

GradMA: A Gradient-Memory-based Accelerated Federated Learning with Alleviated Catastrophic Forgetting

Kangyang Luo, Xiang Li*, Yunshi Lan, Ming Gao
East China Normal University
Shanghai, China

52205901003@stu.ecnu.edu.cn, {xiangli, yslan, mgao}@dase.ecnu.edu.cn

Abstract

Federated Learning (FL) has emerged as a de facto machine learning area and received rapid increasing research interests from the community. However, catastrophic forgetting caused by data heterogeneity and partial participation poses distinctive challenges for FL, which are detrimental to the performance. To tackle the problems, we propose a new FL approach (namely GradMA), which takes inspiration from continual learning to simultaneously correct the server-side and worker-side update directions as well as take full advantage of server’s rich computing and memory resources. Furthermore, we elaborate a memory reduction strategy to enable GradMA to accommodate FL with a large scale of workers. We then analyze convergence of GradMA theoretically under the smooth non-convex setting and show that its convergence rate achieves a linear speed up w.r.t the increasing number of sampled active workers. At last, our extensive experiments on various image classification tasks show that GradMA achieves significant performance gains in accuracy and communication efficiency compared to SOTA baselines. We provide our code here: <https://github.com/lkyddd/GradMA>.

1. Introduction

Federated Learning (FL) [18, 26] is a privacy-preserving distributed machine learning scheme in which workers jointly participate in the collaborative training of a centralized model by sharing model information (parameters or updates) rather than their private datasets. In recent years, FL has shown its potential to facilitate real-world applications, which falls broadly into two categories [10]: the *cross-silo* FL and the *cross-device* FL. The *cross-silo* FL corresponds to a relatively small number of reliable workers, usually organizations, such as healthcare facilities [9] and financial institutions [41], etc. In contrast, for the *cross-*

device FL, the number of workers can be very huge and unreliable, such as mobile devices [26], IoT [27] and autonomous driving cars [22], among others. In this paper, we focus on *cross-device* FL.

The privacy-preserving and communication-efficient properties of the *cross-device* FL make it promising, but it also confronts practical challenges arising from data heterogeneity (i.e., non-iid data distribution across workers) and partial participation [5, 12, 20, 39]. Specifically, the datasets held by real-world workers are generated locally according to their individual circumstances, resulting in the distribution of data on different workers being not identical. Moreover, owing to the flexibility of worker participation in many scenarios (e.g., IoT and mobile devices), workers can join or leave the FL system at will, thus making the set of active workers random and time-varying across communication rounds. Note that we consider a worker participates or is active at round t (i.e., the index of the communication round) if it is able to complete the computation task and send back model information at the end of round t .

The above-mentioned challenges mainly bring catastrophic forgetting (CF) [25, 30, 37] to FL. In a typical FL process, represented by FedAvg [26], a server updates the centralized model by iteratively aggregating the model information from workers that generally is trained over several steps locally before being sent to the server. On the one hand, due to data heterogeneity, the model is updated on private data in local training, which is prone to overfit the current knowledge and forget the previous experience, thus leading to CF [8]. In other words, the updates of the local models are prone to drift and diverge increasingly from the update of the centralized model [12]. This can seriously deteriorate the performance of the centralized model. To ameliorate this issue, a variety of existing efforts regularize the objectives of the local models to align the centralized optimization objective [1, 12, 13, 17, 19]. On the other hand, the server can only aggregate model information from active workers per communication round caused by partial participation. In this case, many existing works directly dis-

*Corresponding author

card [1, 11, 12, 18, 26, 39] or implicitly utilize [7, 28], by means of momentum, the information provided by workers who have participated in the training but dropped out in the current communication round (i.e., stragglers). This results the centralized model, which tends to forget the experience of the stragglers, thus inducing CF. In doing so, the convergence of popular FL approaches (e.g., FedAvg) can be seriously slowed down by stragglers. Moreover, all above approaches solely aggregate the collected information by averaging in the server, ignoring the server’s rich computing and memory resources that could be potentially harnessed to boost the performance of FL [45].

In this paper, to alleviate CF caused by data heterogeneity and stragglers, we bring forward a new FL approach, dubbed as GradMA (**Gradient-Memory-based Accelerated Federated Learning**), which takes inspiration from continual learning (CL) [4, 14, 24, 29, 44] to simultaneously correct the server-side and worker-side update directions and fully utilize the rich computing and memory resources of the server. Concretely, motivated by the success of GEM [24] and OGD [4], two memory-based CL methods, we invoke quadratic programming (QP) and memorize updates to correct the update directions. On the worker side, GradMA harnesses the gradients of the local model in the previous step and the centralized model, and the parameters difference between the local model in the current step and the centralized model as constraints of QP to adaptively correct the gradient of the local model. Furthermore, we maintain a memory state to memorize accumulated update of each worker on the server side. GradMA then explicitly takes the memory state to constrain QP to augment the momentum (i.e., the update direction) of the centralized model. Here, we need the server to allocate memory space to store memory state. However, it may be not feasible in FL scenarios with a large size of workers, which can increase the storage cost and the burden of computing QP largely. Therefore, we carefully craft a memory reduction strategy to alleviate the said limitations. In addition, we theoretically analyze the convergence of GradMA in the smooth non-convex setting.

To sum up, we highlight our contributions as follows:

- We formulate a novel FL approach GradMA, which aims to simultaneously correct the server-side and worker-side update directions and fully harness the server’s rich computing and memory resources. Meanwhile, we tailor a memory reduction strategy for GradMA to reduce the scale of QP and memory cost.
- For completeness, we analyze the convergence of GradMA theoretically in the smooth non-convex setting. As a result, the convergence result of GradMA achieves the linear speed up as the number of selected active workers increases.

- We conduct extensive experiments on four commonly used image classification datasets (i.e., MNIST, CIFAR-10, CIFAR-100 and Tiny-Imagenet) to show that GradMA is highly competitive compared with other state-of-the-art baselines. Meanwhile, ablation studies demonstrate efficacy and indispensability for core modules and key parameters.

2. Related Work

FL with Data Heterogeneity. FedAvg, the classic distributed learning framework for FL, is first proposed by McMahan et al. [26]. Although FedAvg provides a practical and simple solution for aggregation, it still suffers performance deterioration when the data among workers is non-iid [18]. Shortly thereafter, a panoply of modifications for FedAvg have been proposed to handle said issue. For example, FedProx [19] constrains local updates via adding a proximal term to the local objectives. Scaffold [12] uses control variate to augment the local updates. FedDyn [1] dynamically regularizes the objectives of workers to align global and local objectives. Moon [17] corrects the local training by conducting contrastive learning in model-level. Meanwhile, there exists another line of works to improve the global performance of FL through performing knowledge distillation [13, 23, 42, 45, 46] on the server side or worker side. FedMLB [13] architecturally regularizes the local objectives via online knowledge distillation. However, other approaches incur additional communication overhead [42, 46] or pseudo data [23, 45]. Going beyond the aforementioned approaches, FL with momentum is an effective way to tackle worker drift problem caused by data heterogeneity and accelerate the convergence. Specifically, on the server side, FedAvgM [7] maintains a momentum buffer, whereas FedADAM [28] and FedAMS [33] both adopt adaptive gradient-descent methods to speed up training. FedCM [38] keeps a state, carrying global information broadcasted by the server, on the worker side to address data heterogeneity issue. DOMO [36] and Mime [11] maintain momentum buffers on both server side and worker side to improve the training performance.

