On the Convergence of Continual Federated Learning Using Incrementally Aggregated Gradients

Satish Kumar Keshri
IIIT Delhi

Nazreen Shah
IIIT Delhi

Ranjitha Prasad
IIIT Delhi

Abstract

The holy grail of machine learning is to enable Continual Federated Learning (CFL) to enhance the efficiency, privacy, and scalability of AI systems while learning from streaming data. The primary challenge of a CFL system is to overcome global catastrophic forgetting, wherein the accuracy of the global model trained on new tasks declines on the old tasks. In this work, we propose *Continual Federated Learning with Aggregated Gradients* (C-FLAG), a novel replay-memory based federated strategy consisting of edge-based gradient updates on memory and aggregated gradients on the current data. We provide convergence analysis of the C-FLAG approach which addresses forgetting and bias while converging at a rate of $\mathcal{O}(1/\sqrt{T})$ over T communication rounds. We formulate an optimization sub-problem that minimizes catastrophic forgetting, translating CFL into an iterative algorithm with adaptive learning rates that ensure seamless learning across tasks. We empirically show that C-FLAG outperforms several state-of-the-art baselines on both task and class-incremental settings with respect to metrics such as accuracy and forgetting.

1 INTRODUCTION

The concept of lifelong learning in AI is inspired by the basic human nature of learning and adapting to new experiences and knowledge continuously throughout one's life. Continual learning (CL) is an important aspect of lifelong learning, where the key is to gain knowledge of new tasks without forgetting the previously gained expertise. Centralized lifelong learners are well-known Wang et al. (2024). However, increasing privacy concerns, the volume and complexity of data generated by various sources such as sensors, IoT devices, online platforms, and communication bottlenecks have led to the advent of continual federated learning (CFL) mechanisms.

A popular use-case of CFL is edge streaming analytics, where a stream of private data is analyzed continuously at the edge-user Georgiou et al. (2018); Ntumba et al. (2021), enabling organizations to extract insights for data-driven decisions without transmitting the data to a centralized server. Edge streaming analytics is well-suited for autonomous decision-making memory-constrained applications such as industrial IoT Sen et al. (2023); Husom et al. (2023), smart cities ul Haq et al. (2021), autonomous systems Shaheen et al. (2022) and remote monitoring Doshi and Yilmaz (2020). Conventional ML techniques necessitate retraining in order to adapt to the non-stationary streaming data Zenke et al. (2017) while computational and memory constraints restrict the simultaneous processing of previous and current datasets, rendering retraining impossible. Further, edge-based analytics without federation results in models that can only learn from its *direct experience* Yoon et al. (2021). A privacy-preserving strategy that allows continuous learning at the global level while circumventing all the above-mentioned issues is Continual Federated Learning (CFL) Yang et al. (2024).

In CFL, clients train on private data streams and communicate their local parameters to a server, which subsequently shares the global model with the clients. Several issues in CFL, such as inter-client interference Yoon et al. (2021), dynamic arrival of new classes into FL training Dong et al. (2023), and local and global catastrophic forgetting Bakman et al. (2023), have been studied. Memory-based replay techniques Dupuy et al. (2023); Good et al. (2023); Li et al. (2024) offer a direct mechanism to revisit past experiences without requiring access to the historical data of other clients. In particular, episodic replay, wherein a small, fixed-size replay buffer of past data is stored along with new data

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

at each client, has proven to be effective in reducing forgetting Dupuy et al. (2023). However, the replay buffer permits limited access to the dataset from past tasks, resulting in sampling bias Chrysakis and Moens (2023). Therefore, it is essential to jointly manage the bias and federation when developing replay-based CFL methods.

Consider a federated real-time surveillance use-case as depicted in Fig. 1 (right), where the edge analytics task is the continuous monitoring and analysis of data streams to respond to events as they occur. Let $\mathcal{P}^i$ comprise data from all the previous tasks at the i-th client until a given observation point $t = 0$. The i-th client samples data from $\mathcal{P}^i$ and stores it in the buffer $\mathcal{M}^i \subset \mathcal{P}^i$ as depicted in Fig. 1. Gradient updates on the data in $\mathcal{M}^i$ lead to biased gradients since $\mathcal{M}^i$ consists of a subset of data of all the previous tasks. At $t = 0$, the server transitions from the previous to the current task, and the FL model starts to learn from the current non-stationary dataset $\mathcal{C}^i$. The goal of the CFL framework is to learn from $\mathcal{C}^i, \forall i$, while mitigating the effect of catastrophic forgetting on $\mathcal{P}^i$.

Contributions: We propose the novel *Continual Federated Learning with Aggregated Gradients* (C-FLAG) technique, which is a replay-based CFL strategy consisting of local learning steps and global aggregation at the server. We consider the incremental aggregated gradient (IAG) approach, which significantly reduces computation costs and comes with the benefits of variance reduction techniques Mitra et al. (2021). To jointly mitigate the issues of client drift, bias, and forgetting, an *effective gradient*, which is a combination of a single gradient on the memory buffer and multiple gradients on the current task, is proposed, as depicted in Fig. 1 (left). Our contributions are as follows:

- We formulate the CFL problem as a smooth, non-convex finite-sum optimization problem and theoretically demonstrate that the proposed C-FLAG approach converges to a stationary point at a rate of $\mathcal{O}(\frac{1}{\sqrt{T}})$ over T communication rounds.
- We formulate an optimization sub-problem parameterized by the learning rate to minimize catastrophic forgetting, allowing the translation of C-FLAG into an iterative algorithm with adaptive learning rates for seamless learning across tasks.

We evaluate C-FLAG on task-incremental FL setups, where it consistently outperforms baseline methods in terms of both average accuracy and forgetting. We also perform ablation studies on data heterogeneity, varying number of clients, and the size/type of replay buffer. The results show that C-FLAG outperforms well-known and state-of-the-art baselines in mitigating forgetting and enhancing overall model performance.

To the best of our knowledge, this work is the first of its kind to propose a replay-based CFL framework with convergence guarantees. The crux of the theoretical analysis deals with the handling of bias due to memory constraints and characterizing catastrophic forgetting. While prior FL works typically rely on convex or strongly convex assumptions, our analysis extends to the more general non-convex setting.

2 PROBLEM FORMULATION

In a typical FL setting, N edge devices collaboratively solve the finite-sum optimization problem given as:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \phi(\mathbf{x}) = \min_{\mathbf{x}\in\mathbb{R}^d} \textstyle\sum_{i=1}^{N} p_i \phi_i(\mathbf{x}) \tag{1}$$

where $\phi_i(\mathbf{x})$ is the local loss function at the i-th client, $\phi(\mathbf{x})$ is the global loss function and p_i is the weight assigned to i-th client. Each client independently computes gradients on a local private dataset, and the central server receives and aggregates these updates using a predefined strategy McMahan et al. (2017). In contrast, the edge devices in a CFL framework observe private streaming data, and the goal is to adapt the models as the data arrives, without forgetting the knowledge gained from past experiences. Given the global memory and current datasets $\mathcal{C} = \{\mathcal{C}^1, \mathcal{C}^2, \ldots, \mathcal{C}^N\}$ and $\mathcal{P} = \{\mathcal{P}^1, \mathcal{P}^2, \ldots, \mathcal{P}^N\}$, where $\mathcal{P}^i$ and $\mathcal{C}^i$ represents the past and the current task at the i-th client, the continual learning problem is defined as a smooth finite-sum optimization problem given as

$$\min_{\mathbf{x}\in\mathbb{R}^d} h(\mathbf{x}) := \textstyle\sum_{i=1}^{N} p_i h_i(\mathbf{x}), \tag{2}$$

where $h(\cdot)$ is a smooth, non-convex function which decomposes into $f(\mathbf{x})$ and $g(\mathbf{x})$ as

$$h(\mathbf{x}) = \frac{|\mathcal{P}|}{|\mathcal{P}| + |\mathcal{C}|} f(\mathbf{x}) + \frac{|\mathcal{C}|}{|\mathcal{P}| + |\mathcal{C}|} g(\mathbf{x}). \tag{3}$$

Here, $f(\mathbf{x})$ and $g(\mathbf{x})$ represents the restriction of $h(\mathbf{x})$ on the datasets $\mathcal{P} = \{\mathcal{P}^1, \mathcal{P}^2, \ldots, \mathcal{P}^N\}$ and $\mathcal{C} = \{\mathcal{C}^1, \mathcal{C}^2, \ldots, \mathcal{C}^N\}$, respectively, and $|\mathcal{P}| = \sum_{i=1}^{N} |\mathcal{P}^i|$, $|\mathcal{C}| = \sum_{i=1}^{N} |\mathcal{C}^i|$. Additionally, we also define the global functions $f(\mathbf{x})$ and $g(\mathbf{x})$ in terms of the local optimization objectives as $f(\mathbf{x}) := h(\mathbf{x})|_{\mathcal{P}} := \sum_{i=1}^{N} p_i f_i(\mathbf{x})$ and $g(\mathbf{x}) := h(\mathbf{x})|_{\mathcal{C}} := \sum_{i=1}^{N} p_i g_i(\mathbf{x})$. Here, $f_i(\mathbf{x})$ and $g_i(\mathbf{x})$ are the restrictions on the local previous and current datasets, respectively. Episodic memory $\mathcal{M}^i$ consisting of a fixed-size buffer of size at most m_0 stores a subset of the data that arrives prior to the start of the current task ($t = 0$) at the i-th client Dupuy et al. (2023). After sampling, it remains fixed until it trains on the current task over T communication rounds, i.e., $\mathcal{M}^i = \mathcal{M}^i_t \; \forall t \in \{0, 1, \cdots, T-1\}$ and $i \in [N]$. We define the

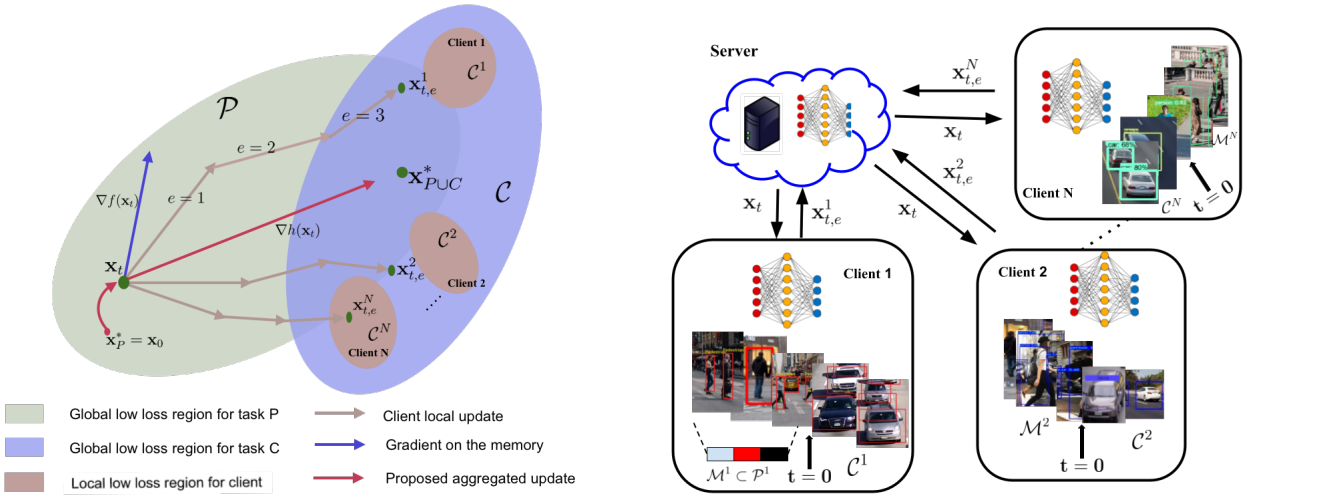

Figure 1: (Left) Illustration of C-FLAG: Initialised at the optimal point of the previous tasks $\mathbf{x}^*_{\mathcal{P}} = \mathbf{x}_0$, at the t-th iteration, i-th client takes E local steps towards its local optimal regions (pink regions). To balance learning and forgetting, C-FLAG takes a single step towards local memory and E steps on the local current data. The global aggregated model moves towards a common global minima $\mathbf{x}^*_{\mathcal{P}\cup\mathcal{C}}$. (Right) Real-time surveillance where a subset of previous tasks are stored in memory until $T = 0$. Data arriving thereafter is the current task $\mathcal{C}^i$.

global memory dataset as $\mathcal{M} := \{\mathcal{M}^1, \mathcal{M}^2, \ldots, \mathcal{M}^N\}$. Since the replay buffer at the edge, $\mathcal{M}^i$, permits limited access to the dataset from past tasks, gradient-based approaches result in bias Chrysakis and Moens (2023).

Streaming data tends to be non-stationary, while the conventional FL framework always assumes stationary data. Consequently, the convergence of any CFL framework is non-intuitive, and hence, it is essential to theoretically prove that the proposed strategy converges for the previous as well as on the new task.

3 C-FLAG ALGORITHM

C-FLAG consists of local learning steps, where an effective gradient is computed at each client, and a global aggregation step is performed at the server. At the i-th client, C-FLAG computes an effective gradient which is a combination of a gradient on the local memory buffer $\mathcal{M}^i$, and E gradient steps on the dataset of the current task $\mathcal{C}_i$. The E gradient steps is obtained using the update rule as follows:

$$\mathbf{x}^i_{t,k+1} = \mathbf{x}^i_{t,k} - \beta_t \left(\nabla g(\mathbf{x}_t) - \nabla g_i(\mathbf{x}_t) + \nabla g'_i(\mathbf{x}^i_{t,k})\right), \tag{4}$$

where $(\nabla g(\mathbf{x}_t) - \nabla g_i(\mathbf{x}_t))$ represents the one-time computation of the difference between the global and the local full gradient of the loss function, respectively. Essentially, $\nabla g(\mathbf{x}_t)$ provides an insight into the global descent direction at the beginning of round t. This descent direction is computed at a stale iterate $\mathbf{x}^i_{t,0} = \mathbf{x}_t$, and not at the current iterate. To account for this deviation, Mitra et al. (2021) proposes IAG as an approximation to $\nabla g_i(\mathbf{x}^i_{t,k})$, which we represent as $\nabla g'_i(\mathbf{x}^i_{t,k})$ defined as follows:

$$\nabla g'_i(\mathbf{x}^i_{t,k}) := \tfrac{1}{|\mathcal{C}^i|}\textstyle\sum_{j\in\mathcal{C}^i} \nabla g_{i,j}(\mathbf{x}^i_{t,\tau^i_{k,j}}). \tag{5}$$

At each local step k, the i-th client computes the gradient of a single, say the j-th component function of $g_i(\mathbf{x}^i_{t,k})$ given by $g_{i,j}(.)$ for $j \in \mathcal{C}^i$. For the remaining $r \in \mathcal{C}^i, r\backslash j$ component functions, the most recently computed gradients are retained. In order to track the component function that is updated, we use an index $\tau^i_{k,j}$ which denotes the most recent local step. At the beginning of the first local epoch ($k = 0$), the i-th client computes the full gradient and sets $\tau^i_{0,j} = 0\ \forall j \in \mathcal{C}^i$.

The evolution of the index $\tau^i_{k,j}$ can be understood using the following toy example. Suppose, the current dataset $\mathcal{C}^i$ has 3 data points, i.e., $\mathcal{C}^i = \{c_1, c_2, c_3\}$. At the beginning of the t-th round $\nabla g'_i(\mathbf{x}^i_{t,0})$ is computed as $\nabla g'_i(\mathbf{x}^i_{t,0}) = \frac{1}{3}(\nabla g_{i,c_1}(\mathbf{x}_0) + \nabla g_{i,c_2}(\mathbf{x}_0) + \nabla g_{i,c_3}(\mathbf{x}_0)) = \nabla g_i(\mathbf{x}_0)$. Hence, $\tau^i_{t,j} = 0$ for all j. At $k = 1$, the client randomly samples c_1, computes $\nabla g_{i,c_1}(\mathbf{x}^i_1)$ and updates $\tau^i_{t,d_1} = 1$. For the other components the previously computed gradients are used, and hence, $\nabla g'_i(\mathbf{x}^i_{t,1}) = \frac{1}{3}(\nabla g_{i,c_1}(\mathbf{x}^i_{t,1}) + \nabla g_{i,c_2}(\mathbf{x}^i_{t,0}) + \nabla g_{i,c_3}(\mathbf{x}^i_{t,0}))$. At $k = 2$, suppose the client randomly samplesc_3, computes the gradient on c_3 and updates $\tau^i_{t,c_3} = 2$, which leads to $\nabla g'_i(\mathbf{x}^i_{t,1}) = \frac{1}{3}(\nabla g_{i,c_1}(\mathbf{x}^i_{t,1}) + \nabla g_{i,c_2}(\mathbf{x}^i_{t,0}) + \nabla g_{i,c_3}(\mathbf{x}^i_{t,2}))$. This process continues until $k = E - 1$. Using delayed gradients at each local epoch, the gradient computation cost drastically decreases as only one component is updated at a time.

To mitigate catastrophic forgetting while transitioning from one task to another, we update the model on both, the memory buffer data and the current data. However, frequent training on the memory data may lead to overfitting and impede learning from the current dataset. Hence, we propose to take one step towards the memory data for every E local step on current data utilizing the memory as a guide for future learning. Accordingly, for each client $i \in [N]$, we obtain the biased gradient on the memory buffer $\mathcal{M}^i$ as

$$\nabla f_i^\dagger(\mathbf{x}) = \nabla f_i(\mathbf{x}) + b_i(\mathbf{x}), \tag{6}$$

where $b_i(\cdot)$ quantifies the bias introduced due to the sampling of the memory data $\forall i \in [N]$. Additionally, we define the average bias as $b(\mathbf{x}) = \sum_{i=1}^N p_i b_i(\mathbf{x})$. After E local epochs, the i-th client communicates $\Delta \mathbf{x}_t^i = \mathbf{x}_{t,E}^i - \alpha_t \nabla f^\dagger(\mathbf{x}_t)$, where $\mathbf{x}_{t,E}^i$ is obtained using (4). The server updates the global model as;

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha_t \textstyle\sum_{i=1}^N p_i \nabla f^\dagger(\mathbf{x}_t) - \beta_t \sum_{i=1}^N p_i \mathbf{x}_{t,E}^i, \tag{7}$$

where, α_t and β_t are the learning rates on the memory and current data, respectively. The steps of the algorithm are presented in Alg. 1, and the pseudocode for `AdapLR` is provided in the supplementary.

Algorithm 1 C-FLAG: Continual Federated Learning with Aggregated Gradients

Require: Step-size α, β, initial model $\mathbf{x}_0$, *AdapFlag*
Ensure: $\mathbf{x}_t$ for $t = 1, \ldots, T$

```
for t = 0, ..., T − 1 do
    For client i = 1, ..., N, compute ∇g_i(x_t) and ∇f_i^†(x_t), and transmit to the server.
    Server computes ∇g(x_t) and ∇f^†(x_t), and broadcasts to each client.
    for client i = 1, 2, ... N in parallel do
        Set x_{t,0}^i = x_t
        for k = 0, ..., E − 1 do
            Compute ∇g_i'(x_{t,k}^i) using (5) and x_{t,k+1}^i using (4).
        end for
        Δx_t^i = AdapLR(x_{t,E}^i, ∇f^†(x_t), α, β, AdapFlag).
        Transmit Δx_t^i to the server.
    end for
    Server computes and broadcasts x_{t+1} using x_{t+1} = x_t − Σ_{i=1}^N p_i Δx_t^i.
end for
```

4 CONVERGENCE ANALYSIS

In this section, we present a theoretical convergence analysis of the proposed memory-based continual learning framework in a non-convex setting. For purposes of brevity, proofs have been delegated to the supplementary. The assumptions are as follows:

Assumption 1. *(L-smoothness). For all $i \in [N]$, f_i, g_i, h_i are L-smooth.*

Assumption 2. *(Bounded Bias). There exists constants $0 \le m_i < 1$ for all $\mathbf{x} \in \mathbb{R}^d$ such that $\|b_i(\mathbf{x})\|^2 \le m_i \|\nabla f_i(\mathbf{x})\|^2, \quad \forall i \in [N]$.*

Assumption 3. *(Bounded memory gradient). There exists $r_i \in \mathbb{R}^+$, such that $\|\nabla f_i^\dagger(\mathbf{x}_t)\| \le r_i \|\nabla g(\mathbf{x}_t)\|$ for all $i \in [N]$.*

Since the biased gradient, $\nabla f_i^\dagger(\mathbf{x}_t)$, on the memory data is correlated with the true gradients, $\nabla f_i(\mathbf{x}_t)$, Assumption 2 is similar to Assumption 4 in Ajalloeian and Stich (2020). We also define the expectation over memory datasets till t-th global iteration as $\mathbb{E}_{\mathcal{M}_t} = \mathbb{E}_{[\mathcal{M}_0:\mathcal{M}_t]}$. Further we denote $\mathbb{E}$ as $\mathbb{E}_{[\mathcal{M}_0:\mathcal{M}_T]}$ over T global iterations.

