FedLMT: Tackling System Heterogeneity of Federated Learning via Low-Rank Model Training with Theoretical Guarantees

Jiahao Liu [1 2] **Yipeng Zhou** [3] **Di Wu** [1 2] **Miao Hu** [1 2] **Mohsen Guizani** [4] **Quan Z. Sheng** [3]

Abstract

Federated learning (FL) is an emerging machine learning paradigm for preserving data privacy. However, diverse client hardware often has varying computation resources. Such *system heterogeneity* limits the participation of resource-constrained clients in FL, and hence degrades the global model accuracy. To enable heterogeneous clients to participate in and contribute to FL training, previous works tackle this problem by assigning customized sub-models to individual clients with model pruning, distillation, or low-rank based techniques. Unfortunately, the global model trained by these methods still encounters performance degradation due to heterogeneous sub-model aggregation. Besides, most methods are heuristic-based and lack convergence analysis. In this work, we propose the FedLMT framework to bridge the performance gap, by assigning clients with a homogeneous pre-factorized low-rank model to substantially reduce resource consumption without conducting heterogeneous aggregation. We theoretically prove that the convergence of the low-rank model can guarantee the convergence of the original full model. To further meet clients' personalized resource needs, we extend FedLMT to pFedLMT, by separating model parameters into common and custom ones. Finally, extensive experiments are conducted to verify our theoretical analysis and show that FedLMT and pFedLMT outperform other baselines with much less communication and computation costs.

[1]School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China [2]Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China [3]School of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, Australia [4]Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE. Correspondence to: Di Wu <wudi27@mail.sysu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

Federated learning (FL) has received unprecedented attention due to its ability to preserve privacy. However, the FL performance is susceptible to the inherent heterogeneity in FL clients (Kairouz et al., 2021) due to giant discrepancies of data distribution and resource capability among FL clients, *e.g.*, storage, communication, and computation. How to handle data heterogeneity has been explored in existing works (Tan et al., 2023; Ye et al., 2023). Yet, how to handle resource capability heterogeneity (namely system heterogeneity) is still not fully studied (Pfeiffer et al., 2023).

The mainstream technique to address system heterogeneity is to assign customized sub-models to individual clients in accordance with their capabilities. These methods can be divided into four main categories. 1) *Width-scale* methods that create sub-models from the global full model by pruning channels (Alam et al., 2022). 2) *Depth-scale* methods which generate sub-models by dividing the global model based on the model depth for layer-wise training (Kim et al., 2023). 3) *Distillation* methods where the server utilizes knowledge distillation to allow clients to own heterogeneous models (Zhang et al., 2022). 4) *Low-rank* based methods where sub-models are created through low-rank decomposition (Yao et al., 2021). However, most of them need to aggregate heterogeneous sub-models, which may impair the performance of the aggregated global model (Zhang et al., 2023). The reason is that heterogeneous sub-models have distinct characteristics and loss landscapes. Simply weighting the average of these sub-models will cause inconsistency and hence introduce errors (Kang et al., 2023). Besides, most methods are heuristic-based and lack convergence analysis.

In this work, we propose FedLMT, a general FL framework with low-rank model training, to tackle system heterogeneity. In FedLMT, each client trains a pre-factorized low-rank model with the same model architecture, which is more lightweight than the original full model and inherently reduces computation and communication costs. Since clients' sub-models are homogeneous, we can bypass the problems caused by heterogeneous aggregation. Indeed, our experiments show that FedLMT outperforms other methods with much lower computation overhead and is more robust under typical heterogeneous FL settings described in (Diao et al.,

2021). The core reason lies in that training a low-rank model can have almost the same performance as training the original full model, though the latter cannot be implemented due to the existence of resource-constrained clients. The convergence analysis is provided in Section 4 to justify the efficacy of low-rank model training. To better meet the resource needs of diverse clients, we further propose pFedLMT, a personalized version of FedLMT, which splits the model parameters into common and custom ones. The common parts are the same among clients while the custom parts are heterogeneous according to clients' own requirements.

Our contributions. Our contributions are fourfold. 1) We propose FedLMT, a low-rank FL training framework to tackle system heterogeneity. Unlike traditional heterogeneous methods, we show that training a homogeneous low-rank model on all clients is more robust and efficient than training heterogeneous models on all clients. 2) We theoretically analyze the convergence property of FedLMT, and to the best of our knowledge, we are among the first to reveal that a converged full model can be reached by training low-rank sub-models decomposed from the full model under non-convex and smooth assumptions. 3) We propose pFedLMT, a personalized version of FedLMT, which splits the model weights into common and custom ones to better adapt to the resource requirements of heterogeneous clients. 4) We conduct extensive experiments to validate our theoretical analysis and show that FedLMT and pFedLMT achieve better model performance than state-of-the-art methods.

2. Related Work

2.1. Federated Learning

Based on FedAvg (McMahan et al., 2017), the most standard FL algorithm, many variants of FedAvg have been proposed to reduce communication cost (Wang et al., 2022; Isik et al., 2023), overcome data heterogeneity (Collins et al., 2021; Marfoq et al., 2022; Xu et al., 2023; Liu et al., 2023), and improve the convergence rate (Karimireddy et al., 2020; Acar et al., 2021; Reddi et al., 2021). Our work focuses on the system heterogeneity challenge but is compatible and can be integrated with methods for solving other challenges.

There are also studies to deal with system heterogeneity where clients are resource-constrained. One thread of work creates customized sub-models by factorizing the global full model with different low-rank decomposition ratios (Yao et al., 2021; Mei et al., 2022). Among them, FedHM (Yao et al., 2021) also uses low-rank model training, but our approach is completely different from FedHM. In FedHM, the server needs to conduct Singular Value Decomposition (SVD) operations frequently to obtain clients' factorized sub-models, which will suffer inferior model utility since SVD operations incur additional approximation errors, while our approach avoids this by training pre-factorized low-rank sub-models to get better model performance. A toy example is presented in Appendix A.1 to illustrate this issue. Besides, although the convergence of FedHM is theoretically derived, the analysis is not convincing since FedHM fails to converge to a stationary solution. More discussions on this are presented in Appendix A.2.

The second thread of work adopts knowledge distillation (Hinton et al., 2015) where the server ensembles the knowledge distilled from clients' heterogeneous models (Li & Wang, 2019; Cho et al., 2022; Zhang et al., 2022). However, most of them require the availability of a public dataset, which conflicts with secure aggregation protocols and is susceptible to backdoor attacks (Wang et al., 2020). Finally, *width-scale* methods are based on model pruning (Caldas et al., 2018; Diao et al., 2021; Horvath et al., 2021; Alam et al., 2022; Hong et al., 2022). This kind of method is similar to previously proposed *slimmable neural network* (Yu & Huang, 2019) where sub-networks with various widths and shared weights are jointly trained with self-distillation (Zhang et al., 2019). However, the inherent drawback of pruning lies in the mismatch of channel parameters when aggregating heterogeneous sub-models from clients (Kim et al., 2023). To avoid this problem, some researchers proposed dividing the global full model based on a fixed model depth (Liu et al., 2022; Kim et al., 2023; Ilhan et al., 2023). Such methods usually need extra classifiers for auxiliary training and deep layers may be trained inadequately. Thereby, extra distillation methods are required to further train deep layers. Despite their success, most of them are heuristic-based and the global full model trained by these methods still suffers performance degradation than training the full model from scratch, while FedLMT is theoretically guaranteed and can bridge the performance gap. Besides, FedLMT outperforms other baselines with less computation and communication costs and can be easily extended to pFedLMT that can handle not only the system heterogeneity but also data heterogeneity, whereas most previous works are limited in only addressing the system heterogeneity.

2.2. Low-rank Model Training

Training a low-rank neural network from scratch has been studied in a centralized setting for a long time (Sainath et al., 2013; Tai et al., 2016; Khodak et al., 2021). However, since low-rank constraints are lossy, the model performance of naive low-rank training is degraded compared with training the original full model. Many practical techniques such as *hybrid model architecture* (Wang et al., 2021) and *Frobenius decay* (Khodak et al., 2021) have been proposed to bridge the performance gap, but theoretical reasons behind the effectiveness of these techniques are poorly understood (Kamalakara et al., 2022). Our framework adopts these technologies and our theory provides a new insight to un-

derstand and explain the effectiveness of existing low-rank techniques, from a theoretical perspective.

In federated and distributed settings, low-rank techniques are mainly used to improve the communication efficiency. (Konečnỳ et al., 2016) introduced low-rank approaches into FL for the first time, but the global model performance is not ideal. FedDLR (Qiao et al., 2021) further improved the performance of the global model by using an ad-hoc adaptive rank selection. In FedDLR, SVD operations are conducted per up/down transmission round, which will also encounter the same problem as FedHM. (Hyeon-Woo et al., 2022) proposed FedPara with low-rank Hadamard product parameterization. This way, they can construct a higher-rank model to achieve a better performance with less communication cost. Pufferfish (Wang et al., 2021) suggested training a homogeneous low-rank model to reduce the communication cost, yet this work has no convergence analysis. Our framework is theoretically guaranteed and we are more focused on solving the system heterogeneity in FL.

It is worth mentioning that the training paradigm of modern deep learning generally follows the pattern of pre-training plus fine-tuning. In this case, users do not need to train the model from scratch but instead focus on fine-tuning the pre-trained model to adapt to their downstream tasks. Low-rank techniques also have many applications in this field. For instance, a series of variants (Dettmers et al., 2024; Kopiczko et al., 2023; Lialin et al., 2023) based on LoRA (Hu et al., 2022) provide an effective approach for large pre-trained model fine-tuning. To some extent, fine-tuning a pre-trained model can be seen as training the model at a carefully selected initial point. In this paper, we pay more attention to training a low-rank model from scratch without the help of pre-training and the main conclusions of this paper can be extended to other fields.

3. Preliminary

We assume that models trained in FL are composed of neural network (NN) layers provided the prevalence and superb recognition capability of NN models.

3.1. Low-rank Factorization of Neural Layers

Fully Connected Layers. A fully connected (FC) layer takes a n-dimensional input denoted by $\mathbf{z}$ to output a m-dimensional vector, *i.e.*, $\mathbf{z}' = \sigma(\mathbf{W}\mathbf{z})$, where $\sigma : \mathcal{R}^m \mapsto \mathcal{R}^m$ can be any element-wise activation function and $\mathbf{W} \in \mathcal{R}^{m\times n}$ represents parameters of the FC layer. $\mathbf{W}$ can be factorized into the product of $\mathbf{U}\mathbf{V}^T$, where $\mathbf{U} \in \mathcal{R}^{m\times r}$, $\mathbf{V} \in \mathcal{R}^{n\times r}$, and $r \ll \min\{m, n\}$ to reduce the computation and memory complexity from $\mathcal{O}(mn)$ to $\mathcal{O}(mr + nr)$.

Convolutional Layers. There are multiple ways to factorize convolution layers (Lebedev et al., 2015; Tucker, 1966; Wang et al., 2021). The strategy in our work is the same as that of in (Khodak et al., 2021) for the convolution layer factorization. Specifically, for a 2D convolution layer with dimension $\mathbf{W} \in \mathcal{R}^{c_{out}\times c_{in}\times k\times k}$ where c_{in} and c_{out} are the number of input and output channels, and k is the size of convolution filters. We first unroll the 4D tensor $\mathbf{W}$ leading to a 2D matrix $\mathbf{W}^{'}$ with dimension $\mathcal{R}^{c_{out}k\times c_{in}k}$. Then, we factorize $\mathbf{W}^{'}$ to obtain $\mathbf{U} \in \mathcal{R}^{c_{out}k\times r}$, $\mathbf{V} \in \mathcal{R}^{c_{in}k\times r}$. Finally, we reshape $\mathbf{U}$ and $\mathbf{V}$ matrices back to 4D filter, yielding $\mathbf{U} \in \mathcal{R}^{c_{out}\times r\times k\times 1}$, $\mathbf{V} \in \mathcal{R}^{r\times c_{in}\times 1\times k}$. Therefore, factorizing a convolution layer returns two 1D convolution layers: 1) the first defined by $\mathbf{V}$ consists of r output channels and filters of size k along one input dimension; 2) the second defined by $\mathbf{U}$ consists of c_{out} output channels and filters of size k along the other input dimension. Usually $r \ll \min\{kc_{in}, kc_{out}\}$ to reduce the computation complexity from $\mathcal{O}(k^2c_{in}c_{out})$ to $\mathcal{O}(kr(c_{in} + c_{out}))$.

Definition 3.1. For an unfactorized layer denoted by $\mathbf{W} \in \mathcal{R}^{m\times n}$, the low-rank ratio of the layer is defined by $\alpha = \frac{r}{\text{rank}(\mathbf{W})}$ where $\text{rank}(\mathbf{W}) \triangleq \min\{m, n\}$.

Here, α measures the size of the low-rank model. The smaller α is, the smaller the low-rank model.

3.2. Training Methods for Low-rank Models

Hybrid Model Architecture. It has been observed that factorizing initial layers may negatively impact the model accuracy (Konečnỳ et al., 2016; Wang et al., 2021). A possible explanation is that initial layers can be regarded as feature extractors and low-rank factorization on these layers will introduce approximation errors. These errors will be accumulated and propagated throughout the entire model, leading to an inferior model performance. To address this issue, the *hybrid model architecture* is designed as follows (Wang et al., 2021). Let $\mathbf{w} = \{W^1, W^2, \cdots, W^{L+1}\}$ denote the weights of a full $(L+1)$-layer model. The weights of the corresponding hybrid low-rank model can be represented by $\mathbf{x} = \{W^1, \cdots, W^\rho, U^{\rho+1}, V^{\rho+1}, \cdots, U^L, V^L, W^{L+1}\}$, where ρ is a hyper-parameter denoting the number of layers that are not factorized. Note that the last classification layer, *i.e.*, W^{L+1}, is usually not factorized (Khodak et al., 2021; Wang et al., 2021). To keep concise, W^{L+1} will not be included in our presentation hereafter. Besides, we assume that the rank ratio α of all layers is identical and fixed.

Initialization and Regularization. The performance of low-rank models can be boosted through a customized initialization called *spectral initialization* (Idelbayev & Carreira-Perpinán, 2020) and a customized regularization named *Frobenius decay* (Khodak et al., 2021). The former uses SVD to initialize low-rank model parameters and the latter applies Frobenius norm penalty, *i.e.*, $\|U^l(V^l)^T\|_F^2$, for each decomposed layer l during the low-rank model training.

