Federated Learning on Non-IID Data Silos: An Experimental Study

Qinbin Li*
National University of Singapore
Singapore
qinbin@comp.nus.edu.sg

Yiqun Diao* Quan Chen
Shanghai Jiao Tong University
Shanghai, China
{diaoyiqun, chen-quan}@sjtu.edu.cn

Bingsheng He
National University of Singapore
Singapore
hebs@comp.nus.edu.sg

***Abstract*—Due to the increasing privacy concerns and data regulations, training data have been increasingly fragmented, forming distributed databases of multiple "data silos" (e.g., within different organizations and countries). To develop effective machine learning services, there is a must to exploit data from such distributed databases without exchanging the raw data. Recently, federated learning (FL) has been a solution with growing interests, which enables multiple parties to collaboratively train a machine learning model without exchanging their local data. A key and common challenge on distributed databases is the heterogeneity of the data distribution among the parties. The data of different parties are usually non-independently and identically distributed (i.e., non-IID). There have been many FL algorithms to address the learning effectiveness under non-IID data settings. However, there lacks an experimental study on systematically understanding their advantages and disadvantages, as previous studies have very rigid data partitioning strategies among parties, which are hardly representative and thorough. In this paper, to help researchers better understand and study the non-IID data setting in federated learning, we propose comprehensive data partitioning strategies to cover the typical non-IID data cases. Moreover, we conduct extensive experiments to evaluate state-of-the-art FL algorithms. We find that non-IID does bring significant challenges in learning accuracy of FL algorithms, and none of the existing state-of-the-art FL algorithms outperforms others in all cases. Our experiments provide insights for future studies of addressing the challenges in "data silos".**

I. Introduction

In recent years, we have witnessed some promising advancement with leveraging machine learning services, such as learned index structures [12], [54] and learned cost estimation [24], [55]. As such, machine learning services have become emerging data-intensive workloads, such as Ease.ml [46], Machine Learning Bazaar [69] and Rafiki [74]. Despite the success of machine learning services, their effectiveness highly relies on large-volume high-quality training data. However, due to the increasing privacy concerns and data regulations such as GDPR [70], training data have been increasingly fragmented, forming distributed databases of multiple "data silos" (e.g., within different organizations and countries). Due to the deployed data regulations, raw data are usually not allowed to transfer across organizations/countries. For example, a multinational corporation (MNC) provides services to users in multiple nations, whose personal data usually cannot be centralized to a single country due to the data regulations in many countries.

To develop effective machine learning services, it is necessary to exploit data from such distributed databases without exchanging the raw data. While there are many studies working on privacy-preserving data management and data mining [3], [31], [60], [64], [66] in a centralized setting, they cannot handle the cases of distributed databases. Thus, how to conduct data mining/machine learning from distributed databases without exchanging local data has become an emerging topic.

To address the above challenge, we borrow the federated learning (FL) [33], [43], [44], [77] approach from the machine learning community. Originally proposed by Google, FL is a promising solution to enable many parties jointly train a machine learning model while keeping their local data decentralized. Here we focus on horizontal federated learning, where the parties share the same feature space but different sample space. Instead of exchanging data and conducting centralized training, each party sends its model to the server, which updates and sends back the global model to the parties in each round. Since their raw data are not exposed, FL is an effective way to address privacy concerns. It has attracted many research interests [9], [25], [35], [42], [45], [52], [75] and been widely used in practice [5], [23], [34]. Thus, we consider FL to develop machine learning services for distributed databases.

One key and common data challenge in such distributed databases is that data distributions in different parties are usually non-independently and identically distributed (non-IID). For example, different areas can have very different disease distributions. Due to the ozone hole, the countries in the Southern Hemisphere may have more skin cancer patients than the Northern Hemisphere. Then, the label distributions differ across parties. Another example is that people have different writing styles even for the same world. In such a case, the feature distributions differ across parties. According to previous studies [28], [35], [47], the non-IID data settings can degrade the effectiveness of machine learning services.

There have been some studies trying to develop effective FL algorithms under non-IID data including FedProx [45], SCAFFOLD [35], and FedNova [72]. However, there lacks an experimental study on systematically understanding their advantages and disadvantages, as the previous studies have very rigid data partitioning strategies among parties, which

*Equal contribution.

are hardly representative and thorough. In the experiments of these studies, they only try one or two partitioning strategies to simulate the non-IID data setting, which does not sufficiently cover different non-IID cases. For example, in FedAvg [56], each party only has samples of two classes. In FedNova [72], the number of samples of each class in each party follows Dirichlet distribution. The above partitioning strategies only cover the label skewed case. Thus, it is a necessity to evaluate those algorithms with a systematic exploration of different non-IID scenarios.

In this paper, we break the barrier of experiments on non-IID data distribution challenges in FL by proposing NIID-Bench. Specifically, we introduce six non-IID data partitioning strategies which thoroughly consider different cases including label distribution skew, feature distribution skew, and quantity skew. Moreover, we conduct extensive experiments on nine datasets to evaluate the accuracy of four state-of-the-art FL algorithms including FedAvg [56], FedProx [45], SCAFFOLD [35], and FedNova [72]. The experimental results provide insights for the future development of FL algorithms. Last, our code is publicly available [1]. Researchers can easily use our code to try different partitioning strategies for the evaluation of existing algorithms or a new algorithm. We also maintain a leaderboard along with our code to rank state-of-the-art federated learning algorithms on different non-IID settings, which can benefit the federated learning community a lot.

Through extensive studies, we have the following key findings. First, we find that non-IID does bring significant challenges in learning accuracy of FL algorithms, and none of the existing state-of-the-art FL algorithms outperforms others in all cases. Second, the effectiveness of FL is highly related to the kind of data skews, e.g., the label distribution skew setting is more challenging than the quantity skew setting. This indicates the importance of having a more comprehensive benchmark on non-IID distributions. Last, in non-IID data setting, instability of the learning process widely exists due to techniques such as batch normalization and partial sampling. This can severely hurt the effectiveness of machine learning services on distributed data silos.

Our main contributions are as follows:

- We identity non-IID data distributions as a key and common challenge in designing effective federated learning algorithms for distributed data silos and develop a benchmark for researchers' study of federated learning on non-IID data.
- We summarize six different partitioning strategies to generate comprehensive non-IID data distribution cases. Among six partitioning strategies, four simple and effective partitioning strategies are designed by our study, while the other two strategies are adopted from existing studies due to their popularity. We also demonstrate the significance of those strategies. None of the previous studies [28], [35], [45], [72] are as comprehensive as ours. For example, paper [28] only covers a single partitioning strategy to generate the label distribution skew setting.
- Using the proposed partitioning strategies, we conduct an extensive experimental study on four state-of-the-art algorithms, including FedAvg [56], FedProx [45], SCAFFOLD [35], and FedNova [72]. Moreover, we provide insightful findings and future directions for data management and learning for distributed data silos, which we believe are more and more common in the future.

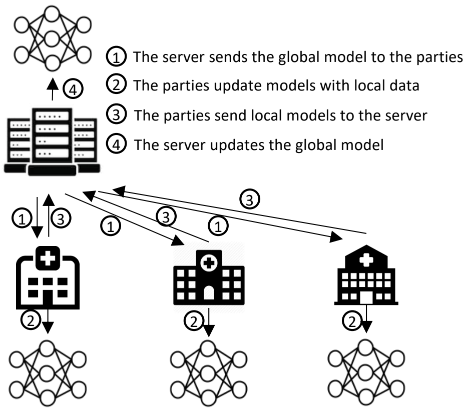

① The server sends the global model to the parties
② The parties update models with local data
③ The parties send local models to the server
④ The server updates the global model
④
①
③
①
③
①
③
②
②
②

Fig. 1. The FedAvg framework.

The remainder of this paper is structured as follows. We introduce the preliminaries in Section II. We review FL algorithms handling non-IID data in Section III, and present our non-IID data partition strategies in Section IV. Section V present the experimental results, followed by the future research directions in Section VI. We discuss the related work in Section VII, and conclude in Section VIII.

II. Preliminaries

A. Notations

Let $\mathcal{D} = \{(\mathbf{x}, y)\}$ denote the global dataset. Suppose there are N parties, denoted as $P_1, ..., P_N$. The local dataset of P_i is denoted as $\mathcal{D}^i = \{(\mathbf{x}_i, y_i)\}$. We use w^t and w_i^t to denote the global model and the local model of party P_i in round t, respectively. Thus, w^t is the output model of the federated learning process.

B. FedAvg

FedAvg [56] has been a de facto approach for FL. The framework of FedAvg is shown in Figure 1. In each round, first, the server sends the global model to the randomly selected parties. Second, each party updates the model with its local dataset. Then, the updated models are sent back to the server. Last, the server averages the received local models as the updated global model. Unlike traditional distributed SGD, the parties update their local model with multiple epochs, which can decrease the number of communication rounds and is much more communication-efficient. However, the local updates may lead to a bad accuracy, as shown in previous studies [28], [35], [47].

[1] https://github.com/Xtra-Computing/NIID-Bench

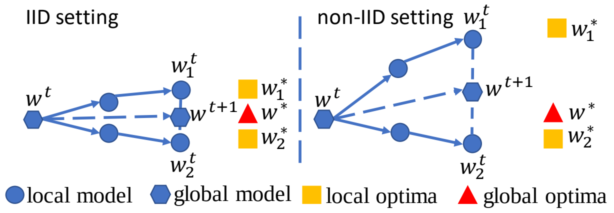


Fig. 2. Example of a drift under the non-IID setting.

C. *Effect of Non-IID Data*

A key challenge in FL is the non-IID data among the parties [33], [43]. Non-IID data can influence the accuracy of FedAvg a lot. Since the distribution of each local dataset is highly different from the global distribution, the local objective of each party is inconsistent with the global optima. Thus, there exists a *drift* in the local updates [35]. In other words, in the local training stage, the local models are updated towards the local optima, which can be far from the global optima. The averaged model may also be far from the global optima especially when the local updates are large (e.g., a large number of local epochs) [35], [45], [71], [72]. Eventually, the converged global model has much worse accuracy than IID setting. Figure 2 demonstrates the issue of FedAvg under the non-IID data setting. Under the IID setting, the global optima w^* is close to the local optima w_1^* and w_2^*. Thus, the averaged model w^{t+1} is also close to the global optima. However, under the non-IID setting, since w^* is far from w_1^*, w^{t+1} can be far from w^*. It is challenging to design an effective FL algorithm under the non-IID setting. We will present the FL algorithms on handling non-IID data in the next section.

III. FL Algorithms on Non-IID Data

There have been some studies [35], [45], [72] trying to address the drift issue in FL. Here we summarize several state-of-the-art and popular approaches as shown in Algorithm 1 (FedAvg [56], FedProx [45], FedNova [72]) and Algorithm 2 (SCAFFOLD [35]). These approaches are all based on FedAvg, and we use colors to mark the parts that specially designed in FedProx (red), SCAFFOLD (blue), and FedNova (orange). Note that the studied approaches have the same objective, i.e., learning an effective global model under the non-IID data setting. There are also other FL studies related to non-IID data setting, such as personalizing the local models for each party [13], [15], [22] and designing robust algorithms against different combinations of local distributions [10], [57], [63], which are out of the scope of this paper.

A. *FedProx*

FedProx [45] improves the local objective based on FedAvg. It directly limits the size of local updates. Specifically, as shown in Line 14 of Algorithm 1, it introduces an additional L_2 regularization term in the local objective function to limit the distance between the local model and the global model. This is a straightforward way to limit the local updates so that the averaged model is not so far from the global optima.

Algorithm 1: A summary of FL algorithms including FedAvg/FedProx/FedNova. We use red and orange colors to mark the part specially included in FedProx and FedNova, respectively.

```
Input: local datasets $\mathcal{D}^i$, number of parties $N$, number
       of communication rounds $T$, number of local
       epochs $E$, learning rate $\eta$
Output: The final model $w^T$
1  Server executes:
2  initialize $x^0$
3  for $t = 0, 1, ..., T-1$ do
4      Sample a set of parties $S_t$
5      $n \leftarrow \sum_{i \in S_t} |\mathcal{D}^i|$
6      for $i \in S_t$ in parallel do
7          send the global model $w^t$ to party $P_i$
8          $\Delta w_i^t, \tau_i \leftarrow$ LocalTraining($i$, $w^t$)
9      For FedAvg/FedProx:
         $w^{t+1} \leftarrow w^t - \eta \sum_{i \in S_t} \frac{|\mathcal{D}^i|}{n} \Delta w_k^t$
10     For FedNova:
         $w^{t+1} \leftarrow w^t - \eta \frac{\sum_{i \in S_t} |\mathcal{D}^i| \tau_i}{n} \sum_{i \in S_t} \frac{|\mathcal{D}^i| \Delta w_i^t}{n \tau_i}$
11 return $w^T$
12 Party executes:
13 For FedAvg/FedNova: $L(w; \mathbf{b}) = \sum_{(x,y) \in \mathbf{b}} \ell(w; x; y)$      损失函数
                                                                        平均
14 For FedProx:
     $L(w; \mathbf{b}) = \sum_{(x,y) \in \mathbf{b}} \ell(w; x; y) + \underbrace{\frac{\mu}{2} \|w - w^t\|^2}$   正则化系数.
15 LocalTraining($i$, $w^t$):
16 $w_i^t \leftarrow w^t$
17 $\tau_i \leftarrow 0$  ?
18 for epoch $k = 1, 2, ..., E$ do
19     for each batch $\mathbf{b} = \{\mathbf{x}, y\}$ of $\mathcal{D}^i$ do     使用小批量数据b
20         $w_i^t \leftarrow w_i^t - \eta \nabla L(w_i^t; \mathbf{b})$          进行求导训练.
21         $\tau_i \leftarrow \tau_i + 1$
22 $\Delta w_i^t \leftarrow w^t - w_i^t$
23 return $\Delta w_i^t$, $\tau_i$ to the server
```

A hyper-parameter μ is introduced to control the weight of the L_2 regularization. Overall, the modification to FedAvg is lightweight and easy to implement. FedProx introduces additional computation overhead and does not introduce additional communication overhead. However, one drawback is that users may need to carefully tune μ to achieve good accuracy. If μ is too small, then the regularization term has almost no effect. If μ is too big, then the local updates are very small and the convergence speed is slow.

