This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3216837

Scalable and Low-Latency Federated Learning with Cooperative Mobile Edge Networking

Zhenxiao Zhang, *Student Member, IEEE,* Zhidong Gao, *Student Member, IEEE,* Yuanxiong Guo, *Senior Member, IEEE,* and Yanmin Gong, *Senior Member, IEEE*

Abstract—Federated learning (FL) enables collaborative model training without centralizing data. However, the traditional FL framework is cloud-based and suffers from high communication latency. On the other hand, the edge-based FL framework that relies on an edge server co-located with mobile base station for model aggregation has low communication latency but suffers from degraded model accuracy due to the limited coverage of edge server. In light of high-accuracy but high-latency cloud-based FL and low-latency but low-accuracy edge-based FL, this paper proposes a new FL framework based on cooperative mobile edge networking called cooperative federated edge learning (CFEL) to enable both high-accuracy and low-latency distributed intelligence at mobile edge networks. Considering the unique two-tier network architecture of CFEL, a novel federated optimization method dubbed cooperative edge-based federated averaging (CE-FedAvg) is further developed, wherein each edge server both coordinates collaborative model training among the devices within its own coverage and cooperates with other edge servers to learn a shared global model through decentralized consensus. Experimental results based on benchmark datasets show that CFEL can largely speed up the convergence speed and reduce the training time to achieve a target model accuracy compared with prior FL frameworks.

Index Terms—Federated learning, mobile edge networks, decentralized optimization, training latency, scalability.

1 Introduction

The proliferation of edge devices such as smartphones and Internet-of-things (IoT) devices, each equipped with rich sensing, computation, and storage resources, leads to tremendous data being generated on a daily basis at the network edge. At the same time, artificial intelligence (AI) and machine learning (ML) are advancing rapidly and enable efficient knowledge extraction from large volumes of data. The convergence of 5G networks and AI/ML leads to many emerging applications with significant economic and societal impacts such as autonomous driving [1], augmented reality [2], real-time video analytics [3], mobile healthcare [4], and smart manufacturing [5]. A salient feature of these emerging domains is the large and continuously streaming data that these applications generate, which must be processed efficiently enough to support real-time learning and decision making based on these data.

The standard ML paradigm requires centralizing the data at the cloud, which involves large amounts of distributed data transferred from the network edge to the cloud with high communication cost and privacy risk. An alternative paradigm is *Federated Learning (FL)*, which enables edge devices to collaboratively learn a shared prediction model under the orchestration of the cloud while keeping all the personal data that may contain private information on device [6]. Compared with the traditional centralized ML, FL is capable of reducing communication cost, improving latency, and enhancing data privacy while obtaining an accurate shared learning model for on-device inference, and therefore has received significant attention recently [7].

Despite of its great potential, FL faces a major bottleneck in communication efficiency. Specifically, in the current cloud-based FL framework, edge devices need to repeatedly download the global model from the remote cloud and upload local model updates of large data size (e.g., million of parameters for modern DNN models) to the cloud for many times in order to learn an accurate shared model. Although communication compression techniques such as quantization and sparsification [8], [9], [10] have been developed to improve the communication efficiency of FL, due to the long-distance and limited-bandwidth transmissions between an edge device and the remote cloud, the model training in cloud-based FL is inevitably slow and fails to meet the latency requirements of delay-sensitive intelligent applications.

As more computing and storage resources are being deployed at the mobile network edge in 5G-and-beyond networks, the edge-based FL framework, where an edge server co-located with mobile base station serves as the aggregator to coordinate FL among its proximate edge devices, is gaining popularity [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. Although this framework can speed up model training by mitigating the cloud bottleneck and saving long-distance data transmission, an edge server can only access a limited number of edge devices and their collected data. As the ML model performance highly depends on the data volume, edge-based FL cannot meet the accuracy requirements of AI-powered applications that could be safety-critical such as autonomous driving and mobile healthcare. To address the limited coverage issue of edge-based FL framework, hierarchical FL framework [21] that relies on the cloud to

- *Z. Zhang, Z. Gao, and Y. Gong are with the Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX, 78249. Y. Guo is with the Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX, 78249. E-mail: {zhidong.gao@my., zhenxiao.zhang@my., yuanxiong.guo@, yanmin.gong@}utsa.edu.*

- *Z. Zhang and Z. Gao both contributed equally to this work.*

coordinate multiple edge servers has been proposed, but it still suffers from high communication latency with the cloud.

In light of the high-accuracy but high-latency cloud-based FL and low-latency but low-accuracy edge-based FL, this paper proposes a new FL framework called cooperative federated edge learning (CFEL) to achieve both high-accuracy and low-latency over wireless edge networks. The key idea of CFEL is to leverage a network of cooperative edge servers located at the wireless edge, rather than relying on a central cloud server or multiple independent edge servers, to facilitate FL among large numbers of edge devices distributed over a wide area. By eliminating the costly communication with the cloud, CFEL can achieve lower model training latency than cloud-based FL, and by tapping into more data from a larger set of edge devices, CFEL can obtain higher model accuracy than edge-based FL. Moreover, due to the distributed system nature of CFEL, there does not exist a single bottleneck, making the framework more scalable than previous frameworks. Although promising, CFEL contains multiple cooperative aggregators rather than a single aggregator as assumed in prior FL frameworks, making the classic federated averaging (FedAvg) algorithm [6] not directly applicable. To address that, we further design an efficient federated optimization method for CFEL dubbed cooperative edge-based federated averaging (CE-FedAvg), wherein each edge server first obtains an edge model from the set of edge devices associated to it using FedAvg and then cooperates with other edge servers to learn a shared global model through decentralized consensus.

In summary, the main contributions of this paper are as follows:

- We propose CFEL, a novel FL framework at mobile edge networks, to achieve both high-accuracy and low-latency model training based on cooperative mobile edge networking. CFEL is more scalable than prior FL frameworks by exploiting multiple aggregators and eliminating a single point of failure.
- Considering the unique network architecture of CFEL, we design a new federated optimization method named CE-FedAvg that can learn a shared global model efficiently over the collective dataset of all edge devices under the orchestration of a distributed network of cooperative edge servers.
- We prove the convergence of CE-FedAvg theoretically and derive its convergence rates under general assumptions about the loss function, data distribution, and network topology. The obtained convergence guarantees are tighter than those in literature and provide new insights about the algorithm design.
- We conduct extensive experiments based on common FL benchmark datasets and demonstrate that CFEL can learn an accurate model within a shorter time than other FL frameworks at mobile edge networks.

2 Related Works

FL at mobile edge networks suffers from high training latency due to limited communication bandwidth. To address this issue, various communication-efficient distributed learning algorithms have been proposed to improve the communication efficiency of FL. Specifically, McMahan et al. [6] proposed FedAvg to reduce the number of communication rounds by running multiple steps of SGD update on devices before aggregating their updates at the server to compute the new model. Various communication compression techniques such as sparsification [10] and quantization [26] were also designed to reduce the size of messages transmitted between the server and devices in each communication round of FL. Considering the resource constraints of mobile edge networks, learning and resource allocation were jointly optimized in [11], [14], [15], [17], [19], [20] to minimize the training latency of FedAvg at mobile edge networks. All of the aforementioned studies assume a single server that aggregates model updates from all devices in each communication round. However, since the coverage of a single edge server is inherently limited, the proposed solutions cannot scale to a large number of devices.

A few recent studies [21], [22], [23], [24], [25] have considered multiple edge servers for FL at mobile edge networks, each responsible for aggregating model updates from a subset of devices. In particular, hierarchical FL and the associated hierarchical federated averaging (Hier-FAvg) optimization algorithm were developed in [21], [22] that relies on a central entity (e.g., the cloud) to coordinate multiple edge servers in a star topology. As the central entity can become the bottleneck and suffer from a single point of failure, the fault-tolerance and scalability of hierarchical FL is still a concern. Alternatively, decentralized coordination among edge servers without relying on a central entity like the setting of CFEL has been considered in [23], [24], [25]. Castiglia et al. [24] proposed Multi-Level Local SGD (MLL-SGD) in a two-tier communication network with heterogeneous workers, but it only considers the IID data distribution. Zhong et al. [23] proposed a similar algorithm called P-FedAvg, but it only consider the convex model, and the global and local model aggregations operate at the same frequency. The concurrent work [25] is mostly related to ours, but as elaborated later, our convergence result is much tighter than theirs and gives new insights on why frequent local model aggregation helps and which system design works better. A detailed comparison between our algorithm and prior algorithms under the same system setting is summarized in Table 1.

