MR.AsyncFL: Staleness-Aware Aggregation for Asynchronous Federated Learning via Model Replacement

Anonymous authors
Paper under double-blind review

Abstract

Asynchronous federated learning (AFL) has become increasingly popular and crucial for scalable, privacy-preserving machine learning across diverse and distributed edge devices. However, a fundamental challenge in FL is the staleness of client updates, which can degrade global model accuracy and slow down convergence as clients operate and communicate independently. Existing AFL methods typically address this issue by down-weighting stale updates, but outdated client information may still persist in the global model over time. In this paper, we propose MR.AsyncFL, a fully AFL framework based on *model replacement*. Upon receiving a new local model from a client, the server replaces that client's previously cached contribution in the global model with the updated one, preserving the invariant that the global model remains a convex combination of the most recent available client models. To support this mechanism, we propose a recursive weight update scheme that preserves normalization in a lightweight and fully asynchronous manner. We further provide a convergence analysis for MR.AsyncFL under bounded staleness and client participation assumptions, and derive an $\mathcal{O}(T^{-1/4})$ convergence rate under a specific parameter scaling. Experiments on CIFAR-10 and CIFAR-100 under both IID and non-IID settings, with and without staleness thresholds, show that MR.AsyncFL consistently outperforms representative asynchronous baselines, such as FedAsync, TWAFL, and Rolling FedAvg, while maintaining strong robustness under severe staleness and system heterogeneity.

1 Introduction

Massive amounts of data are continually generated by various devices, sensors and mobile applications across the Internet. At the same time, privacy regulations and rising public concern about data misuse have made it untenable to centralize sensitive data. Federated learning (FL) has emerged as a promising solution that trains machine learning (ML) models collaboratively across a network of clients while keeping raw data local (McMahan et al., 2016). By exchanging only model updates instead of raw data, FL mitigates privacy leakage and reduces communication burden. This privacy-preserving paradigm has already seen wide adoption in sensitive domains such as smart-healthcare systems (Wu et al., 2020) and autonomous driving applications (Chen et al., 2015) where personal data must not be disclosed. Similar opportunities are being explored in internet-of-things (IoT) and smart-city contexts, where FL enables decentralized model training on distributed sensors and devices (Ma et al., 2025; Pandya et al., 2023).

As FL scales to larger and more diverse networks of mobile phones, edge devices and organizational silos, the underlying infrastructure becomes highly heterogeneous. Clients vary widely in computation power, storage, energy consumption and connectivity. Their datasets are often non-independent and identically distributed (non-IID). Most FL algorithms adopt a synchronous federated learning (SFL) framework, such as FedAvg, in which the server broadcasts a global model, selected clients perform local training, and the server waits for *all* selected clients to upload their updates before aggregation. In such synchronous systems, clients with limited computing power, unstable connectivity, or temporary unavailability become stragglers. Since the server cannot proceed to the next round until updates from all selected clients are received, these slower or offline clients delay the entire training cycle. Consequently, faster clients remain idle, and overall training

speed is constrained by the slowest participant, resulting in poor resource utilization and prolonged training times.

Asynchronous federated learning (AFL) addresses this bottleneck by eliminating strict synchronization at the aggregation stage. Rather than waiting for all selected clients, the server updates the global model upon receiving each client update, thereby preventing stragglers from delaying system-wide progress. By decoupling client updates from strict synchronization, AFL drastically and dramatically reduces waiting time and better utilizes available client resources.

However, the flexibility of AFL introduces a fundamental challenge: staleness. Because local training on clients and global aggregations on the server occur independently, the slow clients may have been training over a long period of time using an outdated global model that had been updated multiple times by faster clients' local updates. Consequently, the server may receive updates computed from stale models from the slow clients. Aggregating such outdated gradients can destabilize training, introduce inconsistent model updates, and degrade final accuracy. The timing of aggregation introduces a trade-off in client model staleness. Fully AFL can accelerate global model convergence by immediately incorporating updates, but it increases staleness, which may amplify training bias and risk divergence. In contrast, SFL eliminates staleness by waiting for all selected clients before aggregation, but this improves consistency at the cost of lower resource utilization and longer overall training time. Designing aggregation algorithms that can robustly handle staleness is therefore a central challenge in asynchronous federated optimization (Singh & Adhikari, 2026). To address this challenge, we introduce MR.AsyncFL, an asynchronous federated learning framework based on model replacement. Our proposed solution is designed to counteract the detrimental impact of staled updates. By incorporating mechanisms that adaptively account for update freshness, MR.AsyncFL enhances training stability, accelerates convergence, and improves final model accuracy in environments with heterogeneous client and network conditions. The main contributions of this paper can be summarized as follows:

- We propose MR.AsyncFL, a novel AFL framework that introduces a stale-model substitution paradigm for asynchronous aggregation. In contrast to existing asynchronous methods that mainly attenuate stale updates through temporal decay or staleness-aware weighting (Xie et al., 2019; Chen et al., 2019; Zhu et al., 2022), MR.AsyncFL uses each newly uploaded local model in two occasions: first, it is used to replaces the same client's previously cached stale contribution, and second, it participates in the aggregation step that produces the next global model. This design reduces the lingering effect of outdated information and helps mitigate staleness-induced drift over time.
- We design a recursive weight normalization scheme that dynamically adjusts each client's contribution based on its update delay and participation history. The proposed scheme maintains normalization without requiring additional buffers or optimizer-specific modifications, enabling stable and efficient integration of stale-model replacement in fully asynchronous settings.
- We establish a convergence guarantee for MR.AsyncFL and show that it achieves an $\mathcal{O}(T^{-1/4})$ rate under certain assumptions.
- We conduct extensive experiments on benchmark datasets under both IID and non-IID settings, with and without staleness-threshold mechanisms. The results demonstrate that MR.AsyncFL consistently achieves faster convergence and higher accuracy compared to representative asynchronous FL baselines, validating both the effectiveness and practical advantages of the proposed framework.

2 Related Work

2.1 Staleness-Aware AFL via Weight Aggregation

In Xie et al. (2019), the authors proposed an asynchronous optimization scheme in which the server performs an immediate aggregation upon the arrival of an update from a client. To mitigate staleness, their method applied a staleness-aware hyperparameter during aggregation, blending the previous global model with the new local update using a simple weighting mechanism. However, its treatment of staleness remains coarse,

since stale updates are only softened through weighting rather than being explicitly corrected or removed. Subsequent works either generalized or refined this idea. TWAFL (Chen et al., 2019) applied time-decay functions with exponential decay to the aggregation update rule, aiming to deal with the staleness issue. WKAFL (Zhou et al., 2022b) went further with a two-stage K-async design that selects gradients with consistent descent directions, clips large-norm gradients, and adjusts the learning rate according to the least staleness. In WKAFL, the two-stage K-async design, gradient clipping, and the staleness-aware learning-rate adjustment are further combined with exponential staleness weighting and momentum to improve robustness on non-IID data. The drawback, however, is the added algorithmic complexity and tuning burden, which may reduce practicality in large-scale asynchronous deployments. FedACA (Zhou et al., 2022a) adopts an "informative update" policy in which a client sends in updates only when its model has changed significantly and the server weights update by a time-related factor. This can reduce unnecessary communication and prioritize more meaningful updates, but its effectiveness depends on the quality of the triggering criterion and it still relies on weighting-based handling of delayed information. TimelyFL (Zhang et al., 2023b) addresses a different but related issue by improving fairness through wall-clock time control and adaptive workload assignment, enabling slower devices to participate more regularly. While this improves inclusiveness, its focus is primarily on participation fairness rather than directly resolving the fundamental effect of stale updates once they are incorporated into the server model. Overall, these methods improve asynchronous FL from different perspectives, including staleness control, communication efficiency, and fairness. Nevertheless, they largely remain within a weighted-aggregation framework, where stale information is moderated rather than explicitly replaced or removed.

2.2 Staleness-Aware AFL via Client Selection

A growing body of research addresses the staleness issue by intelligently controlling which clients are selected to participate in each round (Xu et al., 2023). In other words, new client selection and scheduling strategies are developed to limit staleness at its source while improving overall training efficiency. Hao et al. (2020) proposed a semi-asynchronous protocol in which a selected subset of clients can upload their updates as soon as they finish local training. To mitigate straggler effects, the server maintains a priority-based selection policy that favors clients with larger datasets and higher computing power, while a GAN-based data expansion module is used to balance data volume between clients. This approach accelerates convergence by smoothing out computational and statistical heterogeneity. Zhou et al. (2021) introduced TEA-Fed, a hybrid AFL protocol that allows idle clients to proactively apply for training, with a server-side C-fraction control to limit participation. To handle staleness, TEA-Fed incorporates a staleness-aware weighted aggregation scheme and caches multiple versions of the global model for alignment. This design improves update timeliness and leverages edge availability without centralized scheduling overhead. Zhu et al. (2023a) formulates client selection in AFL as a multi-armed bandit (MAB) problem with fairness constraints. Their FLACOS algorithm combines an upper-confidence-bound (UCB) policy with a virtual queue mechanism to ensure that frequently overlooked clients are eventually selected, thereby balancing training speed and long-term fairness. FLACOS guarantees sub-linear regret while reducing training latency significantly. Chen et al. (2021) focused on heterogeneous IoT environments, where client capabilities and link conditions vary widely. They propose a lightweight node-selection strategy that considers each client's real-time availability and resource conditions. By deferring low-quality updates and prioritizing reliable contributors, their approach improves convergence without requiring synchronization or complex correction mechanisms. Finally, Lee & Lee (2021) addressed AFL in wireless networks by formulating it as an asynchronous learning-aware scheduling (ALS) problem. To capture the trade-off between timeliness and staleness, the authors introduced an effectivity score that penalizes outdated updates while rewarding fresh contributions. Using this score, they designed dynamic scheduling policies using reinforcement learning and dynamic programming. Their approach effectively reduces communication overhead and improves training robustness, particularly in bandwidth-constrained environments. Overall, these methods show that client selection and scheduling can reduce staleness indirectly by controlling which updates are generated and transmitted. Their main strength is that they address staleness together with broader system concerns such as fairness, resource heterogeneity, and communication efficiency. However, most of them do not directly correct the effect of stale updates once those updates have already entered the aggregation process. As a result, they are best viewed as complementary to, rather than replacements for, server-side staleness-handling mechanisms.

3 MR.AsyncFL: Asynchronous Federated learning with Model Replacement

3.1 Problem Setup

We consider an AFL system consisting of a central server and N clients, indexed by $i \in [N] := \{1, \ldots, N\}$. Each client i possesses a private local dataset of size n_i sampled from its local data distribution $\mathcal{D}_i$. The local data is never shared outside the device. The central server maintains a global model $w \in \mathbb{R}^d$.

The objective is to collaboratively learn a global model that minimizes the weighted loss across all clients:

$$\min_{w \in \mathbb{R}^d} F(w) := \sum_{i=1}^{N} r_i F_i(w), \tag{1}$$

where $r_i = n_i / \sum_{j=1}^{N} n_j$ denotes the relative data size of client i, $F_i(w) = \mathbb{E}_{\xi \sim \mathcal{D}_i}[f(w; \xi)]$ is the expected loss over client i's local data distribution, and $f(w, \xi)$ is the loss function for model w on a sample data ξ.

3.2 System Overview and Communication Protocol

Our algorithm follows the standard AFL protocol, similar to FedAsync (Xie et al., 2019), where clients communicate with the server independently upon completing their local updates. The protocol proceeds as follows:

1. **System Initiation**: The server broadcasts the initial global model $w_0^{(g)}$ to a list of selected clients, who in turn start a round of training using their local data and this global model.

