An Efficient Federated Learning Method Enables Larger Local Intervals

YAN SUN

SID: 490605413

Supervisor: Dacheng Tao
Auxiliary Supervisor: Liu Liu

A thesis submitted in fulfilment of
the requirements for the degree of
Master of Philosophy

School of Computer Science
Faculty of Engineering
The University of Sydney
Australia

29 April 2023

Abstract

Federated learning is an emerging distributed machine learning framework that 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 thesis, we propose two novel and practical methods, FedSpeed and its variant FedSpeed-Ing, 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 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. Then, we introduce two inertial momenta on the global update as the FedSpeed-Ing method, which could further improve the optimization speed. Our theoretical analysis indicates that the convergence rate is related to both the communication rounds T and local intervals K with an 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 the 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.

Acknowledgements

I sincerely thank my supervisor Dacheng Tao and auxiliary supervisor Liu Liu. With the their help and guidance, I complete a series of theoretical explorations on federated learning field. Thanks a lot to my mentor Li Shen from the JD Explore Academy to give me enough guidance and inspiration with great patience in the researches, which encourages me to explore in the science.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Background

Since the introduction of deep neural networks in 2006, coupled with the considerable improvement in algorithms and computing power in recent years, and the massive amount of information data under the big data cloud model, artificial intelligence has ushered in a very golden period of improvement. For example, in alpha-go, the model used a total of 0.3 million game records as training data, successfully learned the characteristics and skills of Go, and then defeated the ranked No.1 human players in the world. We can see the great potential of artificial intelligence in the field of deep learning, and we are yearning for artificial intelligence technology to be applied to serve our lives.

The great success of alpha-go makes people naturally hope that this kind of big data-driven artificial intelligence can blossom and bear in all walks of daily life. But the current overall situation is not as optimistic as expected in our view. Except for a limited number of industries, such as computer vision, natural language process, and voice signal processing, there is a tough problem of poor data quality in other fields, e.g. for the radioactive medical data for the specific malignant disease and the structure and properties handled by some rare chemical elements. Data constraints the further development of deep learning. Therefore, most industries still cannot get the wide support of artificial intelligence. The data of most applications still require extensive manual annotation, and the fault tolerance is very high. Because the contents of the data samples are so difficult, empiricism often cannot guarantee the reliability of this kind of dataset. And people who can provide such data annotations are very unique, and there is even a high probability that they need to be the best in the industry

for decades. Including various topics of the medical team, most people can't serve this kind of difficult dataset. At the same time, the data format varies in some fields. A special learning theory is needed to deal with the problems that arise in the specific field.

On the other hand, with the further development of big data, emphasis on data privacy and security has become a worldwide trend. Every leak of public data will cause a huge or even catastrophic information collapse. At present, all walks of life are strengthening protection measures for data security. Studies have shown that data breaches usually do not occur on the client side of data usage. Because before using the window, the data has gone through a lot of unprotected interactions and communications. This unreliable path brings huge holes in the dataset. This has a major impact on the reliability of artificial intelligence. If data privacy cannot be guaranteed, it means that much private data cannot be applied with these technologies. Therefore, in the process of data exchange with the third-party, how to achieve privacy protection and deal with data loss and theft attacks has become a technology that must be considered in AI.

To solve the big data dilemma, the bottleneck has appeared only by traditional methods. Firstly, the user is the owner of the original data, and the data cannot be exchanged between companies without the user's approval. Secondly, the purpose of data modeling cannot be changed until the user agrees to it. Therefore, the data storage model should be a closed system, and the right of the interaction process should be entrusted to the user to execute. However, users cannot give instructions for use all the time, because the speed of data interaction is much higher than our imagination. Therefore, under this premise, the interaction with the data gradually transforms into the interaction with the upper model. If the two data can be mutually assisted, then the interaction between the model is established to improve the performance of the joint data, and its efficiency is higher than the transmission of the original data. At the same time, the model has a certain degree of concealment. In general, we cannot pass the model directly or the original data. It can also be simply understood that the model parameter dimension is generally much lower than the data parameter dimension, so the model is more similar to the embedding of a data feature rather than the data itself.

1.2 Federated Learning (FL)

1.2.1 Framework

How to design a machine learning framework under the premise of meeting data privacy, security, and regulatory requirements, so that artificial intelligence can work together with local private data to solve a public problem more efficiently and accurately is an important topic in the current development of artificial intelligence. To shift the focus of research to how to solve the problem of data islands, we explored a feasible solution that satisfies privacy protection and data security, that is, federated learning. It has the following characteristics:

- 1) Local private data are kept without sharing, and no privacy has been disclosed neither in-laws nor in regulations, which are always violated.

- 2) A system in which multiple participants jointly establish a virtual shared model and obtain benefits together, which is the single target for local clients.

- 3) Ultimate goals and training effects are aligned with distributed training, which means the separate dataset training and concentrated data training should be consistent in performance and efficiency.

- 4) Without mutual local data access rights, only information fusion is achieved at the model level or other privacy protection level.

As the underlying technology of AI development, federated learning will continue to promote innovation and leaps in global AI technology by relying on safe and reliable data protection measures.

With the increase of data privatization and localization requirements in practical applications, the implementation of Federated Learning (**FL**) has gradually become one of the most economic and efficient methods to protect local privacy without accessing root among the edge devices. It is a distributed approach for collaboratively training the global server model

with the decentralized dataset located on the local clients. The global server will handle the training process by communicating with the local nodes to optimize the joint ultimate target. FL uses the computing power of edge nodes to improve the overall training efficiency. In this setting, the communication efficiency, the data heterogeneity drifts, and the compatibility with privacy are the three main issues that limit the development of FL, especially in large-scale training frameworks. In addition, in real-world scenarios, people also should pay more attention to many systemic issues, requirements, and other constraints that are missed in theoretical analysis.

Since (McMahan et al. 2017b) 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. Generally, the size of a local client's cluster is hundreds or even thousands. As a distributed data crossing-silo framework, the most important concern is to reduce the communication costs in the training process, which is traced to communication efficiency. However, FL is far from its maturity, as it still suffers from considerable performance degradation over the heterogeneously distributed data, a very common setting in the practical application of FL.

1.2.2 Main Challenges

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 an inconsistent local solution. Then, the rugged client drifts resulting from the local over-fitting into inconsistent local solutions may make the obtained global model degrade into the average of the client's local parameters. The non-vanishing biases have been studied by several previous works (Charles and Konečny 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.

Here we further explain the terms *local inconsistency* and *local heterogeneous over-fitting*. *local inconsistency* measures the distance between the global model and local models. If the distance is too large, the performance of the average in FL will become very weak. It could be bounded by the variance of the local models to measure the level of inconsistency. *local heterogeneous over-fitting* measure the level of local training. It is usually highly related to the length of the local interval. If we select a large local interval, the local client may overfit on its own dataset which will dramatically affect the global performance.

Though adaptive optimizers on the FL system exhibit amazing performance and excellent potential in practical FL applications, there are still daunting challenges in training practical non-convex deep network scenarios. Our experiments indicate that the global adaptive optimizer harms convergence speed, which is much slower than SGD-based algorithms. Inaccurate gradient estimation from local clients' differences introduces a larger variance in the calculation of second-order momenta and leads to instability of the training process. Local adaptive optimizer that effectively improves the convergence speed suffers from the negative implication of significant over-fitting. A heterogeneous dataset yields huge gaps between aggregated local optimum and global optimum as the client drifts. Therefore, the local heterogeneous dataset causes the unsatisfactory performance of the global model. In federated deep model training, we empirically reveal the lower generalization problem caused by local over-fitting of directly applying the local adaptive optimizer compared to SGD-based algorithms. Furthermore, we introduce two inertial momenta on the global update as the FedSpeed-Ing method, which could further improve the optimization speed.

1.3 Contributions

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**, and an efficient variant **FedSpeed-Ing**. 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 and 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 by 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; 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 and Flammarion 2021) to prevent the local over-fitting problems.

We also provide the theoretical analysis of the 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 $O(1/T)$, 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 (Acar 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.

In the end, we summarize the main contributions as follows:

- We propose two novel and practical federated optimization algorithms, **FedSpeed** and **FedSpeed-Ing**, which apply 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. **FedSpeed** achieves a fast convergence with large local steps and simultaneously maintains high generalization performance. **FedSpeed-Ing** further uses two inertial momenta to accelerate the optimization speed and achieves higher training efficiency.

- 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 in the case of enlarging the local interval.

CHAPTER 2

Literature review

2.1 Federated Learning Framework

Since (McMahan et al. 2017a) firstly propose the federated framework – FedAvg algorithm, to further address the major challenges in the communication costs and the local heterogeneity on the dataset with the fully theoretical analysis (Yang et al. 2021a; Li et al. 2020d; Lin et al. 2020) on the proof of linear speedup property with the large scaled cluster. Several stochastic gradient descent based methods are proposed to implement the federated training process. (Li et al. 2020c) introduce the FedProx algorithm to tackle the local heterogeneity through adopting a proximal term to handle the difference in the local training. Inspired by the excellent effects of variance reduction techniques in stochastic optimization, (Karimireddy et al. 2020b) propose a VR-type method which applies the SVRG (Johnson and Zhang 2013) to alleviate the local client drifts. (Zhang et al. 2021) incorporate the Primal-Dual based method in the training to fix different levels of local heterogeneity and achieve the true global optimal. FedNova (Wang et al. 2020b) perform different local steps on asynchronous aggregation settings and averages the normalized local offset to merge the scaled parameters. (Acar et al. 2021) pay attention on the local consistency and propose the FedDyn method which forces the local objective optimal close to the global optimal. (Yu et al. 2019a) propose to improve the local efficiency of the parallel restarted momentum at the first iteration during each epoch. (Wang et al. 2020a) introduce the Slow-Mo method, which applies the global momentum update to yield smaller gaps between the optimization and generalization performance. (Xu et al. 2021; Ozfatura et al. 2021) both apply the averaged quasi global gradient as a client-level momentum term to approach better performance. Usually the general federated optimization

involves a local client training stage and a global server aggregation step (Asad et al. 2020) and it has been proved to achieve the linear speedup property in (Yang et al. 2021b) even under the partial participation case. With the fast development of the federated learning framework, a series of powerful methods are adopt in the both local and global nodes. Both (Li et al. 2020b) and (Kairouz et al. 2021) draw a detailed overview in this field. However, there are still many problems to be solved in the real-world scenarios and limitations in the federated learning system (Li et al. 2020a; Yang et al. 2019; Konečny et al. 2016; Liu et al. 2022).

2.2 Optimizer Development in FL

Adaptive Optimizer. Adaptive methods in FL greatly benefit from the adaptivity on the heterogeneous dataset. (Duchi et al. 2011; McMahan and Streeter 2010; Kingma and Ba 2015; Li and Orabona 2019; Wu et al. 2019) study several adaptive methods on non-FL settings. A lot of powerful variants are proposed including Adagrad (Duchi et al. 2011), Adadelta (Zeiler 2012), Adam (Kingma and Ba 2015), Amsgrad (Reddi et al. 2018) and Nadam, etc. To be adapted to different tasks. adaptive methods have achieved more excellent empirical performance than SGD. (Reddi et al. 2021) incorporate adaptive optimizer on the global server in FL framework to accelerate the convergence speed in deep network training. (Xie et al. 2019) apply the AdaAlter optimizer on the local clients with the lazily updated denominators. (Chen et al. 2020a) indicate the second-order momenta of local Amsgrad must be averaged to avoid divergence in training process. (Wang et al. 2021b) prove the inconsistency leads to non-vanishing gaps in a toy quadratic example and update the global model by averaging the inverse of the local pre-conditioner matrices. Compared with these works, our proposed method benefits from the fast convergence speed of local adaptive optimizer and takes advantage of local amended technique to mitigate over-fitting on non-iid dataset.

Regularized Term. An efficient way is to adopt the regularization term on the local training process to correct the local objective function, which can achieve the global optimal via a two-staged optimization. (Li et al. 2020c) employs a local proxy function in the training

process and averaged aggregation on the global server. (Pham et al. 2021) introduce the FedDR method with a Douglas-Rachford splitting way to alleviate the drift. (Zhang et al. 2021) use the primal-dual algorithm in the federated system, and (Acar et al. 2021) make a progress on the FedPD to improve a partial merged parameters method with the full merged dual variables in the global server, named FedDyn, which achieves the SOTA results in the regularization baselines. (Wang et al. 2022; Gong et al. 2022) use the alternating direction method of multipliers in the total training process as a extension of the federated primal-dual methods. (Fallah et al. 2020) put forward a personalized federated learning (pFL) framework with the regularization to achieve a better generalization performance. (T Dinh et al. 2020) incorporate the Moreau-Envelope technique in the local training with a stage-wised proxy function to update. (Huang et al. 2021) propose an adaptive parameters for the regularization term to encourage the local devices to aggregate more within the similar neighbourhoods. The efficient regularization-based methods are very important and efficient in the FL field, which allows the faster convergence rate than SGD-based methods.

Momentum-based Term. Inspired by the success of the global correction technique, the exponential moving average term is introduced to federated learning framework to correct the local training. (Liu et al. 2020) adopts the momentum-SGD to the local clients to improve the generalization performance with a convergence analysis. (Wang et al. 2020a) proposes a global momentum method to further improve the stability in the server side. (Xu et al. 2021) incorporate the global offset to the local client as a client-level momentum to correct the heterogeneous drifts. (Ozfatura et al. 2021) combine the global and local momentum update and propose the FedADC algorithm to avoid the local over-fitting. (Reddi et al. 2021) sets a global ADAM optimizer with the momentum update and propose the adaptive federated optimizer in the FL. (Wang et al. 2021a) corrects the pre-conditioner in the global server. Though momentum terms are the biased estimation of global information, they still contribute a lot to the federated frameworks in practical empirical experiments.

VR-based Term. Motivated by the success of VR-techniques in the stochastic local training, several methods are proposed to reduce the heterogeneous inconsistency, which efficiently avoid the local client-drift theoretically. (Karimireddy et al. 2020b) use the SVRG control

variants to correct the heterogeneous offset in the local updates. (Karimireddy et al. 2020a) implement a combination of the local controller and global correction optimizer in each communication round to ensure the local model mimics a centralized or distributed method. (Mitra et al. 2021) propose a global gradient correction term in the local training steps which exploits much of memory in the practical applications. (Murata and Suzuki 2021) incorporate the small second-order heterogeneity of local objectives and suggests randomly picking up one of the local models instead of taking the average of them when clients are synchronized, which can improve the efficiency and reduce the communication costs. (Zhao et al. 2021a) applies the vanilla local SGD update with a little correction with a sampling probability defined manually. (Zhao et al. 2021b) addresses the compression by proposing the compressed VR methods with a error feedback variant.

2.3 Main Challenges in Optimization

Local consistency. (Li et al. 2020c) 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 and Konečny 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. 2021a; Wang et al. 2020c) 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. 2020b; 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, (Acar 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 $O(\frac{1}{T})$ 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 and 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. 2020b) 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. 2019b) and (Wang et al. 2020a) 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. 2020b) and (Chen et al. 2021) focus on the adaptive method to alleviate the biases and improve the efficiency.

CHAPTER 3

Methods

In this section, we will introduce the proposed methods in details. We firstly define the problem setups and preliminaries in the next part. In section 3.2, we will introduce the baseline of FedAvg framework. In section 3.3, we will introduce the design of our proposed FedSpeed and its variant FedSpeed-Ing method in details, including the motivation, the insights and the improvement of different modules designed. We will reveal the connection with the previous proposed method and analyze their difference compared with the baselines. In section 3.4, we discuss the insights of the proposed FedSpeed and its improvement in details. In section 3.5, we provide the theoretical analysis of the proposed FedSpeed, and more details can be referred to the Appendix.

