

# Hierarchical Federated Learning with Momentum Acceleration in Multi-Tier Networks

Zhengjie Yang, Sen Fu, Wei Bao, Dong Yuan, and Albert Y. Zomaya

**Abstract**—In this paper, we propose Hierarchical Federated Learning with Momentum Acceleration (HierMo), a three-tier worker-edge-cloud federated learning algorithm that applies momentum for training acceleration. Momentum is calculated and aggregated in the three tiers. We provide convergence analysis for HierMo, showing a convergence rate of  $\mathcal{O}(\frac{1}{T})$ . In the analysis, we develop a new approach to characterize model aggregation, momentum aggregation, and their interactions. Based on this result, we prove that HierMo achieves a tighter convergence upper bound compared with HierFAVG without momentum. We also propose HierOPT, which optimizes the aggregation periods (worker-edge and edge-cloud aggregation periods) to minimize the loss given a limited training time. By conducting the experiment, we verify that HierMo outperforms existing mainstream benchmarks under a wide range of settings. In addition, HierOPT can achieve a near-optimal performance when we test HierMo under different aggregation periods.

**Index Terms**—Federated learning; momentum; convergence analysis; edge computing

## I. INTRODUCTION

With the advancement of Industry 4.0, Internet of Things (IoT), and Artificial Intelligence, machine learning applications such as image classification [1], automatic driving [2], and automatic speech recognition [3] are rapidly developed. Since the machine learning dataset is distributed in individual users and in many situations they are not willing to share these sensitive raw data, Federated Learning (FL) emerges [4]. It allows workers to participate in the model training without sharing their raw data. Typically, FL is implemented in two tiers, where multiple devices (workers) are distributed and connected to a remote aggregator (usually located in the cloud). A potential issue of the two-tier FL setting is its scalability. The communication overhead between workers and the cloud is proportional to the number of workers, which causes problems when there are a large number of geodistributed workers connecting to the remote cloud via the public Internet.

With the development of edge computing [5], a more effective solution is adding the edge tier between local workers and the remote cloud to address the scalability issue. Different from the typical two-tier architecture, in the three-tier hierarchical architecture as shown in Fig. 1, workers can first communicate with the edge node for edge-level aggregation, and then the edge nodes communicate with the remote cloud for cloud-level aggregation. Each edge node is closer to the workers and is usually connected with them in the same local/edge network, so that the communication cost is much cheaper compared with the two-tier case when the workers directly communicate with the cloud. In Fig. 1, we can see that

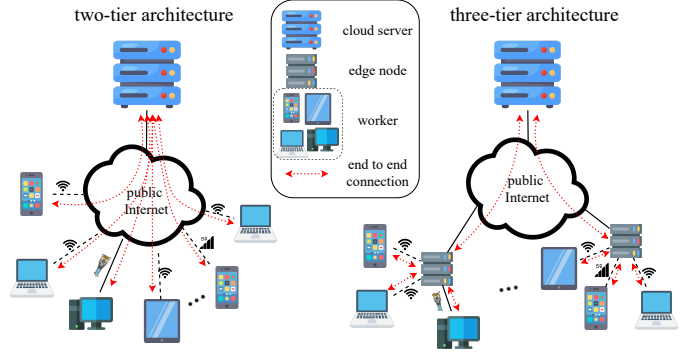


Fig. 1. Two-tier architecture vs. three-tier architecture. 6 connections are through the public Internet in the two-tier architecture but only 2 connections are through the public Internet in the three-tier architecture. Communication burdens are restrained in the local/edge networks.

much of the traffic through the public Internet (left subfigure) is restrained in the local edge networks (right subfigure) due to the existence of the edge nodes. Therefore, the three-tier architecture is a good fit for larger-scale FL, and has attracted attentions from researchers in recent years [6]–[8].

Although the three-tier FL can improve the communication efficiency *in one training iteration* by replacing worker-cloud communication with worker-edge communication, there is also a need to accelerate its convergence performance to reduce the *number of iterations*. One obstacle in the three-tier FL is that each edge node can only aggregate the updates of its local workers, and there is a discrepancy among edge nodes. The edge nodes are to be synchronized in the cloud-level aggregation. The two-level aggregation causes delayed synchronization, leading to less training efficiency. Therefore, it is a strong motivation for us to develop a more efficient algorithm to accelerate the convergence, reducing the number of training iterations in the three-tier hierarchical architecture, and finally improve the overall training efficiency (considering both per-iteration cost and the number of iterations).

Momentum is proved to be an effective mechanism to accelerate model training. Many studies have demonstrated its advantage in both centralized machine learning environment [9]–[12] and two-tier FL environment [13]–[16]. Apart from the conventional gradient descent step, the momentum method conducts additional momentum steps [17] to accelerate convergence. In this paper, we propose Hierarchical Federated Learning with Momentum Acceleration (HierMo), which leverages momentum to accelerate three-tier FL. HierMo is operated as follows: ① In each iteration, each worker locally updates its own model and worker momentum; ② In every

$\tau$  iterations ( $\tau$  is called the worker-edge aggregation period), each edge node receives, averages, and sends back the models and momentum values with its connected workers.<sup>1</sup> ③ In every  $\tau \cdot \pi$  iterations ( $\pi$  is called the edge-cloud aggregation period), the cloud receives, averages, and sends back the models and momentum values with edge nodes. The edge nodes will then distribute them to connected workers. The above ①–③ steps are repeated for multiple rounds until the loss is sufficiently small.

Theoretically, we prove that HierMo is convergent and has an  $\mathcal{O}(\frac{1}{T})$  convergence rate for smooth non-convex problems for a given  $T$  iterations. In this step, we need to address substantial new challenges, compared with two-tier FL. In particular, we develop a new method to characterize the *multi-time cross-two-tier momentum interaction* and *cross-three-tier momentum interaction*, which do not exist in the two-tier FL. After we theoretically prove the convergence, we observe that the worker-edge and edge-cloud aggregation periods  $\tau$  and  $\pi$  are key design variables we aim to optimize. Based on the result of the convergence analysis, we propose HierOPT algorithm, which can find a local optimal  $(\tau, \pi)$  value pair.

In the experiment, we demonstrate the performance of HierMo compared with various mainstream hierarchical FL and momentum-based FL algorithms, including hierarchical FL without momentum (HierFAVG [18] and CFL [19]), two-tier FL with momentum (FedMom [20], SlowMo [21], FedNAG [22], Mime [23], DOMO [24], and FedADC [25]), and two-tier FL without momentum (FedAvg [4]). The experiment is implemented on different kinds of models (linear regression, logistic regress, CNN [26], VGG16 [27], and ResNet18 [28]) based on various real-world datasets (MNIST [29], CIFAR-10 [30], ImageNet [28], [31] for image classification, and UCI-HAR [32] for human activity recognition). The experimental results illustrate that HierMo drastically outperforms benchmarks under a wide range of settings. We also verify HierOPT can output a near-optimal  $(\tau, \pi)$  in the real-world settings. All these results match our expectations by the theoretical analysis.

The contributions of this paper are summarized as follows.

- We have proved that HierMo is convergent and has an  $\mathcal{O}(\frac{1}{T})$  convergence rate for smooth non-convex problems for a given  $T$  iterations under non-i.i.d. data.
- We have proved that as long as learning step size  $\eta$  is sufficiently small, HierMo (with momentum acceleration) achieves the tighter convergence upper bound than HierFAVG (without momentum acceleration).
- We have proposed the new HierOPT algorithm which can find a local optimal pair of  $(\tau^*, \pi^*)$  when total training time is constrained.
- HierMo is efficient and decreases the total training time by 21–70% compared with the mainstream two-tier momentum-based algorithms and three-tier algorithms.
- HierOPT generates the near-optimal pair of  $(\tau^*, \pi^*)$  when the total training time is constrained. HierOPT achieves the near-optimal accuracy with only 0.23–0.29%

(CNN on MNIST) and 0.04–0.16% (CNN on CIFAR10) gap from the real-world optimum.

The rest of the paper is organized as follows. In Section II, we introduce related works. The HierMo algorithm design is described in Section III. In Section IV, we provide theoretical results including the convergence analysis of HierMo and the performance gain of momentum. The algorithm to optimize the aggregation periods, i.e., HierOPT, is proposed in Section V. Section VI provides our experimental results and the conclusion is made in Section VII.

## II. RELATED WORK

### A. Momentum in Machine Learning and Federated Learning

Momentum [33] is a method that helps accelerate gradient descent in the relevant direction by adding a fraction  $\gamma$  of the difference between past and current model vectors. In the classical centralized setting, the update rule of the momentum (Polyak’s momentum) is as follows:

$$\mathbf{m}(t) = \gamma \mathbf{m}(t-1) - \eta \nabla F(\mathbf{w}(t-1)), \quad (1)$$

$$\mathbf{w}(t) = \mathbf{w}(t-1) + \mathbf{m}(t), \quad (2)$$

with  $\gamma \in [0, 1], t = 1, 2, \dots, \mathbf{m}(0) = \mathbf{0}$ , where  $\gamma$  is momentum factor (weight of momentum),  $t$  is update iteration,  $\mathbf{m}(t)$  is momentum term at iteration  $t$ , and  $\mathbf{w}(t)$  is model parameter at iteration  $t$ . Through this method, the momentum term increases for dimensions whose gradients point in the same directions and reduces updates for dimensions whose gradients change directions. As a result, momentum gains faster convergence and reduces oscillation [17], [34].

Momentum has been investigated in both centralized machine learning and FL. In the centralized environment, another form of momentum called Nesterov Accelerate Gradient (NAG) [17], [35] is proposed. NAG<sup>2</sup> calculates the gradient based on an approximation of the next position of the parameters, i.e.,  $\nabla F(\mathbf{w}(t-1) + \gamma \mathbf{m}(t-1))$ , instead of  $\nabla F(\mathbf{w}(t-1))$  in Polyak’s momentum, leading to better convergence performance. In [11], authors study the utilization of momentum in over-parameterized models. [9] provides a unified convergence analysis for both Polyak’s momentum and NAG. [12] studies NAG in stochastic settings.

All the above works show the advantages of momentum to accelerate the centralized training and it attracts researchers’ attention to apply momentum in FL environment. Depending on where the momentum is adopted, we can categorize them into the worker momentum, aggregator momentum, and combination momentum. For the worker momentum (e.g., FedNAG [22] and Mime [23]), momentum acceleration is adopted at workers in each local iteration. However, it is vulnerable to data heterogeneity among workers, which may harm the long-run performance. For the aggregator momentum (e.g., FedMom [20] and SlowMo [21]), the momentum acceleration is adopted only at the aggregator based on the global model and it shares the same property of acceleration as in centralized setting and dampens oscillations [17]. Nevertheless, it is

<sup>1</sup>Each edge node also calculates another momentum for its own usage to further accelerate convergence. See Section III for the detailed algorithm.

<sup>2</sup>There are two mainstream equivalent representations of NAG. In this paper, we employ the representation in [9], [36].

TABLE I  
KEY NOTATIONS

$\eta$	worker model learning rate
$\tau$	worker-edge aggregation period
$\pi$	edge-cloud aggregation period
$\gamma$	worker momentum factor
$\gamma_a$	edge momentum factor
$T$	number of total local (worker) iterations indexed by $t$
$K$	number of total edge aggregations indexed by $k$
$P$	number of total global (cloud) aggregations indexed by $p$
$L$	number of edge nodes indexed by $\ell$
$C_\ell$	number of workers under edge node $\ell$
$N$	number of workers in the system indexed by $\{i, \ell\}$
$\mathbf{x}_{i,\ell}^t$	worker model parameter in worker $\{i, \ell\}$ at iteration $t$
$\mathbf{y}_{i,\ell}^t$	worker momentum parameter in worker $\{i, \ell\}$ at iteration $t$
$\mathbf{y}_{\ell-}^t$	aggregated worker momentum in edge node $\ell$ at iteration $t$
$\mathbf{x}_{\ell-}^t$	aggregated worker model in edge node $\ell$ at iteration $t$
$\mathbf{y}_{\ell+}^t$	updated edge momentum in edge node $\ell$ at iteration $t$
$\mathbf{x}_{\ell+}^t$	updated edge model in edge node $\ell$ at iteration $t$
$\mathbf{y}^t$	worker momentum cloud aggregation in the cloud at iteration $t$
$\mathbf{x}^t$	cloud model in the cloud at iteration $t$

conducted less frequently (every  $\tau$  iterations<sup>3</sup>) compared with worker momentum (every iteration), and the performance gain may not be obvious especially when  $\tau$  is large. To address the above limitations, works in [24], [25], [37] combine the worker and aggregator momenta and they show a better convergence performance than only using either worker or aggregator momentum. The above forms of momentum are only adopted and analyzed in the two-tier FL and we focus on the three-tier scenarios in this paper.

### B. Three-Tier Hierarchical Federated Learning

Three-tier FL has attracted more attention in recent years. Without considering momentum, studies have demonstrated the convergence performance in three-tier FL [18], [19], [38], [39]. The communication overhead can be further optimized in [40]. The convergence analysis extended from two-tier to three-tier FL is not straightforward. Different from two-tier FL where the global aggregation is executed every  $\tau$  local iterations, in three-tier FL, each worker's local model will be first aggregated by the connected edge node every  $\tau$  local iterations, and will then be aggregated by the cloud in another level of every  $\pi$  edge aggregations. Existing two-tier methods can only bound the two-tier effects, but not the three-tier effects. Substantial new challenges are encountered in this paper. When momentum is leveraged in the three-tier scenario, it additionally introduces *multi-time cross-two-tier momentum interaction* and *cross-three-tier momentum interaction*. This is completely different from the two-tier scenario. Existing two-tier analyses cannot deal with the above two new terms. They can only characterize multi-time inner-tier momentum acceleration and one-time cross-two-tier momentum interaction. We devise a two-level virtual update (edge and cloud) method, which is able to bound the aforementioned new terms so that the convergence of HierMo still holds.

<sup>3</sup> $\tau$  the is aggregation period

## III. HIERMO PROBLEM FORMULATION

### A. Overview

We consider a three-tier hierarchical FL system consisting of a cloud server,  $L$  edge nodes, and  $N$  workers. Each edge node  $\ell$  serves  $C_\ell$  workers, and the total number of workers is  $N = \sum_{\ell=1}^L C_\ell$ . Worker  $\{i, \ell\}$  denotes the  $i$ th worker served by edge node  $\ell$ , where  $i = 1, 2, \dots, C_\ell$ . It contains its local dataset with the number of data samples denoted by  $D_{i,\ell}$ . The total training dataset in the cluster of workers served by edge node  $\ell$  is  $D_\ell \triangleq \sum_{i=1}^{C_\ell} D_{i,\ell}$  and the total training dataset  $D \triangleq \sum_{\ell=1}^L D_\ell = \sum_{\ell=1}^L \sum_{i=1}^{C_\ell} D_{i,\ell}$ . The target of three-tier hierarchical FL is to find the stationary point  $\mathbf{x}^*$  that minimizes the global loss function  $F(\mathbf{x})$  that is the weighted average of all workers' loss functions. The problem can be formulated as follows:

$$\min_{\mathbf{x} \in \mathbb{R}^d} F(\mathbf{x}) \triangleq \frac{1}{D} \sum_{\ell=1}^L \sum_{i=1}^{C_\ell} D_{i,\ell} F_{i,\ell}(\mathbf{x}) \quad (3)$$

$$= \sum_{\ell=1}^L \frac{D_\ell}{D} \sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} F_{i,\ell}(\mathbf{x}) \quad (4)$$

$$\triangleq \sum_{\ell=1}^L \frac{D_\ell}{D} F_\ell(\mathbf{x}), \quad (5)$$

where  $d$  is the dimension of  $\mathbf{x}$ ,  $F(\mathbf{x})$  is the global loss function at the cloud server, and  $F_{i,\ell}(\mathbf{x})$  is the local loss function at worker  $\{i, \ell\}$ . (4) is the mathematical transformation from (3) by adding  $D_\ell$ . We also define the edge loss function at edge node  $\ell$  as  $F_\ell(\mathbf{x}) \triangleq \sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} F_{i,\ell}(\mathbf{x})$ , which is the weighted average of edge node  $\ell$ 's connected workers' local loss functions  $F_{i,\ell}(\mathbf{x})$ . Therefore, by replacing  $\sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} F_{i,\ell}(\mathbf{x})$  with  $F_\ell(\mathbf{x})$  in (4), we can directly derive (5), demonstrating that the global loss function is the weighted average of all edge loss functions as  $F(\mathbf{x}) \triangleq \sum_{\ell=1}^L \frac{D_\ell}{D} F_\ell(\mathbf{x})$ . We assume the problem is within the scope of cross-siloed federated learning [41] where all workers are required to participate in the training with siloed data. Each worker represents a repository of data, and data are sensitive and non-i.i.d.. The key notations are summarized in Table I.

### B. Worker Momentum and Edge Momentum

We notice that there are two types of momentum in two-tier FL: One type (i.e., worker momentum) is calculated at each worker and is aggregated; The other type (i.e., aggregator momentum) is calculated at the aggregator. Since both types can accelerate the convergence, we adopt both of them in our work. In the three-tier case in our paper, the worker momentum is individually computed in each worker and aggregated in the edge node (worker momentum edge aggregation) and the cloud (worker momentum cloud aggregation). We still call it *worker momentum* throughout the paper. For the aggregator momentum, we apply it at each edge node. Each edge node computes its own momentum and it is not shared with the workers or the cloud. We call it *edge momentum* throughout this paper.