4. FedLMT: Algorithm and Analysis

4.1. Problem Formulation

Assume that there are N clients with non identically and independently distributed local data $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_N\}$. Traditional FL aims to learn a global L-layer model $\mathbf{w} = \{W^1, W^2, \cdots W^L\}$ by solving the following problem:

$$\min_{\mathbf{w} \in \mathcal{W}_f} f(\mathbf{w}) \triangleq \frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{w}), \tag{1}$$

where the local objective $f_i(\mathbf{w}) \triangleq \mathbb{E}_{\xi_i \sim \mathcal{D}_i}\left[F_i\left(\mathbf{w}; \xi_i\right)\right]$ is the expected loss function of client i, ξ_i is a random sample on client i, and $\mathcal{W}_f$ represents the weight space of the full model $\mathbf{w}$ with d-dimension. In resource-constrained settings, clients may not have enough computation capacity or memory to train the full model $\mathbf{w}$. Therefore, we consider training the corresponding low-rank model $\mathbf{x}$ of the full model $\mathbf{w}$ instead of training $\mathbf{w}$ directly. Using the hybrid model architecture technique, we define $\mathbf{x} = \{W^1, \cdots, W^\rho, U^{\rho+1}, V^{\rho+1}, \cdots, U^L, V^L\}$ as the corresponding low-rank model of $\mathbf{w}$, and the training objective is defined as $g(\mathbf{x}) \triangleq \frac{1}{N}\sum_{i=1}^{N} g_i(\mathbf{x})$ with $g_i(\mathbf{x}) \triangleq \mathbb{E}_{\xi_i \sim \mathcal{D}_i}\left[G_i\left(\mathbf{x}; \xi_i\right)\right]$. Here, $\mathbf{x} \in \mathcal{W}_g$ where $\mathcal{W}_g$ is the low-rank weight space with $d^{'}$-dimension, and usually $d^{'} \ll d$. The general training process for a low-rank model $\mathbf{x}$ using FedAvg (McMahan et al., 2017), named FedLMT, is shown in Algorithm 1. Note that in the Algorithm, we do not need to generate $\mathbf{w}$ explicitly.

In this way, if we can obtain the optimal low-rank model $\mathbf{x}$ in $\mathcal{W}_g$ after training, we can recover it to obtain a full model $\mathbf{w}$ by executing $W^l = U^l(V^l)^T, \forall l > \rho$. At this time, an interesting phenomenon is that $\mathbf{x}$ and $\mathbf{w}$ will achieve the same loss, *i.e.*, $F_i(\mathbf{w}; \xi) = G_i(\mathbf{x}; \xi)$ where ξ is a sample in the supervised learning scenario, because the full model $\mathbf{w}$ is obtained from $\mathbf{x}$. Since the resource requirement for training $\mathbf{x}$ is generally much less than that for training $\mathbf{w}$, we can get $\mathbf{w}$ with less training cost. Extensive prior works (Sainath et al., 2013; Wang et al., 2021; Khodak et al., 2021) have empirically observed that the full model $\mathbf{w}$ obtained by training the low-rank $\mathbf{x}$ achieves almost the same performance as training $\mathbf{w}$ from scratch directly, but the underlying theoretical principles are unknown. Besides, it is not clear whether $\mathbf{w}$ recovered from $\mathbf{x}$ converges in $\mathcal{W}_f$ or not.

In this work, we explore this problem from a theoretical perspective. Intuitively speaking, we first obtain a stationary point $\mathbf{x}^t$ that converges in the low-rank weight space $\mathcal{W}_g$, *i.e.*, $\nabla g(\mathbf{x}^t) = 0$, at the end of Algorithm 1 after t iterations. Next, we prove that the corresponding full model $\mathbf{w}^t$ converges in the full weight space $\mathcal{W}_f$ as well. The key that makes this conclusion valid lies in the low-rank relation between $\mathbf{x}^t$ and $\mathbf{w}^t$. Since $\mathbf{w}^t$ is obtained from $\mathbf{x}^t$ by executing $W^l = U^l(V^L)^T, \forall l > \rho$, using the chain rule,

Algorithm 1 FedLMT

Input: Local epoch E, total iteration T, learning rate γ, a randomly selected client set $\mathcal{N}^0$, initial local low-rank model $\mathbf{x}_i^0 = \mathbf{x}^0 = \{W_{i,t}^1, \cdots, W_{i,t}^\rho, U_{i,t}^{\rho+1}, V_{i,t}^{\rho+1}, \cdots, U_{i,t}^L, V_{i,t}^L\}, \forall i$.
Output: Final global model $\mathbf{x}^T$.
for $t = 1$ **to** T **do**
  for client $i \in \mathcal{N}^{t-1}$ in parallel **do**
    $\mathbf{x}_i^t = \mathbf{x}_i^{t-1} - \gamma \nabla G_i(\mathbf{x}_i^{t-1}, \xi_i^t)$
  end for
  if t divides E **then**
    Each client $i \in \mathcal{N}^{t-1}$ sends $\mathbf{x}_i^t$ to the server
    Server updates $\mathbf{x}^t = \frac{1}{|\mathcal{N}^{t-1}|}\sum_{i=1}^{|\mathcal{N}^{t-1}|} \mathbf{x}_i^t$
    Server randomly samples a new client set $\mathcal{N}^t$
    Server broadcasts $\mathbf{x}^t$ to all chosen clients and replaces the local model
  end if
end for
(Optional) Generate $\mathbf{w}^T$ from $\mathbf{x}^T$.

we can deduce that for all $l > \rho$,

$$\left\|\nabla g^l(\mathbf{x}^t)\right\|_F^2 \geq (\sigma_{min}^2(U_t^l) + \sigma_{min}^2(V_t^l)) \left\|\nabla f^l(\mathbf{w}^t)\right\|_F^2, \tag{2}$$

where $\sigma_{min}(\cdot)$ denotes the smallest singular value of the matrix (see Lemma D.5 for a detailed derivation). Noting that $\left\|\nabla g^l(\mathbf{x}^t)\right\|_F^2 = 0$ at the end of the training in Algorithm 1, if U_t^l and V_t^l are trained carefully so that $\sigma_{min}^2(U_t^l) > 0$ and $\sigma_{min}^2(V_t^l) > 0$, then we can conclude that $\nabla f^l(\mathbf{w}^t) = 0$, implying that it is reasonable to approximate the full model training by the low-rank model training in both convex and non-convex settings. Since non-convex cases are more common, in the following subsection, we will theoretically prove our statement under non-convex settings.

4.2. Convergence Analysis

Our analysis begins with the following assumptions.

Assumption 4.1. For each client i, $f_i(\cdot)$ and $g_i(\cdot)$ are continuously differentiable and smooth with modulus L_s.

Assumption 4.2. (Informal) For each client i, the stochastic gradients $\nabla F_i(\mathbf{w}_i; \xi_i)$ and $\nabla G_i(\mathbf{x}_i; \xi_i)$ are both unbiased, and the second moment of $\nabla F_i(\mathbf{w}_i; \xi_i)$ and $\nabla G_i(\mathbf{x}_i; \xi_i)$ are bounded by constants G^2 and G_g^2, respectively. Besides, $\nabla f_i(\mathbf{w}_i)$ have the σ^2-bounded 2th central moment and the fourth moment of $\nabla F_i(\mathbf{w}_i; \xi_i)$ is bounded by constant G^4.

Assumption 4.3. There exist constants κ_u, κ_v and κ_{uv} such that at each iteration t, for every client i and layer $l \in \{\rho+1, \cdots, L\}$:

$$\left\|U_{i,t}^l\right\|_F \leq \kappa_u, \left\|V_{i,t}^l\right\|_F \leq \kappa_v, \left\|U_{i,t}^l (V_{i,t}^l)^T\right\|_F \leq \kappa_{uv},$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $(\cdot)^T$ means the transpose operation for the matrix.

Assumption 4.1 and Assumption 4.2 are commonly used in convergence analysis under distributed settings (Yu et al., 2019; Bottou et al., 2018). Note that here we need to assume that the fourth moment of ∇F_i is bounded, which is also used in some popular studies (Glasgow et al., 2022; Blanchard et al., 2017). Assumption 4.3 has been widely used for theoretical analysis of machine learning models (Chen et al., 2020; Chérief-Abdellatif, 2020; Xue et al., 2023; Zhao et al., 2024). The intuition of Assumption 4.3 is similar to the regularization technique in deep learning, where a trained model with smaller parameter values is more desirable since it can mitigate the risk of over-fitting (Shalev-Shwartz & Ben-David, 2014). Since the dimension of the model is finite, it is reasonable to assume that the model weights have an upper bound. To complete the proof, we introduce an extra assumption as follows.

Assumption 4.4. Let $\sigma_{min}(\cdot)$ denote the smallest singular value. In training iteration t, we define $\psi_{uv} \triangleq \min_{l,t}\{\sigma_{min}(U_t^l), \sigma_{min}(V_t^l)\}\ \forall l, \forall t$, where U_t^l and V_t^l represents l-layer weights of $\mathbf{x}^t$, *i.e.*, $U_t^l = \sum_i U_{i,t}^l$, $V_t^l = \sum_i V_{i,t}^l$. By the definition of the singular value, we have $\psi_{uv} \geq 0$, and this assumption further requires $\psi_{uv} > 0$.

Assumption 4.4 is supported by existing works. According to the Marchenko-Pastur theory (Marchenko & Pastur, 1967; Bai & Yin, 2008; Vershynin, 2010) (described in Appendix B), if each layer weight $U^l \in \mathcal{R}^{m_l \times r_l}$ in model $\mathbf{x}$ is a Gaussian ensemble with scale $\frac{1}{\sqrt{r_l}}$ (r_l is the input dimension), which roughly aligns with common model parameter initialization schemes proposed in (Glorot & Bengio, 2010; He et al., 2015), and $r_l \ll m_l$. Then, we have $\sigma_{min}(U^l) > 0$ almost surely. We also do some experiments in Appendix F.4 to verify that this statement can be true.

Theorem 4.5. *(Informal) Under Assumption 4.1-Assumption 4.4, let q_0 be a constant and $1 < q_0 < 2$, if $0 < \gamma \leq \min\{\psi_{uv}^{\frac{2}{q_0-1}}, \frac{1}{L_s}, 1\}$, for a full model $\mathbf{w}$, by training its corresponding low-rank model $\mathbf{x}$ using Algorithm 1, for all $T \geq 1$, we have:*

$$\frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\left[\left\|\nabla f(\mathbf{w}^{t-1})\right\|_2^2\right] \leq \frac{2}{\gamma^{q_0}T}(f(\mathbf{w}^0) - f^\star) + \gamma^{2-q_0}\left(\frac{L_s\sigma^2}{2N} + \frac{3(L-\rho)G^2L_s(\kappa_u^4+\kappa_v^4)}{2}\right) + \gamma^{2-q_0}(L-\rho)\frac{G^4}{N^2}(\kappa_{uv}^2 + \kappa_u^2\kappa_v^2(N-1)^2) + \mathcal{O}(\gamma^{3-q_0}), \tag{3}$$

where $f^\star$ is the minimum value of problem (1) and $\mathcal{O}(\cdot)$ ignores higher powers of γ and constant numerical factors.

The proof can be found in Appendix D. Theorem 4.5 reveals the relationship between training $\mathbf{x}$ and training $\mathbf{w}$. With a fixed local epoch E on clients, by setting $\gamma = \frac{1}{\sqrt{T}}$ and $q_0 \to 1$ in Theorem 4.5, the asymptotic convergence rate to obtain the full model $\mathbf{w}$ is $\mathcal{O}(\frac{1}{\sqrt{T}})$ by using Algorithm 1. This indicates that $\mathbf{w}$ can converge in the full weight space $\mathcal{W}_f$ by only training $\mathbf{x}$ in the low-rank weight space $\mathcal{W}_g$, which is much faster than training $\mathbf{w}$ directly and consume less cost. In comparison, previous works (Jiang & Agrawal, 2018; Yu et al., 2019) only show that the convergence rate for training a low-rank model $\mathbf{x}$ in Algorithm 1 is $\mathcal{O}(\frac{1}{\sqrt{T}})$ by setting $\gamma = \frac{1}{\sqrt{T}}$, failing to link with the convergence of $\mathbf{w}$. Besides, in Theorem 4.5, as long as we set the learning rate as $\gamma = \frac{N^{(\frac{1}{q_0}-\frac{1}{2})}}{\sqrt{T}}$, we can deduce that FedLMT has a convergence rate $\mathcal{O}(\frac{1}{(NT)^{1-q_0/2}})$, indicating that FedLMT can achieve a linear speed-up with respect to the number of participating clients. An additional important observation is that if Assumption 4.4 does not hold, which means that ψ_{uv} might be zero, then in order for Theorem 4.5 to be true, we must choose the learning rate γ to be 0. At this time, we cannot efficiently update the model $\mathbf{x}$ and model $\mathbf{w}$. From this point of view, Assumption 4.4 is a key factor for the validity of Theorem 4.5.

Effect of Regularization. Previous works have observed that imposing *Frobenius decay* (*i.e.*, $\left\|UV^T\right\|_F^2$) or *L2 decay* (*i.e.*, $\|U\|_F^2 + \|V\|_F^2$) during a low-rank training can enhance the model performance (Kamalakara et al., 2022) under the centralized training setting, but the underlying theoretical principle is unknown. Here, we provide a new insight to explain the benefits of these techniques. Specifically, we find that these techniques can accelerate the model convergence. From Eq. (3), we can find that if κ_{uv}^2, $\kappa_u^4 + \kappa_v^4$ and $(\kappa_u\kappa_v)^2$ are small, the convergence upper bound will be tighter which means that the error will be small. Since κ_{uv}^2 is the upper bound of $\left\|UV^T\right\|_F^2$ and $\kappa_u^4 + \kappa_v^4$ is the upper bound of $\|U\|_F^4 + \|V\|_F^4$, *Frobenius decay* can reduce κ_{uv}^2, and *L2 decay* can reduce $\kappa_u^4 + \kappa_v^4$ and $(\kappa_u\kappa_v)^2$, and hence they can reduce the error upper bound to speed up the convergence. Besides, since $(\kappa_u\kappa_v)^2$ is the upper bound of $\|U\|_F^2\|V\|_F^2$, we can define an additional regular form $\|U\|_F^2\|V\|_F^2$ which is named as *Kronecker decay* since $\|U\|_F\|V\|_F = \|U \otimes V\|_F$, where $\otimes$ is the Kronecker product. Besides, it is interesting to note that the *Kronecker decay* is actually an upper bound of the *Frobenius decay* since $\left\|UV^T\right\|_F^2 \leq \|U\|_F^2\|V\|_F^2$. The experimental results in Section 6.2 verify that all regularization forms are effective and better results can be obtained by simultaneously using two regularization forms.

