Coordinating Momenta for Cross-Silo Federated Learning

An Xu, Heng Huang∗

Electrical and Computer Engineering Department, University of Pittsburgh, PA, USA
an.xu@pitt.edu, heng.huang@pitt.edu

Abstract

Communication efficiency is crucial for federated learning (FL). Conducting local training steps in clients to reduce the communication frequency between clients and the server is a common method to address this issue. However, this strategy leads to the client drift problem due to non-i.i.d. data distributions in different clients which severely deteriorates the performance. In this work, we propose a new method to improve the training performance in cross-silo FL via maintaining double momentum buffers. In our algorithm, one momentum buffer is used to track the server model updating direction, and the other one is adopted to track the local model updating direction. More important, we introduce a novel momentum fusion technique to coordinate the server and local momentum buffers. We also derive the first theoretical convergence analysis involving both the server and local standard momentum SGD. Extensive deep FL experimental results verify that our new approach has a better training performance than the FedAvg and existing standard momentum SGD variants.

Introduction

With deep learning models becoming prevalent but data-hungry, data privacy emerges as an important issue. Federated learning (FL) (Konečný et al. 2016) was thus introduced to leverage the massive data from different clients for training models without directly sharing nor aggregating data. More recently, an increasing number of FL techniques focus on addressing the cross-silo FL (Kairouz et al. 2019; Gu et al. 2021) problem, which has more and more real-world applications, such as the collaborative learning on health data across multiple medical centers (Liu et al. 2021; Guo et al. 2021) or financial data across different corporations and stakeholders. During the training, the server only communicates the model weights and updates with the participating clients.

However, the deep learning models require many training iterations to converge. Unlike workers in data-center distributed training with large network bandwidth and relatively low communication delay, the clients participating in the collaborative FL system are often faced with much more unstable conditions and slower network links due to the geo-distribution. Typically, (Bonawitz et al. 2019) showed that one communication round in FL could take about 2 to 3 minutes in practice. To address the communication inefficiency, the de facto standard method FedAvg was proposed in (McMahan et al. 2017). In FedAvg, the server sends clients the server model. Each client conducts many local training steps with its local data and sends back the updated model to the server. The server then averages the models received from clients and finishes one round of training. The increased local training steps can reduce the communication rounds and cost. Here we also refer to the idea of FedAvg as periodic averaging, which is closely related to local SGD (Stich 2018; Yu, Jin, and Yang 2019). Another parallel line of works is to compress the communication to reduce the volume of the message (Rothchild et al. 2020; Karimireddy et al. 2019; Gao, Xu, and Huang 2021; Xu, Huo, and Huang 2021, 2020a,b). In this paper, we do not focus on communication compression.

Although periodic averaging methods such as FedAvg greatly improve the training efficiency in FL, a new problem named client drift arises. The data distributions of different clients are non-i.i.d. because we cannot gather and randomly shuffle the client data as data-center distributed training does. Therefore, the stochastic gradient computed at different clients can be highly skewed. Given that we do many local training steps in each training round, skewed gradients will cause local model updating directions to gradually diverge and overfit local data at different clients. This client drift issue can deteriorate the performance of FedAvg drastically (Hsieh et al. 2020; Hsu, Qi, and Brown 2019), especially with a low similarity of the data distribution on different clients and a large number of local training steps.

As a method to reduce variance and smooth the model updating direction (Cutkosky and Orabona 2019) to accelerate optimization, momentum SGD has shown its power in training many deep learning models in various tasks (Sutskever et al. 2013). Local momentum SGD (Yu, Jin, and Yang 2019) (i.e., FedAvgLM) maintains momentum for the stochastic gradient in each training step but requires averaging local momentum buffer at the end of each training round. Therefore, FedAvgLM requires ×2 communication cost compared with FedAvg. One strategy to achieve the same communication cost as FedAvg is resetting the
local momentum buffer to zero at the end of each training round (FedAvgLM(-Z)) (Seide and Agarwal 2016; Wang and Joshi 2018; Wang et al. 2020). Hsu, Qi, and Brown (2019) and Huo et al. (2020) proposed server momentum SGD (FedAvgSM) which maintains the momentum for the average local model update in a training round other than the stochastic gradient. The idea of FedAvgSM has been previously proposed in (Chen and Huo 2016) for speech models and in (Wang et al. 2019) for distributed training, but neither of them has applied it to FL. Hsu, Qi, and Brown (2019) and Huo et al. (2020) empirically showed the ability of FedAvgSM to tackle client drift in FL, and Wang et al. (2019) and Huo et al. (2020) provided the convergence analysis of FedAvgSM. Throughout this paper, we refer to the naive combination of FedAvgSM and FedAvgLM(-Z) as FedAvgSLM(-Z). However, FedAvgSLM(-Z) has no convergence guarantee to the best of our knowledge. Moreover, there is a lack of understanding in the connection between the server and local momenta in FL. Whether we can further improve standard momentum-based methods in FL remains another question.

