# Theoretical Convergence Guaranteed Resource-Adaptive Federated Learning with Mixed Heterogeneity

Yangyang Wang
Shandong University
Qingdao, China

Xiao Zhang*
Shandong University
Qingdao, China

Mingyi Li
Shandong University
Qingdao, China

Tian Lan
George Washington University
Washington, United States

Huashan Chen
Chinese Academy of Sciences
Beijing, China

Hui Xiong
Hong Kong University of Science and Technology
Guangzhou, China

Xiuzhen Cheng
Shandong University
Qingdao, China

Dongxiao Yu*
Shandong University
Qingdao, China

## ABSTRACT

In this paper, we propose an adaptive learning paradigm for resource-constrained cross-device federated learning, in which heterogeneous local submodels with varying resources can be jointly trained to produce a global model. Different from existing studies, the submodel structures of different clients are formed by arbitrarily assigned neurons according to their local resources. Along this line, we first design a general resource-adaptive federated learning algorithm, namely *RA-Fed*, and rigorously prove its convergence with asymptotically optimal rate $O(1/\sqrt{\Gamma^* TQ})$ under loose assumptions. Furthermore, to address both *submodels heterogeneity* and *data heterogeneity* challenges under *non-uniform training*, we come up with a new server aggregation mechanism *RAM-Fed* with the same theoretically proved convergence rate. Moreover, we shed light on several key factors impacting convergence, such as minimum coverage rate, data heterogeneity level, submodel induced noises. Finally, we conduct extensive experiments on two types of tasks with three widely used datasets under different experimental settings. Compared with the state-of-the-arts, our methods improve the accuracy up to 10% on average. Particularly, when submodels jointly train with 50% parameters, *RAM-Fed* achieves comparable accuracy to *FedAvg* trained with the full model.

## CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Computer systems organization → Distributed architectures**.

## KEYWORDS

Federated learning, Limited resources, Heterogeneity

**ACM Reference Format:**
Yangyang Wang, Xiao Zhang, Mingyi Li, Tian Lan, Huashan Chen, Hui Xiong, Xiuzhen Cheng, and Dongxiao Yu. 2023. Theoretical Convergence Guaranteed Resource-Adaptive Federated Learning with Mixed Heterogeneity. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3580305.3599521

## 1 INTRODUCTION

In recent years, with the promulgation of kinds of data regulations such as GDPR and individuals' awareness of privacy data protection, federated learning has drawn rapidly growing interest from both academia and industry. The classical federated learning is centralized with a parameter server, in which model parameters can be learned from local dispersed datasets and then sent to the server for aggregation without sharing local data. Particularly, with the rapid increase of the volume of data generated by massive mobile and IoT devices [11–13, 24], the cross-device federated learning [8, 22] has become a popular distributed computing paradigm.

In real-world cross-device federated learning scenarios, mobile devices are usually equipped with limited resources for computation and communication which seriously restrict the convergence performance of the federated learning algorithms. It would be difficult and unaffordable for the resource-constrained clients to run the full model for coordination in federated learning, especially for the arising large models like ChatGPT [6]. Therefore, kinds of technologies such as model compression [19], model pruning [5], splitting learning [20] have been introduced to reduce model size or communication cost to facilitate cross-device federated learning feasible. For instance, PruneFL [5] selects the important parameters to train with adaptive pruning. SplitFL [20] combined splitting learning with federated learning to split the full model into smaller parts and train them on a server, and distributed clients separately.

In this work, we consider a novel learning paradigm in resource-limited federated learning. Different from the traditional federated learning in which each client needs to update the full model in each

*Corresponding authors. Email: {xiaozhang, dxyu} @sdu.edu.cn


*KDD '23, August 6–10, 2023, Long Beach, CA, USA*

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599521

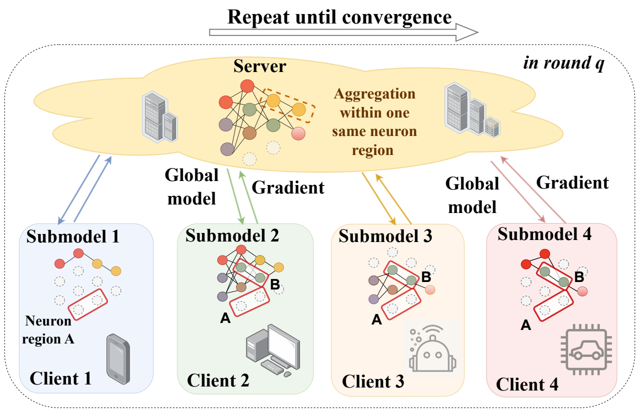



**Figure 1: The novel learning paradigm in resource-limited federated learning. Submodels are formed by arbitrarily assigned neuron regions for clients according to local resources. Training round $q$ is depicted and the training repeats until convergence.**

global epoch, different clients can train different submodels according to their own resource constraints. Thus the resource-adaptive learning paradigm aims to *train heterogeneous local submodels with varying resources and still produce a single global inference model.* Recently, independent subnet training (IST) [26] belongs to this kind of learning paradigm with a strong assumption that hidden neurons are all random uniformly assigned to disjoint computing nodes. Literature [30] achieves this goal by adaptively pruning the shared global full model and establishing sufficient conditions for the heterogeneous submodels to converge. In this work, we consider more general cases without these strong assumptions where existing works would become special cases of our proposed learning paradigm. An example is shown in Fig. 1, the submodel structures of different clients are formed by **arbitrarily assigned neurons** according to their local resources. As the training continues, the submodel structure within the same client could also change continuously due to the changing resources.

Thus in order to achieve this goal, several non-trivial challenges arise. (1) **Submodel heterogeneity.** The arbitrary submodels training induced uncontrollable noises compared with the full model training, which would affect the performance of federated models. (2) **Non-uniform training.** Due to the arbitrarily constructed submodels, it is obvious that not all the neurons of the whole network can be trained in each round. As shown in the depicted Fig. 1, neuron region A is never trained by any clients in this training epoch. The insufficient training would make the convergence of the federated model difficult, which has never been addressed by existing IST and literature [30]. (3) **Data heterogeneity** denotes one same neuron region might be trained in different clients whose data distributions could be not independent and identically distributed (data heterogeneity) [27, 28]. Taking neuron region B in Fig. 1 as an example, which is trained by submodel 2, 3, 4 simultaneously in a certain training round. Especially mixing with submodels heterogeneity, different neuron regions of the full models might be trained by different subsets of clients, which further exacerbates slow convergence [23] and has also been ignored by existing IST and literature [30]. (4) **Theoretical guarantee.** Under the arbitrarily assigned neurons training paradigm, arising with the *submodel heterogeneity*, *non-uniform training*, and *data heterogeneity* challenges, how to theoretically guarantee the convergence rate of our proposed algorithm is unprecedentedly challenging. Little is known about whether such algorithms can converge like standard federated learning methods.

Along this line, we first propose a general *r*esource-*a*daptive *fed*erated learning framework, namely *RA-Fed*, under arbitrary neuron assignments. Within every training round, the server sends the global model to all clients, different clients leverage adaptive online masks to train heterogeneous submodels with varying neuron regions, and then the server receives and aggregates accumulated local updates for each neuron. We give detailed convergence analysis with loose assumptions (e.g., remove bounded gradient and assume the biased mask and compression), which can achieve asymptotically optimal rate $O(1/\sqrt{\Gamma^* TQ})$, where $Q$ is the number of communication rounds, $T$ is the number of local iterations and $\Gamma^*$ is the minimum coverage rate defined in Section 3. Moreover, to mitigate the effects of both *submodel heterogeneity* and *data heterogeneity* under *non-uniform training*, we further proposed *RAM-Fed* with a new server aggregation mechanism, in which the server stores the latest updates for different regions of global model in each client, and reuses it as an approximation for current regions update. We also prove that *RAM-Fed* can achieve the same convergence rate with *RA-Fed*. Finally, extensive experiments are conducted on two widely used datasets, both *RA-Fed* and *RAM-Fed* demonstrating its superiority over other baselines. Our source code is available on github[1]. The main contributions of this paper are summarized as follows:

- To the best of our knowledge, we are the first to propose the arbitrarily assigned neurons based resource-adaptive federated learning paradigm. The heterogeneous local submodels with varying resources can be jointly trained to produce a single global model.
- We design a general resource-adaptive learning algorithm *RA-Fed* under arbitrary neuron assignments. We give detailed convergence analysis with loose assumptions to prove *RA-Fed* can achieve asymptotically optimal rate $O(1/\sqrt{\Gamma^* TQ})$, which can achieve speedup with coverage level $\Gamma^*$.
- Furthermore, in order to mitigate the effects of both *submodel heterogeneity* and *data heterogeneity* under *non-uniform training*, we further propose *RAM-Fed* with a new server aggregation mechanism. We also theoretically prove the *RAM-Fed* can also converge with $O(1/\sqrt{\Gamma^* TQ})$.
- Based on the theoretical convergence analysis, we investigate several key factors impacting convergence rate, such as the minimum coverage rate $\Gamma^*$, data heterogeneity level, submodel induced noises.
- We perform extensive experiments on two different tasks with three datasets by comparing with state-of-the-art algorithms under different experimental settings. Our algorithms improve 10% accuracy compared with the optimal results in baselines. Particularly, *RAM-Fed* with 50% model achieves comparable accuracy to *FedAvg* trained with the full model.

[1] https://github.com/wyy-123-xyy/RA-Fed

In summary, the proposed novel resource-adaptive learning paradigm provides a new insight and rigorously theoretical guarantee for the real-world deployment of arising large models on massive resource-limited devices. Moreover, existing studies would become special cases of our learning paradigm. When $\Gamma^* = N$, *RAM-Fed* achieves the same convergence rate $O(1/\sqrt{NTQ})$ as the vanilla *FedAvg* [16, 25]. When $\Gamma^* = 1$, *RAM-Fed* achieves the same convergence rate $O(1/\sqrt{TQ})$ as *OAP*[2] [30].

## 2 RELATED WORK

In traditional federated learning [22], FedAvg [16] is the widely used aggregation algorithm, which achieves $O(1/\sqrt{NTQ})$ convergence rate with training full global model in each client. However, with the popularity of large models, it would be difficult for devices with limited resources to run the full model under classic federated learning. In recent years, kinds of approaches [14, 15, 21, 26, 30] has been proposed to address the resource-constrained problem. For example, literature [30] focuses on training heterogeneous models with online global model pruning and achieves convergence with strong assumptions (e.g. bounded gradient). IST [26] is proposed by decomposing the fully connected neural network into multiple subnetworks with the same depth. HeteroFL [1] designs a stable framework to train heterogeneous fixed sub-network without theoretical convergence analysis. In addition, to address the limited communication problem, several works [17–19] are proposed. DGC [10] combines gradient sparsity and multiple optimization technologies to greatly reduce communication costs with comparable accuracy. CHOCO-SGD [9] is proposed to realize arbitrary compression level with theoretical convergence on non-convex assumption. Different from any existing studies, in this work, we consider more general cases by proposing a resource-adaptive federated learning paradigm under arbitrarily assigned neurons. We also demonstrate theoretical convergence analysis for the proposed algorithms, existing studies would become special cases of our learning paradigm.