3.1 Problem Setups and Preliminary

We consider the most common and fundamental non-convex minimization problem which is widely used in the practical applications. We consider that the non-convex objective F is a finite-sum problem with several local objective function F_i as the follows:

$$F(\mathbf{x}) \coloneqq \mathbb{E}_{i\sim\mathcal{P}}[F_i(\mathbf{x})], \tag{3.1}$$

$$F_i(\mathbf{x}) \coloneqq \mathbb{E}_{\xi\sim\mathcal{D}_i}[F_i(x,\xi)] \tag{3.2}$$

where $x \in \mathbb{R}^d$ represents for the global parameters, $F : \mathbb{R}^d \to \mathbb{R}$ denotes the global objective function consisting of a bunch of sub-objective function F_i, i denotes the index of the client devices participating in the optimization and $\mathcal{P}$ denotes a distribution on the population of clients set, $F_i : \mathbb{R}^d \to \mathbb{R}$ denotes the sub-objective function which can be applied as a local

loss function F_i and they are often the same across all the clients, ξ denotes the training dataset with the specific distribution $\mathcal{D}_i$, usually in FL problems we consider the data heterogeneity.

Due to the physical limitation in the federated framework, the designed algorithms can not directly compute $F(\mathbf{x})$ or $\nabla F(\mathbf{x})$ on one local client without the full communication connection to the other nodes. It set a leader/central client to exchange the information to the other clients by all-accessing network. This paradigm build a huge community through the central node, which possesses strong performance and convergence guarantee. However, by the centralized processing of large amounts of data, the central node will naturally be tired with greater pressure when calculating or communicating. To overcome this, compressing the transferring information and reducing the communication rounds is one of the important target in FL, which greatly reduces the data pressure of a single node and provides a more flexible and efficient solution.

In practical application, for higher efficiency the clients are usually accessed in a random sample $\mathcal{S}$ in each communication round. The objective function $F(\mathbf{x})$ can be analyzed as a mathematical object and such algorithms are even numerically calculated as part of empirical evaluation in simulations procedures. If the total dataset are fixed in the whole optimization, to approximate the population risk in (3.1), the selection of $\mathcal{S}$ in each round generates errors. On the condition of blind view to the local dataset, local training gaps between clients in $\mathcal{S}$ are unpredictable. Usually if we ignore the potential error, we often use a weighted sum function to approximate. In this "cross-silo" setting, the objective function can take the form of the empirical risk minimization(ERM) with finite clients and each client has finite local data,

$$F(\mathbf{x}) \coloneqq \sum_{i=1}^{M} p_i F_i(\mathbf{x}) \tag{3.3}$$

$$F_i(\mathbf{x}) \coloneqq \frac{1}{|\mathbf{D}_i|} \sum_{\xi \in \mathbf{D_i}} F_i(x, \xi) \tag{3.4}$$

where M is the number of clients and p_i is the weight of client i. Usually we consider the weight as $p_i = |\mathbf{D}_i| / \sum_{i=1}^{M} |\mathbf{D}_i|$, where $\mathbf{D}_i$ denotes the local dataset in client i and $|\mathbf{D}_i|$ denotes the number of the local dataset. It is equal to Equation (3.1) in expectation with a randomly sampled clients set with the union of the dataset of the selected devices.

3.2 FedAvg Method

Algorithm 1 FedAvg Algorithm

Input: Initial $\mathbf{x}_0$, local learning rate η_t, global learning rate η_g, round T, local iteration K
Output: $\mathbf{x}_T$

for t $\in \{0, 1, 2, \cdots, T-1\}$ **do**
  for client i in $\mathcal{S}^{(t)}$ parallel **do**
    communicate $\mathbf{x}_t$ to client i and set $\mathbf{x}_{i,0}^t = \mathbf{x}^t$
    for $k \in \{0, 1, 2, \cdots, K-1\}$ **do**
      compute local stochastic gradient $\mathbf{g}_{i,k}^t$
      $\mathbf{x}_{i,k+1}^t = \mathbf{x}_{i,k}^t - \eta_t \mathbf{g}_{i,k}^t$
    end for
    communicate $\Delta_i^t = \mathbf{x}_{i,K}^t - \mathbf{x}_{i,0}^t$ to the server
  end for
  $\Delta^t = \frac{1}{K\eta_t} \sum_{i \in \mathcal{S}^{(t)}} p_i \Delta_i^t$
  $\mathbf{x}^{t+1} = \mathbf{x}^t - \eta_g \Delta^t$
end for

Based on local-SGD algorithms, the basic method in FL problems is directly to apply the local-SGD with partial participation strategy — *FedAvg*(McMahan et al. 2017b). It is a rough estimate baseline which divides the training into two stages, local training and aggregation. At the first stage in round t, it randomly select the active devices set $\mathcal{S}^{(t)}$, the central server broadcasts the current global model parameters $\mathbf{x}^{(t)}$ to users in $\mathcal{S}^{(t)}$. Each active user sets $\mathbf{x}_{i,0}^t = \mathbf{x}^{(t)}$ as initial point for training. After k local SGD updates, local user sends the $\mathbf{x}_{i,k}^t$ back to the server. The second stage is to aggregate the parameters. According to the definition in Equation (3.3), ignoring the local training gaps, the server can aggregated the parameters as:

$$\mathbf{x}^{t+1} = \frac{\sum_{i \in \mathcal{S}^{(t)}} p_i \mathbf{x}_{i,k}^t}{\sum_{i \in \mathcal{S}^{(t)}} p_i} \tag{3.5}$$

Another description on some recent works will consider the two stage as two separate optimizers — Client-Opt and Server-Opt. They transfer the model parameters as initial point to each other, and finish the training stage alternately. It should be clear that there are no dataset on the server node. Server training often use the minus mean average of the local change as a virtual gradient (some called pseudo-gradient) : $\mathbf{g} := -\sum_{i \in \mathcal{S}^{(t)}} p_i \Delta_i^t / k\eta_t$, where $\Delta_i^t = \mathbf{x}_{i,k}^t - \mathbf{x}_{i,0}^t$. Thus, they are equivalent when setting $\eta_g = K\eta_t$.

The main researches focus on the problems mentioned above. Usually the topic includes "Fast, More precise and Minimum cost". FL could be applied to solve a lot of cooperative learning problems. In practical applications, we need to make reasonable assumptions for different settings. Due to the privacy limitation we always do not expect that the local client maintain high frequency of interactions. Sometimes it is also thought of as a resources limitations. It is hard for a huge system to ask all the clients to participate in training stage. At the same time, we expect less history information when aggregating. Historical redundant information does not effectively help server for training, even sometimes it causes gradient confusion. It is difficult to guarantee convergence when models in different training stages are aggregating. This will also cause time series problems. In practice, for example, there are physical differences between machines and some clients are powerful calculating but some are "tiny" machine. Some works also call it a asynchronous updating. Thus, when we try to optimize a big model with a large number of local users and their larger local dataset, it is core to reduce the costs as possible as we can. It can be thought of a trade-off in accuracy and training costs.

Communication Efficiency To reduce costs in FL training, practical FL focus on how to reduce the communication. The communication cost usually involves the frequency, the matrix information. After total T iterations, if the local stage is K, we have the relation of round $r = T/K$. A measure of the communication cost of an algorithm could be defined as $V * r$, where V is the total numbers of the parameters. The key to reduce costs is to enlarge K and compress V as possible as it can.

Increasing the local interval K, which means a longer local training stage will be applied in local clients. This introduces error to optimal. Even some extremely strict works indicate that one-step FL will converge in some specific settings. However, facing to the most tasks, it is nearly impossible to approach the global optimal through one local training stage. Some recent studies show that there is an additional noise term in the optimizing process. Although it can be ignored when using a decay schedule on step size. It is usually a constant item controlled by η. While the performance will be improved, it can make the analysis on convergence much more difficult. Considering the heterogeneity of the dataset, many works indicate that the

error usually erupts and sometimes it could be a constant error which can not be ameliorated without some new techniques. Thus, it is a trade-off between error and communication costs and convergence rate. When using a larger k, r could be reduced and error will increase as growing. In practical, best local intervals could be thought of as a optional parameters.

Another method is to enlarge the batchsize. Many works has focused on the researches whether local or mini-batch algorithms contribute more for FL optimizations(Woodworth et al. 2020b)(Woodworth et al. 2020a)(Stich 2018). The current experimental results cannot give an absolutely affirmative answer. Although we will not strictly discuss the relationship between these in the training processes, they do have a certain similarity.

3.3 FedSpeed and FedSpeed-Ing Methods

3.3.1 FedSpeed

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 (Acar et al. 2021) for the local updates to eliminate the client drifts when T approaches infinite.

Our proposed FedSpeed is shown in Algorithm 2. 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 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 $\check{\mathbf{x}}_{i,k}^t$; (2) computing the unbiased stochastic gradient $\mathbf{g}_{i,k,2}^t$ with the same sampled mini-batch data in (1) at the $\check{\mathbf{x}}_{i,k}^t$ 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.

Algorithm 2 FedSpeed Algorithm

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 $\mathbf{x}^T$.

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

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 term. In vanilla Fedavg and some other SGD-based algorithms, on the local client we perform many steps of local update at each round. The theoretical analysis shows the clear relationship of the longer local steps are, the greater the impact of heterogeneity is. While more local iterations will reduce the communication costs, it is still a trade-off between performance and efficiency. It always incur some extra errors at each aggregation stages compared with the vanilla SGD. At the same, more local steps will force the local client converge to the local optimal of the local objectives, which is always far away from the global optimal. In order to avoid the client drifts, a naturally idea is to penalize the local models that are not far away from the global models parameters by a regularization item between local and global parameters. The prox term $\|\mathbf{x}_{i,k}^t - \mathbf{x}^t\|^2$ is designed to reduce the local client drifts caused by the different local optimal in the FedProx, which is a federated

optimization algorithm that addresses the challenges of heterogeneity dataset both theoretically and empirically. The key inspiration in developing FedProx is that an interplay exists between systems and statistical heterogeneity in federated learning. It approximates the global optimal by approximate a γ-inexact solution of the prox function $\hat{F}_i(\mathbf{x})$. For a function $\hat{F}_i(\mathbf{x})$, it is said that $\hat{\mathbf{x}}^*$ is a γ-inexact solution if $||\nabla \hat{F}_i(\hat{\mathbf{x}}^*, \mathbf{x}_0)|| \leq \gamma ||\nabla F_i(\mathbf{x}_0, \mathbf{x}_0)||$ for the proxy. It can be used for measuring the amount of local computation of each local iterations at each round. Thus, the federated learning problems are divided into different sub-problems in each client. Some early works indicate that it is important to apply a changeable γ with the number of iterations and local conditions. In the optimization theory, the γ will vanish to zero if the $\hat{F}$ satisfy some conditions. The proximal point optimization will bring some new features if we select a good weight λ. It shares a connection with the averaged SGD, the method to train the deep networks in the data center setting and use a similar proximal term in its objective. Also it needs the bounded dissimilarity assumption in the theoretical analysis. It makes local updates not too far away from the initial global model, and reduce the impact of non-iid while tolerating system heterogeneity. At the same time, the inexact solution is defined, through the inaccurate solution of the local function, the number of local iterations is dynamically adjusted to ensure the accuracy of heterogeneous systems.

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 (Li et al. 2020c). However, as discussed in (Hanzely and 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. From the optimization perspective of view, the prox function do approach the optimal after a very long training process with a requirement of exact solution of the local functions. However, in the practical applications, we can not finish the enough guarantees on this condition. With a few updates, the local clients will approach a mixed solution between local and global optimal which will play a negative impact on the convergence.

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 2 (Line.11), for $\forall\, \mathbf{x} \in \mathbb{R}^d$ we have:

$$\mathbf{x}_{i,K}^t - \mathbf{x}^t = -\gamma\lambda \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t + \gamma\lambda \hat{\mathbf{g}}_i^{t-1}, \tag{3.6}$$

where $\sum_{k=0}^{K-1} \gamma_k = \sum_{k=0}^{K-1} \frac{\eta_l}{\lambda}\big(1 - \frac{\eta_l}{\lambda}\big)^{K-1-k} = \gamma$.

Firstly let $\hat{\mathbf{g}}_i^{-1} = \mathbf{0}$, Equation (3.6) 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 2) and the equation (3.6), we can rewrite this update as:

$$\hat{\mathbf{g}}_i^t = (1 - \gamma)\hat{\mathbf{g}}_i^{t-1} + \gamma\Big(\sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t\Big), \tag{3.7}$$

where γ and γ_k is defined the same as in Equation (3.6).

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. 2020a) 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.

We propose the gradient perturbation in the local training stage instead of the traditional stochastic gradient, which merges an extra gradient ascent step to the vanilla gradient by a hyper-parameter α. While its ascent step usually approximates the worst point in the neighbourhood. This has been studied in many previous works, e.g. for the form of extra gradient and the sharpness aware minimization. In our studies, we perform the extra gradient ascent step instead of the descent step in extra gradient method. It also could be considered as a variant of the sharpness aware minimization method via weighted averaging the ascent step gradient and the vanilla gradient, instead of the normalized gradient. Here we illustrate the implicit of this quasi-gradient $\tilde{\mathbf{g}}$ in our proposed FedSpeed and explain the positive efficiency for the local training from the perspective of objective functions.

Firstly we consider to minimize the non-convex problem $L_p(\mathbf{x})$. To approach the stationary point of L_p, we can simply introduce a penalized gradient term as a extra loss in L_p, which is to solve the problem $\min_{\mathbf{x}}\{L(\mathbf{x}) \triangleq L_p(\mathbf{x}) + \frac{\beta}{2}\|\nabla L_p(\mathbf{x})\|^2\}$. The final optimization target is consistent with the vanilla target, while penalizing gradient term can approach a flatten minimal empirically. We compute the gradient form as follows:

$$\nabla L(\mathbf{x}) = \nabla L_p(\mathbf{x}) + \frac{\beta}{2}\nabla\|\nabla L_p(\mathbf{x})\|^2 = \nabla L_p(\mathbf{x}) + \beta\nabla^2 L_p(\mathbf{x}) \cdot \nabla L_p(\mathbf{x}). \tag{3.8}$$

The update in Equation (3.8) contains second-order Hessian information, which involves a huge amount of parameters for calculation. To further simplify the updates, we consider an

approximation for the gradient form. We expand the function L_p via Taylor expansion as:

$$L_p(\mathbf{x}+\Delta) = L_p(\mathbf{x}) + \nabla L_p(\mathbf{x})\Delta + \frac{1}{2}\Delta^T\nabla^2 L_p(\mathbf{x})\Delta + \mathcal{R}_\Delta, \tag{3.9}$$

where $\mathcal{R}_\Delta = O(\|\Delta\|^2)$ is the infinitesimal to $\|\Delta\|^2$, which is directly omitted in our approximation.

Thus we have the gradient form on Δ as:

$$\nabla L_p(\mathbf{x}+\Delta) \approx \nabla L_p(\mathbf{x}) + \nabla^2 L_p(\mathbf{x})\Delta. \tag{3.10}$$

$\mathcal{R}_\Delta$ is relevant to Δ. We set the $\Delta = \rho\nabla L_p(\mathbf{x})$ and then we have:

$$\nabla^2 L_p(\mathbf{x})\nabla L_p(\mathbf{x}) \approx \frac{1}{\rho}\big(\nabla L_p\big(\mathbf{x}+\rho\nabla L_p(\mathbf{x})\big) - \nabla L_p(\mathbf{x})\big). \tag{3.11}$$

Thus we connect Equation (3.8) and Equation (3.11), we have:

$$\begin{aligned}
\nabla L(\mathbf{x}) &= \nabla L_p(\mathbf{x}) + \beta\nabla^2 L_p(\mathbf{x})\cdot\nabla L_p(\mathbf{x}) \\
&\approx \nabla L_p(\mathbf{x}) + \frac{\beta}{\rho}\big(\nabla L_p\big(\mathbf{x}+\rho\nabla L_p(\mathbf{x})\big) - \nabla L_p(\mathbf{x})\big) \\
&= \big(1-\frac{\beta}{\rho}\big)\nabla L_p(\mathbf{x}) + \frac{\beta}{\rho}\nabla L_p\big(\mathbf{x}+\rho\nabla L_p(\mathbf{x})\big) \\
&= (1-\alpha)\nabla L_p(\mathbf{x}) + \alpha\nabla L_p\big(\mathbf{x}+\rho\nabla L_p(\mathbf{x})\big).
\end{aligned}$$

Here we can see that the balance weight α in our proposed method is actually the ratio of the gradient penalized weight β and the gradient ascent step size ρ. To fix the step size ρ, increasing α means increasing the gradient penalized weight β, which facilitates searching for a flatten stationary point to improve the generalization performance. While the second term of $\nabla L(\mathbf{x})$ can not be directly computed for its nested form, we approximate the second term with the chain rule as follows:

$$\nabla L_p\big(\mathbf{x}+\rho\nabla L_p(\mathbf{x})\big) \approx \nabla L_p(\theta)|_{\theta=\mathbf{x}+\rho\nabla L_p(\mathbf{x})}.$$

Finally we have:

$$\nabla L(\mathbf{x}) \approx (1-\alpha)\nabla L_p(\mathbf{x}) + \alpha\nabla L_p(\theta)|_{\theta=\mathbf{x}+\rho\nabla L_p(\mathbf{x})}. \tag{3.12}$$

The Equation (3.12) provides an understanding for the weighted quasi gradient $\tilde{\mathbf{g}}$ on the local training stage in our proposed FedSpeed. We select an appropriate $0 \leq \beta \leq \rho$ to satisfy the update of perturbation gradient. It executes a gradient ascent step firstly with the step size ρ to $\check{\mathbf{x}}$. Then it generates the stochastic gradient by the same sampled mini-batch data as the ascent step at $\check{\mathbf{x}}$. The quasi-gradient is merged as Equation (3.12) to execute the gradient descent step.

