FEDSPEED: LARGER LOCAL INTERVAL, LESS COMMUNICATION ROUND, AND HIGHER GENERALIZATION ACCURACY

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning is an emerging distributed machine learning framework which jointly trains a global model via a large number of local devices with data privacy protections. Its performance suffers from the non-vanishing biases introduced by the local inconsistent optimal and the rugged client-drifts by the local over-fitting. In this paper, we propose a novel and practical method, FedSpeed, to alleviate the negative impacts posed by these problems. Concretely, FedSpeed applies the prox-correction term on the current local updates to efficiently reduce the biases introduced by the prox-term, a necessary regularizer to maintain the strong local consistency. Furthermore, FedSpeed merges the vanilla stochastic gradient with a perturbation computed from an extra gradient ascent step in the neighborhood, thereby alleviating the issue of local over-fitting. Our theoretical analysis indicates that the convergence rate is related to both the communication rounds T and local intervals K with a upper bound $\mathcal{O}(1/T)$ if setting a proper local interval. Moreover, we conduct extensive experiments on the real-world dataset to demonstrate the efficiency of our proposed FedSpeed, which performs significantly faster and achieves the state-of-the-art (SOTA) performance on the general FL experimental settings than several baselines including FedAvg, FedProx, FedCM, FedAdam, SCAFFOLD, FedDyn, FedADMM, etc.

1 INTRODUCTION

Since McMahan et al. (2017) proposed federated learning (FL), it has gradually evolved into an efficient paradigm for large-scale distributed training. Different from the traditional deep learning methods, FL allows multi local clients to jointly train a single global model without data sharing. However, FL is far from its maturity, as it still suffers from the considerable performance degradation over the heterogeneously distributed data, a very common setting in the practical application of FL.

We recognize the main culprit leading to the performance degradation of FL as *local inconsistency* and *local heterogeneous over-fitting*. Specifically, for canonical local-SGD-based FL method, e.g., FedAvg, the non-vanishing biases introduced by the local updates may eventually lead to inconsistent local solution. Then, the rugged client-drifts resulting from the local over-fitting into inconsistent local solutions may make the obtained global model degrading into the average of client's local parameters. The non-vanishing biases have been studied by several previous works Charles & Konečnỳ (2021); Malinovskiy et al. (2020) in different forms. The inconsistency due to the local heterogeneous data will compromise the global convergence during the training process. Eventually it leads to serious client-drifts which can be formulated as $\mathbf{x}^* \neq \sum_{i\in[m]} \mathbf{x}_i^*/m$. Larger data heterogeneity may enlarge the drifts, thereby degrading the practical training convergence rate and generalization performance.

In order to strengthen the local consistency during the local training process, and avoid the client-drifts resulting from the local over-fitting, we propose a novel and practical algorithm, dubbed as **FedSpeed**. Notably, FedSpeed incorporates two novel components to achieve SOTA performance. i) Firstly, FedSpeed inherits a penalized prox-term to force the local offset to be closer to the initial point at each communication round. However, recognized from Hanzely & Richtárik (2020); Khaled et al. (2019) that the prox-term between global and local solutions may introduce undesirable local training bias, we propose and utilize a prox-correction term to counteract the adverse impact. Indeed, in our

theoretical analysis, the implication of the prox-correction term could be considered as a momentum-based term of the weighted local gradients. Via utilizing the historical gradient information, the bias brought by the prox-term can be effectively corrected. ii) Secondly, to avoid the rugged local over-fitting, FedSpeed incorporates a local gradient perturbation via merging the vanilla stochastic gradient with an extra gradient, which can be viewed as taking an extra gradient ascent step for each local update. Based on the analysis in Zhao et al. (2022); van der Hoeven (2020), we demonstrate that the gradient perturbation term could be approximated as adding a penalized squared $L2$-norm of the stochastic gradients to the original objective function, which can efficiently search for the flatten local minima Andriushchenko & Flammarion (2021) to prevent the local over-fitting problems.

We also provide the theoretical analysis of our proposed FedSpeed and further demonstrate that its convergence rate could be accelerated by setting an appropriate large local interval K. Explicitly, under the non-convex and smooth cases, FedSpeed with an extra gradient perturbation could achieve the fast convergence rate of $\mathcal{O}(1/T)$, which indicates that FedSpeed achieves a tighter upper bound with a proper local interval K to converge, without applying a specific global learning rate or assuming the precision for the local solutions (Durmus et al., 2021; Wang et al., 2022). Extensive experiments are tested on CIFAR-10/100 and TinyImagenet dataset with a standard ResNet-18-GN network under the different heterogeneous settings, which shows that our proposed FedSpeed is significantly better than several baselines, e.g. for FedAvg, FedProx, FedCM, FedPD, SCAFFOLD, FedDyn, on both the stability to enlarge the local interval K and the test generalization performance in the actual training.

To the end, we summarize the main contributions of this paper as follows:

- We propose a novel and practical federated optimization algorithm, **FedSpeed**, which applies a prox-correction term to significantly reduce the bias due to the local updates of the prox-term, and an extra gradient perturbation to efficiently avoid the local over-fitting, which achieves a fast convergence speed with large local steps and simultaneously maintains the high generalization.
- We provide the convergence rate upper bound under the non-convex and smooth cases and prove that FedSpeed could achieve a fast convergence rate of $\mathcal{O}(1/T)$ via enlarging the local training interval $K = \mathcal{O}(T)$ without any other harsh assumptions or the specific conditions required.
- Extensive experiments are conducted on the CIFAR-10/100 and TinyImagenet dataset to verify the performance of our proposed FedSpeed. To the best of our interests, both convergence speed and generalization performance could achieve the SOTA results under the general federated settings. FedSpeed could outperform other baselines and be more robust to enlarging the local interval.

