Similarity-Guided Rapid Deployment of Federated Intelligence over Heterogeneous Edge Computing

Hansong Zhou*, Jingjing Fu‡, Yukun Yuan†, Linke Guo‡ and Xiaonan Zhang*
* Department of Computer Science, Florida State University, Tallahassee, FL 32306, USA
† Department of Computer Science, University of Tenessee at Chattanooga, Chattanooga, TN 37403, USA
‡ Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA
Email: hz21e@fsu.edu, jfu@g.clemson.edu, yukun-yuan@utc.edu, linkeg@clemson.edu, xzhang@cs.fsu.edu

IEEE INFOCOM 2025 - IEEE Conference on Computer Communications | 979-8-3315-4305-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/INFOCOM55648.2025.11044586

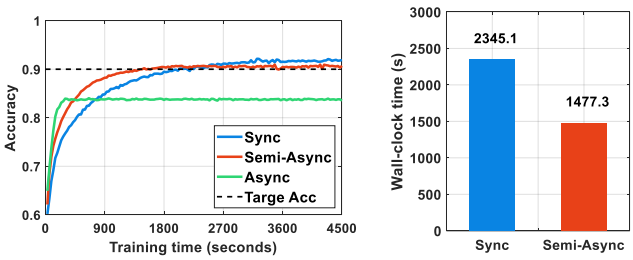
***Abstract*—Edge computing is envisioned to enable rapid federated intelligence on edge devices to satisfy their dynamically changing AI service demands. Semi-Asynchronous FL (Semi-Async FL) enables distributed learning in an asynchronous manner, where the server does not have to wait all local models for improving the global model. Hence, it takes a small time to well-train a global model. However, system heterogeneity in edge computing results in staleness issue, which will deteriorate training accuracy. In this paper, we propose to accelerate Semi-Async FL while ensuring training accuracy by designing a Similarity-Aware Aggregation (SAA) strategy. SAA is able to enhance the aggregation quality and thus decrease the wall-clock time, the training time for a certain accuracy. Particularly, we leverage the global model similarity to describe the local model influence and let those with higher influence contribute more to global aggregation. We further measure the similarity between global model update deviations as directional similarity, which is then used for determining aggregation timing. We theoretically provide a convergence analysis to SAA. Our extensive experimental results empirically show that the proposed SAA strategy reduces up to 53.7% wall-clock time and 59.4% wall-clock round for Semi-Async FL compared with several benchmark schemes.**

***Index Terms*—Semi-Asynchronous Federated learning, Model Staleness, Similarity-Aware Aggregation.**

I. Introduction

Edge computing, as a transformative shift in computing systems, brings computational resources and data storage closer to data sources, which enables the efficient processing and analysis of vast data volumes on edge devices in a timely fashion [1], [2]. Being able to exploit the abundant data on edge devices and their computational abilities, Federated learning (FL) offers valuable insights into various edge computing applications, such as smart cities and intelligent industries [3]–[5]. In synchronous federated learning (Sync FL), the central server maintains a global model and synchronizes updates from all participating clients at each training round. Clients compute local updates using their private data and send them to the server, which aggregates the updates only after all clients have completed their computations. This synchronized aggregation ensures consistency across iterations and continues until the global model converges to a desired accuracy.

The proliferation of edge devices, coupled with their dynamically changing demands for AI services, necessitates rapid training of FL over edge computing. However, the disparate communication and computation capabilities of edge devices lead to system heterogeneity, causing considerable variances



(a) Training accuracy (b) Wall-clock time

Fig. 1: Comparison of different FL schemes (EMNIST)

in *response time* — the time spent on local model updating in FL. In Sync FL, the server must wait for *stragglers*, clients with slow response time, to perform global aggregation. This significantly increases the *wall-clock time*, time spent to achieve a certain accuracy of the global model. This hinders the fast deployment of federated intelligence, particularly in large-scale systems such as Intelligent Transportation Systems [6]. Asynchronous FL (Async FL) addresses this by allowing the server to update the global model once receiving the local model from any client. Nevertheless, Async FL in heterogeneous systems always introduces significant staleness to weak devices, where outdated client updates harm the aggregation quality as shown in Fig. 1. To balance these issues, Semi-Async FL enables the server to perform aggregation from a subset of clients synchronously while allowing others to train local models asynchronously at their own pace. As a result, it manages to achieve a close final accuracy with Sync FL but save $1/3$ wall-clock time. Moreover, Semi-Async FL significantly outperforms Sync FL during the early stages of training, making it an optimal candidate for the rapid deployment of federated intelligence.

Nevertheless, the presence of stragglers results in varying frequencies of clients participating in asynchronous global model updating. This brings staleness in local models as they are trained from the global models of the old versions. The existence of staleness will prevent the global model from learning enough knowledge about the datasets at stale clients, which will decrease the training accuracy in the end [7], [8]. Existing solutions such as weight allocation [9]–[11], device selection [12]–[14], and resource allocation [15]–[17] get less stale local models more involved in aggregation. The rationale behind this is that a less stale local model has higher influence on global model updates. Apart from staleness, are there any

metrics to assess the local model influence to aggregation?

In this paper, we propose a novel Semi-Async FL steered by similarity measurements, aiming to expedite the implementation of federated intelligence over edge computing systems. We attempt to reduce wall-clock time via accommodating uploaded local models with various staleness. Wall-clock time in Semi-Async FL is derived by adding up the duration of each round across all rounds. If we can improve the global model update quality in each round, both individual round duration and the overall number of rounds will decrease. To achieve it, we propose a similarity-aware aggregation (SAA) strategy to answer the following questions: (1) how to figure out influential local models and better engage them in aggregation; and (2) when to asynchronously update the global model. We summarize our key attributions of this paper as follows:

- We describe the influence of a local model update on the global update based on the similarity between the stale global model it used and the fresh model, which is used to adjust its contribution during aggregation accordingly.
- We guide the aggregation through quantifying its stability based on the similarity between two consecutive global model update deviations for higher accuracy.
- We propose a similarity-aware aggregation (SAA) strategy to improve the Semi-Async FL performance with a theoretical convergence guarantee.
- We evaluate the SAA strategy on real-world datasets, demonstrating that SAA dramatically outperforms benchmarks in reducing the wall-clock time.

II. Related Work

Similarity in Federated Learning. There are a few works considering the similarity in FL [18]–[21]. In [18], Zhang et al. designed similarity metrics including the model update latency and similarity between clients' gradients for clustering. KL divergence was applied in [19] to measure the similarity between client models as well as optimize the cluster relationship. The clustering structure is inferred based on the similarity between clients' gradient updates in [20]. In conventional Sync-FL, similarity plays an important role in client selection. In FAIR system [22], the similarity between the upload model and global model was calculated to filter out the low-quality update of clients. Nevertheless, the relation between similarity and model quality is not well addressed. Wang et al. [23] directly considered the similarity among local datasets for device sampling. In their work, D2D offloading and data information sharing are permitted between trusted neighbours. However, the assumption of the trusted clients is not applicable for real-world implementation due to the high cost of device verification. Furthermore, these works use similarity as a supplementary metric for client sampling or clustering only in Sync-FL. Few works in the literature have discussed the similarity in Semi-Async FL.

Staleness in Aggregation. A mainstream approach to mitigate negative influence from stale local models is to re-weight their models during the aggregation. A general weight design is to assign the weight to each client by using a polynomial function based on the round variance. A cluster-based FL mechanism was introduced in [24], where clients in each cluster perform in a synchronous manner. They discuss high-level staleness and set a staleness threshold for each cluster to implement different weight functions at different stale levels. However, those clients who exceed the threshold are either treated as untrained models in global update [25] or simply discarded [8], [26], [27]. Unlike these works, we ground from similarity observations to improve the quality of global model updates, which is able to enhance Semi-Async FL performance in the end.

