IEEE Open Journal of the Computer Society

Received 5 January 2024; accepted 14 January 2024. Date of publication 22 January 2024;
date of current version 9 February 2024. The review of this paper was arranged by Associate Editor Quan Chen.

Digital Object Identifier 10.1109/OJCS.2024.3356599

Slingshot: Globally Favorable Local Updates for Federated Learning

JIALIANG LIU [1], HUAWEI HUANG [2] (Senior Member, IEEE), CHUN WANG [1], SICONG ZHOU [1], RUIXIN LI [1], AND ZIBIN ZHENG [2] (Fellow, IEEE)

[1]School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou 516001, China
[2]School of Software Engineering, Sun Yat-Sen University, Zhuhai 519082, China

CORRESPONDING AUTHOR: HUAWEI HUANG (e-mail: huanghw28@mail.sysu.edu.cn).

This work was supported by the Key-Area Research and Development Program of Guangdong Province under Grant 2021B0101400005.

ABSTRACT Federated Learning (FL), as a promising distributed learning paradigm, is proposed to solve the contradiction between the data hunger of modern machine learning and the increasingly stringent need for data privacy. However, clients naturally present different distributions of their local data and inconsistent local optima, which leads to poor model performance of FL. Many previous methods focus on mitigating objective inconsistency. Although local objective consistency can be guaranteed when the number of communication rounds is infinite, we should notice that the accumulation of global drift and the limitation on the potential of local updates are non-negligible in those previous methods. In this article, we study a new framework for data-heterogeneity FL, in which the local updates in clients towards the global optimum can accelerate FL. We propose a new approach called *Slingshot*. Slingshot's design goals are twofold, i.e., i) to retain the potential of local updates, and ii) to combine local and global trends. Experimental results show that *Slingshot* helps local updates become more globally favorable and outperforms other popular methods under various FL settings. For example, on CIFAR10, *Slingshot* achieves 46.52% improvement in test accuracy and 48.21× speedup for a lightweight neural network named *SqueezeNet*.

INDEX TERMS Federated learning, data heterogeneity, catastrophic forgetting, model performance.

I. INTRODUCTION

In each communication round of a standard FL called *FedAvg* [1], each selected client first receives the global model from a central server and executes stochastic gradient descent (SGD) with local data in several local epochs. The updated local model is then returned to the server for aggregation. Compared to traditional distributed learning, FL protects data privacy by exchanging models instead of the local data of each participant. Therefore, FL can be applied to areas with strict privacy restrictions such as healthcare [2], [3]. On the other hand, FL reduces the aggregation frequency, thereby lowering communication costs [1].

The reality is that data heterogeneity (also known as statistical heterogeneity or non-identically distributed) prevents FL from being largely applied in practice. In the real-world environment, each client often has its own data distribution because of personal preferences and attributes. Combining as much data as possible to train an optimal global model that fits the total data distribution is our expectation for FL. However, in data-heterogeneity settings, FL has been found to converge slowly to a sub-optimal point or not at all [4], [5], particularly if the learning rate has not been specifically optimized.

The poor model performance and slow convergence of data-heterogeneity FL result from inconsistent local optima across clients [6], [7], [8], [9]. Seriously heterogeneous data means that the clients' local optima are far from each other. Thus, the global optimum (the average of all local optima) would be far from each local optimum. Such inconsistencies cause two detrimental effects on FL, i.e., i) local updates deviate from global updates (Local Drift as shown in Fig. 1), and ii) the aggregated global model deviates from the global optimum (Global Drift as shown in Fig. 1).

It is a common approach to address the data heterogeneity problem in FL by alleviating objective inconsistency. These methods can be divided into the following two categories. The first method is to add the regularization term to constrain

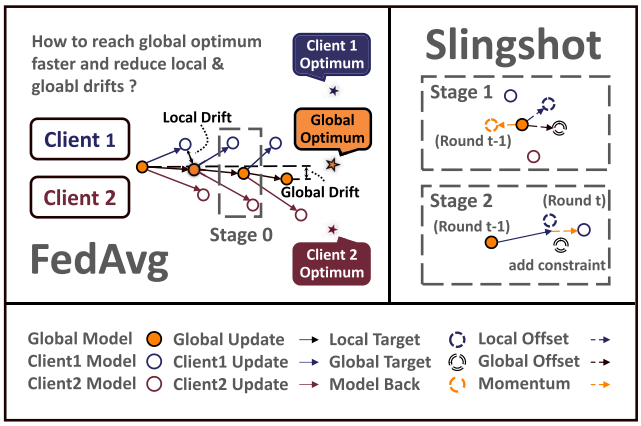


FIGURE 1. Comparison between the conventional federated learning paradigm *FedAvg* [1] and our proposed *Slingshot*. *FedAvg* produces two kinds of drifts when dealing with data heterogeneity. *Slingshot* is designed to facilitate the convergence of local update models toward the global optimum while preserving the quality of local updates.

the distance between the local updated model and the global model [8], [10], [11]. The other one is to set the correction term of Local Drift [6], [12]. However, these methods cannot eliminate objective inconsistency, because the global model in the regularization or correction term also includes drifts [13]. Although all client solutions can be aligned at the end of federated training in these methods, the drift has gradually accumulated during the progressive alignment. In other words, these methods ensure faster convergence of FL, but can not guarantee the converged model is closer to the global optimum. The accumulation of Global Drift leads to poor model performance of FL. In addition, strict restrictions on consistency limit the potential of local updates to improve local models, thus degrading the updates of the global model [14].

In this article, we reconstruct a new perspective on data-heterogeneity FL that prioritizes global benefits over objective consistency. Particularly, we no longer asymptotically reduce the Local Drift to ensure the consistency of clients, but focus on increasing local updates in a globally favorable direction. This is because local updates towards the global optimum can potentially accelerate data-heterogeneity FL. Inspired by this idea, we propose a new method called *Slingshot* by taking the two following design goals into account, i.e., i) to retain the potential of local updates, and ii) to combine local and global trends such that local updates are more globally favorable.

Our study includes the following contributions.

- We provide a new metric called *MGAI* (**M**ean **G**lobal **A**ccuracy **I**ncrease) to measure the effect of local updates of FL algorithms. We show a positive correlation between this metric and the final model performance. Thus, this metric can help researchers compare the final model performance of various FL algorithms during the early testing phase in the laboratory.
- We propose a new method called *Slingshot* that improves the quality of local updates in FL. *Slingshot* adopts two dynamic targets representing local and global trends respectively, which helps the updated local models get closer to the global optimum. Also, it retains the potential of local updates to greatly improve local models.
- Our extensive experiments show that the improved local updates of *Slingshot* make it applicable to various datasets, models, and other settings in FL, with better performance and faster convergence. For example, in a severe data-heterogeneity setting, *Slingshot* reduces the communication rounds of *FedAvg* by 72% on the CIFAR10 dataset.