2. **Client-Side (Local) Update:** After client i performed local training on its private dataset using stochastic gradient descent[1], the updated local model will be uploaded to the server, which triggers server's t^{th} (a global event index[2]) round of aggregation. It is important to note that this local training process may take unknown amount of time, during which the global model may have been updated after one or more rounds of aggregation triggered by other client's upload(s). A local model on client i after training can be expressed as:

$$w_t^{(i)} = w_{t-\delta_t-1}^{(g)} - \eta g_t, \tag{2}$$

   where $\eta > 0$ is the step-size and g_t is a local gradient, and critically, $\delta_t \geq 0$ denotes the staleness due to various delays in the system, such as communication overhead. In other words, the updated model may be trained from an earlier round $t - \delta_t - 1$ instead of the last round $t - 1$.

3. **Model Exchange:** Upon receiving client i's model $w_t^{(i)}$, the server enters round t, which triggers an immediate broadcast of its latest aggregated global model $w_{t-1}^{(g)}$ to this client i before executing the stale model replacement and aggregation step for this event. We adopt this pre-aggregation (Liu et al., 2024) design to speed up the next local training cycle. A natural alternative is a post-aggregation design (Dai et al., 2023), in which the client instead receives the newly updated global model, yielding a fresher initialization. In the situation that server-side aggregation latency is negligible, the practical benefit of pre-aggregation in reducing waiting time becomes modest. As this design choice is orthogonal to our main focus on the aggregation rule design, and it does not affect the correctness analysis of the proposed solution, we leave the experimental comparison of the two variants to the future work.

4. **Server-Side (Global) Aggregation:** After sending the global model to client i, the server will integrate $w_t^{(i)}$ into the global model using our proposed update rule that is discussed in the next subsection.

[1] We use one gradient step for a clearer exposition of our convergence result. Our proof can incorporate multiple steps easily.
[2] We want to emphasize that this index increases each time the server receives an updated model from a client.

For theoretical analysis and practical application purposes, we assume that the staleness is bounded as $\delta_t \leq \tau$ for some constant $\tau > 0$. If a client has not uploaded to the server for more than τ rounds, the server will reset its training by sending the most recent available global model to it to conduct its local training from that point on. This kind of bounded-staleness mechanism is commonly adopted in AFL to limit the adverse effect of severely outdated models on convergence and model quality (Forootani & Iervolino, 2025; Nguyen et al., 2022; Xu et al., 2024).

The proposed global update rule is the critical component of our proposed solution. For comparison purposes, we list the server aggregation rules[3] proposed in FedAsync (Xie et al., 2019) below:

$$\text{FedAsync:} \quad w_t^{(g)} = \gamma w_{t-1}^{(g)} + (1-\gamma) w_t^{(i)}, \tag{3}$$

where $\gamma \in (0, 1)$ is the mixing coefficient or discount factor. γ controls how fast the global model is moved towards the updated local model. Our update bares some similarity to (3), but has a critical difference that we discuss in the next subsection.

3.3 Client Contribution and Global Update

The protocol described in Section 3.2 follows the standard asynchronous FL pipeline. It highlights a key challenge: the uploaded local model $w_t^{(i)}$ is generally stale, since it was trained from an earlier global model $w_{t-\delta_t-1}^{(g)}$ rather than the most recently aggregated global model $w_{t-1}^{(g)}$. In conventional asynchronous aggregation, such stale updates are typically incorporated into the latest global model through staleness-aware weighting or decay, thereby weakening, but not removing the influence of outdated information. This motivates our design with a different perspective: when client i communicates again, its newly uploaded local model can be viewed as a fresher replacement for its previous stale contribution on the server. Based on this intuition, MR.AsyncFL does not merely attenuate the effect of stale information; instead, it directly replaces the stale client-side contribution with the newly arrived one, so that outdated information is removed rather than continuously carried forward in the aggregation process.

Client Contributions to the Global Model To achieve the aforementioned goal, we need to investigate the composition of global model. At any round $t > 0$, when client i uploads an updated model $w_t^{(i)}$, the server-side's global model can be viewed as a cached mixture:

$$w_{t-1}^{(g)} = \sum_{j=1}^{N} c_{t-1}^{(j)} w_{t-1}^{(j)}, \tag{4}$$

where $c_{t-1}^{(j)} \geq 0$ denotes the coefficient assigned to client j in the global model, and $w_{t-1}^{(j)}$ denotes the most recent cached local model from client j on the server. The client weight $c_{t-1}^{(j)}$ indicates how much the client contribute to the global model and thus satisfies $\sum_{j=1}^{N} c_{t-1}^{(j)} = 1$ throughout the training. For the communicating client i, the cached stale model corresponds to $w_{t-\delta_t}^{(i)}$, i.e., $w_{t-1}^{(i)} = w_{t-2}^{(i)} = \cdots = w_{t-\delta_t}^{(i)}$ since there was no update from client i during this period.

If we want to remove client i's old contribution, we need to replace it with the newly uploaded model, yielding

$$\widetilde{w}_{t-1}^{(g)} := w_{t-1}^{(g)} - c_{t-1}^{(i)} w_{t-\delta_t}^{(i)} + c_{t-1}^{(i)} w_t^{(i)}. \tag{5}$$

Based on this, our update to the global model is then

$$\text{MR.AsyncFL:} \quad w_t^{(g)} = \gamma \widetilde{w}_{t-1}^{(g)} + (1-\gamma) w_t^{(i)} \tag{6}$$

$$= \gamma w_{t-1}^{(g)} + (1-\gamma) w_t^{(i)} + \gamma c_{t-1}^{(i)} \left(w_t^{(i)} - w_{t-\delta_t}^{(i)} \right), \tag{7}$$

$$= \sum_{j \neq i} \gamma c_{t-1}^{(j)} w_{t_j}^{(j)} + \left(\gamma c_{t-1}^{(i)} + 1 - \gamma \right) w_t^{(i)}. \tag{8}$$

[3]In the original paper, FedAsync is more generally written as $w_t^{(g)} = (1-\alpha_t) w_{t-1}^{(g)} + \alpha_t w_t^{(i)}$, where $\alpha_t = \alpha(\delta_t + 1)^{-a}$ is staleness-aware. Here, α and a are tunable hyperparameters. Eq. (3) uses the equivalent notation $\gamma_t = 1 - \alpha_t$, and corresponds to the constant-mixing case when γ_t is treated as fixed over time.

Compared to FedAsync (Xie et al., 2019) (i.e., Eq. (3)), our update completely cleanses the old contribution of the participating client and replaced it with the most recent model. To see it more clearly, by expanding Eq. (3), we can see that FedAsync only decays the old contribution rather than removing it

$$\text{FedAsync:} \quad w_t^{(g)} = \sum_j \gamma c_{t-1}^{(i)} w_{t-1}^{(j)} + (1-\gamma) w_t^{(i)} \tag{9}$$

$$= \sum_{j \neq i} \gamma c_{t-1}^{(i)} w_{t-1}^{(j)} + \gamma c_{t-1}^{(i)} w_{t-\delta_t}^{(i)} + (1-\gamma) w_t^{(i)} \tag{10}$$

since $w_{t-\delta_t}^{(i)}$ is outdated and the old contribution $\gamma c_{t-1}^{(i)} w_{t-\delta_t}^{(i)}$ remains after the update. In contrast, our update has an additional term based on model difference as shown in Eq. (7), which removes the contribution of the outdated model completely. To do so, we would need to keep the cached old models, which incurs a memory cost. In practice, various techniques can be used to lower the memory cost on the server side, such as using compressed (e.g., with low-precision) models (Li et al., 2023; Shah & Lau, 2023; Zhu et al., 2023b) and grouping clients into representative clusters (Briggs et al., 2020; Huang et al., 2023). We leave this for future engineering endeavors.

Recursive Weight Update From Eq. (8), the clients coefficient are updated after incorporating $w_t^{(i)}$. The updated global model can still be written as a weighted combination of the most recently cached client models, namely,

$$w_t^{(g)} = \sum_{j=1}^{N} c_t^{(j)} w_t^{(j)}, \tag{11}$$

where $c_t^{(j)}$ are the new weights and other clients' cached models remain unchanged $w_t^{(j)} = w_{t-1}^{(j)}, \forall j \neq i$. By comparing (8) with (11), we can see that the recursive update of the client weights is obtained as

$$c_t^{(j)} = \begin{cases} \gamma c_{t-1}^{(j)}, & j \neq i, \\ \gamma c_{t-1}^{(i)} + (1-\gamma), & j = i. \end{cases} \tag{12}$$

The initial values are set to be $c_0^{(j)} \equiv 1/N$. Other initial values can still work and this choice is mainly for the ease of theoretical analysis. Eq. (12) has a simple interpretation. For the non-participating clients $j \neq i$, the importance will be discounted by γ as they are now one step older or staler. The participating client i will still be first discounted by γ, but also receive a bonus $(1-\gamma)$ due to its recency. Moreover, the normalization of the weights is preserved over time: Given $\sum_{j=1}^{N} c_{t-1}^{(j)} = 1$, we have

$$\sum_{j=1}^{N} c_t^{(j)} = \sum_{j \neq i} \gamma c_{t-1}^{(j)} + \gamma c_{t-1}^{(i)} + (1-\gamma) = \gamma \sum_{j=1}^{N} c_{t-1}^{(j)} + (1-\gamma) = 1. \tag{13}$$

This property is important because it guarantees that the server model remains a convex combination of the cached client models. Consequently, the aggregation rule does not introduce unintended scaling effects into the global iterate, and the update remains consistent with the normalized weighted averaging commonly used in federated optimization (Tedeschini et al., 2025).

In summary, after each communication event, the server replaces the stale cached model of the communicating client with its newly uploaded local model. This representation makes the contribution of each client explicit and provides a convenient basis for deriving the recursive weight update.

3.4 Algorithm Summary

Figure 1 illustrates the general procedure of MR.AsyncFL with three devices. Whenever a client finishes local training, it uploads its trained local model to the server and downloads the latest global model. At each aggregation round, the server performs the aggregation step (8) using three elements: (a) the current global model at hand, (b) the newly received local model from the client that just finished training, and (c)