As the iterations progress according to algorithm 1, two additional quantities, namely an overfitting term, and a catastrophic forgetting term, vary with time. In the following lemma and theorem, we formally introduce these terms and provide a convergence guarantee of C-FLAG on the previous task $\mathcal{P}$.

Lemma 1. *Suppose that the assumptions 1, 2 hold, $\alpha_t < \frac{2}{L(1+m)}$ and $m \in \mathbb{R}^+$. For the sequence $\{\mathbf{x}_t\}_{t=1}^T$ generated by algorithm 1, we have*

$$\|\nabla f(\mathbf{x}_t)\|^2 \le \frac{1}{\alpha_t[1 - \frac{L}{2}\alpha_t(1+m)]}\big(f(\mathbf{x}_t) - f(\mathbf{x}_{t+1}) + B(t) + \Gamma(t)\big), \tag{8}$$

where $B(t)$ is the overfitting term defined as

$$B(t) = (L\alpha_t^2 - \alpha_t)\langle \nabla f(\mathbf{x}_t), b(\mathbf{x}_t)\rangle + \beta_t \langle b(\mathbf{x}_t), \textstyle\sum_{i=1}^N \sum_{k=1}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i)\rangle. \tag{9}$$

Further, $\Gamma(t)$ is the forgetting term defined as

$$\Gamma(t) = L\beta_t^2 \|\textstyle\sum_{i=1}^N \sum_{k=1}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i)\|^2 - \beta_t(1 - L\alpha_t) \langle \nabla f^\dagger(\mathbf{x}_t), \textstyle\sum_{i=1}^N \sum_{k=1}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i)\rangle. \tag{10}$$

As the quantities $B(t)$ and $\Gamma(t)$ accumulate over time, they significantly degrade the performance of the continual learning framework Han et al. (2023). In particular, the term $\langle \nabla f^\dagger(\mathbf{x}_t), \sum_{i=1}^N \sum_{k=1}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i)\rangle$ is a key factor that determines interference and transference Chaudhry et al. (2018); Han et al. (2023) between locally aggregated gradients and the gradient on the memory. We provide a detailed discussion on the aspects of interference and transference as related to deriving adaptive learning rates in Sec. 5.

To derive the convergence of Alg. 1 we telescope over the training iterations for the current dataset, and using Lem. 1 we obtain the following theorem.

Theorem 2. *Suppose that the assumptions 1, 2 hold. Given $F = f(\mathbf{x}_0) - f(\mathbf{x}_T)$, the sequence $\{\mathbf{x}_t\}_{t=1}^T$ generated by algorithm 1 with $\alpha_t = \alpha = \frac{1}{L(m+1)}$ $\forall t \in \{0, 1, \cdots, T-1\}$, and $m \in \mathbb{R}^+$, satisfies*

$$\min_t \mathbb{E}\left[\|\nabla f(\mathbf{x}_t)\|^2\right] \leq \frac{2L(1+m)}{T}\left(F + \sum_{t=0}^{T-1} \mathbb{E}[\Gamma(t)]\right).$$

We observe that the expected convergence of $f(\cdot)$ depends on the initialization and the forgetting term. The overfitting term tends to zero in expectation, as shown in the supplementary material. On the other hand, the cumulative forgetting term accumulates with each server iteration. Hence, a tight upper bound on the convergence of the previous task necessitates an upper bound on $\Gamma(t)$, which we do in the sequel.

It is also essential to obtain the convergence guarantee on the current task if the updates are obtained using Alg. 1, where both the memory and the current dataset are used together. This leads to deriving the convergence rate of the global loss function, $h(\cdot)$, which learns jointly with respect to the current task and the replay memory at each client.

Lemma 3. *Given assumption 1, the sequence $\{\mathbf{x}_t\}_{t=1}^T$ generated with learning rates $\alpha_t = \beta_t = \alpha = \frac{1}{30LE}$, for the restriction of $h(\cdot)$ on $\mathcal{M} \cup \mathcal{C}$ given as $\tilde{h}(\mathbf{x}_t) = h_{|_{\mathcal{M}\cup\mathcal{C}}}(\mathbf{x}_t)$ we have*

$$\min_t \|\nabla \tilde{h}(\mathbf{x}_t)\|^2 \leq \tfrac{60L}{T}\left(\tilde{h}(\mathbf{x}_0) - \tilde{h}(\mathbf{x}_T)\right). \quad (11)$$

From the above lemma, we observe that using the proposed update rule, the overall global loss function reaches a stationary point in sub-linear $\mathcal{O}(\frac{1}{T})$ time complexity. Since the global loss function converges on $\mathcal{M} \cup \mathcal{C}$, the convergence on current task C is evident from the fact that $\mathcal{C} \subset \mathcal{M} \cup \mathcal{C}$. In the next lemma, we will provide an upper bound on the cumulative forgetting term.

Lemma 4. *Suppose that the assumptions 1, 2, 3 hold and the step-sizes satisfy $\alpha_t = \alpha < \frac{2}{L(1+m)}$ and $\beta_t = \beta < \frac{c}{\sqrt{T}} \forall t \in \{0, 1, \cdots, T-1\}$ and for some $c, m \in \mathbb{R}^+$. Then the following holds for the forgetting term $\Gamma(t)$:*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\Gamma(t)] < \mathcal{O}\left(\frac{1}{T} + \frac{1}{\sqrt{T}}\right). \quad (12)$$

One key observation from the above lemma is that using constant step sizes for updates on the memory data (via α) and diminishing step sizes for updates on the current data (via β) is crucial for a tight bound on cumulative forgetting. If we use constant step-sizes for updates on both memory and current data instead of diminishing rates, then the cumulative forgetting term is only guaranteed to converge with constant time complexity $\mathcal{O}(1)$. Finally, in the next theorem, we show that C-FLAG converges on the previous task, alleviating the problem of catastrophic forgetting.

Theorem 5. *Let $\{\mathbf{x}_t\}_{t=1}^T$ be the sequence generated by algorithm 1, and the step-sizes satisfy $\alpha_t = \frac{1}{L(m+1)}$ and $\beta_t < \frac{c}{\sqrt{T}} \forall \{0, 1, \cdots, T-1\}$ and for some $c, m \in \mathbb{R}^+$. Then, we obtain the following rate of convergence*

$$\min_t \mathbb{E}\left[\|\nabla f(\mathbf{x}_t)\|^2\right] < \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) \quad (13)$$

From the above, we obtain a sub-linear $\mathcal{O}(\frac{1}{\sqrt{T}})$ convergence rate for the proposed CFL problem. Theoretically, we show that diminishing step sizes for the current task is important to have a better upper bound on the convergence on f. In the centralized continual learning setup, Han et al. (2023) also provides $\mathcal{O}(\frac{1}{\sqrt{T}})$ convergence rate, but they impose additional constraints on the learning rates. Without these constraints, convergence on the past task is not guaranteed in their work and may diverge at the rate $\mathcal{O}(\sqrt{T})$.

5 C-FLAG: ADAPTIVE LEARNING RATES

In the previous sections, we provided theoretical convergence guarantees and the rate of convergence of the proposed C-FLAG framework. We observed that improved convergence on the previous task is possible if the cumulative forgetting term is as small as possible. In particular, in Theorem 2 we discussed that the convergence rate is dependent on the cumulative forgetting term $\sum_{t=0}^{T-1}\mathbb{E}[\Gamma(t)]$, and additionally, Lem. 4 specifies the constraints on α_t and β_t for obtaining an upper bound on the average of the cumulative forgetting terms over T iterations. In this section, we translate the analysis into an easily implementable solution where learning rates α_t and β_t are adapted to achieve lower forgetting at each iteration. Ideally, the adaptable learning rates solve the following constrained optimization problem

$$\min_{\alpha_t, \beta_t} \mathbb{E}[\Gamma(t)] \quad \text{subj. to } \alpha_t < \frac{2}{L(1+m)} \forall t < T. \quad (14)$$

This is a popular strategy in centralized continual learning, where adjusting learning rates trades off between learning new information and retaining previous knowledge Han et al. (2023). The above formulation allows us to adapt the learning rates to effectively balance learning and forgetting at the global level, leading to a practically implementable solution. Since the learning rate is within the constraint, it also enjoys theoretical convergence guarantees on both $f(\mathbf{x})$ and $g(\mathbf{x})$.

To derive the adaptive rates, we analyze the catastrophic forgetting term at the end of the t-th communication round ((10) in Lem. 1) given as

$$\Gamma(t) = \frac{L\beta_t^2}{2}\left\|\sum_{i=1}^{N}\sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i)\right\|^2 - \beta_t(1-L\alpha_t)\Lambda_t, \tag{15}$$

where we denote $\Lambda_t = \sum_{i=1}^{N} p_i \Lambda_{t,i}$ and $\Lambda_{t,i} = \langle \nabla f^\dagger(\mathbf{x}_t), \sum_{k=0}^{E-1} \nabla g_i'(\mathbf{x}_{t,k}^i)\rangle$. We observe from (15) that the first term is always non-negative, $(1-L\alpha_t) > 0$ (for $m > 1$) and $\beta_t > 0$ and hence, the term Λ_t is crucial to optimize $\Gamma(t)$. If $\Lambda_t > 0$, it results in a favorable decrease in $\Gamma(t)$ while enhancing learning on the previous and current task, and this case is termed as *transference*. Further, $\Lambda_t \leq 0$ leads to an increase in $\Gamma(t)$ resulting in forgetting, and this case is termed as *interference*. Since $\Gamma(t)$ is a quadratic polynomial in β_t, the optimal value of β_t and $\mathbb{E}[\Gamma(t)]$ is obtained as

$$\beta_t^* = \frac{(1-L\alpha_t)\Lambda_t}{L\left\|\sum_{i=1}^{N}\sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i)\right\|^2}, \text{and}$$
$$\mathbb{E}[\Gamma^*(t)] = -\frac{(1-L\alpha_t)^2\Lambda_t^2}{2L\left\|\sum_{i=1}^{N}\sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i)\right\|^2}. \tag{16}$$

A subtle trick that can be used here is as follows: at the beginning of the training at $t = 0$, the server fixes $\alpha_t = \beta_t = \alpha \forall t \in \{0, 1, \cdots, T-1\}$ and clients train on their datasets for E local epochs. Then at the end of each global communication round t, if the server observes a case of transference ($\Lambda_t > 0$) it rescales the client updates by $\frac{\beta_t^*}{\alpha}$ to get the effective learning rate as β_t^*. However, the above analysis handles transference on an average basis and not a per-client basis. This implies that some clients may observe interference locally ($\Lambda_t^i \leq 0$), but their contribution is nullified while computing Λ_t. Hence, we consider the approach where we minimize the clients' contribution to $\Gamma(t)$ individually to better control cumulative forgetting.

In order to provide the client specific analysis, let $\mathbf{a}_i = \sum_{k=0}^{E-1} \nabla g_i'(\mathbf{x}_{t,k}^i)$ and the client interaction terms $C_{i,j} = \langle \mathbf{a}_i, \mathbf{a_j} \rangle$. The first term in $\Gamma(t)$ can be written as:

$$||\sum_{i=1}^{N} p_i \mathbf{a}_i||^2 = \sum_{i=1}^{N} p_i^2 \left\|\mathbf{a}_i\right\|^2 + \underbrace{\sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N} p_i p_j C_{i,j}}_{A_{i,j}}.$$

The client interaction terms play an important role in the analysis. If $C_{i,j} \leq 0$, it can further reduce $\Gamma(t)$ and this leads to alleviated forgetting, while $C_{i,j} > 0$ leads to an increase in $\Gamma(t)$ and increased forgetting. Due to privacy concerns, these inter-client interaction terms cannot be utilised directly. In our study, we consider two cases; (a) the *average case* where $A_{i,j} = 0$, and (b) the *worst case* where $C_{i,j} > 0 \; \forall i, j \in [N]$ and $i \neq j$. We derive adaptive learning rates by analyzing the average and the worst case individually. As mentioned earlier, we may fix the learning rate at the beginning of the training on the current task and rescale based on the interference or transference case. Additionally, we denote the catastrophic forgetting obtained using the adapted rates as $\Gamma_{i,ad}(t)$ and $\Gamma_{ad}(t)$ for the i-th client and the server, respectively. The adaptive rates obtained after tackling the interference and transference are presented in Table 1. In both the average and the worst case when the i-th client interferes with the past learning, we adapt α_t to $\alpha_{t,i}$ while retaining β_t as α. On the other hand, in the case of transferring clients, we adapt β_t to $\beta_{t,i}$ and retain α_t as α. We summarise the proposed adaptive rates in Table 1.

Table 1: Table for adaptive learning rates to optimize catastrophic forgetting. In the table, I stands for interference ($\Lambda_{t,i} \leq 0$), and T stands for transference ($\Lambda_{t,i} > 0$).

Case	Type	$\alpha_{t,i}$	$\beta_{t,i}$
Average	I	$\alpha(1 - \frac{\Lambda_{t,i}}{\|\nabla f^\dagger(\mathbf{x}_t)\|^2})$	α
	T	α	$\frac{(1-L\alpha)\Lambda_{t,i}}{Lp_i\|\mathbf{a}_i\|^2}$
Worst	I	$\alpha(1 - \frac{\Lambda_{t,i}}{\|\nabla f^\dagger(\mathbf{x}_t)\|^2})$	α
	T	α	$\frac{(1-L\alpha)\Lambda_{t,i}}{LNp_i\|\mathbf{a}_i\|^2}$

The following lemmas show that our proposed choice of adaptive $\alpha_{t,i}$ and $\beta_{t,i}$ in the case of interference and transference, respectively, leads to lower forgetting, i.e., $\mathbb{E}[\Gamma_{i,ad}(t)] \leq \mathbb{E}[\Gamma_i(t)]$.

Lemma 6. *In the case of interference at the i-th client, for both the average and worst cases, adaptive rates* $\alpha_{t,i} = \alpha(1 - \frac{\Lambda_{t,i}}{\|\nabla f^\dagger(\mathbf{x}_t)\|^2})$ *and* $\beta_{t,i} = \alpha$ *lead to smaller forgetting, that is* $\mathbb{E}[\Gamma_{i,ad}(t)] \leq \mathbb{E}[\Gamma_i(t)]$.

Lemma 7. *Client-wise adaptive rates lead to improved forgetting,* $\mathbb{E}[\Gamma_{ad}(t)] \leq \mathbb{E}[\Gamma(t)]$.

6 RELATED WORKS AND DISCUSSIONS

Memory Based Approaches: In order to mitigate catastrophic forgetting (French (1999)), memory-based approaches store a subset of data to be leveraged during training. One way to utilise the memory data is to impose constraints while optimizing loss while learning new tasks. Centralized continual learning approaches such as GEM Lopez-Paz and Ranzato (2017) and A-GEM Chaudhry et al. (2018) impose these constraints through gradient projection such that the loss on the previous tasks does not increase. Episodic replay

(ER-replay) memory-based approaches directly use the gradients on the memory during training Chaudhry et al. (2019), Rolnick et al. (2019). Achieving the right balance in learning the past tasks and new ones, known as the stability-plasticity trade-off Liang and Li (2023), is crucial for continual learners. In continual federated learning (CFL), Dupuy et al. (2023) highlights the role of ER-replay memory at each client. Existing methods with ER-replay-based CFL include exemplar Hendryx et al. (2021) and prototypical-class Shenaj et al. (2023) based methods. In Zizzo et al. (2022), authors demonstrate that direct application of replay methods on temporal shifts in client behavior leads to poor results, and as a solution, they propose data sharing between clients employing differential privacy. A drawback of existing approaches is that while the convergence behavior has been demonstrated through empirical results, theoretical guarantees on their convergence are unfounded. In contrast, C-FLAG effectively utilises the replay memory while maintaining the stability-plasticity trade-off by performing a single step of training on the memory data as a guide for multiple local steps on the current task dataset using adaptive learning rates. We also provide theoretical convergence guarantees by addressing catastrophic forgetting.

Stale Gradients for Client Drift Mitigation: The incrementally aggregated gradient (IAG) Blatt et al. (2007) method significantly reduces computation costs but introduces stale gradients, which can deviate from the true gradient direction. In the federated setting, FedTrack Mitra et al. (2021) uses IAG and demonstrates that it improves the convergence rate in comparison to non-IAG methods. Furthermore, stale server gradients from the last round are used to reduce the client drift between two communication rounds. C-FLAG takes inspiration from IAG-based updates to reduce client drift and jointly learn from memory and current data to mitigate forgetting.

Comparison with NCCL (Han et al. (2023)): NCCL is a centralized continual learning algorithm that greedily adapts learning rates at each iteration and trains a model jointly on memory and current data. NCCL is a strategy that uses adaptive step sizes for continual learning when the optimization problem is smooth and non-convex. NCCL overcomes several shortcomings in Lopez-Paz and Ranzato (2017) and Chaudhry et al. (2018) in the context of adapting the learning rates. C-FLAG is a federated counterpart of NCCL as an adaptive learner, when data is available at the edge user. As FL involves local training, the global greedy adaptation of learning rates in every epoch is not feasible. In our approach, IAG helps to adapt the learning rates based on the accumulated gradients. This allows C-FLAG to optimize the learning rates at each client while reducing global catastrophic forgetting.

7 EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we present the experimental results to demonstrate the efficacy of the proposed C-FLAG algorithm in task-incremental settings.

Benchmarks: We demonstrate the experimental results on continual learning benchmarks such as Split-CIFAR10, Split-CIFAR100, and Split-TinyImageNet. Split-CIFAR10 and Split-CIFAR100 are derived from the CIFAR10 and CIFAR100 Krizhevsky and Hinton (2009) datasets, where the entire dataset is split into 5 tasks. Each task in Split-CIFAR10 has two classes, while each task in Split-CIFAR100 contains 20 classes. In the TinyImageNet Le and Yang (2015) dataset, data is divided into 10 tasks, each consisting of 20 classes. For non-IID splits, we employ the Dirichlet partitioning technique to distribute task-specific data among the participating clients. This allows us to regulate the data heterogeneity in FL using the Dirichlet parameter ζ. In this work, we use $\zeta = 0.1$ and $\zeta = 10^5$ for simulating the non-IID and IID scenarios, respectively. We use a ResNet-18 He et al. (2015) backbone for classification on Split-CIFAR10 and Split-CIFAR100 datasets, whereas we use a ResNet-50 backbone for classification on Split-TinyImageNet dataset. on all the datasets. Other details of the experimental configuration have been delegated to the appendix.

Baselines: We compare the proposed method in both task-incremental and class-incremental setups against the following baselines. The primary distinction between these two setups lies in the usage of task/class identity. In the task-incremental setup, task identity is required during both the training and testing phases, whereas in the class-incremental setup, task information is needed only during training. Another key difference is how the classifier heads are handled: in the class-incremental setup, classifier heads are appended progressively as new classes are introduced using training, while in the task-incremental setup, all classifier heads are initialized as a list at the beginning of training and selected based on the task identity for training and testing. We now proceed to discuss the baselines used for the different settings, marking each baseline with 'TI' for task-incremental and 'CI' for class-incremental, respectively. In all the cases, FedAvg McMahan et al. (2017) is used to adapt the centralized continual learning techniques to the federated setting.

- `Fine-FL` (TI, CI): A naive baseline where a client model trains on the current data.

- `EWC-FL` (Elastic weight consolidation in FL Kirkpatrick et al. (2017))(TI, CI): This is a Fisher information based regularization method implemented in a federated manner.
- `NCCL-FL` Han et al. (2023)(TI, CI): We implement the centralized NCCL technique in a federated manner.
- `Erg-FL` Chaudhry et al. (2019)(TI): Ering-FL is an experience-replay based continual learning method implemented in a federated manner.
- `FedTrk` Mitra et al. (2021)(TI, CI): FedTrack is an FL technique that uses IAG for local updates.

Table 2: Average accuracy and forgetting for non-IID setting on Split-CIFAR10, Split-CIFAR100, and Split-TinyImageNet (TinyIN) with 5 clients.