B. *FedNova*

Another recent study, FedNova [72], improves FedAvg in the aggregation stage. It considers that different parties may conduct different numbers of local steps (i.e., the number of mini-batches in the local training) each round. This can happen when parties have different computation power given the same time constraint or parties have different local dataset size given the same number of local epochs and batch size. Intuitively, the parties with a larger number of local steps will have a larger local update, which will have a more significant influence on the global updates if simply averaged. Thus, to ensure that the global updates are not biased, FedNova normalizes and scales the local updates of each party according to their number of local steps before updating the global model (see Line 10 of Algorithm 1). FedNova also only introduces lightweight modifications to FedAvg, and negligible computation overhead when updating the global model.

C. SCAFFOLD

SCAFFOLD [35] models non-IID as introducing variance among the parties and applies the variance reduction technique [32], [65]. It introduces control variates for the server (i.e., c) and parties (i.e., c_i), which are used to estimate the update direction of the server model and the update direction of each client. Then, the drift of local training is approximated by the difference between these two update directions. Thus, SCAFFOLD corrects the local updates by adding the drift in the local training (Line 20 of Algorithm 2). SCAFFOLD proposes two approaches to update the local control variates (Line 23 of Algorithm 2), by computing the gradient of the local data at the global model or by reusing the previously computed gradients. The second approach has a lower computation cost while the first one may be more stable. Compared with FedAvg, intuitively, SCAFFOLD doubles the communication size per round due to the additional control variates.

D. Other Studies

When preparing this paper, there are other contemporary works [2], [40], [48], [73] on federated learning under non-IID setting. [2] proposes FedDyn, which adds a regularization term in the local training based on the global model and the model from the previous round. [48] proposes FedBN for feature shift non-IID setting, where the client batch-norm layers are updated locally without communicating to the server. [73] applies a monitor to detect class imbalance in the training process, and proposes a new loss function to address it. [40] proposes model-contrastive learning. Their approach corrects the local training by comparing the representations learned by the current local model, the local model from the previous round, and the global model. We leave the comparison between these studies as future studies.

E. Motivation of this study

Non-IID is a key and common data challenge for developing effective federated learning algorithms. Although previous studies [35], [45], [72] have demonstrated preliminary and promising results over FedAvg on non-IID data, as we will summarize in Table I in later section, all above studies have evaluated only one or two non-IID distributions, and tried rigid data partitioning strategies in the experiments. There is still no standard benchmark or a systematic study to evaluate the effectiveness of these FL algorithms. This motivates us to develop a benchmark with more comprehensive data distributions as well as data partitioning strategies, and then we can evaluate the pros and cons of existing algorithms and outline the challenges and opportunities for future federated learning on non-IID data.

Algorithm 2: The SCAFFOLD algorithm. We use blue color to mark the part specially included in SCAFFOLD compared with FedAvg.

```
Input: same as Algorithm 1
Output: The final model $w^T$
1  Server executes:
2  initialize $x^0$
3  $c^t \leftarrow \mathbf{0}$
4  for $t = 0, 1, ..., T-1$ do
5      Randomly sample a set of parties $S_t$
6      $n \leftarrow \sum_{i \in S_t} |\mathcal{D}^i|$
7      for $i \in S_t$ in parallel do
8          send the global model $w^t$ to party $P_i$
           $\Delta w_i^t, \Delta c \leftarrow$ LocalTraining($i$, $w^t$, $c^t$)
9      $w^{t+1} \leftarrow w^t - \eta \sum_{i \in S_t} \frac{|\mathcal{D}^i|}{n} \Delta w_k^t$
10     $c^{t+1} \leftarrow c^t + \frac{1}{N} \Delta c$
11 return $w^T$
12 Party executes:
13 $L(w; \mathbf{b}) = \sum_{(x,y) \in \mathbf{b}} \ell(w; x; y)$
14 $c_i \leftarrow \mathbf{0}$
15 LocalTraining($i$, $w^t$, $c^t$):
16 $w_i^t \leftarrow w^t$
17 $\tau_i \leftarrow 0$
18 for epoch $k = 1, 2, ..., E$ do
19     for each batch $\mathbf{b} = \{\mathbf{x}, y\}$ of $\mathcal{D}^i$ do
20         $w_i^t \leftarrow w_i^t - \eta(\nabla L(w_i^t; \mathbf{b}) - c_i^t + c)$
21         $\tau_i \leftarrow \tau_i + 1$
22 $\Delta w_i^t \leftarrow w^t - w_i^t$
23 $c_i^* \leftarrow (i) \nabla L(w_i^t), or (ii) c_i - c + \frac{1}{\tau_i \eta}(w^t - w_i^t)$
24 $\Delta c \leftarrow c_i^* - c_i$
25 $c_i \leftarrow c_i^*$
26 return $\Delta w_i^t$, $\Delta c$ to the server
```

IV. Simulating Non-IID Data Setting

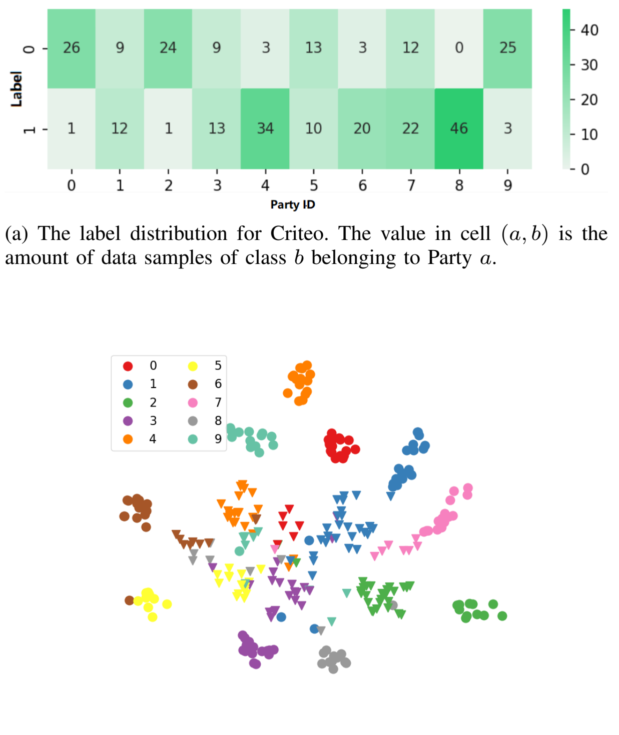
As existing studies only adopt limited partitioning strategies, they cannot represent a comprehensive view of non-IID cases. To bridge this gap, we develop a benchmark named NIID-Bench.

A. Research Problems

We need to address two key research problems. The first one is on data sets: whether to use real-world non-IID datasets or synthetic datasets. The second one is on how to design comprehensive non-IID scenarios.

For the first problem, we choose to synthesize the distributed non-IID datasets by partitioning a real-world dataset into multiple smaller subsets. Many existing studies [35], [56], [72] use the partitioning approach to simulate the non-IID federated setting. Compared with using real federated datasets [6], [29], adopting partitioning strategies has the following advantages. First, while it is challenging to evaluate the imbalance properties (e.g., imbalanced level and imbalanced case) in real federated datasets, partitioning strategies can easily quantify and control the imbalance properties of the local data. Thus, researchers can easily investigate the behavior of algorithms by trying different imbalanced settings, which is essential to the development of FL algorithms. Second, when using synthetic datasets, one can easily set different factors (e.g., number of parties, size of data) that are important in the FL experiments. However, a real federated dataset usually corresponds to a fixed federated setting. Last, due to data regulation and privacy concerns, meaningful real federated datasets are difficult to obtain [29]. Even if we can obtain such real datasets, they do not have the previous two advantages of synthetic data sets. It is more flexible to develop partitioning strategies on existing widely used public datasets, which already have lots of centralized training knowledge as reference, as well as to simulate different non-IID scenarios. There are also limitations of using generated datasets compared with using real federated datasets. The generated datasets may not fully capture the real data distributions, which can be complicated and challenging to quantity. Note that the usage of generated federated datasets and real federated datasets are orthogonal. It is an interesting future study to find and study meaningful real-world data sets and application scenarios.

For the second problem, an existing study [33] gives a very good and comprehensive summary on non-IID data cases from a distribution perspective. Specifically, considering the local data distribution $P(x_i, y_i) = P(x_i|y_i)P(y_i)$ or $P(x_i, y_i) = P(y_i|x_i)P(x_i)$, the previous study [33] summaries five different non-IID cases: (1) label distribution skew (i.e., $P(y_i)$ is different among parties); (2) feature distribution skew (i.e., $P(x_i)$ is different among parties); (3) same label but different features (i.e., $P(x_i|y_i)$ is different among parties); (4) same features but different labels (i.e., $P(y_i|x_i)$ is different among parties); (5) quantity skew (i.e., $P(x_i, y_i)$ is same but the amount of data is different among parties). Here the third case is mainly related to vertical FL (the parties share the same sample IDs but different features). As mentioned in the third paragraph of Section I, we focus on horizontal FL in this paper, where each party shares the same feature space but owns different samples. The fourth case is not applicable in most FL studies, which assume there is a common knowledge $P(y|x)$ among the parties to learn. Otherwise, techniques such as domain adaption [61] or personalized federated learning (i.e., each party learns a personalized local model) [13], [15] can be applied in federated learning, which is out of the scope of our paper. Thus, we consider label distribution skew, feature distribution skew, and quantity skew as possible non-IID data distribution cases in this paper. While the five non-IID data cases cover all possible single type of skew, there may be mixed types of skew, which we will discuss in Section V-G.



(a) The label distribution for Criteo. The value in cell (a, b) is the amount of data samples of class b belonging to Party a.

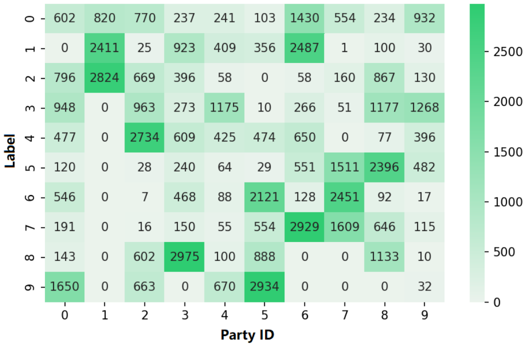
(b) The feature distribution for Digits. The triangles are the visualized features of SVHN and the circles are the visualized features of MNIST.

Fig. 3. The non-IID properties of *criteo* and *digits*.

We use two real-world datasets, *Criteo* [11] and *Digits* [61], to demonstrate the non-IID properties. Criteo contains feature values and click feedback for millions of display ads, which can be used for clickthrough rate prediction. Digits contains multiple subsets for digit classification. In Criteo, taking each user as a party, we select ten parties and draw the label distribution as shown in Figure 3a. We can observe that there exists both label distribution skew (e.g., Party 0 and Party 4) and quantity skew (e.g., Party 0 and Party 8) among the parties. In Digits, taking each subset (e.g., *MNIST* and *SVHN*) as a party, we train a model using these subsets and draw the feature distribution using t-SNE [53] as shown in Figure 3b. For each class, althougth MNIST and SVHN have the same label, the feature distributions of MNIST and SVHN are significantly different from each other. Feature skew exists in the Digits dataset. These two examples show that the considered non-IID data cases are reasonable and practical.

B. Label Distribution Skew

In label distribution skew, the label distributions $P(y_i)$ vary across parties. Such a case is common in practice. For example, some hospitals are more specialized in several specific kinds of diseases and have more patient records on them. To simulate label distribution skew, we introduce two different label imbalance settings: quantity-based label imbalance and distribution-based label imbalance.



