiveness of the proposed method has been demonstrated in experimental analysis by comparing it with other baselines in public datasets.

1 INTRODUCTION

In recent years, much of the success in machine learning has depended on reams of data. In society, huge amounts of data are often generated at different devices all over the place, e.g., data collected by different corporate servers and end devices like a mobile phone. Due to the privacy or integration limits, centralize all data into the server to train the machine learning algorithm is a mission impossible in certain application scenario. Aim at this situation, Federated Learning advocates a decentralized training scheme that training an acceptable machine learning model by aggregating the locally learned parameters without any data integration (McMahan et al., 2017). Since there is no need to centralize or direct access to data by outsiders, federated learning has successfully alleviated the application limits to a certain extent.

Early research like FedAvg (McMahan et al., 2017) focused on training a single shared model for all distributed end devices and expect the global model to know all training data. For that to happen, the frequently used assumption is that the data from remote ends have to be independent and identically distributed(IID) which is not always the case. For example, traffic data collected by the traffic sensors would be reflected by the geographical situation, some sensors located at school or hospitals mostly collect traffic data that is slow and stop frequently. The sensors on the freeway capture cars that run at a stable speed. As a result, data across all traffic sensors are highly skewed with significant differences in the distribution. With FedAvg, the same initial parameters for client models can optimize multiple different models due to the heterogeneity in local data distributions. Some of which may be in completely different optimization directions and those model parameters will cancel each other on the server aggregation process, resulting in an ineffective learning process and non-convergence of the global model.

Most of the existing work attempts to address the data Non-IID issue from two aspects. On one hand, the instinctive idea is the data-based approach which modifies the data distribution of clients by sharing or augmenting a certain amount of data to alleviate the data non-IID issue. Zhao et al. (Zhao et al., 2018) construct a shared proper subset of data between all clients to warm up the global

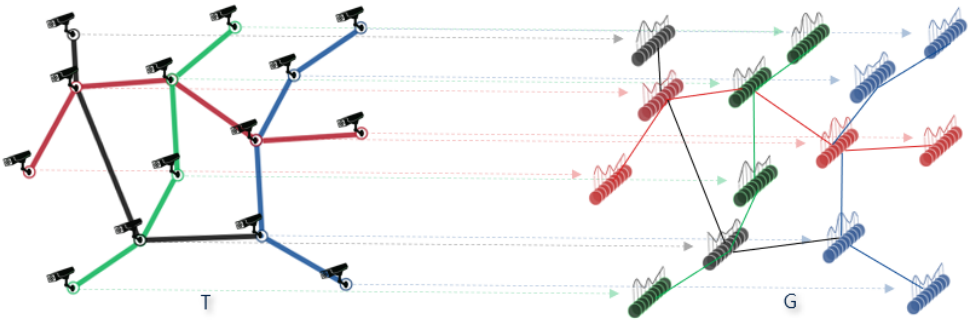


Figure 1: The traffic sensor collects local traffic information to form multiple data centers and the structural relationship between centers can be constructed according to the actual road conditions.

model. Some works (Duan et al., 2019; Li et al., 2020) regard data non-IID as a data imbalance issue between clients. Thus, clients can adjust their own data through data augmentation methods to achieve the data IID among the whole federated learning system. Although data-based approaches can significantly improve the performance of federated learning with Non-IID data, the shortcoming is obvious and fatal. Any kind of modification with data itself violates the basic and vital purpose of federated learning and increases the risk of data privacy. On the other hand, personalizing the global model with local data or an extra local layer can yield a similar result. One of the early ideas was to build a high-quality global model by employing the meta-learning mechanism(Fallah et al., 2020) so that the client could achieve better performance with few extra on-device optimizations. By different means, some work (Arivazhagan et al., 2019) has focused on improving clients' ability to personalize the global model according to their own needs. Compared with data-based approaches, this kind of method not only improves the performance of the federated learning system but also ensures that the system does not require any modification or aggregation of other data.