To address the above challenging problems, in this paper, we propose a new double momentum SGD (DOMO) algorithm, which leverages double momentum buffers to track the server and local model updating directions separately. We introduce a novel momentum fusion technique to coordinate the server and local momentum buffers. More importantly, we provide the theoretical analysis for the convergence of our new method for non-convex problems. Our new algorithm focuses on addressing the cross-silo FL problem, considering the recently increasing needs on it as described at the beginning of this section. We also regard it as time-consuming to compute the full local gradient and focuses on stochastic methods. We summarize our major technical contributions as follows:

• Propose a new double momentum SGD (DOMO) method with a novel momentum fusion technique.
• Derive the first convergence analysis involving both server and local standard momentum SGD in non-convex settings and under mild assumptions, show incorporating server momentum’s convergence benefits over local momentum SGD for the first time, and provide new insights into their connection.
• Conduct deep FL experiments to show that DOMO can improve the test accuracy by up to 5% compared with the state-of-the-art momentum-based method when training VGG-16 on CIFAR-10, while the naive combination FedAvgSLM(-Z) may sometimes hurt the performance compared with FedAvgSM.

**Background and Related Work**

FL can be formulated as an optimization problem of $\min_{x \in \mathbb{R}^d} \frac{1}{K} \sum_{k=0}^{K-1} f^{(k)}(x)$, where $f^{(k)}$ is the local loss function on client $k$, $x$ is the model weights with $d$ as its dimension, and $K$ is the number of clients. Other basic notations are listed below. In FedAvg, the client trains the local model for $P$ steps using stochastic gradient $\nabla F^{(k)}(x_{r;p}^{(k)}, \xi_{r;p}^{(k)})$ and sends local model update $x_{r,P} - x_{r;0}$ to server. Server then takes an average and updates the server model via $x_{r+1} = x_r - \alpha \sum_{k=0}^{K-1} (x_{r,P}^{(k)} - x_{r;0})$.

**Basic notations:**

• Training round (total): $r$ ($R$); Local training steps (total): $p$ ($P$); Client (total): $k$ ($K$);
• Global training step (total): $t = rP + p$ ($T = RP$);
• Server, local learning rate: $\alpha$, $\eta$;
• Momentum fusion constant $\beta$; Server, local momentum constant: $\mu_s$, $\mu_l$;
• Server, local momentum buffer: $m_r$, $m_{r,p}$; Server, (average) local model: $x_r$, $x_{r,p}$ ($\overline{x}_{r,p}$);
• Local stochastic gradient: $\nabla F^{(k)}(x_{r,p}^{(k)}, \xi_{r,p}^{(k)})$, where $\xi_{r,p}^{(k)}$ is the sampling random variable;
• Local full gradient: $\nabla F^{(k)}(x_r^{(k)}) = \mathbb{E}_{\xi_{r,p}^{(k)}}[\nabla F^{(k)}(x_{r,p}^{(k)}, \xi_{r,p}^{(k)})]$ (unbiased sampling).

**Momentum-based.** State-of-the-art method server momentum SGD (FedAvgSM) maintains a server momentum buffer with the local model update $\frac{1}{P} \sum_{k=0}^{K-1} (x_{r,P}^{(k)} - x_{r;0}^{(k)})$ to update the server model. While local momentum SGD maintains a local momentum buffer with $\nabla F^{(k)}(x_{r,p}^{(k)}, \xi_{r,p}^{(k)})$ to update the local model. The communication costs compared with FedAvg are $\times 1, \times 2$ and $\times 1$ for FedAvgSM, FedAvgLM and FedAvgLM-Z respectively.