	0	1	2	3	4	5	6	7	8	9
0	602	820	770	237	241	103	1430	554	234	932
1	0	2411	25	923	409	356	2487	1	100	30
2	796	2824	669	396	58	0	58	160	867	130
3	948	0	963	273	1175	10	266	51	1177	1268
4	477	0	2734	609	425	474	650	0	77	396
5	120	0	28	240	64	29	551	1511	2396	482
6	546	0	7	468	88	2121	128	2451	92	17
7	191	0	16	150	55	554	2929	1609	646	115
8	143	0	602	2975	100	888	0	0	1133	10
9	1650	0	663	0	670	2934	0	0	0	32

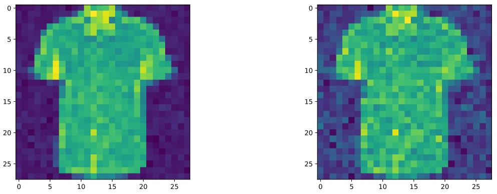
Fig. 4. An example of distribution-based label imbalance partition on MNIST [38] dataset with $\beta = 0.5$. The value in each rectangle is the number of data samples of a class belonging to a certain party.

a) Quantity-based label imbalance: Here each party owns data samples of a fixed number of labels. This is first introduced in the experiments of FedAvg [56], where the data samples with the same label are divided into subsets and each party is only assigned 2 subsets with different labels. Following FedAvg, such a setting is also used in many other studies [19], [45]. [16] considers a highly extreme case, where each party only has data samples with a single label. We introduce a general partitioning strategy to set the number of labels that each party has. Suppose each party only has data samples of k different labels. We first randomly assign k different label IDs to each party. Then, for the samples of each label, we randomly and equally divide them into the parties which own the label. In this way, the number of labels in each party is fixed, and there is no overlap between the samples of different parties. For ease of presentation, we use $\#C = k$ to denote such a partitioning strategy.

b) Distribution-based label imbalance: Another way to simulate label imbalance is that each party is allocated a proportion of the samples of each label according to Dirichlet distribution. Dirichlet distribution is commonly used as prior distribution in Bayesian statistics [30] and is an appropriate choice to simulate real-world data distribution. Specifically, we sample $p_k \sim Dir_N(\beta)$ and allocate a $p_{k,j}$ proportion of the instances of class k to party j. Here $Dir(\cdot)$ denotes the Dirichlet distribution and β is a concentration parameter ($\beta > 0$). This partitioning strategy was first used in [78] and has been used in many recent studies [41], [50], [71], [72]. An advantage of this approach is that we can flexibly change the imbalance level by varying the concentration parameter β. If β is set to a smaller value, then the partition is more unbalanced. An example of such a partitioning strategy is shown in Figure 4. For ease of presentation, we use $p_k \sim Dir(\beta)$ to denote such a partitioning strategy.

C. Feature Distribution Skew

In feature distribution skew, the feature distributions $P(x_i)$ vary across parties although the knowledge $P(y_i|\mathbf{x}_i)$ is same. For example, cats may vary in coat colors and patterns in different areas. Here we introduce three different settings to simulate feature distribution skew: noise-based feature imbalance, synthetic feature imbalance, and real-world feature imbalance.



(a) add noises from $Gau(0.001)$ (b) add noises from $Gau(0.01)$

Fig. 5. An example of adding noises on FMNIST [76] dataset. On party P_1, noises sampled from $Gau(0.001)$ are added into its images. On party P_2, noises sampled from $Gau(0.01)$ are added into its images.

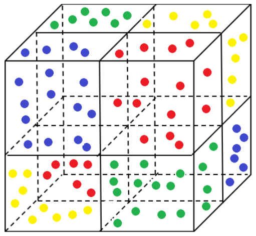


Fig. 6. The visualization of our FCUBE dataset. The data points within the upper four cubes have label 0 and within the lower four cubes have label 1. There are a total of eight cubes with four colors. The data points with the same color are assigned to a party.

a) Noise-based feature imbalance: We first divide the whole dataset into multiple parties randomly and equally. For each party, we add different levels of Gaussian noise to its local dataset to achieve different feature distributions. We choose Gaussian noise due to its popularity especially in images [79]. Specifically, given user-defined noise level σ, we add noises $\hat{\mathbf{x}} \sim Gau(\sigma \cdot i/N)$ for Party P_i, where $Gau(\sigma \cdot i/N)$ is a Gaussian distribution with mean 0 and variance $\sigma \cdot i/N$. Users can change σ to increase the feature dissimilarity among the parties. Figure 5 is an example of noise-based feature imbalance on FMNIST dataset [76]. For ease of presentation, we use $\hat{\mathbf{x}} \sim Gau(\sigma)$ to present such a partitioning strategy.

b) Synthetic feature imbalance: We generate a synthetic feature imbalance federated dataset named FCUBE. Suppose the distribution of data points is a cube in three dimensions (i.e, (x_1, x_2, x_3)) which have two different labels classified by plane $x_1 = 0$. As shown in Figure 6, we divide the cube into 8 parts by planes $x_1 = 0$, $x_2 = 0$, and $x_3 = 0$. Then, we allocate two parts which are symmetric of (0,0,0) to a subset for each party. In this way, feature distribution varies among parties while labels are still balanced.

c) Real-world feature imbalance: The EMNIST dataset [8] collects handwritten characters/digits from different writers. Then, like [6], it is natural to partition the dataset into different parties according to the writers. Since the character features usually differ among writers (e.g, stroke width, slant), there is a natural feature distribution skew among different parties. Specifically, for the digit images of EMNIST, we

TABLE I
THE EXPERIMENTAL SETTINGS IN EXISTING STUDIES AND OUR BENCHMARK. NOTE THAT THE QUANTITY-BASED, NOISED-BASED, AND QUANTITY SKEW PARTITIONING STRATEGIES IN THE EXISTING STUDIES ARE DIFFERENT FORM THE STRATEGIES PROPOSED IN OUR STUDY.

Partitioning strategies		FedAvg	FedProx	SCAFFOLD	FedNova	NIID-Bench
Label distribution skew	quantity-based	✓	✓	✗	✗	✓
	distribution-based	✗	✗	✓	✓	✓
Feature distribution skew	noise-based	✗	✗	✗	✗	✓
	synthetic	✗	✓	✗	✗	✓
	real-world	✗	✓	✗	✗	✓
Quantity skew		✗	✗	✗	✓	✓

divide and assign the writers (and their digits) into each party randomly and equally. Since each party has different writers, the feature distributions are different among the parties. Like [6], we call this federated dataset as FEMNIST.

D. Quantity Skew

In quantity skew, the size of the local dataset $|\mathcal{D}^i|$ varies across parties. Although data distribution may still be consistent among the parties, it is interesting to see the effect of the quantity imbalance in FL. Like distribution-based label imbalance setting, we use Dirichlet distribution to allocate different amounts of data samples into each party. We sample $q \sim Dir_N(\beta)$ and allocate a q_j proportion of the total data samples to P_j. The parameter β can be used to control the imbalance level of the quantity skew. For ease of presentation, we use $q \sim Dir(\beta)$ to denote such a partitioning strategy.

E. Experiments in Existing Studies

Table I compares the partitioning strategies in NIID-bench with the experimental settings in existing studies. We can observe that each study only covers partial non-IID cases. It is impossible to directly compare the results presented in different papers. In contrast, NIID-bench consists of six partitioning strategies, which are more comprehensive and representative for representing different non-IID data cases.

V. EXPERIMENTS

To investigate the effectiveness of existing FL algorithms on non-IID data setting, we conduct extensive experiments on nine public datasets, including six image datasets (i.e., MNIST [38], CIFAR-10 [36], FMNIST [76], SVHN [58], FCUBE, FEMNIST [6]) and three tabular datasets (i.e., adult, rcv1, and covtype)[2]. The statistics of the datasets are summarized in Table II. For the image datasets, we use a CNN, which has two 5x5 convolution layers followed by 2x2 max pooling (the first with 6 channels and the second with 16 channels) and two fully connected layers with ReLU activation (the first with 120 units and the second with 84 units). For the tabular datasets, we use a MLP with three hidden layers. The numbers of hidden units of three layers are 32, 16, and 8. The number of parties is set to 10 by default, except for FCUBE where the number of parties is set to 4. All parties participate in every round to eliminate the effect of randomness brought by party sampling by default [56]. We use the SGD optimizer with learning rate 0.1 for rcv1 and learning rate 0.01 for the other datasets (tuned from $\{0.1, 0.01, 0.001\}$) and momentum 0.9. The batch size is set to 64 and the number of local epochs is set to 10 by default.

TABLE II
THE STATISTICS OF DATASETS IN THE EXPERIMENTS.

Datasets	#training instances	#test instances	#features	#classes
MNIST	60,000	10,000	784	10
FMNIST	60,000	10,000	784	10
CIFAR-10	50,000	10,000	1,024	10
SVHN	73,257	26,032	1,024	10
adult	32,561	16,281	123	2
rcv1	15,182	5,060	47,236	2
covtype	435,759	145,253	54	2
FCUBE	4,000	1,000	3	2
FEMNIST	341,873	40,832	784	10

Benchmark metrics. We use the top-1 accuracy on the test dataset as a metric to compare the studied algorithms. We run all the studied algorithms for the same number of rounds for fair comparison. The number of rounds is set to 50 by default unless specified.

Due to the page limit, for the experiments on the effect of batch size and model architecture, please refer to Appendix D and E of the technical report [39], respectively.

A. Overall Accuracy Comparison

The accuracy of existing approaches including FedAvg, FedProx, SCAFFOLD, and FedNova under different non-IID data settings is shown in Table III. For comparison, we also present the results for IID scenarios (i.e., homogeneous partitions). Next we show the insights from different perspectives.

1) Comparison among different non-IID settings:

Finding (1): The label distribution skew case where each party only has samples of a single class is the most challenging setting, while the feature distribution skew and quantity skew setting have little influence on the accuracy of FedAvg.

From Table III, we can observe that there is a gap between the accuracy of existing algorithms on several non-IID data settings and on the homogeneous setting. First, among different non-IID data settings, all studied FL algorithms perform worse on the label distribution skew case. Second, in label distribution skew setting, the algorithms have the worst accuracy when each party only has data from a single label. As expected, the accuracy increases as the number of classes in each party increases. Third, for feature distribution skew setting, except for CIFAR-10, existing algorithms have

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

TABLE III

The top-1 accuracy of different approaches. We run three trials and report the mean accuracy and standard derivation. For FedProx, we tune μ from $\{0.001, 0.01, 0.1, 1\}$ and report the best accuracy.

category	dataset	partitioning	FedAvg	FedProx	SCAFFOLD	FedNova
Label distribution skew	MNIST	$p_k \sim Dir(0.5)$	98.9%±0.1%	98.9%±0.1%	**99.0%±0.1%**	98.9%±0.1%
		$\#C = 1$	29.8%±7.9%	**40.9%±23.1%**	9.9%±0.2%	39.2%±22.1%
		$\#C = 2$	**97.0%±0.4%**	96.4%±0.3%	95.9%±0.3%	94.5%±1.5%
		$\#C = 3$	**98.0%±0.2%**	97.9%±0.4%	96.6%±1.5%	**98.0%±0.3%**
	FMNIST	$p_k \sim Dir(0.5)$	88.1%±0.6%	88.1%±0.9%	88.4%±0.5%	**88.5%±0.5%**
		$\#C = 1$	11.2%±2.0%	**28.9%±3.9%**	12.8%±4.8%	14.8%±5.9%
		$\#C = 2$	**77.3%±4.9%**	74.9%±2.6%	42.8%±28.7%	70.4%±5.1%
		$\#C = 3$	80.7%±1.9%	**82.5%±1.9%**	77.7%±3.8%	78.9%±3.0%
	CIFAR-10	$p_k \sim Dir(0.5)$	68.2%±0.7%	67.9%±0.7%	**69.8%±0.7%**	66.8%±1.5%
		$\#C = 1$	10.0%±0.0%	**12.3%±2.0%**	10.0%±0.0%	10.0%±0.0%
		$\#C = 2$	49.8%±3.3%	**50.7%±1.7%**	49.1%±1.7%	46.5%±3.5%
		$\#C = 3$	**58.3%±1.2%**	57.1%±1.2%	57.8%±1.4%	54.4%±1.1%
	SVHN	$p_k \sim Dir(0.5)$	86.1%±0.7%	86.6%±0.9%	**86.8%±0.3%**	86.4%±0.6%
		$\#C = 1$	11.1%±0.0%	**19.6%±0.0%**	6.7%±0.0%	10.6%±0.8%
		$\#C = 2$	**80.2%±0.8%**	79.3%±0.9%	62.7%±11.6%	75.4%±4.8%
		$\#C = 3$	82.0%±0.7%	**82.1%±1.0%**	77.2%±2.0%	80.5%±1.2%
	adult	$p_k \sim Dir(0.5)$	78.4%±0.9%	**80.5%±0.7%**	76.4%±0.0%	52.3%±26.7%
		$\#C = 1$	**82.5%±2.2%**	76.4%±0.0%	23.6%±0.0%	50.8%±0.9%
	rcv1	$p_k \sim Dir(0.5)$	48.2%±0.7%	**70.3%±13.3%**	64.4%±24.3%	49.3%±2.1%
		$\#C = 1$	**51.8%±0.7%**	**51.8%±0.7%**	**51.8%±0.7%**	**51.8%±0.7%**
	covtype	$p_k \sim Dir(0.5)$	**77.2%±7.4%**	70.9%±0.7%	67.7%±14.9%	74.8%±12.9%
		$\#C = 1$	48.8%±0.1%	**59.1%±2.1%**	49.6%±1.4%	50.4%±1.4%
number of times that performs the best			8	11	4	3
Feature distribution skew	MNIST	$\hat{\mathbf{x}} \sim Gau(0.1)$	**99.1%±0.1%**	**99.1%±0.1%**	**99.1%±0.1%**	**99.1%±0.1%**
	FMNIST		89.1%±0.3%	89.0%±0.2%	**89.3%±0.0%**	89.0%±0.1%
	CIFAR-10		68.9%±0.3%	69.3%±0.2%	**70.1%±0.2%**	68.5%±1.3%
	SVHN		**88.1%±0.5%**	**88.1%±0.2%**	**88.1%±0.4%**	**88.1%±0.4%**
	FCUBE	synthetic	**99.8%±0.2%**	**99.8%±0.0%**	99.7%±0.3%	99.7%±0.1%
	FEMNIST	real-world	**99.4%±0.0%**	99.3%±0.1%	**99.4%±0.1%**	99.3%±0.1%
number of times that performs the best			4	3	5	2
Quantity skew	MNIST	$q \sim Dir(0.5)$	**99.2%±0.1%**	**99.2%±0.1%**	99.1%±0.1%	99.1%±0.1%
	FMNIST		89.4%±0.1%	**89.7%±0.3%**	88.8%±0.4%	86.1%±2.9%
	CIFAR-10		**72.0%±0.3%**	71.2%±0.6%	62.4%±4.1%	10.0%±0.0%
	SVHN		88.3%±1.0%	**88.4%±0.4%**	11.0%±7.4%	41.3%±21.1%
	adult		82.2%±0.1%	**84.8%±0.2%**	81.6%±4.5%	43.2%±33.9%
	rcv1		96.7%±0.3%	**96.8%±0.4%**	49.0%±1.9%	51.8%±0.7%
	covtype		**88.1%±0.2%**	84.6%±0.2%	63.2%±20.8%	51.2%±3.2%
number of times that performs the best			3	5	0	0
Homogeneous partition	MNIST	IID	99.1%±0.1%	99.1%±0.1%	**99.2%±0.0%**	99.1%±0.1%
	FMNIST		89.6%±0.3%	89.5%±0.2%	**89.7%±0.2%**	89.4%±0.2%
	CIFAR-10		70.4%±0.2%	70.2%±0.1%	**71.5%±0.3%**	69.5%±1.0%
	SVHN		**88.5%±0.5%**	**88.5%±0.8%**	88.0%±0.8%	88.4%±0.5%
	FCUBE		99.7%±0.1%	99.6%±0.2%	99.8%±0.1%	**99.9%±0.1%**
	FEMNIST		99.3%±0.1%	**99.4%±0.1%**	**99.4%±0.0%**	99.3%±0.0%
	adult		82.6%±0.4%	**84.8%±0.2%**	83.8%±2.5%	82.6%±0.0%
	rcv1		**96.8%±0.4%**	96.6%±0.6%	80.9%±27.8%	96.6%±0.4%
	covtype		87.9%±0.1%	85.2%±0.0%	**88.0%±2.3%**	87.9%±0.2%
number of times that performs the best			2	3	5	1