In addition to the two main directions mentioned above, there are some other studies (Briggs et al., 2020; Sattler et al., 2020). Several recent servers have carried out relevant analysis (Zhu et al., 2021). However, both data-based and fine-tuning-based approaches ignore a very important piece of information. With federated learning, any client in the system is bound to have a variety of complex relationships(as shown in figure 1) which have not been investigated in any previous studies. We hypothesize that the effects of this relationship include but are not limited to data distribution. Thus we came up with a novel structure federated learning framework(SFL) which employee the graph convolutional network(GCN) to exploits the inherent topological structure connecting client ends and allows the personalized parameters and model on each end collaboratively to update at the server. Specifically, the proposed SFL trains personalized models for each end, and uploads local-learned parameters to the central server to simultaneously update each personalized parameter for each client model through a GCN. Topological structure information between clients can effectively alleviate the loss of accuracy caused by data non-IID. Contrast experiments on real-world structured data sets have validated the superiority of the proposed structured federated learning framework.

The main contributions are summarized as follows:

- we reveal the ubiquitous scenarios in federated learning which clients are significantly affected by the nearby connected neighbors, while the server is able to employ the topological information between the client ends;
- we construct a novel structured federated learning architecture for personalized client models to exploit the structured information for the first time while respecting each client end's peculiarity;
- Experiments with both image and traffic datasets have confirmed our hypothesize and validate the effectiveness of our proposed structure federated learning framework.

2 RELATED WORK

Addressing the data non-IID issue has been a high concern topic in federated learning. SCAFFOLD (Karimireddy et al., 2020) proves that a drift exists in each local update when data is heterogeneous (non-IID), contributing to the unstable convergence on FedAvg, and corrects the client-drift in its local updates by controlling the direction of variance reduction in each client according to the update detection in the server model. Except for the non-IID data, another reason responsible for the unsatisfying performance of FedAvg is the single one global model for all various data distribution stored in terminal clients.

The immediate idea is to modify the distribution of the data to address the situation. (Zhao et al., 2018) proposed a data-sharing strategy, the core concept is very simple yet effective. By constructing a shared dataset within the server for a model warm-up and passing part of this shared dataset into all clients so that the client model is trained by both partial shared data and local data. Although this greatly contributes to the final performance loss caused by data non-IID, the drawbacks are obvious. Sharing data accusing servers and clients defeats the main purpose of federated learning to protect data privacy. A similar strategy is also used in (Yoshida et al., 2020). While enjoying the improvement of the performance brought by the shared data, (Yoshida et al., 2020) reduces the number of clients who need to share their data, this improves the practicability of this approach. Contrary to the idea of sharing data, (Tuor et al., 2021) proposed to reduce the client training samples to alleviate the issue. A benchmark model trained on a small benchmark dataset has been used to select the relevance of individual data samples at each client. In this way, the distribution of training data will be controlled in a certain distribution.

Other than data sharing, some studies try to trade data non-IID as a data unbalance problem then use augmentation methods to address the issue. (Duan et al., 2019) proposed to let all clients send their label distribution information to the server for calculation of the mean values, if the client's data is below those mean values, the corresponding data is generated to achieve a data balance thus solve the data non-IID issue. Compared to collecting label data directly from clients, XorMixFL (Shin et al., 2020) proposed a method that collects the encoded data samples from clients to form a balanced dataset at the server for global model training. Another typical approach for data augmentation is to train a data generator in the presence of non-IID data. Instead of training a global model with uploaded data or data seed, those approaches train a data generator with a small amount of uploaded data then send it to all clients. With this well-trained generative model, all clients can construct an IID local dataset. In any case, these great methods rely on the aforementioned data sharing operation. Unfortunately, director encrypted transfer data between clients and server violates the nature of federated learning and raise the concerns of data privacy. Thus, most of these methods are not acceptable in practice.