**Adaptive Methods.** (Reddi et al. 2020) applied the idea of using server statistics as in server momentum SGD to adaptive optimizers including Adam (Kingma and Ba 2014), AdaGrad (Duchi, Hazan, and Singer 2011), and Yogi (Zaheer et al. 2018). (Reddi et al. 2020) showed that server learning rate should be smaller than $O(1)$ in terms of complexity, but the exact value was unknown.

**Inter-client Variance Reduction.** Variance reduction in FL (Karimireddy et al. 2020b; Acar et al. 2020) refers to correct client drift caused by non-i.i.d. data distribution on different clients following the variance reduction convention. In contrast, traditional stochastic variance reduced methods that are popular in convex optimization (Johnson and Zhang 2013; Defazio, Bach, and Lacoste-Julien 2014) can be seen as intra-client variance reduction. Scaffold (Karimireddy et al. 2020b) proposed to maintain a control variate $c_k$ on each client $k$ and add $\frac{1}{K} \sum_{k=0}^{K-1} c_k - c_k$ to gradient $\nabla F^{(k)}_{r,p}(x_{r,p}^{(k)}, \xi_{r,p}^{(k)})$ when conducting local training. A prior work VRL-SGD (Liang et al. 2019) was built on a similar idea with $c_k$ equal to the average local gradients in the last training round. Both Scaffold and VRL-SGD have to maintain and communicate local statistics, which makes the clients stateful and requires $\times 2$ communication cost. Mime (Karimireddy et al. 2020a) proposed to apply server statistics locally to address this issue. However, Mime has to compute the full local gradient which can be prohibitive in cross-silo FL. It also needs $\times 2$ communication cost. Besides, Mime’s theoretical results are based on Storm (Cutkosky and Orabona 2019) but their algorithm is based on Polyak’s momentum. Though theoretically appealing, the variance reduction technique has shown to be ineffective in
Algorithm 1: FL with double momenta.

1: **Input:** local training steps $P \geq 1$, #rounds $R$, #clients $K$, server (local) learning rate $\alpha$ ($\eta$), server (local) momentum constant $\mu_s$ ($\mu_r$), momentum fusion constant $\beta$.
2: **Initialize:** Server, local momentum buffer $m_0, m_r, 0 = 0$. Local model $x_{t, 0} = x_0$.
3: for $r = 0, 1, \ldots, R - 1$ do
   4:   **Client k:**
   5:   $(r \geq 1)$ Receive $x_r^{(k)} \leftarrow x_r \cdot m_r = \frac{1}{\alpha\eta^p}(x_{r-1} - x_r)$.
   6:   $m_r^{(k)} \leftarrow 0$. // Reset local momentum.
   7:   for $p = 0, 1, \ldots, P - 1$ do
   8:     Option I: $x_r^{(k)} \leftarrow x_r^{(k)} - \eta \beta_m \text{Pm}_r \cdot 1_{p=0}$ // Pre-momentum fusion (DOMO)
   9:     $m_{r, p+1} = \mu_m m_{r, p} + \nabla F(x_r^{(k)}, \xi_r^{(k)})$
   10:    Option II: $x_r^{(k)} = x_{r-1} - \eta m_{r, p+1} - \eta \beta_m m_r$. // Intra-momentum fusion (DOMO-S)
   11:   end for
   12:   Send $d_r^{(k)} = \frac{1}{P} \sum_{p=0}^{P-1} m_{r, p+1}$ to the server.
   13: **Server:**
   14:   Receive $d_r^{(k)}$ from client $k \in [K]$. $m_{r+1} = m_r + \frac{1}{R} \sum_{k=1}^{K} d_r^{(k)}$
   15:   $x_{r+1} = x_r - \alpha \eta P \text{m}_r + 1 \text{. Send } x_{r+1} \text{ to client } k \in [K]$
   16: **end for**
   17: **Output:** $x_R$

practical neural networks’ optimization (Defazio and Bottou 2018). (Defazio and Bottou 2018) showed that common tricks such as data augmentation, batch normalization (Ioffe and Szegedy 2015), and dropout (Srivastava et al. 2014) broke the transformation locking and deviated practice from variance reduction’s theory.