2 Related Work

McMahan et al. (2017) propose the federated framework with the properties of jointly training with several unbalance and non-iid local dataset via communicating with lower costs during the total training stage. The general FL optimization involves a local client training stage and a global server update operation Asad et al. (2020) and it has been proved to achieve a linear speedup property in Yang et al. (2021). With the fast development of the FL, a series of efficient optimization method are applied in the federated framework. Li et al. (2020b) and Kairouz et al. (2021) introduce a detailed overview in this field. There are still many difficulties to be solved in the practical scenarios, while in this paper we focus to highlight the two main challenges of the local inconsistent solution and client-drifts due to heterogeneous over-fitting, which are two acute limitations in the federated optimization Li et al. (2020a); Yang et al. (2019); Konečnỳ et al. (2016); Liu et al. (2022).

Local consistency. Sahu et al. (2018) study the non-vanishing biases of the inconsistent solution in the experiments and apply a prox-term regularization, an extra penalized L_2-norm term on local updates to force the local solution be close to the initial point at round t. FedProx utilizes the bounded local updates by penalizing parameters to provide a good guarantee of consistency. Charles & Konečnỳ (2021); Malinovskiy et al. (2020) show that the local learning rate decay can balance the trade-off between the convergence rate and the local inconsistency with the rate of $\mathcal{O}(\eta_l(K-1))$. Furthermore, Wang et al. (2021; 2020b) through a simple counterexample to show that using adaptive optimizer or different hyper-parameters on local clients leads to an additional gaps. They propose a local correction technique to alleviate the biases. Wang et al. (2020a); Tan et al. (2022) consider the different local settings and prove that in the case of asynchronous aggregation, the inconsistency bias will no longer be eliminated by local learning rate decay. They propose a novel aggregation method

FedNova to weighted each local offset instead of the model parameters. Zhang et al. (2021) apply the primal dual method instead of the primal method to solve a series of sub-problems on the local clients and alternately updates the primal and dual variables which can achieve the fast convergence rate of $\mathcal{O}(\frac{1}{T})$ with the local solution precision assumption. Based on FedPD, Durmus et al. (2021) propose FedDyn as a variants via averaging all the dual variables (the average quantity can then be viewed as the global gradient) under the partial participation settings, which can also achieve the same $\mathcal{O}(\frac{1}{T})$ under the assumption that exact local solution can be found by the local optimizer. Wang et al. (2022); Gong et al. (2022) propose two other variants to apply different dual variable aggregation strategies under partial participation settings. These methods benefit from applying the prox-term Li et al. (2019); Chen & Chao (2020) or higher efficient optimization methods Bischoff et al. (2021); Yang et al. (2022) to control the local consistency.

Client-drifts. Karimireddy et al. (2020) firstly demonstrate the client-drifts for federated learning framework to indicate the negative impact on the global model when each local client over-fits to the local heterogeneous dataset. They propose SCAFFOLD via a variance reduction technique to mitigate this drifts. Yu et al. (2019) and Wang et al. (2019) introduce the momentum instead of the gradient to the local and global update respectively to improve the generalization performance. To maintain the property of consistency, Xu et al. (2021) propose a novel client-level momentum term to improve the local training process. Ozfatura et al. (2021) incorporate the client-level momentum with local momentum to further control the biases. In recent Gao et al. (2022); Kim et al. (2022), they propose a drift correction term as a penalized loss on the original local objective functions with a global gradient estimation. Chen et al. (2020) and Chen et al. (2021) focus on the adaptive method to alleviate the biases and improve the efficiency.

Our proposed FedSpeed inherits the prox-term at local update to guarantee the local consistency during local training. Different from the previous works, we adopt an extra prox-correction term to reduce the bias during the local training introduced by the update direction towards the last global model parameters. This ensures that the local update could be corrected towards the global minima. Furthermore, we incorporate a gradient perturbation update to enhance the generalization performance of the local model, which merges a gradient ascent step. FedSpeed has been shown to accelerate the convergence as fast as $\mathcal{O}(1/T)$ by increasing the local interval under non-convex smooth case.

3 METHODOLOGY

In this part, we will introduce the preliminaries and our proposed method. We will explain the implicit meaning for each variables and demonstrate the FedSpeed algorithm inference in details.

Notations and preliminary. Let m be the number of total clients. We denote $\mathcal{S}^t$ as the set of active clients at round t. K is the number of local updates and T is the communication rounds. $(\cdot)_{i,k}^t$ denotes variable $(\cdot)$ at k-th iteration of t-th round in the i-th client. $\mathbf{x}$ is the model parameters. $\mathbf{g}$ is the stochastic gradient computed by the sampled data. $\tilde{\mathbf{g}}$ is the weighted quasi-gradient computed as defined in Algorithm 1. $\hat{\mathbf{g}}$ is the prox-correction term. We denote $\langle \cdot, \cdot \rangle$ as the inner product for two vectors and $\| \cdot \|$ is the Euclidean norm of a vector. Other symbols are detailed at their references.

As the most FL frameworks, we consider to minimize the following finite-sum non-convex problem:

$$F(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^{m} F_i(\mathbf{x}), \tag{1}$$

where $F\colon \mathbb{R}^d \to \mathbb{R}$, $F_i(\mathbf{x}) := \mathbb{E}_{\varepsilon_i \sim \mathcal{D}_i} F_i(\mathbf{x}, \varepsilon_i)$ is objective function in the client i, and ε_i represents for the random data samples obeying the distribution $\mathcal{D}_i$. m is the total number of clients. In FL, $\mathcal{D}_i$ may differ across the local clients, which may introduce the client drifts by the heterogeneous data.

3.1 FEDSPEED ALGORITHM

In this part, we will introduce our proposed method to alleviate the negative impact of the heterogeneous data and reduces the communication rounds. We are inspired by the dynamic regularization Durmus et al. (2021) for the local updates to eliminate the client drifts when T approaches infinite.