III. Preliminary Case Studies and Motivation

We first outline the Semi-Async FL workflow, focusing on the staleness issue, followed by our observations related to model similarity that inspire our SAA strategy.

A. Semi-Async FL

Semi-Async FL allows the server to execute aggregation upon receiving some local models that are trained from global models derived several rounds ago. As in Fig. 2, in round $k+1$, Clients A and B upload their local models calculated from stale global model w_g^k that are derived from round $k-1$. The server then aggregates these local models to update the global model while Clients C and D are working on their own pace. Unfortunately, the staleness in local models will cause an increasing error to the aggregation [26].

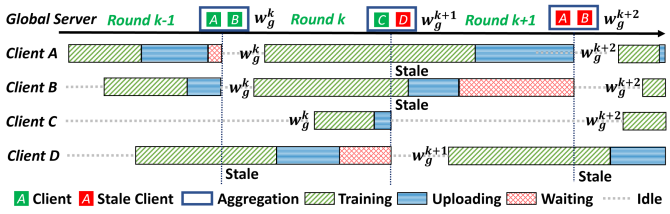
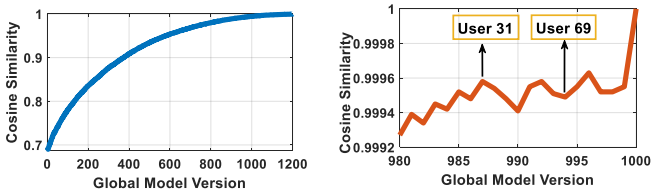


Fig. 2: Working procedure of the Semi-Async FL

B. Global Model Similarity

To shed light on the local model influence during aggregation in Semi-Async FL, we conduct an empirical study with 100 clients on a classification task using the CIFAR-10 dataset [28] over 1200 rounds. The server performs aggregation once receiving 10 local models. The data is split in a Non-IID setting via the Dirichlet distribution with a concentration parameter $r = 0.1$, creating severe imbalances among clients. Fig. 3a illustrates the changes in cosine similarity between the outdated and the newest global models during training. While the general trend suggests that more outdated models tend to exhibit lower similarity to the newest model, exceptions are observed, which persists throughout the training process. For demonstration purposes, we randomly select the 1000th aggregation for illustration and present the similarity between this round and its neighboring rounds in Fig. 3b. Here, the local models of clients 31 and 69 are trained from the global models derived in rounds 987 and 994, respectively. Although the outdated global model used by client 31 lags behind that used by client 69 by 7 rounds, it has a higher similarity with

the newest global model, indicating that the contribution of client 31 aligns more closely with the direction of the global update (if the similarity is 1, the outdated model is identical to the current global model, meaning staleness does not exist). This observation suggests that simply using the number of outdated rounds to measure a client's staleness and its local model influence on the global update does not accurately reflect the real situation.



(a) Over 1200 aggregations (b) Partial aggregations

Fig. 3: Cosine similarity between global models

C. Global model Update Stability.

We further investigate how the deviation of the global model update between consecutive aggregations affects the convergence performance under the same experimental settings as above. In particular, the global model deviation between two consecutive rounds k and $k-1$ is denoted as $\triangle w_g^k = w_g^k - w_g^{k-1}$, where w_g^k is the global model obtained in round k. We define the model similarity between two consecutive global models as $\Theta\langle w_g^k, w_g^{k-1}\rangle$ and corresponding directional similarity as $\Theta\langle \triangle w_g^k, \triangle w_g^{k-1}\rangle$, where $\Theta\langle A, B\rangle$ is the cosine function. The former one tells the similarity of the results while the latter one reflects the consistency of the driving forces behind such changes of the results. We calculate the mean values and the standard variances (Std) of directional similarities with different concentration parameters.

	Non-IID (Dirichlet Distribution)					IID
r	0.1	0.2	0.3	0.5	1	-
Mean	-0.421	-0.4191	-0.419	-0.423	-0.4237	-0.428
Std	0.1167	0.1056	0.1000	0.0956	0.0858	0.0812
Acc(%)	60.08	63.43	64.64	66.86	67.34	69.88

TABLE I: Cosine similarity between the global model variances under different data distribution

As illustrated in Table. I, $\Theta\langle \triangle w_g^k, \triangle w_g^{w^k}\rangle$ has a negative mean value, indicating that deviation substantially vibrates between two consecutive rounds. Meanwhile, the final accuracy is negatively correlated with the Std. of the directional similarity that reflects the stability of the global model update.

IV. System Model

We present an edge computing system involving a server and a set of clients $\mathcal{M} = \{1, 2, \cdots, M\}$. Clients, such as smartphones and autonomous vehicles, are equipped with heterogeneous computation and communication modules.

A. Model Training

The server and clients cooperatively perform Semi-Async FL to train a global model with the objective of minimizing the following global loss function,

$$\min_{w\in\mathbb{R}^d} F(w) := \sum_{m=1}^{M} p_m F_m(w), \tag{1}$$

where $F_m(w)$ is the local loss function of client m. The weight p_m is assigned to $F_m(w)$ when updating the global model.

The entire training procedure is divided into multiple rounds. In round k, the server temporarily caches the local models uploaded by a set of clients $\{m \in \mathcal{M}^k | \mathcal{M}^k \subseteq \mathcal{M}\}$ in its buffer [27]. At a specific time, those local models are aggregated to update the global model from w_g^k to w_g^{k+1} as

$$w_g^{k+1} = w_g^k + \eta_g \sum_{m\in\mathcal{M}^k} p_m (w_m^{k+1} - w_g^k), \tag{2}$$

where η_g is the global learning rate and w_m^{k+1} denotes the updated local model from the client m. Due to the asynchronous training manner, w_m^{k+1} may be trained from a stale global model. For client m, we use **round variance** τ_m to represent the round difference between the current round k and the round when it initiated the last training $k-\tau_m$. Hence, given a local learning rate η_m, the local update for client m is described as

$$w_m^{k+1}(\tau_m) = w_g^{k-\tau_m} - \eta_m \nabla F_m(w_g^{k-\tau_m}). \tag{3}$$

B. Computation and Communication

Next, we describe the time cost of the local model training and uploading for each client. It is divided into two main parts.

Computation. After retrieving the global model, client m performs E_m epochs of local training. Assuming the computation time per epoch is f_m^k, the computation time for the local training on client m in round k is given by

$$T_m^{k,comp} = E_m f_m^k. \tag{4}$$

Communication. Client m then uploads its local model to the server over the stationary wireless environment [29]. Denote the size of the local model in bits as Q_m. The time spent in uploading its local model in round k is expressed as

$$T_m^{k,trans} = \frac{Q_m}{B_m^k \log_2(1 + \frac{P_m^k |h_m^k|^2}{B_m^k N_0})}, \tag{5}$$

where B_m^k is the bandwidth allocated to client m in round k; P_m^k denotes its up-link transmit power; and N_0 is the power spectral density (PSD) of Additive White Gaussian Noise. The overall time for each local model update by client m is

$$T_m^k = T_m^{k,comp} + T_m^{k,trans}. \tag{6}$$

Because of the abundant computation resource and the high transmit power at the server, we ignore the computation time at the server and the global model distribution time.

