Partial Model Averaging in Federated Learning: Performance Guarantees and Benefits

Sunwoo Lee[1], **Anit Kumar Sahu**[2], **Chaoyang He**[1], **Salman Avestimehr**[1]
[1]University of Southern California
[2]Amazon Alexa AI
[1]{sunwool,chaoyang.he,avestime}@usc.edu
[2]anitsah@amazon.com

Abstract

Local Stochastic Gradient Descent (SGD) with periodic model averaging (FedAvg) is a foundational algorithm in Federated Learning. The algorithm independently runs SGD on multiple workers and periodically averages the model across all the workers. When local SGD runs with many workers, however, the periodic averaging causes a significant model discrepancy across the workers making the global loss converge slowly. While recent advanced optimization methods tackle the issue focused on non-IID settings, there still exists the model discrepancy issue due to the underlying periodic model averaging. We propose a partial model averaging framework that mitigates the model discrepancy issue in Federated Learning. The partial averaging encourages the local models to stay close to each other on parameter space, and it enables to more effectively minimize the global loss. Given a fixed number of iterations and a large number of workers (128), the partial averaging achieves up to 2.2% higher validation accuracy than the periodic full averaging.

Introduction

Local Stochastic Gradient Descent (Local SGD) with periodic model averaging has been recently shown to be a promising alternative to vanilla synchronous SGD (Robbins and Monro 1951). The algorithm runs SGD on multiple workers independently and averages the model parameters across all the workers periodically. FedAvg (McMahan et al. 2017) is built around local SGD and has been shown to be effective in Federated Learning to solve problems involving non-Independent and Identically Distributed (non-IID) data. Several studies have shown that local SGD achieves linear speedup with respect to number of workers for convex and non-convex problems (Stich 2018; Yu, Yang, and Zhu 2019; Yu, Jin, and Yang 2019; Wang and Joshi 2018b; Haddadpour et al. 2019).

While the periodic model averaging dramatically reduces the communication cost in distributed training, it causes model discrepancy across all the workers. Due to variance of stochastic gradients and data heterogeneity, the independent local training steps can disperse the models over a wide region in the parameter space. Averaging a large number of such different local models can significantly distract the convergence of global loss as compared to synchronous SGD that only has one global model. The model discrepancy can adversely affect the convergence both in IID and non-IID settings. To scale up the training to hundreds, thousands, or even millions of workers in Federated Learning, it is crucial to address this issue.

Many researchers have put much effort into addressing the model discrepancy issue in non-IID settings. Variance Reduced Local-SGD (VRL-SGD) (Liang et al. 2019) and SCAFFOLD (Karimireddy et al. 2020) make use of extra control variates to accelerate the convergence by reducing variance of stochastic gradients. FedProx (Li et al. 2020) adds a proximal term to each local loss to suppress the distance among the local models. FedNova (Wang et al. 2020) normalizes the magnitude of local updates across the workers so that the model averaging less distracts the global loss. All these algorithms employ the periodic model averaging as a backbone of the model aggregation. Thus, although they mitigate the model discrepancy caused by the data heterogeneity, the issue still exists due to the underlying periodic model averaging scheme.

Breaking the convention of periodic full model averaging, we propose a *partial model averaging* framework to tackle the model discrepancy issue in Federated Learning. Instead of allowing the workers independently update the full model parameters within each communication round, our framework synchronizes a distinct subset of the model parameters every iteration. Such frequent synchronizations encourage all the local models to stay close to each other on parameter space, and thus the global loss is not strongly distracted when averaging many local models. Our empirical study shows that the partial model averaging effectively suppresses the degree of model discrepancy during the training, and it results in making the global loss converge faster than the periodic averaging. Within a fixed iteration budget, the faster convergence of the loss most likely results in achieving a higher validation accuracy in Federated Learning. We also theoretically analyze the convergence property of the proposed algorithm for smooth and non-convex problems considering both IID and non-IID data settings.

We focus on how the partial model averaging affects the classification performance when it replaces the underlying periodic model averaging scheme in Federated Learning. We

evaluate the performance of the proposed framework across a variety of computer vision and natural language processing tasks. Given a fixed number of iterations and a large number of workers (128), the partial averaging shows a faster convergence and achieves up to 2.2% higher validation accuracy than the periodic averaging. In addition, the partial averaging consistently accelerates the convergence across various degrees of the data heterogeneity. These results demonstrate that the partial averaging effectively mitigates the adverse impact of the model discrepancy on the federated neural network training. The partial averaging method has the same communication cost as the periodic averaging and does not require extra computations.

Contributions – We highlight our contributions below.

1. We propose a novel partial model averaging framework for large-scale Federated Learning. The framework tackles the model discrepancy in a foundational model averaging level. Our theoretical analysis provides a convergence guarantee for non-convex problems, achieving linear speedup with respect to the number of workers.
2. We explore benefits of the proposed partial averaging framework. Our empirical study demonstrates that the global loss is not strongly distracted when partially averaging the local models, which results in a faster convergence. We also report extensive experimental results across various benchmark datasets and models.
3. The partial averaging framework is readily applicable to any Federated Learning algorithms. Our study introduces promising future works regarding how to harmonize the layer-wise model aggregation scheme with many Federated Learning algorithms such as FedProx, FedNova, SCAFFOLD, and adaptive averaging interval methods.

Background

Local SGD with Periodic Model Averaging – We consider federated optimization problems of the form

$$\min_{\mathbf{x}\in\mathbb{R}^d} \left[F(\mathbf{x}) := \sum_{i=1}^{m} p_i F_i(\mathbf{x}) \right], \tag{1}$$

where $p_i = n_i/n$ is the ratio of local data to the total dataset, and $F_i(\mathbf{x}) = \frac{1}{n_i}\sum_{\xi\in\mathcal{D}} f_i(\mathbf{x},\xi)$ is the local objective function of client i. n is the global dataset size and n_i is the local dataset size.

The model averaging can be expressed as follows:

$$\mathbf{u}_k = \sum_{i=1}^{m} p_i \mathbf{x}_k^i, \tag{2}$$

where m is the number of workers (local models), $\mathbf{x}_k^i$ is the local model of worker i at iteration k, and $\mathbf{u}_k$ is the averaged model. Note that, $p_i = 1/m$ when the data is IID. The parameter update rule of local SGD with periodic averaging (FedAvg) is as follows.

$$\mathbf{x}_{k+1}^i = \begin{cases} \sum_{i=1}^{m} p_i[\mathbf{x}_k^i - \mu g(\mathbf{x}_k^i)], & k \bmod \tau \text{ is } 0 \\ \mathbf{x}_k^i - \mu g(\mathbf{x}_k^i), & \text{otherwise} \end{cases} \tag{3}$$

Algorithm 1: Local SGD with partial model avg.

Input: Initial parameters $\mathbf{x}_0$, learning rate η, and model averaging interval τ

1 **for** $k = 1$ *to* K **do**
2 A local SGD step: $\mathbf{x}_k^i = \mathbf{x}_{k-1}^i - \mu g(\mathbf{x}_{k-1}^i)$;
3 $j \leftarrow k \bmod \tau$;
4 Average j^{th} subset of the model across all m workers: $\mathbf{u}_{(j,k)} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_{(j,k)}^i$;
5 Each worker updates j^{th} subset of the local model: $\mathbf{x}_{(j,k)}^i = \mathbf{u}_{(j,k)}$;
6 **end**
7 Return $\mathbf{u}_K = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_K^i$

where τ is the model averaging interval and $g(\cdot)$ is a stochastic gradient computed from a random training sample $\boldsymbol{\xi}$. This update rule allows all the workers to independently update their own models for every τ iterations.

Model Discrepancy – Assuming the local optimizers are stochastic optimization methods, the most typical training algorithm for neural network training, all m local models can move toward different directions on parameter space due to the variance of the stochastic gradients. In Federated Learning, the data heterogeneity also makes such an effect more significant. We call the difference between the local models and the global model *model discrepancy*. If the degree of model discrepancy is large, the local models are more likely attracted to different minima adversely affecting the convergence of global loss. Note that synchronous SGD does not have such an issue since it guarantees all the workers always view the same model parameters.

Partial Model Averaging Framework

Algorithm 1 presents local SGD with partial model averaging. Each worker independently runs SGD until the stop condition is satisfied (K iterations). After every SGD step, the algorithm averages a distinct subset of model parameters across all m workers. Each subset consists of $\frac{d}{\tau}$ parameters, where d is the total number of model parameters and τ is the model averaging interval. In this setting, each subset is averaged after every τ iterations. At the end of the training, Algorithm 1 returns the fully-averaged model $\mathbf{u}_K$.

Figure 1 shows schematic illustrations of the periodic averaging (**a**) and the partial averaging (**b**). They show the expected movement of two local models on the parameter space within one communication round ($\tau = 3$). While the periodic averaging allows fully-independent local updates, the partial averaging frequently synchronizes a part of the model parameters suppressing the model discrepancy.

In this work, we use mini-batch SGD as a local solver for simplicity. The framework can be applied to any advanced optimizers by simply changing the parameter update rule at line 2. For instance, FedProx (Li et al. 2020) can be applied by replacing the $g(\mathbf{x}_k^i)$ term with the gradient computed from the FedProx loss function.

Note that Algorithm 1 does not specify how to partition

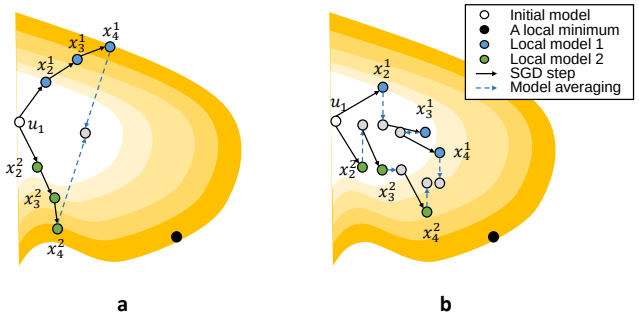


Figure 1: Example illustrations of **a)**: periodic averaging and **b)**: partial averaging with two workers ($\tau = 3$). While the periodic averaging allows fully-independent local updates, the partial averaging frequently synchronize a part of model suppressing the model discrepancy.

the model parameters to τ subsets. As long as the entire parameters are synchronized at least once within τ iterations, it is theoretically guaranteed to have the same maximum bound of the convergence rate. We discuss the impact of the model partitioning on the training results in Appendix.

Convergence Analysis

Preliminaries

Notations – To consider the model partitions in the convergence analysis, we borrow partition-wise notations and assumptions from (You et al. 2019). All vectors in this paper are column vectors. $\mathbf{x} \in \mathbb{R}^d$ denotes the parameters of one local model and m is the number of workers. The model is partitioned into τ subsets such that $\mathbf{x}_j \in \mathbb{R}^{d_j}$ for $j \in \{1, \cdots, \tau\}$, where $\sum_{j=1}^{\tau} d_j = d$. We use $g_j(\mathbf{x}, \xi)$ to denote the gradient of $f(\cdot)$ with respect to $\mathbf{x}_j$, where ξ is a single training sample. For convenience, we use $g_j(\mathbf{x})$ instead. The gradient computed from the whole training samples with respect to $\mathbf{x}_j$ is denoted by $\nabla_j F(\mathbf{x})$. L_j is Lipschitz constant of $f(\cdot)$ with respect to $\mathbf{x}_j$. $L_{\max}$ indicates the maximum Lipschitz constant among all τ model partitions: $\max(L_j), j \in \{1, \cdots, \tau\}$. Likewise, $\sigma^2 = \sum_{j=1}^{\tau} \sigma_j^2$. We provide all the proofs in Appendix.

Convergence Analysis for IID Data

Assumptions – We analyze the convergence rate of Algorithm 1 under the following assumptions.

1. Smoothness: $f(\cdot)$ is L_j-smooth for all $\mathbf{x}_j$;
2. Unbiased gradient: $\mathbb{E}_\xi[g_j(x)] = \nabla_j F(x)$;
3. Bounded variance: $\mathbb{E}_\xi[\|g_j(x) - \nabla_j F(x))\|^2] \leq \sigma_j^2$, where σ_j^2 is a positive constant;

Theorem 1. *Suppose all m local models are initialized to the same point $\mathbf{u}_1$. Under Assumption $1 \sim 3$, if Algorithm 1 runs for K iterations using the learning rate η that satisfies $L_{\max}^2 \eta^2 \tau(\tau-1) + \eta L_{\max} \leq 1$, then the average-squared gradient norm of $\mathbf{u}_k$ is bounded as follows*

$$\mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{2}{\eta K}\mathbb{E}\left[F(\mathbf{u}_1) - F(\mathbf{u}_{inf})\right] + \frac{\eta}{m}\sum_{j=1}^{\tau} L_j \sigma_j^2 + \eta^2(\tau-1)\sum_{j=1}^{\tau} L_j^2 \sigma_j^2 \quad (4)$$

Remark 1. For non-convex smooth objective functions and IID data, local SGD with the partial model averaging ensures the convergence of the model to a stationary point. Particularly, the convergence rate is not dependent on the partition size or the synchronization order across the partitions. That is, as long as the entire model parameters are covered at least once in τ iterations, Algorithm 1 guarantees the convergence.

Remark 2. (linear speedup) If the learning rate $\eta = \frac{\sqrt{m}}{\sqrt{K}}$, the complexity of (4) becomes

$$\mathcal{O}\left(\frac{1}{\sqrt{mK}}\right) + \mathcal{O}\left(\frac{m}{K}\right),$$

where all the constants are removed by $\mathcal{O}$. Thus, if $K > m^3$, the first term dominates the second term achieving linear speedup. Note that the partial averaging has the same complexity of the convergence rate as the periodic averaging method (Wang and Joshi 2018b).

Convergence Analysis for Non-IID Data

For non-IID convergence analysis, we use an assumption on the data heterogeneity that is presented in (Wang et al. 2020).

Assumptions – Our analysis is based on the following assumptions.

1. Smoothness: $f(\cdot)$ is L_j-smooth for all $\mathbf{x}_j$;
2. Unbiased gradient: $\mathbb{E}_s[g_{(i,j)}(x)] = \nabla_j F_i(x)$;
3. Bounded variance: $\mathbb{E}_s[\|g_{(i,j)}(x) - \nabla_j F_i(x))\|^2] \leq \sigma_j^2$, where σ_j^2 is a positive constant;
4. Bounded Dissimilarity: For any sets of weights $\{p_i \geq 0\}_{i=1}^m, \sum_{i=1}^m p_i = 1$, there exist constants $\beta^2 \geq 1$ and $\kappa^2 \geq 0$ such that $\sum_{i=1}^m p_i \|\nabla F_i(\mathbf{x})\|^2 \leq \beta^2 \|\sum_{i=1}^m p_i \nabla F_i(\mathbf{x})\|^2 + \kappa^2$;

Theorem 2. *Suppose all m local models are initialized to the same point $\mathbf{u}_1$. Under Assumption $1 \sim 4$, if Algorithm 1 runs for K iterations and the learning rate satisfies $\eta \leq \frac{1}{L_{\max}} \min\left\{\frac{1}{2}, \frac{1}{\sqrt{2\tau(\tau-1)(2\beta^2+1)}}\right\}$, the average-*

squared gradient norm of $\mathbf{u}_k$ *is bounded as follows*

$$\mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{4}{\eta K}\left(\mathbb{E}\left[F(\mathbf{u}_1) - F(\mathbf{u}_{inf})\right]\right) + 4\eta\sum_{i=1}^{m}p_i^2\sum_{j=1}^{\tau}L_j\sigma_j^2 + 3\eta^2(\tau-1)\sum_{j=1}^{\tau}L_j^2\sigma_j^2 + 6\eta^2\tau(\tau-1)\sum_{j=1}^{\tau}L_j^2\kappa_j^2.$$

Remark 3. For non-convex smooth objective functions and non-IID data, local SGD with the partial model averaging ensures the convergence to a stationary point. Likely to IID data, the partition size or the synchronization order across the partitions do not affect the bound.

Remark 4. (linear speedup) If the learning rate $\eta = \frac{\sqrt{m}}{\sqrt{K}}$ and $p_i = \frac{1}{m}, \forall i \in \{1, \cdots, m\}$, the complexity of the above maximum bound becomes

$$\mathcal{O}\left(\frac{1}{\sqrt{mK}}\right) + \mathcal{O}\left(\frac{m}{K}\right),$$

where all the constants are removed by $\mathcal{O}$. Thus, if $K^3 > m$, the first first term becomes dominant, and it achieves linear speedup. Although the exact bounds cannot be directly compared due to the different assumptions, our analysis shows that the partial averaging method has the same complexity of the convergence rate as the periodic averaging method (Wang et al. 2020).

Impact of Partial Model Averaging on Local Models

We empirically analyze the impact of the partial averaging on the statistical efficiency of local SGD. Figure 2 shows the squared distance between the global model $\mathbf{u}_k$ and the local model $\mathbf{x}_k^i$ averaged across all m workers. The distance is collected from CIFAR-10 (ResNet20) training with $m = 128$ workers. The left chart shows the distance comparison between the periodic averaging and the partial averaging at the first 500 iterations and the right chart shows the comparison in the middle of training (iteration $3000 \sim 3500$). It is clearly observed that the partial averaging effectively suppresses the maximum degree of model discrepancy. While the periodic averaging has a wide spectrum of the distance within each communication round, the partial averaging shows a stable distance across the iterations.

When analyzing the convergence stochastic optimization methods, the difference between the local gradients and the global gradients is usually bounded by the distance between the corresponding model parameters under a smoothness assumption on the objective function. The shorter distance among the models bounds the gradient difference more tightly, and it makes the loss more efficiently converge. We verify such an effect by comparing the local loss and the global loss curves. We collect full-batch training loss of all

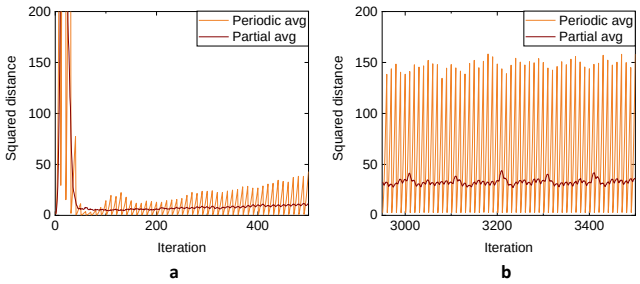


Figure 2: The model discrepancy (the squared distance between the global model $\mathbf{u}_k$ and the local model $\mathbf{x}_k^i$ averaged across all m workers) comparison between the periodic averaging and the partial averaging. The curves are collected from ResNet20 (CIFAR-10) training. **a)**: the curves for the first 500 iterations. **b)**: the curves for the iteration $3000 \sim 3500$.

