TWO IMPORTANT THINGS TO DO BEFORE FEDERATED TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) aims to coordinate multiple data sources to train a machine learning model with data privacy. The existing FL mainly focuses on privacy preservation, algorithm optimization, and reducing communication costs and so on. However, there are still some unexplored issues in federated learning. In this work, we proposed two novel questions on FL algorithm selection and clients contribution prediction more applicable to the current FL service scenario. We suggest some solutions and thoughts on the above two issues.

1 INTRODUCTION

Federated Learning (FL) was proposed to learning a global model through multiple data owners (federation) who are coordinated by a centralized server without sharing raw data (McMahan et al., 2017). In real-world situations, with increasing demand for model training with privacy, there are servers in data market starting to provide FL service for different federations to collaboratively train a model for a global task. Consider a new federation requests to purchase the FL service, the server should firstly select an appropriate training algorithm and conduct a trail training, this is the process of *algorithm selection*. This process could provide the situation that some clients have high-quality data but they might be irrelevant in this particular training task. After deciding the training algorithm, the participating clients will provide their data in this multi-player game. The malicious clients and free-riders should be detected ahead, and the contribution calculation should be more transparent. Some research consider evaluating the contribution of clients (Song et al., 2019; Wang et al., 2020; 2022; Liu et al., 2022) using gradient-based Shapely Value (G-SV). However, the drawback of their evaluation schemes is the "post-hoc compensation" , many risk-averse clients may not be willing to fully provide the good enough data to collaborate with others without knowing other information. If the clients could obtain their potential future contribution in this task, the collaboration will be more effective, and this is the process of *contribution prediction*.

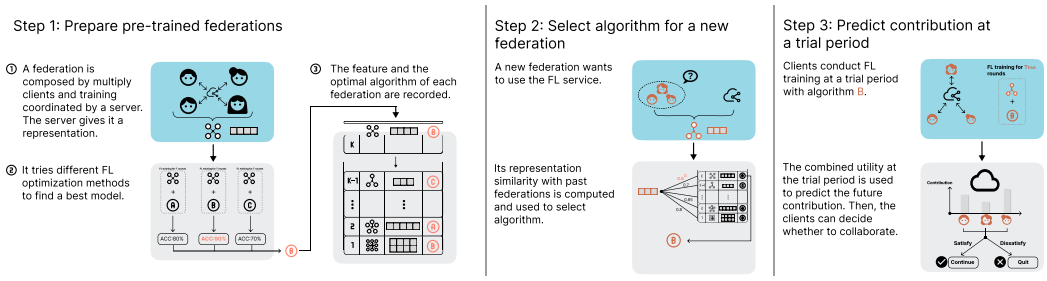

Figure 1: Pipeline for Algorithm Selection and Contribution Prediction

2 PROBLEM FORMULATION

Suppose there are K different federations, and each federation has client set $N_k, k \in [1, K]$, where $|N_k| = n_k$ and n_k may be different in different federations. In a particular k-th federation, each client has data set $D_i, i \in [1, n_k]$. Following the classical framework FedAvg

(McMahan et al., 2017), a central server aggregates local trained models from clients at the end of each round and updates the global model. The G-SV of client i in t-th round is defined as: $\phi_i^t = \sum_{S \subseteq N \setminus \{i\}} \binom{|N|-1}{S}^{-1} \left(V^t\left(S \cup \{i\}\right) - V^t\left(S\right)\right)$, where S indicates any subset of clients without client i. $V^t(S)$ is the utility function as the performance of the model aggregated by local models of set S in the t-th round on the global test set. We use $\{V_k^t(\{S\}), S \subseteq N_k\}$ to represent the performance of all subsets of client set N_k in k-th federation , simplified as $U_k^t(N_k)$, where $V_k^t(\{S\})$ is the performance of k-th federation in t-th round.