C. Problem Formulation

In Semi-Async FL, since the clients' contribution to the global model update and their response times vary across rounds, the duration of every round is not consistent as well. We leverage wall-clock time, denoted as T_{wc}, to assess the performance of Semi-Async FL. This term represents the

overall training duration required to achieve a certain accuracy. Correspondingly, the total count of rounds is labeled as a wall-clock round. Assuming the wall-clock rounds as K_{wc} and the duration of the k-th round as ΔT^k, Semi-Async FL aims to optimize the training process by minimizing T_{wc} as follows

$$\min \sum_{k=1}^{K_{wc}} \Delta T^k = \sum_{k=1}^{K_{wc}} (T_{M^k}^k - T_1^k), \tag{7a}$$

$$\text{s.t.} \quad F(w^{K_{wc}}) \leq F(w^*) + \epsilon \tag{7b}$$

$$p_m \in [0,1], \forall m \in \mathcal{M} \tag{7c}$$

$$M^k \leq |\mathcal{M}|, \forall k. \tag{7d}$$

T_1^k and $T_{M^k}^k$ denote the time consumed for the first and last local updates in round k, respectively, where $M^k = |\mathcal{M}^k|$. Their expressions are in Eq. (6). Constraints (7b) and (7c) encapsulate the loss requirement and establish the boundary for the weight, respectively. Furthermore, Constraint (7d) prevents a deadlock, where the server awaits further local updates despite all clients having uploaded their local models.

V. Semi-Async FL with SAA strategy

We aim to minimize the wall-clock time of Semi-Async FL with a performance guarantee through a two-fold task: reducing the duration of each round and decreasing the number of wall-clock rounds. To achieve this, we focus on enhancing aggregation quality, which is able to address both objectives simultaneously. Drawing inspiration from our insights into the similarity-based local model influence and the stability of global model updates, we propose a novel Similarity-Aware Aggregation (SAA) strategy for Semi-Async FL.

A. Global Similarity-based Weight Allocation

Following the insight from our first observation, we formally measure the local model influence via the cosine similarity between the current global model w_g^k and the stale global model $w_g^{k-\tau_m}$ used for obtaining that local model as

$$s_g^{m,k} = \Theta\langle w_g^k, w_g^{k-\tau_m} \rangle, -1 \leq s_g^{m,k} \leq 1, \tag{8}$$

where a larger $s_g^{m,k}$ indicates that the local model $w_m^{k+1}(\tau_m)$ is more influential to current aggregation since the stale global model it used for local update is more similar to the fresh one.

However, as illustrated in Fig. 3b, the absolute values of $s_g^{m,k}$ for diverse local models are too proximate to discernibly differentiate their impacts during the aggregation. To tackle this issue, we leverage the Hinge function to augment model similarity, employing it as the new weight to better emphasize the distinct influence of each local model as follows,

$$p_m^k = \frac{\beta}{1 - s_g^{m,k} + \beta}. \tag{9}$$

In Eq. (9), $p_m^k \in [\frac{\beta}{2+\beta}, 1]$; and β, as a positive hyperparameter, is the scaling factor. The global update in Eq. (2) then becomes

$$w_g^{k+1} = w_g^k + \eta_g \sum_{m \in \mathcal{M}^k} \frac{\beta(w_m^{k+1} - w_g^k)}{1 - s_g^{m,k} + \beta}. \tag{10}$$

From Eq. (10), a larger weight is allocated to a more influential local model, enabling it to contribute more to the aggregation.

B. Directional Similarity-based Aggregation

The other challenge in enhancing global model update quality is to determine the aggregation timing. While integrating local models from a larger number of clients with Non-IID data distribution can improve aggregation quality, the server will spend a longer time waiting for multiple local models, thereby substantially increasing the wall-clock time. On the other hand, frequent aggregation will compromise aggregation quality since the global model cannot learn sufficient knowledge from a limited number of clients. This underlines the trade-off between waiting time and global model quality. To solve this problem, we design a directional similarity-based scheme, where the aggregation stability is used to guide the aggregation timing. Since the training accuracy is positively correlated to the standard deviation (Std.) of directional similarity as mentioned in Section III, our aggregation scheme aims at controlling the Std. to strike a balance between training accuracy and wall-clock time.

Consistent with Section III, we denote the global model update deviation in round k as $\triangle w_g^k$ and the direction similarity metric as $C^k = \Theta\langle \triangle w_g^k, \triangle w_g^{k-1} \rangle$. The stability of aggregation is reflected by the Std. of C^k. Assuming the server has collected a set of local models $\mathcal{M}_k$ in round k, C^k is then expressed as

$$\begin{aligned} C_{\mathcal{M}_k}^k &= \Theta\langle \triangle w_g^k, \triangle w_g^{k-1} \rangle \\ &= \Theta\langle \sum_{m \in \mathcal{M}_k} p_m (w_m^{k+1} - w_g^k), \triangle w_g^{k-1} \rangle \\ &= \frac{\sum_{m \in \mathcal{M}_k} \Theta\langle w_m^{k+1} - w_g^k, \triangle w_g^{k-1} \rangle \parallel w_m^{k+1} - w_g^k \parallel}{\sum_{m=1}^{n} \parallel w_m^{k+1} - w_g^k \parallel}, \end{aligned} \tag{11}$$

where the third equation is derived by substituting Eq. (3) into the second equation. Upon comparing the first and third equations, it is evident that the directional similarity is fundamentally the weighted collective of individual contribution $\Theta\langle w_m^{k+1} - w_g^k, \triangle w_g^{k-1} \rangle$. For a more holistic comprehension, Fig. 4 computes both individual contributions and directional similarity upon the arrival of a new local model at the server, under the same experiment setting in Section III-B.

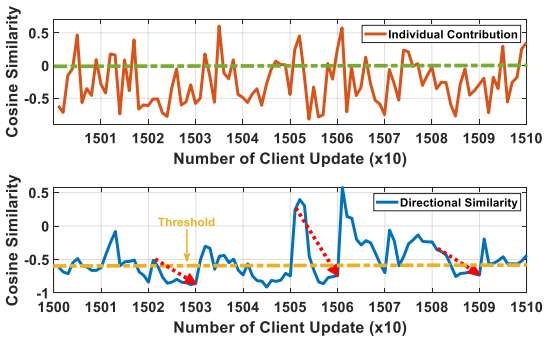


Fig. 4: Individual contribution vs. directional similarity

Fig.4 shows that individual contributions typically are negative, implying that the directional similarity tends to approach

the lower bound of -1 as more local models participate in the aggregation. We establish a threshold ρ where the server will not aggregate local models from the buffer until the collected models meet the following conditions

$$C^k_{\mathcal{M}^k} \leq \rho, \quad -1 \leq \rho \leq 1, \tag{12}$$

We define $M^k(\rho)$ as the number of models aggregated when the threshold ρ is reached. With the presence of the threshold ρ, the directional similarity is constrained within $[-1, \rho]$, which limits the Std. of the directional similarity to a relatively narrow range. A lower ρ would substantially increase the number of local models needed for depressing the directional similarity below the threshold in each round, hence prolonging the duration of each round. Conversely, increasing ρ would impair the quality of aggregation as the server can only learn from the limited data knowledge from a small set of clients. Notably, when $\rho = 1$, the server performs exactly FedAsync [26]. Based on this, we reformulate the wall-clock time minimization problem in Eq. $(7a)$ into the problem of determining the aggregation timing, where

$$\min_{\rho} \sum_{k=1}^{K_{wc}} \Delta T^k = \sum_{k=1}^{K_{wc}} (T^k_{M^k(\rho)} - T^k_1)) \tag{13a}$$

$$\text{s.t.} \quad (7b) - (7c) \tag{13b}$$

$$M^k(\rho) \in [M_l(\rho), M_u(\rho)]. \tag{13c}$$

The lower bound $M_l(\rho)$ in Constraint (13c) guarantees the minimum number of participating local models, and the upper bound $M_u(\rho)$ avoids the case that the threshold cannot be satisfied even with an excessive number of local models.