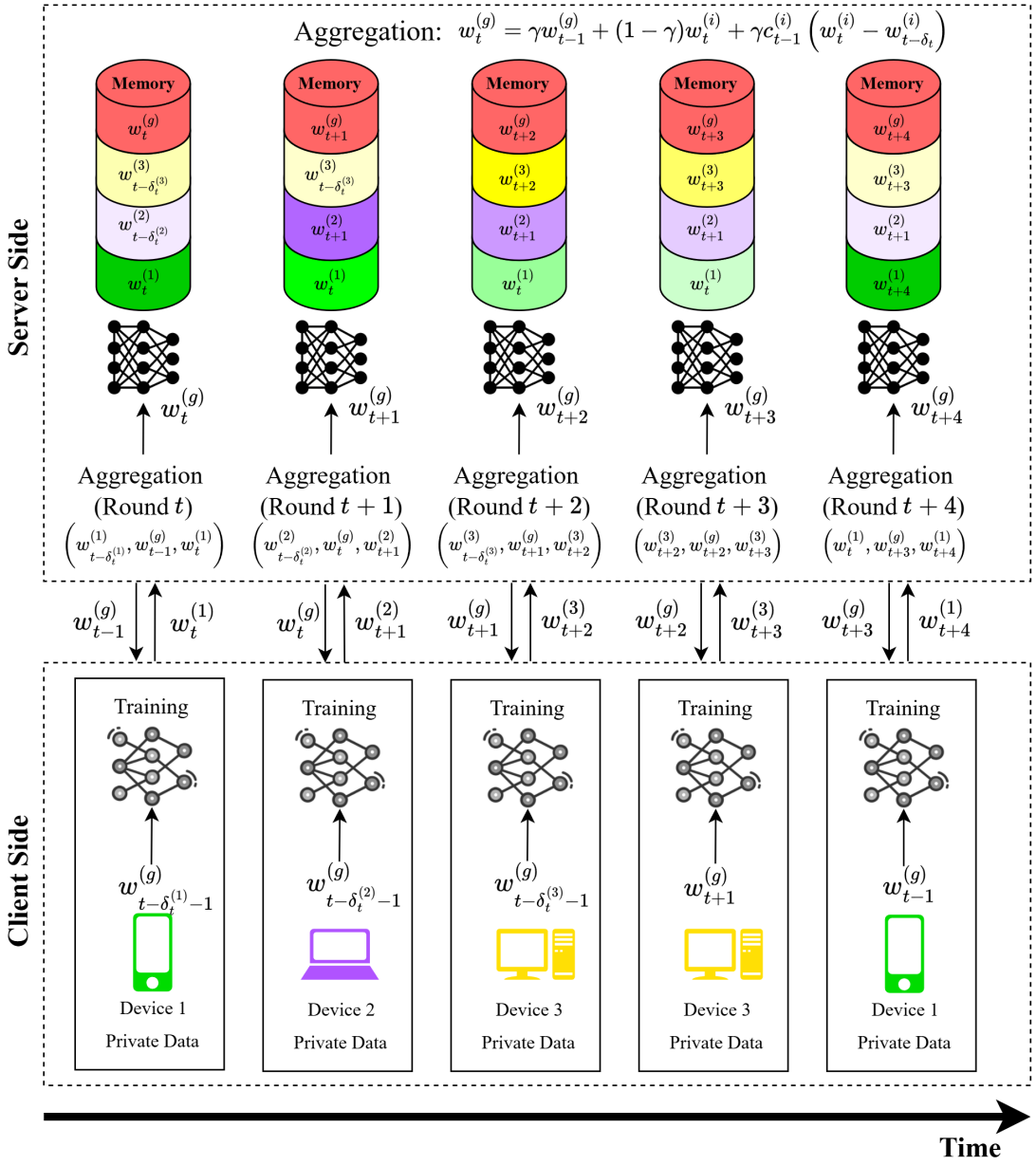

Figure 1: This figure demonstrates a toy running example of three clients in a timeline for MR.AsyncFL. The intensity of the color for each block represents the recency of the local model stored in server memory for a particular client: freshly updated models are shown with more vivid colors, while fading colors indicate increasing staleness as local training proceeds without new uploads. At the very top, the global model memory is shown, which is overwritten after every aggregation event. The figure further illustrates the MR.AsyncFL update procedure: whenever a client completes local training, it uploads its updated model to the server. The server aggregates using (a) the current global model, (b) the newly received local model, and (c) the most recently saved local model from the same client. Notably, Device 3 and Device 1 rejoin in subsequent rounds, and their previous local models are directly retrieved from the server's memory without needing to request this information from the clients. This mechanism is crucial for realizing the model-replacement in our proposed update equation, enabling precise staleness compensation in asynchronous aggregation.

the most recently saved local model from the same client. In this example, Device 3 (yellow) finishes its

training quickly from $t+2$ to $t+3$, thus participates consecutively. Notice that its previous local model $w_{t+2}^{(3)}$ is used by the server for the aggregation step at time $t+3$. A similar situation occurs at time $t+4$, where Device 1 (green) uploads $w_{t+4}^{(1)}$, and the server uses Device 1's previously saved local model, $w_t^{(1)}$, for the current round's update. This highlights the necessity for the server to maintain and reuse the latest local models received from each device, as these may be required for future aggregation steps depending on the order and timing of client participation.

Algorithm 1 MR.AsyncFL: Asynchronous Server and Client Procedures

Require: Mixing parameter $\gamma \in (0, 1)$, N clients

Client Side Procedure:
For each client i, run *asynchronously*:
while Client i is available **do**
  At time $t - \delta_t$, download latest global model $w_{t-\delta_t-1}^{(g)}$ from server
  Perform local training with Eq. (2) (possibly over many server events) to obtain $w_t^{(i)}$
  At time t, upload $w_t^{(i)}$ to server
end while
Server Side Procedure:
Initialize $w_0^{(g)}$ and set $c_0^{(j)} \leftarrow 1/N, \forall j$
while server is running **do**
  Upon receiving upload $w_t^{(i)}$ from client i
  Retrieve $w_{t-\delta_t}^{(i)}$ # Previously stored local model for client i
  Update global model $w_t^{(g)}$ using Eq. (8)
  Update weights $c_t^{(i)}$ and $c_t^{(j)}$ for $j \neq i$ using Eq. (12)
  Store $w_t^{(i)}$ as latest model from client i
  $t \leftarrow t + 1$
end while

Algorithm 1 formalizes the asynchronous operation of MR.AsyncFL, where both clients and the server proceed *independently*. Rather than relying on fixed rounds or global synchronization, clients update at their own pace and the server integrates incoming updates immediately upon arrival. To manage staleness, the server tracks the last received model from each client and computes a model difference relative to the previous upload. This design enables stale-aware aggregation without maintaining large control buffers or requiring optimizer-specific mechanisms. The recursive weighting scheme naturally balances client contributions over time and ensures that the protocol remains lightweight and easy to integrate into existing AFL systems.

4 Convergence Analysis

In this section, we establish the convergence properties of our algorithm. All proofs are deferred in the appendix. We start with the assumptions.

Assumption 1 (Objective Properties)**.** *The global objective $F : \mathbb{R}^d \to \mathbb{R}$ is bounded from below (i.e., $F(w) \geq F^*, \forall w$) and L-smooth: $\|\nabla F(u) - \nabla F(v)\| \leq L\|u - v\|$ $\forall u, v$.*

Assumption 2 (Stochastic gradient)**.** *With filtration $\{\mathcal{F}_t\}_{t\geq 0}$, the realized client update satisfies*

$$g_t = \nabla F_i\big(w_{t-\delta_t-1}^{(g)}\big) + \xi, \qquad \mathbb{E}[\xi \mid \mathcal{F}_{t-1}] = 0. \tag{14}$$

Assumption 3 (Bounded Staleness)**.** *The staleness is bounded by $\delta_t \leq \tau$ where $\tau > 0$ is a finite positive integer.*

These are standard assumptions in AFL (Forootani & Iervolino, 2025; Nguyen et al., 2022; Xu et al., 2024). Assumption 3 indicates that a client should respond to the server in at most τ rounds. Otherwise, it will be reset and start training using a more recent global model as mentioned in the previous section.

Assumption 4 (Participation Probability)**.** *The participation probability of client i is r_i. The maximum of these probabilities is bounded by $r \in [1/N, 1)$. Specifically, using $\mathbf{1}_t^i \in \{0, 1\}$ to indicate whether client i participates in round t, we have $r_i = \mathbb{E}[\mathbf{1}_t^i] \leq r, \forall i, t$.*

This assumption helps us analyze the upper bound of the client weights as seen later in Lemma 1, which is critical in deriving the final convergence rate.

Assumption 5 (Bounded Local Update)**.** *For any client and t, the local gradient update is bounded*

$$\|g_t\| \leq G. \tag{15}$$

This assumption is not strictly necessary and can be replaced by a different one on the variance of the gradient (Nguyen et al., 2022; Xu et al., 2024). However, it simplifies the analysis and makes the final bound less cluttered.

Assumption 6. *The discount factor γ and participation bound r satisfy $\rho := 4\tau^2[(1-\gamma)^2 + \gamma^2 r] < 1$.*

This final assumption ensures a contraction property in the proof. A similar assumption can be found in FedAsync (Xie et al., 2019).

Our proof heavily rely on a decomposition of the global model update:

$$\begin{aligned} \Delta_t &:= w_t^{(g)} - w_{t-1}^{(g)} && (16)\\ &= (1-\gamma)\left(w_{t-\delta_t-1}^{(g)} - w_{t-1}^{(g)}\right) + \gamma c_{t-1}^{(i)}\left(w_{t-\delta_t-1}^{(g)} - w_{t-\delta_t}^{(i)}\right) - \theta_t^{(i)}\eta g_t. && (17)\end{aligned}$$

where $\theta_t^{(i)} := (1-\gamma) + \gamma c_{t-1}^{(i)}$. The proof starts with the L-smoothness of the objective function

$$\mathbb{E}\left[F\left(w_t^{(g)}\right)\right] \leq F\left(w_{t-1}^{(g)}\right) + \underbrace{\mathbb{E}\left[\left\langle \nabla F\left(w_{t-1}^{(g)}\right), \Delta_t\right\rangle \;\middle|\; \mathcal{F}_{t-1}\right]}_{\text{Term A}} + \underbrace{\frac{L}{2}\mathbb{E}\left[\|\Delta_t\|^2 \mid \mathcal{F}_{t-1}\right]}_{\text{Term B}}. \tag{18}$$

We will bound Term A and Term B, and apply telescoping sum to get the final result. When it is clear from the context, we will ignore the condition on the filtration $\mathcal{F}_{t-1}$ and explicitly mention it when necessary. We start with several supporting lemmas.

Since the client weight are updated by Eq. (12), it would be non-trivial to see if some client can have a dominating contribution/weight that is close to one. The following lemma answers this question and ensures that the weight is bounded by r.

Lemma 1 (Bound for Client Weight)**.** *Under Assumption 4, the expected client weight is bounded by $\mathbb{E}\left[c_t^{(i)}\right] \leq r$. Moreover, $\mathbb{E}\left[(c_t^{(i)})^2\right] \leq r$:*

With this lemma, we can now derive upper bounds for (the average of) Term A and Term B respectively with the following lemmas:

Lemma 2 (Bound for Term B of (18))**.** *Under Assumptions 3 to 6, we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\Delta_t\|^2] \leq C_\Delta \eta^2, \tag{19}$$

where $C_\Delta := \frac{4G^2\left((1-\gamma)^2 + 2\gamma r\right)}{1-\rho} > 0$.

Lemma 3 (Bound for Term A of (18))**.** *Under Assumptions 1 to 6, we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\langle \nabla F\left(w_{t-1}^{(g)}\right), \Delta_t\right\rangle\right] \leq \left[\frac{1-\gamma+2\gamma r}{\eta} + \eta L^2\right]\frac{C_w}{T} + \eta\gamma r G^2 - \frac{\eta(1-\gamma)}{4T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right], \tag{20}$$

where $C_w = \tau^2 C_\Delta$.

With Lemmas 2 and 3, we are ready to show the main convergence result.

Theorem 1 (Convergence)**.** *Under Assumptions 1 to 6, we have*

$$\min_{t=1,\dots,T} \mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right] \leq \frac{4}{1-\gamma}\left(\frac{F\left(w_0^{(g)}\right) - F_*}{\eta T} + C_w\left[\frac{1-\gamma+2\gamma r}{2} + \eta^2 L^2\right] + \gamma r G^2 + LC_\Delta \eta\right). \quad (21)$$

Furthermore, by choosing

$$\eta = \frac{1}{\sqrt{T}}, \qquad r = \frac{2}{N}, \qquad \gamma = 1 - \frac{1}{T^{1/4}}, \qquad T = N^2,$$

we obtain a convergence rate of $\mathcal{O}\big(T^{-1/4}\big)$.

This theorem shows that, with a reasonable participation ($r = 2/N$) and sufficient training ($T = N^2$), the minimum expected gradient on the iterates decays at a rate of $\mathcal{O}(T^{-1/4})$.