Here is the guide for the subsequent sections. In Section II, we review related work in the field of data-heterogeneity federated learning and catastrophic forgetting. In Section III, we discuss the underlying cause of *FedAvg's* poor performance with heterogeneous data is ineffective local updates. We also define a new metric called *MGAI* to measure the value of local updates. Following that, Section IV introduces the proposed method *Slingshot*, detailing its motivation and designs. Experimental setups and results are presented in Section V. Finally, we conclude in Section VI with a summary of key contributions. We hope our study will stimulate more researchers to discuss what kind of local updates are more beneficial to global updates and global models.

II. RELATED WORK

A. DATA-HETEROGENEITY FEDERATED LEARNING

The data heterogeneity among all clients is a key challenge in FL. A common solution to addressing the data-heterogeneity issue is to alleviate objective inconsistency by reducing the Local Drifts aforementioned. The representative studies are reviewed as follows. *FedProx* [10] simply adds a proximal term to the objective. *Scaffold* [6] views the Local Drift as "client-variance" and explores *control variate* to correct for the Local Drift. *Moon* [8] adopts a model-level contrastive loss by comparing representations learned by global models, local models, and previous local models. *FedDyn* [11] adds linear and quadratic penalty terms that dynamically modify the clients' objective to ensure objective consistency in the limit. *FedDC* [12] decouples local training from global training and bridges the Local Drift with a local-drift variable. *FedGA* [15] promotes the alignment of gradients across clients with an implicit regularization.

In addition, some methods improve on the original weighted aggregation of *FedAvg* with a focus on the server side. These methods are designed to reduce Global Drift. For example, *FedNova* [13] starts with various amounts of local updates and normalizes all local gradients before the aggregation. Considering the permutation invariance of neural network parameters, *FedMa* [16] matches and averages similar weights layer by layer in the aggregation phase. To mitigate feature drifts of heterogeneous data, *FedBN* [17] does not aggregate the parameters of local BatchNorm layers. *FedOPT* [18] converts the various dynamic optimizers into federated versions, and applies them to the global updates on the server. However, none of the mentioned methods can

integrate with Local-Drift-reduction methods well. In fact, Global Drift is actually the vector sum of Local Drift [9]. Thus, the Local-Drift-reduction methods are not orthogonal to the Global-Drift-reduction methods in FL. We focus on the Local-Drift-reduction methods, which are directly related to the raw data and are more popular recently [12], [15].

From these previous studies mentioned above, we observe that paying too much attention to objective consistency could be harmful to FL. Because of the strong constraints or corrections in these methods, the length of gradient descent of the local model could be too short. Aggregating these less updated models, the server collects less novel information per communication round, thus increasing the communication rounds [7]. In addition, the global model in the regularization or correction term also includes drifts and the cumulative Global Drift in these methods can not be ignored. Thus, we try to find a more general and intuitive way to solve the issue of data heterogeneity.

B. CATASTROPHIC FORGETTING IN FEDERATED LEARNING

Catastrophic forgetting refers to an important problem when neural networks train the current task but forget the knowledge of the previous tasks [19], if given a series of tasks with different data distributions. Some recent papers attribute the poor performance of data-heterogeneity FL to the catastrophic forgetting across clients [20], [21]. Because of inconsistent objectives, the global model forgets the previous knowledge when each client executes local SGD, and local models forget the locally learned knowledge when the updated models are aggregated by servers. The biggest challenge in tackling data-heterogeneity FL and catastrophic forgetting is how to balance the knowledge with different data distributions. Consequently, we can learn from the methods of catastrophic forgetting to address data heterogeneity FL.

The authors of [22], [23] classified the most related papers on catastrophic forgetting into three main categories, i.e., regularization [24], [25], [26], replay [27], [28], [29] and parameter isolation approaches [30], [31], [32], [33]. It is a common agreement for regularization-based methods to be applied for data-heterogeneity FL, such as *FedProx* and *FedDyn* aforementioned. The replay-based methods have been also proposed for FL, e.g., [7] corrects the classifier of neural network by replaying virtual representations generated from the Gaussian distribution. Inspired by the parameter-isolation approaches [32], [33], we find that a globally favorable local update in FL is actually fixing those model parameters that improve both local and global accuracy while updating other model parameters. Thus, we adopt two dynamic targets in the local training phase to help local updates improve both local and global accuracy. Moreover, these targets are not proximal and do not unduly constrain the potential of local training to update other parameters.

III. INEFFECTIVE LOCAL UPDATES

In this section, we define the optimization goals of federated learning and analyze the deep reasons why conventional federated learning performs poor model performance in the case of heterogeneous data. At last, we propose a new metric to validate our analysis. Table 1 provides explanations for important symbols and notations.

TABLE 1. Symbols and Notations

k, r	the index of client, communication round
R	the # of communication round
N, $[N]$	the # of clients, the client set
$\mathcal{S}$	the selected subset of $[N]$
η	the learning rate of local updates
ω^*	the optimal global model
ω_k^*	the local optimum of client k
ω	the global model
ω_k	the local model of client k
D	the whole dataset in a theoretical sense
D_k	the local dataset of client k
s	the data sample of local dataset
f	the supervised loss function
l	the loss computed through the loss function and input
$\overline{\omega}$	the averaged model after the aggregation
ω^+	the global model after one step of ideal global update
Δ_k	the computed test accuracy difference for *MGAI*
μ	controlling the weight of regularization term
α	controlling *Slingshot's* targets building
$\omega_{loc}, \omega_{glo}$	local target, global target of *Slingshot*
$\omega_k^{pre}, \omega_k^{rec}$	previous updated model, last received global model
g	the gradient aggregated through weighted averaging
m	the momentum of global update

A. PRELIMINARIES

Assuming that there are a number of $N \in \mathbb{N}^+$ clients participating FL. We denote ω as the global model. Thus, the global optimum in FL is defined as follows.

$$\omega^* = \arg\min \left[f(\omega) \triangleq \frac{1}{|D|} \sum_{k=1}^{N} \sum_{s \in D_k} f(\omega; s) \right], \quad (1)$$

where D_k is the local dataset on client $k(\in [N])$, the whole dataset $D = \cup_{k \in [N]} D_k$ is the union of D_k, $|D|$ indicates the number of data samples in all clients, s is a data sample in D_k, and $f(\omega; s)$ means the supervised loss function given ω and s, such as the common *cross-entropy loss*.

