FedCluster: Boosting the Convergence of Federated Learning via Cluster-Cycling

Cheng Chen*
Department of ECE
University of Utah
u0952128@utah.edu

Ziyi Chen*
Department of ECE
University of Utah
ziyi.chen@utah.edu

Yi Zhou
Department of ECE
University of Utah
yi.zhou@utah.edu

Bhavya Kailkhura
Lawrence Livermore National Lab
San Francisco, US
kailkhura1@llnl.gov

***Abstract*—We develop FedCluster – a novel federated learning framework with improved optimization efficiency, and investigate its theoretical convergence properties. The FedCluster groups the devices into multiple clusters that perform federated learning cyclically in each learning round. Therefore, each learning round of FedCluster consists of multiple cycles of meta-update that boost the overall convergence. In nonconvex optimization, we show that FedCluster with the devices implementing the local stochastic gradient descent (SGD) algorithm achieves a faster convergence rate than the conventional federated averaging (FedAvg) algorithm in the presence of device-level data heterogeneity. We conduct experiments on deep learning applications and demonstrate that FedCluster converges significantly faster than the conventional federated learning under diverse levels of device-level data heterogeneity for a variety of local optimizers.**

***Index Terms*—Federated learning, clustering, SGD**

I. Introduction

Federated learning has become an emerging distributed machine learning framework that enables edge computing at a large scale [21], [25], [30]. As opposed to traditional centralized machine learning that collects all the data at a central server to perform learning, federated learning exploits the distributed computation power and data of a massive number of edge devices to perform distributed machine learning while preserving full data privacy. In particular, federated learning has been successfully applied to the areas of Internet of things (IoT), autonomous driving, health care, etc. [25]

The original federated learning framework was proposed in [30], where the federated averaging (FedAvg) training algorithm was developed. Specifically, in each learning round of FedAvg, a subset of devices are activated to download a model from the cloud server and train the model using their local data for multiple stochastic gradient descent (SGD) iterations. Then, the devices upload the trained local models to the cloud, where the local models are aggregated and averaged to obtain an updated global model to be used in the next round of learning. In this learning process, the data are privately kept in the devices. However, the convergence rate of the FedAvg algorithm is heavily affected by the device-level data heterogeneity of the devices, which has been shown both empirically and theoretically to slow down the convergence of FedAvg [27], [43].

To alleviate the negative effect of device-level data heterogeneity and facilitate the convergence of federated learning, various algorithms have been developed in the literature. For example, the federated proximal (FedProx) algorithm studied in [26], [35] proposed to regularize the local loss function with the square distance between the local model and the global model, which helps to reduce the heterogeneity of the local models. Other federated learning algorithms apply variance reduction techniques to reduce the variance of local stochastic gradients caused by device-level data heterogeneity, e.g., FedMAX [5], FEDL [8], VRL-SGD [29], FedSVRG [21], [32] and PR-SPIDER [37]. Moreover, [17] uses control variates to control the local model difference, which is similar to the variance reduction techniques. The MIME framework [18] combines arbitrary local update algorithms with server-level statistics to control local model difference. In FedNova [38], the devices adopt different numbers of local SGD iterations and use normalized local gradients to adapt to the device-level data heterogeneity. Although these federated learning algorithms achieve improved performance over the traditional FedAvg algorithm, their optimization efficiency is unsatisfactory due to the infrequent model update in the cloud, i.e., every round of these federated learning algorithms only performs one global update (i.e., the model average step) to update the global model. Such an update rule does not fully utilize the big data to make the maximum optimization progress, and the convergence rates of these federated learning algorithms do not benefit from the large population of the devices. Therefore, it is much desired to develop an advanced federated learning framework that achieves improved optimization convergence and efficiency with the same amount of resources as those of the existing federated learning frameworks.

In this paper, we propose a novel federated learning framework named FedCluster, in which the devices are grouped into multiple clusters to perform federated learning in a cyclic way. The proposed federated learning framework has the flexibility to implement any conventional federated learning algorithms and can boost their convergence with the same amount of computation & communication resources. We summarize our contributions as follows.

*The authors contributed equally to this work.

This work was supported by the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

A. Our Contributions

We propose FedCluster, a novel federated learning framework that groups the devices into multiple clusters for flexible management of the devices. In each learning round, the clusters of devices are activated to perform federated learning in a cyclic order. In particular, the FedCluster framework has the following amenable properties: 1) The clusters of devices in FedCluster can implement any federated learning algorithm, e.g., FedAvg, FedProx, etc. Hence, FedCluster provides a general optimization framework for federated learning; and 2) In each learning round, FedCluster updates the global model multiple times while consuming the same amount of computation & communication resources as those consumed by the traditional federated learning framework.

Theoretically, we proved that in nonconvex optimization, FedCluster with the devices implementing the local SGD algorithm converges at a sub-linear rate $\mathcal{O}(\frac{1}{\sqrt{TME}})$, where T, M, E correspond to the number of learning rounds, clusters and local SGD iterations, respectively. As a comparison, the convergence rate of the conventional FedAvg is in the order of $\mathcal{O}(\frac{1}{\sqrt{TE}})$ [28], which is slower than FedCluster with local SGD by a factor of $\sqrt{M}$. In addition, our convergence rate only depends on cluster-level data heterogenity denoted as H_{cluster} (see (2)), while FedAvg depends on a device-level data heterogeneity H_{device} that is larger than H_{cluster}. Therefore, FedCluster with local SGD achieves a faster convergence rate and suffers less from the device-level data heterogeneity than FedAvg.

Empirically, we compare the performance of FedCluster with that of the conventional centralized federated learning with different local optimizers in deep learning applications. We show that FedCluster achieves significantly faster convergence than the conventional centralized federated learning under different levels of device-level data heterogeneity and local optimizers. We also explore the impact of the number of clusters and cluster-level data heterogeneity on the performance of FedCluster.

B. Related Work

The conventional federated learning framework along with the FedAvg algorithm was first introduced in [30]. The convergence rate of FedAvg was studied in strongly convex optimization [20], [27], [40], convex optimization [2], [19], [20], [40] and nonconvex optimization [20], [28]. [20] studied the convergence rate of decentralized SGD, a generalization of FedAvg. To address the device-level data heterogeneity issue in federated learning, [26], [35] proposed the FedProx algorithm that adds the square distance between the local model and the global model to the local loss function, [34] applied adaptive optimizers such as Adagrad, Adam, and YOGI to the update of the global model, and variance reduction techniques have been applied to develop advanced federated learning algorithms including Federated SVRG [21], [32], FEDL [8], FedMAX [5], VRL-SGD [29] and PR-SPIDER [37]. [41] proposed a FedMeta algorithm that uses a keep-trace gradient descent strategy originated from model-agnostic meta-learning [11], but the local updates involve high-order derivatives. [33] developed an ASD-SVRG algorithm to address the heterogeneity of the local smoothness parameters, but the algorithm suffers from a high computation cost due to frequent communication and full gradient computation in the variance reduction scheme. [36] performed FedAvg and bisection clustering in turn until convergence, which is time consuming without theoretical convergence guarantee. [4] also used clustering scheme where each cluster performs FedAvg and periodically communicate with neighbor clusters in a decentralized manner. The semi-federated learning framework [6] also proposed to cluster the devices, but the devices take only one local SGD update per learning round and require D2D communication. The Federated Augmentation (FAug) [14] used generative adversarial networks to augment the local device dataset to make it closer to an i.i.d. dataset.

Some other works focus on personalized federated learning. For example, [10] trained different local models instead of a shared global model. [7], [12] trained a mixture of global and local models and [1] combined the global and local models to make prediction for online learning. Readers can refer to [23] for an overview of personalized federated learning. Other studies of federated learning include privacy [13], [15], [39], adversarial attacks [3], [15] and fairness [31]. Please refer to [16] for a comprehensive overview of these topics.

II. FedCluster: Federated Learning with Cluster-Cycling

In this section, we introduce the FedCluster framework and compare it with the traditional federated learning framework. To provide a fair comparison, we impose the following practical resource constraint on the devices in federated learning systems.

