Targeting Collaborative Fairness in Federated Learning

Abstract

In current deep learning paradigms, the standalone framework tends to result in overfitting and low utility. This problem can be addressed by collaborative/federated learning that leverages a parameter server to aggregate local model updates. However, all the existing collaborative learning frameworks have overlooked an important aspect of participation: collaborative fairness. In particular, all parties can get same or similar models, even the ones who contributes nearly nothing. To address this issue, we make the first-ever investigation on the collaborative fairness, and propose a novel reputation-based method to facilitate fairness in collaborative learning. We experimentally demonstrate that using our proposed method, fairness and accuracy in federated learning can be effectively achieved at the same time. Moreover, our framework provides a viable solution to detect and isolate the low-reputation party such as the free-riders in the collaborative learning system.

1 Introduction

Training complex deep networks on large-scale datasets is computationally expensive and may not be feasible for a single party in practice. Moreover, the data owned by a single party may be very homogeneous, resulting in overfitting which negatively impacts accuracy when the model is applied to previously unseen data, *i.e.*, poor generalizability. Therefore, there is a high demand to perform federated learning [Yang *et al.*2019].

In the current federated learning paradigm [McMahan *et al.*2016], all participants receive the same federated model at the end of collaborative model training regardless of their contributions. This is obviously unfair, because all parties including the low-contribution parties can get the same global model. However, in reality, some parties may contribute more compared with other parties, while some parties may contribute nearly nothing or even negatively. This also makes the paradigm vulnerable to free-riding participants. For example, several banks may want to work together to build model to predict the creditworthiness of small and medium enterprises. However, but larger banks with more data maybe reluctant to train their local model based on high quality local data for fear of smaller banks benefiting from the shared FL model and eroding its market share [Yang *et al.*2019]. Without the guarantee of privacy and the promise of collaborative fairness, participants with high quality and large datasets may be discouraged from joining federated learning, thereby negatively affect the formation of a healthy FL ecosystem. Existing research on fairness mostly focuses on protecting sensitive attributes or reducing the variance of the prediction distribution across participants [Cummings *et al.*2019, Jagielski *et al.*2018]. The problem of treating federated learning participants fairly remains open [Yang *et al.*2019].

To overcome this problem, it is essential to develop a fair federated learning framework that respects collaborative fairness and accuracy at the same time. In this paper, we address the problem of treating FL participants fairly based on their contributions to build a healthy FL ecosystem. We refer to the proposed framework as the Collaborative Fair Federated Learning (CFFL) framework. Unlike existing work such as [Yu *et al.*2020] which uses monetary rewards to incentivize good behaviour, our proposed solution fundamentally changes the current FL paradigm so that participants may not receive the same FL model in the end. Instead, each of them will receive a final FL model with performance reflecting their individual contributions to the federation. CFFL achieves collaborative fairness through the evaluation of reputation that considers the contribution of each party during collaborative learning process.

To the best of our knowledge, this paper is the first to achieve collaborative fairness in federated learning through adjusting the level of performance of the version of the FL model allocated to each participant based on his contribution. Extensive experiments based on benchmark datasets demonstrate that CFFL achieves high fairness, delivers comparable accuracy to existing distributed deep learning framework, and outperforms standalone deep learning framework.

The rest of this paper is organized as follows. Section 2 reviews the related literature on fairness in federated learning which are major problems we aim to tackle. Section 3 presents technical details of the proposed CFFL framework. Section 4 evaluates the performance of CFFL in terms of accuracy and fairness for different SGD frameworks, followed by discussions in Section 5. Section 6 concludes the paper and points out potential future research directions.

2 Related Work

In this section, we review relevant literature on fairness in federated learning to position our research in relation to existing research. Existing approach for promoting collaborative fairness among federated learning participants is based on incentive schemes. In general, participants shall receive payoffs that is commensurate with their contributions. Equal division is an example of egalitarian profit-sharing [Yang *et al.*2017]. Under this scheme, the available total payoff at a given round is equally divided among all participants. Under the Individual profit-sharing scheme [Yang *et al.*2017], each participant i's own contribution to the collective (assuming the collective only contains i) is used to determine his share of the total payoff.