FL with Partial Participation. In addition to data heterogeneity issue, another key hurdle to FL stems from partial participation. The causes for partial participation can be roughly classified into two categories. One is the difference in the computing power and communication speed of different workers. A natural way to cope with this situation is to allow asynchronous updates [2, 34, 40]. The other is the different availability mode, in which workers can abort the training midway (i.e., stragglers) [10]. To do so, many approaches may collect information from only a subset of workers to update the centralized model [1, 7, 11, 12, 18, 26, 39]. However, the server in the mentioned approaches simply ignores and discards the information of the stragglers,

which can lead to other problems such as under-utilization of computation and memory [45], slower convergence [18], and biased/unfair use of workers’ information [10]. Recently, MIFA [5] corrects the gradient bias by exploiting the memorized latest updates of all workers, which avoids excessive delays caused by inactive workers and mitigates CF to some extent.

Continual Learning. CL is a training paradigm that focuses on scenarios with a continuously changing class distribution of each task and aims at overcoming CF. Existing works for CL can be roughly divided into three branches: expansion-based methods [21, 44], regularization-based methods [14, 32] and memory-based methods [4, 24, 29]. Note that unlike CL, we focus on alleviating CF in distributed data, not sequential data. There are a handful of recent studies that consider FL with CL. For example, FedWeIT [43] focuses on sequential data. FedCurv [30] trains objectives based on all-reduce protocol. FedReg [37] and FCCL [8] require generated pseudo data and public data, respectively.

3. Preliminaries

This section defines the objective function for FL and introduces QP.

In practice, FL is designed to minimize the empirical risk over data distributed across multiple workers without compromising local data. The following optimization problem is often considered:

$$\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N \left[f_i(\mathbf{x}) = \frac{1}{n_i} \sum_{r=1}^{n_i} F_i(\mathbf{x}; \xi_r^{(i)}) \right], \quad (1)$$

where N is the number of workers. Moreover, the local objective $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ measures the local empirical risk over data distribution \mathcal{D}_i , i.e., $\xi_r^{(i)} \sim \mathcal{D}_i$, with n_i samples available at i -th worker. Note that \mathcal{D}_i can be different among workers. In this work, we consider the typical centralized setup where N workers are connected to one central server.

Next, we introduce QP, which is a fundamental optimization problem with well-established solutions and can be widely seen in the machine learning community, to correct the server-side and work-side update directions. In this paper, we can model our goal via QP, which is posed in the following primal form:

$$\min_{\tilde{\mathbf{p}}} \frac{1}{2} \|\mathbf{p} - \tilde{\mathbf{p}}\|^2 \quad \text{s.t.} \quad \langle \tilde{\mathbf{p}}, \mathbf{M}[i] \rangle \geq 0, \forall i \in [C], \quad (2)$$

where $\mathbf{p} \in \mathbb{R}^d$ and $\mathbf{M} \in \mathbb{R}^{d \times C}$ ($C \in \mathbb{N}$). One can see that the goal of (2) is to seek a vector $\tilde{\mathbf{p}}$ that is positively correlated with $\mathbf{M}[i] \in \mathbb{R}^d, \forall i \in [C]$ while being close to \mathbf{p} . By discarding the constant term $\mathbf{p}^\top \mathbf{p}$, we rewrite (2) as:

$$\min_{\tilde{\mathbf{p}}} \frac{1}{2} \tilde{\mathbf{p}}^\top \tilde{\mathbf{p}} - \mathbf{p}^\top \tilde{\mathbf{p}} \quad \text{s.t.} \quad \mathbf{M}^\top \tilde{\mathbf{p}} \succeq \mathbf{0} \in \mathbb{R}^C. \quad (3)$$

Algorithm 1 GradMA: A Gradient-Memory-based Accelerated Federated Learning

- 1: **Input:** learning rates (η_l, η_g) , the number of all workers N , the number of sampled active workers per communication round S , control parameters (β_1, β_2) , synchronization interval I and memory size m ($S \leq m \leq \min\{d, N\}$).
 - 2: Initial state $\mathbf{x}_0^{(i)} = \mathbf{x}_0 \in \mathbb{R}^d (\forall i \in [N]), \tilde{\mathbf{m}}_0 = \mathbf{0}$.
 - 3: Initial counter = $\{c(i) = 0, \forall i \in [N]\}$.
 - 4: Initial memory state $\mathbf{D} = \{\}$.
 - 5: $buf = \{\}, new_buf = \{\}$.
 - 6: **for** $t = 0, 1, \dots, T - 1$ **do**
 - 7: **On server:**
 - 8: Server samples a subset \mathcal{S}_t with S active workers and transmits \mathbf{x}_t to \mathcal{S}_t .
 - 9: $counter, \mathbf{D}, buf, new_buf \leftarrow \text{mem_red}(m, \mathcal{S}_t, counter, \mathbf{D}, buf, new_buf)$.
 - 10: **On workers:**
 - 11: **for** $i \in \mathcal{S}_t$ **parallel do**
 - 12: $\mathbf{x}_{t+1}^{(i)} = \text{Worker_Update}(\mathbf{x}_t^{(i)}, \mathbf{x}_t, \eta_l, I)$,
 - 13: sends $\mathbf{d}_{t+1}^{(i)} = \mathbf{x}_t - \mathbf{x}_{t+1}^{(i)}$ to server.
 - 14: **end for**
 - 15: **On server:**
 - 16: $\mathbf{D}, \mathbf{x}_{t+1}, \tilde{\mathbf{m}}_{t+1} = \text{Server_Update}([\mathbf{d}_{t+1}^{(i)}, i \in \mathcal{S}_t], \tilde{\mathbf{m}}_t, \mathbf{D}, \eta_g, \beta_1, \beta_2, buf, new_buf)$.
 - 17: Sends \mathbf{x}_{t+1} to sampled active workers in the next round.
 - 18: $new_buf = \{\}$.
 - 19: **end for**
 - 20: **Output:** \mathbf{x}_T
-

However, this is a QP problem on d variables, which are updates of the model. Generally, d can be enormous, resulting in d being much larger than C . We thus solve the dual formulation of the above QP problem:

$$\min_{\mathbf{z}} \frac{1}{2} \mathbf{z}^\top \mathbf{M}^\top \mathbf{M} \mathbf{z} + \mathbf{p}^\top \mathbf{M} \mathbf{z} \quad \text{s.t.} \quad \mathbf{z} \succeq \mathbf{0} \in \mathbb{R}^C. \quad (4)$$

Once we solve for the optimal dual variable \mathbf{z}^* , we can recover the optimal primal solution as $\tilde{\mathbf{p}} = \mathbf{M} \mathbf{z}^* + \mathbf{p}$.

4. Proposed Approach: GradMA

We now present the proposed FL approach GradMA, see Alg. 1 for complete pseudo-code. Note that the communication cost of GradMA is the same as that of FedAvg. Next, we detail core modules of GradMA, which include the memory reduction strategy (i.e., `mem_red()`), `Worker_Update()` and `Server_Update()` on lines 9, 12 and 16 of Alg. 1, respectively.