TABLE 1: Comparison of algorithms in multi-server FL setting.

Algorithm	non-IID	non-convex	fault tolerance	local aggregation benefit
Hier-FAvg [21]	✓	✓	✗	✗
Hier-FAvg [22]	✓	✓	✗	✓
P-FedAvg [23]	✓	✗	✓	✗
MLL-SGD [24]	✗	✓	✓	✓
SE-FEEL [25]	✓	✓	✓	✗
Ours	✓	✓	✓	✓

3 System Model and Problem Formulation

Consider a CFEL system depicted in Fig. 1. Assume a set of m clusters in the system. Each cluster $i \in [m]$ contains

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3216837

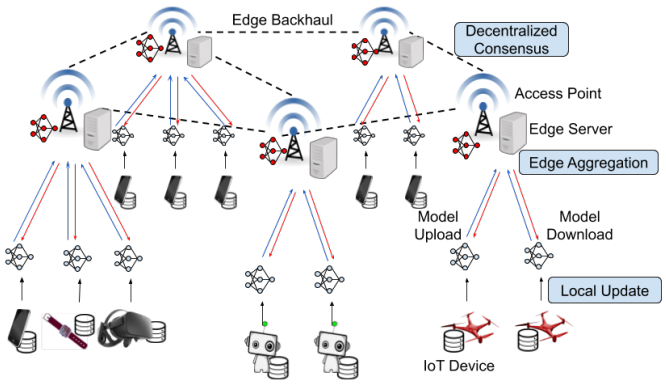


Fig. 1: CFEL: Cooperative Federated Edge Learning.

a single edge server co-located with the base station and a set of devices $\mathcal{S}_i$ with $n_i = |\mathcal{S}_i|$. Devices in $\mathcal{S}_i$ only communicate with the server in the same cluster using the device-edge links. Define the set of all devices in the system as $\mathcal{S} = \cup_{i=1}^{m} \mathcal{S}_i$, and the total number of devices $n = |\mathcal{S}|$. The edge servers communicate with each other over the edge backhaul. The communication pattern of edge backhual is represented as an undirected and connected graph $\mathcal{G} = \{V, E\}$, where V denotes the set of of all edge servers, and each edge in the graph $(i, j) \in E$ denotes the link between edge servers i and j. Let $\mathcal{N}_i = \{j : (i, j) \in E\}$ be the set of neighbors of server i in the graph $\mathcal{G}$. A list of main notations used in the paper is summarized in Table 2. Furthermore, let $\|\cdot\|, \|\cdot\|_{\mathrm{F}}$ and $\|\cdot\|_{\mathrm{op}}$ denote the ℓ_2 vector norm, Frobenius norm and matrix operator norm, respectively.

The goal of FL is to find a global model $\mathbf{x} \in \mathbb{R}^d$ that solves the following optimization problem:

$$\min_{\mathbf{x}} F(\mathbf{x}) = \frac{1}{n} \sum_{k=1}^{n} F_k(\mathbf{x}), \tag{1}$$

where $F_k(\mathbf{x}) = \mathbb{E}_{z \sim \mathcal{D}_k}[\ell_k(\mathbf{x}; z)]$ is the local objective function of device k, and $\mathcal{D}_k$ is the data distribution of device k. Here ℓ_k is the loss function defined by the learning task, and z represents a data sample from distribution $\mathcal{D}_k$.

To solve (1) while satisfying the communication constraints of CFEL, we decompose the problem into multiple subproblems, each for a cluster. The local objective function of the i-th cluster is defined as

$$\min_{\mathbf{x}} f_i(\mathbf{x}) = \frac{1}{n_i} \sum_{k \in \mathcal{S}_i} F_k(\mathbf{x}), \tag{2}$$

which represents the average loss over all devices in cluster i. Then the global objective function (1) can be rewritten as:

$$\min_{\mathbf{x}} F(\mathbf{x}) = \sum_{i=1}^{m} \frac{n_i}{n} f_i(\mathbf{x}). \tag{3}$$

In CFEL, devices in the systems collaboratively solve the above optimization problem under the coordination of the edge servers in their clusters without sharing the raw data.

4 Learning Algorithm Design for CFEL

Since the CFEL system in Fig. 1 contains multiple clusters without a central aggregator, the classic FedAvg algorithm is not directly applicable. In this section, we propose a new federated optimization method called *Cooperative Edge-based Federated Averaging* (CE-FedAvg) to efficiently solve (3).

TABLE 2: Summary of main notations.

Notation	Definition
i, j	Index for cluster
k, k'	Index for device
l	Index for global round
r	Index for edge round
s	Index for local iteration
t	Index for global iteration
n	Total number of devices
m	Total number of edge servers/clusters
$[m]$	$\{1, 2, \ldots, m\}$
$\mathcal{S}_i$	Set of devices in cluster i
$\mathcal{S}$	Set of all devices
n_i	Number of devices in cluster i
$\mathcal{G}$	Communication graph for edge backhaul
$\mathbf{y}_{l,r}^{(i)}$	Edge model of cluster i
$\mathcal{D}_k$	Data distribution of device k
$F_k(\cdot)$	Local objective function of device k
$f_i(\cdot)$	Local objective function of cluster i
$\mathbf{x}_{l,r,s}^{(k)}$	Local model of device k
$\mathbf{g}_k$	Stochastic gradient of device k
η	Local learning rate
τ	Intra-cluster aggregation period
$q\tau$	Inter-cluster aggregation period
$\mathcal{N}_i$	Set of neighbors of edge server i
$\mathbf{H}$	Mixing matrix
π	Number of gossip steps per round
ζ	Second largest eigenvalue of $\mathbf{H}$
σ^2	Bounded variance
ϵ^2	Inter-cluster divergence
ϵ_i^2	Intra-cluster divergence of cluster i
i_k	Cluster index of device k

4.1 Algorithm Description

Algorithm 1 describes our proposed CE-FedAvg algorithm for CFEL. The overall training process of CE-FedAvg is divided into multiple global rounds wherein each cluster first performs q edge rounds of intra-cluster collaboration independently and then communicates with other clusters for inter-cluster collaboration.

At the beginning of the r-th edge round in the l-th global round (i.e., (r, l)-th round), the edge server in each cluster i first broadcasts its current edge model $\mathbf{y}_{l,r}^{(i)}$ to the associated devices $\mathcal{S}_i$ under its coverage in the system. Then, devices in each cluster initialize their local models to be the received edge model and run τ iterations of SGD to update their local models in parallel. Let $\mathbf{x}_{l,r,s}^{(k)}$ denote the local model of device k at the s-th local iteration of (r, l)-th round. We have the following update equations for each device $k \in \mathcal{S}_i$:

$$\mathbf{x}_{l,r,0}^{(k)} \leftarrow \mathbf{y}_{l,r}^{(i)}, \tag{4}$$

$$\mathbf{x}_{l,r,s+1}^{(k)} \leftarrow \mathbf{x}_{l,r,s}^{(k)} - \eta \mathbf{g}_k(\mathbf{x}_{l,r,s}^{(k)}), \forall s = 0, \ldots, \tau - 1, \tag{5}$$

where η is the local learning rate, and $\mathbf{g}_k(\mathbf{x}_{l,r,s}^{(k)})$ is the stochastic gradient computed over a mini-batch θ_k sampled from the local data distribution $\mathcal{D}_k$. Next, their final updated local models $\{\mathbf{x}_{l,r,\tau}^{(k)}, \forall k \in S_i\}$ are sent to the edge server i for intra-cluster model aggregation, and each edge server

Algorithm 1 Proposed CE-FedAvg Algorithm.