The Labour Union game [Gollapudi *et al.*2017] profit-sharing scheme determines a participant's share of the total payoff based on his marginal contribution to the utility of the collective formed by his predecessors (i.e. each participant's marginal contribution is computed based on the same sequence as they joined the federation). The Fair-value game scheme [Gollapudi *et al.*2017] is a marginal loss-based scheme. Under this scheme, a participant's share of the total payoff is determined by the sequence following which the participants leave a federation. The Shapley game profit-sharing scheme [Gollapudi *et al.*2017] is also a marginal contribution-based scheme. Unlike the Labour Union game, Shapley game aims to eliminate the effect of the participants joining the collective in different sequences in order to more fairly estimate their marginal contributions to the collective. Thus, it averages the marginal contribution for each participant under all different permutations of him joining the collective relative to other participants. This approach is computationally expensive.

For gradient-based federated learning approaches, the gradient information can be regarded as a useful source of data. However, in these cases, output agreement-based rewards are hard to apply as mutual information requires a multi-task setting which is usually not present in such cases. Thus, among these three categories of schemes, model improvement is the most relevant way of designing rewards for federated learning. There are two emerging federated learning incentive schemes focused on model improvement.

In addition to the contributions made by participants, [Yu *et al.*2020] proposed a joint objective optimization-based approach to take costs and waiting time into account in order to achieve additional notions of fairness when distributing payoffs to FL participants. Different from the aforementioned approaches, the proposed CFFL framework does not utilize monetary payoffs to achieve fair treatment of FL participants. Instead, it allocates each of them a different version of the FL model with performance commensurate with his contributions. This represents a alternative paradigm to existing federated learning in which all participants receive the same final FL model.

3 The CFFL Framework

3.1 Collaborative Fairness Definition

The contribution difference lies in the fact that different parties may have different capacities to generate the training data, and there may exist unpredictable random errors during data collection and storage. On the other hand, different parties have different sharing levels, some parties are more willing to share their information, while some are more private. Since our focus here is to distribute different variants of the final FL model to participants based on their contributions, the notion of fairness most relevant for our purpose is *Fairness through Awareness*. Unlike the traditional distributed/federated learning, where all parties can have access to the same global model, our design does not force a single global model onto local models. Instead, each local model is updated separately by improving their local accuracy. Therefore, parties finally converges to different local models, thus ensuring fairness. The fundamental principle behind fairness is that a party with high-contribution should be rewarded more than a party with low-contribution party. Under this context, we define collaborative fairness as:

Definition 1. *Collaborative fairness. In collaborative learning systems, a high-contribution party is deserved to be rewarded with a better performing local model than a low-contribution party. Specially, fairness can be quantified by the correlation coefficient between the contributions of parties and their respective final model accuracies.*

3.2 Fairness via Reputation

During collaborative learning process, the server audits the claims of each party based on a characterization of the influence of its local model updates on the validation accuracy in each communication round. The server keeps a reputation list, and updates the reputations of all parties as per the contributions of their released gradients. In this way, the reputation of each party keeps changing, reflecting real-time contribution and thus delivering better fairness. The high-contribution party will be highly rated by the server, while the low-contribution party will be detected or even isolated from the collaborative learning system, avoiding the low-contribution party from dominating the whole system.

In our CFFL framework, the server uses leave-one-out strategy to quantify the reputation of party j based on the usefulness of party j's gradients in each communication round. Specifically, the server evaluates the change of validation accuracy by removing party j's gradients from the updated model parameter $\boldsymbol{w}'_t$ that combines all parties' gradients, *i.e.*, using the combined gradients with and without party j's gradients to evaluate the validation accuracy, which yield acc and acc_j respectively, the difference between acc and acc_j reflects how party j affects validation accuracy. The server computes the reputation c_j of party j at the current communication round by passing an "accuracy factor" $x = \frac{acc}{acc+acc_j}$ through a sigmoid function f in Eq. (1).