In *FedAvg* [1], each client first lets local model $\omega_k = \omega$, then executes mini-batch gradient descent at its device. In fact, with its local data, each client k can only help the local model get closer to the local optimum by executing local SGD following *FedAvg* [1]:

$$\omega_k = \omega_k - \eta \nabla f_k(\omega_k), \ k \in [N], \quad (2)$$

$$f_k(\omega_k) \triangleq \frac{1}{|D_k|} \sum_{s \in D_k} f(\omega_k; s), \ k \in [N], \quad (3)$$

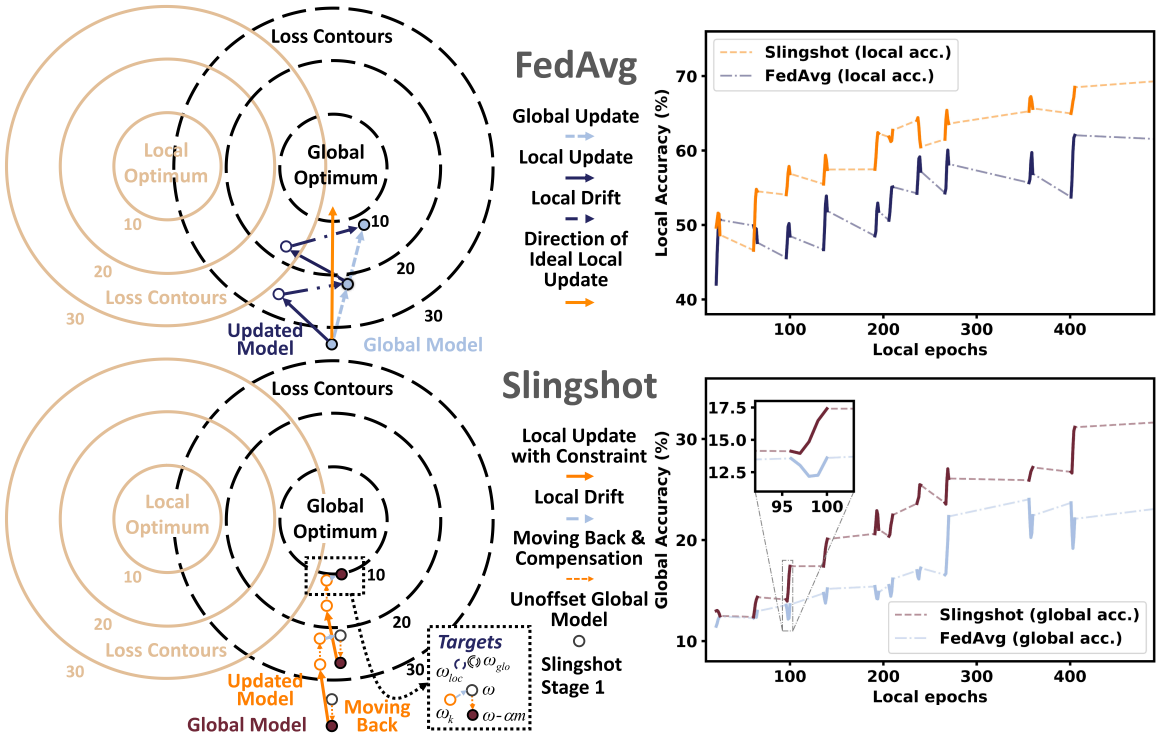


FIGURE 2. **Underlying cause of *FedAvg's* poor performance with heterogeneous data is the fact that local updates of a client have little or even negative impacts on global accuracy. In the right subgraph, the solid lines indicate the periods during which a client performs local SGD, and the dashed lines represent the intervals when this client is in idle status. While *FedAvg's* local updates can significantly improve local accuracy, their impacts on the global accuracy of local models remain marginal. In contrast, *Slingshot* strikes a balance between local and global accuracy and improves the quality of local updates. Local update consists of multiple iterations, which is omitted as one iteration for simplicity in this figure.**

where η is the local learning rate. After local training, client k returns the updated model ω_k to the central FL server for the aggregation of model parameters

$$\overline{\omega} = \sum_{k=1}^{N} \frac{|D_k|}{|D|} \omega_k, \tag{4}$$

where $\overline{\omega}$ is the averaged model. And the server uses $\overline{\omega}$ as the new global model ω.

B. HETEROGENEOUS DATA MAKES UPDATES INEFFECTIVE

In a centralized learning, after one step of global update, the global model will be changed to

$$\omega^+ = \omega - \eta \nabla f(\omega). \tag{5}$$

It is obvious that $\overline{\omega}$ is worse than ω^+ because the direction of $-\nabla f_k(\omega)$ is towards the local optimum

$$\omega_k^* = \arg\min \sum_{s \in D_k} f(\omega; s)/|D_k|, \ k \in [N]. \tag{6}$$

and there is a large angle between the directions of $-\nabla f(\omega)$ and $-\nabla f_k(\omega)$ especially when D_k is significantly different from D. The difference between $-\nabla f(\omega)$ and $-\nabla f_k(\omega)$ is the mentioned Local Drift. The difference between ω^+ and $\overline{\omega}$ is Global Drift. As shown in Fig. 2, the direction of the local update deviates from the direction of the ideal local update in *FedAvg*. In consequence, the speed of the global model approaching the global optimum is slowed down because local updates have limited contributions to achieving the global optimum.

We demonstrate this inference by testing a client's local accuracy (using a local test set with the same distribution as the local train set) and global accuracy (using the global test set) on CIFAR10. As shown in Fig. 2, when this client executes local SGD under *FedAvg*, the local accuracy increases significantly, while the global accuracy increases a little or even decreases. In fact, the overall local accuracy also increases slowly. This is because, in each communication round, all local models start from the global model sent by the central server. The slowly-growing global accuracy implies that the starting point of local accuracy in each round grows slowly. Therefore, the bad model-training performance and slow convergence of data-heterogeneity FL are fundamentally induced by the ineffective local updates. When the local updated model is not much closer to the global optimum than the global model before training, such a local update is ineffective.

C. MEAN GLOBAL ACCURACY INCREASE

In order to better explore what local updates are globally beneficial, we integrate the findings from the previous section and first define a metric called *MGAI:* Mean Global Accuracy Increase. A study shows that the final test accuracy of FL is greatly affected by the early phase of the training process [34].

This is because if the global model is optimized to a sub-optimal point far away from the global optimum in the early training phase, the complex loss surface in data-heterogeneity FL prevents the global model from escaping the sub-optimal point. Thus, we focus on the global impact of local updates in the early training phase. Note that measuring the accuracy of a local model simply with a local test set or measuring the accuracy of a global model simply with a global test set can not directly account for the global impact of local updates. The effective measurement scheme is to test the effectiveness of local updates with a global test set.

Specifically, in each early critical communication round, we measure the test accuracy of local models on the global test set twice. The first measurement occurs before the first local epoch, and the second measurement is conducted after each selected client sends the updated model to the FL server. The difference between the two measurements $\Delta_k \to \infty$ indicates an effective local update of client k, while $\Delta_k \to -\infty$ indicates an ineffective one. Then we compute the mean $\frac{1}{|\mathcal{S}|}\sum_{k\in\mathcal{S}}\Delta_k$ of five communication rounds, and the *MGAI* is obtained.

MGAI measures the increase of local updates on global accuracy. The experimental results show that *MGAI* is positively correlated with the final model performance. This validates the idea that globally beneficial local updates can improve the model performance of FL. Note that measuring *MGAI* can be done directly on the server side. As the training task is initiated on the server, we can collect the test set on the server and measure *MGAI* in the early stages of training. In each communication round, the first measurement of test accuracy occurs before the local models are updated. At this time, each local model is equal to the global model. The second measurement of test accuracy occurs before aggregation. At this time, the updated models are sent to the server. Although such testing consumes some additional resources, it allows users to select effective algorithms at an early phase. For example, for a task to train a thousand rounds, it may only take 50 rounds of *MGAI* measurement to select an effective algorithm, which can save a lot of training resources.