	CIFAR10		CIFAR100		TinyIN	
	Acc(↑)	Fgt(↓)	Acc(↑)	Fgt(↓)	Acc(↑)	Fgt(↓)
`Fine-FL`	54.10	6.08	38.66	20.87	23.72	23.85
`EWC-FL`	53.55	**5.22**	38.83	19.20	26.94	20.29
`NCCL-FL`	63.35	12.37	32.25	29.49	28.49	**8.73**
`Erg-FL`	**79.11**	8.32	31.25	32.40	21.90	19.67
`C-FLAG`	65.02	5.82	**43.47**	**16.76**	**28.63**	9.52

Table 3: Average accuracy and forgetting for IID setting on Split-CIFAR10, Split-CIFAR100, And Split-TinyImageNet (TinyIN) with 5 clients.

	CIFAR10		CIFAR100		TinyIN	
	Acc(↑)	Fgt(↓)	Acc(↑)	Fgt(↓)	Acc(↑)	Fgt(↓)
`Fine-FL`	72.64	26.51	49.82	30.00	30.17	31.91
`EWC-FL`	75.98	22.34	50.48	30.16	33.18	28.38
`NCCL-FL`	83.43	17.10	41.65	39.23	28.93	9.51
`FedTrk`	79.86	17.62	29.85	18.11	33.32	28.57
`Erg-FL`	84.01	16.50	38.70	43.66	31.25	32.40
`C-FLAG`	**89.28**	**7.23**	**66.85**	**7.30**	**44.30**	**7.65**

We also benchmark the performance of C-FLAG in the class-incremental setup by employing baselines such as LwF-FL (Learning without forgetting) Li and Hoiem (2017)(CI), iCARL-FL Rebuffi et al. (2017)(CI) and TARGET Zhang et al. (2023)(CI). However, due to lack of space, we defer the results of the class-incremental setup to the supplementary.

Metrics: The metrics used to evaluate the C-FLAG and the benchmarks are as follows:
1. Average accuracy (Acc): It is the average global model accuracy of all the tasks at the end of the final task in the CFL process.
2. Forgetting (Fgt): It is the average forgetting evaluated on the global model. For S tasks, the forgetting is defined as Forget $= \frac{1}{S-1}\sum_{i=1}^{S-1}(a_{i,i} - a_{S,i})$, where $a_{i,j}$ denotes the global model accuracy of task-j after training on the i-th task.

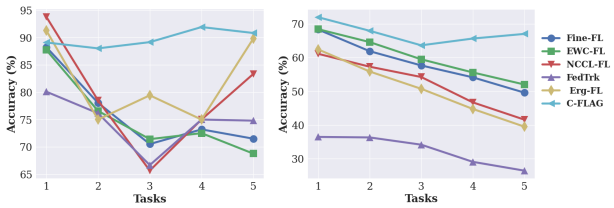

Figure 2: Average accuracy across tasks for IID splits of Split-CIFAR10 (Left) and Split-CIFAR100 (Right).

Comparison with Benchmarks: We perform experiments to benchmark the performance of the proposed C-FLAG approach as compared to the task-incremental baselines listed above. From Table 2 and Table 3, we observe that the proposed C-FLAG technique outperforms all the baselines for the Split-CIFAR10, Split-CIFAR100, and Split-TinyImageNet datasets with respect to accuracy. In particular, we see that our technique leads to a very low value of forgetting for all the datasets. All entries in Table 2 and Table 3 are averaged over 3 seeds, with standard deviation values provided in the supplementary material.

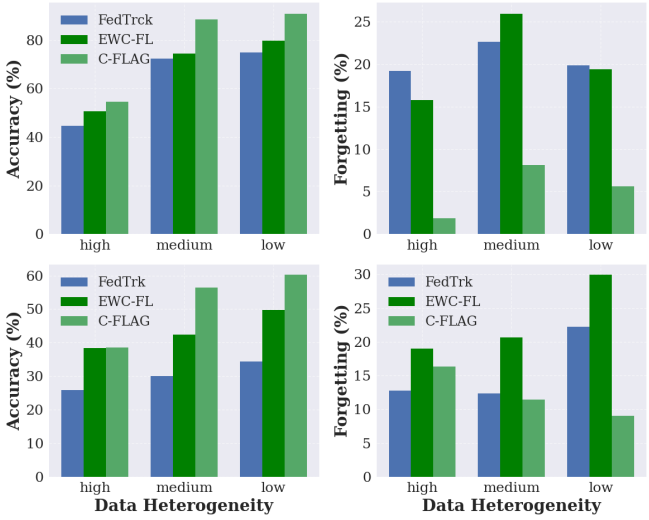

Figure 3: Varying heterogeneity for C-FLAG, EWC-FL and FedTrk techniques on Split-CIFAR10 (Top) and Split-CIFAR100 (Bottom).

In Fig. 2, we depict the average accuracy plots on Split-CIFAR10 and Split-CIFAR100 datasets for the proposed technique in the task-incremental setup as compared to the baselines. We observe that C-FLAG outperforms the baselines by demonstrating the highest average accuracy and lowest forgetting after each task.

Ablation Study: We perform ablations on data heterogeneity, the number of clients, and varying memory sampling sizes.

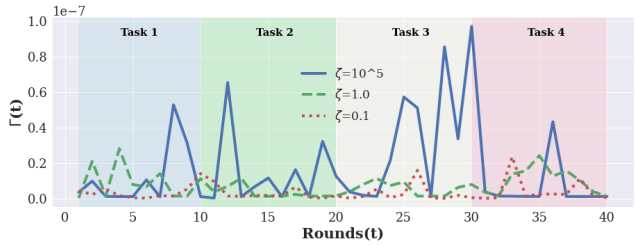

Figure 4: Evolution of $\Gamma(t)$ against progressing tasks on Split-CIFAR10 dataset for varying heterogeneity.

In Fig. 3, we illustrate the effectiveness of the proposed C-FLAG method across varying levels of data heterogeneity: low ($\zeta = 10^5$), medium ($\zeta = 1.0$) and high ($\zeta = 0.1$), controlled by the Dirichlet parameter ζ. The results clearly show that C-FLAG outperforms the EWC-FL and FedTrk baselines consistently across all levels of data heterogeneity. In Fig. 4, we also examine the evolution of the forgetting term $\Gamma(t)$ as a function of communication rounds as the tasks progress. An interesting observation is that $\Gamma(t)$ in the IID case exhibits a larger variation as compared to the non-IID scenarios. This can be attributed to the fact that, in the IID case, the data distribution among clients is more uniform, which enables the global model to exhibit greater plasticity, thereby providing more opportunities for learning new patterns. Consequently, at the end of each round, the global model tends to forget more while assimilating new knowledge. In contrast, in non-IID scenarios, clients experience data drift, which limits the model's adaptability. From Fig. 4, we also observe that whenever $\Gamma(t)$ shoots up, adaptive learning rates effectively help to mitigate this increase.

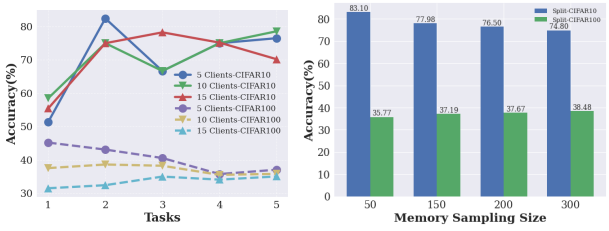

Figure 5: Varying clients (left) and varying memory sample size (right) for C-FLAG on non-IID partitions of Split-CIFAR10 and Split-CIFAR100 dataset.

The scalability of C-FLAG with varying clients can be observed in Fig. 5 (Left) for both, Split-CIFAR10 and Split-CIFAR100 datasets since the accuracy performance of C-FLAG remains consistent. We analyze the effect of varying sizes of the memory sample in C-FLAG, on non-IID partitions of Split-CIFAR10 and Split-CIFAR100 datasets using sampling sizes of 50, 150, 200, and 300 per task. Note that the initial buffer size is fixed at 400 for this experiment. We observe from Fig. 5 (Right), that for Split-CIFAR10, accuracy decreases as the sampling size increases, whereas in Split-CIFAR100, accuracy improves with larger sampling sizes. Split-CIFAR10 and Split-CIFAR100 have 2 and 20 classes per task respectively. As memory size increases, the model has to accommodate constraints from a larger memory data. Unlike CIFAR10, in CIFAR100, the model has enough latent representational space to accommodate the constraints due to the higher number of classes per task.

8 CONCLUSIONS AND FUTURE WORK

We proposed the novel C-FLAG algorithm, which is a replay-memory based federated strategy that combines the edge-based gradient updates on memory and aggregated gradients on the current data. We also presented the convergence analysis of C-FLAG. In C-FLAG, the server initiates the transition between tasks at clients, while the previous task data is selectively sampled into a memory buffer for each client. The global parameter update of C-FLAG is an aggregation of a single step on memory and a delayed gradient-based local step on the current data at each client. We demonstrate that the convergence on the previous task is largely dependent on the suppression of the catastrophic forgetting term. We extend the analysis further by optimizing the catastrophic forgetting term and derive adaptive learning rates that ensure seamless continual learning. Empirically, we show that C-FLAG outperforms the baselines w.r.t. metrics such as accuracy and forgetting in a task-incremental setup. The limitations of C-FLAG include the need to store data in the memory. However, since each client maintains its memory buffer at the edge, privacy constraints are not violated. C-FLAG requires an additional communication for obtaining the average memory gradients $\nabla f^{\dagger}(\mathbf{x})$ and the average gradients of the clients on current data $\nabla g(\mathbf{x})$ at the beginning of each server round.

Acknowledgments

We acknowledge the financial support provided by the iHub-Anubhuti-IIITD Foundation, set up under the NM-ICPS scheme of the Department of Science and Technology, Government of India. We also acknowledge the Prime Minister's Fellowship for Doctoral Research (ANRF-FICCI) from ANRF and LightMetrics, Bengaluru, India, and the DRDO CARS project for their generous funding and support.

References

Ahmad Ajalloeian and Sebastian U Stich. On the convergence of sgd with biased gradients. *arXiv preprint arXiv:2008.00051*, 2020.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Yahya H Ezzeldin, and Salman Avestimehr. Federated orthogonal training: Mitigating global catastrophic forgetting in continual federated learning. In *The Twelfth International Conference on Learning Representations*, 2023.

Doron Blatt, Alfred O Hero, and Hillel Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, 2007.

Antonio Carta, Lorenzo Pellegrini, Andrea Cossu, Hamed Hemati, and Vincenzo Lomonaco. Avalanche: A pytorch library for deep continual learning. *Journal of Machine Learning Research*, 24(363): 1–6, 2023. URL `http://jmlr.org/papers/v24/23-0130.html`.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet Kumar Dokania, Philip HS Torr, and Marcaurelio Ranzato. On tiny episodic memories in continual learning. arxiv. *Learning*, 6(7), 2019.

Aristotelis Chrysakis and Marie-Francine Moens. Online bias correction for task-free continual learning. In *International Conference on Learning Representations*, 2023. URL `https://api.semanticscholar.org/CorpusID:259298434`.

Jiahua Dong, Hongliu Li, Yang Cong, Gan Sun, Yulun Zhang, and Luc Van Gool. No one left behind: Real-world federated class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

Keval Doshi and Yasin Yilmaz. Continual learning for anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 254–255, 2020.

Christophe Dupuy, Jimit Majmudar, Jixuan Wang, Tanya G Roosta, Rahul Gupta, Clement Chung, Jie Ding, and Salman Avestimehr. Quantifying catastrophic forgetting in continual federated learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4): 128–135, 1999.

Zacharias Georgiou, Moysis Symeonides, Demetris Trihinas, George Pallis, and Marios D Dikaiakos. Streamsight: A query-driven framework for streaming analytics in edge computing. In *2018 IEEE/ACM 11th International Conference on Utility and Cloud Computing (UCC)*, pages 143–152. IEEE, 2018.

Jack Good, Jimit Majmudar, Christophe Dupuy, Jixuan Wang, Charith Peris, Clement Chung, Richard Zemel, and Rahul Gupta. Coordinated replay sample selection for continual federated learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 331–342, 2023.

Seungyub Han, Yeongmo Kim, Taehyun Cho, and Jungwoo Lee. On the convergence of continual learning with adaptive methods. In *Uncertainty in Artificial Intelligence*, pages 809–818. PMLR, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL `https://arxiv.org/abs/1512.03385`.

Sean M Hendryx, Dharma Raj KC, Bradley Walls, and Clayton T Morrison. Federated reconnaissance: Efficient, distributed, class-incremental learning. *arXiv preprint arXiv:2109.00150*, 2021.

Erik Johannes Husom, Sagar Sen, Arda Goknil, Simeon Tverdal, and Phu H Nguyen. Reptile: a tool for replay-driven continual learning in iiot. In *Proceedings of the 13th International Conference on the Internet of Things*, pages 204–207, 2023.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015. URL `https://api.semanticscholar.org/CorpusID:16664790`.

Yichen Li, Qunwei Li, Haozhao Wang, Ruixuan Li, Wenliang Zhong, and Guannan Zhang. Towards efficient replay in federated incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12820–12829, 2024.

Zhizhong Li and Derek Hoiem. Learning without forgetting. 2017.

Yan-Shuo Liang and Wu-Jun Li. Adaptive plasticity improvement for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7816–7825, 2023.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

Aritra Mitra, Rayana Jaafar, George J Pappas, and Hamed Hassani. Federated learning with incrementally aggregated gradients. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 775–782. IEEE, 2021.

Patient Ntumba, Nikolaos Georgantas, and Vassilis Christophides. Efficient scheduling of streaming operators for iot edge analytics. In *2021 Sixth International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 1–8, 2021. doi: 10.1109/FMEC54266.2021.9732409.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Paper.pdf`.

Sagar Sen, Simon Myklebust Nielsen, Erik Johannes Husom, Arda Goknil, Simeon Tverdal, and Leonardo Sastoque Pinilla. Replay-driven continual learning for the industrial internet of things. In *2023 IEEE/ACM 2nd International Conference on AI Engineering–Software Engineering for AI (CAIN)*, pages 43–55. IEEE, 2023.

Khadija Shaheen, Muhammad Abdullah Hanif, Osman Hasan, and Muhammad Shafique. Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks. *Journal of Intelligent & Robotic Systems*, 105(1):9, 2022.

Donald Shenaj, Marco Toldo, Alberto Rigon, and Pietro Zanuttigh. Asynchronous federated continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5055–5063, 2023.

Qazi Mazhar ul Haq, Shanq-Jang Ruan, Muhammad Amirul Haq, Said Karam, Jeng Lun Shieh, Peter Chondro, and De-Qin Gao. An incremental learning of yolov3 without catastrophic forgetting for smart city applications. *IEEE Consumer Electronics Magazine*, 11(5):56–63, 2021.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024. doi: 10.1109/TPAMI.2024.3367329.

Xin Yang, Hao Yu, Xin Gao, Hao Wang, Junbo Zhang, and Tianrui Li. Federated continual learning via knowledge fusion: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *International Conference on Machine Learning*, pages 12073–12086. PMLR, 2021.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

Jie Zhang, Chen Chen, Weiming Zhuang, and Lingjuan Lyu. Target: Federated class-continual learning via exemplar-free distillation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4759–4770, 2023. doi: 10.1109/ICCV51070.2023.00441.

Giulio Zizzo, Ambrish Rawat, Naoise Holohan, and Seshu Tirupathi. Federated continual learning with differentially private data sharing. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes, in Supplementary]
   (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Material for "On the Convergence of Continual Federated Learning Using Incrementally Aggregated Gradients"

1 PROOFS AND EXTENDED DISCUSSIONS

Table 4: This table provides a summary of the key notations.

Notation	Description
$\mathcal{P}^i$	Data from all the previous tasks at the i-th client
$\mathcal{M}^i$	Memory buffer at the i-th client
$\mathcal{C}^i$	Current dataset at the i-th client
$h_i(\mathbf{x})$	Local loss function at the i-th client
$f_i(\mathbf{x})$	Restriction of $h_i(\mathbf{x})$ on $\mathcal{P}^i$
$g_i(\mathbf{x})$	Restriction of $h_i(\mathbf{x})$ on $\mathcal{C}^i$
$\mathbf{x}_{t,k}^i$	Model weight at the i-th client after t-th communication round and at the end of the k-th local epoch
$\mathbf{x}_t$	Server model weight at the end of the t-th communication round
$\nabla g_i'(\mathbf{x}_{t,k}^i)$	Delayed gradient at the i-th client after t-th communication round and k-th local epoch
$\tau_{k,j}^i$	Index for the most recent local step at which gradient was computed for the k-th datapoint at the i-th client
$b_i(\mathbf{x}_t)$	Bias at the i-th client
$\nabla f^\dagger(\mathbf{x}_t)$	Gradient on the memory data
$\Gamma(t)$	Global catastrophic forgetting term
$h_{\lvert_{\mathcal{M}\cup\mathcal{C}}}(\mathbf{x}_t)$	Restriction of $h(\mathbf{x}_t)$ on $\mathcal{M}\cup\mathcal{C}$

In this section, we present the theoretical convergence analysis of C-FLAG, the proposed memory-based continual learning framework in a non-convex setting, along with the detailed proofs.

First, we provide derivations of some results which will be used in the sequel to complete the proofs of the main theorems and lemmas. Notations-wise, we use the same notations as in the main manuscript.

Client drift after E local epochs: At the beginning of the t-th global iteration, each client $i \in [N]$, receives current global model weight $\mathbf{x}_t$. Each client updates its weights for E epochs to obtain $\mathbf{x}_{t,E}^i$ resulting in a client drift from the current global model given as:

$$\mathbf{x}_{t,E}^i - \mathbf{x}_t = -\beta_t \sum_{k=0}^{E-1} \Big(\nabla g(\mathbf{x}_t) - \nabla g_i(\mathbf{x}_t) + \nabla g_i'(\mathbf{x}_{t,k}^i) \Big) \tag{17}$$

$$= -\beta_t \sum_{k=0}^{E-1} \Big(\nabla g(\mathbf{x}_t) - \nabla g_i(\mathbf{x}_t) + \nabla g_i'(\mathbf{x}_{t,k}^i) \Big) \tag{18}$$

$$= -\beta_t \sum_{k=0}^{E-1} \Big(\nabla g(\mathbf{x}_t) - \nabla g_i(\mathbf{x}_t) \Big) - \beta_t \sum_{k=0}^{E-1} \nabla g_i'(\mathbf{x}_{t,k}^i) \tag{19}$$

$$= -\beta_t E \Big(\nabla g(\mathbf{x}_t) - \nabla g_i(\mathbf{x}_t) \Big) - \beta_t \sum_{k=0}^{E-1} \nabla g_i'(\mathbf{x}_{t,k}^i). \tag{20}$$

Summing up the weighted drifts across clients, we get

$$\sum_{i=1}^{N} p_i(\mathbf{x}_{t,E}^i - \mathbf{x}_t) = -\beta_t E \underbrace{\sum_{i=1}^{N} p_i \Big(\nabla g(\mathbf{x}_t) - \nabla g_i(\mathbf{x}_t) \Big)}_{=0} - \beta_t \sum_{i=1}^{N} \sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \tag{21}$$

$$= -\beta_t \sum_{i=1}^{N} \sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i). \tag{22}$$

Hence, we see that the average client drift is a function of the aggregated delayed gradients at each client.

Server Update Rule: At the end of the t-th global iteration, i-th client sends $(\mathbf{x}_{t,E}^i - \alpha_t \nabla f^\dagger(\mathbf{x}_t^i))$ to the server and server aggregates its model weights to get $\mathbf{x}_{t+1}$ as follows:

$$\mathbf{x}_{t+1} = \sum_{i=1}^{N} p_i(\mathbf{x}_{t,E}^i - \alpha_t \nabla f^\dagger(\mathbf{x}_t^i)) \tag{23}$$

$$\overset{(a)}{=} \mathbf{x}_t - \alpha_t \nabla f^\dagger(\mathbf{x}_t) - \beta_t \sum_{i=1}^{N} \sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \tag{24}$$

$$= \mathbf{x}_t - \alpha_t \sum_{i=1}^{N} p_i \nabla f_i^\dagger(\mathbf{x}_t) - \beta_t \sum_{i=1}^{N} \sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \tag{25}$$

where (a) follows from (22).

1.1 Essential Lemmas and Proofs

For purposes of clarity, we restate the assumptions as follows:

Assumption 4. *(L-smoothness). For all $i \in [N]$, f_i, g_i, h_i are L-smooth.*

Assumption 5. *(Bounded Bias). There exists constants $0 \leq m_i < 1$ for all $\mathbf{x} \in \mathbb{R}^d$ such that $\|b_i(\mathbf{x})\|^2 \leq m_i \|\nabla f_i(\mathbf{x})\|^2, \quad \forall i \in [N]$.*

Assumption 6. *(Bounded memory gradient). There exists $r_i \in \mathbb{R}^+$, such that $\|\nabla f_i^\dagger(\mathbf{x}_t)\| \leq r_i \|\nabla g(\mathbf{x}_t)\|$ for all $i \in [N]$.*

Additionally, we note that $g_i(\mathbf{x})$ is defined as a sum of it component functions as $g_i(\mathbf{x}) = \frac{1}{|\mathcal{C}^i|} \sum_{j \in \mathcal{C}^i} g_{i,j}(\mathbf{x})$, where each of the $g_{i,j}$s are also L-smooth.