From Theorem 4.5, we have the following corollary.

Corollary 4.6. *(Informal) Under Theorem 4.5, the upper bound is a linear function of ρ and can be represented as $C_1\rho + C_2, (C_2 \geq 0)$ where C_1 and C_2 are related to γ.*

If we choose $0 < \gamma < \min\{\frac{3(\kappa_u^4+\kappa_v^4)}{4L_s E^2}, \psi_{uv}^{\frac{2}{q_0-1}}, \frac{1}{L_s}, 1\}$*, the error bound can be minimized by setting* $\rho = L$*.*

Effect of the Hybrid Model Architecture. From Corollary 4.6, we find that there exists a trade-off between the model convergence and the model compression, tuned by ρ. As ρ gets larger, the error bound in Eq. (3) gets smaller while the size of the low-rank model is larger. In particular, when $\rho = L$ which means the full model $\mathbf{w}$ is trained, it can minimize the upper bound error and the model converges at the fastest rate. The experiments in Section 6.2 verify this point. In practice, heterogeneous FL systems can choose ρ properly according to Corollary 4.6 to adapt to the clients' limited resources.

Discussion. Theorem 4.5 and Corollary 4.6 show that we can obtain a converged large model in the large weight space by training its corresponding low-rank model. The size of the low-rank model is tuned by α and ρ, which can be adjusted as needed. Since the size of the low-rank model is smaller than that of the original large model, we thereby diminish the communication, computation, and storage overhead during the FL training while maintaining a high model utility. These advantages indicate that it is feasible to employ FedLMT in solving the system heterogeneity in FL.

5. pFedLMT: Personalized FedLMT

In practice, clients may have different needs. For example, clients with sufficient computations may desire a larger model for a better performance (*system heterogeneity*) or clients may prefer a customized model due to their unique data distribution (*data heterogeneity*). However, most previous works only consider one of these two challenges, and ignore the other one. In this section, we show how to extend FedLMT to tackle both problems at the same time.

Inspired by (Collins et al., 2021), in this paper, we propose pFedLMT to tackle data heterogeneity and system heterogeneity. For a low-rank model $\mathbf{x} = \{W^1, \cdots, W^\rho, U^{\rho+1}, V^{\rho+1}, \cdots, U^L, V^L\}$, we denote those parts which are not factorized $\mathbf{p} = \{W^1, \cdots, W^\rho\}$ as the *common* layers and those factorized parts $\mathbf{q} = \{U^{\rho+1}, V^{\rho+1}, \cdots, U^L, V^L\}$ as the *custom* layers of $\mathbf{x}$.

Overview. In pFedLMT, we first initialize a full model $\mathbf{w} = \{W^1, W^2, \cdots W^L\}$. Then, we only factorize the *custom* layers with different rank ratios α_i to generate a customized low-rank model $\mathbf{x}_i$ for each client i adapting to its resource capacity β_i. Following the setup of previous works (Diao et al., 2021; Kim et al., 2023), we assign a large model to client i with large β_i by setting large α_i. During the training process, each client only uploads the *common* layers to the server for aggregation while the *custom* layers are retained locally. The pseudo-code of pFedLMT is presented in Algorithm 3 in Appendix C.

We point out several advantages of pFedLMT. 1) By setting the commonly undecomposed layers, all clients can learn a globally shared representation of all clients' local data (Collins et al., 2021; Oh et al., 2022); 2) By aggregating the common layers, we can avoid the aggregation of heterogeneous parts of models which may lead to parameter mismatch (Kim et al., 2023; Zhang et al., 2023); 3) The custom layers enable clients with sufficient resources to train a larger model for better performance and learn a personalized model to adapt to local data distribution.

Convergence guarantee for pFedLMT. pFedLMT can be abstracted into the general algorithm FedSim mentioned in (Pillutla et al., 2022). Therefore, the convergence analysis of pFedLMT under the non-convex setting can be proved by reusing the method in (Pillutla et al., 2022). Although pFedLMT is a special case of FedSim, it is worth emphasizing that our contribution lies in the low-rank constraints imposed by our method to handle the system heterogeneity. Besides, the analysis in (Pillutla et al., 2022) can not be used to prove Theorem 4.5 since they only prove that $\mathbf{x}$ is convergent and do not explore the link between the convergence of $\mathbf{w}$ and $\mathbf{x}$, and thereby there is no theoretical guarantee of the FL performance achieved by training low-rank models.

6. Experiments

6.1. Experiment Setup

Datasets, Tasks and Models. We conduct experiments on five benchmark datasets: SVHN (Netzer et al., 2011), CIFAR10, CIFAR100 (Krizhevsky et al., 2009), Tiny-ImageNet (TINY) (Chrabaszcz et al., 2017) and WikiText2 (Merity et al., 2016). The former four datasets are used for image classification, while we conduct a masked language modeling task with a 15% masking rate for the last dataset according to (Diao et al., 2021). We use the ResNet-18 model which is the same as that in (Mei et al., 2022) for image classification. For WikiText2, we train a Transformer which is the same as that in (Alam et al., 2022). More details can be found in Appendix E.1. Codes to reproduce the main results are available here: https://github.com/Sherrylife/FedLMT.

Baselines. We compare FedLMT with other state-of-the-art methods including FedAvg with full resource availability where all clients can train the full model in each round, *width-scale* methods (FedDropout (Caldas et al., 2018), HeteroFL (Diao et al., 2021) and FedRolex (Alam et al., 2022)), *depth-scale* methods (DepthFL (Kim et al., 2023), ProgFed (Wang et al., 2022)) and *low-rank* based methods like FedHM (Yao et al., 2021) and FLANC (Mei et al., 2022). The implementation details are described in Appendix E.3.

Table 1. Number of parameters / [# of FLOPs] of clients with different model capacities using ResNet-18.

METHOD	β_1	β_2	β_3	β_4
WIDTH-SCALE	701K [35.8M]	2.80M [141M]	6.29M [315M]	11.2M [559M]
DEPTH-SCALE	449K [231M]	1.58M [404M]	5.15M [557M]	13.2M [692M]
FEDHM	927K [296M]	1.96M [324M]	5.28M [415M]	11.2M [576M]
FLANC	531K [57.1M]	2.36M [207M]	6.01M [436M]	11.5M [744M]
FEDLMT(OURS)	422K [23.5M]	1.43M [140M]	3.43M [283M]	—
PFEDLMT	927K [296M]	1.96M [324M]	5.28M [415M]	11.2M [576M]

Table 2. The performance of different methods under 'β_4 - β_3 - β_2' settings. ACC means top-1 test accuracy, COMM means the total communication cost including download and upload among all clients, and FLOPs denotes the total floating operations during FL training.

TASK		FEDAVG	FEDDROPOUT	HETEROFL	FEDHM	FEDROLEX	DEPTHFL	FLANC	FEDLMT
CIFAR10	ACC	91.91	73.31	85.02	83.33	89.11	86.79	75.83	**91.03**
	COMM(GB)	223.5	134.7	134.7	122.6	134.7	132.9	132.0	**28.62**
	FLOPS(1E12)	11.18	6.75	6.75	8.77	6.75	11.01	9.22	**2.80**
CIFAR100	ACC	72.20	64.84	63.59	66.10	68.56	69.35	58.32	**71.08**
	COMM(GB)	335.2	201.5	201.5	183.3	201.5	198.6	197.6	**42.93**
	FLOPS(1E12)	16.77	10.10	10.10	13.12	10.10	16.49	13.80	**4.20**
SVHN	ACC	94.39	93.68	92.08	94.26	94.62	92.41	88.05	**95.35**
	COMM(GB)	223.5	134.7	134.7	122.6	134.7	132.9	132.0	**28.62**
	FLOPS(1E12)	11.18	6.75	6.75	8.77	6.75	11.01	9.22	**2.80**
TINY	ACC	42.71	30.38	28.88	36.30	32.82	44.84	31.53	**48.53**
	COMM(GB)	335.2	201.5	201.5	183.3	201.5	198.6	197.6	**42.93**
	FLOPS(1E12)	67.02	40.36	40.36	52.50	40.36	65.94	55.20	**16.74**
WIKITEXT2	PERPLEXITY	3.52	4157.1	3.06	—	3.14	—	—	**2.93**
	COMM(GB)	10.36	7.50	7.50	—	7.50	—	—	**2.65**
	FLOPS(1E12)	0.39	0.275	0.275	—	0.275	—	—	**0.106**

Data Heterogeneity. Unless otherwise stated, for image classification tasks, the data is distributed in a non-IID manner, as in (Kim et al., 2023), a Dirichlet distribution $z_c \sim \text{Dir}(\eta)(\eta = 0.5)$ is used to allocate samples to client m with a fraction of $p_{c,m}$ of all training instances belonging to class c. For WikiText2, we uniformly assign balanced data examples to clients according to (Diao et al., 2021).

System Heterogeneity. We consider four different client model capacities $\boldsymbol{\beta} = \{\beta_1, \beta_2, \beta_3, \beta_4\}$ according to (Kim et al., 2023). Table 1 depicts the model size and the number of FLOPs of baselines with different model capacities. Here β_4 means training the original full model and the partition details of each method can be found in Appendix E.3. We notice that there are a total number of 10 model capacity combinations, as shown in Table 2. Unless stated otherwise, the same number of clients are allocated to each of the different model capacities. For example, 'β_4 - β_3 - β_2 -β_1' means 25% of the clients are allocated to each of β_1, β_2, β_3 and β_4 local models; 'β_2 -β_1' means that 50% clients are allocated to each of β_1 and β_2 local models. Note that at this time no client can train the complete full model.

Evaluation Metrics. For image classification tasks, we use the average top-1 test accuracy which is defined as the mean test accuracy of the global full model on each of the client's local datasets as our evaluation metric. For natural language tasks, we use perplexity as the evaluation metric. The model performance is better if the perplexity is smaller.

Training Setting. Similar to (Diao et al., 2021), in each communication round, we sample 10 out of 100 clients for image classification tasks and 5 out of 100 clients for natural language tasks. The details of the hyper-parameters for the model training are presented in Appendix E.3.

6.2. Experimental Results and Analysis

Performance Comparison. We first consider a setup with only three model capacities 'β_4 - β_3- β_2'. For FedAvg, we assume all clients train the largest full model with capacity β_4. For FedLMT, all clients train the low-rank model with the smallest capacity β_2. For other methods, the same number of clients are allocated to each of the three different model capacities. Table 2 shows the final performance of the global full model obtained by each method. We observe that even when training the smallest model, FedLMT is still better than other baselines with less communication cost and computation overhead on five datasets. This improvement lies in that we train a homogeneous model on clients so that we can avoid the problem caused by heterogeneous models in the model aggregation stage (Zhang et al., 2023).

Impact of Model Distribution. We further evaluate the performance of the global model obtained by each method under different client model capacity distributions on the CIFAR10 dataset, and the results are reported in Table 3.

Table 3. Impact of client model heterogeneity distribution on model accuracy using CIFAR10 dataset.

MODEL DISTRIBUTION	FEDAVG	FEDDROPOUT	HETEROFL	FEDHM	FEDROLEX	DEPTHFL	FLANC	FEDLMT (OURS)
β_4	91.91	89.79	91.91	91.56	91.90	88.99	90.65	—
β_4- β_3	—	85.08	88.40	82.11	91.75	88.78	82.41	**91.98**
β_4 - β_3- β_2	—	73.31	85.02	83.33	89.11	86.79	75.83	**91.03**
β_4 - β_3 - β_2 -β_1	—	61.13	82.05	83.64	84.80	82.77	65.95	**86.27**
β_3	—	80.57	29.94	79.73	86.03	87.79	90.96	**91.98**
β_3 - β_2	—	63.90	32.11	81.87	83.54	84.20	82.35	**91.03**
β_3 - β_2 -β_1	—	47.30	31.95	81.83	72.49	80.54	75.32	**86.27**
β_2	—	55.04	15.92	79.45	61.15	81.49	90.56	**91.03**
β_2 -β_1	—	33.71	19.61	81.92	52.20	77.72	82.88	**86.27**
β_1	—	20.59	12.92	82.38	36.67	73.85	**88.08**	86.27

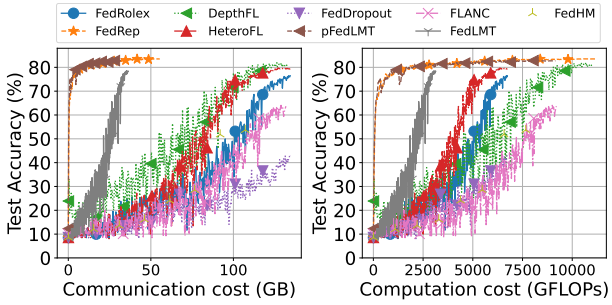


Figure 1. Performance of different methods under typical personalized FL settings with Dir($\eta = 0.1$) on the CIFAR10 dataset.

Note that no matter how the distribution is, FedLMT always trains a homogeneous model adapting to clients with the smallest model capacity. We find that for *depth-scale* and *width-scale* methods, the lower the capacity of clients involved in training, the worse the performance of the final model trained. This is particularly evident in HeteroFL: when no client can load a completely large model for training (*e.g.*, without capacity β_4), a giant performance decline happens due to the static pruning of HeteroFL. Whereas, for low-rank based methods FedHM and FLANC, they are more robust to the model distribution. Besides, we can find that when clients train a homogeneous model (*e.g.*, 'β_1' setting), the performance of low-rank based heterogeneous methods (FedHM, FLANC) is even better than that of the model in heterogeneous setting (*e.g.*, 'β_3 - β_2- β_1' setting) with much less cost, which manifests the negative effects caused by heterogeneous sub-model aggregation (Zhang et al., 2023). While FedLMT achieves better results by only training the homogeneous smallest low-rank model, we believe this finding will inspire more research in the future. Despite that FedLMT is slightly less effective than FLANC under the 'β_1' setting, FedLMT consumes less than 50% of the computation of FLANC and a much lower communication cost.