a very close accuracy compared with the IID setting. Last, in quantity skew setting, FedAvg has almost no accuracy loss. Since the weighted averaging is adopted in FedAvg, it can already handle the quantity imbalance well. Overall, the label distribution skew influences the accuracy of FL algorithms most among all non-IID settings. There is room for existing algorithms to be improved to handle scenarios such as quantity-based label imbalance.

We draw a decision tree to summarize the suitable FL algorithm for each non-IID setting as shown in Figure 7 according to our observations. This decision tree is helpful for users to choose the algorithm for their learning according to the non-IID distribution and the datasets. For example, if the local datasets are likely to have feature distribution skew (e.g., the digits from different writers), then SCAFFOLD may be the best algorithm for FL. If the local datasets have almost the same data distribution but different sizes (e.g., databases with different capacities), then FedProx is likely the appropriate algorithm. If there is no prior knowledge on the local datasets, how to determine the distribution is a challenging problem and more research efforts are needed (see Section VI-A).

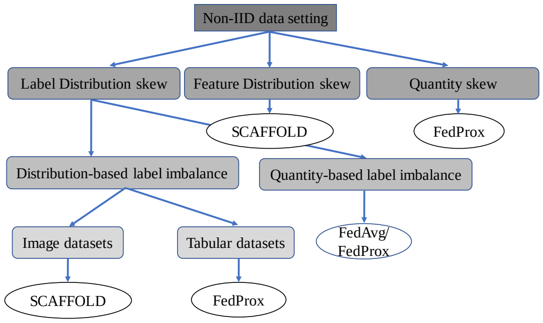


Fig. 7. The decision tree to determine the (almost) best FL algorithm given the non-IID setting.

2) Comparison among different algorithms:

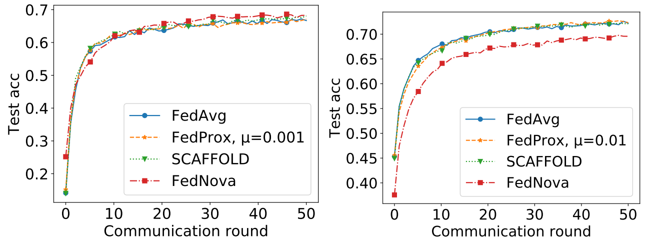
Finding (2): No algorithm consistently outperforms the other algorithms in all settings. The state-of-the-art algorithms significantly outperform FedAvg only in several cases.

We have the following observations in aspect of different algorithms. First, in label distribution skew and quantity skew cases, FedProx usually achieves the best accuracy. In feature distribution skew case, SCAFFOLD usually achieves the best accuracy. Second, in some cases (e.g., $p_k \sim Dir(0.5)$, feature distribution skew and quantity skew), the improvement of the three non-IID FL algorithms is insignificant compared with FedAvg, which is smaller than 1%. Third, when $\#C = 1$, FedProx can significantly outperform FedAvg, SCAFFOLD and FedNova. Fourth, for SCAFFOLD, its accuracy is quite unstable. It can significantly outperform the other two approaches in some cases (e.g., $Dir(0.5)$ and $K = 1$ on CIFAR-10). However, it may also have much worse accuracy than the other two approaches (e.g., $K = 1$ and $K = 2$ on SVHN). Last, for FedNova, it does not show much superiority compared with other FL algorithms. Compared with the accuracy of FedAvg on the homogeneous partition, there is still a lot of room for improvement in the non-IID setting.

3) Comparison among different tasks:

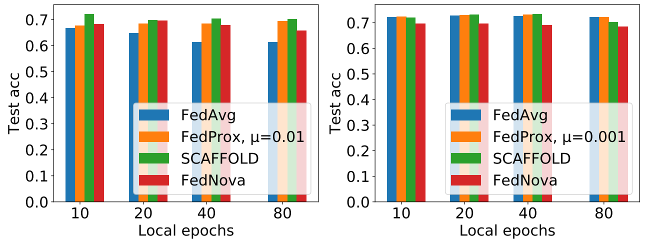
Finding (3): CIFAR-10 and tabular datasets are challenging tasks under non-IID settings. MNIST is a simple task under most non-IID settings where the studied algorithms perform similarly well.

Among nine different datasets, while heterogeneity significantly degrades the accuracy of FL algorithms on CIFAR-10 and tabular datasets, such influence is smaller in other datasets. Among image datasets, the classification task on CIFAR-10 is more complex than the other datasets in a centralized setting. Thus, when each party only has a skewed subset, the task will be more challenging and the accuracy is worse. Also, it is interesting that all the four algorithms cannot handle tabular datasets well in the non-IID setting. The accuracy loss is quite large especially for the label distribution skew case. We suggest that the challenging tasks like CIFAR-10 and rcv1 should be included in the benchmark for distributed data silos.



(a) $p_k \sim Dir(0.5)$ (b) $\hat{\mathbf{x}} \sim Gau(0.1)$

Fig. 8. The training curves of different approaches on CIFAR-10.



(a) $p_k \sim Dir(0.5)$ (b) $\hat{\mathbf{x}} \sim Gau(0.1)$

Fig. 9. The test accuracy with different numbers of local epochs on CIFAR-10.

B. Communication Efficiency

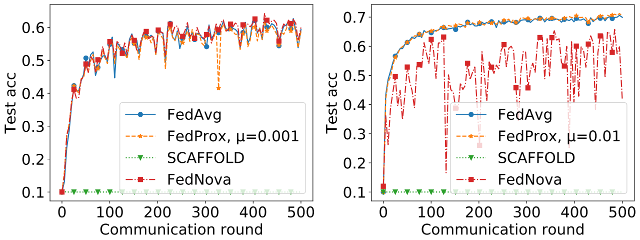
Finding (4): FedProx has almost the same convergence speed compared with FedAvg, while SCAFFOLD and FedNova are more unstable in training.

Figure 8 shows the training curves of the studied algorithms on CIFAR-10. Here we try two different partitioning strategies that cover label skew and feature skew. For the results on other partitioning strategies and other datasets, please refer to Appendix A of our technical report [39]. For FedProx, we show the curve with the best μ. First, for the $\#C = 1$ setting, FedAvg and FedProx are very unstable, while SCAFFOLD and FedNova even cannot improve as the number of rounds increases. Second, for the $q \sim Dir(0.5)$ setting, FedNova is quite unstable and the accuracy changes rapidly as the number of communication rounds increases. Moreover, FedProx is very close to FedAvg during the whole training process in many cases. Since the best μ is always small, the regularization term in FedProx has little influence on the training. Thus, FedProx and FedAvg usually have similar convergence speed and final accuracy. How to achieve stable learning and fast convergence is still an open problem on non-IID data.

C. Robustness to Local Updates

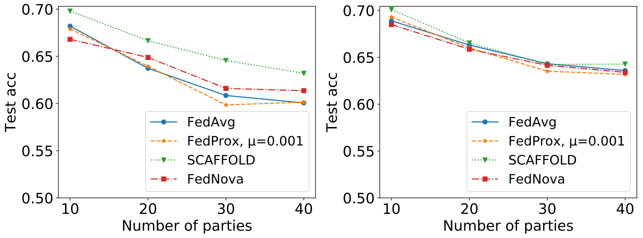
Finding (5): The number of local epochs can have a large effect on the accuracy of existing algorithms. The optimal value of the number of local epochs is very sensitive to non-IID distributions.

We vary the number of local epochs from $\{10, 20, 40, 80\}$ and report the final accuracy on CIFAR-10 in Figure 9. Please refer to Appendix B of our technical report [39] for the results of other settings and datasets. On the one hand, we can find



(a) $p_k \sim Dir(0.5)$ (b) $q \sim Dir(0.5)$

Fig. 10. The training curves of different approaches on CIFAR-10 with 100 parties and sample fraction 0.1.



(a) $p_k \sim Dir(0.5)$ (b) $\hat{\mathbf{x}} \sim Gau(0.1)$

Fig. 11. The test accuracy with different number of parties on CIFAR-10.

that the number of local epochs has a large effect on the accuracy of FL algorithms. For example, when $\#C = 2$, the accuracy of all algorithms generally degrades significantly when the number of local epochs is set to 80. On the other hand, the optimal number of local epochs differ in different settings. For example, when $\#C = 1$ and $\#C = 2$, the optimal number of local epochs is 20 for FedAvg, and is 10 on the settings $p_k \sim Dir(0.5)$ and $\#C = 3$. In summary, existing algorithms are not robust enough against large local updates. Non-IID distributions have to be considered to determine the best number of local epochs.

D. Party Sampling

Finding (6): In the partial participation setting, SCAFFOLD cannot work effectively, while the other FL algorithms have a very unstable accuracy during training.

In some scenarios, not all the data silos will participate the entire training process. In such a setting, the sampling technique is usually applied (Line 6 of Algorithm 1). To simulate this scenario, we set the number of parties to 100 and the sample fraction to 0.1. We run experiments on CIFAR-10 and the results are shown in Figure 10. Please refer to Appendix C of the technical report [39] for the results with other partitioning strategies. We can find that the training curves are quite unstable in most non-IID settings. Due to the sampling technique, the local distributions among different rounds can vary, and thus the averaged gradients may have very different directions among rounds. Moreover, we can find that SCAFFOLD has a bad accuracy on all settings. Since the frequency of updating local control variates (Lines 23-25 of Algorithm 2) is low, the estimation of the update direction may be very inaccurate using the control variates.

E. Scalability

Finding (7): The accuracy of all approaches decrease when increasing the number of parties.

We study the effect of number of clients on studied approaches as shown in Figure 11. Here we run all approaches for 50 rounds. We can observe that the accuracy decreases significantly when increasing the number of clients. When the number of parties is large, the amount of local data is small and it is easy to overfit in the local training stage. How to design effective and communication-efficient algorithms on a large-scale setting with small data in the client is still an open problem.

TABLE IV
THE COMPUTATION TIME (SECOND) AND COMMUNICATION SIZE (MB) PER ROUND OF DIFFERENT APPROACHES.

	MNIST	CIFAR-10	adult	rcv1
FedAvg	73s	193s	15s	66s
FedProx	133s	233s	44s	76s
SCAFFOLD	77s	197s	14s	66s
FedNova	73s	189s	17s	65s
FedAvg	1.95MB	2.73MB	0.20MB	66.54MB
FedProx	1.95MB	2.73MB	0.20MB	66.54MB
SCAFFOLD	3.91MB	5.46MB	0.41MB	133.08MB
FedNova	1.95MB	2.73MB	0.20MB	66.54MB

F. Efficiency

Finding (8): The computation overhead of FedProx is large compared with FedAvg. Moreover, the communication cost of SCAFFOLD is twice of that of FedAvg.