In contrast to those data-based approaches, efforts are being made to personalize the global model in various ways. Few efforts concentrate on on-device personalizing the optimization of the client model after receiving the global model from the server(Wang et al., 2019). Those methods normally start from the FedAvg and then perform two kinds of fine-tuning 1) train a better initial shared model and 2) local optimization. Per-FedAvg (Fallah et al., 2020) leverages Model-Agnostic Meta-Learning(MAML) to generate a global model which is easier for the clients to perform on-device personalization. Similarly, (Jiang et al., 2019) characterize the intimacy of underlying distribution between client data and measure the affinity score (distribution distance) using the 1-Wasserstein metric. By doing so, such frameworks calculate the personalized variant of federated training architecture to allow a more tailored model for each client. Besides, (Chen et al., 2018) propose FedMeta which treats it as a multi-task learning problem and train a global meta-learner instead of a global model and then send it to clients for local optimization. However, in those approaches, the training and penalization procedures are completely disconnected, which results in potentially sub-optimal personalized models. There is another type of fine-tuning-based approach that does not have this concern. They let the client models have not only base layers that are synchronized from the server but also personalization layers which only trained by local data. Both (Arivazhagan et al., 2019) and (Liang et al., 2020) followed this idea, with the former treating base layers as shallow layers and the latter the opposite.

Although the aforementioned architectures, to some extent, mitigated the performance degradation caused by the data non-IID. There is still one aspect, structured information between clients, that

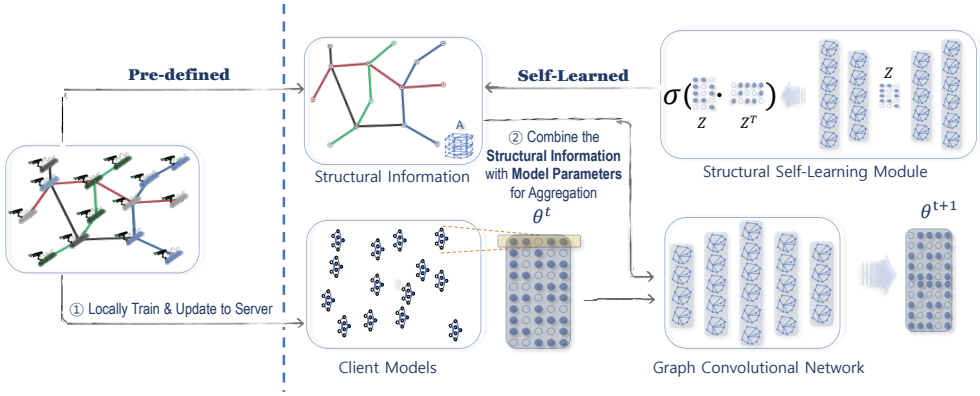


Figure 2: The overview of structured federated learning(SFL).

has always been overlooked. Whether it's data-based, fine-tuning-based, or others, they all tend to homogenize all the client ends which intrinsically against the client's (node's) peculiarity in the structured data. Therefore, we explore the use of graph convolutional networks(GCNS) to deal with the structural relationship between clients to optimize the server aggregation process. Structural data and GCNs are ubiquitous in many fields for several tasks (Pan et al., 2016b; 2017; 2016a). The most advantage of GNNs is the ability to capture the complex relationships between concepts(also called nodes). At present, the vast majority of GCNs follow the k-hop aggregation framework. Each node will only aggregate with its k-order neighbors which are ideal for the FL server to aggregate the model parameters from clients. Recently, some under-progress literature in the Arxiv (e.g., GraphFL(Wang et al., 2020)) tends to explore the topological information among the clients under the federated training scheme. However, these works merely replace the globe model with some classic graph neural networks (e.g., GCNs), never essentially leverage the inherent topological inter-dependence between the client ends. Overall, our proposed method is the first attempt to use graph neural networks to introduce structural information between clients into the server aggregation process.

3 Problem Formulation

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

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

where $F_k(W)$ is the supervised loss of the K-th participant that has dataset D_K, and all participants 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 to be modelled as a bi-level optimization problem.