This is just a simple approximation for the gradient perturbation to help for understanding the implicit of the quasi-gradient and its performance in the training stage. Actually the error of the approximation depends a lot on ρ. The smaller ρ, the higher the accuracy of this estimation, but the smaller ρ, the less efficient the optimizer performs.

3.3.2 FedSpeed-Ing Variant

Here we introduce a variant of FedSpeed, which enjoys the both prox-term and prox-correction term and further employs the inertial momentum on the global state and correction term as dual variable in the primal problem.

We highlight the improvements as the blue part, which are two **In**ertial-**g**radient terms on the global state and the prox-correction term. We call it FedSpeed-Ing method. The same as FedSpeed, it adopts a prox term and prox-correction term to perform the local update, which means it can inherent the faster and better performance. The difference is, i) in the global server when it update the global model $\mathbf{x}^t$, it firstly aggregate the corrected local model as the same as FedSpeed, then it performs a step of inertial momentum at the current state with one-step historical state. This is similar with the look ahead method, which try to predict the next global state and set the prediction state as the proxy target. ii) in the local client, we also do a prediction on the prox-correction term to match the prox-term.

Combining the two stages of training, we can see how it works. First, local client subtracts the historic gradient of the previous round $t-1$ and in the server node, the average of all local gradients is used as the global gradient to re-compensate to the global model parameters during the aggregation. Obviously, this process is very similar to the application of variance

Algorithm 3 FedSpeed-Ing Variant Algorithm

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 $\mathbf{x}^T$.

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

reduction techniques such as SCAFFOLD and FedDANE. The key problem is how large the gap between the global gradient estimate $\mathbf{h}$ and the global gradient will be. At the same time, we also need to think about the impact of this separation form on local updates. When the local gradient correction term is introduced, we consider the final state of convergence. When the local most convergent point is approached, we usually consider : $\mathbf{x_t} = \mathbf{x}_{t-1}$, which means that at the optimal we have $\nabla F_i(\mathbf{x}_t) = \nabla F_i(\mathbf{x}_{t-1})$. FedAvg does not have such an equation relationship. In traditional algorithms, this is the condition of the local optimal point. We have discussed the relationship between the local optimal and the global optimal before. In non-iid dataset, there is usually no deterministic connection between them. Moreover, generally speaking, when global optimal is achieved in FedAvg, there is often no such relationship between local gradients. However, the local training of FedDyn can offset this part of the error very well. At any convergence point, local training can achieve the above conditions, or in other words, only when the above conditions are met, the global model will converge to

a stable optimal point. Therefore, we can understand that all local training will eventually achieve the same solution condition as the global model, in another words, the same stable optimal. This is a very good property. This shows that our local results are consistent.

As long as the algorithm can be guaranteed to converge normally, then in the local optimization process, an approximate estimation of the gradient can be achieved, which is:

$$\frac{1}{M}\sum_{i\in\mathbf{M}}\nabla F_i(\mathbf{x}_t) \approx \frac{1}{M}\sum_{i\in\mathbf{M}}\nabla F_i(\mathbf{x}_{i,k}^t) \tag{3.13}$$

where local optimal is close to the global optimal gradually. Therefore, in the later of the optimization process, we can consider the FedDyn as a algorithm similar to using variance reduction techniques. The estimation of $\mathbf{h}$ finally converges on the true global gradients. In the global aggregation, the correction term can be regarded as a compensation. This also illustrates the cleverness of the dynamic regularization. Without the participation of all customers, at final convergence, the global optimal can achieve convergence rate similar to SCAFFOLD. In the theoretical convergence rate of FedDyn, the convergence rate of $O(\frac{1}{T})$ can be achieved, which also illustrates the effect of the algorithm. In the experiments, FedDyn can maintain training with a larger learning rate, which means that its convergence speed will be faster in general. In general, it implements the constraint training of dynamic regularization terms, and provides an application of variance reduction techniques for prox-based algorithms. Moreover, it does not need to transmit the estimation of the global gradient back to each clients, which reduce the pressure of communication efficiency. This is actually a very important progress, an acceleration method to achieve zero extra communication. To balance communication efficiency and convergence, it is a very good application and powerful algorithm.

The optimization of the regularization is actually a trade-off. At the same time there is a more intuitive advantage — it can bring additional mathematical properties to the original objective function, such as smoothness or convexity, etc. Generally speaking, mathematical properties often lead to better results and more universal application algorithms. Therefore, this type of algorithm is very representative in solving federated learning problems. In this part, we only introduced three recent works, which does not mean that the other works are unimportant. Many solid researches have been published in the view of the optimization methods. These

three tasks represent the main development of this type of algorithm in recent years. Next we will introduce another technology.

3.4 Understanding of FedSpeed

To express the essential insights in the updates of the Algorithm 2, we introduce two auxiliary sequences. From the different perspective of views, we can clearly demonstrate the performance of each modules and their inner-connections in the total optimization.

Firstly 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 at the last iteration, we have the total update offset as:

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

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. From the perspective of federated local update, we can construct the update of $\mathbf{u}_{i,k}^t$ as

$$\mathbf{u}_{i,k+1}^t = \mathbf{u}_{i,k}^t - \frac{\lambda\gamma_k}{\gamma}(\gamma\tilde{\mathbf{g}}_{i,k}^t + (1-\gamma)\hat{\mathbf{g}}_i^{t-1}). \tag{3.14}$$

Here, we see that the Equation (3.14) indicates the variable $\mathbf{u}$ updates as a client-level momentum method with a quasi exponential decay learning rate, whose initial value is λ. γ is a constant when K is fixed, which plays a role as the coefficient of the linear combinations.

The term $\hat{\mathbf{g}}$ is the local averaged offset, which matches the client-level momentum in the FedCM method (Xu et al. 2021). It should be noticed that the stochastic gradient is calculated at the raw $\mathbf{x}$, and the virtual local update only reveals the insights of the FedSpeed. In the original FedCM method, the local learning rate η_l is a fixed value in the total local training process and decayed per round. Here we can find their difference: FedSpeed adopts a decayed local learning rate γ_k at k-th iteration. This provides the different importance for the local update, which handles the local momentum term method.

Based on the definition of $\{\mathbf{u}^t\}$, we introduce the second auxiliary sequences $\{\mathbf{z}^t = \mathbf{u}^t + \frac{1-\gamma}{\gamma}(\mathbf{u}^t - \mathbf{u}^{t-1})\}_{t>0}$. According to the analysis above, we also introduce K virtual states $\{\mathbf{z}_{i,k}^t\}$. This is a very common transformation which is widely used in the local momentum methods. It could be considered as a inner-momentum on the raw states, which is a "look-ahead" step to the current state. From this point, we can further indicate the local update as a vanilla FedAvg method which can be easy to understand. 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.

we expand the the auxiliary sequence $\mathbf{z}^t$ as:

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

$$
\begin{aligned}
&= -\lambda \frac{1}{m} \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t - \frac{1-\gamma}{\gamma} \frac{1}{m} \sum_{i\in[m]} (\mathbf{u}^t - \mathbf{u}^{t-1} + \lambda \hat{\mathbf{g}}_i^{t-1}) \\
&= -\lambda \frac{1}{m} \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t - \frac{1-\gamma}{\gamma} \frac{1}{m} \sum_{i\in[m]} (\mathbf{x}_{i,K}^{t-1} - \mathbf{x}_{i,K}^{t-2} + \lambda \hat{\mathbf{g}}_i^{t-1}) \\
&= -\lambda \frac{1}{m} \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t - \frac{1-\gamma}{\gamma} \frac{1}{m} \sum_{i\in[m]} (\mathbf{x}_{i,K}^{t-1} - \mathbf{x}_{i,0}^{t-1} + \lambda \hat{\mathbf{g}}_i^{t-1} - \lambda \hat{\mathbf{g}}_i^{t-2}) \\
&= -\lambda \frac{1}{m} \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t.
\end{aligned}
$$

The same as $\mathbf{u}$, we can construct the local update of $\mathbf{z}_{i,k}^t$ as:

$$
\mathbf{z}_{i,k+1}^t = \mathbf{z}_{i,k}^t - \frac{\lambda \gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t. \tag{3.15}
$$

This also indicates the advantage of the proposed FedSpeed, which shows the higher efficiency with a decayed momentum-based local learning rate.

3.5 Convergence Analysis

In this section, we will provide the theoretical analysis on the FedSpeed method. In the section 3.5.1, we show some usual assumptions on the function F to prepare for the proof, which are common assumptions in the federated and stochastic cases. In the section 3.5.2, we will introduce some important lemmas and theoretical conclusions we prove in this part. In the section 3.5.3, we will provide the detailed theorem on the convergence of FedSpeed, including some important discussions on the hyper-parameters and selections of the fixed weights. More detailed proofs can be referred to the Appendix.

3.5.1 Assumptions

In this section, we will introduce the assumptions we use in our proof. We will discuss the problems more generally through the analysis of the heterogeneity and local updates to

present the current progress in this field and many problems we still face to. Here are some basic assumptions and preliminaries.

- 1) In each round t, each active client will take a local training with k steps on local private dataset. Usually the parameters such as learning rate η_t and k is a given constant or determined variable.

- 2) It must be aggregated on the server client or some group of leader, even each client in decentralized setting.

- 3) There are finite clients to participate in the training process. Therefore, we consider the problem definition as $\frac{1}{M}\sum_{i\in[M]} F_i(x)$.

- 4) It will be selected a part of clients to be active for training, which is to reduce the communicate efficiency. This is called partial participation and we denote the active set as $\mathcal{S}^{(t)}$.

- 5) Special properties of functions F. Usually the deep neural network is considered as a universal fitter to fit a specific function. We often solve temporarily simplified problems, step by step. Function properties usually include L-smooth, convex and strong convex, etc.

- 6) Gradient bound. This prat contains three main types of assumptions. Firstly, usually we need a bounded gradient, which is:

$$||\nabla F(\mathbf{x})||^2 \leq G^2. \tag{3.16}$$

  This is a conventional assumption. Usually we will assume a initial point in the convex optimization process. The initial point usually retains a relatively large gradient value and the optimal point holds zero. In non-convex optimization, we usually use L-smooth assumptions to ensure that this.

Secondly, the stochastic gradients bound is needed in analysis. We often have the assumptions as:

$$\mathbb{E}[\mathbf{g}_i(\mathbf{x}_{i,k}^t)|\mathbf{x}_{i,k}^t] = \nabla F(\mathbf{x}_{i,k}^t), \tag{3.17}$$

$$\mathbb{E}[||\mathbf{g}_i(\mathbf{x}_{i,k}^t) - \nabla F(\mathbf{x}_{i,k}^t)||^2|\mathbf{x}_{i,k}^t] \leq \sigma^2. \tag{3.18}$$

The purpose of this assumption is to ensure the stability of the stochastic gradient. This is also one of the very common assumptions in many other optimization methods.

Thirdly, we often call it dissimilarity of each clients. In the FL problem, this assumption is necessary. Assuming that the local data is completely differentiated, in fact, we cannot guarantee that local training is reasonable. In the data space, we even need to assume that the distance between each local optimal is so large that their local convergence rate will have nothing to do with global convergence. Therefore, we need to use assumptions to ensure that each local dataset has a certain similarity but not exactly the same. The most common form of hypothesis is as follows :

$$\frac{1}{M}\sum_{i\in[M]} ||\nabla F_i(\mathbf{x}) - \nabla f(\mathbf{x})||^2 \leq \sigma_g^2. \tag{3.19}$$

This directly bounds the gradient difference between global and local loss function, which is widely used to measure the heterogeneity of the local dataset in FL problems. There is also another assumption form to measure this distance:

$$\frac{1}{M}\sum_{i\in[M]} ||\nabla F_i(\mathbf{x})||^2 \leq \mathbf{G}^2 + B^2||\nabla F(\mathbf{x})||^2. \tag{3.20}$$

This is mentioned in SCAFFOLD which give a new analysis framework with the new assumption. Moreover, a supplementary assumption is also given in the original text:

$$||\nabla^2 F_i(\mathbf{x}) - \nabla^2 F(\mathbf{x})||^2 \leq \delta. \tag{3.21}$$

At present, these kinds of gradient bounded assumptions are more common in the theoretical analysis. Some works use bounded L1 norm of the gradients, which is the

same as L2 norm essentially. In the federated learning problems, we have multiple local iterates on clients and we want to ensure all of them are approaching the global optimal. To handle iterates from multiple clients, the upper bound of the gradient is very important.

In this thesis, we main select the following as the based assumptions:

ASSUMPTION 3.5.1. *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 3.5.2. *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 3.5.3. *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 3.5.1 guarantees a Lipschitz continuity and Assumption 3.5.2 guarantees the stochastic gradient is bounded by zero mean and constant variance. Assumption 3.5.3 is the heterogeneity bound for the non-iid dataset, which is widely used in many previous works (Reddi et al. 2021; Yang et al. 2021b; Xu et al. 2021; Wang et al. 2021a). Our theoretical analysis depends on the above assumptions to explore the comprehensive properties in the local training process.

3.5.2 Important Lemmas

In this part, we will introduce some important lemmas firstly, which are the main technique contributions in this paper and some basic proofs of the final convergence analysis.

LEMMA 1. *For* $\forall\, \mathbf{x}_{i,k}^t \in \mathbb{R}^d$ *and* $i \in \mathcal{S}^t$, *we denote* $\delta_{i,k}^t = \mathbf{x}_{i,k}^t - \mathbf{x}_{i,k-1}^t$ *with setting* $\delta_{i,0}^t = 0$, *and* $\Delta_{i,K}^t = \sum_{k=0}^K \delta_{i,k}^t = \mathbf{x}_{i,K}^t - \mathbf{x}_{i,0}^t$, *under the update rule in Algorithm 2, we have:*

$$\Delta_{i,K}^t = -\lambda\gamma \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t + \gamma\lambda \hat{\mathbf{g}}_i^{t-1}, \tag{3.22}$$

where $\sum_{k=0}^{K-1} \gamma_k = \sum_{k=0}^{K-1} \frac{\eta_l}{\lambda}\big(1 - \frac{\eta_l}{\lambda}\big)^{K-1-k} = \gamma = 1 - (1 - \frac{\eta_l}{\lambda})^K$.