To compare the efficiency of different FL algorithms, we show the overall computation time and communication costs of each approach in Table IV. We can observe that the computation costs of FedAvg, SCAFFOLD, and FedNova are close. FedProx has a much higher computation cost than the other algorithms. From Algorithm 1, FedProx directly modifies the objective, which causes additional computation overhead in the gradient descent of each batch. FedNova and SCAFFOLD only introduce very small number of addition and multiplication operations each round, which is negligible. For the communication costs, since SCAFFOLD needs to communicate control variates in each round as shown in Algorithm 2, its communication cost is twice of that of the other algorithms.

G. Mixed Types of Skew

Finding (9): FL is more challenging when there exists mixed types of skew among the local data.

In practice, there may exist mixed types of skew among parties. Here we combine multiple partitioning strategies to generate such cases. We try two different settings: 1) we first divide the whole dataset into each party by the distribution-based label imbalanced partitioning strategy. Then, we add noises to the data of each party according to the noise-based feature imbalance strategy. Therefore, there exists both label

TABLE V
THE PERFORMANCE OF DIFFERENT APPROACHES WITH DIFFERENT IMBALANCE CASES ON CIFAR-10.

Case 1	FedAvg	FedProx	SCAFFOLD	FedNova
label skew	68.2%	67.9%	69.8%	66.8%
feature skew	68.9%	69.3%	70.1%	68.5%
label and feature skew	66.1%	64.8%	67.8%	65.9%

Case 2	FedAvg	FedProx	SCAFFOLD	FedNova
feature skew	68.9%	69.3%	70.1%	68.5%
quantity skew	72.0%	71.2%	62.4%	10.0%
feature and quantity skew	69.1%	69.2%	62.2%	10.0%

distribution skew and feature distribution skew among the local data of different parties. 2) we first divide the whole dataset into each party by the quantity imbalanced partitioning strategy. Then, we add noises to the data of each party according to the noise-based feature imbalance strategy. Therefore, there exists both feature distribution skew and quantity skew among the local data of different parties. The results are shown in Table V.

For the first case, we can observe that the accuracies of all approaches degrade when there exists mixed types of skew compared with a single type of skew, which is reasonable since both label imbalance and feature imbalance bring challenges in the training process.

For the second case, while quantity skew does not affect the accuracy of FedAvg and FedProx, the accuracy of both feature and quantity skew setting is close to the accuracy of the feature skew setting. However, for SCAFFOLD and FedNova, the accuracy of both feature and quantity skew setting is poor since quantity skew degrades the accuracy significantly.

Overall, as we observe more significant model quality degradation in mixed non-IID settings, it is an important direction to design algorithms for settings with mixed types of skew, which are common in reality. For example, the images taken in different areas have different label distributions, while the feature distributions also differ due to the cameras (e.g., contrast).

H. Insights on the Experimental Results

We summarized the insights from the experimental studies as follows.

- The design and evaluation of future FL algorithms should consider more comprehensive settings, including different non-IID data partitioning strategies and tasks. There is not a single studied algorithm that consistently outperforms the other algorithms or has a good performance in all settings. Thus, it is still a promising research direction to address issues in distributed data silos with FL.
- Accuracy and communication efficiency are two important metrics in the evaluation of FL algorithms under non-IID data settings. Our study demonstrates the trade-off between them, and also the stability of those two metrics in the training process.
- FL introduces new training factors (e.g., number of local epochs, batch normalization, party sampling, number of parties) compared with centralized training due to non-IID data setting, while some training factors share the similar behavior as the centralized training (e.g., batch size). These challenging factors deserve more attention in the evaluation of future FL studies.
- Mixed types of skew brings more challenges than a single type of skew. As we observe more significant model quality degradation in mixed non-IID settings, it is important to investigate effective algorithms working on multiple types of skew, which is more practical in reality.

VI. FUTURE DIRECTIONS

We present some following promising future directions for data management and federated learning on non-IID distributed databases.

A. Opportunities for data management

Integration with learned database systems: Existing learned systems are mostly based on centralized databases, such as learned index structures [12], [54] and learned cost estimation [24], [55]. We believe that, as the concerns on data privacy and data regulation grow, we will see more distributed databases and existing learned systems and algorithms need to be revisited. For example, it could be very interesting to enable federated search and develop learned index structures for multiple "data silos" without exchanging the local data.

Light-weight data techniques for profiling non-IID data: From our experimental study, different non-IID distributions have a large effect on the accuracy and stability of FL algorithms. Thus, it would be helpful if we can know the non-IID distribution in prior before conducting FL. This made a decade of database research relevant, such as data sampling [7] and sketching [20]. Another potential approach is to use mete data to represent the non-IID distributions. However, it is still an open problem on how to extend current statistics estimation (such as cardinality estimation) to non-IID distribution.

Non-IID resistant sampling for partial participation: As in Finding (8), the sampling approach can bring instability in FL. Instead of random sampling, selective sampling according to the data distribution features of the parties may significantly increase the learning stability. One inspiration is from the skew resistant data techniques [18], [37], which can be potentially extended to the partial participation in FL training. Moreover, stratified sampling [59] can be a good solution. By classifying the parties to subgroups, representative parties can be selected in each round in a more balanced way [1].

Privacy-preserving data mining: Although there is no raw data transfer in FL, the model may still leak sensitive information about the training data due to possible inference attacks [17], [68]. Thus, techniques such as differential privacy [14] are useful to protect the local databases. How to decrease the accuracy loss while ensuring the differential privacy guarantee is a challenge research direction.

Query on Federated Databases: As we focus on distributed databases due to privacy concerns, federated databases [67] also need to be revisited. On the one hand, how to combine the SQL query with machine learning on federated databases is an important problem. On the other hand, how to preserve the data privacy while supporting both query and learning on federated databases also needs to be investigated.

B. Opportunities for better FL design

A Party with a Single Label: From Table III, the accuracy of FL algorithms is very bad if each party only has data of a single label. This setting is seemingly unrealistic. However, it has many real-world applications in practice. For example, we can use FL to train a speaker recognition model, while each mobile device only has the voices of its single user.

Fast Training: From Figure 8, the training speed of existing FL algorithms are usually close to each other. FedProx, SCAFFOLD, and FedNova do not show much superiority on the communication efficiency. To improve the training speed, researchers can work on the following two directions. One possible solution is to develop communication-efficient FL algorithms with only a few rounds. There are some studies [21], [41] that propose FL algorithms using a single communication round. In their studies, a public dataset is needed, which may potentially limit the applications. Another possible solution is to develop fast initialization approach to reduce the number of rounds while achieving the same accuracy for FL. In the experiments of a previous study [41], they show that their approach is also promising if applied as an initialization step.

Automated Parameter Tuning for FL: FL algorithms suffer from large local updates. The number of local epochs is an important parameter in FL. While one traditional way is to develop approaches robustness to the local updates, another way is to design efficient parameter tuning approaches for FL. A previous paper [9] studies Bayesian optimization in the federated setting, which can be used to search hyper-parameters. Approaches for the setting of number of local epochs need to be investigated.

Towards Robust Algorithms against Different Non-IID Settings: As in Finding (2), no algorithm consistently performs the best in all settings. It is a natural question whether and how we can develop a robust algorithm for different non-IID settings. We may have to first investigate the common characteristics of FL processes under different non-IID settings. The intuitions of existing algorithms are same: the local model updates towards the local optima, and the averaged model is far from the global optima. We believe the design of FL algorithms under non-IID settings can be improved if we can observe more detailed and common behaviours in the training.

Aggregation of Heterogeneous Batch Normalization: From our Finding (7), simple averaging is not a good choice for batch normalization. Since the batch normalization in each party records the statistics of local data distribution, there is also heterogeneity among the batch normalization layers of different parties. The averaged batch normalization layer may not catch the local distribution after sending back to the parties. A possible solution is to only average the learned parameters but leave the statistics (i.e., mean and variance) alone [4]. More specialized designs for particular layers in deep learning need to be investigated.

VII. Related Work

Although the existing study [33] provides non-IID data cases, it does not provide the partitioning strategies to generate the corresponding non-IID data distributions. We go beyond the previous study and summarize six different partitioning strategies to generate three non-IID data distribution cases. Among these six partitioning strategies, the two partitioning strategies in Section IV-A-b and Section IV-B-c are adopted from existing FL studies due to their popularity, while the other four effective partitioning strategies are designed by our study. Next, we introduce these partitioning strategies in detail.

There are some existing benchmarks for federated learning [6], [26], [29], [49]. LEAF [6] provides some realistic federated datasets including images and texts. Specifically, LEAF partitions the existing datasets according to its data recourses, e.g., partitioning the data in Extended MNIST [8] based on the writer of the digit or character. OARF [29] proposes federated datasets by combining multiple related real-world public datasets. Moreover, it provides various metrics including utility, communication overhead, privacy loss, and mimics the federated systems in the real world. However, both LEAF and OARF do not provide an algorithm-level comparison. FedML [26] provides reference implementations of federated learning algorithms such as FedAvg, FedNOVA [72] and FedOpt [62]. There are no new datasets, metrics, and settings in FedML. FLBench [49] is proposed for isolated data island scenario. Its framework covers domains including medical, finance, and AIoT. However, currently, FLBench is not open-sourced and it does not provide any experiments.

The above benchmarks do not provide analysis of existing federated learning algorithms on different non-IID settings, which is our focus in this paper. To the best of our knowledge, there is one existing benchmark [51] for federated learning on the non-IID data setting. However, it only provides two partitioning approaches: random split and split by labels. In this paper, we provide comprehensive partitioning strategies and datasets to cover different non-IID settings. Moreover, we conduct extensive experiments to compare and analyze existing federated learning algorithms.

VIII. Conclusion

There has been a growing interest in exploiting distributed databases (e.g., in different organizations and countries) to improve the effectiveness of machine learning services. In this paper, we study non-IID data as one key challenge in such distributed databases, and develop a benchmark named NIID-bench. Specifically, we introduce six data partitioning strategies which are much more comprehensive than the previous studies. Furthermore, we conduct comprehensive experiments to compare existing algorithms and demonstrate their strength and weakness. This study sheds light on some future directions to build effective machine learning services on distributed databases.

References

[1] S. AbdulRahman, H. Tout, A. Mourad, and C. Talhi. Fedmccs: multicriteria client selection model for optimal iot federated learning. *IEEE Internet of Things Journal*, 8(6):4723–4735, 2020.

[2] D. A. E. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, and V. Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

[3] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 439–450, 2000.

[4] M. Andreux, J. O. du Terrail, C. Beguier, and E. W. Tramel. Siloed federated learning for multi-centric histopathology datasets. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*, pages 129–139. Springer, 2020.

[5] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. M. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander. Towards federated learning at scale: System design. In *SysML*, 2019.

[6] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[7] S. Chaudhuri, R. Motwani, and V. Narasayya. Random sampling for histogram construction: How much is enough? *ACM SIGMOD Record*, 27(2):436–447, 1998.

[8] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[9] Z. Dai, B. K. H. Low, and P. Jaillet. Federated bayesian optimization via thompson sampling. *Advances in Neural Information Processing Systems*, 33, 2020.

[10] Y. Deng, M. M. Kamani, and M. Mahdavi. Distributionally robust federated averaging. *Advances in Neural Information Processing Systems*, 33, 2020.

[11] Diemert Eustache, Meynet Julien, P. Galland, and D. Lefortier. Attribution modeling increases efficiency of bidding in display advertising. In *Proceedings of the AdKDD and TargetAd Workshop, KDD, Halifax, NS, Canada, August, 14, 2017*, page To appear. ACM, 2017.

[12] J. Ding, U. F. Minhas, J. Yu, C. Wang, J. Do, Y. Li, H. Zhang, B. Chandramouli, J. Gehrke, D. Kossmann, D. Lomet, and T. Kraska. Alex: An updatable adaptive learned index. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, page 969–984, New York, NY, USA, 2020. Association for Computing Machinery.

[13] C. T. Dinh, N. H. Tran, and T. D. Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 2020.

[14] C. Dwork. Differential privacy. *Encyclopedia of Cryptography and Security*, pages 338–340, 2011.

[15] A. Fallah, A. Mokhtari, and A. Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33, 2020.

[16] X. Y. Felix, A. S. Rawat, A. K. Menon, and S. Kumar. Federated learning with only positive labels. *arXiv preprint arXiv:2004.10342*, 2020.

[17] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333. ACM, 2015.

[18] S. Ganguly, P. B. Gibbons, Y. Matias, and A. Silberschatz. Bifocal sampling for skew-resistant join size estimation. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, page 271–281, New York, NY, USA, 1996. Association for Computing Machinery.

[19] R. C. Geyer, T. Klein, and M. Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[20] A. C. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 389–398, 2002.

[21] N. Guha, A. Talwlkar, and V. Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

[22] F. Hanzely, S. Hanzely, S. Horváth, and P. Richtárik. Lower bounds and optimal algorithms for personalized federated learning. *Advances in Neural Information Processing Systems*, 2020.

[23] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[24] S. Hasan, S. Thirumuruganathan, J. Augustine, N. Koudas, and G. Das. Deep learning models for selectivity estimation of multi-attribute queries. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, page 1035–1050, New York, NY, USA, 2020. Association for Computing Machinery.

[25] C. He, M. Annavaram, and S. Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems*, 33, 2020.

[26] C. He, S. Li, J. So, M. Zhang, H. Wang, X. Wang, P. Vepakomma, A. Singh, H. Qiu, L. Shen, et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

[27] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] T.-M. H. Hsu, H. Qi, and M. Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[29] S. Hu, Y. Li, X. Liu, Q. Li, Z. Wu, and B. He. The oarf benchmark suite: Characterization and implications for federated learning systems. *arXiv preprint arXiv:2006.07856*, 2020.