---

**Algorithm 1** HierMo algorithm.
 

---

**Input:**  $\tau, \pi, T = K\tau = P\tau\pi, \eta, \gamma, \gamma_a$ 
**Output:** Final cloud (global) model parameter  $\mathbf{x}_{(a)}^T$ 

```

1: For each worker, initialize:  $\mathbf{x}_{i,\ell}^0$  as same value for all  $i, \ell$ , and
    $\mathbf{y}_{i,\ell}^0 = \mathbf{x}_{i,\ell}^0$ 
2: For each edge node, initialize:  $\mathbf{x}_{\ell(a)}^0 = \mathbf{x}_{i,\ell}^0$ , and  $\mathbf{y}_{\ell(a)}^0 = \mathbf{x}_{\ell(a)}^0$ 
3: for  $t = 1, 2, \dots, T$  do
4:   For each worker  $i = 1, 2, \dots, N$  in parallel,
5:      $\mathbf{y}_{i,\ell}^t \leftarrow \mathbf{x}_{i,\ell}^{t-1} - \eta \nabla F_{i,\ell}(\mathbf{x}_{i,\ell}^{t-1})$ 
     // Worker momentum update
6:      $\mathbf{x}_{i,\ell}^t \leftarrow \mathbf{y}_{i,\ell}^t + \gamma(\mathbf{y}_{i,\ell}^t - \mathbf{y}_{i,\ell}^{t-1})$ 
     // Worker model update
7:   if  $t == k\tau$  where  $k = 1, \dots, K$  then
8:     For each edge node  $\ell = 1, 2, \dots, L$  in parallel,
9:        $\mathbf{y}_{\ell-}^{k\tau} \leftarrow \sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} \mathbf{y}_{i,\ell}^{k\tau}$ 
       // Worker momentum edge aggregation
10:       $\mathbf{y}_{\ell+}^{k\tau} \leftarrow \mathbf{x}_{\ell+}^{(k-1)\tau} - \sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} (\mathbf{x}_{\ell+}^{(k-1)\tau} - \mathbf{x}_{i,\ell}^{k\tau})$ 
       // Edge momentum update
11:       $\mathbf{x}_{\ell+}^{k\tau} \leftarrow \mathbf{y}_{\ell+}^{k\tau} + \gamma_a (\mathbf{y}_{\ell+}^{k\tau} - \mathbf{y}_{\ell+}^{(k-1)\tau})$ 
       // Edge model update
12:      Set  $\mathbf{y}_{i,\ell}^{k\tau} \leftarrow \mathbf{y}_{\ell-}^{k\tau}$  for all worker  $i \in C_\ell$ 
       // Edge aggregated worker momentum
       re-distribution to workers
13:      Set  $\mathbf{x}_{i,\ell}^{k\tau} \leftarrow \mathbf{x}_{\ell+}^{k\tau}$  for all worker  $i \in C_\ell$ 
       // Edge model re-distribution to workers
14:    end if
15:    if  $t == p\tau\pi$  where  $p = 1, 2, \dots, P$  then
16:      Aggregate  $\mathbf{y}^{p\tau\pi} \leftarrow \sum_{\ell=1}^L \frac{D_\ell}{D} \mathbf{y}_{\ell-}^{p\tau\pi}$ 
      // Worker momentum cloud aggregation
17:      Aggregate  $\mathbf{x}^{p\tau\pi} \leftarrow \sum_{\ell=1}^L \frac{D_\ell}{D} \mathbf{x}_{\ell+}^{p\tau\pi}$ 
      // Edge model cloud aggregation
18:      Set  $\mathbf{y}_{\ell-}^{p\tau\pi} \leftarrow \mathbf{y}^{p\tau\pi}$  for all edge node  $\ell \in L$ 
      // Cloud aggregated worker momentum
      re-distribution to edge nodes
19:      Set  $\mathbf{x}_{\ell+}^{p\tau\pi} \leftarrow \mathbf{x}^{p\tau\pi}$  for all edge node  $\ell \in L$ 
      // Cloud model re-distribution to edge nodes
20:      Set  $\mathbf{y}_{i,\ell}^{p\tau\pi} \leftarrow \mathbf{y}_{\ell-}^{p\tau\pi}$  for all worker  $i \in C_\ell, \ell \in L$ 
      // Cloud aggregated worker momentum
      re-distribution from edge nodes to workers
21:      Set  $\mathbf{x}_{i,\ell}^{p\tau\pi} \leftarrow \mathbf{x}_{\ell+}^{p\tau\pi}$  for all worker  $i \in C_\ell, \ell \in L$ 
      // Cloud model re-distribution
      from edge nodes to workers
22:    end if
23:  end for

```

---

### C. HierMo Algorithm

In Algorithm 1, we propose a momentum-based three-tier hierarchical FL algorithm, named as HierMo, which applies both worker momentum and edge momentum. HierMo aims to find the final cloud model  $\mathbf{x}_{(a)}^T$  to solve the formula (3). It conducts  $T$  local iterations,  $K$  edge aggregations, and  $P$  cloud aggregations, where  $T = K\tau = P\tau\pi$ ,  $\tau$  is the worker-edge aggregation period, and  $\pi$  is the edge-cloud aggregation period.

1) *Worker update:* In each local iteration  $t$ , each worker  $\{i, \ell\}$  computes its worker update, which includes two things: ① worker momentum update  $\mathbf{y}_{i,\ell}^t$  (Line 5) and ② worker model update  $\mathbf{x}_{i,\ell}^t$  (Line 6). ① and ② follow the Nesterov Accelerated Gradient (NAG) [35] momentum update and are conducted every iteration. Through this way, each worker can utilize its own worker momentum acceleration.

2) *Edge update:* When  $t = k\tau, k = 1, 2, \dots, K$ , each edge node  $\ell$  receives workers' momenta and models in  $C_\ell$  and performs edge update, which includes two operations: ① Worker momentum edge aggregation  $\mathbf{y}_{\ell-}^{k\tau}$  (Line 9) with re-distribution (Line 12). Through this way, some straggler workers with high data-heterogeneity whose local momenta  $\mathbf{y}_{i,\ell}^{k\tau}$  pointing to an inappropriate direction can be refined from  $\mathbf{y}_{\ell-}^{k\tau}$ . ② Edge momentum  $\mathbf{y}_{\ell+}^{k\tau}$  and model  $\mathbf{x}_{\ell+}^{k\tau}$  update (Lines 10–11) with model re-distribution (Line 13). Since the computation of edge momentum and model update is based on the edge model, it is equivalent to perform it in edge setting involving all workers' dataset under edge node  $\ell$  ( $D_\ell = \sum_{i=1}^{C_\ell} D_{i,\ell}$ ). By doing so, it dampens oscillations [17] within the edge node. Please note that ① and ② are two operations on the same edge node, so that we use subscript “-” and “+” to label the momentum/model right after operations ① and ② respectively. Finally, both ① and ② are conducted in each edge node every  $\tau$  iterations.

3) *Cloud update:* When  $t = p\tau\pi, p = 1, 2, \dots, P$ , the cloud receives edge aggregated worker momentum  $\mathbf{y}_{\ell-}^{p\tau\pi}$  and edge model  $\mathbf{x}_{\ell+}^{p\tau\pi}$  for all  $\ell \in L$  and performs cloud update, which includes two things: ① Worker momentum cloud aggregation  $\mathbf{y}^{p\tau\pi}$  (Line 16) and re-distribution (Lines 18 and 20). Through this way, all edge nodes and workers receive the cloud aggregated worker momentum and mitigate the disadvantage caused by non-i.i.d. data heterogeneity. ② Edge model cloud aggregation  $\mathbf{x}^{p\tau\pi}$  (Line 17) and cloud model re-distribution (Lines 19 and 21). Please note that the cloud will re-distribute the momentum and model to all edge nodes and all edge nodes will then distribute them to all workers when  $t$  is a multiple of  $\tau\pi$ .

## IV. CONVERGENCE ANALYSIS OF HIERMO

In this section, we present the theoretical analysis of HierMo. We first provide preliminaries. Then, we introduce the concept of virtual update which is a significant intermediate step to conduct convergence analysis. Afterward, we show the convergence guarantee of HierMo. Finally, we compare the convergence upper bound of HierMo and HierFAVG to analyze the performance gain of momentum.

### A. Preliminaries

We assume  $F_{i,\ell}(\cdot)$  satisfies the following standard conditions that are commonly adopted in the literature [13], [22], [42].

**Assumption 1.**  $F_{i,\ell}(\mathbf{x})$  is  $\rho$ -Lipschitz, i.e.,  $\|F_{i,\ell}(\mathbf{x}_1) - F_{i,\ell}(\mathbf{x}_2)\| \leq \rho \|\mathbf{x}_1 - \mathbf{x}_2\|$  for any  $\mathbf{x}_1, \mathbf{x}_2, i, \ell$ .

**Assumption 2.**  $F_{i,\ell}(\mathbf{x})$  is  $\beta$ -smooth, i.e.,  $\|\nabla F_{i,\ell}(\mathbf{x}_1) - \nabla F_{i,\ell}(\mathbf{x}_2)\| \leq \beta \|\mathbf{x}_1 - \mathbf{x}_2\|$  for any  $\mathbf{x}_1, \mathbf{x}_2, i, \ell$ .

**Assumption 3.** (Bounded diversity) The variance of local gradient to edge gradient is bounded. i.e.,  $\|\nabla F_{i,\ell}(\mathbf{x}) - \nabla F_\ell(\mathbf{x})\| \leq \delta_{i,\ell}$  for  $\forall i, \forall \ell$ , and  $\forall \mathbf{x}$ . We also define  $\delta_\ell$  as the weighted average of  $\delta_{i,\ell}$  and  $\delta$  as the weighted average of  $\delta_\ell$ , i.e.,  $\delta_\ell \triangleq \sum_{i \in C_\ell} \frac{D_{i,\ell}}{D_\ell} \delta_{i,\ell}$  and  $\delta \triangleq \sum_{\ell \in L} \frac{D_\ell}{D} \delta_\ell$ .

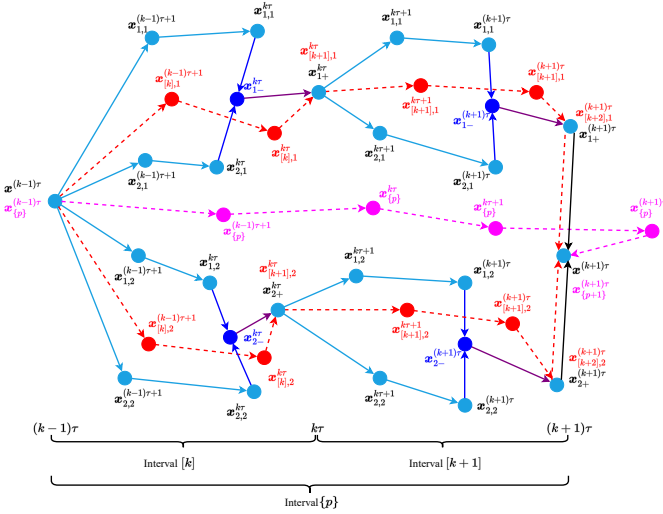


Fig. 2. Illustration of  $\mathbf{x}_{\ell-}^t$ ,  $\mathbf{x}_{\ell+}^t$ ,  $\mathbf{x}_{[k],\ell}^t$ ,  $\mathbf{x}_{\{p\}}^t$ , and  $\mathbf{x}^t$ , when  $N = 4$ ,  $\tau = 2$ ,  $\pi = 2$  with each edge node serving 2 workers. Cyan lines show worker model update. Blue lines show worker model edge aggregation. Purple lines show edge model accelerated by edge momentum. Black lines show edge model cloud aggregation. Red dashed lines show edge model virtual update. Magenta dashed lines show cloud model virtual update.

According to Assumptions 1 and 2, and applying the Triangle Inequality to  $F_{i,\ell}(\mathbf{x})$ , it is straightforward to show that  $F_\ell(\mathbf{x})$  is  $\rho$ -Lipschitz and  $\beta$ -smooth. Applying the Triangle Inequality to  $F_\ell(\mathbf{x})$ , we can also derive that  $F(\mathbf{x})$  is  $\rho$ -Lipschitz and  $\beta$ -smooth. Assumptions 1 and 2 indicate that the function and the gradient of the function are not changing too fast. Assumption 3 indicates that the data distributed to all workers are heterogeneous and non-i.i.d..  $\delta_{i,\ell}$  is used to quantify the level of gradient divergence and is different at different workers.

### B. Virtual Update

In order to index the edge aggregation and cloud aggregation, we divide the total  $T$  local iterations into  $K$  edge intervals and  $P$  cloud intervals.  $T = K\tau = P\tau\pi$ . We use  $[k]$  to denote the edge interval  $t \in [(k-1)\tau, k\tau]$  for  $k = 1, 2, \dots, K$ , and  $\{p\}$  to denote the cloud interval  $t \in [(p-1)\tau\pi, p\tau\pi]$  for  $p = 1, 2, \dots, P$ . Please note that the edge aggregation occurs at the end of each edge interval and the cloud aggregation occurs at the end of each cloud interval. Therefore, each edge interval  $[k]$  contains  $\tau$  local iterations with one edge aggregation, and each cloud interval  $\{p\}$  contains  $\pi$  edge intervals with one cloud aggregation, i.e.,  $\{p\} = \cup_k [k]$  for  $k = (p-1)\pi + 1, (p-1)\pi + 2, \dots, p\pi$ .

At the beginning of edge interval  $[k]$  when  $t = (k-1)\tau$ , we set *edge virtual update*

$$\mathbf{y}_{[k],\ell}^{(k-1)\tau} \leftarrow \mathbf{y}_{\ell-}^{(k-1)\tau}, \quad (6)$$

$$\mathbf{x}_{[k],\ell}^{(k-1)\tau} \leftarrow \mathbf{x}_{\ell+}^{(k-1)\tau}, \quad (7)$$

for each edge node  $\ell$ , where  $\mathbf{y}_{[k],\ell}^{(k-1)\tau}$  and  $\mathbf{x}_{[k],\ell}^{(k-1)\tau}$  are set as the virtual aggregated values right after the edge aggregation occurs. Then, we further conduct *edge virtual update* as if

model and momentum updates are conducted in the edge node. When  $t \in ((k-1)\tau, k\tau]$ , we conduct *edge virtual update* as

$$\mathbf{y}_{[k],\ell}^t \leftarrow \mathbf{x}_{[k],\ell}^{t-1} - \eta \nabla F_\ell(\mathbf{x}_{[k],\ell}^{t-1}), \quad (8)$$

$$\mathbf{x}_{[k],\ell}^t \leftarrow \mathbf{y}_{[k],\ell}^t + \gamma(\mathbf{y}_{[k],\ell}^t - \mathbf{y}_{[k],\ell}^{t-1}). \quad (9)$$

We repeat (6)–(9) for each edge interval  $[k]$  where  $k = 1, 2, \dots, K$ . Please note that only if  $t = k\tau$ ,  $k = 1, \dots, K$ ,  $\mathbf{y}_{\ell-}^t$  and  $\mathbf{x}_{\ell+}^t$  are computed. For ease of analysis, we define intermediate value  $\mathbf{x}_{\ell-}^t = \sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} \mathbf{x}_{i,\ell}^t$  and  $\mathbf{y}_{\ell-}^t = \sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} \mathbf{y}_{i,\ell}^t$  that are meaningful at any iteration  $t$ .

Same as edge intervals, for each cloud interval  $\{p\}$  where  $p = 1, 2, \dots, P$ , the *cloud virtual update* is also conducted:

$$\mathbf{y}_{\{p\}}^{(p-1)\tau\pi} \leftarrow \mathbf{y}^{(p-1)\tau\pi}, \quad (10)$$

$$\mathbf{x}_{\{p\}}^{(p-1)\tau\pi} \leftarrow \mathbf{x}^{(p-1)\tau\pi}, \quad (11)$$

when  $t = (p-1)\tau\pi$ , and

$$\mathbf{y}_{\{p\}}^t \leftarrow \mathbf{x}_{\{p\}}^{t-1} - \eta \nabla F(\mathbf{x}_{\{p\}}^{t-1}), \quad (12)$$

$$\mathbf{x}_{\{p\}}^t \leftarrow \mathbf{y}_{\{p\}}^t + \gamma(\mathbf{y}_{\{p\}}^t - \mathbf{y}_{\{p\}}^{t-1}), \quad (13)$$

when  $p \in ((p-1)\tau\pi, p\tau\pi]$ .

By applying virtual updates on edge nodes and the cloud, we can bound the gap between real updates and these virtual updates that can be then used to prove the convergence. Since in HierMo, the momenta and the models are aggregated on both edge nodes and the cloud, it brings much more challenges to conduct convergence analysis. The virtual update is an important intermediate process for convergence analysis and is one of our contributions in this paper.

Fig. 2 illustrates the evolution of  $\mathbf{x}_{\ell-}^t$ ,  $\mathbf{x}_{\ell+}^t$ ,  $\mathbf{x}_{[k],\ell}^t$ ,  $\mathbf{x}_{\{p\}}^t$ , and  $\mathbf{x}^t$  when  $\tau = 2$ ,  $\pi = 2$ . There are 2 edge nodes and each edge node serves 2 workers (in total 4 workers in Fig. 2). After every 2 local updates, there is an edge aggregation, and after every 2 edge aggregations (4 local updates), there is a cloud aggregation. Please note ①  $\mathbf{x}_{[k],\ell}^{k\tau}$  and  $\mathbf{x}_{[k+1],\ell}^{k\tau}$  are different.  $\mathbf{x}_{[k],\ell}^{k\tau}$  is calculated from  $\mathbf{x}_{[k],\ell}^{(k-1)\tau}$  after  $\tau$  edge virtual updates, while  $\mathbf{x}_{[k+1],\ell}^{k\tau}$  is directly given by  $\mathbf{x}_{\ell+}^{k\tau}$ . ②  $\mathbf{x}_{\ell-}^{k\tau}$  and  $\mathbf{x}_{\ell+}^{k\tau}$  are different.  $\mathbf{x}_{\ell-}^{k\tau}$  is the intermediate value that is used for edge model/momentum update, while  $\mathbf{x}_{\ell+}^{k\tau}$  is calculated from  $\mathbf{x}_{\ell-}^{k\tau}$  during edge model/momentum update. ③  $\mathbf{x}_{\{p\}}^{(k+1)\tau}$  and  $\mathbf{x}_{\{p+1\}}^{(k+1)\tau}$  are different.  $\mathbf{x}_{\{p\}}^{(k+1)\tau}$  is calculated from  $\mathbf{x}_{\{p\}}^{(k-1)\tau}$  after  $\tau \cdot \pi$  cloud virtual updates, while  $\mathbf{x}_{\{p+1\}}^{(k+1)\tau}$  is directly given by  $\mathbf{x}^{(k+1)\tau}$ .

### C. Convergence Analysis

In this section, we provide the convergence analysis of HierMo. In Theorem 1, we first focus on worker models under each edge node  $\ell$  to bound the distance between edge intermediate value  $\mathbf{x}_{\ell-}^t$  and edge virtual update  $\mathbf{x}_{[k],\ell}^t$  within interval  $[k]$ .

**Theorem 1.** For any edge interval  $[k]$ ,  $\forall t \in ((k-1)\tau, k\tau]$  and  $\forall \ell \in L$ , we have

$$\|\mathbf{x}_{\ell-}^t - \mathbf{x}_{[k],\ell}^t\| \leq h(t - (k-1)\tau, \delta_\ell), \quad (14)$$

where  $h(x, \delta_\ell)$  is

$$h(x, \delta_\ell) = \eta\delta_\ell \left( I(\gamma A)^x + J(\gamma B)^x - \frac{1}{\eta\beta} - \frac{\gamma^2(\gamma^x - 1) - (\gamma - 1)x}{(\gamma - 1)^2} \right), \quad (15)$$

and  $A, B, I,$  and  $J$  are constants defined in Appendix A, for  $0 < \gamma < 1$  and any positive integer  $x$ .