C. Semi-Async FL with SAA strategy

Combining the global similarity-based weight function and the directional similarity-based aggregation scheme, we propose an innovative Similarity-Aware Aggregation (SAA) strategy in Semi-Async FL. Our strategy is illustrated in Alg. 1 and Alg. 2. Particularly, each client uploads the local model and the version of the global model it used for training that local model to the server, as indicated in Line 2 and 3 of Alg. 2. The server caches a list of stale global models, which is updated every round following Line 15 in Alg. 1. The purpose of this list is to calculate the global similarity for each uploaded local model through Eq. (8).

Line 4-13 in Alg. 1 delineate ail the directional similarity-based aggregation scheme. Given the heterogeneity of computational resources of clients and the random Non-IID distribution of data, establishing a mapping from the threshold ρ to the set of participating clients $M^k(\rho)$ is extremely challenging. Directly solving for ρ is impractical, for which we design a heuristic approach to obtain the optimal value of ρ for aggregation timing. Specifically, we execute the SAA strategy with a batch of ρ and select the one with the smallest wall-clock time as the optimal value. Based on the experiments detailed in the following sections, we empirically provide a reasonable range for ρ, which is applicable to a variety of datasets.

We highlight the advantages of SAA over traditional Semi-Async FL methods [27]. The dynamic buffer in SAA allows for better control over aggregation stability, addressing a limitation in traditional methods that rely on a fixed buffer size without considering adaptability. Meanwhile, the server in SAA caches a list of stale global models, which enables a more precise measurement of local model influence compared to relying solely on round variance. Notably, the storage overhead for caching such a list is minimal, especially when compared to the terabytes of raw data typically handled by a high-performance server.

Algorithm 1 Semi-Async FL with SAA Strategy - Server

Input: Model cache list L_w, global learning rate η_g, aggregation threshold ρ, buffer size bound $[M_l(\rho), M_u(\rho)]$.
Output: Trained global model w_g, wall-clock time T_{wc}.

1: Aggregation round $k = 0$, global model version $O_g = 0$.
2: **while** $F(w^k_g) - F(w^*) > \epsilon$ **do**
3: Reset the directional similarity $C^k = 0$. Clear the buffer and set its size as $M^k = 0$.
4: **while** $C^k > \rho$ **do**
5: Receive trained local model w^{k+1}_m and client model version O^k_m from client m through wireless channel.
6: Calculate model staleness p^k_m by Eq. (8) and (9).
7: Update the directional similarity C^k by Eq. (11).
8: Buffer size increases by $M^k \leftarrow M^k + 1$
9: **if** $M^k = M_u(\rho)$ **break**
10: **end while**
11: **while** $M^k \leq M_l(\rho)$ **do**
12: Repeat Step 5 - 8.
13: **end while**
14: Update global model w^{k+1}_g by Eq. (10).
15: Update the cache list $L_w \leftarrow \{L_w, w^{k+1}_g\}$.
16: Global version increases by $O_g \leftarrow O_g + 1$.
17: Aggregation round increases by $k \leftarrow k + 1$.
18: **end while**

Algorithm 2 Semi-Async FL with SAA Strategy - Client

Input: Global model parameter from Server w_g, global model version O_g, base local learning rate η, decaying factor γ.
Output: Trained local model parameter w^{k+1}_m, version of the global model used for training O^k_m.

1: Initialize the local model $w^k_m \leftarrow w^k_g$ and the optimizer with learning rate $\eta_m \leftarrow \gamma^k \eta_m$.
2: Update the local model to w^{k+1}_m by Eq. (3).
3: Record the local model version $O^k_m \leftarrow O_g$.
Return: Trained local model w^{k+1}_m, model version O^k_m.

D. Convergence Analysis

We analyze the convergence rate of the Semi-Async FL with the SAA strategy. Referring to [27], we make the following common assumptions on the FL model:

Assumption 1 (*L-smoothness*) The F_m is L-smooth with a positive L for client $\forall m \in \mathcal{M}$. For $\forall w_1, w_2$, $F_m(w_2) - F_m(w_1) \leq \langle \nabla F_m(w_1), w_2 - w_1 \rangle + \frac{L}{2} \parallel w_2 - w_1 \parallel^2$

According to the Theorem 2.1.8 in [30], the following condition holds as well,

$$\parallel \nabla F_m(w_2) - \nabla F_m(w_1) \parallel^2 \leq L \langle \nabla F_m(w_2) - \nabla F_m(w_1), w_2 - w_1 \rangle \quad (14)$$

Assumption 2 (μ-*convexity*) The F_m is strong convex with a positive μ for client $\forall m \in \mathcal{M}$. For $\forall w_1, w_2$, $F_m(w_2) - F_m(w_1) \geq \langle \nabla F_m(w_1), w_2 - w_1 \rangle + \frac{\mu}{2} \parallel w_2 - w_1 \parallel^2$.

According to Theorem 2.1.19 in [30], the following also holds

$$\langle \nabla F_m(w_2), w_2 - w_1 \rangle \leq F_m(w_2) - F_m(w_1) + \frac{\parallel \nabla F_m(w_2) - \nabla F_m(w_1) \parallel^2}{2\mu} \quad (15)$$

Assumption 3 (*Global optimal*) Assume that there exists w^* that minimizes the global loss function as $\nabla F(w^*) = 0$.

Assumption 4 (*Local optimal bound*) Given w^*, the local loss function is bound as $\parallel \nabla F_m(w^*) \parallel^2 \leq G$ for all $m \in \mathcal{M}$.

Assumption 5 (*Local learning rate bound*) For any local optimizer, the learning rate η_m is bounded as $\eta_m \in [\eta_l, \eta_u]$, where both lower bound and upper bound are non-negative.

Theorem 1. If $\frac{\eta_u^2}{\eta_l} < \frac{L^2}{2\mu}$ and $\eta_g < \frac{2+\beta}{\beta M_l(\rho)(2\eta_l\mu - \eta_u^2 L^2)}$ hold, the loss of the global model in Alg. 1 satisfies the following equation after k rounds

$$\begin{aligned} & F(w_g^k) - F(w^*) \\ & \leq [1 - \frac{\eta_g \beta M_l(\rho)}{2+\beta}(2\eta_l\mu - \eta_u^2 L^2)]^{\frac{k}{1+\tau_{max}}} (F(w^0) - F(w^*)) \\ & + \frac{(2+\beta) M_u(\rho) \eta_u^2 LG}{\beta M_l(\rho)(2\eta_l\mu - \eta_u^2 L^2)}. \end{aligned} \quad (16)$$

The first item on the right side can be presented as $\zeta^k(F(w^0) - F(w^*))$, where the ζ denotes the convergence rate, and the second item is a residual error, denoted as ϵ. The proof is referred to in the Appendix.

Corollary 1. There is a trade-off in the faster convergence rate and smaller residual error when selecting the scaling factor β. Hence, there exists an optimal value of β.

Corollary 2. Given a fixed β and η_g, the upper bound of buffer size $M_l(\rho)$ should be smaller than $\lfloor \frac{2+\beta}{\beta \eta_g (2\eta_l\mu - \eta_u^2 L^2)} \rfloor$ to guarantee the convergence of the leaning model. The function $\lfloor x \rfloor$ is the maximum integer that smaller than x.