$$c_j = f(x) = \frac{1}{1 + exp(-15 * (x - 0.5))} \tag{1}$$

Here x stands for the accuracy ratio between the validation accuracy using the combined gradients of all parties and the validation accuracy using the combined gradients excluding party j's gradients, hence it can be further expressed as:

$$x = \frac{acc}{acc + acc_j} = \frac{acc}{2 * acc + \Delta} \tag{2}$$

where $acc_j = acc + \Delta$, Δ indicates the impact of removing party j, the more positive the value of Δ, the better the validation accuracy after removing party j, hence the lower the contribution of party j. To be more specific, if party j has no impact, $\Delta = 0, x = 0.5, c_j = 0.5$; if party j contributes negatively, $acc_j > acc$, then $\Delta > 0, x < 0.5, c_j < 0.5$; if party j contributes positively, $acc_j < acc$, then $\Delta < 0, x > 0.5, c_j > 0.5$. The server computes reputation of each party based on the its contribution in each round, then integrates its historical reputation to update its reputation in the reputation list. In the follow-up rounds, the number of the updated global parameter to be downloaded will be dependent on the reputation of each party. The higher the reputation of party j in the reputation list, the more likely party j will be allocated with a more complete model from the server.

Algorithm 1 Fairness-aware federated learning

Input: $C, \boldsymbol{w_g}, \Delta \boldsymbol{w}_j, \lambda_j, \boldsymbol{w}_i, V$.
In each communication round, each party sends gradients to the server, and server updates its reputation and determines how much this party can download. Each party is initialized with the same parameter to start with.
Role: Party j
if $j \in C \setminus i$ **then**
Downloads the allocated global parameter $\boldsymbol{w_g^j}$ from server and replaces the corresponding elements in local model $\boldsymbol{w}_j$ with $\boldsymbol{w_g^j}$;
Runs SGD on local data and updates the local parameters as $\boldsymbol{w}'_j$;
Computes gradient vector $\Delta \boldsymbol{w}_j = \boldsymbol{w}'_j - \boldsymbol{w}_j$;
$\lambda_j * \Delta \boldsymbol{w}_j$ are grouped into set S as $\Delta(\boldsymbol{w}_j)^S$, which are selected according to the "largest values" criterion: sort gradients in $\Delta \boldsymbol{w}_j$, and upload $\lambda_j * \Delta \boldsymbol{w}_j$ of them, starting from the largest.
end if
Role: Server
Parameter update: $\boldsymbol{w}'_g = \boldsymbol{w}_g + \sum_{j \in C} \Delta(\boldsymbol{w}_j)^S$.
for $j \in C$ **do**
$\boldsymbol{w}_g^{j\prime} = \boldsymbol{w}'_g - \Delta(\boldsymbol{w}_j)^S$, where $\boldsymbol{w}_g$ is the server's global parameters of previous epoch.
$acc \leftarrow (\boldsymbol{w}'_g, V_i), acc_j \leftarrow (\boldsymbol{w}_g^{j\prime}, V_i)$
$x = \frac{acc}{acc + acc_j}$
$c_j{}' = \frac{c_j + f(x)}{2}$, where f refers to the sigmoid reputation mapping function in Eq. (1).
end for
reputation normalisation: $c_j{}' = \frac{c_j{}'}{\sum_{j \in C} c_j{}'}$
if $c_j{}' < c_{th}$ **then**
server flags party j as "low-contribution" and removes it, then runs reputation normalisation again.
end if

3.3 Quantification of Fairness

In FL system, collaborative fairness should be quantified from the point of view of the whole system. In this work, we quantify collaborative fairness through the correlation coefficient between party contributions (*i.e.*, standalone model accuracy which characterizes the learning capability of each party on its own local dataset) and party rewards (*i.e.*, final model accuracies of different parties).

Specifically, we take party contributions as the X-axis, which represents the contributions of different parties from the system view. In particular, we characterize different parties' contributions by their standalone model accuracies, as the party who has local data with better generalization empirically contributes more. In summary, the X-axis can be expressed by Equation 3, where $sacc_j$ denotes the standalone model accuracy of party j:

$$\boldsymbol{x} = \{sacc_1, \cdots, sacc_n\} \tag{3}$$

Similarly, we take party rewards (*i.e.*, final model accuracies of different parties) as the Y-axis, as expressed by Equation 4, acc_j denotes the final model accuracy of party j:

$$\boldsymbol{y} = \{acc_1, \cdots, acc_n\} \tag{4}$$

As the Y-axis measures local model performance of different parties after collaboration, it is expected to be positively correlated with the X-axis to deliver good fairness. Hence, we formally quantify collaborative fairness in Equation 5:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y} \tag{5}$$

where $\bar{x}$ and $\bar{y}$ are the sample means of $\boldsymbol{x}$ and $\boldsymbol{y}$, s_x and s_y are the corrected standard deviations. The range of fairness is within [-1,1], with higher values implying good fairness. Conversely, negative coefficient implies poor fairness.