Please note that when  $t = (k - 1)\tau$  for all  $[k]$ , we have  $\|\mathbf{x}_{\ell-}^t - \mathbf{x}_{[k],\ell}^t\| = 0 = h(0, \delta_\ell)$ , which also satisfies (15). Also,  $F_\ell(\mathbf{x})$  is  $\rho$ -Lipschitz, so that we also have

$$F_\ell(\mathbf{x}_{\ell-}^t) - F_\ell(\mathbf{x}_{[k],\ell}^t) \leq \rho h(t - (k - 1)\tau, \delta_\ell). \quad (16)$$

*Proof sketch.* We first obtain the worker momentum upper bound  $\|\mathbf{y}_{i,\ell}^t - \mathbf{y}_{[k],\ell}^t\|$  for each worker  $\{i, \ell\}$ . Based on it and worker momentum update rules in Lines 5–6 in Algorithm 1, we bound the worker model parameter gap  $\|\mathbf{x}_{i,\ell}^t - \mathbf{x}_{[k],\ell}^t\|$ . Then, we extend above two bounds to obtain edge aggregated worker momentum upper bound  $\|\mathbf{y}_{\ell-}^t - \mathbf{y}_{[k],\ell}^t\|$ . Finally, the gap of edge model parameter  $\|\mathbf{x}_{\ell-}^t - \mathbf{x}_{[k],\ell}^t\|$  is obtained. See Appendix A for the complete proof.  $\square$

In Theorem 2, we then bound the edge momentum update between  $\mathbf{x}_{\ell+}^{k\tau}$  and  $\mathbf{x}_{\ell-}^{k\tau}$  within interval  $[k]$ .

**Theorem 2.** For any edge interval  $[k]$  in any edge node  $\ell \in L$ , suppose  $0 < \gamma < 1, 0 < \gamma_a < 1$ , and any  $\tau = 1, 2, \dots$ , we have

$$\|\mathbf{x}_{\ell+}^{k\tau} - \mathbf{x}_{\ell-}^{k\tau}\| \leq s(\tau), \quad (17)$$

where  $s(\tau)$  is

$$s(\tau) = \gamma_a \tau \eta \rho (\gamma \mu + \gamma + 1) \quad (18)$$

and constant  $\mu$  is defined in Appendix E.

*Proof sketch.* Based on the edge momentum update rules in Lines 10–11 in Algorithm 1, we can derive  $\mathbf{x}_{\ell+}^{k\tau} - \mathbf{x}_{\ell-}^{k\tau} = \gamma_a (\mathbf{x}_{\ell-}^{k\tau} - \mathbf{x}_{\ell-}^{(k-1)\tau}) = \gamma_a \sum_{t=(k-1)\tau}^{k\tau-1} (\mathbf{x}_{\ell-}^{t+1} - \mathbf{x}_{\ell-}^t)$ . Then we prove the bound of  $\|\mathbf{x}_{\ell-}^{t+1} - \mathbf{x}_{\ell-}^t\|$  based on the definition of intermediate value where  $\mathbf{x}_{\ell-}^t = \sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} \mathbf{x}_{i,\ell}^t$ , and then the result is obtained. See Appendix E for the complete proof.  $\square$

By combining the results of Theorem 1 and Theorem 2, we can telescope the bound within edge interval  $[k]$  to the cloud interval  $\{p\}$  where  $k = (p-1)\pi + 1, (p-1)\pi + 2, \dots, p\pi$ . Then, we are ready to bound the gap between weighted average of edge virtual update  $\sum_{\ell=1}^L \frac{D_\ell}{D} \mathbf{x}_{[p\pi],\ell}^{p\tau\pi}$  and cloud virtual update  $\mathbf{x}_{\{p\}}^{p\tau\pi}$  in Theorem 3.

**Theorem 3.** For any cloud interval  $\{p\}$ ,  $0 < \gamma < 1$ , and  $0 < \gamma_a < 1$ , when edge interval  $[k] = [p\pi]$  (the last edge interval in cloud interval  $\{p\}$ ), and  $\forall \tau, \pi \in \{1, 2, \dots\}$  we have

$$\|\mathbf{x}_{[p\pi]}^{p\tau\pi} - \mathbf{x}_{\{p\}}^{p\tau\pi}\| \leq h(\tau\pi, \delta) + \pi \sum_{\ell=1}^L \frac{D_\ell}{D} (h(\tau, \delta_\ell) + s(\tau)), \quad (19)$$

where we define  $\mathbf{x}_{[p\pi]}^{p\tau\pi} \triangleq \sum_{\ell=1}^L \frac{D_\ell}{D} \mathbf{x}_{[p\pi],\ell}^{p\tau\pi}$ , for  $\forall \ell \in L$ .

*Proof sketch.* We propose an intermediate sequence of edge virtual update on the cloud  $\mathbf{x}_{\{p\},\ell}^{p\tau\pi}$ . We then bound  $\|\mathbf{x}_{[p\pi]}^{p\tau\pi} - \mathbf{x}_{\{p\},\ell}^{p\tau\pi}\|$  and  $\|\mathbf{x}_{\{p\},\ell}^{p\tau\pi} - \mathbf{x}_{\{p\}}^{p\tau\pi}\|$  respectively to obtain the final result. See Appendix F for complete proof.  $\square$

**Theorem 4.** Under the following conditions: (1)  $0 < \beta\eta(\gamma + 1) \leq 1$ ,  $0 < \gamma < 1$ ,  $0 < \gamma_a < 1$ , and  $\forall \tau, \pi \in \{1, 2, \dots\}$ ; (2)  $\exists \varepsilon > 0$ , (2.1)  $\omega\alpha\sigma^2 - \frac{\rho j(\tau, \pi, \delta_\ell, \delta)}{\tau\pi\varepsilon^2} > 0$ ; (2.2)  $F(\mathbf{x}_{\{p\}}^{p\tau\pi}) - F(\mathbf{x}^*) \geq \varepsilon, \forall p$ ; and (2.3)  $F(\mathbf{x}^T) - F(\mathbf{x}^*) \geq \varepsilon$  are satisfied; Algorithm 1 gives

$$F(\mathbf{x}^T) - F(\mathbf{x}^*) \leq \frac{1}{T \left( \omega\alpha\sigma^2 - \frac{\rho j(\tau, \pi, \delta_\ell, \delta)}{\tau\pi\varepsilon^2} \right)}. \quad (20)$$

where  $j(\tau, \pi, \delta_\ell, \delta)$  is

$$j(\tau, \pi, \delta_\ell, \delta) = h(\tau\pi, \delta) + (\pi + 1) \sum_{\ell=1}^L \frac{D_\ell}{D} ((h(\tau, \delta_\ell) + s(\tau))). \quad (21)$$

We define  $F(\mathbf{x}^*)$  as the minimum value, if there exists some  $\varphi > 0$  such that  $F(\mathbf{x}^*) \leq F(\mathbf{x})$  for all  $\mathbf{x}$  within distance  $\varphi$  of  $\mathbf{x}^*$ . Constant  $\mu$  is defined in Appendix E and constants  $\omega, \sigma$ , and  $\alpha$  are defined in Appendix H.

*Proof sketch.* We first analyze the convergence of  $F(\mathbf{x}_{\{p\}}^{t+1}) - F(\mathbf{x}_{\{p\}}^t)$  within cloud interval  $\{p\}$  when  $t \in [(p-1)\tau\pi, p\tau\pi)$ . Then, we merge  $h(\tau, \delta_\ell)$ ,  $s(\tau)$ , and result of Theorem 3 to handle the overall effects and telescope the gap of overall effects to all  $P$  cloud intervals, and then the final result is obtained. See Appendix H for complete proof.  $\square$

Please note in the proof of Theorems 2, 3, and 4, we have characterized the multi-time cross-two-tier momentum interaction and cross-three-tier momentum interaction brought by the three-tier FL. To analyze  $\pi$  times cross-two-tier momentum interactions, we devise a new telescope form to bound these new deviations (Equations (49)–(51) and (58)). To analyze cross-three-tier momentum interaction, we devise a new mechanism to analyze such momentum interactions across multi-tiers (Equations (52)–(58) and (64)–(67)).

We have demonstrated that the gap between the global loss function value  $F(\mathbf{x}^T)$  and the stationary point  $F(\mathbf{x}^*)$  is upper bounded by a function of  $T$  ( $T = K\tau = P\tau\pi$ ) which is inversely proportional to  $T$ . It converges with the convergence rate  $\mathcal{O}(\frac{1}{T})$  for smooth non-convex problems under non-i.i.d. data distribution. We also give the following observations based on the above theorems.

**Observation 1.** The overall gap in Theorem 4,  $F(\mathbf{x}^T) - F(\mathbf{x}^*)$  decreases when  $T$  is larger. From Appendix G, we have  $h(x) \geq 0$  for any  $x = 1, 2, \dots$ , and it increases with  $x$ . According to (18),  $s(\tau)$  increases with  $\tau$ . According to (21),  $j(\tau, \pi)$  increases with  $\tau$  and  $\pi$ . Therefore, the value of  $\frac{\rho j(\tau, \pi)}{\tau\pi\varepsilon^2}$  increases with  $\tau$  and  $\pi$  so as to increase the overall bound  $F(\mathbf{x}^T) - F(\mathbf{x}^*)$ . However, in order to let the Condition (2.1) in Theorem 4 hold, we cannot set a very large  $\tau$  and  $\pi$ , implying that convergence is guaranteed when  $j(\tau, \pi)$  is below a certain threshold. Experiments on the effects of  $\tau$  and  $\pi$  further verify that larger  $\tau$  and  $\pi$  decreases the convergence performance.

In Theorem 5, we further eliminate the value  $\varepsilon$  in Theorem 4 and further demonstrate the bound between the final loss function value that the algorithm can obtain  $F(\mathbf{x}^f)$  and the stationary point  $F(\mathbf{x}^*)$ , where we define

$$\mathbf{x}^f \triangleq \arg \min_{\mathbf{x} \in \{\mathbf{x}^{p\tau\pi}, p=1,2,\dots,P\}} F(\mathbf{x}). \quad (22)$$

**Theorem 5.** *Under the following condition:  $0 < \beta\eta(\gamma+1) \leq 1$ ,  $0 < \gamma < 1$ ,  $0 < \gamma_a < 1$ , and  $\forall \tau, \pi \in \{1, 2, \dots\}$ , we have*

$$F(\mathbf{x}^f) - F(\mathbf{x}^*) \leq \frac{1}{2T\omega\alpha\sigma^2} + \rho j(\tau, \pi, \delta_\ell, \delta) \quad (23)$$

$$+ \sqrt{\frac{1}{4T^2\omega^2\alpha^2\sigma^4} + \frac{\rho j(\tau, \pi, \delta_\ell, \delta)}{\omega\alpha\sigma^2\tau\pi}} \triangleq f_{HierMo}(T).$$

*Proof.* See Appendix I for complete proof.  $\square$

Theorem 5 will be used in Section IV-D and Section V to compare the convergence upper bound and formulate the optimization problem respectively.

#### D. Comparison between HierMo and HierFAVG

In this section, we theoretically quantify the performance gain brought by HierMo compared with HierFAVG (without momentum). The convergence upper bound of HierFAVG can be derived from [18] as follows:

$$F(\hat{\mathbf{x}}^f) - F(\mathbf{x}^*) \leq \frac{1}{2T\omega\hat{\alpha}\sigma^2} + \rho \hat{j}(\tau, \pi, \delta_\ell, \delta) \quad (24)$$

$$+ \sqrt{\frac{1}{4T^2\omega^2\hat{\alpha}^2\sigma^4} + \frac{\rho \hat{j}(\tau, \pi, \delta_\ell, \delta)}{\omega\hat{\alpha}\sigma^2\tau\pi}} \triangleq f_{HierFAVG}(T).$$

The definitions of  $\hat{\alpha}$  and  $\hat{j}(\cdot)$  can be found in [18].

To prevent the gradient descent from overshooting [43], it is common to choose a very small  $\eta$ . The following theorem is made when  $\eta \rightarrow 0^+$ .

**Theorem 6.** *When  $0 < \beta\eta(\gamma+1) \leq 1$ ,  $0 < \gamma < 1$ ,  $0 < \gamma_a < 1$ , and  $\forall \tau, \pi \in \{1, 2, \dots\}$ , HierMo outperforms HierFAVG, i.e.,*

$$f_{HierFAVG}(T) - f_{HierMo}(T) > 0$$

for any  $T$  and  $\eta \rightarrow 0^+$ .

*Proof.* See Appendix J for detailed proof.  $\square$

The above theorem indicates that HierMo leads to a tighter convergence upper bound compared with HierFAVG, showing that HierMo theoretically outperforms HierFAVG.

## V. AGGREGATION PERIOD OPTIMIZATION BY HIEROPT

We have proved that HierMo is convergent in section IV. We observe that the worker-edge and edge-cloud aggregation periods  $\tau$  and  $\pi$  are two key design variables that will influence the convergence performance. The values of  $\tau$  and  $\pi$  will also influence the usage of communication and computation resources in the real-world training process. Therefore, we aim to optimize these two variables and formulate an optimization problem: Under a given total training time denoted as  $\Psi$ , how the HierMo algorithm achieves the best performance (min global model loss).

We denote the worker computation delay for one iteration as  $\Theta_w$ , edge computation delay for one edge aggregation as  $\Theta_e$ , and cloud computation delay for one cloud aggregation as  $\Theta_c$ . We also denote the worker communication delay to the edge as  $\Phi_{w2e}$  and edge communication delay to the cloud as  $\Phi_{e2c}$ . All the above values are assumed to be given as they can be measured in the real world. We assume each worker  $\{i, \ell\}$  communicates with connected edge node  $\ell$  in parallel and each edge node  $\ell$  communicates with cloud in parallel [8], [18], [44]. The above assumptions are commonly adopted in the literature [42], [44]. As a result, the total training time for HierMo is calculated as follows

$$\Psi \triangleq P \cdot (\tau\pi\Theta_w + \pi\Theta_e + \Theta_c + \pi\Phi_{w2e} + \Phi_{e2c}), \quad (25)$$

where  $P$  is the total number of cloud aggregations ( $P = \frac{T}{\tau\pi}$ ).

In order to find the optimal pair of  $(\tau, \pi)$ , we target to minimize (23), where (23) demonstrates the bound between the global loss and the stationary point [18], [42]. By incorporating the constraints, the optimization problem can be formulated as follows

$$\min_{\tau, \pi} \frac{1}{2T\omega\alpha\sigma^2} + \rho j(\tau, \pi, \delta_\ell, \delta) \quad (26)$$

$$+ \sqrt{\frac{1}{4T^2\omega^2\alpha^2\sigma^4} + \frac{\rho j(\tau, \pi, \delta_\ell, \delta)}{\omega\alpha\sigma^2\tau\pi}},$$

$$\text{s.t. } P \cdot (\tau\pi\Theta_w + \pi\Theta_e + \Theta_c + \pi\Phi_{w2e} + \Phi_{e2c}) = \Psi, \quad (26a)$$

$$T = P\tau\pi, \quad (26b)$$

$$\tau \geq 1, \quad (26c)$$

$$\pi \geq 1. \quad (26d)$$

From constraints (26a) and (26b), we obtain

$$\frac{1}{T} = \frac{\Theta_e + \Phi_{w2e}}{\Psi} \frac{1}{\tau} + \frac{\Theta_c + \Phi_{e2c}}{\Psi} \frac{1}{\tau\pi} + \frac{\Theta_w}{\Psi}. \quad (27)$$

Substituting (27) into (26), we can eliminate the equation constraints. We also define

$$q(\tau, \pi) \triangleq \frac{1}{2T\omega\alpha\sigma^2} \quad (28)$$

$$= \frac{\Theta_e + \Phi_{w2e}}{2\Psi\omega\alpha\sigma^2} \frac{1}{\tau} + \frac{\Theta_c + \Phi_{e2c}}{2\Psi\omega\alpha\sigma^2} \frac{1}{\tau\pi} + \frac{\Theta_w}{2\Psi\omega\alpha\sigma^2}.$$

The problem (26) can be re-formulated as

$$\min_{\tau, \pi} q(\tau, \pi) + \rho j(\tau, \pi, \delta_\ell, \delta) + \sqrt{q^2(\tau, \pi) + \frac{\rho j(\tau, \pi, \delta_\ell, \delta)}{\omega\alpha\sigma^2\tau\pi}}, \quad (29)$$

$$\text{s.t. } \tau \geq 1, \quad (29a)$$

$$\pi \geq 1. \quad (29b)$$

It is non-trivial to find a closed-form optimal pair of  $(\tau, \pi)$  in the three-tier hierarchical FL because problem (29) includes both polynomial and exponential terms of  $\tau$  and  $\pi$ , where the exponential term is nest-embedded in  $h(\cdot)$  that is embedded in  $j(\cdot)$ . Even if for a two-tier FL problem, the objective function of the bound is complicated, and it is still infeasible to find an optimal solution in closed form [42], [44]. In what follows, we propose the Hierarchical Optimizing Periods (HierOPT) algorithm to find a local optimal solution to problem (29).