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

where each participant has a uique personalized model M_i parameterised by v_i, and w^* is an optimal global model to minimise the loss as mentioned in the E.q. 1. R is the regulrasation term to control the local updates, 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, different personalized FL 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 to below bi-level optimization problem.

$$\begin{aligned} \min_{v_{1:K}} \quad & \sum_{i=1}^{K} \left(F_i(v_i) + \lambda[R(v_i, w^*) + R(v_i, u_i^*)]\right) \\ s.t. \quad & w \in \arg\min_{w} G(F_1(w), ..., F_K(w)) \\ & u_i \in \arg\min_{u} \sum_{j \in \mathcal{N}(i)} A_{j,i} S(u_j, u) \end{aligned} \tag{3}$$

where the $A_{i,j} \in \{0, 1\}$ from adjancent matrix is to indicate the neigbhouhood between two participants i and j, and the $S(w_i, w_j)$ is to measure the distance, e.g. Eculidean distance, between the i-th client and its neighbour j using their parameters w_i and w_j.

In many real application scenario, the adjacent matrix A across participants is usually not existing, thus it needs to be learnt. For this case, we need formulate the optimization problem as below.

$$\begin{aligned} \min_{v_{1:K}, A} \quad & \sum_{i=1}^{K} \left(F_i(v_i) + \lambda[R(v_i, w^*) + R(v_i, u_i^*)]\right) + \gamma G(A) \\ s.t. \quad & w^* \in \arg\min_{w} G(F_1(w), ..., F_K(w)) \\ & u_i^* \in \arg\min_{u} \sum_{j \in \mathcal{N}(i)} A_{j,i} S(u_j, u) \end{aligned} \tag{4}$$

where $G(.)$ is a regularsation term for the toplogical information of the learnt graph. In particular, we expect the learnt graph structure with adjancent matrix A is sparse while preserving proximity relationship among participants. There are various way to measure the proximity betweeen two participants, for example, distance of model parameters, local accuracy using the same model, and external descriptive features.

4.1 OPTIMIZATION

To solve the optimization problem in Eq. 3, we could 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 aggregate global model $R(v_i, w^*)$, and distance between local model and structure-based aggregated personalized model $R(v_i, u_i^*)$. Then, we conduct model aggregation at the server to update w and $\{u_i\}_i^K$. In particular, we can use a GCN (Graph Convolution Network) to implement the structure-based model aggregation by constructing the graph G: K clients represent the node in the graph, a pre-defined adjacent matrix A, and each node's attribute u_i is initialized by its local model v_i. The GCN will automatically update each node's model u_i by aggregating its neighbors' model in the graph. It will satisfy Contraint 2 in E.q. 2. Moreover, the global model will be updated by aggregating all personalized models u_i which is to satisfy Constraint 1 in Eq. 3. This gradient-based aggregation is equivalent to the read-out operator in the GCN.

To solve the optimization problem in Eq. 4, we can 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. 4, as below.

$$\min_{A} \sum_{i=1}^{K} \left(\lambda[R(v_i, w^*) + R(v_i, u_i^*)] + \gamma G(A)\right) \tag{5}$$

We can construct the graph using the learnt 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

We implement the optimization procedure in an algorithm as shown in Algorithm 1. The optimization goal will be iteratively achieved 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 exists, we will add an optional step for structure learning.

Algorithm 1 Structural Federated Learning - Server.

Initialize $\lambda_0, \eta, A, \{v_i^{(0)}\}_{i=1}^K \leftarrow v$
for each communication round t = 0, 1, ..., T **do**
  $\lambda = 1[t > 0] \times \lambda_0$
  Local updating:
  for each client i = 1, 2, ..., K in parallel **do**
    Update v_i for s local steps:
    $v_i^{(t)} \leftarrow v_i^{(t)} - \eta \nabla \left(F_i(v_i^{(t)}) + \lambda [R(v_i^{(t)}, w^{(t)}) + R(v_i^{(t)}, u_i^{(t)})] \right)$
    $v_i^{(t+1)} \leftarrow v_i^{(t)}$
  end for
  Structure-based aggregating:
  $\{u_i^{(t+1)}\}_{i=1}^K \leftarrow \{v_i^{(t)}\}_{i=1}^K$
  Update $u_i^{(t+1)}$ for m steps of $GCN(A, \{u_i^{(t+1)}\}_{i=1}^K)$
  $w^{(t+1)} \leftarrow GCN\_readout(\{u_i^{(t+1)}\}_{i=1}^K)$
  (Optional) Structure learning:
  $A \leftarrow Structure\_learn(\{v_i^{(t+1)}, u_i^{(t+1)}, w^{(t+1)}\}_{i=1}^K)$. // Could be a part of GCN with fully connected links
end for