4 Experimental Evaluation

4.1 Datasets

We implement experiments on two benchmark image datasets. The first is the MNIST dataset[1] for handwritten digit recognition consisting of 60,000 training examples and 10,000 test examples. Each example is a 32x32 gray-level image [Shokri and Shmatikov2015], with digits locating at the center of the image. The second is the Adult Census dataset, which includes 48,843 records with 14 sensitive attributes, including age, race, education level, marital status, and occupation, etc[2]. This dataset is commonly used to predict whether an individual makes over 50K dollars in a year (binary). There are total total 48,842 records, with 24% (11687) records over 50K, while the remaining 76% (37155) under 50K. We manually balance the dataset to have 11687 records over 50K and 11687 records under 50K, resulting in total 23374 records. Among which, we choose 80% records as training set, while the remaining 20% as test set. For all datasets, we randomly choose 10% training examples as validation set.

[1] http://yann.lecun.com/exdb/mnist/
[2] http://archive.ics.uci.edu/ml/datasets/Adult

4.2 Baselines

We demonstrate the effectiveness of our proposed CFFL framework by comparison with the following two representative frameworks. In all frameworks, stochastic gradient descent (SGD) is applied to each party.

1. *Standalone* framework assumes parties train standalone models on local dataset without any collaboration. This framework delivers maximum privacy, but minimum utility, because each party is susceptible to falling into local optima when training alone. In particular, we remark that there is no concrete definition of collaborative fairness in the standalone framework, because parties do not collaborate in this framework.
2. *Distributed* framework enables parties to train independently and concurrently, and chooses a fraction of parameters to be uploaded at each iteration. In particular, as demonstrated in [Shokri and Shmatikov2015], Distributed Selective SGD (DSSGD) can achieve even higher accuracy than the centralized SGD because updating only a small fraction of parameters at each round acts as a regularization technique to avoid overfitting by preventing the neural network weights from jointly "remembering" the training data. Hence, we take DSSGD for the analysis of the distributed framework and omit the centralized framework. In particular, we follow [Shokri and Shmatikov2015] to choose the upload rate $\theta_u = 10\%$, then gradients are uploaded according to the "largest values" criterion.

4.3 Experimental Setup

For MNIST dataset, we take *multi-layer perceptron* (MLP) as local model architecture. For local model training, we set the learning rate as 0.001, learning rate decay as 1e-7, and batch size as 1. In addition, to reduce the impact of different initializations and avoid non-convergence, each party is initialized with the same parameter w_0, then local training is run on individual training data to update local model parameter $\boldsymbol{w_i}$. The sharing level of each party is fixed to 0.1. To boost fairness, we let each party individually train 10 epochs before collaborative learning starts. For all the experiments, we empirically set the reputation threshold as $c_{th} = \frac{1}{|C|-1} * \frac{2}{3}$ via grid search, where $|C|$ is the number of alive parties in the system.

In FL, the data across different parties can be inherently heterogeneous, violating iid assumption. In particular, we consider two practical scenarios, data size imbalanced and class number imbalanced. For data size imbalanced scenario, to simulate different parties owing different amount of data, we follow a power law to randomly partition total $\{3000, 6000, 12{,}000\}$ examples among $\{5,10,20\}$ parties respectively. Similarly, for Adult dataset, total $\{5000, 10000, 20000\}$ examples are randomly partitioned among $\{5,10,20\}$ parties. We remark that the purpose of allocating 600 MNIST examples for each party is to fairly compare with Shokri *et al.* [Shokri and Shmatikov2015], in which each party is allocated with 600 MNIST examples (small number of local examples to simulate data scarity which necessitates collaboration). Therefore, for MNIST, we simulate the total examples of 3000 (5 parties) up to 12,000 (20 parties). For larger datasets like 300,000 examples, it would require 500 parties, imposing heavy requirement on deployment, while delivering similar results.