Personalization Study. To evaluate pFedLMT, we consider a scenario where data distribution is extremely pathological with Dir($\eta = 0.1$) followed by (Chen & Chao, 2022) and resource constraints differ among clients with 'β_4 - β_3- β_2' setting. Figure 1 shows that pFedLMT consumes less computation and communication costs to achieve a better model utility than other baselines. Here, FedRep (Collins et al., 2021), an advanced personalization method, is shown as the performance upper bound since each client is well-resourced and trains the same original full model. It can be seen that pFedLMT can achieve the same performance as FedRep with less communication and computation cost while other methods cannot do this.

Hyper-parameters. To explore the impact of the *hybrid model architecture* and rank ratio α, we train a ResNet-18 model on CIFAR10 and CIFAR100 datasets without regularization under FL settings. The results are shown in Figure 2 where the value of ρ is the number of undecomposed layers. For example, $\rho = 1$ means decomposing all layers except the first one; $\rho = 15$ means decomposing the model starting from the 16th layer. We find that when the model is highly compressed (*e.g.*, $\alpha = 0.05$ and $\rho = 1$), the performance of the low-rank model will significantly drop. At this time, the larger the ρ, the better the model performance, which means that the *hybrid model architecture* can effectively mitigate the performance degradation. For a fixed ρ, as α increases, implying that the model is larger, the improvement brought by *hybrid model architecture* is less since at this time the model is large enough to learn complex representations. Figure 4 in Appendix F shows the effect of ρ on the model performance and the convergence rate under different learning rates γ with fixed $\alpha = 0.2$. Per Corollary 4.6, the model converges faster with the increase of ρ. More discussions and results can be found in Appendix F.1.

Effect of Regularization. According to Theorem 4.5, *Frobenius decay* (FD), *L2 decay* (L2) and *Kronecker decay* (KD) should be effective according to our theory. Table 4 verifies our conjecture by training a low-rank ResNet-18 model with $\rho = 3$ and $\alpha = 0.2$ on the CIFAR10 and CIFAR100 datasets under the centralized setting. The training details can be found in Appendix E.2. Here NONE means training the low-rank model with no regularization and the last three lines mean that training the low-rank model us-

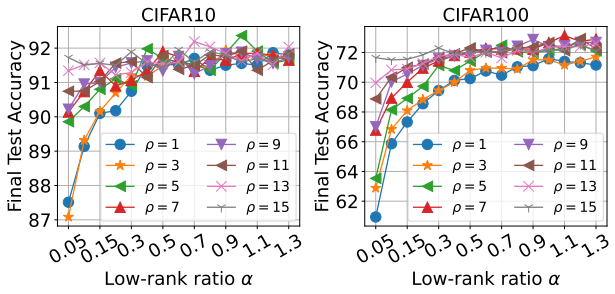


Figure 2. Effect of low-rank ratio α under different hybrid model architectures.

Table 4. Effect of different types of regularization under the centralized setting. The results of the original full model are *italicized*. ACC means the final top-1 test accuracy and ROUND (x%) means the training rounds required to reach the target test accuracy x%.

REGULAR TERM	CIFAR10 *(93.06%)*		CIFAR100 *(72.24%)*	
	ACC (%)	ROUND (90%)	ACC (%)	ROUND (65%)
NONE	91.27	142	68.54	117
L2	91.76	136	69.43	101
FD	92.67	129	71.21	110
KD	91.99	131	69.91	99
L2+KD	92.60	119	70.05	88
L2+FD	**92.91**	**107**	72.10	83
KD+FD	92.70	124	**72.54**	**81**

ing two forms of regularization at the same time. From Table 4, we find that when using regular terms, the number of rounds required to achieve the target accuracy is lower, which means regular terms can make the model converge faster, and the effect is more obvious under the action of two regular terms, especially under the combination of L2+FD or FD+KD. According to our theory, this is because L2 and KD reduce the value of κ_u and κ_v directly (*i.e.*, the upper bound of U and V) and FD aims to reduce the value of κ_{uv} (*i.e.*, the upper bound of UV^T). Therefore, a single regular form only guarantees the reduction of one of the terms κ_u (or κ_v) and κ_{uv}. The combination of L2+FD or FD+KD can minimize the term κ_u (or κ_v) and κ_{uv} at the same time, so as to achieve a faster convergence rate. Interestingly, in addition to speeding up the model convergence, the use of regularization also seems to allow the model to converge to a better solution, thereby reducing the generalization error. This is evident when two regular terms are used.

Supplementary experiments. We conduct supplementary experiments including the effect of model compression among FedLMT, *width-scale* way and *depth-scale* way, and comparison with ProgFed. All these results are presented in Appendix F due to the page limit.

7. Conclusion and Future Work

In this paper, we propose FedLMT, a low-rank FL training framework, and build a theoretical foundation for it. We show that FedLMT can bridge the performance by avoiding heterogeneous aggregation compared with other advanced heterogeneous methods. Besides, our theory reveals that a large converged model can be obtained with less training cost by training it in the low-rank weight space, and also provides new insights to support existing popular low-rank model training techniques. Finally, we extend FedLMT to pFedLMT to demonstrate how to handle system and data heterogeneity at the same time. Extensive experimental results verify the theory and the effectiveness of our methods.

The numerical results indicate some promising results in terms of efficiency and performance even when the low-rank ratio is small. However, the physical implications of the low-rank connection between the low-rank model and the original model are still unclear. Some researchers have found that deep neural networks can be trained in low-dimensional subspaces (Li et al., 2022; Arora et al., 2019), while in general, it is difficult to obtain a converged high-dimensional model by training it in a low-dimensional space since we may encounter saddle points during model optimization. As a result, a precise explanation of how the low-rank connection between low-dimensional space and high-dimensional space can guarantee the convergence of the high-dimensional model is necessary. Moreover, the impact of the smallest singular value of the model parameters and the low-rank ratio on the model performance and the convergence rate is not well-reflected by Theorem 4.5. More investigation of the correlation between the low-rank ratio and convergence will allow us to better understand how the low-rank model training affects the original high-dimensional model.

Acknowledgements

This work was supported by the National Key R&D Program of China under Grant 2023YFB3001900, Shenzhen Science and Technology Program (Grant No. KJZD20230923113901004), the Science and Technology Planning Project of Guangdong Province under Grant 2023A0505020006, the National Natural Science Foundation of China under Grant U2001209, 62072486, ARC DP230100233, 2023 Australia-Germany Joint Research Cooperation scheme under Grant No.57702286.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. We do not perceive any potential negative consequences for society and ethics.

References

Acar, D. A. E., Zhao, Y., Navarro, R. M., Mattina, M., Whatmough, P. N., and Saligrama, V. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

Alam, S., Liu, L., Yan, M., and Zhang, M. Fedrolex: Model-heterogeneous federated learning with rolling sub-model extraction. *Advances in Neural Information Processing Systems*, 35:29677–29690, 2022.

Arora, S., Cohen, N., Hu, W., and Luo, Y. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32, 2019.

Bai, Z.-D. and Yin, Y.-Q. Limit of the smallest eigenvalue of a large dimensional sample covariance matrix. In *Advances In Statistics*, pp. 108–127. World Scientific, 2008.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

Caldas, S., Konečny, J., McMahan, H. B., and Talwalkar, A. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018.

Chen, H. and Chao, W. On bridging generic and personalized federated learning for image classification. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Chen, M., Li, X., and Zhao, T. On generalization bounds of a family of recurrent neural networks. In Chiappa, S. and Calandra, R. (eds.), *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1233–1243. PMLR, 2020.

Chérief-Abdellatif, B.-E. Convergence rates of variational inference in sparse deep learning. In *International Conference on Machine Learning*, pp. 1831–1842. PMLR, 2020.

Cho, Y. J., Manoel, A., Joshi, G., Sim, R., and Dimitriadis, D. Heterogeneous ensemble knowledge transfer for training large models in federated learning. In Raedt, L. D. (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pp. 2881–2887. ijcai.org, 2022.

Chrabaszcz, P., Loshchilov, I., and Hutter, F. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Collins, L., Hassani, H., Mokhtari, A., and Shakkottai, S. Exploiting shared representations for personalized federated learning. In *International Conference on Machine Learning*, pp. 2089–2099. PMLR, 2021.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

Diao, E., Ding, J., and Tarokh, V. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Glasgow, M. R., Yuan, H., and Ma, T. Sharp bounds for federated averaging (local sgd) and continuous perspective. In *International Conference on Artificial Intelligence and Statistics*, pp. 9050–9090. PMLR, 2022.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hong, J., Wang, H., Wang, Z., and Zhou, J. Efficient split-mix federated learning for on-demand and in-situ customization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Horvath, S., Laskaridis, S., Almeida, M., Leontiadis, I., Venieris, S., and Lane, N. Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. *Advances in Neural Information Processing Systems*, 34:12876–12889, 2021.

Hsu, T.-M. H., Qi, H., and Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Hyeon-Woo, N., Ye-Bin, M., and Oh, T. Fedpara: Low-rank hadamard product for communication-efficient federated learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Idelbayev, Y. and Carreira-Perpinán, M. A. Low-rank compression of neural nets: Learning the rank of each layer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8049–8059, 2020.

Ilhan, F., Su, G., and Liu, L. Scalefl: Resource-adaptive federated learning with heterogeneous clients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24532–24541, 2023.

Isik, B., Pase, F., Gündüz, D., Weissman, T., and Zorzi, M. Sparse random networks for communication-efficient federated learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Jiang, P. and Agrawal, G. A linear speedup analysis of distributed deep learning with sparse and quantized communication. *Advances in Neural Information Processing Systems*, 31, 2018.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Kamalakara, S. R., Locatelli, A., Venkitesh, B., Ba, J., Gal, Y., and Gomez, A. N. Exploring low rank training of deep neural networks. *arXiv preprint arXiv:2209.13569*, 2022.

Kang, H., Cha, S., Shin, J., Lee, J., and Kang, J. Nefl: Nested federated learning for heterogeneous clients. *arXiv preprint arXiv:2308.07761*, 2023.

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

Khaled, A., Mishchenko, K., and Richtárik, P. Tighter theory for local sgd on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pp. 4519–4529. PMLR, 2020.

Khodak, M., Tenenholtz, N., Mackey, L., and Fusi, N. Initialization and regularization of factorized neural layers. *arXiv preprint arXiv:2105.01029*, 2021.

Kim, M., Yu, S., Kim, S., and Moon, S. Depthfl : Depthwise federated learning for heterogeneous clients. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Kopiczko, D. J., Blankevoort, T., and Asano, Y. M. Vera: Vector-based random matrix adaptation. *arXiv preprint arXiv:2310.11454*, 2023.

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

Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I. V., and Lempitsky, V. S. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Li, D. and Wang, J. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

Li, T., Tan, L., Huang, Z., Tao, Q., Liu, Y., and Huang, X. Low dimensional trajectory hypothesis is true: Dnns can be trained in tiny subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3411–3420, 2022.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Lialin, V., Muckatira, S., Shivagunde, N., and Rumshisky, A. Relora: High-rank training through low-rank updates. In *Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ NeurIPS 2023)*, 2023.

Liu, J., Wu, J., Chen, J., Hu, M., Zhou, Y., and Wu, D. Feddwa: Personalized federated learning with dynamic weight adjustment. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pp. 3993–4001. ijcai.org, 2023.

Liu, R., Wu, F., Wu, C., Wang, Y., Lyu, L., Chen, H., and Xie, X. No one left behind: Inclusive federated learning over heterogeneous devices. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3398–3406, 2022.

Marchenko, V. A. and Pastur, L. A. Distribution of eigenvalues for some sets of random matrices. *Matematicheskii Sbornik*, 114(4):507–536, 1967.

Marfoq, O., Neglia, G., Vidal, R., and Kameni, L. Personalized federated learning through local memorization. In *International Conference on Machine Learning*, pp. 15070–15092. PMLR, 2022.

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

Mei, Y., Guo, P., Zhou, M., and Patel, V. Resource-adaptive federated learning with all-in-one neural composition. *Advances in Neural Information Processing Systems*, 35: 4270–4284, 2022.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M. I., et al. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pp. 561–577, 2018.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. The role of over-parametrization in generalization of neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Oh, J., Kim, S., and Yun, S. Fedbabu: Toward enhanced representation for federated image classification. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

Pfeiffer, K., Rapp, M., Khalili, R., and Henkel, J. Federated learning for computationally-constrained heterogeneous devices: A survey. *ACM Computing Surveys*, 2023.

Pillutla, K., Malik, K., Mohamed, A.-R., Rabbat, M., Sanjabi, M., and Xiao, L. Federated learning with partial model personalization. In *International Conference on Machine Learning*, pp. 17716–17758. PMLR, 2022.

Qiao, Z., Yu, X., Zhang, J., and Letaief, K. B. Communication-efficient federated learning with dual-side low-rank compression. *arXiv preprint arXiv:2104.12416*, 2021.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Sainath, T. N., Kingsbury, B., Sindhwani, V., Arisoy, E., and Ramabhadran, B. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6655–6659. IEEE, 2013.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Tai, C., Xiao, T., Wang, X., and E, W. Convolutional neural networks with low-rank regularization. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

Tan, A. Z., Yu, H., Cui, L., and Yang, Q. Towards personalized federated learning. *IEEE Trans. Neural Networks Learn. Syst.*, 34(12):9587–9603, 2023.

Tucker, L. R. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

Vershynin, R. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Wang, H., Sreenivasan, K., Rajput, S., Vishwakarma, H., Agarwal, S., Sohn, J.-y., Lee, K., and Papailiopoulos, D. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33:16070–16084, 2020.

Wang, H., Agarwal, S., and Papailiopoulos, D. Pufferfish: Communication-efficient models at no extra cost. *Proceedings of Machine Learning and Systems*, 3:365–386, 2021.

Wang, H.-P., Stich, S., He, Y., and Fritz, M. Progfed: effective, communication, and computation efficient federated learning by progressive training. In *International Conference on Machine Learning*, pp. 23034–23054. PMLR, 2022.