Initialization: initial edge models $\mathbf{y}_{0,0}^{(i)}$, $\forall i \in [m]$, edge backhaul graph $\mathcal{G}$, mixing matrix $\mathbf{H} \in [0,1]^{m\times m}$, intra-cluster aggregation period τ, inter-cluster aggregation period $q\tau$, and number of gossip steps π.
for each global round $l = 0, \ldots, p-1$ **do**
  for each cluster $i \in [m]$ **in parallel do**
    for each edge round $r = 0, \ldots, q-1$ **do**
      for each device $k \in \mathcal{S}_i$ **in parallel do**
        $\mathbf{x}_{l,r,0}^{(k)} \leftarrow \mathbf{y}_{l,r}^{(i)}$
        for $s = 0, \ldots, \tau-1$ **do**
          Compute a stochastic gradient $\mathbf{g}_k$ over a mini-batch θ_k sampled from $\mathcal{D}_k$
          $\mathbf{x}_{l,r,s+1}^{(k)} \leftarrow \mathbf{x}_{l,r,s}^{(k)} - \eta \mathbf{g}_k(\mathbf{x}_{l,r,s}^{(k)})$
        end for
      end for
      $\mathbf{y}_{l,r+1}^{(i)} \leftarrow \frac{1}{n_i}\sum_{k\in\mathcal{S}_i} \mathbf{x}_{l,r,\tau}^{(k)}$
    end for
    $\mathbf{y}_{l+1,0}^{(i)} \leftarrow \sum_{j\in\{i\}\cup\mathcal{N}_i} \mathbf{H}_{j,i}^{\pi}\mathbf{y}_{l,q}^{(j)}$
  end for
end for

$i \in [m]$ updates its edge model $\mathbf{y}_{l,r+1}^{(i)}$ by averaging the received local models from all associated devices as follows:

$$\mathbf{y}_{l,r+1}^{(i)} \leftarrow \frac{1}{n_i}\sum_{k\in\mathcal{S}_i} \mathbf{x}_{l,r,\tau}^{(k)}. \tag{6}$$

Then, the same procedure repeats in the next edge round $r+1$.

After q edge rounds, the edge servers communicate with each other over the edge backhaul for inter-cluster model aggregation by averaging their models with neighboring servers in π times using gossip protocol as follows:

$$\mathbf{y}_{l+1,0}^{(i)} \leftarrow \sum_{j\in\{i\}\cup\mathcal{N}_i} \mathbf{H}_{j,i}^{\pi}\mathbf{y}_{l,q}^{(j)}. \tag{7}$$

Here $\mathcal{N}_i = \{j : (j,i) \in E\}$ denotes the neighbors of server i in the graph $\mathcal{G}$, and $\mathbf{H} \in [0,1]^{m\times m}$ denotes the mixing matrix with each element $\mathbf{H}_{j,i}$ being the weight assigned by server i to server j. Note that $\mathbf{H}_{j,i} > 0$ only if servers i and j are directly connected in the edge backhaul. Finally, the algorithm goes to the next global round $l+1$ until p global rounds in total.

Notably, CE-FedAvg inherits the privacy benefits of classic FL schemes by keeping the original data on device and sharing only model parameters. Furthermore, CE-FedAvg is compatible with existing privacy-preserving techniques in FL such as secure aggregation [27], [28], differential privacy [29], [30], [31], and shuffling [32] since only the sum rather than individual values is needed for the intra-cluster and inter-cluster model aggregations.

4.2 Runtime Analysis of CE-FedAvg

We now present a runtime analysis of CE-FedAvg. Here, the communication time of downloading models from the edge server by each device is ignored because the download bandwidth is usually much larger than upload bandwidth for the device-to-edge communication in practice [7]. Similarly, the computation time for model aggregation at edge servers is ignored because the involved computation workload is rather small compared to the computation capabilities of edge servers.

In each global round of CE-FedAvg, the total delay consists of the computation time for performing $q\tau$ steps of SGD update, the communication time for performing q rounds of intra-cluster model aggregation, and the communication time for performing one round of inter-cluster model aggregation consisting of π steps of gossip averaging. Therefore, the total runtime of CE-FedAvg after p global rounds can be estimated as

$$p \times \left[\max_k \frac{q\tau C}{c_k} + \frac{qW}{b_{d2e}} + \frac{\pi W}{b_{e2e}}\right], \tag{8}$$

where C is the computation workload of performing one step of SGD update, c_k is the processing capability of device k, W is the model size, b_{d2e} is the uplink bandwidth from device to edge server, and b_{e2e} is the bandwidth between two connecting edge servers in the backhaul.

4.3 Prior Algorithms as Special Cases

When the topology of edge backhaul $\mathcal{G}$ is fully connected and the edge models from all servers are averaged in each global aggregation round, CE-FedAvg essentially reduces to Hier-FAvg [22] with the same model update rule. Also, when there exists only one cluster, and all devices send their local models to a single edge server for model aggregation after τ local iterations (i.e., $m = 1$). CE-FedAvg reduces to FedAvg [6]. Moreover, when each cluster only contains one edge device, and each device communicates with its neighboring device after $q\tau$ iterations (i.e., $n = m$), CE-FedAvg reduces to decentralized local SGD [33]. Therefore, the existing algorithms can be viewed as special cases of CE-FedAvg. However, due to the generality of CE-FedAvg, its convergence analysis presents significant new challenges. As one of the main contributions in this paper, the convergence analysis of CE-FedAvg will be elaborated in the next section.

5 Convergence Analysis of CE-FedAvg

In this section, we first describe the convergence results of CE-FedAvg with respect to the gradient norm of the objective function $F(\cdot)$ and compare CE-FedAvg with prior learning algorithms. Then we analyze the impact of various learning parameters on the convergence rates of CE-FedAvg.

5.1 Assumptions

Before stating our results, we make the following assumptions to facilitate our convergence analysis.

Assumption 1 (Smoothness)**.** *Each local objective function $F_k : \mathbb{R}^d \to \mathbb{R}$ is L-smooth for all $k \in \mathcal{S}$, i.e.,*

$$\|\nabla F_k(\mathbf{x}) - \nabla F_k(\mathbf{x}')\| \le L\|\mathbf{x} - \mathbf{x}'\|,\ \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d.$$

Assumption 2 (Unbiased Gradient and Bounded Variance)**.** *The local mini-batch stochastic gradient is an unbiased estimator of the local gradient: $\mathbb{E}_{\theta_k}[\mathbf{g}_k(\mathbf{x})] = \nabla F_k(\mathbf{x})$ and has bounded variance: $\mathbb{E}_{\theta_k}[\|\mathbf{g}_k(\mathbf{x}) - \nabla F_k(\mathbf{x})\|^2] \le \sigma^2, \forall \mathbf{x} \in \mathbb{R}^d, k \in \mathcal{S}$.*

Assumption 3 (Lower Bounded)**.** *There exists a constant $F_{\inf}$ such that*

$$F(\mathbf{x}) \geq F_{\inf}, \forall \mathbf{x} \in \mathbb{R}^d.$$

Assumption 4 (Mixing Matrix)**.** *The graph $\mathcal{G} := (V, E)$ is strongly connected and the mixing matrix $\mathbf{H} \in [0,1]^{m \times m}$ defined on it satisfies the following:*

1) *If $(i,j) \in E$, then $\mathbf{H}_{i,j} > 0$; otherwise, $\mathbf{H}_{i,j} = 0$.*
2) *$\mathbf{H}$ is doubly stochastic, i.e., $\mathbf{H}^\intercal = \mathbf{H}$.*
3) *The magnitudes of all eigenvalues except the largest one are strictly less than 1, i.e., $\zeta = \max\{|\lambda_2(\mathbf{H})|, |\lambda_n(\mathbf{H})|\} < \lambda_1(\mathbf{H}) = 1$*

Assumption 5 (Bounded Intra-Cluster Divergence)**.** *For each cluster $i \in V$, there exists a constant $\epsilon_i \geq 0$ such that $\forall \mathbf{x} \in \mathbb{R}^d$,*

$$\frac{1}{n_i} \sum_{k \in \mathcal{S}_i} \|\nabla f_i(\mathbf{x}) - \nabla F_k(\mathbf{x})\|^2 \leq \epsilon_i^2.$$

If the local objective functions of edge devices are identical to each other within a cluster, then we have $\epsilon_i^2 = 0$.