## 3 PRELIMINARIES

Given the resource-constrained cross-device federated learning paradigm, there exist $N$ clients, and all clients collaboratively learn a single global inference model with parameter $\theta$. The goal is to optimize the empirical risk minimization like traditional setting:

$$min_{\theta\in\mathbb{R}^d} F(\theta) := \frac{1}{N}\sum_{n=1}^{N} F_n(\theta) \tag{1}$$

where $F_n(\theta) := \mathbb{E}_{\xi_n \sim D_n}[F_n(\theta, \xi_n)]$ is the local loss function of client $n$ on local dataset $D_n$.

Definition 1. Neuron regions. *The global inference model contains $|\mathcal{K}|$ neuron regions with varying number of neurons. In extreme cases, each model neuron can be regarded as a separate region.*

In our proposed resource-adaptive learning paradigm, due to the arbitrarily assigned neurons, each client can train a submodel with multiple varying neuron regions according to their own online heterogeneous resource constraints. Specifically, the adaptive online mask strategy is utilized to obtain the submodel for each client. For instance, $\theta_q$ is defined as the initial global model in round $q$, $m_{q,n}$ denotes the mask generated by client $n$ in round $q$. Thus, $\theta_q \odot m_{q,n}$ defines the submodel with multiple neuron regions of client $n$ in round $q$. Comparing with the full model training, the submodels training induced noises are assumed as follows:

Assumption 1. *Mask-induced noises: Existing $w_1 \in [0, 1)$, the mask-induced noise on client n and any q is bounded:*

$$\|\theta_q - \theta_q \odot m_{q,n}\| \le w_1^2 \|\theta_q\|^2 \tag{2}$$

Since every submodel is constructed by arbitrarily multiple neuron regions, we let $S_q$ be the trained neuron regions set in round $q$. It is worth noting that $S_q = \mathcal{K}$ denotes all neuron regions can be trained in round $q$, while $S_q \subseteq \mathcal{K}$ denotes only parts of neuron region are trained in round $q$, which is the main difference from any existing learning paradigms. Then we let $\mathcal{N}_q^i$ be the clients set whose submodels train neuron region $i \in S_q$ in round $q$ and $\Gamma_q^i$ be the number of clients in $\mathcal{N}_q^i$.

For algorithm design and convergence analysis, we define a crucial indicator namely minimum coverage rate $\Gamma^*$ as follows:

$$\Gamma^* = \min_{q,i} \Gamma_q^i, i \in S_q, \forall q \tag{3}$$

$\Gamma^*$ measures the minimum number of submodels training the corresponding neuron region $i$ in all rounds. Intuitively, the larger $\Gamma^*$ indicates the neuron region $i$ can be trained sufficiently but might face higher data heterogeneity.

## 4 ALGORITHM DESIGN

In this section, we design a novel resource-adaptive learning paradigm in cross-device federated learning scenarios. Due to the limited and continuously changing resources in device clients, different submodels can be trained with arbitrarily varying neuron regions

**Table 1: Frequently used notations**

| Notations | Descriptions |
|---|---|
| $\|\cdot\|$ | the vector $\ell_2$ norm or the matrix spectral norm depending on the argument |
| $\mathcal{K}$ | the set of all neuron regions |
| $S_q$ | the trained neuron regions set in round $q$ |
| $\lvert S_q \rvert$ | the number of trained neuron regions in round $q$ |
| $S^*$ | minimum number of trained neuron regions:$S^* = \min_q \lvert S_q\rvert, \forall q$ |
| $N_q^i$ | the set of clients training neuron region $i$ in round $q$ |
| $\Gamma_q^i$ | $\Gamma_q^i = \lvert N_q^i \rvert$ the number of clients in $N_q^i$ |
| $\Gamma^*$ | minimum coverage rate: $\Gamma^* = \min_{q,i} \Gamma_q^i, i \in S_q, \forall q$ |
| $\Delta_{q,n}$ | the accumulated local updates from client $n$ on itself submodel in round $q$ |
| $\Delta_{q,n}^i$ | the accumulated local updates from client $n$ on neuron region $i$ in round $q$ |
| $m_{q,n}$ | the mask of client $n$ in round $q$ |
| $u_{q+1,n}^i$ | the latest update from client $n$ on neuron region $i$ in round $q$ |
| $\theta_q^i$ | the neuron region $i$ of global model in round $q$ |
| $\mathcal{C}(\cdot)$ | the arbitrary compressor |
| $\gamma$ | the step size (learning rate) |

[2]In our paper, $TQ$ denotes the total number of SGDs. While in some related work, only one notation is utilized to represent the total number of SGDs in convergence rate.

**Algorithm 1:** RA-Fed

**Initialize:** subdataset $\mathcal{D}_n$ on $N$ clients, mask policy $P$, $\theta_1$ **for** $q = 1$ *to* $Q$ **do**
  **for** $n = 1$ *to* $N$ *(all workers in parallel)* **do**
    Generate mask $m_{q,n} = P(\complement(\theta_q), n)$
    Generate submodel $\theta_{q,n,0} = \complement(\theta_q) \odot m_{q,n}$
    *# Update local submodel with multiple neuron regions:*
    **for** *epoch* $t = 1$ *to* $T$ **do**
      $\theta_{q,n,t} = \theta_{q,n,t-1} - \gamma \nabla F_n(\theta_{q,n,t-1}, \xi_{n,t-1}) \odot m_{q,n}$
    **endfor**
    $\Delta_{q,n} = \frac{\theta_{q,n,0} - \theta_{q,n,T}}{\gamma}$
  **endfor**
  *# Update all neuron regions of global model:*
  **for** *region* $i = 1$ *to* $K$ **do**
    Find $N_q^i = \{n : m_{q,n}^i = \mathbf{1}\}$
    **if** $\Gamma_q^i = 0$ **then**
      Update $\theta_{q+1}^i = \theta_q^i$
    **else**
      Update $\theta_{q+1}^i = \theta_q^i - \gamma \frac{1}{\Gamma_q^i} \sum_{n \in N_q^i} \Delta_{q,n}^i$
    **end**
  **endfor**
  $\theta_{q+1} = \sum_{i=1}^{K} \theta_{q+1}^i$
**endfor**

according to their own resource constraints in our work. The non-uniform training leads to that not all the neuron regions of the full model can be trained in each round. In addition, one same neuron region might be trained by different clients in each round, which might face high data heterogeneity. Therefore, we first propose a general resource-adaptive learning algorithm, namely *RA-Fed* to address the arising challenges. Moreover, to further mitigate the effects of both submodel heterogeneity and data heterogeneity, we propose *RAM-Fed* with a new server aggregation mechanism. The details are shown as follows.

## 4.1 RA-Fed Algorithm

In order to achieve resource-adaptive learning, we propose the *RA-Fed* algorithm, whose training process is shown in Algorithm 1. First, the server sends the globally full model to all clients, different clients leverage adaptive online masks to train heterogeneous submodels with varying neuron regions, and then the server receives and aggregates accumulated local updates for each neuron. The details of *RA-Fed* in the $q$-th round are described as follows:

- Mask generation: Online mask $m_{q,n}$ would be generated according to its resource constraints within each client.
- Submodel construction: Each client $n$ leverage adaptive online mask $m_{q,n}$ to generate heterogeneous local submodel $\theta_{q,n,0}$ with multiple neuron regions.
- Local submodel update: Each client $n$ calculates local gradients and update local submodel with $T$ iterations: $\theta_{q,n,t} = \theta_{q,n,t-1} - \gamma \nabla F_n(\theta_{q,n,t-1}, \xi_{n,t-1}) \odot m_{q,n}$.
- Uploading local updates: Each client $n$ calculates accumulated local updates on local submodel: $\Delta_{q,n} = \frac{\theta_{q,n,0} - \theta_{q,n,T}}{\gamma}$.
- Neuron regions aggregation: For each neuron region $i$, the server calculates the number of clients which local submodels contains neuron region $i$: $\Gamma_q^i$. 1) If $\Gamma_q^i = 0$, neuron region $i$ in round $q$ is not trained: Update $\theta_{q+1}^i = \theta_q^i$. 2) Otherwise, the neuron region $i$ is trained by at least one client: Update $\theta_{q+1}^i = \theta_q^i - \gamma \frac{1}{\Gamma_q^i} \sum_{n \in N_q^i} \Delta_{q,n}^i$.
- Full global model generation: The full model is constructed based on all neuron regions: $\theta_{q+1} = \sum_{i=1}^{K} \theta_{q+1}^i$.

The convergence of the *RA-Fed* algorithm is theoretically proved in Sec. 5.1.

## 4.2 RAM-Fed Algorithm

Except for the submodel heterogeneity and the non-uniform training, data heterogeneity also seriously restricts the convergence and performance of the proposed resource-adaptive learning algorithms. The core challenge of the mixed heterogeneity is that one neuron region might be trained by partial clients simultaneously, or even not be trained in one round due to arbitrariness. Obviously, the key is to ensure each neuron region can be updated by all clients in each round. Inspired by the idea of memorized latest updates [2, 4], we further propose the *RAM-Fed* algorithm with a new server aggregation mechanism to further mitigate the effects of the mixed heterogeneity.

In *RAM-Fed*, the server stores the latest updates from all clients on each neuron region. Specifically, in Algorithm 2, $\Delta_{q,n}^i$ represents the current updates from client $n$ ($n \in N_q^i$) on neuron region $i$ in round $q$. To maintain the latest updates from client $n$ ($n \in N$), after each round, we perform the following step for all clients:

$$u_{q+1,n}^i = \begin{cases} \Delta_{q,n}^i & if \quad n \in N_q^i \\ u_{q,n}^i & if \quad n \notin N_q^i \end{cases} \tag{4}$$

By this way, $u_{q+1,n}^i$ maintains the latest updates from all clients on neuron region $i$ in round $q$.