PROOF. According to the update rule of Line.11 in Algorithm Algorithm 1, we have:

$$\begin{aligned}
\delta_k &= \Delta_{i,k}^t - \Delta_{i,k-1}^t = \mathbf{x}_{i,k}^t - \mathbf{x}_{i,k-1}^t \\
&= -\eta_l\big(\tilde{\mathbf{g}}_{i,k-1}^t - \hat{\mathbf{g}}_i^{t-1} + \frac{1}{\lambda}(\mathbf{x}_{i,k-1}^t - \mathbf{x}_{i,0}^t)\big) = -\eta_l(\tilde{\mathbf{g}}_{i,k-1}^t - \hat{\mathbf{g}}_i^{t-1} + \frac{1}{\lambda}\Delta_{i,k-1}^t).
\end{aligned}$$

Then We can formulate the iterative relationship of $\Delta_{i,k}^t$ as:

$$\Delta_{i,k}^t = \Delta_{i,k-1}^t - \eta_l(\tilde{\mathbf{g}}_{i,k-1}^t - \hat{\mathbf{g}}_i^{t-1} + \frac{1}{\lambda}\Delta_{i,k-1}^t) = (1 - \frac{\eta_l}{\lambda})\Delta_{i,k-1}^t - \eta_l(\tilde{\mathbf{g}}_{i,k-1}^t - \hat{\mathbf{g}}_i^{t-1}).$$

Taking the iteration on k and we have:

$$\begin{aligned}
\mathbf{x}_{i,K}^t - \mathbf{x}_{i,0}^t = \Delta_{i,K}^t &= (1 - \frac{\eta_l}{\lambda})^K \Delta_{i,0}^t - \eta_l \sum_{k=0}^{K-1} (1 - \frac{\eta_l}{\lambda})^{K-1-k}(\tilde{\mathbf{g}}_{i,k}^t - \hat{\mathbf{g}}_i^{t-1}) \\
&\overset{(a)}{=} -\eta_l \sum_{k=0}^{K-1} (1 - \frac{\eta_l}{\lambda})^{K-1-k}(\tilde{\mathbf{g}}_{i,k}^t - \hat{\mathbf{g}}_i^{t-1}) \\
&= -\lambda \sum_{k=0}^{K-1} \frac{\eta_l}{\lambda}(1 - \frac{\eta_l}{\lambda})^{K-1-k}(\tilde{\mathbf{g}}_{i,k}^t - \hat{\mathbf{g}}_i^{t-1}) \\
&= -\lambda \sum_{k=0}^{K-1} \frac{\eta_l}{\lambda}(1 - \frac{\eta_l}{\lambda})^{K-1-k}\tilde{\mathbf{g}}_{i,k}^t + \big(1 - (1 - \frac{\eta_l}{\lambda})^K\big)\lambda\hat{\mathbf{g}}_i^{t-1} \\
&= -\lambda\gamma \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t + \gamma\lambda\hat{\mathbf{g}}_i^{t-1}.
\end{aligned}$$

(a) applies $\Delta_{i,0}^t = \delta_{i,0}^t = 0$.

□

This lemma indicate the very important properties in the prox-based method, which is, the local update could be transferred to a SGD-type update with a decayed local learning rate γ_k, which focuses on the importance of the related gradient information. If we ignore the prox-correction term, FedSpeed performs as the momentum-based method with a exponential moving averaged aggregation.

LEMMA 2. *Under the update rule in Algorithm Algorithm 1, we have:*

$$\hat{\mathbf{g}}_i^t = (1-\gamma)\hat{\mathbf{g}}_i^{t-1} + \gamma \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t. \tag{3.23}$$

where $\sum_{k=0}^{K-1} \gamma_k = \sum_{k=0}^{K-1} \frac{\eta_l}{\lambda}\big(1-\frac{\eta_l}{\lambda}\big)^{K-1-k} = \gamma = 1-(1-\frac{\eta_l}{\lambda})^K$.

PROOF. According to the update rule of Line.13 in Algorithm Algorithm 1, we have:

$$\begin{aligned}
\hat{\mathbf{g}}_i^t &= \hat{\mathbf{g}}_i^{t-1} - \frac{1}{\lambda}(\mathbf{x}_{i,K}^t - \mathbf{x}_{i,0}^t) \\
&\overset{(a)}{=} \hat{\mathbf{g}}_i^{t-1} + \frac{\eta_l}{\lambda} \sum_{k=0}^{K-1} \big(1-\frac{\eta_l}{\lambda}\big)^{K-1-k} (\tilde{\mathbf{g}}_{i,k}^t - \hat{\mathbf{g}}_i^{t-1}) \\
&= \hat{\mathbf{g}}_i^{t-1} + \frac{\eta_l}{\lambda} \sum_{k=0}^{K-1} \big(1-\frac{\eta_l}{\lambda}\big)^{K-1-k} \tilde{\mathbf{g}}_{i,k}^t - \frac{\eta_l}{\lambda}\Big(\sum_{k=0}^{K-1} \big(1-\frac{\eta_l}{\lambda}\big)^{K-1-k}\Big)\hat{\mathbf{g}}_i^{t-1} \\
&= \hat{\mathbf{g}}_i^{t-1} + \frac{\eta_l}{\lambda} \sum_{k=0}^{K-1} \big(1-\frac{\eta_l}{\lambda}\big)^{K-1-k} \tilde{\mathbf{g}}_{i,k}^t - \frac{\eta_l}{\lambda} \frac{1-(1-\frac{\eta_l}{\lambda})^K}{\frac{\eta_l}{\lambda}} \hat{\mathbf{g}}_i^{t-1} \\
&= (1-\frac{\eta_l}{\lambda})^K \hat{\mathbf{g}}_i^{t-1} + \frac{\eta_l}{\lambda} \sum_{k=0}^{K-1} \big(1-\frac{\eta_l}{\lambda}\big)^{K-1-k} \tilde{\mathbf{g}}_{i,k}^t \\
&= (1-\gamma)\hat{\mathbf{g}}_i^{t-1} + \gamma \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t.
\end{aligned}$$

(a) applies Lemma 1. □

This shows the update rule of the correction term via the combination of the local quasi-gradient $\tilde{\mathbf{g}}$. The correction term is updated as a momentum term with the coefficient γ ($\gamma < 1$). The second term is the averaged local quasi-gradient weighted by γ, which implicates the

local offset as the update information. Here, the correction term is to adjust the local update towards to the global optimal.

3.5.3 Convergence Analysis

Based on the assumptions above, we prove that the convergence of the non-convex smooth function F holds:

THEOREM 3.5.4. *Under the Assumptions 3.5.1-3.5.3, when the perturbation learning rate satisfies $\rho \leq \frac{1}{\sqrt{6}\alpha L}$, and the local learning rate satisfies $\eta_l \leq \min\{\frac{1}{32\sqrt{3K}L}, 2\lambda\}$, and the local interval satisfies $K \geq \lambda/\eta_l$, let $\kappa = \frac{1}{2} - 3\alpha^2L^2\rho^2 - 1536\eta_l^2L^2K$ is a positive constant with selecting the proper η_l and ρ, the auxiliary sequence $\mathbf{z}^t$ generated by executing the Algorithm 2 satisfies:*

$$\frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}\|\nabla F(\mathbf{z}^t)\|^2 \leq \frac{2(F(\mathbf{z}^1)-F^*)}{\lambda\kappa T} + \frac{64\eta_l L^2K}{\kappa m T}\sum_{i\in[m]}\mathbb{E}\|\hat{\mathbf{g}}_i^0\|^2 + \frac{32\lambda^2L^2}{\kappa T}\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^0\|^2 + \Phi, \tag{3.24}$$

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

$$\Phi = \frac{1}{\kappa}\left(32\lambda\eta_l^2L^2K(16\sigma_g^2+\sigma_l^2) + \lambda\alpha^2L^2\rho^2(3\sigma_g^2+\sigma_l^2)\right), \tag{3.25}$$

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

COROLLARY 3.5.5. *Let $\rho = O(1/\sqrt{T})$ with the upper bound of $\rho \leq 1/\sqrt{6}\alpha L$, and let $\eta_l = O(1/K)$ with the lower bound of $\eta_l \geq \lambda/K$, when the local interval K is long enough with $K = O(T)$, the proposed FedSpeed achieves a fast convergence rate of $O(1/T)$.*

REMARK 3.5.6. *Compared with the other prox-based works, e.g. for (Acar 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 $O(1/T)$.*

TABLE 3.1: Convergence rate for non-convex smooth cases in some baselines and the proposed FedSpeed.

Method	Convergence	Assumption
FedAvg	$\mathcal{O}\left(\frac{1}{\sqrt{mKT}}+\frac{mK}{T}\right)$	-
FedAdam	$\mathcal{O}\left(\frac{1}{\sqrt{mKT}}+\frac{1}{KT}+\frac{\sqrt{m}}{\sqrt{KT^3}}\right)$	specific η_g
SCAFFOLD	$\mathcal{O}\left(\frac{1}{\sqrt{mKT}}+\frac{1}{T}\right)$	specific η_g
FedCM	$\mathcal{O}\left(\sqrt{\frac{1}{mT}+\frac{1}{mKT}}+\sqrt[3]{\frac{1}{T^2}+\frac{1}{mT^2}+\frac{1}{mKT^2}}\right)$	specific η_g
FedProx	$\mathcal{O}(\frac{1}{T})$	local exact solution
FedPD	$\mathcal{O}\left(\frac{1}{T}+\epsilon\right)$	local ϵ-stationarity[2]
FedDyn	$\mathcal{O}\left(\frac{1}{T}\right)$	local exact solution
FedADMM	$\mathcal{O}\left(\frac{1}{T}+\frac{1}{mT}\sum_i\sum_t\epsilon_{i,t}^2\right)$	local $\epsilon_{i,t}$-close solution[3]
FedSpeed	$\mathcal{O}\left(\frac{1}{T}+\frac{1}{K}\right)$	**-**

[1] m : the number of clients, K : the local interval, T : the communication round.
[2] solve the local sub-problem $F_i(\mathbf{x})$ to satisfy $\|\nabla F_i(\mathbf{x}^t)\|^2 \leq \epsilon$.
[3] solve the local sub-problem $F_i(\mathbf{x})$ to satisfy $\|F_i(\mathbf{x}^t) - F_i^*\| \leq \epsilon_{i,t}$.

REMARK 3.5.7. *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. 2021b), 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. 2020b) contribute the convergence as* $\mathcal{O}(1/\sqrt{SKT})$ *under the constant local interval, and (Reddi et al. 2021) 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 4.3. 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.*

We 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 achieves the fast convergence rate of $O(\frac{1}{T})$ when local interval setting $K = O(T)$. Generally, the dominate term of the convergence rate achieves $\max\{O(\frac{1}{T}), O(\frac{1}{K})\}$. We summarize the convergence rate of FedSpeed and some baselines in the Table 3.1, which indicates that FedSpeed achieves a tighter upper bound on local interval K to converge without applying a specific global learning rate or assuming the precision for the local solutions.

CHAPTER 4

Results

In this part, we will introduce our experimental setups, including dataset, hyper-parameters selection and implementation details firstly. In section 4.1, we introduce the basic setups including the dataset, deep model, hyper-parameters selections, heterogeneity introduction, and other empirical settings. We present the convergence and generalization performance in Section 4.2, and study the hyper-parameter sensitivity and ablation experiments in Section 4.3.

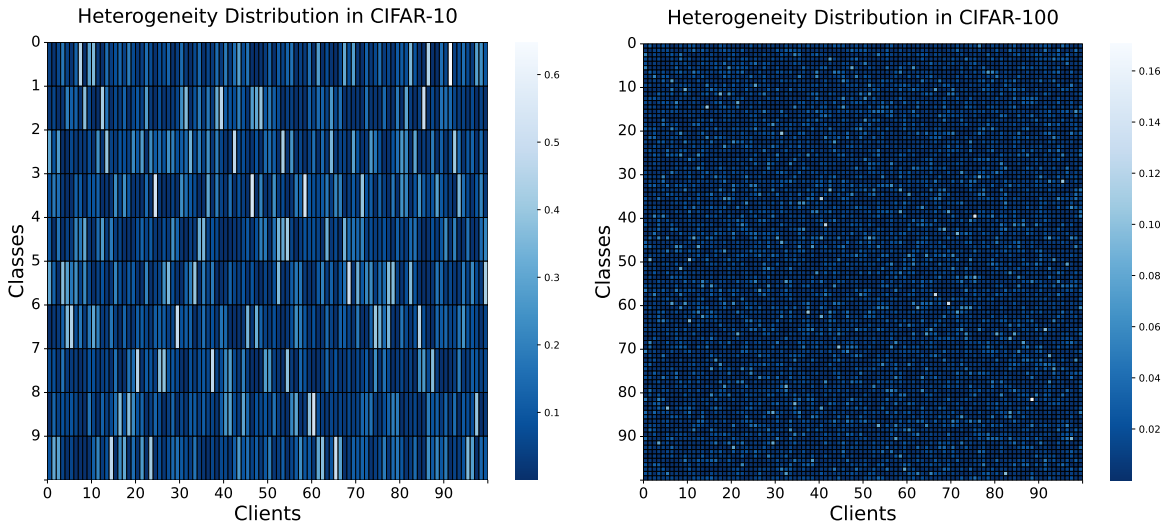
4.1 Setups

4.1.1 Dataset and backbones.

We test the experiments on CIFAR-10, CIFAR-100 (Krizhevsky, Hinton et al. 2009) and TinyImagenet. We 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 (Hsieh et al. 2020) to avoid the invalid aggregation.

TABLE 4.1: Dataset introductions.

Dataset	Training Data	Test Data	Class	Size
CIFAR-10	50,000	10,000	10	3×32×32
CIFAR-100	50,000	10,000	100	3×32×32
TinyImagenet	100,000	10,000	200	3×64×64



(a) CIFAR-10. (b) CIFAR-100.

FIGURE 4.1: Heat maps for different dataset under heterogeneity weight equals to 0.6 for Dirichlet distribution.

Dataset and Backbones. Extensive experiments are tested on CIFAR-10/100 dataset. We test on the two different settings as 10% participation of total 100 clients and 2% participation of total 500 clients. CIFAR-10 dataset contains 50,000 training data and 10,000 test data in 10 classes. Each data sample is a 3×32×32 color image. CIFAR-100 (Krizhevsky, Hinton et al. 2009) includes 50,000 training data and 10,000 test data in 100 classes as 500 training samples per class. TinyImagenet involves 100,000 training images and 10,000 test images in 200 classes for 3×64×64 color images, as shown in Table 4.1. To fairly compare with the other baselines, 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 as implemented in the previous works, e.g. for (Karimireddy et al. 2020b; Acar et al. 2021; Xu et al. 2021). We follow the (Hsieh et al. 2020) to replace the batch normalization layer with group normalization layer, which can be aggregated directly by averaging. These are all common setups in many previous works.

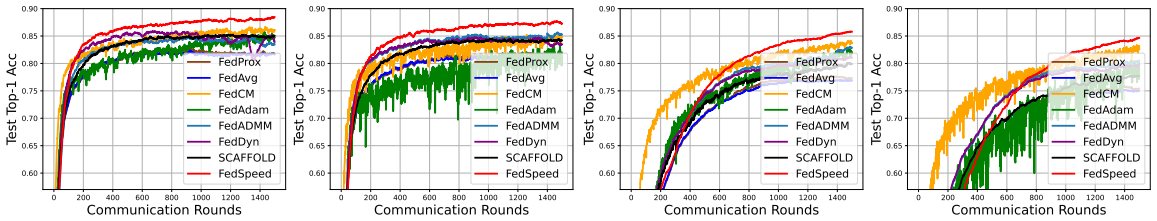
Dataset Partitions. To fairly compare with the other baselines, we follow the (Hsu et al. 2019) to introduce the heterogeneity via splitting the total dataset by sampling the label ratios from the Dirichlet distribution. An additional parameter is used to control the level

of the heterogeneity of the entire data partition. In order to visualize the distribution of heterogeneous data, we make the heat maps of the label distribution in different dataset, as shown in Figure 4.1. Since the heat map of 500 clients cannot be displayed normally, we show 100 clients case. It could be seen that for heterogeneity weight equals to 0.6, about 10% to 20% of the categories dominate on each client, which is white block in the Figure 4.1. The IID dataset is totally averaged in each client.