Assumption 6 (Bounded Inter-Cluster Divergence)**.** *There exists a constant $\epsilon \geq 0$ such that $\forall \mathbf{x} \in \mathbb{R}^d$,*

$$\sum_{i=1}^{m} \frac{n_i}{n} \|\nabla f_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \epsilon^2.$$

If the local objective functions of clusters are identical to each other, then we have $\epsilon^2 = 0$.

Assumptions 1, 2, and 3 are standard in the analysis of SGD [34]. Assumption 4 follows the decentralized optimization literature [35] and ensures that the gossip step converges to the average of all the vectors shared between the nodes in the graph $\mathcal{G}$. Here, smaller ζ indicates better connectivity between edge servers. For example, for complete graphs and bipartite graphs, $\zeta = 0$ and $\zeta = 1$, respectively. Assumptions 5 and 6 capture the dissimilarities of local objectives within a single and across different clusters due to data heterogeneity, respectively.

Note that most prior work in literature [25], [33], [36] uses the following global divergence assumption to capture the data heterogeneity:

Assumption 7 (Bounded Global Divergence)**.** *There exists a constant $\hat{\epsilon} \geq 0$ such that $\forall \mathbf{x} \in \mathbb{R}^d$,*

$$\frac{1}{n} \sum_{k=1}^{n} \|\nabla F_k(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \hat{\epsilon}^2.$$

To see the relationship between Assumptions 5–6 and Assumption 7, we can split the global divergence into the intra-cluster and inter-cluster divergences as follows:

$$\frac{1}{n} \sum_{k=1}^{n} \|\nabla F_k(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 = \sum_{i=1}^{m} \frac{n_i}{n} \|\nabla f_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 + \sum_{i=1}^{m} \frac{n_i}{n} \times \frac{1}{n_i} \sum_{k \in \mathcal{S}_i} \|\nabla f_i(\mathbf{x}) - \nabla F_k(\mathbf{x})\|^2. \quad (9)$$

As discussed later in Section 5.5, by decomposing the global divergence bound into two components, our assumptions enable a tighter convergence analysis for CE-FedAvg to capture the benefit of local aggregation in accelerating convergence.

5.2 Update Rule for CE-FedAvg Algorithm

Since edge servers are essentially stateless in CE-FedAvg, we focus on how device models evolve in the convergence analysis. We define $t = lq\tau + r\tau + s$, where $l \in [0, p-1]$, $r \in [0, q-1]$ and $s \in [0, \tau - 1]$, as the global iteration index, and $T = pq\tau$ as the total number of global training iterations in Algorithm 1. Then we can rewritten the local model $\mathbf{x}_{l,r,s}^{(k)}$ as $\mathbf{x}_t^{(k)}$. Without loss of generality, we denote the range of device indices for cluster $i \in [m]$ as $\left[\sum_{j \leq i-1} n_j + 1, \sum_{j \leq i} n_j\right]$ with $n_0 = 0$.

The system behavior of CE-FedAvg can be summarized by the following update rule for device models:

$$\mathbf{X}_{t+1} = (\mathbf{X}_t - \eta \mathbf{G}_t)\mathbf{W}_t, \quad (10)$$

where $\mathbf{X}_t = [\mathbf{x}_t^{(1)}, \ldots, \mathbf{x}_t^{(n)}] \in \mathbb{R}^{d \times n}$, $\mathbf{G}_t = [\mathbf{g}_1(\mathbf{x}_t^{(1)}), \ldots, \mathbf{g}_n(\mathbf{x}_t^{(n)})] \in \mathbb{R}^{d \times n}$, and $\mathbf{W}_t \in \mathbb{R}^{n \times n}$ is a time-varying operator capturing the three stages in CE-FedAvg: SGD update, intra-cluster model aggregation, and inter-cluster model aggregation. Specifically, $\mathbf{W}_t$ is defined as follows:

$$\mathbf{W}_t = \begin{cases} \mathbf{B}^\intercal \text{diag}(\mathbf{c}) \mathbf{H}^\pi \mathbf{B}, & (t+1) \bmod q\tau = 0 \\ \mathbf{B}^\intercal \text{diag}(\mathbf{c}) \mathbf{B}, & (t+1) \bmod \tau = 0 \\ & \text{and } (t+1) \bmod q\tau \neq 0 \\ \mathbf{I}_{n \times n}, & \text{otherwise,} \end{cases} \quad (11)$$

where $\mathbf{B} \in \{0,1\}^{m \times n}$ is a binary matrix with each element $\mathbf{B}_{i,k}$ denoting if device k belongs to cluster i (i.e., $\mathbf{B}_{i,k} = 1$) or not (i.e., $\mathbf{B}_{i,k} = 0$), $\mathbf{c} = [1/n_1, \ldots, 1/n_m] \in \mathbb{R}^m$, and $\text{diag}(\mathbf{c}) \in \mathbb{R}^{m \times m}$ is a diagonal matrix with the elements of vector $\mathbf{c}$ on the main diagonal. Specifically, for the stage of SGD update (i.e., $(t+1) \bmod \tau \neq 0$), $\mathbf{W}_t$ is the identity matrix because there is no communication between edge devices after SGD update; for the stage of intra-cluster model aggregation (i.e., $(t+1) \bmod \tau = 0$ and $(t+1) \bmod q\tau \neq 0$), $\mathbf{B}^\intercal \text{diag}(\mathbf{c}) \mathbf{B}$ captures the model averaging within each cluster independently after SGD update; and for the stage of inter-cluster model aggregation (i.e., $(t+1) \bmod q\tau = 0$), $\mathbf{B}^\intercal \text{diag}(\mathbf{c}) \mathbf{H}^\pi \mathbf{B}$ captures the model aggregation within each cluster followed by π steps of gossip averaging across clusters.

To facilitate the convergence analysis, we first introduce the quantities of interests. Multiplying $\mathbf{1}_n/n$ on both sides in (10), we get

$$\mathbf{X}_{t+1} \frac{\mathbf{1}_n}{n} = \mathbf{X}_t \frac{\mathbf{1}_n}{n} - \eta \mathbf{G}_t \frac{\mathbf{1}_n}{n}, \quad (12)$$

where $\mathbf{W}_t$ disappears due to the fact that $\mathbf{1}_n/n$ is a right eigenvector of $\mathbf{B}^\intercal \text{diag}(\mathbf{c}) \mathbf{H}^\pi \mathbf{B}$ and $\mathbf{B}^\intercal \text{diag}(\mathbf{c}) \mathbf{B}$ with eigenvalue of 1. Then define the average model as

$$\mathbf{u}_t = \mathbf{X}_t \frac{\mathbf{1}_n}{n}. \quad (13)$$

After rearranging, one can obtain

$$\mathbf{u}_{t+1} = \mathbf{u}_t - \frac{\eta}{n} \sum_{k=1}^{n} \mathbf{g}_k(\mathbf{x}_t^{(k)}). \quad (14)$$

Note that the averaged local model $\mathbf{u}_t$ is updated via performing the perturbed SGD contributed by all devices. In the following, we will focus on the convergence of the averaged model $\mathbf{u}_t$, which is a common practice in distributed optimization literature [37].

5.3 Convergence Results

We now provide the main theoretical results of the paper in Theorem 1 and Corollary 1. We only provide the proof sketch here and include the detailed proofs in the appendices. Define the following constants:

$$\Omega_1 = \frac{\zeta^{2\pi}}{1-\zeta^{2\pi}}, \quad \Omega_2 = \frac{1}{1-\zeta^{2\pi}} + \frac{2}{1-\zeta^{\pi}} + \frac{\zeta^{\pi}}{(1-\zeta^{\pi})^2}, \tag{15}$$

and $n \times n$ matrix $\mathbf{A} = \mathbf{1}_n \mathbf{1}_n^\intercal / n$. For the sake of presentation, we use $\mathbf{V}$ to denote $\mathbf{B}^\intercal \mathrm{diag}(\mathbf{c}) \mathbf{B}$ in the following.