Xu, J., Tong, X., and Huang, S. Personalized federated learning with feature alignment and classifier collaboration. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Xue, J., Liu, M., Sun, S., Wang, Y., Jiang, H., and Jiang, X. Fedbiad: Communication-efficient and accuracy-guaranteed federated learning with bayesian inference-based adaptive dropout. In *2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 489–500. IEEE, 2023.

Yao, D., Pan, W., O'Neill, M. J., Dai, Y., Wan, Y., Jin, H., and Sun, L. Fedhm: Efficient federated learning for heterogeneous models via low-rank factorization. *arXiv preprint arXiv:2111.14655*, 2021.

Ye, M., Fang, X., Du, B., Yuen, P. C., and Tao, D. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.

Yu, H., Yang, S., and Zhu, S. 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, pp. 5693–5700, 2019.

Yu, J. and Huang, T. S. Universally slimmable networks and improved training techniques. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1803–1811, 2019.

Zhang, K., Dai, Y., Wang, H., Xing, E., Chen, X., and Sun, L. Memory-adaptive depth-wise heterogenous federated learning. *arXiv preprint arXiv:2303.04887*, 2023.

Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., and Ma, K. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3713–3722, 2019.

Zhang, L., Wu, D., and Yuan, X. Fedzkt: Zero-shot knowledge transfer towards resource-constrained federated learning with heterogeneous on-device models. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, pp. 928–938. IEEE, 2022.

Zhao, J., Zhang, Z., Chen, B., Wang, Z., Anandkumar, A., and Tian, Y. Galore: Memory-efficient LLM training by gradient low-rank projection. *CoRR*, abs/2403.03507, 2024. doi: 10.48550/ARXIV.2403.03507. URL https://doi.org/10.48550/arXiv.2403.03507.

Zou, D., Long, P. M., and Gu, Q. On the global convergence of training deep linear resnets. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

A. Discussion about FedHM (Yao et al., 2021)

Algorithm 2 FedHM and FedHMv2

Input: Local epoch E, total iteration T, learning rate γ, a set of randomly selected clients $\mathcal{N}^0$, the initial full model $\mathbf{w}^0 = \{W^1, W^2, \cdots, W^L\}$ with L layers, hyper parameters ρ, rank shrinkage ratios $\{\alpha_i\}_{i=1}^N$ for N clients, $\alpha_1 = \alpha_2 = \cdots = \alpha_N$, SVD executing period T_s, the initial low rank models $\{\mathbf{x}_i^0\}_{i=1}^N$ for N clients where $\mathbf{x}_i^0 = \{W^1, \cdots, W^\rho, U^{\rho+1}, V^{\rho+1}, \cdots, U^L, V^L\}$ is obtained through SVD according to the corresponding α_i and ρ.
Output: Final full model $\mathbf{w}^t$.
for $t = 1$ **to** T **do**
  for client $i \in \mathcal{N}^{t-1}$ in parallel **do**
    $\mathbf{x}_i^t = \mathbf{x}_i^{t-1} - \gamma \nabla G_i(\mathbf{x}_i^{t-1}, \xi_i^t)$
  end for
  if t divides E **then**
    Each client $i \in \mathcal{N}^{t-1}$ sends $\mathbf{x}_i^t$ to the server
    Server recovers the low rank models $\{\mathbf{x}_i^t\}_{i\in\mathcal{N}^{t-1}}$ from clients to the corresponding full models $\{\mathbf{w}_i^t\}_{i\in\mathcal{N}^{t-1}}$
    Server updates the global full model $\mathbf{w}^t = \frac{1}{|\mathcal{N}^{t-1}|}\sum_{i=1}^{|\mathcal{N}^{t-1}|} \mathbf{w}_i^t$
    Server randomly samples a new client set $\mathcal{N}^t$
    Server factorizes $\mathbf{w}^t$ using SVD according to α_i to get new $\{\mathbf{x}_i^t\}_{i\in\mathcal{N}^t}$ and broadcasts them to all chosen clients
    if t divides T_s **then**
      Server factorizes $\mathbf{w}^t$ using SVD according to α_i to get new $\mathbf{x}^t$ and broadcasts $\mathbf{x}^t$ to all chosen clients
    else
      Server updates the global low rank model $\mathbf{x}^t = \frac{1}{|\mathcal{N}^{t-1}|}\sum_{i=1}^{|\mathcal{N}^{t-1}|} \mathbf{x}_i^t$
      Server broadcasts $\mathbf{x}^t$ to all chosen clients in $\mathcal{N}^t$
    end if
  end if
end for

A.1. A Toy Example

Despite the flexible design of FedHM, in this section, we empirically show that flexibly assigning heterogeneous sub-models to clients in FedHM will introduce significant approximation errors over multiple SVD operations. Conversely, the performance of our approach is better by training a homogeneous pre-factorized sub-model to avoid frequently conducting SVD operations.

To adapt to the typical heterogeneous FL setting, we simulate a non-IID (non-identically and independently distributed) scenario using the CIFAR10 dataset where 100 clients collaboratively train a ResNet-18 model over $2,000$ communication rounds. In this setting, clients can not train the original full model $\mathbf{w}$ due to limited resources, and can only train the sub-models factorized from the full model. Following from (Yao et al., 2021), we use *hybrid model architecture* technique proposed in (Wang et al., 2021) to decompose the original full model $\mathbf{w}$ from the 10th layer to the 17th layer and adjust the compression ratio to construct low-rank sub-model $\mathbf{x}$, which is only 20.7% the size of the original full model. To be specific, for a full model $\mathbf{w} = \{W^1, W^2, \cdots, W^L\}$ with L layers, the corresponding low-rank model $\mathbf{x}$ can be represented as $\mathbf{x} = \{W^1, \cdots, W^\rho, U^{\rho+1}, V^{\rho+1}, \cdots, U^L, V^L\}$ by executing $W^l = U^l(V^l)^T$ using SVD for $l = \rho+1, \cdots, L$, where $(\cdot)^T$ means the transpose operation of matrix and $\rho+1$ represents the index of the first decomposed layer. In each communication round, only 10% of the clients are sampled for model training, and other hyper parameters for training can be found in Table 6 in Appendix E.3.

The workflows of FedHM (Algorithm 2 in the original paper[1]) and the modified version (FedHMv2) we make are present in Algorithm 2: they share most procedures except the highlighted steps. Here the full model $\mathbf{w} = \{W^1, W^2, \cdots, W^L\}$ with L layers is recovered from the low-rank model $\mathbf{x}$ by executing $W^l = U^l(V^l)^T$ for $l = \rho+1, \cdots, L$. To visualize the effect of SVD operation on model performance, we make the following two changes in FedHMv2. 1) We set $\alpha_1 = \alpha_2 = \cdots = \alpha_N$ for all N clients participating in the training process, which means that all clients train the homogeneous low-rank sub-

[1] https://arxiv.org/pdf/2111.14655.pdf

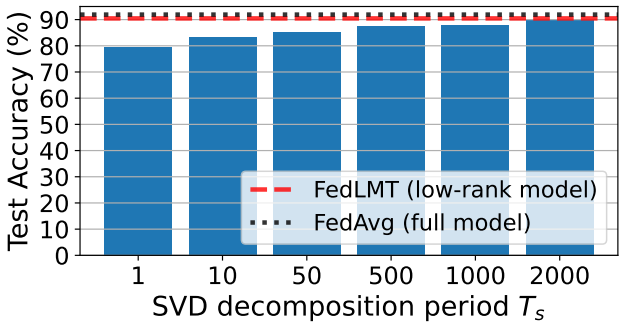


Figure 3. The impact of SVD operation on model accuracy using FedHMv2. Here T_s is the interval to conduct SVD. For example, $T_s = 10$ means that SVD is conducted per 10 rounds. The results of the bar chart represent the accuracy of FedHMv2 under different Ts. The dotted line represents the accuracy of training the original full model using FedAvg and the dashed line represents the accuracy of training the pre-factorized low-rank model using FedAvg (FedLMT).

models. 2) We introduce an additional variable T_s to control the execution frequency of SVD. For example, $T_s = 1$ means that we do SVD every communication round, just as the original FedHM does. $T_s = 10$ means that FedHMv2 executes the SVD operation every 10 rounds. The larger T_s is, the fewer times the SVD operation is executed.

In FedHMv2, given that these sub-models are homogeneous, we can tune the frequency to conduct SVD for every T_s round where T_s varies from 1 to $2,000$. We show the final model accuracy of FedHMv2 under different values of T_s, and as a comparison, we also show the results of training homogeneous low-rank sub-models without SVD operations (FedLMT) and training the original full model using FedAvg (McMahan et al., 2017). Figure 3 shows the final results. Notably, the final model accuracy of FedHMv2 is higher as T_s is larger. In particular, when $T_s = 2,000$, implying that no SVD is executed, the accuracy of the low-rank model trained by FedHMv2 is very close to that of the full model trained by FedAvg. If SVD is conducted more frequently with a smaller T_s, the model accuracy is lower, *e.g.*, when $T_s = 1$, the original FedHM algorithm is executed, and the model performance is the worst. This toy example shows the potential that training a homogeneous low-rank model without SVD operation can improve model performance by minimizing approximation errors, which motivates us to narrow our analysis on the case with homogeneous sub-models.

A.2. Discussion about Theoretical Results in FedHM

The upper bound in Theorem 1 proposed in (Yao et al., 2021) is a function of the learning rate γ and can be abbreviated as:

$$\mathcal{O}(\frac{1}{TE\eta} + C_1\eta + C_2\eta^2 + C_3(1 + \frac{1}{E\eta})), \tag{4}$$

where T is the number of total iteration, E is the number of local epoch and C_1, C_2, C_3 are constants. The authors point out that the establishment condition of Theorem 1 in their paper is $0 < \eta \leq \frac{1}{L_g}$ where L_g is a Lipschitz constant. As the number of iteration T increases, the first term $\frac{1}{TE\eta}$ can approach 0. For the last three terms, when $\eta \to 0$, we have $C_1\eta \to 0$ and $C_2\eta^2 \to 0$, whereas the last term $\frac{C_3}{E\eta}$ goes to infinity. The only possibility that $\frac{C_3}{E\eta}$ approaches 0 is that $E \to \infty$, which means that each client trains an infinite number of local iterations, and this is impractical. Besides, there is also an independent term C_3 not affected by η, which will not be 0.

B. Marchenko-Pastur Theory

The Marchenko-Pastur (MP) theory defines the distribution of singular values of Gaussian random matrices in the infinite limit but is applicable to finite matrices with very reasonable error bounds. Specifically, for a $N \times M (N > M)$ gaussian random matrix X where each element $x_{ij} \sim \mathcal{N}(0, \sigma^2)$, MP theory defines the distribution of the eigenvalue of matrix $\frac{XX^T}{N}$ as:

$$\rho(\lambda) = \begin{cases} \frac{N}{2\pi\sigma^2 M} \frac{\sqrt{(\lambda^+ - \lambda)(\lambda - \lambda^-)}}{\lambda} & \text{if } \lambda \in [\lambda^-, \lambda^+] \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

where $\lambda^+ = \sigma^2\left(1+\sqrt{\frac{M}{N}}\right)^2$ and $\lambda^- = \sigma^2\left(1-\sqrt{\frac{M}{N}}\right)^2$ represent the largest and smallest eigenvalue, respectively. Since the eigenvalues of XX^T are the squares of the singular values of X, the smallest singular value of X can be greater than 0.

C. pFedLMT Algorithm

Algorithm 3 gives the details of the pFedLMT algorithm.

Algorithm 3 pFedLMT

Input: Local epoch E, total iteration T, learning rate γ, a set of randomly selected clients $\mathcal{N}^0$, the initial low-rank model $\mathbf{x}_i^0 = (\mathbf{p}^0, \mathbf{q}_i^0)$ according to $\beta_i, \forall i$. $\mathbf{p}$ are the *common* layers and $\mathbf{q}_i$ are the *custom* layers of client i.
Output: Personalized models $\{\mathbf{x}_1^t, \cdots, \mathbf{x}_N^t\}$.
for $t = 1$ **to** T **do**
  for client $i \in \mathcal{N}^{t-1}$ in parallel **do**
    $\mathbf{q}_i^t = \mathbf{q}_i^{t-1} - \gamma \nabla_{\mathbf{q}_i^{t-1}} G_i(\mathbf{x}_i^{t-1}, \xi_i^t)$
    $\mathbf{p}^t = \mathbf{p}^{t-1} - \gamma \nabla_{\mathbf{p}^{t-1}} G_i(\mathbf{x}_i^{t-1}, \xi_i^t)$
  end for
  if t divides E **then**
    Each client i in $\mathcal{N}^{t-1}$ sends $\mathbf{p}_i^t$ to the server
    Server updates $\mathbf{p}^t = \frac{1}{|\mathcal{N}^{t-1}|}\sum_{i=1}^{|\mathcal{N}^{t-1}|} \mathbf{p}_i^t$
    Server randomly samples a new client set $\mathcal{N}^t$
    Server broadcasts $\mathbf{p}^t$ to all chosen clients and replaces the *common* layers of clients' local models
  end if
end for

D. Convergence Analysis

In this section, we first review the background of the problem and some of the assumptions and mathematical properties that need to be used for the proof, and then we give a concrete proof process.

D.1. Preliminary

Table 5. Notations.

Symbol	Description
N	the number of all clients
i	the index of client
l	the index of the layer in the specific neural network
$\mathbf{x}$	the parameters of low rank model
$\mathbf{w}$	the parameters of the recovered full model
$\mathcal{D}_i$	the dataset of client i
f	the loss function for full model $\mathbf{w}$
g	the loss function for low rank model $\mathbf{x}$
∇f	the gradient of function f for full model $\mathbf{w}$
∇g	the gradient of function g for low rank model $\mathbf{x}$
t	the index of all iterations
γ	the learning rate
T	the total number of iterations
E	the number of local iterations to be executed
L	the number of layers in the specific neural network
ρ	the number of layers which are not decomposed

Algorithm 4 FedLMT: Federated Learning with Low-rank Model Training

Input: initial low-rank model $\bar{\mathbf{x}}^0 = \mathbf{x}_i^0 = \{W^1, \cdots, W^\rho, U^{\rho+1}, V^{\rho+1}, \cdots, U^L, V^L\}$ for all clients, local epochs E, the number of total iterations T such that $T \bmod E = 0$, learning rate γ.
Output: Final low-rank model $\bar{\mathbf{x}}^t$.
for $t = 1$ **to** T **do**
  for $i = 1$ **to** N **do**
    Compute stochastic gradient $G_i(\mathbf{x}_i^{t-1}; \xi_i^t)$.
    $\mathbf{x}_i^t = \begin{cases} \frac{1}{N}\sum_{j=1}^N (\mathbf{x}_j^{t-1} - \gamma G_j(\mathbf{x}_j^{t-1}; \xi_j^t)), & \text{if } t \bmod E = 0 \\ \mathbf{x}_i^{t-1} - \gamma G_i(\mathbf{x}_i^{t-1}; \xi_i^t), & \text{otherwise.} \end{cases}$
  end for
end for
$\bar{\mathbf{x}}^t = \frac{1}{N}\sum_{j=1}^N \mathbf{x}_i^t$.