5 EXPERIMENT

We conduct several empirical experiments on two different tasks to demonstrate SFL's superior performance and universality. First, we experiment with the traffic prediction task with pure RNN to study the performance of SFL in the real-world scenario. Second, we artificially partitioned the image dataset to construct a more challenging scenario to test the ability of SFL and specifically the structural self-learning module. We also perform a couple of case studies to better understand the role of each component. What's more, we perform a combination of SFL and other solutions to data non-IID issues to demonstrate that our method is independent of existing approaches and can be arbitrarily combined to further improve the performance of a federated learning system.

Data sets. The traffic datasets are ideal for validating our hypothesis, as it comes with natural topological structure and per-user data non-iid partition which all collected in the real world. We use four traffic datasets, METR-LA, PEMS-BAY, PEMS-D4, and PEMS-D8 to observe the performance of the SFL in different real-world scenarios. The statistics are provided in Table 1 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. We separated the training-set, validation-set, and test-set in a 70% 20% and 10% ratio. For the image classification task, we used benchmark datasets with the same train/test splits as in previous works which are MNIST(LeCun et al., 1998), CIFAR-10, and cifar-100(Krizhevsky et al., 2009). To simulate extreme data conditions to test the customization capability of the evaluated frameworks, we artificially partitioned three datasets into non-IID splits as in (McMahan et al., 2017). All datasets are being sorted then split into $n \times k$

Table 1: The Statistics of Traffic Datasets

Data	# Samples	# Nodes	# Edges	I/O Length
METR-LA	34272	207	1722	12
PEMS-BAY	52116	325	2694	12
PEMS-D4	16969	307	209	12
PEMS-D8	17833	170	137	12

	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
WaveNet	4.45	13.62	8.93	2.35	5.87	5.43	23.01	17.62	36.05	17.27	12.03	31.32
DCRNN	3.60	10.50	7.60	2.07	4.90	4.74	23.19	19.86	35.67	16.91	15.34	25.91
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 2: Traffic Prediction Performance

shards equally, and assign each of n clients k shards. This creates a pathological non-IID partition in terms of label distribution skew among devices, as most clients will only have examples at most of k classes.

Models and Frameworks. 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), FedPer(Arivazhagan et al., 2019) and LG-FedAvg(Liang et al., 2020). A brief introduction of the frameworks is provided in Appendix. 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 MNIST, we implement a simple CNN architecture same as in (McMahan et al., 2017) and we use ResNet-18(He et al., 2016) for a more complex task, CIFAR-10, and CIFAR-100. For a fair comparison, without any additional statement, all reported results are based on same training setting as follow, we employ SGD with learning rate 0.001 as the optimizer for all training operation, use 128 for batch size and the number of total communication round as 20. It is worth mentioning that higher capacity models and larger communication rounds can always bring higher performance on any of those datasets. As such, the gold of our experiment is to compare the relative performance of these frameworks with the same basic models rather than the specific number.

Performance Comparison. The performance of SFL in traffic prediction task comparing with other baselines are provided in Table 2. We use SFL* denotes 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 whole result can be looked at in three parts. First, for METR-LA and PEMS-BAY there is a 25% and 18% performance improvement in terms of MAE separately. Because the two datasets have relatively more nodes and complex structural information(edges) as stated in Table 2, using 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 methods 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 the absence information, thus bringing more 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, the relationships are fragile. The results confirm that the performance lower-bound of SFL remains slightly better than the traditional methods.