[30] J. Huang. Maximum likelihood estimation of dirichlet distribution parameters. *CMU Technique Report*, 2005.

[31] N. Hynes, D. Dao, D. Yan, R. Cheng, and D. Song. A demonstration of sterling: A privacy-preserving data marketplace. *Proceedings of the VLDB Endowment*, 11(12):2086–2089, 2018.

[32] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26:315–323, 2013.

[33] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[34] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, pages 1–7, 2020.

[35] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020.

[36] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[37] Y. Kwon, M. Balazinska, B. Howe, and J. Rolia. Skew-resistant parallel processing of feature-extracting scientific user-defined functions. In *Proceedings of the 1st ACM Symposium on Cloud Computing*, SoCC '10, page 75–86, New York, NY, USA, 2010. Association for Computing Machinery.

[38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[39] Q. Li, Y. Diao, Q. Chen, and B. He. Federated learning on non-iid data silos: An experimental study. *arXiv preprint arXiv:2102.02079*, 2021.

[40] Q. Li, B. He, and D. Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[41] Q. Li, B. He, and D. Song. Practical one-shot federated learning for cross-silo setting. *IJCAI*, 2021.

[42] Q. Li, Z. Wen, and B. He. Practical federated gradient boosting decision trees. In *AAAI*, pages 4642–4649, 2020.

[43] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, and B. He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *arXiv preprint arXiv:1907.09693*, 2019.

[44] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873*, 2019.

[45] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020.

[46] T. Li, J. Zhong, J. Liu, W. Wu, and C. Zhang. Ease.ml: Towards multi-tenant resource sharing for machine learning workloads. 11(5):607–620, Jan. 2018.

[47] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2020.

[48] X. Li, M. JIANG, X. Zhang, M. Kamp, and Q. Dou. Fed{bn}: Federated learning on non-{iid} features via local batch normalization. In *International Conference on Learning Representations*, 2021.

[49] Y. Liang, Y. Guo, Y. Gong, C. Luo, J. Zhan, and Y. Huang. An isolated data island benchmark suite for federated learning. *arXiv preprint arXiv:2008.07257*, 2020.

[50] T. Lin, L. Kong, S. U. Stich, and M. Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[51] L. Liu, F. Zhang, J. Xiao, and C. Wu. Evaluation framework for large-scale federated learning. *arXiv preprint arXiv:2003.01575*, 2020.

[52] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang. A secure federated transfer learning framework. *IEEE Intelligent Systems*, 2020.

[53] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[54] R. Marcus, A. Kipf, A. van Renen, M. Stoian, S. Misra, A. Kemper, T. Neumann, and T. Kraska. Benchmarking learned indexes. *Proc. VLDB Endow.*, 14(1):1–13, Sept. 2020.

[55] R. Marcus, P. Negi, H. Mao, C. Zhang, M. Alizadeh, T. Kraska, O. Papaemmanouil, and N. Tatbul. Neo: A learned query optimizer. *Proc. VLDB Endow.*, 12(11):1705–1718, July 2019.

[56] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[57] M. Mohri, G. Sivek, and A. T. Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019.

[58] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[59] J. Neyman. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. In *Breakthroughs in statistics*, pages 123–150. Springer, 1992.

[60] C. Niu, Z. Zheng, F. Wu, X. Gao, and G. Chen. Trading data in good faith: Integrating truthfulness and privacy preservation in data markets. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 223–226. IEEE, 2017.

[61] X. Peng, Z. Huang, Y. Zhu, and K. Saenko. Federated adversarial domain adaptation. In *International Conference on Learning Representations*, 2020.

[62] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[63] A. Reisizadeh, F. Farnia, R. Pedarsani, and A. Jadbabaie. Robust federated learning: The case of affine distribution shifts. *Advances in Neural Information Processing Systems*, 2020.

[64] S. J. Rizvi and J. R. Haritsa. Maintaining data privacy in association rule mining. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*, pages 682–693. Elsevier, 2002.

[65] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

[66] S. Shastri, V. Banakar, M. Wasserman, A. Kumar, and V. Chidambaram. Understanding and benchmarking the impact of gdpr on database systems. *Proc. VLDB Endow.*, 13(7):1064–1077, Mar. 2020.

[67] A. P. Sheth and J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys (CSUR)*, 22(3):183–236, 1990.

[68] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[69] M. J. Smith, C. Sala, J. M. Kanter, and K. Veeramachaneni. The machine learning bazaar: Harnessing the ml ecosystem for effective system development. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, page 785–800, New York, NY, USA, 2020. Association for Computing Machinery.

[70] P. Voigt and A. Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017.

[71] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020.

[72] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

[73] L. Wang, S. Xu, X. Wang, and Q. Zhu. Addressing class imbalance in federated learning. In *AAAI*, 2021.

[74] W. Wang, J. Gao, M. Zhang, S. Wang, G. Chen, T. K. Ng, B. C. Ooi, J. Shao, and M. Reyad. Rafiki: Machine learning as an analytics service system. *Proc. VLDB Endow.*, 12(2):128–140, Oct. 2018.

[75] Y. Wu, S. Cai, X. Xiao, G. Chen, and B. C. Ooi. Privacy preserving vertical federated learning for tree-based models. *Proceedings of the VLDB Endowment*, 2020.

[76] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[77] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[78] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni. Bayesian nonparametric federated learning of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019.

[79] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

Appendix

A. *Training Curves*

Figures 12, 13, 14, 15, and 16 show the training curves of different approaches on the studied datasets except CIFAR-10.

B. *Number of Local Epochs*

Figures 12, 18, 19, 20, and 21 show the accuracy with different number of local epochs on the studied datasets except CIFAR-10.

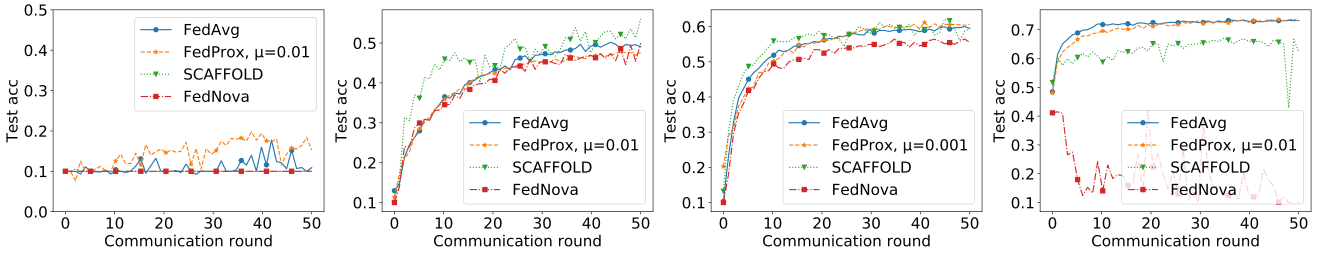
C. *Party Sampling*

Figure 22 shows the training curves of the studied approaches on CIFAR-10 under the party sampling setting.

D. *Batch Size*

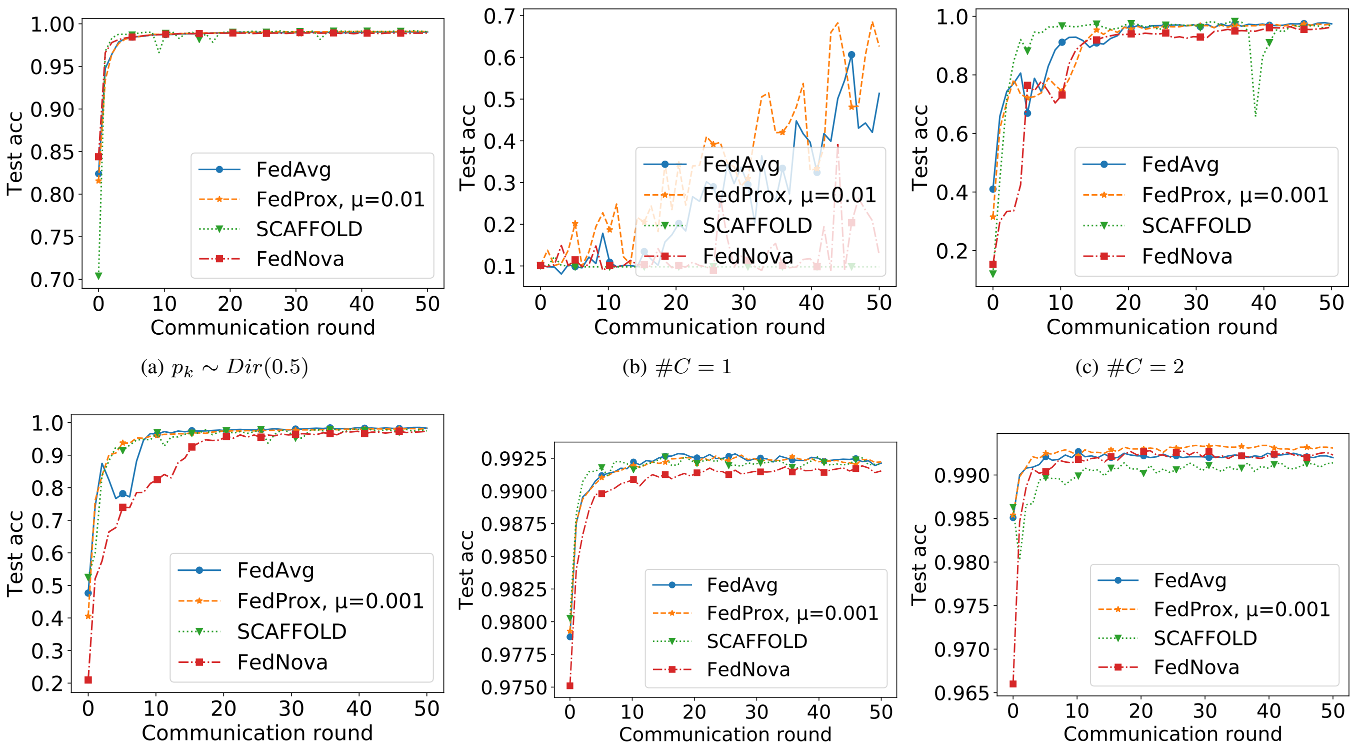
Finding (10): The heterogeneity of local data does not appear to influence the behaviors of different choices of batch sizes.

Batch size is an important hyper-parameter in deep learning. We choose FedAvg and FedProx as the representative algorithms and study the effect of batch size in FL by varying it from 16 to 256 as shown in Figure 23. Like centralized training, a large batch size slows down the learning process. Moreover, four studied algorithms have similar behaviours given different batch sizes. The results demonstrate that there is no clear relationship between the setting of batch size and the heterogeneity of local data. The knowledge of the behaviors of different batch sizes still applies in the non-IID federated setting.



(a) $\#C = 1$ (b) $\#C = 2$ (c) $\#C = 3$ (d) $q \sim Dir(0.5)$

Fig. 12. The training curves of different approaches on CIFAR-10.



(a) $p_k \sim Dir(0.5)$ (b) $\#C = 1$ (c) $\#C = 2$

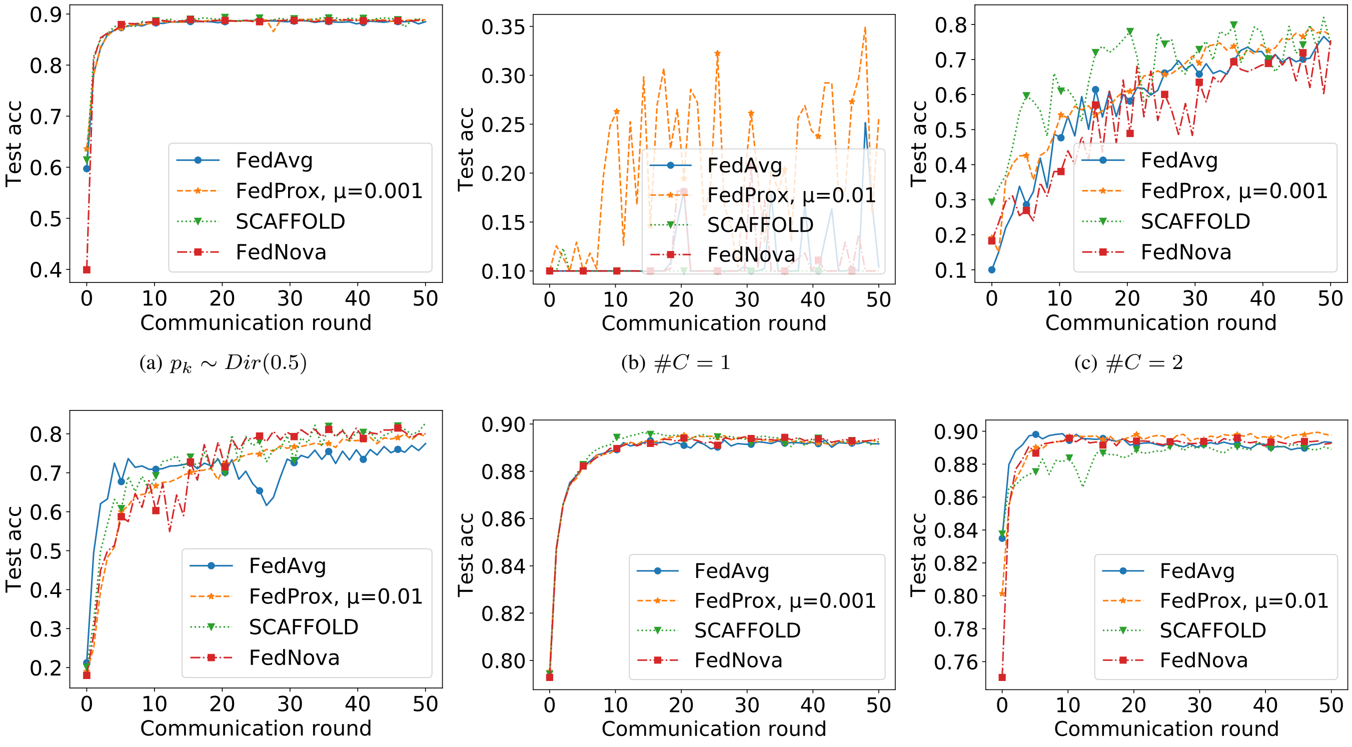
(d) $\#C = 3$ (e) $\hat{\mathbf{x}} \sim Gau(0.1)$ (f) $p \sim Dir(0.5)$

Fig. 13. The training curves of different approaches on MNIST.