Traditional federated learning is to learn a global full model $\mathbf{w} = \{W^1, W^2, \cdots, W^L\}$ by solving the following problem:

$$\min_{\mathbf{w} \in \mathcal{W}_f} f(\mathbf{w}) \triangleq \frac{1}{N}\sum_{i=1}^N f_i(\mathbf{w}), \tag{6}$$

where $\mathcal{W}_f$ denotes the full model weight space with d dimensions, N is the number of client and each $f_i(\mathbf{w}) \triangleq \mathbb{E}_{\xi_i \in \mathcal{D}_i}[F_i(\mathbf{w}; \xi_i)]$. Let $\mathbf{x} = \{W^1, \cdots, W^\rho, U^{\rho+1}, V^{\rho+1}, \cdots, U^L, V^L\}$ is the corresponding low-rank model of $\mathbf{w}$, if we want to train a low rank model $\mathbf{x}$, we actually solve the following problem:

$$\min_{\mathbf{x} \in \mathcal{W}_g} g(\mathbf{x}) \triangleq \frac{1}{N}\sum_{i=1}^N g_i(\mathbf{x}), \tag{7}$$

where $\mathcal{W}_g$ denotes the low rank model weight space with d' dimensions and each $g_i(\mathbf{x}) \triangleq \mathbb{E}_{\xi_i \in \mathcal{D}_i}[G_i(\mathbf{x}; \xi_i)]$. Then, we use the hybrid low rank model $\mathbf{x}$ to recover the corresponding full model $\mathbf{w}$: $\mathbf{x} \in \mathcal{W}_g \mapsto \mathbf{w} \in \mathcal{W}_f$ by executing $W^l = U^l (V^l)^T$ for $l = \rho + 1, \cdots, L$ where $(\cdot)^T$ means the transpose operation of matrix. It is important to observe that for any identical sample $\xi = (z, y)$ where z is the input and y is the label under supervised learning scenario, the full model $\mathbf{w}$ and its corresponding low-rank model $\mathbf{x}$ will get the same result, i.e. $F_i(\mathbf{w}; \xi) = G_i(\mathbf{x}; \xi)$. Therefore, if we can obtain a good low-rank model $\mathbf{x}$, we can recover it and then gain the full model $\mathbf{w}$ with the same model performance. We use FedAvg (McMahan et al., 2017) to train the low-rank model $\mathbf{x}$ under a FL scenario and rewrite Algorithm 1 in the form of Algorithm 4 for ease of proof. Note that Algorithm 1 and Algorithm 4 are mathematically equivalent. For a fix iteration index t, we define $\mathbf{x}_i^t = \left\{W_{i,t}^1, \cdots, W_{i,t}^\rho, U_{i,t}^{\rho+1}, V_{i,t}^{\rho+1}, \cdots, U_{i,t}^L, V_{i,t}^L\right\}$ to represent the low rank model of client i at iteration t and the corresponding full model is $\mathbf{w}_i^t = \left\{W_{i,t}^1, \cdots, W_{i,t}^\rho, \cdots, W_{i,t}^L\right\}$. Since $\mathbf{w}_i^t$ is recovered form $\mathbf{x}_i^t$, the full rank model also can be represented as $\mathbf{w}_i^t = \left\{W_{i,t}^1, \cdots, W_{i,t}^\rho, U_{i,t}^{\rho+1}(V_{i,t}^{\rho+1})^T, \cdots, U_{i,t}^L (V_{i,t}^L)^T\right\}$. Here $U_{i,t}^l$ and $V_{i,t}^l$ mean the lth layer low rank model parameters of client i at iteration t, $U_{i,t}^l \in \mathcal{R}^{m_l \times r_l}$ is a $m_l \times r_l$ matrix, $V_{i,t}^l \in \mathcal{R}^{n_l \times r_l}$ is a $n_l \times r_l$ matrix, and r_l means the low rank scale and usually $r_l \ll \min\{m_l, n_l\}$. ρ means we decompose the full rank model only from layer $\rho + 1$ to layer L. $W_{i,t}^l \in \mathcal{R}^{m_l \times n_l}$ is a $m_l \times n_l$ matrix and $W_{i,t}^l = U_{i,t}^l (V_{i,t}^l)^T$ for all $l \geq \rho + 1$, In this work, we only consider the case where the decomposition ratio of each layer is identical, i.e. $r^l = r, \forall l = \rho + 1, \cdots, L$. Next, we define

$$\bar{\mathbf{x}}^t \triangleq \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i^t \tag{8}$$

as the average of the local low-rank model overall N clients at the tth iteration. It is immediate that

$$\bar{\mathbf{x}}^t = \bar{\mathbf{x}}^{t-1} - \gamma \frac{1}{N}\sum_{i=1}^N \nabla G_i(\mathbf{x}_i^{t-1}). \tag{9}$$

Correspondingly, for each layer l, we can define $\bar{W}_t^l \triangleq \frac{1}{N} W_{i,t}^l$, $\bar{U}_t^l \triangleq \frac{1}{N} U_{i,t}^l$, $\bar{V}_t^l \triangleq \frac{1}{N} V_{i,t}^l$. Hence, $\bar{\mathbf{x}}^t =$

$\left\{\bar{W}_t^1, \cdots, \bar{W}_t^\rho, \bar{U}_t^{\rho+1}, \bar{V}_t^{\rho+1}, \cdots, \bar{U}_t^L, \bar{V}_t^L\right\}$, and the corresponding full model $\bar{\mathbf{w}}^t$ of $\bar{\mathbf{x}}^t$ can be represented as

$$\bar{\mathbf{w}}^t = \left\{\bar{W}_t^1, \cdots, \bar{W}_t^\rho, \bar{U}_t^{\rho+1}(\bar{V}_t^{\rho+1})^T, \cdots, \bar{U}_t^L(\bar{V}_t^L)^T\right\}. \tag{10}$$

Since we train the low rank model $\mathbf{x}$ using FedAvg (McMahan et al., 2017), assuming the total number of iterations is T, we can get a sequence $X' = \{\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \cdots, \bar{\mathbf{x}}_t, \cdots, \bar{\mathbf{x}}_T\}$ at the end of the training in Algorithm 4. Since $\bar{\mathbf{w}}^t$ can be recovered from $\bar{\mathbf{x}}^t$, we also have another imaginary sequence $W' = \{\bar{\mathbf{w}}_1, \bar{\mathbf{w}}_2, \cdots, \bar{\mathbf{w}}_t, \cdots, \bar{\mathbf{w}}_T\}$. We emphasize that in the whole training process, only the sequence X' exists, while the sequence W' is only the theoretically existing sequence, since we do not need to explicitly recover $\bar{\mathbf{w}}_t$ from $\bar{\mathbf{x}}_t$ at each training iteration. According to the existing convergence analysis (Yu et al., 2019; Li et al., 2020; Khaled et al., 2020; Glasgow et al., 2022) of FedAvg, X' is convergent and since $\bar{\mathbf{w}}_t$ is recovered from $\bar{\mathbf{x}}_t$, W' is also convergent at the end of Algorithm 4. However, it only means $\mathbf{x}^t$ is converged in the low rank model weight space $\mathcal{W}_g$, *i.e.,* $\nabla g(\bar{\mathbf{x}}^t; \xi) = 0$, and we can't conclude $\nabla f(\bar{\mathbf{w}}^T; \xi) = 0$ in the full model weight space $\mathcal{W}_f$. Theorem 4.5 proves that if we can obtain the optimal low-rank model $\mathbf{x}$ in $\mathcal{W}_g$ after training using Algorithm 4, then we can conclude that the corresponding full model $\mathbf{w}$ will also converge in the full weight space $\mathcal{W}_f$. This theorem implies that it is reasonable to obtain the full model by training the corresponding low-rank model with much less training cost.

Next, we describe some important properties of learning with factorized layers. For each iteration t, client i and layer $l > \rho$, according to the chain rule for derivatives, the stochastic gradient of layer l of the hybrid model $\mathbf{x}_i^t$ satisfies

$$\nabla G_i^l\left(\mathbf{x}_i^t; \xi_i^{t+1}\right) = \begin{bmatrix} \nabla F_i^l\left(\mathbf{w}_i^t; \xi_i^{t+1}\right) V_{i,t}^l \\ \nabla F_i^l\left(\mathbf{w}_i^t; \xi_i^{t+1}\right)^T U_{i,t}^l \end{bmatrix} \text{ and } \nabla g_i^l\left(\mathbf{x}_i^t; \xi_i^{t+1}\right) = \begin{bmatrix} \nabla f_i^l\left(\mathbf{w}_i^t; \xi_i^{t+1}\right) V_{i,t}^l \\ \nabla f_i^l\left(\mathbf{w}_i^t; \xi_i^{t+1}\right)^T U_{i,t}^l \end{bmatrix}. \tag{11}$$

Note that in Eq. (11), both $\nabla G_i^l\left(\mathbf{x}_i^t; \xi_i^{t+1}\right)$ and $\nabla g_i^l\left(\mathbf{x}_i^t; \xi_i^{t+1}\right)$ are $(m_l + n_l) \times r$ matrices. For ease of writing, without causing ambiguity, we use the abbreviation $\nabla g_i^l\left(\mathbf{x}_i^t\right)$ to denote $\nabla g_i^l\left(\mathbf{x}_i^t; \xi_i^{t+1}\right)$ and $\nabla G_i^l\left(\mathbf{x}_i^t\right)$ to denote $\nabla G_i^l\left(\mathbf{x}_i^t; \xi_i^{t+1}\right)$. Furthermore, since $\nabla g(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^N \nabla g_i(\mathbf{x})$, we have

$$\nabla G^l\left(\bar{\mathbf{x}}^t\right) = \begin{bmatrix} \nabla F^l\left(\bar{\mathbf{w}}^t\right) \bar{V}_t^l \\ \nabla F^l\left(\bar{\mathbf{w}}^t\right)^T \bar{U}_t^l \end{bmatrix} \text{ and } \nabla g^l\left(\bar{\mathbf{x}}^t\right) = \begin{bmatrix} \nabla f^l\left(\bar{\mathbf{w}}^t\right) \bar{V}_t^l \\ \nabla f^l\left(\bar{\mathbf{w}}^t\right)^T \bar{U}_t^l \end{bmatrix}. \tag{12}$$

The following equations also hold.

$$\begin{aligned} \left\|\nabla G_i^l\left(\mathbf{x}_i^{t-1}\right)\right\|_F^2 &= \left\|\nabla F_i^l\left(\mathbf{w}_i^{t-1}\right) V_{i,t}^l\right\|_F^2 + \left\|\nabla F_i^l\left(\mathbf{w}_i^{t-1}\right)^T U_{i,t}^l\right\|_F^2. \\ \left\|\nabla g_i^l\left(\mathbf{x}_i^{t-1}\right)\right\|_F^2 &= \left\|\nabla f_i^l\left(\mathbf{w}_i^{t-1}\right) V_{i,t}^l\right\|_F^2 + \left\|\nabla f_i^l\left(\mathbf{w}_i^{t-1}\right)^T U_{i,t}^l\right\|_F^2. \end{aligned} \tag{13}$$

Besides, the whole gradient of $\mathbf{x}_i^{t-1}$ and $\mathbf{w}_i^{t-1}$ can be represented as

$$\begin{aligned} \nabla g_i\left(\mathbf{x}_i^{t-1}\right) &= \{\nabla g_i^1\left(\mathbf{x}_i^{t-1}\right), \cdots, \nabla g_i^L\left(\mathbf{x}_i^{t-1}\right)\}, \\ \nabla G_i\left(\mathbf{x}_i^{t-1}\right) &= \{\nabla G_i^1\left(\mathbf{x}_i^{t-1}\right), \cdots, \nabla G_i^L\left(\mathbf{x}_i^{t-1}\right)\}, \\ \nabla f_i\left(\mathbf{w}_i^{t-1}\right) &= \{\nabla f_i^1\left(\mathbf{w}_i^{t-1}\right), \cdots, \nabla f_i^L\left(\mathbf{x}_i^{t-1}\right)\}, \\ \nabla F_i\left(\mathbf{w}_i^{t-1}\right) &= \{\nabla F_i^1\left(\mathbf{w}_i^{t-1}\right), \cdots, \nabla F_i^L\left(\mathbf{w}_i^{t-1}\right)\}, \end{aligned} \tag{14}$$

where $\nabla g_i\left(\mathbf{x}_i^{t-1}\right)$, $\nabla G_i\left(\mathbf{x}_i^{t-1}\right)$, $\nabla f_i\left(\mathbf{w}_i^{t-1}\right)$ and $\nabla F_i\left(\mathbf{w}_i^{t-1}\right)$ are vectors, and

$$\begin{aligned} \left\|\nabla g_i\left(\mathbf{x}_i^{t-1}\right)\right\|_2^2 &= \sum_{l=1}^L \left\|\nabla g_i^l\left(\mathbf{x}_i^{t-1}\right)\right\|_F^2, \quad \left\|\nabla G_i\left(\mathbf{x}_i^{t-1}\right)\right\|_2^2 = \sum_{l=1}^L \left\|\nabla G_i^l\left(\mathbf{x}_i^{t-1}\right)\right\|_F^2 \\ \left\|\nabla f_i\left(\mathbf{w}_i^{t-1}\right)\right\|_2^2 &= \sum_{l=1}^L \left\|\nabla f_i^l\left(\mathbf{w}_i^{t-1}\right)\right\|_F^2, \quad \left\|\nabla F_i\left(\mathbf{w}_i^{t-1}\right)\right\|_2^2 = \sum_{l=1}^L \left\|\nabla F_i^l\left(\mathbf{w}_i^{t-1}\right)\right\|_F^2. \end{aligned} \tag{15}$$

In the following, we will try to show that the full model sequence $W' = \{\bar{\mathbf{w}}_1, \bar{\mathbf{w}}_2, \cdots, \bar{\mathbf{w}}_t, \cdots, \bar{\mathbf{w}}_T\}$ obtained by Algorithm 4 will also converge to a local stationary point in the full model weight space $\mathcal{W}_f$ under non-convex and smooth assumptions.