Then, when updating the neuron region $i$, we can use the latest updates $u_{q,n}^i$ ($n \in N$) in round $q-1$ and current updates $\Delta_{q,n}^i$ ($n \in N_q^i$) in round $q$ to compute an approximation aggregated update $\mathrm{v}_q^i$ from all clients. Specifically, in round $q$, if the neuron region $i$ is not trained, then the neuron region $i$ will be updated by the average latest update from all clients: $\mathrm{v}_q^i = \frac{1}{N} \sum_{n=1}^{N} u_{q,n}^i$. Otherwise, if the neuron region $i$ is trained by clients $N_q^i$, then the neuron region $i$ will be updated by $\Delta_{q,n}^i$ ($n \in N_q^i$) and $u_{q,n}^i$ ($n \notin N_q^i$): $\mathrm{v}_q^i = \frac{1}{N} \sum_{n=1}^{N} u_{q,n}^i + \frac{1}{\Gamma_q^i} \sum_{n \in N_q^i} (\Delta_{q,n}^i - u_{q,n}^i)$. Noted that we give higher weight to current client updates $\Delta_{q,n}^i$ ($n \in N_q^i$) as compared to previous client updates $u_{q,n}^i$ ($n \notin N_q^i$) following FedVARP [4]. Thus, this can correct the update bias (only partial clients update error compared with all clients update) in neuron regions using the latest updates from all clients in each round. The convergence of the *RAM-Fed* algorithm is theoretically proved in Sec. 5.2.

**Algorithm 2:** RAM-Fed

**Initialize:** subdataset $\mathcal{D}_n$ on $N$ clients, mask policy $P$, $\theta_1$, $u_{1,n}$, $(n = 1, ..., N)$
**for** $q = 1$ *to* $Q$ **do**
  **for** $n = 1$ *to* $N$ *(all workers in parallel)* **do**
    Generate mask $m_{q,n} = P(\mathcal{C}(\theta_q), n)$
    Generate submodel $\theta_{q,n,0} = \mathcal{C}(\theta_q) \odot m_{q,n}$
    *# Update local submodel with multiple neuron regions:*
    **for** *epoch* $t = 1$ *to* $T$ **do**
      $\theta_{q,n,t} = \theta_{q,n,t-1} - \gamma \nabla F_n(\theta_{q,n,t-1}, \xi_{n,t-1}) \odot m_{q,n}$
    **endfor**
    $\Delta_{q,n} = \frac{\theta_{q,n,0} - \theta_{q,n,T}}{\gamma}$
  **endfor**
  *# Update all neuron regions of global model:*
  **for** *region* $i = 1$ *to* $K$ **do**
    Find $N_q^i = \{n : m_{q,n}^i = \mathbf{1}\}$
    **if** $\Gamma_q^i = 0$ **then**
      Update $v_q^i = \frac{1}{N} \sum_{n=1}^{N} u_{q,n}^i$
    **else**
      Update $v_q^i = \frac{1}{N} \sum_{n=1}^{N} u_{q,n}^i + \frac{1}{\Gamma_q^i} \sum_{n \in N_q^i} (\Delta_{q,n}^i - u_{q,n}^i)$
    **end**
    $\theta_{q+1}^i = \theta_q^i - \gamma v_q^i$
    *# Store the latest update for each client:*
    **for** $n = 1$ *to* $N$ **do**
      $u_{q+1,n}^i = \begin{cases} \Delta_{q,n}^i & if \quad n \in N_q^i \\ u_{q,n}^i & if \quad n \notin N_q^i \end{cases}$
    **endfor**
  **endfor**
  $\theta_{q+1} = \sum_{i=1}^{K} \theta_{q+1}^i$
**endfor**

# 5 CONVERGENCE ANALYSIS

In the section, we show the convergence rate of our proposed *RA-Fed* and *RAM-Fed* algorithms. Firstly, we give some commonly used assumptions in federated learning:

Assumption 2. *Lipschitzian Condition: Every function $F_n(\cdot)$ is with L-Lipschitzian gradient: $\forall n \in [N], \theta, \varphi \in R^d$*

$$\|\nabla F_n(\theta) - \nabla F_n(\varphi)\| \leq L\|\theta - \varphi\| \tag{5}$$

Assumption 3. *Bounded compression: An operator $\mathcal{C} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a $w$-approximate compressor over $w_2$ for $w_2 \in (0, 1]$ if*

$$\mathbb{E}\|\mathcal{C}(\theta) - \theta\|^2 \leq w_2^2 \mathbb{E}\|\theta\|^2. \quad \forall \theta \in \Omega \tag{6}$$

Assumption 4. *Bounded variance: There exists $\sigma > 0$:*

$$\mathbb{E}_{\xi_{n,t} \sim D_n} \|\nabla F_n(\theta_{q,n,t}; \xi_{n,t}) - \nabla F_n(\theta_{q,n,t})\|^2 \leq \sigma^2, \quad \forall q, n, t \tag{7}$$

*$\sigma > 0$ bounds the variance of stochastic gradient.*

Assumption 5. *Bounded data heterogeneity level: There exists $\delta > 0$:*

$$\|\nabla F_n(\theta_q) - \nabla F(\theta_q)\|^2 \leq \delta^2 \tag{8}$$

*$\delta > 0$ bounds the effect of heterogeneous data.*

## 5.1 Convergence analysis of RA-Fed

Lemma 1. *Deviation of local submodel and global model: Let all assumptions hold.*

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\|\theta_{q,n,t-1} - \theta_q\|^2 &\leq 4\gamma^2 T \sigma^2 + 32\gamma^2 T^2 \delta^2 \\ &+ 32\gamma^2 T^2 \sum_{i \in S_q} \mathbb{E}\|\nabla F^i(\theta_q)\|^2 + 4w^2 \mathbb{E}\|\theta_q\|^2 \end{aligned} \tag{9}$$

Lemma 1 bounds the difference between local submodel and global model. It indicates that the effects of local submodel training: $\theta_{q,n,t-1} - \theta_{q,n,0}$ and mask and compression error: $\theta_{q,n,0} - \theta_q$. Note that $\theta_{q,n,0} - \theta_q$ can be split into mask error $\mathcal{C}(\theta_q) \odot m_{n,q} - \mathcal{C}(\theta_q)$ and compression error $\mathcal{C}(\theta_q) - \theta_q$.

Theorem 1. *Let all assumptions hold. Suppose that the step size $\gamma$ satisfies the following relationships:*

$$\begin{cases} 8\gamma^2 L^2 T^2 \leq \frac{1}{2} \Rightarrow \gamma \leq \frac{1}{4LT} \\ 32\gamma^2 T^2 \frac{N}{\Gamma^*} L^2 \leq \frac{1}{8} \Rightarrow \gamma \leq \frac{\sqrt{\Gamma^*}}{16TL\sqrt{N}} \\ 96L^3 \gamma^3 T^3 \frac{N}{\Gamma^*} \leq \frac{1}{8} \Rightarrow \gamma \leq \frac{(\Gamma^*)^{\frac{1}{3}}}{768^{\frac{1}{3}} LTN^{\frac{1}{3}}} \\ \frac{3}{2} L\gamma T \leq \frac{1}{8} \Rightarrow \gamma \leq \frac{1}{12TL} \end{cases}$$

*Therefore, the step size $\gamma$ is defined as:*

$$0 \leq \gamma \leq \min\{\frac{1}{12TL}, \frac{\sqrt{\Gamma^*}}{16TL\sqrt{N}}, \frac{(\Gamma^*)^{\frac{1}{3}}}{768^{\frac{1}{3}} LTN^{\frac{1}{3}}}\}$$

*Then, for all $Q \geqslant 1$, we have:*

$$\begin{aligned} \frac{1}{Q} \sum_{q=1}^{Q} \sum_{i \in S_q} \mathbb{E}\|\nabla F^i(\theta_q)\|^2 &\leq \frac{8\mathbb{E}[F(\theta_1)]}{T\gamma Q} \\ &+ (64w^2 \frac{N}{\Gamma^*} L^2 + 96L^3 \gamma T \frac{N}{\Gamma^*} w^2) \frac{1}{Q} \sum_{q=1}^{Q} \mathbb{E}\|\theta_q\|^2 \\ &+ \frac{8N}{\Gamma^*} (32\gamma^2 T^2 L^2 + 1 + 96L^3 \gamma^3 T^3 + 3L\gamma T) \delta^2 \\ &+ \gamma L \frac{8N}{\Gamma^*} (4\gamma TL + \frac{3}{2} + 12L^2 \gamma^2 T^2) \sigma^2, \end{aligned} \tag{10}$$

*where $2w_1^2 w_2^2 + 2w_1^2 + w_2^2 = w^2$*

Theorem 1 shows the convergence rate of algorithm *RA-Fed* by giving the upper bound on the average gradient of all clients for all trained neuron regions.

**Remark 1** Impact of the number of trained neuron regions $|S_q|$. Our algorithm is novel with stronger generalization, in which not all neuron regions can be trained in each round. Specifically, regions $(\mathcal{K} - S_q)$ can not be trained in round $q$. It is obvious that in identical settings, the larger $|S_q|$, the more neuron regions trained, the more gradients on neuron regions can be bounded, the better the convergence rate. Furthermore, to reduce the impact of partial neuron regions not being updated in some rounds on the convergence rate, we design a new server aggregation mechanism in Algorithm 2, which achieves the same convergence rate and ensures that all neuron regions can be updated in each round.

**Remark 2:** Impact of the mask-induced noise $w_1$ and compression noise $w_2$.

Our convergence result shows that the smaller noises $w^2 = 2w_1^2w_2^2 + 2w_1^2 + w_2^2$ would lead to a faster convergence rate and better performance in federated learning. Besides, it is worth noting that the mask-induced noise is also highly related to $S_q$ and $\Gamma^*$.

**Remark 3:** Impact of the data heterogeneity level $\delta$.

In our learning paradigm, we consider heterogeneous data distributions in real-world scenarios. The larger $\delta$ denotes the higher the data heterogeneity level and the slower convergence rate. Therefore, to reduce the impact of data heterogeneity on learning performance, we propose the *RAM-Fed* shown in Algorithm 2.

Next, by choosing the appropriate convergence rate $\gamma$ and the parameters representing data heterogeneity levels $\delta$, we can obtain the following corollary.

Corollary 1. *Let all assumptions hold. Supposing that the step size $\gamma = O(\sqrt{\frac{\Gamma^*}{TQ}})$ and that $\delta = O(\frac{1}{\sqrt{TQ}})$, when the constant $C > 0$ exists, the convergence rate can be expressed as follows:*

$$\frac{1}{Q}\sum_{q=1}^{Q}\sum_{i\in S_q}\mathbb{E}\|\nabla F^i(\theta_q)\|^2 \leq C(\frac{1}{\sqrt{\Gamma^* TQ}}+\frac{1}{Q}+\frac{1}{\Gamma^* TQ}+\frac{1}{Q^{1.5}}+\frac{1}{Q^2}+\frac{1}{Q^{2.5}}) \tag{11}$$

*Corollary 1 indicates that when Q is sufficiently large, the term $O(1/\sqrt{\Gamma^* TQ})$ will dominate the convergence rate and the convergence increases with the number of $\Gamma^*$.*

The detailed theoretical proof of Theorem 1 and Corollary 1 are provided in Supplement.