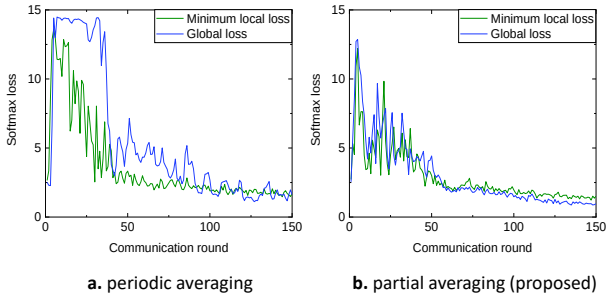


Figure 3: The comparison between the minimum local loss among all the workers and the global loss. The curves are collected from ResNet20 (CIFAR-10) training for 150 communication rounds. **a)**: The full-batch training loss curves of the periodic averaging. We compare the minimum local loss and the global loss curves. **b)**: The same curves of the partial averaging.

individual local models (local loss) and compare it to the loss of the global model (global loss) at the end of each communication round. Figure 3.**a** and 3.**b** show the loss curves of the periodic averaging and the partial averaging, respectively. While the periodic averaging makes the global loss frequently spikes, the partial averaging shows the global loss that goes down more smoothly along with the minimum local loss. Figure 4 shows the loss curves of four different datasets. The partial averaging achieves a faster convergence than the periodic averaging in all the experiments. This empirical analysis demonstrates that the partial averaging accelerates the convergence of the global loss by mitigating the degree of model discrepancy.

Communication Cost

For large-scale deep learning applications on High-Performance Computing (HPC) systems, the fully-distributed communication model is typically used. The most popular communication pattern for model averaging is *allreduce* operation. In Federated Learning, server-client

communication model is more commonly used. Considering the independent and heterogeneous client-side compute nodes, the individual communication pattern (*send* and *receive* operations) can be a better fit for model averaging. Regardless of the communication patterns, the periodic averaging and the partial averaging have the same total communication cost. Given the local model size d, the periodic averaging method requires one inter-process communication of all the d parameters after every τ iterations. The proposed partial averaging performs one communication at every iteration, but only aggregates $\frac{d}{\tau}$ parameters at once. Thus, if τ is the same, the two averaging methods have the same total communication cost.

One potential drawback of the proposed method is the increased number of inter-process communications. While having the same total data size to be transferred, the partial averaging method requires more frequent communications than the periodic full averaging method, and it results in increasing the total latency cost. One may consider adjusting the number of model partitions to reduce the latency cost while degrading the expected classification performance. We consider making a practical trade-off between the latency cost and the statistical efficiency as an important future work.

Experiments

In this section, we present key experimental results that demonstrate efficacy of the partial averaging framework for Federated Learning. Additional experimental results can be found in Appendix.

Experimental Settings

We implemented our experiments using TensorFlow 2.4.0 (Abadi et al. 2015). All the experiments were conducted on a GPU cluster that has four compute nodes each of which has two NVIDIA V100 GPUs. Because of the limited compute resources, we simulate the large-scale local SGD training such that all m local models are distributed to p processes ($m > p$), and each process sequentially trains the given $\frac{m}{p}$ local models. When averaging the parameters, they are aggregated and summed up across the local models owned by each process first, and then reduced across all the processes using MPI communications.

We perform extensive Computer Vision experiments using popular benchmark datasets: CIFAR-10 and CIFAR-100 (Krizhevsky, Hinton et al. 2009), SVHN (Netzer et al. 2011), Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), and Federated Extended MNIST (Caldas et al. 2018). We also run Natural Language Processing (sentiment analysis) experiments using IMDB dataset (Maas et al. 2011). Due to the limited space, the details about the datasets and the model architectures are provided in Appendix. We use momentum SGD with a coefficient of 0.9 and apply gradual warmup (Goyal et al. 2017) to the first 5 epochs to stabilize the training. All the reported performance results are average accuracy across three separate runs. Due to the limited space, we report the final accuracy only and show all the full learning curves in Appendix.

Experiments with IID Data

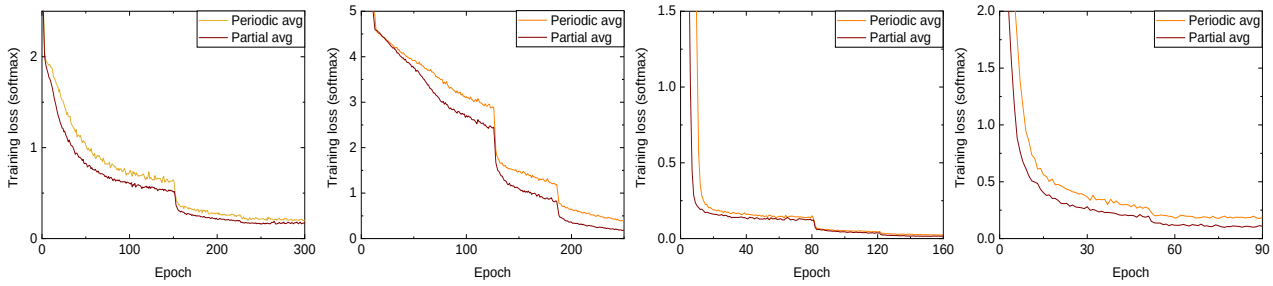
We use the hyper-parameter settings shown in the reference works, and further tune only the learning rate based on a grid search. Table 1 presents our experimental results achieved using 128 workers. The partial averaging achieves a higher validation accuracy than the periodic averaging in all the experiments. This comparison demonstrates that the partial averaging method effectively accelerates the local SGD for IID data. We can also see that the accuracy consistently drops in all the experiments as the averaging interval τ increases. While the larger interval improves the scaling efficiency by reducing the total communication cost, it can harm the statistical efficiency of local SGD.

We also present CIFAR-10 classification results with extended epochs in Table 2. The partial averaging catches up with the synchronous SGD accuracy ($92.63 \pm 0.2\%$) faster than the periodic averaging. We ran synchronous SGD using 128 batch size and 0.1 learning rate for 300 epochs. One insight is that the degree of model discrepancy indeed strongly affects the final accuracy. The synchronous SGD can be considered as a special case where the averaging interval is 1. That is, the degree of model discrepancy is always 0, and thus synchronous SGD achieves a higher accuracy than any local SGD settings. This indirectly explains why the partial averaging achieves a higher accuracy than the periodic averaging. The lower the degree of model discrepancy across the workers, the higher the accuracy.

Experiments with Non-IID Data

Data Heterogeneity Settings – To evaluate the performance of the proposed framework in realistic Federated Learning environments, we also run experiments under two settings: non-IID data and partial device participation. First, we generate synthetic heterogeneous data distributions based on Dirichlet's distribution. We use concentration coefficients of 0.1, 0.5, and 1.0 to evaluate the proposed framework across different degrees of data heterogeneity. Second, we use three different device participation ratios, 25% and 50% (*cross-edge*) and 100% (*cross-silo*). For *cross-edge* Federated Learning settings, we randomly select a subset of the workers for training at every communication round. Note that, since the partial averaging method synchronizes only a subset of parameters at once, extra communications are required to send out the whole local model parameters to other workers at the end of every communication round. To make a fair comparison with respect to the communication cost, we use a 10% longer interval for the partial averaging and re-distribute the local models after every 10 communication rounds. Under this setting, the two averaging methods have a similar total communication cost while the partial averaging has a slightly higher degree of data heterogeneity.

Accuracy Comparison – We fix all the factors that affect the training time: the number of workers, the number of training iterations, and the averaging interval, and then we tune the local batch size and learning rate. Figure 5 shows the loss curves of CIFAR-10 and FEMNIST training. Figure 5**.a** and **c** show the curves for the *cross-edge* settings and **b** and **d** show the curves for the *cross-silo* settings. Regardless of the ratio of participation, the partial averaging



a. CIFAR-10 **b.** CIFAR-100 **c.** SVHN **d.** Fashion-MNIST

Figure 4: (IID data) The training loss comparison between the periodic averaging and the partial averaging across four different datasets: **a.** CIFAR-10 (ResNet-20), **b.** CIFAR-100 (WideResNet-28-10), **c.** SVHN (WideResNet-16-8), and **d.** Fashion-MNIST (VGG-11). 128 workers are used for training.

Table 1: The classification performance comparisons using IID data. The learning rate is fine-tuned based on a grid search for all individual settings.

dataset	model	batch size (LR)	workers	epochs	avg interval	periodic avg	partial avg
CIFAR-10	ResNet20	32 (1.2)	128	300	2	$91.19 \pm 0.2\%$	**91.89** $\pm 0.1\%$
					4	$89.80 \pm 0.2\%$	**90.58** $\pm 0.2\%$
					8	$85.70 \pm 0.3\%$	**88.17** $\pm 0.1\%$
CIFAR-100	WRN28-10	32 (1.2)		250	2	$77.64 \pm 0.2\%$	**79.15** $\pm 0.1\%$
					4	$76.07 \pm 0.2\%$	**77.03** $\pm 0.2\%$
					8	$60.82 \pm 0.2\%$	**62.32** $\pm 0.2\%$
SVHN	WRN16-8	64 (0.2)		160	4	$98.15 \pm 0.1\%$	**98.54**$\pm 0.1\%$
					16	$98.02 \pm 0.2\%$	**98.13**$\pm 0.1\%$
					64	$97.54 \pm 0.1\%$	**97.78**$\pm 0.1\%$
Fasion-MNIST	VGG-11	32 (0.2)		90	2	$92.33 \pm 0.1\%$	**94.01**$\pm 0.1\%$
		32 (0.1)			4	$91.80 \pm 0.1\%$	**93.03**$\pm 0.1\%$
		32 (0.08)			8	$90.48 \pm 0.1\%$	**92.21**$\pm 0.1\%$
IMDB review	LSTM	10 (0.6)		90	2	$88.14 \pm 0.1\%$	**89.22**$\pm 0.1\%$
					4	$88.78 \pm 0.2\%$	**89.27**$\pm 0.1\%$
					8	$88.53 \pm 0.2\%$	**88.74**$\pm 0.3\%$

Table 2: CIFAR-10 classification results with extended training epochs. The partial averaging accuracy catches up with the sync SGD accuracy ($92.63 \pm 0.2\%$) faster than the periodic averaging.

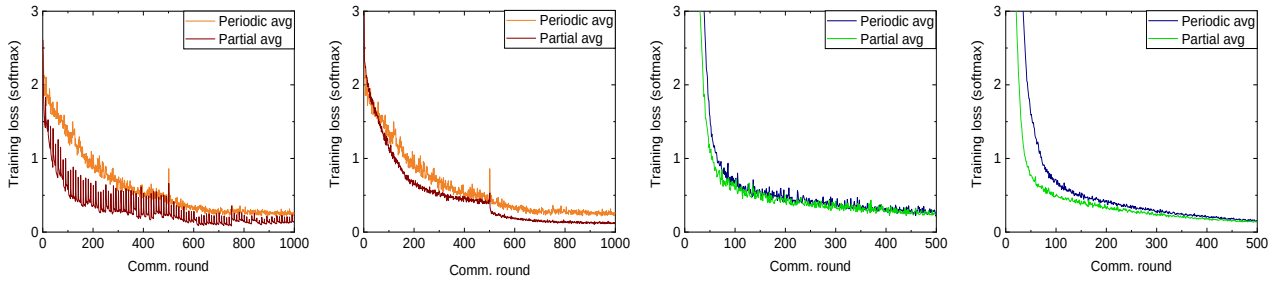
dataset	model	# of workers	avg interval	epochs	periodic avg.	partial avg. (proposed)
CIFAR-10	ResNet20	128	4	300	$89.80 \pm 0.2\%$	**90.58** $\pm 0.2\%$
				400	$90.16 \pm 0.1\%$	**91.70** $\pm 0.2\%$
				500	$91.19 \pm 0.2\%$	**92.20** $\pm 0.1\%$

effectively accelerates the convergence of the training loss. Due to the data heterogeneity, Federated Learning usually requires more iterations to converge than the training in centralized environments. That is, the faster convergence likely results in achieving a higher validation accuracy within a fixed iteration budget.

Table 3 shows our best-tuned hyper-parameter settings and the accuracy results of the three different problems (CIFAR-10 and FEMNIST classifications and IMDB sentiment analysis). Note that we set $\alpha \geq 0.5$ for IMDB because the Dirichlet's concentration coefficient lower than 0.5 makes some workers not assigned with any training samples. Given a fixed iteration budget, as expected, the partial averaging achieves a higher accuracy than the periodic averaging in all the experiments. These results verifies that the partial averaging effectively mitigates the adverse impact of the model discrepancy on the global loss convergence in non-IID settings.

Related Work

Post-local SGD – Lin et al. proposed *post-local* SGD in (Lin et al. 2018). The algorithm begins the training with a single worker and then increases the number of workers once the learning rate is decayed. This approach makes the model converge much faster than pure local SGD because the training does not suffer from the model discrepancy in the early training epochs. However, it significantly undermines the degree of parallelism making it less practical. The authors



a. CIFAR-10 (cross-edge) **b.** CIFAR-10 (cross-silo) **c.** FEMNIST (cross-edge) **d.** FEMNIST (cross-silo)

Figure 5: (Non-IID data) The training loss comparison between the periodic averaging and the partial averaging across two different datasets: **a.** CIFAR-10 (*cross-edge*), **b.** CIFAR-10 (*cross-silo*), **c.** FEMNIST (*cross-edge*), and **d.** FEMNIST (*cross-silo*). In the *cross-edge* settings, 25% of random workers participate in training.

Table 3: Classification experiments using non-IID data. We conduct the experiments with various degrees of data heterogeneity (Dir(α)) and device selection ratio settings. The ResNet20 is trained for $10,000$ iterations. The LSTM and CNN are trained for $2,000$ iterations.

dataset	batch size (LR)	workers	avg interval	active ratio	Dir(α)	periodic avg	partial avg
CIFAR-10 (ResNet20)	32 (0.4)	128	10	100%	1	$90.38 \pm 0.1\%$	**91.54** $\pm 0.1\%$
					0.5	$90.18 \pm 0.1\%$	**91.56** $\pm 0.1\%$
					0.1	$89.92 \pm 0.2\%$	**91.31** $\pm 0.1\%$
				50%	1	$89.98 \pm 0.2\%$	**90.61** $\pm 0.2\%$
					0.5	$89.51 \pm 0.3\%$	**91.02** $\pm 0.3\%$
					0.1	$88.99 \pm 0.3\%$	**90.64** $\pm 0.2\%$
				25%	1	$89.32 \pm 0.3\%$	**91.00** $\pm 0.2\%$
					0.5	$88.73 \pm 0.4\%$	**90.16** $\pm 0.3\%$
	32 (0.2)				0.1	$87.70 \pm 0.4\%$	**88.95** $\pm 0.3\%$
IMDB reviews (LSTM)	10 (0.4)	128	10	100%	1	$88.03 \pm 0.2\%$	**88.68** $\pm 0.2\%$
					0.5	$87.72 \pm 0.2\%$	**88.40** $\pm 0.3\%$
	10 (0.2)			50%	1	$83.79 \pm 0.3\%$	**85.83** $\pm 0.3\%$
					0.5	$83.00 \pm 0.2\%$	**84.82** $\pm 0.2\%$
	10 (0.1)			25%	1	$81.13 \pm 0.3\%$	**83.40** $\pm 0.2\%$
					0.5	$80.02 \pm 0.2\%$	**82.01** $\pm 0.3\%$
FEMNIST	32 (0.1)	128	4	100%	-	$83.93 \pm 0.4\%$	**85.34** $\pm 0.3\%$
				50%	-	$85.27 \pm 0.3\%$	**85.81** $\pm 0.1\%$
	32 (0.05)			25%	-	$85.73 \pm 0.2\%$	**85.90** $\pm 0.1\%$

use up to 16 workers for training and achieve a comparable accuracy to that of synchronous SGD.

Variance Reduced Stochastic Methods – Variance reduced stochastic methods, such as SVRG (Johnson and Zhang 2013) or SAGA (Defazio, Bach, and Lacoste-Julien 2014), are known to improve the convergence rate of SGD. Recently, Liang et al. successfully applied the variance reduction technique to local SGD (Liang et al. 2019). Despite the faster convergence thanks to the reduced stochastic variance, the variance reduction techniques are known to harm the generalization performance (Defazio and Bottou 2018). This limitation is aligned with the fact that the convergence under a low noise condition can adversely affect the generalization performance (Li, Wei, and Ma 2019; Lewkowycz et al. 2020).

We implemented VRL-SGD (Liang et al. 2019) using TensorFlow based on the open-source[1] of the reference work. Due to the limited space, we provide the detailed experimental results in Appendix. While VRL-SGD effectively accelerates the convergence of the training loss, we could observe a non-negligible gap of the validation accuracy between local SGD with the partial averaging and VRL-SGD. Note that VRL-SGD performs extra computations to obtain average gradient deviations while our proposed model aggregation scheme does not have such computations.

Adaptive Model Averaging Interval – Some researchers have proposed adaptive model averaging interval methods (Wang and Joshi 2018a; Haddadpour et al. 2019). The common principle behind these works is that the communication cost of model averaging can be reduced by adjusting the averaging frequency based on the training progress at run-time. The proposed partial averaging method is readily applicable to these adaptive interval methods because the proposed method is not dependent on any interval settings. One can expect even better scaling efficiency if the largest averaging

[1] https://github.com/zerolxf/VRL-SGD

interval for each part of the model can be found.

Conclusion

We proposed a partial model averaging framework for Federated Learning. Our analysis and experimental results demonstrate the efficacy of the partial averaging in large-scale local SGD. The proposed framework is readily applicable to any Federated Learning applications. Breaking the conventional assumption of periodic model averaging can considerably broaden potential design options for Federated Learning algorithms. We consider harmonizing the partial averaging with many existing advanced Federated Learning algorithms such as FedProx, FedNova, and SCAFFOLD as a critical future work.