We also ran experiments on the image dataset to further validate the SFL. Table 3 state our method's stable performance in three levels of image prediction tasks. With the mimics of extreme data non-iid environment state before, the attention mechanism in FedAtt is not only useless but also leads

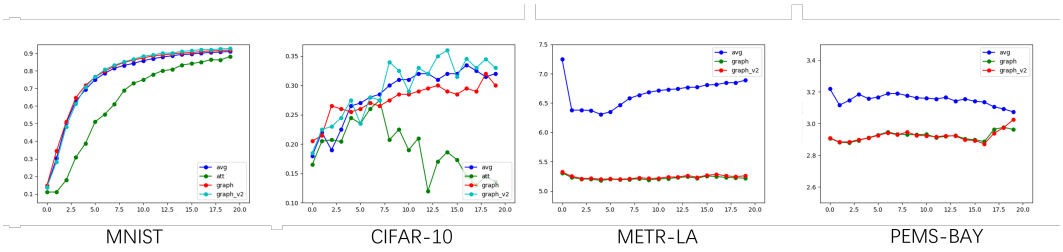
to the complete failure of convergence. We can make two observations regards the SFL from the results. 1) the pre-defined structural information helps the federated learning process and provides better overall performance. 2) Construct structural information based on pure label skew is not sufficient enough, our structural self-learning module can do better and discover multiple types of relationships between clients.

To sum up, the results confirm our conjecture about the structural relationship that exists between federated learning clients. By using GCN to aggregate client model parameters with their structural information, the data non-IID issue has been alleviated thus greatly improve the final performance. In addition, the structure self-learning module can complete the missing structural information in an unsupervised fashion. All of these achievements are based on compliance with privacy-preserving principles.

	FedAvg	FedAtt	FedPer	LG-FEDAVG	SFL	SFL*
MNIST	90.96	88.14	91.26	91.53	91.77	**92.65**
CIFAR-10	33.50	12.94	33.24	34.43	32.20	**36.82**
CIFAR-100	11.50	4.70	11.00	11.85	12.03	**12.20**

Table 3: Image Classification Performance

Comparison of convergence. The experiment results prove our conjecture that the topological structure relationship between clients can effectively alleviate the performance degradation caused by data non-IID. We also observed the convergence process of frameworks in Fig5 to better understand their behaviors in different datasets. With a simple task like MNIST, our SFL with structure learning is fast and stable compare with others. In other datasets, extreme data environments presented different challenges, and our framework demonstrated superior resistance to interference.



Visualization. We visualized the structural information from samples MNIST and PEMS-BAY respectively. In Figure 3, the small squares in different colors represent the adjacent connection between sample clients to others, with deeper color, represent a stronger connection relationship. In traffic prediction tasks, the learned adjacent connection approaches the pre-defined adjacent values, but the learned adjacent on image classification tasks is a little "deeper" than the pre-defined. This is consistent with our observations earlier. Two different manifestations are because the pre-defined value on MNIST is constructed by pure label skew which is more incomplete than that of in PEMS-BAY. And the missing structural information is supplemented through the self-learning module. This again shows the ability of SFL to learn complex client relationships.

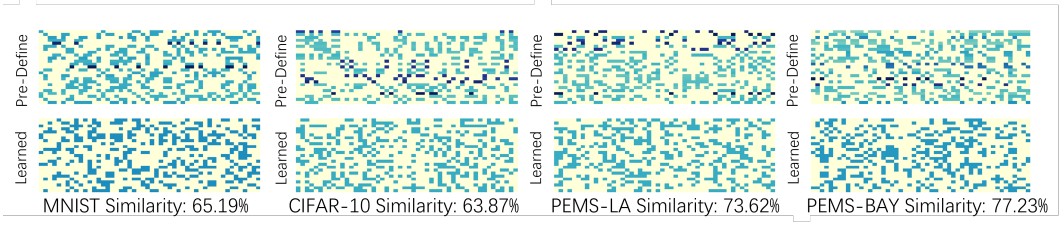


Figure 3: Visualization of adjacent matrix

Ablation study: We take METR-LA as the observation object to study the sensitivity of SFL performance to the choice of hyper-parameter. First, we fixed the rest of the framework and looked at the training epoch of the structure learning process, the result shows only a small amount of training is needed to obtain relatively stable results without a heavy computing burden on the server. Second, we study the effect of communication frequency and communication round, with a consistent total client training epoch, only a small amount of communication is required to ensure a relatively smooth performance. However, blindly increasing the total training epoch of the client model could not guarantee better performance. Overall, the performance of SFL does not rely on radical parameter adjustment, it only takes a little extra resource to get the stable performance. The observation figures related are in Appendix.