**Remark 4:** Impact of the minimum coverage rate $\Gamma^*$.

The Corollary 1 demonstrates that our proposed *RA-Fed* algorithm can converge to $O(1/\sqrt{\Gamma^* TQ})$ under arbitrary adaptive online mask. Except for the non-trained neuron regions from the global model, others can be trained by at least $\Gamma^*$ submodels in each round. Intuitively when fixing other impacting factors, as $\Gamma^*$ increases, the more frequently the neuron region can be trained, so the faster *RA-Fed* can converge to a stationary point.

## 5.2 Convergence analysis of RAM-Fed

Assumption 6. *Number of continuously non-trained rounds: We define the total number of rounds that client n has not trained neuron region i continuously as $\tau^i_{q,n}$:*

$$\tau_q = \max_{n,i}\tau^i_{q,n}, n \in N, i \in \mathcal{K} \tag{12}$$

Assumption 7. *Bounded gradient: In algorithm 2, the expected squared norm of stochastic gradients is bounded uniformly, for constant $G > 0$ and $\forall n, q, t$:*

$$\mathbb{E}\|\nabla F_n(\theta_{q,n,t}, \xi_{q,n,t})\|^2 \leq G. \tag{13}$$

Lemma 2. *Deviation of average submodel stochastic gradient between round q and round $q - \tau_q$: Let all assumptions hold.*

$$\begin{aligned}
&\sum_{i\in S_q}\mathbb{E}\|\frac{1}{\Gamma_q^i}\sum_{n\in N_q^i}\frac{1}{T}\sum_{t=1}^{T}\nabla F_n^i(\theta_{q,n,t-1}, \xi_{n,t-1})\\
&-\frac{1}{\Gamma_q^i}\sum_{n\in N_q^i}\frac{1}{T}\sum_{t=1}^{T}\nabla F_n^i(\theta_{q-\tau_q,n,t-1}, \xi_{n,t-1})\|^2 \qquad (14)\\
&\leq 6\frac{N}{\Gamma^*}\sigma^2 + 18\frac{N}{\Gamma^*}L^2(4\gamma^2T\sigma^2 + 32\gamma^2T^2\delta^2 + 32\gamma^2T^2G)\\
&+ 36\frac{N}{\Gamma^*}L^2w^2(\mathbb{E}\|\theta_q\|^2 + \mathbb{E}\|\theta_{q-\tau_q}\|^2)\\
&+ 9\frac{N}{\Gamma^*}L^2(3(\tau_q)^2\gamma^2T^2G(1+\frac{2N}{\Gamma^*}) + (\tau_q)^2\gamma^2T^2G)
\end{aligned}$$

Lemma 2 bounds the difference of average local submodel stochastic gradient between round $q$ and round $q - \tau_q$ for all trained neuron regions $S_q$.

Theorem 2. *Let all assumptions hold. Suppose that the step size $\gamma$ satisfies the following relationships:*

$$\begin{cases} 8\gamma^2L^2T^2 \leq \frac{1}{2} \Rightarrow \gamma \leq \frac{1}{4LT} \\ \frac{L}{2}\gamma^2T^2 - \frac{T\gamma}{2} < 0 \Rightarrow \gamma < \frac{1}{LT} \end{cases}$$

*Therefore, the step size $\gamma$ is defined as:*

$$0 \leq \gamma \leq \frac{1}{4LT}$$

*Then, for all $Q \geqslant 1$, we have:*

$$\begin{aligned}
&\frac{1}{Q}\sum_{q=1}^{Q}\mathbb{E}\|\nabla F(\theta_q)\|^2 \leq \frac{2\mathbb{E}[F(\theta_1)]}{T\gamma Q}\\
&+ (48\tau + 384 + 1152\frac{N}{\Gamma^*}L^2 + 72\frac{N}{\Gamma^*}\tau + 72\frac{N}{\Gamma^*}L^2\tau + 108(\frac{N}{\Gamma^*})^2L^2\tau)\gamma^2T^2G\\
&+ (48\gamma^2T + 12\frac{N}{\Gamma^*} + 6 + 144\frac{N}{\Gamma^*}L^2\gamma^2T)\sigma^2 + 128(3 + 9\frac{N}{\Gamma^*}L^2)\gamma^2T^2\delta^2\\
&+ 8w^2(6 + 9\frac{N}{\Gamma^*}L^2)\frac{1}{Q}\sum_{q=1}^{Q}\mathbb{E}\|\theta_{q-\tau_q}\|^2 + 72\frac{N}{\Gamma^*}L^2w^2\frac{1}{Q}\sum_{q=1}^{Q}\mathbb{E}\|\theta_q\|^2,
\end{aligned} \tag{15}$$

*where $\frac{1}{Q}\sum_{q=1}^{Q}(\tau_q)^2 = \tau$*

Theorem 2 shows the convergence rate of *RAM-Fed* algorithm by giving the upper bound on the average gradient of all neuron regions on all clients. It is worth noting that in the convergence result, we ensure that all neuron regions can be updated in each round compared with Algorithm 1. As shown in the theoretical result of *RAM-Fed*, we remove $S_q$ to achieve the bound of gradients on all neuron regions, which is consistent with our optimal goal.

**Remark 5** Impact of the maximum number of continuously non-trained rounds $\tau_q$

In our convergence analysis, we need to satisfy: $q - \tau_q > 0 \Rightarrow \tau_q < q$. Recall that $\tau_q = \max_{n,i}\tau^i_{q,n}$ means until round $q$, the maximum number of non-trained for all neuron regions on all clients. Therefore, in our algorithm, we can only ensure that all neuron regions are trained on all clients in the first round. In this case, inequality $\tau_q < q$ always holds. Moreover, the result indicates that the larger $\tau_q$, the worse the convergence rate.

Next, by choosing the appropriate convergence rate $\gamma$, we can obtain the following corollary.

Corollary 2. *Let all assumptions hold. Supposing that the step size $\gamma = O(\sqrt{\frac{\Gamma^*}{TQ}})$ and $\sigma$ is sufficiently small, when the constant $C > 0$ exists, the convergence rate can be expressed as follows:*

$$\frac{1}{Q}\sum_{q=1}^{Q}\mathbb{E}\|\nabla F(\theta_q)\|^2 \leq C(\frac{1}{\sqrt{\Gamma^* TQ}} + \frac{1}{Q} + \frac{1}{\Gamma^* Q} + \frac{1}{Q^2}) \tag{16}$$

**Table 2: Performance comparison on MLP-MNIST and CNN-CIFAR10. '-' means this method doesn't work under corresponding mask level setting. Bold is the optimal result except for FedAvg with full model training, underlined is the suboptimal result.**

| Methods | Mask level | MLP-MNIST (Accuracy %) | | | | CNN-CIFAR10 (Accuracy %) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $\alpha = 0.01$ | $\alpha = 0.05$ | $\alpha = 0.1$ | $\alpha = 0.15$ | $\alpha = 0.1$ | $\alpha = 0.2$ | $\alpha = 0.3$ | $\alpha = 0.4$ |
| FedAvg | Full | 87.9 | 91.2 | 94.2 | 96.4 | 61.9 | 64.5 | 70.6 | 75.8 |
| SplitFL | SameStr. | 52.2 | 69.9 | 72.4 | 67.6 | 37.6 | 45.8 | 50.1 | 63.8 |
| | Arb. | - | - | - | - | - | - | - | |
| IST | U.A. | 66.9 | 80.5 | 87.0 | 91.5 | 24.5 | 25.0 | 27.1 | 27.9 |
| | Arb. | - | - | - | - | - | - | - | - |
| PruneFL | L-Arb. | 80.8 | 87.5 | 91.5 | 94.8 | 48.7 | 51.4 | 53.6 | 60.0 |
| | S-Arb. | 64.3 | 79.1 | 86.4 | 91.7 | 29.5 | 39.7 | 42.4 | 43.7 |
| | MIX-Arb. | 78.8 | 85.5 | 90.0 | 94.2 | 38.6 | 49.7 | 53.2 | 56.8 |
| OAP | L-Arb. | 65.8 | 77.7 | 84.9 | 92.2 | 45.9 | 49.4 | 50.5 | 56.0 |
| | S-Arb. | 37.8 | 71.7 | 75.9 | 84.4 | 16.8 | 18.0 | 27.9 | 35.4 |
| | MIX-Arb. | 61.5 | 74.5 | 80.9 | 90.4 | 45.5 | 43.8 | 54.0 | 56.8 |
| RA-Fed(ours) | L-Arb. | **85.2** | **89.0** | **92.8** | **95.6** | **49.2** | **54.8** | **57.8** | **65.1** |
| | S-Arb. | **78.1** | **86.4** | **90.5** | **94.1** | **33.4** | **44.5** | **46.2** | **51.1** |
| | MIX-Arb. | **82.7** | **87.1** | **91.0** | **94.8** | **49.6** | **53.4** | **54.6** | **61.6** |
| RAM-Fed(ours) | L-Arb. | **88.0** | **90.1** | **95.4** | **96.6** | **52.7** | **55.6** | **61.8** | **65.3** |
| | S-Arb. | **85.0** | **87.8** | **94.2** | **94.9** | **39.5** | **47.5** | **49.2** | **51.1** |
| | MIX-Arb. | **86.5** | **89.6** | **94.5** | **95.7** | **50.8** | **53.7** | **60.5** | **61.4** |

*Corollary 2 indicates that when $Q$ is sufficiently large, the term $O(1/\sqrt{\Gamma^* TQ})$ will dominate the convergence rate and the convergence increases with $\Gamma^*$.*

**Remark 6** Recall that $\Gamma^* = \min_{q,i} \Gamma_q^i, i \in S_q, \forall q$ measures the minimum number of submodels training the corresponding neuron region $i \in S_q$ in all rounds. Thus, it is obvious that $1 \leq \Gamma^* \leq N$. When $\Gamma^* = N$, all submodels can train neuron region $i \in S_q$, which achieves the same convergence rate $O(1/\sqrt{NTQ})$ as the vanilla *FedAvg* [16, 25]. When $\Gamma^* = 1$, we achieves the same convergence rate $O(1/\sqrt{TQ})$ as *OAP* [30].

The detailed theoretical proof of Theorem 2 and Corollary 2 are provided in Supplement.

# 6 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate kinds of federated learning paradigms. The details are shown as follows.

## 6.1 Datasets and baseliens

*6.1.1 Task and dataset description.* We perform all approaches on two different types of machine learning tasks: image classification on *MNIST* and *CIFAR10*, text classification on *AGnews* [29].