Assumption 1 (Resource constraint)**.** *In each round of federated learning, every device has a maximum budget of two communications (one download and one upload) and a fixed number of local updates.*

Fig. 1 (Left) illustrates a particular learning round of the traditional centralized federated learning system. To elaborate, in each learning round, a subset of devices are activated to download a model from the cloud server and train the model using their local data for multiple iterations. The training algorithm can be any stochastic optimization algorithm, e.g., local SGD in FedAvg [30] and regularized local SGD in FedProx [26]. After the local training, the activated devices upload their trained local models to the cloud, where the local models are aggregated and averaged to obtain an updated model to be used in the next learning round. We note that the devices of the traditional federated learning system satisfy the resource constraint in Assumption 1. However, the convergence of the traditional federated learning process is slowed down by two important factors: 1) the heterogeneous local datasets of the devices and 2) the infrequent global model update in each learning round.

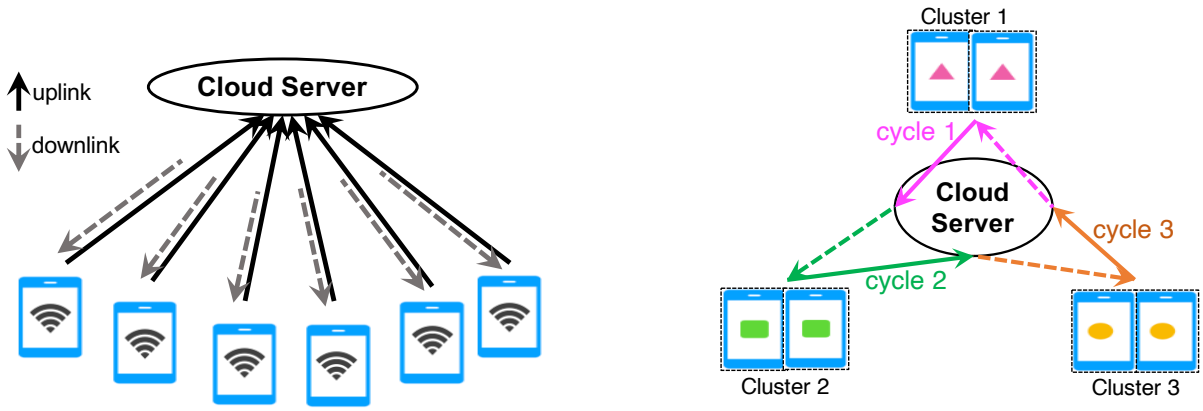

Fig. 1. Left: Traditional federated learning system. Right: FedCluster system with cluster-cycling.

To improve the optimization efficiency and flexibility of federated learning, we propose the FedCluster framework as illustrated in Fig. 1 (Right). To elaborate, in the FedCluster system, devices are grouped into multiple clusters using a certain clustering approach (elaborated later). In each learning round, the system performs multiple cycles of federated learning through the clusters in a cyclical way. Specifically, as illustrated in Fig. 1 (Right), in the first cycle of a learning round, a subset of devices of Cluster 1 are activated to perform federated learning, i.e., they download a model from the cloud server, perform local trainings using a certain algorithm (e.g., SGD) and upload the trained local models to the cloud for model averaging. Then, in the next cycle, a subset of devices of Cluster 2 are activated to perform federated learning. Following this strategy, all the clusters take turns to perform federated learning in a cyclic order. We note that the devices of FedCluster satisfy the resource constraint in Assumption 1, as each device is only activated at most once per learning round.

Comparison: The key differences between the FedCluster framework and the traditional federated learning framework are in two-fold. First, in each learning round, FedCluster updates the global model multiple times (equals to number of clusters), whereas the traditional federated learning updates it only once. Hence, FedCluster is expected to make more optimization progress per learning round. In fact, as the devices in federated learning are usually busy and unavailable, the clusters in FedCluster provide much flexibility to schedule the learning tasks for the devices and make frequent updates on the global model (see the discussion in the next paragraph.). Second, the traditional federated learning process suffers from the device-level data heterogeneity. In comparison, as we show later in the analysis section, the convergence of the FedCluster learning process is affected by a smaller cluster-level data heterogeneity.

Generality and flexibility: We further elaborate various aspects of generality and flexibility of the FedCluster framework.

- *Generality:* The traditional federated learning framework in Fig. 1 (Left) can be viewed as a special case of the FedCluster framework with only one cluster that includes all the devices.
- *Algorithm:* The clusters of FedCluster can implement any federated learning algorithms that are compatible with the traditional federated learning framework, e.g., FedAvg [30], FedProx [26], etc.
- *Clustering:* In FedCluster, the way to cluster the devices depends on the specific application scenario. Below we provide several representative clustering approaches.
  1) Random uniform clustering: The devices are grouped into multiple clusters of equal size uniformly at random. In this case, the clusters are homogeneous and have similar data statistics.
  2) Timezone-based clustering: In mobile networks where the devices are smart phones that are distributed over the world, one can group the devices based on either their timezones or GPS locations. This scenario fits the FedCluster system well because many smart phones are available only at a particular local time (e.g., midnight) to perform federated learning. In this case, the federated learning process cycles through the smart phones in different timezones.
  3) Availability-based clustering: A more general approach is to divide each learning round into multiple time slots. Each device determines its available time slot to perform federated learning. Then, the available devices within each time slot form a cluster.

III. Convergence Analysis of FedCluster with Local SGD

In this section, we analyze the convergence of FedCluster in smooth nonconvex optimization.

A. Problem Formulation and Algorithm

We consider a federated learning system that consists of n devices. Each device k possesses a local data set $\mathcal{D}^{(k)}$ with $|\mathcal{D}^{(k)}|$ data samples. Denote $\mathcal{D} = \cup_{k=1}^{n} \mathcal{D}^{(k)}$ as the total dataset and denote $p_k = |\mathcal{D}^{(k)}|/|\mathcal{D}|$ as the proportion of data possessed by the device k. Then, we aim to solve the following empirical risk minimization problem

$$\min_{w \in \mathbb{R}^d} \; f(w) = \sum_{k=1}^{n} p_k f(w; \mathcal{D}^{(k)}), \tag{P}$$

where $f(w;\mathcal{D}^{(k)}) = \frac{1}{|\mathcal{D}^{(k)}|}\sum_{\xi\in\mathcal{D}^{(k)}} f(w;\xi)$ corresponds to the average loss on the local dataset.

In FedCluster, we group all the devices $\{1,2,...,n\}$ into M clusters $\{\mathcal{S}_1,\mathcal{S}_2,...,\mathcal{S}_M\}$. The learning process of FedCluster with local SGD is presented in Algorithm 1. To elaborate, in each learning round of FedCluster, we first sample a subset of devices from each cluster to be activated. Then, in the inner cycles of this learning round, the clusters sequentially perform federated learning using the FedAvg algorithm. Specifically in each cycle, the cloud server sends the current model to the activated devices of the cluster to initialize their local models. Then, these devices perform multiple local SGD iterations in parallel and send the trained local models to the cloud for model averaging. After that, the updated global model is sent to the activated devices of the cluster in the next cycle.

Algorithm 1 FedCluster with local SGD

Input: Initialization model $W_0\in\mathbb{R}^d$, learning rate $\eta_{j,K,t}$.
for rounds $j=0,1,\ldots,T-1$ **do**
  for cycles $K=0,1,...,M-1$ **do**
    Sample a subset of devices $S_{K+1}^{(j)}$ from cluster $\mathcal{S}_{K+1}$.
    Cloud sends W_{jM+K} to the sampled devices.
    for all devices $k\in S_{K+1}^{(j)}$ **in parallel do**
      Initialize $w_{j,K,0}^{(k)} = W_{jM+K}$.
      for $t=0,\ldots,E-1$ **do**
        Sample a local data point $\xi_{j,K,t}^{(k)}\in\mathcal{D}^{(k)}$ uniformly at random. Update $w_{j,K,t+1}^{(k)} = w_{j,K,t}^{(k)} - \eta_{j,K,t}\nabla f(w_{j,K,t}^{(k)};\xi_{j,K,t}^{(k)})$.
      end
      Send the local model $w_{j,K,E}^{(k)}$ to the cloud.
    end
    Cloud computes $W_{jM+K+1} = \sum_{k\in\mathcal{S}'_{j,\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} w_{j,K,E}^{(k)}$.
  end
end
Output: W_{TM}

Remark: Algorithm 1 can be easily extended to the partial participation setting in which only a subset of the devices in every cluster are randomly selected to participate in each learning round.