IV. THE METHOD OF *SLINGSHOT*

Inspired by the observation from Fig. 2 and the analysis described in previous section, we propose *Slingshot* to accelerate data-heterogeneity FL. *Slingshot* exploits the directions and the quality of local updates, which is based on the thinking of how to improve *MGAI*. The design goal of *Slingshot* is to enforce each local update becoming more favorable to the update of a global model such that their local updates can help the updated local model approach the global optimum. With the goal in mind, the challenge is that each client does not know where the global optimum is.

To assist each client to capture the globally favorable direction, *Slingshot* has the following two changes compared to *FedAvg*. Firstly, each local update is guided by two dynamic targets. Secondly, the global model is moved back before local updates and is compensated after local updates. These two changes are elaborated in Designs 1 and 2, respectively. Before describing the two designs, we depict the motivation of Design 1 as follows.

Algorithm 1: *Slingshot.*

Input: learning rate η, hyper-parameters (α, μ), trainsets D_k, global momentum $m = \mathbf{0}$, global model ω, previous updated models ω_k^{pre}, last received global models ω_k^{rec}, $k \in [N]$. */* All models are of the same architecture and are initialized in the same way. */*

Output: final global model ω.

for each round $r = 1,..., R$ **do**
  $\omega \leftarrow \omega - \alpha m$ */* Move back global model. */*
  Sample clients $\mathcal{S}$ from $\{1,..., N\}$.
  for each client $k \in \mathcal{S}$ **in parallel do**
    Local model $\omega_k \leftarrow \omega$
    $\omega_{loc} \leftarrow \omega + \alpha(\omega_k^{pre} - \omega_k^{rec})$ */* Build loc. targets. */*
    $\omega_{glo} \leftarrow \omega + \alpha(\omega - \omega_k^{rec})$ */* Build glo. targets. */*
    $\omega_k \leftarrow$ **LocalUpdate**$(\omega_k, \omega_{loc}, \omega_{glo})$
    $\omega_k^{rec}, \omega_k^{pre} \leftarrow \omega, \omega_k$ */* Update saved models. */*
  end for
  Aggregated gradient $g \leftarrow FedAvg(\omega_k - \omega), k \in \mathcal{S}$
  $\omega \leftarrow \omega + g$ */* Global update. */*
  $\omega \leftarrow \omega + \alpha m$ */* Compensate global model. */*
  $m \leftarrow \eta m + g$ */* Momentum update. */*
end for

LocalUpdate $(\omega_k, \omega_{loc}, \omega_{glo})$:

for each local epoch **do**
  for batch $b_i \in D_k$ **do**
    $l_{ce} \leftarrow CrossEntropyLoss(b_i; \omega_k)$
    $l_{ss} \leftarrow \frac{\mu}{2}(\|\omega_k - \omega_{loc}\|^2 + \|\omega_k - \omega_{glo}\|^2)$
    $l \leftarrow l_{ce} + l_{ss}$
    $\omega_k \leftarrow \omega_k - \eta\nabla l$
  end for
end for
return ω_k

A. MOTIVATION: BALANCING THE LOCAL AND GLOBAL PERFORMANCE

Ineffective local updates in data-heterogeneity FL originates from the fact that all clients share a common set of global model parameters but the local and global optima are different. Motivated by the parameter-isolation approaches [32], [33] ensuring minimal drop in performance, we argue that the key to data-heterogeneity FL is to balance the local and global performance. However, the local updates in *FedAvg* are unbalanced, which aims at local optima but ignores the global optimum. On the other hand, FedProx-like methods add a strong penalty term written as $\frac{\mu}{2}\|\omega_k - \omega\|^2$, where μ controls the weight of this penalty term. This penalty term prevents the updated local ω_k model from getting too far from the global model ω. In fact, such stringent penalty term limits the effective forwarding of local updates towards local optima.

Then global updates are also constrained accordingly. Therefore, a globally-favorable local update should simultaneously consider the update trends of both the local and global models, while preserving its potential of moving towards local and global optima.

B. DESIGN 1: SETTING TWO DYNAMIC TARGETS

We construct two dynamic targets for each local update (as shown in lines 6-7 of Algorithm 1), i.e., local target ω_{loc} and global target ω_{glo}. They imply the local and global trends, respectively. For client $k \in [N]$, ω_k^{rec} is the global model last received from the central server, and ω_k^{pre} is the previous updated local model sent to the server. Both targets are constructed by global model ω adding accumulative gradients. The local target ω_{loc} is constructed as

$$\omega_{loc} = \omega + \alpha(\omega_k^{pre} - \omega_k^{rec}), \tag{7}$$

and the global target is constructed as

$$\omega_{glo} = \omega + \alpha(\omega - \omega_k^{rec}). \tag{8}$$

where α is a hyper-parameter controlling how far the target is from the global model. Then the local update of client k is limited by two distances when fitting local data (as shown as Local Update in Algorithm 1). One is the distance between the local model ω_k and the local target, represented by $\frac{\mu}{2}\|\omega_k - \omega_{loc}\|^2$. And the other one is the distance between the local model and the global target, represented by $\frac{\mu}{2}\|\omega_k - \omega_{glo}\|^2$. Such two targets direct each client to find an updated model that gets closer to its local optimum with a minimal drop in global performance.

C. DESIGN 2: MOVING GLOBAL MODEL BACK

Although the penalty terms $\frac{\mu}{2}(\|\omega_k - \omega_{loc}\|^2 + \|\omega_k - \omega_{glo}\|^2)$ help local updates more globally favorable, they also prevent the local updates from converging to the local optimum fast. Note that there is a trade-off in the size of the hyper-parameters α. When α is too large, the dynamic targets are too far from the local model, reducing the impact of targets on local updates. On the other hand, these dynamic targets are close to the global model. Our proposed *Slingshot* is then equivalent to *FedProx* [10], which limits the potential of local updates to improve local models. To prevent the value of hyper-parameters α from unduly affecting the performance of *Slingshot*, we further improve local updates.

In the state-of-the-art data-heterogeneity FL methods, local updates are mainly affected by local datasets and regularization terms (or correction terms). These methods often design regularization terms (or correction terms) elaborately, but ignore the raw effects of local datasets. When is the direction toward a local optimum approximately equal to the direction toward the global optimum? The answer is when the global model is far away from both local and global optimum.

Let the local dataset guide the local model to the global optimum as much as possible. We propose a method that increases the distance between the global model and the global optimum before local training but decreases the distance after training. Specifically, we let each client move the global model back αm before local updates

$$\omega \leftarrow \omega - \alpha m, \tag{9}$$

and compensate it after local updates (as shown in line 13 of Algorithm 1), where m is the global momentum, i.e. the momentum of global updates. This momentum is updated as shown in line 14 of Algorithm 1.

It is worth noting that Design 2 is fundamentally different from the conventional FL methods using momentum [18], [35]. Although we use server-side momentum, this momentum is not really incorporated into the optimizer. This is because the momentum that we add to the global model cancels out before and after the local training, while the momentum added to the model in the conventional FL methods is maintained. Considering that it is difficult to directly correct the global update, (especially in the scenario of Big Data and deep models, even a minor correction to the global model will have a huge impact) we do not directly correct the global update. The reason we introduce server-side momentum is to implicitly improve local updates, i.e., to implicitly add the gradient pointing to the global optimum on the local update. Our experiments in Section V-B prove that both of our proposed designs can improve the performance of FL. Also, *Slingshot's* performance is less sensitive to hyper-parameter α because of Design 2.