E. Model Architectures

Finding (11): A simple averaging of batch normalization layers introduces instability in non-IID setting.

In the previous experiments, the models we use are simple CNNs and MLPs. Here we try more complex models including VGG-9 and ResNet-50 [27]. The experimental results on CIFAR-10 are shown in Figure 24. Overall, while the final accuracies of using VGG-9 and ResNet-50 are usually close, training a ResNet-50 appears to more unstable. ResNet-50 uses batch normalization to standardize the inputs to a layer. A challenge in training ResNet-50 is how to aggregate the batch normalization layers. While the local batch normalization layers can handle the local distribution well, a simple averaging of these layers may not be able to catch the statistics of global distribution and introduces more instability.



(a) $p_k \sim Dir(0.5)$

(b) $\#C = 1$

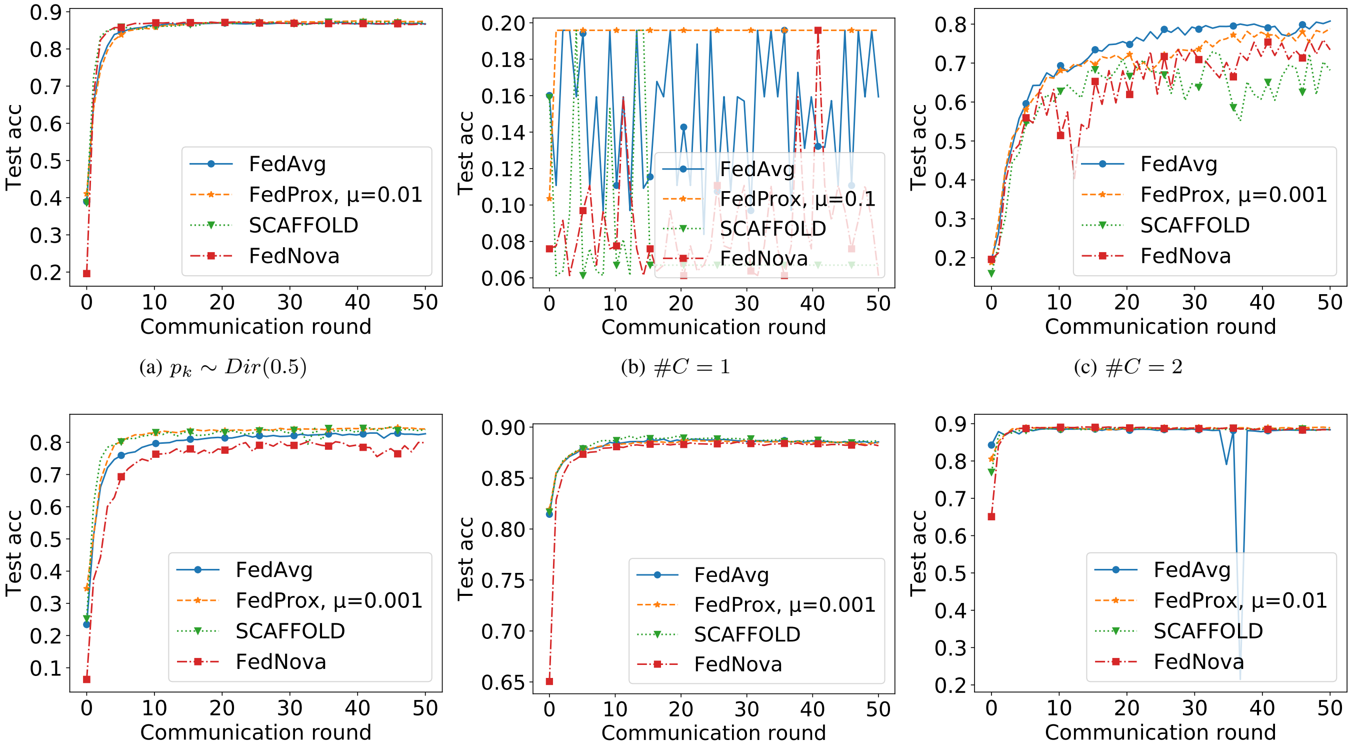
(c) $\#C = 2$

(d) $\#C = 3$

(e) $\hat{\mathbf{x}} \sim Gau(0.1)$

(f) $p \sim Dir(0.5)$

Fig. 14. The training curves of different approaches on FMNIST.



(a) $p_k \sim Dir(0.5)$

(b) $\#C = 1$

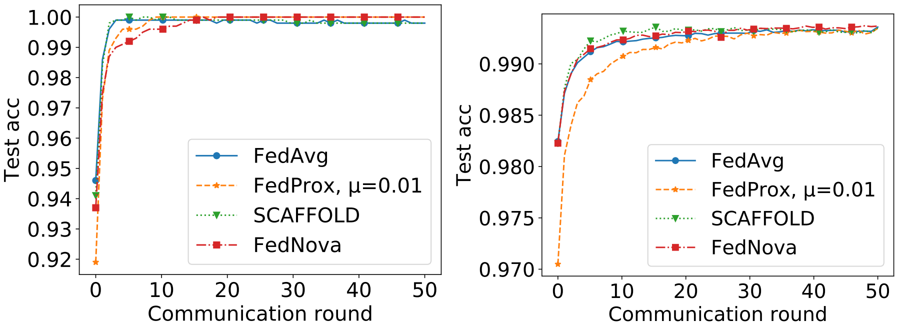
(c) $\#C = 2$

(d) $\#C = 3$

(e) $\hat{\mathbf{x}} \sim Gau(0.1)$

(f) $p \sim Dir(0.5)$

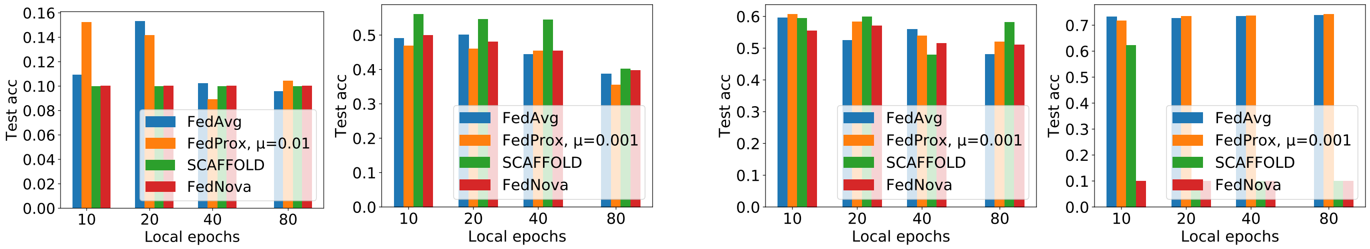
Fig. 15. The training curves of different approaches on SVHN.



(a) FCUBE

(b) FEMNIST

Fig. 16. The training curves of different approaches on FCUBE and FEMNIST.



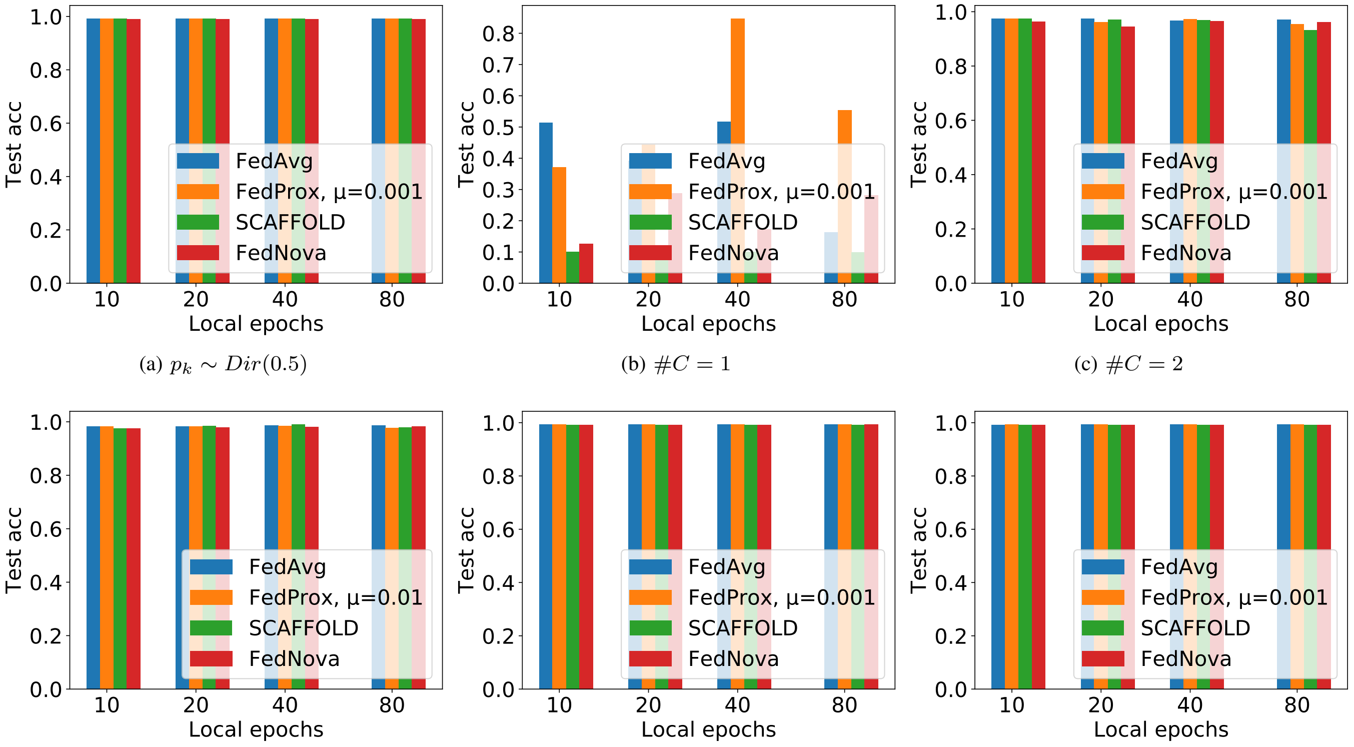
(a) $\#C = 1$

(b) $\#C = 2$

(c) $\#C = 3$

(d) $q \sim Dir(0.5)$

Fig. 17. The test accuracy with different numbers of local epochs on CIFAR-10.



(a) $p_k \sim Dir(0.5)$

(b) $\#C = 1$

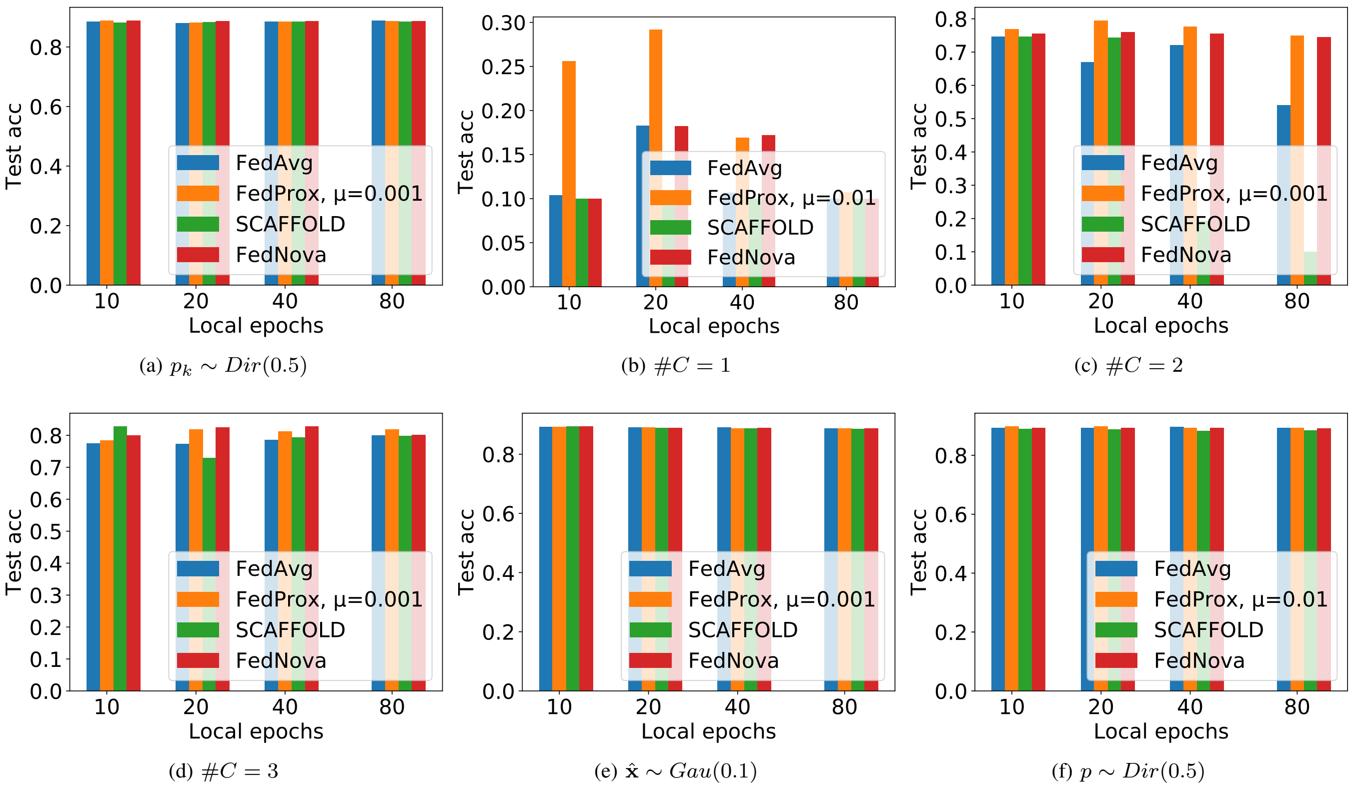
(c) $\#C = 2$

(d) $\#C = 3$

(e) $\hat{\mathbf{x}} \sim Gau(0.1)$

(f) $p \sim Dir(0.5)$

Fig. 18. The test accuracy with different number of local epochs on MNIST.