Societal Impacts – This research work does not have any potential adverse impacts on society. Our study aims to improve the statistical efficiency of Federated Learning algorithms making better use of the provided hardware resources. Consequently, large-scale Federated Learning applications may finish the neural network training faster, and it can results in reducing the CO2 footprint. The faster training also may reduce the electricity consumption.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; and Zheng, X. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org.

Caldas, S.; Duddu, S. M. K.; Wu, P.; Li, T.; Konečnỳ, J.; McMahan, H. B.; Smith, V.; and Talwalkar, A. 2018. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*.

Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *arXiv preprint arXiv:1407.0202*.

Defazio, A.; and Bottou, L. 2018. On the ineffectiveness of variance reduced optimization for deep learning. *arXiv preprint arXiv:1812.04529*.

Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; and He, K. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

Haddadpour, F.; Kamani, M. M.; Mahdavi, M.; and Cadambe, V. R. 2019. Local sgd with periodic averaging: Tighter analysis and adaptive synchronization. *arXiv preprint arXiv:1910.13598*.

Johnson, R.; and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26: 315–323.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. SCAFFOLD: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, 5132–5143. PMLR.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Lewkowycz, A.; Bahri, Y.; Dyer, E.; Sohl-Dickstein, J.; and Gur-Ari, G. 2020. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems*, volume 2, 429–450.

Li, Y.; Wei, C.; and Ma, T. 2019. Towards explaining the regularization effect of initial large learning rate in training neural networks. *arXiv preprint arXiv:1907.04595*.

Liang, X.; Shen, S.; Liu, J.; Pan, Z.; Chen, E.; and Cheng, Y. 2019. Variance reduced local SGD with lower communication complexity. *arXiv preprint arXiv:1912.12844*.

Lin, T.; Stich, S. U.; Patel, K. K.; and Jaggi, M. 2018. Don't Use Large Mini-Batches, Use Local SGD. *arXiv preprint arXiv:1808.07217*.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142–150. Portland, Oregon, USA: Association for Computational Linguistics.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, 1273–1282. PMLR.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Robbins, H.; and Monro, S. 1951. A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Stich, S. U. 2018. Local SGD converges fast and communicates little. *arXiv preprint arXiv:1805.09767*.

Wang, J.; and Joshi, G. 2018a. Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD. *arXiv preprint arXiv:1810.08313*.

Wang, J.; and Joshi, G. 2018b. Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms. *arXiv preprint arXiv:1808.07576*.

Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; and Poor, H. V. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. *arXiv preprint arXiv:2007.07481*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

You, Y.; Li, J.; Reddi, S.; Hseu, J.; Kumar, S.; Bhojanapalli, S.; Song, X.; Demmel, J.; Keutzer, K.; and Hsieh, C.-J. 2019. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*.

Yu, H.; Jin, R.; and Yang, S. 2019. On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In *International Conference on Machine Learning*, 7184–7193. PMLR.

Yu, H.; Yang, S.; and Zhu, S. 2019. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 5693–5700.

Appendix

Preliminaries

Herein, we provide the proofs of all Theorems and Lemmas presented in the paper.

Notations – We first define a few notations for our analysis.

- m: the number of local models (clients)
- K: the number of total training iterations
- τ: the model averaging interval
- $\mathbf{x}^i_{(j,k)}$: a local model partition j of client i at iteration k
- $\mathbf{u}_{(j,k)}$: a model partition j averaged across all m clients at iteration k
- $\mathbf{g}^i_{(j,k)}$: a stochastic gradient of client i with respect to the local model partition j at iteration k
- $\nabla_j F_i(\cdot)$: a local full-batch gradient of client i with respect to the model partition j
- L_{max}: the maximum Lipschitz constant across all τ model partitions; $L_{max} = max(L_j), j \in \{1, \cdots, \tau\}$

Vectorization – We further define a vectorized form of the local model partition and its gradients as follows.

$$\mathbf{x}_{(j,k)} = vec(\mathbf{x}^1_{(j,k)}, \mathbf{x}^2_{(j,k)}, \cdots, \mathbf{x}^m_{(j,k)}) \tag{5}$$

$$\mathbf{g}_{(j,k)} = vec(\mathbf{g}^1_{(j,k)}, \mathbf{g}^2_{(j,k)}, \cdots, \mathbf{g}^m_{(j,k)}) \tag{6}$$

$$\mathbf{f}_{(j,k)} = vec(\nabla_j F_1(\mathbf{x}^1_k), \nabla_j F_2(\mathbf{x}^2_k), \cdots, \nabla_j F_m(\mathbf{x}^m_k)) \tag{7}$$

Averaging Matrix – We first define a full-averaging matrix $\mathbf{J}_j$ for each model partition j as follows.

$$\mathbf{J}_j = \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^\top \otimes \mathbf{I}_{d_j}, \qquad j \in \{1, \cdots, \tau\}, \tag{8}$$

where $\otimes$ indicates Kronecker product, $\mathbf{1}_m \in \mathbb{R}^m$ is a vector of ones, and $\mathbf{I}_{d_j} \in \mathbb{R}^{d_j \times d_j}$ is an identity matrix.

Then, we define a time-varying partial-averaging matrix for each model partition $\mathbf{P}_{(j,k)} \in \mathbb{R}^{md_j \times md_j}$ for $j \in \{1, \cdots, \tau\}$.

$$\mathbf{P}_{(j,k)} = \begin{cases} \mathbf{J}_j & \text{if } k \bmod \tau \text{ is } j \\ \mathbf{I}_j & \text{if } k \bmod \tau \text{ is not } j, \end{cases} \tag{9}$$

where $\mathbf{I}_j$ is an identity matrix of size $md_j \times md_j$.

Here we present an example where $d = 3$, $\tau = 2$, and $m = 2$. If $d_0 = 2$ and $d_1 = 1$, we have $\mathbf{P}_{(j,k)}$ as follows.

$$\mathbf{P}_{(0,0)} = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}, \mathbf{P}_{(0,1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{10}$$

$$\mathbf{P}_{(1,0)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{P}_{(1,1)} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}. \tag{11}$$

For instance, if $j = 0$ and (k mod τ) is j, the first partition of the model is averaged by multiplying $\mathbf{P}_{(0,k)}$ by $\mathbf{x}_{(0,k)}$ as follows.

$$\mathbf{P}_{(0,k)}\mathbf{x}_{(0,k)} = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} x^{(0,0)} \\ x^{(0,1)} \\ x^{(1,0)} \\ x^{(1,1)} \end{bmatrix} = \begin{bmatrix} (x^{(0,0)} + x^{(1,0)})/2 \\ (x^{(0,1)} + x^{(1,1)})/2 \\ (x^{(0,0)} + x^{(1,0)})/2 \\ (x^{(0,1)} + x^{(1,1)})/2 \end{bmatrix}, \tag{12}$$

where $x^{(i,j)}$ indicates the parameter j of worker i.

The averaging matrix $\mathbf{P}_{(j,k)}$ has the following properties.

1. $\mathbf{P}_{(j,k)}\mathbf{1}_{md_j} = \mathbf{1}_{md_j}, \forall j \in \{1, \cdots, \tau\}$.
2. $\mathbf{J}_j\mathbf{P}_{(j,k)}$ is $\mathbf{J}_j, \forall k \in \{1, \cdots, K\}$.
3. $\mathbf{P}_{(j,k)}\mathbf{P}_{(j,k')}$ is symmetric, $\forall k, k' \in \{1, \cdots, K\}$.
4. $\mathbf{P}_{(j,k)}\mathbf{P}_{(j,k')}$ consists only of $m \times m$ diagonal blocks, $\forall k, k' \in \{1, \cdots, K\}$.

Using the vectorized form of the parameters (5) and gradients (6), the parameter update rule of FedAvg is

$$\mathbf{x}^i_k = \mathbf{P}_k(\mathbf{x}^i_{k-1} - \eta\mathbf{g}^i_{k-1}), \tag{13}$$

where η is the learning rate.

Convergence Analysis for IID Data

Now, we provide the proof of main Theorem under IID data settings.

Proof of Theorem 1

Theorem 1. *Suppose all m local models are initialized to the same point $\mathbf{u}_1$. Under Assumption $1 \sim 3$, if Algorithm 1 runs for K iterations using the learning rate η that satisfies $L_{\max}^2\eta^2\tau(\tau-1)+\eta L_{\max} \leq 1$, then the average-squared gradient norm of $\mathbf{u}_k$ is bounded as follows*

$$\mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{2}{\eta K}\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{inf})\right]+\frac{\eta}{m}\sum_{j=1}^{\tau}L_j\sigma_j^2+\eta^2(\tau-1)\sum_{j=1}^{\tau}L_j^2\sigma_j^2 \tag{14}$$

Proof. Based on Lemma **??** and **??**, we have

$$\begin{aligned}\mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{2}{\eta K}\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{inf})\right]+\sum_{j=1}^{\tau}\frac{\eta L_j\sigma_j^2}{m}+\sum_{j=1}^{\tau}\left(\frac{\eta L_j-1}{mK}\sum_{i=1}^{m}\sum_{k=1}^{K}\|\nabla_j F(\mathbf{x}_k^i)\|^2\right)\\ +\sum_{j=1}^{\tau}L_j^2\left(\eta^2(\tau-1)\sigma_j^2+\frac{\eta^2\tau(\tau-1)}{mK}\sum_{i=1}^{m}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla_j F(\mathbf{x}_k^i)\right\|^2\right]\right)\end{aligned} \tag{15}$$

After a minor rearrangement, we have

$$\begin{aligned}\mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{2}{\eta K}\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{inf})\right]+\sum_{j=1}^{\tau}\frac{\eta L_j\sigma_j^2}{m}+\sum_{j=1}^{\tau}L_j^2\eta^2(\tau-1)\sigma_j^2\\ +\sum_{j=1}^{\tau}\left(\frac{L_j^2\eta^2\tau(\tau-1)+\eta L_j-1}{mK}\sum_{i=1}^{m}\sum_{k=1}^{K}\|\nabla_j F(\mathbf{x}_k^i)\|^2\right)\end{aligned} \tag{16}$$

If the learning rate η satisfies $L_{\max}^2\eta^2\tau(\tau-1)+\eta L_{\max} \leq 1$, we have

$$\begin{aligned}\mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] &\leq \frac{2}{\eta K}\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{inf})\right]+\sum_{j=1}^{\tau}\frac{\eta L_j\sigma_j^2}{m}+\sum_{j=1}^{\tau}L_j^2\eta^2(\tau-1)\sigma_j^2\\ &= \frac{2}{\eta K}\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{inf})\right]+\frac{\eta}{m}\sum_{j=1}^{\tau}L_j\sigma_j^2+\eta^2(\tau-1)\sum_{j=1}^{\tau}L_j^2\sigma_j^2\end{aligned} \tag{17}$$

We finish the proof. □

Proof of Lemma 1

Lemma 1. *(framework) Under Assumption $1 \sim 3$, if $\eta \leq \frac{1}{L}$, Algorithm 1 ensures*

$$\mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{2}{\eta K}\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{inf})\right]+\frac{\eta}{m}\sum_{j=1}^{\tau}L_j\sigma_j^2+\frac{1}{mK}\sum_{i=1}^{m}\sum_{k=1}^{K}\sum_{j=1}^{\tau}L_j^2\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^i\|^2 \tag{18}$$

Proof. Based on Assumption 1, we have

$$\mathbb{E}\left[F(\mathbf{u}_{k+1})-F(\mathbf{u}_k)\right] \leq -\eta\sum_{j=1}^{\tau}\mathbb{E}\left[\langle\nabla_j F(\mathbf{u}_k),\frac{1}{m}\sum_{i=1}^{m}\mathbf{g}_{(j,k)}^i\rangle\right]+\sum_{j=1}^{\tau}\frac{\eta^2 L_j}{2}\left(\mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^{m}\mathbf{g}_{(j,k)}^i\right\|^2\right]\right). \tag{19}$$

The first term on the right-hand side in (19) can be re-written as follows.

$$-\eta\sum_{j=1}^{\tau}\left(\mathbb{E}\left[\langle\nabla_j F(\mathbf{u}_k),\frac{1}{m}\sum_{i=1}^{m}\mathbf{g}_{(j,k)}^i\rangle\right]\right) = -\eta\sum_{j=1}^{\tau}\left(\frac{1}{m}\sum_{i=1}^{m}\langle\nabla_j F(\mathbf{u}_k),\nabla_j F(\mathbf{x}_k^i)\rangle\right) \tag{20}$$

$$= -\eta\sum_{j=1}^{\tau}\left(\frac{1}{2m}\sum_{i=1}^{m}\left(\|\nabla_j F(\mathbf{u}_k)\|^2+\|\nabla_j F(\mathbf{x}_k^i)\|^2-\|\nabla_j F(\mathbf{u}_k)-\nabla_j F(\mathbf{x}_k^i)\|^2\right)\right) \tag{21}$$

$$= -\eta\sum_{j=1}^{\tau}\left(\frac{1}{2}\|\nabla_j F(\mathbf{u}_k)\|^2+\frac{1}{2m}\sum_{i=1}^{m}\|\nabla_j F(\mathbf{x}_k^i)\|^2-\frac{1}{2m}\sum_{i=1}^{m}\|\nabla_j F(\mathbf{u}_k)-\nabla_j F(\mathbf{x}_k^i)\|^2\right), \tag{22}$$

where (21) holds based on a basic equality: $2a^\top b = \|a\|^2 + \|b\|^2 - \|a-b\|^2$.

The second term on the right-hand side in (19) is bounded as follows.

$$
\begin{aligned}
&\sum_{j=1}^{\tau} \frac{\eta^2 L_j}{2} \left(\mathbb{E}\left[\left\| \frac{1}{m} \sum_{i=1}^{m} \mathbf{g}_{(j,k)}^i \right\|^2 \right] \right) \\
&\quad = \sum_{j=1}^{\tau} \frac{\eta^2 L_j}{2} \left(\mathbb{E}\left[\left\| \frac{1}{m} \sum_{i=1}^{m} \mathbf{g}_{(j,k)}^i - \mathbb{E}\left[\frac{1}{m} \sum_{i=1}^{m} \mathbf{g}_{(j,k)}^i \right] \right\|^2 \right] + \left\| \mathbb{E}\left[\frac{1}{m} \sum_{i=1}^{m} \mathbf{g}_{(j,k)}^i \right] \right\|^2 \right) \qquad (23) \\
&\quad = \sum_{j=1}^{\tau} \frac{\eta^2 L_j}{2} \left(\mathbb{E}\left[\left\| \frac{1}{m} \sum_{i=1}^{m} \mathbf{g}_{(j,k)}^i - \frac{1}{m} \sum_{i=1}^{m} \nabla_j F(\mathbf{x}_k^i) \right\|^2 \right] + \left\| \frac{1}{m} \sum_{i=1}^{m} \nabla_j F(\mathbf{x}_k^i) \right\|^2 \right) \\
&\quad = \sum_{j=1}^{\tau} \frac{\eta^2 L_j}{2} \left(\frac{1}{m^2} \sum_{i=1}^{m} \mathbb{E}\left[\left\| \mathbf{g}_{(j,k)}^i - \nabla_j F(\mathbf{x}_k^i) \right\|^2 \right] + \left\| \frac{1}{m} \sum_{i=1}^{m} \nabla_j F(\mathbf{x}_k^i) \right\|^2 \right) \qquad (24) \\
&\quad \leq \sum_{j=1}^{\tau} \frac{\eta^2 L_j}{2} \left(\frac{1}{m^2} \sum_{i=1}^{m} \mathbb{E}\left[\left\| \mathbf{g}_{(j,k)}^i - \nabla_j F(\mathbf{x}_k^i) \right\|^2 \right] + \frac{1}{m} \sum_{i=1}^{m} \left\| \nabla_j F(\mathbf{x}_k^i) \right\|^2 \right) \qquad (25) \\
&\quad \leq \sum_{j=1}^{\tau} \frac{\eta^2 L_j}{2} \left(\frac{\sigma_j^2}{m} + \frac{1}{m} \sum_{i=1}^{m} \left\| \nabla_j F(\mathbf{x}_k^i) \right\|^2 \right), \qquad (26)
\end{aligned}
$$

where (23) follows a basic equality: $\mathbb{E}[\|\mathbf{x}\|^2] = \mathbb{E}[\|\mathbf{x} - \mathbb{E}[\mathbf{x}]\|^2] + \|\mathbb{E}[\mathbf{x}]\|^2$ for any random vector $\mathbf{x}$. (24) holds because $\mathbf{g}_{(j,k)}^i - \nabla_j F(\mathbf{x}_k^i)$ has 0 mean and is independent across i. (25) holds based on the convexity of ℓ_2 norm and Jensen's inequality. Then, plugging in (22) and (26) into (19), we have

$$
\begin{aligned}
&\mathbb{E}\left[F(\mathbf{u}_{k+1}) - F(\mathbf{u}_k)\right] \\
&\quad \leq -\eta \sum_{j=1}^{\tau} \left(\frac{1}{2} \|\nabla_j F(\mathbf{u}_k)\|^2 + \frac{1}{2m} \sum_{i=1}^{m} \|\nabla_j F(\mathbf{x}_k^i)\|^2 - \frac{1}{2m} \sum_{i=1}^{m} \|\nabla_j F(\mathbf{u}_k) - \nabla_j F(\mathbf{x}_k^i)\|^2 \right) \\
&\qquad + \sum_{j=1}^{\tau} \frac{\eta^2 L_j}{2} \left(\frac{\sigma_j^2}{m} + \frac{1}{m} \sum_{i=1}^{m} \left\| \nabla_j F(\mathbf{x}_k^i) \right\|^2 \right) \\
&\quad = \sum_{j=1}^{\tau} \left(-\frac{\eta}{2} \|\nabla_j F(\mathbf{u}_k\|^2 - \frac{\eta}{2m} \sum_{i=1}^{m} \|\nabla_j F(\mathbf{x}_k^i)\|^2 + \frac{\eta}{2m} \sum_{i=1}^{m} \|\nabla_j F(\mathbf{u}_k) - \nabla_j F(\mathbf{x}_k^i)\|^2 \right) \\
&\qquad + \sum_{j=1}^{\tau} \left(\frac{\eta^2 L_j \sigma_j^2}{2m} + \frac{\eta^2 L_j}{2m} \sum_{i=1}^{m} \|\nabla_j F(\mathbf{x}_k^i)\|^2 \right) \qquad (27)
\end{aligned}
$$