Compatibility analysis: Unlike most personalized methods based on FedAvg, the SFL tackle the data non-iid issues by involving the structural information between clients during the server aggregation step which is a new perspective that had never been explored. Therefore, it can theoretically be combined with the existing solution to further improve the performance. Motivated by this assumption, we conduct experiments that superimposed other personalization strategies on the SFL for both traffic prediction and image classification tasks. We trained the PEMS-BAY and MNIST datasets in the way described above for 20 communication rounds. Instead of applying personalized fun-tuning based on the shared global model from FedAvg, we apply the personalization process on top of the SFL, the result is provided in Table 4. In both tasks, the SFL can combine with existing methods to further improve the performance of federated learning without any conflict.

	FedAvg	FedAtt	SFL	SFL+LG	SFL+PER
MNIST(Acc %)	90.96	88.14	91.77	**92.36**	91.96
PEMS-BAY(MAE)	10.73	12.58	6.47	4.95	**4.82**

Table 4: Compatibility Performance

6 CONCLUSION

In this paper, we study a completely new scenario for the first time. Due to privacy concerns and the cost of data interchange, multiple small amounts of data are generated and stored separately on different devices. Each device has the ability to communicate with the server(not for data transition) and there are structural relationships between all clients that are either pre-defined on the server or can be obtained using unsupervised learning. For this scenario, we introduce a graph neural network into a federated learning schema to form a new framework and validate it on both real-world and artificial datasets. The experiment results demonstrate a brand new aggregation mechanism to boost the server aggregation effectiveness without infringing on clients' data privacy. The results in Table 2 and Table 3 show how our method performs in different scenarios, the richness of structural information directly determines the extent to which our method can improve the overall performance. We also observe the convergence curve of the frameworks in different datasets which are good agreement with the previous results. Finally, we test the SFL for compatibility with other existing personalized frameworks. From the empirical results, the excellent performance does not conflict with existing data non-IID optimization methods.

REFERENCES

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

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)*, pp. 1–9. IEEE, 2020.

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.

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

Moming Duan, Duo Liu, Xianzhang Chen, Yujuan Tan, Jinting Ren, Lei Qiao, and Liang Liang. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In *2019 IEEE 37th international conference on computer design (ICCD)*, pp. 246–254. IEEE, 2019.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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)*, pp. 1–8. IEEE, 2019.

Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

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*, pp. 5132–5143. PMLR, 2020.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

Yan Li, Ethan X Fang, Huan Xu, and Tuo Zhao. International conference on learning representations 2020. In *International Conference on Learning Representations 2020*, 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.

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*, pp. 1273–1282. PMLR, 2017.

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

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*, pp. 1474–1475. IEEE, 2016b.

Shirui Pan, Jia Wu, Xingquan Zhu, Guodong Long, and Chengqi Zhang. Task sensitive feature exploration and learning for multitask graph classification. *IEEE transactions on cybernetics*, 47 (3):744–758, 2017.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 2020.

MyungJae Shin, Chihoon Hwang, Joongheon Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Xor mixup: Privacy-preserving data augmentation for one-shot federated learning. *arXiv preprint arXiv:2006.05148*, 2020.

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.

Tiffany Tuor, Shiqiang Wang, Bong Jun Ko, Changchang Liu, and Kin K Leung. Overcoming noisy and irrelevant data in federated learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 5020–5027. IEEE, 2021.

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.

Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Françoise Beaufays, and Daniel Ramage. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 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.

Naoya Yoshida, Takayuki Nishio, Masahiro Morikura, Koji Yamamoto, and Ryo Yonetani. Hybrid-fl for wireless networks: Cooperative learning mechanism using non-iid data. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–7. IEEE, 2020.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *arXiv preprint arXiv:2106.06843*, 2021.