Algorithm 2 Worker_Update($\mathbf{x}', \mathbf{x}, \eta_l, I$)

- 1: Sets $\mathbf{x}_{-1}^{(i)} = \mathbf{x}', \mathbf{x}_0^{(i)} = \mathbf{x}$.
 - 2: **for** $\tau = 0, 1, \dots, I - 1$ **do**
 - 3: $\mathbf{g}_\tau^{(i)} = \nabla f_i(\mathbf{x}_\tau^{(i)})$,
 - 4: $\mathbf{G}_\tau^{(i)} = [\nabla f_i(\mathbf{x}_{\tau-1}^{(i)}), \nabla f_i(\mathbf{x}), \mathbf{x}_\tau^{(i)} - \mathbf{x}_t]$,
 - 5: $\tilde{\mathbf{g}}_\tau^{(i)} = \text{QP}_l(\mathbf{g}_\tau^{(i)}, \mathbf{G}_\tau^{(i)})$,
 - 6: $\mathbf{x}_{\tau+1}^{(i)} = \mathbf{x}_\tau^{(i)} - \eta_l \tilde{\mathbf{g}}_\tau^{(i)}$.
 - 7: **end for**
 - 8: **Output:** $\mathbf{x}_I^{(i)}$.
-

4.1. Correcting gradient for the worker side

Throughout the local update, we leverage QP to perform correcting gradient directions, see Alg. 2. Here, the input of QP (marked as QP_l for distinction) is $\mathbf{p} \leftarrow \mathbf{g}_\tau^{(i)}$ and $\mathbf{M} \leftarrow \mathbf{G}_\tau^{(i)} \in \mathbb{R}^{d \times 3}$ (line 5 of Alg. 2), and its output is the following vector $\tilde{\mathbf{g}}_\tau^{(i)}$, which is positively correlated with $\nabla f_i(\mathbf{x}_{\tau-1}^{(i)})$, $\nabla f_i(\mathbf{x})$ and $\mathbf{x}_\tau^{(i)} - \mathbf{x}_t$ while ensuring the minimum $\|\mathbf{g}_\tau^{(i)} - \tilde{\mathbf{g}}_\tau^{(i)}\|$:

$$\begin{aligned} \tilde{\mathbf{g}}_\tau^{(i)} &= \mathbf{G}_\tau^{(i)} \mathbf{z}_\tau^* + \mathbf{g}_\tau^{(i)} \\ &= z_{\tau,1}^* \nabla f_i(\mathbf{x}_{\tau-1}^{(i)}) + z_{\tau,2}^* \nabla f_i(\mathbf{x}) + z_{\tau,3}^* (\mathbf{x}_\tau^{(i)} - \mathbf{x}_t) + \mathbf{g}_\tau^{(i)}, \end{aligned} \quad (5)$$

where $\mathbf{z}_\tau^* = [z_{\tau,1}^*, z_{\tau,2}^*, z_{\tau,3}^*]^\top$ and $\mathbf{z}_\tau^* \succeq \mathbf{0} \in \mathbb{R}^3$. Essentially, the output of QP_l is a conical combination and serves as an update direction for local training. Particularly, when $z_{\tau,1}^* = 0$, $z_{\tau,2}^* = 0$ and $z_{\tau,3}^* > 0$, Eq. (5) is equivalent to the local update of FedProx [19]. The difference is that the control parameter μ in FedProx is a hyper-parameter, while $z_{\tau,3}^*$ is determined adaptively by QP_l . Specifically, when $\mathbf{g}_\tau^{(i)}$ is positively correlated with $\mathbf{x}_\tau^{(i)} - \mathbf{x}_t$, i.e., $\langle \mathbf{g}_\tau^{(i)}, \mathbf{x}_\tau^{(i)} - \mathbf{x}_t \rangle \geq 0$, $z_{\tau,3}^*$ is approximately equal to 0; otherwise, $z_{\tau,3}^*$ is greater than 0. In other words, $\mathbf{x}_\tau^{(i)} - \mathbf{x}_t$ acts as a hard constraint only when $\mathbf{g}_\tau^{(i)}$ is negatively correlated with $\mathbf{x}_\tau^{(i)} - \mathbf{x}_t$, which makes $\tilde{\mathbf{g}}_\tau^{(i)}$ focus more on local information. Moreover, the calculation mechanisms for $z_{\tau,1}^*$ and $z_{\tau,2}^*$ are the same as that for $z_{\tau,3}^*$. When $z_{\tau,1}^* > 0$ and $z_{\tau,2}^* > 0$, it indicates that the update direction $\tilde{\mathbf{g}}_\tau^{(i)}$ takes into account the previous step and global information about the model, which is inspired by CL [4, 24]. Intuitively, Eq. (5) adaptively taps previous and global knowledge, thus effectively mitigating CF caused by data heterogeneity.

4.2. Correcting update direction for the server side

Now, we describe the proposed update process of the centralized model on the server side, see Alg. 3 for details. For ease of presentation, we define the number of local updates of workers that the server can store as the memory size m . To elaborate, we assume that there is enough memory

space on the server such that $m = N$. In this way, at communication round t , the update process can be streamlined, which takes the form:

$$\mathbf{d}_{t+1} = \frac{1}{S} \sum_{i \in \mathcal{S}_t} \mathbf{d}_{t+1}^{(i)}, \mathbf{m}_{t+1} = \beta_1 \tilde{\mathbf{m}}_t + \mathbf{d}_{t+1}, \quad (6)$$

$$\mathbf{D}[i] \leftarrow \begin{cases} \beta_2 \mathbf{D}[i] + \mathbf{d}_{t+1}^{(i)}, & i \in \mathcal{S}_t \\ \beta_2 \mathbf{D}[i], & i \notin \mathcal{S}_t \end{cases}, \quad (7)$$

$$\tilde{\mathbf{m}}_{t+1} = \text{QP}_g(\mathbf{m}_{t+1}, \mathbf{D}), \mathbf{x}_{t+1} = \mathbf{x}_t - \eta_g \tilde{\mathbf{m}}_{t+1}. \quad (8)$$

As shown in Eq. (7), we propose that the server allocates memory space to maintain a memory state \mathbf{D} , which is updated in a momentum-like manner to memorize the accumulated updates of all workers. Each worker only uploads update $\mathbf{d}_{t+1}^{(i)}$ ($i \in [N]$) to the server, and as such the risk of data leakage is greatly reduced. By memorizing accumulated updates of inactive workers, GradMA avoids waiting for any straggler when facing heterogeneous workers with different availability, so as to effectively overcome the adverse effects caused by partial participation. This is different from the recently proposed MIFA [5] (see Alg. 6 in Appendix .1), which stores the latest updates of all workers to perform averaging. However, such a straightforward and naive implementation of integration implicitly increases statistical heterogeneity in situations where different workers have varying data distributions, which can induce bias.