In the following lemmas, we show that the expectation of the total bias term across clients, $b(\mathbf{x}) = \sum_{i=1}^{N} p_i b_i(\mathbf{x})$, is zero. We also provide an upper bound on the squared norm of $b(\mathbf{x})$.

Lemma 8. *If $\mathcal{M}_0^i$ is uniformly sampled from $\mathcal{P}^i$ at each client $i \in [N]$, then $\mathbb{E}_{\mathcal{M}_t}[b(\mathbf{x})] = \sum_{i=1}^{N} p_i \mathbb{E}_{\mathcal{M}_t^i}[b_i(\mathbf{x})] = 0 \; \forall \mathbf{x} \in \mathbb{R}^d$.*

Proof. From the definition of the bias function in the main manuscript 6, we have $b_i(\mathbf{x}) = \nabla f_i^\dagger(\mathbf{x}) - \nabla f_i(\mathbf{x})$, where the biased gradient is computed on the memory data $\mathcal{M}_t^i$ at t-th time-step for the i-th client. In the episodic memory (ring buffer) scheme, at each client $i \in [N]$ we construct the memory data as $\mathcal{M}_t^i = \mathcal{M}_0^i \; \forall t \in \{0, 1, \cdots, T-1\}$ and the initial memory $\mathcal{M}_0^i$ is uniformly chosen from the IID data stream of the past task dataset $\mathcal{P}_i$. Using this, we obtain

$$\mathbb{E}_{\mathcal{M}_t^i}[\nabla f_i^\dagger(\mathbf{x})] = \mathbb{E}_{\mathcal{M}_0^i}[\nabla f_i^\dagger(\mathbf{x})] = \nabla f_i(\mathbf{x}), \tag{26}$$

which leads to $\mathbb{E}_{\mathcal{M}_t^i}[b_i(\mathbf{x})] = 0 \; \forall \mathbf{x} \in \mathbb{R}^d$. Hence, $\mathbb{E}_{\mathcal{M}_t}[b(\mathbf{x})] = \sum_{i=1}^{N} p_i \mathbb{E}_{\mathcal{M}_t^i}[b_i(\mathbf{x})] = 0$. □

Lemma 9. *Given assumption 2 holds, there exists a $m \in \mathbb{R}^+$, $\forall \mathbf{x} \in \mathbb{R}^d$ such that*

$$\|b(\mathbf{x})\|^2 = \left\|\sum_{i=1}^{N} p_i b_i(\mathbf{x})\right\|^2 \leq m \|\nabla f(\mathbf{x})\|^2 . \tag{27}$$

Proof. From the assumption 2, we have $\|b_i(x)\|^2 \leq m_i \|\nabla f_i(\mathbf{x})\|^2$. We assume there exists some $m \in \mathbb{R}^+$ such that

$$m_i \|\nabla f_i(\mathbf{x})\|^2 \leq m \|\nabla f(\mathbf{x})\|^2 \ \forall i \in [N]. \tag{28}$$

Using this we obtain the following:

$$\|b(\mathbf{x})\|^2 = \left\|\sum_{i=1}^{N} p_i b_i(\mathbf{x})\right\|^2 \overset{(a)}{\leq} \sum_{i=1}^{N} p_i \|b_i(\mathbf{x})\|^2 \overset{(b)}{\leq} \sum_{i=1}^{N} p_i m \|\nabla f(\mathbf{x})\|^2 = m \|\nabla f(\mathbf{x})\|^2 . \tag{29}$$

where (a) follows from the Jensen's inequality and (b) follows from (28). □

Lemma 10. *Suppose that the bounded memory gradient assumption 3 holds. Then there exists a constant $r \in \mathbb{R}^+$, such that $\left\|\nabla f^\dagger(\mathbf{x}_t)\right\| \leq r \|\nabla g(\mathbf{x}_t)\|$ holds.*

Proof. Using the triangle inequality, the bounded memory gradient assumption, and choosing $r = max\{r_i | i \in [N]\}$, we obtain

$$\left\|\nabla f^\dagger(\mathbf{x}_t)\right\| = \left\|\sum_{i=1}^{N} p_i \nabla f_i^\dagger(\mathbf{x}_t)\right\| \leq \sum_{i=1}^{N} p_i \left\|\nabla f_i^\dagger(\mathbf{x}_t)\right\| \leq \sum_{i=1}^{N} p_i r_i \|\nabla g(\mathbf{x}_t)\| \leq r \|\nabla g(\mathbf{x}_t)\| \tag{30}$$

□

1.2 Proofs of the Lemmas and Theorems

Since we use delayed gradients which is an estimate of the local gradient on the current task's dataset there is an associated gradient estimation error, $\mathbf{e}_{t,k}^i$, at each local epoch k given by

$$\mathbf{e}_{t,k}^i = \nabla g_i'(\mathbf{x}_{t,k}^i) - \nabla g_i(\mathbf{x}_{t,k}^i). \tag{31}$$

Using the above, we rephrase (4) with the additional gradient error term as

$$\mathbf{x}_{t,k+1}^i = \mathbf{x}_{t,k}^i - \beta_t \left(\nabla g(\mathbf{x}_t) - \nabla g_i(\mathbf{x}_t) + \nabla g_i(\mathbf{x}_{t,k}^i) + \mathbf{e}_{t,k}^i\right). \tag{32}$$

In this section, we provide proof of the lemmas and theorems stated in the main manuscript. First, we start with the Lem. 1 followed by Theorem 2.

Lemma 1: Suppose that the assumptions 1, 2 hold and $\alpha_t < \frac{2}{L(1+m)}$. For the sequence $\{\mathbf{x}_t\}_{t=1}^T$ generated by the algorithm 1 , we have

$$\|\nabla f(\mathbf{x}_t)\|^2 \leq \frac{1}{\alpha_t[1 - \frac{L}{2}\alpha_t(1+m)]}\big(f(\mathbf{x}_t) - f(\mathbf{x}_{t+1}) + B(t) + \Gamma(t)\big), \tag{33}$$

where $B(t)$ is the overfitting term defined as

$$B(t) = (L\alpha_t^2 - \alpha_t)\langle \nabla f(\mathbf{x}_t), b(\mathbf{x}_t)\rangle + \beta_t \langle b(\mathbf{x}_t), \textstyle\sum_{i=1}^{N}\sum_{k=1}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i)\rangle. \tag{34}$$

Further, $\Gamma(t)$ is the forgetting term defined as

$$\Gamma(t) = \frac{L}{2}\beta_t^2 \|\textstyle\sum_{i=1}^{N}\sum_{k=1}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i)\|^2 - \beta_t(1 - L\alpha_t)\langle \nabla f^\dagger(\mathbf{x}_t), \textstyle\sum_{i=1}^{N}\sum_{k=1}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i)\rangle. \tag{35}$$

Proof. We start our analysis using L- smoothness of $f(\cdot)$ and the server update rule (25). By L-smoothness of $f(\cdot)$, we have

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \leq \underbrace{\langle \nabla f(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle}_{A} + \underbrace{\frac{L}{2} \left\| \mathbf{x}_{t+1} - \mathbf{x}_t \right\|^2}_{B}. \tag{36}$$

We separately upper bound the terms A and B. Using the update rule given in (24) and separating the terms involving gradient over the memory and current data, we obtain

$$\begin{aligned} A &= \langle \nabla f(\mathbf{x}_t), \mathbf{x}_{t+1} - \mathbf{x}_t \rangle & (37)\\ &= \langle \nabla f(\mathbf{x}_t), -\alpha_t \sum_{i=1}^{N} p_i \nabla f_i^{\dagger}(\mathbf{x}_t) - \beta_t \sum_{i=1}^{N} \sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \rangle & (38)\\ &= \underbrace{-\alpha_t \left\langle \nabla f(\mathbf{x}_t), \sum_{i=1}^{N} p_i \nabla f_i^{\dagger}(\mathbf{x}_t) \right\rangle}_{AI} \underbrace{-\beta_t \left\langle \nabla f(\mathbf{x}_t), \sum_{i=1}^{N} \sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \right\rangle}_{AII}. & (39) \end{aligned}$$

In the above, the first term (AI) measures the effect of the previous task's memory data gradients, and the second term (AII) measures the effect of the current task's delayed gradients. Considering AI and using the biased gradient $\nabla f_i^{\dagger}(\mathbf{x}) = \nabla f_i(\mathbf{x}) + b_i(\mathbf{x})$ we have the following:

$$\begin{aligned} AI &= -\alpha_t \left\langle \nabla f(\mathbf{x}_t), \sum_{i=1}^{N} p_i \nabla f_i^{\dagger}(\mathbf{x}_t) \right\rangle & (40)\\ &= -\alpha_t \left\langle \nabla f(\mathbf{x}_t), \sum_{i=1}^{N} p_i (\nabla f_i(\mathbf{x}_t) + b_i(\mathbf{x}_t)) \right\rangle & (41)\\ &= -\alpha_t \langle \nabla f(\mathbf{x}_t), \nabla f(\mathbf{x}_t) \rangle - \alpha_t \langle \nabla f(\mathbf{x}_t), b(\mathbf{x}_t) \rangle & (42)\\ &= -\alpha_t \left\| \nabla f(\mathbf{x}_t) \right\|^2 - \alpha_t \langle \nabla f(\mathbf{x}_t), b(\mathbf{x}_t) \rangle. & (43) \end{aligned}$$

Next, we simplify the term AII as follows:

$$\begin{aligned} AII &= -\beta_t \left\langle \nabla f(\mathbf{x}_t), \sum_{i=1}^{N} \sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \right\rangle & (44)\\ &= -\beta_t \left\langle \nabla f^{\dagger}(\mathbf{x}_t) - b(\mathbf{x}_t), \sum_{i=1}^{N} \sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \right\rangle & (45)\\ &= -\beta_t \left\langle \nabla f^{\dagger}(\mathbf{x}_t), \sum_{i=1}^{N} \sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \right\rangle + \beta_t \left\langle b(\mathbf{x}_t), \sum_{i=1}^{N} \sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \right\rangle & (46) \end{aligned}$$

Next we consider the term B, which represents the difference between two consecutive global model weights generated through our proposed algorithm 1. Using (25), we have the following:

$$\begin{aligned} B &= \frac{L}{2} \left\| \mathbf{x}_{t+1} - \mathbf{x}_t \right\|^2 & (47)\\ &= \frac{L}{2} \left\| \alpha_t \sum_{i=1}^{N} p_i \nabla f_i^{\dagger}(\mathbf{x}_t) + \beta_t \sum_{i=1}^{N} \sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \right\| & (48)\\ &= \underbrace{\frac{L}{2} \alpha_t^2 \left\| \sum_{i=1}^{N} p_i \nabla f_i^{\dagger}(\mathbf{x}_t) \right\|^2}_{BI} + \underbrace{\frac{L}{2} \beta_t^2 \left\| \sum_{i=1}^{N} \sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \right\|^2}_{BII}\\ &\quad + \underbrace{L \alpha_t \beta_t \left\langle \sum_{i=1}^{N} p_i \nabla f_i^{\dagger}(\mathbf{x}_t), \sum_{i=1}^{N} \sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \right\rangle}_{BIII} & (49) \end{aligned}$$

Similar to our previous approach, next we will bound each of the terms in B separately. Using the definitions of biased gradients, $\nabla f(\mathbf{x}_t)$ and $b(\mathbf{x}_t)$, we obtain

$$BI = \frac{L}{2}\alpha_t^2 \left\| \sum_{i=1}^{N} p_i \nabla f_i^{\dagger}(\mathbf{x}_t) \right\|^2 \tag{50}$$

$$= \frac{L}{2}\alpha_t^2 \left\| \sum_{i=1}^{N} p_i (\nabla f_i(\mathbf{x}_t) + b_i(\mathbf{x}_t)) \right\|^2 \tag{51}$$

$$= \frac{L}{2}\alpha_t^2 \left\| \nabla f(\mathbf{x}_t) + b(\mathbf{x}_t) \right\|^2 \tag{52}$$

$$= \frac{L}{2}\alpha_t^2 \left\| \nabla f(\mathbf{x}_t) \right\|^2 + \frac{L}{2}\alpha_t^2 \left\| b(\mathbf{x}_t) \right\|^2 + L\alpha_t^2 \langle \nabla f(\mathbf{x}_t), b(\mathbf{x}_t) \rangle. \tag{53}$$

We keep the term BII as it is. Finally, we simplify $BIII$ as

$$BIII = L\alpha_t\beta_t \left\langle \sum_{i=1}^{N} p_i \nabla f_i^{\dagger}(\mathbf{x}_t), \sum_{i=1}^{N}\sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \right\rangle = \sum_{k=0}^{E-1} L\alpha_t\beta_t \left\langle \nabla f^{\dagger}(\mathbf{x}_t), \sum_{i=1}^{N} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \right\rangle \tag{54}$$

Finally putting all the resulting terms in (36), we obtain the following:

$$\begin{aligned} f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \leq & -\alpha_t \left\| \nabla f(\mathbf{x}_t) \right\|^2 - \alpha_t \langle \nabla f(\mathbf{x}_t), b(\mathbf{x}_t) \rangle - \beta_t \left\langle \nabla f^{\dagger}(\mathbf{x}_t), \sum_{i=1}^{N}\sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \right\rangle \\ & + \beta_t \left\langle b(\mathbf{x}_t), \sum_{i=1}^{N}\sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \right\rangle + \frac{L}{2}\alpha_t^2 \left\| \nabla f(\mathbf{x}_t) \right\|^2 + \frac{L}{2}\alpha_t^2 \left\| b(\mathbf{x}_t) \right\|^2 \\ & + L\alpha_t^2 \langle \nabla f(\mathbf{x}_t), b(\mathbf{x}_t) \rangle + \frac{L}{2}\beta_t^2 \left\| \sum_{i=1}^{N}\sum_{k=0}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \right\|^2 \\ & + \sum_{k=0}^{E-1} L\alpha_t\beta_t \left\langle \nabla f^{\dagger}(\mathbf{x}_t), \sum_{i=1}^{N} p_i (\nabla g_i'(\mathbf{x}_{t,k}^i) \right\rangle \\ \leq & -\alpha_t (1 - \frac{L}{2}\alpha_t) \left\| \nabla f(\mathbf{x}_t) \right\|^2 + B(t) + \Gamma(t) + \frac{L}{2}\alpha_t^2 m \left\| \nabla f(\mathbf{x}_t)) \right\|^2 \text{ (using Lem. 9)} \end{aligned} \tag{55}$$

$$= -\alpha_t \big[1 - \frac{L}{2}\alpha_t(1+m)\big] \left\| \nabla f(\mathbf{x}_t) \right\|^2 + B(t) + \Gamma(t), \tag{56}$$

where the overfitting term $B(t)$ is given as

$$B(t) = (L\alpha_t^2 - \alpha_t)\langle \nabla f(\mathbf{x}_t), b(\mathbf{x}_t) \rangle + \beta_t \langle b(\mathbf{x}_t), \textstyle\sum_{i=1}^{N}\sum_{k=1}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \rangle, \tag{57}$$

and the forgetting term $\Gamma(t)$ is given as

$$\Gamma(t) = \frac{L}{2}\beta_t^2 \| \textstyle\sum_{i=1}^{N}\sum_{k=1}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \|^2 - \beta_t(1 - L\alpha_t)\langle \nabla f^{\dagger}(\mathbf{x}_t), \sum_{i=1}^{N}\sum_{k=1}^{E-1} p_i \nabla g_i'(\mathbf{x}_{t,k}^i) \rangle. \tag{58}$$

Using $\alpha_t < \frac{2}{L(1+m)}$ and re-arranging the terms, we obtain

$$\left\| \nabla f(\mathbf{x}_t) \right\|^2 \leq \frac{1}{\alpha_t[1 - \frac{L}{2}\alpha_t(1+m)]} \left(f(\mathbf{x}_t) - f(\mathbf{x}_{t+1}) + B(t) + \Gamma(t) \right). \tag{59}$$

Theorem 2: Suppose that the assumptions 1, 2 hold. Given $F = f(\mathbf{x}_0) - f(\mathbf{x}_T)$, the sequence $\{\mathbf{x}_t\}_{t=1}^{T}$ generated by algorithm 1 with $\alpha_t = \alpha = \frac{1}{L(m+1)}$ $\forall t \in \{0, 1, \cdots, T-1\}$, and $m \in \mathbb{R}^+$, satisfies

$$\min_t \mathbb{E}\left[\left\| \nabla f(\mathbf{x}_t) \right\|^2 \right] \leq \frac{2L(1+m)}{T} \left(F + \sum_{t=0}^{T-1} \mathbb{E}[\Gamma(t)] \right).$$

Proof. Taking expectation with respect to the choice of the memory $\mathcal{M}_t$ on both sides of (59), we obtain

$$\mathbb{E}_{\mathcal{M}_t}\Big[\|\nabla f(\mathbf{x}_t)\|^2\Big] \leq \mathbb{E}_{\mathcal{M}_t}\Big[\frac{1}{\alpha_t[1-\frac{L}{2}\alpha_t(1+m)]}\left(f(\mathbf{x}_t)-f(\mathbf{x}_{t+1})+B(t)+\Gamma(t)\right)\Big]. \tag{60}$$

Given the learning rate $\alpha_t = \alpha$ for all $t \in \{0, 1, \cdots, T-1\}$ and taking average over T iterations ($t = 0$ to $T-1$), of the above, we obtain

$$\min_t \mathbb{E}\Big[\|\nabla f(\mathbf{x}_t)\|^2\Big] \leq \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\Big[\|\nabla f(\mathbf{x}_t)\|^2\Big] \tag{61}$$

$$\leq \frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{\alpha[1-\frac{L}{2}\alpha(1+m)]}\Big(f(\mathbf{x}_t)-f(\mathbf{x}_{t+1})+\mathbb{E}[B(t)+\Gamma(t)]\Big) \tag{62}$$

$$\leq \frac{1}{T\alpha[1-\frac{L}{2}\alpha(1+m)]}\Big(f(\mathbf{x}_0)-f(\mathbf{x}_T)+\sum_{t=0}^{T-1}\mathbb{E}[B(t)+\Gamma(t)]\Big) \tag{63}$$

$$\overset{(a)}{=} \frac{1}{T\alpha[1-\frac{L}{2}\alpha(1+m)]}\Big(F+\sum_{t=0}^{T-1}\mathbb{E}[\Gamma(t)]\Big) \tag{64}$$

$$\overset{(b)}{=} \frac{2L(1+m)}{T}\big(F+\sum_{t=0}^{T-1}\mathbb{E}[\Gamma(t)]\big), \tag{65}$$

where (a) follows from Lem. 8 and (b) follows using the step size $\alpha = \frac{1}{L(m+1)}$.

Next, we provide the proof for Lem. 3 as present in the main manuscript.