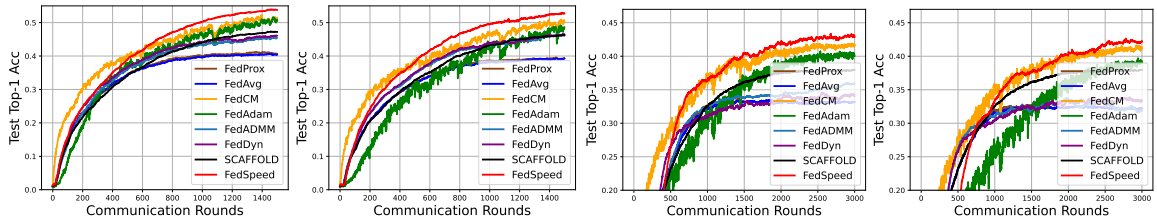
Implementation details. We select each hyper-parameters within the appropriate range and present the combinations under the best performance. To fairly compare these baseline methods, we fix the most hyper-parameters for all methods under the same setting. For the 10% participation of total 100 clients 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.

4.2 Experiments

CIFAR-10. Our proposed FedSpeed is robust to different participation cases. Figure 4.2 (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 4.2 (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%.



(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 4.2: 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.

CIFAR-100 and TinyImagenet. As shown in Figure 4.2 (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 4.2 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 4.2 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. 2021; Xu et al. 2021). FedAdam and FedCM could maintain the low

TABLE 4.2: 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**

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 (Acar et al. 2021) claim that 5 local epochs are approximately enough for the ϵ-close solution, there is still an unpredictable local biases.

TABLE 4.3: Training wall-clock time.

	Times (s/Round)	Rounds	Total (s)	Cost Ratio
FedAvg	10.44	-	-	-
FedProx	11.33	-	-	-
FedAdam	14.74	1343	19795.8	4.31×
SCAFFOLD	14.34	654	9378.3	2.03×
FedCM	13.22	622	8222.8	1.78×
FedDyn	14.11	400	5644.0	1.22×
FedSpeed	16.42	281	4614.0	1×
FedSpeed-Ing	16.48	266	4383.7	0.95×

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.

Training Speed.

Table 4.4 shows the communication rounds required to achieve the target test accuracy. At the beginning of training, FedCM performs faster than others and usually achieve a high accuracy finally. FedSpeed is faster in the middle and late stages of training. We bold the data for the top-2 in each test and generally FedCM and FedSpeed significantly performs well on the training speed.

TABLE 4.4: Communication rounds required to achieve the target accuracy. On CIFAR-10/100 it trains 1,500 rounds and on TinyImagenet it trains 3,000 rounds. "-" means the test accuracy can not achieve the target accuracy within the fixed training rounds. **DIR** represents for the Dirichlet distribution with the heterogeneity weight equal to 0.6. Local interval K is set as 5 on CIFAR-10 (100-10%) and 2 on others. Other hyper-parameters are introduced above.

Dataset	CIFAR-10 (100-10%)				CIFAR-10 (500-2%)			
Heterogeneity	IID.		DIR.		IID.		DIR.	
Target Acc. (%)	80.0	85.0	80.0	85.0	75.0	82.5	75.0	82.5
FedAvg	344	-	472	-	772	-	1357	-
FedProx	338	-	465	-	720	-	1151	-
FedAdam	324	1343	689	-	613	1476	878	-
SCAFFOLD	207	654	272	-	628	-	967	-
FedCM	**109**	620	192	1092	**325**	**1160**	**449**	**1399**
FedDyn	**121**	**400**	166	-	547	-	673	-
FedADMM	169	917	**174**	**756**	505	1440	687	-
FedSpeed	136	**280**	**169**	**380**	**495**	**926**	**662**	**1148**

Dataset	CIFAR-100 (500-2%)				TinyImagenet (500-2%)			
Heterogeneity	IID.		DIR.		IID.		DIR.	
Target Acc.	40.0	50.0	40.0	50.0	33.0	40.0	33.0	40.0
FedAvg	1013	-	-	-	1615	-	-	-
FedProx	957	-	-	-	1588	-	-	-
FedAdam	614	1277	847	-	1151	2495	1584	-
SCAFFOLD	720	-	784	-	949	-	1187	-
FedCM	**505**	**1150**	**526**	**1336**	**661**	**1360**	**817**	**1843**
FedDyn	661	-	703	-	1419	-	2559	-
FedADMM	687	-	715	-	921	-	2711	-
FedSpeed	**522**	**973**	**541**	**1038**	**684**	**1373**	**962**	**1885**

We test the time on the A100-SXM4-40GB GPU and show the performance in the Table 4.2. Experimental setups are the same as the CIFAR-10 10% participation among total 100 clients

TABLE 4.5: Comparison on different heterogeneous dataset.

α_1	IID	Dir-0.6	Dir-0.3	Drops (i.i.d. > Dir-0.6)	Drops (Dir-0.6 > Dir-0.3)
FedAvg	77.01	75.21	71.96	1.80	3.25
FedAdam	82.92	80.55	76.87	2.37	3.68
SCAFFOLD	80.11	77.71	74.34	2.40	3.37
FedCM	84.20	83.48	81.02	0.72	2.46
FedDyn	83.36	80.57	77.33	2.79	3.24
FedSpeed	85.80	84.79	82.68	1.01	2.11

on the DIR-0.6 dataset. The rounds in the table are the communication rounds required that the test accuracy achieves accuracy 85%. "-" means it can not achieve the target accuracy.

FedSpeed is slower due to the requirement of computing an extra gradient. So it gets slower in one single update, approximately 1.57× wall-clock time costs than FedAvg. But its convergence process is very fast. For the final convergence speed, FedSpeed still has a considerable advantage over other algorithms. The issue is possibly one of the improvements for FedSpeed in the future. For example, introduces a single-call gradient method to save half the costs during backpropagation. We are also currently trying to introduce new module to save the cost.

4.3 Ablation Study

Different heterogeneity. We test on the Dir-0.3 setups on CIFAR-10 and show the results as Table 4.3, the other settings are the same as the test in the text. The (i.i.d. > Dir-0.6) is the difference between the IID dataset and the Dir-0.6 dataset and (Dir-0.6 > Dir-0.3) is the difference between the Dir-0.6 dataset and the DIR-0.3 dataset. FedSpeed can outperform the others on the Dir-0.3 setups whose heterogeneity is much stronger than Dir-0.6 setups. the heterogeneity becomes stronger, FedSpeed can still maintain a stable generalization performance. The correction term helps to correct the biases during the local training, while the gradient perturbation term helps to resist the local over-fitting on the heterogeneous dataset. FedSpeed can benefit from avoiding falling into the biased optima.

TABLE 4.6: Comparison of different modules.

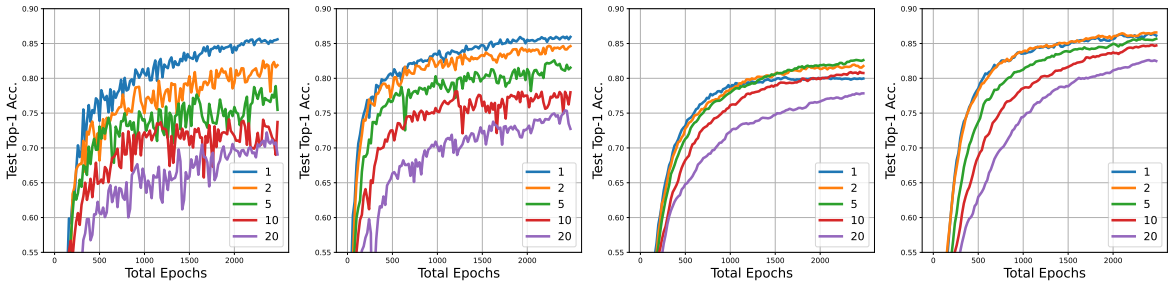
Prox-term	Prox-correction term	Gradient perturbation	Accuracy (%)
-	-	-	81.92
√	-	-	82.24
√	√	-	83.94
√	-	√	83.88
√	√	√	85.70

Improvements of Different Modules. From the practical training point of view, compared with the vanilla FedAvg, FedSpeed adds three main modules: (1) prox-term, (2) prox-correction term, and (3) gradient perturbation. We test the performance of 500 communication rounds of the different combination of the modules above on the CIFAR-10 with the settings of 10% participating ratio of total 100 clients. The Table4.3 shows their performance.

From the table above, we can clearly see the performance of different modules. The prox-term is proposed by the FedProx. But due to some issues we point out in this thesis, this term has also a negative impact on the performance in FL. When the prox-correction term is introduced in, it improves the performance from 82.24% to 83.94%. When the gradient perturbation is introduced in, it improves the performance from 82.24% to 83.88%. While FedSpeed applies them together and achieves a 3.46% improvement.

Different performance of these modules:

As introduced in this thesis, the prox-term 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. Thus we utilize the prox-correction term to correct the inconsistent biases during the local training. About the function of gradient perturbation, we refer to a theoretical explanation in the main text, and its proof is provided in the supplementary material due to the space limitations. This perturbation is similar to utilize a penalized gradient term to the objective function during local optimization process. The additional penalty will bring better properties to the local state, e.g. for flattened minimal and smoothness. For federated learning, the smoother the local minima is, the more flatness the



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

FIGURE 4.3: 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.

model merged on the server will be. FedSpeed benefits from these two modules to improve the performance and achieves the SOTA results.

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 4.3, 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 4.3 (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 4.7. ρ is consistent with the sharpness aware minimization (Foret et al. 2020) which can search for a flat local minimal. Table 4.7 shows

TABLE 4.7: 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

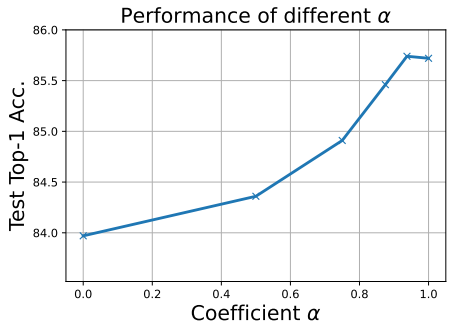


FIGURE 4.4: Performance of different α.

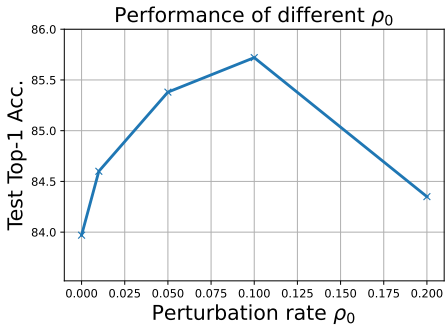


FIGURE 4.5: Performance of different ρ_0.

the performance of utilizing the different ρ_0 on CIFAR-10 by 500 communication rounds under the 10% participation of total 100 clients setting.

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 increase the extra orders. And the experimental results shown in Table 4.8, indicate that the generalization performance improves by increasing α.

TABLE 4.8: 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

CHAPTER 5

Conclusion

In this thesis, 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 $O(1/T)$ 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.

Summary and Future

In recent years, the distribution of clients-silo and the strengthening of the supervision of data privacy are becoming important challenges in the next stage of artificial intelligence. The emergence of federated learning breaks the data barrier and further provides development ideas for artificial intelligence. It enables multiple data owners to jointly establish a common model under the premise of protecting local data, thereby achieving mutual benefit under the premise of protecting privacy and data security. This chapter briefly introduces the basic concepts, architecture and technical principles of federated learning. At the same time, it introduces many technological development routes and development directions from a theoretical perspective. It is expected that in the future, federated learning can break down data barriers in various fields and industries and provide stronger information benefits for different participants while protecting privacy and data security.

In the future, federated learning face many challenges. Its safety is still one of our main concerns. For those who have administrator access to the client device, malicious attacks can be carried out by controlling the client. Maliciously manipulated clients can check all communication information during their participation in the iteration process, and even tamper with the exchange data to achieve the purpose of destruction. Clients that remain neutral will indirectly lead to training failure due to attacks. Malicious attacks on the server side are more serious. A maliciously manipulated server will disrupt interactive communications and cause the entire network to be paralyzed. At the same time, in the process of model output and deployment, it may also be subject to malicious attacks. How to ensure privacy in this situation is a great challenge.

The value of federated learning lies in breaking the data silos. By encouraging nodes with the same data structure or different to participate in the training together and improve the overall effect of the training. The related technologies and developments introduced in this chapter are also important researches in the future. The most important of these is the study of non-iid dataset. In practical scenarios, data inconsistencies are widespread. Since the virtual training process only occurs in the local stage, data processing and data enhancement in the traditional sense are biased. The exploration of data heterogeneity still requires a lot of theoretical researches. In addition, techniques such as fine-tuning, transfer learning, and meta learning are also being continuously introduced into federated learning to explore how to solve the impact of non-iid dataset.

Another concern is the aggregation method. In federated learning, in addition to dealing with parameters similar to traditional deep learning or traditional machine learning (such as learning rate, batch size, regularization, etc.), the aggregation rules need to be considered. Especially in practice when many assumptions cannot be met, how to achieve reasonable model aggregation has become an important research topic. Many model search algorithms are also difficult to implement due to the independence settings of federated learning, such as AutoML and NAS.

Limited bandwidth communication and equipment unreliability is also a research direction. The content of this part cannot be regarded as a bottleneck in the practical application now.

Because the existing large-scale equipment and frameworks are usually physically connected (such as large server clusters, etc.), communication delays are usually not observed. However, in the future, federated learning will eventually become widely used, so the connection between different devices and the stability of the device must be considered as one of the most important conditions. It is not clear about the application of low-efficiency equipment, but the scene of local training of different performance devices has appeared widely. Many compression and quantization techniques have been used in federated learning algorithms, which has been proved by practical experiments that the wall-clock time can be effectively reduced while training.

Bibliography

Acar, Durmus Alp Emre et al. (2021). 'Federated Learning Based on Dynamic Regularization'. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=B7v4QMR6Z9w.

Andriushchenko, Maksym and Nicolas Flammarion (2021). 'Understanding Sharpness-Aware Minimization'. In.

Asad, Muhammad, Ahmed Moustafa and Takayuki Ito (2020). 'FedOpt: Towards communication efficiency and privacy preservation in federated learning'. In: *Applied Sciences* 10.8, p. 2864.

Bischoff, Sebastian et al. (2021). 'On Second-order Optimization Methods for Federated Learning'. In: *arXiv preprint arXiv:2109.02388*.

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

Chen, Congliang et al. (2021). 'Quantized adam with error feedback'. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 12.5, pp. 1–26.

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

Chen, Xiangyi, Xiaoyun Li and Ping Li (2020a). 'Toward Communication Efficient Adaptive Gradient Method'. In: *FODS '20: ACM-IMS Foundations of Data Science Conference, Virtual Event, USA, October 19-20, 2020*. Ed. by Jeannette M. Wing and David Madigan. ACM, pp. 119–128. DOI: 10.1145/3412815.3416891. URL: https://doi.org/10.1145/3412815.3416891.

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

Duchi, John C., Elad Hazan and Yoram Singer (2011). 'Adaptive Subgradient Methods for Online Learning and Stochastic Optimization'. In: *J. Mach. Learn. Res.* 12, pp. 2121–2159. URL: http://dl.acm.org/citation.cfm?id=2021068.

Fallah, Alireza, Aryan Mokhtari and Asuman Ozdaglar (2020). 'Personalized federated learning: A meta-learning approach'. In: *arXiv preprint arXiv:2002.07948*.

Foret, Pierre et al. (2020). 'Sharpness-aware minimization for efficiently improving generalization'. In: *arXiv preprint arXiv:2010.01412*.

Gao, Liang et al. (2022). 'FedDC: Federated Learning with Non-IID Data via Local Drift Decoupling and Correction'. In: *arXiv preprint arXiv:2203.11751*.

Gong, Yonghai, Yichuan Li and Nikolaos M Freris (2022). 'FedADMM: A Robust Federated Deep Learning Framework with Adaptivity to System Heterogeneity'. In: *arXiv preprint arXiv:2204.03529*.