Our proposed FeedSpeed is shown in Algorithm 1. At the beginning of each round t, a subset of clients $\mathcal{S}^t$ are required to participate in the current training process. The global server will

Algorithm 1 FedSpeed Algorithm Framework

Input: model parameters $\mathbf{x}^0$, total communication rounds T, local gradient controller $\hat{\mathbf{g}}_i^{-1} = 0$, penalized weight λ.
Output: model parameters $\mathbf{x}^T$.

1: **for** $t = 0, 1, 2, \cdots, T-1$ **do**
2: select active clients-set $\mathcal{S}^t$ at round t
3: **for** client $i \in \mathcal{S}^t$ parallel **do**
4: communicate $\mathbf{x}^t$ to local client i and set $\mathbf{x}_{i,0}^t = \mathbf{x}^t$
5: **for** $k = 0, 1, 2, \cdots, K-1$ **do**
6: sample a minibatch $\varepsilon_{i,k}^t$ and do
7: compute unbiased stochastic gradient: $\mathbf{g}_{i,k,1}^t = \nabla F_i(\mathbf{x}_{i,k}^t; \varepsilon_{i,k}^t)$
8: update the extra step: $\breve{\mathbf{x}}_{i,k}^t = \mathbf{x}_{i,k}^t + \rho \mathbf{g}_{i,k,1}^t$
9: compute unbiased stochastic gradient: $\mathbf{g}_{i,k,2}^t = \nabla F_i(\breve{\mathbf{x}}_{i,k}^t; \varepsilon_{i,k}^t)$
10: compute quasi-gradient: $\tilde{\mathbf{g}}_{i,k}^t = (1-\alpha)\mathbf{g}_{i,k,1}^t + \alpha \mathbf{g}_{i,k,2}^t$
11: update the gradient descent step: $\mathbf{x}_{i,k+1}^t = \mathbf{x}_{i,k}^t - \eta_l\big(\tilde{\mathbf{g}}_{i,k}^t - \hat{\mathbf{g}}_i^{t-1} + \frac{1}{\lambda}(\mathbf{x}_{i,k}^t - \mathbf{x}^t)\big)$
12: **end for**
13: $\hat{\mathbf{g}}_i^t = \hat{\mathbf{g}}_i^{t-1} - \frac{1}{\lambda}(\mathbf{x}_{i,K}^t - \mathbf{x}^t)$
14: communicate $\hat{\mathbf{x}}_i^t = \mathbf{x}_{i,K}^t - \lambda \hat{\mathbf{g}}_i^t$ to the global server
15: **end for**
16: $\mathbf{x}^{t+1} = \frac{1}{S}\sum_{i \in \mathcal{S}^t} \hat{\mathbf{x}}_i^t$
17: **end for**

communicate the parameters $\mathbf{x}^t$ to the active clients for local training. Each active local client performs three stages: (1) computing the unbiased stochastic gradient $\mathbf{g}_{i,k,1}^t = \nabla F_i(\mathbf{x}_{i,k}^t; \varepsilon_{i,k}^t)$ with a randomly sampled mini-batch data $\varepsilon_{i,k}^t$ and executing a gradient ascent step in the neighbourhood to approach $\breve{\mathbf{x}}_{i,k}^t$; (2) computing the unbiased stochastic gradient $\mathbf{g}_{i,k,2}^t$ with the same sampled mini-batch data in (1) at the $\breve{\mathbf{x}}_{i,k}^t$ and merging the $\mathbf{g}_{i,k,1}^t$ with $\mathbf{g}_{i,k,2}^t$ to introduce a basic perturbation to the vanilla descent direction; (3) executing the gradient descent step with the merged quasi-gradient $\tilde{\mathbf{g}}_{i,k}^t$, the prox-term $\|\mathbf{x}_{i,k}^t - \mathbf{x}^t\|^2$ and the local prox-correction term $\hat{\mathbf{g}}_i^{t-1}$. After K iterations local training, prox-correction term $\hat{\mathbf{g}}_i^{t-1}$ will be updated as the weighted sum of the current local offset $(\mathbf{x}_{i,K}^t - \mathbf{x}_{i,0}^t)$ and the historical offsets momentum. Then we communicate the amended model parameters $\hat{\mathbf{x}}_i^t = \mathbf{x}_{i,K}^t - \lambda \hat{\mathbf{g}}_i^t$ to the global server for aggregation. On the global server, a simple average aggregation is applied to generate the current global model parameters $\mathbf{x}^{t+1}$ at round t.

Prox-correction term. In the general optimization, the prox-term $\|\mathbf{x}_{i,k}^t - \mathbf{x}^t\|^2$ is a penalized term for solving the non-smooth problems and it contributes to strengthen the local consistency in the FL framework by introducing a penalized direction in the local updates as proposed in Sahu et al. (2018). However, as discussed in Hanzely & Richtárik (2020), it simply performs as a balance between the local and global solutions, and there still exists the non-vanishing inconsistent biases among the local solutions, i.e., the local solutions are still largely deviated from each other, implying that local inconsistency is still not eliminated, which limits the efficiency of the federated learning framework.

To further strengthen the local consistency, we utilize a prox-correction term $\hat{\mathbf{g}}_i^t$ which could be considered as a previous local offset momentum. According to the local update, we combine the $\mathbf{x}_{i,k-1}^t$ term in the prox term and the local state, setting the weight as $(1 - \frac{\eta_l}{\lambda})$ multiplied to the basic local state. As shown in the local update in Algorithm 1 (Line.11), for $\forall\, \mathbf{x} \in \mathbb{R}^d$ we have:

$$\mathbf{x}_{i,K}^t - \mathbf{x}^t = -\gamma\lambda \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t + \gamma\lambda \hat{\mathbf{g}}_i^{t-1}, \tag{2}$$

where $\sum_{k=0}^{K-1} \gamma_k = \sum_{k=0}^{K-1} \frac{\eta_l}{\lambda}\big(1 - \frac{\eta_l}{\lambda}\big)^{K-1-k} = \gamma$. Proof details can be referred to the **Appendix**.