D. PROPERTY ANALYSIS

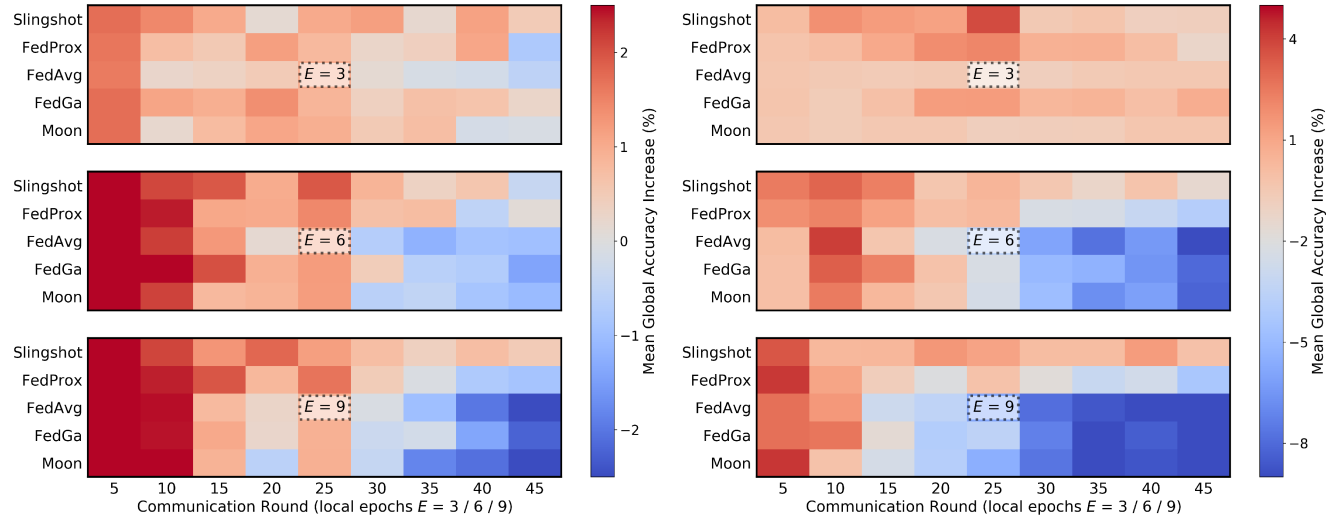
Compared with the classical FL algorithm *FedAvg*, most data-heterogeneity FL methods have additional resource consumption [6], [8], [10]. To the best of our knowledge, an efficient data-heterogeneity FL method that does not require preserving additional models or gradient states has not yet emerged. *Slingshot* also needs to maintain some historical models locally on the clients to improve the training, such as ω_k^{pre} and ω_k^{rec}. These models consume additional memory resources. On the other hand, the effectiveness of some methods depends heavily on synchronizing the extra saved models. For example, the conventional data-heterogeneity FL algorithm *Scaffold* [6] incurs 2×communication overhead for synchronizing the global variate c. However, *Slingshot* does not require this kind of synchronization, eliminating additional communication resource consumption. This means that *Slingshot's* communication overhead per round is equal to that of *FedAvg*.

V. EXPERIMENTS

A. TRAINING CONFIGURATION

1) DATASETS AND MODELS

We test our proposed *Slingshot* through the experiments conducting on five common image-classification datasets: CIFAR10, CIFAR100 [36], FashionMNIST [37], EMNIST [38] and SVHN [39]. These datasets can be easily downloaded from Pytorch [40]. For CIFAR10, CIFAR100 and SVHN,



(a) *MGAI* on CIFAR10

(b) *MGAI* on FashionMNIST

FIGURE 3. **Improved local updates of *Slingshot*. We measure the metric called *MGAI* (Mean Global Accuracy Increase). *MGAI* measures the impacts of local updates on global accuracy. A larger *MGAI* means that the locally updated model performs better on the global test set. The concentration of the Dirichlet distribution β is set to 0.3.**

we use a CNN with two 5×5 convolutional layers as used in *FedAvg* [1]. For FashionMNIST and EMNIST, the classic LeNet [41] is adopted (the number of input channels of the first layer is changed to 1).

2) THE PARTITION OF DATASETS

The whole test dataset is stored at the FL server, but each client has only a subset of the train dataset during the entire FL training. We adopt a popular approach that appeared in FL papers to partition datasets [7], [8], [12]. Concretely, for each class, a distribution of data samples across all clients is generated by the Dirichlet distribution ($Dir(\beta)$). The parameter β is the concentration of the Dirichlet distribution, and a lower concentration leads to a more heterogeneous distribution of data. To implement different degrees of data heterogeneity, we set $\beta = 0.1, 0.3$, and 0.5, respectively. In these challenging settings, the data distribution and the amount of data can vary widely from client to client.

3) FEDERATED LEARNING SETTINGS

In each communication round, a central FL server randomly samples a specified number of clients to perform local SGD via a fixed random seed. For CIFAR10, CIFAR100, and SVHN, we set 1000 communication rounds, and 10 out of 100 clients are selected for each round. For FashionMNIST, we set 300 communication rounds, and 10 out of 200 clients are selected for each round. For EMNIST, we set 200 communication rounds, and 10 out of 100 clients are selected. Unless otherwise specified, we set the number of local epochs in each round to 5, the batch size in the local update phase to 64, the initial learning rate to 0.1, the decay rate of learning rate in each round to 0.998 [11], the momentum to 0.9, and the weight decay to 10^{-4}, respectively. We also apply the same data-augmentation techniques for each dataset.

4) BASELINES AND HYPER-PARAMETERS

We compare *Slingshot* with the standard FL paradigm *FedAvg*, as well as other three popular methods in data-heterogeneity FL including *FedProx* [10], *FedGa* [15], and *Moon* [8]. For CIFAR100, we tune μ of *FedProx* from {0.1, 0.01, 0.001} and use the best μ 0.01, α of *FedGa* is set to 0.05 tuned from {0.1, 0.05, 0.025}, μ of *Moon* is set to 0.01 tuned from {1, 0.1, 0.01, 0.001} and α of *Slingshot* is set to 0.2 tuned from {0.2, 0.1, 0.05}. For CIFAR10, the best μ of *Moon* is changed to 0.001, and the best α of *Slingshot* is 0.1. In order to fairly compare the generalization of these methods, their specific hyper-parameters for the other tasks are set to the best for CIFAR10. We explore the sensitivity of *Slingshot's* hyper-parameters α and show the results in Fig. 6. All the settings of μ in *Slingshot* are the same as those set in *FedProx*.