2.1 Algorithm Selection

In this process, each federation should initially conduct one round FL training and record the utility set $U_k^1(N_k), k \in [1, K]$ as the representation feature of this federation, which is equivalent to $\{V_k^1(S_\alpha^k), S_\alpha^k \subseteq N_k, \alpha \in [1, n_k]\}$, where S_α^k is a particular subset of N_k that $|S_\alpha^k| = \alpha$. We measure the similarity between federation k and j via

$$d_{k,j} = \frac{\sum_{\alpha=1}^{m} \mathrm{Var}(S_\alpha^k) \times \mathrm{Var}(S_\alpha^j)}{\sqrt{(\sum_{\alpha=1}^{m} \mathrm{Var}(S_\alpha^k)^2) \times (\sum_{\alpha=1}^{m} \mathrm{Var}(S_\alpha^j)^2)}}, \tag{1}$$

where $m = \min(n_k, n_j)$, and choose the algorithm of the client k by $\arg\max_k d_{k,j}$ for the new client j. Table 1 shows the experiment results.

2.2 Contribution Prediction

After selecting the FL algorithm, we can not accurately predict the future contribution with G-SV due to its unstable transition, as we observed in Figure 3. As another attempt, the new federation should be trained for R_{trial} rounds in **a trial period**. We decompose ϕ_i^t into $N-1$ combined utility sets and find that the overall changes of each set are more stable. More details can be found in Figure 4

Proposition 1 *Given the utility set of a federation, the gradient-based SV of client i in t-th round can be decomposed to $|N|-1$* ***combined utility sets****:*

$$\phi_i^t \approx \sum_{n=1}^{|N|-1} \left(\frac{1}{\binom{|N|-1}{n-1}} + \frac{1}{\binom{|N|-1}{n}} \right) \sum_{i \in S^n \subseteq N} V_t(S^n).$$

The proof of Proposition 1 is shown in Appendix A.2. Let $P_n^t = \sum_{i \in S^n \subseteq N} V_t(S^n)$ be the sum of a combined utility set which it is used to for prediction. For clients i, it owns $N-1$ combined utility set, each D_n is composed by $\{P_n^t, t \in [1, R_{trial}]\}$. Then, we use power law function f to fit the training data D_n by least squares minimization and obtain model f_n. Finally, we use model f_n to predict the future contribution $\hat{P}_n^T$ and combine $\sum_{n=1}^{|N|-1} \hat{P}_n^T$ into $\hat{\phi}_i^t$.

3 Conclusion

We proposed two novel questions about how to select a suitable algorithm for a new federation and predict future contribution of the clients to make the FL process more complete. For a new federation, we utilized its utility set as its representation to compare with past federations and select the optimal algorithm. Then, we find gradient-based sv is unstable and provide an approach to decompose the utility set into different combined utility values which changes more stably. Our experimental results gives some interesting findings about the two questions.

URM Statement

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2023 Tiny Papers Track.

REFERENCES

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. *Advances in Neural Information Processing Systems*, 31, 2018.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pp. 2242–2251. PMLR, 2019.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1167–1176. PMLR, 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.

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

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

Xin-Chun Li and De-Chuan Zhan. Fedrs: Federated learning with restricted softmax for label distribution non-iid data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 995–1005, 2021.

Zelei Liu, Yuanyuan Chen, Yansong Zhao, Han Yu, Yang Liu, Renyi Bao, Jinpeng Jiang, Zaiqing Nie, Qian Xu, and Qiang Yang. Contribution-aware federated learning for smart healthcare. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 12396–12404, 2022.

Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. Bounding the estimation error of sampling-based shapley value approximation. *arXiv preprint arXiv:1306.4265*, 2013.

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.

Lloyd S Shapley et al. A value for n-person games. 1953.

Tianshu Song, Yongxin Tong, and Shuyue Wei. Profit allocation for federated learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pp. 2577–2586. IEEE, 2019.

Junhao Wang, Lan Zhang, Anran Li, Xuanke You, and Haoran Cheng. Efficient participant contribution evaluation for horizontal and vertical federated learning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 911–923. IEEE, 2022.

Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. A principled approach to data valuation for federated learning. *Federated Learning: Privacy and Incentive*, pp. 153–167, 2020.

Jingfeng Zhang, Cheng Li, Antonio Robles-Kelly, and Mohan Kankanhalli. Hierarchically fair federated learning. *arXiv preprint arXiv:2004.10386*, 2020.

Wennan Zhu, Peter Kairouz, Brendan McMahan, Haicheng Sun, and Wei Li. Federated heavy hitters discovery with differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pp. 3837–3847. PMLR, 2020.

A APPENDIX

A.1 RELATED WORKS

A.1.1 FEDERATED LEARNING

As the society growing emphasis on privacy and safety of the client data, it is difficult to learn a well-performed model with insufficient data. To solve the problem, federated learning was first proposed by (McMahan et al., 2017) to allow multiple clients to jointly train a shared model coordinated by a centralized server without exposing their local data. Following (McMahan et al., 2017), many researchers have made progress on improving the efficiency and effectiveness of algorithm (Li et al., 2020; Karimireddy et al., 2020; Acar et al., 2021; Li & Zhan, 2021) and preserving the privacy of client data (Agarwal et al., 2018; Zhu et al., 2020). (Li et al., 2020) claimed that FedAvg can not converge well when the setting becomes more realistic which means the huge difference between the data distribution of clients (statistical heterogeneity), and proposed FedProx to add a proximal loss term to FedAvg for alleviating statistical heterogeneity. To solve the misalignment of optimization goals between the global model and local models of clients, (Karimireddy et al., 2020; Acar et al., 2021) respectively introduced control variates and a dynamic regularizer into local training and global model aggregation. (Li & Zhan, 2021) proposed to "Restricted Softmax" to limit classification weights of missing classes under scenarios of missing classes. They successfully saved transmission costs and improved the training efficiency of FL. Different from the aforementioned works, we mainly focus on the part before the optimization algorithm in the federated process e.g., algorithm selection and contribution prediction.

A.1.2 CONTRIBUTION EVALUATION IN FL

A fair and rational contribution evaluation is critical in the incentive mechanism. Many contribution evaluation methods are derived from Shapley Value (SV) which is the classical solution in cooperative game theory(Shapley et al., 1953; Maleki et al., 2013; Ghorbani & Zou, 2019; Jia et al., 2019). However, it is because not only the SV calculation growing exponentially with the number of clients, but the heavy computation and communication overheads in FL, SV can not be directly applied to FL. Most of the existing studies have computed approximate SV by utilizing gradients of all clients to reconstruct a sub-model instead of retraining a sub-model(Song et al., 2019) which we call gradient-based SV. To further improve the efficiency of SV computation, some works introduced Monte-Carlo sampling and designed round truncation scheme to reduce unnecessary steps(Song et al., 2019; Wang et al., 2020; 2022; Liu et al., 2022). In previous works, only when the contribution is computed by gradient-based SV after training terminates, can we distinguish the contributions of clients and find some negative clients, but their attack has already caused overall losses. In addition, we find the gradient-based SV between each round has no obvious rules which are difficult to predict future contribution directly as shown in Figure 3. In addition, current studies about federated learning have never been concerned with selecting which appropriate algorithm for a new particular federation. (Zhang et al., 2020) claimed that data evaluation based on SV is not suitable for FL due to its model dependency and evaluation metric dependency. Their experiment found that SV of ten clients between five learning algorithms to learn a binary classification model returned different outcomes, whereas we think that the contribution of clients in FL should be related to data, model, FL algorithms, etc. For example, in the non-identically distributed (non-iid) case, for a client with a rare class, different algorithms will lead to different contributions. Moreover, due to data privacy and data heterogeneity, it is difficult to choose one federated learning algorithm to tackle all the scenarios. This leads to a new question: how to select an appropriate FL algorithm for each federation?