Lemma 3: Given assumption 1, the sequence $\{\mathbf{x}_t\}_{t=1}^T$ generated with learning rates $\alpha_t = \beta_t = \alpha = \frac{1}{30LE}$, for the restriction of $h(\cdot)$ on $\mathcal{M}\cup\mathcal{C}$ given as $\tilde{h}(\mathbf{x}_t) = h_{|_{\mathcal{M}\cup\mathcal{C}}}(\mathbf{x}_t)$ we have

$$\min_t \|\nabla\tilde{h}(\mathbf{x}_t)\|^2 \leq \tfrac{60L}{T}\left(\tilde{h}(\mathbf{x}_0)-\tilde{h}(\mathbf{x}_T)\right). \tag{66}$$

Proof. We use a common learning rate $\beta_t = \alpha_t \forall t \in \{0, 1, \cdots, T-1\}$. Using the update rules (22) and (25) to update the global objective function on the restriction $\mathcal{M}\cup\mathcal{C}$, denoted as $\tilde{h}(\cdot)$, the update rule reduces into the following local and server update rules:

$$\text{Local: } \mathbf{x}^i_{t,k+1} = \mathbf{x}^i_{t,k} - \beta_t\Big(\nabla\tilde{h}(\mathbf{x}_t) - \nabla\tilde{h}_i(\mathbf{x}_t) + \nabla\tilde{h}'_i(\mathbf{x}^i_{t,k})\Big), \tag{67}$$

$$\text{Server: } \mathbf{x}_{t+1} = \mathbf{x}_t - \beta_t\sum_{i=1}^N p_i\nabla\tilde{h}(\mathbf{x}_t) - \beta_t\sum_{i=1}^N\sum_{k=0}^{E-1} p_i\nabla\tilde{h}'_i(\mathbf{x}^i_{t,k}), \tag{68}$$

where $\nabla\tilde{h}'_i(\mathbf{x}^i_{t,k})$ is the delayed gradient on $\mathcal{M}\cup\mathcal{C}$. Then we note that the problem is similar to Lem. 1 and can be proved using the L-smoothness of $h(.)$. To prove convergence we follow the same steps as in Theorem 2 of Mitra et al. (2021) and obtain

$$\tilde{h}(\mathbf{x}_{t+1}) - \tilde{h}(\mathbf{x}_t) \leq (-\beta_t + 10\beta_t^2LE^2 + 80\beta_t^4L^3E^4 - \beta_tE)\left\|\tilde{h}(\mathbf{x}_t)\right\|^2 \tag{69}$$

$$< (-\beta_tE + 10\beta_t^2LE^2 + 80\beta_t^4L^3E^4)\left\|\tilde{h}(\mathbf{x}_t)\right\|^2 \tag{70}$$

$$\overset{(a)}{\leq} -\beta_tE(1-15\beta_tLE)\left\|\tilde{h}(\mathbf{x}_t)\right\|^2, \tag{71}$$

where (a) follows from using $\beta_t \leq \frac{1}{4LE}$. Rearranging the above equation and using $\beta_t = \frac{1}{30LE}$, we obtain

$$\left\|\tilde{h}(\mathbf{x}_t)\right\|^2 \leq \frac{1}{\beta_tE(1-15\beta_tLE)}\big(\tilde{h}(\mathbf{x}_t)-\tilde{h}(\mathbf{x}_{t+1})\big) \tag{72}$$

$$= 60L\big(\tilde{h}(\mathbf{x}_t)-\tilde{h}(\mathbf{x}_{t+1})\big). \tag{73}$$

Finally, using telescopic summing, we obtain the desired result as

$$\min_t \left\|\nabla \tilde{h}(\mathbf{x}_t)\right\|^2 \leq \frac{60L}{T} \sum_{t=0}^{T-1} \big(\tilde{h}(\mathbf{x}_0) - \tilde{h}(\mathbf{x}_T)\big). \tag{74}$$

Hence, our proposed algorithm provides convergence guarantees if it jointly learns from the current task and the replay-memory at each client. Next, we provide an upper bound on the squared norm of $\nabla g(\mathbf{x}_t)$.

We define $\mathcal{D} = \{\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^N\}$ as the collection of subsets of both the current task and memory data where each $\mathcal{D}^i$ satisfies $\mathcal{C}^i \subset \mathcal{D}^i \subset \mathcal{M}^i \cup \mathcal{C}^i$. Additionally, $h_{i_{|_{\mathcal{D}^i}}}$ is defined as the restriction of h_i on $\mathcal{D}^i$.

Lemma 11. *Given the sequence $\{\mathbf{x}_t\}_{t=1}^T$ as generated by algorithm 1, the upper bound for the global loss gradient for the current data $\mathcal{C} = \{\mathcal{C}^1, \mathcal{C}^2, \ldots, \mathcal{C}^N\}$ satisfies*

$$\|\nabla g(\mathbf{x}_t)\|^2 = \left\|\sum_{i=1}^N p_i \nabla g_i(\mathbf{x}_t)\right\|^2 \leq 2\left\|\nabla \tilde{h}(\mathbf{x}_t)\right\|^2 + 2\omega^2, \tag{75}$$

where $\omega^2 := \sum_{i=1}^N p_i \sup_{\mathcal{C}^i \subset \mathcal{D}^i \subset \mathcal{M}^i \cup \mathcal{C}^i} \omega^2(h_i; \mathcal{D}^i)$ *and* $\omega^2(h_i; \mathcal{D}^i) := \sup_{\mathbf{x}} \left\|\nabla h_{i_{|_{\mathcal{D}^i}}}(\mathbf{x}_t) - \nabla \tilde{h}_i(\mathbf{x}_t)\right\|^2$.

Proof. Using the definitions of $\omega^2(h_i; \mathcal{D}^i)$ and Jensen's inequality, we obtain

$$\sup_{\mathbf{x}} \left\|\nabla g(\mathbf{x}_t) - \nabla \tilde{h}(\mathbf{x}_t)\right\|^2 = \sup_{\mathbf{x}} \left\|\nabla h_{|_{\mathcal{C}}}(\mathbf{x}_t) - \nabla \tilde{h}(\mathbf{x}_t)\right\|^2 \tag{76}$$

$$= \sup_{\mathbf{x}} \left\|\sum_{i=1}^N p_i \big(\nabla h_{i|_{\mathcal{C}^i}}(\mathbf{x}_t) - \nabla \tilde{h}_i(\mathbf{x}_t)\big)\right\|^2 \tag{77}$$

$$\leq \sup_{\mathbf{x}} \sum_{i=1}^N p_i \left\|\nabla h_{i|_{\mathcal{C}^i}}(\mathbf{x}_t) - \nabla \tilde{h}_i(\mathbf{x}_t)\right\|^2. \tag{78}$$

Using $\sup_{\mathbf{x}} \left(\sum_{i=1}^N \psi_i(\mathbf{x})\right) \leq \sum_{i=1}^N \sup_{\mathbf{x}} \psi_i(\mathbf{x})$ for $\psi_i : \mathbb{R}^d \to \mathbb{R}\ \forall i \in [N]$ in (78), we obtain

$$\sup_{\mathbf{x}} \left\|\nabla g(\mathbf{x}_t) - \nabla \tilde{h}(\mathbf{x}_t)\right\|^2 \leq \sum_{i=1}^N p_i \sup_{\mathbf{x}} \left\|\nabla h_{i|_{\mathcal{C}^i}}(\mathbf{x}_t) - \nabla \tilde{h}_i(\mathbf{x}_t)\right\|^2 \tag{79}$$

$$\leq \sum_{i=1}^N p_i \sup_{\mathcal{C}^i \subset \mathcal{D}^i \subset \mathcal{M}^i \cup \mathcal{C}^i} \sup_{\mathbf{x}} \left\|\nabla h_{i|_{\mathcal{D}^i}}(\mathbf{x}_t) - \nabla \tilde{h}_i(\mathbf{x}_t)\right\|^2 \tag{80}$$

$$\leq \sum_{i=1}^N p_i \sup_{\mathcal{C}^i \subset \mathcal{D}^i \subset \mathcal{M}^i \cup \mathcal{C}^i} \omega^2(h_i; \mathcal{D}^i) \tag{81}$$

$$= \omega^2 \tag{82}$$

Finally, using the above result, we obtain

$$\|\nabla g(\mathbf{x}_t)\|^2 = \left\|\nabla g(\mathbf{x}_t) - \nabla \tilde{h}(\mathbf{x}_t) + \nabla \tilde{h}(\mathbf{x}_t)\right\|^2 \tag{83}$$

$$\leq 2\left\|\nabla \tilde{h}(\mathbf{x}_t)\right\|^2 + 2\left\|\nabla g(\mathbf{x}_t) - \nabla \tilde{h}(\mathbf{x}_t)\right\|^2 \tag{84}$$

$$\leq 2\left\|\nabla \tilde{h}(\mathbf{x}_t)\right\|^2 + 2\omega^2. \tag{85}$$

□

1.3 Convergence Analysis of $\Gamma(t)$

To show the convergence of $\Gamma(t)$, we expand it using (31) to obtain

$$\Gamma(t) = \frac{L}{2}\beta_t^2 \|\sum_{i=1}^{N}\sum_{k=0}^{E-1} p_i(\mathbf{e}_{t,k}^i + \nabla g_i(\mathbf{x}_{t,k}^i))\|^2 - \beta_t(1-L\alpha_t)\langle \nabla f^\dagger(\mathbf{x}_t), \sum_{i=1}^{N}\sum_{k=0}^{E-1} p_i(\mathbf{e}_{t,k}^i + \nabla g_i(\mathbf{x}_{t,k}^i))\rangle \tag{86}$$

Further for brevity, we denote $\boldsymbol{\rho}_{t,k}^i = \nabla g_i(\mathbf{x}_{t,k}^i)$, $\bar{\boldsymbol{\rho}}_{t,k} = \sum_{i=1}^N p_i \boldsymbol{\rho}_{t,k}^i$, and $\bar{\mathbf{e}}_{t,k} = \sum_{i=1}^N p_i \mathbf{e}_{t,k}^i$. Using the inequality $\left\|\sum_{i=1}^{M} \mathbf{x}_i\right\|^2 \leq M \sum_{i=1}^{N} \|\mathbf{x}_i\|^2$ for $M = 2$, the first term of $\Gamma(t)$ can be expanded as

$$\begin{aligned}
&\frac{L}{2}\beta_t^2 \left\|\sum_{i=1}^{N}\sum_{k=0}^{E-1} p_i(\nabla g_i(\mathbf{x}_{t,k}^i) + \mathbf{e}_{t,k}^i)\right\|^2 \\
&\leq L\beta_t^2 \left\|\sum_{i=1}^{N}\sum_{k=0}^{E-1} p_i \nabla g_i(\mathbf{x}_{t,k}^i)\right\|^2 + L\beta_t^2 \left\|\sum_{i=1}^{N}\sum_{k=0}^{E-1} p_i \mathbf{e}_{t,k}^i\right\|^2 &(87)\\
&= L\beta_t^2 \|\sum_{k=0}^{E-1} \bar{\boldsymbol{\rho}}_{t,k}\|^2 + L\beta_t^2 \|\sum_{k=0}^{E-1} \bar{\mathbf{e}}_{t,k}\|^2. &(88)
\end{aligned}$$

The second term in $\Gamma(t)$ can be expanded as

$$\begin{aligned}
&-\beta_t(1-L\alpha_t)\langle \nabla f^\dagger(\mathbf{x}_t), \sum_{i=1}^{N}\sum_{k=0}^{E-1} p_i(\mathbf{e}_{t,k}^i + \nabla g_i(\mathbf{x}_{t,k}^i))\rangle \\
&= -\beta_t(1-L\alpha_t)\langle \nabla f^\dagger(\mathbf{x}_t), \textstyle\sum_{k=1}^{E-1} \bar{\boldsymbol{\rho}}_{t,k}\rangle - \beta_t(1-L\alpha_t)\langle \nabla f^\dagger(\mathbf{x}_t), \sum_{k=0}^{E-1} \bar{\mathbf{e}}_{t,k}\rangle. &(89)
\end{aligned}$$

Using (88) and (89) in (86), we obtain

$$\begin{aligned}
\Gamma(t) = L\beta_t^2 \|\sum_{k=0}^{E-1} \bar{\boldsymbol{\rho}}_{t,k}\|^2 - \beta_t(1-L\alpha_t)\langle \nabla f^\dagger(\mathbf{x}_t), \sum_{k=0}^{E-1} \bar{\boldsymbol{\rho}}_{t,k}\rangle \\
- \beta_t(1-L\alpha_t)\langle \nabla f^\dagger(\mathbf{x}_t), \sum_{k=0}^{E-1} \bar{\mathbf{e}}_{t,k}\rangle + L\beta_t^2 \left\|\sum_{k=0}^{E-1} \bar{\mathbf{e}}_{t,k}\right\|^2. &(90)
\end{aligned}$$

Lemma 12. *Let assumption 1 hold. For all $k \in \{0, \ldots, E-1\}$ and given any $t \in \{0, 1, \cdots T-1\}$, the gradient error* (31) *due to the incrementally aggregated gradients in the proposed algorithm is bounded as follows:*

$$\left\|\mathbf{e}_{t,k}^i\right\| \leq \beta_t LE \left\|\nabla g(\mathbf{x}_t)\right\| + 3\beta_t L^2 E \max_{0\leq b\leq k-1} \left\|\mathbf{x}_{t,b}^i - \mathbf{x}_t\right\|. \tag{91}$$

Proof. Expanding $\mathbf{e}_{t,k}^i$ as given in (31), we have

$$\begin{aligned}
\left\|\mathbf{e}_{t,k}^i\right\| &= \left\|\nabla g_i'(\mathbf{x}_{t,k}^i) - \nabla g_i(\mathbf{x}_{t,k}^i)\right\| &(92)\\
&= \left\|\frac{1}{|\mathcal{C}^i|}\sum_{j\in\mathcal{C}^i} \left\{\nabla g_{i,j}\left(\mathbf{x}_{t,\tau_{k,j}^i}^i\right) - \nabla g_{i,j}\left(\mathbf{x}_{t,k}^i\right)\right\}\right\| &(93)\\
&\overset{(a)}{\leq} \frac{1}{|\mathcal{C}^i|}\sum_{j\in\mathcal{C}^i} L\left\|\mathbf{x}_{t,\tau_{k,j}^i}^i - \mathbf{x}_{t,k}^i\right\| &(94)
\end{aligned}$$

where (a) follows from triangle inequality and L-smoothness of g_i.

Further simplifying, we note that

$$\frac{1}{|\mathcal{C}^i|}\sum_{j\in\mathcal{C}^i} L\left\|\mathbf{x}^i_{t,\tau^i_{k,j}}-\mathbf{x}^i_{k,j}\right\| \overset{(b)}{=} \frac{1}{|\mathcal{C}^i|}\sum_{j\in\mathcal{C}^i} L\left\|\sum_{l=\tau^i_{k,j}}^{k-1}\mathbf{x}^i_{t,l}-\mathbf{x}^i_{t,l+1}\right\| \tag{95}$$

$$\overset{(c)}{\leq} \frac{1}{|\mathcal{C}^i|}\sum_{j\in\mathcal{C}^i} L\sum_{l=\tau^i_{k,j}}^{k-1}\left\|\mathbf{x}^i_{t,l}-\mathbf{x}^i_{t,l+1}\right\| \tag{96}$$

$$\overset{(d)}{\leq} \frac{1}{|\mathcal{C}^i|}\sum_{j\in\mathcal{C}^i} L\sum_{l=0}^{k-1}\left\|\mathbf{x}^i_{t,l}-\mathbf{x}^i_{t,l+1}\right\| \tag{97}$$

$$= L\sum_{l=0}^{k-1}\left\|\mathbf{x}^i_{t,l}-\mathbf{x}^i_{t,l+1}\right\|, \tag{98}$$

where (b) follows from the properties of IAG with $0\leq\tau^i_{k,j}\leq k$, (c) follows using triangle inequality, (d) follows since each entry in the summation is non-negative and $0\leq\tau^i_{k,j}$. Using (4), we have

$$L\sum_{l=0}^{k-1}\left\|\mathbf{x}^i_{t,l}-\mathbf{x}^i_{t,l+1}\right\| = L\sum_{l=0}^{k-1}\left\|\beta_t\left(\nabla g(\mathbf{x}_t)-\nabla g_i(\mathbf{x}_t)+\nabla g'_i(\mathbf{x}^i_{t,l})\right)\right\| \tag{99}$$

$$= L\beta_t\sum_{l=0}^{k-1}\left[\left\|\nabla g(\mathbf{x}_t)-\nabla g_i(\mathbf{x}_t)+\nabla g_i(\mathbf{x}^i_{t,l})+\mathbf{e}^i_{t,l}\right\|\right] \tag{100}$$

$$\overset{(e)}{\leq} L\beta_t\sum_{l=0}^{k-1}\left[\left\|\nabla g(\mathbf{x}_t)\right\|+L\left\|\mathbf{x}^i_{t,l}-\mathbf{x}_t\right\|+\left\|\mathbf{e}^i_{t,l}\right\|\right], \tag{101}$$

where (e) follows from triangle inequality and L-smoothness of g_i. Further, using (94), we obtain

$$\left\|\mathbf{e}^i_{t,l}\right\| \leq \frac{1}{|\mathcal{C}^i|}\sum_{j\in\mathcal{C}^i} L\left\|\mathbf{x}^i_{t,\tau^i_{l,j}}-\mathbf{x}^i_{k,j}\right\| \tag{102}$$

$$\overset{(a)}{\leq} \frac{1}{|\mathcal{C}^i|}\sum_{j\in\mathcal{C}^i} L\left[\left\|\mathbf{x}^i_{l,\tau^i_{l,j}}-\mathbf{x}_t\right\|+\left\|\mathbf{x}^i_{t,\tau^i_{l,j}}-\mathbf{x}_t\right\|\right] \tag{103}$$

$$\overset{(b)}{\leq} L\max_{0\leq b\leq l}\left[\left\|\mathbf{x}^i_{t,b}-\mathbf{x}_t\right\|+\left\|\mathbf{x}^i_{t,b}-\mathbf{x}_t\right\|\right] \tag{104}$$

$$= 2L\max_{0\leq b\leq l}\{\left\|\mathbf{x}^i_{t,b}-\mathbf{x}_t\right\|\}, \tag{105}$$

where (a) follows using Triangle inequality and (b) follows since $0\leq\tau^i_{l,j}\leq l$ leads to $\left\|\mathbf{x}^i_{t,\tau^i_{l,j}}-\mathbf{x}_t\right\| \leq \max_{0\leq b\leq\tau^i_{l,j}}\{\left\|\mathbf{x}^i_{t,b}-\mathbf{x}_t\right\|\}\leq\max_{0\leq b\leq l}\{\left\|\mathbf{x}^i_{t,b}-\mathbf{x}_t\right\|\}$. Substituting (105) in (101), we obtain

$$\left\|\mathbf{e}^i_{t,k}\right\| \leq L\beta_t\sum_{l=0}^{k-1}\left[\left\|\nabla g(\mathbf{x}_t)\right\|+L\left\|\mathbf{x}^i_{t,l}-\mathbf{x}_t\right\|+2L\max_{0\leq b\leq l}\{\left\|\mathbf{x}^i_{t,b}-\mathbf{x}_t\right\|\}\right] \tag{106}$$

$$\leq L\beta_t\sum_{l=0}^{k-1}\left[\nabla g(\mathbf{x}_t)+L\max_{0\leq b\leq l}\{\left\|\mathbf{x}^i_{t,b}-\mathbf{x}_t\right\|+2L\max_{0\leq b\leq l}\{\left\|\mathbf{x}^i_{t,b}-\mathbf{x}_t\right\|\}\right] \tag{107}$$

$$= L\beta_t\sum_{l=0}^{k-1}\left[\nabla g(\mathbf{x}_t)+3L\max_{0\leq b\leq l}\{\left\|\mathbf{x}^i_{t,b}-\mathbf{x}_t\right\|\right] \tag{108}$$

$$\leq \beta_t LE\left\|\nabla g(\mathbf{x}_t)\right\|+3\beta_t L^2E\max_{0\leq b\leq k-1}\{\left\|\mathbf{x}^i_{t,b}-\mathbf{x}_t\right\|\}, \tag{109}$$

where the last inequality in the above expression holds since $k\leq E$. □

Lemma 13. *Let assumption 1 hold and let $\beta_t \leq \frac{1}{4LE}$. Then for $k \in \{0, \ldots, E-1\}$ and given any $t \in \{0, 1, \cdots T-1\}$, the bound on the client drift is given by*

$$\left\|\mathbf{x}_{t,k}^i - \mathbf{x}_t\right\| \leq 4\beta_t k \left\|\nabla g(\mathbf{x}_t)\right\| \leq 4\beta_t E \left\|\nabla g(\mathbf{x}_t)\right\|. \tag{110}$$

Proof. For any $i \in [N]$, from (4) and expanding $\nabla g_i'(\mathbf{x}_{t,k}^i)$ using (31), we obtain

$$\begin{aligned}
\left\|\mathbf{x}_{t,k+1}^i - \mathbf{x}_t\right\| &= \left\|\mathbf{x}_{t,k}^i - \mathbf{x}_t - \beta_t(\nabla g(\mathbf{x}_t) - \nabla g_i(\mathbf{x}_t) + \nabla g_i'(\mathbf{x}_{t,k}^i))\right\| && (111)\\
&= \left\|\mathbf{x}_{t,k}^i - \mathbf{x}_t - \beta_t(\nabla g(\mathbf{x}_t) - \nabla g_i(\mathbf{x}_t) + \nabla g_i(\mathbf{x}_{t,k}^i) + \mathbf{e}_{t,k}^i)\right\| && (112)\\
&\leq \left\|\mathbf{x}_{t,k}^i - \mathbf{x}_t\right\| + \beta_t \left\|\nabla g_i(\mathbf{x}_{t,k}^i) - \nabla g_i(\mathbf{x}_t)\right\| + \beta_t \left\|\nabla g(\mathbf{x}_t)\right\| + \beta_t \left\|\mathbf{e}_{t,k}^i\right\|. && (113)
\end{aligned}$$

where the last inequality in the above follows from triangle inequality. Using L-smoothness of g_i as given in assumption 1 in (113), we have

$$\begin{aligned}
\left\|\mathbf{x}_{t,k+1}^i - \mathbf{x}_t\right\| &\leq \left\|\mathbf{x}_{t,k}^i - \mathbf{x}_t\right\| + \beta_t L \left\|\mathbf{x}_{t,k}^i - \mathbf{x}_t\right\| + \beta_t \left\|\nabla g(\mathbf{x}_t)\right\| + \beta_t \left\|\mathbf{e}_{t,k}^i\right\| && (114)\\
&= (1 + \beta_t L) \left\|\mathbf{x}_{t,k}^i - \mathbf{x}_t\right\| + \beta_t \left\|\nabla g(\mathbf{x}_t)\right\| + \beta_t \left\|\mathbf{e}_{t,k}^i\right\| && (115)\\
&\overset{(a)}{\leq} (1 + \beta_t L) \left\|\mathbf{x}_{t,k}^i - \mathbf{x}_t\right\| + \beta_t \left\|\nabla g(\mathbf{x}_t)\right\| + \beta_t^2 LE \left\|\nabla g(\mathbf{x}_t)\right\| + 3\beta_t^2 L^2 E \max_{0 \leq b \leq k-1} \{\left\|\mathbf{x}_{t,b}^i - \mathbf{x}_t\right\| && (116)\\
&\overset{(b)}{\leq} (1 + \beta_t L) \left\|\mathbf{x}_{t,k}^i - \mathbf{x}_t\right\| + 2\beta_t \left\|\nabla g(\mathbf{x}_t)\right\| + \beta_t L \max_{0 \leq b \leq k-1} \{\left\|\mathbf{x}_{t,b}^i - \mathbf{x}_t\right\|\}, && (117)
\end{aligned}$$

where (a) follows from Lem. 12, and (b) follows using $\beta_t \leq \frac{1}{4LE} < \frac{1}{3LE}$.