Firstly let $\hat{\mathbf{g}}_i^{-1} = \mathbf{0}$, Equation (2) indicates that the local offset will be transferred to a exponential average of previous local gradients when applying the prox-term, and the updated formation of the local offset is independent of the local learning rate η_l. This is different from the vanilla SGD-based methods, e.g. FedAvg, which treats all local updates fairly. γ_k changes the importance of the historical

gradients. As K increases, previous updates will be weakened by exponential decay significantly for $\eta_l < \lambda$. Thus, we apply the prox-correction term to balance the local offset. According to the iterative formula for $\hat{\mathbf{g}}_i^t$ (Line.13 in Algorithm 1) and the equation (2), we can rewrite this update as:

$$\hat{\mathbf{g}}_i^t = (1-\gamma)\hat{\mathbf{g}}_i^{t-1} + \gamma\Big(\sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t\Big), \tag{3}$$

where γ and γ_k is defined the same as in Equation (2). Proof details can be referred to the **Appendix**.

Note that $\hat{\mathbf{g}}_i^t$ performs as a momentum term of the historical local updates before round t, which can be considered as a estimation of the local offset at round t. At each local iteration k of round t, $\hat{\mathbf{g}}_i^{t-1}$ provides a correction for the local update to balance the impact of the prox-term to enhance the contribution of those descent steps executed firstly at each local stages. It should be noted that $\hat{\mathbf{g}}_i^{t-1}$ is different from the global momentum term mentioned in Wang et al. (2019) which aggregates the average local updates to improve the generalization performance. After the local training, it updates the current information. Then we subtract the current $\hat{\mathbf{g}}_i^t$ from the local models $\mathbf{x}_{i,K}^t$ to counteract the influence in the local stages. Finally it sends the post-processed parameters $\hat{\mathbf{x}}_{i,K}^t$ to the global server.

Gradient perturbation. Gradient perturbations significantly improves generalization for deep models. An extra gradient ascent in the neighbourhood can effectively express the curvature near the current parameters. Referring to the analysis in Zhao et al. (2022), we show that the quasi-gradient $\tilde{\mathbf{g}}$, which merges the extra ascent step gradient and the vanilla gradient, could be approximated as penalizing a square term of the $L2$-norm of the gradient on the original function. On each local client to solve the stationary point of $\min_{\mathbf{x}}\{F_i(\mathbf{x}) + \beta\|\nabla F_i(\mathbf{x})\|^2\}$ can search for a flat minima. Flatten loss landscapes will further mitigate the local inconsistency due to the averaging aggregation on the global server on heterogeneous dataset. Detailed discussions can be referred to the **Appendix**.

4 Convergence Analysis

In this part we will demonstrate the theoretical analysis of our proposed FedSpeed and illustrate the convergence guarantees under the specific hyperparameters. Due to the space limitations, more details could be referred to the **Appendix**. Some standard assumptions are stated as follows.

Assumption 4.1 ***(L-Smoothness)*** *For the non-convex function F_i holds the property of smoothness for all $i \in [m]$, i.e., $\|\nabla F_i(\mathbf{x}) - \nabla F_i(\mathbf{y})\| \leq L\|\mathbf{x}-\mathbf{y}\|$, for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.*

Assumption 4.2 ***(Bounded Stochastic Gradient)*** *The stochastic gradient $\mathbf{g}_{i,k}^t = \nabla F_i(\mathbf{x}_{i,k}^t, \varepsilon_{i,k}^t)$ with the randomly sampled data $\varepsilon_{i,k}^t$ on the local client i is an unbiased estimator of ∇F_i with bounded variance, i.e., $\mathbb{E}[\mathbf{g}_{i,k}^t] = \nabla F_i(\mathbf{x}_{i,k}^t)$ and $\mathbb{E}\|\mathbf{g}_{i,k}^t - \nabla F_i(\mathbf{x}_{i,k}^t)\|^2 \leq \sigma_l^2$, for all $\mathbf{x}_{i,k}^t \in \mathbb{R}^d$.*

Assumption 4.3 ***(Bounded Heterogeneity)*** *The dissimilarity of the dataset among the local clients is bounded by the local and global gradients, i.e., $\mathbb{E}\|\nabla F_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \sigma_g^2$, for all $\mathbf{x} \in \mathbb{R}^d$.*

Assumption 4.1 guarantees a Lipschitz continuity and Assumption 4.2 guarantees the stochastic gradient is bounded by zero mean and constant variance. Assumption 4.3 is the heterogeneity bound for the non-iid dataset, which is widely used in many previous works (Reddi et al., 2020; Yang et al., 2021; Xu et al., 2021; Wang et al., 2021; Karimi et al., 2021). Our theoretical analysis depends on the above assumptions to explore the comprehensive properties in the local training process.

Proof sketch. To express the essential insights in the updates of the Algorithm 1, we introduce two auxiliary sequences. Considering the $\mathbf{u}^t = \frac{1}{m}\sum_{i\in[m]} \mathbf{x}_{i,K}^t$ as the mean averaged parameters of the last iterations in the local training among the local clients. Based on $\{\mathbf{u}^t\}$, we introduce the auxiliary sequences $\{\mathbf{z}^t = \mathbf{u}^t + \frac{1-\gamma}{\gamma}(\mathbf{u}^t - \mathbf{u}^{t-1})\}_{t>0}$. Combining the local update and the Equation (3):

$$\mathbf{u}^{t+1} = \mathbf{u}^t - \lambda \frac{1}{m} \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \left(\gamma \tilde{\mathbf{g}}_{i,k}^t + (1-\gamma)\hat{\mathbf{g}}_i^{t-1}\right). \tag{4}$$