We adopt the following standard assumptions on the loss function of the problem (P) [27].

Assumption 2. *The loss function in the problem (P) satisfies the following conditions.*

1) *For any data point ξ, function $f(\cdot;\xi)$ is L-smooth and bounded below.*
2) *For any $w\in\mathbb{R}^d$, the variance of stochastic gradients sampled by each device k is bounded by s_k^2, i.e.,*
$$\mathbb{E}_{\xi\sim\mathcal{D}^{(k)}}\|\nabla f(w;\xi)-\nabla f(w;\mathcal{D}^{(k)})\|^2 \le s_k^2.$$
3) *There exists $G>0$ such that for all $w_{j,K,t}^{(k)}$ and $\overline{w}_{j,K,t}$ (defined in (5)) generated by Algorithm 1, it holds that $\mathbb{E}_{\xi\sim\mathcal{D}^{(k)}}\|\nabla f(w_{j,K,t}^{(k)};\xi)\|^2 \le G^2$ and $\mathbb{E}_{\xi\sim\mathcal{D}^{(k)}}\|\nabla f(\overline{w}_{j,K,t};\xi)\|^2 \le G^2$.*

We note that the item 3 of Assumption 2 directly implies that for any $j=0,1,\ldots,T-1$, $K,K'=0,1,\ldots,M-1$, $t=0,1,\ldots,E-1$
$$\max\left\{\|\nabla f(\overline{w}_{j,K,t};\mathcal{D}^{(\mathcal{S}_{K'})})\|, \|\nabla f(\overline{w}_{j,K,t})\|\right\} \le G. \quad (1)$$

B. Convergence Result

In this subsection, we analyze the convergence rate of FedCluster with local SGD specified in Algorithm 1 under nonconvexity of the loss function in the problem (P). In particular, we consider the simplified full participation setup, in which all the devices are activated in each learning round (i.e., $S_{K+1}^{(j)} = \mathcal{S}_{K+1}$). To simplified the analysis, we assume the clusters are chosen cyclically with reshuffle in each round to perform federated learning.

Throughout the analysis, we define $q_K := \sum_{k\in\mathcal{S}_K} p_k$ and $f(w;\mathcal{D}^{(\mathcal{S}_K)}) := q_K^{-1}\sum_{k\in\mathcal{S}_K} p_k f(w;\mathcal{D}^{(k)})$, which characterizes the total loss of the cluster $\mathcal{S}_K$. Then, we adopt the following definition of cluster-level data heterogeneity that corresponds to the variance of the gradient on the local data possessed by the clusters.

$$H_{\text{cluster}} := \sup_{w\in\mathbb{R}^d}\Big(\mathbb{E}\sum_{K=1}^{M} q_K\|\nabla f(w;\mathcal{D}^{(\mathcal{S}_K)}) - \nabla f(w)\|^2\Big). \quad (2)$$

We obtain the following convergence result of FedCluster with local SGD in the nonconvex setting. Please refer to the appendix for the details of the proof.

Theorem 1. *Let Assumption 2 hold and assume that $f(\cdot;\xi)$ is nonconvex for any data sample ξ. Choose learning rate $\eta_{j,K,t}\equiv(TME)^{-\frac{1}{2}}$ and choose E,M,T such that $ME\le\frac{C}{8LG^2}$, $T\ge L^2\max(1,\frac{16}{EM})$. Then, under full participation of the devices, the output of Algorithm 1 satisfies*
$$\frac{1}{T}\sum_{j=0}^{T-1}\mathbb{E}\Big\|\nabla f(W_{jM})\Big\|^2 \le \frac{2C}{\sqrt{TME}}, \quad (3)$$
where the constant C is defined as
$$C = 2\mathbb{E}\big(f(W_0) - \inf_{w\in\mathbb{R}^d} f(w)\big) + 4L\Big(H_{cluster} + \sum_{K=1}^{M} q_K^{-1}\sum_{k\in\mathcal{S}_K} p_k^2 s_k^2\Big). \quad (4)$$
Furthermore, in order to achieve a solution that satisfies $\frac{1}{T}\sum_{j=0}^{T-1}\mathbb{E}\|\nabla f(W_{jM})\|^2\le\epsilon$, the required number of learning rounds satisfies $T\propto\frac{C^2}{ME}$.

Therefore, under cluster-cycling, FedCluster with local SGD enjoys a convergence rate $\mathcal{O}(\frac{1}{\sqrt{TME}})$ in nonconvex optimization, which is faster than the convergence rate $\mathcal{O}(\frac{1}{\sqrt{TE}})$ of the FedAvg algorithm [28]. Also, the above convergence rate of FedCluster depends on the cluster-level data heterogeneity H_{cluster}, whereas the convergence rate of the FedAvg algorithm

depends on the larger device-level data heterogeneity $H_{\text{device}} := \sup_{w\in\mathbb{R}^d}(\mathbb{E}\sum_{k=1}^n p_k\|\nabla f(w;\mathcal{D}^{(k)}) - \nabla f(w)\|^2)$ (It is easy to show that $H_{\text{cluster}} \leq H_{\text{device}}$). The complexity result (3) also implies a trade-off between the number of clusters M and the constant C. Specifically, with more clusters (i.e., a larger M), it can be shown that both the cluster-level data heterogeneity H_{cluster} and the local variance $\sum_{K=1}^M q_K^{-1}\sum_{k\in\mathcal{S}_K} p_k^2 s_k^2$ in C increase. Hence, a proper choice of M that minimizes $\frac{C}{\sqrt{M}}$ can yield the fastest convergence rate in nonconvex case. On the other hand, given a fixed number of clusters M, a proper clustering approach that minimizes the cluster-level data heterogeneity H_{cluster} can also improve the convergence rate. We further explore the impact of these factors on the convergence speed of FedCluster in the following experiment section.

IV. Experiments

A. Experiment Setup

In this section, we compare the performance of FedCluster with that of the traditional federated learning in deep learning applications. We consider completing a standard classification tasks on two datasets – CIFAR-10 [22] and MNIST [24], using the AlexNet model [42] and the cross-entropy loss. We simulate 1000 devices for both FedCluster and the traditional federated learning system, and each device possesses 500 data samples. Specifically, the dataset of each device is specified by a major class and a device-level data heterogeneity ratio $\rho_{\text{device}} \in [0.1, 1]$. Take the CIFAR-10 dataset as an example, each of its ten classes is assigned as the major class of 100 devices. Then, $\rho_{\text{device}} \times 100\%$ of the samples of each device are sampled from the major class, and $(1-\rho_{\text{device}})/9 \times 100\%$ of the samples are sampled from each of the other classes. Hence, a larger ρ_{device} corresponds to a higher device-level data heterogeneity. For FedCluster, by default we cluster the devices into 10 clusters uniformly at random.

For the traditional federated learning system, in every learning round we randomly activate 10% of the devices to participate in the training. By default, these activated devices run $E = 20$ local SGD steps with batch size 30 using fine-tuned learning rates $\eta_t \equiv 0.1$ and 0.05 for CIFAR-10 and MNIST, respectively. For the FedCluster system, in every learning cycle we randomly activate 10% of the devices of the corresponding cluster to participate in the training. These activated devices run $E = 20$ local SGD steps with batch size 30 using the learning rates $\eta_t \equiv 0.01$ and 0.005 for CIFAR-10 and MNIST, respectively. In particular, to make a fair comparison, these choices of learning rates are one tenth of those adopted by the traditional federated learning system, as in each cycle only one of the ten clusters is involved (hence less number of data samples are used). We note that the learning rates adopted by FedCluster are not fine-tuned. We also implement a centralized SGD as a baseline, which adopts 1000 iterations per learning round, batch size 60 and fine-tuned learning rates 0.01 and 0.005 for CIFAR-10 and MNIST, respectively. This ensures that the federated learning algorithms and the centralized SGD consume the same number of samples (i.e., 60k training samples) per learning round. All experiments with a given model and dataset are initialized at the same point.