Comparison to FedAsync. Our convergence analysis shares an important similarity with FedAsync (Xie et al., 2019) in that both proofs initially contain a non-vanishing residual term caused by asynchrony and staleness. In our case, this error floor arises through the participation-related term r, which appears in the bound of our Theorem 1; however, under the large-scale cross-device FL regime, this floor can be driven to zero by choosing $r = 2/N$ and allowing the effective client population N to grow with the training horizon T, e.g., through the scaling $T = N^2$. This is particularly natural in open cross-device federated systems, where the client pool can be very large and new clients may join over time. By contrast, the FedAsync analysis removes its residual term by requiring the minimum number of local updates per client, $H_{\min}$, to grow with T at a rate of $H_{\min} = T^{1/5}$. The bound in their Theorem 1 contains residual terms proportional to $1/H_{\min}$, so asymptotic vanishing relies on increasingly large local computation on each participating device. While mathematically valid, such a condition is less well aligned with practical cross-device FL, where clients are typically resource-constrained and only available for short, intermittent local training sessions. From this perspective, our scaling assumption is better matched to the intended deployment regime of large, dynamic cross-device federated learning, whereas the FedAsync condition places the asymptotic burden on ever-longer local training.

5 Experiments

5.1 Experimental Setup

We conduct experiments on the CIFAR-10 and CIFAR-100 datasets (Krizhevsky et al., 2009), both widely adopted benchmarks in image classification and FL research. CIFAR-10 consists of 60,000 color images of size 32×32 pixels, evenly distributed across 10 classes, with 50,000 images for training and 10,000 for testing. CIFAR-100 shares the same image size and total number of samples, but spans 100 classes with 600 images per class (500 for training and 100 for testing), making it a substantially more challenging benchmark due to its larger label space and higher inter-class similarity.

All experiments are carried out in a FL setting with $N = 30$ clients. We include both IID and non-IID settings because they reflect two complementary regimes for evaluating asynchronous federated learning. The IID case serves as a controlled setting in which client distributions are well aligned, making it easier to isolate the effect of asynchronous aggregation and staleness. The non-IID case, on the other hand, better reflects practical FL deployments and is substantially more challenging, since stale updates may also be biased toward client-specific data distributions. For the IID scenario, the training data is randomly shuffled and equally partitioned among all clients, ensuring each client receives a balanced mixture of all classes. For the non-IID scenario, we employ a Dirichlet distribution-based partitioning strategy (Hsu et al., 2019), which allocates each client a proportion of data from each class drawn from a Dirichlet distribution with concentration parameter set to 0.5. This value is commonly adopted in FL literature to create a moderate degree of statistical heterogeneity across clients.

We further consider experiments both without a maximum staleness bound and with a maximum staleness bound. The former serves as a harsher empirical stress-test, in which algorithms must cope with arbitrary highly stale updates and the effectiveness of their aggregation rules becomes especially important. The latter reflects the bounded-staleness regime commonly adopted in practical AFL systems, where excessively outdated updates are discarded and clients are resynchronized with a recent global model. Studying both cases allows us to evaluate MR.AsyncFL both in a theory-aligned setting and in a more challenging unconstrained setting, thereby illustrating not only its behavior under bounded staleness, which is assumed in our analysis, but also its empirical robustness when such constraint is removed.

5.2 Model Setup

We use PyTorch to implement the local models for the clients. The local and global model architecture closely follows the design specified in Zhang et al. (2023a) and is based on ResNet18, a widely adopted convolutional neural network for image classification tasks in FL research. ResNet18 is chosen for its favorable trade-off between representational power and computational efficiency, with skip connections that enable stable training on both IID and non-IID data. The architecture consists of an initial convolutional layer with 64 channels, followed by batch normalization and ReLU activation, and four sequential residual layers with channel sizes of 64, 128, 256, and 512. Each residual block includes either an identity shortcut or a projection shortcut when the input and output dimensions differ. The final layers perform global average pooling and map the output classes using a fully connected layer. All experiments were conducted on multiple virtual machines, each configured with high-performance GPUs, including Nvidia Titan V, RTX 3060, and RTX 4090 models, all operating under Ubuntu 24.04.

5.3 Baselines

We compare the proposed method against three representative AFL baselines:

FedAsync (Xie et al., 2019) performs an immediate update of the global model upon receiving a local model from any client, combining the previous global model and the new local model using a fixed or staleness-adjusted step size.

TWAFL (Chen et al., 2019) employs a rolling average approach: at each global update, it takes the weighted average of every client's latest local model, rather than only combining the previous global with the incoming update. The aggregation weights are computed based on the staleness of each local model. This method generalizes rolling FedAvg by explicitly accounting for client staleness in the weighting scheme.

Rolling FedAvg is the staleness-agnostic version of TWAFL, aggregating the latest model from every client using uniform (or sample-size based) weights, but without any adjustment for staleness. Rolling FedAvg can be seen as the base of TWAFL, where all clients contribute equally regardless of the delay or frequency of their updates.

5.4 Results and Analysis

Each algorithm is evaluated with its best-performing hyperparameters found through systematic experimentation. For FedAsync, the update follows a staleness-aware rule in the original paper, which achieves the best overall results in our experiments. Their mixing coefficient is computed as $\gamma_t = 1 - \alpha(\delta_t + 1)^{-a}$ where α, a are hyperparameters. The optimal values determined in our experiments are $\alpha = 0.1$ and $a = 0.5$, and these are applied consistently across all plots. We also experimented with a constant-mixing version of FedAsync, but its performance was generally worse than the staleness-aware setting with $\alpha = 0.1$ and $a = 0.5$; therefore, we report the stronger staleness-aware variant throughout. For TWAFL, we tune the temporal decay factor, and the best performance under IID data without a maximum staleness threshold is obtained with an effective decay coefficient of 0.5. As shown in Figure 2, MR.AsyncFL achieves optimal performance at $\gamma = 0.8$. Among the four values we tested, performance consistently improves as γ increases; this trend is qualitatively aligned with our theoretical analysis, which considers a regime where $\gamma \to 1$ as T increases. Rolling FedAvg, in contrast, does not include any tunable staleness or decay hyperparameters, as it aggregates client updates without regard to their staleness. In Figures 3 through 10, the shaded re-

gions represent the standard deviation over three runs with different random seeds, where the randomness arises primarily from the asynchronous client participation order, which can significantly affect AFL training dynamics.

CIFAR-10 Figure 3 presents the global accuracy curves for all four algorithms under IID data without any maximum staleness threshold. In this setting, TWAFL achieves its best performance with a decay coefficient of 0.5. The results show that MR.AsyncFL attains the highest global accuracy, followed by TWAFL and FedAsync, while Rolling FedAvg lags significantly behind due to its inability to handle staleness.

Figure 4 shows the results for the same algorithms and IID data, but with a maximum staleness threshold set to 20 global timestamps. TWAFL achieves its best results with decay coefficient set to 0.6 in this scenario. The performance ranking remains the same, with MR.AsyncFL maintaining its lead, followed by TWAFL, then FedAsync, and finally Rolling FedAvg. However, the global accuracy values are noticeably higher for all methods, and the training curves are smoother compared to Figure 3. This improvement is due to the prevention of highly outdated client updates from being included in the aggregation process.

Figure 5 considers the non-IID data and a maximum staleness threshold of 20. For TWAFL, the optimal value for its decay coefficient is 0.4 in this setting. Rolling FedAvg continues to operate without any staleness or decay parameters. Under these conditions, all algorithms experience reduced global accuracy and increased fluctuation, reflecting the greater difficulty of learning from heterogeneous data. The gap between TWAFL and FedAsync narrows, indicating that TWAFL is more sensitive to the effects of non-IID data. MR.AsyncFL, while still the top performer, also shows a reduction in accuracy compared to the IID cases. Rolling FedAvg remains the lowest-performing algorithm, which is consistent with its lack of any staleness control.

Figure 6 considers non-IID data with a maximum staleness threshold of 20. For TWAFL, the optimal value for its decay coefficient is 0.4 in this setting. An interesting observation is that TWAFL exhibits a much larger performance drop in the non-IID setting when the maximum staleness threshold is removed, compared with the corresponding IID case shown in Figure 3 and 4. A plausible explanation is rooted in the temporally weighted aggregation mechanism of TWAFL, which down-weights stale updates but does not explicitly discard them. Under non-IID data, stale local models are not only outdated but also more strongly biased toward client-specific distributions. As a result, when no staleness threshold is imposed, the aggregation becomes more sensitive to the coexistence of highly stale updates and recently arrived but distribution-skewed client models, leading to larger fluctuations and lower final accuracy. By contrast, in the IID case, client updates are more statistically aligned, so the absence of a staleness threshold is less detrimental. This interpretation is also consistent with the original TWAFL paper, which notes that temporally weighted aggregation may introduce fluctuations because some high-quality local models can receive insufficient weight in certain communication rounds.

Across all figures, we emphasize that the results reflect careful hyperparameter tuning for each algorithm, particularly for the staleness-related parameters. For FedAsync, we adopt a staleness function $s_a(t-\tau)$ and use the aggregation weight α_t as described above, with $\alpha = 0.1$ and $a = 0.5$ selected based on their empirical performance. For TWAFL, we use decay coefficient of 0.5 for IID data without threshold, 0.6 for IID data with threshold, and 0.4 for the two non-IID data scenarios. Rolling FedAvg does not require such tuning, as it lacks any staleness-awareness by design. The application of a maximum staleness threshold universally improves accuracy and convergence stability for all algorithms, though it does not alter their relative ranking. Meanwhile, non-IID data settings reduce accuracy and increase the sensitivity to hyperparameters, especially for algorithms like TWAFL. The consistently poor performance of Rolling FedAvg across all scenarios further highlights the necessity of staleness-aware aggregation in AFL.

CIFAR-100 To further evaluate the scalability and robustness of MR.AsyncFL, we extend our experiments to the CIFAR-100 dataset, which is considerably more challenging due to its 100 classes and higher inter-class similarity. This setting better reflects real-world FL scenarios with large output spaces and complex distributions. To obtain more stable training behavior in this harder setting, we increase the total number of FL epochs from 1000 to 4000. Using the same four experimental settings as in the CIFAR-10 study, we report results for IID data without a maximum staleness threshold (Figure 7), IID data with a maximum

Table 1: Global accuracy (%) across three seeds for each method (CIFAR-10). Entries are mean ± std. Bold indicates the highest mean accuracy per column.

Method	No Stale Thres. (IID)	Stale Thres. = 20 (IID)	No Stale Thres. (non-IID)	Stale Thres. = 20 (non-IID)
MR.AsyncFL	**80.77 ± 2.33**	**83.17 ± 1.21**	**77.88 ± 1.23**	**78.68 ± 0.98**
FedAsync	75.28 ± 2.29	77.35 ± 2.87	71.13 ± 2.48	72.10 ± 2.72
TWAFL	79.19 ± 2.57	81.33 ± 1.90	73.16 ± 1.25	76.62 ± 1.54
Rolling FedAvg	53.70 ± 2.39	53.70 ± 2.39	48.68 ± 2.09	48.68 ± 2.09

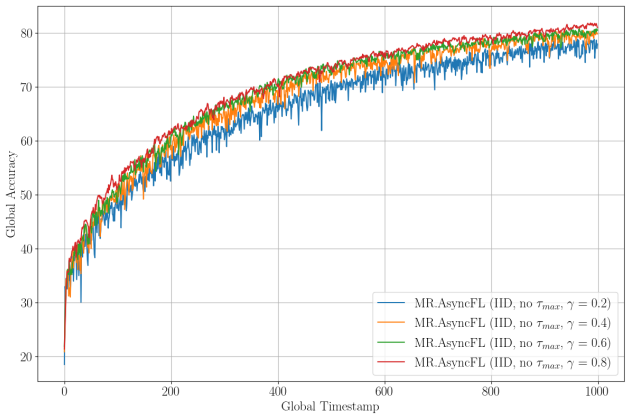

Figure 2: Global Accuracy Comparison of MR.AsyncFL under IID data distribution without maximum staleness threshold for different decay parameters (CIFAR-10).