If we introduce K virtual states $\mathbf{u}_{i,k}^t$, it could be considered as a momentum-based update of the prox-correction term $\hat{\mathbf{g}}_i^{t-1}$ with the coefficient γ. And the prox-correction term $\hat{\mathbf{g}}_i^t = -\frac{1}{\lambda}\left(\mathbf{u}_{i,K}^t - \mathbf{u}_{i,0}^t\right)$,

which implies the global update direction in the local training process. And $\mathbf{z}^t$ updates as:

$$\mathbf{z}^{t+1} = \mathbf{z}^t - \lambda \frac{1}{m} \sum_{i \in [m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t, \tag{5}$$

Detailed proofs can be referred in the **Appendix**. After mapping $\mathbf{x}^t$ to $\mathbf{u}^t$, the local update could be considered as a client momentum-like method with a normalized weight parameterized by γ_k. Further, after mapping $\mathbf{u}^t$ to $\mathbf{z}^t$, the entire update process will be simplified to a SGD-type method with the quasi-gradients $\tilde{\mathbf{g}}$. $\mathbf{z}^t$ contains the penalized prox-term in the total local training stage. Though a prox-correction term is applied to eliminate the local biases, $\mathbf{x}^t$ maintains to be beneficial from the update of penalizing the prox-term. The prox-correction term plays the role as exponential average of the global offset. Then we introduce our proof of the convergence rate for the FedSpeed algorithm:

Theorem 4.4 *Under the Assumptions 4.1-4.3, when the perturbation learning rate satisfies $\rho \leq \frac{1}{\sqrt{6}\alpha L}$, and the local learning rate satisfies $\eta_l \leq \min\{\frac{1}{32\sqrt{3KL}}, 2\lambda\}$, and the local interval satisfies $K \geq \lambda/\eta_l$, let $\kappa = \frac{1}{2} - 3\alpha^2 L^2 \rho^2 - 1536\eta_l^2 L^2 K$ is a positive constant with selecting the proper η_l and ρ, the auxiliary sequence $\mathbf{z}^t$ in Equation (5) generated by executing the Algorithm 1 satisfies:*

$$\frac{1}{T}\sum_{t=1}^{T-1} \mathbb{E}\|\nabla F(\mathbf{z}^t)\|^2 \leq \frac{2(F(\mathbf{z}^1) - F^*)}{\lambda \kappa T} + \frac{64\eta_l L^2 K}{\kappa m T} \sum_{i \in [m]} \mathbb{E}\|\hat{\mathbf{g}}_i^0\|^2 + \frac{32\lambda^2 L^2}{\kappa T} \mathbb{E}_t \| \frac{1}{m} \sum_{i \in [m]} \hat{\mathbf{g}}_i^0 \|^2 + \Phi, \tag{6}$$

where F is a non-convex objective function F^* is the optimal of F. The term Φ is:

$$\Phi = \frac{1}{\kappa}\left(32\lambda\eta_l^2 L^2 K(16\sigma_g^2 + \sigma_l^2) + \lambda\alpha^2 L^2 \rho^2 (3\sigma_g^2 + \sigma_l^2)\right), \tag{7}$$

*where α is the perturbation weight. More proof details can be referred to the **Appendix**.*

Corollary 4.5 *Let $\rho = \mathcal{O}(1/\sqrt{T})$ with the upper bound of $\rho \leq 1/\sqrt{6}\alpha L$, and let $\eta_l = \mathcal{O}(1/K)$ with the lower bound of $\eta_l \geq \lambda/K$, when the local interval K is long enough with $K = \mathcal{O}(T)$, the proposed FedSpeed achieves a fast convergence rate of $\mathcal{O}(1/T)$.*

Remark 4.6 *Compared with the other prox-based works, e.g. for (Durmus et al., 2021; Wang et al., 2022; Gong et al., 2022), their proofs rely on the harsh assumption that local client must approach an exact stationary point or ϵ-inexact stationary point in the local training per round. It cannot be strictly satisfied in the practical federated learning framework with the current theoretical analysis of the last iteration point on the non-convex case. We relax this assumption through enlarging the local interval and prove that federated prox-based methods can also achieve the convergence of $\mathcal{O}(1/T)$.*

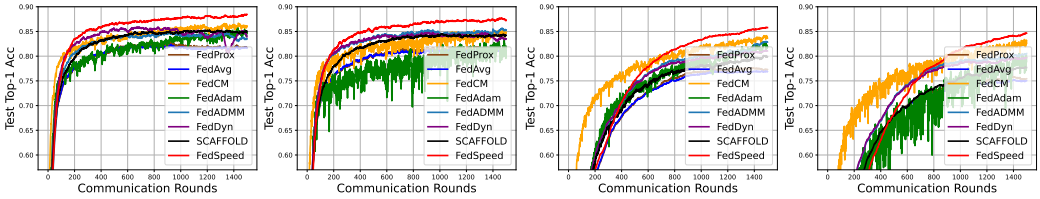
Remark 4.7 *Compared with the other current methods, FedSpeed can improve the convergence rate by increasing the local interval K, which is a good property for the practical federated learning framework. For the analysis of FedAvg (Yang et al., 2021), under the same assumptions, it achieves $\mathcal{O}(1/\sqrt{SKT} + K/T)$ which restricts the value of K to not exceed the order of T. Karimireddy et al. (2020) contribute the convergence as $\mathcal{O}(1/\sqrt{SKT})$ under the constant local interval, and (Reddi et al., 2020) proves the same convergence under the strict coordinated bounded variance assumption for the global full gradient in the FedAdam. Our experiments also verify this characteristic in Section 5.3. Most current algorithms are affected by increasing K in the training while FedSpeed shows the good stability under the enlarged local intervals and shrunk communication rounds.*

5 EXPERIMENTS

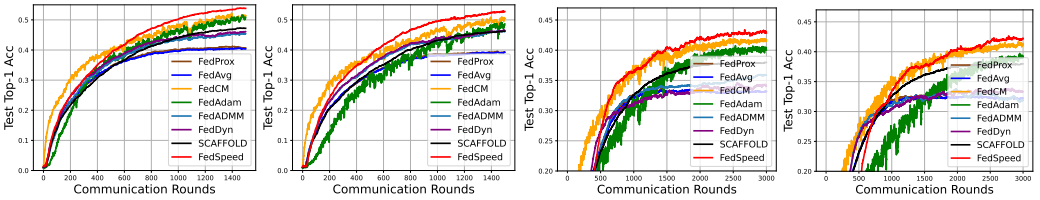
In this part, we will introduce our experimental setups, including dataset, hyper-parameters selection and implementation details firstly. We present the convergence and generalization performance in Section 5.2, and study the hyper-parameter sensitivity and ablation experiments in Section 5.3.