B. Comparison under Different Device-level Data Heterogeneities

We first compare FedCluster (with local SGD) with the conventional FedAvg under different levels of device-level data heterogeneity ratios, i.e., $\rho_{\text{device}} = 0.1, 0.4, 0.7, 0.9$. We present the train loss and test accuracy results on CIFAR-10 in Fig. 2, where the top row presents the results of $\rho_{\text{device}} = 0.1, 0.4$ and the bottom row presents the results of $\rho_{\text{device}} = 0.7, 0.9$. It can be seen that FedCluster achieves faster convergence than FedAvg in terms of both train loss and test accuracy under different levels of ρ_{device}, which demonstrates the advantage of FedCluster in both training efficiency and generalization ability with heterogeneous data.

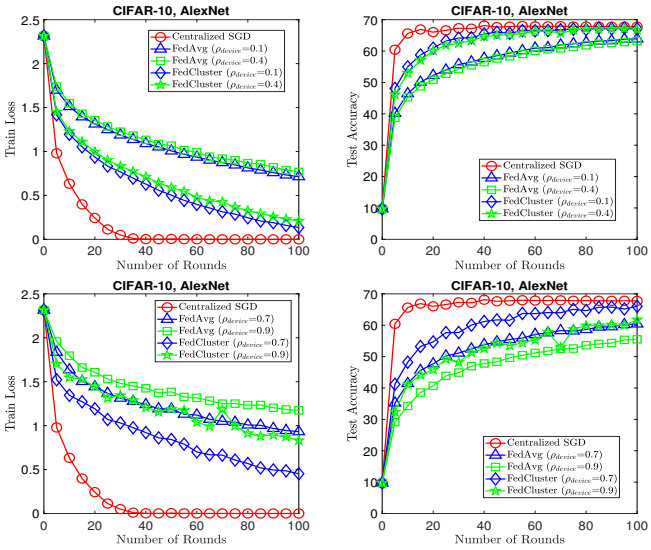

Fig. 2. Comparison between FedCluster and FedAvg under different device-level data heterogeneities on CIFAR-10.

In the following Fig. 3, we present the train loss and test accuracy results on MNIST, and one can make similar observations as above. In particular, by comparing Fig. 2 with Fig. 3, it seems that FedCluster is more advantageous when the data is more complex.

C. Comparison under Different Optimizers

We further compare FedCluster and the traditional federated learning with different choices of local optimizers, including 1) SGD-momentum (SGDm) with m = 0.5; 2) Adam with (β_1, β_2) = (0.9, 0.999), $\epsilon = 10^{-8}$; and 3) FedProx with μ = 0.1. The learning rate for Adam is chosen to be small enough to ensure convergence, while the other optimizers use the default learning rate. We set $\rho_{\text{device}} = 0.1, 0.5$. Fig. 4 presents the training results with different local optimizers, where the first column shows the results on CIFAR-10 and the second column shows the results on MNIST. The testing results are similar to the training results and hence are omitted. It can be seen that FedCluster significantly outperforms the traditional federated

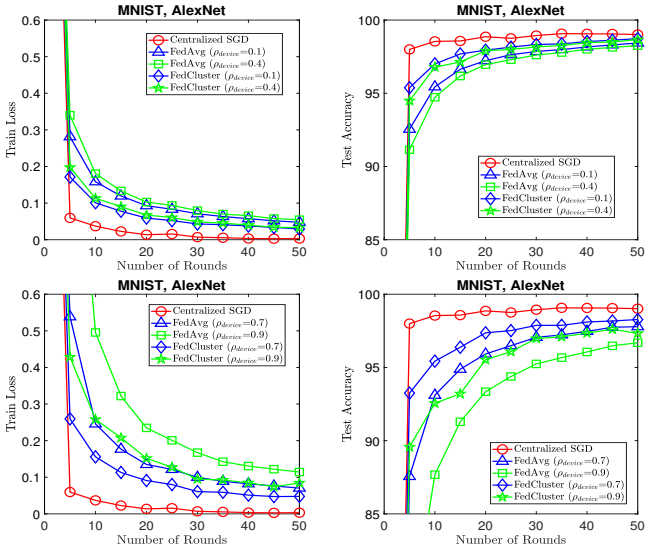

Fig. 3. Comparison between FedCluster and FedAvg under different device-level data heterogeneities on MNIST.

learning under all choices of local optimizers and all levels of device-level data heterogeneity. Again, FedCluster seems to be more advantageous when dealing with more complex datasets.

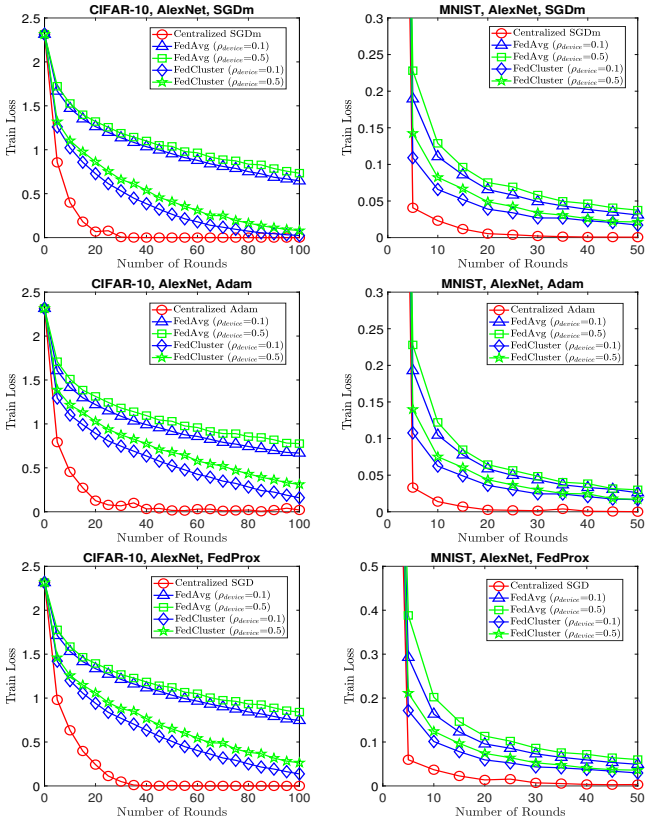

Fig. 4. Comparison between FedCluster and FedAvg under different local optimizers on CIFAR-10 (left) and MNIST (right).

D. Effect of Number of Clusters

We further explore the impact of the number of clusters on the performance of FedCluster with local SGD. We set $\rho_{\text{device}} = 0.1, 0.5$ and explore the number of clusters choices $M = 5, 10, 20$. The training results on CIFAR-10 and MNIST are presented in the first and second row of Fig. 5, respectively. From the top row, it can be seen that FedCluster outperforms FedAvg under all choices of number of clusters on the complex CIFAR-10 dataset. In particular, a larger number of clusters leads to faster convergence of FedCluster, as indicated by Theorem 1. From the bottom row, similar observations can be made on the MNIST dataset except that FedCluster with 5 clusters has comparable performance to FedAvg. In practice, the number of clusters of FedCluster is limited by the number of learning cycles allowed within a learning round.