D.2. Assumptions

Assumption D.1. For every client i, both the loss function $f_i(\cdot)$ and $g_i(\cdot)$ are continuously differentiable and have L_s-Lipschitz continuous gradient.

Assumption D.2. For every client i, the stochastic gradient $\nabla F_i(\mathbf{w}_i^t;\xi_i^t)$ and $\nabla G_i(\mathbf{x}_i^t;\xi_i^t)$ are both unbiased estimate, i.e., $\mathbb{E}\left[\nabla F_i(\mathbf{w}_i^t;\xi_i^t)\right] = \nabla f_i(\mathbf{w}_i^t)$, $\mathbb{E}\left[\nabla G_i(\mathbf{x}_i^t;\xi_i^t)\right] = \nabla g_i(\mathbf{x}_i^t)$ for all $t \in \mathbb{N}$. Besides, there exist constants $\sigma > 0$ $\sigma_g > 0$, $G > 0$ and $G_g > 0$ such that:

$$\begin{aligned}
\mathbb{E}_{\xi_i\sim\mathcal{D}_i}\left\|\nabla F_i\left(\mathbf{w};\xi_i\right) - \nabla f_i(\mathbf{w})\right\|^2 \le \sigma^2, \forall \mathbf{w}, \forall i,\\
\mathbb{E}_{\xi_i\sim\mathcal{D}_i}\left\|\nabla F_i\left(\mathbf{w};\xi_i\right)\right\|^2 \le G^2, \forall \mathbf{w}, \forall i,\\
\mathbb{E}_{\xi_i\sim\mathcal{D}_i}\left\|\nabla F_i\left(\mathbf{w};\xi_i\right)\right\|^4 \le G^4, \forall \mathbf{w}, \forall i,\\
\mathbb{E}_{\xi_i\sim\mathcal{D}_i}\left\|\nabla G_i\left(\mathbf{x};\xi_i\right) - \nabla g_i(\mathbf{x})\right\|^2 \le \sigma_g^2, \forall \mathbf{x}, \forall i,\\
\mathbb{E}_{\xi_i\sim\mathcal{D}_i}\left\|\nabla G_i\left(\mathbf{x};\xi_i\right)\right\|^2 \le G_g^2, \forall \mathbf{x}, \forall i.
\end{aligned}$$

Assumption D.3. There exists constants κ_u, κ_v and κ_{uv} such that at each iteration t and for every client i:

$$\left\|U_{i,t}^l\right\|_F \le \kappa_u, \left\|V_{i,t}^l\right\|_F \le \kappa_v, \left\|U_{i,t}^l(V_{i,t}^l)^T\right\|_F \le \kappa_{uv}, \quad \forall l \in \{\rho+1, \cdots, L\},$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $(\cdot)^T$ means the transpose operation.

Assumption D.4. There exist a constant $\psi_{uv} > 0$ such that at each iteration t and for every client i:

$$\sigma_{min}(\bar{U}_t^l) \ge \psi_{uv}, \sigma_{min}(\bar{V}_t^l) \ge \psi_{uv}, \quad \forall l \in \{\rho+1, \cdots, L\},$$

where $\sigma_{min}(\cdot)$ denotes the smallest singular value.

D.3. Supporting Lemmas and Corresponding Proofs

We first prove Lemma D.5, Lemma D.6, Lemma D.7, Lemma D.8 and Lemma D.9, and then we give the proof of Theorem D.10 (which is Theorem 4.5 in the main paper) and Corollary D.12 (which is Corollary 4.6 in the main paper).

Lemma D.5. *Using Algorithm 1, at iteration t, given a full model $\mathbf{w}^t = \{W_t^1, W_t^2, \cdots, W_t^L\}$ and the corresponding low-rank model $\mathbf{x}^t = \{W_t^1, \cdots, W_t^\rho, U_t^{\rho+1}, V_t^{\rho+1}, \cdots, U_t^L, V_t^L\}$, for $l = \rho+1, \cdots, L$, we have*

$$\left\|\nabla g^l(\mathbf{x}^t)\right\|_F^2 \ge \left(\sigma_{min}^2(V_t^l) + \sigma_{min}^2(U_t^l)\right)\left\|\nabla f^l(\mathbf{w}^t)\right\|_F^2, \tag{16}$$

where $\sigma_{min}(\cdot)$ represents the smallest singular value of any matrix and $\|\cdot\|_F$ denotes the Fronbenius norm.

Proof. Since $\mathbf{w}$ is recovered from $\mathbf{x}$ by executing $W^l = U^l(V^l)^T$, $\forall l > \rho$, using the derivative chain rule, we have

$$\begin{aligned}
\left\|\nabla g^l(\mathbf{x}^t)\right\|_F^2 &= \left\|\nabla f^l(\mathbf{w}^t)V_t^l\right\|_F^2 + \left\|(\nabla f^l(\mathbf{w}^t))^T U_t^l\right\|_F^2\\
&= \left\|((V_t^l)^T\nabla f^l(\mathbf{w}^t)^T)\right\|_F^2 + \left\|(U_t^l)^T\nabla f^l(\mathbf{w}^t)\right\|_F^2\\
&\overset{(a)}{\ge} \left(\sigma_{min}^2(V_t^l) + \sigma_{min}^2(U_t^l)\right)\left\|\nabla f^l(\mathbf{w}^t)\right\|_F^2,
\end{aligned} \tag{17}$$

where (a) follows from the basic inequality $\sigma_{min}(U)\|V\|_F \le \|UV\|_F$ for any matrix $U \in \mathcal{R}^{m\times r}$ and matrix $V \in \mathcal{R}^{r\times n}$, and the proof of this inequality can be found in Lemma B.3 proposed in (Zou et al., 2020).

□

Lemma D.6. *Under Assumption D.1 and Assumption D.2, at each iteration t and for every client i it follows that*

$$\mathbb{E}\left[\left\|\bar{\mathbf{x}}^t - \mathbf{x}_i^t\right\|_2^2\right] \le 4\gamma^2E^2G_g^2, \tag{18}$$

where $\bar{\mathbf{x}}^t$ is defined in Eq. (8) and G is the constant defined in Assumption D.2.

Proof. For the fixed $t \geq 1$ and $i \in \{1, 2, \cdots, N\}$, noting that the aggregated model $\bar{\mathbf{x}}^t \triangleq \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i^t$ is updated every E iterations, we can analyze it separately based on the value of t.

1. t mod $E \neq 0$. At this time, there exists a largest $t_0 \leq t$ such that $\bar{\mathbf{x}}^{t_0} = \mathbf{x}_i^{t_0}$. Besides, t_0 also meets the condition $t - t_0 \leq E$ such that

$$\mathbf{x}_i^t = \mathbf{x}_i^{t_0} - \gamma \sum_{\tau=t_0+1}^{t} \nabla G_i(\mathbf{x}_i^{\tau-1}) = \bar{\mathbf{x}}^{t_0} - \gamma \sum_{\tau=t_0+1}^{t} \nabla G_i(\mathbf{x}_i^{\tau-1}). \tag{19}$$

By Eq. (8), we have

$$\bar{\mathbf{x}}^t = \bar{\mathbf{x}}^{t_0} - \gamma \frac{1}{N} \sum_{i=1}^N \sum_{\tau=t_0+1}^{t} \nabla G_i(\mathbf{x}_i^{\tau-1}). \tag{20}$$

Thus, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\bar{\mathbf{x}}^t - \mathbf{x}_i^t\right\|_2^2\right] = \mathbb{E}\left[\left\|\gamma \sum_{\tau=t_0+1}^{t} \frac{1}{N}\sum_{i=1}^N \nabla G_i(\mathbf{x}_i^{\tau-1}) - \gamma \sum_{\tau=t_0+1}^{t} \nabla G_i(\mathbf{x}_i^{\tau-1})\right\|_2^2\right] \\
&= \gamma^2 \mathbb{E}\left[\left\|\sum_{\tau=t_0+1}^{t} \frac{1}{N}\sum_{i=1}^N \nabla G_i(\mathbf{x}_i^{\tau-1}) - \sum_{\tau=t_0+1}^{t} \nabla G_i(\mathbf{x}_i^{\tau-1})\right\|_2^2\right] \\
&\overset{(a)}{\leq} 2\gamma^2 \mathbb{E}\left[\left\|\sum_{\tau=t_0+1}^{t} \frac{1}{N}\sum_{i=1}^N \nabla G_i(\mathbf{x}_i^{\tau-1})\right\|_2^2 + \left\|\sum_{\tau=t_0+1}^{t} \nabla G_i(\mathbf{x}_i^{\tau-1})\right\|_2^2\right] \\
&\overset{(b)}{\leq} 2\gamma^2 (t-t_0) \mathbb{E}\left[\sum_{\tau=t_0+1}^{t} \left\|\frac{1}{N}\sum_{i=1}^N \nabla G_i(\mathbf{x}_i^{\tau-1})\right\|_2^2 + \sum_{\tau=t_0+1}^{t} \left\|\nabla G_i(\mathbf{x}_i^{\tau-1})\right\|_2^2\right] \\
&\overset{(c)}{\leq} 2\gamma^2 (t-t_0) \mathbb{E}\left[\sum_{\tau=t_0+1}^{t} \frac{1}{N}\sum_{i=1}^N \left\|\nabla G_i(\mathbf{x}_i^{\tau-1})\right\|_2^2 + \sum_{\tau=t_0+1}^{t} \left\|\nabla G_i(\mathbf{x}_i^{\tau-1})\right\|_2^2\right] \\
&\overset{(d)}{\leq} 4\gamma^2 E^2 G_g^2.
\end{aligned} \tag{21}$$

where (a)-(c) follows by using the inequality $\|\sum_{i=1}^n \mathbf{z}_i\|_2^2 \leq n\sum_{i=1}^n \|\mathbf{z}_i\|_2^2$ for any vector $\mathbf{z}_i$ and any positive integer n (using $n = 2$ for (a), $n = t - t_0$ for (b) and $n = N$ for (c)); and (d) follows from Assumption D.2.

2. t mod E = 0. At this time we have

$$\mathbf{x}_i^t = \frac{1}{N}\sum_{j=1}(\mathbf{x}_j^{t-1} - \gamma \nabla G_j(\mathbf{x}_j^{t-1})) = \bar{\mathbf{x}}^{t-1} - \gamma \frac{1}{N}\sum_{j=1}^N \nabla G_j(\mathbf{x}_j^{t-1}). \tag{22}$$

By the definition of $\bar{\mathbf{x}}^t$, we have

$$\begin{aligned}
\bar{\mathbf{x}}^t &= \frac{1}{N}\sum_{i=1}^N \mathbf{x}_i^t = \frac{1}{N}\sum_{i=1}^N (\frac{1}{N}\sum_{j=1}(\mathbf{x}_j^{t-1} - \gamma \nabla G_j(\mathbf{x}_j^{t-1}))) \\
&= \bar{\mathbf{x}}^{t-1} - \gamma \frac{1}{N}\sum_{j=1}^N \nabla G_j(\mathbf{x}_j^{t-1}).
\end{aligned} \tag{23}$$

Hence, $\mathbb{E}\left[\|\bar{\mathbf{x}}^t - \mathbf{x}_i^t\|_2^2\right] = 0 \leq 4\gamma^2 E^2 G_g^2$.

In conclusion, Lemma D.6 is valid. □

Lemma D.7. *Under Assumption D.1, Assumption D.2 and Assumption D.3, at each iteration t and for every client i it follows that*

$$\mathbb{E}\left[\left\|\bar{\mathbf{w}}^t - \mathbf{w}_i^t\right\|_2^2\right] \leq \Gamma, \tag{24}$$

where $\Gamma = 4\rho\gamma^2E^2G^2 + 6\gamma^2(\kappa_u^4 + \kappa_v^4)E^2G^2(L-\rho)(2+\gamma^2E^2G^2)$.

Proof. At this time the partial layers of model $\bar{\mathbf{w}}^t$ are recovered after low-rank training from $\bar{\mathbf{x}}^t$, we need to distinguish between the layer obtained by low-rank restoration and the normal layer in the proof. Note that

$$\mathbb{E}\left[\left\|\bar{\mathbf{w}}^t - \mathbf{w}_i^t\right\|_2^2\right] = \mathbb{E}\left[\sum_{l=1}^{\rho}\left\|\bar{W}_t^l - W_{i,t}^l\right\|_F^2 + \sum_{l=\rho+1}^{L}\left\|\bar{U}_t^l(\bar{V}_t^l)^T - U_{i,t}^l(V_{i,t}^l)^T\right\|_F^2\right] \tag{25}$$

We first consider the item $\mathbb{E}\left[\sum_{l=1}^{\rho}\left\|\bar{W}_t^l - W_{i,t}^l\right\|_F^2\right]$. Similar to Lemma D.6, we consider two cases when t mod $E \neq 0$ and when t mode $E = 0$.