**Other**. There are some other settings of FL including heterogeneous optimization (Li et al. 2018; Wang et al. 2020), fairness (Mohri, Sivek, and Suresh 2019; Li et al. 2020, 2021), personalization (T Dinh, Tran, and Nguyen 2020; Fallah, Mohktari, and Ozdaglar 2020; Jiang et al. 2019; Shamsian et al. 2021), etc. These different settings, variance reduction techniques, and server statistics can sometimes be combined. Here we focus on momentum-based FL methods.

**New Double Momentum SGD (DOMO)**

In this section, we address the connection and coordination of server and local momenta to improve momentum-based methods by introducing our new double momentum SGD (DOMO) algorithm. We maintain both the server and local statistics (momentum buffers). Nevertheless, the local momentum buffer does not make the clients stateful because the local momentum buffer will be reset to zero at the end of each training round for every client.

**Motivation.** We observe that FedAvgSM applies momentum SGD update after aggregating the local model update from clients in the training round $r$. Specifically, it updates the model at the server via $x_{r+1} = x_r - \alpha \eta P \text{m}_{r+1} = x_r - \alpha \eta P \text{m}_r - \frac{\alpha \eta P}{R} \sum_{k=1}^{K} d^{(k)}$ (Algorithm 1 lines 14 and 15). We can see that the server momentum buffer $m_r$ is only applied at the server side after the clients finish the training round $r$. Therefore, FedAvgSM fails to take advantage of the server momentum buffer $m_r$ during the local training at the client side in the training round $r$. The same issue exists for FedAvgSLM(-Z), where the local optimizer is also momentum SGD.

Recognizing this issue, we propose DOMO (summarized in Algorithm 1 where I denotes the indicator function) by utilizing server momentum statistics $m_r$ to help local training in round $r$ at the client side as it provides information on global updating direction. The whole framework can be briefly summarized in the following steps.

1. Receive the initial model from the server at the beginning of the training round.
2. Fuse the server momentum buffer in local training steps.
3. Remove the server momentum’s effect from the local model update before sending it to the server.
4. Aggregate local model updates from clients to update model and statistics at the server.

To avoid incurring additional communication cost than FedAvg, we 1) infer the server momentum buffer $m_r$ via the current and last initial model ($x_r, x_{r-1}$), and 2) reset the local momentum buffer $m_r$ to zero instead of averaging. Note that for FedAvgSLM, the local momentum buffer has to be averaged.

**Momentum Fusion in Local Training Steps.** In each local training step, the local momentum buffer is updated following the standard momentum SGD method (Algorithm 1 line 8). We propose two options to fuse server momentum into local training steps. The default Option I is DOMO and the Option II is DOMO-S with “S” standing for “scatter”. In DOMO, we apply server momentum buffer $m_r$ with coefficient $\beta P$ and learning rate $\eta$ to the local model before the local training starts. $\beta$ is the momentum fusion constant. DOMO-S is an heuristic extension of DOMO by evenly scatter- ing this procedure to all the $P$ local training steps. Therefore, the coefficient becomes $\beta$ instead of $\beta P$. Intuitively, the local model updating direction should be adjusted by the direction of the server momentum buffer to alleviate the client drift issue. Furthermore, DOMO-S follows this motivation in a more fine-grained way by adjusting each local momentum SGD training step with the server momentum buffer.

**Pre-Momentum, Intra-Momentum, Post-Momentum.** To help understand the connection between the server and local momenta better, here we propose new concepts called pre-momentum, intra-momentum, and post-momentum. The naive combination FedAvgSLM(-Z) can be interpreted as post-momentum because the current server momentum buffer $m_r$ is applied at the end of the training round and after all the local momentum SGD training steps are finished. Therefore, FedAvgSLM(-Z) has no momentum fusion to help local training. While our proposed DOMO can be regarded as pre-momentum because it applies the current server momentum buffer $m_r$ at the beginning of the training round ($p = 0$) and before the local momentum SGD
Convergence Analysis

In this section, we will discuss our convergence analysis framework with double momenta and the potential difficulty for the naive combination FedAvgSLM(-Z). There has been little theoretical analysis in existing literature for FedAvgSLM(-Z) possibly due to this theoretical difficulty. After that, we will show how the motivation of DOMO addresses it. This is the first convergence analysis involving both server and local standard momentum SGD to the best of our knowledge. Both resetting and averaging local momentum buffer are considered, though we only reset it in Algorithm 1 for less communication. Please refer to the Supplementary Material for proof details.