**Image classification:** We train CNN with 2 convolution layers and 3 hidden layers on *CIFAR10*, MLP with 2 hidden layers on *MNIST*. *CIFAR10* contains 10 categories with 50k training images and 10k testing images. *MNIST* contains 10 categories with 60k training images and 10k testing images. **Text classification:** We train *FastText* [7] with 1 embedding layer and 2 hidden layers on *AGNews*, *AGNews* contains 4 categories with 120k training news articles and 7600 testing news articles.

*6.1.2 Dataset partition.* We use Dirichlet distribution Dir($\alpha$) [3] to set up different data heterogeneity levels. The smaller $\alpha$ represents the stronger heterogeneity levels.

*6.1.3 Baselines and Metrics.* We compare our learning paradigm with related state-of-the-art methods in resource-limited federated learning: **Fedavg** [16], **SplitFL** [20] , **IST** [26], **PruneFL** [5] and **OAP** [30]. We evaluate all approaches on two important evaluation Metrics in resource-limited federated learning. The *Mask level* measures the average local submodel size and the arbitrariness level of submodels. The *Accuracy* measures the performance of different learning paradigms on various tasks.

## 6.2 Experimental setup

**Submodel setup:** The submodels are designed based on arbitrarily assigned neurons, which denotes that not all the neurons of the full network can be trained in each round. To achieve this goal, we randomly select partial neuron regions not to be trained periodically. We design three different mask levels to generate different numbers of submodel parameters. *L* or *S* denotes that submodels train 50% or 25% parameters of the full model respectively. Differently, *MIX* denotes that 50% submodels are trained with 50% parameters, while others are with 25% parameters. Specifically, in each round, the global model $\theta$ can be split into 4 neuron regions, $\theta = \{\theta^1, \theta^2, \theta^3, \theta^4\}$. Considering arbitrarily assigned neuron regions, when the mask level is set to *L-Arb.*, each client can adaptively select 2 neuron regions (e.g. $\{\theta^1, \theta^2\}$) to train. Thus, at most 6 types of heterogeneous submodels can be generated. Under *MIX-Arb.* setting, we randomly select 50% submodels with 2 adaptively chosen neuron regions, while others are with 1 adaptively chosen neuron region for training. In different rounds, the submodel structure within the same client could change due to the varying resources. It is worth noting that for *FedAvg*, full models need to be trained. The

**Table 3: Performance comparison on FastText-AGNews. '-' means this method doesn't work under corresponding mask level setting.**

| Methods | Mask level | Accuracy(%) | | | |
|---|---|---|---|---|---|
| | | $\alpha = 0.05$ | $\alpha = 0.1$ | $\alpha = 0.15$ | $\alpha = 0.20$ |
| FedAvg | Full | 73.7 | 71.8 | 82.0 | 82.2 |
| SplitFL | SameStr. | 50.4 | 73.4 | 82.0 | 83.4 |
| | Arb. | - | - | - | - |
| IST | U.A. | 51.6 | 52.3 | 57.7 | 62.8 |
| | Arb. | - | - | - | - |
| PruneFL | L-Arb. | 67.3 | 72.0 | 80.3 | 80.4 |
| | S-Arb.. | 57.8 | 59.6 | 71.2 | 72.2 |
| | MIX-Arb. | 66.2 | 70.3 | 79.5 | 80.3 |
| OAP | L-Arb. | 45.8 | 44.1 | 51.7 | 53.3 |
| | S-Arb. | - | - | - | - |
| | MIX-Arb. | 37.3 | 39.7 | 46.2 | 49.4 |
| RA-Fed (ours) | L-Arb. | **73.3** | **78.1** | **84.6** | **84.9** |
| | S-Arb. | **66.7** | **72.9** | **83.0** | **84.1** |
| | MIX-Arb. | **70.8** | **77.3** | **83.9** | **84.7** |
| RAM-Fed (ours) | L-Arb. | **77.6** | **86.6** | **89.3** | **89.4** |
| | S-Arb. | **73.1** | **85.3** | **87.1** | **88.7** |
| | MIX-Arb. | **76.8** | **86.5** | **88.4** | **89.2** |

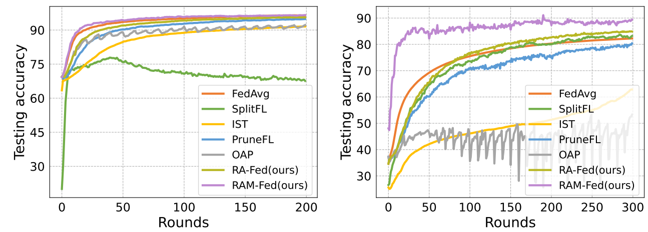
arbitrarily assigned neuron regions are not allowed in *SplitFL* and *IST*. According to their original definitions, the same submodel structures need to be constructed in every client in *SplitFL*, and neurons are uniformly assigned (U.A.) to different clients.

**Training setup:** In our experiments, we train tasks with momentum SGD optimizer on 10 clients. The batch size is set to 128. The momentum parameter is set to 0.5. The number of local iterations is set to 5. The learning rate $\gamma$ is set to 0.01 on *MLP-MNIST*, 0.05 on *CNN-CIFAR10* and 0.1 on *FastText-AGNews*. For all approachs, we use an identical experiment setup, and run all experiments on ten GeForce RTX 3090 GPUs.

## 6.3 Numerical results

About the results on image classification tasks in table 2, we can observe that:

- On the whole, our algorithms outperform all baselines under different mask levels and data heterogeneity level settings. Except *FedAvg* with full model, in testing accuracy, *RA-Fed* nearly improves 2%-20% and *RAM-Fed* improves 4%-22%. Comparing with the baselines excepting FedAvg, *RA-Fed* and *RAM-Fed* improve accuracy by 8.5% and 10% on average respectively. Particularly, RAM-Fed with 25% submodel achieves comparable accuracy to PruneFL with 50% model.
- Compared with *FedAvg*, we observe that *FedAvg* is slightly higher than our algorithms in testing accuracy but *RAM-Fed* with L-Arb. mask level slightly outperforms *FedAvg* on *MLP-MNIST*, which demonstrates that our algorithm is robust in the resource-limited learning environment.
- Especially in our proposed *RAM-Fed* algorithm, it greatly improves the performance in non-uniform training and achieves the highest accuracy under high data heterogeneity levels, which further indicates that the new aggregation mechanism in *RAM-Fed* effectively mitigates the impact of non-uniform and data heterogeneity.
- For the impact of mask level, all algorithms nearly perform worse due to the larger induced noises when the mask level varies from *L-Arb.* to *S-Arb.*. It indicates that the mask-induced error is a key factor impacting performance which is consistent with our theoretical analysis.
- With the increment of data heterogeneity level $\alpha$, all methods' accuracy generally becomes worse. But *RAM-Fed* decreases slightly which demonstrates that *RAM-Fed* is more robust in data heterogeneous scenarios.
- In general, comparing with other baselines, *PruneFL* can achieve higher accuracy, which is due to the fact that important parameters are selected for training in every round.




(a) MLP-MNIST (b) FastText-AGNews

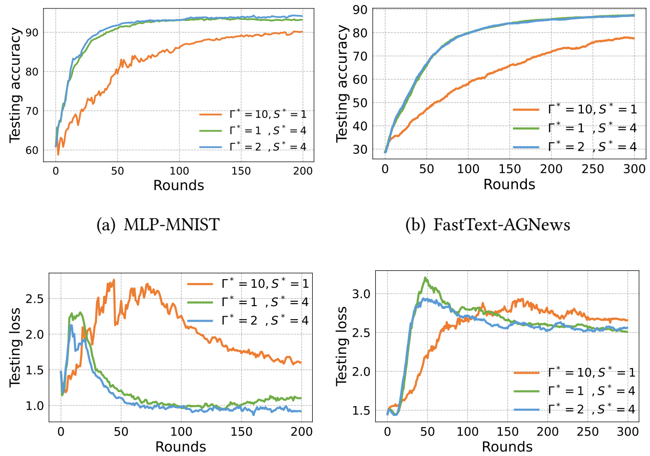
**Figure 2: Training process of different learning paradigms.**

About results on text classification task in table 3, we conclude:

- Noticeably, in any mask level, *RAM-Fed* performs better than other algorithms, nearly achieving 7%-30% improvements. Comparing with the baselines, *RA-Fed* and *RAM-Fed* improve accuracy by 9% and 13% on average respectively.
- Surprisingly, *RAM-Fed* algorithm nearly performs better than *FedAvg*, which might be because partial stale gradients could be larger than the current gradients with the right direction. This further demonstrates the effectiveness of our proposed submodels joint novel training mechanism.
- *RA-Fed* and *RAM-Fed* algorithms all achieve higher accuracy than *PruneFL*, which indicates that adaptive strategy could perform better than high-wight parameter selection method.
- Even comparing with *SplitFL* and *IST* with uniform training, *RA-Fed* and *RAM-Fed* all achieve better performance with arbitrarily assigned neurons.

The convergence process of different learning paradigms on image classification (*L-Arb.* mask level, $\alpha = 0.15$ on *MLP-MNIST*) and text classification task (*L-Arb.* mask level, $\alpha = 0.2$ on *FastText-AGNews*) are depicted in Fig 2.

- As shown in Fig. 2(a) and Fig. 2(b), *RA-Fed* and *RAM-Fed* have similar convergence trends with *FedAvg* on *MLP-MNIST*, this is because neuron regions can be trained sufficiently with training continues. Surprisingly, *RAM-Fed* achieves better performance significantly compared with other algorithms on *FastText-AGNews*.
- *OAP* diverges with obvious fluctuations during training, while *IST* converges slowly. *SplitFL* has the worst performance on *MLP-MNIST*, which is might due to the over-fitting of the same submodel structures across all clients.