(a) $p_k \sim Dir(0.5)$

(b) $\#C = 1$

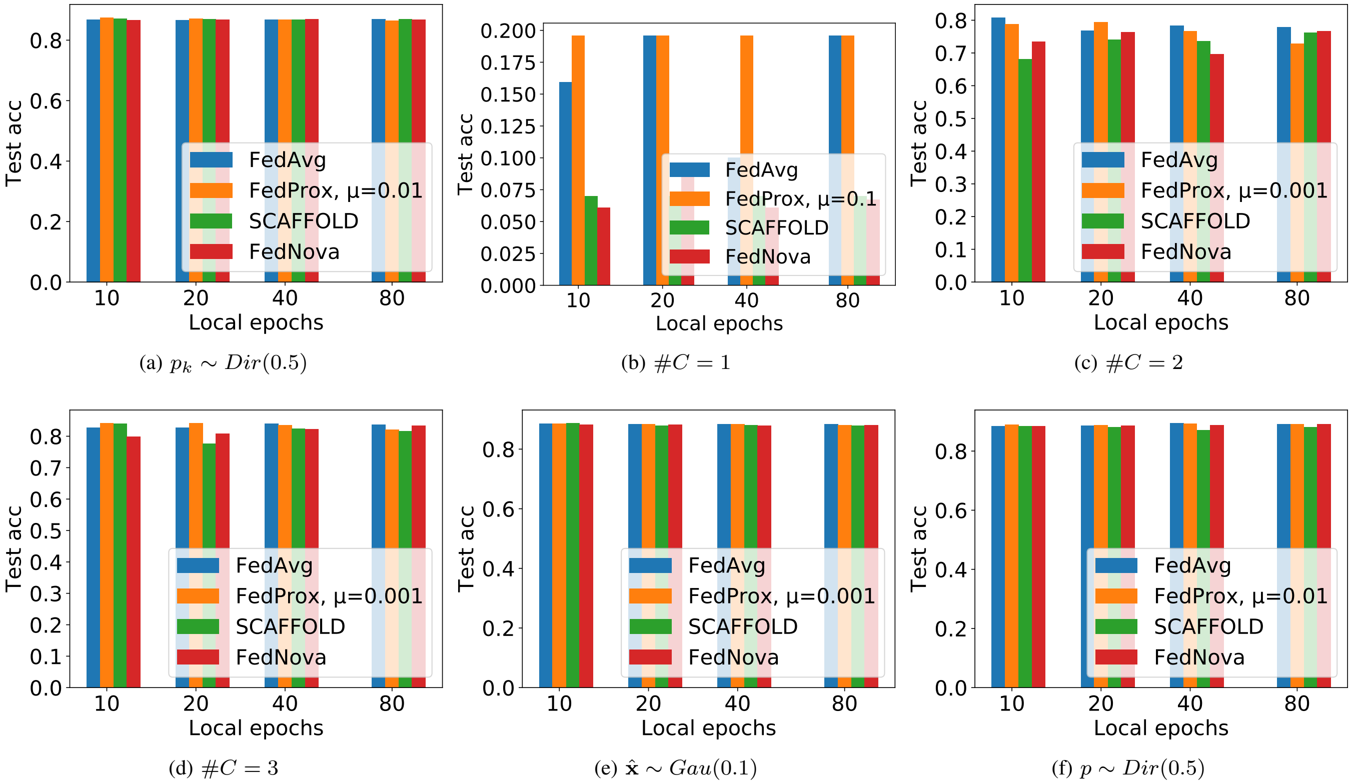
(c) $\#C = 2$

(d) $\#C = 3$

(e) $\hat{\mathbf{x}} \sim Gau(0.1)$

(f) $p \sim Dir(0.5)$

Fig. 19. The test accuracy with different number of local epochs on FMNIST.



(a) $p_k \sim Dir(0.5)$

(b) $\#C = 1$

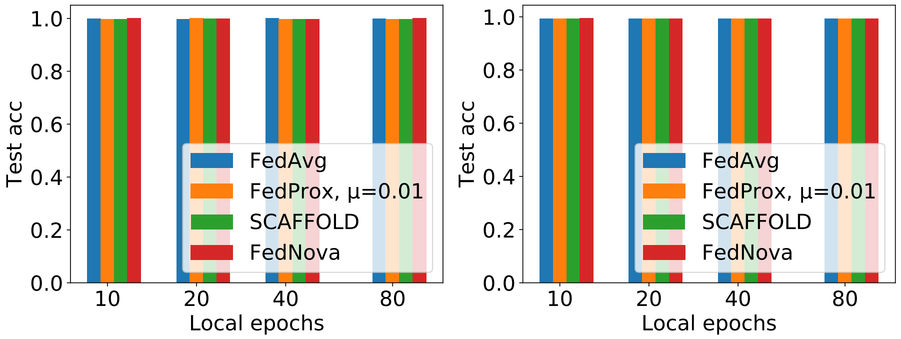
(c) $\#C = 2$

(d) $\#C = 3$

(e) $\hat{\mathbf{x}} \sim Gau(0.1)$

(f) $p \sim Dir(0.5)$

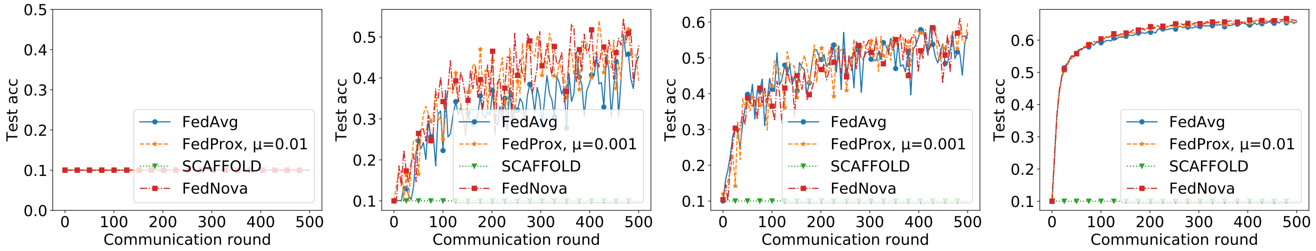
Fig. 20. The test accuracy with different number of local epochs on SVHN.



(a) FCUBE

(b) FEMNIST

Fig. 21. The test accuracy with different number of local epochs on FCUBE and FEMNIST.



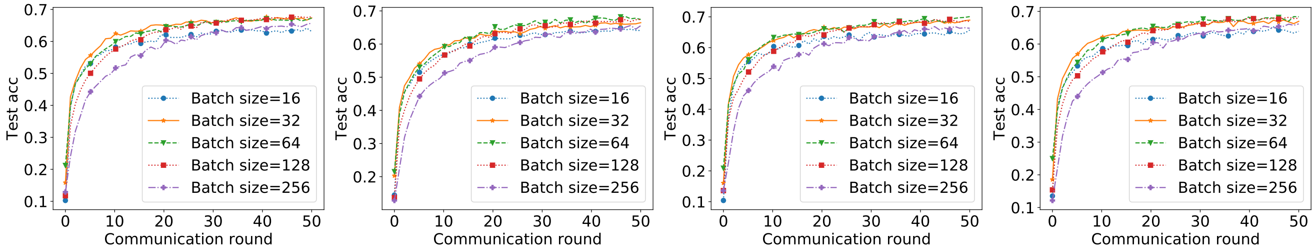
(a) $\#C = 1$

(b) $\#C = 2$

(c) $\#C = 3$

(d) IID

Fig. 22. The training curves of different approaches on CIFAR-10 with 100 parties and sample fraction 0.1.



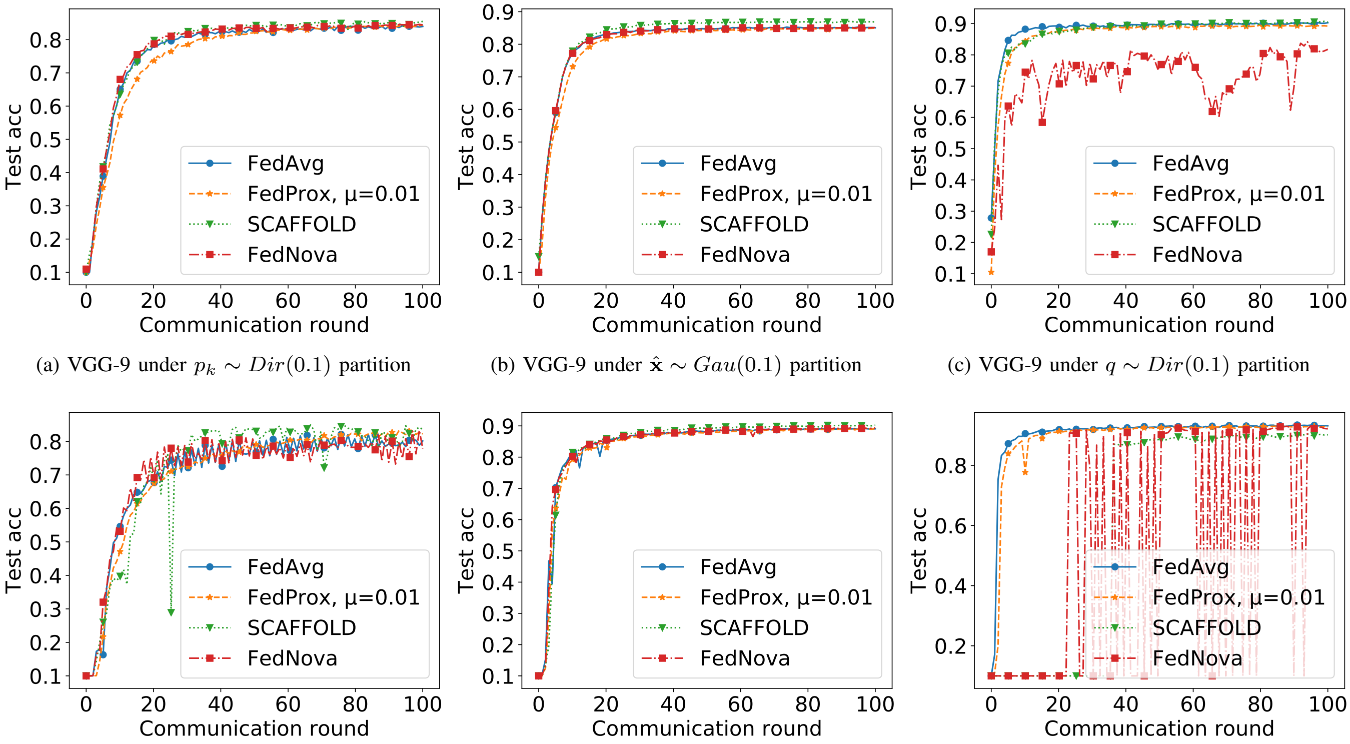
(a) FedAvg

(b) FedProx ($\mu = 0.01$)

(c) SCAFFOLD

(d) FedNova

Fig. 23. The training curves of different batch sizes on CIFAR-10 under $p_k \sim Dir(0.5)$ partition.



(a) VGG-9 under $p_k \sim Dir(0.1)$ partition

(b) VGG-9 under $\hat{\mathbf{x}} \sim Gau(0.1)$ partition

(c) VGG-9 under $q \sim Dir(0.1)$ partition

(d) ResNet-50 under $p_k \sim Dir(0.1)$ partition

(e) ResNet-50 under $\hat{\mathbf{x}} \sim Gau(0.1)$ partition

(f) ResNet-50 under $q \sim Dir(0.1)$ partition

Fig. 24. The training curves of VGG-9/ResNet-50 on CIFAR-10 under different partitions.