We consider non-convex smooth objective function satisfying Assumption 1. We also assume that the local stochastic gradient is an unbiased estimation of the local full gradient and has a bounded variance in Assumption 2. Furthermore, we bound the non-i.i.d. data distribution across clients in Assumption 3, which is widely employed in existing works as well. From Lemma 2, we can see that for DOMO, we have (z_{r,p} - x_{r,p})_D^{0} = (1 - \frac{\alpha}{\mu_f}) \mu_f \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} \sum_{p=0}^{P-1} m_{r,p}^{(k)}; while for FedAvgSLM(-Z), we have (z_{r,p} - x_{r,p})_{FedAvgSLM(-Z)} = (z_{r,p} - x_{r,p})_{DOMO} + \frac{\mu_f}{\mu_s - \mu_f} \exp P_{r,p}.

Furthermore, assume that \alpha \geq (1 - \mu_s)(1 - \mu_f); let h = \frac{1}{\mu_s - \mu_f} \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} \sum_{p=0}^{P-1} m_{r,p}^{(k)}, while for FedAvgSLM(-Z), we have

\begin{align}
\sum_{r=0}^{R-1} \sum_{p=0}^{P-1} \frac{1}{K} \sum_{k=0}^{K-1} \nabla F^{(k)}(x_{r,p}, y_{r,p}) = \nabla f(x), \forall x \in \mathbb{R}^d. Its variance is also bounded, i.e., E_x \|
abla F(x, \xi) - \nabla f(x)\|^2 \leq \sigma^2, \forall x \in \mathbb{R}^d.
\end{align}

Assumption 3. (Bounded Non-i.i.d. Distribution (Yu, Jin, and Yang 2019; Reddi et al. 2020; Wang et al. 2019; Karimireddy et al. 2020a)) For any client k \in [K] and x \in \mathbb{R}^d, there exists B \geq 0 and G \geq 0, the variance of the local full gradient in each client is upper bounded so that \frac{1}{K} \sum_{k=0}^{K} \nabla f^{(k)}(x) - \nabla f(x)\|^2 \leq G^2.

Lemma 1. (DOMO updating rule) Let 0 \leq r \leq R - 1 and 0 \leq p \leq P - 1. Let \hat{y}_{r,p} = x_0 - \frac{\alpha \eta}{(1 - \mu_s)\mu_f} \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} \sum_{p=0}^{P-1} m_{r,p}^{(k)} and z_{r,p} = \frac{1}{1 - \mu_s} \hat{y}_{r,p} - \frac{\mu_f}{1 - \mu_s} \hat{y}_{r,p-1} where \hat{y}_{0,0} = \hat{y}_{0,0} = x_0, then

\begin{align}
z_{r,p+1} = z_{r,p} - \frac{\alpha \eta}{(1 - \mu_s)(1 - \mu_f)\mu_f} \sum_{k=0}^{K-1} \nabla F^{(k)}(x_{r,p}, y_{r,p}).
\end{align}

The key of the proof is to find a novel auxiliary sequence \{z_{r,p}\} that not only has a concise update rule than the mixture of server and local momentum SGD, but also is close to the average local model \{x_{r,p}\}. One difficulty is to analyze the server model update at the end of the training round (x_{r,p} \rightarrow x_{r+1}) due to server momentum. To tackle it, we design z_{r,p} = z_{r+1,0} to facilitate the analysis at the end of the training round. Lemma 1 gives such an auxiliary sequence. Before to analyze the convergence of \{x_{r,p}\}, we summarize in Section “Background & Related Works”.

Assumption 1. (L-Lipschitz Smoothness) The global objective function f(\cdot) and local objective function f^{(k)} are L-smooth, i.e., \|
abla f^{(k)}(x) - \nabla f^{(k)}(y)\|_2 \leq L\|x - y\|_2 and \|
abla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \forall x, y \in \mathbb{R}^d, k \in [K].

Assumption 2. (Unbiased Gradient and Bounded Variance) The stochastic gradient \nabla F^{(k)}(x, \xi) is an unbiased estimation of the full gradient \nabla f^{(k)}(x), i.e., E_\xi \nabla F(x, \xi) = \nabla f(x), \forall x \in \mathbb{R}^d.
Figure 1: CIFAR-10 training curves using the VGG-16 model with various data similarity $s$.