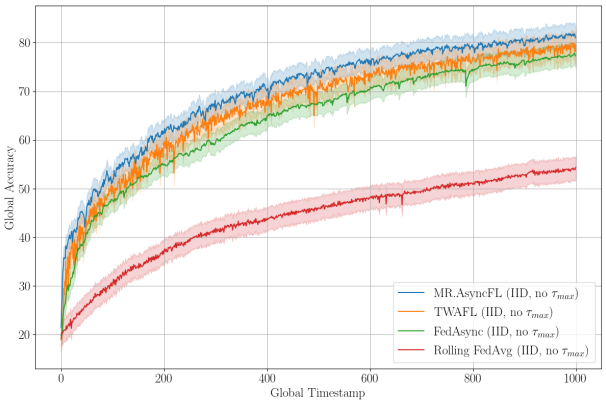

Figure 3: Global Accuracy Comparison under IID data distribution without maximum staleness threshold (CIFAR-10).

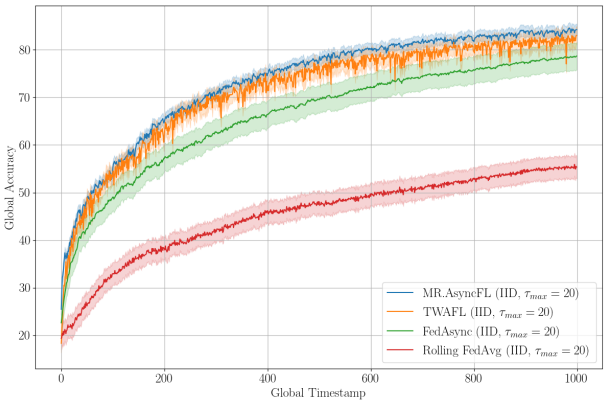

Figure 4: Global Accuracy Comparison under IID data distribution with maximum staleness threshold of 20 global timestamps (CIFAR-10).

staleness threshold of 20 (Figure 8), non-IID data with a maximum staleness threshold of 20 (Figure 9), and non-IID data without a maximum staleness threshold (Figure 10). Overall, the same qualitative trends observed on CIFAR-10 remain consistent: MR.AsyncFL achieves the best performance across all settings, the staleness threshold improves stability and final accuracy for all methods, and the non-IID setting remains more challenging than the IID setting. As expected, the absolute global accuracies are uniformly lower than those on CIFAR-10 because CIFAR-100 is a more difficult classification task.

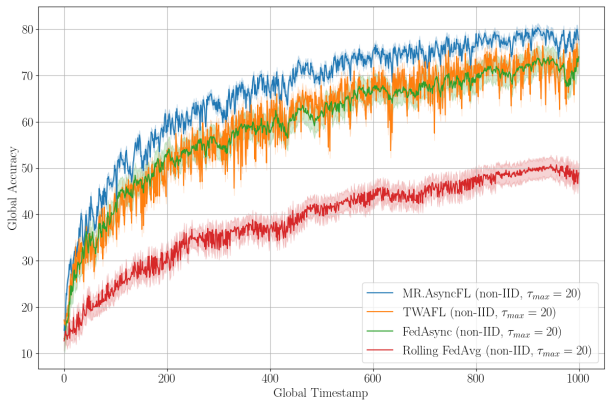

Figure 5: Global Accuracy Comparison under non-IID data distribution with maximum staleness threshold of 20 global timestamps (CIFAR-10).

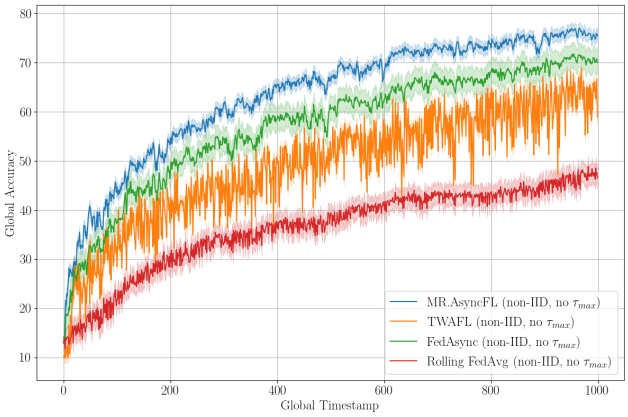

Figure 6: Global Accuracy Comparison under Non-IID Data Distribution without Maximum Staleness Threshold (CIFAR-10)

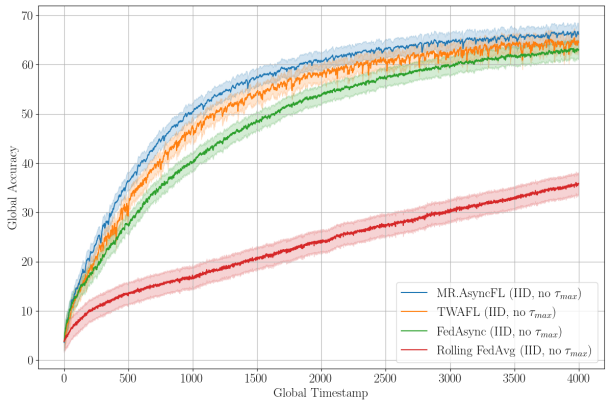

Figure 7: Global Accuracy Comparison under IID data distribution without maximum staleness threshold (CIFAR-100).

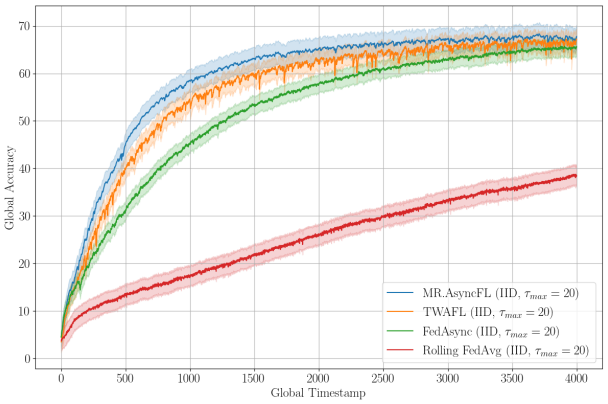

Figure 8: Global Accuracy Comparison under IID data distribution with maximum staleness threshold of 20 global timestamps (CIFAR-100).

In addition to the global accuracy curves, we also report the mean and standard deviation of the final accuracies across three random seeds in Table 1 for CIFAR-10 and Table 2 for CIFAR-100. These tables provide a compact numerical summary of the best global accuracies obtained under each experimental setting.

Table 2: Global accuracy (%) across available seeds for each method (CIFAR-100). Entries are mean ± std. Bold indicates the highest mean accuracy per column.

Method	**No Stale Thres. (IID)**	**Stale Thres. = 20 (IID)**	**No Stale Thres. (non-IID)**	**Stale Thres. = 20 (non-IID)**
MR.AsyncFL	**66.46 ± 1.81**	**68.19 ± 2.42**	**63.96 ± 1.39**	**65.19 ± 1.55**
FedAsync	65.14 ± 1.84	65.88 ± 1.90	62.13 ± 2.02	63.36 ± 1.86
TWAFL	65.33 ± 1.90	67.63 ± 2.20	61.81 ± 1.80	63.71 ± 1.90
Rolling FedAvg	40.22 ± 2.20	40.22 ± 2.15	38.48 ± 2.20	38.48 ± 2.15

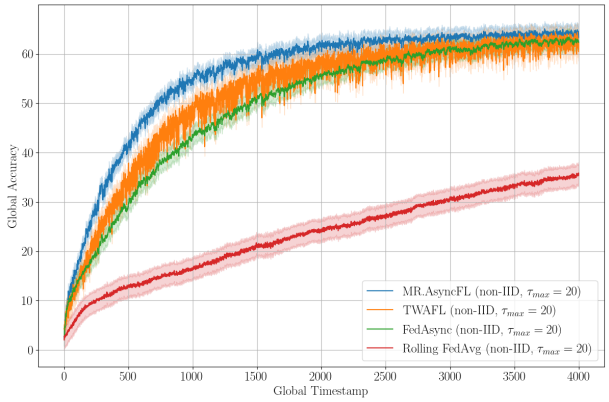

Figure 9: Global Accuracy Comparison under non-IID data distribution with maximum staleness threshold of 20 global timestamps (CIFAR-100).

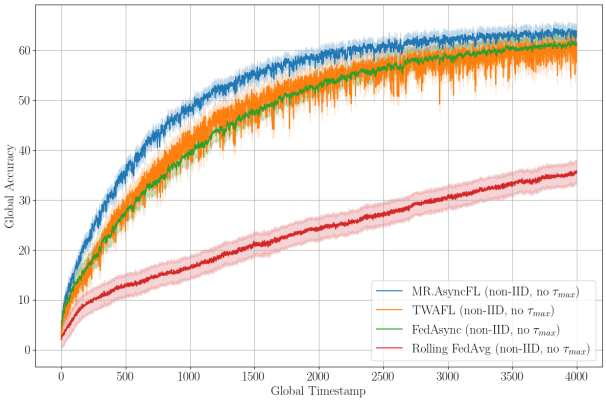

Figure 10: Global Accuracy Comparison under non-IID data distribution without maximum staleness threshold (CIFAR-100).

6 Conclusion

In this paper, we proposed MR.AsyncFL, a fully asynchronous FL framework that addresses stale updates from slow clients through a stale-model replacement mechanism. Unlike conventional AFL methods that only attenuate stale updates during aggregation, MR.AsyncFL replaces client i's previously cached local model in the server-side mixture with the new one, forming an adjusted intermediate model. It then combines this intermediate model with the same newly uploaded local model to produce the next global model. This yields a more faithful representation of the latest client state in the global model and provides an interpretable weighted-mixture view of server aggregation.

Furthermore, to make this update practical in fully asynchronous settings, we introduced a recursive participation-weight scheme that preserves normalization without requiring auxiliary buffers or optimizer-specific modifications. We also provide a convergence analysis showing that, under bounded staleness, MR.AsyncFL achieves an $\mathcal{O}(T^{-1/4})$ convergence rate measured by the minimum expected squared gradient norm over T iterations. Empirically, across CIFAR-10 and CIFAR-100 under IID and non-IID settings, with and without staleness thresholds, MR.AsyncFL consistently achieved the best accuracy among the compared asynchronous baselines, with particularly clear advantages in more challenging heterogeneous settings.

References

C. Briggs, Z. Fan, and P. Andras. Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, UK, 2020.

C. Chen, A. Seff, A. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015.

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*, 31(10):4229–4238, 2019.

Z. Chen, W. Liao, K. Hua, C. Lu, and W. Yu. Towards asynchronous federated learning for heterogeneous edge-powered internet of things. *Digital Communications and Networks*, 7(3):317–326, 2021.

M. Dai, Y. Zhao, J. Yuan, S. Kianoush, S. Savazzi, and B. Li. Federated learning based on asynchronous and adjusted client training. *Physical Communication*, 61:102164, 2023.

A. Forootani and R. Iervolino. Asynchronous federated learning with non-convex client objective functions and heterogeneous dataset. *IEEE Transactions on Artificial Intelligence*, 2025.

J. Hao, Y. Zhao, and J. Zhang. Time efficient federated learning with semi-asynchronous communication. In *Proceedings of IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS)*, Hong Kong, 2020.