E. Complexity Analysis

We mainly focus on the time complexity of the server because the working procedure of clients is similar to that in the conventional Semi-Async FL. For the learning model with N-dimensions of parameters, we define the time complexity of each element-wise operation on the entire model as $\mathcal{O}(N)$. For the original Semi-Async FL with a fixed buffer size M, the total time complexity for each round is $\mathcal{O}(MN)$. The extra computation of SAA consists of the operation of calculating the global model similarity and directional similarity. The cosine similarity operation requires $(4N + \delta)$ computations, where δ refers to scalar computations and $\delta \ll N$. In the worst case, the buffer size always reaches its maximum $M_u(\rho)$. Assuming that L times of searching for the optimal ρ are executed, the time complexity becomes $\mathcal{O}(M_u(\rho)NL)$, which is still linear in this worst case.

VI. Performance Evaluation

We conduct all experiments under FLSiM frame [27] on a desktop with GeForce RTX 3060 graphic card.

A. Experiment Setup

System setting. We evaluate the SAA strategy with both IID and Non-IID data distribution at multiple clients in edge computing systems. The distance between each client and the server is uniformly distributed between 100 and 500 meters. Client transmission power (P) is fixed at 23dBm, with a bandwidth (B_m) of 20MHz centered at a frequency of 2100MHz. We establish the Power Spectral Density (PSD) of Additive White Gaussian Noise (AWGN) as $1e^{-}14.7mW/Hz$. To accurately simulate the wireless environment in edge computing systems, we adopt the widely accepted WINNER II channel model [31] that is commonly used in urban areas. Furthermore, in alignment with [27], we assume that the computation time per epoch adheres to a half-normal distribution ($f_m \sim NH(0, \sigma^2)$) with $\sigma = 0.8$.

Dataset and models. We conduct extensive experiments on three real-world datasets with different neural network models. The details are shown in Table. III. We apply SGD optimizers, Cosine Annealing learning rate schedulers, and cross-entropy loss functions for all experiments. The global learning rate η_g during the aggregation is set as 1.

Since we are comparing the training efficiency of proposed schemes with benchmarks, the experiments will be cut off once the server has aggregated $10,000$ local models in total for CINIC-10 and CIFAR-100 datasets, and $20,000$ for the EMNIST dataset in all schemes, which are adequate for the convergence in most cases.

Dataset	EMNIST	CINIC-10	CIFAR-100
Model	LeNet-5	MobileNet-V2	ResNet18
# of clients	100	50	20
Learning rate	0.001	0.0005	0.01
Batch size	128	64	32

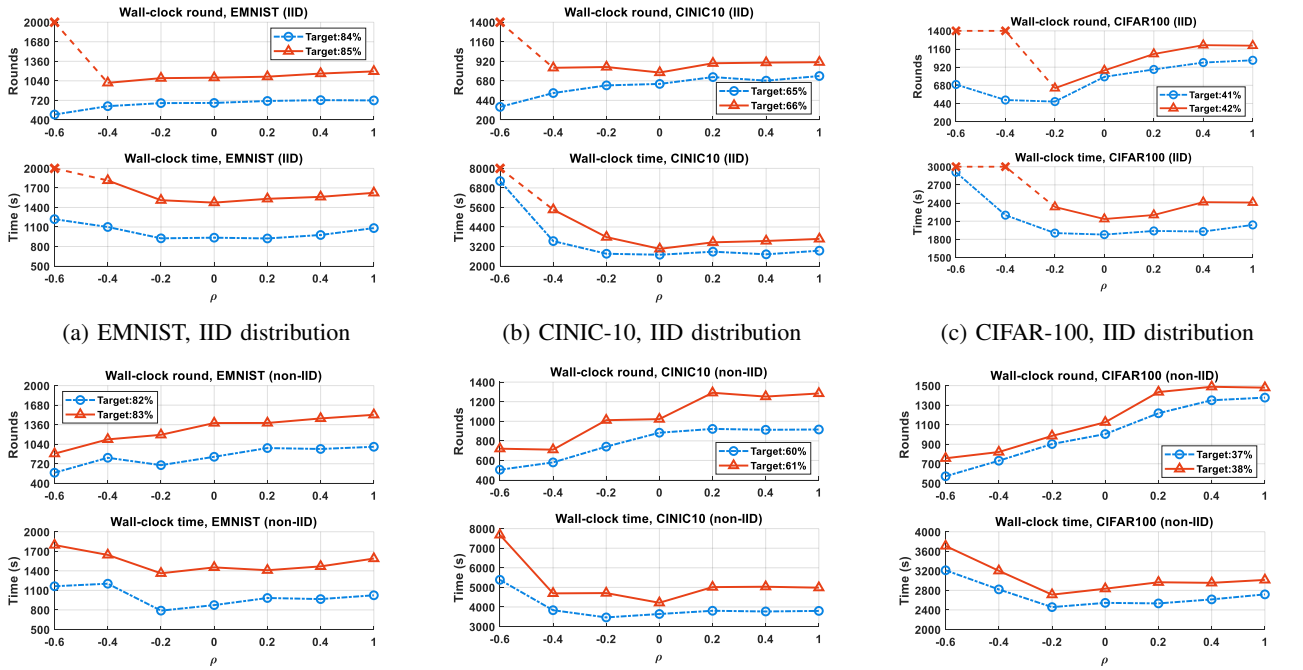
TABLE III: Dataset and training model

Benchmark schemes. We compare the Semi-Async FL with our SAA strategy with three benchmark schemes.

- FedAsync [26]: The standard fully-Async FL, where aggregation is performed once the server receives a local model from any client. Thus, the number of total rounds is equal to the total amount of client updates.
- FedBuff [27]: The Semi-Async FL with a fixed buffer size and a round variance-based weight function. The total number of total rounds is equal to the client updates divided by the buffer size
- FedMAX: Its buffer size equals the buffer upper bound in SAA, which is significantly larger than that in FedBuff.

Dataset (Model)	Evaluation Metric	Weight Parameter	Constant \	Polynomial $\alpha = 0.3$	Polynomial 0.7	Similarity $\beta = 1e-5$	Similarity $1e-4$	Similarity $1e-3$	Similarity $1e-2$
EMNIST (LeNet5)	Final Acc.	IID	0.8431	0.8454	0.8475	0.8477	**0.8547**	0.8542	0.8442
		Non-IID	0.8211	0.8197	0.8220	0.8240	**0.8370**	0.8350	0.8277
	WC Time (s)	IID (84%)	1420.4	1432.3	1293.3	1209.3	**1083.1**	1131.3	1244.1
		Non-IID (83%)	1824.3	1733.8	1669.1	1650.7	1588.2	**1587.4**	1629.3
CINIC-10 (MobileNetV2)	Final Acc.	IID	0.6608	0.6643	0.6638	0.6680	**0.6743**	0.6729	0.6690
		Non-IID	0.6324	0.6340	0.6408	0.6410	0.6510	**0.6520**	0.6430
	WC Time (s)	IID (65%)	3880.1	4012.7	3882.4	3722.1	**3667.5**	3706.6	3768.3
		Non-IID (60%)	5103.1	5111.7	5003.8	5013.8	**4984.9**	4974.3	5023.8
CIFAR-100 (ResNet18)	Final Acc.	IID	0.4170	0.4211	0.4228	0.4205	0.4306	**0.4340**	0.4242
		Non-IID	0.4043	0.4083	0.4075	0.4089	**0.4151**	0.4146	0.4045
	WC Time (s)	IID (41%)	2345.2	2287.1	2163.2	2071.6	2037.9	**2025.3**	2123.4
		Non-IID (38%)	3014.8	3008.2	3011.6	2991.4	**2812.7**	2841.5	3008.7

TABLE II: The final accuracy (Final Acc.) and wall-clock time (WC Time) under different weight functions