5.1 SETUP

Dataset and backbones. We test the experiments on CIFAR-10, CIFAR-100 Krizhevsky et al. (2009) and TinyImagenet. Due to the space limitations we introduce these datasets in the **Appendix**. We



(a) 10% participation of 100 clients on CIFAR-10. (b) 2% participation of 500 clients on CIFAR-10.



(c) 2% participation of 500 clients on CIFAR-100. (d) 2% participation of 500 clients on TinyImagenet.

Figure 1: The top-1 accuracy in communication rounds of all compared methods on CIFAR-10/100 and TinyImagenet. Communication rounds are set as 1500 for CIFAR-10/100, 3000 for TinyImagenet. In each group, the left shows the performance on IID dataset while the right shows the performance on the non-IID dataset, which are split by setting heterogeneity weight of the Dirichlet as 0.6.

follow the Hsu et al. (2019) to introduce the heterogeneity via splitting the total dataset by sampling the label ratios from the Dirichlet distribution. We train and test the performance on the standard ResNet-18 He et al. (2016) backbone with the 7×7 filter size in the first convolution layer with BN-layers replaced by GN Wu & He (2018); Hsieh et al. (2020) to avoid the invalid aggregation.

Implementation details. We select each hyper-parameters within the appropriate range and present the combinations under the best performance. To fairly compare these baseline methods, we fix the most hyper-parameters for all methods under the same setting. For the 10% participation of total 100 clients training, we set the local learning rate as 0.1 initially and set the global learning rate as 1.0 for all methods except for FedAdam which applies 0.1 on global server. The learning rate decay is set as multiplying 0.998 per communication round except for FedDyn, FedADMM and FedSpeed which apply 0.9995. Each active local client trains 5 epochs with batchsize 50. Weight decay is set as $1e$-3 for all methods. The weight for the prox-term in FedProx, FedDyn, FedADMM and FedSpeed is set as 0.1. For the 2% participation, the learning rate decay is adjusted to 0.9998 for FedDyn and FedSpeed. Each active client trains 2 epochs with batchsize 20. The weight for the prox-term is set as 0.001. The other hyper-parameters specific to each method will be introduced in the **Appendix**.

Baselines. We compare several classical and efficient methods with the proposed FedSpeed in our experiments, which focus on the local consistency and client-drifts, including FedAvg McMahan et al. (2017), FedAdam Reddi et al. (2020), SCAFFOLD Karimireddy et al. (2020), FedCM Xu et al. (2021), FedProx Sahu et al. (2018), FedDyn Durmus et al. (2021) and FedADMM Wang et al. (2022). FedAdam applies adaptive optimizer to improve the performance on the global updates. SCAFFOLD and FedCM utilize the global gradient estimation to correct the local updates. FedProx introduces the prox-term to alleviate the local inconsistency. FedDyn and FedADMM both employ the different variants of the primal-dual method to reduce the local inconsistency. Due to the limited space, more detailed description and discussions on these compared baselines are placed in the **Appendix**.

5.2 Experiments

CIFAR-10. Our proposed FedSpeed is robust to different participation cases. Figure 1 (a) shows the results of 10% participation of total 100 clients. For the IID splits, FedSpeed achieves 6.1% ahead of FedAvg as 88.5%. FedDyn suffers the instability when learning rate is small, which is the similar phenomenon as mentioned in Xu et al. (2021). When introducing the heterogeneity, FedAdam suffers from the increasing variance obviously with the accuracy dropping from 85.7% to 83.2%. Figure 1 (b) shows the impact from reducing the participation. FedAdam is lightly affected by this change while the performance degradation of SCAFFOLD is significant which drops from 85.3% to 80.1%.

Table 1: Test accuracy (%) on the CIFAR-10/100 and TinyImagenet under the 2% participation of 500 clients with IID and non-IID dataset. The heterogeneity is applied as Dirichlet-0.6 (**DIR.**).

Method	CIFAR-10		CIFAR-100		TinyImagenet	
	IID.	DIR.	IID.	DIR.	IID.	DIR.
FedAvg	77.01	75.21	40.68	39.33	33.58	32.71
FedProx	77.68	75.97	41.29	39.69	33.71	32.78
FedAdam	82.92	80.55	51.65	49.29	40.85	39.71
SCAFFOLD	80.11	77.71	47.38	46.33	38.03	37.54
FedCM	84.20	83.48	52.35	50.98	41.90	41.67
FedDyn	83.36	80.57	46.18	46.60	34.69	33.92
FedADMM	81.29	79.71	45.51	46.65	36.03	33.83
FedSpeed	**85.80**	**84.79**	**53.93**	**52.88**	**43.38**	**42.75**