For $k = 0$, it is straightforward since $\mathbf{x}_{t,0}^i = \mathbf{x}_t$. Going further, we demonstrate that $\left\|\mathbf{x}_{t,k}^i - \mathbf{x}_t\right\| \leq 4\beta_t k \left\|\nabla g(\mathbf{x}_t)\right\|$ for $k \in \{1, 2, \cdots, E-1\}$ using mathematical induction. First, we verify for $k = 1$. For $k = 1$, we have

$$\begin{aligned}
\left\|\mathbf{x}_{t,1}^i - \mathbf{x}_t\right\| &= \left\|\beta_t(\nabla g(\mathbf{x}_t) - \nabla g_i(\mathbf{x}_t) + \nabla g_i'(\mathbf{x}_{t,0}^i))\right\| && (118)\\
&\overset{(a)}{\leq} \beta_t \left\|\nabla g(\mathbf{x}_t)\right\| + \beta_t \left\|\nabla g_i(\mathbf{x}_t) - \nabla g_i'(\mathbf{x}_{t,0}^i)\right\| && (119)\\
&\overset{(b)}{=} \beta_t \left\|\nabla g(\mathbf{x}_t)\right\| \leq 4\beta_t \left\|\nabla g(\mathbf{x}_t)\right\|. && (120)
\end{aligned}$$

where (a) follows using triangle inequality and (b) follows from $\nabla g_i'(\mathbf{x}_{t,0}^i) = \frac{1}{|\mathcal{C}^i|}\sum_{j \in \mathcal{C}^i} \nabla g_{i,j}(\mathbf{x}_{t,\tau_{0,j}^i}^i) = \frac{1}{|\mathcal{C}^i|}\sum_{j \in \mathcal{C}^i} \nabla g_{i,j}(\mathbf{x}_t) = \nabla g_i(\mathbf{x}_t)$ because $\tau_{0,j}^i = 0$ and $\mathbf{x}_{t,0}^i = \mathbf{x}_t$. Hence, the induction base step holds.

We state the induction argument by first assuming that the condition is true for $k = l$, i.e.,

$$\left\|\mathbf{x}_{t,l}^i - \mathbf{x}_t\right\| \leq 4\beta_t l \left\|\nabla g(\mathbf{x}_t)\right\|. \tag{121}$$

Now, we need to demonstrate that the condition holds for $k = l + 1$. From (117), we obtain

$$\begin{aligned}
\left\|\mathbf{x}_{t,l+1}^i - \mathbf{x}_t\right\| &\leq (1 + \beta_t L) \left\|\mathbf{x}_{t,l}^i - \mathbf{x}_t\right\| + 2\beta_t \left\|\nabla g(\mathbf{x}_t)\right\| + \beta_t L \max_{0 \leq b \leq l-1} \{\left\|\mathbf{x}_{t,b}^i - \mathbf{x}_t\right\|\} && (122)\\
&= \left\|\mathbf{x}_{t,l}^i - \mathbf{x}_t\right\| + 2\beta_t \left\|\nabla g(\mathbf{x}_t)\right\| + \beta_t L(\left\|\mathbf{x}_{t,l}^i - \mathbf{x}_t\right\| + \max_{0 \leq b \leq l-1} \{\left\|\mathbf{x}_{t,b}^i - \mathbf{x}_t\right\|\}) && (123)\\
&\overset{(a)}{\leq} 4\beta_t l \left\|\nabla g(\mathbf{x}_t)\right\| + 2\beta_t \left\|\nabla g(\mathbf{x}_t)\right\| + \beta_t L(4\beta_t l + 4\beta_t l) \left\|\nabla g(\mathbf{x}_t)\right\| && (124)\\
&\overset{(b)}{\leq} \left[4\beta_t l + 2\beta_t + 2\beta_t(4\beta_t LE)\right] \left\|\nabla g(\mathbf{x}_t)\right\| && (125)\\
&\overset{(c)}{\leq} (4\beta_t l + 2\beta_t + 2\beta_t) \left\|\nabla g(\mathbf{x}_t)\right\| = 4\beta_t(l+1) \left\|\nabla g(\mathbf{x}_t)\right\|, && (126)
\end{aligned}$$

where (a) follows using the induction hypothesis, (b) follows since $l \leq E$, (c) follows from $\beta_t \leq \frac{1}{4LE}$.

Since the statement is true for $k = 1$ and true for $k = l + 1$ when it is true for $k = l$. Then, using the principle of mathematical induction, the statement is true for all $k \in \{0, 1, \cdots, E-1\}$. Finally, $\left\|\mathbf{x}_{t,k}^i - \mathbf{x}_t\right\| \leq 4\beta_t k \left\|\nabla g(\mathbf{x}_t)\right\| \leq 4\beta_t E \left\|\nabla g(\mathbf{x}_t)\right\|$ holds since $k < E$. □

Corollary 14. *Suppose that $\beta_t \leq \frac{1}{12LE}$. Then from the Lem. 12 and Lem. 13, it follows that*

$$\left\|\mathbf{e}_{t,k}^i\right\| \leq 2\beta_t LE \left\|\nabla g(\mathbf{x}_t)\right\| \tag{127}$$

Proof. Using the result of Lem. 12 and Lem. 13, we obtain

$$\left\|\mathbf{e}_{t,k}^i\right\| \leq \beta_t LE \left\|\nabla g(\mathbf{x}_t)\right\| + 3\beta_t L^2 E \max_{0\leq b\leq k-1} \left\|\mathbf{x}_{t,b}^i - \mathbf{x}_t\right\| \tag{128}$$

$$\overset{(a)}{\leq} \beta_t LE \left\|\nabla g(\mathbf{x}_t)\right\| + 3\beta_t L^2 E(4\beta_t E \left\|\nabla g(\mathbf{x}_t)\right\| \leq (\beta_t LE \left\|\nabla g(\mathbf{x}_t)\right\| + 12\beta_t^2 L^2 E^2) \left\|\nabla g(\mathbf{x}_t)\right\| \tag{129}$$

$$\overset{(b)}{\leq} 2\beta_t LE \left\|\nabla g(\mathbf{x}_t)\right\|, \tag{130}$$

where (a) follows from Lem. 3 and (b) follows from using $\beta_t \leq \frac{1}{12LE}$.

□

Lemma 15. *Suppose assumption 1 holds. Then at the t-th iteration at each client $i \in [N]$, the sequence of local updates, $\{\mathbf{x}_{t,k}^i\}_{k=0}^{E-1}$, generated by our proposed algorithm satisfies the following:*

$$\textit{(I)} \ \|\sum_{k=0}^{E-1} \bar{\boldsymbol{\rho}}_{t,k}\| \leq 4L\beta_t E^2 \left\|\nabla g(\mathbf{x}_t)\right\| + E \left\|\nabla g(\mathbf{x}_t)\right\|, \tag{131}$$

$$\textit{(II)} \ \|\sum_{k=0}^{E-1} \bar{\boldsymbol{\rho}}_{t,k}\|^2 \leq 32\beta_t^2 L^2 E^4 \left\|\nabla g(\mathbf{x}_t)\right\|^2 + 2E^2 \left\|\nabla g(\mathbf{x}_t)\right\|^2. \tag{132}$$

Proof. To prove (I), we start from the definition of $\bar{\boldsymbol{\rho}}_{t,k}$ as given after (86) , to obtain the following:

$$\left\|\sum_{k=0}^{E-1} \bar{\boldsymbol{\rho}}_{t,k}\right\| = \left\|\sum_{k=0}^{E-1}\sum_{i=1}^{N} p_i \nabla g_i(\mathbf{x}_{t,k}^i)\right\| \leq \sum_{k=0}^{E-1} \left\|\sum_{i=1}^{N} p_i \nabla g_i(\mathbf{x}_{t,k}^i)\right\|, \tag{133}$$

where the inequality follows from the triangle inequality. Adding and subtracting $\nabla g_i(\mathbf{x}_t)$, we have

$$\left\|\sum_{k=0}^{E-1} \bar{\boldsymbol{\rho}}_{t,k}\right\| \leq \sum_{k=0}^{E-1} \left\|\sum_{i=1}^{N} p_i \big(\nabla g_i(\mathbf{x}_{t,k}^i) - \nabla g_i(\mathbf{x}_t)\big)\right\| + \sum_{k=0}^{E-1} \left\|\sum_{i=1}^{N} p_i \big(\nabla g_i(\mathbf{x}_t)\big)\right\| \tag{134}$$

$$\overset{(b)}{\leq} \sum_{k=0}^{E-1}\sum_{i=1}^{N} p_i \left\|\nabla g_i(\mathbf{x}_{t,k}^i) - \nabla g_i(\mathbf{x}_t)\right\| + \sum_{k=0}^{E-1} \left\|\nabla g(\mathbf{x}_t)\right\| \tag{135}$$

$$\overset{(c)}{\leq} \sum_{k=0}^{E-1}\sum_{i=1}^{N} p_i L \left\|\mathbf{x}_{t,k}^i - \mathbf{x}_t\right\| + E \left\|\nabla g(\mathbf{x}_t)\right\| \tag{136}$$

$$\overset{(d)}{\leq} \sum_{k=0}^{E-1}\sum_{i=1}^{N} p_i L(4\beta_t E) \left\|\nabla g(\mathbf{x}_t)\right\| + E \left\|\nabla g(\mathbf{x}_t)\right\| \tag{137}$$

$$= 4L\beta_t E^2 \left\|\nabla g(\mathbf{x}_t)\right\| + E \left\|\nabla g(\mathbf{x}_t)\right\|, \tag{138}$$

where (b) follows from triangle inequality and the definition of $\nabla g_i(\mathbf{x}_t)$, (c) follows from the L-smoothness assumption, and (d) follows from Lem. 13.

To prove (II), we start from the definition of $\bar{\boldsymbol{\rho}}_{t,k}$ and use $\nabla g(\mathbf{x}_t) = \sum_{i=1}^{N} p_i \nabla g_i(\mathbf{x}_t)$ to obtain the following:

$$\left\|\sum_{k=0}^{E-1} \bar{\boldsymbol{\rho}}_{t,k}\right\|^2 = \left\|\sum_{k=0}^{E-1}\sum_{i=1}^{N} p_i \boldsymbol{\rho}_{t,k}^i\right\|^2 \overset{(a)}{\leq} E\sum_{k=0}^{E-1}\left\|\sum_{i=1}^{N} p_i \boldsymbol{\rho}_{t,k}^i\right\|^2 \tag{139}$$

$$\leq E\sum_{k=0}^{E-1}\left\|\sum_{i=1}^{N} p_i\big(\nabla g_i(\mathbf{x}_{t,k}^i) - \nabla g_i(\mathbf{x}_t) + \nabla g_i(\mathbf{x}_t)\big)\right\|^2 \tag{140}$$

$$\leq 2E\sum_{k=0}^{E-1}\left\|\sum_{i=1}^{N} p_i\big(\nabla g_i(\mathbf{x}_{t,k}^i) - \nabla g_i(\mathbf{x}_t)\big)\right\|^2 + 2E\sum_{k=0}^{E-1}\left\|\nabla g(\mathbf{x}_t)\right\|^2 \tag{141}$$

$$\overset{(b)}{\leq} 2E\sum_{k=0}^{E-1}\sum_{i=1}^{N} p_i\left\|\nabla g_i(\mathbf{x}_{t,k}^i) - \nabla g_i(\mathbf{x}_t)\right\|^2 + 2E^2\left\|\nabla g(\mathbf{x}_t)\right\|^2 \tag{142}$$

$$\overset{(c)}{\leq} 2E\sum_{k=0}^{E-1}\sum_{i=1}^{N} p_i\left\|L\big(\mathbf{x}_{t,k}^i - \mathbf{x}_t\big)\right\|^2 + 2E^2\left\|\nabla g(\mathbf{x}_t)\right\|^2 \tag{143}$$

$$\overset{(d)}{\leq} 2E\sum_{k=0}^{E-1}\sum_{i=1}^{N} p_i\big(4\beta_t LE\left\|\nabla g(\mathbf{x}_t)\right\|\big)^2 + 2E^2\left\|\nabla g(\mathbf{x}_t)\right\|^2 \tag{144}$$

$$= 32\beta_t^2 L^2 E^4\left\|\nabla g(\mathbf{x}_t)\right\|^2 + 2E^2\left\|\nabla g(\mathbf{x}_t)\right\|^2, \tag{145}$$

where (a),(b) follows from using Jensen's inequality, (c) follows from L-smoothness assumption, and (d) follows from Lem. 13. □

In the following lemma, we analyze the effect of the delayed gradient at the t-th iteration as observed on the average error across E local steps.

Lemma 16. *The average error accumulated across E local steps due to the delayed gradient at the t-th iteration of the proposed algorithm satisfies*

$$\left\|\sum_{k=0}^{E-1} \bar{\mathbf{e}}_{t,k}\right\| \leq 2\beta_t LE^2\left\|\nabla g(\mathbf{x}_t)\right\|. \tag{146}$$

Proof. We start from the definition of $\bar{\mathbf{e}}_{t,k}$ as given after (86), and then using triangle inequality and corollary 14, we obtain

$$\left\|\sum_{k=0}^{E-1} \bar{\mathbf{e}}_{t,k}\right\| = \left\|\sum_{k=0}^{E-1}\sum_{i=1}^{N} p_i \mathbf{e}_{t,k}^i\right\| \leq \sum_{k=0}^{E-1}\sum_{i=1}^{N} p_i\left\|\mathbf{e}_{t,k}^i\right\| \leq \sum_{k=0}^{E-1}\sum_{i=1}^{N} p_i\big(2\beta_t LE\left\|\nabla g(\mathbf{x}_t)\right\|\big) = 2\beta_t LE^2\left\|\nabla g(\mathbf{x}_t)\right\|. \tag{147}$$

□

Next we provide the proof of the Lem. 4 present in the main manuscript.

Lemma 4 Suppose that the assumptions 1, 2, 3 hold and the step-sizes satisfy $\alpha_t = \alpha < \frac{2}{L(1+m)}$ and $\beta_t = \beta < \frac{c}{\sqrt{T}} \forall t \in \{0, 1, \cdots, T-1\}$ and for some $c, m \in \mathbb{R}^+$. Then the following holds for the forgetting term $\Gamma(t)$:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\Gamma(t)] < \mathcal{O}\bigg(\frac{1}{T} + \frac{1}{\sqrt{T}}\bigg). \tag{148}$$

Proof. From (90) we obtain $\Gamma(t)$ as

$$\Gamma(t) = \underbrace{L\beta_t^2 \left\| \sum_{i=1}^{N} \sum_{k=0}^{E-1} \bar{\boldsymbol{\rho}}_{t,k} \right\|^2}_{T1} \underbrace{-\beta_t(1-L\alpha_t)\left\langle \nabla f^{\dagger}(\mathbf{x}_t), \sum_{k=0}^{E-1} \bar{\boldsymbol{\rho}}_{t,k} \right\rangle}_{T2} \underbrace{-\beta_t(1-L\alpha_t)\left\langle \nabla f^{\dagger}(\mathbf{x}_t), \sum_{k=0}^{E-1} \bar{\mathbf{e}}_{t,k} \right\rangle}_{T3}$$
$$\underbrace{+L\beta_t^2 \left\| \sum_{k=0}^{E-1} \bar{\mathbf{e}}_{t,k} \right\|^2}_{T4}. \tag{149}$$

To complete the proof, we will bound each of the terms ($T1$, $T2$, $T3$, and $T4$) separately. Using Lem. 15 and 16, we obtain

$$T1 \leq 32\beta_t^4 L^3 E^4 \left\|\nabla g(\mathbf{x}_t)\right\|^2 + 2\beta_t^2 LE^2 \left\|\nabla g(\mathbf{x}_t)\right\|^2 \tag{150}$$
$$T4 \leq 4\beta_t^4 L^3 E^4 \left\|\nabla g(\mathbf{x}_t)\right\|^2. \tag{151}$$

To bound the other two terms ($T2$ and $T3$), we use the bounded bias assumption, Lem. 10 along with Lem. 15, 16 and Cauchy-Schwartz inequality, to obtain

$$T2 = -\beta_t(1-L\alpha_t)\left\langle \nabla f^{\dagger}(\mathbf{x}_t), \sum_{k=0}^{E-1} \bar{\boldsymbol{\rho}}_{t,k} \right\rangle \leq \beta_t(1-L\alpha_t) \left\|\nabla f^{\dagger}(\mathbf{x}_t)\right\| \left\| \sum_{k=0}^{E-1} \bar{\boldsymbol{\rho}}_{t,k} \right\| \tag{152}$$
$$\leq \beta_t(1-L\alpha_t) r \left\|\nabla g(\mathbf{x}_t)\right\| \left(4\beta_t LE^2 \left\|\nabla g(\mathbf{x}_t)\right\| + E \left\|\nabla g(\mathbf{x}_t)\right\|\right) \tag{153}$$
$$= \left[4\beta_t^2(1-L\alpha_t) rLE^2 + \beta_t(1-L\alpha_t) rE\right] \left\|\nabla g(\mathbf{x}_t)\right\|^2. \tag{154}$$

For the term $T3$, we obtain

$$T3 = -\beta_t(1-L\alpha_t)\left\langle \nabla f^{\dagger}(\mathbf{x}_t), \sum_{k=0}^{E-1} \bar{\mathbf{e}}_{t,k} \right\rangle \leq \beta_t(1-L\alpha_t) \left\|\nabla f^{\dagger}(\mathbf{x}_t)\right\| \left\| \sum_{k=0}^{E-1} \bar{\mathbf{e}}_{t,k} \right\| \tag{155}$$
$$\leq \beta_t(1-L\alpha_t) r \left\|\nabla g(\mathbf{x}_t)\right\| \left(2\beta_t LE^2 \left\|\nabla g(\mathbf{x}_t)\right\|\right) = 2r\beta_t^2(1-L\alpha_t) LE^2 \left\|\nabla g(\mathbf{x}_t)\right\|^2. \tag{156}$$

Using (150), (151), (154), and (156) in (149), we obtain

$$\Gamma(t) \leq \underbrace{32\beta_t^4 L^3 E^4 \left\|\nabla g(\mathbf{x}_t)\right\|^2 + 2\beta_t^2 LE^2 \left\|\nabla g(\mathbf{x}_t)\right\|^2}_{T1}$$
$$+ \underbrace{4\beta_t^2(1-L\alpha_t) rLE^2 \left\|\nabla g(\mathbf{x}_t)\right\|^2 + \beta_t(1-L\alpha_t) rE \left\|\nabla g(\mathbf{x}_t)\right\|^2}_{T2}$$
$$+ \underbrace{2r\beta_t^2(1-L\alpha_t) LE^2 \left\|\nabla g(\mathbf{x}_t)\right\|^2}_{T3} + \underbrace{4\beta_t^4 L^3 E^4 \left\|\nabla g(\mathbf{x}_t)\right\|^2}_{T4} \tag{157}$$
$$= \left[36\beta_t^4 L^3 E^4 + 2\beta_t^2 LE^2 + r\beta_t E(1-L\alpha_t)\big(4\beta_t LE + 2\beta_t LE + 1\big)\right] \left\|\nabla g(\mathbf{x}_t)\right\|^2 \tag{158}$$
$$\overset{(a)}{\leq} \left[36\beta_t^4 L^3 E^4 + 2\beta_t^2 LE^2 + r\beta_t E(1-L\alpha_t)\big(6\beta_t LE + 1\big)\right] \left(2\left\|\nabla \tilde{h}(\mathbf{x}_t)\right\|^2 + 2\omega^2\right) \tag{159}$$
$$= \left[72\beta_t^4 L^3 E^4 + 4\beta_t^2 LE^2 + 2r\beta_t E(1-L\alpha_t)\big(6\beta_t LE + 1\big)\right] \left(\left\|\nabla \tilde{h}(\mathbf{x}_t)\right\|^2 + \omega^2\right) \tag{160}$$
$$\overset{(b)}{\leq} \frac{Q_t}{Z_t}[\tilde{h}(\mathbf{x}_t) - \tilde{h}(\mathbf{x}_{t+1})] + Q_t\omega^2, \tag{161}$$

where (a) follows using Lem. 11, (b) follows from (70), $Q_t \triangleq 72\beta_t^4 L^3 E^4 + 4\beta_t^2 LE^2 + 2r\beta_t E(1-L\alpha_t)\big(6\beta_t LE + 1\big)$ and $Z_t \triangleq \beta_t E(1 - 80\beta_t^3 L^3 E^3 - 10\beta_t LE)$.