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

H. Huang, W. Shi, Y. Feng, C. Niu, G. Cheng, J. Huang, and Z. Liu. Active client selection for clustered federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(11):16424–16438, 2023.

A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

H. Lee and J. Lee. Adaptive transmission scheduling in wireless networks for asynchronous federated learning. *IEEE Journal on Selected Areas in Communications*, 39(12):3673–3687, 2021.

Z. Li, H. Li, and L. Meng. Model compression for deep neural networks: A survey. *MDPI Computers*, 12 (3):60, 2023.

J. Liu, J. Jia, T. Che, C. Huo, J. Ren, Y. Zhou, H. Dai, and D. Dou. Fedasmu: Efficient asynchronous federated learning with dynamic staleness-aware model update. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)*. AAAI Press, 2024.

X. Ma, W. Shi, and J. Wen. An enhanced combinatorial contextual neural bandit approach for client selection in federated learning. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, 10(2):1–23, 2025.

H.B. McMahan, E. Moore, D. Ramage, S. Hampson, and B.A. Arcas. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

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 (AISTATS)*, Valencia, Spain, 2022.

S. Pandya, G. Srivastava, R. Jhaveri, M.R. Babu, S. Bhattacharya, P.K.R. Maddikunta, S. Mastorakis, M.J. Piran, and T.R. Gadekallu. Federated learning for smart cities: A comprehensive survey. *Sustainable Energy Technologies and Assessments*, 55:102987, 2023. ISSN 2213-1388.

S.M. Shah and V.K.N. Lau. Model compression for communication efficient federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9):5937–5951, 2023.

N. Singh and M. Adhikari. Staleness-aware semi-asynchronous federated learning with adaptive learning rates and non-iid data in edge networks. *IEEE Internet of Things Journal*, 13(4):6683–6691, 2026.

B.C. Tedeschini, S. Savazzi, and M. Nicoli. Weighted average consensus algorithms in distributed and federated learning. *IEEE Transactions on Network Science and Engineering*, 12(2):1369–1382, 2025.

Q. Wu, K. He, and X. Chen. Personalized federated learning for intelligent iot applications: A cloud-edge based framework. *IEEE Open Journal of the Computer Society*, 1:35–44, 2020.

C. Xie, S. Koyejo, and I. Gupta. Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934*, 2019.

C. Xu, Y. Qu, Y. Xiang, and L. Gao. Asynchronous federated learning on heterogeneous devices: A survey. *Computer Science Review*, 50:100595, 2023.

H. Xu, Z. Zhang, S. Di, B. Liu, K.A. Alharthi, and J. Cao. Fedfa: A fully asynchronous training paradigm for federated learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, Jeju, Korea, 2024.

F. Zhang, X. Liu, S. Lin, G. Wu, X. Zhou, J. Jiang, and X. Ji. No one idles: Efficient heterogeneous federated learning with parallel edge and server computation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 41399–41413. PMLR, 23–29 Jul 2023a.

T. Zhang, L. Gao, S. Lee, M. Zhang, and S. Avestimehr. Timelyfl: Heterogeneity-aware asynchronous federated learning with adaptive partial training. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Vancouver, BC, Canada, 2023b.

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*, pp. 30–37, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384049.

S. Zhou, Y. Huo, S. Bao, B. Landman, and A. Gokhale. Fedaca: An adaptive communication-efficient asynchronous framework for federated learning. In *Proceedings of IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, CA, USA, 2022a.

Z. Zhou, Y. Li, X. Ren, and S. Yang. Towards efficient and stable k-asynchronous federated learning with unbounded stale gradients on non-iid data. *IEEE Transactions on Parallel and Distributed Systems*, 33 (12):3291–3305, 2022b.

F. Zhu, J. Hao, Z. Chen, Y. Zhao, B. Chen, and X. Tan. Stafl: Staleness-tolerant asynchronous federated learning on non-iid dataset. *Electronics*, 11(3):314, 2022.

H. Zhu, Y. Zhou, H. Qian, Y. Shi, X. Chen, and Y. Yang. Online client selection for asynchronous federated learning with fairness consideration. *IEEE Transactions on Wireless Communications*, 22(4):2493–2506, 2023a.

X. Zhu, J. Wang, W. Chen, and K. Sato. Model compression and privacy preserving framework for federated learning. *Future Generation Computer Systems*, 140:376–389, 2023b.

A Proof of Lemma 1

Lemma 1 (Bound for Client Weight)**.** *Under Assumption 4, the expected client weight is bounded by* $\mathbb{E}\left[c_t^{(i)}\right] \leq r$*. Moreover,* $\mathbb{E}\left[(c_t^{(i)})^2\right] \leq r$*:*

Proof. The update of a client can be expressed as

$$c_t^{(i)} = \gamma c_{t-1}^{(i)} + \mathbf{1}_t^i(1-\gamma). \tag{22}$$

where $\mathbf{1}_t^i$ is an indicator function, indicating whether client i participates in round t. For a client with participation probability $r_i \in [0, r]$, in expectation, it satisfies

$$\mathbb{E}\left[c_t^{(i)} \middle| \mathcal{F}_{t-1}\right] = \gamma c_{t-1}^{(i)} + r_i(1-\gamma). \tag{23}$$

Keep expanding and take full expectation, we have

$$\mathbb{E}\left[c_t^{(i)}\right] = \gamma^t c_0^{(i)} + r_i(1-\gamma)\sum_{s=0}^{t-1}\gamma^s \tag{24}$$

$$= \gamma^t c_0^{(i)} + r_i(1-\gamma)\frac{1-\gamma^t}{1-\gamma} \tag{25}$$

$$= r_i + \gamma^t(c_0^{(i)} - r_i). \tag{26}$$

There are two cases, if $r_i \geq c_0^{(i)} = 1/N$, then $(c_0^{(i)} - r_i)$ is negative and $\mathbb{E}\left[c_t^{(i)}\right] \leq r_i \leq r$. If $r_i < 1/N$, then $\mathbb{E}\left[c_t^{(i)}\right] = (1-\gamma^t)r_i + \gamma^t c_0^{(i)} < 1/N \leq r$ by Assumption 4. Therefore, $\mathbb{E}\left[c_t^{(i)}\right] \leq r$ holds for all clients and all time t.

Next, we derive a bound on the square of the weight $\mathbb{E}[(c_t^{(i)})^2]$. To do so, we first derive its variance and add it to the square of the expectation. Given $c_{t-1}^{(i)}$, $c_t^{(i)}$ is Bernoulli with a gap of $(1-\gamma)$ between the two possible outcomes (see (12)). Thus, its conditional variance is

$$\mathbb{V}[c_t^{(i)}|\mathcal{F}_{t-1}] = r_i(1-r_i)(1-\gamma)^2, \tag{27}$$

which is a constant independent of $c_{t-1}^{(i)}$. Then by the law of total variance,

$$\begin{aligned}
\mathbb{V}[c_t^{(i)}] &= \mathbb{E}[\mathbb{V}[c_t^{(i)}|\mathcal{F}_{t-1}]] + \mathbb{V}[\mathbb{E}[c_t^{(i)}|\mathcal{F}_{t-1}]] && (28)\\
&= r_i(1-r_i)(1-\gamma)^2 + \mathbb{V}[\gamma c_{t-1}^{(i)} + r_i(1-\gamma)] && \text{From (27) and (23)} \quad (29)\\
&= r_i(1-r_i)(1-\gamma)^2 + \gamma^2 \mathbb{V}[c_{t-1}^{(i)}]. && \text{Variance properties} \quad (30)
\end{aligned}$$

Continuing the expansion gives

$$\begin{aligned}
\mathbb{V}[c_t^{(i)}] &= \gamma^{2t}\mathbb{V}[c_0^{(i)}] + r_i(1-r_i)(1-\gamma)^2 \sum_{s=0}^{t-1} \gamma^{2k} && (31)\\
&= r_i(1-r_i)(1-\gamma)^2 \cdot \frac{1-\gamma^{2t}}{1-\gamma^2} && (32)\\
&= \frac{r_i(1-r_i)(1-\gamma)(1-\gamma^{2t})}{1+\gamma} && (33)\\
&\leq \frac{r_i(1-r_i)(1-\gamma)}{1+\gamma} && (34)\\
&\leq r_i(1-r_i). && (35)
\end{aligned}$$

Note from (26) that $\mathbb{E}[c_t^{(i)}]$ is monotonically increasing or decreasing in t depending on whether $r_i < 1/N$ or $r_i \geq 1/N$, so $\max_t \mathbb{E}[c_t^{(i)}] = \max\{1/N, r_i\}$. Then

$$\begin{aligned}
\mathbb{E}[(c_t^{(i)})^2] &= \mathbb{E}[c_t^{(i)}]^2 + \mathbb{V}[c_t^{(i)}] && (36)\\
&\leq \max\{1/N, r_i\}^2 + r_i(1-r_i) && (37)
\end{aligned}$$

There are two cases, if $r_i \geq 1/N$, the bound is

$$\mathbb{E}[(c_t^{(i)})^2] \leq r_i^2 + r_i(1-r_i) = r_i \leq r. \tag{38}$$

If $r_i < 1/N$, the bound is

$$\mathbb{E}[(c_t^{(i)})^2] \leq \frac{1}{N^2} + r_i(1-r_i). \tag{39}$$

The RHS is a quadratic function of r_i, and it is increasing in $[0, 1/2]$ (thus increasing in $[0, 1/N]$ since $N \geq 2$). This means the maximum is achieved at $r_i = 1/N$ and the bound becomes

$$\mathbb{E}[(c_t^{(i)})^2] \leq \frac{1}{N^2} + \frac{1}{N}\left(1 - \frac{1}{N}\right) = \frac{1}{N} \leq r \tag{40}$$

due to Assumption 4. This finishes the proof and $\mathbb{E}[(c_t^{(i)})^2] \leq r$ for all clients and all time t. □

B Proof of Lemma 2

Lemma 2 (Bound for Term B of (18))**.** *Under Assumptions 3 to 6, we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\Delta_t\|^2] \leq C_\Delta \eta^2, \tag{19}$$

where $C_\Delta := \frac{4G^2\left((1-\gamma)^2+2\gamma r\right)}{1-\rho} > 0$.