Therefore, the core idea of this paper is how to leverage memorized information to overcome the above challenge effectively. To tackle the challenge, we apply QP (marked as QP_g for distinction) to seek an update direction $\tilde{\mathbf{m}}_{t+1}$ that is positively correlated with buffers $\mathbf{D}[i] \in \mathbb{R}^d, \forall i \in [N]$ while being close to \mathbf{m}_{t+1} . Concretely, the input of QP_g is $\mathbf{p} \leftarrow \mathbf{m}_{t+1}$ and $\mathbf{M} \leftarrow \mathbf{D} \in \mathbb{R}^{d \times N}$, and its output is $\tilde{\mathbf{m}}_{t+1}$ (see Eq. (8)), which takes the form $\tilde{\mathbf{m}}_{t+1} = \mathbf{D} \mathbf{z}_{t+1}^* + \mathbf{m}_{t+1}$, where $\mathbf{z}_{t+1}^* = [z_{t+1,1}^*, \dots, z_{t+1,N}^*]^\top \succeq \mathbf{0} \in \mathbb{R}^N$ is determined adaptively by QP_g . Inherently, QP_g takes advantage of the accumulated updates of all workers stored on \mathbf{D} to correct the update direction \mathbf{m}_{t+1} and circumvents the centralized model from forgetting stragglers' knowledge, thereby alleviating CF induced by partial participation. In particular, one can easily observe that $\tilde{\mathbf{m}}_{t+1} = \mathbf{m}_{t+1}$ holds if $m = 0$ (that is, $\mathbf{D} = \mathbf{0}$). The update process of Alg. 3 is then consistent with that of FedAvgM [7] on the server side. Consequently, Alg. 3 can be considered as an extension of FedAvgM in terms of augmenting updates through allocating memory.

4.3. A Practical Memory Reduction Strategy

It is well known that in realistic FL scenarios, on the one hand, the number of workers may be large; the size of the model may be huge on the other hand, leading to large-scale QP as well as high memory demanding for server to store \mathbf{D} , which is infeasible and unnecessary in practice.

Algorithm 3 Server_Update($(\mathbf{d}_{t+1}^{(i)}, i \in \mathcal{S}_t], \tilde{\mathbf{m}}_t, \mathbf{D}, \eta_g, \beta_1, \beta_2, \text{buf}, \text{new_buf})$)

```

1:  $\mathbf{d}_{t+1} = \frac{1}{S} \sum_{i \in \mathcal{S}_t} \mathbf{d}_{t+1}^{(i)}, \mathbf{m}_{t+1} = \beta_1 \tilde{\mathbf{m}}_t + \mathbf{d}_{t+1}$ .
2: for  $c(i) \in \text{buf}$  do
3:   if  $i \in \mathcal{S}_t$  then
4:      $\mathbf{D}[i] \leftarrow \begin{cases} \beta_2 \mathbf{D}[i] + \mathbf{d}_{t+1}^{(i)}, c(i) \notin \text{new\_buf} \\ \mathbf{d}_{t+1}^{(i)}, c(i) \in \text{new\_buf} \end{cases}$ .
5:   else if  $i \notin \mathcal{S}_t$  then
6:      $\mathbf{D}[i] \leftarrow \beta_2 \mathbf{D}[i]$ .
7:   end if
8: end for
9:  $\tilde{\mathbf{m}}_{t+1} = \text{QP}_g(\mathbf{m}_{t+1}, \mathbf{D}), \mathbf{x}_{t+1} = \mathbf{x}_t - \eta_g \tilde{\mathbf{m}}_{t+1}$ .
10: Output:  $\mathbf{D}, \mathbf{x}_{t+1}, \tilde{\mathbf{m}}_{t+1}$ .

```

Algorithm 4 mem_red($m, \mathcal{S}, c, \mathbf{D}, \text{buf}, \text{new_buf}$)

```

1: for  $i \in \mathcal{S}$  do
2:   if  $c(i) \in \text{buf}$  then
3:      $c(i) \leftarrow c(i) + 1$ .
4:   else if  $c(i) \notin \text{buf}$  then
5:     if  $\text{Length}(\text{buf}) = m$  then
6:        $\text{old\_buf} = \{\}$ .
7:       for  $k \in \text{buf}$  do
8:         if  $k \notin \mathcal{S}$  then
9:            $\text{old\_buf} \leftarrow \text{old\_buf} \cup \{c(k)\}$ .
10:        end if
11:       end for
12:       Discarding  $c(i')$  with the smallest value from  $\text{old\_buf}$  and set  $c(i') = 0$ .
13:       Discarding  $\mathbf{D}[i']$  from memory state  $\mathbf{D}$ .
14:     end if
15:      $c(i) \leftarrow c(i) + 1$ .
16:      $\text{buf} \leftarrow \text{buf} \cup \{c(i)\}$ .
17:      $\text{new\_buf} \leftarrow \text{new\_buf} \cup \{c(i)\}$ .
18:   end if
19: end for
20: Output:  $c, \mathbf{D}, \text{buf}, \text{new\_buf}$ 

```

Therefore, we propose a memory reduction strategy to alleviate this deficiency, which ensures that the size of \mathbf{D} does not exceed a pre-given m and $S \leq m \leq \min\{d, N\}$, see Alg. 4 for details. The design ethos of the memory reduction strategy is to keep as much useful information as possible in a given m . Specifically, at communication round t , the server samples S active workers and performs that $c(i) \leftarrow c(i) + 1$ ($i \in \mathcal{S}$) (lines 3 and 15 of Alg. 4). When the memory used is less than the given one, $c(i) \notin \text{buf}$ of sampled active workers enter the buffers buf and new_buf in turn (lines 16-17 of Alg. 4). Once the memory used is equal to the given one, $c(i')$ with the smallest value in old_buf is discarded and set $c(i') = 0$. Also, $\mathbf{D}[i']$ is discarded from

\mathbf{D} (lines 12-13 of Alg. 4).

5. Convergence Results for GradMA

We now present a convergence analysis of GradMA in the smooth non-convex setting. And the following assumptions are considered.

Assumption 1 (Global function below bounds). *Set $f^* = \inf_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x})$ and $f^* > -\infty$.*

Assumption 2 (L -smooth). $\forall i \in [N]$, *the local functions f_i are differentiable, and there exist constant $L > 0$ such that for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$.*

Assumption 3 (Bounded data heterogeneity). *The degree of heterogeneity of the data distribution across workers can be quantified as $\|\nabla f_i(\mathbf{x}) - \nabla f(\mathbf{x})\|^2 \leq \rho^2$, for any $i \in [N]$ and some constant $\rho \geq 0$.*

Assumption 4 (Bounded optimal solution error for QP_l). *Given $\mathbf{g}^{(i)} = \nabla f_i(\mathbf{x})$ (see Alg. 2), then there exists $\varepsilon_l > 0$ such that $\|\mathbf{g}^{(i)} - \tilde{\mathbf{g}}^{(i)}\|^2 \leq \varepsilon_l^2$.*

Assumption 5 (Bounded optimal solution error for QP_g). *Given $\beta_2 \in [0, 1)$ and \mathbf{m} (see Alg. 3), then there exists $\varepsilon_g > 0$ such that $\|\mathbf{m} - \tilde{\mathbf{m}}\|^2 \leq \frac{\varepsilon_g^2}{1 - \beta_2}$.*

Assumptions 1 and 2 are commonly used in the analysis of distribution learning [12, 19, 35]. Assumption 3 quantifies inter-worker variances, i.e., data heterogeneity [12, 19]. Assumptions 4 and 5 are necessary for the theoretical analysis of GradMA, which constrain the upper bound on the optimal solution errors of QP_l and QP_g , respectively. Intuitively, the assumptions hold if the local updates for all worker make sense [3]. Note that the upper bound for Assumption 5 follows an intuitive observation: more accumulated updates of workers (i.e., the larger β_2) can provide more accumulated update information for the centralized model. Next, we state our convergence results for GradMA.