(a) EMNIST, IID distribution (b) CINIC-10, IID distribution (c) CIFAR-100, IID distribution

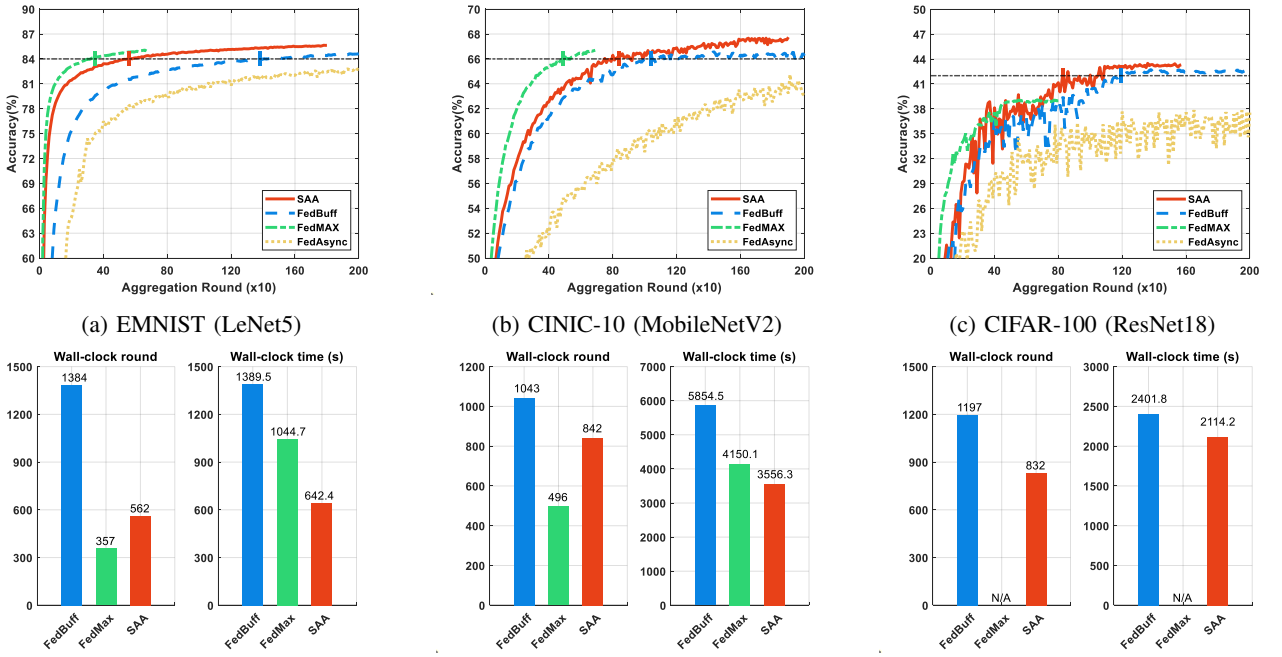
(d) EMNIST, non-IID distribution (r=0.3) (e) CINIC-10, non-IID distribution (r=0.3) (f) CIFAR-100,non-IID distribution(r=0.9)

Fig. 5: The impact of ρ on the wall-clock round and wall-clock time for different target accuracy

Performance of the similarity-based weight function. We compare the proposed weight function in Eq. (9) with the constant weight and the polynomial weight [32], both of which measure the staleness with round variance. The concentration parameter for non-IID setting is set to $r = 0.3$ for EMNIST and CINIC-10 whereas $r = 0.9$ for CIFAR-100. Note that the directional similarity-based aggregation scheme is not applied here for the ablation study. Table. II shows that the proposed weight function with proper scaling factor considerably improves the accuracy compared with others. Moreover, it dominates the reduction of the wall-clock time under various settings. Especially for the EMNIST dataset with a large number of clients, SAA reduces about 33% wall-clock time under IID data distribution. These observations indicate that the similarity-based weight function remarkably improves the global model update quality.

Note that either the scaling factor β is too large like 0.1 or too small like $1e-5$ cannot bring a significant benefit. This owes to the fact that a large β will enforce the weight to approach 1 whereas a small β is not sufficient to augment the local model influence in our weight function. As indicated in Table. II, $\beta = 1e-4$ demonstrates to be a generally good setting for SAA strategy among most of the datasets and models and thus it is suggested to be used in further research.

Impact of the aggregation threshold. We investigate the impact of threshold ρ on wall-clock time and wall-clock rounds for low and high accuracy requirements, the results of which are shown in Fig. 5. First of all, compared to the results shown in Table. II, introducing the directional similarity-based aggregation scheme dramatically improves the performance on both metrics. In details, the wall-clock time generally decreases as ρ becomes smaller, particularly in the case of non-IID distributions. It is worthwhile to note that the high accuracy of in EMNIST (85%) cannot be satisfied when $\rho = -0.6$ and in CIFAR-100 (42%) when $\rho \leq -0.4$. This is because a small ρ will induce the server to aggregate more local



(a) EMNIST (LeNet5) (b) CINIC-10 (MobileNetV2) (c) CIFAR-100 (ResNet18)

(d) EMNIST, Target Accuracy (84%) (e) CINIC-10, Target Accuracy (66%) (f) CIFAR-100, Target Accuracy (42%)

Fig. 6: Comparison of accuracy, wall-clock time and wall-clock rounds with different dataset and models

models each round to meet the threshold requirement, resulting in insufficient rounds of aggregation. In contrast, when the threshold is too large, i.e., $\rho \geq 0.2$, SAA will merely benefit from the aggregation scheme with trivial improvement. This is because a large ρ encourages more frequent aggregation pattern but lower update quality of global model. Empirically, a suggested value for the aggregation threshold ρ is around -0.2 and this is applied in the following experiments.

FL scheme	FedAsync	FedBuff	SAA	FedMax
average round variance	40.0	8.6	4.7	3.2
standard deviation	21.3	3.5	1.8	1.2
maximum round variance	122	22	18	14

TABLE IV: Version variance of different schemes

Performance of the SAA strategy. Fig. 6 compares the SAA strategy with benchmarks under the same suggested values of scaling factors and aggregation threshold. Within a limited total number of local model updates, FedAsync cannot reach a target accuracy in any dataset. The reason is demonstrated in Table. IV using the example of EMNIST dataset. The fully-Async scheme suffers from an ultra-high round variance with a mean value of 40. Compared with the conventional Semi-Async FL scheme with a fixed aggregation buffer, our SAA strategy reduces 45% average round variance with its stability-guided dynamic aggregation scheme.

For EMNIST and CINIC-10 datasets, although FedMAX can reduce more wall-clock rounds than the SAA strategy, too many local models being aggregated each round deteriorates the global model update quality. Consequently, FedMAX, compared to SAA, causes 38% more wall-clock time in EMNIST and even worse, the accuracy requirement in CIFAR-100 cannot be satisfied. In contrast, the SAA reduces 53.7%, 39.3%, and 12.0% wall-clock time, as well as 59.4%, 23.9%, and 30.5% wall-clock round on three datasets compared with the conventional Semi-Async FL scheme. Hence the dynamic buffer applied in SAA demonstrates a dominant performance in reducing the wall-clock time and wall-clock round in the case of limited local updates.

VII. Conclusion

In this paper, we accelerate the Semi-Async FL deployment to adapt to rapidly changing AI service demands in edge computing. Aiming at minimizing wall-clock time, we design an aggregation strategy SAA based on two observations on similarity. Specifically, the influence of a local model can be better measured by the cosine similarity between the stale and fresh global model than the round variance. Additionally, the global model update stability, described by the standard deviation of directional similarity, exerts a positive impact on model accuracy. Therefore, we first design a global similarity-based weight function to engage local models with higher influence in the aggregation. We then develop a directional similarity-based aggregation scheme to determine the aggregation timing with great control on the global model update stability. In theory, we provide the convergence guarantee to SAA. Our experimental results empirically demonstrated that SAA outperforms benchmarks in terms of training efficiency and final accuracy with different data datasets and models.