B. IMPROVED LOCAL UPDATES

Our key idea is to help local updates more globally favorable in data-heterogeneity FL. Thus, we compare the global impacts of local updates of various federated learning baselines. Our proposed metric *MGAI* measures the increase in global accuracy of the locally updated models, which shows the effectiveness of local updates. As shown in Fig. 3, *Slingshot* improves the value of local updates compared to other baselines, especially with a large number of local epochs. This is because Slingshot can not only appropriately guide the directions of local updates but also preserve the possibility of a large improvement of the updated model.

Effective local update results in good performance of the global model. We compare the performance of *Slingshot* with

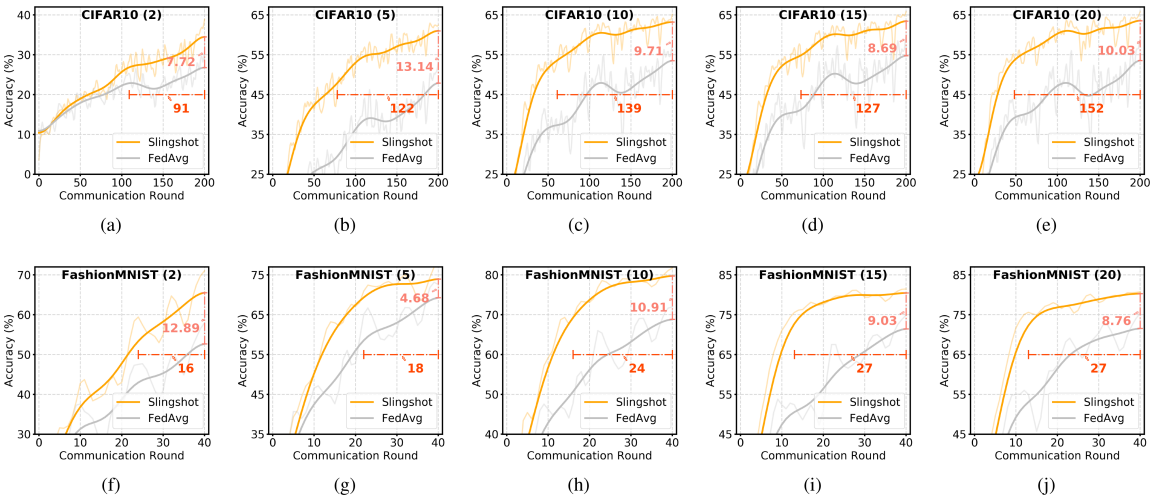


FIGURE 4. **Comparison of the early critical stage performance between *Slingshot* and *FedAvg* under the different number of local epochs. The number in each black bracket represents the number of local epochs. In addition, we mark the differences in performance and communication efficiency between *Slingshot* and *FedAvg*, respectively. The oscillation is the true test accuracy curve.**

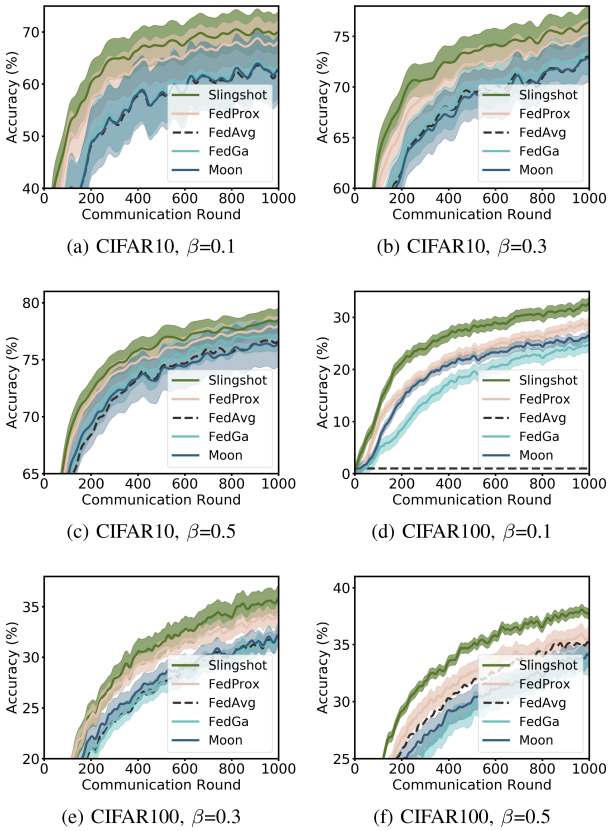


FIGURE 5. **Global accuracy v.s. communication rounds. A smaller β represents a greater degree of heterogeneous data. *FedAvg* is stuck at 1% accuracy in (c). The color band represents the oscillation of the true accuracy curve of the corresponding method.**

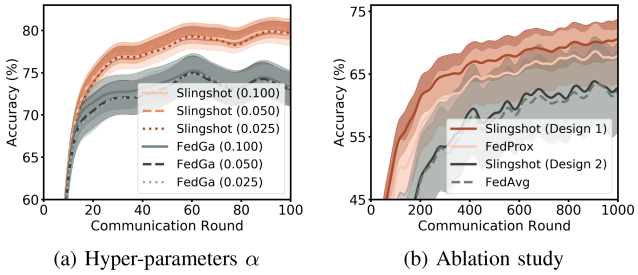


FIGURE 6. **Effectiveness of *Slingshot*. (a) Testing hyper-parameters α of *Slingshot* on EMNIST. (b) Ablation study on CIFAR10. The color band represents the oscillation of the true accuracy curve of the corresponding method.**

that of *FedAvg* in the early stages (known as critical periods). The experimental results show that *MGAI* is positively correlated with model performance. *Slingshot* has better model performance than *FedAvg*, with a higher *MGAI*. Note that the number of local epochs in a communication round is a crucial parameter of FL. A simple and crude way to reduce the impact of heterogeneous data in *FedAvg* is to reduce the number of local epochs. However, decreasing local epochs seriously increases communication costs. It is generally believed that a large number of local epochs causes a serious challenge of data heterogeneity. Thus, we conduct five groups of experiments corresponding to local epochs = 2, 5, 10, 15, and 20 respectively, to test the performance under various degrees of data heterogeneity. Fig. 4 shows that the acceleration of *Slingshot* in the early stage is robust for different local epochs. This observation suggests that *Slingshot* works well under various degrees of data heterogeneity.