(a) MLP-MNIST (b) FastText-AGNews

(c) MLP-MNIST (d) FastText-AGNews

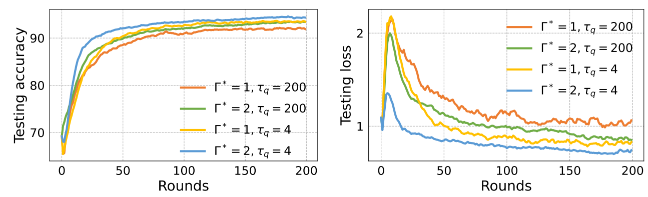
**Figure 3: The impact of $\Gamma^*$ and $S^*$ of RA-Fed**

## 6.4 Impact of key factors

*6.4.1 Impact of minimum coverage rate* $\Gamma^*$ *and minimum number of trained neuron regions* $S^*$. Based on the above analysis, the proposed adaptive learning paradigm is essential and effective. Combined with our theoretical analysis, we study two key factors impacting convergence and accuracy: $\Gamma^*$ and $S^*$. Fixing other impacting factors in *RA-Fed*, we set *S-Arb.* mask level, $\alpha = 0.15$ on *MLP-MNIST* task, and $\alpha = 0.5$ on *FastText-AGNews* task. Through varying $\Gamma^*$ and $S^*$, we set three combinations: ($\Gamma^*$=1, $S^*$=4), ($\Gamma^*$=2, $S^*$=4), ($\Gamma^* = 10$, $S^*$=1). As shown in Fig. 3, we have some observations:

- When fixing $S^* = 4$, we find that $\Gamma^*$=2 performs slightly better than $\Gamma^*$=1 in testing accuracy. As shown in testing loss, the trends clearly show that $\Gamma^*$=2 can converge faster. It is worth noting that the testing loss increases rapidly at the initial rounds, and then decreases slowly. This is mainly because only partial neurons can be trained at first, but as the training continues, all neuron regions can be trained sufficiently. In addition, we observe that in testing loss, the $\Gamma^*$=2 curve decreases earlier than $\Gamma^*$=1, which is consistent with our theoretical analysis.
- The larger $\Gamma^*$ does not mean the higher accuracy and faster convergence. Considering an extreme example with the largest $\Gamma^*$ (e.g. $\Gamma^* = 10$, $S^*$=1), in this case, only one neuron region is trained by ten submodels in each round. However, the testing accuracy decreases significantly and testing loss converges very slowly which indicates that when a large number of neuron regions are not trained, the model performance becomes poor. Therefore, we can conclude that the performance and convergence rate are impacted by multiple factors comprehensively.

*6.4.2 Impact of the maximum number of continuously non-trained rounds* $\tau_q$. We further consider two key factors impacting convergence and accuracy in *RAM-Fed*: $\Gamma^*$ and $\tau_q$. Fixing other impacting factors in *RAM-Fed*: we set *S-Arb.* as mask level, $\alpha = 0.15$ on *MLP-MNIST* task. Considering $\Gamma^*$ and $\tau_q$, we set four combinations: ($\Gamma^*$=1, $\tau_q$=200), ($\Gamma^*$=2, $\tau_q$=200), ($\Gamma^*$=1, $\tau_q$=4), ($\Gamma^*$=2, $\tau_q$=4). As shown in Fig. 4, we have some observations:




(a) MLP-MNIST (b) MLP-MNIST

**Figure 4: The impact of $\Gamma^*$ and $\tau_q$ of RAM-Fed**

- When fixing $\Gamma^*$, $\tau_q$=4 performs better than $\tau_q$=200 in testing accuracy and loss. Thus, when $\tau_q$ is very larger (e.g. $\tau_q$=200), some neurons can only be updated by stale gradients continuously, which causes a bias compared with the right descent direction.
- When fixing $\tau_q$, it is obvious that as $\Gamma^*$ increases, the performance becomes better, which further indicates that the larger $\Gamma^*$, the more fully training the neuron regions.
- On the whole, $\Gamma^*$ and $\tau_q$ play important roles in convergence. For *RAM-Fed*, the optimal $\Gamma^*$ and $\tau_q$ can significantly improve performance.

# 7 CONCLUSION

In traditional cross-device federated learning, massive devices are usually equipped with limited resources for computation and communication which would be unaffordable to run the full model for coordination. To this end, we designed an adaptive learning paradigm, in which heterogeneous local submodels with arbitrarily assigned neurons can be jointly trained to produce a single global model. In order to address the arising *submodels heterogeneity*, *non-uniform training* and *data heterogeneity* challenges, we proposed general *RA-Fed* algorithm and *RAM-Fed* with a new server aggregation mechanism. We theoretically proved the proposed *RA-Fed* and *RAM-Fed* can both converge with asymptotically optimal rate $O(1/\sqrt{\Gamma^* TQ})$ under given assumptions. We investigated several key factors impacting convergence, such as minimum coverage rate, data heterogeneity level, submodel induced noises. Extensive experiments were conducted on two types of tasks with three widely used datasets. Compared with the state-of-the-art baselines, our algorithms improved the accuracy up to 10% on average. Particularly, *RAM-Fed* with 50% model achieved comparable accuracy compared with *FedAvg* with full model, even outperforming *FedAvg*.

# ACKNOWLEDGMENTS

This work was supported in part by the National Key R& D Program of China under Grant No. 2022YFF0712100, in part by the National Natural Science Foundation of China under Grant 62202273, in part by National Science Fund for Excellent Young Scholars of China under Grant 62122042, in part by Major Basic Research Program of Shandong Provincial Natural Science Foundation under Grant ZR2022ZD02, in part by Shandong Provincial Natural Science Foundation of China under Grant ZR2021QF044, in part by the Fundamental Research Funds for the Central Universities.

# REFERENCES


[1] Enmao Diao, Jie Ding, and Vahid Tarokh. 2021. HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=TNkPBBYFkXg

[2] Xinran Gu, Kaixuan Huang, Jingzhao Zhang, and Longbo Huang. 2021. Fast Federated Learning in the Presence of Arbitrary Device Unavailability. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 12052–12064. https://proceedings.neurips.cc/paper/2021/hash/64be20f6dd1dd46adf110cf871e3ed35-Abstract.html

[3] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification. *CoRR* abs/1909.06335 (2019). arXiv:1909.06335 http://arxiv.org/abs/1909.06335

[4] Divyansh Jhunjhunwala, Pranay Sharma, Aushim Nagarkatti, and Gauri Joshi. 2022. Fedvarp: Tackling the variance due to partial client participation in federated learning. In *Uncertainty in Artificial Intelligence, Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI 2022, 1-5 August 2022, Eindhoven, The Netherlands (Proceedings of Machine Learning Research, Vol. 180)*, James Cussens and Kun Zhang (Eds.). PMLR, 906–916. https://proceedings.mlr.press/v180/jhunjhunwala22a.html

[5] Yuang Jiang, Shiqiang Wang, Bong Jun Ko, Wei-Han Lee, and Leandros Tassiulas. 2019. Model Pruning Enables Efficient Federated Learning on Edge Devices. *CoRR* abs/1909.12326 (2019). arXiv:1909.12326 http://arxiv.org/abs/1909.12326

[6] Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is ChatGPT A Good Translator? A Preliminary Study. *CoRR* abs/2301.08745 (2023). https://doi.org/10.48550/arXiv.2301.08745 arXiv:2301.08745

[7] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, Mirella Lapata, Phil Blunsom, and Alexander Koller (Eds.). Association for Computational Linguistics, 427–431. https://doi.org/10.18653/v1/e17-2068

[8] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. 2021. Breaking the centralized barrier for cross-device federated learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (Eds.). 28663–28676. https://proceedings.neurips.cc/paper/2021/hash/f0e6be4ce76ccfa73c5a540d992d0756-Abstract.html

[9] Anastasia Koloskova, Tao Lin, Sebastian U. Stich, and Martin Jaggi. 2020. Decentralized Deep Learning with Arbitrary Communication Compression. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=SkgGCkrKvH

[10] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J. Dally. 2017. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. *CoRR* abs/1712.01887 (2017). arXiv:1712.01887 http://arxiv.org/abs/1712.01887

[11] Fan Liu, Hao Liu, and Wenzhao Jiang. 2022. Practical Adversarial Attacks on Spatiotemporal Traffic Forecasting Models. In *Advances in Neural Information Processing Systems*, Vol. 35. 19035–19047.

[12] Hao Liu, Ying Li, Yanjie Fu, Huaibo Mei, Jingbo Zhou, Xu Ma, and Hui Xiong. 2020. Polestar: An intelligent, efficient and national-wide public transportation routing engine. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2321–2329.

[13] Hao Liu, Qiyu Wu, Fuzhen Zhuang, Xinjiang Lu, Dejing Dou, and Hui Xiong. 2021. Community-Aware Multi-Task Transportation Demand Prediction. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*. 320–327.

[14] Jianchun Liu, Hongli Xu, Yang Xu, Zhenguo Ma, Zhiyuan Wang, Chen Qian, and He Huang. 2021. Communication-efficient asynchronous federated learning in resource-constrained edge computing. *Comput. Networks* 199 (2021), 108429. https://doi.org/10.1016/j.comnet.2021.108429

[15] Xiaolong Ma, Minghai Qin, Fei Sun, Zejiang Hou, Kun Yuan, Yi Xu, Yanzhi Wang, Yen-Kuang Chen, Rong Jin, and Yuan Xie. 2022. Effective Model Sparsification by Scheduled Grow-and-Prune Methods. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=xa6otUDdP2W

[16] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Xiaojin (Jerry) Zhu (Eds.). PMLR, 1273–1282. http://proceedings.mlr.press/v54/mcmahan17a.html

[17] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and Communication-Efficient Federated Learning from Non-IID Data. *CoRR* abs/1903.02891 (2019). arXiv:1903.02891 http://arxiv.org/abs/1903.02891

[18] Navjot Singh, Deepesh Data, Jemin George, and Suhas N. Diggavi. 2021. SQuARM-SGD: Communication-Efficient Momentum SGD for Decentralized Optimization. *IEEE J. Sel. Areas Inf. Theory* 2, 3 (2021), 954–969. https://doi.org/10.1109/JSAIT.2021.3103920

[19] Hanlin Tang, Xiangru Lian, Shuang Qiu, Lei Yuan, Ce Zhang, Tong Zhang, and Ji Liu. 2019. DeepSqueeze: Parallel Stochastic Gradient Descent with Double-Pass Error-Compensated Compression. *CoRR* abs/1907.07346 (2019). arXiv:1907.07346 http://arxiv.org/abs/1907.07346

[20] Chandra Thapa, Mahawaga Arachchige Pathum Chamikara, Seyit Camtepe, and Lichao Sun. 2022. SplitFed: When Federated Learning Meets Split Learning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 8485–8493. https://ojs.aaai.org/index.php/AAAI/article/view/20825

[21] Zhiyuan Wang, Hongli Xu, Jianchun Liu, He Huang, Chunming Qiao, and Yangming Zhao. 2021. Resource-Efficient Federated Learning with Hierarchical Aggregation in Edge Computing. In *40th IEEE Conference on Computer Communications, INFOCOM 2021, Vancouver, BC, Canada, May 10-13, 2021*. IEEE, 1–10. https://doi.org/10.1109/INFOCOM42981.2021.9488756

[22] Qi Xia, Winson Ye, Zeyi Tao, Jindi Wu, and Qun Li. 2021. A survey of federated learning for edge computing: Research problems and solutions. *High-Confidence Computing* 1, 1 (2021), 100008. https://doi.org/10.1016/j.hcc.2021.100008

[23] Xin Yao, Tianchi Huang, Rui-Xiao Zhang, Ruiyu Li, and Lifeng Sun. 2019. Federated Learning with Unbiased Gradient Aggregation and Controllable Meta Updating. *CoRR* abs/1910.08234 (2019). arXiv:1910.08234 http://arxiv.org/abs/1910.08234