In Algorithm 2, for convenience, we define the objective function (29) as  $\mathcal{R}(\tau, \pi)$  with respect to  $\tau$  and  $\pi$ . We also

**Algorithm 2** HierOPT algorithm.

---

**Input:**  $\Psi, \Theta_w, \Theta_e, \Theta_c, \Phi_{w2e}, \Phi_{e2c}$   
**Output:**  $\tau^*$  and  $\pi^*$ 


---

```

1: Initialize  $\tau_0$  and  $\pi_0$  as random positive integers,  $i = 0$  as the
   index of search iteration.
2: while true do
3:   Calculate  $\mathcal{R}'(\tau_i)$ 
4:   if  $\mathcal{R}'(\tau_i) > 0$  then
5:      $\tau_{i+1} \leftarrow \max\{\tau_i - 1, 1\}$ 
6:   else if  $\mathcal{R}'(\tau_i) < 0$  then
7:      $\tau_{i+1} \leftarrow \tau_i + 1$ 
8:   end if
9:   Calculate  $\mathcal{R}'(\pi_i)$ 
10:  if  $\mathcal{R}'(\pi_i) > 0$  then
11:     $\pi_{i+1} \leftarrow \max\{\pi_i - 1, 1\}$ 
12:  else if  $\mathcal{R}'(\pi_i) < 0$  then
13:     $\pi_{i+1} \leftarrow \pi_i + 1$ 
14:  end if
15:  Record  $(\tau_i, \pi_i)$ 
16:  if the pair of values  $(\tau_i, \pi_i)$  is visited before then
17:    Set  $\tau^* \leftarrow \tau_i$  and  $\pi^* \leftarrow \pi_i$ 
18:    BREAK
19:  end if
20:   $i \leftarrow i + 1$ 
21: end while

```

---

define the partial derivative of  $\tau$  and  $\pi$  as  $\mathcal{R}'(\tau)$  and  $\mathcal{R}'(\pi)$  respectively. Since  $\mathcal{R}(\tau, \pi)$  is in closed-form,  $\mathcal{R}'(\tau)$  and  $\mathcal{R}'(\pi)$  are also in closed-form and can be calculated numerically given any  $\pi$  and  $\tau$  respectively. Algorithm 2 is operated as follows: ① We take turns to calculate  $\mathcal{R}'(\tau)$  (Lines 3–8) and  $\mathcal{R}'(\pi)$  (Lines 9–14). When the gradient is greater than zero, implying that the objective function has the trend to increase, we decrease the value by 1 (Lines 5 and 11). When the gradient is less than zero, implying that the objective function has the trend to decrease, we increase the value by 1 (Lines 7 and 13). Due to constraints (29a) and (29b), we restrict the values of  $\tau$  and  $\pi$  to be equal or greater than 1. ② If the pair of value  $(\tau, \pi)$  is visited before (Lines 16–19), it means Algorithm 2 converges and  $(\tau, \pi)$  oscillates within a number of feasible value pairs (because  $\tau$  and  $\pi$  can only be integers). In this case, we find a local optimal pair of  $(\tau^*, \pi^*)$  and we can exit the algorithm.

## VI. EXPERIMENTAL RESULTS

In this section, we evaluate the convergence performance of HierMo compared with three typical categories of benchmark algorithms: ① three-tier FL without momentum (HierFAVG [18] and CFL [19]), ② two-tier FL with momentum (DOMO [24], FedADC [25], FedMom [20], SlowMo [21], FedNAG [22], and Mime [23]), and ③ two-tier FL without momentum (FedAvg [4]). For the two-tier benchmarks, we assume that the edge nodes do not exist and the workers are directly connected to the cloud. We then discuss the effects of  $\tau$  and  $\pi$  respectively and their joint effects. Afterwards, we explicitly quantify different levels of non-i.i.d. data and analyze their effects. Finally, we perform a trace-driven simulation of the three-tier hierarchical FL environment as if real-world hierarchical FL is implemented so that we can test the overall training time. Through this way, we verify that  $(\tau^*, \pi^*)$

derived in Section V leads to near-optimal performance in the realistic scenario.

### A. Experiment on Convergence of HierMo

1) *Experimental Setup:* We employ four real-world datasets including MNIST [29], CIFAR-10 [30], and ImageNet [28], [31] for image classification, and UCI-HAR [32] for human activity recognition. All training and testing samples are randomly shuffled and distributed to workers. Please note there is no restriction on how the data is distributed at different workers, therefore, the level of non-i.i.d. data distribution captured by  $\delta_{i,\ell}$  is different for each worker  $\{i, \ell\}$ . The training is run on a GPU tower server with 4 NVIDIA GeForce RTX 2080Ti GPUs.

We use five models including linear regression, logistic regression, CNN, VGG16, and ResNet18. The CNN model's structure is the classic one in [26], which has two  $5 \times 5$  convolutional layers with 32 and 64 channels respectively. In each convolutional layer,  $2 \times 2$  max pooling is used. The last three layers are fully connected layers with ReLU activation and softmax. The structure of VGG16 and ResNet18 can be found in [27], [28] respectively. We use mini-batch in all experiments, and the batch size is 64. We set the learning rate  $\eta = 0.01$ . Other hyper-parameters will be specified in each experiment.

In this experiment, we focus on the convergence performance (i.e., accuracy given the number of iterations) of different algorithms. We do not consider the real-world delay for now. The results do not depend on hardware but on the algorithm itself. Therefore, we can create several virtual machines within a single server to carry out the experiment. (Even if real-world hardware is used in the experiment, it will still give the same results.) The experiment on the optimization considering real-world delay will be discussed in Section VI-B.

2) *Performance Comparison:* In Table II, we compare the convergence performance of HierMo with benchmark algorithms. The numbers show the accuracy when different algorithms are run for  $T$  iterations. The experiment is conducted on linear regression, logistic regression, CNN, VGG16, and ResNet18. We set  $T = 1000$  (MNIST),  $T = 4000$  (UCI-HAR), or  $T = 10000$  (CIFAR10 and ImageNet),  $\gamma = 0.5$ ,  $\gamma_a = 0.5$ . There are 4 workers and 2 edge nodes with each edge node serving 2 workers (three-tier algorithm). There are 4 workers directly served by the cloud (two-tier algorithm). For two-tier algorithms, we set  $\tau = 20$  (convex model) or  $\tau = 40$  (non-convex model). For three-tier algorithms, we set  $\tau = 10, \pi = 2$  (convex model) or  $\tau = 20, \pi = 2$  (non-convex model). Please note that since  $\pi$  does not exist for two-tier algorithms, we set  $\tau$  value for two-tier algorithms equal to  $\tau\pi$  value for three-tier algorithms for a fair comparison. These hyper-parameters are typically used in existing works [8], [13], [14], [18], [22].

In all cases, HierMo outperforms all other benchmarks. This confirms that applying momentum on both worker-level and edge-level with three-tier architecture achieves the best performance.



TABLE II  
PERFORMANCE COMPARISON OF DIFFERENT FL ALGORITHMS (ACCURACY %).

	Linear on MNIST	Logistic on MNIST	CNN on MNIST	CNN on CIFAR10	VGG16 on CIFAR10	ResNet18 on ImageNet	CNN on UCI-HAR
HierMo	<b>85.97</b> $\pm$ 0.03	<b>89.23</b> $\pm$ 0.04	<b>96.13</b> $\pm$ 0.07	<b>64.18</b> $\pm$ 0.08	<b>90.06</b> $\pm$ 0.15	<b>69.64</b> $\pm$ 0.12	<b>88.36</b> $\pm$ 0.06
HierFAVG [18]	83.62 $\pm$ 0.03	87.00 $\pm$ 0.05	93.40 $\pm$ 0.07	38.46 $\pm$ 0.13	89.46 $\pm$ 0.12	68.63 $\pm$ 0.10	54.56 $\pm$ 0.11
CFL [19]	83.36 $\pm$ 0.04	86.98 $\pm$ 0.06	93.58 $\pm$ 0.06	38.79 $\pm$ 0.11	89.80 $\pm$ 0.11	68.87 $\pm$ 0.09	69.19 $\pm$ 0.09
DOMO [24]	85.79 $\pm$ 0.05	89.02 $\pm$ 0.05	95.90 $\pm$ 0.05	59.39 $\pm$ 0.07	88.53 $\pm$ 0.09	67.05 $\pm$ 0.10	88.15 $\pm$ 0.06
FedADC [25]	85.51 $\pm$ 0.04	88.18 $\pm$ 0.05	95.09 $\pm$ 0.07	56.00 $\pm$ 0.11	89.38 $\pm$ 0.08	67.76 $\pm$ 0.12	85.14 $\pm$ 0.09
FedMom [20]	84.84 $\pm$ 0.06	88.05 $\pm$ 0.05	94.74 $\pm$ 0.05	54.87 $\pm$ 0.07	88.03 $\pm$ 0.10	66.91 $\pm$ 0.11	84.69 $\pm$ 0.07
SlowMo [21]	84.82 $\pm$ 0.06	88.00 $\pm$ 0.06	94.88 $\pm$ 0.05	54.43 $\pm$ 0.06	88.47 $\pm$ 0.09	66.84 $\pm$ 0.09	83.03 $\pm$ 0.10
FedNAG [22]	84.97 $\pm$ 0.04	88.14 $\pm$ 0.05	95.04 $\pm$ 0.06	55.54 $\pm$ 0.09	88.33 $\pm$ 0.06	66.81 $\pm$ 0.14	84.69 $\pm$ 0.06
Mime [23]	84.41 $\pm$ 0.06	87.73 $\pm$ 0.06	93.89 $\pm$ 0.08	48.24 $\pm$ 0.15	81.76 $\pm$ 0.11	64.33 $\pm$ 0.21	76.75 $\pm$ 0.11
FedAvg [4]	83.57 $\pm$ 0.04	86.89 $\pm$ 0.05	93.31 $\pm$ 0.08	37.79 $\pm$ 0.19	88.27 $\pm$ 0.15	66.59 $\pm$ 0.09	53.31 $\pm$ 0.12

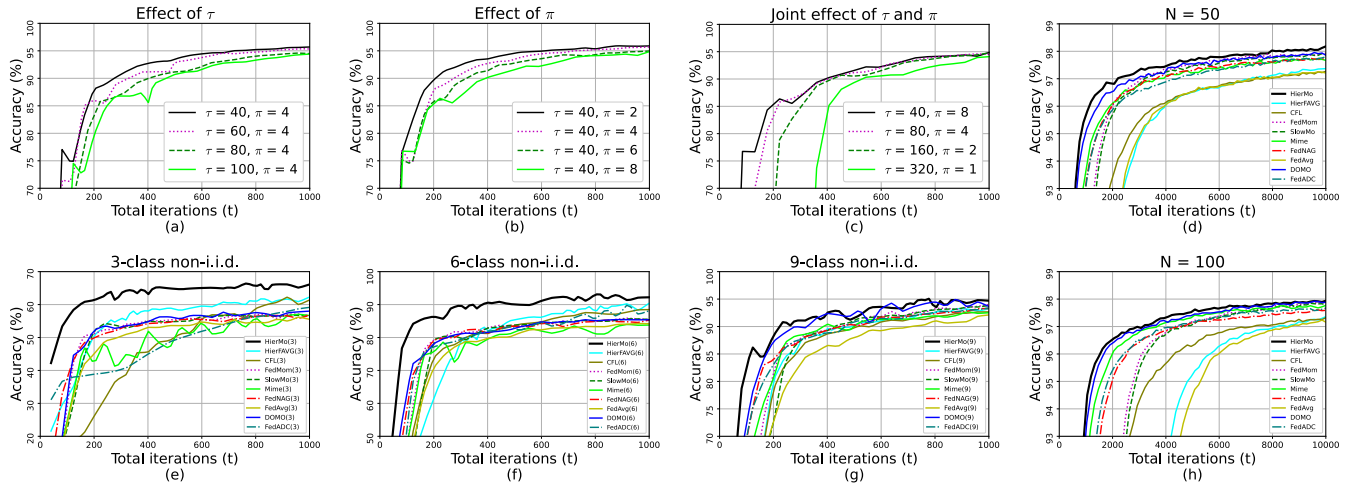


Fig. 3. (a)–(c): Accuracy comparison for HierMo under different settings of worker-edge aggregation period  $\tau$  and edge-cloud aggregation period  $\pi$  when CNN is trained on MNIST. (e)–(g): Accuracy comparison under 3-class (e), 6-class (f), and 9-class (g) non-i.i.d. data when CNN is trained on MNIST. (d) and (h): Accuracy comparison for large  $N$  ( $N = 50$  and  $N = 100$ ) when CNN is trained on MNIST.

Comparing HierMo with HierFAVG and CFL, we observe that HierMo  $>$  CFL  $>$  HierFAVG. (We use “ $>$ ” to indicate “is better than” for presentation convenience.) This verifies that the momentum can accelerate the convergence in three-tier architecture.

Comparing HierMo with DOMO and FedADC, we observe that HierMo  $>$  DOMO  $>$  FedADC. This verifies that when two types of momentum are applied, the three-tier architecture outperforms the two-tier architecture. This is because the additional edge aggregation can decrease the effect of data heterogeneity among workers under the same edge node, so as to improve the performance.

Comparing DOMO and FedADC with FedMom, SlowMo, FedNAG, and Mime, we observe that DOMO  $>$  FedADC  $>$  FedNAG  $>$  FedMom  $\approx$  SlowMo  $>$  Mime. This confirms that using combined worker momentum and aggregator momentum can accelerate the convergence compared with those using momentum only on workers or only on the aggregator. For worker momentum only or aggregator momentum only algorithms, we can still observe their acceleration compared with FedAvg. We also observe Mime may not perform well. Sometimes, it is even worse than FedAvg. This is because Mime uses the fixed momentum value in worker momentum update, where such

value can be refreshed only in the global aggregation phase. As a result, the momentum value may be stale, especially when  $\tau$  is as large as 40.

Comparing HierFAVG and CFL with two-tier momentum-based algorithms (DOMO, FedADC, FedMom, SlowMo, FedNAG, and Mime), we observe that for DNN, HierFAVG and CFL outperform two-tier momentum-based algorithms, while for convex model and CNN, the later is better. This shows that for complicated models, the three-tier architecture plays a more significant role to accelerate the convergence while for less complicated models, the momentum plays a more significant role to accelerate the convergence.

We also compare the training accuracy when more workers ( $N = 50$  and  $N = 100$ ) participate the training to demonstrate the cross-siloed FL [41] (typically up to one hundred participants). The results in Fig. 3(d) and (h) show the same trend as results in Table II.

3) *Effects of  $\tau$  and  $\pi$* : In Fig. 3, we evaluate the effects of  $\tau$  and  $\pi$ , and their joint effects. The curves in the figure show the accuracy when CNN is trained on MNIST. We set  $T = 1000$ ,  $\gamma = 0.5$ ,  $\gamma_a = 0.5$ . There are 16 workers and 4 edge nodes with each edge node serving 4 workers.

When  $\pi$  and  $\tau$  are fixed in Fig. 3(a) and Fig. 3(b) respectively, we observe that larger  $\tau$  or  $\pi$  lowers the performance.

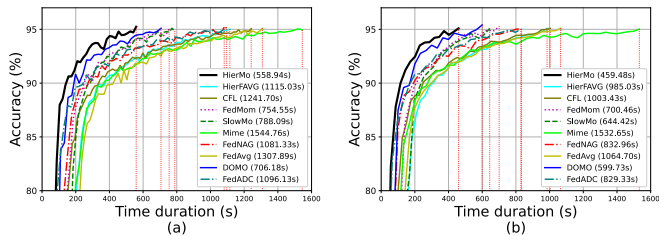


Fig. 4. Comparison of total training time to reach 0.95 accuracy under two different settings when CNN is trained on MNIST. The time to reach 0.95 accuracy is labeled in the legends. (a):  $\gamma = 0.5, \gamma_a = 0.5, \tau = 20$  (two-tier) or  $\tau = 10, \pi = 2$  (three-tier). (b):  $\gamma = 0.5, \gamma_a = 0.5, \tau = 40$  (two-tier) or  $\tau = 20, \pi = 2$  (three-tier).

This observation matches our expectation and verifies the result of Theorem 4 showing that the larger  $\tau$  or  $\pi$  leads to larger convergence upper bound.

When  $\tau \cdot \pi$  (the product of  $\tau$  and  $\pi$ ) is fixed in Fig. 3(c), we observe that smaller  $\tau$  (larger  $\pi$ ) leads to better performance. This shows that more frequent edge aggregation is more effective compared with more frequent cloud aggregation.

4) *Effects of non-i.i.d. data distribution:* In Fig. 3(e)–(g), we evaluate the effects of different levels of non-i.i.d. data distribution. We train CNN on MNIST with the setting  $\tau = 40$  (two-tier) or  $\tau = 20, \pi = 2$  (three-tier),  $N = 4, L = 2$ , and  $T = 1000$ . The curves show the training accuracy. To quantify the level of non-i.i.d. data distribution, we explicitly assign only  $x < 10$  out of 10 classes of data for each worker. (Each worker has data samples from a subset of classes.) The class of data is randomly allocated to each worker. Smaller  $x$  represents higher level of non-i.i.d. setting. We use *3-class non-i.i.d.*, *6-class non-i.i.d.*, and *9-class non-i.i.d.* to represent high, middle and low level of non-i.i.d. data respectively.

We observe that HierMo > HierFAVG > DOMO > FedADC > FedNAG > CFL > FedMom > SlowMo > Mime  $\approx$  FedAvg in most cases. This is consistent with the results in Table II, showing that HierMo outperforms all benchmarks under any levels of non-i.i.d. data distribution. We also observe higher level of non-i.i.d. setting decreases convergence performance for all algorithms. Specifically, HierMo achieves 66.11% accuracy for high level non-i.i.d. data, while achieving 92.21% accuracy and 94.70% accuracy for middle and low level non-i.i.d. data respectively. This matches our expectations where higher level of non-i.i.d. setting causes more data divergence that is denoted by larger  $\delta$ , and therefore lowers the accuracy.

## B. Experiment on Trace-driven simulation of HierMo

1) *Experimental Setup:* We emulate the real-world three-tier hierarchical FL environment to test the performance of HierMo in the following two aspects. ① To reach a target training accuracy (0.95), we compare the total training time of HierMo and benchmarks. ② For a given total training time  $\Psi$ , we compare the performance of HierMo under different  $(\tau, \pi)$  and verify that  $(\tau^*, \pi^*)$  derived by HierOPT is near optimal.

We train the CNN on MNIST in the GPU tower server to keep the trace of the sequence of iterations. We use real-world devices as workers (one laptop with Intel Core i3 M380 CPU, three Android phones: Nubia z17s with Qualcomm Snapdragon 835 CPU, Realme GT Neo with MTK Dimensity 1200 CPU, Redmi K30 Ultra with MTK Dimensity 1000+ CPU) to sample worker computation delays. We use Macbook Pro 2018 with Intel Core i7-8750H CPU as the edge node to sample the edge computation delays. The GPU tower server is regarded as the cloud server and the cloud computation delays are sampled on it. The workers are connected to a HUAWEI Honor router X2+ with 5GHz WiFi. The edge node is also connected to the router with a wired cable (1 Gbps Ethernet). The router is then connected to the public Internet.

The cloud server is connected to the Internet via another ISP’s access network. The worker communication delays are sampled between the workers and the edge node. The edge communication delays are sampled between the edge node and the server via the public Internet. Please note that for two-tier FL algorithms, since the workers directly communicate with the cloud, the worker-to-cloud communication delays are sampled as the delays from the devices to the server. We use the trace of the sequence of iterations and the sampled delays to figure out the overall delays as if the training process is conducted in real-world three-tier or two-tier FL environment. Please note that such approach to use a digital representation of physical objects to conduct the experiment is widely used in distributed systems, IoT, Industry 4.0, and machine learning applications [45], [46]. It can generate a convincing system performance evaluation without deploying physical devices.