After dividing both sides of (27) by $\frac{\eta}{2}$ and rearranging, we have

$$
\begin{aligned}
\sum_{j=1}^{\tau} \|\nabla_j F(\mathbf{u}_k)\|^2 \leq \frac{2}{\eta} \mathbb{E}\left[F(\mathbf{u}_k) - F(\mathbf{u}_{k+1})\right] + \sum_{j=1}^{\tau} \frac{\eta L_j \sigma_j^2}{m} + \sum_{j=1}^{\tau} \left(\frac{\eta L_j - 1}{m} \sum_{i=1}^{m} \|\nabla_j F(\mathbf{x}_k^i)\|^2 \right) \\
+ \sum_{j=1}^{\tau} \left(\frac{1}{m} \sum_{i=1}^{m} \|\nabla_j F(\mathbf{u}_k) - \nabla_j F(\mathbf{x}_k^i)\|^2 \right) \qquad (28)
\end{aligned}
$$

Taking expectation on both sides of (28) and averaging it over K iterations, we have

$$\begin{aligned}\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\left(\sum_{j=1}^{\tau}\|\nabla_j F(\mathbf{u}_k)\|^2\right)\right] \leq & \frac{2}{\eta K}\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{k+1})\right]+\sum_{j=1}^{\tau}\frac{\eta L_j\sigma_j^2}{m} \\ & +\sum_{j=1}^{\tau}\left(\frac{\eta L_j-1}{mK}\sum_{i=1}^{m}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla_j F(\mathbf{x}_k^i)\|^2\right]\right) \\ & +\sum_{j=1}^{\tau}\left(\frac{1}{mK}\sum_{i=1}^{m}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla_j F(\mathbf{u}_k)-\nabla_j F(\mathbf{x}_k^i)\|^2\right]\right) \end{aligned} \tag{29}$$

If $\eta \leq \frac{1}{L_{max}}$, then

$$\begin{aligned}\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\left(\sum_{j=1}^{\tau}\|\nabla_j F(\mathbf{u}_k)\|^2\right)\right] &\leq \frac{2}{\eta K}\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{k+1})\right]+\sum_{j=1}^{\tau}\frac{\eta L_j\sigma_j^2}{m} \\ &\quad +\sum_{j=1}^{\tau}\left(\frac{1}{mK}\sum_{i=1}^{m}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla_j F(\mathbf{u}_k)-\nabla_j F(\mathbf{x}_k^i)\|^2\right]\right) \\ &\leq \frac{2}{\eta K}\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{k+1})\right]+\sum_{j=1}^{\tau}\frac{\eta L_j\sigma_j^2}{m} \\ &\quad +\frac{1}{mK}\sum_{i=1}^{m}\sum_{k=1}^{K}\left(\sum_{j=1}^{\tau}L_j^2\,\mathbb{E}\left[\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^i\|^2\right]\right), \end{aligned} \tag{30}$$

where (30) holds based on Assumption 1. Finally, summing up the gradients of τ model partitions, we have

$$\begin{aligned}\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] &\leq \frac{2}{\eta K}\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{k+1})\right]+\sum_{j=1}^{\tau}\frac{\eta L_j\sigma_j^2}{m}+\frac{1}{mK}\sum_{i=1}^{m}\sum_{k=1}^{K}\sum_{j=1}^{\tau}L_j^2\,\mathbb{E}\left[\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^i\|^2\right] \\ &\leq \frac{2}{\eta K}\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{inf})\right]+\frac{\eta}{m}\sum_{j=1}^{\tau}L_j\sigma_j^2+\frac{1}{mK}\sum_{i=1}^{m}\sum_{k=1}^{K}\sum_{j=1}^{\tau}L_j^2\,\mathbb{E}\left[\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^i\|^2\right] \end{aligned}$$

We complete the proof.

□

Proof of Lemma 2

Lemma 2. *(model discrepancy) Under Assumption* $1 \sim 3$*, Algorithm 1 ensures*

$$\begin{aligned}&\frac{1}{mK}\sum_{k=1}^{K}\sum_{i=1}^{m}\sum_{j=1}^{\tau}L_j^2\,\mathbb{E}\left[\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^i\|^2\right] \\ &\quad\leq \sum_{j=1}^{\tau}L_j^2\left(\eta^2(\tau-1)\sigma_j^2+\frac{\eta^2\tau(\tau-1)}{mK}\sum_{i=1}^{m}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla_j F(\mathbf{x}_k^i)\right\|^2\right]\right).\end{aligned}$$

Proof. The averaged distance of each partition j can be re-written using the vectorized form of the parameters as follows.

$$\sum_{i=1}^{m}\left\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^i\right\|^2=\left\|\mathbf{J}_j\mathbf{x}_{(j,k)}-\mathbf{x}_{(j,k)}\right\|^2=\left\|(\mathbf{J}_j-\mathbf{I}_j)\mathbf{x}_{(j,k)}\right\|^2. \tag{31}$$

According to the parameter update rule, we have

$$\begin{aligned}(\mathbf{J}_j-\mathbf{I}_j)\mathbf{x}_{(j,k)} &= (\mathbf{J}_j-\mathbf{I}_j)\mathbf{P}_{(j,k-1)}(\mathbf{x}_{(j,k-1)}-\eta\mathbf{g}_{(j,k-1)}) \\ &= (\mathbf{J}_j-\mathbf{I}_j)\mathbf{P}_{(j,k-1)}\mathbf{x}_{(j,k-1)}-(\mathbf{J}_j-\mathbf{P}_{(j,k-1)})\eta\mathbf{g}_{(j,k-1)},\end{aligned} \tag{32}$$

where the second equality holds because $\mathbf{J}_j\mathbf{P}_j=\mathbf{J}_j$ and $\mathbf{I}_j\mathbf{P}_j=\mathbf{P}_j$.

Then, expanding the expression of $\mathbf{x}_{(j,k-1)}$, we have

$$\begin{aligned}(\mathbf{J}_j - \mathbf{I}_j)\mathbf{x}_{(j,k)} &= (\mathbf{J}_j - \mathbf{I}_j)\mathbf{P}_{(j,k-1)}(\mathbf{P}_{(j,k-2)}(\mathbf{x}_{(j,k-2)} - \eta\mathbf{g}_{(j,k-2)})) - (\mathbf{J}_j - \mathbf{P}_{(j,k-1)})\eta\mathbf{g}_{(j,k-1)} \\ &= (\mathbf{J}_j - \mathbf{I}_j)\mathbf{P}_{(j,k-1)}\mathbf{P}_{(j,k-2)}\mathbf{x}_{(j,k-2)} - (\mathbf{J}_j - \mathbf{P}_{(j,k-1)}\mathbf{P}_{(j,k-2)})\mu\mathbf{g}_{(j,k-2)} \\ &\quad - (\mathbf{J}_j - \mathbf{P}_{(j,k-1)})\mu\mathbf{g}_{(j,k-1)}. \end{aligned} \tag{33}$$

Repeating the same procedure for $\mathbf{x}_{(j,k-2)}, \mathbf{x}_{(j,k-3)}, \cdots, \mathbf{x}_{(j,2)}$, we have

$$\begin{aligned}(\mathbf{J}_j - \mathbf{I}_j)\mathbf{x}_{(j,k)} &= (\mathbf{J}_j - \mathbf{I}_j)\prod_{s=1}^{k-1}\mathbf{P}_{(j,s)}\mathbf{x}_{(j,1)} - \eta\sum_{s=1}^{k-1}(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)})\mathbf{g}_{(j,s)} \\ &= -\eta\sum_{s=1}^{k-1}(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)})\mathbf{g}_{(j,s)}, \end{aligned} \tag{34}$$

where the second equality holds since $\mathbf{x}^i_{(j,1)}$ is all the same among m workers and thus $(\mathbf{J}_j - \mathbf{I}_j)\mathbf{x}_{(j,1)}$ is 0.

Then, we have

$$\begin{aligned}&\frac{1}{mK}\sum_{k=1}^{K}\sum_{i=1}^{m}\sum_{j=1}^{\tau}L_j^2\,\mathbb{E}[\|\mathbf{u}_{(j,k)} - \mathbf{x}^i_{(j,k)}\|^2] \\ &= \frac{1}{mK}\sum_{k=1}^{K}\sum_{j=1}^{\tau}L_j^2\,\mathbb{E}[\|(\mathbf{J}_j - \mathbf{I}_j)\mathbf{x}_{(j,k)}\|^2] \\ &= \frac{\eta^2}{mK}\sum_{k=1}^{K}\sum_{j=1}^{\tau}L_j^2\,\mathbb{E}[\|\sum_{s=1}^{k-1}(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)})\mathbf{g}_{(j,s)}\|^2] \\ &= \frac{\eta^2}{mK}\sum_{k=1}^{K}\sum_{j=1}^{\tau}L_j^2\,\mathbb{E}\left[\left\|\sum_{s=1}^{k-1}(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)})(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}) + \sum_{s=1}^{k-1}(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)})\mathbf{f}_{(j,s)}\right\|^2\right] \\ &\le \frac{2\eta^2}{mK}\sum_{j=1}^{\tau}L_j^2\left(\underbrace{\sum_{k=1}^{K}\mathbb{E}\left[\left\|\sum_{s=1}^{k-1}(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)})(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)})\right\|^2\right]}_{T_1} + \underbrace{\sum_{k=1}^{K}\mathbb{E}\left[\left\|\sum_{s=1}^{k-1}(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)})\mathbf{f}_{(j,s)}\right\|^2\right]}_{T_2}\right), \end{aligned} \tag{35}$$

where the last inequality holds based on a basic inequality: $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$. Now we focus on bounding the above two terms, T_1 and T_2, separately.

Bounding T_1 – We first partition K iterations into three subsets: the first j iterations, the next $K-\tau$ iterations, and the final $\tau - j$ iterations. We bound the first j iterations as follows.

$$\begin{aligned}\sum_{k=1}^{j}\mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right)\right\|^2\right] &= \sum_{k=1}^{j}\sum_{s=1}^{k-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right)\right\|^2\right] && (36) \\ &= \sum_{k=1}^{j}\sum_{s=1}^{k-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j - \mathbf{I}_j\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right)\right\|^2\right] && (37) \\ &\le \sum_{k=1}^{j}\sum_{s=1}^{k-1}\mathbb{E}\left[\left\|\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right\|^2\right] && (38) \\ &= \sum_{k=1}^{j}\sum_{s=1}^{k-1}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\mathbf{g}^i_{(j,s)} - \nabla_j F_i(\mathbf{x}^i_s)\right\|^2\right] \\ &\le \sum_{k=1}^{j}\sum_{s=1}^{k-1}m\sigma_j^2 = \frac{j(j-1)}{2}m\sigma_j^2, && (39)\end{aligned}$$

where (36) holds because $\mathbf{g}_{(j,s)} - \nabla_j F(\mathbf{x}_s)$ has a mean of 0 and independent across s; (37) holds because $\prod_{l=s}^{k-1} \mathbf{P}_{(j,l)}$ is $\mathbf{I}_j$ when $k < j$; (38) holds based on Lemma 2.

Then, the next $K - \tau$ iterations of T_1 are bounded as follows.

$$\sum_{k=j+1}^{K-\tau+j} \mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right)\right\|^2\right] = \sum_{k=j+1}^{K-\tau+j}\sum_{s=1}^{k-1} \mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right)\right\|^2\right]. \quad (40)$$

Without loss of generality, we replace k with $a\tau + b + j$ where a is the communication round and b is the local update step. Note that these iterations are $\frac{K}{\tau} - 1$ full communication rounds. So, (40) can be re-written and bounded as follows.

$$\begin{aligned}
&\sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\sum_{s=1}^{a\tau+b+j-1} \mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right)\right\|^2\right] \\
&= \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\sum_{s=1}^{a\tau+j} \mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right)\right\|^2\right] \\
&\quad + \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\sum_{s=a\tau+j+1}^{a\tau+b+j-1} \mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right\|^2\right)\right] \\
&= \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\sum_{s=a\tau+j+1}^{a\tau+b+j-1} \mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right\|^2\right)\right] \quad (41) \\
&= \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\sum_{s=a\tau+j+1}^{a\tau+b+j-1} \mathbb{E}\left[\left\|\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right\|^2\right] \quad (42) \\
&\leq \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\sum_{s=a\tau+j+1}^{a\tau+b+j-1} m\sigma_j^2 \\
&= \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}(b-1)m\sigma_j^2 \\
&= \sum_{a=0}^{K/\tau-2}\frac{\tau(\tau-1)}{2}m\sigma_j^2 = (\frac{K}{\tau}-1)\frac{\tau(\tau-1)}{2}m\sigma_j^2, \quad (43)
\end{aligned}$$

where (41) holds because $\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)}$ becomes 0 when $s \leq a\tau + j$; (42) holds based on Lemma 2.

Finally, the last $\tau - j$ iterations are bounded as follows.

$$\begin{aligned}
&\sum_{k=K-\tau+j+1}^{K} \mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right)\right\|^2\right] \\
&= \sum_{k=K-\tau+j+1}^{K}\sum_{s=1}^{k-1} \mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right)\right\|^2\right] \quad (44) \\
& \quad (45)
\end{aligned}$$

where (44) holds because $\mathbf{g}_{(j,s)} - \nabla_j F(\mathbf{x}_s)$ has 0 mean and independent across s. Replacing k with $a\tau + b + j$, we have

$$\begin{aligned}
&\sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\sum_{s=1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right)\right\|^2\right]\right) \\
&= \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\sum_{s=1}^{a\tau+j}\mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right)\right\|^2\right]\right) \\
&\quad + \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right)\right\|^2\right]\right) \\
&= \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right)\right\|^2\right]\right) \qquad (46) \\
&= \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\mathbf{g}_{(j,s)} - \mathbf{f}_{(j,s)}\right\|^2\right]\right) \qquad (47) \\
&\leq \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\sigma_j^2\right) = \sum_{b=1}^{\tau-j}(b-1)m\sigma_j^2 = \frac{(\tau-j)(\tau-j-1)}{2}m\sigma_j^2, \qquad (48)
\end{aligned}$$

where (46) holds because $\mathbf{J}_j - \prod_{l=s}^{K-\tau+b+j-1}\mathbf{P}_{(j,l)}$ becomes 0 when $s \leq K - \tau + j$; (47) holds based on Lemma 2.

Based on (39), (43), and (48), T_1 is bounded as follows.

$$\begin{aligned}
T_1 &\leq \frac{j(j-1)}{2}m\sigma_j^2 + (\frac{K}{\tau}-1)\frac{\tau(\tau-1)}{2}m\sigma_j^2 + \frac{(\tau-j)(\tau-j-1)}{2}m\sigma_j^2 \\
&= \frac{j(j-1)+(\tau-j)(\tau-j-1)}{2}m\sigma_j^2 + (\frac{K}{\tau}-1)\frac{\tau(\tau-1)}{2}m\sigma_j^2 \\
&\leq \frac{\tau(\tau-1)}{2}m\sigma_j^2 + (\frac{K}{\tau}-1)\frac{\tau(\tau-1)}{2}m\sigma_j^2 \qquad (49) \\
&= mK\frac{(\tau-1)}{2}\sigma_j^2, \qquad (50)
\end{aligned}$$

where (49) holds because $0 < j \leq \tau$. Here, we finish bounding T_1.

Bounding T_2 – Likely to T_1, we partition T_2 to three subsets and bound them separately. We begin with the first j iterations.

$$\begin{aligned}
&\sum_{k=1}^{j}\mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right] \\
&\leq \sum_{k=1}^{j}(k-1)\sum_{s=1}^{k-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right] \qquad (51) \\
&= \sum_{k=1}^{j}(k-1)\sum_{s=1}^{k-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,s)}\right\|^2\right] \qquad (52) \\
&\leq \frac{j(j-1)}{2}\sum_{k=1}^{j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right], \qquad (53)
\end{aligned}$$

where (51) holds based on the convexity of ℓ_2 norm and Jensen's inequality; (52) holds based on Lemma 2.

Then, the next $K-\tau$ iterations of T_2 are bounded as follows.

$$
\begin{aligned}
&\sum_{k=j+1}^{K-\tau} \mathbb{E}\left[\left\|\sum_{s=1}^{k-1}(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)})\mathbf{f}_{(j,s)}\right\|^2\right] \\
&= \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\mathbb{E}\left[\left\|\sum_{s=1}^{a\tau+b+j-1}(\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)})\mathbf{f}_{(j,s)}\right\|^2\right] \\
&= \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\mathbb{E}\left[\left\|\sum_{s=a\tau+j+1}^{a\tau+b+j-1}(\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)})\mathbf{f}_{(j,s)}\right\|^2\right] &(54)\\
&= \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\left((b-1)\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|(\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)})\mathbf{f}_{(j,s)}\right\|^2\right]\right) &(55)\\
&\leq \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\left((b-1)\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,s)}\right\|^2\left\|(\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)})\right\|_{op}^2\right]\right) &(56)\\
&\leq \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\left((b-1)\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,s)}\right\|^2\right]\right) &(57)\\
&\leq \frac{\tau(\tau-1)}{2}\sum_{a=0}^{K/\tau-2}\left(\sum_{s=a\tau+j+1}^{a\tau+\tau+j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,s)}\right\|^2\right]\right) \\
&\leq \frac{\tau(\tau-1)}{2}\sum_{k=j+1}^{K-\tau+j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right], &(58)
\end{aligned}
$$

where (54) holds because $\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)}$ becomes 0 when $s \leq a\tau + j$. (55) holds based on the convexity of ℓ_2 norm and Jensen's inequality. (56) holds based on Lemma 1. (57) holds based on Lemma 2.

Finally, the last $\tau - j$ iterations of T_2 are bounded as follows.

$$\begin{aligned}
&\sum_{k=K-\tau+j+1}^{K} \mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right] \\
&= \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\mathbb{E}\left[\left\|\sum_{s=1}^{a\tau+b+j-1}\left(\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right]\right) \\
&= \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\mathbb{E}\left[\left\|\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\left(\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right]\right) \qquad (59) \\
&= \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left((b-1)\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right]\right) \qquad (60) \\
&\leq \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left((b-1)\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,s)}\right\|^2\right]\right) \qquad (61) \\
&\leq \frac{(\tau-j)(\tau-j-1)}{2}\sum_{a=K/\tau-1}^{K/\tau-1}\left(\sum_{s=a\tau+j+1}^{a\tau+\tau-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,s)}\right\|^2\right]\right) \\
&= \frac{(\tau-j)(\tau-j-1)}{2}\sum_{k=K-\tau+j+1}^{K-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right], \qquad (62)
\end{aligned}$$

where (59) holds because $\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}$ becomes 0 when $s \leq a\tau + j$; (60) holds based on the convexity of ℓ_2 norm and Jensen's inequality; (61) holds based on Lemma 2.