Hanzely, Filip and Peter Richtárik (2020). 'Federated Learning of a Mixture of Global and Local Models'. In: *CoRR* abs/2002.05516. arXiv: 2002.05516. URL: https://arxiv.org/abs/2002.05516.

He, Kaiming et al. (2016). 'Deep Residual Learning for Image Recognition'. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, pp. 770–778. DOI: 10.1109/CVPR.2016.90. URL: https://doi.org/10.1109/CVPR.2016.90.

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

Hsieh, Kevin et al. (2020). 'The non-iid data quagmire of decentralized machine learning'. In: *International Conference on Machine Learning*. PMLR, pp. 4387–4398.

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

Huang, Yutao et al. (2021). 'Personalized Cross-Silo Federated Learning on Non-IID Data.' In: *AAAI*, pp. 7865–7873.

Johnson, Rie and Tong Zhang (2013). 'Accelerating Stochastic Gradient Descent using Predictive Variance Reduction'. In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. Ed. by Christopher J. C. Burges et al., pp. 315–323. URL: https://proceedings.neurips.cc/paper/2013/hash/ac1dd209cbcc5e5d1c6e28598e8cbbe8-Abstract.html.

Kairouz, Peter et al. (2021). 'Advances and open problems in federated learning'. In: *Foundations and Trends® in Machine Learning* 14.1–2, pp. 1–210.

Karimireddy, Sai Praneeth et al. (2020a). 'Mime: Mimicking centralized stochastic algorithms in federated learning'. In: *arXiv preprint arXiv:2008.03606*.

Karimireddy, Sai Praneeth et al. (2020b). 'SCAFFOLD: Stochastic Controlled Averaging for Federated Learning'. In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 5132–5143. URL: http://proceedings.mlr.press/v119/karimireddy20a.html.

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

Kim, Geeho, Jinkyu Kim and Bohyung Han (2022). 'Communication-Efficient Federated Learning with Acceleration of Global Momentum'. In: *arXiv preprint arXiv:2201.03172*.

Kingma, Diederik P. and Jimmy Ba (2015). 'Adam: A Method for Stochastic Optimization'. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.6980.

Konečny, Jakub et al. (2016). 'Federated learning: Strategies for improving communication efficiency'. In: *arXiv preprint arXiv:1610.05492*.

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

Li, Li et al. (2020a). 'A review of applications in federated learning'. In: *Computers & Industrial Engineering* 149, p. 106854.

Li, Tian et al. (2019). 'Feddane: A federated newton-type method'. In: *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, pp. 1227–1231.

Li, Tian et al. (2020b). 'Federated learning: Challenges, methods, and future directions'. In: *IEEE Signal Processing Magazine* 37.3, pp. 50–60.

Li, Tian et al. (2020c). 'Federated Optimization in Heterogeneous Networks'. In: *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*. Ed. by Inderjit S. Dhillon, Dimitris S. Papailiopoulos and Vivienne Sze. mlsys.org. URL: https://proceedings.mlsys.org/book/316.pdf.

Li, Xiang et al. (2020d). 'On the Convergence of FedAvg on Non-IID Data'. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=HJxNAnVtDS.

Li, Xiaoyu and Francesco Orabona (2019). 'On the Convergence of Stochastic Gradient Descent with Adaptive Stepsizes'. In: *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*. Ed. by Kamalika Chaudhuri and Masashi Sugiyama. Vol. 89. Proceedings of Machine Learning Research. PMLR, pp. 983–992. URL: http://proceedings.mlr.press/v89/li19c.html.

Lin, Tao et al. (2020). 'Don't Use Large Mini-batches, Use Local SGD'. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=B1eyO1BFPr.

Liu, Ji et al. (2022). 'From distributed machine learning to federated learning: A survey'. In: *Knowledge and Information Systems*, pp. 1–33.

Liu, Wei et al. (2020). 'Accelerating federated learning via momentum gradient descent'. In: *IEEE Transactions on Parallel and Distributed Systems* 31.8, pp. 1754–1766.

Malinovskiy, Grigory et al. (2020). 'From local SGD to local fixed-point methods for federated learning'. In: *International Conference on Machine Learning*. PMLR, pp. 6692–6701.

McMahan, Brendan et al. (2017a). 'Communication-Efficient Learning of Deep Networks from Decentralized Data'. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*. Ed. by Aarti Singh and Xiaojin (Jerry) Zhu. Vol. 54. Proceedings of Machine Learning Research. PMLR, pp. 1273–1282. URL: http://proceedings.mlr.press/v54/mcmahan17a.html.

McMahan, Brendan et al. (2017b). 'Communication-efficient learning of deep networks from decentralized data'. In: *Artificial intelligence and statistics*. PMLR, pp. 1273–1282.

McMahan, H. Brendan and Matthew J. Streeter (2010). 'Adaptive Bound Optimization for Online Convex Optimization'. In: *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*. Ed. by Adam Tauman Kalai and Mehryar Mohri. Omnipress, pp. 244–256. URL: http://colt2010.haifa.il.ibm.com/papers/COLT2010proceedings.pdf%5C#page=252.

Mitra, Aritra et al. (2021). 'Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients'. In: *Advances in Neural Information Processing Systems* 34, pp. 14606–14619.

Murata, Tomoya and Taiji Suzuki (2021). 'Bias-variance reduced local SGD for less heterogeneous federated learning'. In: *arXiv preprint arXiv:2102.03198*.

Ozfatura, Emre, Kerem Ozfatura and Deniz Gündüz (2021). 'FedADC: Accelerated Federated Learning with Drift Control'. In: *IEEE International Symposium on Information Theory, ISIT 2021, Melbourne, Australia, July 12-20, 2021*. IEEE, pp. 467–472. DOI: 10.1109/ISIT45174.2021.9517850. URL: https://doi.org/10.1109/ISIT45174.2021.9517850.

Pham, Nhan H. et al. (2021). 'Federated Learning with Randomized Douglas-Rachford Splitting Methods'. In: *CoRR* abs/2103.03452. arXiv: 2103.03452. URL: https://arxiv.org/abs/2103.03452.

Reddi, Sashank J., Satyen Kale and Sanjiv Kumar (2018). 'On the Convergence of Adam and Beyond'. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=ryQu7f-RZ.

Reddi, Sashank J. et al. (2021). 'Adaptive Federated Optimization'. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=LkFG3lB13U5.

Stich, Sebastian U (2018). 'Local SGD converges fast and communicates little'. In: *arXiv preprint arXiv:1805.09767*.

T Dinh, Canh, Nguyen Tran and Josh Nguyen (2020). 'Personalized federated learning with moreau envelopes'. In: *Advances in Neural Information Processing Systems* 33, pp. 21394–21405.

Tan, Yue et al. (2022). 'FedProto: Federated Prototype Learning across Heterogeneous Clients'. In: *AAAI Conference on Artificial Intelligence*. Vol. 1.

Wang, Han, Siddartha Marella and James Anderson (2022). 'FedADMM: A Federated Primal-Dual Algorithm Allowing Partial Participation'. In: *arXiv preprint arXiv:2203.15104*.

Wang, Jianyu et al. (2020a). 'SlowMo: Improving Communication-Efficient Distributed SGD with Slow Momentum'. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. URL: https://openreview.net/forum?id=SkxJ8REYPH.

Wang, Jianyu et al. (2020b). 'Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization'. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. URL: https://proceedings.neurips.cc/paper/2020/hash/564127c03caab942e503ee6f810f54fd-Abstract.html.

— (2020c). 'Tackling the objective inconsistency problem in heterogeneous federated optimization'. In: *Advances in neural information processing systems* 33, pp. 7611–7623.

Wang, Jianyu et al. (2021a). 'Local Adaptivity in Federated Learning: Convergence and Consistency'. In: *arXiv preprint arXiv:2106.02305*.

— (2021b). 'Local Adaptivity in Federated Learning: Convergence and Consistency'. In: *CoRR* abs/2106.02305. arXiv: 2106.02305. URL: https://arxiv.org/abs/2106.02305.

Woodworth, Blake, Kumar Kshitij Patel and Nathan Srebro (2020a). 'Minibatch vs local sgd for heterogeneous distributed learning'. In: *arXiv preprint arXiv:2006.04735*.

Woodworth, Blake et al. (2020b). 'Is local SGD better than minibatch SGD?' In: *International Conference on Machine Learning*. PMLR, pp. 10334–10343.

Wu, Xiaoxia, Simon S. Du and Rachel Ward (2019). 'Global Convergence of Adaptive Gradient Methods for An Over-parameterized Neural Network'. In: *CoRR* abs/1902.07111. arXiv: 1902.07111. URL: http://arxiv.org/abs/1902.07111.

Xie, Cong et al. (2019). 'Local AdaAlter: Communication-Efficient Stochastic Gradient Descent with Adaptive Learning Rates'. In: *CoRR* abs/1911.09030. arXiv: 1911.09030. URL: http://arxiv.org/abs/1911.09030.

Xu, Jing et al. (2021). 'FedCM: Federated Learning with Client-level Momentum'. In: *CoRR* abs/2106.10874. arXiv: 2106.10874. URL: https://arxiv.org/abs/2106.10874.

Yang, Haibo, Minghong Fang and Jia Liu (2021a). 'Achieving Linear Speedup with Partial Worker Participation in Non-IID Federated Learning'. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=jDdzh5ul-d.

— (2021b). 'Achieving Linear Speedup with Partial Worker Participation in Non-IID Federated Learning'. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=jDdzh5ul-d.

Yang, Peng et al. (2022). 'Over-the-Air Federated Learning via Second-Order Optimization'. In: *arXiv preprint arXiv:2203.15488*.

Yang, Qiang et al. (2019). 'Federated learning'. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 13.3, pp. 1–207.

Yu, Hao, Rong Jin and Sen Yang (2019a). 'On the Linear Speedup Analysis of Communication Efficient Momentum SGD for Distributed Non-Convex Optimization'. In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov.

Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 7184–7193. URL: http://proceedings.mlr.press/v97/yu19d.html.

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

Zeiler, Matthew D. (2012). 'ADADELTA: An Adaptive Learning Rate Method'. In: *CoRR* abs/1212.5701. arXiv: 1212.5701. URL: http://arxiv.org/abs/1212.5701.

Zhang, Xinwei et al. (2021). 'FedPD: A Federated Learning Framework With Adaptivity to Non-IID Data'. In: *IEEE Trans. Signal Process.* 69, pp. 6055–6070. DOI: 10.1109/TSP.2021.3115952. URL: https://doi.org/10.1109/TSP.2021.3115952.

Zhao, Haoyu, Zhize Li and Peter Richtárik (2021a). 'FedPAGE: A fast local stochastic gradient method for communication-efficient federated learning'. In: *arXiv preprint arXiv:2108.04755*.

Zhao, Haoyu et al. (2021b). 'Faster rates for compressed federated learning with client-variance reduction'. In: *arXiv preprint arXiv:2112.13097*.

Zhao, Yang, Hao Zhang and Xiuyuan Hu (2022). 'Penalizing Gradient Norm for Efficiently Improving Generalization in Deep Learning'. In: *arXiv preprint arXiv:2202.03599*.

1 Appendix A Proof Details

In this part we will demonstrate the proofs of all formula mentioned in this thesis. Each formula is presented in the form of a lemma.

Firstly we state some important lemmas applied in the proof.

LEMMA 3. *(Bounded global update) The global update* $\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^t$ *holds the upper bound of:*

$$\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2 \leq \frac{1}{\gamma}\left(\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t}\|^2\right) + \mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t\|^2.$$

PROOF. According to the lemma 2,we have:

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

Take the L2-norm and we have:

$$\begin{aligned}\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^t\|^2 &= \|(1-\gamma)\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1} + \gamma\frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t\|^2 \\ &\leq (1-\gamma)\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2 + \gamma\|\frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t\|^2.\end{aligned}$$

Thus we have the following recursion,

$$\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2 \leq \frac{1}{\gamma}\left(\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t}\|^2\right) + \mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t\|^2.$$

□

LEMMA 4. *(Bounded local update) The local update* $\hat{\mathbf{g}}_i^t$ *holds the upper bound of:*

$$\frac{1}{m}\sum_{i\in[m]}\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2 \leq \frac{P}{\gamma}\frac{1}{m}\sum_{i\in[m]}\left(\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\hat{\mathbf{g}}_i^{t}\|^2\right) + \frac{24PL^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t - \mathbf{x}^t\|^2 + 12P\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2 + P(12\sigma_g^2 + \sigma_l^2),$$

where $\frac{1}{P} = 1 - \frac{24\lambda^2L^2(1-2\gamma)^2}{\gamma^2}$.

PROOF. According to the lemma2, we have:

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

Take the L2-norm and we have:

$$\begin{aligned}
\|\hat{\mathbf{g}}_i^t\|^2 &= \|(1-\gamma)\hat{\mathbf{g}}_i^{t-1} + \gamma\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t\|^2 \\
&\overset{(a)}{\leq} (1-\gamma)\|\hat{\mathbf{g}}_i^{t-1}\|^2 + \gamma\|\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t\|^2 \\
&\overset{(b)}{\leq} (1-\gamma)\|\hat{\mathbf{g}}_i^{t-1}\|^2 + \gamma\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\|\tilde{\mathbf{g}}_{i,k}^t\|^2 \\
&= (1-\gamma)\|\hat{\mathbf{g}}_i^{t-1}\|^2 + \sum_{k=0}^{K-1}\gamma_k\|\tilde{\mathbf{g}}_{i,k}^t\|^2.
\end{aligned}$$

(a) and (b) apply the Jensen inequality.

Thus we have the following recursion:

$$\frac{1}{m}\sum_{i\in[m]}\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2 \leq \frac{1}{\gamma}\frac{1}{m}\sum_{i\in[m]}\left(\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\hat{\mathbf{g}}_i^{t}\|^2\right) + \frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\tilde{\mathbf{g}}_{i,k}^t\|^2.$$

Here we provide a loose upper bound as a constant for the quasi-stochastic gradient:

$$
\begin{aligned}
&\frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\tilde{\mathbf{g}}_{i,k}^t\|^2 \\
&=\frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|(1-\alpha)\mathbf{g}_{i,k,1}^t+\alpha\mathbf{g}_{i,k,2}^t\|^2 \\
&=\frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{g}_{i,k,1}^t+\alpha(\mathbf{g}_{i,k,2}^t-\mathbf{g}_{i,k,1}^t)\|^2 \\
&\leq\frac{2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\left(\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k}^t)\|^2+\alpha^2\mathbb{E}_t\|\nabla F_i(\check{\mathbf{x}}_{i,k}^t)-\nabla F_i(\mathbf{x}_{i,k}^t)\|^2\right)+\sigma_l^2 \\
&\leq\frac{2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\left(\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k}^t)\|^2+\alpha^2L^2\rho^2\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k}^t)\|^2\right)+\sigma_l^2 \\
&\leq\frac{4}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k}^t)-\nabla F_i(\mathbf{z}^t)+\nabla F_i(\mathbf{z}^t)-\nabla F(\mathbf{z}^t)+\nabla F(\mathbf{z}^t)\|^2+\sigma_l^2 \\
&\leq\frac{12L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{z}^t\|^2+12\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2+(12\sigma_g^2+\sigma_l^2) \\
&\leq\frac{12L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t+\mathbf{x}^t-\mathbf{u}^t+\mathbf{u}^t-\mathbf{z}^t\|^2 \\
&\quad+12\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2+(12\sigma_g^2+\sigma_l^2) \\
&\leq\frac{24L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2+24L^2\|\mathbf{x}^t-\mathbf{u}^t+\mathbf{u}^t-\mathbf{z}^t\|^2+(12\sigma_g^2+\sigma_l^2) \\
&\quad+12\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2 \\
&\leq\frac{24L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2+\frac{24L^2\lambda^2(1-2\gamma)^2}{\gamma^2}\frac{1}{m}\sum_i\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2 \\
&\quad+12\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2+(12\sigma_g^2+\sigma_l^2).
\end{aligned}
$$