Lemma 1 (Convergence Decomposition)**.** *Under Assumptions 1, 2, and 3, if the learning rate $\eta \leq \frac{1}{L}$, the iterates of Algorithm 1 satisfy:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(\mathbf{u}_t)\|^2 \leq \underbrace{\frac{2(F(\mathbf{x}_1)-F_{\inf})}{\eta T} + \frac{\eta L \sigma^2}{n}}_{\textit{fully sync SGD}} + \underbrace{\frac{2L^2}{nT}\Big(\sum_{t=0}^{T-1}\mathbb{E}\|\mathbf{X}_t(\mathbf{V}-\mathbf{A})\|_{\mathrm{F}}^2 + \sum_{t=0}^{T-1}\mathbb{E}\|\mathbf{X}_t(\mathbf{I}-\mathbf{A})\|_{\mathrm{F}}^2\Big)}_{\textit{residual error}}.$$

Proof: The proof is provided in Appendix D in the supplementary text. □

Lemma 1 aims to provide the composition of the total convergence error bound. The *residual error* provides hints on how to derive the convergence properties of CE-FedAvg. Specifically, the first term $\|\mathbf{X}_t(\mathbf{V}-\mathbf{A})\|_{\mathrm{F}}^2$ represents the inter-cluster error between the global average model $\mathbf{X}_t\mathbf{A}$ and the edge server models $\mathbf{X}_t\mathbf{V}$. The second term $\|\mathbf{X}_t(\mathbf{I}-\mathbf{A})\|_{\mathrm{F}}^2$ represents the intra-cluster error between the device models $\mathbf{X}_t\mathbf{I}$ and the edge server models $\mathbf{X}_t\mathbf{V}$. Next, we will provide the upper bounds for these two terms.

Lemma 2 (Bounded inter-cluster error)**.** *Under Assumptions 1, 2, 4 and 6, the iterates of Algorithm 1 satisfy:*

$$\frac{1}{nT}\sum_{t=0}^{T-1}\mathbb{E}\|\mathbf{X}_t(\mathbf{V}-\mathbf{A})\|_{\mathrm{F}}^2 \leq \frac{2\eta^2(\Omega_1 q\tau + \frac{m-1}{n}q\tau)\sigma^2}{1-4\eta^2L^2q^2\tau^2\Omega_2} + \frac{4\eta^2q^2\tau^2\Omega_2}{1-4\eta^2L^2q^2\tau^2\Omega_2}\left(\epsilon^2 + \frac{L^2}{nT}\sum_{t=0}^{T-1}\mathbb{E}\|\mathbf{X}_t(\mathbf{V}-\mathbf{I})\|_{\mathrm{F}}^2\right).$$

Proof: The proof is provided in Appendix E in the supplementary text. □

Lemma 2 shows that the inter-cluster error contains the intra-cluster error. Next, we will bound the intra-cluster error.

Lemma 3 (Bounded intra-cluster error)**.** *Under Assumptions 1, 2, 4, 5, the iterates of Algorithm 1 satisfy:*

$$\frac{1}{nT}\sum_{t=0}^{T-1}\mathbb{E}\|\mathbf{X}_t(\mathbf{I}-\mathbf{V})\|_{\mathrm{F}}^2 \leq \frac{(\frac{n-m}{n})\eta^2\tau\sigma^2}{1-2\eta^2L^2\tau^2} + \frac{2\eta^2\tau^2\sum_{i=1}^m \frac{n_i}{n}\epsilon_i^2}{1-2\eta^2L^2\tau^2}.$$

Proof: The proof is provided in Appendix F in the supplementary text. □

Lemma 3 gives the upper bound of intra-cluster error. Combining Lemmas 1, 2 and 3 and choosing a proper learning rate, we can derive the following convergence bound:

Theorem 1 (Convergence of CE-FedAvg)**.** *Let Assumptions 1–6 hold, and let Ω_1, Ω_2, L, σ, ϵ, ϵ_i be as defined therein. If the learning rate satisfies*

$$\eta \leq \min\{\frac{1}{2L\tau}, \frac{1}{2\sqrt{2\Omega_2}Lq\tau}\}, \tag{16}$$

then for any $T > 0$, the iterates of Algorithm 1 for CE-FedAvg satisfy

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(\mathbf{u}_t)\|^2 \leq \frac{2(F(\mathbf{x}_1)-F_{\inf})}{\eta T} + \frac{\eta L\sigma^2}{n} + 8\eta^2L^2(\Omega_1 q\tau + \frac{m-1}{n}q\tau)\sigma^2 + 16\eta^2L^2q^2\tau^2\Omega_2\epsilon^2 + 8\frac{n-m}{n}\eta^2L^2\tau\sigma^2 + 16L^2\eta^2\tau^2\sum_{i=1}^m\frac{n_i}{n}\epsilon_i^2. \tag{17}$$

Proof: Substituting the results in Lemmas 2 and 3 into Lemma 1, we have:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(\mathbf{u}_t)\|^2 \leq \frac{2(F(\mathbf{x}_1)-F_{\inf})}{\eta T} + \frac{\eta L\sigma^2}{n} + \frac{4\eta^2L^2(\Omega_1 q\tau + \frac{m-1}{n}q\tau)\sigma^2}{1-4\eta^2L^2q^2\tau^2\Omega_2} + \frac{8\eta^2L^2q^2\tau^2\Omega_2\epsilon^2}{1-4\eta^2L^2q^2\tau^2\Omega_2} + \left(\frac{4\eta^2L^2q^2\tau^2\Omega_2}{1-4\eta^2L^2q^2\tau^2\Omega_2}+1\right)\left(\frac{2(\frac{n-m}{n})\eta^2L^2\tau\sigma^2}{1-2\eta^2L^2\tau^2} + \frac{4\eta^2L^2\tau^2\sum_{i=1}^m\frac{n_i}{n}\epsilon_j^2}{1-2\eta^2L^2\tau^2}\right). \tag{18}$$

When $\eta \leq \min\{\frac{1}{2L\tau}, \frac{1}{2\sqrt{2\Omega_2}Lq\tau}\}$, we have:

$$\frac{1}{1-4\eta^2L^2q^2\tau^2\Omega_2} \leq 2, \tag{19}$$

$$\frac{4\eta^2L^2q^2\tau^2\Omega_2}{1-4\eta^2L^2q^2\tau^2\Omega_2} \leq 1, \quad \frac{1}{1-2\eta^2L^2\tau^2} \leq 2. \tag{20}$$

Putting (19) and (20) into (18), we arrive at the conclusion. □

Further, by setting the learning rate to be $\eta = \frac{1}{L}\sqrt{\frac{n}{T}}$, we can obtain the following corollary:

Corollary 1. *For CE-FedAvg, under Assumptions 1-6, if the learning rate is $\eta = \frac{1}{L}\sqrt{\frac{n}{T}}$ when $T > 4\tau^2 n\max\{1, 2\Omega_2 q^2\}$, then*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(\mathbf{u}_t)\|^2 \leq O(\frac{1}{\sqrt{T}}) + O(\frac{q\tau+\tau}{T}) + O(\frac{q^2\tau^2+\tau^2}{T}).$$

Corollary 1 provides some notable insights. First, the last two terms show the trade-off between communication cost and convergence. While smaller communication periods q and τ speed up the convergence and reduce the convergence error, they also increase the overall communication cost. Second, the error increases w.r.t. the magnitude of $q^2\tau^2$. Thus, the convergence rate of $O(1/\sqrt{T})$ can be guaranteed by ensuring the total iteration number satisfies $T > q^4\tau^4$.

5.4 Comparison of Iteration Complexity

In the following, we consider the extreme cases of CE-FedAvg and show that our analysis recovers the results of prior algorithms that can be treated as special cases of CE-FedAvg.