A.2 PROOF OF PROPOSITION 1

The contribution of client i in t-th round is:

$$\phi_i^t = \sum_{S \subseteq N \setminus \{i\}} \frac{V_t(S \cup \{i\}) - V_t(S)}{\binom{|N|-1}{|S|}} \tag{2}$$

$$= \sum_{n=1}^{|N|-1} \sum_{S^n \subseteq N \setminus \{i\}} \frac{V_t\left(S^n \cup \{i\}\right) - V_t\left(S^n\right))}{\binom{|N|-1}{n}} + \frac{V_t(\{i\}) - V_t(\emptyset)}{\binom{|N|-1}{0}} \tag{3}$$

$$= \sum_{n=1}^{|N|-1} \frac{1}{\binom{|N|-1}{n}} \left\{ \sum_{i \in S^n \subseteq N} V_t\left(S^{n+1}\right) - \sum_{S^n \subseteq N \setminus \{i\}} V_t\left(S^n\right) \right\} + \frac{V_t(\{i\}) - V_t(\emptyset)}{\binom{|N|-1}{0}} \tag{4}$$

$$= \sum_{n=1}^{|N|-1} \frac{1}{\binom{|N|-1}{n}} \left\{ \sum_{i \in S^n \subseteq N} V_t\left(S^{n+1}\right) - \left(\sum_{S^n \subseteq N} V_t\left(S^n\right) - \sum_{i \in S^n \subseteq N} V_t\left(S^n\right) \right) \right\} + \frac{V_t(\{i\}) - V_t(\emptyset)}{\binom{|N|-1}{0}} \tag{5}$$

$$= \sum_{n=1}^{|N|-1} \frac{1}{\binom{|N|-1}{n}} \left\{ \sum_{i \in S^n \subseteq N} V_t\left(S^{n+1}\right) + \sum_{i \in S^n \subseteq N} V_t\left(S^n\right) \right\} - \sum_{n=1}^{|N|-1} \sum_{S^n \subseteq N} \frac{V_t\left(S^n\right)}{\binom{|N|-1}{n}} + \frac{V_t(\{i\}) - V_t(\emptyset)}{\binom{|N|-1}{0}} \tag{6}$$

$$= \sum_{n=1}^{|N|-1} \left(\frac{1}{\binom{|N|-1}{n-1}} + \frac{1}{\binom{|N|-1}{n}} \right) \sum_{i \in S^n \subseteq N} V_t(S^n) \underbrace{+ V_t(N) - V_t(\emptyset) - \sum_{n=1}^{|N|-1} \sum_{S^n \subseteq N} \frac{V_t\left(S^n\right)}{\binom{|N|-1}{n}}}_{\text{Shared part}} \tag{7}$$

$$\approx \sum_{n=1}^{|N|-1} \left(\frac{1}{\binom{|N|-1}{n-1}} + \frac{1}{\binom{|N|-1}{n}} \right) \sum_{i \in S^n \subseteq N} V_t(S^n) \tag{8}$$

$$\tag{9}$$

B EXPERIMENTS

In this part, we will briefly introduce the datasets and models involved in the experiments, as well as the performance on two tasks. More experiment details can be found in the Appendix.

B.1 EXPERIMENT SETUP

We refer to previous works on FL (Acar et al., 2021) and use CNN for CIFAR-10(Krizhevsky et al., 2009) and CIFAR-10s, both of them are used to image classification tasks. CIFAR-10s includes 10 'fine' classes randomly sampled from CIFAR-100. To construct different kinds of federations, we choose two non-IID data partitioning ways which include Dirichlet distribution partitioning and label partitioning. Label partitioning could generate more heterogeneous data. We choose two FL algorithms FedDyn (Acar et al., 2021) and FedRS (Li & Zhan, 2021) as available algorithms of federations.