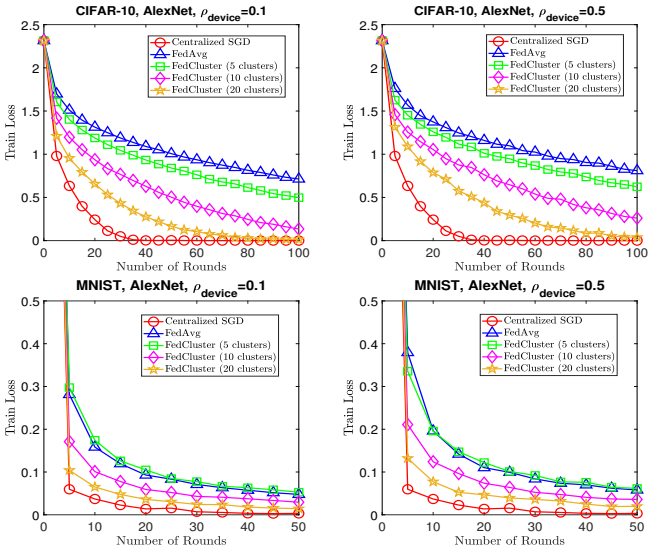

Fig. 5. Comparison between FedCluster and FedAvg under different number of clusters on CIFAR-10 (top) and MNIST (bottom).

E. Comparison under Different Cluster-level Data Heterogeneities

We introduce an additional cluster-level data heterogeneity ratio $\rho_{\text{cluster}} \in [0.1, 1]$ to specify how the devices of FedCluster are clustered. Specifically, each cluster is assigned a different major class and $\rho_{\text{cluster}} \times 100\%$ of the devices in each cluster are assigned the same major class, whereas $(1 - \rho_{\text{cluster}})/9 \times 100\%$ of the devices in the cluster are assigned a different major class. Hence, a larger ρ_{cluster} indicates a higher cluster-level data heterogeneity. We fix $\rho_{\text{device}} = 0.5$ and compare FedCluster with FedAvg under different levels of cluster-level data heterogeneity, i.e., $\rho_{\text{cluster}} = 0.1, 0.5, 0.9$. The training results on CIFAR-10 and MNIST are presented in Fig. 6. It can be seen that FedCluster achieves faster convergence than FedAvg under all levels of ρ_{cluster}. In addition, a smaller ρ_{cluster} (i.e., lower cluster-level data heterogeneity) leads to slightly faster convergence of FedCluster, which is also indicated by Theorem 1.

V. Conclusion

In this paper, we propose a novel federated learning framework named FedCluster. The FedCluster framework groups the edge devices into multiple clusters and activates them

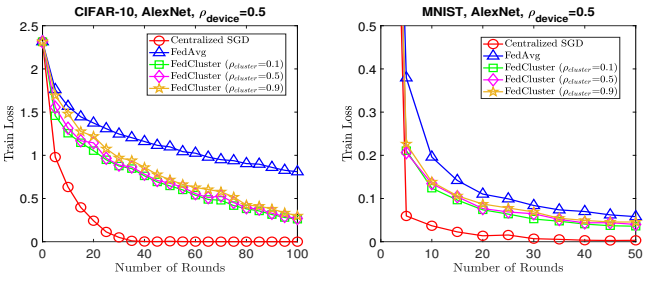

Fig. 6. Comparison between FedCluster and FedAvg under different cluster-level data heterogeneities on CIFAR-10 (left) and MNIST (right).

cyclically to perform federated learning in each learning round. We provide theoretical convergence analysis to show that FedCluster with local SGD achieves faster convergence than the conventional FedAvg algorithm in nonconvex optimization, and the convergence of FedCluster is less dependent on the device-level data heterogeneity. Our experiments on deep federated learning corroborate the theoretical findings. In the future, we expect and hope that FedCluster can be implemented in practical federated learning systems to demonstrate its fast convergence and provide great flexibility in scheduling the workload for the devices. Also, it is interesting to explore the impact of the clustering approach and the random cyclic order on the performance of FedCluster.

REFERENCES

[1] A. Agarwal, J. Langford, and C.-Y. Wei. Federated residual learning. *ArXiv:2003.12880*, 2020.

[2] A. K. R. Bayoumi, K. Mishchenko, P. Richtarik. Tighter Theory for Local SGD on Identical and Heterogeneous Data. *In Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108, pages 4519-4529, 26-28 Aug.

[3] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo. Analyzing federated learning through an adversarial lens. *In Proc. International Conference on Machine Learning (ICML)*, volume 97, pages 634–643, 09–15 Jun 2019.

[4] T. Castiglia, A. Das, S. Patterson. Multi-Level Local SGD for Heterogeneous Hierarchical Networks. *ArXiv:2007.13819*, 2020.

[5] W. Chen, K. Bhardwaj, and R. Marculescu. Fedmax: mitigating activation divergence for accurate and communication-efficient federated learning. *ArXiv:2004.03657*, 2020.

[6] Z. Chen, D. Li, M. Zhao, S. Zhang, and J. Zhu. Semi-federated learning. *IEEE Wireless Communications and Networking Conference*, 2020.

[7] Y. Deng, M. M. Kamani, and M. Mahdavi. Adaptive personalized federated learning. *ArXiv:2003.13461*, 2020.

[8] C. Dinh, N. H. Tran, M. N. Nguyen, C. S. Hong, W. Bao, A. Zomaya, and V. Gramoli. Federated learning over wireless networks: convergence analysis and resource allocation. *ArXiv:1910.13067*, 2019.

[9] C. T. Dinh, N. H. Tran, T. D. Nguyen, W. Bao, A. Y. Zomaya, B. B. Zhou. Federated Learning with Proximal Stochastic Variance Reduced Gradient Algorithms. *In Prof. International Conference on Parallel Processing (ICPP)*, pages 1-11, Aug 2020.

[10] H. Eichner, T. Koren, H. B. McMahan, N. Srebro, and K. Talwar. Semi-cyclic stochastic gradient descent. *In Proc. International Conference on Machine Learning (ICML)*, volume 97, pages 1764-1773, 09–15 Jun 2019.

[11] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *In Proc. International Conference on Machine Learning (ICML)*, volume 70, pages 1126–1135, 2017.

[12] F. Hanzely and P. Richtárik. Federated learning of a mixture of global and local models. *ArXiv:2002.05516*, 2020.

[13] R. Hu, Y. Gong, and Y. Guo. Cpfed: communication-efficient and privacy-preserving federated learning. *ArXiv:2003.13761*, 2020.

[14] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *The second Neurips Workshop on Machine Learning on the Phone and other Consumer Devices (MLPCD 2)*.

[15] R. Jin, Y. Huang, X. He, H. Dai, and T. Wu. Stochastic-sign sgd for federated learning with theoretical guarantees. *ArXiv:2002.10940*, 2020.

[16] 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:1912.04977*, 2019.

[17] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. *In Proc. International Conference on Machine Learning (ICML)*, Jun 2020.

[18] S. P. Karimireddy, M. Jaggi, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, A. T. Suresh. Mime: Mimicking Centralized Stochastic Algorithms in Federated Learning. *ArXiv:2008.03606*, 2020.

[19] A. Khaled, K. Mishchenko, and P. Richtárik. First analysis of local gd on heterogeneous data. *ArXiv:1909.04715*, 2019.

[20] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. U. Stich. A Unified Theory of Decentralized SGD with Changing Topology and Local Updates. *In Proc. International Conference on Machine Learning (ICML)*, Jun 2020.

[21] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik. Federated optimization: distributed machine learning for on-device intelligence. *ArXiv:1610.02527*, 2016.

[22] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[23] V. Kulkarni, M. Kulkarni, and A. Pant. Survey of personalization techniques for federated learning. *ArXiv:2003.08673*, 2020.

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

[25] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[26] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. *In Proc. Machine Learning and Systems (MLSys)*, pages 429–450, 2020.

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

[28] X. Li, W. Yang, S. Wang, and Z. Zhang. Communication efficient decentralized training with multiple local updates. *ArXiv:1910.09126*, 2019.

[29] X. Liang, S. Shen, J. Liu, Z. Pan, E. Chen, and Y. Cheng. Variance reduced local sgd with lower communication complexity. *ArXiv:1912.12844*, 2019.