[24] Hongzheng Yu, Zekai Chen, Xiao Zhang, Xu Chen, Fuzhen Zhuang, Hui Xiong, and Xiuzhen Cheng. 2023. FedHAR: Semi-Supervised Online Learning for Personalized Federated Human Activity Recognition. *IEEE Transactions on Mobile Computing* 22, 6 (2023), 3318–3332. https://doi.org/10.1109/TMC.2021.3136853

[25] Hao Yu, Sen Yang, and Shenghuo Zhu. 2019. Parallel Restarted SGD with Faster Convergence and Less Communication: Demystifying Why Model Averaging Works for Deep Learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 5693–5700. https://doi.org/10.1609/aaai.v33i01.33015693

[26] Binhang Yuan, Cameron R. Wolfe, Chen Dun, Yuxin Tang, Anastasios Kyrillidis, and Chris Jermaine. 2022. Distributed Learning of Fully Connected Neural Networks using Independent Subnet Training. *Proc. VLDB Endow.* 15, 8 (2022), 1581–1590. https://www.vldb.org/pvldb/vol15/p1581-wolfe.pdf

[27] Xiao Zhang, Qilin Wang, Ziming Ye, Haochao Ying, and Dongxiao Yu. 2023. Federated Representation Learning With Data Heterogeneity for Human Mobility Prediction. *IEEE Transactions on Intelligent Transportation Systems* (2023), 1–12. https://doi.org/10.1109/TITS.2023.3252029

[28] Xiao Zhang, Yangyang Wang, Shuzhen Chen, Cui Wang, Dongxiao Yu, and Xiuzhen Cheng. 2023. Robust communication-efficient decentralized learning with heterogeneity. *Journal of Systems Architecture* (2023), 102900. https://doi.org/10.1016/j.sysarc.2023.102900

[29] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (Eds.). 649–657. https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html

[30] Hanhan Zhou, Tian Lan, Guru Prasadh Venkataramani, and Wenbo Ding. [n. d.]. Federated Learning with Online Adaptive Heterogeneous Local Models. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*.

# A SUPPLEMENT

## A.1 Part One

Let us start the proof of RA-Fed from $L$-Lipschitzian Condition:

$$\mathbb{E}[F(\theta_{q+1})] - \mathbb{E}[F(\theta_q)] \le \underbrace{\mathbb{E}[< \nabla F(\theta_q), \theta_{q+1} - \theta_q >]}_{U_1} + \frac{L}{2}\underbrace{\mathbb{E}[\|\theta_{q+1} - \theta_q\|^2]}_{U_2}$$

bound $U_1$:

$$\begin{aligned}
&\mathbb{E} < \nabla F(\theta_q), \theta_{q+1} - \theta_q > \le -T\gamma \sum_{i\in S_q} \mathbb{E}\|\nabla F^i(\theta_q)\|^2 \\
&+ \frac{T\gamma}{2} \sum_{i\in S_q} \mathbb{E}\|\nabla F^i(\theta_q)\|^2 + 32\gamma^3 T^3 \frac{N}{\Gamma^*} L^2 \sum_{i\in S_q} \mathbb{E}\|\nabla F^i(\theta_q)\|^2 \\
&+ 8w^2 T\gamma \frac{N}{\Gamma^*} L^2 \mathbb{E}\|\theta_q\|^2 + 4\gamma^3 T^2 \frac{N}{\Gamma^*} L^2\sigma^2 + 32\gamma^3 T^3 \frac{N}{\Gamma^*} L^2\delta^2 + T\gamma\frac{N}{\Gamma^*}\delta^2
\end{aligned}$$

bound $U_2$:

$$\begin{aligned}
&\frac{L}{2}\mathbb{E}\|\theta_{q+1} - \theta_q\|^2 \\
&\le \frac{3}{2} LT\gamma^2 \frac{N}{\Gamma^*}\sigma^2 + 12L^3\gamma^4 \frac{N}{\Gamma^*} T^3\sigma^2 + 96L^3\gamma^4 T^4 \frac{N}{\Gamma^*}\delta^2 \\
&+ 96L^3\gamma^4 T^4 \frac{N}{\Gamma^*} \sum_{i\in S_q} \mathbb{E}\|\nabla F^i(\theta_q)\|^2 + 12L^3\gamma^2 T^2 \frac{N}{\Gamma^*} w^2 \mathbb{E}\|\theta_q\|^2 \\
&+ \frac{3}{2} L\gamma^2 T^2 \sum_{i\in S_q} \mathbb{E}\|\nabla F^i(\theta_q)\|^2 + 3L\frac{N}{\Gamma^*}\gamma^2 T^2\delta^2
\end{aligned}$$

Last,we have:

$$\begin{aligned}
&\mathbb{E}[F(\theta_{Q+1})] - \mathbb{E}[F(\theta_1)] = \sum_{q=1}^{Q} \mathbb{E}[F(\theta_{q+1})] - \sum_{q=1}^{Q} \mathbb{E}[F(\theta_q)] \\
&\le \sum_{q=1}^{Q} \mathbb{E}[< \nabla F(\theta_q), \theta_{q+1} - \theta_q >] + \sum_{q=1}^{Q} \frac{L}{2}\mathbb{E}\|\theta_{q+1} - \theta_q\|^2
\end{aligned}$$

Plugging $U_1, U_2$ into above equation, we have:

$$\begin{aligned}
&\mathbb{E}[F(\theta_{Q+1})] - \mathbb{E}[F(\theta_1)] \\
&\overset{a}{\le} -\frac{T\gamma}{8} \sum_{q=1}^{Q} \sum_{i\in S_q} \mathbb{E}\|\nabla F^i(\theta_q)\|^2 \\
&+ (8w^2 T\gamma \frac{N}{\Gamma^*} L^2 + 12L^3\gamma^2 T^2 \frac{N}{\Gamma^*} w^2) \sum_{q=1}^{Q} \mathbb{E}\|\theta_q\|^2 \\
&+ T\gamma Q\frac{N}{\Gamma^*}(32\gamma^2 T^2 L^2 + 1 + 96L^3\gamma^3 T^3 + 3L\gamma T)\delta^2 \\
&+ \gamma^2 TLQ\frac{N}{\Gamma^*}(4\gamma TL + \frac{3}{2} + 12L^2\gamma^2 T^2)\sigma^2
\end{aligned}$$

where $a$ follows because:

$$32\gamma^2 T^2 \frac{N}{\Gamma^*} L^2 \le \frac{1}{8} \Rightarrow \gamma \le \frac{\sqrt{\Gamma^*}}{16TL\sqrt{N}}$$

$$96L^3\gamma^3 T^3 \frac{N}{\Gamma^*} \le \frac{1}{8} \Rightarrow \gamma \le \frac{(\Gamma^*)^{\frac{1}{3}}}{768^{\frac{1}{3}} LTN^{\frac{1}{3}}}$$

$$\frac{3}{2} L\gamma T \le \frac{1}{8} \Rightarrow \gamma \le \frac{1}{12TL}$$

Therefore, we have:

$$\begin{aligned}
&\frac{T\gamma}{8} \sum_{q=1}^{Q} \sum_{i\in S_q} \mathbb{E}\|\nabla F^i(\theta_q)\|^2 \le \mathbb{E}[F(\theta_1)] - \mathbb{E}[F(\theta_{Q+1})] \\
&+ (8w^2 T\gamma \frac{N}{\Gamma^*} L^2 + 12L^3\gamma^2 T^2 \frac{N}{\Gamma^*} w^2) \sum_{q=1}^{Q} \mathbb{E}\|\theta_q\|^2 \\
&+ T\gamma Q\frac{N}{\Gamma^*}(32\gamma^2 T^2 L^2 + 1 + 96L^3\gamma^3 T^3 + 3L\gamma T)\delta^2 \\
&+ \gamma^2 TLQ\frac{N}{\Gamma^*}(4\gamma TL + \frac{3}{2} + 12L^2\gamma^2 T^2)\sigma^2
\end{aligned}$$

dividing both sides by $Q$ and $\frac{T\gamma}{8}$:

$$\begin{aligned}
&\frac{1}{Q} \sum_{q=1}^{Q} \sum_{i\in S_q} \mathbb{E}\|\nabla F^i(\theta_q)\|^2 \le \frac{8\mathbb{E}[F(\theta_1)]}{T\gamma Q} \\
&+ (64w^2 \frac{N}{\Gamma^*} L^2 + 96L^3\gamma T\frac{N}{\Gamma^*} w^2)\frac{1}{Q} \sum_{q=1}^{Q} \mathbb{E}\|\theta_q\|^2 \\
&+ \frac{8N}{\Gamma^*}(32\gamma^2 T^2 L^2 + 1 + 96L^3\gamma^3 T^3 + 3L\gamma T)\delta^2 \\
&+ \gamma L\frac{8N}{\Gamma^*}(4\gamma TL + \frac{3}{2} + 12L^2\gamma^2 T^2)\sigma^2
\end{aligned}$$

Supposing that the step size $\gamma = O(\sqrt{\frac{\Gamma^*}{TQ}})$ and that $\delta = O(\frac{1}{\sqrt{TQ}})$, when the constant $C > 0$ exists, the convergence rate can be expressed as follows:

$$\frac{1}{Q} \sum_{q=1}^{Q} \sum_{i\in S_q} \mathbb{E}\|\nabla F^i(\theta_q)\|^2 \le C(\frac{1}{\sqrt{\Gamma^* TQ}} + \frac{1}{Q} + \frac{1}{\Gamma^* TQ} + \frac{1}{Q^{1.5}} + \frac{1}{Q^2} + \frac{1}{Q^{2.5}})$$