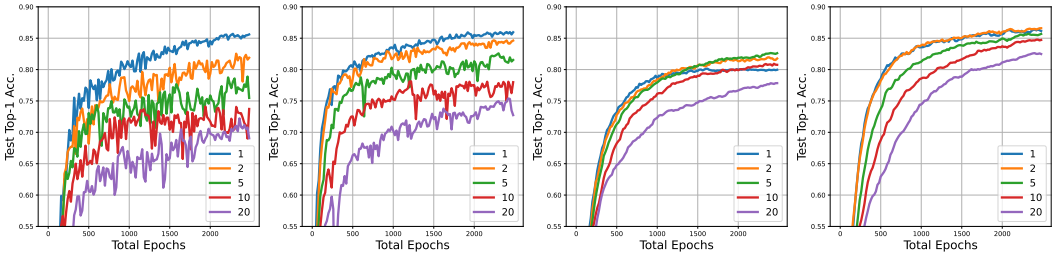
CIFAR-100 & **TinyImagenet.** As shown in Figure 1 (c) and (d), the performance of FedSpeed on the CIFAR-100 and TinyImagenet with low participating setting performs robustly and achieves approximately 1.6% and 1.8% improvement ahead of the FedCM respectively. As the participation is too low, the impact from the heterogeneous data becomes weak gradually with a similar test accuracy. SCAFFOLD is still greatly affected by a low participation ratio, which drops about 3.3% lower than FedAdam. FedCM converges fast at the beginning of the training stage due to the benefits from strong consistency limitations. FedSpeed adopts to update the prox-correction term and converges faster with its estimation within several rounds and then FedSpeed outperforms other methods.

Table 1 shows the accuracy under the low participation ratio equals to 2%. Our proposed FedSpeed outperforms on each dataset on both IID and non-IID settings. Table 1 shows the accuracy under the low participation ratio equals to 2%. Our proposed FedSpeed outperforms on each dataset on both IID and non-IID settings. We observe the similar results as mentioned in Reddi et al. (2020); Xu et al. (2021). FedAdam and FedCM could maintain the low consistency in the local training stage with a robust results to achieve better performance than others. While FedDyn is affected greatly by the number of training samples in the dataset, which is sensitive to the partial participation ratios.

Large local interval for the prox-term. From the IID case to the non-IID case, the heterogeneous dataset introduces the local inconsistency and leads to the severe client-drifts problem. Almost all the baselines suffer from the performance degradation. High local consistency usually supports for a large interval as for their bounded updates and limited offsets. Applying prox-term guarantees the local consistency, but it also has an negative impact on the local training towards the target of weighted local optimal and global server model. FedDyn and FedADMM succeed to apply the primal-dual method to alleviate this influence as they change the local objective function whose target is reformed by a dual variable. These method can mitigate the local offsets caused by the prox-term and they improve about 3% ahead of the FedProx on CIFAR-10. However, the primal-dual method requires a local ϵ-close solution. In the non-convex optimization it is difficult to determine the selection of local training interval K under this requirement. Though Durmus et al. (2021) claim that 5 local epochs are approximately enough for the ϵ-close solution, there is still an unpredictable local biases.

FedSpeed directly applies a prox-correction term to update K epochs and avoids the requirement for the precision of local solution. This ensures that the local optimization stage does not introduce the bias due to the error of the inexact solution. Moreover, the extra ascent step can efficiently improve the performance of local model parameters. Thus, the proposed FedSpeed can improve 3% than FedDyn and FedADMM and achieve the comparable performance as training on the IID dataset.

An interesting experimental phenomenon is that the performance of SCAFFOLD gradually degrades under the low participation ratio. It should be noticed that under the 10% participation case, SCAFFOLD performs as well as the FedCM. It benefits from applying a global gradient estimation to correct the local updates, which can weaken the client-drifts by a quasi gradient towards to the global optimal. Actually the estimation variance is related to the participation ratio, which means that their efficiencies rely on the enough number of clients. When the participation ratio decreases to be extremely low, their performance will also be greatly affected by the huge biases in the local training.



(a) FedAdam. (b) FedCM. (c) SCAFFOLD. (d) **FedSpeed.**

Figure 2: Performance of FedAdam, FedCM, SCAFFOLD and FedSpeed with local epochs $E = 1, 2, 5, 10, 20$ on the 10% participation case of total 100 clients on CIFAR-10. We fix $T \times E = 2500$ as the equaled total training epochs to illustrate the performance of increasing E and decreasing T.

5.3 Hyperparameters Sensitivity

Local interval K. To explore the acceleration on T by applying a large interval K, we fix the total training epochs E. It should be noted that K represents for the iteration and E represents for the epoch. A larger local interval can be applied to accelerate the convergence in many previous works theoretically, e.g. for SCAFFOLD and FedAdam, while empirical studies are usually unsatisfactory. As shown in Figure 2, in the FedAdam and FedCM, when K increases from 1 to 20, the accuracy drops about 13.7% and 10.6% respectively. SCAFFOLD is affected lightly while its performance is much lower. In Figure 2 (d), FedSpeed applies the larger E to accelerate the communication rounds T both on theoretical proofs and empirical results, which stabilizes to swing within 3.8% lightly.

Learning rate ρ for gradient perturbation. In the simple analysis, ρ can be selected as a proper value which has no impact on the convergence complexity. By noticing that if $\alpha \neq 0$, ρ could be selected irrelevant to η_l. To achieve a better performance, we apply the ascent learning rate $\rho = \rho_0/\|\nabla F_i\|$ to in the experiments, where ρ_0 is a constant value selected from the Table 2. ρ is consistent with the sharpness aware minimization Foret et al. (2020) which can search for a flat local minimal. Table 2 shows the performance of utilizing the different ρ_0 on CIFAR-10 by 500 communication rounds under the 10% participation of total 100 clients setting. Due to the space limitations more details could be referred to the **Appendix**.

Table 2: Performance of different ρ_0 with $\alpha = 1$.

ρ_0	0	0.01	0.05	**0.1**	0.2
Acc.	83.97	84.6	85.38	**85.72**	84.35

Perturbation weight α. α determines the degree of influence of the perturbation gradient term to the vanilla stochastic gradient on the local training stage. It is a trade-off to balance the ratio of the perturbation term. We select the α from 0 to 1 and find FedSpeed can converge with any $\alpha \in [0, 1]$. Though the theoretical analysis demonstrates that by applying a $\alpha > 0$ in the term Φ will not increasing the extra orders. And the experimental results shown in Table 3, indicates that the generalization performance improves by increasing α. We provide an explanation for α in the **Appendix**.