Based on (53), (58), and (62), T_2 is bounded as follows.

$$\begin{aligned}
T_2 &\leq \frac{j(j-1)}{2}\sum_{k=1}^{j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right] \\
&\quad + \frac{\tau(\tau-1)}{2}\sum_{k=j+1}^{K-\tau}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right] \\
&\quad + \frac{(\tau-j)(\tau-j-1)}{2}\sum_{k=K-\tau+j+1}^{K-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right] \\
&\leq \frac{\tau(\tau-1)}{2}\left(\sum_{k=1}^{K}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right]\right) \qquad (63)
\end{aligned}$$

Here, we finish bounding T_2.

Final result – By plugging in (50) and (63) into (35), we have

$$\begin{aligned}
&\frac{1}{mK}\sum_{k=1}^{K}\sum_{i=1}^{m}\sum_{j=1}^{\tau}L_j^2\,\mathbb{E}[\|\mathbf{u}_{(j,k)} - \mathbf{x}_{(j,k)}^i\|^2] \\
&\leq \frac{2\eta^2}{mK}\sum_{j=1}^{\tau}L_j^2\left(mK\frac{(\tau-1)}{2}\sigma_j^2 + \frac{\tau(\tau-1)}{2}\left(\sum_{k=1}^{K}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right]\right)\right) \\
&= \sum_{j=1}^{\tau}L_j^2\left(\eta^2(\tau-1)\sigma_j^2 + \frac{\eta^2\tau(\tau-1)}{mK}\left(\sum_{k=1}^{K}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right]\right)\right) \\
&= \sum_{j=1}^{\tau}L_j^2\left(\eta^2(\tau-1)\sigma_j^2 + \frac{\eta^2\tau(\tau-1)}{mK}\left(\sum_{i=1}^{m}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla_j F(\mathbf{x}_k^i)\right\|^2\right]\right)\right) \qquad (64)
\end{aligned}$$

Here, we complete the proof. □

Proof of Other Lemmas

Lemma 1. *Consider a real matrix $\mathbf{A} \in \mathbb{R}^{md_j \times md_j}$ and a real vector $\mathbf{b} \in \mathbb{R}^{md_j}$. If A is symmetric and $\mathbf{b} \neq \mathbf{0}_{md_j}$, we have*

$$\|\mathbf{A}\mathbf{b}\| \leq \|\mathbf{A}\|_{op}\|\mathbf{b}\| \tag{65}$$

Proof.

$$\begin{aligned}\|\mathbf{A}\mathbf{b}\|^2 &= \frac{\|\mathbf{A}\mathbf{b}\|^2}{\|\mathbf{b}\|^2}\|\mathbf{b}\|^2 \\ &\leq \|\mathbf{A}\|_{op}^2\|\mathbf{b}\|^2 \end{aligned}\tag{66}$$

where (66) holds based on the definition of operator norm. □

Lemma 2. *Given an identity matrix $\mathbf{I}_j \in \mathbb{R}^{md_j \times md_j}$ and a full-averaging matrix $\mathbf{J}_j \in \mathbb{R}^{md_j \times md_j}$,*

$$\|(\mathbf{I}_j - \mathbf{J}_j)\,\mathbf{x}\|^2 \leq \|\mathbf{x}\|^2 . \tag{67}$$

Proof. Since $\mathbf{J}_j$ is a real symmetric matrix, it can be decomposed into $\mathbf{J}_j = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$, where $\mathbf{Q}$ is a orthogonal eigenvector matrix and $\mathbf{\Lambda}$ is a diagonal eigenvalue matrix. By the definition, $\mathbf{J}_j$ the sum of every column is 1, and thus its eigenvalue is either 1 or 0. Because $\mathbf{J}_j$ has only two different columns, $\mathbf{\Lambda} = \text{diag}\{1, 1, 0, \cdots, 0\}$. By the definition of the identity matrix $\mathbf{I}_j$, it can be decomposed into $\mathbf{Q}\mathbf{\Lambda_i}\mathbf{Q}^\top$, where $\mathbf{\Lambda}_i = \text{diag}\{1, 1, \cdots, 1\}$. Then, we have

$$\mathbf{I}_j - \mathbf{J}_j = \mathbf{Q}(\mathbf{\Lambda_i} - \mathbf{\Lambda})\mathbf{Q}^\top. \tag{68}$$

Thus, the eigenvalue matrix of $\mathbf{I}_j - \mathbf{J}_j$ is $\mathbf{\Lambda_i} - \mathbf{\Lambda} = \text{diag}\{0, 0, 1, \cdots, 1\}$. By the definition of operator norm,

$$\|(\mathbf{I}_j - \mathbf{J}_j)\|_{op} = \sqrt{\lambda_{max}\left((\mathbf{I}_j - \mathbf{J}_j)^\top(\mathbf{I}_j - \mathbf{J}_j)\right)} = \sqrt{\lambda_{max}\left(\mathbf{I}_j - \mathbf{J}_j\right)} = 1 \tag{69}$$

Finally, based on Lemma 1, it follows

$$\|(\mathbf{I}_j - \mathbf{J}_j)\,\mathbf{x}\|^2 \leq \|\mathbf{I}_j - \mathbf{J}_j\|_{op}^2\,\|\mathbf{x}\|^2 = \|\mathbf{x}\|^2 \tag{70}$$

□

Convergence Analysis for Non-IID Data

We provide the proofs of Theorems and Lemmas under non-IID data settings.

Preliminaries

In addition to the conventional assumptions including the smoothness of each local objective function, the unbiased local stochastic gradients, and the bounded local variance, we highlight the following assumption on the bounded dissimilarity of the gradients across clients for non-IID analysis.

Assumption 4. *(Bounded Dissimilarity). There exist constants $\beta_j^2 \geq 1$ and $\kappa_j^2 \geq 0$ such that $\frac{1}{m}\sum_{i=1}^m \|\nabla_j F_i(\mathbf{x})\|^2 \leq \beta_j^2\|\frac{1}{m}\sum_{i=1}^m \nabla_j F_i(\mathbf{x})\|^2 + \kappa_j^2$. If the data is IID, $\beta_j^2 = 1$ and $\kappa_j^2 = 0$.*

Proof of Theorem 2

Theorem 2. *Suppose all m local models are initialized to the same point $\mathbf{u}_1$. Under Assumption $1 \sim 4$, if Algorithm 1 runs for K iterations and the learning rate satisfies $\eta \leq \frac{1}{L_{\max}}\min\left\{1, \frac{1}{\sqrt{2\tau(\tau-1)(2\beta^2+1)}}\right\}$, the average-squared gradient norm of $\mathbf{u}_k$ is bounded as follows*

$$\begin{aligned}\mathbb{E}\left[\frac{1}{K}\sum_{i=1}^K \|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{4}{\eta K}\left(\mathbb{E}\left[F(\mathbf{u}_1) - F(\mathbf{u}_{inf})\right]\right) + \frac{2\eta}{m}\sum_{j=1}^{\tau} L_j\sigma_j^2 \\ + 3\eta^2(\tau-1)\sum_{j=1}^{\tau} L_j^2\sigma_j^2 + 6\eta^2\tau(\tau-1)\sum_{j=1}^{\tau} L_j^2\kappa_j^2.\end{aligned}\tag{71}$$

Proof. Based on Lemma **??** and **??**, we have

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{2}{\eta K}\left(\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{inf})\right]\right)+\frac{\eta}{m}\sum_{j=1}^{\tau}L_j\sigma_j^2 + \sum_{j=1}^{\tau}L_j^2\left(\frac{\eta^2(\tau-1)\sigma_j^2}{1-A_j}+\frac{A_j\beta_j^2}{KL_j^2(1-A_j)}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla_j F(\mathbf{u}_k)\|^2\right]+\frac{A_j\kappa_j^2}{L_j^2(1-A_j)}\right),$$

where $A_j = 2\eta^2\tau(\tau-1)L_j^2$. After re-writing the left-hand side and a minor rearrangement, we have

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{\tau}\mathbb{E}\left[\|\nabla_j F(\mathbf{u}_k)\|^2\right] \leq \frac{2}{\eta K}\left(\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{inf})\right]\right)+\frac{\eta}{m}\sum_{j=1}^{\tau}L_j\sigma_j^2 + \frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{\tau}\frac{A_j\beta_j^2}{1-A_j}\mathbb{E}\left[\|\nabla_j F(\mathbf{u}_k)\|^2\right] + \sum_{j=1}^{\tau}L_j^2\left(\frac{\eta^2(\tau-1)\sigma_j^2}{1-A_j}+\frac{A_j\kappa_j^2}{L_j^2(1-A_j)}\right).$$

By moving the third term on the right-hand side to the left-hand side, we have

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{\tau}\left(1-\frac{A_j\beta_j^2}{1-A_j}\right)\mathbb{E}\left[\|\nabla_j F(\mathbf{u}_k)\|^2\right] \leq \frac{2}{\eta K}\left(\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{inf})\right]\right)+\frac{\eta}{m}\sum_{j=1}^{\tau}L_j\sigma_j^2 + \sum_{j=1}^{\tau}L_j^2\left(\frac{\eta^2(\tau-1)\sigma_j^2}{1-A_j}+\frac{A_j\kappa_j^2}{L_j^2(1-A_j)}\right). \tag{72}$$

If $A_j \leq \frac{1}{2\beta_j^2+1}$, then $\frac{A_j\beta_j^2}{1-A_j} \leq \frac{1}{2}$. Therefore, (72) can be simplified as follows.

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{\tau}\mathbb{E}\left[\|\nabla_j F(\mathbf{u}_k)\|^2\right] \leq \frac{4}{\eta K}\left(\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{inf})\right]\right)+\frac{2\eta}{m}\sum_{j=1}^{\tau}L_j\sigma_j^2 + 2\sum_{j=1}^{\tau}L_j^2\left(\frac{\eta^2(\tau-1)\sigma_j^2}{1-A_j}\right)+2\sum_{j=1}^{\tau}\frac{A_j\kappa_j^2}{1-A_j}.$$

The learning rate condition $A_j \leq \frac{1}{2\beta_j^2+1}$ also ensures that $\frac{1}{1-A_j} \leq 1+\frac{1}{2\beta_j^2}$. Based on Assumption 4, $\frac{1}{2\beta_j^2} \leq \frac{2}{3}$, and thus $\frac{1}{1-A_j} \leq \frac{2}{3}$. Therefore, we have

$$\begin{aligned}\frac{1}{K}\sum_{k=1}^{K}\sum_{j=1}^{\tau}\mathbb{E}\left[\|\nabla_j F(\mathbf{u}_k)\|^2\right] &\leq \frac{4}{\eta K}\left(\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{inf})\right]\right)+\frac{2\eta}{m}\sum_{j=1}^{\tau}L_j\sigma_j^2 \\ &\quad + 3\sum_{j=1}^{\tau}L_j^2\eta^2(\tau-1)\sigma_j^2+6\sum_{j=1}^{\tau}\eta^2\tau(\tau-1)L_j^2\kappa_j^2 \\ &= \frac{4}{\eta K}\left(\mathbb{E}\left[F(\mathbf{u}_1)-F(\mathbf{u}_{inf})\right]\right)+\frac{2\eta}{m}\sum_{j=1}^{\tau}L_j\sigma_j^2 \\ &\quad + 3\eta^2(\tau-1)\sum_{j=1}^{\tau}L_j^2\sigma_j^2+6\eta^2\tau(\tau-1)\sum_{j=1}^{\tau}L_j^2\kappa_j^2.\end{aligned}$$

We complete the proof. □

Learning rate constraints – We have two learning rate constraints as follows.

$$\eta \leq \frac{1}{L_{\max}} \qquad \text{Lemma ??}$$
$$2\eta^2\tau(\tau-1)L_j^2 \leq \frac{1}{2\beta_j^2+1} \qquad \text{Theorem 2}$$

By merging the two constraints, we can have a single learning rate constraint as follows.

$$\eta \leq \frac{1}{L_{\max}} \min\left\{1, \frac{1}{\sqrt{2\tau(\tau-1)(2\beta^2+1)}}\right\} \tag{73}$$

Proof of Lemma 3

Lemma 3. *(model discrepancy) Under Assumption* $1 \sim 4$*, local SGD with the partial model averaging ensures*

$$\begin{aligned}&\frac{1}{mK}\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^i\right\|^2\right]\\ &\quad\leq \frac{\eta^2(\tau-1)\sigma_j^2}{1-A_j}+\frac{A_j\beta_j^2}{KL_j^2(1-A_j)}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla_j F(\mathbf{u}_k)\right\|^2\right]+\frac{A_j\kappa_j^2}{L_j^2(1-A_j)}\end{aligned} \tag{74}$$

where $A_j = 2\eta^2\tau(\tau-1)L_j^2$.

Proof. We begin with re-writing the weighted average of the squared distance using the vectorized form of the local models as follows.

$$\frac{1}{m}\sum_{i=1}^{m}\left\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^i\right\|^2 = \left\|\mathbf{J}_j\mathbf{x}_{(j,k)}-\mathbf{x}_{(j,k)}\right\|^2 = \left\|(\mathbf{J}_j-\mathbf{I}_j)\mathbf{x}_{(j,k)}\right\|^2 \tag{75}$$

Then, according to the parameter update rule, we have

$$\begin{aligned}(\mathbf{J}_j-\mathbf{I}_j)\mathbf{x}_{(j,k)} &= (\mathbf{J}_j-\mathbf{I}_j)\mathbf{P}_{(j,k-1)}(\mathbf{x}_{(j,k-1)}-\eta\mathbf{g}_{(j,k-1)}) &(76)\\ &= (\mathbf{J}_j-\mathbf{I}_j)\mathbf{P}_{(j,k-1)}\mathbf{x}_{(j,k-1)}-(\mathbf{J}_j-\mathbf{P}_{(j,k-1)})\eta\mathbf{g}_{(j,k-1)}, &(77)\end{aligned}$$

where the second equality holds because $\mathbf{J}_j\mathbf{P}_j = \mathbf{J}_j$ and $\mathbf{I}_j\mathbf{P}_j = \mathbf{P}_j$.

Then, expanding the expression of $\mathbf{x}_{(j,k-1)}$, we have

$$\begin{aligned}(\mathbf{J}_j-\mathbf{I}_j)\mathbf{x}_{(j,k)} &= (\mathbf{J}_j-\mathbf{I}_j)\mathbf{P}_{(j,k-1)}(\mathbf{P}_{(j,k-2)}(\mathbf{x}_{(j,k-2)}-\eta\mathbf{g}_{(j,k-2)}))-(\mathbf{J}_j-\mathbf{P}_{(j,k-1)})\eta\mathbf{g}_{(j,k-1)}\\ &= (\mathbf{J}_j-\mathbf{I}_j)\mathbf{P}_{(j,k-1)}\mathbf{P}_{(j,k-2)}\mathbf{x}_{(j,k-2)}-(\mathbf{J}_j-\mathbf{P}_{(j,k-1)}\mathbf{P}_{(j,k-2)})\eta\mathbf{g}_{(j,k-2)}-(\mathbf{J}_j-\mathbf{P}_{(j,k-1)})\eta\mathbf{g}_{(j,k-1)}.\end{aligned}$$

Repeating the same procedure for $\mathbf{x}_{(j,k-2)}, \mathbf{x}_{(j,k-3)}, \cdots, \mathbf{x}_{(j,2)}$, we have

$$\begin{aligned}(\mathbf{J}_j-\mathbf{I}_j)\hat{\mathbf{x}}_{(j,k)} &= (\mathbf{J}_j-\mathbf{I}_j)\prod_{s=1}^{k-1}\mathbf{P}_{(j,s)}\mathbf{x}_{(j,1)}-\eta\sum_{s=1}^{k-1}(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)})\mathbf{g}_{(j,s)}\\ &= -\eta\sum_{s=1}^{k-1}(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)})\mathbf{g}_{(j,s)}, &(78)\end{aligned}$$

where (78) holds because $\mathbf{x}_{(j,1)}^i$ is the same across all the workers and thus $(\mathbf{J}_j-\mathbf{I}_j)\mathbf{x}_{(j,1)} = 0$.

Based on (78), we have

$$
\begin{aligned}
&\frac{1}{mK}\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^{i}\right\|^{2}\right]\\
&\quad=\frac{1}{K}\sum_{k=1}^{K}\left(\mathbb{E}\left[\left\|\left(\mathbf{J}_{j}-\mathbf{I}_{j}\right)\mathbf{x}_{(j,k)}\right\|^{2}\right]\right)\\
&\quad=\frac{1}{K}\sum_{k=1}^{K}\left(\eta^{2}\,\mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_{j}-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\mathbf{g}_{(j,s)}\right\|^{2}\right]\right)\\
&\quad=\frac{1}{K}\sum_{k=1}^{K}\left(\eta^{2}\,\mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_{j}-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)+\sum_{s=1}^{k-1}\left(\mathbf{J}_{j}-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^{2}\right]\right)\\
&\quad\leq\frac{2\eta^{2}}{K}\left(\underbrace{\sum_{k=1}^{K}\mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_{j}-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^{2}\right]}_{T_{3}}+\underbrace{\sum_{k=1}^{K}\mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_{j}-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^{2}\right]}_{T_{4}}\right) \qquad (79)
\end{aligned}
$$

where (79) holds based on the convexity of ℓ_2 norm and Jensen's inequality. Now, we focus on bounding T_3 and T_4, separately.