- **Comparison to Hier-FAvg.** When the topology of edge backhaul is fully connected, the value of ζ becomes 0, and the model update rule of CE-FedAvg is essentially the same as that of Hier-FAvg. Therefore, our convergence result in Theorem 1 reduce to those of Hier-FAvg in [22]. Meanwhile, Theorem 1 shows that the fully connected network topology gives the fastest convergence speed in terms of iteration complexity among all the connected topologies because it has the smallest values of Ω_1 and Ω_2.
- **Comparison to FedAvg.** When $m = 1$ and $q = 1$, all devices communicate with a single edge server after τ local iterations and $\epsilon = 0$. In this case, the proposed CE-FedAvg algorithm reduces to FedAvg, and the iteration complexity of CE-FedAvg reduces to $O(\frac{1}{\eta T}) + O(\frac{\eta\sigma^2}{n}) + O(\eta^2\tau\sigma^2) + O(\eta^2\tau^2\epsilon_i^2)$. This coincides with the complexity of FedAvg given in [36].
- **Comparison to Decentralized Local SGD.** When $n = m$ and $\tau = 1$, each edge server only coordinates one device and communicates with neighboring servers after q iterations and $\epsilon_i = 0$. The proposed CE-FedAvg algorithm reduces to decentralized local SGD, and the iteration complexity of CE-FedAvg reduces to $O(\frac{1}{\eta T}) + O(\frac{\eta\sigma^2}{n}) + O(\eta^2 q\sigma^2) + O(\eta^2 q^2\epsilon^2)$. This coincides with the complexity of decentralized local SGD given in [33].

5.5 Discussions

In the following, we compare our main results with prior work and analyze the impacts of cluster-level data distribution and cluster size on algorithmic convergence in CE-FedAvg.

Remark 1 (Comparison with SE-FEEL). We compare our convergence result with that of a concurrent work SE-FEEL [25] that analyzes CE-FedAvg only under the global divergence assumption 7. Specifically, [25] provides a convergence rate of

$$O(\frac{1}{\eta T}) + O(\frac{\eta\sigma^2}{n}) + O(\eta^2 q\tau\sigma^2) + O(\eta^2 q^2\tau^2\hat{\epsilon}^2).$$

According to the above result, q and τ have the same effect on the convergence bound, which cannot show any benefit of intra-cluster model aggregation. In comparison, our work shows a much tighter convergence rate of

$$O(\frac{1}{\eta T}) + O(\frac{\eta\sigma^2}{n}) + O(\eta^2\tau(q+1)\sigma^2) + O(\eta^2\tau^2(q^2\epsilon^2 + \sum_{i=1}^{m}\epsilon_i^2)).$$

We can observe that both intra-cluster aggregation period τ and inter-cluster aggregation period $q\tau$ affect the convergence bound. In particular, given a fixed inter-cluster aggregation period $q\tau$, more frequent intra-cluster aggregation (i.e., a smaller τ) leads to faster convergence and lower convergence error. This clearly shows the benefit of intra-cluster model aggregation in CE-FedAvg.

Remark 2 (Effect of cluster size). We analyze the impact of cluster size on the convergence of CE-FedAvg under a fixed number of devices n. For the IID setting (i.e., $\epsilon^2 = \epsilon_i^2 = 0, \forall i$), the terms containing m in (17) is

$$\frac{4\eta^2 L^2\sigma^2\tau(2q-1)}{n}m. \tag{21}$$

As $4\eta^2 L^2\sigma^2\tau(2q-1)/n$ is always positive, a smaller value of m leads to a lower convergence error bound. For the non-IID setting, cluster size m can also affect the inter-cluster and intra-cluster divergences (i.e., Assumptions 5 and 6). For simplicity, assume all clusters have the same number of devices, i.e., $n_i = n/m, \forall i \in [m]$. We have the following lemma:

Lemma 4. *Under equal cluster sizes, the inter-cluster divergence in Assumption 6 can be written as:*

$$\frac{1}{m}\sum_{i=1}^{m}\|\nabla f_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 = \frac{m}{n^2}\sum_{i=1}^{m}\|\sum_{k\in\mathcal{V}_i}\nabla F_k(\mathbf{x})\|^2 - \frac{1}{n^2}\|\sum_{k=1}^{n}\nabla F_k(\mathbf{x})\|^2. \tag{22}$$

Proof: The proof is provided in Appendix G in the supplementary text. □

Suppose we combine any $\rho > 1$ existing clusters (assume the cluster index $i = 1, \ldots, \rho$ without loss of generality) into a new cluster. According to the Cauchy–Schwarz inequality, we have

$$\sum_{i=1}^{\rho}\left\|\sum_{k\in\mathcal{V}_i}\nabla F_k(\mathbf{x})\right\|^2 \geq \frac{1}{\rho}\left\|\sum_{k\in\bigcup_{i=1}^{\rho}\mathcal{V}_i}\nabla F_k(\mathbf{x})\right\|^2. \tag{23}$$

Therefore, by Lemma 4, given the same set of devices and random grouping, it is easy to see that in the RHS of (22), the first term decreases as m decreases while the second term remains the same. Therefore, the inter-cluster divergence decreases as m decreases, corresponding to faster convergence in CE-FedAvg.

Remark 3 (Effect of cluster-level data distribution). We investigate the impact of cluster-level data distribution (IID and non-IID) on the convergence of CE-FedAvg. According to (9) which shows that the global divergence can be decomposed into the inter-cluster and intra-cluster divergences, we can obtain the following:

$$\hat{\epsilon}^2 = \epsilon^2 + \sum_{i=1}^{m}\frac{n_i}{n}\epsilon_i^2. \tag{24}$$

Therefore, given certain data distributions on devices (i.e., the global divergence $\hat{\epsilon}^2$ is fixed), decreasing the inter-cluster divergence ϵ^2 will increase the intra-cluster divergence $\sum_{i=1}^{m} n_i\epsilon_i^2/n$. According to (17) of Theorem 1, since $16\eta^2 L^2 q^2\tau^2\Omega_2 > 8\eta^2 L^2\tau^2$, this will lead to a lower total convergence bound. In particular, when the cluster-level data distribution is IID (i.e., $\epsilon^2 = 0$), the convergence bound in Theorem 1 is the smallest, and CE-FedAvg converges at the fastest speed.

6 Experiments

6.1 Experimental Setup

We consider a CFEL system with 64 devices and 8 edge servers. Each edge server is connected with 8 devices, and

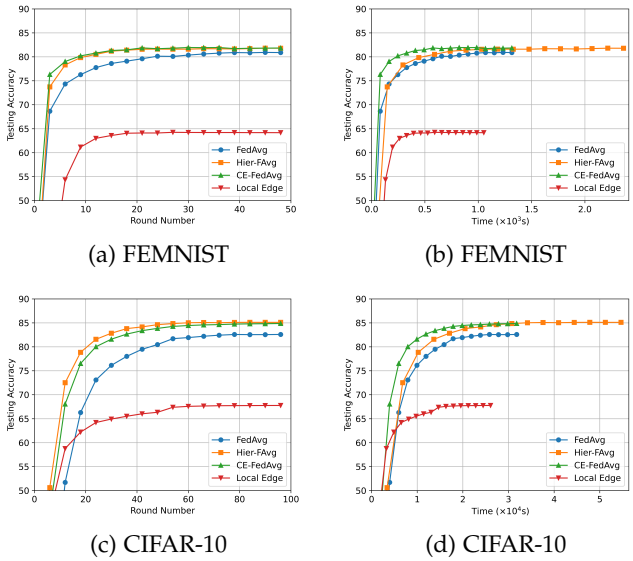
This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3216837

edge servers are interconnected in edge backhaul with a ring topology. In the experiments, we consider the image classification task on two common FL datasets: FEMNIST [38] and CIFAR-10 [39]. The FEMNIST dataset is the federated splitting version of EMNIST dataset which includes 3,550 users. We randomly sample 64 users to simulate the non-IID data distribution for experimentation. Each user's local data is divided into 90% and 10% for training and testing, respectively. The common testing dataset is composed of the testing data from all devices. The model trained on FEMNIST is a CNN with two 3×3 convolutional layers (each with 32 channels and ReLu activation followed with 2×2 max pooling), a full connected layer with 1024 units and ReLu activation, and a final softmax output layer (6,603,710 total parameters) [40]. The CIFAR-10 dataset contains 50,000 training images and 10,000 testing images. To simulate the non-IID data distribution, by default, the 50,000 training images are partitioned across devices following the Dirichlet distribution [41] with concentration parameter of 0.5. We train a modified VGG-11 (9,750,922 total parameters) on CIFAR-10. The original 10,000 testing images are used as the common testing dataset. For each dataset, the common testing set is used to evaluate the generalization performances of the trained models.