Acknowledgement

The work of L. Guo is partially supported NSF under grant CNS-2008049, CCF-2312616, and CCF-2427875. The work of X. Zhang is supported NSF under grant CCF-2312617.

APPENDIX

Proof of Theorem 1: For simplicity, we use $\sum_{\mathcal{M}^{k-1}}$ to represent the sum of m clients $\sum_{m\in\mathcal{M}^{k-1}(\rho)}$. With the convexity assumption of F, we have

$$\begin{aligned}
&F(w_g^k) - F(w^*)\\
&= F(w_g^{k-1} + \eta_g \sum_{\mathcal{M}^{k-1}} p_m(w_m^k - w_g^{k-1})) - F(w^*)\\
&\le (1-\eta_g \sum_{\mathcal{M}^{k-1}} p_m)F(w_g^{k-1}) + \eta_g \sum_{\mathcal{M}^{k-1}} p_m F(w_m^k) - F(w^*)\\
&= (1-\eta_g \sum_{\mathcal{M}^{k-1}} p_m)T_{k-1} + \eta_g \sum_{\mathcal{M}^{k-1}} p_m T_{\tau_m}^m, \qquad (17)
\end{aligned}$$

where $T_{k-1} = F(w^{k-1})_g - F(w^*)$ and $T_{\tau_m}^m = F(w_m^k) - F(w^*)$. The $T_{\tau_m}^m$ is represented as

$$\begin{aligned}
T_{\tau_m}^m &= F(w_m^k) - F(w^*)\\
&= (F(w_g^{\tau_m}) - F(w^*)) + (F(w_m^k) - F(w_g^{\tau_m})). \qquad (18)
\end{aligned}$$

Assumption 1 can be easily extended to the global loss function, which is commonly applied in FL [26] [27]. Then, we have the second item in $T_{\tau_m}^m$ as

$$\begin{aligned}
&F(w_m^k) - F(w_g^{\tau_m})\\
&\le \frac{L}{2} \parallel w_m^k - w_g^{\tau_m} \parallel^2 + \langle \nabla F(w_g^{\tau_m}), w_m^k - w_g^{\tau_m}\rangle\\
&\le \eta_i^2 L \parallel \nabla F_m(w_g^{\tau_m}) - \nabla F_m(w^*) \parallel^2 + \eta_i^2 L \parallel \nabla F_m(w^*) \parallel^2\\
&\quad -\eta_i \langle \nabla F(w_g^{\tau_m}), \nabla F_m(w_g^{\tau_m})\rangle. \qquad (19)
\end{aligned}$$

From Assumption 4, we obtain that

$$\begin{aligned}
&\sum_{\mathcal{M}^{k-1}} p_m(F(w_m^k) - F(w_g^{\tau_m}))\\
&\le \eta_u^2 L \sum_{\mathcal{M}^{k-1}} p_m \parallel \nabla F_m(w_g^{\tau_m}) - \nabla F_m(w^*) \parallel^2 \qquad (20)\\
&\quad +\eta_u^2 L \sum_{\mathcal{M}^{k-1}} p_m \parallel \nabla F_m(w^*) \parallel^2 \qquad (21)\\
&\quad -\eta_l \sum_{\mathcal{M}^{k-1}} p_m \langle \nabla F(w_g^{\tau_m}), \nabla F_m(w_g^{\tau_m})\rangle. \qquad (22)
\end{aligned}$$

Given the definition of the global loss function, it is easy to *[XZ: approve]* that $\sum_{\mathcal{M}^{k-1}} p_m \nabla F_m(w^*) = \nabla F(w^*) = 0$. We also have $\sum_{\mathcal{M}^{k-1}} p_m \nabla F_m(w_g^{\tau_m}) = \sum_{\mathcal{M}^{k-1}} p_m \nabla F(w_g^{\tau_m})$. Combining L-smoothness of function and convexity given in Theorem 2.1.19 [33], Eq. (20) is expressed as

$$\begin{aligned}
&\eta_u^2 L \sum_{\mathcal{M}^{k-1}} p_m \parallel \nabla F_m(w_g^{\tau_m}) - \nabla F_m(w^*) \parallel^2\\
&\le (\eta_u L)^2 \sum_{\mathcal{M}^{k-1}} p_m \langle \nabla F_m(w_g^{\tau_m}) - \nabla F_m(w^*), w_g^{\tau_m} - w^*\rangle\\
&\le (\eta_u L)^2 \sum_{\mathcal{M}^{k-1}} p_m \langle \nabla F(w_g^{\tau_m}), w_g^{\tau_m} - w^*\rangle\\
&\le (\eta_u L)^2 \sum_{\mathcal{M}^{k-1}} p_m (F(w_g^{\tau_m}) - F(w^*) + \frac{\parallel \nabla F(w_g^{\tau_m}) \parallel^2}{2\mu})\\
&\le \frac{(\eta_u L)^2}{2\mu} \sum_{\mathcal{M}^{k-1}} p_m \parallel \nabla F(w_g^{\tau_m}) \parallel^2 . \qquad (23)
\end{aligned}$$

We first uniform the expression of Eq. (22) and Eq. (23) by rewriting Eq. (22) as $-\eta_l \sum_{\mathcal{M}^{k-1}} p_m \parallel \nabla F(w_g^{\tau_m}) \parallel^2$. We then sum them to obtain

$$\begin{aligned}
&-(\eta_l - \frac{(\eta_u L)^2}{2\mu}) \sum_{\mathcal{M}^{k-1}} p_m \parallel \nabla F(w_g^{\tau_m}) \parallel^2\\
&= -(\eta_l - \frac{(\eta_u L)^2}{2\mu}) \sum_{\mathcal{M}^{k-1}} p_m \parallel \nabla F(w_g^{\tau_m}) - \nabla F(w^*) \parallel^2\\
&\le -(2\eta_l\mu - \eta_u^2 L^2) \sum_{\mathcal{M}^{k-1}} p_m (F(w_g^{\tau_m}) - F(w^*)). \qquad (24)
\end{aligned}$$

By integrating Eq. (18), (21) and (24), we have

$$\begin{aligned}
\sum_{\mathcal{M}^{k-1}} p_m T_{\tau_m}^m &= \eta_u^2 L \sum_{\mathcal{M}^{k-1}} p_m \parallel \nabla F_m(w^*) \parallel^2\\
&+(1-(2\eta_l\mu - \eta_u^2 L^2)) \sum_{\mathcal{M}^{k-1}} p_m (F(w_g^{\tau_m}) - F(w^*)). \qquad (25)
\end{aligned}$$

With $a^k = 1-\eta_g \sum_{\mathcal{M}^{k-1}} p_m$, $b_i^k = \eta_g p_m[1-(2\eta_l\mu-\eta_u^2 L^2)]$ and $c^k = \eta_g \eta_u^2 L \sum_{\mathcal{M}^{k-1}} p_m \parallel \nabla F_m(w^*) \parallel^2$, the target function is transformed to

$$F(w_g^k) - F(w^*) = T_k = a^k T_{k-1} + \sum_{\mathcal{M}^{k-1}} b_i^k T_{\tau_m} + c^k \qquad (26)$$