Bounding T_3 – We first partition K iterations into three subsets: the first j iterations, the next $K-\tau$ iterations, and the final $\tau-j$ iterations. We bound the first j iterations of T_3 as follows.

$$
\begin{aligned}
\sum_{k=1}^{j}\mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_{j}-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^{2}\right] &\leq\sum_{k=1}^{j}\sum_{s=1}^{k-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_{j}-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^{2}\right] &(80)\\
&\leq\sum_{k=1}^{j}\sum_{s=1}^{k-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_{j}-\mathbf{I}_{j}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^{2}\right] &(81)\\
&\leq\sum_{k=1}^{j}\sum_{s=1}^{k-1}\mathbb{E}\left[\left\|\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right\|^{2}\right] &(82)\\
&=\sum_{k=1}^{j}\sum_{s=1}^{k-1}\frac{1}{m}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\mathbf{g}_{(j,s)}^{i}-\nabla_{j}F_{i}(\mathbf{x}_{s}^{i})\right\|^{2}\right]\\
&\leq\sum_{k=1}^{j}\sum_{s=1}^{k-1}\sigma_{j}^{2}\\
&=\frac{j(j-1)}{2}\sigma_{j}^{2}, &(83)
\end{aligned}
$$

where (81) holds because $\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}$ is $\mathbf{I}_j$ when $k<j$; (82) holds based on Lemma 2.

Then, the next $K-\tau$ iterations of T_3 can be bounded as follows.

$$
\begin{aligned}
&\sum_{k=j+1}^{K-\tau+j}\mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_{j}-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^{2}\right]\\
&\qquad=\sum_{k=j+1}^{K-\tau+j}\sum_{s=1}^{k-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_{j}-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^{2}\right] &(84)
\end{aligned}
$$

Replacing k with $a\tau+b+j$, (84) can be re-written and bounded as follows.

$$
\begin{aligned}
&\sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\sum_{s=1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^2\right]\\
&\quad=\sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\sum_{s=1}^{a\tau+j}\mathbb{E}\left[\left\|\left(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^2\right]\\
&\qquad+\sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^2\right]\\
&\quad=\sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^2\right] &(85)\\
&\quad\le\sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right\|^2\right] &(86)\\
&\quad=\sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\frac{1}{m}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\mathbf{g}_{(j,s)}^i-\nabla_j F_i(\mathbf{x}_s^i)\right\|^2\right]\\
&\quad\le\sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\sigma_j^2 &(87)\\
&\quad=\sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}(b-1)\sigma_j^2\\
&\quad=\sum_{a=0}^{K/\tau-2}\frac{\tau(\tau-1)}{2}\sigma_j^2\\
&\quad=\left(\frac{K}{\tau}-1\right)\frac{\tau(\tau-1)}{2}\sigma_j^2, &(88)
\end{aligned}
$$

where (41) holds because $\mathbf{J}_j-\prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)}$ becomes 0 when $s\le a\tau+j$; (86) holds based on Lemma 2. (87) holds based on Assumption 6.

Finally, the last $\tau-j$ iterations are bounded as follows.

$$
\begin{aligned}
&\sum_{k=K-\tau+j+1}^{K}\mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^2\right]\\
&\quad=\sum_{k=K-\tau+j+1}^{K}\sum_{s=1}^{k-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^2\right] &(89)
\end{aligned}
$$

where (89) holds because $\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}$ has a mean of 0 and independent across s.

By replacing k with $a\tau+b+j$ in (89), we have

$$\begin{aligned}
&\sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\sum_{s=1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^2\right]\right)\\
&=\sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\sum_{s=1}^{a\tau+j}\mathbb{E}\left[\left\|\left(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^2\right]\right)\\
&\quad+\sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^2\right]\right)\\
&=\sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\left(\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right)\right\|^2\right]\right) &(90)\\
&=\sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\mathbf{g}_{(j,s)}-\mathbf{f}_{(j,s)}\right\|^2\right]\right) &(91)\\
&=\sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\frac{1}{m}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\left(\mathbf{g}_{(j,s)}^i-\nabla_j F_i(\mathbf{x}_{(j,s)}^i)\right)\right\|^2\right]\right)\\
&\le\sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\sigma_j^2 &(92)\\
&=\sum_{b=1}^{\tau-j}(b-1)\sigma_j^2=\frac{(\tau-j)(\tau-j-1)}{2}\sigma_j^2, &(93)
\end{aligned}$$

where (90) holds because $\mathbf{J}_j-\prod_{l=s}^{K-\tau+b+j-1}\mathbf{P}_{(j,l)}$ becomes 0 when $s\le K-\tau+j$; (91) holds based on Lemma 2; (92) holds based on Assumption 6.

Summing up (83), (88), and (93), we have

$$\begin{aligned}
T_3&\le\frac{j(j-1)}{2}\sigma_j^2+(\frac{K}{\tau}-1)\frac{\tau(\tau-1)}{2}\sigma_j^2+\frac{(\tau-j)(\tau-j-1)}{2}\sigma_j^2\\
&=\frac{j(j-1)+(\tau-j)(\tau-j-1)}{2}\sigma_j^2+(\frac{K}{\tau}-1)\frac{\tau(\tau-1)}{2}\sigma_j^2\\
&\le\frac{\tau(\tau-1)}{2}m\sigma_j^2+(\frac{K}{\tau}-1)\frac{\tau(\tau-1)}{2}\sigma_j^2 &(94)\\
&=K\frac{(\tau-1)}{2}\sigma_j^2, &(95)
\end{aligned}$$

where (94) holds because $0<j\le\tau$. Here, we finish bounding T_3.

Bounding T_4 – Likely to T_3, we partition T_4 to three subsets and bound them separately. The T_4 at the first j iterations can be bounded as follows.

$$\begin{aligned}
\sum_{k=1}^{j}\mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right]&\le\sum_{k=1}^{j}(k-1)\sum_{s=1}^{k-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j-\prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right]\\
&=\sum_{k=1}^{j}(k-1)\sum_{s=1}^{k-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j-\mathbf{I}_j\right)\mathbf{f}_{(j,s)}\right\|^2\right] &(96)\\
&\le\sum_{k=1}^{j}(k-1)\sum_{s=1}^{k-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,s)}\right\|^2\right] &(97)\\
&=\frac{j(j-1)}{2}\sum_{k=1}^{j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right], &(98)
\end{aligned}$$

where (96) holds because $\prod_{l=1}^{k-1} \mathbf{P}_{(j,l)}$ is $\mathbf{I}_j$ when $k < j$; (97) holds based on Lemma 2.

Then, the next $K - \tau$ iterations of T_2 are bounded as follows.

$$\sum_{k=j+1}^{K-\tau} \mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right] = \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\mathbb{E}\left[\left\|\sum_{s=1}^{a\tau+b+j-1}\left(\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right]$$

$$= \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\mathbb{E}\left[\left\|\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\left(\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right] \tag{99}$$

$$= \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\left((b-1)\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right]\right) \tag{100}$$

$$= \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\left((b-1)\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j - \mathbf{I}_j\right)\mathbf{f}_{(j,s)}\right\|^2\right]\right)$$

$$\leq \sum_{a=0}^{K/\tau-2}\sum_{b=1}^{\tau}\left((b-1)\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,s)}\right\|^2\right]\right) \tag{101}$$

$$\leq \frac{\tau(\tau-1)}{2}\sum_{a=0}^{K/\tau-2}\left(\sum_{s=a\tau+j+1}^{a\tau+\tau+j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,s)}\right\|^2\right]\right)$$

$$\leq \frac{\tau(\tau-1)}{2}\sum_{k=j+1}^{K-\tau+j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right], \tag{102}$$

where (99) holds because $\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)}$ becomes 0 when $s \leq a\tau + j$; (100) holds based on the convexity of ℓ_2 norm and Jensen's inequality; (101) holds based on Lemma 2.

Finally, the last $\tau - j$ iterations of T_4 are bounded as follows.

$$\sum_{k=K-\tau+j+1}^{K} \mathbb{E}\left[\left\|\sum_{s=1}^{k-1}\left(\mathbf{J}_j - \prod_{l=s}^{k-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right] = \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\mathbb{E}\left[\left\|\sum_{s=1}^{a\tau+b+j-1}\left(\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right]\right)$$

$$= \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left(\mathbb{E}\left[\left\|\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\left(\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right]\right) \tag{103}$$

$$= \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left((b-1)\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}\mathbf{P}_{(j,l)}\right)\mathbf{f}_{(j,s)}\right\|^2\right]\right) \tag{104}$$

$$= \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left((b-1)\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\left(\mathbf{J}_j - \mathbf{I}_j\right)\mathbf{f}_{(j,s)}\right\|^2\right]\right)$$

$$\leq \sum_{a=K/\tau-1}^{K/\tau-1}\sum_{b=1}^{\tau-j}\left((b-1)\sum_{s=a\tau+j+1}^{a\tau+b+j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,s)}\right\|^2\right]\right) \tag{105}$$

$$\leq \frac{(\tau-j)(\tau-j-1)}{2}\sum_{a=K/\tau-1}^{K/\tau-1}\left(\sum_{s=a\tau+j+1}^{a\tau+\tau-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,s)}\right\|^2\right]\right)$$

$$= \frac{(\tau-j)(\tau-j-1)}{2}\sum_{k=K-\tau+j+1}^{K-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right], \tag{106}$$

where (103) holds because $\mathbf{J}_j - \prod_{l=s}^{a\tau+b+j-1}$ becomes 0 when $s \leq a\tau + j$; (104) holds based on the convexity of ℓ_2 norm and Jensen's inequality; (105) holds based on Lemma 2.

Based on (98), (102), and (106), T_4 is bounded as follows.

$$T_4 \leq \frac{j(j-1)}{2}\sum_{k=1}^{j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right] + \frac{\tau(\tau-1)}{2}\sum_{k=j+1}^{K-\tau+j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right]$$
$$+ \frac{(\tau-j)(\tau-j-1)}{2}\sum_{k=K-\tau+j+1}^{K-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right]$$
$$\leq \frac{\tau(\tau-1)}{2}\left(\sum_{k=1}^{j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right] + \sum_{k=j+1}^{K-\tau+j-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right] + \sum_{k=K-\tau+j+1}^{K-1}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right]\right) \tag{107}$$
$$\leq \frac{\tau(\tau-1)}{2}\left(\sum_{k=1}^{K}\mathbb{E}\left[\left\|\mathbf{f}_{(j,k)}\right\|^2\right]\right)$$
$$= \frac{\tau(\tau-1)}{2m}\left(\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\nabla_j F_i(\mathbf{x}_k^i)\right\|^2\right]\right), \tag{108}$$

where (107) holds because $0 < j \leq \tau$. Here, we finish bounding T_4.

By plugging in (95) and (108) into (79), we have

$$\begin{aligned}
&\frac{1}{mK}\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^{i}\right\|^{2}\right] \\
&\quad\leq\frac{2\eta^{2}}{K}\left(K\frac{(\tau-1)}{2}\sigma_{j}^{2}+\frac{\tau(\tau-1)}{2m}\left(\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\nabla_{j}F_{i}(\mathbf{x}_{k}^{i})\right\|^{2}\right]\right)\right) \\
&\quad=\eta^{2}(\tau-1)\sigma_{j}^{2}+\frac{\eta^{2}\tau(\tau-1)}{mK}\left(\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\nabla_{j}F_{i}(\mathbf{x}_{k}^{i})\right\|^{2}\right]\right) \qquad (109)
\end{aligned}$$

The local gradient term on the right-hand side in (109) can be rewritten using the following inequality.

$$\begin{aligned}
\mathbb{E}\left[\left\|\nabla_{j}F_{i}(\mathbf{x}_{k}^{i})\right\|^{2}\right]&=\mathbb{E}\left[\left\|\nabla_{j}F_{i}(\mathbf{x}_{k}^{i})-\nabla_{j}F_{i}(\mathbf{u}_{k})+\nabla_{j}F_{i}(\mathbf{u}_{k})\right\|^{2}\right] \\
&\leq 2\,\mathbb{E}\left[\left\|\nabla_{j}F_{i}(\mathbf{x}_{k}^{i})-\nabla_{j}F_{i}(\mathbf{u}_{k})\right\|^{2}\right]+2\,\mathbb{E}\left[\left\|\nabla_{j}F_{i}(\mathbf{u}_{k})\right\|^{2}\right] \qquad (110) \\
&\leq 2L_{j}^{2}\,\mathbb{E}\left[\left\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^{i}\right\|^{2}\right]+2\,\mathbb{E}\left[\left\|\nabla_{j}F_{i}(\mathbf{u}_{k})\right\|^{2}\right], \qquad (111)
\end{aligned}$$

where (110) holds based on the convexity of ℓ_2 norm and Jensen's inequality.

Plugging in (111) into (109), we have

$$\begin{aligned}
&\frac{1}{mK}\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^{i}\right\|^{2}\right] \\
&\quad\leq\eta^{2}(\tau-1)\sigma_{j}^{2}+\frac{2\eta^{2}\tau(\tau-1)L_{j}^{2}}{mK}\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^{i}\right\|^{2}\right] \\
&\qquad+\frac{2\eta^{2}\tau(\tau-1)}{mK}\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\nabla_{j}F_{i}(\mathbf{u}_{k})\right\|^{2}\right] \qquad (112)
\end{aligned}$$

After a minor rearranging, we have

$$\begin{aligned}
&\frac{1}{mK}\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^{i}\right\|^{2}\right] \\
&\quad\leq\frac{\eta^{2}(\tau-1)\sigma_{j}^{2}}{1-2\eta^{2}\tau(\tau-1)L_{j}^{2}}+\frac{2\eta^{2}\tau(\tau-1)}{mK(1-2\eta^{2}\tau(\tau-1)L_{j}^{2})}\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\nabla_{j}F_{i}(\mathbf{u}_{k})\right\|^{2}\right] \qquad (113)
\end{aligned}$$

Let us define $A_j = 2\eta^2\tau(\tau-1)L_j^2$. Then (113) is simplified as follows.

$$\begin{aligned}
&\frac{1}{mK}\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^{i}\right\|^{2}\right] \\
&\quad\leq\frac{\eta^{2}(\tau-1)\sigma_{j}^{2}}{1-A_{j}}+\frac{A_{j}}{mKL_{j}^{2}(1-A_{j})}\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\nabla_{j}F_{i}(\mathbf{u}_{k})\right\|^{2}\right]
\end{aligned}$$

Based on Assumption 4, we have

$$\begin{aligned}
&\frac{1}{mK}\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\mathbf{u}_{(j,k)}-\mathbf{x}_{(j,k)}^{i}\right\|^{2}\right] \\
&\quad\leq\frac{\eta^{2}(\tau-1)\sigma_{j}^{2}}{1-A_{j}}+\frac{A_{j}\beta_{j}^{2}}{KL_{j}^{2}(1-A_{j})}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla_{j}F(\mathbf{u}_{k})\right\|^{2}\right]+\frac{A_{j}\kappa_{j}^{2}}{L_{j}^{2}(1-A_{j})}
\end{aligned}$$

Here, we complete the proof. □

Additional Experimental Results

In this section, we provide detailed experimental settings and additional experimental results that support our proposed algorithm.

Datasets and Models

CIFAR-10 and CIFAR-100 – CIFAR-10 and CIFAR-100 is benchmark image datasets for classification. Both datasets have 50K training samples and 10K validation samples. Each sample is a 32×32 RGB image. We use ResNet-20 and Wide-ResNet-28-10 for CIFAR-10 and CIFAR-100 classification, respectively. We apply weight decay using a parameter of 0.0001 for ResNet-20 and 0.0005 for Wide-ResNet-28-10.

SVHN – SVHN is an image dataset that consists of 73K training samples and 26K test samples. It also has additional 530K training samples. Each sample is a 32×32 RGB image. We use Wide-ResNet-16-8 for classification experiments. We apply weight decay using a parameter of 0.0001.

Fashion-MNIST – Fashion-MNIST is an image dataset that has 50K training samples and 10K test samples. Each sample is a 28×28 gray image. We use VGG-11 for classification experiments. We apply weight decay using a parameter of 0.0001.

IMDB review – IMDB consists of 50K movie reviews for natural language processing. For IMDB sentiment analysis experiments, we use a LSTM model that consists of one embedding layer followed by one bidirectional LSTM layer of size 256. We also applied dropout with a probability of 0.3 to both layers. The maximum number of words in the embedding layer is $10,000$ and the output dimension is 256. We do not apply weight decay for LSTM training.

Federated Extended MNIST – FEMNIST consists of $805,263$ pictures of hand-written digits and characters. The data is intrinsically heterogeneous such that $3,550$ writers provide different numbers of pictures. We use a CNN that consists of 2 convolution layers and 2 fully-connected layers. We provide the reference to the model architecture in the main paper. When training the model, we use a random 10% of the writer's samples only.

Experimental Results

Local Model Re-distribution When re-distributing the models to a new set of active workers, there are two available design options. First, the aggregated local models can be fully averaged and then distributed to other active workers. This option slightly sacrifices the statistical efficiency due to the full averaging while the local data privacy is better protected. Second, the aggregated local models can be re-distributed to other active workers without averaging. This option provides a good statistical efficiency while potentially having a privacy issue. In our experiments, we found that both options outperforms the periodic averaging. All the performance results reported in the main paper are obtained using the second option.

Here we compare the performance of these two design options in Table 4. We set τ to 11 and re-distribute the local models to new active workers after every 110 iterations so that the total communication cost is the same as the periodic averaging setting. First, interestingly, the second design option provides the better accuracy than the first design option in most of the settings. This result demonstrates that the full model averaging likely harms the statistical efficiency regardless of the frequency. Second, both design options consistently outperforms the periodic averaging. In this work, we simply choose a random subset of workers as the new active workers. Studying the impact of different device selection schemes on the convergence properties and the accuracy can be an interesting future work.

Model Partitioning When synchronizing a part of model in Algorithm 1, the model can be partitioned in many different ways. Table 5 and 6 show the CIFAR-10 classification performance comparison between *channel-partition* and *layer-partition* for IID and non-IID data settings, respectively. We do not see a large difference between the two different partitioning methods. This result demonstrates that, because every parameter is guaranteed to be averaged after every τ iterations, the order of synchronizations does not strongly affect the performance.

Learning Curves of IID Data Experiments We present the training loss and validation accuracy curves collected in our experiments.

CIFAR-10 – Figure 6 shows the learning curves of ResNet-20 training on CIFAR-10. The averaging interval τ is set to 2, 4, and 8 (**a**, **b**, **c**). The learning rate is decayed by a factor of 10 after 150 and 225 epochs. We clearly see that the partial averaging makes the training loss converge faster. In addition, the partial averaging achieves a higher validation accuracy than the periodic averaging after the same number of training epochs. The performance gap between the periodic averaging and the partial averaging becomes more significant as τ increases.