For class number imbalanced scenario,

4.4 Experimental Results

Table 1 lists the calculated fairness of the distributed framework and our CFFL over MNIST and Adult datasets under different settings of $\{5,10,20\}$ parties.All the fairness results are averaged over five trails. As is evidenced by the high positive values of fairness, with all of them above 0.5, CFFL achieves reasonably good fairness, confirming the intuition behind fairness: the party who contributed more is reward with a better model. In contrast, the distributed framework exhibits bad fairness with significantly lower values than that of CFFL in all cases, and even negative values in some cases, manifesting the lack of fairness in the distributed framework. This is because in the distributed framework, all the participating parties can get access to the same global model, no matter how much one party contributes, which is obviously unfair.

Table 1: Fairness of distributed framework and our CFFL over MNIST and Adult datasets under different party settings (P-k indicates there are k parties in the experiments).

	MNIST		Adult	
	Distributed	CFFL	Distributed	CFFL
P5	-0.68	**0.84**		
P10	-0.20	**0.79**		
P20	-0.27	**0.84**		

Table 2: Accuracy [%] over MNIST and Adult of varying party number settings, achieved by *Standalone*, *Distributed* (DSSGD) and our CFFL.

Framework	MNIST			Adult		
	P5	P10	P20	P5	P10	P20
Distributed	92.03	94.33	95.52			
Standalone	88.16	89.41	89.52			
CFFL	92.82	94.06	94.88			

Table 2 provides the accuracy on MNIST and Adult datasets of $\{5,10,20\}$ parties using baseline frameworks in Section **??**, and our proposed CFFL. Here we report the best accuracy because our CFFL enables parties to converge to different local models after collaborative learning, and we expect the most contributive party derives a local model with maximum accuracy approximating the distributed framework. For both MNIST and Adult datasets, we show the worst accuracy for standalone SGD (minimum utility, maximum privacy). In particular, CFFL obtains comparable accuracy to the distributed framework using DSSGD, and always achieves higher accuracy than the standalone SGD. For example, as shown in Table 2, for MNIST dataset of 20 parties, our CFFL achieves -95% test accuracy, which is higher than the standalone SGD 89%, and comparable to the distributed framework using DSSGD.

The above fairness results in Table 1, and accuracy results in Table 2 demonstrate that *our proposed CFFL achieves both reasonable fairness and comparable accuracy*.

Party Convergence. To investigate the impact of our CFFL on individual convergence, Fig. 1 further depicts the accuracy trajectory of each party when running Standalone framework and our CFFL over MNIST across 100 communication rounds. For the sake of brevity, we only report experimental results obtained for the collaboration among 5 parties and 10 parties. It can be observed that our CFFL consistently delivers better accuracy than the standalone model obtained by any individual party, at the cost of slower convergence and more fluctuation. However, most parties can converge within the first 20 rounds, except those with lower standalone accuracy. For example, in Figure 1 (d), party 1 and party 2 encounter higher fluctuations compared with the other parties with higher standalone accuracies. More importantly, these figures confirm that our CFFL enforces all parties to converge to different local models, which are better than their standalone models without any collaboration, thereby offering fairness as claimed.

To speed up convergence and alleviate fluctuations, we further experiment with larger number of local epochs, larger local batch size, and higher learning rate. As corroborated by Fig. **??**, by setting $B = 10, E = 5, lr = 0.15$, each party can converge faster, without affecting both accuracy and fairness. For example, for P10 in Figure **??** (d), it needs 65 communication rounds for all parties to converge using $B = 1, E = 1, lr = 0.001$, while it only needs 50 communication rounds using $B = 10, E = 5, lr = 0.15$ in Figure **??** (d). However, this faster convergence and less fluctuations come at the cost of local computation at each party.

5 Discussions

Fairness in heterogeneous settings. Sharing model updates is typically limited only to homogeneous FL architectures, *i.e.*, the same model architecture is shared with all parties. In heterogeneous settings, parties may train different types of local model, instead of sharing model updates, parties can share model predictions on the unlabeled public dataset. The server then quantifies the reputation of each party based on their local prediction. We remark that sharing model predictions is model agnostic, meaning that this should work for almost any kind of machine architecture. We evaluate FL with heterogeneous model architectures, and show its effectiveness without compromising the final accuracy of party models. We use Purchase data and 5 fully connected models, which we call M1, M2, M3, M4, and M5, with hidden layer sizes $\{\}$, $\{1024\}$, $\{512, 256\}$, $\{1024, 256\}$, and $\{1024, 512, 256\}$ respectively. Note that, M1 has lower capacity, thus should deliver lower accuracy than the other four models.