Considering $\alpha_t = \alpha$, $\beta_t = \beta$ $\forall t \in \{0, 1, \cdots, T-1\}$, we obtain $Q_t = Q = 72\beta^4L^3E^4 + 4\beta^2LE^2 + 2r\beta E(1 - L\alpha)\big(6\beta LE + 1\big)$ and $Z_t = Z = \beta E(1 - 80\beta^3L^3E^3 - 10\beta LE)$. Telescoping the above from $t = 0$ to $T-1$ we obtain

$$\sum_{t=0}^{T-1} \Gamma(t) \leq \sum_{t=0}^{T-1} \left[\frac{Q}{Z} \{\tilde{h}(\mathbf{x}_t) - \tilde{h}(\mathbf{x}_{t+1})\} + Q\omega^2 \right] \tag{162}$$

$$= \frac{Q}{Z}\big[\tilde{h}(\mathbf{x}_0) - \tilde{h}(\mathbf{x}_T)\big] + QT\omega^2. \tag{163}$$

Putting back the values of Q, Z and using $\Delta_{\tilde{h}} = \tilde{h}(\mathbf{x}_0) - \tilde{h}(\mathbf{x}_T)$, we obtain

$$\sum_t \Gamma(t) \leq \frac{72\beta^4L^3E^4 + 4\beta^2LE^2 + 2r\beta E(1 - L\alpha)\big(4\beta L^2E + 2\beta LE + 1\big)}{\beta E(1 - 80\beta^3L^3E^3 - 10\beta LE)} \Delta_{\tilde{h}}$$

$$+ T\big[72\beta^4L^3E^4 + 4\beta^2LE^2 + 2r\beta E(1 - L\alpha)\big(6\beta LE + 1\big)\big]\omega^2 \tag{164}$$

$$\leq \mathcal{O}\left(\frac{\beta^3L^4 + \beta L + \beta L^2 + \beta L + 1}{1 - \beta^3L^3 - 10\beta L} \Delta_{\tilde{h}} + T\big(\beta^4L^3 + \beta^2L + \beta\big(\beta L + 1\big)\big)\omega^2 \right). \tag{165}$$

The ω term depends on the memory choice $\mathcal{D} = \{\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^N\}$. To handle this randomness, we choose

$$\mathcal{D}^* = \underset{C \subset D \subset P \cup C}{\operatorname{argmax}} \frac{\beta^3L^4 + \beta L^2 + 2\beta L + 1}{1 - \beta^3L^3 - 10\beta L} \Delta_{\tilde{h}} \tag{166}$$

Taking expectation on both sides of (165), averaging over T and using $\beta < \frac{c}{\sqrt{T}}$ for some $c > 0$, we obtain

$$\frac{1}{T}\sum_t \mathbb{E}[\Gamma(t)] < \frac{1}{T}\mathcal{O}\left(\frac{\beta^3 + \beta + 1}{1 - \beta^3 - \beta} \Delta_{h_{|_{\mathcal{D}^*}}} + T\big(\beta^4 + \beta^2 + \beta\big)\omega^2 \right) < \mathcal{O}\left(\frac{1}{T} + \frac{1}{\sqrt{T}} \right). \tag{167}$$

Next, we provide the proof for the Theorem 5 present in the main manuscript.

Theorem 5. Let $\{\mathbf{x}_t\}_{t=1}^T$ be the sequence generated by algorithm 1, and the step-sizes satisfy $\alpha_t = \frac{1}{L(m+1)}$ and $\beta_t < \frac{c}{\sqrt{T}} \forall \{0, 1, \cdots, T-1\}$ and for some $c, m \in \mathbb{R}^+$. Then, we obtain the following rate of convergence

$$\min_t \mathbb{E}\left[\|\nabla f(\mathbf{x}_t)\|^2 \right] < \mathcal{O}\left(\frac{1}{\sqrt{T}} \right). \tag{168}$$

Proof. From Theorem 2 and Lem. 4, we have the following two results

$$\min_t \mathbb{E}\left[\|\nabla f(\mathbf{x}_t)\|^2 \right] \leq \frac{2L(1+m)}{T} \left(f(\mathbf{x}_0) - f(\mathbf{x}_T) + \sum_{t=0}^{T-1} \mathbb{E}[\Gamma(t)] \right),$$

$$\frac{1}{T}\sum_t \mathbb{E}[\Gamma(t)] < \mathcal{O}\left(\frac{1}{T} + \frac{1}{\sqrt{T}} \right).$$

Using these together, we obtain

$$\min_t \mathbb{E}\left[\|\nabla f(\mathbf{x}_t)\|^2 \right] < \mathcal{O}\left(\frac{1}{\sqrt{T}} \right).$$

Theorem 5 gives the rate of convergence for the previous task. Next, we will analyze the IFO complexity of the C-FLAG algorithm 1.

Corollary 17. *Let the step-sizes satisfy $\alpha_t = \frac{1}{L(m+1)}$ and $\beta_t < \frac{c}{\sqrt{T}} \forall t \in \{0, 1, \cdots, T-1\}$, and for some $c \in \mathbb{R}^+$. Then the IFO complexity of the algorithm 1 to obtain an ϵ-accurate solution is:*

$$IFO\ calls = \mathcal{O}\left(\frac{1}{\epsilon^2} \right). \tag{169}$$

Proof. Recall that the IFO complexity of an algorithm for an ϵ-accurate solution is given as

$$T(\epsilon) = \min\{T : \min_t \ \mathbb{E}[\|\nabla f(\mathbf{x}_t)\|^2] \leq \epsilon\} \tag{170}$$

For each step, an IFO call is utilised in calculating gradients for that step. From Theorem 2, we get

$$\min_t \mathbb{E}\left[\|\nabla f(\mathbf{x}_t)\|^2\right] \leq \frac{2L(1+m)}{T}\left(F + \sum_{t=0}^{T-1}\mathbb{E}[\Gamma(t)]\right) \tag{171}$$

Hence, $\mathbb{E}\left[\|\nabla f(\mathbf{x}_t)\|^2\right] \to \epsilon$ leads to $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$. □

Thus far, we presented the analysis with a constant step-size for α_t and diminishing step-size for β_t. For completeness, in the sequel, we discuss the effect of constant learning rates on the overall convergence rate.

Lemma 18. *Suppose that the assumptions 1, 2, 3 hold and the learning rates satisfy $\alpha_t < \frac{2}{L(1+m)}$ and $\beta_t = \frac{1}{60LE}$ $\forall t \in \{0, 1, \cdots, T-1\}$. Then we obtain the following bound on $\Gamma(t)$:*

$$\sum_{t=0}^{T-1}\frac{\mathbb{E}[\Gamma(t)]}{T} < O\left(\frac{1}{T}+1\right). \tag{172}$$

Proof. From (90) we obtain $\Gamma(t)$ as

$$\Gamma(t) = \underbrace{L\beta_t^2\left\|\sum_{i=1}^{N}\sum_{k=0}^{E-1}\bar{\boldsymbol{\rho}}_{t,k}\right\|^2}_{T1} \underbrace{-\beta_t(1-L\alpha_t)\left\langle \nabla f^{\dagger}(\mathbf{x}_t), \sum_{k=0}^{E-1}\bar{\boldsymbol{\rho}}_{t,k}\right\rangle}_{T2} \underbrace{-\beta_t(1-L\alpha_t)\left\langle \nabla f^{\dagger}(\mathbf{x}_t), \sum_{k=0}^{E-1}\bar{\mathbf{e}}_{t,k}\right\rangle}_{T3}$$
$$\underbrace{+L\beta_t^2\left\|\sum_{k=0}^{E-1}\bar{\mathbf{e}}_{t,k}\right\|^2}_{T4}. \tag{173}$$

Next we proceed similar to the proof of Theorem 5 provided above, and from (161) we obtain,

$$\Gamma(t) < \frac{Q_t}{Z_t}[\tilde{h}(\mathbf{x}_t) - \tilde{h}(\mathbf{x}_{t+1})] + Q_t\omega^2, \tag{174}$$

where $Q_t = 72\beta_t^4L^3E^4 + 4\beta_t^2LE^2 + 2r\beta_tE(1-L\alpha_t)\big(6\beta_tLE+1\big)$ and $Z_t = \beta_tE(1-80\beta_t^3L^3E^3 - 10\beta_tLE)$.

Considering $\alpha_t = \alpha$, $\beta_t = \beta$ $\forall t \in \{0, 1, \cdots, T-1\}$, we obtain $Q_t = Q = 72\beta^4L^3E^4 + 4\beta^2LE^2 + 2r\beta E(1 - L\alpha)\big(6\beta LE+1\big)$ and $Z_t = Z = \beta E(1-80\beta^3L^3E^3 - 10\beta LE)$. By summing up the above equation from $t=0$ to $T-1$ we obtain

$$\sum_{t=0}^{T-1}\Gamma(t) \leq \sum_{t=0}^{T-1}\left[\frac{Q}{Z}\{\tilde{h}(\mathbf{x}_t) - \tilde{h}(\mathbf{x}_{t+1})\} + Q\sup_{C\subset D\subset M\cup C}\omega^2_{h|_D}\right] \tag{175}$$
$$= \frac{Q}{Z}\left[\tilde{h}(\mathbf{x}_0) - \tilde{h}(\mathbf{x}_T)\right] + QT\omega^2. \tag{176}$$

Assuming that $\beta \leq \frac{1}{2LE}$ and finally using $\beta = \frac{1}{60LE}$, we obtain the following inequalities

$$\frac{Q}{Z} \leq 16r(1-L\alpha) + 22, \tag{177}$$
$$Q \leq \frac{11}{60L} + \frac{2r}{15L}(1-L\alpha). \tag{178}$$

Using (177) and (178) in (176), we obtain

$$\sum_{t=0}^{T-1}\Gamma(t) \leq \big[16r(1-L\alpha)+22)\big]\big[\tilde{h}(\mathbf{x}_0)-\tilde{h}(\mathbf{x}_T)\big] + \bigg[\frac{11}{60L}+\frac{2r}{15L}(1-L\alpha)\bigg]T\omega^2. \tag{179}$$

Considering average over T iterations and denoting $\Delta_{\tilde{h}} = \tilde{h}(\mathbf{x}_0)-\tilde{h}(\mathbf{x}_T)$, we obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\Gamma(t) \leq \frac{16r(1-L\alpha)+22}{T}\Delta_{|_D h} + \bigg[\frac{11}{60L}+\frac{2r}{15L}(1-L\alpha)\bigg]\omega^2. \tag{180}$$

To handle the randomness of the memory choice, we choose

$$D^* = \underset{C\subset D\subset P\cup C}{\operatorname{argmax}} \frac{16r(1-L\alpha)+22}{T}\Delta_{\tilde{h}}. \tag{181}$$

Taking expectations on both sides, we obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\Gamma(t)] < \mathcal{O}\bigg(\frac{16r(1-L\alpha)+22}{T}\Delta_{h_{|_{D^*}}} + \bigg[\frac{11}{60L}+\frac{2r}{15L}(1-L\alpha)\bigg]\omega^2\bigg) \tag{182}$$

$$< \mathcal{O}\bigg(\frac{1}{T}+1\bigg). \tag{183}$$

So, if we take constant step-sizes for both α_t and β_t, then in our proposed method, the average of the cumulative forgetting terms still converges with constant time complexity $\mathcal{O}(1)$. □

2 PROPOSED ALGORITHM

The proposed C-FLAG algorithm was presented in Sec. 3 of the main manuscript. Here, we complete the algorithm by presenting the pseudocode for `AdapLR` where *AdapFlag* is a boolean variable with $AdapFlag = True$ implies that the algorithm employs adaptive learning rates.

Algorithm 2 C-FLAG: Continual Federated Learning with Aggregated Gradients

Require: Step-size α, β, initial model $\mathbf{x}_0$, *AdapFlag*
Ensure: $\mathbf{x}_t$ for $t = 1, \ldots, T$

for $t = 0, \ldots, T-1$ **do**
  For client $i = 1, \ldots, N$, compute $\nabla g_i(\mathbf{x}_t)$ and $\nabla f_i^{\dagger}(\mathbf{x}_t)$, and transmit to the server.
  Server computes $\nabla g(\mathbf{x}_t)$ and $\nabla f^{\dagger}(\mathbf{x}_t)$, and broadcasts to each client.
  for client $i = 1, 2, \ldots N$ in parallel **do**
    Set $\mathbf{x}_{t,0}^i = \mathbf{x}_t$
    for $k = 0, \ldots, E-1$ **do**
      Compute $\nabla g_i'(\mathbf{x}_{t,k}^i)$ using (5) and $\mathbf{x}_{t,k+1}^i$ using (4).
    end for
    $\Delta\mathbf{x}_t^i = \texttt{AdapLR}(\mathbf{x}_{t,E}^i, \nabla f^{\dagger}(\mathbf{x}_t), \alpha, \beta, AdapFlag)$.
    Transmit $\Delta\mathbf{x}_t^i$ to the server.
  end for
  Server computes and broadcasts $\mathbf{x}_{t+1}$ using $\mathbf{x}_{t+1} = \mathbf{x}_t - \sum_{i=1}^{N} p_i \Delta\mathbf{x}_t^i$.
end for

Algorithm 3 AdapLR

Require: $\mathbf{x}_{t,E}^i, \nabla f^{\dagger}(\mathbf{x}_t), \alpha, \beta, AdapFlag$
Ensure: $\Delta\mathbf{x}_t^i$

if not *AdapFlag* **then**
  return $\Delta\mathbf{x}_t^i = \alpha_t \nabla f^{\dagger}(\mathbf{x}_t) + \mathbf{x}_{t,E}^i$
end if
Compute $\mathbf{a}_i = \mathbf{x}_t - E\beta_t(\nabla g(\mathbf{x}_t) - \nabla g_i(\mathbf{x}_t)) - \mathbf{x}_{t,E}^i$.
Compute $\Lambda_{t,i} = \langle \nabla f^{\dagger}(\mathbf{x}_t), \mathbf{a}_i \rangle$.
if $\Lambda_{t,i} > 0$ **then**
  $\alpha_{t,i} = \alpha$
  $\beta_{t,i} = \frac{(1-L\alpha)\Lambda_{t,i}}{LNp_i\|\mathbf{a}_i\|^2}$ // Worst case
else
  $\alpha_{t,i} = \alpha(1 - \frac{\Lambda_{t,i}}{\|\nabla f^{\dagger}(\mathbf{x}_t)\|^2})$
  $\beta_{t,i} = \beta$
end if
$\mathbf{a}_i = \frac{\beta_{t,i}}{\beta} * \mathbf{a}_i$ // re-scaling with adaptive rate
$\Delta\mathbf{x}_t^i = \alpha_{t,i}\nabla f^{\dagger}(\mathbf{x}_t) + E\beta_{t,i}(\nabla g(\mathbf{x}_t) - \nabla g_i(\mathbf{x}_t)) + \mathbf{a}_i$
return $\Delta\mathbf{x}_t^i$

3 C-FLAG WITH ADAPTIVE LEARNING RATES

In Sec. 5 of the main manuscript, we presented the analysis regarding the effect of choosing adaptive learning rates in C-FLAG. Essentially, this translated the convergence analysis into an easily implementable solution where learning rates are adapted to achieve lower forgetting at each iteration. In the following subsections, we provide detailed discussions on the average and the worst-case scenarios. The results obtained here are encapsulated in Table 1 of the main manuscript.

3.1 Average Case

In the average case, we denote the catastrophic forgetting as $\Gamma_{av}(t)$ and $\Gamma_{i,av}(t)$ for the server and the i-th client respectively. Using the average case condition given as $\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} p_i p_j C_{i,j} = 0$, the forgetting terms can be rewritten as

$$\begin{aligned}\Gamma(t) = \Gamma_{av}(t) &= \frac{L\beta_t^2}{2}\sum_{i=1}^{N} p_i^2 \left\|\mathbf{a}_i\right\|^2 - \beta_t(1-L\alpha_t)\sum_{i=1}^{N} p_i \Lambda_{t,i} \\ &= \sum_{i=1}^{N} p_i \left[\frac{L\beta_t^2}{2} p_i \left\|\mathbf{a}_i\right\|^2 - \beta_t(1-L\alpha_t)\Lambda_{t,i}\right] = \sum_{i=1}^{N} p_i \Gamma_{i,av}(t), \end{aligned} \tag{184}$$

where $\Gamma_{i,av}(t) = \frac{L\beta_t^2}{2}\sum_{i=1}^{N} p_i \left\|\mathbf{a}_i\right\|^2 - \beta_t(1-L\alpha_t)\sum_{i=1}^{N}\Lambda_{t,i}$. Subsequently, taking expectation on both sides of (184), we obtain

$$\mathbb{E}[\Gamma_{av}(t)] = \sum_{i=1}^{N} p_i \mathbb{E}[\Gamma_{i,av}(t)]. \tag{185}$$

We aim to minimize $\Gamma_{av}(t)$ by minimizing individual clients' contribution, $\Gamma_{i,av}(t)$. We achieve this by obtaining client-specific adaptive learning rates $\alpha_{t,i}$ and $\beta_{t,i}$ for $i \in [N]$. Since $\Gamma_{i,av}(t)$ is a quadratic polynomial in β_t, we obtain its solution $\beta_{t,i}^* = \frac{(1-L\alpha_t)\Lambda_{t,i}}{Lp_i\|\mathbf{a}_i\|^2}$ which leads to $\mathbb{E}[\Gamma_{i,av}^*] = -\frac{[(1-L\alpha_t)\Lambda_{t,i}]^2}{2Lp_i\|\mathbf{a}_i\|^2}$ for a fixed α_t. For the transference case ($\Lambda_{t,i} > 0$), we choose $\beta_{t,i} = \beta_{t,i}^*$ and $\alpha_{t,i} = \alpha_t = \alpha$, which is a feasible solution to our optimization problem 14 as in the main manuscript, and consequently, i-th client's contribution towards the global forgetting becomes negative since $\mathbb{E}[\Gamma_{i,av}^*] < 0$. However, in the case of interference ($\Lambda_{t,i} \leq 0$), $\beta_{t,i}^*$ is negative, which violates the constraint of our optimization problem. Moreover, $\Gamma_{i,ad}(t)$ is monotonically increasing in β_t for any feasible β_t. Hence, we propose to adapt the α_t to find $\alpha_{t,i}$. Recall that each client takes E local gradient steps on the current data before server aggregation. Consequently, the overall effect of i-th client's accumulated gradients, $\mathbf{a}_i$, on the direction of $\nabla f^\dagger(\mathbf{x}_t)$, is given by $\langle \frac{\nabla f^\dagger(\mathbf{x}_t)}{\|\nabla f^\dagger(\mathbf{x}_t)\|}, \mathbf{a}_i\rangle \frac{\nabla f^\dagger(\mathbf{x}_t)}{\|\nabla f^\dagger(\mathbf{x}_t)\|} = \frac{\Lambda_{t,i}}{\|\nabla f^\dagger(\mathbf{x}_t)\|^2}\nabla f^\dagger(\mathbf{x}_t)$. To negate this effect, we propose to adapt α_t, at the i-th client as $\alpha_{t,i} = \alpha(1 - \frac{\Lambda_{t,i}}{\|\nabla f^\dagger(\mathbf{x}_t)\|^2})$ and keep $\beta_{t,i} = \alpha$.