CIFAR-100 – Figure 7 shows the learning curves of WideResNet-28-10 training on CIFAR-100. The hyper-parameter settings are shown in Table 1. The averaging interval τ is set to 2, 4, and 8 (**a**, **b**, **c**). The learning rate is decayed by a factor of 10 after 120 and 185 epochs. Overall, the two different averaging methods show a significant performance difference. When the averaging interval is large (8), both averaging methods show a significant drop of accuracy but the partial averaging still outperforms the periodic averaging.

SVHN – Figure 8 shows the learning curves of WideResNet-16-8 training on SVHN. The hyper-parameter settings are shown in Table 1. The averaging interval τ is set to 4, 16, and 64 (**a**, **b**, **c**). The learning rate is decayed by a factor of 10 after 80 and 120

Table 4: CIFAR-10 (ResNet20) classification accuracy comparison between two design options: **(1)**: average the aggregated local models before re-distributing to new active workers, **(2)**: re-distribute the local models to new active workers without averaging.

dataset	batch size (LR)	workers	avg interval	active ratio	Dir(α)	design **(1)**	design **(2)**
CIFAR-10 (ResNet20)	32 (0.4)	128	11	100%	1	$90.74 \pm 0.1\%$	**91.54** $\pm 0.1\%$
					0.5	$90.53 \pm 0.1\%$	**91.43**$\pm 0.1\%$
					0.1	$90.39 \pm 0.2\%$	**91.08** $\pm 0.1\%$
				50%	1	**90.69**$\pm 0.1\%$	$90.64 \pm 0.2\%$
					0.5	$90.23 \pm 0.2\%$	**91.02**$\pm 0.3\%$
					0.1	$89.89 \pm 0.2\%$	**90.17**$\pm 0.2\%$
				25%	1	$89.64 \pm 0.3\%$	**91.00** $\pm 0.2\%$
					0.5	$89.39 \pm 0.3\%$	**90.16**$\pm 0.3\%$
	32 (0.2)				0.1	$88.32 \pm 0.2\%$	**88.95** $\pm 0.3\%$

Table 5: CIFAR-10 classification performance comparison between *channel-partition* and *layer-partition* (IID data). We do not see any meaningful difference between the two partitioning methods.

dataset	model	# of workers	epochs	avg interval	channel-partition	layer-partition
CIFAR-10	ResNet20	128	300	2	**91.89**$\pm 0.1\%$	$91.81 \pm 0.1\%$
				4	$90.56 \pm 0.2\%$	**90.58**$\pm 0.2\%$
				8	**87.13**$\pm 0.1\%$	$87.10 \pm 0.1\%$

epochs. We could use a relatively longer averaging interval than the other experiments without much losing the performance due to the large number of training samples. The partial averaging slightly outperforms the periodic averaging in all the settings.

Fashion-MNIST – Figure 9 shows the learning curves of VGG-11 training on Fashion-MNIST. The hyper-parameter settings are shown in Table 1. The averaging interval τ is set to 2, 4, and 8 (**a**, **b**, **c**). The learning rate is decayed by a factor of 10 after 50 and 75 epochs. The partial averaging consistently outperforms the periodic averaging for all the different averaging interval settings.

IMDB reviews – Figure 10 shows the learning curves of LSTM training on IMDB. The hyper-parameter settings are shown in Table 1. The averaging interval τ is set to 2, 4, and 8 (**a**, **b**, **c**). The learning rate is decayed by a factor of 10 after 60 and 80 epochs. Although the final accuracy is not significantly different between the two averaging methods, the partial averaging accuracy is still consistently higher than that of the periodic averaging.

Learning Curves of non-IID Data Experiments Here, we present the learning curves for non-IID data experiments. We summarize two key observations on the learning curves as follows. First, the partial averaging provides smoother training loss curves than the periodic full averaging as well as a faster convergence. Especially when the averaging interval is large ($\tau = 8$), the periodic averaging curves fluctuate significantly while the partial averaging curves are stable. Second, the validation curves show noticeable differences. The partial averaging shows a sharp increase of validation curves when the learning rate is decayed. It has been known that the high degree of noise in the early training can improve the generalization performance. This pattern of validation curves is well aligned with the presented final accuracy.

CIFAR-10 – Figure 11 shows the learning curves of ResNet20 training on CIFAR-10 under more realistic Federated Learning settings. **a, b, c**: Training loss curves with activation ratio of 25%, 50%, and 100%, respectively. **d, e, f**: Validation accuracy curves with activation ratio of 25%, 50%, and 100%, respectively. The three columns correspond to Dirichlet's concentration parameters of 0.1, 0.5, and 1.0, respectively. The learning rate is decayed by a factor of 10 after 5000 and 7500 iterations. We see that the partial averaging shows a faster convergence of training loss as well as a higher accuracy in all the experiment.

IMDB – Figure 12 shows the learning curves of LSTM training on IMDB under more realistic Federated Learning settings. **a, b, c**: Training loss curves with activation ratio of 25%, 50%, and 100%, respectively. **d, e, f**: Validation accuracy curves with activation ratio of 25%, 50%, and 100%, respectively. The two columns correspond to Dirichlet's concentration parameters of 0.5 and 1.0, respectively. The learning rate is decayed by a factor of 10 after 1500 and 1800 iterations. Likely to CIFAR-10, the partial averaging shows superior classification performance than the periodic averaging. The performance gap is even larger than that of the same IMDB sentiment analysis with IID settings.

FEMNIST – Figure 13 shows the learning curves of CNN training on FEMNIST. Because the data distribution is already heterogeneous across the workers, we adjust the ratio of device activation. **a, b, c** show the learning curves with 25%, 50%, and 100% activation ratios, respectively. The partial model averaging achieves the higher accuracy in all the settings. Especially, the training loss curves show a significant difference between the two model averaging methods.

Table 6: CIFAR-10 (ResNet20) classification accuracy comparison between two model partitioning options: **(1)**: *channel-partitioning*, **(2)**: *layer-partitioning*.

dataset	batch size (LR)	workers	avg interval	active ratio	Dir(α)	*layer-partitioning*	*channel-partitioning*
CIFAR-10 (ResNet20)	32 (0.4)	128	11	100%	1	91.21 $\pm$ 0.1%	**91.54** $\pm$0.1%
					0.5	**91.56**$\pm$0.2%	91.43 $\pm$ 0.1%
					0.1	**91.31**$\pm$0.1%	91.08 $\pm$ 0.1%
				50%	1	**90.66**$\pm$0.1%	90.61 $\pm$ 0.2%
					0.5	90.97 $\pm$ 0.2%	**91.02**$\pm$0.3%
					0.1	90.09 $\pm$ 0.3%	**90.64**$\pm$0.2%
				25%	1	90.48 $\pm$ 0.3%	**91.00**$\pm$0.2%
					0.5	89.39 $\pm$ 0.3%	**90.02**$\pm$0.2%
	32 (0.2)				0.1	88.32 $\pm$ 0.2%	**88.92**$\pm$0.1%

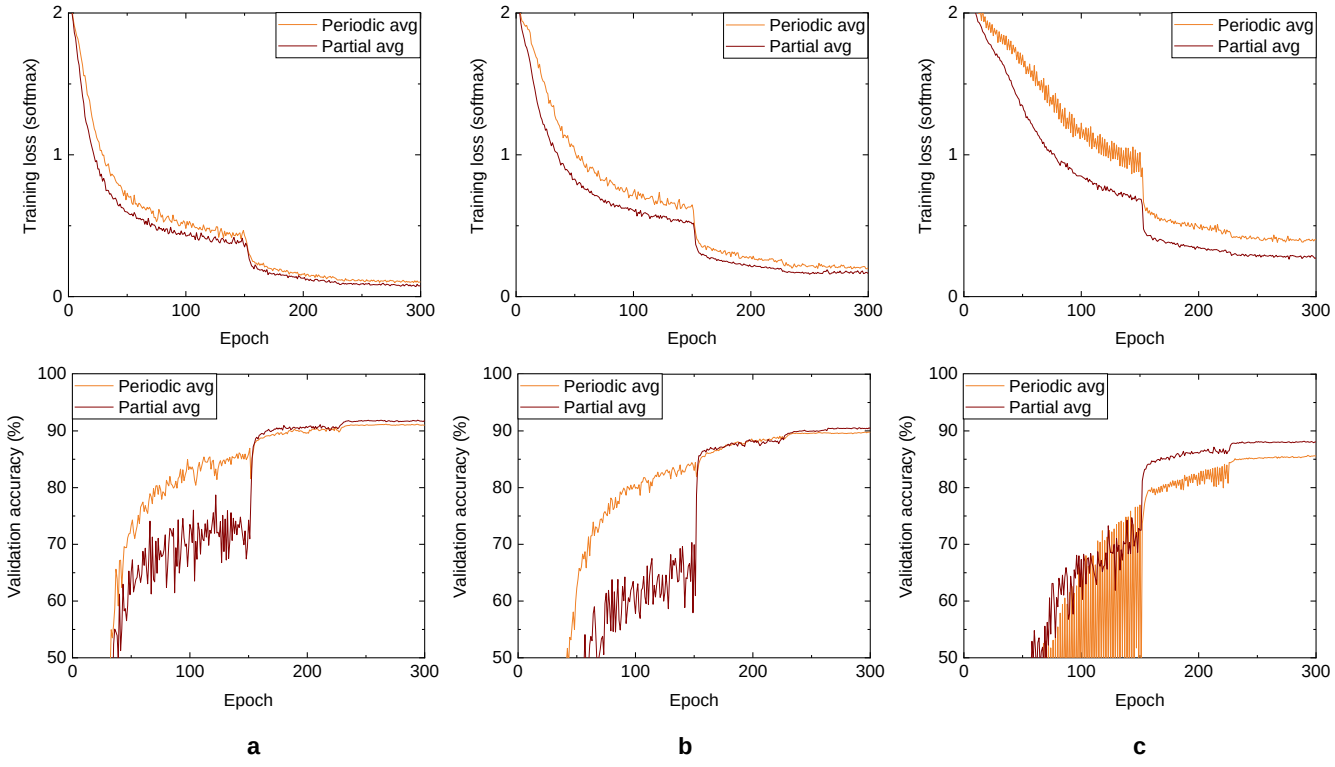


Figure 6: The learning curves of ResNet-20 (CIFAR-10) training. The number of workers is 128 and the hyper-parameters are shown in Table 1. The top charts are the training loss and the bottom charts are the validation accuracy. The averaging interval τ is set to 2, 4, and 8 (**a**, **b**, and **c**).

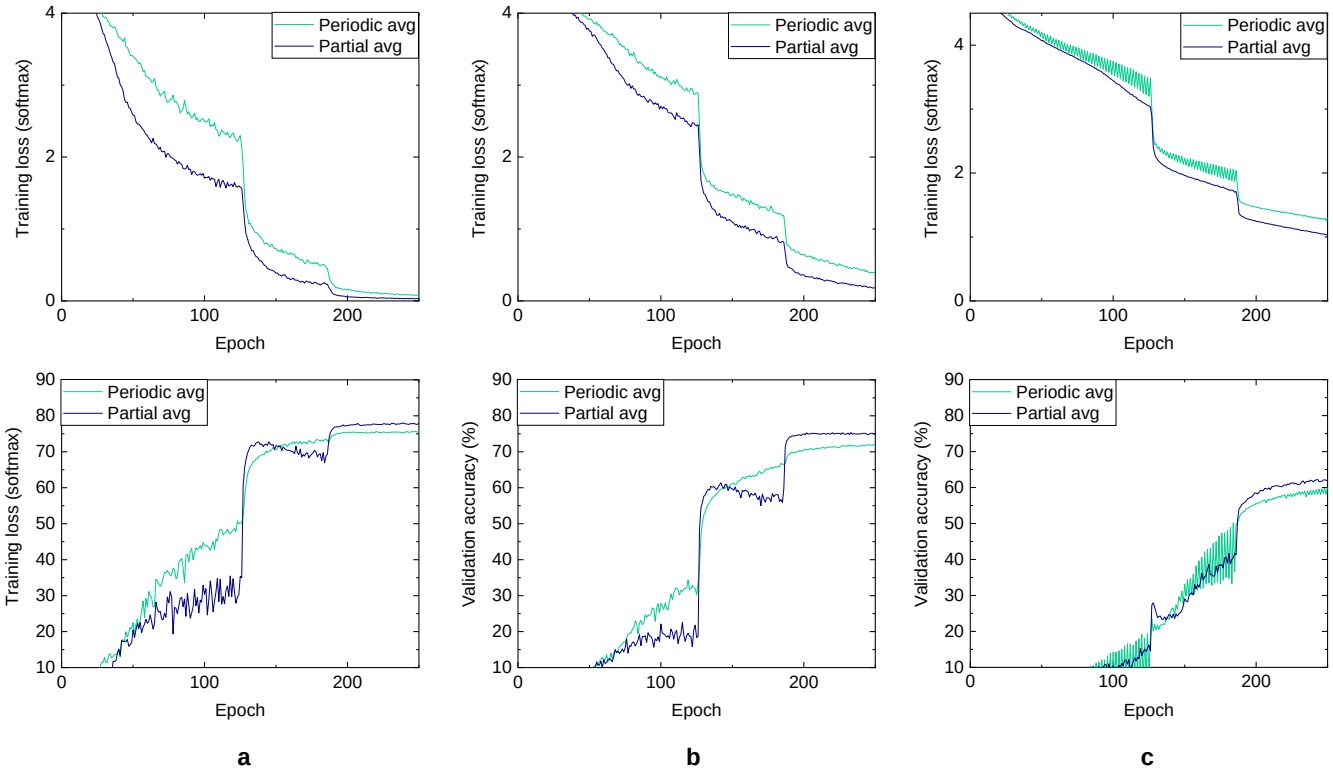


Figure 7: The learning curves of WideResNet-28-10 (CIFAR-100) training. The number of workers is 128 and the hyper-parameters are shown in Table 1. The top charts are the training loss and the bottom charts are the validation accuracy. The averaging interval τ is set to 2, 4, and 8 (**a**, **b**, and **c**).

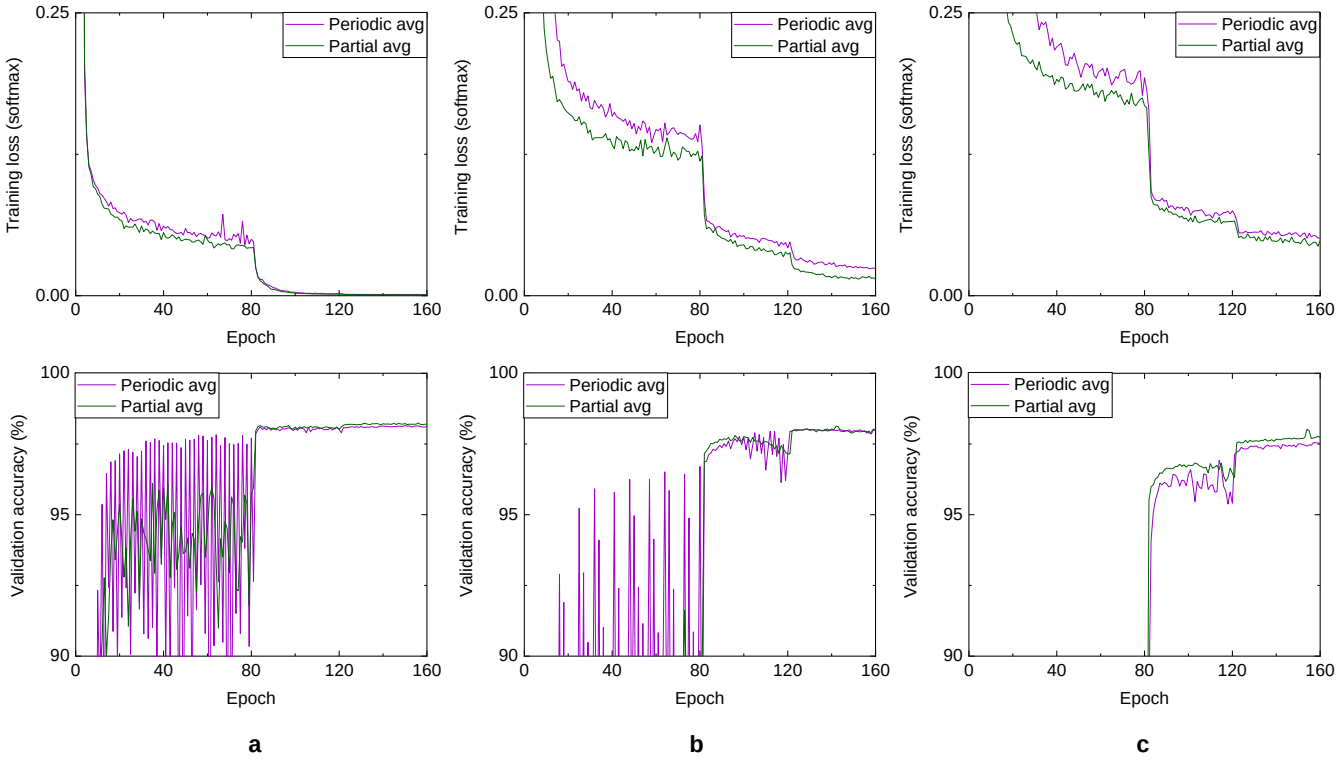


Figure 8: The learning curves of WideResNet-16-8 (SVHN) training. The number of workers is 128 and the hyper-parameters are shown in Table 1. The top charts are the training loss and the bottom charts are the validation accuracy. The averaging interval τ is set to 4, 16, and 64 (**a**, **b**, and **c**).