C. CONVERGENCE COMPARISON

1) NON-IID DATASET AND PARTIAL PARTICIPANTS

The reason why the early training stage is called the critical round is that the early training often determines the quality of the final model and the overall convergence speed of the algorithm. Our experimental results also prove that the convergence rate and model performance in the early training stage imply the eventual convergence rate and model

TABLE 2. Global Model Performance and Communication Efficiency for Five Benchmark Datasets

Non-IID	Dirichlet conc. = 0.1					Dirichlet conc. = 0.3					Dirichlet conc. = 0.5				
Method	FedAvg	FedProx	FedGa	Moon	Slingshot	FedAvg	FedProx	FedGa	Moon	Slingshot	FedAvg	FedProx	FedGa	Moon	Slingshot
Dataset 1	CIFAR10: Sample CNN, 1000 rounds														
Top-1 acc.	63.10	68.22	63.78	63.27	**70.18**	72.85	75.04	73.20	72.72	**76.18**	76.62	77.87	77.10	76.35	**78.39**
R (69±6 %)	908	380	880	898	**254**	457	300	444	484	**208**	625	412	522	661	**346**
R (67±6 %)	663	290	619	647	**219**	352	221	340	378	**166**	380	265	348	372	**230**
Dataset 2	CIFAR100: Sample CNN, 1000 rounds														
Top-1 acc.	1.00	28.68	24.60	26.13	**32.88**	31.68	33.97	31.53	31.82	**36.94**	35.10	35.67	33.94	33.48	**37.87**
R (31±4 %)	/	813	/	/	**372**	861	552	902	849	**370**	914	835	/	/	**508**
R (29±4 %)	/	602	822	/	**304**	640	395	658	593	**294**	637	553	881	841	**382**
Dataset 3	FashionMNIST: LeNet, 300 rounds														
Top-1 acc.	80.00	82.62	83.91	80.20	**84.34**	84.54	85.13	84.85	84.23	**86.02**	85.96	85.82	85.73	85.83	**86.17**
R (84±2 %)	/	230	218	/	**163**	248	150	190	276	**91**	/	/	/	/	**262**
R (82±2 %)	283	167	**121**	280	123	89	68	84	88	**56**	111	100	99	104	**85**
Dataset 4	EMNIST: LeNet, 200 rounds														
Top-1 acc.	75.62	80.21	75.69	75.63	**81.57**	78.14	82.31	78.17	77.79	**83.20**	78.33	82.50	78.25	78.11	**83.34**
R (77±2 %)	193	25	179	195	**22**	73	18	74	78	**17**	/	**20**	/	/	**20**
R (75±2 %)	51	**17**	48	50	**17**	25	14	22	25	**12**	46	**14**	45	49	**14**
Dataset 5	SVHN: Sample CNN, 1000 rounds														
Top-1 acc.	82.45	87.40	84.54	84.58	**88.45**	89.40	90.30	89.70	89.47	**91.25**	90.76	90.67	90.66	91.03	**91.97**
R (86±3 %)	/	202	630	/	**150**	300	130	197	289	**93**	472	275	493	385	**170**
R (84±3 %)	628	142	379	627	**98**	197	83	126	144	**77**	220	121	262	187	**88**

The R (a±b %) means the number of communication rounds that achieve the target accuracy. For Dirichlet concentration = 0.1, 0.3, and 0.5, the target accuracy is (a-b) %, a %, and (a+b) %, respectively. The "/" indicates a failure to achieve the target accuracy.

performance. In addition, the positive relationship between *MGAI* and the final model performance is also confirmed. Thus, researchers can predict the final performance of various FL algorithms by measuring *MGAI* during the early testing phase.

Table 2 shows the convergence of the mentioned methods for five benchmark datasets by comparing both the top-1 accuracy of the global model and the number of rounds that achieve the target accuracy. In order to better compare the convergence, we set higher target accuracies for the settings with larger β (smaller challenge). Compared to other baselines, *Slingshot* shows the best model performance and communication efficiency in almost all settings. For CIFAR10 ($\beta = 0.1$), the top-1 accuracy of *Slingshot* is 7.08% higher than that of *FedAvg*, and the number of communication rounds to achieve 63% accuracy is reduced by 72%. *Slingshot* is also 5.30× faster than *FedAvg* on EMNIST with fewer challenges of convergence. *Slingshot's* local updates are more globally beneficial and not pulled by different local optima, thus *Slingshot* achieves the target accuracy with fewer communication rounds.

Among the five dataset tasks, the most difficult one is CIFAR100. For CIFAR100 with heterogeneous data, some methods even cannot train effectively. For instance, when $\beta = 0.1$, the global accuracy in *FedAvg* is stuck at 1%. However, in this case of significantly heterogeneous data, *Slingshot* still converges best. Sometimes, other methods are no better than vanilla *FedAvg* while *Slingshot* is always better than *FedAvg* with heterogeneous data. It is because *Slingshot* is not overly constrained by objective consistency and not seriously affected by cumulative Global Drift such as *FedProx*, but it is more general to choose the globally favorable direction to update.

TABLE 3. Top-1 Accuracy, With IID Dataset and Full Participants

Method	FedAvg	FedProx	FedGa	Moon	Slingshot
CIFAR10	80.47	80.9	80.34	80.85	81.3
FashionMNIST	89.21	89.20	89.23	88.98	89.45

The test accuracy results of training models on CIFAR10 and CIFAR100 are shown in Fig. 5. In each communication round, only a subset of clients are selected by the central server, and these clients may have little data in data-heterogeneity settings. In addition, the selected clients may have data that are useless or even have negative impacts on global training at this stage. Thus, the test accuracy results of all methods have some degree of oscillation. However, *Slingshot* captures the globally favorable direction even if with oscillation, resulting in faster convergence.

2) IID DATASET AND FULL PARTICIPANTS

Table 3 shows that *Slingshot* does not degrade performance with full participants and IID dataset, where all clients have the same data distribution and the same amount of data. Due to the feature drifts among samples and the randomness of stochastic gradient descent, *Slingshot* is also effective under the setting of IID and full participants.

D. ROBUSTNESS AND EFFECTIVENESS

It is not our intention to get a competitive method by tuning hyper-parameters. Specific hyper-parameters, however, definitely affect the convergence of all methods. We explore the sensitivity of *Slingshot's* parameter α in Fig. 6(a). This sensitivity is similar to that of *FedGa* while *Slingshot* outperforms

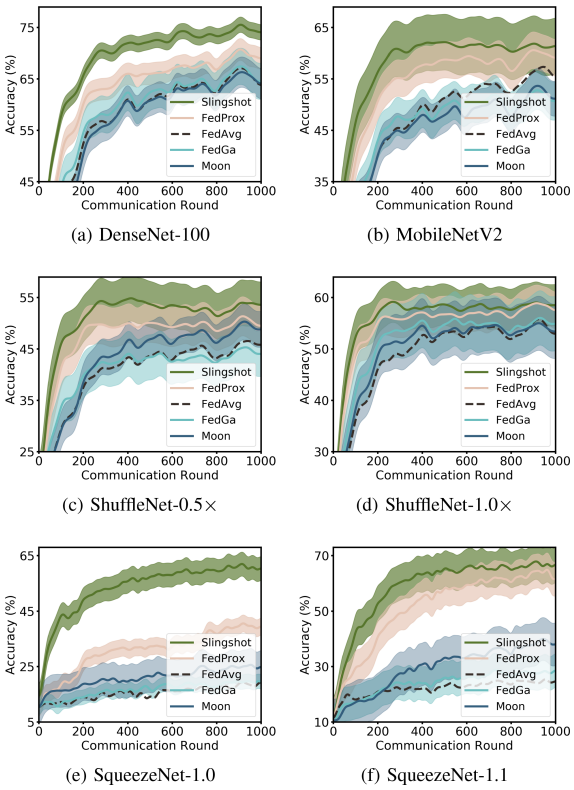


FIGURE 7. Model performance and convergence when training different models on CIFAR10 ($\beta = 0.1$). The color band represents the oscillation of the true accuracy curve of the corresponding method.