Theorem 1 *Assume Assumptions 1-5 exist. Let $\eta_l \leq \frac{1}{160^{0.5} L I}$, $\eta_g \eta_l \leq \frac{(1 - \beta_1)^2 S(N - 1)}{I L (\beta_1 S(N - 1) + 4N(S - 1))}$ and $320 I^2 \eta_l^2 L^2 + \frac{64 I \eta_g \eta_l L (1 + 40 I^2 \eta_l^2 L^2)}{(1 - \beta_1)^2} \frac{N - S}{S(N - 1)} \leq 1$. For all $t \in [0, \dots, T - 1]$, the following relationship generated by Alg. 1 holds:*

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \left[\|\nabla f(\mathbf{x}_t)\|^2 \right] \leq \frac{8(1 - \beta_1)(f(\mathbf{x}_0) - f^*)}{I \eta_g \eta_l T} + C_1 \varepsilon_l^2 + C_2 \varepsilon_g^2 + C_3 \rho^2,$$

where the expectation \mathbb{E} is w.r.t the sampled active workers per communication round, and $C_1 = 8 + 320 I^2 \eta_l^2 L^2 + \frac{64 I \eta_g \eta_l L (1 + 40 I^2 \eta_l^2 L^2)}{(1 - \beta_1)^2} \frac{N - S}{S(N - 1)}$, $C_2 = \frac{20 \eta_g L}{(1 - \beta_1)^2 (1 - \beta_2) I \eta_l} + \frac{8}{(1 - \beta_2) I^2 \eta_l^2}$, $C_3 = C_1 - 8$.

A detailed proof of Theorem 1 is presented in the Appendix .3.

Corollary 1 Assume Assumptions 1-5 exist. We set $\eta_l = \frac{1}{T^{0.5}LI}$, $\eta_g = \frac{S^{0.5}}{I^{0.5}}$, $\varepsilon_l = \frac{1}{T^{0.5}}$ and $\varepsilon_g = \frac{I^{0.25}}{T^{0.75}S^{0.25}L}$. For $T \geq \max \left\{ 160, \frac{(\beta_1 S(N-1) + 4N(S-1))^2}{I^2(1-\beta_1)^4 S(N-1)^2}, \frac{(b+(b^2+1280)^{0.5})^2}{4} \right\}$ where $b = \frac{128(N-S)}{(1-\beta_1)^2 I^{0.5} S^{0.5} (N-1)}$ in Theorem 1, we have:

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} [\|\nabla f(\mathbf{x}_t)\|^2] = \mathcal{O} \left(\frac{I^{0.5}}{S^{0.5} T^{0.5}} + \frac{1}{T} \right).$$

An immediate observation from Corollary 1 is that GradMA can achieve the linear speed up as the number of sampled active workers S increases. This convergence rate matches the well-known best result in FL approaches in literature [39] under the smooth non-convex setting.

6. Empirical Study

In this section, we empirically investigate GradMA on four datasets (MNIST [16], CIFAR-10, CIFAR-100 [15] and Tiny-Imagenet¹) commonly used for image classification tasks.

6.1. Experimental Setup

To gauge the effectiveness of Worker.Update() and Server.Update(), we perform ablation study of GradMA. For this purpose, we design Alg. 7 (marked as GradMA-W) and Alg. 8 (marked as GradMA-S), as specified in Appendix .1. Meanwhile, we compare other baselines, including FedAvg [26], FedProx [19], MOON [17], FedMLB [13], Scaffold [12], FedDyn [1], MimeLite [11], MIFA [5] and slow-momentum variants of FedAvg, FedProx, MIFA, MOON and FedMLB (i.e., FedAvgM [7], FedProxM, MIFAM, MOONM and FedMLBM), in terms of test accuracy and communication efficiency in different FL scenarios. For fairness, we divide the baselines into three groups based on FedAvg’s improvements on the **worker side**, **server side**, or **both**. See Table 1 and Table 2 for details. Furthermore, on top of GradMA-S, we empirically study the effect of the control parameters (β_1 , β_2) and verify the effectiveness of men_red() by setting varying memory sizes m .

All our experiments are performed on a centralized network with 100 workers. And we fix synchronization interval $I = 5$. To explore the performances of the approaches, we set up multiple different scenarios w.r.t. the number of sampled active workers S per communication round and data heterogeneity. Specifically, we set $S \in \{5, 10, 50\}$. Moreover, we use Dirichlet process $Dp(\omega)$ [1,46] to strictly partition the training set of each dataset across 100 workers. We set $\omega \in \{0.01, 0.1, 1.0\}$. A visualization of the

¹<http://cs231n.stanford.edu/tiny-imagenet-200.zip>

data partitions for the four datasets at varying ω values can be found in Fig. 8 in Appendix .2. For MNIST, a neural network (NN) with three linear hidden layers is implemented for each worker. For CIFAR-10 (CIFAR-100, Tiny-Imagenet), each worker runs a Lenet-5 [16] (VGG-11 [31], Resnet20 [6]) architecture. Due to the space limitation, we relegate detailed hyper-parameters tuning and full experimental results to Appendix .2.

6.2. Performance Analysis

Effects of data heterogeneity. From Table 1, one can see that the performances of all approaches degrade severely with decreasing ω on MNIST, CIFAR-10 and Tiny-Imagenet, with GradMA being the only approach that is robust while surpasses other baselines with an overwhelming margin against most scenarios. In particular, the higher data heterogeneity, the more superior performance for GradMA. Also, as shown in Table 2 and Fig. 3, GradMA requires much less communication rounds to reach a given test accuracy compared to baselines against most scenarios. These results validate our idea in the sense that the advantage of GradMA comes from the effective adaptive utilization of workers’ information on both the worker side and server side, which alleviates negative impacts caused by the discrepancy of data distributions among workers.

Impacts of stragglers. We explore the impacts of different S on MNIST, CIFAR-100 and Tiny-Imagenet. A higher S means more active workers upload updates per communication round. From Table 1, we can clearly see that the performance of all approaches improves uniformly with increasing S on CIFAR-100 and Tiny-Imagenet, where GradMA consistently dominates other baselines in terms of test accuracy. Meanwhile, Fig. 3 shows that the learning efficiency of GradMA consistently outperforms other baselines (see Appendix .2 for more results). However, for MNIST, the test accuracy for most of the approaches does not intuitively improve with increasing S . We conjecture that for simple classification tasks and models, the more active workers participating in training, the more prone the centralized model is to overfitting.

Comments on GradMA-W and GradMA-S. We now

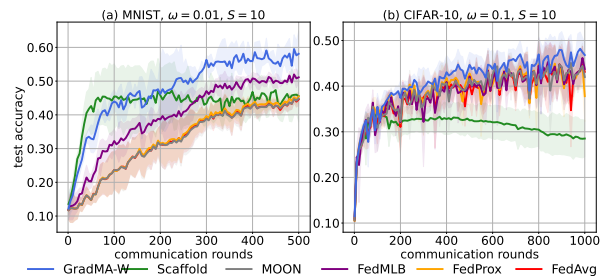


Figure 1. Test accuracy curves selected of GradMA-W as well as baselines over MNIST and CIFAR-10.

Table 1. Top test accuracy (%) overview given different FL scenarios.