[30] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. *In Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pages 1273–1282, 20–22 Apr 2017.

[31] M. Mohri, G. Sivek, and A. T. Suresh. Agnostic federated learning. *In Proc. International Conference on Machine Learning (ICML)*, volume 97, pages 4615–4625, 09–15 Jun 2019.

[32] A. Nagar. Privacy-preserving blockchain based federated learning with differential data sharing. *ArXiv:1912.04859*, 2019.

[33] I. Ramazanli, H. Nguyen, H. Pham, S. Reddi, and B. Poczos. Adaptive sampling distributed stochastic variance reduced gradient for heterogeneous distributed datasets. *ArXiv:2002.08528*, 2020.

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

[35] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith. On the convergence of federated optimization in heterogeneous networks. *ArXiv:1812.06127*, 2018.

[36] F. Sattler, K. R. Müller, W. Samek. Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints. *ArXiv:1910.01991*, 2019.

[37] P. Sharma, P. Khanduri, S. Bulusu, K. Rajawat, and P. K. Varshney. Parallel restarted spider—communication efficient distributed nonconvex optimization with optimal computation com- plexity. *ArXiv:1912.06036*, 2019.

[38] J. Wang, Q. Liu, H. Liang, G. Joshi, H. V. Poor. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. *ArXiv:2007.07481*, 2020.
[39] W. Wei, L. Liu, M. Loper, K.-H. Chow, M. E. Gursoy, S. Truex, and Y. Wu. A framework for evaluating gradient leakage attacks in federated learning. *ArXiv:2004.10397*, 2020.
[40] B. Woodworth, K. K. Patel, N. Srebro. Minibatch vs Local SGD for Heterogeneous Distributed Learning. *ArXiv:2006.04735*, 2020.
[41] X. Yao, T. Huang, R.-X. Zhang, R. Li, and L. Sun. Federated learning with unbiased gradient aggregation and controllable meta updating. *NeurIPS Workshop on federated learning*, 2019.
[42] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *In Proc. International Conference on Learning Representations (ICLR)*, 2017.
[43] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra. Federated learning with non-iid data. *ArXiv:1806.00582*, 2018.

APPENDIX A
SUPPORTING LEMMAS

In this section, we present and prove some supporting lemmas for the convenience of proving the main Theorem 1 in the next subsection.

Throughout the analysis, we consider a random reshuffle of the clusters. To elaborate, for every j-th round, we define a random permutation $\sigma_j : \{1,\dots,M\} \to \{1,\dots,M\}$, and the reshuffled clusters $\mathcal{S}_{\sigma_j(1)}, \dots, \mathcal{S}_{\sigma_j(M)}$ sequentially perform federated learning. For every t-th iteration of the K-th cycle in the j-th learning round, we define the following weighted average of the local models obtained by the devices in the active cluster $\mathcal{S}_{\sigma_j(K+1)}$.

$$\overline{w}_{j,K,t} := \sum_{k\in\mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} w^{(k)}_{j,K,t}. \tag{5}$$

Then, by the local update rule of the FedCluster in Algorithm 1, it holds that

$$\overline{w}_{j,K,t+1} = \overline{w}_{j,K,t} - \eta_{j,K,t} g_{j,K,t}, \tag{6}$$

where $g_{j,K,t}$ is defined as

$$g_{j,K,t} := \sum_{k\in\mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \nabla f(w^{(k)}_{j,K,t}; \xi^{(k)}_{j,K,t}). \tag{7}$$

We also define the following expectation of $g_{j,K,t}$ over the set of random variables $\xi_{j,K,t} := \{\xi^{(k)}_{j,K,t}\}_{k\in\mathcal{S}_{\sigma_j(K+1)}}$.

$$\begin{aligned}\overline{g}_{j,K,t} =& \mathbb{E}_{\xi_{j,K,t}}[g_{j,K,t}] \\ =& \sum_{k\in\mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \nabla f(w^{(k)}_{j,K,t}; \mathcal{D}^{(k)}).\end{aligned}$$

Lemma 1. *Let Assumption 2 hold. Then, $g_{j,K,t}$ and $\overline{g}_{j,K,t}$ satisfy*

$$\mathbb{E}\|\overline{g}_{j,K,t} - g_{j,K,t}\|^2 \le \sum_{K=1}^{M} q_K^{-1} \sum_{k\in\mathcal{S}_K} p_k^2 s_k^2 \tag{8}$$

Proof.

$$\begin{aligned}&\mathbb{E}\|\overline{g}_{j,K,t} - g_{j,K,t}\|^2 \\ &= \mathbb{E}\Big\| \sum_{k\in\mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \\ &\quad [\nabla f(w^{(k)}_{j,K,t}; \xi^{(k)}_{j,K,t}) - \nabla f(w^{(k)}_{j,K,t}; \mathcal{D}^{(k)})]\Big\|^2 \qquad (9) \\ &\overset{(i)}{=} \mathbb{E}\Big[\sum_{k\in\mathcal{S}_{\sigma_j(K+1)}} \frac{p_k^2}{q^2_{\sigma_j(K+1)}} \\ &\quad \mathbb{E}_{\xi_{j,K,t}} \Big\| \nabla f(w^{(k)}_{j,K,t}; \xi^{(k)}_{j,K,t}) - \nabla f(w^{(k)}_{j,K,t}; \mathcal{D}^{(k)}) \Big\|^2 \Big] \\ &\overset{(ii)}{\le} \mathbb{E}\Big(\sum_{k\in\mathcal{S}_{\sigma_j(K+1)}} \frac{p_k^2 s_k^2}{q^2_{\sigma_j(K+1)}} \Big) \\ &= \sum_{K=1}^{M} q_K \sum_{k\in\mathcal{S}_K} \frac{p_k^2 s_k^2}{q_K^2} \\ &= \sum_{K=1}^{M} q_K^{-1} \sum_{k\in\mathcal{S}_K} p_k^2 s_k^2,\end{aligned}$$

where (i) uses the facts that $\mathbb{E}_{\xi^{(k)}_{j,K,t}}[\nabla f(w^{(k)}_{j,K,t}; \xi^{(k)}_{j,K,t}) - \nabla f(w^{(k)}_{j,K,t}; \mathcal{D}^{(k)})] = 0$ and that $\sigma_j(K+1)$ and all the samples in $\{\xi^{(k)}_{j,K,t}\}_{k\in\mathcal{S}_{\sigma_j(K+1)}}$ are independent, and (ii) follows from item 2 of Assumption 2. □

Lemma 2. *Suppose the learning rate $\eta_{j,K,t}$ is non-increasing with regard to $(jM+K)E+t$. Then, for any j, K, t it holds that*

$$\begin{aligned}&\mathbb{E}\|\overline{w}_{j,K,t} - W_{jM}\|^2 \\ &\le \mathbb{E}\Big[\sum_{k\in\mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \Big\| w^{(k)}_{j,K,t} - W_{jM} \Big\|^2 \Big] \\ &\le \eta^2_{j,0,0} E^2 M^2 G^2 \qquad (10)\end{aligned}$$

Proof. The first inequality in (10) follows from Jensen's inequality applied to $\|\cdot\|^2$ and the fact that $\overline{w}_{j,K,t} = \sum_{k\in\mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} w^{(k)}_{j,K,t}$. Next, we prove the second inequality.