![Graphs showing test accuracy vs. communication round for different momentum values.]

Figure 2: CIFAR-10 test accuracy (%) with various server momentum constant $\mu_s$, local momentum constant $\mu_l$, $\mu_s = 0$ corresponds to FedAvgLM, $\mu_l = 0$ corresponds to FedAvgLM, $\mu_s = 0 & \mu_l = 0$ corresponds to FedAvg, and $\mu_s \neq 0 & \mu_l \neq 0$ corresponds to DOMO.

$P^2 \sum_{r=0}^{R-1} \sum_{p=0}^{P-1} \|m_r\|^2$. Intuitively, if we ignore the constant and simply assume that $\|m_r\|^2$ is of the same complexity as $\|\nabla F\|^2$, then it causes an additional term of complexity $O(RP^3\|\nabla F\|^2)$ in the inconsistency bound, much larger than the complexity $O(RP^2\|\nabla F\|^2)$ for DOMO in the R.H.S. of Eq. (1). This also means that FedAvgSLM(-Z) is more sensitive to $P$ and may even hurt the performance when $P$ is large as in FL.

Tighten Inconsistency Bound with DOMO. In Lemma 2, we also show that DOMO can tighten the inconsistency bound. Specifically, set $\alpha = (1 - \mu_s)(1 - \mu_l)$ and DOMO can scale the inconsistency bound down to about $(1 - \mu_l)^2$ of that in FedAvgLM(-Z) ($\alpha = 1, \mu_s = 0$). Take the popular momentum constant $\mu_l = 0.9$ as an example, the inconsistency bound of DOMO is reduced to $(1 - \mu_l)^2 = 1\%$ compared with FedAvgLM(-Z). Therefore, momentum fusion not only addresses the difficulty for FedAvgSLM(-Z), but also helps the local momentum SGD training. To the best of our knowledge, this is the first time to show that incorporating server momentum leads to convergence benefits over local momentum. It is also intuitively reasonable that server momentum buffer carries historical local momentum information. But we note that this improvement analysis has not reached the optimal yet due to inequality scaling. Therefore, we fine-tune $\alpha$ and $\beta$ for the best performance in practice.

When $\alpha = 1 - \mu_s$ and $\beta = \mu_s$, we can see that DOMO has the same inconsistency bound as FedAvgLM(-Z) ($\alpha = 1, \mu_s = 0$). Consider the momentum buffer as a smoothed updating direction. Suppose the update of server momentum buffer $m_{r+1} = m_r + \Delta_r$ becomes steady with $\Delta_r \to \Delta$ (the sum of local momentum buffer in local training), then $m_r$ will be approximately equal to $\frac{\alpha}{1 - \mu_l}$. The coefficient $\frac{\alpha}{1 - \mu_l}$ leads to a different magnitude of the server and local momenta. Setting $\beta = \mu_s$ in Lemma 2 gives $\alpha = 1 - \mu_s$, balancing the difference by a smaller server learning rate.

With the above lemmas, we have the following convergence analysis theorem for our new algorithm.

**Theorem 1.** (Convergence of DOMO) Assume Assumptions 1, 2 and 3 exist. Let $\alpha \leq \frac{1 - \mu_s}{2L} < \frac{\mu_s}{4L} < \frac{\mu_s}{2L}$, then for DOMO with either resetting or averaging local momentum buffer in Algorithm 1 line 5, we have

$$\frac{1}{R \tilde{P}} \sum_{r=0}^{R-1} \sum_{p=0}^{P-1} E\|\nabla f(x_{r,p})\|^2 \leq \frac{2(1 - \mu_s)(f(x_0) - f_s)}{\eta R \tilde{P}} + 9\eta L^2 \tilde{P}^2 G^2 \frac{(1 - \mu_l)}{(1 - \mu_l)^2} + \frac{\eta L \alpha^2 (1 + \frac{3\eta LP}{(1 - \mu_l)^2}) + 2\mu_s^2 \eta L (1 - \mu_l)^2 R K}{(1 - \mu_l)^2}$$