Alg.s	MNIST+NN, $S = 10$			MNIST+NN, $\omega = 0.01$		CIFAR-10+Lenet-5, $S = 10$			CIFAR-100+VGG-11, $\omega = 0.1$			Tiny-Imagenet+Resnet20, (ω, S)		
	$\omega = 1.0$	$\omega = 0.1$	$\omega = 0.01$	$S = 5$	$S = 50$	$\omega = 1.0$	$\omega = 0.1$	$\omega = 0.01$	$S = 5$	$S = 10$	$S = 50$	(0.01, 5)	(1.0, 5)	(1.0, 10)
FedAvg	98.22±0.05	97.11±0.39	46.19±1.29	49.65±3.88	68.32±4.16	69.48±8.28	47.86±5.26	20.97±3.73	56.02±0.37	61.22±0.16	64.78±0.43	7.50±0.32	41.80±0.55	42.90±0.12
FedProx	98.16±0.09	97.19±0.31	46.82±0.96	49.89±3.67	67.97±4.27	71.58±4.66	48.63±4.92	20.40±3.85	55.94±0.71	61.25±0.09	64.69±0.27	7.51±0.46	41.82±0.29	42.58±0.69
FedMLB	98.31±0.06	97.26±0.40	54.53±0.39	57.22±3.07	68.44±3.26	69.28±6.56	48.99±4.94	20.81±3.26	53.80±0.16	59.20±0.27	64.06±0.30	7.98±0.34	42.83±0.13	43.59±0.80
MOON	98.18±0.12	97.11±0.31	46.26±1.35	50.39±5.16	68.75±4.50	71.11±7.94	48.84±5.16	19.39±3.99	55.37±0.34	60.58±0.60	64.48±0.42	7.70±0.38	41.68±0.22	42.80±0.54
Scaffold	97.63±0.37	93.94±1.18	50.86±7.46	39.97±4.88	49.54±2.28	53.33±6.63	35.91±2.14	15.55±1.33	32.22±0.92	34.72±0.80	45.70±0.76	7.20±0.33	40.96±0.23	43.02±0.30
GradMA-W	98.15±0.10	97.01±0.23	63.34±3.75	65.39±0.96	65.13±2.54	72.33±3.84	50.25±3.94	18.99±4.06	56.43±0.51	61.38±0.11	64.96±0.36	9.98±0.22	43.68±0.23	44.57±0.45
FedAvgM	98.29±0.18	97.20±0.30	53.77±0.32	57.87±3.64	67.80±5.58	71.04±7.29	51.91±4.46	21.02±3.52	55.85±0.28	61.32±0.29	64.88±0.25	16.96±1.08	41.91±0.23	42.57±0.14
MIFA	98.02±0.12	96.88±0.56	66.92±2.53	56.04±3.92	52.84±4.89	71.41±5.81	50.60±11.87	23.78±2.04	50.37±1.02	58.74±0.42	64.71±0.31	8.88±0.33	41.42±0.22	42.83±0.13
MIFAM	98.02±0.15	96.90±0.44	67.15±2.23	55.28±6.05	53.35±6.84	73.48±1.37	52.13±9.71	24.17±1.24	49.30±0.86	58.91±0.24	64.61±0.33	12.01±0.32	41.94±0.06	43.17±0.09
GradMA-S	98.38±0.09	97.35±0.28	74.52±1.71	75.93±0.97	69.09±3.83	78.76±1.96	64.60±5.87	28.41±2.43	59.08±0.43	63.23±0.22	65.63±0.35	20.93±1.49	48.83±1.06	49.65±0.72
FedProxM	98.26±0.08	97.13±0.34	54.50±0.79	58.59±4.58	69.00±4.42	78.00±1.61	51.22±5.14	21.80±3.72	55.63±0.31	63.15±0.12	64.78±0.11	18.30±0.79	37.98±0.10	45.27±0.19
FedMLBM	98.26±0.16	97.35±0.30	61.12±1.48	64.12±4.17	68.78±3.28	73.70±4.62	49.90±5.82	21.53±2.93	53.91±0.78	60.44±0.34	64.85±0.18	17.32±0.82	44.62±0.32	45.18±0.27
MOONM	98.21±0.13	97.04±0.42	62.34±8.91	57.98±5.51	68.82±4.43	73.96±4.11	50.06±6.14	20.19±3.10	56.01±0.25	62.06±0.19	65.37±0.17	16.78±0.95	42.43±0.39	42.78±0.46
Feddyn	97.92±0.12	96.03±0.46	59.39±2.29	65.36±5.20	57.68±4.30	74.94±2.48	41.93±2.22	17.94±3.52	52.95±1.63	58.48±0.18	61.71±0.25	17.89±0.95	44.37±0.57	44.86±0.15
MimeLite	98.19±0.07	97.10±0.31	54.86±13.36	51.04±4.15	69.41±4.15	77.98±1.48	53.27±1.69	20.73±3.33	58.00±0.51	63.29±0.49	64.68±0.33	8.29±0.29	41.05±0.21	41.56±0.18
GradMA	98.39±0.04	97.34±0.35	77.97±1.28	75.51±1.94	66.68±3.03	79.92±0.59	65.91±5.10	30.81±1.78	59.47±0.58	63.49±0.47	65.68±0.25	23.52±1.32	49.29±0.86	50.54±0.56

Table 2. Communication rounds to reach given test accuracy ac under different FL scenarios. Note that since Scaffold and MimeLite have twice as much communication load per communication round as the other approaches, we use $2\times$ to show the distinction.

Alg.s	MNIST, $S = 10$			MNIST, $\omega = 0.01$		CIFAR-10, $S = 10$			CIFAR-100, $\omega = 1.0$			Tiny-Imagenet+Resnet20, (ω, S)		
	$w = 1.0$ $ac = 95\%$	$w = 0.1$ $ac = 95\%$	$w = 0.01$ $ac = 45\%$	$S = 5$ $ac = 40\%$	$S = 50$ $ac = 50\%$	$w = 1.0$ $ac = 55\%$	$w = 0.1$ $ac = 45\%$	$w = 0.01$ $ac = 15\%$	$S = 5$ $ac = 30\%$	$S = 10$ $ac = 40\%$	$S = 50$ $ac = 60\%$	(0.01, 5) $ac = 5\%$	(1.0, 5) $ac = 30\%$	(1.0, 10) $ac = 35\%$
FedAvg	25	115	493	299	117	177	882	106	437	435	1,071	986	906	1,116
FedProx	25	120	478	283	117	141	766	197	429	435	1,081	986	906	1,091
FedMLB	22	115	280	203	78	245	694	63	511	548	1,404	791	806	966
MOON	25	116	493	283	117	96	616	247	475	458	1,088	986	906	1,116
Scaffold ($2\times$)	26	-	60	-	-	-	-	-	806	-	-	971	971	1,181
GradMA-W	30	111	125	78	172	168	582	136	389	420	1,027	821	841	936
FedAvgM	19	116	280	138	79	175	605	73	429	397	654	286	821	991
MIFA	42	85	67	62	89	156	393	54	631	567	1,075	716	1,021	1,151
MIFAM	39	80	60	62	80	152	325	25	677	547	883	656	956	1,076
GradMA-S	16	43	38	38	56	83	101	14	297	266	579	131	231	251
FedProxM	22	115	280	138	96	97	631	67	468	397	665	291	821	611
FedMLBM	16	87	191	113	77	185	653	69	522	522	922	296	681	796
MOONM	19	115	186	210	95	78	766	85	461	417	697	291	821	991
Feddyn	19	73	136	128	96	66	-	83	441	356	736	371	396	406
MimeLite ($2\times$)	32	115	277	284	113	92	385	134	355	329	675	1,016	871	1,071
GradMA	18	49	37	29	105	49	130	11	274	253	559	146	231	231