For CE-FedAvg, we set the mixing matrix $\mathbf{H}$ following Assumption 4 and the number of gossip steps in each global aggregation round $\pi = 10$ by default. To demonstrate the effectiveness of CE-FedAvg, we compare it with three baselines: FedAvg [6], Hier-FAvg [21] and Local-Edge. For fair comparison, the baseline algorithms are adapted as follows:

- *FedAvg:* In every global round, each device performs $q\tau$ iterations of SGD update and uploads its updated model to the cloud for global aggregation. This corresponds to the traditional cloud-based FL framework.
- *Hier-FAvg:* In every global round, each device first alternatively performs τ iterations of SGD update and uploads its updated model to the associated edge server for local aggregation for $q-1$ times. Next, each device performs τ iterations of SGD update and uploads its updated model to the cloud for global aggregation. This corresponds to the hierarchical FL framework.
- *Local-Edge:* In every global round, each device alternatively performs τ iterations of SGD update and uploads its updated model to the associated edge server for local aggregation for q times without collaboration between edge servers. This corresponds to the edge-based FL framework.

For all experiments, we use mini-batch SGD with momentum of 0.9 to train the localmodel with batch size of 50. The learning rate of each algorithm is tuned from $\{0.01, 0.05, 0.1\}$ for CIAFR-10 and from $\{0.1, 0.06, 0.03, 0.01\}$ for FEMNIST using grid search. Following the implementation in [42], instead of doing τ local training steps per device, we perform τ epochs of training over each device's dataset. Moreover, to account for varying numbers of gradient steps per device, we weight the average of device models by each device's local sample size. We run each experiment with 5 random seeds and report the average. All algorithms are implemented using Pytorch on an Ubuntu server with 4 NVIDIA RTX 8000 GPUs.



(a) FEMNIST (b) FEMNIST

(c) CIFAR-10 (d) CIFAR-10

Fig. 2: Convergence rate and runtime comparisons of CE-FedAvg and the baseline algorithms when $\tau = 2$ and $q = 8$ for FEMNIST and CIFAR-10 datasets. (a) and (c) show how the accuracy changes over global round; (b) and (d) show how the accuracy changes over runtime.

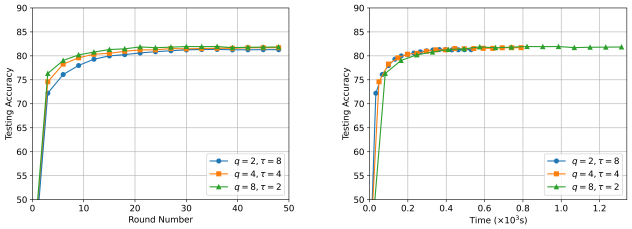
We estimate the total training time as the sum of computing time and communication time. We use thop[1] to measure the computation workload in terms of the number of floating point operations (FLOPs). The number of FLOPs needed for each training sample per iteration is 920.67 MFLOPs for VGG-11 on CIFAR-10 and 13.30 MFLOPs for CNN on FEMNIST, respectively. The edge devices are assumed to be iPhone X whose processing capacity is 691.2 GFLOPS. Following [25], we assume the edge servers are connected in a ring topology via high-speed links with bandwidth of 50 Mbps unless otherwise specified. The devices and edge servers are connected via wireless links whose bandwidth is 10 Mbps per device, and the bandwidth between each device and the cloud is set to be 1 Mbps [43].

6.2 Experimental Results

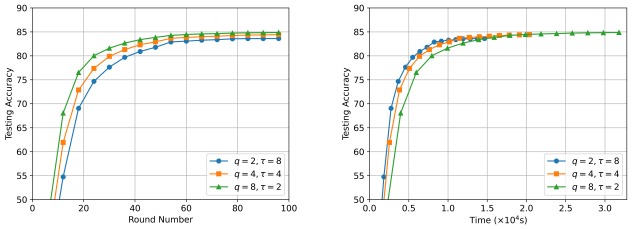
We first compare the convergence speed and runtime of CE-FedAvg and the baseline algorithms while fixing $\tau = 2$ and $q = 8$. For CE-FedAvg and Local-Edge, we measure the average test accuracy of edge models in each global round, and for FedAvg and Hier-FAvg, we measure the test accuracy of cloud model in each global round. Fig. 2 shows the convergence process. From the figure, we can observe that in terms of global round number, Hier-FAvg generally converges faster than CE-FedAvg by aggregating all local models centrally in the global model aggregation stage. Both Hier-FAvg and CE-FedAvg converge faster than FedAvg by using local model aggregation before global model aggregation. Furthermore, Local-Edge converges to a much lower model accuracy because a smaller amount of

1. https://pypi.org/project/thop/

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2022.3216837

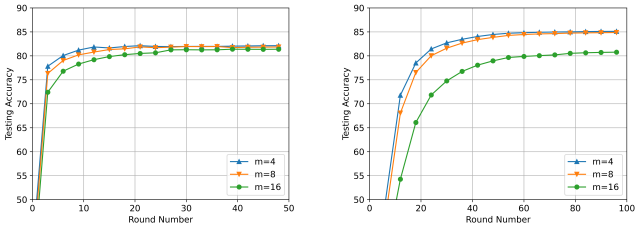


(a) FEMNIST (b) FEMNIST



(c) CIFAR-10 (d) CIFAR-10

Fig. 3: Convergence rate and runtime comparisons of CE-FedAvg for CIFAR-10 and FEMNIST datasets under different τ when $q\tau = 16$.



(a) FEMNIST (b) CIFAR-10

Fig. 4: Testing accuracy vs. round number of CE-FedAvg under different cluster number m for CIFAR-10 and FEMNIST datasets when fixing $n = 64$, $\tau = 2$, and $q = 8$.

data is used to train each edge model. On the other hand, in terms of runtime, CE-FedAvg can achieve a better time-to-accuracy than all baseline algorithms by leveraging a distributed network of cooperative edge servers to perform fast local and global model aggregations. Specifically, On FEMNIST the runtime of CE-FedAvg necessary to achieve a target test accuracy of 80% is 62.5% and 58.3% less than that of FedAvg and Hier-FedAvg, respectively. On CIFAR-10, the runtime of CE-FedAvg necessary to achieve a target test accuracy of 80% is 50.0% and 41.8% less than that of FedAvg and Hier-FedAvg, respectively.

Next, we vary τ from $\{2, 4, 8\}$ while fixing $q\tau = 16$ and compare the performances of CE-FedAvg on FEMNIST and CIFAR-10 in Fig. 3. From the figure, we can observe that CE-FedAvg can converge faster in terms of global round number as τ decreases. This demonstrates the benefit of frequent local model aggregation in improving the convergence speed, matching the theoretical analysis in Remark 1. However, in terms of runtime, a smaller τ incurs longer communication delay for local model aggregation in each global round, which could lead to inferior performance on time-to-accuracy. For instance, to achieve a target test

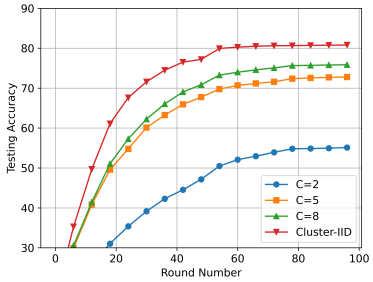


Fig. 5: Testing accuracy vs. round number for CE-FedAvg on CIFAR-10 dataset under different cluster-level data distributions when fixing $n = 64$, $\tau = 2$, and $q = 8$. Here C denotes the number of label classes each cluster has.

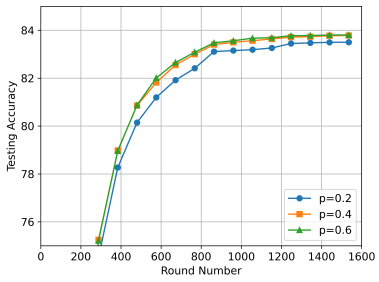


Fig. 6: Testing accuracy vs. round number for CE-FedAvg on CIFAR-10 dataset under different edge backhaul topologies when fixing $n = 64$, $\tau = 1$, $q = 1$, and $\pi = 1$.

accuracy of 80% on FEMNIST, the time needed for $\tau = 2$ is 24.6% and 24.2% more than that of $\tau = 4$ and $\tau = 8$. To achieve a target test accuracy of 80% on CIFAR-10, the time needed for $\tau = 2$ is 4% and 24% more than that of $\tau = 4$ and $\tau = 8$.