B.2 EXPERIMENT RESULTS AND ANALYSIS

Algorithm Selection. We simulate three past federations with various data heterogeneity levels (Dir<label) on CIFAR-10s. Similarly, three new federations are constructed by CIFAR-10 and the same heterogeneity levels. The task of algorithm selection is to select the best FL algorithm for each new federation. The performance of FedDyn and FedRS on four federations are listed in Table 1. From the similarity matrix in Figure 2, CIFAR-10 Dir (New) matches with CIFAR-10s Dir (Past). Similarly, CIFAR-10 Label (New) is closest to CIFAR-10s Dir (Past). However, the optimal algorithms of CIFAR-10 Label and CIFAR-10s Dir are different in Table 1. This may be because

Federations/Algorithms			FedDyn	FedRS
1	cifar10	Dir	**84.6**	81.46
2		Label	60.14	**67.48**
$3 \rightarrow 1$	cifar10s	Dir	**59.1**	56.8
$4 \rightarrow 1$		Label	21.9	**46.0**

Table 1: The similarity between different federations. $i \rightarrow j$ denotes that the past federation j is most similar to the new federation i. Red indicates the best accuracy of each federation.

there is fewer data on CIFAR-10s Dir, which makes it difficult to capture the characteristics of the federation.

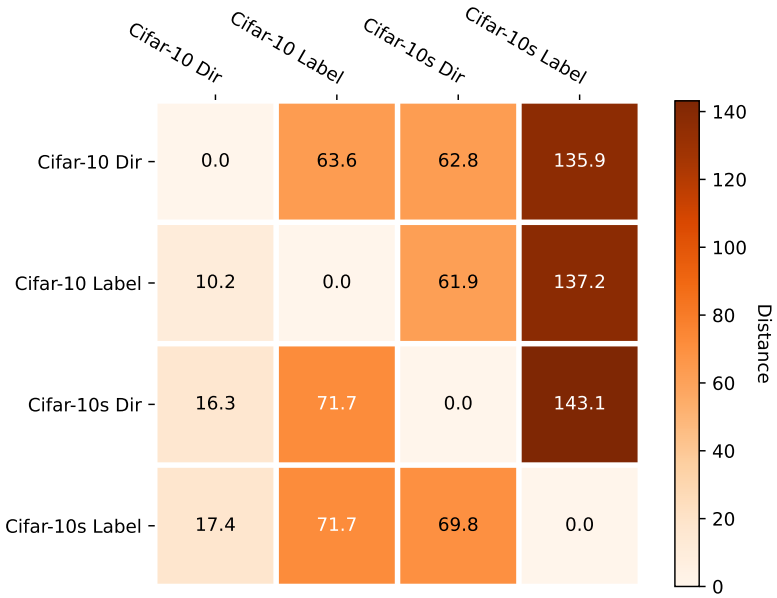

Figure 2: The similarity between new and past federations

Contribution Prediction. Figure 3 shows the L2 norm between i-th round SV and 300-th round SV changes unstably which lead to more difficulty to predict. Therefore, we compose the SV value into the combined utility values transition of which is more regular.

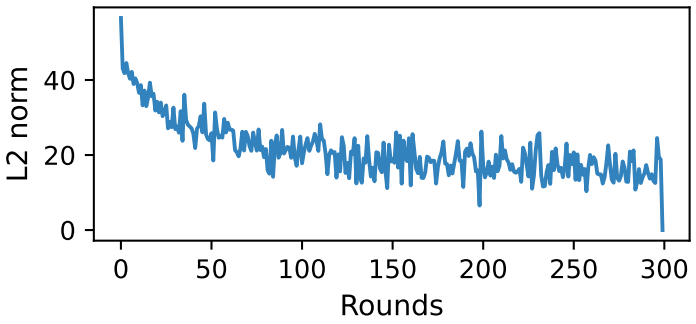

Figure 3: The instability of G-SV in FL training.

B.3 ANALYSIS OF UTILITY TRANSITION

Figure 4 shows the variation of the combined utility of client 0-3 in different communication round. we find it accord with power law and become our prediction goal.