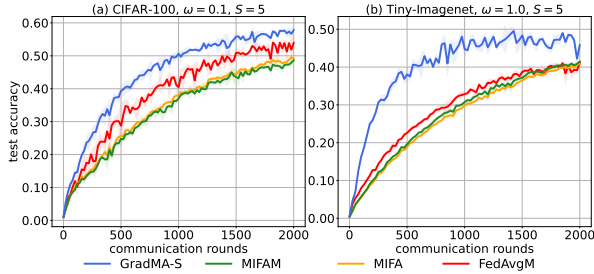


Figure 2. Test accuracy curves selected of GradMA-S as well as baselines over CIFAR-100 and Tiny-Imagenet.

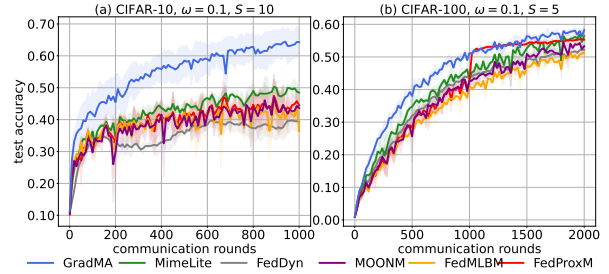


Figure 3. Test accuracy curves selected of GradMA as well as baselines over CIFAR-10 and CIFAR-100.

discuss the empirical performances of GradMA-W and GradMA-S and observe that GradMA-S beats GradMA-W by a significant margin in different FL scenarios, and even slightly outperforms GradMA in a few cases (see Table 1 and Table 2). To put it differently, GradMA leads GradMA-S in most FL scenarios, suggesting that the combination of Worker_Update() and Server_Update() can have a positive effect and thus improve performance. Meanwhile, GradMA-W trumps baselines in most cases, which suggests that Worker_Update() can mitigate the issue of CF and thus augment the centralized model. In addition, we can draw an

empirical conclusion that correcting the update direction of the centralized model on the server can greatly boost accuracy compared to correcting that of the local model for each worker. Selected learning curves shown in Fig. 1 and Fig. 4 verify the above statements.

Next, we further explore effects of (β_1, β_2) and m on the performance of GradMA-S on MNIST and CIFAR-10.

Varying control parameters (β_1, β_2) . In order to explore effects of (β_1, β_2) in more detail, we set $\beta_1, \beta_2 \in \{0.0, 0.1, 0.3, 0.5, 0.7, 0.9\}$. And we fix $m = 100$ and $S = 10$. Notice that some similarities exist between GradMA-S

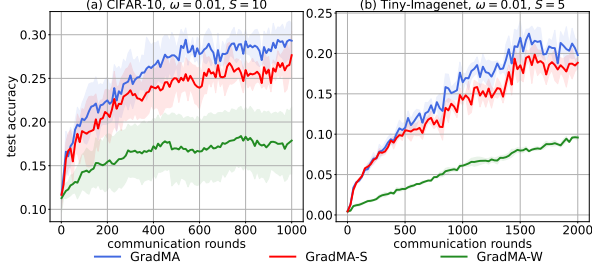


Figure 4. Test accuracy curves selected of GradMA, GradMA-S and GradMA-W over CIFAR-10 and Tiny-Imagenet.

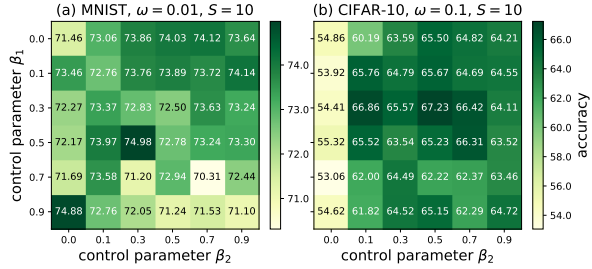


Figure 5. Top test accuracy (%) overview for GradMA-S with varying control parameters (β_1 , β_2) on MNIST and CIFAR-10.

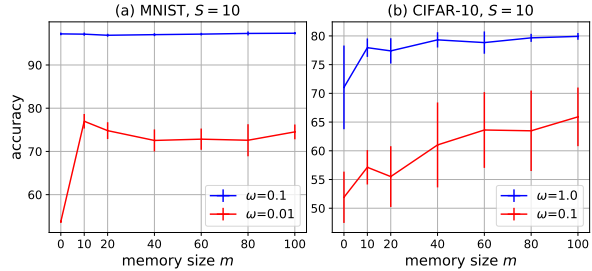


Figure 6. Top test accuracy (%) overview for GradMA-S with varying memory sizes m on MNIST and CIFAR-10.

and MIFA (MIFAM) when $\beta_1 = 0.0$ and $\beta_2 = 0.0$ ($\beta_1 = 0.0$ and $\beta_2 > 0.0$), i.e., they both memorize the latest updates of stragglers at the server side. From Table 1 and Fig. 5 (refer to Appendix .2 for more results), we can see that GradMA-S with $\beta_2 = 0.0$ considerably beats MIFA and MIFAM regardless of value of β_1 . Furthermore, we observe that GradMA-S with $\beta_2 > 0.0$ outperforms GradMA-S with $\beta_2 = 0.0$ in most cases, and the best test accuracy is located in the region of $\beta_2 > 0.0$. This indicates that the accumulated updates of stragglers can provide more effective update information for the centralized model to refine the performance of GradMA-S compared to the latest updates of stragglers.

Varying memory sizes m . In a real-world FL scenario, the memory space on the server side determines the value of the tunable parameter m for GradMA-S. Here, we fix $S = 10$ and set $m \in \{0, 10, 20, 40, 60, 80, 100\}$ to carefully look into the performance of GradMA-S with varying m .

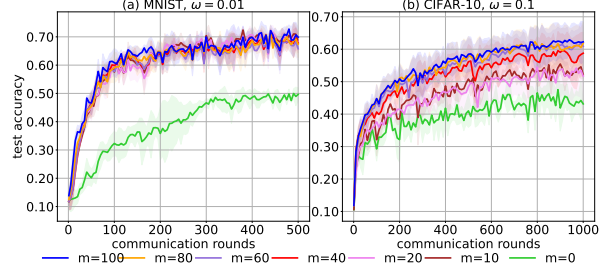


Figure 7. Test accuracy curves selected of GradMA-S with varying memory sizes m on MNIST and CIFAR-10.

Notably, GradMA-S with $m = 0$ is equivalent to FedAvgM. From Fig. 6, FedAvgM performs comparably to GradMA-S with $m > 0$ for MNIST under moderate data heterogeneity setting (i.e., $\omega = 0.1$). In contrast, the performance of FedAvgM sharply degrades and is seriously worse than that of GradMA-S with $m > 0$ under high data heterogeneity setting (i.e., $\omega = 0.01$). Meanwhile, for CIFAR-10, GradMA-S with $m > 0$ consistently surpasses FedAvgM, even under mild data heterogeneity setting (i.e., $\omega = 1.0$). Besides, we can see that the performance of GradMA-S does not intuitively and monotonically improve with increasing m . This indicates that the quality of the memory reduction strategy is an essential ingredient affecting the performance of GradMA-S for a given m . Therefore, how to tailor a more effective memory reduction strategy is one of our future works. From Fig. 7, the learning curves selected also echo the said statements (see Appendix .2 for more results).