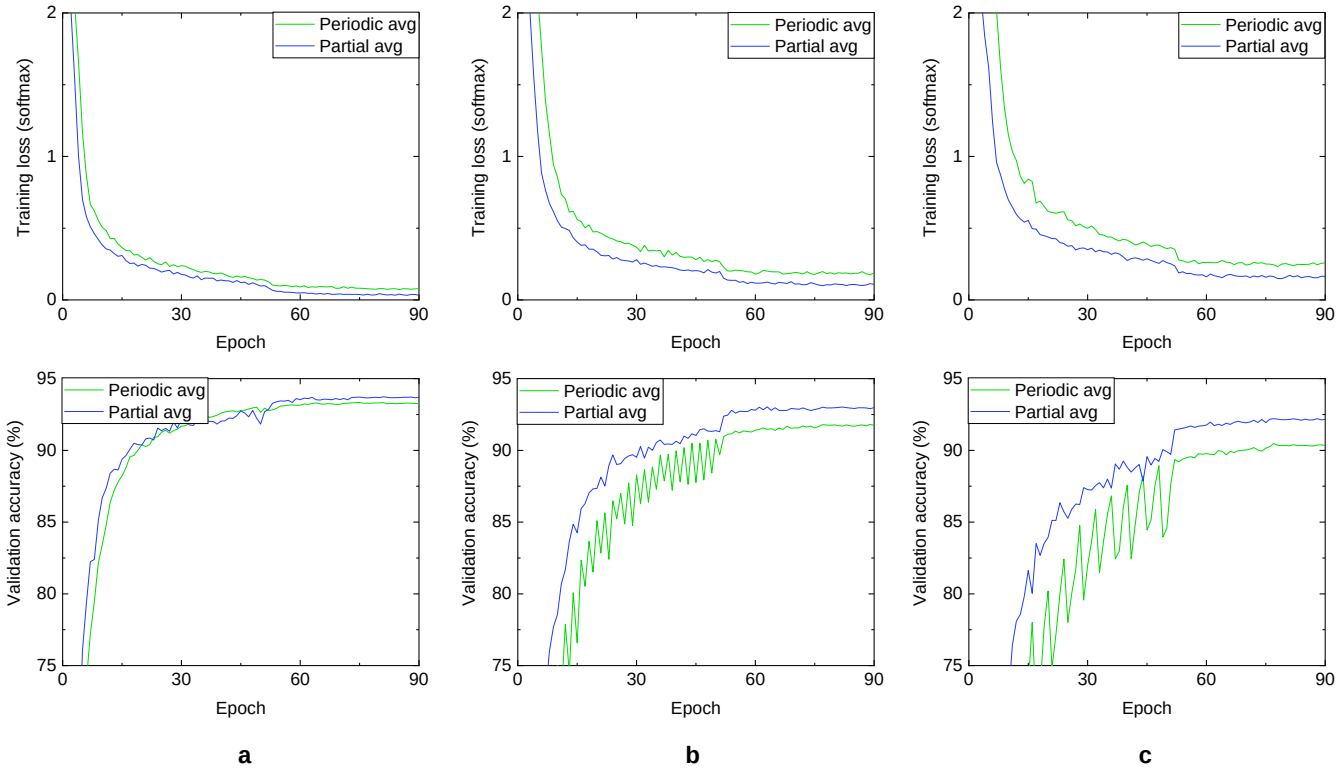


Figure 9: The learning curves of VGG-11 (Fashion-MNIST) training. The number of workers is 128 and the hyper-parameters are shown in Table 1. The top charts are the training loss and the bottom charts are the validation accuracy. The averaging interval τ is set to 2, 4, and 8 (**a**, **b**, and **c**).

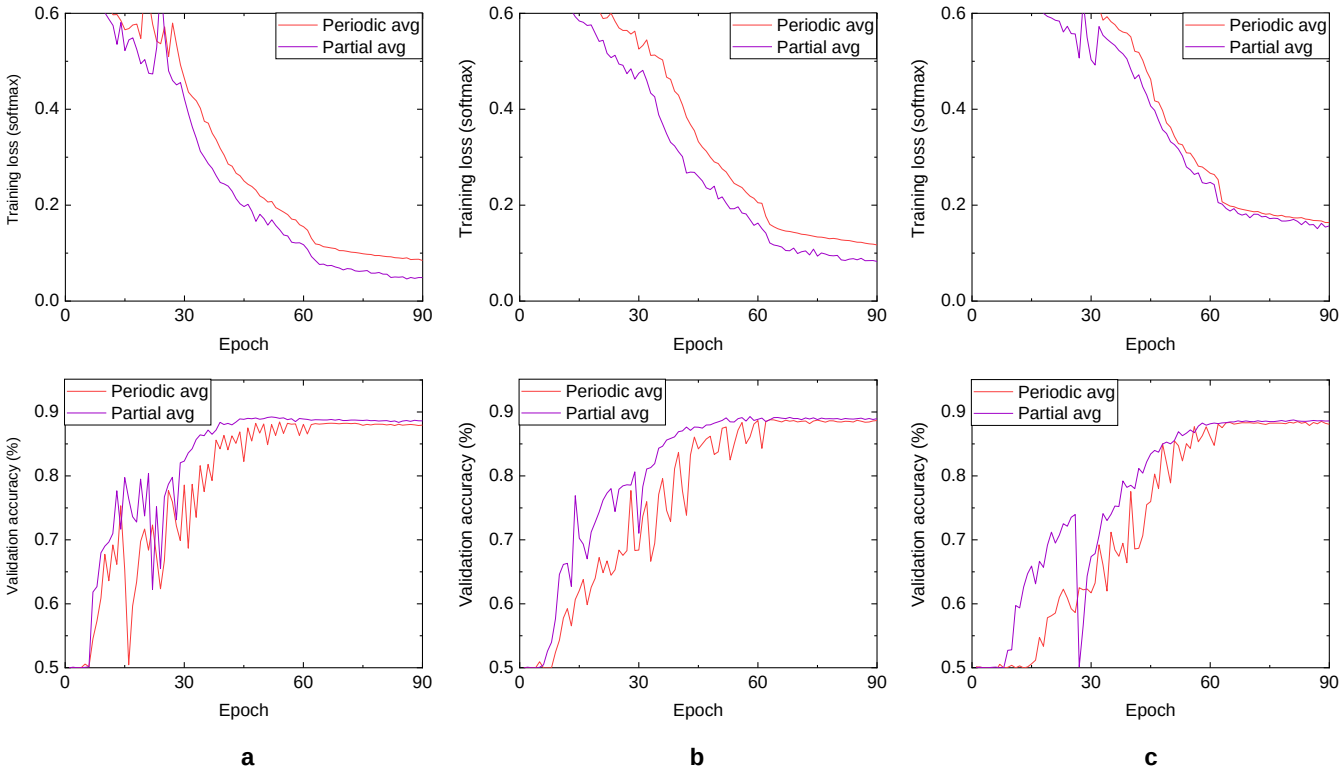


Figure 10: The learning curves of LSTM (IMDB) training. The number of workers is 128 and the hyper-parameters are shown in Table 1. The top charts are the training loss and the bottom charts are the validation accuracy. The averaging interval τ is set to 2, 4, and 8 (**a**, **b**, and **c**).

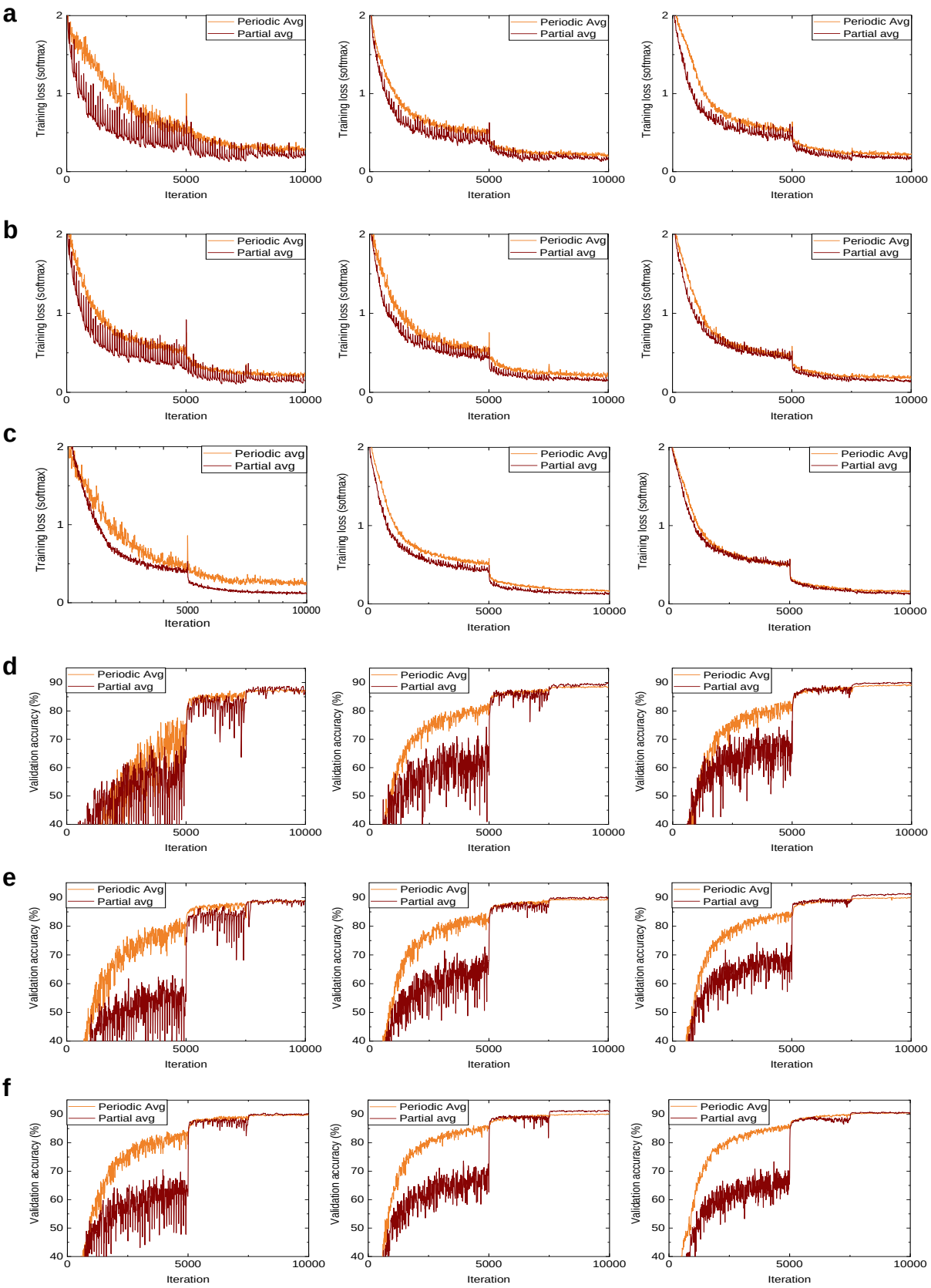


Figure 11: The learning curves of CIFAR-10 with various degrees of data heterogeneity and ratios of the active workers. **a, b, c**: Training loss curves with activation ratio of 25%, 50%, and 100%, respectively. **d, e, f**: Validation accuracy curves with activation ratio of 25%, 50%, and 100%, respectively. The three columns correspond to Dirichlet's concentration parameters of 0.1, 0.5, and 1.0, respectively.

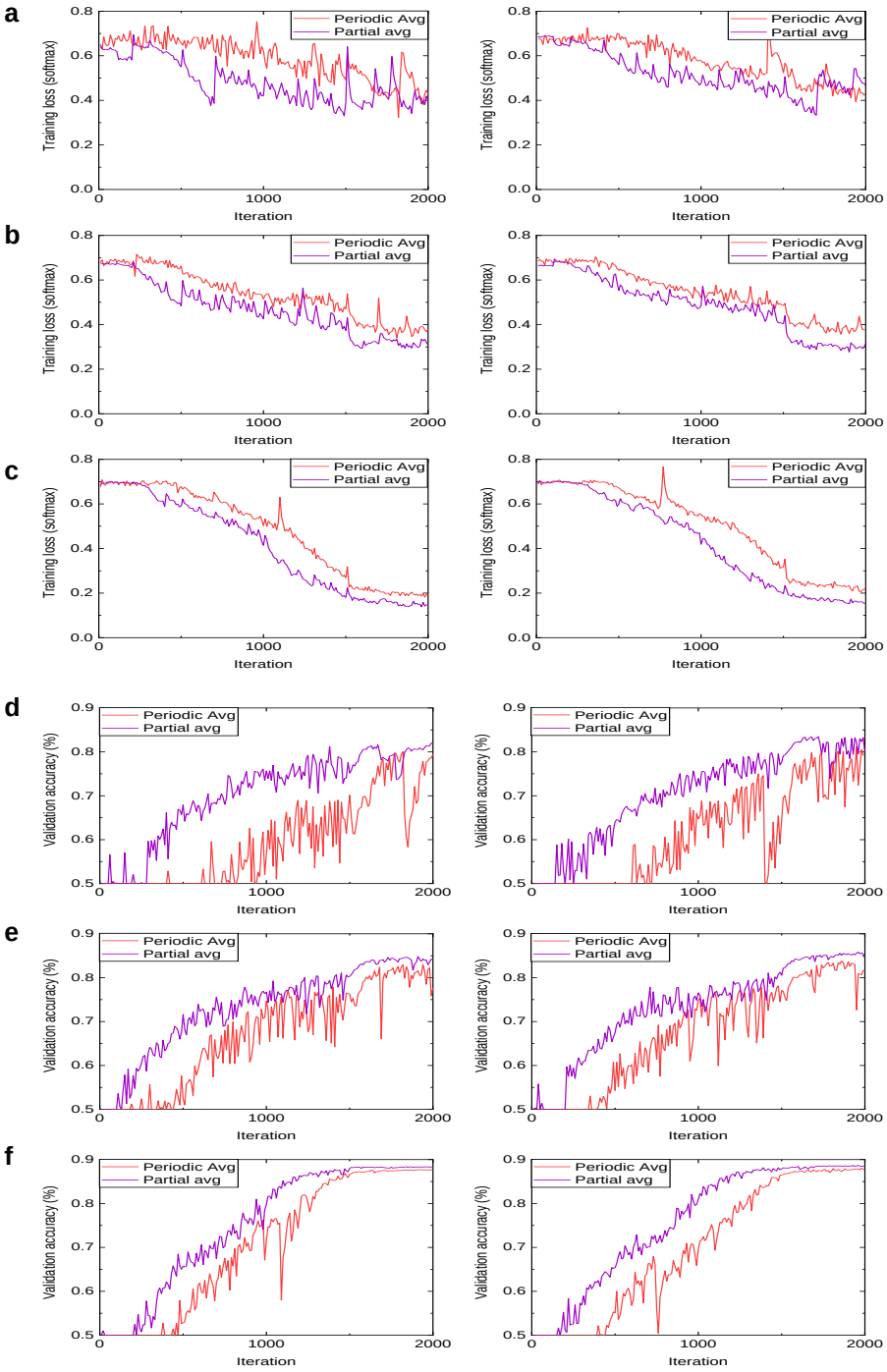


Figure 12: The learning curves of IMDB with various degrees of data heterogeneity and ratios of the active workers. **a, b, c**: Training loss curves with activation ratio of 25%, 50%, and 100%, respectively. **d, e, f**: Validation accuracy curves with activation ratio of 25%, 50%, and 100%, respectively. The two columns correspond to Dirichlet's concentration parameters of 0.5 and 1.0, respectively.

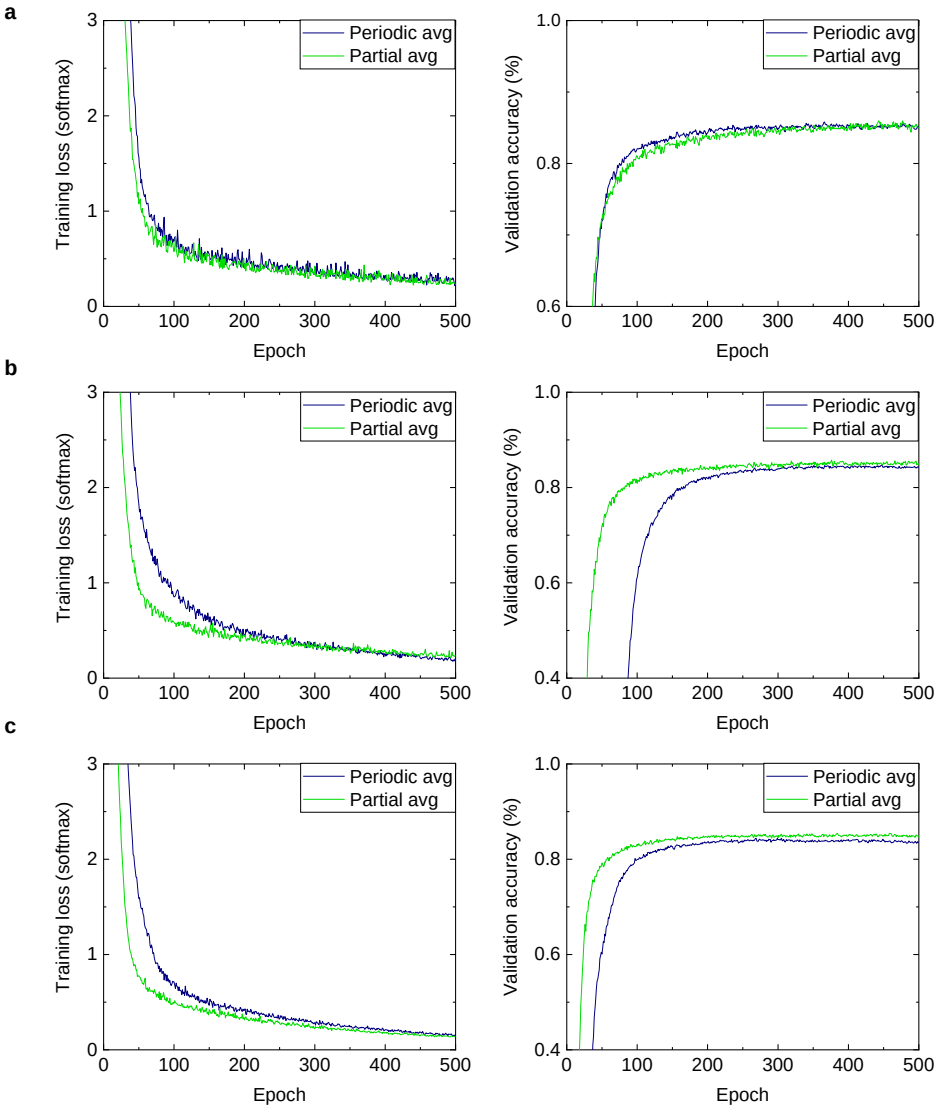


Figure 13: The learning curves of FEMNIST with different ratios of the active workers. **a, b, c**: learning curves with 25%, 50%, and 100% of random devices, respectively. The left charts are the training loss curves and the right charts are the validation accuracy curves.