2) *Total Training Time Comparison:* In Fig. 4, we compare the total training time of HierMo and benchmarks when CNN is trained on MNIST. The experiment is conducted under two settings: ①  $\gamma = 0.5, \gamma_a = 0.5, \tau = 20$  (two-tier) or  $\tau = 10, \pi = 2$  (three-tier) and ②  $\gamma = 0.5, \gamma_a = 0.5, \tau = 40$  (two-tier) or  $\tau = 20, \pi = 2$  (three-tier). There are 4 workers and 2 edge nodes with each edge node serving 2 workers (three-tier algorithm). There are 4 workers directly served by the cloud (two-tier algorithm).

We observe that to reach the accuracy 0.95, HierMo spends 558.94s under setting ① and 459.48s under setting ② while other benchmarks spend 706.18s–1544.76s under setting ① and 599.73s–1532.65s under setting ② respectively. This demonstrates that HierMo is efficient and decreases the total training time by 21–70% compared with the benchmarks.

3) *Performance of HierOPT:* In Fig. 5, we illustrate the performance of HierOPT. In this experiment, CNN is trained on MNIST and CIFAR10. We set  $\gamma = 0.5, \gamma_a = 0.5, \Psi = 400s$  or  $\Psi = 200s$  (MNIST), and  $\Psi = 6000s$  or  $\Psi = 3000s$  (CIFAR10). There are 16 workers and 4 edge nodes with each edge node serving 4 workers. All constants in the objective function (29) can be sampled in advance of the training process [42], [44].

We show the accuracy under different pairs of  $(\tau, \pi)$  and flag  $(\tau^*, \pi^*)$  derived by HierOPT. The darker color in the chromatography indicates a higher training accuracy. The red cross indicates the derived  $(\tau^*, \pi^*)$  by HierOPT. We observe that in all figures, HierOPT can find near-optimal solutions.

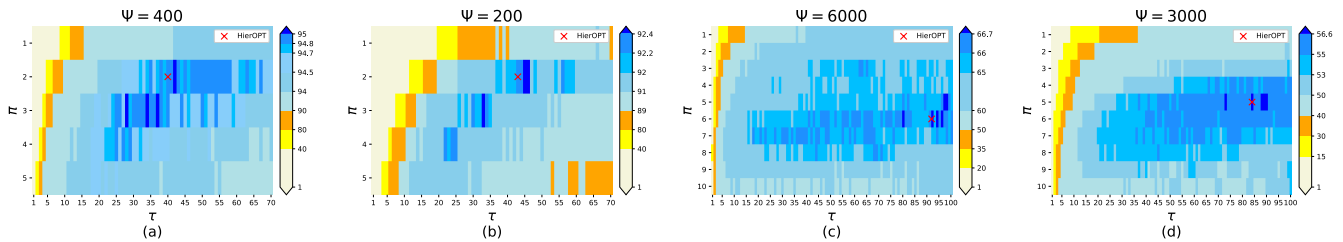


Fig. 5. Accuracy comparison for HierMo with derived pair of  $(\tau^*, \pi^*)$  by HierOPT (red cross) and different pairs of  $(\tau, \pi)$  under limited total training time  $\Psi$ . The darker color indicates the higher training accuracy (in %). (a):  $\Psi = 400$ s on MNIST. (b):  $\Psi = 200$ s on MNIST. (c):  $\Psi = 6000$ s on CIFAR10. (d):  $\Psi = 3000$ s on CIFAR10.

In Fig. 5(a), when  $\Psi = 400$ s, the optimal accuracy is 95.05%, with optimal  $(\tau, \pi) = (42, 2)$ , while HierOPT finds  $(\tau^*, \pi^*) = (40, 2)$ , with accuracy 94.82%, only a 0.23% gap from the optimum. In Fig. 5(b), when  $\Psi = 200$ s, the optimal accuracy is 92.52%, with optimal  $(\tau, \pi) = (46, 2)$ , while HierOPT finds  $(\tau^*, \pi^*) = (43, 2)$ , with accuracy 92.23%, only a 0.29% gap from the optimum. For CIFAR10, HierOPT can still find the near-optimal  $(\tau^*, \pi^*)$ , with only 0.04% (67.09% to 67.05%) and 0.16% (56.82% to 56.66%) gap from the real-world optimum, when  $\Psi = 6000$ s and  $\Psi = 3000$ s respectively.

## VII. CONCLUSION

In this paper, we propose HierMo, a three-tier hierarchical FL algorithm that applies momentum to accelerate convergence. We provide convergence analysis for HierMo, showing that it converges with a rate of  $\mathcal{O}(\frac{1}{T})$  for smooth non-convex problems under non-i.i.d. data. In the analysis, we develop a new two-level virtual update (edge and cloud) method to characterize the multi-time cross-two-tier momentum interaction and the cross-three-tier momentum interaction. The performance gain of momentum is also quantified. We also propose HierOPT to derive a near-optimal setting of worker-edge and edge-cloud aggregation periods  $(\tau, \pi)$  under a limited total training time. We verify that HierMo outperforms existing mainstream benchmarks under a wide range of settings. In addition, HierOPT can achieve a near-optimal performance when we test HierMo under different values of  $(\tau, \pi)$ .

## APPENDIX

### A. Proof of Theorem 1

1) *Equivalent Update*: First, we define  $\mathbf{v}_{i,\ell}^t \triangleq \mathbf{y}_{i,\ell}^t - \mathbf{y}_{i,\ell}^{t-1}$  with  $\mathbf{v}_{i,\ell}^0 = \mathbf{0}$  for all  $i, \ell$ . We can obtain  $\mathbf{x}_{i,\ell}^{t-1} = \mathbf{y}_{i,\ell}^{t-1} + \gamma \mathbf{v}_{i,\ell}^{t-1}$ . The worker momentum/model update in Lines 5–6 in Algorithm 1 can then be equivalently written as

$$\mathbf{v}_{i,\ell}^t \leftarrow \gamma \mathbf{v}_{i,\ell}^{t-1} - \eta \nabla F_{i,\ell}(\mathbf{x}_{i,\ell}^{t-1}), \quad (30)$$

$$\mathbf{x}_{i,\ell}^t \leftarrow \mathbf{x}_{i,\ell}^{t-1} + \gamma \mathbf{v}_{i,\ell}^t - \eta \nabla F_{i,\ell}(\mathbf{x}_{i,\ell}^{t-1}). \quad (31)$$

The aggregated value  $\mathbf{v}_\ell^t$  and the intermediate value  $\mathbf{x}_{\ell-}^t$  can also be equivalently written as

$$\mathbf{v}_\ell^t \leftarrow \sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} \mathbf{v}_{i,\ell}^t, \quad \mathbf{x}_{\ell-}^t \leftarrow \sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} \mathbf{x}_{i,\ell}^t. \quad (32)$$

Similarly, the edge and cloud virtual updates (8)–(9) and (12)–(13) can be equivalently written as

$$\begin{aligned} \mathbf{v}_{[k],\ell}^t &\leftarrow \gamma \mathbf{v}_{[k],\ell}^{t-1} - \eta \nabla F_\ell(\mathbf{x}_{[k],\ell}^{t-1}), \\ \mathbf{x}_{[k],\ell}^t &\leftarrow \mathbf{x}_{[k],\ell}^{t-1} + \gamma \mathbf{v}_{[k],\ell}^t - \eta \nabla F_\ell(\mathbf{x}_{[k],\ell}^{t-1}), \end{aligned} \quad (33)$$

$$\begin{aligned} \mathbf{v}_{\{p\}}^t &\leftarrow \gamma \mathbf{v}_{\{p\}}^{t-1} - \eta \nabla F(\mathbf{x}_{\{p\}}^{t-1}), \\ \mathbf{x}_{\{p\}}^t &\leftarrow \mathbf{x}_{\{p\}}^{t-1} + \gamma \mathbf{v}_{\{p\}}^t - \eta \nabla F(\mathbf{x}_{\{p\}}^{t-1}). \end{aligned} \quad (34)$$

We employ the above equivalent update format (30)–(34) to complete the proof in the rest of the Appendix.

2) *Constant Definition*: We define the constants as follows, which are more conveniently used in the rest of the Appendix.

$$\begin{aligned} A &\triangleq \frac{(1 + \eta\beta)(1 + \gamma) + \sqrt{(1 + \eta\beta)^2(1 + \gamma)^2 - 4\gamma(1 + \eta\beta)}}{2\gamma}, \\ B &\triangleq \frac{(1 + \eta\beta)(1 + \gamma) - \sqrt{(1 + \eta\beta)^2(1 + \gamma)^2 - 4\gamma(1 + \eta\beta)}}{2\gamma}, \\ I &\triangleq \frac{\gamma A + A - 1}{(A - B)(\gamma A - 1)}, J \triangleq \frac{\gamma B + B - 1}{(A - B)(1 - \gamma B)}, \\ U &\triangleq \frac{\frac{1 + \eta\beta + \eta\beta\gamma}{\gamma} - B}{A - B} = \frac{A - 1}{A - B}, V \triangleq \frac{A - \frac{1 + \eta\beta + \eta\beta\gamma}{\gamma}}{A - B} = \frac{1 - B}{A - B}. \end{aligned}$$

3) *Subscript  $\ell$* : Since Theorem 1 focuses on a specific edge node  $\ell$ , for presentation convenience, in the proofs of Theorem 1 (including Lemmas 1–3), we ignore all subscript  $\ell$ . We use  $\mathbf{x}_i, \mathbf{v}_i, \mathbf{x}, \mathbf{v}, \mathbf{x}_{[k]}, \mathbf{v}_{[k]}, F_i, F, D_i, D, \delta_i, \delta$ , and  $C$  to represent  $\mathbf{x}_{i,\ell}, \mathbf{v}_{i,\ell}, \mathbf{x}_{\ell-}, \mathbf{v}_\ell, \mathbf{x}_{[k],\ell}, \mathbf{v}_{[k],\ell}, F_{i,\ell}, F_\ell, D_{i,\ell}, D_\ell, \delta_{i,\ell}, \delta_\ell$ , and  $C_\ell$  respectively. Please note that in the proofs of the theorems other than Theorem 1, we do not ignore subscript  $\ell$ .

4) *Prerequisite Lemmas for the Proof of Theorem 1*: To prove Theorem 1, the progress mainly includes four steps. (1) We first introduce an important equality in Lemma 1, which will be used to prove Lemma 2. (2) We bound  $\|\mathbf{x}_i^t - \mathbf{x}_{[k]}^t\|$  in Lemma 2 based on Lemma 1. (3) Based on the result of Lemma 2, we then bound  $\|\mathbf{v}^t - \mathbf{v}_{[k]}^t\|$  in Lemma 3. Please note that the proofs of Lemmas 1–3 are in Appendix B–D respectively. (4) Finally, based on the result of Lemma 3, we bound  $\|\mathbf{x}^t - \mathbf{x}_{[k]}^t\|$ , which concludes Theorem 1.

**Lemma 1.** *Given*

$$a_t = \frac{\delta_i}{\beta} \left( \frac{1+\eta\beta+\eta\beta\gamma}{\gamma} \frac{A^t - B^t}{A-B} - \frac{1+\eta\beta+\eta\beta\gamma}{\gamma} \frac{A^t - B^t}{A-B} \right), \quad (35)$$

$$A + B = \frac{1 + \eta\beta + \eta\beta\gamma + \gamma}{\gamma} = \frac{(1 + \eta\beta)(1 + \gamma)}{\gamma},$$

$$AB = \frac{1 + \eta\beta}{\gamma}, \quad (36)$$

where  $t = 0, 1, 2, \dots, 0 < \gamma < 1, \eta\beta > 0$ , we have  $(1 + \eta\beta)a_{t-1} + \eta\beta\gamma \sum_{i=0}^{t-1} a_i = \gamma a_t$ .

**Lemma 2.** *For any interval  $[k], \forall t \in [(k-1)\tau, k\tau]$ , we have  $\|\mathbf{x}_i^t - \mathbf{x}_{[k]}^t\| \leq f_i(t - (k-1)\tau)$ , where we define the function  $f_i(x)$  as  $f_i(x) \triangleq \frac{\delta_i}{\beta}(\gamma^x(UA^x + VB^x) - 1)$  and the function  $u(x)$  as  $u(x) \triangleq \gamma^x(UA^x + VB^x) - 1$ .*

**Lemma 3.** *For any interval  $[k], \forall t \in [(k-1)\tau, k\tau]$ , we have:*

$$\|\mathbf{v}^t - \mathbf{v}_{[k]}^t\| \leq \eta\delta \left( \frac{U(\gamma A)^{t_0}}{\gamma(A-1)} + \frac{V(\gamma B)^{t_0}}{\gamma(B-1)} - \frac{\gamma^{t_0} - 1}{\gamma - 1} \right),$$

where  $t_0 = t - (k-1)\tau$ .

5) *Derivation of Theorem 1:* From (31) and (32), we have

$$\mathbf{x}^t = \mathbf{x}^{t-1} + \gamma\mathbf{v}^t - \eta \frac{\sum_{i=1}^C D_i \nabla F_i(\mathbf{x}_i^{t-1})}{D}. \quad (37)$$

From (33) and (37), and according to  $\beta$ -smoothness, Lemma 2, the definition of  $f_i(x)$  and  $u(x)$ , and Assumption 3, we have

$$\begin{aligned} \|\mathbf{x}^t - \mathbf{x}_{[k]}^t\| &= \|\mathbf{x}^{t-1} + \gamma\mathbf{v}^t - \eta \frac{\sum_{i=1}^C D_i \nabla F_i(\mathbf{x}_i^{t-1})}{D} \\ &\quad - \mathbf{x}_{[k]}^{t-1} - \gamma\mathbf{v}_{[k]}^t + \eta \nabla F(\mathbf{x}_{[k]}^{t-1})\| \\ &\leq \|\mathbf{x}^{t-1} - \mathbf{x}_{[k]}^{t-1}\| + \gamma\|\mathbf{v}^t - \mathbf{v}_{[k]}^t\| + \eta\delta u(t-1 - (k-1)\tau). \end{aligned}$$

Then, according to Lemma 3, we have

$$\begin{aligned} &\|\mathbf{x}^t - \mathbf{x}_{[k]}^t\| - \|\mathbf{x}^{t-1} - \mathbf{x}_{[k]}^{t-1}\| \\ &\leq \gamma\eta\delta \left( \frac{U(\gamma A)^{t_0}}{\gamma(A-1)} + \frac{V(\gamma B)^{t_0}}{\gamma(B-1)} - \frac{\gamma^{t_0} - 1}{\gamma - 1} \right) \\ &\quad + \eta\delta(\gamma^{t_0-1}(UA^{t_0-1} + VB^{t_0-1}) - 1) \end{aligned} \quad (38)$$

$$\begin{aligned} &= \eta\delta \left( \frac{U(\gamma A)^{t_0-1}}{A-1}(\gamma A + A - 1) + \frac{V(\gamma B)^{t_0-1}}{B-1}(\gamma B + B - 1) \right. \\ &\quad \left. - \frac{\gamma^{t_0+1} - 1}{\gamma - 1} \right). \end{aligned} \quad (39)$$

When  $t = (k-1)\tau$ , we have  $\|\mathbf{x}^t - \mathbf{x}_{[k]}^t\| = 0$ . When  $t \in ((k-1)\tau, k\tau]$ , we sum up (39) for  $t, t-1, \dots, (k-1)\tau + 1$ , leading to

$$\begin{aligned} \|\mathbf{x}^t - \mathbf{x}_{[k]}^t\| &\leq \sum_{x=1}^{t_0} \eta\delta \left( \frac{U(\gamma A)^{x-1}}{A-1}(\gamma A + A - 1) \right. \\ &\quad \left. + \frac{V(\gamma B)^{x-1}}{B-1}(\gamma B + B - 1) - \frac{\gamma^{x+1} - 1}{\gamma - 1} \right) \\ &= \eta\delta \left[ I((\gamma A)^{t_0} - 1) + J((\gamma B)^{t_0} - 1) \right. \\ &\quad \left. - \frac{\gamma^2(\gamma^{t_0} - 1) - (\gamma - 1)t_0}{(\gamma - 1)^2} \right] \\ &= \eta\delta \left[ I(\gamma A)^{t_0} + J(\gamma B)^{t_0} - \frac{1}{\eta\beta} - \frac{\gamma^2(\gamma^{t_0} - 1) - (\gamma - 1)t_0}{(\gamma - 1)^2} \right] \\ &= h(t_0), \end{aligned}$$

where  $I = \frac{\gamma A + A - 1}{(A - B)(\gamma A - 1)}$  and  $J = \frac{\gamma B + B - 1}{(A - B)(1 - \gamma B)}$  (as defined before).  $I + J = \frac{1}{\eta\beta}$ .  $t_0 = t - (k-1)\tau$ . We complete the proof of Theorem 1.

**B. Proof of Lemma 1**

Based on the definitions of  $U, V$ , and  $a_t$ , we have  $a_t = \frac{\delta_i}{\beta}(UA^t + VB^t)$ . According to the inverse theorem of Vieta's formulas, we have

$$\gamma x^2 - (1 + \eta\beta + \eta\beta\gamma + \gamma)x + \eta\beta + 1 = 0, \quad (40)$$

where  $x$  values are the roots of the quadratic equation. The discriminant of the quadratic equation is positive.

$$\begin{aligned} \Delta &= (1 + \eta\beta + \eta\beta\gamma + \gamma)^2 - 4(1 + \eta\beta)\gamma \\ &> (1 + \eta\beta + \gamma)^2 - 4(1 + \eta\beta)\gamma = ((1 + \eta\beta) - \gamma)^2 > 0. \end{aligned}$$

Thus, the roots of (40) can be expressed as  $A$  and  $B$ . Therefore, we can obtain

$$\begin{aligned} &(1 + \eta\beta)a_{t-1} + \eta\beta\gamma \sum_{i=0}^{t-1} a_i - \gamma a_t \\ &= (1 + \eta\beta) \frac{\delta_i}{\beta} (UA^{t-1} + VB^{t-1}) + \eta\beta\gamma \frac{\delta_i}{\beta} U \frac{A^t - 1}{A - 1} \\ &\quad + \eta\beta\gamma \frac{\delta_i}{\beta} V \frac{B^t - 1}{B - 1} - \gamma \frac{\delta_i}{\beta} UA^t - \gamma \frac{\delta_i}{\beta} VB^t \\ &= \frac{\delta_i}{\beta} \left[ \frac{A^{t-1}U}{1 - A} (\gamma A^2 - (1 + \eta\beta + \eta\beta\gamma + \gamma)A + 1 + \eta\beta) \right. \\ &\quad \left. + \frac{B^{t-1}V}{1 - B} (\gamma B^2 - (1 + \eta\beta + \eta\beta\gamma + \gamma)B + 1 + \eta\beta) \right] \\ &\quad - \frac{\delta_i}{\beta} \eta\beta\gamma \left( \frac{U}{A-1} + \frac{V}{B-1} \right) \\ &= 0 - \eta\delta_i\gamma \left( \frac{U}{A-1} + \frac{V}{B-1} \right) = 0. \end{aligned}$$

We complete the proof of Lemma 1.