7. Conclusions

In this paper, we propose a novel FL approach GradMA, which corrects the update directions of the server and workers simultaneously. Specifically, on the worker side, GradMA utilizes the gradients of the local model in the previous step and the centralized model, and the parameters difference between the local model in the current round and the centralized model as constraints of QP to adaptively correct the update direction of the local model. On the server side, GradMA takes the memorized accumulated gradients of all workers as constraints of QP to augment the update direction of the centralized model. Meanwhile, we provide the convergence analysis theoretically of GradMA in the smooth non-convex setting. Also, we conduct extensive experiments to verify the superiority of GradMA.

8. Acknowledgments

This work has been supported by the National Natural Science Foundation of China under Grant No.U1911203, and the National Natural Science Foundation of China under Grant No.61977025.

References

- [1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *ICLR*, 2021. [1](#), [2](#), [6](#), [13](#)
- [2] Dmitrii Avdiukhin and Shiva Kasiviswanathan. Federated learning under arbitrary communication patterns. In *International Conference on Machine Learning*, pages 425–435, 2021. [2](#)
- [3] Yasaman Esfandiari, Sin Yong Tan, Zhanhong Jiang, Aditya Balu, Ethan Herron, Chinmay Hegde, and Soumik Sarkar. Cross-gradient aggregation for decentralized learning from non-iid data. In *International Conference on Machine Learning*, pages 3036–3046. PMLR, 2021. [5](#)
- [4] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3762–3773, 2020. [2](#), [3](#), [4](#)
- [5] Xinran Gu, Kaixuan Huang, Jingzhao Zhang, and Longbo Huang. Fast federated learning in the presence of arbitrary device unavailability. *Advances in Neural Information Processing Systems*, 34:12052–12064, 2021. [1](#), [3](#), [4](#), [6](#), [13](#)
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [6](#), [13](#)
- [7] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019. [2](#), [4](#), [6](#), [13](#)
- [8] Wenke Huang, Mang Ye, and Bo Du. Learn from others and be yourself in heterogeneous federated learning. In *CVPR*, pages 10143–10153, 2022. [1](#), [3](#)
- [9] Meirui Jiang, Zirui Wang, and Qi Dou. Harmofl: Harmonizing local and global drifts in federated learning on heterogeneous medical images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1087–1095, 2022. [1](#)
- [10] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021. [1](#), [2](#), [3](#)
- [11] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Mime: Mimicking centralized stochastic algorithms in federated learning. *arXiv preprint arXiv:2008.03606*, 2020. [2](#), [6](#), [13](#)
- [12] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*, pages 5132–5143, 2020. [1](#), [2](#), [5](#), [6](#), [12](#)
- [13] Jinkyu Kim, Geeho Kim, and Bohyung Han. Multi-level branched regularization for federated learning. In *International Conference on Machine Learning*, pages 11058–11073, 2022. [1](#), [2](#), [6](#), [12](#), [14](#)
- [14] 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. [2](#), [3](#)
- [15] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009. [6](#)
- [16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. [6](#), [13](#)
- [17] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10713–10722, 2021. [1](#), [2](#), [6](#), [12](#), [14](#)
- [18] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020. [1](#), [2](#), [3](#)
- [19] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020. [1](#), [2](#), [4](#), [5](#), [6](#), [12](#)
- [20] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019. [1](#)
- [21] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *ICML*, pages 3925–3934. PMLR, 2019. [3](#)
- [22] Yijing Li, Xiaofeng Tao, Xuefei Zhang, Junjie Liu, and Jin Xu. Privacy-preserved federated learning for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2021. [1](#)
- [23] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *NIPS*, 33:2351–2363, 2020. [2](#)
- [24] David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. *NIPS*, 30, 2017. [2](#), [3](#), [4](#)
- [25] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. [1](#)
- [26] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, pages 1273–1282, 2017. [1](#), [2](#), [6](#), [12](#)
- [27] Dinh C Nguyen, Ming Ding, Pubudu N Pathirana, Aruna Seneviratne, Jun Li, and H Vincent Poor. Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3):1622–1658, 2021. [1](#)

- [28] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020. [2](#)
- [29] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. *ICLR*, 2021. [2](#), [3](#)
- [30] Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. Overcoming forgetting in federated learning on non-iid data. *NIPS*, 2019. [1](#), [3](#)
- [31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [6](#), [13](#)
- [32] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *CVPR*, pages 184–193, 2021. [3](#)
- [33] Yujia Wang, Lu Lin, and Jinghui Chen. Communication-efficient adaptive federated learning. *arXiv preprint arXiv:2205.02719*, 2022. [2](#)
- [34] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Asynchronous federated optimization. *arXiv preprint arXiv:1903.03934*, 2019. [2](#)
- [35] Ran Xin, Usman Khan, and Soumya Kar. A hybrid variance-reduced method for decentralized stochastic non-convex optimization. In *ICML*, pages 11459–11469, 2021. [5](#)
- [36] An Xu and Heng Huang. Coordinating momenta for cross-silo federated learning. *AAAI*, 36:8735–8743, 2022. [2](#)
- [37] Chencheng Xu, Zhiwei Hong, Minlie Huang, and Tao Jiang. Acceleration of federated learning with alleviated forgetting in local training. *ICLR*, 2022. [1](#), [3](#)
- [38] Jing Xu, Sen Wang, Liwei Wang, and Andrew Chi-Chih Yao. Fedcm: Federated learning with client-level momentum. *arXiv preprint arXiv:2106.10874*, 2021. [2](#)
- [39] Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-iid federated learning. In *ICLR*, 2021. [1](#), [2](#), [6](#)
- [40] Haibo Yang, Xin Zhang, Prashant Khanduri, and Jia Liu. Anarchic federated learning. In *ICML*, pages 25331–25363, 2022. [2](#)
- [41] Wensi Yang, Yuhang Zhang, Kejiang Ye, Li Li, and Cheng-Zhong Xu. Ffd: A federated learning based method for credit card fraud detection. In *International conference on big data*, pages 18–32, 2019. [1](#)
- [42] Dezhong Yao, Wanning Pan, Yutong Dai, Yao Wan, Xiaofeng Ding, Hai Jin, Zheng Xu, and Lichao Sun. Local-global knowledge distillation in heterogeneous federated learning with non-iid data. *arXiv preprint arXiv:2107.00051*, 2021. [2](#)
- [43] Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *ICML*, pages 12073–12086, 2021. [3](#)
- [44] Jaehong Yoon, Saehoon Kim, Eunho Yang, and Sung Ju Hwang. Scalable and order-robust continual learning with additive parameter decomposition. *arXiv preprint arXiv:1902.09432*, 2019. [2](#), [3](#)
- [45] Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *CVPR*, pages 10174–10183, 2022. [2](#), [3](#)
- [46] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *ICML*, pages 12878–12889, 2021. [2](#), [6](#), [13](#)