We applies the Jensen inequality, the basic inequality $\|\sum_{i=1}^n\mathbf{a}_i\|^2\leq n\sum_{i=1}^n\|\mathbf{a}_i\|^2$, and the upper bound of $\rho\leq\frac{1}{\alpha L}$. Combining the above inequalities, let $\frac{1}{P}=1-\frac{24L^2\lambda^2(1-2\gamma^2)}{\gamma^2}$ is the

constant, we have:

$$\frac{1}{m}\sum_{i\in[m]}\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2 \leq \frac{P}{\gamma}\frac{1}{m}\sum_{i\in[m]}\left(\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\hat{\mathbf{g}}_i^{t}\|^2\right) + \frac{24PL^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t - \mathbf{x}^t\|^2 + 12P\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2 + P(12\sigma_g^2 + \sigma_l^2).$$

□

L-smoothness of the Function F

For the general non-convex case, according to the Assumptions and the smoothness of F, we take the conditional expectation at round $t+1$ and expand the $F(\mathbf{z}^{t+1})$ as:

$$\begin{aligned}
\mathbb{E}_t[F(\mathbf{z}^{t+1})] &\leq F(\mathbf{z}^t) + \mathbb{E}_t\langle\nabla F(\mathbf{z}^t), \mathbf{z}^{t+1} - \mathbf{z}^t\rangle + \frac{L}{2}\mathbb{E}_t\|\mathbf{z}^{t+1} - \mathbf{z}^t\|^2 \\
&= F(\mathbf{z}^t) + \langle\nabla F(\mathbf{z}^t), \mathbb{E}_t[\mathbf{z}^{t+1}] - \mathbf{z}^t\rangle + \frac{L}{2}\mathbb{E}_t\|\mathbf{z}^{t+1} - \mathbf{z}^t\|^2 \\
&= F(\mathbf{z}^t) + \mathbb{E}_t\langle\nabla F(\mathbf{z}^t), -\lambda\frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t\rangle + \frac{L}{2}\mathbb{E}_t\|\mathbf{z}^{t+1} - \mathbf{z}^t\|^2 \\
&= F(\mathbf{z}^t) - \lambda\mathbb{E}_t\langle\nabla F(\mathbf{z}^t), \frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t - \nabla F(\mathbf{z}^t) + \nabla F(\mathbf{z}^t)\rangle \\
&\quad + \frac{L}{2}\mathbb{E}_t\|\mathbf{z}^{t+1} - \mathbf{z}^t\|^2 \\
&= F(\mathbf{z}^t) - \lambda\|\nabla F(\mathbf{z}^t)\|^2 \underbrace{-\lambda\mathbb{E}_t\langle\nabla F(\mathbf{z}^t), \frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t - \nabla F(\mathbf{z}^t)\rangle}_{\textbf{R1}} \\
&\quad + \frac{L}{2}\underbrace{\mathbb{E}_t\|\mathbf{z}^{t+1} - \mathbf{z}^t\|^2}_{\textbf{R2}}.
\end{aligned}$$

Bounded R1

Note that **R1** can be bounded as:

$$
\begin{aligned}
\mathbf{R1} &= -\lambda \mathbb{E}_t \langle \nabla F(\mathbf{z}^t), \frac{1}{m} \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t - \nabla F(\mathbf{z}^t) \rangle \\
&\overset{(a)}{=} -\lambda \mathbb{E}_t \langle \nabla F(\mathbf{z}^t), \frac{1}{m} \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \tilde{\mathbf{g}}_{i,k}^t - \frac{1}{m} \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \nabla F_i(\mathbf{z}^t) \rangle \\
&\overset{(b)}{=} \frac{\lambda}{2} \|\nabla F(\mathbf{z}^t)\|^2 + \frac{\lambda}{2} \mathbb{E}_t \| \frac{1}{m} \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \big(\mathbb{E}\tilde{\mathbf{g}}_{i,k}^t - \nabla F_i(\mathbf{z}^t) \big) \|^2 - \frac{\lambda}{2m^2} \mathbb{E}_t \| \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \mathbb{E}\tilde{\mathbf{g}}_{i,k}^t \|^2 \\
&\overset{(c)}{\leq} \frac{\lambda}{2} \|\nabla F(\mathbf{z}^t)\|^2 + \underbrace{\frac{\lambda}{2} \frac{1}{m} \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \mathbb{E}_t \| \mathbb{E}\tilde{\mathbf{g}}_{i,k}^t - \nabla F_i(\mathbf{z}^t) \|^2}_{\mathbf{R1}.\mathbf{a}} - \frac{\lambda}{2m^2} \mathbb{E}_t \| \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \mathbb{E}\tilde{\mathbf{g}}_{i,k}^t \|^2.
\end{aligned}
$$

(a) applies the fact that $\frac{1}{m} \sum_{i\in[m]} \nabla F_i(\mathbf{z}^t) = \nabla F(\mathbf{z}^t)$. (b) applies $-\langle \mathbf{x}, \mathbf{y} \rangle = \frac{1}{2} \big(\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \|\mathbf{x} + \mathbf{y}\|^2 \big)$. (c) applies the Jensen's inequality and the fact that $\sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} = 1$.

According to the update rule we have:

$$
\begin{aligned}
\mathbb{E}\tilde{\mathbf{g}}_{i,k}^t &= (1-\alpha) \mathbb{E} \left[\mathbf{g}_{i,k,1}^t \right] + \alpha \mathbb{E} \left[\mathbf{g}_{i,k,2}^t \right] = (1-\alpha) \mathbb{E} \left[\nabla F_i(\mathbf{x}_{i,k}^t; \varepsilon_{i,k}^t) \right] + \alpha \mathbb{E} \left[\nabla F_i(\check{\mathbf{x}}_{i,k}^t; \varepsilon_{i,k}^t) \right] \\
&= (1-\alpha) \nabla F_i(\mathbf{x}_{i,k}^t) + \alpha \nabla F_i(\check{\mathbf{x}}_{i,k}^t) = (1-\alpha) \nabla F_i(\mathbf{x}_{i,k}^t) + \alpha \nabla F_i(\mathbf{x}_{i,k}^t + \rho \mathbf{g}_{i,k,1}^t).
\end{aligned}
$$

Let $\rho \leq \frac{1}{\sqrt{3}\alpha L}$, thus we could bound the term $\mathbf{R1}.\mathbf{a}$ as follows:

$$
\begin{aligned}
&\frac{1}{m} \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \mathbb{E}_t \| \mathbb{E}\tilde{\mathbf{g}}_{i,k}^t - \nabla F_i(\mathbf{z}^t) \|^2 \\
= &\frac{1}{m} \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \mathbb{E}_t \| (1-\alpha) \nabla F_i(\mathbf{x}_{i,k}^t) + \alpha \nabla F_i(\mathbf{x}_{i,k}^t + \rho \mathbf{g}_{i,k,1}^t) - \nabla F_i(\mathbf{z}^t) \|^2 \\
= &\frac{1}{m} \sum_{i\in[m]} \sum_{k=0}^{K-1} \frac{\gamma_k}{\gamma} \mathbb{E}_t \| \nabla F_i(\mathbf{x}_{i,k}^t) - \nabla F_i(\mathbf{z}^t) + \alpha \Big(\nabla F_i(\mathbf{x}_{i,k}^t + \rho \mathbf{g}_{i,k,1}^t) - \nabla F_i(\mathbf{x}_{i,k}^t) \Big) \|^2
\end{aligned}
$$

$$\leq \frac{2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k}^t)-\nabla F_i(\mathbf{z}^t)\|^2+\frac{2\alpha^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\nabla F_i(\breve{\mathbf{x}}_{i,k}^t)-\nabla F_i(\mathbf{x}_{i,k}^t)\|^2$$

$$\leq \frac{2L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{z}^t\|^2+\frac{2\alpha^2L^2\rho^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{g}_{i,k,1}^t\|^2$$

$$= \frac{2L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t+\mathbf{x}^t-\mathbf{u}^t+\mathbf{u}^t-\mathbf{z}^t\|^2+\frac{2\alpha^2L^2\rho^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{g}_{i,k,1}^t\|^2$$

$$\leq \frac{4L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2+\frac{4L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|(\mathbf{x}^t-\mathbf{u}^t)+(\mathbf{u}^t-\mathbf{z}^t)\|^2$$

$$+\frac{2\alpha^2L^2\rho^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{g}_{i,k,1}^t-\nabla F_i(\mathbf{x}_{i,k}^t)\|^2+\frac{2\alpha^2L^2\rho^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k}^t)\|^2$$

$$\leq \frac{4L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2+\frac{2\alpha^2L^2\rho^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k}^t)\|^2+2\alpha^2L^2\rho^2\sigma_l^2$$

$$+4L^2\mathbb{E}_t\|(\mathbf{x}^t-\mathbf{u}^t)+(\mathbf{u}^t-\mathbf{z}^t)\|^2$$

$$= \frac{4L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2+\frac{2\alpha^2L^2\rho^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k}^t)\|^2+2\alpha^2L^2\rho^2\sigma_l^2$$

$$+4L^2\mathbb{E}_t\|-\frac{1}{m}\sum_{i\in[m]}\lambda\hat{\mathbf{g}}_i^{t-1}+\frac{\gamma-1}{\gamma}(\mathbf{u}^t-\mathbf{u}^{t-1})\|^2$$

$$= \frac{4L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2+\frac{2\alpha^2L^2\rho^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k}^t)\|^2+2\alpha^2L^2\rho^2\sigma_l^2$$

$$+4L^2\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\Big((\mathbf{u}^t-\mathbf{u}^{t-1}+\lambda\hat{\mathbf{g}}_i^{t-1})-\frac{1}{\gamma}(\mathbf{u}^t-\mathbf{u}^{t-1}+\lambda\hat{\mathbf{g}}_i^{t-1})+(\frac{1-2\gamma}{\gamma})\lambda\hat{\mathbf{g}}_i^{t-1}\Big)\|^2$$

$$= \frac{4L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2+\frac{2\alpha^2L^2\rho^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k}^t)\|^2+2\alpha^2L^2\rho^2\sigma_l^2$$

$$+\frac{4\lambda^2L^2(1-2\gamma)^2}{\gamma^2}\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2$$

$$= \frac{4L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2+\frac{4\lambda^2L^2(1-2\gamma)^2}{\gamma^2}\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2+2\alpha^2L^2\rho^2\sigma_l^2$$

$$+\frac{2\alpha^2L^2\rho^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k}^t)-\nabla F_i(\mathbf{z}^t)+\nabla F_i(\mathbf{z}^t)-\nabla F(\mathbf{z}^t)+\nabla F(\mathbf{z}^t)\|^2$$

$$\overset{(a)}{\leq}\frac{4L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2+\frac{4\lambda^2L^2(1-2\gamma)^2}{\gamma^2}\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2+2\alpha^2L^2\rho^2\sigma_l^2$$

$$+\frac{2L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{z}^t\|^2+6\alpha^2L^2\rho^2\sigma_g^2+6\alpha^2L^2\rho^2\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2$$

$$\leq\frac{8L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2+\frac{8\lambda^2L^2(1-2\gamma)^2}{\gamma^2}\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2+2\alpha^2L^2\rho^2\sigma_l^2$$

$$+6\alpha^2L^2\rho^2\sigma_g^2+6\alpha^2L^2\rho^2\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2.$$

$$\overset{(b)}{\leq}\frac{8L^2}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2+\frac{8\lambda^2L^2(1-2\gamma)^2}{\gamma^3}\left(\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2-\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^t\|^2\right)$$

$$+\frac{8\lambda^2L^2(1-2\gamma)^2}{\gamma^2}\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t\|^2+2\alpha^2L^2\rho^2\sigma_l^2+6\alpha^2L^2\rho^2\sigma_g^2+6\alpha^2L^2\rho^2\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2.$$

(a) applies the bound of ρ as $\rho\leq\frac{1}{\sqrt{3}\alpha L}$. (b) applies the lemma 3. These others use the fact $\mathbb{E}[x-\mathbb{E}[x]]^2=\mathbb{E}[x^2]-[\mathbb{E}[x]]^2$ and $\|\mathbf{x}+\mathbf{y}\|^2\leq(1+a)\|\mathbf{x}\|^2+(1+\frac{1}{a})\|\mathbf{y}\|^2$.

We denote $\mathbf{c}^t=\frac{1}{m}\sum_{i\in m}\sum_{k=0}^{K-1}(\gamma_k/\gamma)\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2$ term as the local offset after k iterations updates, we firstly consider the $\mathbf{c}_k^t=\frac{1}{m}\sum_{i\in m}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2$ and it can be bounded as:

$$\mathbf{c}_k^t=\frac{1}{m}\sum_{i\in[m]}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}^t\|^2=\frac{1}{m}\sum_{i\in[m]}\mathbb{E}_t\|\mathbf{x}_{i,k}^t-\mathbf{x}_{i,k-1}^t+\mathbf{x}_{i,k-1}^t-\mathbf{x}_{i,0}^t\|^2$$

$$=\frac{1}{m}\sum_{i\in[m]}\mathbb{E}_t\|-\eta_l(\tilde{\mathbf{g}}_{i,k-1}^t-\hat{\mathbf{g}}_i^{t-1})+(1-\frac{\eta_l}{\lambda})(\mathbf{x}_{i,k-1}^t-\mathbf{x}_{i,0}^t)\|^2$$

$$\leq(1+a)(1-\frac{\eta_l}{\lambda})^2\frac{1}{m}\sum_{i\in[m]}\mathbb{E}_t\|\mathbf{x}_{i,k-1}^t-\mathbf{x}_{i,0}^t\|^2+(1+\frac{1}{a})\frac{\eta_l^2}{m}\sum_{i\in[m]}\mathbb{E}_t\|\tilde{\mathbf{g}}_{i,k-1}^t-\hat{\mathbf{g}}_i^{t-1}\|^2$$

$$=(1+a)(1-\frac{\eta_l}{\lambda})^2\mathbf{c}_{k-1}^t+(1+\frac{1}{a})\frac{\eta_l^2}{m}\sum_{i\in[m]}\mathbb{E}_t\|(1-\alpha)\mathbf{g}_{i,k-1,1}^t+\alpha\mathbf{g}_{i,k-1,2}^t-\hat{\mathbf{g}}_i^{t-1}\|^2$$