Define $\xi_j := \{\xi^{(k)}_{j,K,t} : K = 0,1,\dots,M-1; t = 0,1,\dots,E-1; k \in \mathcal{S}_{\sigma_j(K+1)}\}$. Then,

$$\begin{aligned}&\mathbb{E}_{\xi_j}\|w^{(k)}_{j,K,t} - W_{jM}\|^2 \\ &\le \mathbb{E}_{\xi_j} \Big\| w^{(k)}_{j,K,t} - W_{jM+K} + \sum_{K'=0}^{K-1} (W_{jM+K'+1} - W_{jM+K'}) \Big\|^2 \\ &\overset{(i)}{\le} (K+1)\mathbb{E}_{\xi_j}\|w^{(k)}_{j,K,t} - w^{(k)}_{j,K,0}\|^2 \\ &\quad + (K+1)\sum_{K'=0}^{K-1} \mathbb{E}_{\xi_j}\|\overline{w}_{j,K',E} - \overline{w}_{j,K',0}\|^2 \\ &\overset{(ii)}{\le} M\mathbb{E}_{\xi_j}\Big\| \sum_{s=0}^{t-1} \eta_{j,K,s} \nabla f(w_{j,K,s}; \xi^{(k)}_{j,K,s}) \Big\|^2 \\ &\quad + M\sum_{K'=0}^{K-1} \mathbb{E}_{\xi_j}\Big\| \sum_{s=0}^{E-1} \eta_{j,K',s} g_{j,K',s} \Big\|^2 \\ &\overset{(iii)}{\le} M\Big(\sum_{s=0}^{t-1} \eta_{j,K,s}\Big) \sum_{s=0}^{t-1} \eta_{j,K,s} \mathbb{E}_{\xi_j} \|\nabla f(w_{j,K,s}; \xi^{(k)}_{j,K,s})\|^2\end{aligned}$$

$$
\begin{aligned}
&+ M \sum_{K'=0}^{K-1} \Big(\sum_{s'=0}^{E-1} \eta_{j,K',s'} \Big) \sum_{s=0}^{E-1} \eta_{j,K',s} \mathbb{E}_{\xi_j} \|g_{j,K',s}\|^2 \\
&\overset{(iv)}{\leq} \eta_{j,0,0}^2 G^2 \Big(Mt^2 + MKE^2 \Big) \\
&\overset{(v)}{\leq} \eta_{j,0,0}^2 E^2 M^2 G^2,
\end{aligned}
$$

where (i) applies Jensen's inequality to $\|\cdot\|^2$ and uses the fact that $W_{jM+K'+1} = \overline{w}_{j,K',E} = \overline{w}_{j,K'+1,0} = w_{j,K'+1,0}^{(k)}$ ($0 \leq K' \leq M-2$, $k \in \mathcal{S}_{\sigma_j(K'+1)}$) based on the update rule of Algorithm 1, (ii) uses (6) and $K \leq M-1$, (iii) applies Jensen's inequality, (iv) uses item 3 of Assumption 2, the inequality (11) below and the fact that $\eta_{j,K,s} \leq \eta_{j,0,0}$ for $K, s \geq 0$, and (v) uses the fact that $K \leq M-1$ and $0 \leq t \leq E-1$.

$$
\begin{aligned}
&\mathbb{E}_{\xi_j} \|g_{j,K,t}\|^2 \\
&= \mathbb{E}_{\xi_j} \Big\| \sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \nabla f(w_{j,K,t}^{(k)}; \xi_{j,K,t}^{(k)}) \Big\|^2 \\
&\leq \mathbb{E}_{\xi_j} \Big[\sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \Big\| \nabla f(w_{j,K,t}^{(k)}; \xi_{j,K,t}^{(k)}) \Big\|^2 \Big] \\
&\leq \mathbb{E}_{\xi_j} \Big[\sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} G^2 \Big] \leq G^2. \qquad (11)
\end{aligned}
$$

□

Appendix B
Proof of Theorem 1

By the L-smoothness of the objective function f, we obtain that

$$
\begin{aligned}
&\mathbb{E}[f(\overline{w}_{j,K,t+1}) - f(\overline{w}_{j,K,t})] \\
&\leq \mathbb{E}\langle \nabla f(\overline{w}_{j,K,t}), \overline{w}_{j,K,t+1} - \overline{w}_{j,K,t} \rangle \\
&\quad + \frac{L}{2} \mathbb{E} \|\overline{w}_{j,K,t+1} - \overline{w}_{j,K,t}\|^2 \\
&\overset{(i)}{=} \mathbb{E}\Big\langle \nabla f(\overline{w}_{j,K,t}), \\
&\quad - \eta_{j,K,t} \sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \nabla f(w_{j,K,t}^{(k)}; \xi_{j,K,t}^{(k)}) \Big\rangle \\
&\quad + \frac{L}{2} \mathbb{E} \Big\| \eta_{j,K,t} \sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \nabla f(w_{j,K,t}^{(k)}; \xi_{j,K,t}^{(k)}) \Big\|^2 \\
&\overset{(ii)}{\leq} -\eta_{j,K,t} \mathbb{E}\Big\langle \nabla f(\overline{w}_{j,K,t}), \\
&\quad \sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \nabla f(w_{j,K,t}^{(k)}; \mathcal{D}^{(k)}) \Big\rangle \\
&\quad + 2L\eta_{j,K,t}^2 \mathbb{E} \Big\| \nabla f(W_{jM}) \Big\|^2 \\
&\quad + 2L\eta_{j,K,t}^2 \mathbb{E} \Big\| \sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \nabla f(W_{jM}; \mathcal{D}^{(k)}) \\
&\quad - \nabla f(W_{jM}) \Big\|^2 + 2L\eta_{j,K,t}^2 \mathbb{E} \Big\| \sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \\
&\quad [\nabla f(w_{j,K,t}^{(k)}; \mathcal{D}^{(k)}) - \nabla f(W_{jM}; \mathcal{D}^{(k)})] \Big\|^2 \\
&\quad + 2L\eta_{j,K,t}^2 \mathbb{E} \Big\| \sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \\
&\quad [\nabla f(w_{j,K,t}^{(k)}; \xi_{j,K,t}^{(k)}) - \nabla f(w_{j,K,t}^{(k)}; \mathcal{D}^{(k)})] \Big\|^2 \\
&\overset{(iii)}{\leq} -\eta_{j,K,t} \mathbb{E}\Big\langle \nabla f(\overline{w}_{j,K,t}), \\
&\quad \sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} [\nabla f(w_{j,K,t}^{(k)}; \mathcal{D}^{(k)}) - \nabla f(W_{jM}; \mathcal{D}^{(k)})] \Big\rangle \\
&\quad - \eta_{j,K,t} \mathbb{E}\Big\langle \nabla f(\overline{w}_{j,K,t}) - \nabla f(W_{jM}), \\
&\quad \sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \nabla f(W_{jM}; \mathcal{D}^{(k)}) \Big\rangle \\
&\quad - \eta_{j,K,t} \mathbb{E}\Big\langle \nabla f(W_{jM}), \sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \nabla f(W_{jM}; \mathcal{D}^{(k)}) \Big\rangle \\
&\quad + 2L\eta_{j,K,t}^2 \mathbb{E} \Big\| \nabla f(W_{jM}) \Big\|^2 \\
&\quad + 2L\eta_{j,K,t}^2 \mathbb{E} \Big\| \nabla f(W_{jM}; \mathcal{D}^{(\mathcal{S}_{\sigma_j(K+1)})}) - \nabla f(W_{jM}) \Big\|^2 \\
&\quad + 2L\eta_{j,K,t}^2 \mathbb{E} \Big[\sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \\
&\quad \Big\| \nabla f(w_{j,K,t}^{(k)}; \mathcal{D}^{(k)}) - \nabla f(W_{jM}; \mathcal{D}^{(k)}) \Big\|^2 \Big] \\
&\quad + 2L\eta_{j,K,t}^2 \sum_{K=1}^{M} q_K^{-1} \sum_{k \in \mathcal{S}_K} p_k^2 s_k^2 \\
&\overset{(iv)}{\leq} \eta_{j,K,t} \mathbb{E}\Big\{ \Big\| \nabla f(\overline{w}_{j,K,t}) \Big\| \Big\| \sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \\
&\quad [\nabla f(w_{j,K,t}^{(k)}; \mathcal{D}^{(k)}) - \nabla f(W_{jM}; \mathcal{D}^{(k)})] \Big\| \Big\} \\
&\quad + \eta_{j,K,t} \mathbb{E}\Big\{ \Big\| \nabla f(\overline{w}_{j,K,t}) - \nabla f(W_{jM}) \Big\| \\
&\quad \Big\| \nabla f(W_{jM}; \mathcal{D}^{(\mathcal{S}_{\sigma_j(K+1)})}) \Big\| \Big\} \\
&\quad - \eta_{j,K,t} \mathbb{E} \Big\| \nabla f(W_{jM}) \Big\|^2 + 2L\eta_{j,K,t}^2 \mathbb{E} \Big\| \nabla f(W_{jM}) \Big\|^2 \\
&\quad + 2L\eta_{j,K,t}^2 H_{\text{cluster}} \\
&\quad + 2L^3 \eta_{j,K,t}^2 \mathbb{E} \Big[\sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \Big\| w_{j,K,t}^{(k)} - W_{jM} \Big\|^2 \Big] \\
&\quad + 2L\eta_{j,K,t}^2 \sum_{K=1}^{M} q_K^{-1} \sum_{k \in \mathcal{S}_K} p_k^2 s_k^2 \\
&\overset{(v)}{\leq} \eta_{j,K,t} G \mathbb{E}\Big\{ \sum_{k \in \mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \\
&\quad \Big\| \nabla f(w_{j,K,t}^{(k)}; \mathcal{D}^{(k)}) - \nabla f(W_{jM}; \mathcal{D}^{(k)}) \Big\| \Big\} \\
&\quad + \eta_{j,K,t} LG \mathbb{E} \Big\| \overline{w}_{j,K,t} - W_{jM} \Big\| \\
&\quad - \eta_{j,K,t} (1 - 2L\eta_{j,K,t}) \mathbb{E} \Big\| \nabla f(W_{jM}) \Big\|^2 \\
&\quad + 2L\eta_{j,K,t}^2 H_{\text{cluster}} + 2L^3 \eta_{j,K,t}^2 \eta_{j,0,0}^2 E^2 M^2 G^2
\end{aligned}
$$