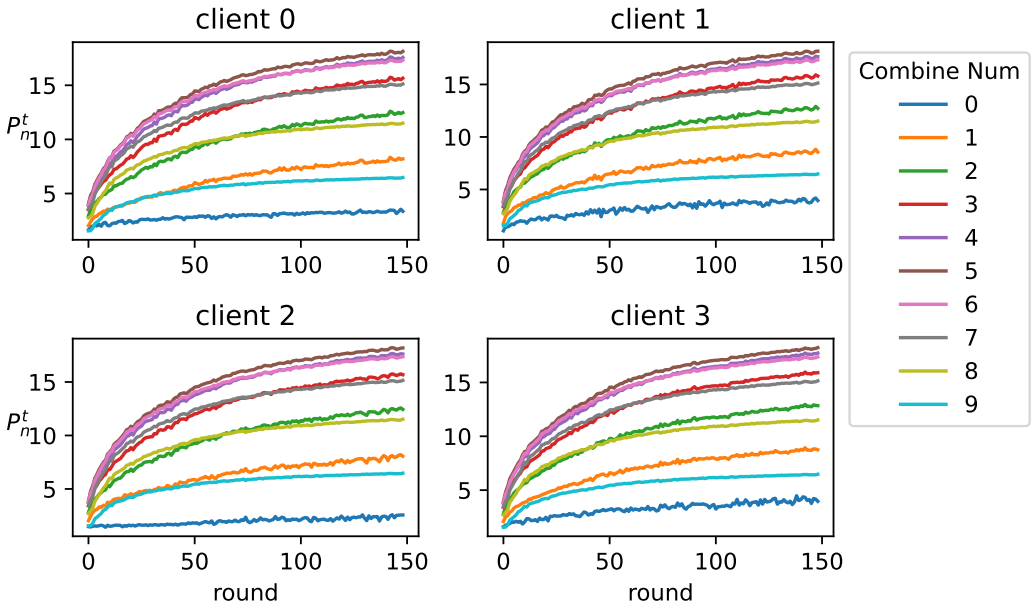

Figure 4: The transition of combined utility values

B.4 DATA DISTRIBUTION AND MODEL PERFORMANCE

The first row in Figure 5 illustrates that the data distribution of 4 federations and the distribution of federations with the same partitioning is similar. Then, the second row shows the performance of two FL algorithms in different federations. We find FedRS do well in scenarios of missing class, meanwhile, FedDyn has better performance on less heterogeneous data.

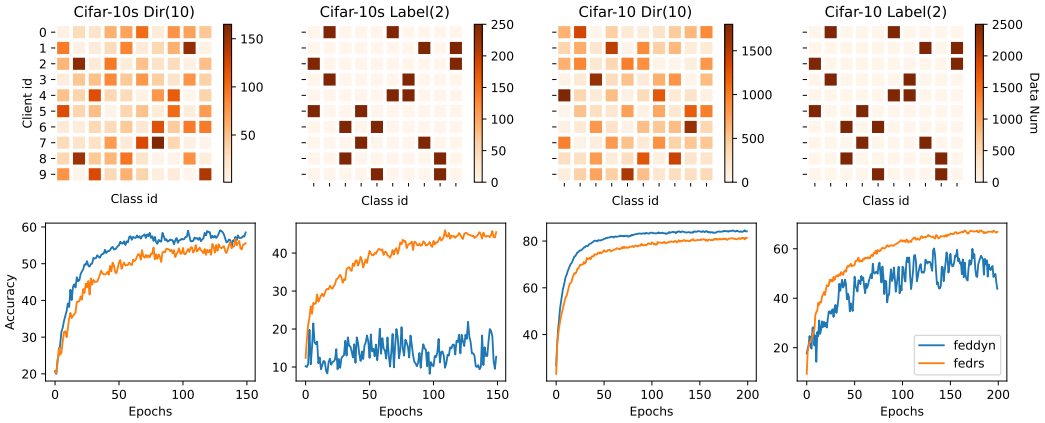

Figure 5: Comparsion of four federations