Reputation threshold. If the server finds out that the reputation of one party is lower than the threshold c_{th}, implying a potentially low-contribution party, it will be isolated from the collaborative learning system. Here, c_{th} is mainly used to detect and isolate the extremely low-contribution party, and it should be agreed by the majority of parties. However, it should not be too small or too large as fairness and accuracy may be affected. For example, too small c_{th} might allow low-contribution party to sneak into the collaborative learning system without being detected and isolated. On the contrary, too large c_{th} might isolate most participants in the system.

our framework is more relevant to practical applications to businesses [Lyu *et al.*2020], such as biomedical or financial institutions where the collaboration fairness is a more concerned problem.

6 Conclusion and Future Work

This paper initiates the research problem of collaborative fairness in federated learning, and proposes a novel collaborative fair federated learning framework called CFFL. A notion of reputation is introduced to quantify party contribution across communication rounds. The experimental results demonstrate that our CFFL achieves comparable accuracy to the distributed framework, and always delivers better results than the standalone framework, confirming the applicability of our proposed framework. A number of avenues for further work are appealing. In particular, we would like to study how to quantify fairness in more complex settings. We also expect to deploy our system into a wide spectrum of real-world applications.

References

[Cummings *et al.*, 2019] Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. On the compatibility of privacy and fairness, 2019.

[Gollapudi *et al.*, 2017] Sreenivas Gollapudi, Kostas Kollias, Debmalya Panigrahi, and Venetia Pliatsika. Profit sharing and efficiency in utility games. In *ESA*, pages 1–16, 2017.

[Jagielski *et al.*, 2018] Matthew Jagielski, Michael Kearns, Jieming Mao, Alina Oprea, Aaron Roth, Saeed Sharifi-Malvajerdi, and Jonathan Ullman. Differentially private fair learning. *arXiv preprint arXiv:1812.02696*, 2018.

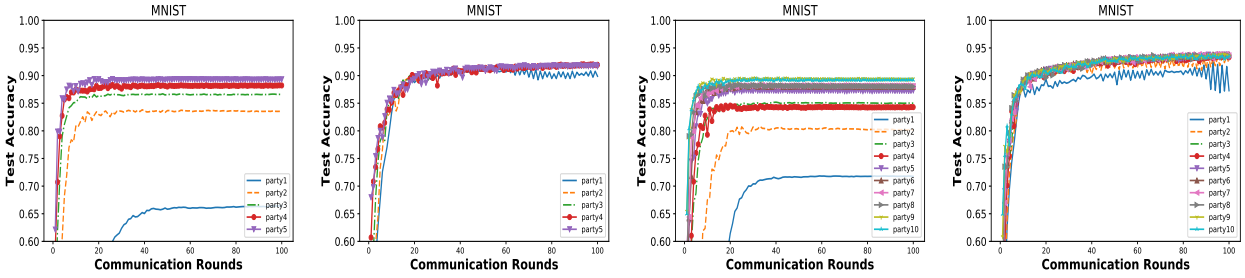
[Lyu *et al.*, 2020] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020.

[McMahan *et al.*, 2016] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Aguera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2016.

[Shokri and Shmatikov, 2015] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1310–1321. ACM, 2015.

[Yang *et al.*, 2017] Shuo Yang, Fan Wu, Shaojie Tang, Xiaofeng Gao, Bo Yang, and Guihai Chen. On designing data quality-aware truth estimation and surplus sharing method for mobile crowdsensing. *IEEE Journal on Selected Areas in Communications*, 35(4):832–847, 2017.

[Yang *et al.*, 2019] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. *Federated Learning*. Morgan & Claypool Publishers, 2019.



(a) Standalone (P5) (b) CFFL (P5) (c) Standalone (P10) (d) CFFL (P10)

Figure 1: Individual convergence for MNIST MLP using Standalone framework and our CFFL (B=1, E=1, lr=0.001).

[Yu *et al.*, 2020] Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. A fairness-aware incentive scheme for federated learning. In *Proceedings of the 3rd AAAI/ACM Conference on AI, Ethics, and Society (AIES-20)*, pages 393–399, 2020.