## A.2 Part Two

Let us start the proof of RAM-Fed from $L$-Lipschitzian Condition:

$$\begin{aligned}
&\sum_{i\in S_q} \mathbb{E} < \nabla F^i(\theta_q), \theta^i_{q+1} - \theta^i_q > = \sum_{i\in S_q} \mathbb{E} < \nabla F^i(\theta_q), -\gamma \mathrm{v}^{\mathrm{i}}_{\mathrm{q}} > \\
&= -\frac{T\gamma}{2} \sum_{i\in S_q} \mathbb{E}\|\nabla F^i(\theta_q)\|^2 \\
&- \frac{T\gamma}{2} \sum_{i\in S_q} \mathbb{E}\| \frac{1}{N} \sum_{n=1}^{N} \frac{1}{T} \sum_{t=1}^{T} \nabla F^i_n(\theta_{q-\tau_q,n,t-1}, \xi_{n,t-1}) \\
&+ \frac{1}{\Gamma^i_q} \sum_{n\in N^i_q} \frac{1}{T} \sum_{t=1}^{T} \nabla F^i_n(\theta_{q,n,t-1}, \xi_{n,t-1}) \\
&- \frac{1}{\Gamma^i_q} \sum_{n\in N^i_q} \frac{1}{T} \sum_{t=1}^{T} \nabla F^i_n(\theta_{q-\tau_q,n,t-1}, \xi_{n,t-1})\|^2 \\
&+ \frac{T\gamma}{2} \underbrace{\sum_{i\in S_q} \mathbb{E}\|\nabla F^i(\theta_q) - \frac{1}{N} \sum_{n=1}^{N} \frac{1}{T} \sum_{t=1}^{T} \nabla F^i_n(\theta_{q-\tau_q,n,t-1}, \xi_{n,t-1}) - \frac{1}{\Gamma^i_q} \sum_{n\in N^i_q} \frac{1}{T} \sum_{t=1}^{T} \nabla F^i_n(\theta_{q,n,t-1}, \xi_{n,t-1})}_{T_1}
\end{aligned}$$

$$\underbrace{+\frac{1}{\Gamma_q^i}\sum_{n\in N_q^i}\frac{1}{T}\sum_{t=1}^{T}\nabla F_n^i(\theta_{q-\tau_q,n,t-1},\xi_{n,t-1})\|^2}_{T_1}$$

bound $T_1$:

$$\begin{aligned}&\sum_{i\in S_q}\mathbb{E}\|\nabla F^i(\theta_q)-\frac{1}{N}\sum_{n=1}^{N}\frac{1}{T}\sum_{t=1}^{T}\nabla F_n^i(\theta_{q-\tau_q,n,t-1},\xi_{n,t-1})\\&-\frac{1}{\Gamma_q^i}\sum_{n\in N_q^i}\frac{1}{T}\sum_{t=1}^{T}\nabla F_n^i(\theta_{q,n,t-1},\xi_{n,t-1})\\&+\frac{1}{\Gamma_q^i}\sum_{n\in N_q^i}\frac{1}{T}\sum_{t=1}^{T}\nabla F_n^i(\theta_{q-\tau_q,n,t-1},\xi_{n,t-1})\|^2\\&\le(32(\tau_q)^2+256+1152\frac{N}{\Gamma^*}L^2+48\frac{N}{\Gamma^*}(\tau_q)^2+72\frac{N}{\Gamma^*}L^2(\tau_q)^2\\&+108\frac{N^2}{(\Gamma^*)^2}L^2(\tau_q)^2)\gamma^2T^2G\\&+(32\gamma^2T+12\frac{N}{\Gamma^*}+4+144\frac{N}{\Gamma^*}L^2\gamma^2T)\sigma^2\\&+128\gamma^2T^2\delta^2(2+9\frac{N}{\Gamma^*}L^2)\\&+8w^2(4+9\frac{N}{\Gamma^*}L^2)\mathbb{E}\|\theta_{q-\tau_q}\|^2+72\frac{N}{\Gamma^*}L^2w^2\mathbb{E}\|\theta_q\|^2\end{aligned}$$

For another term in $L$-Lipschitzian condition, we have:

$$\begin{aligned}&\frac{L}{2}\sum_{i\in S_q}\mathbb{E}\|\theta_{q+1}^i-\theta_q^i\|^2\\&=\frac{L}{2}\gamma^2T^2\sum_{i\in S_q}\mathbb{E}\|\frac{1}{N}\sum_{n=1}^{N}\frac{1}{T}\sum_{t=1}^{T}\nabla F_n^i(\theta_{q-\tau_q,n,t-1},\xi_{n,t-1})\\&+\frac{1}{\Gamma_q^i}\sum_{n\in N_q^i}\frac{1}{T}\sum_{t=1}^{T}\nabla F_n^i(\theta_{q,n,t-1},\xi_{n,t-1})\\&-\frac{1}{\Gamma_q^i}\sum_{n\in N_q^i}\frac{1}{T}\sum_{t=1}^{T}\nabla F_n^i(\theta_{q-\tau_q,n,t-1},\xi_{n,t-1})\|^2\end{aligned}$$

For $i\in S_q$, we get:

$$\begin{aligned}&\sum_{i\in S_q}\mathbb{E}<\nabla F^i(\theta_q),\theta_{q+1}^i-\theta_q^i>+\frac{L}{2}\sum_{i\in S_q}\mathbb{E}\|\theta_{q+1}^i-\theta_q^i\|^2\\&\overset{b}{\le}-\frac{T\gamma}{2}\sum_{i\in S_q}\mathbb{E}\|\nabla F^i(\theta_q)\|^2+\frac{T\gamma}{2}(T_1)\end{aligned}$$

where $b$ follows because: $\frac{L}{2}\gamma^2T^2-\frac{T\gamma}{2}<0\Rightarrow\gamma<\frac{1}{LT}$. Then:

$$\begin{aligned}&\sum_{i\in K-S_q}\mathbb{E}<\nabla F^i(\theta_q),\theta_{q+1}^i-\theta_q^i>=\sum_{i\in K-S_q}\mathbb{E}<\nabla F^i(\theta_q),-\gamma \mathrm{v}_\mathrm{q}^\mathrm{i}>\\&=-\frac{T\gamma}{2}\sum_{i\in K-S_q}\mathbb{E}\|\nabla F^i(\theta_q)\|^2\\&-\frac{T\gamma}{2}\sum_{i\in K-S_q}\mathbb{E}\|\frac{1}{N}\sum_{n=1}^{N}\frac{1}{T}\sum_{t=1}^{T}\nabla F_n^i(\theta_{q-\tau_q,n,t-1},\xi_{n,t-1})\|^2\\&+\frac{T\gamma}{2}\sum_{i\in K-S_q}(T_2)\end{aligned}$$

For another term in $L$-Lipschitzian condition, we have:

$$\begin{aligned}&\frac{L}{2}\sum_{i\in K-S_q}\mathbb{E}\|\theta_{q+1}^i-\theta_q^i\|^2\\&=\frac{L}{2}\gamma^2T^2\sum_{i\in K-S_q}\mathbb{E}\|\frac{1}{N}\sum_{n=1}^{N}\frac{1}{T}\sum_{t=1}^{T}\nabla F_n^i(\theta_{q-\tau_q,n,t-1},\xi_{n,t-1})\|^2\end{aligned}$$

For $i\in K-S_q$, we get:

$$\begin{aligned}&\sum_{i\in K-S_q}\mathbb{E}<\nabla F^i(\theta_q),\theta_{q+1}^i-\theta_q^i>+\frac{L}{2}\sum_{i\in K-S_q}\mathbb{E}\|\theta_{q+1}^i-\theta_q^i\|^2\\&\overset{b}{\le}-\frac{T\gamma}{2}\sum_{i\in K-S_q}\mathbb{E}\|\nabla F^i(\theta_q)\|^2+\frac{T\gamma}{2}(T_2)\end{aligned}$$

Combining $i\in S_q$ and $i\in K-S_q$:

$$\begin{aligned}&\mathbb{E}[F(\theta_{q+1})]-\mathbb{E}[F(\theta_q)]\\&\le\mathbb{E}<\nabla F(\theta_q),\theta_{q+1}-\theta_q>+\frac{L}{2}\mathbb{E}\|\theta_{q+1}-\theta_q\|^2\\&=\sum_{i\in S_q}\mathbb{E}<\nabla F^i(\theta_q),\theta_{q+1}^i-\theta_q^i>+\frac{L}{2}\sum_{i\in S_q}\mathbb{E}\|\theta_{q+1}^i-\theta_q^i\|^2\\&+\sum_{i\in K-S_q}\mathbb{E}<\nabla F^i(\theta_q),\theta_{q+1}^i-\theta_q^i>+\frac{L}{2}\sum_{i\in K-S_q}\mathbb{E}\|\theta_{q+1}^i-\theta_q^i\|^2\\&\le-\frac{T\gamma}{2}\mathbb{E}\|\nabla F(\theta_q)\|^2+\frac{T\gamma}{2}(T_1+T_2)\end{aligned}$$

Then, we can obtain:

$$\begin{aligned}&\mathbb{E}[F(\theta_{Q+1})]-\mathbb{E}[F(\theta_1)]=\sum_{q=1}^{Q}\mathbb{E}[F(\theta_{q+1})]-\sum_{q=1}^{Q}\mathbb{E}[F(\theta_q)]\\&\le-\frac{T\gamma}{2}\sum_{q=1}^{Q}\mathbb{E}\|\nabla F(\theta_q)\|^2+\frac{T\gamma}{2}\sum_{q=1}^{Q}(T_1+T_2)\end{aligned}$$

Re-arranging the terms:

$$\frac{T\gamma}{2}\sum_{q=1}^{Q}\mathbb{E}\|\nabla F(\theta_q)\|^2\le\mathbb{E}[F(\theta_1)]-\mathbb{E}[F(\theta_{Q+1})]+\frac{T\gamma}{2}\sum_{q=1}^{Q}(T_1+T_2)$$

Letting $\frac{1}{Q}\sum_{q=1}^{Q}(\tau_q)^2=\tau$ and dividing both sides by $\frac{T\gamma Q}{2}$

$$\begin{aligned}&\frac{1}{Q}\sum_{q=1}^{Q}\mathbb{E}\|\nabla F(\theta_q)\|^2\le\frac{2\mathbb{E}[F(\theta_1)]}{T\gamma Q}\\&+(48\tau+384+1152\frac{N}{\Gamma^*}L^2+72\frac{N}{\Gamma^*}\tau+72\frac{N}{\Gamma^*}L^2\tau+108(\frac{N}{\Gamma^*})^2L^2\tau)\gamma^2T^2G\\&+(48\gamma^2T+12\frac{N}{\Gamma^*}+6+144\frac{N}{\Gamma^*}L^2\gamma^2T)\sigma^2+128(3+9\frac{N}{\Gamma^*}L^2)\gamma^2T^2\delta^2\\&+8w^2(6+9\frac{N}{\Gamma^*}L^2)\frac{1}{Q}\sum_{q=1}^{Q}\mathbb{E}\|\theta_{q-(\tau_q)}\|^2+72\frac{N}{\Gamma^*}L^2w^2\frac{1}{Q}\sum_{q=1}^{Q}\mathbb{E}\|\theta_q\|^2\end{aligned}$$

Supposing that the step size $\gamma=O(\sqrt{\frac{\Gamma^*}{TQ}})$ and $\sigma$ is sufficiently small, when the constant $C>0$ exists, the convergence rate can be expressed as follows:

$$\frac{1}{Q}\sum_{q=1}^{Q}\mathbb{E}\|\nabla F(\theta_q)\|^2\le C(\frac{1}{\sqrt{\Gamma^*TQ}}+\frac{1}{Q}+\frac{1}{\Gamma^*Q}+\frac{1}{Q^2})$$

All proof details are available on Github [3].

[3]https://github.com/wyy-123-xyy/RA-Fed