**C. Proof of Lemma 2**

To prove Lemma 2, (1) we first bound the gap of  $\|\mathbf{v}_i^t - \mathbf{v}_{[k]}^t\|$ ; (2) then we bound the gap of  $\|\mathbf{x}_i^t - \mathbf{x}_{[k]}^t\|$ , which concludes Lemma 2.

When  $t = (k-1)\tau$ , we know  $\mathbf{x}_i^t = \mathbf{x}^t = \mathbf{x}_{[k]}^t$  by the definition of  $\mathbf{x}_{[k]}^t$  and the aggregation rules. Hence, we have  $\|\mathbf{x}_i^t - \mathbf{x}_{[k]}^t\| = 0$ . Meanwhile, when  $t = (k-1)\tau$ , we have  $x = 0$  and  $f_i(0) = 0$  (Lemma 2 holds).

When  $t \in ((k-1)\tau, k\tau]$ , we bound the momentum gap

$$\begin{aligned} &\|\mathbf{v}_i^t - \mathbf{v}_{[k]}^t\| \\ &= \|\gamma\mathbf{v}_i^{t-1} - \eta\nabla F_i(\mathbf{x}_i^{t-1}) - (\gamma\mathbf{v}_{[k]}^{t-1} - \eta\nabla F(\mathbf{x}_{[k]}^{t-1}))\| \\ &= \|\gamma(\mathbf{v}_i^{t-1} - \mathbf{v}_{[k]}^{t-1}) - \eta[\nabla F_i(\mathbf{x}_i^{t-1}) - \nabla F_i(\mathbf{x}_{[k]}^{t-1}) \\ &\quad + \nabla F_i(\mathbf{x}_{[k]}^{t-1}) - \nabla F(\mathbf{x}_{[k]}^{t-1})]\| \\ &\stackrel{(a)}{\leq} \gamma\|\mathbf{v}_i^{t-1} - \mathbf{v}_{[k]}^{t-1}\| + \eta\|\nabla F_i(\mathbf{x}_i^{t-1}) - \nabla F_i(\mathbf{x}_{[k]}^{t-1})\| \\ &\quad + \eta\|\nabla F_i(\mathbf{x}_{[k]}^{t-1}) - \nabla F(\mathbf{x}_{[k]}^{t-1})\| \\ &\stackrel{(b)}{\leq} \gamma\|\mathbf{v}_i^{t-1} - \mathbf{v}_{[k]}^{t-1}\| + \eta\beta\|\mathbf{x}_i^{t-1} - \mathbf{x}_{[k]}^{t-1}\| + \eta\delta_i, \end{aligned} \quad (41)$$

where (a) is from triangle inequality and (b) is from  $\beta$ -smoothness and Assumption 3.

We use  $\gamma^0, \gamma^1, \dots, \gamma^{t-(k-1)\tau-1}$  as multipliers to multiply (41) when  $t, t-1, \dots, (k-1)\tau+1$ , respectively.

$$\begin{aligned} & \|\mathbf{v}_i^t - \mathbf{v}_{[k]}^t\| \leq \gamma \|\mathbf{v}_i^{t-1} - \mathbf{v}_{[k]}^{t-1}\| + \eta\beta \|\mathbf{x}_i^{t-1} - \mathbf{x}_{[k]}^{t-1}\| + \eta\delta_i, \\ & \gamma \|\mathbf{v}_i^{t-1} - \mathbf{v}_{[k]}^{t-1}\| \leq \gamma(\gamma \|\mathbf{v}_i^{t-2} - \mathbf{v}_{[k]}^{t-2}\| + \eta\beta \|\mathbf{x}_i^{t-2} - \mathbf{x}_{[k]}^{t-2}\| + \eta\delta_i), \\ & \dots \\ & \gamma^{t-(k-1)\tau-1} \|\mathbf{v}_i^{(k-1)\tau+1} - \mathbf{v}_{[k]}^{(k-1)\tau+1}\| \leq \gamma^{t-(k-1)\tau-1} \\ & (\gamma \|\mathbf{v}_i^{(k-1)\tau} - \mathbf{v}_{[k]}^{(k-1)\tau}\| + \eta\beta \|\mathbf{x}_i^{(k-1)\tau} - \mathbf{x}_{[k]}^{(k-1)\tau}\| + \eta\delta_i). \end{aligned}$$

For convenience, we define  $G_i(t) \triangleq \|\mathbf{x}_i^t - \mathbf{x}_{[k]}^t\|$ . Summing up all of the above inequalities with respect to  $b \in [1, t-(k-1)\tau]$ , we have

$$\begin{aligned} \|\mathbf{v}_i^t - \mathbf{v}_{[k]}^t\| & \leq \eta\beta \sum_{b=1}^{t-(k-1)\tau} \gamma^{b-1} G_i(t-b) + \eta\delta_i \sum_{b=1}^{t-(k-1)\tau} \gamma^{b-1} \\ & + \gamma^{t-(k-1)\tau} \|\mathbf{v}_i^{(k-1)\tau} - \mathbf{v}_{[k]}^{(k-1)\tau}\|. \end{aligned}$$

When  $t = (k-1)\tau$ , we know that  $\mathbf{v}_i^t = \mathbf{v}^t = \mathbf{v}_{[k]}^t$  by the definition of  $\mathbf{v}_{[k]}^t$  and aggregation rules. Then we have  $\|\mathbf{v}_i^{(k-1)\tau} - \mathbf{v}_{[k]}^{(k-1)\tau}\| = 0$ , so that the last term of above inequality is zero and

$$\|\mathbf{v}_i^t - \mathbf{v}_{[k]}^t\| \leq \eta\beta \sum_{b=1}^{t-(k-1)\tau} \gamma^{b-1} G_i(t-b) + \eta\delta_i \sum_{b=1}^{t-(k-1)\tau} \gamma^{b-1}. \quad (42)$$

Now, we can bound the gap between  $\mathbf{x}_i^t$  and  $\mathbf{x}_{[k]}^t$ . When  $t \in ((k-1)\tau, k\tau]$ , we have

$$\begin{aligned} & \|\mathbf{x}_i^t - \mathbf{x}_{[k]}^t\| \\ & \stackrel{(a)}{=} \|\mathbf{x}_i^{t-1} + \gamma \mathbf{v}_i^t - \eta \nabla F_i(\mathbf{x}_i^{t-1}) - (\mathbf{x}_{[k]}^{t-1} + \gamma \mathbf{v}_{[k]}^t - \eta \nabla F(\mathbf{x}_{[k]}^{t-1}))\| \\ & = \|\mathbf{x}_i^{t-1} - \mathbf{x}_{[k]}^{t-1} + \gamma(\mathbf{v}_i^t - \mathbf{v}_{[k]}^t) - \eta[\nabla F_i(\mathbf{x}_i^{t-1}) - \nabla F(\mathbf{x}_{[k]}^{t-1})] \\ & \quad + \nabla F_i(\mathbf{x}_i^{t-1}) - \nabla F(\mathbf{x}_{[k]}^{t-1})\| \\ & \stackrel{(b)}{\leq} \|\mathbf{x}_i^{t-1} - \mathbf{x}_{[k]}^{t-1}\| + \gamma \|\mathbf{v}_i^t - \mathbf{v}_{[k]}^t\| + \eta\beta \|\mathbf{x}_i^{t-1} - \mathbf{x}_{[k]}^{t-1}\| + \eta\delta_i \\ & = (\eta\beta + 1) \|\mathbf{x}_i^{t-1} - \mathbf{x}_{[k]}^{t-1}\| + \gamma \|\mathbf{v}_i^t - \mathbf{v}_{[k]}^t\| + \eta\delta_i, \end{aligned} \quad (43)$$

where (a) is from (31) and (33), and (b) is from triangle inequality,  $\beta$ -smoothness, and Definition 3.

Substituting (42) into (43) and using  $G_i(t)$  to denote  $\|\mathbf{x}_i^t - \mathbf{x}_{[k]}^t\|$  for  $t, t-1, \dots, (k-1)\tau+1$ , we have

$$\begin{aligned} G_i(t) & \leq (\eta\beta + 1) G_i^{t-1} + \eta\beta\gamma \sum_{b=1}^{t-(k-1)\tau} \gamma^{b-1} G_i(t-b) \\ & \quad + \eta\delta_i\gamma \sum_{b=1}^{t-(k-1)\tau} \gamma^{b-1} + \eta\delta_i \\ & = (\eta\beta + 1) G_i^{t-1} + \eta\beta\gamma \sum_{b=1}^{t-(k-1)\tau} \gamma^{b-1} G_i(t-b) \\ & \quad + \eta\delta_i \sum_{b=0}^{t-(k-1)\tau} \gamma^b. \end{aligned} \quad (44)$$

For convenience, we define  $g_i(x) \triangleq \frac{\delta_i}{\beta}(UA^x + VB^x)$ . We have  $f_i(x) = \gamma^x g_i(x) - \frac{\delta_i}{\beta}$ .

We use induction to prove  $G_i(t) \leq f_i(t - (k-1)\tau)$ ,  $\forall t \in [(k-1)\tau, k\tau]$ . First of all, we know that it is true when  $t = (k-1)\tau$  because  $G_i((k-1)\tau) = f_i(0)$ . Then, we assume that  $G_i(c) \leq f_i(c - (k-1)\tau)$  holds for all  $c \in [(k-1)\tau, t]$ , and we show it also holds for  $t$ .

$$\begin{aligned} & G_i(t) \\ & \stackrel{(a)}{\leq} (\eta\beta + 1) f_i(t - 1 - (k-1)\tau) \\ & \quad + \eta\beta \sum_{b=1}^{t-(k-1)\tau} \gamma^b f_i(t - b - (k-1)\tau) + \eta\delta_i \sum_{b=0}^{t-(k-1)\tau} \gamma^b \\ & \stackrel{(b)}{=} (\eta\beta + 1) \left( \gamma^{t-1-(k-1)\tau} g_i(t - 1 - (k-1)\tau) - \frac{\delta_i}{\beta} \right) \\ & \quad + \eta\beta \sum_{b=1}^{t-(k-1)\tau} \left( \gamma^{t-(k-1)\tau} g_i(t - b - (k-1)\tau) - \gamma^b \frac{\delta_i}{\beta} \right) \\ & \quad + \eta\delta_i \sum_{b=0}^{t-(k-1)\tau} \gamma^b \\ & = \gamma^{t-1-(k-1)\tau} ((\eta\beta + 1) g_i(t - 1 - (k-1)\tau) \\ & \quad + \eta\beta\gamma \sum_{b=1}^{t-(k-1)\tau} g_i(t - b - (k-1)\tau)) - \frac{\delta_i}{\beta} \\ & \stackrel{(c)}{=} \gamma^{t-(k-1)\tau} g_i(t - (k-1)\tau) - \frac{\delta_i}{\beta} = f_i(t - (k-1)\tau), \end{aligned}$$

where (a) is from (44), (b) is from definition of  $f_i(x)$ , and (c) is from Lemma 1 and  $G_i(t) = a_t$ . We complete the proof of Lemma 2.

#### D. Proof of Lemma 3

Based on the definition of  $u(x)$  in Lemma 2, we get  $f_i(x) = \frac{\delta_i}{\beta} u(x)$ . From (31) and (32), we have

$$\mathbf{v}^t = \gamma \mathbf{v}^{t-1} - \eta \frac{\sum_{i=1}^C D_i \nabla F_i(\mathbf{x}_i^{t-1})}{D}. \quad (45)$$

For  $t \in ((k-1)\tau, k\tau]$ , we have

$$\begin{aligned} & \|\mathbf{v}^t - \mathbf{v}_{[k]}^t\| \\ & \stackrel{(a)}{=} \|\gamma \mathbf{v}^{t-1} - \eta \frac{\sum_{i=1}^C D_i \nabla F_i(\mathbf{x}_i^{t-1})}{D} - \gamma \mathbf{v}_{[k]}^{t-1} + \eta \nabla F(\mathbf{x}_{[k]}^{t-1})\| \\ & \leq \gamma \|\mathbf{v}^{t-1} - \mathbf{v}_{[k]}^{t-1}\| + \eta \frac{\sum_{i=1}^C D_i \|\nabla F_i(\mathbf{x}_i^{t-1}) - \nabla F(\mathbf{x}_{[k]}^{t-1})\|}{D} \\ & \stackrel{(b)}{\leq} \gamma \|\mathbf{v}^{t-1} - \mathbf{v}_{[k]}^{t-1}\| + \eta\beta \frac{\sum_{i=1}^C D_i f_i(t - 1 - (k-1)\tau)}{D} \\ & \stackrel{(c)}{=} \gamma \|\mathbf{v}^{t-1} - \mathbf{v}_{[k]}^{t-1}\| + \eta\delta u(t - 1 - (k-1)\tau), \end{aligned} \quad (46)$$

where (a) is from (45) and (33); (b) is from  $\beta$ -smoothness and Lemma 2; and (c) is from definition of  $f_i(x)$  and Assumption 3.

We use  $\gamma^0, \gamma^1, \dots, \gamma^{t-(k-1)\tau-1}$  as multipliers to multiply (46) when  $t, t-1, \dots, (k-1)\tau+1$ , respectively.

$$\begin{aligned} \|\mathbf{v}^t - \mathbf{v}_{[k]}^t\| &\leq \gamma \|\mathbf{v}^{t-1} - \mathbf{v}_{[k]}^{t-1}\| + \eta \delta u(t-1 - (k-1)\tau), \\ \gamma \|\mathbf{v}^{t-1} - \mathbf{v}_{[k]}^{t-1}\| &\leq \gamma^2 (\|\mathbf{v}^{t-2} - \mathbf{v}_{[k]}^{t-2}\| + \gamma \eta \delta u(t-2 - (k-1)\tau)), \\ &\dots \\ \gamma^{t-(k-1)\tau-1} \|\mathbf{v}^{(k-1)\tau+1} - \mathbf{v}_{[k]}^{(k-1)\tau+1}\| \\ &\leq \gamma^{t-(k-1)\tau} \|\mathbf{v}^{(k-1)\tau} - \mathbf{v}_{[k]}^{(k-1)\tau}\| + \gamma^{t-1-(k-1)\tau} \eta \delta u(0). \end{aligned}$$

Summing up all of the above inequalities, and according to  $\|\mathbf{v}^{(k-1)\tau} - \mathbf{v}_{[k]}^{(k-1)\tau}\| = 0$ , we have

$$\begin{aligned} \|\mathbf{v}^t - \mathbf{v}_{[k]}^t\| &\leq \eta \delta \sum_{b=1}^{t-(k-1)\tau} \gamma^{t-b-(k-1)\tau} u(b-1) \quad (47) \\ &= \eta \delta \left( \gamma^{t-1-(k-1)\tau} U \sum_{b=1}^{t-(k-1)\tau} A^{b-1} \right. \\ &\quad \left. + \gamma^{t-1-(k-1)\tau} V \sum_{b=1}^{t-(k-1)\tau} B^{b-1} - \sum_{b=1}^{t-(k-1)\tau} \gamma^{b-1} \right) \\ &= \eta \delta \left( \gamma^{t_0-1} U \frac{A^{t_0} - 1}{A-1} + \gamma^{t_0-1} V \frac{B^{t_0} - 1}{B-1} - \frac{\gamma^{t_0} - 1}{\gamma - 1} \right) \\ &= \eta \delta \left( \frac{U(\gamma A)^{t_0}}{\gamma(A-1)} + \frac{V(\gamma B)^{t_0}}{\gamma(B-1)} - \frac{\gamma^{t_0} - 1}{\gamma - 1} \right) \\ &\quad - \eta \delta \gamma^{t_0-1} \left( \frac{U}{A-1} + \frac{V}{B-1} \right) \\ &= \eta \delta \left( \frac{U(\gamma A)^{t_0}}{\gamma(A-1)} + \frac{V(\gamma B)^{t_0}}{\gamma(B-1)} - \frac{\gamma^{t_0} - 1}{\gamma - 1} \right) \quad (48) \end{aligned}$$

where  $t_0 = t - (k-1)\tau$ . We complete the proof of Lemma 3.

### E. Proof of Theorem 2

Based on the edge momentum update rules in Lines 10–11 in Algorithm 1, and (31) we have

$$\begin{aligned} \mathbf{x}_{\ell+}^{k\tau} - \mathbf{x}_{\ell-}^{k\tau} &= \gamma_a (\mathbf{x}_{\ell-}^{k\tau} - \mathbf{x}_{\ell-}^{(k-1)\tau}) = \gamma_a \sum_{t=(k-1)\tau}^{k\tau-1} (\mathbf{x}_{\ell-}^{t+1} - \mathbf{x}_{\ell-}^t) \\ &= \gamma_a \sum_{t=(k-1)\tau}^{k\tau-1} \sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} (\mathbf{x}_{i,\ell}^{t+1} - \mathbf{x}_{i,\ell}^t) \\ &= \gamma_a \sum_{t=(k-1)\tau}^{k\tau-1} \sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} (\gamma^2 \mathbf{v}_{i,\ell}^t - \eta(\gamma+1) \nabla F_{i,\ell}(\mathbf{x}_{i,\ell}^t)), \quad (49) \end{aligned}$$

and we define

$$\mu \triangleq \max_{p \in [1, P], \forall t, \ell, i} \left\{ \frac{\|\gamma(\mathbf{v}_{\{p\}}^t)\|}{\|\eta \nabla F(\mathbf{x}_{\{p\}}^t)\|}, \frac{\|\gamma(\mathbf{v}_{i,\ell}^t)\|}{\|\eta \nabla F_{i,\ell}(\mathbf{x}_{i,\ell}^t)\|} \right\}. \quad (50)$$

Because  $F_{i,\ell}(\cdot)$  is  $\rho$ -Lipschitz, and according to [47, Lecture 2, Lemma 1], we have  $\|\nabla F_{i,\ell}(\cdot)\|^2 \leq \rho^2$ . Therefore, based on

the definition of  $\mu$  and (49), we can derive

$$\begin{aligned} &\|\mathbf{x}_{\ell+}^{k\tau} - \mathbf{x}_{\ell-}^{k\tau}\| \\ &\leq \gamma_a \sum_{t=(k-1)\tau}^{k\tau-1} \sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} \|\gamma^2 \mathbf{v}_{i,\ell}^t - \eta(\gamma+1) \nabla F_{i,\ell}(\mathbf{x}_{i,\ell}^t)\| \\ &\leq \gamma_a \sum_{t=(k-1)\tau}^{k\tau-1} \sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} (\gamma \mu \eta + \eta(\gamma+1)) \rho \\ &= \gamma_a \tau \rho \eta (\gamma \mu + \gamma + 1). \quad (51) \end{aligned}$$

We complete the proof of Theorem 2.