3.2 Worst Case: $C_{i,j} > 0$

In the worst case, we denote the catastrophic forgetting as $\Gamma_w(t)$ and $\Gamma_{i,w}(t)$ for the server and the i-th client respectively. The forgetting terms can be rewritten as follows:

$$\Gamma(t) = \Gamma_w(t) \tag{186}$$

$$= \frac{L\beta_t^2}{2}\Big[\sum_{i=1}^{N} p_i^2 \left\|\mathbf{a}_i\right\|^2 + \sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N} p_i p_j C_{i,j}\Big] - \beta_t(1-L\alpha_t)\sum_{i=1}^{N} p_i \Lambda_{t,i} \tag{187}$$

$$\overset{(a)}{\leq} \frac{L\beta_t^2}{2}\sum_{i=1}^{N} p_i^2 N \left\|\mathbf{a}_i\right\|^2 - \beta_t(1-L\alpha_t)\sum_{i=1}^{N} p_i \Lambda_{t,i} \tag{188}$$

$$= \sum_{i=1}^{N} p_i \Big[\frac{L\beta_t^2}{2} p_i N \left\|\mathbf{a}_i\right\|^2 - \beta_t(1-L\alpha_t)\Lambda_{t,i}\Big] \tag{189}$$

$$= \sum_{i=1}^{N} p_i \Gamma_{i,w}(t), \tag{190}$$

where (a) follows from the inequality $2p_ip_jC_{i,j} = 2\langle p_ia_i, p_ja_j\rangle \leq p_i^2\left\|a_i\right\|^2 + p_j^2\left\|a_j\right\|^2$ and we denote $\Gamma_{i,w}(t) = \frac{L\beta_t^2}{2}\sum\limits_{i=1}^{N} p_i N\left\|\mathbf{a}_i\right\|^2 - \beta_t(1-L\alpha_t)\sum\limits_{i=1}^{N}\Lambda_{t,i}$. Subsequently, taking expectation on both sides of (190), we obtain

$$\mathbb{E}[\Gamma_w(t)] \leq \sum_{i=1}^{N} p_i \mathbb{E}[\Gamma_{i,w}(t)]. \tag{191}$$

Similar to the previous section, we analyze and minimize the contribution of each client individually. In the worst case, for the i-th client, $\mathbb{E}[\Gamma_{i,w}(t)] = \mathbb{E}[\frac{L\beta_t^2}{2}p_iN\left\|\mathbf{a}_i\right\|^2 - \beta_t(1-L\alpha_t)\Lambda^i]$, and the optimal learning rate $\beta_{t,i}^* = \frac{(1-L\alpha_t)\Lambda_{t,i}}{LNp_i\left\|\mathbf{a}_i\right\|^2}$ and $\mathbb{E}[\Gamma_{i,w}^*] = -\frac{[(1-L\alpha_t)\Lambda_{t,i}]^2}{2LNp_i\left\|\mathbf{a}_i\right\|^2}$. Observe that the optimal choice of $\beta_{t,i}^*$ leads to $\mathbb{E}[\Gamma_{i,w}^*] < 0$. Similar to the average case, in the case of interference ($\Lambda_{t,i} \leq 0$), we propose $\alpha_{t,i} = \alpha(1 - \frac{\Lambda_{t,i}}{\left\|\nabla f^\dagger(\mathbf{x}_t)\right\|^2})$. Then clients can scale their learning rates by either $\alpha_{t,i} = \alpha$ or $\beta_{t,i} = \alpha$ at the end of every E local epochs.

3.3 Analysis of Adaptive Rates

From the previous sections, we see that clients having transference effect by adapting β_t to $\beta_{t,i}^*$ leads to minimized catastrophic forgetting in both, the average and the worst case. In this section, first, we show that in the interference case our proposed choice of adaptive $\alpha_{t,i}$ leads to lesser forgetting, that is $\mathbb{E}[\Gamma_{i,ad}(t)] \leq \mathbb{E}[\Gamma_i(t)]$. This holds for both the average and worst-case scenarios.

Next, we provide the proofs for Lem. 6 and Lem. 7.

Lemma 6. In the case of interference at the i-th client, for both the average and worst cases, adaptive rates $\alpha_{t,i} = \alpha(1 - \frac{\Lambda_{t,i}}{\left\|\nabla f^\dagger(\mathbf{x}_t)\right\|^2})$ and $\beta_{t,i} = \alpha$ lead to smaller forgetting, that is $\mathbb{E}[\Gamma_{i,ad}(t)] \leq \mathbb{E}[\Gamma_i(t)]$.

Proof. In the case of interference we have $\Lambda_{t,i} \leq 0$ and $\alpha_{t,i} = \alpha(1 - \frac{\Lambda_{t,i}}{\left\|\nabla f^\dagger(\mathbf{x}_t)\right\|^2})$. Using these we obtain

$$\alpha_{t,i} = \alpha_t(1 - \frac{\Lambda_{t,i}}{\left\|\nabla f^\dagger(\mathbf{x}_t)\right\|^2}) \geq \alpha_t \tag{192}$$

From this, we get

$$1 - L\alpha_{t,i} \leq 1 - L\alpha_t \tag{193}$$

Next, multiplying both sides by $-\beta_t\Lambda_{t,i}$, we get

$$-\beta_t(1-L\alpha_{t,i})\Lambda_{t,i} \leq -\beta_t(1-L\alpha_t)\Lambda_{t,i}, \tag{194}$$

Finally, adding $\frac{L\beta_t^2}{2}\sum_{i=1}^{N} p_i^2 \|\mathbf{a}_i\|^2$ or $\frac{L\beta_t^2}{2}\sum_{i=1}^{N} p_i^2 N \|\mathbf{a}_i\|^2$ on both sides based on the average or worst case it leads to $\mathbb{E}[\Gamma_{i,ad}(t)] \leq \mathbb{E}[\Gamma_i(t)]$.

Lemma 7. Client-wise adaptive rates lead to improved forgetting, $\mathbb{E}[\Gamma_{ad}(t)] \leq \mathbb{E}[\Gamma(t)]$.

Proof. Suppose that out of N clients, for r clients we observe interference($\Lambda_{t,i} \leq 0$), and for the rest $(N-r)$ clients, we observe transference ($\Lambda_{t,i} > 0$). Without loss of generality, consider the first r clients are interfering, that is $\Lambda_{t,i} \leq 0$ for $i \in \{1, 2, \cdots, r\}$ and $\Lambda_{t,i} > 0$ for $i \in \{r+1, r+2, \cdots, N\}$. Hence, we obtain

$$\mathbb{E}[\Gamma_{ad}(t)] = \sum_{i=1}^{N} p_i \mathbb{E}[\Gamma_{i,ad}(t)] \tag{195}$$

$$= \sum_{i=1}^{r} p_i \mathbb{E}[\Gamma_{i,ad}(t)] + \sum_{i=r+1}^{N} p_i \mathbb{E}[\Gamma_{i,ad}(t)] \tag{196}$$

$$\overset{(a)}{\leq} \sum_{i=1}^{r} p_i \mathbb{E}[\Gamma_i(t)] + \sum_{i=r+1}^{N} p_i \mathbb{E}[\Gamma_i^*(t)] \tag{197}$$

$$\overset{(b)}{\leq} \sum_{i=1}^{r} p_i \mathbb{E}[\Gamma_i(t)] + \sum_{i=r+1}^{N} p_i \mathbb{E}[\Gamma_i(t)] \tag{198}$$

$$= \sum_{i=1}^{N} p_i \mathbb{E}[\Gamma_i(t)] \tag{199}$$

$$= \mathbb{E}[\Gamma(t)], \tag{200}$$

where (a) follows from Lem. 6 and noting that for both the average and worst cases, $\mathbb{E}[\Gamma_{i,ad}(t)] = \mathbb{E}[\Gamma_i^*(t)]$, and (b) follows from $\mathbb{E}[\Gamma_i^*(t)] \leq \mathbb{E}[\Gamma_i(t)]$ for $i \in \{r+1, r+2, \cdots, N\}$. This proves that for both the average and the worst cases, using adaptive learning rates is better than constant learning rates.

4 ADDITIONAL EXPERIMENTS

In this section, we present additional experimental results. Unless specified otherwise, we run all tasks for 20 communication rounds each, allowing each client to perform 2 epochs of local updates. We consistently use a federated setup with 5 clients, maintaining a 100% client participation rate. For Split-CIFAR10 and Split-CIFAR100, we use a mini-batch size of 128, whereas for Split-TinyImageNet, the mini-batch size is set to 32. In this work, the theory of C-FLAG applies to training using gradient descent (GD). However, employing mini-batch GD is a common practice in machine learning to reduce the computational load on edge devices. For this, we adopt the ADAM optimizer for every local client. We use a class-balanced ring buffer memory with an initial size of 400 samples per task for each client while sampling 200 data points from the buffer at each step. Please note that the values in all the plots in this section and the main manuscript correspond to results from a single seed, while the values in the table represent averages over three seeds. The details of the hyperparameters used are provided in Table 5. We utilised the Avalanche library Carta et al. (2023) to generate different task sequences from the Split-CIFAR10, Split-CIFAR100, and Split-TinyImageNet datasets, which consist of 5, 5, and 10 tasks, respectively. The value of L, which is used for calculating adaptive learning rates, is chosen to be 5. We trained our models on NVIDIA A100-PCIE-40GB GPU. In the following subsections, we compare the performance of the proposed C-FLAG with the baselines presented in the main manuscript for the task-incremental and class-incremental settings.

Table 5: Training hyperparameters for different datasets.

Hyperparameter	Split-CIFAR10	Split-CIFAR100	Split-TinyImageNet
Learning rate	0.0001	0.0001	0.0001
Seed	1234, 1235, 1236	1234, 1235, 1236	1234, 1235, 1236

Table 6: **Task incremental**: Performance metrics on different datasets for non-IID and IID settings with 5 clients and 5, 5, 10 tasks, respectively. *Acc* and *Forget* denote average classification accuracy and average forgetting.

	Split-CIFAR10		Split-CIFAR100		TinyImageNet	
	Acc$_{(\uparrow)}$	Forget$_{(\downarrow)}$	Acc$_{(\uparrow)}$	Forget$_{(\downarrow)}$	Acc$_{(\uparrow)}$	Forget$_{(\downarrow)}$
Non-IID Setting						
Finetune-FL	54.10 ± 5.78	6.08 ± 8.60	38.66 ± 1.60	20.87 ± 0.82	23.72 ± 1.82	23.85 ± 0.92
EWC-FL	53.55 ± 4.56	5.22 ± 7.46	38.83 ± 1.33	19.20 ± 1.33	26.94 ± 1.91	20.29 ± 0.26
NCCL-FL	63.35 ± 7.62	12.37 ± 0.83	32.25 ± 0.95	29.49 ± 1.46	28.49 ± 1.19	$\mathbf{8.73 \pm 0.28}$
Erg-FL	$\mathbf{79.11 \pm 7.9}$	8.32 ± 7.42	31.25 ± 1.05	32.40 ± 1.69	21.90 ± 0.96	19.67 ± 0.98
C-FLAG	65.02 ± 6.22	$\mathbf{5.82 \pm 0.9}$	$\mathbf{43.47 \pm 1.64}$	$\mathbf{16.76 \pm 1.85}$	$\mathbf{28.63 \pm 0.91}$	9.52 ± 0.89
IID Setting						
Finetune-FL	72.64 ± 0.98	26.51 ± 1.31	49.82 ± 0.51	30.00 ± 1.32	30.17 ± 0.69	31.91 ± 0.22
EWC-FL	75.98 ± 5.10	22.34 ± 4.19	50.48 ± 1.10	30.16 ± 0.81	33.18 ± 0.99	28.38 ± 0.48
NCCL-FL	83.43 ± 5.38	17.10 ± 4.64	41.65 ± 0.58	39.23 ± 0.62	28.93 ± 0.73	9.51 ± 0.98
FedTrack	79.86 ± 7.16	17.62 ± 4.46	29.85 ± 3.32	18.11 ± 3.15	33.32 ± 1.43	28.57 ± 4.04
Erg-FL	84.01 ± 9.96	16.50 ± 11.57	38.70 ± 1.35	43.66 ± 1.48	31.25 ± 1.05	32.40 ± 1.69
C-FLAG	$\mathbf{89.28 \pm 4.98}$	$\mathbf{7.23 \pm 5.14}$	$\mathbf{66.85 \pm 0.77}$	$\mathbf{7.30 \pm 0.81}$	$\mathbf{44.3 \pm 0.71}$	$\mathbf{7.65 \pm 0.78}$

Table 7: **Class incremental**: Performance metrics on different datasets for non-IID and IID settings with 5 clients and 5, 5, 10 tasks, respectively. *Acc* and *Forget* denote average classification accuracy and average forgetting.

	Split-CIFAR10		Split-CIFAR100	
	Acc$_{(\uparrow)}$	Forget$_{(\downarrow)}$	Acc$_{(\uparrow)}$	Forget$_{(\downarrow)}$
Non-IID Setting				
Finetune-FL	14.18 ± 3.22	$\mathbf{2.31 \pm 4.82}$	16.24 ± 1.03	45.50 ± 1.80
EWC-FL	12.92 ± 2.53	6.12 ± 5.20	17.25 ± 1.29	40.26 ± 2.09
LwF-FL	12.45 ± 3.54	5.62 ± 5.17	22.38 ± 3.66	31.07 ± 2.58
iCARL-FL	15.84 ± 4.23	50.75 ± 12.95	21.40 ± 1.53	32.46 ± 2.23
TARGET	13.68 ± 3.48	9.73 ± 7.04	$\mathbf{23.22 \pm 1.93}$	27.65 ± 5.01
NCCL-FL	14.42 ± 1.22	76.93 ± 2.46	9.96 ± 0.40	47.76 ± 1.05
FedTrack	12.21 ± 1.04	61.89 ± 5.16	3.53 ± 0.18	$\mathbf{17.83 \pm 0.29}$
C-FLAG	$\mathbf{17.06 \pm 2.63}$	43.04 ± 13.27	13.94 ± 1.13	51.49 ± 0.72
IID Setting				
Finetune-FL	24.25 ± 4.89	55.54 ± 13.09	20.04 ± 1.30	65.67 ± 1.61
EWC-FL	25.28 ± 4.07	57.04 ± 5.44	22.20 ± 1.00	62.69 ± 1.35
LwF-FL	43.01 ± 8.21	32.07 ± 16.54	$\mathbf{35.86 \pm 0.98}$	35.72 ± 0.81
iCARL-FL	$\mathbf{47.98 \pm 2.43}$	49.92 ± 3.16	24.13 ± 1.72	48.89 ± 1.47
TARGET	20.08 ± 3.12	21.56 ± 10.38	31.04 ± 2.94	26.75 ± 5.61
NCCL-FL	16.91 ± 0.50	93.15 ± 2.10	12.40 ± 0.39	60.61 ± 0.83
FedTrack	14.54 ± 1.33	81.48 ± 3.64	5.49 ± 0.26	$\mathbf{22.02 \pm 1.73}$
C-FLAG	26.37 ± 3.80	$\mathbf{7.66 \pm 7.55}$	31.68 ± 1.54	47.97 ± 2.64

Tables 6 and 7 present the average accuracy performance for both task-incremental and class-incremental settings, along with standard deviation values. Our findings consistently show that the proposed approach effectively mitigates catastrophic forgetting while achieving higher average accuracy than the baselines in the task-incremental setup. In the class-incremental setup, the results are comparable overall, with our approach outperforming the baselines in select cases.

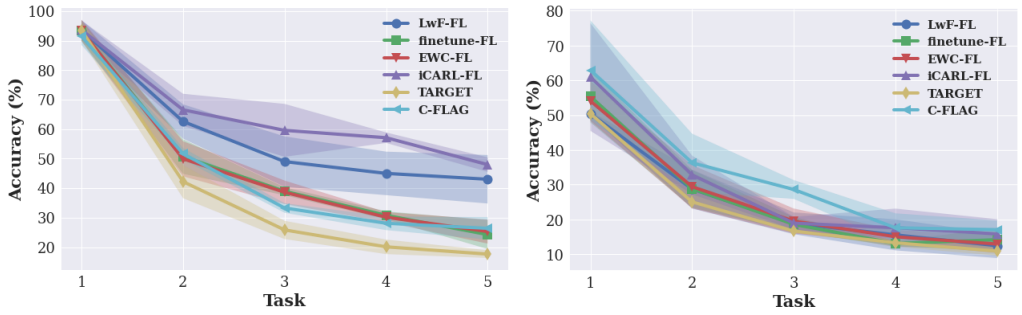

Figure 6: Average accuracy plots on Split-CIFAR10 (Left: IID, Right: non-IID) in the class-incremental setup.

Fig. 6 illustrates the average accuracy performance of our method on the Split-CIFAR10 dataset in the class-incremental setup, demonstrating its superiority over the baselines in the non-IID setting. The baselines considered include class-incremental techniques such as TARGET, iCARL-FL, and LwF-FL.

5 ADDITIONAL ABLATION STUDIES

In this section, we provide additional ablation studies on the task-incremental setting.

5.1 Varying Memory Sizes

We provide the results for varying memory sizes (m_0) in Fig. 7. We have analysed 6 combinations of memory sample sizes on Split-CIFAR10 and Split-CIFAR100 datasets for both IID and non-IID settings. Note that we have fixed the sampling size as half of the corresponding memory size. It is observed that varying memory size affects the final average accuracy and forgetting. The general trend indicates an improvement in accuracy as we increase the size of the memory buffer. This trend is more evident in the non-IID partitioning scenario. The Split-CIFAR100 dataset shows greater sensitivity to memory size variations compared to Split-CIFAR10 in both IID and non-IID settings. This is likely due to the higher complexity and larger number of classes in CIFAR100, which demand more memory for mitigating catastrophic forgetting.

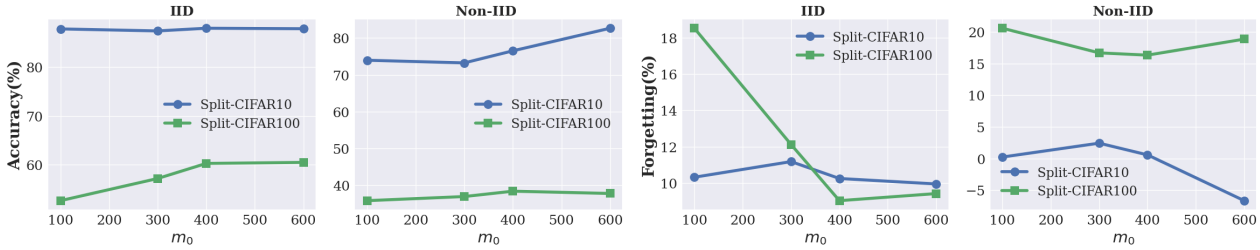

Figure 7: Average accuracies (left) and forgetting (right) for varying memory sizes (m_0).

From Fig. 7, we observe that in the non-IID Split-CIFAR10 setting, forgetting becomes slightly negative for a larger memory size (600), indicating that a large memory size reinforces patterns that encourage improved transference. Note that the negative value of forgetting indicates that the current model performs extremely well. We refer to this phenomenon as transference in Sec. 5.

5.2 Varying Number of Clients

Table 8: Varying number of clients for the Split-CIFAR10 and Split-CIFAR100 datasets

Dataset	Clients-50	Clients-25	Clients-15	Clients-10	Clients-5
Split-CIFAR10 (IID)	(90.67,2.39)	(90.77,3.53)	(91.05,4.28)	(90.52,5.22)	(87.89,10.25)
Split-CIFAR100 (Non-IID)	(35.20,2.28)	(32.60,4.27)	(35.48,6.38)	(35.95,8.58)	(38.43,16.33)

Our proposed C-FLAG framework is agnostic to the number of clients and can be applied in cross-silo and cross-device settings. In the main manuscript, we have demonstrated the effect of different numbers of clients in Fig. 5 (left) of Sec. 7, where we observed that the proposed technique is robust to the change in the number of clients. In other words, varying the number of clients does not deteriorate the continual learning abilities of the proposed method. Here, we are extending the analysis by accommodating 25 and 50 clients to demonstrate that C-FLAG can be implemented in cross-device settings as well. For instance, in the case of IID partitioning of the Split-CIFAR10 dataset and non-IID partitioning of the Split-CIFAR100 dataset, we obtain the average accuracies and forgetting as mentioned in Table 8.