Table 3: Performance of different α with $\rho_0 = 0.1$.

α	0	0.5	0.75	0.875	**0.9375**	1.0
Acc.	83.97	84.36	84.91	85.46	**85.74**	85.72

6 Conclusion

In this paper, we propose a novel and practical federated method FedSpeed which applies a prox-correction term to neutralize the bias due to prox-term in each local training stage and utilizes a perturbation gradient weighted by an extra gradient ascent step to improve the local generalization performance. We provide the theoretical analysis to guarantee its convergence and prove that FedSpeed benefits from a larger local interval K to achieve a fast convergence rate of $\mathcal{O}(1/T)$ without any other harsh assumptions. We also conduct extensive experiments to highlight the significant improvement and efficiency of our proposed FedSpeed, which is consistent with the properties of our analysis. This work inspires the FL framework design to focus on the local consistency and local higher generalization performance to implement the high-efficient method to federated learning.

REFERENCES

Maksym Andriushchenko and Nicolas Flammarion. Understanding sharpness-aware minimization. 2021.

Muhammad Asad, Ahmed Moustafa, and Takayuki Ito. Fedopt: Towards communication efficiency and privacy preservation in federated learning. *Applied Sciences*, 10(8):2864, 2020.

Sebastian Bischoff, Stephan Günnemann, Martin Jaggi, and Sebastian U Stich. On second-order optimization methods for federated learning. *arXiv preprint arXiv:2109.02388*, 2021.

Zachary Charles and Jakub Konečnỳ. Convergence and accuracy trade-offs in federated learning and meta-learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2575–2583. PMLR, 2021.

Congliang Chen, Li Shen, Haozhi Huang, and Wei Liu. Quantized adam with error feedback. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(5):1–26, 2021.

Hong-You Chen and Wei-Lun Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. *arXiv preprint arXiv:2009.01974*, 2020.

Xiangyi Chen, Xiaoyun Li, and Ping Li. Toward communication efficient adaptive gradient method. In *Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference*, pp. 119–128, 2020.

Alp Emre Durmus, Zhao Yue, Matas Ramon, Mattina Matthew, Whatmough Paul, and Saligrama Venkatesh. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. *arXiv preprint arXiv:2203.11751*, 2022.

Yonghai Gong, Yichuan Li, and Nikolaos M Freris. Fedadmm: A robust federated deep learning framework with adaptivity to system heterogeneity. *arXiv preprint arXiv:2204.03529*, 2022.

Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

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*, pp. 770–778, 2016.

Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pp. 4387–4398. PMLR, 2020.

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.

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.

Belhal Karimi, Xiaoyun Li, and Ping Li. Fed-lamb: Layerwise and dimensionwise locally adaptive optimization algorithm. *CoRR*, abs/2110.00532, 2021.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. First analysis of local gd on heterogeneous data. *arXiv preprint arXiv:1909.04715*, 2019.

Geeho Kim, Jinkyu Kim, and Bohyung Han. Communication-efficient federated learning with acceleration of global momentum. *arXiv preprint arXiv:2201.03172*, 2022.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020a.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smithy. Feddane: A federated newton-type method. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 1227–1231. IEEE, 2019.

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, 2020b.

Ji Liu, Jizhou Huang, Yang Zhou, Xuhong Li, Shilei Ji, Haoyi Xiong, and Dejing Dou. From distributed machine learning to federated learning: A survey. *Knowledge and Information Systems*, pp. 1–33, 2022.

Grigory Malinovskiy, Dmitry Kovalev, Elnur Gasanov, Laurent Condat, and Peter Richtarik. From local sgd to local fixed-point methods for federated learning. In *International Conference on Machine Learning*, pp. 6692–6701. PMLR, 2020.

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

Emre Ozfatura, Kerem Ozfatura, and Deniz Gündüz. Fedadc: Accelerated federated learning with drift control. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 467–472. IEEE, 2021.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 3:3, 2018.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. Fedproto: Federated prototype learning across heterogeneous clients. In *AAAI Conference on Artificial Intelligence*, volume 1, 2022.

Dirk van der Hoeven. Exploiting the surrogate gap in online multiclass classification. *Advances in Neural Information Processing Systems*, 33:9562–9572, 2020.

Han Wang, Siddartha Marella, and James Anderson. Fedadmm: A federated primal-dual algorithm allowing partial participation. *arXiv preprint arXiv:2203.15104*, 2022.

Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. Slowmo: Improving communication-efficient distributed sgd with slow momentum. *arXiv preprint arXiv:1910.00643*, 2019.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020a.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020b.

Jianyu Wang, Zheng Xu, Zachary Garrett, Zachary Charles, Luyang Liu, and Gauri Joshi. Local adaptivity in federated learning: Convergence and consistency. *arXiv preprint arXiv:2106.02305*, 2021.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Jing Xu, Sen Wang, Liwei Wang, and Andrew Chi-Chih Yao. Fedcm: Federated learning with client-level momentum. *arXiv preprint arXiv:2106.10874*, 2021.

Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-iid federated learning. *arXiv preprint arXiv:2101.11203*, 2021.

Peng Yang, Yuning Jiang, Ting Wang, Yong Zhou, Yuanming Shi, and Colin N Jones. Over-the-air federated learning via second-order optimization. *arXiv preprint arXiv:2203.15488*, 2022.

Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3):1–207, 2019.

Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pp. 7184–7193. PMLR, 2019.

Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. Fedpd: A federated learning framework with adaptivity to non-iid data. *IEEE Transactions on Signal Processing*, 69:6055–6070, 2021.

Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. *arXiv preprint arXiv:2202.03599*, 2022.