### F. Proof of Theorem 3

First, we define edge virtual update which is meaningful in cloud interval  $\{p\}$  as  $\mathbf{y}_{\{p\},\ell}^t$  and  $\mathbf{x}_{\{p\},\ell}^t$ . The value synchronization and edge virtual update on  $\{p\}$  are conducted as

$$\mathbf{y}_{\{p\},\ell}^{(p-1)\tau\pi} \leftarrow \mathbf{y}^{(p-1)\tau\pi}, \quad (52)$$

$$\mathbf{x}_{\{p\},\ell}^{(p-1)\tau\pi} \leftarrow \mathbf{x}^{(p-1)\tau\pi}, \quad (53)$$

when  $t = (p-1)\tau\pi$ , and

$$\mathbf{y}_{\{p\},\ell}^t \leftarrow \mathbf{x}_{\{p\},\ell}^{t-1} - \eta \nabla F_\ell(\mathbf{x}_{\{p\},\ell}^{t-1}), \quad (54)$$

$$\mathbf{x}_{\{p\},\ell}^t \leftarrow \mathbf{y}_{\{p\},\ell}^t + \gamma (\mathbf{y}_{\{p\},\ell}^t - \mathbf{y}_{\{p\},\ell}^{t-1}), \quad (55)$$

when  $p \in ((p-1)\tau\pi, p\tau\pi]$ . According to Theorem 1, we have proved the gap between intermediate worker update on the edge  $\sum_{i=1}^{C_\ell} \frac{D_{i,\ell}}{D_\ell} \mathbf{x}_{i,\ell}^t$  and edge virtual update  $\mathbf{x}_{\{p\},\ell}^t$ . Equivalently, the gap between the intermediate edge virtual update on the cloud  $\sum_{\ell=1}^L \frac{D_\ell}{D} \mathbf{x}_{\{p\},\ell}^t$  and the cloud virtual update  $\mathbf{x}_{\{p\}}^t$  can be derived as the same way as Theorem 1. The only difference is the gradient divergence. The edge-level gradient divergence is  $\delta_\ell$  and the cloud-level gradient divergence is  $\delta$ . Therefore, for any cloud interval  $\{p\}, \forall t \in [(p-1)\tau\pi, p\tau\pi], \forall \ell \in L$ , we have

$$\left\| \sum_{\ell=1}^L \frac{D_\ell}{D} \mathbf{x}_{\{p\},\ell}^t - \mathbf{x}_{\{p\}}^t \right\| \leq h(t - (p-1)\tau\pi, \delta). \quad (56)$$

At the end of cloud interval  $\{p\}$ , when  $t = p\tau\pi$ , we have

$$\left\| \sum_{\ell=1}^L \frac{D_\ell}{D} \mathbf{x}_{\{p\},\ell}^{p\tau\pi} - \mathbf{x}_{\{p\}}^{p\tau\pi} \right\| \leq h(\tau\pi, \delta). \quad (57)$$

Based on the definition of  $\mathbf{x}_{\{p\}}^{p\tau\pi}$  in Theorem 3 and the definition of  $\mathbf{x}_{\{p\},\ell}^{p\tau\pi}$ , we obtain

$$\begin{aligned} &\left\| \mathbf{x}_{\{p\}}^{p\tau\pi} - \sum_{\ell=1}^L \frac{D_\ell}{D} \mathbf{x}_{\{p\},\ell}^{p\tau\pi} \right\| \leq \sum_{\ell=1}^L \frac{D_\ell}{D} \left\| \mathbf{x}_{\{p\},\ell}^{p\tau\pi} - \mathbf{x}_{\{p\},\ell}^{p\tau\pi} \right\| \\ &\leq \pi \sum_{\ell=1}^L \frac{D_\ell}{D} (h(\tau, \delta_\ell) + s(\tau)). \quad (58) \end{aligned}$$

Combining (57) and (58), we complete the proof of Theorem 3.

### G. Proof of Monotone of $h(x)$

To prove the monotone increasing of  $h(x)$ , it is equivalent to prove  $h(x) - h(x-1) \geq 0$  for all integer  $x \geq 1$ .

When  $x = 0$  or  $x = 1$ , because  $IA + JB = \frac{1+\eta\beta+\eta\beta\gamma}{\eta\beta\gamma}$ , we have  $h(0) = \eta\delta(I + J - \frac{1}{\eta\beta}) = 0$  and  $h(1) = \eta\delta\left(\gamma(IA + JB) - \frac{1}{\eta\beta} - \gamma - 1\right) = 0$ . Then, when  $x > 1$ , according to the definitions of  $A, B, U$ , and  $V$ , we can obtain that  $\gamma A > 1, 0 < \gamma B < 1, \frac{1}{\gamma+1} < B < 1, I > 0, J > 0, U > 0, V > 0$ , and  $U + V = 1$ . Then, we have

$$U(\gamma A)^i + V(\gamma B)^i \geq (1 + \eta\beta + \eta\beta\gamma)^i \quad (59)$$

holds  $\forall i = 0, 1, \dots$ . This is because: ① When  $i = 0$ ,  $U(\gamma A)^i + V(\gamma B)^i = (1 + \eta\beta + \eta\beta\gamma)^i = 1$ , (59) holds. ② When  $i = 1$ , we have  $U(\gamma A)^i + V(\gamma B)^i = \gamma(UA + VB) = \gamma\left(\frac{A-1}{A-B}A + \frac{1-B}{A-B}B\right) = \gamma(A + B - 1) = 1 + \eta\beta + \eta\beta\gamma$ . (59) still holds. ③ When  $i > 1$ , according to Jensen inequality, and because any function  $n(x) = x^i$  is convex, we have  $U(\gamma A)^i + V(\gamma B)^i \geq (\gamma UA + \gamma VB)^i = (1 + \eta\beta + \eta\beta\gamma)^i$ . (59) still holds.

According to (59) and the definition of  $u(x)$  in Lemma 2, we have  $u(x) = U(\gamma A)^x + V(\gamma B)^x - 1 \geq (1 + \eta\beta + \eta\beta\gamma)^x - 1 > 0$ . Then, we have

$$\begin{aligned} h(x) - h(x-1) &= \eta\delta \left( \frac{U(\gamma A)^x (\gamma A + A - 1)}{\gamma A (A - 1)} \right. \\ &\quad \left. + \frac{V(\gamma B)^x (\gamma B + B - 1)}{\gamma B (B - 1)} - \frac{\gamma^{x+1} - 1}{\gamma - 1} \right) \\ &\stackrel{(a)}{=} \eta\delta \left( \frac{U(\gamma A)^x}{\gamma(A-1)} + \frac{V(\gamma B)^x}{\gamma(B-1)} - \frac{\gamma^x - 1}{\gamma - 1} \right) \\ &\quad + \eta\delta(\gamma^{x-1}(UA^{x-1} + VB^{x-1}) - 1) \\ &\stackrel{(b)}{=} \eta\delta \sum_{b=1}^x \gamma^{x-b} u(b-1) + \eta\delta u(x-1) > 0, \end{aligned}$$

where (a) is because (39) equals (38); (b) is because (48) equals (47),  $x = t - (k-1)\tau$ , and the definition of  $u(x)$ . To conclude, we have proven that  $h(0) = h(1) = 0$  and  $h(x)$  increases with  $x$  when  $x \geq 1$ .

### H. Proof of Theorem 4

For convenience, we define  $c_{\{p\}}(t) \triangleq F(\mathbf{x}_{\{p\}}^t) - F(\mathbf{x}^*)$  for a given cloud interval  $\{p\}$ , where  $t \in [(p-1)\tau\pi, p\tau\pi]$ . We also define the following constants in this subsection.

$$\begin{aligned} \omega &\triangleq \min_{p \in [1, P], t \in \{p\}} \frac{1}{\|\mathbf{x}_{\{p\}}^t - \mathbf{x}^*\|^2}, \\ \sigma &\triangleq \min_{p \in [1, P], t_1, t_2 \in \{p\}} \frac{\|\nabla F(\mathbf{x}_{\{p\}}^{t_1})\|}{\|\nabla F(\mathbf{x}_{\{p\}}^{t_2})\|}, \\ \alpha &\triangleq \eta(\gamma + 1) \left( 1 - \frac{\beta\eta(\gamma + 1)}{2} \right) - \frac{\beta\eta^2\gamma^2\mu^2}{2} \\ &\quad - \eta\gamma\mu(1 - \beta\eta(\gamma + 1)). \end{aligned} \quad (60)$$

According to the convergence lower bound of any gradient descent methods given in [36, Theorem 3.14], we always have  $c_{\{p\}}(t) > 0$  for any  $t$  and  $p$ . Then we derive the upper bound

of  $c_{\{p\}}(t+1) - c_{\{p\}}(t)$ , where  $t \in [(p-1)\tau\pi, p\tau\pi - 1]$ . Because  $F(\cdot)$  is  $\beta$ -smooth, according to [36, Lemma 3.4], we have

$$\begin{aligned} c_{\{p\}}(t+1) - c_{\{p\}}(t) &= F(\mathbf{x}_{\{p\}}^{t+1}) - F(\mathbf{x}_{\{p\}}^t) \\ &\leq \langle \nabla F(\mathbf{x}_{\{p\}}^t), \mathbf{x}_{\{p\}}^{t+1} - \mathbf{x}_{\{p\}}^t \rangle + \frac{\beta}{2} \|\mathbf{x}_{\{p\}}^{t+1} - \mathbf{x}_{\{p\}}^t\|^2 \\ &= \gamma \langle \nabla F(\mathbf{x}_{\{p\}}^t), \mathbf{v}_{\{p\}}^{t+1} \rangle - \eta \|\nabla F(\mathbf{x}_{\{p\}}^t)\|^2 \\ &\quad + \frac{\beta}{2} \|\gamma \mathbf{v}_{\{p\}}^{t+1} - \eta \nabla F(\mathbf{x}_{\{p\}}^t)\|^2 \\ &\stackrel{(a)}{=} -\eta(\gamma + 1) \left( 1 - \frac{\beta\eta(\gamma + 1)}{2} \right) \|\nabla F(\mathbf{x}_{\{p\}}^t)\|^2 \\ &\quad + \frac{\beta\gamma^4}{2} \|\mathbf{v}_{\{p\}}^t\|^2 + \gamma^2 (1 - \beta\eta(\gamma + 1)) \langle \nabla F(\mathbf{x}_{\{p\}}^t), \mathbf{v}_{\{p\}}^t \rangle \\ &\stackrel{(b)}{\leq} \left( -\eta(\gamma + 1) \left( 1 - \frac{\beta\eta(\gamma + 1)}{2} \right) + \frac{\beta\eta^2\gamma^2\mu^2}{2} \right. \\ &\quad \left. + \eta\gamma\mu(1 - \beta\eta(\gamma + 1)) \right) \|\nabla F(\mathbf{x}_{\{p\}}^t)\|^2, \end{aligned} \quad (62)$$

where (a) is replacing  $\mathbf{v}_{\{p\}}^{t+1}$  by (34) and rearranging the formula; (b) is because  $\|\gamma \mathbf{v}_{\{p\}}^t\| \leq \mu \|\eta \nabla F(\mathbf{x}_{\{p\}}^t)\|$  with the definition of  $\mu$ . According to Cauchy-Schwarz inequality, we can obtain  $\langle \nabla F(\mathbf{x}_{\{p\}}^t), \mathbf{v}_{\{p\}}^t \rangle \leq \|\nabla F(\mathbf{x}_{\{p\}}^t)\| \|\mathbf{v}_{\{p\}}^t\| \leq \frac{\mu\eta}{\gamma} \|\nabla F(\mathbf{x}_{\{p\}}^t)\|^2$ . According to the definition of  $\alpha$ , and Condition (2.1) of Theorem 4 with  $h(\tau, \delta_\ell) \geq 0$  and  $h(\tau\pi, \delta) \geq 0$  which are proved in Appendix G, we have  $\alpha > 0$ . Then from (62), we have

$$c_{\{p\}}(t+1) \leq c_{\{p\}}(t) - \alpha \|\nabla F(\mathbf{x}_{\{p\}}^t)\|^2. \quad (63)$$

Because  $F(\cdot)$  is  $\rho$ -Lipschitz, and according to [47, Lecture 2, Lemma 1], there exists a point  $\mathbf{x}_{\{p\}}^{t_2}$  such that  $F(\mathbf{x}_{\{p\}}^t) - F(\mathbf{x}^*) = \langle \nabla F(\mathbf{x}_{\{p\}}^{t_2}), \mathbf{x}_{\{p\}}^t - \mathbf{x}^* \rangle$ . Hence, by Cauchy-Schwarz inequality, we have  $c_{\{p\}}(t) = F(\mathbf{x}_{\{p\}}^t) - F(\mathbf{x}^*) \leq \|\nabla F(\mathbf{x}_{\{p\}}^{t_2})\| \|\mathbf{x}_{\{p\}}^t - \mathbf{x}^*\|$ . Based on the definition of  $\sigma$ , and replacing  $t$  with  $t_1$ , we have  $\|\nabla F(\mathbf{x}_{\{p\}}^t)\| \geq \sigma \|\nabla F(\mathbf{x}_{\{p\}}^{t_2})\|$ . Thus,  $\|\nabla F(\mathbf{x}_{\{p\}}^t)\| \geq \sigma \|\nabla F(\mathbf{x}_{\{p\}}^{t_2})\| \geq \frac{\sigma c_{\{p\}}(t)}{\|\mathbf{x}_{\{p\}}^t - \mathbf{x}^*\|}$ . Substituting above inequality into (63), and noting

$\omega \leq \frac{1}{\|\mathbf{x}_{\{p\}}^t - \mathbf{x}^*\|^2}$  by the definition of  $\omega$ , we get  $c_{\{p\}}(t+1) \leq c_{\{p\}}(t) - \frac{\alpha\sigma^2 c_{\{p\}}(t)^2}{\|\mathbf{x}_{\{p\}}^t - \mathbf{x}^*\|^2} \leq c_{\{p\}}(t) - \omega\alpha\sigma^2 c_{\{p\}}(t)^2$ . Because  $\alpha > 0$ ,  $c_{\{p\}}(t) > 0$ , and (63), we have  $0 < c_{\{p\}}(t+1) \leq c_{\{p\}}(t)$ . Dividing both sides by  $c_{\{p\}}(t+1)c_{\{p\}}(t)$ , we get  $\frac{1}{c_{\{p\}}(t)} \leq \frac{1}{c_{\{p\}}(t+1)} - \omega\alpha\sigma^2 \frac{c_{\{p\}}(t)}{c_{\{p\}}(t+1)}$ . We note that  $\frac{c_{\{p\}}(t)}{c_{\{p\}}(t+1)} \geq 1$ . Thus,  $\frac{1}{c_{\{p\}}(t+1)} - \frac{1}{c_{\{p\}}(t)} \geq \omega\alpha\sigma^2 \frac{c_{\{p\}}(t)}{c_{\{p\}}(t+1)} \geq \omega\alpha\sigma^2$ . Summing up the above inequality by  $t \in [(p-1)\tau\pi, p\tau\pi - 1]$ , we have  $\frac{1}{c_{\{p\}}(p\tau\pi)} - \frac{1}{c_{\{p\}}((p-1)\tau\pi)} = \sum_{t=(p-1)\tau\pi}^{p\tau\pi-1} \left( \frac{1}{c_{\{p\}}(t+1)} - \frac{1}{c_{\{p\}}(t)} \right) \geq \sum_{t=(p-1)\tau\pi}^{p\tau\pi-1} \omega\alpha\sigma^2 = \tau\pi\omega\alpha\sigma^2$ . Then, we sum up the above inequality by  $p \in [1, P]$ , after rearranging the left-hand side and noting that  $T = P\tau\pi$ , we can get

$$\begin{aligned} &\sum_{p=1}^P \left( \frac{1}{c_{\{p\}}(p\tau\pi)} - \frac{1}{c_{\{p\}}((p-1)\tau\pi)} \right) \\ &= \frac{1}{c_{\{p\}}(T)} - \frac{1}{c_{\{1\}}(0)} - \sum_{p=1}^{P-1} \left( \frac{1}{c_{\{p+1\}}(p\tau\pi)} - \frac{1}{c_{\{p\}}(p\tau\pi)} \right) \\ &\geq P\tau\pi\omega\alpha\sigma^2 = T\omega\alpha\sigma^2. \end{aligned} \quad (64)$$

Following (64), we note that

$$\begin{aligned}
& \frac{1}{c_{\{p+1\}}(p\tau\pi)} - \frac{1}{c_{\{p\}}(p\tau\pi)} = \frac{c_{\{p\}}(p\tau\pi) - c_{\{p+1\}}(p\tau\pi)}{c_{\{p\}}(p\tau\pi)c_{\{p+1\}}(p\tau\pi)} \\
& = \frac{F(\mathbf{x}_{\{p\}}^{p\tau\pi}) - F(\mathbf{x}_{\{p+1\}}^{p\tau\pi})}{c_{\{p\}}(p\tau\pi)c_{\{p+1\}}(p\tau\pi)} = \frac{F(\mathbf{x}_{\{p\}}^{p\tau\pi}) - F(\mathbf{x}^{p\tau\pi})}{c_{\{p\}}(p\tau\pi)c_{\{p+1\}}(p\tau\pi)} \\
& = \frac{F(\mathbf{x}_{\{p\}}^{p\tau\pi}) - F(\mathbf{x}_{\{p\pi\}}^{p\tau\pi}) + (F(\mathbf{x}_{\{p\pi\}}^{p\tau\pi}) - F(\mathbf{x}^{p\tau\pi}))}{c_{\{p\}}(p\tau\pi)c_{\{p+1\}}(p\tau\pi)} \\
& \stackrel{(a)}{\geq} \frac{-\rho \sum_{\ell=1}^L \frac{D_\ell}{D} (h(\tau, \delta_\ell) + s(\tau))}{c_{\{p\}}(p\tau\pi)c_{\{p+1\}}(p\tau\pi)} \\
& \quad + \frac{(b) -\rho \left( h(\tau\pi, \delta) + \pi \sum_{\ell=1}^L \frac{D_\ell}{D} (h(\tau, \delta_\ell) + s(\tau)) \right)}{c_{\{p\}}(p\tau\pi)c_{\{p+1\}}(p\tau\pi)} \\
& = \frac{-\rho j(\tau, \pi, \delta_\ell, \delta)}{c_{\{p\}}(p\tau\pi)c_{\{p+1\}}(p\tau\pi)}, \tag{65}
\end{aligned}$$

where (a) is because of combining Theorem 1 and Theorem 2; (b) is because of Theorem 3.