Then, we investigate how the cluster number m affects the performance of CE-FedAvg. We select $m = \{4, 8, 16\}$ while fixing the total number of devices $n = 64$, corresponding to $\{16, 8, 4\}$ randomly assigned devices per cluster. Fig. 4 depicts the testing accuracy vs. round number. As can be observed from the figure, decreasing m leads to better convergence because more devices cooperate with each other during each edge round, and the divergence of their models is smaller. This is consistent with the analysis in Remark 2.

After that, we study the impact of cluster-level data distribution on the performance of CE-FedAvg. In CFEL, there are two levels of non-IID data distribution: device-level and cluster-level, corresponding to the intra-cluster and inter-cluster divergence properties. Note that even though the data distribution of device exhibits heterogeneity, the data distribution of cluster can be homogeneous. Specifically, we consider the following two cases on cluster-level data distribution for CIFAR-10 dataset:

- *Cluster IID:* The 50,000 training images are first evenly partitioned in an IID fashion across $m = 8$ clusters with each cluster having 6250 images. Then, within each cluster, we sort the 6250 images by label, evenly divide them into 16 shards, and then assign

each of 8 devices 2 shards such that devices will only have images of two labels. The data distribution among clusters is IID in this case.

- *Cluster Non-IID:* We first sort the 50,000 training images by label, evenly divide them into $C \times 8$ shards, and then assign C shards to each of the 8 clusters such that each cluster roughly has images of C labels. We set $C = \{2, 5, 8\}$ in the experiment. Then, within each cluster, we sort the assigned images by label, evenly divide them into 16 shards, and then assign each of 8 clusters 2 shards such that devices will only have images of two labels. The data distribution among clusters is non-IID in this case.

We compare the performances of CE-FedAvg under the above cases in Fig. 5. The result shows that CE-FedAvg converges much faster under the Cluster IID than Cluster Non-IID case. Therefore, if the grouping strategy of devices can be controlled in practice, it makes sense to group devices to follow the IID fashion across clusters to accelerate the convergence and reduce runtime of CE-FedAvg. Furthermore, as C increases, the inter-cluster divergence increases while the global divergence is fixed, the convergence speed of CE-FedAvg will decrease correspondingly, matching our theoretical analysis in Remark 3.

Finally, we evaluate the convergence of CE-FedAvg under varying edge backhaul topologies in Fig. 6. We generate random network topologies by Erdős-Rényi model with edge probability $p = \{0.2, 0.4, 0.6\}$. As observed in the figure, a more connected network topology (i.e., a larger value of p and smaller value of ξ) generally accelerates the convergence and leads to a higher model accuracy achieved after 1500 communication rounds in CE-FedAvg. This matches our theoretical results in Theorem 1.

7 Conclusion

In this paper, we proposed CFEL, a novel FL framework that integrates a distributed network of cooperative edge servers for fast model aggregation and achieves scalable and low-latency model learning at mobile edge networks. Specifically, a new federated optimization algorithm called CE-FedAvg was developed under the proposed CFEL, and its convergence properties were analyzed under the general non-convex and non-IID setting. Experiments demonstrated that compared with other FL frameworks, CFEL can largely reduce the training time to achieve a target model accuracy. For future work, we will investigate computational heterogeneity and rigorous privacy protection in CFEL.

References

[1] H. Zhang, J. Bosch, and H. H. Olsson, "Real-time end-to-end federated learning: An automotive case study," in *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2021, pp. 459–468.

[2] A. Sheth, U. Jaimini, K. Thirunarayan, and T. Banerjee, "Augmented personalized health: how smart data with IoTs and AI is about to change healthcare," in *2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI)*. IEEE, 2017, pp. 1–6.

[3] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *Computer*, vol. 50, no. 10, pp. 58–67, 2017.

[4] A. Subasi, M. Radhwan, R. Kurdi, and K. Khateeb, "IoT based mobile healthcare system for human activity recognition," in *2018 15th Learning and Technology Conference (L&T)*. IEEE, 2018, pp. 29–34.

[5] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," *Journal of Manufacturing Systems*, vol. 48, pp. 157–169, 2018.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[7] 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," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[8] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. [Online]. Available: https://arxiv.org/abs/1610.05492

[9] D. Rothchild, A. Panda, E. Ullah, N. Ivkin, I. Stoica, V. Braverman, J. Gonzalez, and R. Arora, "FetchSGD: Communication-efficient federated learning with sketching," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8253–8265.

[10] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright, "ATOMO: Communication-efficient learning via atomic sparsification," in *Advances in Neural Information Processing Systems*, 2018, pp. 9850–9861.

[11] N. H. Tran, W. Bao, A. Zomaya, N. M. NH, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1387–1395.

[12] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 491–506, 2019.

[13] G. Zhu, Y. Du, D. Gündüz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 2120–2135, 2020.

[14] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.

[15] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 317–333, 2019.

[16] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2155–2169, 2020.

[17] J. Ren, G. Yu, and G. Ding, "Accelerating dnn training in wireless federated edge learning systems," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 219–232, 2020.

[18] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an intelligent edge: wireless communication meets machine learning," *IEEE Communications Magazine*, vol. 58, no. 1, pp. 19–25, 2020.

[19] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7690–7703, 2020.

[20] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient resource management for federated edge learning with CPU-GPU heterogeneous computing," *IEEE Transactions on Wireless Communications*, vol. 20, no. 12, pp. 7947–7962, 2021.

[21] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[22] J. Wang, S. Wang, R.-R. Chen, and M. Ji, "Demystifying why local aggregation helps: Convergence analysis of hierarchical SGD," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[23] Z. Zhong, Y. Zhou, D. Wu, X. Chen, M. Chen, C. Li, and Q. Z. Sheng, "P-fedavg: parallelizing federated learning with theoretical guarantees," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[24] T. Castiglia, A. Das, and S. Patterson, "Multi-level local SGD: Distributed SGD for heterogeneous hierarchical networks," in *International Conference on Learning Representations*, 2020.

[25] Y. Sun, J. Shao, Y. Mao, J. H. Wang, and J. Zhang, "Semi-decentralized federated edge learning for fast convergence on non-iid data," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 1898–1903.

[26] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2021–2031.

[27] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[28] Y. Guo and Y. Gong, "Practical collaborative learning for crowdsensing in the internet of things with differential privacy," in *2018 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2018, pp. 1–9.

[29] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *International Conference on Learning Representations*, 2018.

[30] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong, "Personalized federated learning with differential privacy," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9530–9539, 2020.

[31] R. Hu, Y. Gong, and Y. Guo, "Federated learning with sparsification-amplified privacy and adaptive optimization," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, Z.-H. Zhou, Ed. ijcai.org, 2021, pp. 1463–1469. [Online]. Available: https://doi.org/10.24963/ijcai.2021/202

[32] A. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. T. Suresh, "Shuffled model of differential privacy in federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 2521–2529.

[33] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of local-update sgd algorithms," *Journal of Machine Learning Research*, vol. 22, no. 213, pp. 1–50, 2021.

[34] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[35] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized sgd with changing topology and local updates," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5381–5393.

[36] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7184–7193.

[37] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[38] S. Caldas, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "LEAF: A benchmark for federated settings," in *Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.

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

[40] S. Li, Y. Cheng, Y. Liu, W. Wang, and T. Chen, "Abnormal client behavior detection in federated learning," in *International Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.

[41] H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," 2019. [Online]. Available: https://arxiv.org/abs/1909.06335

[42] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *International Conference on Learning Representations*, 2020.

[43] J. Yuan, M. Xu, X. Ma, A. Zhou, X. Liu, and S. Wang, "Hierarchical federated learning through LAN-WAN orchestration," *arXiv preprint arXiv:2010.11612*, 2020.