A APPENDIX

A.1 BASELINE DETAILES

privacy non-preserved models

- WaveNet: A deep neural network for generating raw temporal data.
- DCRNN: Using bidirectional random walks on the graph to capture the spatial dependency and encoder-decoder architecture for temporal dependency.
- STGCN: A spatial-temporal graph model which employ the dilated convolution and graph convolution operation for time series prediction task.
- Graph WaveNet: A spatial-temporal graph mode which learning the adaptive dependency matrix through node embedding to capture the hidden spatial dependency in the data.

privacy preserved models

- FedAvg: A federated learning framework that averaging all clients' information during the server aggregation process.
- FedAtt: The extensions of FedAvg which employ attention mechanism for clients information aggregation.
- FedPer: A personalized federated learning framework which trains shallow shared global layers to extracts high-level representations and uses personalization layers for classifications.
- LG-FEDAVG: In contrary to FedPer, the personalization layers are shallow layers of the neural network and most layers of client models are shared, global models.
- SFL: A graph convolutional network is used to introduce the structural information into the process of server aggregation.

privacy non-preserved models for traffic prediction

- WaveNet: A deep neural network for generating raw temporal data.
- DCRNN: Using bidirectional random walks on the graph to capture the spatial dependency and encoder-decoder architecture for temporal dependency.
- STGCN: A spatial-temporal graph model which employs the dilated convolution and graph convolution operation for the time series prediction task.
- Graph WaveNet: A spatial-temporal graph mode which learning the adaptive dependency matrix through node embedding to capture the hidden spatial dependency in the data.

A.2 DATASETS DESCRIPTION

- METR-LA: Traffic flow dataset collected from 207 loop sensors at the highway of Los Angeles County over 4 months from Mar 1st, 2012 to Jun 30th, 2012.
- PEMS-BAY: Traffic flow dataset collected by 325 sensors in the Bay Area over 6 months.
- PEMS-D4: Traffic flow dataset collected by the Caltrans Performance Measurement System(PEMS) every 30 seconds in San Francisco Bay Area, containing data from 307 sensors.
- PEMS-D8: Traffic data in San Bernaridino from July to August in 2016, data from 170 sensors are being collected.
- MNIST: The MNIST database of handwritten digits, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST.
- CIFAR-10: The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.
- CIFAR-100: This dataset is just like the CIFAR-10, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class.

A.3 EFFICIENCY ANALYSIS

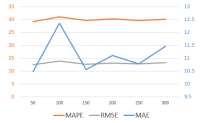
Training a model over decentralized data with federated learning is normally considered ineffective due to the massive amounts of communication(Konečnỳ et al., 2016). So we paid special attention to the analysis and experiment to show the extra cost of SFL compared to traditional methods. The additional cost of federated learning is concentrated in two parts compared to centralized training. 1) communication time where the parameters transfer between server and clients. 2) aggregation time which is cost by the server to aggregate clients' information to generate a new model. For the first part, our SFL didn't impose any additional transmission volume or frequency. Only model parameters need to be transferred between server and clients. For the second part, we need to perform k additional graph convolution operations and train the structure learning module to learn the information if needed. The number of clients would affect the training time by change the size of the adjacency matrix A and parameter matrix Θ which will not affect the computation complexity. Thanks to PyTorch's optimization of matrix multiplication, the extra computation cost is negligible. Given the number of the client as n, For structure self-learning module, although the computation complexity is $O(n^2)$, it only happened at every communication round with a reasonable training time. Table 5 provide more intuitive results to show that our method only requires limited extra resource.

	Training Time (Aggregation Time)		
	2-layer	3-layer	5-layer
FedAvg		9.28(0.016)	
FedAtt		7.47(0.0617)	
SFL	5.14(0.1265)	5.03(0.2056)	4.62(0.2938)
SL-SFL	5.67(0.051)	5.43(0.060)	5.24(0.05638)

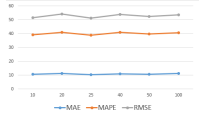
Table 5: Time cost of training and aggregation (wall clock)

A.4 ABLATION STUDY

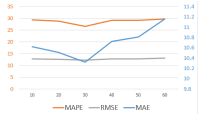
(i): structural learning epoch (j): communication frequency (k): trainig epoch between communication



(a)



(b)



(c)