From (63), we can get  $F(\mathbf{x}_{\{p\}}^t) \geq F(\mathbf{x}_{\{p\}}^{t+1})$  for any  $t \in [(p-1)\tau\pi, p\tau\pi)$ . Recalling Condition (2.2) in Theorem 4, where  $F(\mathbf{x}_{\{p\}}(p\tau\pi)) - F(\mathbf{x}^*) \geq \varepsilon$  for all  $p$ , we can obtain  $c_{\{p\}}(t) = F(\mathbf{x}_{\{p\}}^t) - F(\mathbf{x}^*) \geq \varepsilon$  for all  $t \in [(p-1)\tau\pi, p\tau\pi]$  and  $p$ . Thus,  $c_{\{p\}}(p\tau\pi)c_{\{p+1\}}(p\tau\pi) \geq \varepsilon^2$ . According to Appendix G, we have  $h(\tau, \delta_\ell) \geq 0$  and  $h(\tau\pi, \delta) \geq 0$ . Then substituting above inequalities into (65), we obtain  $\frac{1}{c_{\{p+1\}}(p\tau\pi)} - \frac{1}{c_{\{p\}}(p\tau\pi)} \geq \frac{-\rho j(\tau, \pi, \delta_\ell, \delta)}{\varepsilon^2}$ . Substituting the above inequality into (64) and rearrange, we get

$$\frac{1}{c_{\{p\}}(T)} - \frac{1}{c_{\{1\}}(0)} \geq T\omega\alpha\sigma^2 - (P-1) \frac{\rho j(\tau, \pi, \delta_\ell, \delta)}{\varepsilon^2}. \tag{66}$$

Recalling Condition (2.3) in Theorem 4, where  $F(\mathbf{x}^T) - F(\mathbf{x}^*) \geq \varepsilon$ , and noting that  $c_{\{p\}}(T) \geq \varepsilon$ , we get  $(F(\mathbf{x}^T) - F(\mathbf{x}^*))c_{\{p\}}(T) \geq \varepsilon^2$ . Thus,

$$\begin{aligned}
& \frac{1}{F(\mathbf{x}^T) - F(\mathbf{x}^*)} - \frac{1}{c_{\{p\}}(T)} = \frac{c_{\{p\}}(T) - (F(\mathbf{x}^T) - F(\mathbf{x}^*))}{(F(\mathbf{x}^T) - F(\mathbf{x}^*))c_{\{p\}}(T)} \\
& = \frac{F(\mathbf{x}_{\{p\}}^T) - F(\mathbf{x}^T)}{(F(\mathbf{x}^T) - F(\mathbf{x}^*))c_{\{p\}}(T)} \\
& \geq \frac{-\rho j(\tau, \pi, \delta_\ell, \delta)}{(F(\mathbf{x}^T) - F(\mathbf{x}^*))c_{\{p\}}(T)} \geq -\frac{\rho j(\tau, \pi, \delta_\ell, \delta)}{\varepsilon^2}, \tag{67}
\end{aligned}$$

where the first inequality follows the same method to prove (65).

Combining (66) with (67), we get  $\frac{1}{F(\mathbf{x}^T) - F(\mathbf{x}^*)} - \frac{1}{c_{\{1\}}(0)} \geq T\omega\alpha\sigma^2 - P \frac{\rho j(\tau, \pi, \delta_\ell, \delta)}{\varepsilon^2} = T\omega\alpha\sigma^2 - T \frac{\rho j(\tau, \pi, \delta_\ell, \delta)}{\tau\pi\varepsilon^2} = T \left( \omega\alpha\sigma^2 - \frac{\rho j(\tau, \pi, \delta_\ell, \delta)}{\tau\pi\varepsilon^2} \right)$ . Noting that  $c_{\{1\}}(0) = F(\mathbf{x}_{\{1\}}^0) - F(\mathbf{x}^*) > 0$ , the above inequality can be expressed as  $\frac{1}{F(\mathbf{x}^T) - F(\mathbf{x}^*)} \geq T \left( \omega\alpha\sigma^2 - \frac{\rho j(\tau, \pi, \delta_\ell, \delta)}{\tau\pi\varepsilon^2} \right)$ . Recalling Condition (2.1) in Theorem 4, where  $\omega\alpha\sigma^2 - \frac{\rho j(\tau, \pi, \delta_\ell, \delta)}{\tau\pi\varepsilon^2} > 0$ , we obtain that the right-hand side of above inequality is greater than zero. Therefore, taking the reciprocal of the above inequality, we finally complete the proof of Theorem 4.

## I. Proof of Theorem 5

At the beginning, we see that Condition (1) in Theorem 4 holds due to the Condition in Theorem 5 ( $0 < \beta\eta(\gamma+1) \leq 1$ ,  $0 < \gamma < 1$ ,  $0 < \gamma_a < 1$ , and  $\forall \tau, \pi \in \{1, 2, \dots\}$ ).

1)  $\rho j(\tau, \pi) = 0$ : In this case, there is an arbitrarily small  $\varepsilon > 0$  that let Conditions (2.1)–(2.3) in Theorem 4 hold. In this case, Theorem 4 holds. We also note that the right-hand side of (23) is equivalent to the right-hand side of (20) when  $\rho j(\tau, \pi) = 0$ . According to the definition of  $\mathbf{x}^f$  in (22), we have  $F(\mathbf{x}^f) - F(\mathbf{x}^*) \leq F(\mathbf{x}^T) - F(\mathbf{x}^*) \leq \frac{1}{T\omega\alpha\sigma^2}$ , which satisfies the result in Theorem 4 directly. Thus, Theorem 5 holds when  $\rho j(\tau, \pi) = 0$ .

2)  $\rho j(\tau, \pi) > 0$ : In this case, we aim to find an  $\varepsilon$  satisfying Condition (2.1), but Conditions (2.2) and (2.3) cannot be satisfied together so that  $F(\mathbf{x}^f) - F(\mathbf{x}^*)$  can be bounded. We first define an  $\varepsilon_0$ , then we claim that any  $\varepsilon > \varepsilon_0$  is what we want to find.

We set  $\varepsilon_0$  as the root of the following equation,

$$\varepsilon_0 = \frac{1}{T \left( \omega\alpha\sigma^2 - \frac{\rho j(\tau, \pi)}{\tau\pi\varepsilon_0^2} \right)}. \tag{68}$$

The positive root is

$$\varepsilon_0 = \frac{1}{2T\omega\alpha\sigma^2} + \sqrt{\frac{1}{4T^2\omega^2\alpha^2\sigma^4} + \frac{\rho j(\tau, \pi)}{\omega\alpha\sigma^2\tau\pi}}. \tag{69}$$

Through this way, since  $\omega\alpha\sigma^2 - \frac{\rho j(\tau, \pi)}{\tau\pi\varepsilon^2}$  increases with  $\varepsilon$ ,  $\varepsilon > \varepsilon_0$  will lead to Condition (2.1).

Next, using the proof by contradiction, we can prove that when  $\varepsilon > \varepsilon_0$ , there does not exist  $\varepsilon > \varepsilon_0$  that satisfies both Conditions (2.2) and (2.3) in Theorem 4 at the same time.

We assume that there exists such  $\varepsilon > \varepsilon_0$ , so that Conditions (2.1)–(2.3) hold and thus Theorem 4 holds. Then we have  $F(\mathbf{x}^T) - F(\mathbf{x}^*) \leq \frac{1}{T(\omega\alpha\sigma^2 - \frac{\rho j(\tau, \pi)}{\tau\pi\varepsilon^2})} < \frac{1}{T(\omega\alpha\sigma^2 - \frac{\rho j(\tau, \pi)}{\tau\pi\varepsilon_0^2})} = \varepsilon_0$ , which contradicts the Condition (2.3) in Theorem 4.

Therefore, for any  $\varepsilon > \varepsilon_0$ , one of the following (A) or (B) holds. (A)  $\exists p \in [1, P]$  such that  $F(\mathbf{x}_{\{p\}}^{p\tau\pi}) - F(\mathbf{x}^*) \leq \varepsilon_0$  or (B)  $F(\mathbf{x}^T) - F(\mathbf{x}^*) \leq \varepsilon_0$ . (A) or (B) gives

$$\min \left\{ \min_{p \in [1, P]} F(\mathbf{x}_{\{p\}}^{p\tau\pi}); F(\mathbf{x}^T) \right\} - F(\mathbf{x}^*) \leq \varepsilon_0. \tag{70}$$

According to (67), when  $t = p\tau\pi$ , we have  $F(\mathbf{x}^{p\tau\pi}) \leq F(\mathbf{x}_{\{p\}}^{p\tau\pi}) + \rho j(\tau, \pi)$  for any cloud interval  $\{p\}$ . Combining it with (70), we have  $\min_{p \in [1, P]} F(\mathbf{x}^{p\tau\pi}) - F(\mathbf{x}^*) \leq \varepsilon_0 + \rho j(\tau, \pi)$ . Recalling the definition of  $\mathbf{x}^f$  in (22),  $T = P\tau\pi$ , and combining  $\mathbf{x}^f$  with above inequality, we get  $F(\mathbf{x}^f) - F(\mathbf{x}^*) \leq \varepsilon_0 + \rho j(\tau, \pi)$ . Substituting (69) into above inequality, we finally get the result in (23), which completes the proof of Theorem 5.



### J. Proof of Theorem 6

When  $\eta \rightarrow 0^+$ , we have  $\gamma A \simeq 1$ ,  $\gamma B \simeq \gamma$ , and  $J \simeq \frac{\gamma^2}{(1-\gamma)^2}$ . Therefore,

$$\begin{aligned} & \lim_{\eta \rightarrow 0^+} h(\tau, \delta_\ell) \\ &= \lim_{\eta \rightarrow 0^+} \eta \delta_\ell \left[ I(\gamma A)^\tau + J(\gamma B)^\tau - \frac{1}{\eta \beta} - \frac{\gamma^2(\gamma^\tau - 1) - (\gamma - 1)\tau}{(\gamma - 1)^2} \right] \\ &= \lim_{\eta \rightarrow 0^+} \eta \delta_\ell \left( I - \frac{1}{\eta \beta} \right) \\ &= \lim_{\eta \rightarrow 0^+} \eta \delta_\ell \left( \frac{1}{(1-\gamma)(\gamma A - 1)} - \frac{1}{\eta \beta} \right) \\ &= \frac{\delta_\ell}{1-\gamma} \lim_{\eta \rightarrow 0^+} \frac{\eta}{\gamma A - 1} - \frac{\delta_\ell}{\beta} \\ &= \frac{\delta_\ell}{1-\gamma} \lim_{\eta \rightarrow 0^+} \frac{1}{(\gamma A - 1)'} - \frac{\delta_\ell}{\beta} \\ &= \frac{\delta_\ell}{1-\gamma} \frac{1-\gamma}{\beta} - \frac{\delta_\ell}{\beta} = 0 \end{aligned}$$

where the second last line is because of the L'Hôpital's rule. Then, we can derive  $s(\cdot) \simeq 0$ . Afterwards, we have  $j(\cdot) \simeq 0$  and  $\hat{j}(\cdot) \simeq 0$ . Therefore,  $f_{HierMo}(T) \simeq \frac{1}{T\omega\alpha\sigma^2}$  and  $f_{HierFAVG}(T) \simeq \frac{1}{T\omega\hat{\alpha}\sigma^2}$ . Based on the conditions in Theorem 6, we have  $\alpha > \hat{\alpha}$ . Therefore, we have  $f_{HierFAVG}(T) - f_{HierMo}(T) > 0$ , which completes the proof of Theorem 6.

### REFERENCES

- [1] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International journal of Remote sensing*, vol. 28, no. 5, pp. 823–870, 2007.
- [2] J. E. Naranjo, C. González, R. García, and etc., "Power-steering control architecture for automatic driving," *IEEE transactions on intelligent transportation systems*, vol. 6, no. 4, pp. 406–415, 2005.
- [3] D. Yu and L. Deng, *Automatic speech recognition*. Springer, 2016, vol. 1.
- [4] B. McMahan, E. Moore, D. Ramage, and etc., "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.
- [5] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.
- [6] M. S. H. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, "Hierarchical federated learning across heterogeneous cellular networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8866–8870.
- [7] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *IJCNN*, 2020, pp. 1–9.
- [8] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Hierarchical quantized federated learning: Convergence analysis and system design," *arXiv preprint arXiv:2103.14272*, 2021.
- [9] Y. Yan, T. Yang, Z. Li, Q. Lin, and Y. Yang, "A unified analysis of stochastic momentum methods for deep learning," in *IJCAI*, 2018, pp. 2955–2961.
- [10] C. Liu and M. Belkin, "Accelerating SGD with momentum for over-parameterized learning," in *International Conference on Learning Representations*, 2020.
- [11] S. Vaswani, F. Bach, and M. Schmidt, "Fast and faster convergence of sgd for over-parameterized models and an accelerated perceptron," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1195–1204.
- [12] M. Assran and M. Rabbat, "On the convergence of nesterov's accelerated gradient method in stochastic settings," in *Proceedings of the 37th International Conference on Machine Learning*, 2020, pp. 410–420.
- [13] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating federated learning via momentum gradient descent," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 8, pp. 1754–1766, 2020.
- [14] 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.
- [15] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-iid federated learning," in *ICLR*, 2021.
- [16] H. Gao, A. Xu, and H. Huang, "On the convergence of communication-efficient local sgd for federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021.
- [17] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.
- [18] 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.
- [19] Z. Wang, H. Xu, J. Liu, H. Huang, C. Qiao, and Y. Zhao, "Resource-efficient federated learning with hierarchical aggregation in edge computing," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.
- [20] Z. Huo, Q. Yang, B. Gu, L. C. Huang et al., "Faster on-device training using new federated momentum algorithm," *arXiv preprint arXiv:2002.02090*, 2020.
- [21] J. Wang, V. Tantia, N. Ballas, and M. Rabbat, "SlowMo: Improving communication-efficient distributed sgd with slow momentum," in *International Conference on Learning Representations*, 2020.
- [22] Z. Yang, W. Bao, D. Yuan, N. H. Tran, and A. Y. Zomaya, "Federated learning with nesterov accelerated gradient momentum method," *arXiv preprint arXiv:2009.08716*, 2020.
- [23] S. P. Karimireddy, M. Jaggi, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Mime: Mimicking centralized stochastic algorithms in federated learning," *arXiv preprint arXiv:2008.03606*, 2020.
- [24] A. Xu and H. Huang, "Coordinating momenta for cross-silo federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8735–8743.
- [25] E. Ozfatura, K. Ozfatura, and D. Gündüz, "FedADC: Accelerated federated learning with drift control," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE Press, 2021, p. 467–472.
- [26] yeggasd, A. Trask, and froessler, *FL on MNIST using a CNN*, 2021. [Online]. Available: <https://notebook.community/OpenMined/PySyft/examples/tutorials/Part-6-Federated-Learning-on-MNIST-using-a-CNN>
- [27] S. Gross, S. Chintala, N. Hug, L. Yeager, and E. R. etc., *Pytorch-VGG*, may 2021. [Online]. Available: <https://github.com/pytorch/vision/blob/master/torchvision/models/vgg.py>
- [28] T. Moon and T. Ryffel, *Pytorch-Tiny-ImageNet*, jun 2020. [Online]. Available: <https://github.com/tjmoon0104/pytorch-tiny-imagenet>
- [29] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [30] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," *N/A*, 2009.
- [31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [32] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*, 2013, pp. 437–442.
- [33] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.
- [34] G. Goh, "Why momentum really works," *Distill*, 2017. [Online]. Available: <http://distill.pub/2017/momentum>
- [35] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence of  $1/k^2$ ," *Doklady ANSSSR (translated as Soviet.Math.Docl.)*, vol. 269, pp. 543–547, 1983.
- [36] S. Bubeck, "Convex optimization: Algorithms and complexity," *arXiv preprint arXiv:1405.4980*, 2014.
- [37] Z. Yang, S. Fu, W. Bao, D. Yuan, and A. Y. Zomaya, "FastSlowMo: Federated learning with combined worker and aggregator momenta," *IEEE Transactions on Artificial Intelligence*, 2022.
- [38] 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.
- [39] T. Castiglia, A. Das, and S. Patterson, "Multi-level local sgd for heterogeneous hierarchical networks," *arXiv preprint arXiv:2007.13819*, 2020.

- [40] Y. Deng, F. Lyu, J. Ren, Y. Zhang, Y. Zhou, Y. Zhang, and Y. Yang, "Share: Shaping data distribution at edge for communication-efficient hierarchical federated learning," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, 2021, pp. 24–34.
- [41] 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.
- [42] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.
- [43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [44] C. T. Dinh, N. H. Tran, T. D. Nguyen, W. Bao, A. Y. Zomaya, and B. B. Zhou, "Federated learning with proximal stochastic variance reduced gradient algorithms," in *49th International Conference on Parallel Processing-ICPP*, 2020, pp. 1–11.
- [45] S. Y. Teng, M. Touš, W. D. Leong, B. S. How, H. L. Lam, and V. Máša, "Recent advances on industrial data-driven energy savings: Digital twins and infrastructures," *Renewable and Sustainable Energy Reviews*, vol. 135, p. 110208, 2021.
- [46] J. C. Kirchhof, L. Malcher, and B. Rumpe, "Understanding and improving model-driven iot systems through accompanying digital twins," in *Proceedings of the 20th ACM SIGPLAN ICPG: Concepts and Experiences*, 2021, pp. 197–209.
- [47] I. Mitliagkas and J. Gallego, "Ifit 6085: Theoretical principles for deep learning," in *University of Montreal*. University of Montreal, 2021. [Online]. Available: <http://mitliagkas.github.io/ift6085-dl-theory-class/>