$$
\begin{aligned}
&+ 2L\eta_{j,K,t}^2 \sum_{K=1}^{M} q_K^{-1} \sum_{k\in\mathcal{S}_K} p_k^2 s_k^2 \\
&\overset{(vi)}{\leq} \eta_{j,K,t} LG \sqrt{\mathbb{E}\Big[\sum_{k\in\mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \Big\| w_{j,K,t}^{(k)} - W_{jM} \Big\|^2\Big]} \\
&+ \eta_{j,K,t} LG \sqrt{\mathbb{E}\Big\| \overline{w}_{j,K,t} - W_{jM} \Big\|^2} - \frac{\eta_{j,K,t}}{2} \mathbb{E}\Big\| \nabla f(W_{jM}) \Big\|^2 \\
&+ 2L\eta_{j,K,t}^2 H_{\text{cluster}} + 2L^3 \eta_{j,K,t}^2 \eta_{j,0,0}^2 E^2 M^2 G^2 \\
&+ 2L\eta_{j,K,t}^2 \sum_{K=1}^{M} q_K^{-1} \sum_{k\in\mathcal{S}_K} p_k^2 s_k^2 \\
&\overset{(vii)}{\leq} 2\eta_{j,K,t} LG \sqrt{\eta_{j,0,0}^2 E^2 M^2 G^2} - \frac{\eta_{j,K,t}}{2} \mathbb{E}\Big\| \nabla f(W_{jM}) \Big\|^2 \\
&+ 2L\eta_{j,K,t}^2 H_{\text{cluster}} + 2L^3 \eta_{j,K,t}^2 \eta_{j,0,0}^2 E^2 M^2 G^2 \\
&+ 2L\eta_{j,K,t}^2 \sum_{K=1}^{M} q_K^{-1} \sum_{k\in\mathcal{S}_K} p_k^2 s_k^2 \\
&\overset{(viii)}{=} \Big(\frac{2L}{T} + \frac{2L^3}{T^2}\Big) G^2 - \frac{1}{2\sqrt{TME}} \mathbb{E}\Big\| \nabla f(W_{jM}) \Big\|^2 \\
&+ \frac{2L}{TME} \Big(H_{\text{cluster}} + \sum_{K=1}^{M} q_K^{-1} \sum_{k\in\mathcal{S}_K} p_k^2 s_k^2 \Big),
\end{aligned} \tag{12}
$$

where (i) uses (6) & (7), (ii) uses the equality that $\mathbb{E}_{\xi_{j,K,t}} \nabla f(w_{j,K,t}^{(k)}; \xi_{j,K,t}^{(k)}) = \nabla f(w_{j,K,t}^{(k)}; \mathcal{D}^{(k)})$ and the inequality that $\| \sum_{k=1}^m x_k \|^2 \leq m \sum_{k=1}^m \|x_k\|^2$, (iii) applies Jensen's inequality to $\| \cdot \|^2$ and uses (8) & (9), (iv) uses Cauchy-Schwartz inequality, the heterogeneity definition in (2), the L-smoothness of $f(\cdot; \mathcal{D}^{(k)})$, the equality that $\mathbb{E}_{\sigma_j(K+1)} \Big[\sum_{k\in\mathcal{S}_{\sigma_j(K+1)}} \frac{p_k}{q_{\sigma_j(K+1)}} \nabla f(W_{jM}; \mathcal{D}^{(k)}) \Big] = \nabla f(W_{jM})$ and the fact that W_{jM} is independent of $\sigma_j(K+1)$, (v) uses 1 & 10, the L-smoothness of f and $W_{jM} = \overline{w}_{j,0,0}$ and applies Jensen's inequality to $\| \cdot \|^2$, (vi) uses the L-smoothness of $f(\cdot; \mathcal{D}^{(k)})$, $\eta_{j,K,t} \equiv (TME)^{-\frac{1}{2}} \leq \frac{1}{4L}$ (since $T \geq \frac{16L^2}{EM}$) and the inequality that $\mathbb{E}\|X\| \leq \sqrt{\mathbb{E}\|X\|^2}$ for any random vector X, (vii) uses (10) and (viii) substitutes $\eta_{j,K,t} \equiv (TME)^{-\frac{1}{2}}$ into this equation.

Note that $W_{jM+K'+1} = \overline{w}_{j,K',E} = \overline{w}_{j,K'+1,0}$ ($0 \leq K' \leq M-2$), and $W_{(j+1)M} = \overline{w}_{j,M-1,E} = \overline{w}_{j+1,0,0}$. Hence, by telescoping (12) over $j = 0, 1, \ldots, T-1$, $K = 0, 1, \ldots, M-1$, $t = 0, 1, \ldots, E-1$, it holds that

$$
\begin{aligned}
&\mathbb{E}[f(W_{TM}) - f(W_0)] \\
&\leq 2LMEG^2 \Big(1 + \frac{L^2}{T}\Big) - \frac{1}{2} \sqrt{\frac{ME}{T}} \sum_{j=0}^{T-1} \mathbb{E}\Big\| \nabla f(W_{jM}) \Big\|^2 \\
&\quad + 2L \Big(H_{\text{cluster}} + \sum_{K=1}^{M} q_K^{-1} \sum_{k\in\mathcal{S}_K} p_k^2 s_k^2 \Big),
\end{aligned}
$$

which by using $f(W_{TM}) \geq \inf_w f(w)$ and the assumption that $T \geq L^2$ further implies

$$
\frac{1}{T} \sum_{j=0}^{T-1} \mathbb{E}\Big\| \nabla f(W_{jM}) \Big\|^2 \leq \frac{C}{\sqrt{TME}} + C_2 \sqrt{\frac{ME}{T}} \leq \frac{2C}{\sqrt{TME}}, \tag{13}
$$

where C is defined in (4), $C_2 = 8LG^2$, and the second inequality uses the assumption that $ME \leq C/C_2 = \frac{C}{8LG^2}$. Therefore, to achieve an ϵ-stationary point, we need

$$
\frac{2C}{\sqrt{TME}} \leq \epsilon, \tag{14}
$$

which implies that

$$
T \geq \frac{4C^2\epsilon^{-2}}{ME} \propto \frac{C^2}{ME}. \tag{15}
$$