$$
\begin{aligned}
&= (1+\frac{1}{a})\frac{\eta_l^2}{m}\sum_{i\in[m]}\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k-1}^t)-\hat{\mathbf{g}}_i^{t-1}+\alpha(\nabla F_i(\check{\mathbf{x}}_{i,k-1}^t)-\nabla F_i(\mathbf{x}_{i,k-1}^t))\|^2\\
&\quad+(1+\frac{1}{a})\eta_l^2\sigma_l^2+(1+a)(1-\frac{\eta_l}{\lambda})^2\mathbf{c}_{k-1}^t\\
&\le (1+\frac{1}{a})\frac{3\eta_l^2}{m}\sum_{i\in[m]}\left(\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k-1}^t)\|^2+\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2+\alpha^2L^2\rho^2\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k-1}^t)\|^2\right)\\
&\quad+(1+\frac{1}{a})\eta_l^2\sigma_l^2+(1+a)(1-\frac{\eta_l}{\lambda})^2\mathbf{c}_{k-1}^t\\
&\le (1+\frac{1}{a})\frac{4\eta_l^2}{m}\sum_{i\in[m]}\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k-1}^t)\|^2+(1+\frac{1}{a})\frac{3\eta_l^2}{m}\sum_{i\in[m]}\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2+(1+\frac{1}{a})\eta_l^2\sigma_l^2\\
&\quad+(1+a)(1-\frac{\eta_l}{\lambda})^2\mathbf{c}_{k-1}^t\\
&\le (1+\frac{1}{a})\frac{4\eta_l^2}{m}\sum_{i\in[m]}\mathbb{E}_t\|\nabla F_i(\mathbf{x}_{i,k-1}^t)-\nabla F_i(\mathbf{x}^t)+\nabla F_i(\mathbf{x}^t)-\nabla F_i(\mathbf{z}^t)+\nabla F_i(\mathbf{z}^t)-\nabla F(\mathbf{z}^t)\\
&\quad+\nabla F(\mathbf{z}^t)\|^2+(1+\frac{1}{a})\frac{3\eta_l^2}{m}\sum_{i\in[m]}\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2+(1+\frac{1}{a})\eta_l^2\sigma_l^2+(1+a)(1-\frac{\eta_l}{\lambda})^2\mathbf{c}_{k-1}^t\\
&\le (1+\frac{1}{a})\frac{16\eta_l^2L^2}{m}\sum_{i\in[m]}\mathbb{E}_t\|\mathbf{x}_{i,k-1}^t-\mathbf{x}^t\|^2+(1+\frac{1}{a})16\eta_l^2L^2\|\mathbf{x}^t-\mathbf{z}^t\|^2+(1+\frac{1}{a})\eta_l^2(16\sigma_g^2+\sigma_l^2)\\
&\quad+(1+\frac{1}{a})16\eta_l^2\|\nabla F(\mathbf{z}^t)\|^2+(1+\frac{1}{a})\frac{3\eta_l^2}{m}\sum_{i\in[m]}\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2+(1+a)(1-\frac{\eta_l}{\lambda})^2\mathbf{c}_{k-1}^t\\
&\le \left[(1+a)(1-\frac{\eta_l}{\lambda})^2+(1+\frac{1}{a})16\eta_l^2L^2\right]\mathbf{c}_{k-1}^t+(1+\frac{1}{a})\eta_l^2(16\sigma_g^2+\sigma_l^2)\\
&\quad+(1+\frac{1}{a})16\eta_l^2\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2+(1+\frac{1}{a})\eta_l^2\left[3+\frac{16\lambda^2L^2(1-2\gamma)^2}{\gamma^2}\right]\frac{1}{m}\sum_{i\in[m]}\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2\\
&= \left[(1+a)(1-\frac{\eta_l}{\lambda})^2+(1+\frac{1}{a})16\eta_l^2L^2\right]\mathbf{c}_{k-1}^t+(1+\frac{1}{a})\eta_l^2(16\sigma_g^2+\sigma_l^2)\\
&\quad+(1+\frac{1}{a})\eta_l^2L^2\left(88P-16\right)\mathbf{c}^t+(1+\frac{1}{a})\frac{2\eta_l^2(P-1)}{3}(12\sigma_g^2+\sigma_l^2)\\
&\quad+(1+\frac{1}{a})16\eta_l^2\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2+(1+\frac{1}{a})\eta_l^2(44P-8)\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2\\
&\quad+(1+\frac{1}{a})\frac{2\eta_l^2(P-1)}{3\gamma}\frac{1}{m}\sum_{i\in[m]}\left(\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2-\mathbb{E}_t\|\hat{\mathbf{g}}_i^t\|^2\right)
\end{aligned}
$$

When P satisfies the condition of $P \leq 2$, which means $\frac{1}{P} = 1 - \frac{24\lambda^2 L^2 (1-2\gamma)^2}{\gamma^2} \geq \frac{1}{2}$, then we have the constant of $\frac{2(P-1)}{3} \leq \frac{2}{3} < 1$, let the last $12\sigma_g^2$ enlarged to $16\sigma_g^2$ for convenience, we have:

$$\mathbf{c}_k^t \leq \left[(1+a)(1-\frac{\eta_l}{\lambda})^2 + (1+\frac{1}{a})16\eta_l^2 L^2\right]\mathbf{c}_{k-1}^t + 2(1+\frac{1}{a})\eta_l^2(16\sigma_g^2+\sigma_l^2) + 160(1+\frac{1}{a})\eta_l^2 L^2 \mathbf{c}^t$$
$$96(1+\frac{1}{a})\eta_l^2 \mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2 + 2(1+\frac{1}{a})\frac{\eta_l^2}{\gamma}\frac{1}{m}\sum_{i\in[m]}\left(\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\hat{\mathbf{g}}_i^t\|^2\right).$$

Here we get the recursion formula between the $\mathbf{c}_k^t$ and $\mathbf{c}_{k-1}^t$. Actually we need to upper bound the $\mathbf{c}^t = \sum_{k=0}^{K-1}(\gamma_k/\gamma)\mathbf{c}_k^t$, thus let the weight satisfies that:

$$(1+a)(1-\frac{\eta_l}{\lambda})^2 + (1+\frac{1}{a})16\eta_l^2 L^2 \leq \frac{\gamma_{K-2}}{\gamma_{K-1}} = \frac{\gamma_{K-3}}{\gamma_{K-2}} = \cdots = \frac{\gamma_1}{\gamma_0} = 1 - \frac{\eta_l}{\lambda},$$

let $\eta_l \leq \lambda$ and thus we have:

$$\begin{aligned}
\mathbf{c}^t &= \sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbf{c}_k^t \\
&\leq 2(1+\frac{1}{a})\frac{\eta_l^2}{\gamma}\sum_{k'=0}^{K-1}\left(\sum_{k=0}^{k'-1}\gamma_k\right)\Big(16\sigma_g^2+\sigma_l^2+48\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2+80L^2\mathbf{c}^t \\
&\quad + \frac{1}{m\gamma}\sum_{i\in[m]}\left(\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\hat{\mathbf{g}}_i^t\|^2\right)\Big) \\
&\overset{(a)}{\leq} 2(1+\frac{1}{a})\eta_l^2\sum_{k'=0}^{K-1}\left(\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\right)\Big(16\sigma_g^2+\sigma_l^2+48\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2+80L^2\mathbf{c}^t \\
&\quad + \frac{1}{m\gamma}\sum_{i\in[m]}\left(\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\hat{\mathbf{g}}_i^t\|^2\right)\Big) \\
&= 2(1+\frac{1}{a})\eta_l^2 K\left(16\sigma_g^2+\sigma_l^2+48\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2 + \frac{1}{m\gamma}\sum_{i\in[m]}\left(\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\hat{\mathbf{g}}_i^t\|^2\right)\right)
\end{aligned}$$

$$+ 160(1 + \frac{1}{a})\eta_l^2 L^2 K \mathbf{c}^t.$$

(a) enlarge the sum from $k^{'}$ to $K-1$ where $k^{'} \leq K-1$.

Let η_l satisfies the upper bound of $\eta_l \leq \frac{1}{\sqrt{320(1+1/a)KL}}$ for convenience, we can bound the $\mathbf{c}^t$ as:

$$\mathbf{c}^t = 4(1 + \frac{1}{a})\eta_l^2 K \left(16\sigma_g^2 + \sigma_l^2 + 48\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2 + \frac{1}{m\gamma}\sum_{i\in[m]} \left(\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\hat{\mathbf{g}}_i^t\|^2 \right) \right).$$

Let the a satisfies $a = 1$ for convenience, we summarize the extra terms above and bound the term $\mathbf{R1}.\mathbf{a}$ as:

$$\begin{aligned}
\mathbf{R1}.\mathbf{a} &= \frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}_t\|\mathbb{E}[\tilde{\mathbf{g}}_{i,k}^t] - \nabla F_i(\mathbf{z}^t)\|^2 \\
&\leq 8L^2\mathbf{c}^t + \frac{8\lambda^2L^2(1-2\gamma)^2}{\gamma^3}\left(\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^t\|^2 \right) + 2\alpha^2L^2\rho^2\sigma_l^2 \\
&\quad + \frac{8\lambda^2L^2(1-2\gamma)^2}{\gamma^2}\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t\|^2 + 6\alpha^2L^2\rho^2\sigma_g^2 + 6\alpha^2L^2\rho^2\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2 \\
&\leq \frac{8\lambda^2L^2(1-2\gamma)^2}{\gamma^3}\left(\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^t\|^2 \right) + 2\alpha^2L^2\rho^2\sigma_l^2 + 6\alpha^2L^2\rho^2\sigma_g^2 \\
&\quad + \frac{8\lambda^2L^2(1-2\gamma)^2}{\gamma^2}\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t\|^2 + \frac{64\eta_l^2L^2K}{m\gamma}\sum_{i\in[m]}\left(\mathbb{E}_t\|\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\hat{\mathbf{g}}_i^t\|^2 \right) \\
&\quad + 3072\eta_l^2L^2K\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2 + 6\alpha^2L^2\rho^2\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2 + 64\eta_l^2L^2K(16\sigma_g^2 + \sigma_l^2).
\end{aligned}$$

thus we can bound the $\mathbf{R1}$ as follow:

$$\mathbf{R1} \leq \frac{\lambda}{2}\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2 + \frac{\lambda}{2}\mathbf{R1}.\mathbf{a} - \frac{\lambda}{2m^2}\mathbb{E}_t\|\sum_{i\in[m]}\sum_{k}^{K-1}\frac{\gamma_k}{\gamma}\mathbb{E}[\tilde{\mathbf{g}}_{i,k}^t]\|^2$$

$$\leq \left(\frac{\lambda}{2} + 3\lambda\alpha^2L^2\rho^2 + 1536\lambda\eta_l^2L^2K\right)\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2 + \frac{32\lambda\eta_l L^2K}{\gamma m}\sum_{i\in[m]}\left(\mathbb{E}\|\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}\|\hat{\mathbf{g}}_i^t\|^2\right)$$
$$+ \frac{4\lambda^3L^2(1-2\gamma)^2}{\gamma^3}\left(\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^t\|^2\right) + \lambda\alpha^2L^2\rho^2(3\sigma_g^2+\sigma_l^2)$$
$$+ \frac{4\lambda^3L^2(1-2\gamma)^2}{\gamma^2}\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t\|^2 + 32\lambda\eta_l^2L^2K(16\sigma_g^2+\sigma_l^2).$$

We notice that **R1** contains the same term with a negative weight, thus we can set another constrains for λ to eliminate this term. We will prove it in the next part. **Bounded Global Gradient**

As we have bounded the term **R1** and **R2**, according to the smoothness inequality, we combine the inequalities above and get the inequality:

$$\mathbb{E}_t[F(\mathbf{z}^{t+1})] \leq F(\mathbf{z}^t) - \lambda\|\nabla F(\mathbf{z}^t)\|^2 + \mathbf{R1} + \frac{L}{2}\mathbf{R2}$$
$$= F(\mathbf{z}^t) - \left(\frac{\lambda}{2} - 3\lambda\alpha^2L^2\rho^2 - 1536\lambda\eta_l^2L^2K\right)\|\nabla F(\mathbf{z}^t)\|^2 + \lambda\alpha^2L^2\rho^2(3\sigma_g^2+\sigma_l^2)$$
$$+ \left(\frac{4\lambda^3L^2(1-2\gamma)^2}{\gamma^2} + \frac{\lambda^2L}{2m^2} - \frac{\lambda}{2m^2}\right)\mathbb{E}_t\|\sum_{i\in[m]}\sum_{k=0}^{K-1}\frac{\gamma_k}{\gamma}\tilde{\mathbf{g}}_{i,k}^t\|^2$$
$$+ \frac{32\lambda\eta_l L^2K}{\gamma m}\sum_{i\in[m]}\left(\mathbb{E}\|\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}\|\hat{\mathbf{g}}_i^t\|^2\right) + 32\lambda\eta_l^2L^2K(16\sigma_g^2+\sigma_l^2)$$
$$+ \frac{4\lambda^3L^2(1-2\gamma)^2}{\gamma^3}\left(\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^t\|^2\right).$$

We follow as (Yang et al. 2021b) to set λ that it satisfies $\frac{4\lambda^3L^2(1-2\gamma)^2}{\gamma^2} + \frac{\lambda^2L}{2m^2} - \frac{\lambda}{2m^2} \leq 0$, which is easy to verified that λ has a upper bound for the quadratic inequality. Thus, the stochastic gradient term is diminished by this λ. We denote the constant $\lambda\kappa = \frac{\lambda}{2} - 3\lambda\alpha^2L^2\rho^2 -$

$1536\lambda\eta_l^2L^2K$ and take the full expectation on the bounded global gradient as:

$$\begin{aligned}\lambda\kappa\mathbb{E}\|\nabla F(\mathbf{z}^t)\|^2 \leq \Big(\mathbb{E}F(\mathbf{z}^t) - \mathbb{E}F(\mathbf{z}^{t+1})\Big) &+ \frac{32\lambda\eta_l L^2K}{\gamma m}\sum_{i\in[m]}\Big(\mathbb{E}\|\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}\|\hat{\mathbf{g}}_i^t\|^2\Big)\\ &+ \frac{4\lambda^3L^2(1-2\gamma)^2}{\gamma^3}\left(\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^{t-1}\|^2 - \mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^t\|^2\right)\\ &+ 32\lambda\eta_l^2L^2K(16\sigma_g^2+\sigma_l^2) + \lambda\alpha^2L^2\rho^2(3\sigma_g^2+\sigma_l^2).\end{aligned}$$

Take the full expectation and telescope sum on the inequality above and applying the fact that $F^* \leq F(\mathbf{x})$ for $\mathbf{x}\in\mathbb{R}^d$, we have:

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}_t\|\nabla F(\mathbf{z}^t)\|^2 &\leq \frac{1}{\lambda\kappa T}\big(F(\mathbf{z}^1) - \mathbb{E}_t[F(\mathbf{z}^T)]\big) + \frac{32\eta_l L^2K}{\kappa\gamma mT}\sum_{i\in[m]}\Big(\mathbb{E}\|\hat{\mathbf{g}}_i^0\|^2 - \mathbb{E}\|\hat{\mathbf{g}}_i^t\|^2\Big)\\ &+ \frac{4\lambda^2L^2(1-2\gamma)^2}{\kappa\gamma^3T}\left(\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^0\|^2 - \mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^t\|^2\right)\\ &+ \frac{1}{\kappa}\Big(32\lambda\eta_l^2L^2K(16\sigma_g^2+\sigma_l^2) + \lambda\alpha^2L^2\rho^2(3\sigma_g^2+\sigma_l^2)\Big)\\ &\leq \frac{1}{\lambda\kappa T}\big(F(\mathbf{z}^0) - F^*\big) + \frac{32\eta_l L^2K}{\kappa\gamma mT}\sum_{i\in[m]}\mathbb{E}\|\hat{\mathbf{g}}_i^0\|^2\\ &+ \frac{4\lambda^2L^2(1-2\gamma)^2}{\kappa\gamma^3T}\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^0\|^2\\ &+ \frac{1}{\kappa}\Big(32\lambda\eta_l^2L^2K(16\sigma_g^2+\sigma_l^2) + \lambda\alpha^2L^2\rho^2(3\sigma_g^2+\sigma_l^2)\Big)\end{aligned}$$

Here we summarize the conditions and some constrains in the above conclusion. Firstly we should note that $\gamma = 1-(1-\frac{\eta_l}{\lambda})^K < 1$ when $\eta_l \leq 2\lambda$. Thus we have $1/\gamma > 1$. When K satisfies that $K \geq \frac{\lambda}{\eta_l}$, $(1-\frac{\eta_l}{\lambda})^K \leq e^{-\frac{\eta_l}{\lambda}K} \leq e^{-1}$, and then $\gamma > 1-e^{-1}$ and $1/\gamma < \frac{e}{e-1} < 2$. To let $\kappa = \frac{1}{2} - 3\alpha^2L^2\rho^2 - 1536\eta_l^2L^2K > 0$ hold, ρ and η_l satisfy that $\rho < \frac{1}{\sqrt{6}\alpha L}$ and $\eta_l < \frac{1}{32\sqrt{3K}L}$.

$$\frac{1}{T}\sum_{t=1}^{T-1}\mathbb{E}\|\nabla F(\mathbf{z}^t)\|^2 \leq \frac{2(F(\mathbf{z}^1)-F^*)}{\lambda\kappa T} + \frac{64\eta_l L^2 K}{\kappa T}\frac{1}{m}\sum_{i\in[m]}\mathbb{E}\|\hat{\mathbf{g}}_i^0\|^2 + \frac{32\lambda^2 L^2}{\kappa T}\mathbb{E}_t\|\frac{1}{m}\sum_{i\in[m]}\hat{\mathbf{g}}_i^0\|^2$$
$$+ \frac{1}{\kappa}\left(32\lambda\eta_l^2 L^2 K(16\sigma_g^2+\sigma_l^2) + \lambda\alpha^2 L^2\rho^2(3\sigma_g^2+\sigma_l^2)\right).$$