Proof. From the definition of Δ_t (17) and triangle inequality

$$\|\Delta_t\| \leq (1-\gamma)\|w^{(g)}_{t-\delta_t-1} - w^{(g)}_{t-1}\| + \gamma c^{(i)}_{t-1}\|w^{(g)}_{t-\delta_t-1} - w^{(i)}_{t-\delta_t}\| + \theta^{(i)}_t \eta \|g_t\|. \tag{41}$$

For the second term on the RHS, it measures the deviation between $w^{(g)}_{t-\delta_t-1}$ and $w^{(i)}_{t-\delta_t}$, which actually share the same starting point $w^{(g)}_{t-\delta_t-\delta_{t-\delta_t}-1}$. That is, $w^{(i)}_{t-\delta_t}$ is obtained by local training from the *last* participation (at round $t-\delta_t$ as opposed to the current participation at round t) of client i, taking a gradient step from $w^{(g)}_{t-\delta_t-\delta_{t-\delta_t}-1}$, while $w^{(g)}_{t-\delta_t-1}$ is the global model updated from $w^{(g)}_{t-\delta_t-\delta_{t-\delta_t}-1}$ by incorporating updates from clients *other than* i (when client i is doing local training). Denote $b_t := \delta_t + \delta_{t-\delta_t}$ (i.e., how much we need to trace back; the time pair $(t-\delta_t, t-b_t-1)$ is then analogous to $(t, t-\delta_t-1)$ in this client's last participation). We have

$$\begin{aligned}\|w^{(g)}_{t-\delta_t-1} - w^{(i)}_{t-\delta_t}\| &\leq \|w^{(g)}_{t-\delta_t-1} - w^{(g)}_{t-b_t-1}\| + \|w^{(i_t)}_{t-\delta_t} - w^{(g)}_{t-b_t-1}\|. &(42)\\ &= \|w^{(g)}_{t-\delta_t-1} - w^{(g)}_{t-b_t-1}\| + \eta\|g_{t-\delta_t}\|. &(43)\end{aligned}$$

Then

$$\|\Delta_t\| \leq (1-\gamma)\|w^{(g)}_{t-\delta_t-1} - w^{(g)}_{t-1}\| + \gamma c^{(i)}_{t-1}\|w^{(g)}_{t-\delta_t-1} - w^{(g)}_{t-b_t-1}\| + \gamma c^{(i)}_{t-1}\eta\|g_{t-\delta_t}\| + \theta^{(i)}_t\eta\|g_t\|. \tag{44}$$

Applying $(a+b+c+d)^2 \leq 4(a^2+b^2+c^2+d^2)$, Assumption 5 and the triangle inequality gives

$$\|\Delta_t\|^2 \leq 4(1-\gamma)^2\|w^{(g)}_{t-\delta_t-1} - w^{(g)}_{t-1}\|^2 + 4(\gamma c^{(i)}_{t-1})^2\|w^{(g)}_{t-\delta_t-1} - w^{(g)}_{t-b_t-1}\|^2 + 4(\gamma c^{(i)}_{t-1}\eta)^2G^2 + 4(\theta^{(i)}_t\eta)^2G^2 \tag{45}$$

$$\leq 4(1-\gamma)^2\left(\sum_{s=1}^{\delta_t}\|\Delta_{t-s}\|\right)^2 + 4(\gamma c^{(i)}_{t-1})^2\left(\sum_{s=1}^{b_t}\|\Delta_{t-\delta_t-s}\|\right)^2 + 4\eta^2(\gamma^2(c^{(i)}_{t-1})^2 + (\theta^{(i)}_t)^2)G^2 \tag{46}$$

Since each summation window contains at most τ terms due to Assumption 3, using $(\sum_{i=1}^n a_i)^2 \leq n\sum_i a_i^2$, we obtain

$$\left(\sum_{s=1}^{\delta_t}\|\Delta_{t-s}\|\right)^2 \leq \delta_t\sum_{s=1}^{\delta_t}\|\Delta_{t-s}\|^2 \leq \tau\sum_{s=1}^{\delta_t}\|\Delta_{t-s}\|^2, \tag{47}$$

$$\left(\sum_{s=1}^{b_t}\|\Delta_{t-\delta_t-s}\|\right)^2 \leq b_t\sum_{s=1}^{b_t}\|\Delta_{t-\delta_t-s}\|^2 \leq \tau\sum_{s=1}^{b_t}\|\Delta_{t-\delta_t-s}\|^2. \tag{48}$$

Plugging these to (46), taking full expectation with Lemma 1, we have

$$\mathbb{E}\left[\|\Delta_t\|^2\right] \leq 4(1-\gamma)^2\tau\sum_{s=1}^{\delta_t}\mathbb{E}[\|\Delta_{t-s}\|^2] + 4\gamma^2 r\tau\sum_{s=1}^{b_t}\mathbb{E}[\|\Delta_{t-\delta_t-s}\|^2] + 4\eta^2G^2\left(\gamma^2\mathbb{E}\left[(c^{(i)}_{t-1})^2\right] + \mathbb{E}\left[(\theta^{(i)}_t)^2\right)\right). \tag{49}$$

Furthermore, from the definition of $\theta_t^{(i)}$,

$$\mathbb{E}\left[\left(\theta_t^{(i)}\right)^2\right] = (1-\gamma)^2 + 2\gamma(1-\gamma)\,\mathbb{E}\left[c_{t-1}^{(i)}\right] + \gamma^2\mathbb{E}\left[\left(c_{t-1}^{(i)}\right)^2\right]. \tag{50}$$

Using Lemma 1, $\mathbb{E}\left[c_{t-1}^{(i)}\right] \le r$ and $\mathbb{E}\left[\left(c_{t-1}^{(i)}\right)^2\right] \le r$, so

$$\mathbb{E}\left[\left(\theta_t^{(i)}\right)^2\right] \le (1-\gamma)^2 + 2\gamma(1-\gamma)r + \gamma^2 r. \tag{51}$$

Therefore, the whole gradient-source term (i.e., the term involving G) in (49) can be bounded by

$$4\eta^2G^2\left(\gamma^2 r + (1-\gamma)^2 + 2\gamma(1-\gamma)r + \gamma^2 r\right) = 4\eta^2G^2\left((1-\gamma)^2 + 2\gamma r\right). \tag{52}$$

Summing over t gives

$$\sum_{t=1}^{T}\mathbb{E}\left[\|\Delta_t\|^2\right] \le 4(1-\gamma)^2\tau\sum_{t=1}^{T}\sum_{s=1}^{\delta_t}\mathbb{E}[\|\Delta_{t-s}\|^2] + 4\gamma^2 r\tau\sum_{t=1}^{T}\sum_{s=1}^{b_t}\mathbb{E}[\|\Delta_{t-\delta_t-s}\|^2] + 4T\eta^2G^2\Big((1-\gamma)^2 + 2\gamma r\Big). \tag{53}$$

For each double sum on the RHS, each term $\mathbb{E}[\|\Delta_k\|^2]$ for $k = 0, \dots T$ appears at most τ times. Hence,

$$\sum_{t=1}^{T}\mathbb{E}\left[\|\Delta_t\|^2\right] \le \underbrace{4\tau^2[(1-\gamma)^2 + \gamma^2 r]}_{\rho}\sum_{t=1}^{T}\mathbb{E}\left[\|\Delta_t\|^2\right] + 4T\eta^2G^2\Big((1-\gamma)^2 + 2\gamma r\Big). \tag{54}$$

Since $\rho < 1$ by assumption, rearranging gives

$$(1-\rho)\sum_{t=1}^{T}\mathbb{E}\left[\|\Delta_t\|^2\right] \le 4T\eta^2G^2\Big((1-\gamma)^2 + 2\gamma r\Big) \tag{55}$$

$$\iff \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\Delta_t\|^2\right] \le \frac{4T\eta^2G^2\Big((1-\gamma)^2 + 2\gamma r\Big)}{T(1-\rho)} = C_\Delta\eta^2, \tag{56}$$

where

$$C_\Delta := \frac{4G^2\Big((1-\gamma)^2 + 2\gamma r\Big)}{1-\rho}.$$

□

We also have the following corollary.

Corollary 1 (Bound on Average Global Drift)**.** *Under the same assumptions as Lemma 2, we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|w_{t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2\right] = C_w\eta^2, \tag{57}$$

where $C_w := \tau^2 C_\Delta$.

Proof. Using a similar argument, we have

$$\|w_{t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2 = \left(\sum_{s=1}^{\delta_t}\|\Delta_{t-s}\|\right)^2 \le \tau\sum_{s=1}^{\delta_t}\|\Delta_{t-s}\|^2 \tag{58}$$

Taking expectation, summing over t and noticing that each $\mathbb{E}[\|\Delta_t\|^2]$ appears at most τ times, we have

$$\sum_{t=1}^{T}\mathbb{E}[\|w_{t-1}^{(g)}-w_{t-\delta_t-1}^{(g)}\|^2]\leq\tau^2\sum_{t=1}^{T}\mathbb{E}[\|\Delta_t\|^2]. \tag{59}$$

Then from Lemma 2,

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|w_{t-1}^{(g)}-w_{t-\delta_t-1}^{(g)}\|^2]&\leq\frac{\tau^2}{T}\sum_{t=1}^{T}\mathbb{E}[\|\Delta_t\|^2] &(60)\\ &\leq\tau^2C_\Delta\eta^2 &(61)\\ &=C_w\eta^2 &(62)\end{aligned}$$

where $C_w:=\tau^2C_\Delta$. □

C Proof of Lemma 3

Lemma 3 (Bound for Term A of (18))**.** *Under Assumptions 1 to 6, we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\langle\nabla F\left(w_{t-1}^{(g)}\right),\Delta_t\rangle\right]\leq\left[\frac{1-\gamma+2\gamma r}{\eta}+\eta L^2\right]\frac{C_w}{T}+\eta\gamma rG^2-\frac{\eta(1-\gamma)}{4T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right], \tag{20}$$

where $C_w=\tau^2C_\Delta$.

Proof. Applying (17) to the Term A.

$$\begin{aligned}\text{Term A}=(1-\gamma)\big\langle\nabla F(w_{t-1}^{(g)}),w_{t-\delta_t-1}^{(g)}-w_{t-1}^{(g)}\big\rangle &(63)\\ +\mathbb{E}\left[\big\langle\nabla F(w_{t-1}^{(g)}),\gamma c_{t-1}^{(i)}\left(w_{t-\delta_t-1}^{(g)}-w_{t-\delta_t}^{(i)}\right)\big\rangle\right] &(64)\\ -\mathbb{E}\left[\big\langle\nabla F(w_{t-1}^{(g)}),\theta_t^{(i)}\eta g_t\big\rangle\right] &(65)\end{aligned}$$

For the **first part** (63), by Young's inequality for inner product ($\langle a,b\rangle\leq\frac{\alpha}{2}\|a\|^2+\frac{1}{2\alpha}\|b\|^2,\forall\alpha>0$), we can choose $\alpha=\eta$ to get

$$(1-\gamma)\big\langle\nabla F(w_{t-1}^{(g)}),w_{t-\delta_t-1}^{(g)}-w_{t-1}^{(g)}\big\rangle\leq(1-\gamma)\left[\frac{\eta}{2}\|\nabla F(w_{t-1}^{(g)})\|^2+\frac{1}{2\eta}\|w_{t-\delta_t-1}^{(g)}-w_{t-1}^{(g)}\|^2\right] \tag{66}$$

For the **second part** (64), by choosing $\alpha=\eta$ we get

$$\begin{aligned}&\mathbb{E}\left[\big\langle\nabla F(w_{t-1}^{(g)}),\gamma c_{t-1}^{(i)}\left(w_{t-\delta_t-1}^{(g)}-w_{t-\delta_t}^{(i)}\right)\big\rangle\right]\\ &=\mathbb{E}\left[\gamma c_{t-1}^{(i)}\big\langle\nabla F(w_{t-1}^{(g)}),w_{t-\delta_t-1}^{(g)}-w_{t-\delta_t}^{(i)}\big\rangle\right] &(67)\\ &\leq\mathbb{E}\Big[\gamma c_{t-1}^{(i)}\left(\frac{\eta}{2}\|\nabla F(w_{t-1}^{(g)})\|^2+\frac{1}{2\eta}\|w_{t-\delta_t-1}^{(g)}-w_{t-\delta_t}^{(i)}\|^2\right)\Big] &(68)\end{aligned}$$

Moreover, from the proof of Lemma 2 (see (43)), we know that

$$\begin{aligned}\|w_{t-\delta_t-1}^{(g)}-w_{t-\delta_t}^{(i)}\|^2&\leq2\|w_{t-b_t-1}^{(g)}-w_{t-\delta_t-1}^{(g)}\|^2+2\eta^2\|g_{t-\delta_t}\|^2. &(69)\\ &\leq2\|w_{t-b_t-1}^{(g)}-w_{t-\delta_t-1}^{(g)}\|^2+2\eta^2G^2. &(70)\end{aligned}$$

where the last step is due to Assumption 5. Therefore, (68) becomes

$$\mathbb{E}\left[\left\langle \nabla F(w_{t-1}^{(g)}), \gamma c_{t-1}^{(i)} \left(w_{t-\delta_t-1}^{(g)} - w_{t-\delta_t}^{(i)} \right) \right\rangle\right] \leq \mathbb{E}\Big[\gamma c_{t-1}^{(i)} \left(\frac{\eta}{2} \|\nabla F(w_{t-1}^{(g)})\|^2 + \frac{1}{\eta} \|w_{t-b_t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2 + \eta G^2 \right) \Big]. \tag{71}$$