**Complexity.** According to Theorem 1, let $\eta = O\left(K^{1/2} R^{-1/2} P^{-1/2}\right)$ and $P = O(K^{-1} R^{-1})$, then we have a convergence rate $\frac{1}{R \tilde{P}} \sum_{r=0}^{R-1} \sum_{p=0}^{P-1} E\|\nabla f(x_{r,p})\|^2 = O\left(K^{-1} R^{-1/2} P^{-1/2}\right).$
O(K^{-\frac{1}{2}}R^{-\frac{1}{2}}P^{-\frac{1}{2}}) regarding iteration complexity, which achieves a linear speedup regarding the number of clients K. DOMO also has a communication complexity of $\frac{1}{T}$ when resetting local momentum buffer, which increases with a larger number of clients K but decreases with a larger communication rounds R. It becomes $\frac{1}{T}$ for averaging local momentum buffer, but 2 is a constant and does not affect the theoretical complexity. Note that there is no $\mu_s$ in Theorem 1 because it is eliminated in Lemma 2 by setting $\beta = \mu_s$. 

### Experimental Results

#### Settings

All experiments are implemented using PyTorch (Paszke et al. 2019) and run on a cluster where each node is equipped with 4 Tesla P40 GPUs and 64 Intel(R) Xeon(R) CPU E5-2683 v4 cores @ 2.10GHz. We compare the following momentum-based FL methods: 1) FedAvg, 2) FedAvgSM (i.e., server momentum SGD), 3) FedAvgLM (i.e., local momentum SGD), 4) FedAvgLM-Z (i.e., local momentum SGD with resetting local momentum buffer), 5) FedAvgSLM (i.e., FedAvgSM + FedAvgLM), 6) FedAvgSLM-Z (i.e., FedAvgSM + FedAvgLM-Z, Algorithm 1 Option III), 7) DOMO (i.e., Algorithm 1 Option I), and 8) DOMO-S (i.e., Algorithm 1 Option II). In particular, FedAvg, FedAvgSM, FedAvgLM-Z, FedAvgSLM-Z, DOMO and DOMO-S have the same communication cost. FedAvgLM and FedAvgSLM need $\times2$ communication cost.

We perform careful hyper-parameters tuning for all methods. The local momentum constant $\mu_l$ is selected from {0.9, 0.8, 0.6, 0.4, 0.2}. We select the server momentum constant $\mu_s$ from {0.9, 0.6, 0.3}. The base learning rate is selected from {..., $4 \times 10^{-1}$, $2 \times 10^{-1}$, $1 \times 10^{-1}$, $5 \times 10^{-2}$, $1 \times 10^{-2}$, $5 \times 10^{-3}$, ...}. The server learning rate is an alpha from {0.2, 0.4, 0.6, 0.8, 0.9, 1.0}. The momentum fusion constant $\beta$ is selected from {0.2, 0.4, 0.6, 0.8, 0.9, 1.0}. Following (Karimireddy et al. 2020b; Hsu, Qi, and Brown 2019; Wang et al. 2020), we use local epoch $E$ instead of local training steps $P$ in experiments. $E = 1$ is identical to one pass training of local data. We test local epoch $E \in \{0.5, 1, 2\}$ and $E = 1$ by default.

#### Data Similarity $s$

We follow (Karimireddy et al. 2020b) to simulate the non-i.i.d. data distribution. Specifically, fraction $s$ of the data is randomly selected and allocated to clients, while the remaining fraction $1 - s$ are allocated according to the label. The data similarity is hence $s$. We run experiments with data similarity $s$ in {5%, 10%, 20%}. By default, the data similarity is set to 10% and the number of clients (GPUs) $K = 16$ following (Wang et al. 2020). For all experiments, We report the mean and standard deviation metrics in the form of (mean ± std) over 3 runs with different random seeds for allocating data to clients.

#### Dataset

We train VGG-16 (Simonyan and Zisserman 2014) and ResNet-56 (He et al. 2016) models on CIFAR-10/100¹ (Krizhevsky 2009), and ResNet-20 on SVHN² image classification tasks. Please refer to the Supplementary Material for details.

#### Performance

We illustrate the experimental results in Figures 1, 2, and 3 with test accuracy (mean ± std) reported in the brackets.