LEMMA 1: Under the assumption that a^k, b_i^k and c^k are non-negative, we define $\theta^k = a^k + \sum_{\mathcal{M}^{k-1}} b_i^k < 1$; $\zeta = \theta_{max}^{\frac{1}{1+\tau_{max}}}$ and $\epsilon = \frac{c_{max}}{1-\theta_{max}}$. If Eq. (26) holds, it is true that

$$T_k \le \zeta^k T_0 + \epsilon. \qquad (27)$$

Lemma 1 is proved in [8]. In our scenario, θ should satisfy

$$\begin{aligned}
\theta^k &= 1 - \eta_g \sum_{\mathcal{M}^{k-1}} p_m + \eta_g \sum_{\mathcal{M}^{k-1}} p_m[1-(2\eta_l\mu - \eta_u^2 L^2)]\\
&= 1 - \eta_g(2\eta_l\mu - \eta_u^2 L^2) \sum_{\mathcal{M}^{k-1}} p_m \in (0,1). \qquad (28)
\end{aligned}$$

We then have the bound of local learning rate that satisfies

$$\frac{\eta_u^2}{\eta_l} < \frac{L^2}{2\mu}. \qquad (29)$$

Define the range of the sum of weights as $\sum_{\mathcal{M}^{k-1}} p_m \in [p_l, p_u]$. Following the aggregation with the magnified global model similarity in Eq. (10), we have $p_m \in [\frac{\beta}{2+\beta}, 1]$. Based on Assumption 6, we have the boundary $p_u = M_u(\rho)$ and $p_l = \frac{\beta}{2+\beta} M_l(\rho)$. The global learning rate should satisfy

$$\eta_g < \frac{2+\beta}{\beta M_l(\rho)(2\eta_l\mu - \eta_u^2 L^2)}. \qquad (30)$$

We then have $\theta_{max} = 1 - \frac{\beta\eta_g}{2+\beta} M_l(\rho)(2\eta_l\mu - \eta_u^2 L^2)$. Following the bounded local optimal in Assumption 4, we have $c_{max} = M_u(\rho)\eta_g\eta_u^2 LG$. Hence, if Eq. (29) and Eq. (30) are both satisfied, the following formulation holds

$$F(w_g^k) - F(w^*) \le \zeta^k (F(w_g^0) - F(w^*)) + \epsilon, \qquad (31)$$

where $\zeta = [1-\frac{\eta_g\beta M_l(\rho)}{2+\beta}(2\eta_l\mu-\eta_u^2 L^2)]^{\frac{1}{1+\tau_{max}}}$ and the residual error $\epsilon = \frac{(2+\beta)M_u(\rho)\eta_u^2 LG}{\beta M_l(\rho)(2\eta_l\mu-\eta_u^2 L^2)}$.

REFERENCES

[1] 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.

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

[3] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Communications Surveys & Tutorials*, 2021.

[4] Y. Xiao, G. Shi, and M. Krunz, "Towards ubiquitous ai in 6g with federated learning," *arXiv preprint arXiv:2004.13563*, 2020.

[5] P. S. Bouzinis, P. D. Diamantoulakis, and G. K. Karagiannidis, "Wireless federated learning (wfl) for 6g networks4 part i: Research challenges and future trends," *IEEE Communications Letters*, vol. 26, no. 1, pp. 3–7, 2021.

[6] W. Duan, J. Gu, M. Wen, G. Zhang, Y. Ji, and S. Mumtaz, "Emerging technologies for 5g-iov networks: Applications, trends and opportunities," *IEEE Network*, vol. 34, no. 5, pp. 283–289, 2020.

[7] H. R. Feyzmahdavian, A. Aytekin, and M. Johansson, "A delayed proximal gradient method with linear convergence rate," in *2014 IEEE international workshop on machine learning for signal processing (MLSP)*. IEEE, 2014, pp. 1–6.

[8] Q. Ma, Y. Xu, H. Xu, Z. Jiang, L. Huang, and H. Huang, "Fedsa: A semi-asynchronous federated learning mechanism in heterogeneous edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3654–3672, 2021.

[9] G. Shi, L. Li, J. Wang, W. Chen, K. Ye, and C. Xu, "Hysync: Hybrid federated learning with effective synchronization," in *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2020, pp. 628–633.

[10] C. Zhou, H. Tian, H. Zhang, J. Zhang, M. Dong, and J. Jia, "Tea-fed: time-efficient asynchronous federated learning for edge computing," in *Proceedings of the 18th ACM International Conference on Computing Frontiers*, 2021, pp. 30–37.

[11] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, "Asynchronous online federated learning for edge devices with non-iid data," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 15–24.

[12] J. Hao, Y. Zhao, and J. Zhang, "Time efficient federated learning with semi-asynchronous communication," in *2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, 2020, pp. 156–163.

[13] A. Imteaj and M. H. Amini, "Fedar: Activity and resource-aware federated learning model for distributed mobile robots," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2020, pp. 1153–1160.

[14] C.-H. Hu, Z. Chen, and E. G. Larsson, "Device scheduling and update aggregation policies for asynchronous federated learning," in *2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2021, pp. 281–285.

[15] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning and permissioned blockchain for digital twin edge networks," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2276–2288, 2021.

[16] ——, "Communication-efficient federated learning and permissioned blockchain for digital twin edge networks," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2276–2288, 2020.

[17] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146–1159, 2019.

[18] Y. Zhang, M. Duan, D. Liu, L. Li, A. Ren, X. Chen, Y. Tan, and C. Wang, "Csafl: A clustered semi-asynchronous federated learning framework," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–10.

[19] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G. Xing, "Clusterfl: A similarity-aware federated learning system for human activity recognition," in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '21. New York, NY, USA: ACM, 2021, p. 54–66.

[20] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3710–3722, 2021.

[21] M. Duan, D. Liu, X. Ji, Y. Wu, L. Liang, X. Chen, Y. Tan, and A. Ren, "Flexible clustered federated learning for client-level data distribution shift," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2661–2674, 2022.

[22] Y. Deng, F. Lyu, J. Ren, Y.-C. Chen, P. Yang, Y. Zhou, and Y. Zhang, "Fair: Quality-aware federated learning with precise user incentive and model aggregation," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[23] S. Wang, M. Lee, S. Hosseinalipour, R. Morabito, M. Chiang, and C. G. Brinton, "Device sampling for heterogeneous federated learning: Theory, algorithms, and implementation," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[24] 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*, 2021, pp. 1–10.

[25] W. Wu, L. He, W. Lin, R. Mao, C. Maple, and S. Jarvis, "Safa: A semi-asynchronous protocol for fast federated learning with low overhead," *IEEE Transactions on Computers*, vol. 70, no. 5, pp. 655–668, 2021.

[26] X. Cong, K. Oluwasanmi, and G. Indranil, "Asynchronous federated optimization," arXiv:1903.03934, 2019.

[27] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, and D. Huba, "Federated learning with buffered asynchronous aggregation," in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, vol. 151. PMLR, 28–30 Mar 2022, pp. 3581–3607.

[28] A. Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto*, 05 2012.

[29] Q. Tang, R. Xie, F. R. Yu, T. Huang, and Y. Liu, "Decentralized computation offloading in iot fog computing system with energy harvesting: A dec-pomdp approach," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4898–4911, 2020.

[30] N. Yurii, "Introductory lectures on convex optimization: A basic course," *Springer*, vol. 87, 2003.

[31] P. Kyösti, J. Meinilä, and L. Hentila, "Winner ii channel models," *Radio Technologies and Concepts for IMT-Advanced*, pp. 39–92, 02 2008.

[32] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, pp. 4229–4238, 2020.

[33] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, 1st ed. Springer Publishing Company, Incorporated, 2014.