Combining the first and second parts gives (by the definition of $\theta_t^{(i)}$ and Lemma 1)

$$(1-\gamma)\left\langle \nabla F(w_{t-1}^{(g)}), w_{t-\delta_t-1}^{(g)} - w_{t-1}^{(g)} \right\rangle + \mathbb{E}\left[\left\langle \nabla F(w_{t-1}^{(g)}), \gamma c_{t-1}^{(i)} \left(w_{t-\delta_t-1}^{(g)} - w_{t-\delta_t}^{(i)} \right) \right\rangle\right] \leq \mathbb{E}\left[\theta_t^{(i)} \frac{\eta}{2} \|\nabla F(w_{t-1}^{(g)})\|^2\right] + \frac{1-\gamma}{2\eta} \|w_{t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2 + \frac{\gamma r}{\eta} \|w_{t-b_t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2 + \eta\gamma r G^2. \tag{72}$$

For the **third part** (65), by the polarization identity ($-\langle a, b\rangle = \frac{1}{2}(\|a-b\|^2 - \|a\|^2 - \|b\|^2)$), we have

$$-\mathbb{E}\left[\left\langle \nabla F(w_{t-1}^{(g)}), \theta_t^{(i)} \eta g_t \right\rangle\right] = -\eta \langle \nabla F(w_{t-1}^{(g)}), \nabla F(w_{t-\delta_t-1}^{(g)}) \rangle \cdot \mathbb{E}[\theta_t^{(i)}] \qquad \text{(1) and Assumption 2} \tag{73}$$

$$\leq \frac{\eta}{2}\Big[\|\nabla F(w_{t-1}^{(g)}) - \nabla F(w_{t-\delta_t-1}^{(g)})\|^2 - \|\nabla F(w_{t-1}^{(g)})\|^2 - \|\nabla F(w_{t-\delta_t-1}^{(g)})\|^2\Big]\mathbb{E}[\theta_t^{(i)}] \qquad \text{Polarization} \tag{74}$$

$$\leq \frac{\eta}{2}\Big[L^2 \|w_{t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2 - \|\nabla F(w_{t-1}^{(g)})\|^2 - \|\nabla F(w_{t-\delta_t-1}^{(g)})\|^2\Big]\mathbb{E}[\theta_t^{(i)}]. \qquad \text{Assumption 1} \tag{75}$$

By $\|a-b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, choosing $a = \nabla F(w_{t-\delta_t-1}^{(g)}), b = \nabla F(w_{t-\delta_t-1}^{(g)}) - \nabla F(w_{t-1}^{(g)})$ offers a connection between the two gradients

$$\|\nabla F(w_{t-1}^{(g)})\|^2 \leq 2\|\nabla F(w_{t-\delta_t-1}^{(g)})\|^2 + 2\|\nabla F(w_{t-\delta_t-1}^{(g)}) - \nabla F(w_{t-1}^{(g)})\|^2 \tag{76}$$

$$\iff -\|\nabla F(w_{t-\delta_t-1}^{(g)})\|^2 \leq -\frac{1}{2}\|\nabla F(w_{t-1}^{(g)})\|^2 + \|\nabla F(w_{t-\delta_t-1}^{(g)}) - \nabla F(w_{t-1}^{(g)})\|^2 \tag{77}$$

$$\implies -\frac{\eta}{2}\|\nabla F(w_{t-\delta_t-1}^{(g)})\|^2 \leq -\frac{\eta}{4}\|\nabla F(w_{t-1}^{(g)})\|^2 + \frac{\eta L^2}{2}\|w_{t-\delta_t-1}^{(g)} - w_{t-1}^{(g)}\|^2. \tag{78}$$

Plugging this into (75) gives

$$-\mathbb{E}\left[\left\langle \nabla F(w_{t-1}^{(g)}), \theta_t^{(i)} \eta g_t \right\rangle\right] \leq \eta \left[L^2 \|w_{t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2 - \frac{3}{4}\|\nabla F(w_{t-1}^{(g)})\|^2\right]\mathbb{E}[\theta_t^{(i)}]. \tag{79}$$

Combining the bounds on these three parts (plugging (72) and (79) to (63)), we have

$$\text{Term A} \leq \frac{1-\gamma}{2\eta}\|w_{t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2 + \frac{\gamma r}{\eta}\|w_{t-b_t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2 + \eta\gamma r G^2 + \eta L^2 \|w_{t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2 - \frac{\eta}{4}\|\nabla F(w_{t-1}^{(g)})\|^2 \mathbb{E}[\theta_t^{(i)}] \tag{80}$$

$$\leq \frac{1-\gamma}{2\eta}\|w_{t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2 + \frac{\gamma r}{\eta}\|w_{t-b_t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2 + \eta\gamma r G^2 + \eta L^2 \|w_{t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2 - \frac{\eta(1-\gamma)}{4}\|\nabla F(w_{t-1}^{(g)})\|^2, \tag{81}$$

where the last inequality is due to $\theta_t^{(i)} \geq 1-\gamma$. Finally, we sum over t for Term A, in which case the sum of $\|w_{t-b_t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2$ can be replaced by the sum of $\|w_{t-1}^{(g)} - w_{t-\delta_t-1}^{(g)}\|^2$ since the latter sum has more terms.

Then we apply Corollary 1 to get

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\langle\nabla F\left(w_{t-1}^{(g)}\right),\Delta_t\rangle\right] \tag{82}$$

$$\leq\left[\frac{1-\gamma+2\gamma r}{\eta}+\eta L^2\right]\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|w_{t-1}^{(g)}-w_{t-\delta_t-1}^{(g)}\|^2\right]+\eta\gamma r G^2-\frac{\eta(1-\gamma)}{4T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right], \tag{83}$$

$$\leq\left[\frac{1-\gamma+2\gamma r}{\eta}+\eta L^2\right]\frac{C_w}{T}+\eta\gamma r G^2-\frac{\eta(1-\gamma)}{4T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right], \tag{84}$$

which concludes the proof. □

D Proof of Theorem 1

Theorem 1 (Convergence)**.** *Under Assumptions 1 to 6, we have*

$$\min_{t=1,\ldots,T}\mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right]\leq\frac{4}{1-\gamma}\left(\frac{F\left(w_0^{(g)}\right)-F_*}{\eta T}+C_w\left[\frac{1-\gamma+2\gamma r}{2}+\eta^2L^2\right]+\gamma r G^2+LC_\Delta\eta\right). \tag{21}$$

Furthermore, by choosing

$$\eta=\frac{1}{\sqrt{T}},\qquad r=\frac{2}{N},\qquad \gamma=1-\frac{1}{T^{1/4}},\qquad T=N^2,$$

we obtain a convergence rate of $\mathcal{O}\big(T^{-1/4}\big)$.

Proof. Applying telescoping sum to (18) (with iterative expectation/tower rule) and Lemmas 2 and 3, we have

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[F\left(w_t^{(g)}\right)\right]\leq\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[F\left(w_{t-1}^{(g)}\right)\right]+\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\langle\nabla F\left(w_{t-1}^{(g)}\right),\Delta_t\rangle\right]+\frac{L}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\Delta_t\|^2\right] \tag{85}$$

$$\leq\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[F\left(w_{t-1}^{(g)}\right)\right]+\left[\frac{1-\gamma+2\gamma r}{\eta}+\eta L^2\right]\frac{C_w}{T}$$

$$+\eta\gamma r G^2-\frac{\eta(1-\gamma)}{4T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right]+LC_\Delta\eta^2 \tag{86}$$

Rearranging gives (note that F is bounded from below by some value F^* by Assumption 1)

$$\frac{\eta(1-\gamma)}{4T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right]\leq\frac{F\left(w_0^{(g)}\right)-F_*}{T}+\left[\frac{1-\gamma+2\gamma r}{\eta}+\eta L^2\right]\frac{C_w}{T}+\eta\gamma r G^2+LC_\Delta\eta^2. \tag{87}$$

Each $\mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right]$ is bounded from below by $\min_{t=1,\ldots,T}\mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right]$, hence

$$\frac{\eta(1-\gamma)}{4T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right]\geq\frac{\eta(1-\gamma)}{4T}\min_{t=1,\ldots,T}\mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right]\sum_{t=1}^{T}1 \tag{88}$$

$$=\frac{\eta(1-\gamma)}{4}\min_{t=1,\ldots,T}\mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right] \tag{89}$$

Plugging this back to (87) gives

$$\min_{t=1,\dots,T} \mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right] \leq \frac{4}{1-\gamma}\left(\frac{F\left(w_0^{(g)}\right)-F_*}{\eta T} + \left[\frac{1-\gamma+2\gamma r}{\eta^2}+L^2\right]\frac{C_w}{T} + \gamma r G^2 + LC_\Delta \eta\right), \quad (90)$$

which completes the first part of the theorem.

By choosing $\eta = \frac{1}{\sqrt{T}}$, the bound is simplified to

$$\min_{t=1,\dots,T} \mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right] \leq \frac{4}{1-\gamma}\left(\frac{F\left(w_0^{(g)}\right)-F_*}{\sqrt{T}} + C_w(1-\gamma+2\gamma r) + \frac{L^2 C_w}{T} + \gamma r G^2 + \frac{LC_\Delta}{\sqrt{T}}\right). \quad (91)$$

Then by choosing

$$r = \frac{2}{N}, \qquad \gamma = 1 - \frac{1}{T^{1/4}}, \qquad T = N^2,$$

we have

$$\begin{aligned}
\min_{t=1,\dots,T} \mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right] &\leq 4T^{1/4}\Big((F(w_0^{(g)})-F^*)T^{-1/2} + C_w(T^{-1/4} + 2(1-T^{-1/4})(2T^{-1/2})) \\
&\quad + L^2 C_w T^{-1} + (1-T^{-1/4})(2T^{-1/2})G^2 + LC_\Delta T^{-1/2}\Big) \quad (92)\\
&= 4T^{1/4}\Big((F(w_0^{(g)})-F^*)T^{-1/2} + C_w(T^{-1/4} + 4T^{-1/2} - 4T^{-3/4} + L^2 T^{-1}) \\
&\quad + (2T^{-1/2} - 2T^{-3/4})G^2 + LC_\Delta T^{-1/2}\Big) \quad (93)\\
&= 4\Big((F(w_0^{(g)})-F^*)T^{-1/4} + C_w(1 + 4T^{-1/4} - 4T^{-1/2} + L^2 T^{-3/4}) \\
&\quad (2T^{-1/4} - 2T^{-1/2})G^2 + LC_\Delta T^{-1/4}\Big). \quad (94)
\end{aligned}$$

Note that for C_Δ,

$$C_\Delta = \frac{4G^2\Big((1-\gamma)^2 + 2\gamma r\Big)}{1-\rho} = \frac{4G^2}{1-\rho}\left(T^{-1/2} + 2(1-T^{-1/4})(2T^{-1/2})\right) = \mathcal{O}(T^{-1/2}). \quad (95)$$

Then $C_w = \tau^2 C_\Delta = \mathcal{O}(T^{-1/2})$. Plugging these to (94) gives

$$\min_{t=1,\dots,T} \mathbb{E}\left[\|\nabla F(w_{t-1}^{(g)})\|^2\right] = \mathcal{O}(T^{-1/4}). \quad (96)$$

□