---

¹https://www.cs.toronto.edu/~kriz/cifar.html
²http://ufldl.stanford.edu/housenumbers/
We can see that DOMO improves the test accuracy over the best with data similarity of VGG-16 under different local epoch poly lower test accuracy. When best counterpart by 2.13% respectively regarding the test accuracy when s complicated. In particular, DOMO improves FedAvgSLM(-Z) is much better than without it (FedAvgSM ≫ FedAvgLM(-Z)), in accordance with (Hsu, Qi, and Brown 2019; Huo et al. 2020).

Varying Data Similarity $s$. We plot the training curves under different data similarity settings in Figure 1. We can see that the improvement of DOMO and DOMO-S over other momentum-based methods increases with the data similarity $s$ decreasing. This property makes our proposed method favorable in FL where the data heterogeneity can be complicated. In particular, DOMO improves FedAvgSLM(-Z), FedAvgSM, and FedAvg by 5.00%, 5.68%, and 23.14% respectively regarding the test accuracy when $s = 5\%$. When $s = 10\%$ and $s = 20\%$, DOMO improves over the best counterpart by 2.13% and 1.02%, while DOMO-S improves by 1.41% and 0.85% respectively.

Varying the Server and Local Momentum Constant $\mu_s$, $\mu_l$. We explore the various combinations of server and local momentum constant $\mu_s$ and $\mu_l$ of DOMO and report the test accuracy in Figure 2. $\mu_s = 0.9$ and $\mu_l = 0.6$ work best regardless of the data similarity $s$ and the algorithm we use. Deviating from $\mu_s = 0.9$ and $\mu_l = 0.6$ leads to gradually lower test accuracy.

Varying the Local Epoch $E$. We plot the training curves of VGG-16 under different local epoch $E$ settings in Figure 3 with data similarity $s = 10\%$. The number of local training steps $P = 49$ and 196 respectively when $E = 0.5$ and 2. We can see that DOMO improves the test accuracy over the best counterpart by 1.25% and 1.70% when $E = 0.5$ and 2 respectively.

Varying Hyper-parameters $\alpha$ and $\beta$. We explore the combinations of hyper-parameters $\alpha$ and $\beta$ and report the corresponding test accuracy in Table ??, $\alpha = 1.0$ and $\beta = 0.9$ work best.

Varying Model. We also plot the training curves of ResNet-56 in the right plot of Figure 3 which exhibit a similar pattern. DOMO improves the best counterpart by 1.35% when data similarity $s = 10\%$ and local epoch $E = 1$. FedAvgSLM(-Z) is inferior to FedAvgSM, implying that a naive combination of FedAvgSM and FedAvgLM can hurt the performance. In contrast, DOMO and DOMO-S improve FedAvgSLM by 2.45% and 2.03% respectively.

Varying Dataset. The SVHN test accuracy is summarized in Table ?? and we can see that DOMO and DOMO-S improve the counterpart by 1.45% and 1.56% respectively. The CIFAR-100 test accuracy is summarized in Table ?? and we can see that DOMO and DOMO-S improve the best counterpart by 2.46% and 1.68% respectively when training VGG-16. They improve the counterpart by 2.39% and 0.89% respectively when training ResNet-56.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>87.79 ± 0.72</td>
<td>88.81 ± 0.49</td>
<td>88.86 ± 0.19 (87.93 ± 0.98)</td>
<td>88.67 ± 0.32 (88.89 ± 0.62)</td>
<td>90.45 ± 0.56</td>
<td>90.34 ± 0.83</td>
</tr>
</tbody>
</table>

Table 2: SVHN test accuracy (%) when training ResNet-20.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20.77 ± 1.31</td>
<td>35.14 ± 1.70</td>
<td>38.29 ± 1.01 (35.45 ± 2.04)</td>
<td>60.01 ± 0.28 (57.89 ± 2.79)</td>
<td>61.69 ± 0.41</td>
<td>62.47 ± 0.73</td>
</tr>
<tr>
<td>39.61 ± 0.66</td>
<td>61.92 ± 0.43</td>
<td>46.65 ± 1.38 (45.09 ± 0.26)</td>
<td>62.95 ± 0.51 (63.45 ± 0.61)</td>
<td>64.34 ± 0.59</td>
<td>65.84 ± 0.30</td>
</tr>
</tbody>
</table>

References


Liu, Q.; Chen, C.; Qin, J.; Dou, Q.; and Heng, P.-A. 2021. Feddg: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency


8743