FedGa for all α. What we want to emphasize is that the designs of *Slingshot* are generalized to help solve the problem of data heterogeneity. Fig. 6(b) shows both *Slingshot's* designs can speed up data-heterogeneity FL and Design 1 has a larger impact than Design 2. Despite using a similar loss function adopted by *FedProx*, *Slingshot* is more effective than *FedProx*. This is because *Slingshot* constrains local updates to a globally favorable direction rather than a global model given a Global Drift.

Since deep models are transmitted in FL instead of data, FL is often associated with high communication costs. It is natural to choose a lightweight model in FL. However, experimental results show that lightweight models that are more effective in centralized training are possibly more difficult to train in data-heterogeneity FL. This is because such lightweight models are compressed. Compared with the sparse model, their gradient norm is larger, so they are more susceptible to the drifts of FL. Fig. 7 shows that *Slingshot* can be applied to various lightweight models to tackle the problem of data heterogeneity. For example, *Slingshot* achieves 46.52% improvement on test accuracy and 48.21× speedup for a lightweight neural network named *SqueezeNet-1.0*.

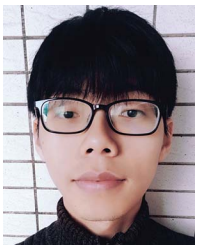
VI. CONCLUSION

A major challenge in federated learning (FL) is the data-heterogeneity problem. Previous approaches have paid plenty of attention to alleviating data-heterogeneity FL by focusing on objective inconsistency. Differently, we find that it is a significant challenge to guarantee local updates that are globally favorable. To address this challenge, we carried out the following two attempts. First, we propose a new metric called Mean Global Accuracy Increase (*MGAI*) to evaluate what kind of local updates are globally favorable. *MGAI* helps the researchers predict the final performance of various FL algorithms during the early testing phase. Meanwhile, this metric helps researchers understand that truly effective local updates for FL should point to global optimum and be of appropriate length. Such local updates can greatly improve *MGAI* and the final performance of the global model. Thus, we propose a new method of FL called *Slingshot* that exploits the globally favorable direction and the quality of local updates. Our experiments demonstrate a faster convergence of FL training under the proposed *Slingshot*, as well as its robustness and effectiveness. Hopefully, our studies can spark more discussions about the directions and the quality of local updates in FL.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A.Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*2017, pp. 1273–1282.
[2] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Mach. Intell.*, vol. 2, no. 6, pp. 305–311, 2020.
[3] S. Warnat-Herresthal et al., "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021.
[4] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-IID data," in *Proc. Int. Conf. Learn. Representations*, 2019.
[5] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-IID data silos: An experimental study," in *Proc. IEEE 38th Int. Conf. Data Eng.*, 2022, pp. 965–978.
[6] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143.
[7] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, "No fear of heterogeneity: Classifier calibration for federated learning with non-IID data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 5972–5984.
[8] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10713–10722.
[9] M. Jiang, Z. Wang, and Q. Dou, "Harmofl: Harmonizing local and global drifts in federated learning on heterogeneous medical images," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 1087–1095.
[10] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, 2020, pp. 429–450.
[11] A. E. Durmus, Z. Yue, M. Ramon, M. Matthew, W. Paul, and S. Venkatesh, "Federated learning based on dynamic regularization," in *Proc. Int. Conf. Learn. Representations*, 2021.
[12] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu, and C.-Z. Xu, "FedDC: Federated learning with non-IID data via local drift decoupling and correction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10112–10121.
[13] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 7611–7623.
[14] M. Mendieta, T. Yang, P. Wang, M. Lee, Z. Ding, and C. Chen, "Local learning matters: Rethinking data heterogeneity in federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8397–8406.

[15] Y. Dandi, L. Barba, and M. Jaggi, "Implicit gradient alignment in distributed and federated learning," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 6454–6462.
[16] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," in *Int. Conf. Learn. Representations*, 2020.
[17] X. Li, M. JIANG, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated learning on non-IID features via local batch normalization," in *Proc. Int. Conf. Learn. Representations*, 2020.
[18] S. Reddi et al., "Adaptive federated optimization," in *Proc. Int. Conf. Learn. Representations*, 2021.
[19] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychol. Learn. Motivation*, vol. 24. Cambridge, MA, USA: Academic Press, pp. 109–165, 1989.
[20] W. Huang, M. Ye, and B. Du, "Learn from others and be yourself in heterogeneous federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10143–10153.
[21] L. Qu et al., "Rethinking architecture design for tackling data heterogeneity in federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10061–10071.
[22] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Netw.*, vol. 113, pp. 54–71, 2019.
[23] M. De Lange et al., "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, Jul. 2022.
[24] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, 2017.
[25] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2017.
[26] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3987–3995.
[27] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6470–6479.
[28] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 350–360.
[29] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2001–2010.
[30] A. A. Rusu et al., "Progressive neural networks," 2016, *arXiv:1606.04671*.
[31] C. Fernando et al., "Pathnet: Evolution channels gradient descent in super neural networks," 2017, *arXiv:1701.08734*.
[32] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7765–7773.
[33] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 67–82.
[34] G. Yan, H. Wang, and J. Li, "Seizing critical learning periods in federated learning," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 8788–8796.
[35] A. Xu and H. Huang, "Coordinating momenta for cross-silo federated learning," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 8735–8743.
[36] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," 2009.
[37] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[38] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2017, pp. 2921–2926.
[39] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.
[40] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.
[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.



JIALIANG LIU is currently working toward the master's degree with the School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China. His research interest include federated learning and distributed systems.

HUAWEI HUANG (Senior Member, IEEE) received the Ph.D. degree from The University of Aizu, Aizuwakamatsu, Japan, in 2016. He is currently an Associate Professor with Sun Yat-Sen University, Guangzhou, China. He was a research Fellow of JSPS, and a program-specific Assistant Professor with Kyoto University, Kyoto, Japan. His research interests include blockchain and distributed computing. He was the lead guest Editor of multiple blockchain special issues at IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, and IEEE OPEN JOURNAL OF THE COMPUTER SOCIETY. He was also a TPC Chair of multiple blockchain conferences and workshops.

CHUN WANG is currently working toward the master's degree with the School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China. His research interest include federated learning and distributed systems.

SICONG ZHOU is currently working toward the master's degree with the School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China. His research interest include federated learning and distributed systems.

RUIXIN LI is currently working toward the master's degree with the School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China. His research interest include federated learning and distributed systems.

ZIBIN ZHENG (Fellow, IEEE) received the Ph.D. degree from the Chinese University of Hong Kong, Hong Kong, in 2012. He is currently a full Professor with the School of Software Engineering, Sun Yat-Sen University, Guangzhou, China. His research interests include service computing and cloud computing. Prof. Zheng was the recipient of the Outstanding Ph.D. Dissertation Award of the Chinese University of Hong Kong in 2012, ACM SIGSOFT Distinguished Paper Award at ICSE in 2010, Best Student Paper Award at ICWS2010, and IBM Ph.D. Fellowship Award in 2010. He was a PC Member of IEEE CLOUD, ICWS, SCC, ICSOC, and SOSE.