1. t mode $E \neq 0$. At this time, there still exists a largest $t_0 \leq t$ such that $W_{i,t}^l = \bar{W}_{t_0}^l - \gamma\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})$ and $\bar{W}_t^l = \bar{W}_{t_0}^l - \gamma\frac{1}{N}\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})$, so

$$\begin{aligned}
&\mathbb{E}\left[\sum_{l=1}^{\rho}\left\|\bar{W}_t^l - W_{i,t}^l\right\|_F^2\right] = \sum_{l=1}^{\rho}\mathbb{E}\left[\left\|\bar{W}_t^l - W_{i,t}^l\right\|_F^2\right] \\
&= \sum_{l=1}^{\rho}\gamma^2\mathbb{E}\left[\left\|\sum_{\tau=t_0+1}^{t}\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{\tau-1}) - \sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2\right] \\
&\overset{(a)}{\leq} \sum_{l=1}^{\rho}2\gamma^2\mathbb{E}\left[\left\|\sum_{\tau=t_0+1}^{t}\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2 + \left\|\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2\right] \\
&\overset{(b)}{\leq} \sum_{l=1}^{\rho}2\gamma^2(t-t_0)\mathbb{E}\left[\sum_{\tau=t_0+1}^{t}\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2 + \sum_{\tau=t_0+1}^{t}\left\|\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2\right] \\
&\overset{(c)}{\leq} \sum_{l=1}^{\rho}2\gamma^2(t-t_0)\mathbb{E}\left[\sum_{\tau=t_0+1}^{t}\frac{1}{N}\sum_{i=1}^{N}\left\|\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2 + \sum_{\tau=t_0+1}^{t}\left\|\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2\right] \\
&\overset{(d)}{\leq} 4\rho\gamma^2E^2G^2,
\end{aligned} \tag{26}$$

   where (a)-(c) follows by using the inequality $\left\|\sum_{i=1}^{n}\mathbf{Z}_i\right\|_F^2 \leq n\sum_{i=1}^{n}\left\|\mathbf{Z}_i\right\|_F^2$ for any matrix $\mathbf{Z}_i$ and any positive integer n (using $n=2$ for (a), $n = t - t_0$ for (b) and $n = N$ for (c)); and (d) follows from Assumption D.2.

2. t mod $E = 0$. Similar to the proof of Lemma D.6, at this time we have $\mathbb{E}\left[\sum_{l=1}^{\rho}\left\|\bar{W}_t^l - W_{i,t}^l\right\|_F^2\right] = 0 \leq 4\rho\gamma^2E^2G^2$.

Overall, we have

$$\mathbb{E}\left[\sum_{l=1}^{\rho}\left\|\bar{W}_t^l - W_{i,t}^l\right\|_F^2\right] \leq 4\rho\gamma^2E^2G^2.$$

Next we consider the item $\mathbb{E}\left[\sum_{l=\rho+1}^{L}\left\|\bar{U}_t^l(\bar{V}_t^l)^T - U_{i,t}^l(V_{i,t}^l)^T\right\|_F^2\right]$, similarly, we need to analyze case t mod $E \neq 0$ and case t mod $E = 0$.

1. t mod $E \neq 0$. At this time there exists a largest integer $t_0 \leq t$ such that $U_{i,t}^l = \bar{U}_{t_0}^l - \gamma\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l$, $\bar{U}_t^l = \bar{U}_{t_0}^l - \gamma\frac{1}{N}\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l$, $V_{i,t}^l = \bar{V}_{t_0}^l - \gamma\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})^TU_{i,\tau-1}^l$, and $\bar{V}_t^l = \bar{V}_{t_0}^l - \gamma\frac{1}{N}\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})^TU_{i,\tau-1}^l$.

We further note that

$$\begin{aligned}
&\bar{U}_t^l(\bar{V}_t^l)^T - U_{i,t}^l(V_{i,t}^l)^T = \\
&-\gamma(\sum_{\tau=t_0+1}^{t}\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l - \sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l)(\bar{V}_{t_0}^l)^T \\
&-\gamma\bar{U}_{t_0}^l(\sum_{\tau=t_0+1}^{t}\frac{1}{N}\sum_{i=1}^{N}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1}) - \sum_{\tau=t_0+1}^{t}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1})) \\
&+\gamma^2[(\frac{1}{N}\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l)(\frac{1}{N}\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1}))] \\
&-\gamma^2[(\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l)(\sum_{\tau=t_0+1}^{t}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1}))].
\end{aligned} \tag{27}$$

Let

$$A^l = -\gamma(\sum_{\tau=t_0+1}^{t}\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l - \sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l)(\bar{V}_{t_0}^l)^T, \tag{28}$$

$$B^l = -\gamma\bar{U}_{t_0}^l(\sum_{\tau=t_0+1}^{t}\frac{1}{N}\sum_{i=1}^{N}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1}) - \sum_{\tau=t_0+1}^{t}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1})), \tag{29}$$

$$\begin{aligned}
C^l = &\gamma^2[(\frac{1}{N}\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l)(\frac{1}{N}\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1}))] \\
&-\gamma^2[(\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l)(\sum_{\tau=t_0+1}^{t}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1}))].
\end{aligned} \tag{30}$$

Note that

$$\begin{aligned}
&\mathbb{E}[\|A^l\|_F^2] \\
&= \gamma^2\mathbb{E}\left[\left\|(\sum_{\tau=t_0+1}^{t}\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l - \sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l)(\bar{V}_{t_0}^l)^T\right\|_F^2\right] \\
&\overset{(a)}{\leq} 2\gamma^2\mathbb{E}\left[\left\|(\sum_{\tau=t_0+1}^{t}\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l)(\bar{V}_{t_0}^l)^T\right\|_F^2\right] \\
&\quad + 2\gamma^2\mathbb{E}\left[\left\|(\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l)(\bar{V}_{t_0}^l)^T\right\|_F^2\right] \\
&\overset{(b)}{\leq} 2\gamma^2\kappa_v^2\mathbb{E}\left[\left\|\sum_{\tau=t_0+1}^{t}\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l\right\|_F^2 + \left\|\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l\right\|_F^2\right] \\
&\overset{(c)}{\leq} 2\kappa_v^2\gamma^2(t-t_0)\mathbb{E}\left[\sum_{\tau=t_0+1}^{t}\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l\right\|_F^2\right] \\
&\quad + 2\kappa_v^2\gamma^2(t-t_0)\mathbb{E}\left[\sum_{\tau=t_0+1}^{t}\|\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l\|_F^2\right] \\
&\overset{(d)}{\leq} 2\kappa_v^2\gamma^2(t-t_0)[(t-t_0)G^2\kappa_v^2 + (t-t_0)G^2\kappa_v^2] \\
&\leq 4\gamma^2\kappa_v^4E^2G^2,
\end{aligned} \tag{31}$$

where (a) and (c) follows by using the inequality $\|\sum_{i=1}^{n}\mathbf{Z}_i\|_F^2 \le n\sum_{i=1}^{n}\|\mathbf{Z}_i\|_F^2$ for any matrix $\mathbf{Z}_i$ and any positive integer n (using $n=2$ for (a), $n=t-t_0$ for (c)); (b) follows from the basic inequality $\|PQ\|_F^2 \le \|P\|_F^2\|Q\|_F^2$ for any matrix P and Q and Assumption D.3; (d) follows from Assumption D.2 and Assumption D.3. Similarly, noting that there is a symmetrical form between A and B in mathematical form, it is easy to find that

$$
\begin{aligned}
&\mathbb{E}[\|B^l\|_F^2] \\
&= \gamma^2\mathbb{E}\left[\left\|\bar{U}_{t_0}^l\Big(\sum_{\tau=t_0+1}^{t}\frac{1}{N}\sum_{i=1}^{N}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1}) - \sum_{\tau=t_0+1}^{t}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1})\Big)\right\|_F^2\right] \\
&\le 2\kappa_u^2\gamma^2(t-t_0)\mathbb{E}\left[\sum_{\tau=t_0+1}^{t}\left\|(U_{i,\tau-1}^l)^T\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2\right] \\
&\quad + 2\kappa_u^2\gamma^2(t-t_0)\mathbb{E}\left[\sum_{\tau=t_0+1}^{t}\left\|(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2\right] \\
&\le 4\gamma^2\kappa_u^4E^2G^2.
\end{aligned}
\tag{32}
$$

What's more, we can find that

$$
\begin{aligned}
&\mathbb{E}[\|C^l\|_F^2] \\
&\le 2\gamma^4\mathbb{E}\left[\left\|(\frac{1}{N}\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l)(\frac{1}{N}\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1}))\right\|_F^2\right] \\
&\quad + 2\gamma^4\mathbb{E}\left[\left\|(\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l)(\sum_{\tau=t_0+1}^{t}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1}))\right\|_F^2\right] \\
&\overset{(a)}{\le} 2\gamma^4\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l\right\|_F^2\cdot\left\|\frac{1}{N}\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2\right] \\
&\quad + 2\gamma^4\mathbb{E}\left[\left\|\sum_{\tau=t_0+1}^{t}\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l\right\|_F^2\cdot\left\|\sum_{\tau=t_0+1}^{t}(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2\right] \\
&\le \frac{2\gamma^4(t-t_0)^2}{N^2}\underbrace{\mathbb{E}\left[\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}\left\|\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l\right\|_F^2\cdot\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}\left\|(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2\right]}_{D_1} \\
&\quad + 2\gamma^4(t-t_0)^2\underbrace{\mathbb{E}\left[\sum_{\tau=t_0+1}^{t}\left\|\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l\right\|_F^2\cdot\sum_{\tau=t_0+1}^{t}\left\|(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2\right]}_{D_2}
\end{aligned}
\tag{33}
$$

where (a) follows from the basic inequality $\|PQ\|_F^2 \le \|P\|_F^2\|Q\|_F^2$ for any matrix P and Q. Next, we consider D_1 and D_2 respectively. To facilitate writing, we denote $P_{i,\tau-1}^l = \left\|\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l\right\|_F^2$ and $Q_{i,\tau-1}^l =$

$\left\|(U_{i,\tau-1}^l)^T \nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^2$. Noting that $P_{i,\tau-1}^l \geq 0$ and $Q_{i,\tau-1}^l \geq 0$, we have

$$\begin{aligned}
D_1 &= \mathbb{E}\left[\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t} P_{i,\tau-1}^l \cdot \sum_{j=1}^{N}\sum_{\tau'=t_0+1}^{t} Q_{j,\tau'-1}^l\right] \\
&\overset{(a)}{=} \frac{1}{2}\mathbb{E}\left[\left|\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t} P_{i,\tau-1}^l\right|^2 + \left|\sum_{j=1}^{N}\sum_{\tau'=t_0+1}^{t} Q_{j,\tau'-1}^l\right|^2\right] \\
&\quad - \frac{1}{2}\mathbb{E}\left[\left|\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t} P_{i,\tau-1}^l - \sum_{j=1}^{N}\sum_{\tau'=t_0+1}^{t} Q_{j,\tau'-1}^l\right|^2\right] \\
&\overset{(b)}{\leq} \frac{1}{2}\mathbb{E}\left[\left|\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t} P_{i,\tau-1}^l\right|^2 + \left|\sum_{j=1}^{N}\sum_{\tau'=t_0+1}^{t} Q_{j,\tau'-1}^l\right|^2\right] \\
&\overset{(c)}{\leq} \frac{1}{2}N(t-t_0)\mathbb{E}\left[\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}\left|P_{i,\tau-1}^l\right|^2 + \sum_{j=1}^{N}\sum_{\tau'=t_0+1}^{t}\left|Q_{j,\tau'-1}^l\right|^2\right] \\
&= \frac{1}{2}N(t-t_0)\mathbb{E}\left[\sum_{i=1}^{N}\sum_{\tau=t_0+1}^{t}\left(\left|P_{i,\tau-1}^l\right|^2 + \left|Q_{i,\tau-1}^l\right|^2\right)\right],
\end{aligned} \tag{34}$$

where (a) follows from the basic identity $ab = \frac{1}{2}(a^2 + b^2 - (a-b)^2)$ for any two real numbers, (b) follows from $a^2 + b^2 - (a-b)^2 \leq a^2 + b^2$ and (c) follows from $\left|\sum_{i=1}^n z_i\right|^2 \leq n\sum_{i=1}^n |z_i|^2$ for any real number z_i and any positive integer n. Since

$$\begin{aligned}
\left|P_{i,\tau-1}^l\right|^2 + \left|Q_{j,\tau-1}^l\right|^2 &= \left\|\nabla F_i^l(\mathbf{w}_i^{\tau-1})V_{i,\tau-1}^l\right\|_F^4 + \left\|(U_{i,\tau-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^4 \\
&\leq (\kappa_u^4 + \kappa_v^4)\left\|\nabla F_i^l(\mathbf{w}_i^{\tau-1})\right\|_F^4 \\
&\leq (\kappa_u^4 + \kappa_v^4)G^4,
\end{aligned} \tag{35}$$

where the first inequality comes from $\|PQ\|_F^4 \leq \|P\|_F^4 \|Q\|_F^4$ and the last inequality comes from Assumption D.2. Therefore, we finally have

$$D_1 \leq \frac{1}{2}(\kappa_u^4 + \kappa_v^4)(t-t_0)^2 G^4 N^2. \tag{36}$$

In the same way, we have

$$\begin{aligned}
D_2 = \mathbb{E}\left[\sum_{\tau=t_0+1}^{t} P_{i,\tau-1}^l \cdot \sum_{\tau'=t_0+1}^{t} Q_{j,\tau'-1}^l\right] &\leq \frac{1}{2}\mathbb{E}\left[\left|\sum_{\tau=t_0+1}^{t} P_{i,\tau-1}^l\right|^2 + \left|\sum_{\tau'=t_0+1}^{t} Q_{j,\tau'-1}^l\right|^2\right] \\
\leq \frac{1}{2}(t-t_0)\mathbb{E}\left[\sum_{\tau=t_0+1}^{t}\left(\left|P_{i,\tau-1}^l\right|^2 + \left|Q_{j,\tau-1}^l\right|^2\right)\right] &\leq \frac{1}{2}(\kappa_u^4 + \kappa_v^4)(t-t_0)^2 G^4.
\end{aligned} \tag{37}$$

Substituting Eq. (36) and Eq. (37) into Eq. (33) yields

$$\mathbb{E}[\left\|C^l\right\|_F^2] \leq \gamma^4(\kappa_u^4 + \kappa_v^4)(t-t_0)^4 G^4 + \gamma^4(\kappa_u^4 + \kappa_v^4)(t-t_0)^4 G^4 \leq 2\gamma^4(\kappa_u^4 + \kappa_v^4)E^4 G^4. \tag{38}$$

Combining the results of Eq. (31), Eq. (32), and Eq. (38), we get

$$\begin{aligned}
&\mathbb{E}\left[\sum_{l=\rho+1}^{L}\left\|\bar{U}_t^l(\bar{V}_t^l)^T - U_{i,t}^l(V_{i,t}^l)^T\right\|_F^2\right] = \mathbb{E}\left[\sum_{l=\rho+1}^{L}\left\|A^l + B^l + C^l\right\|_F^2\right] \\
&\leq \sum_{l=\rho+1}^{L} 3\mathbb{E}\left[\left\|A^l\right\|_F^2 + \left\|B^l\right\|_F^2 + \left\|C^l\right\|_F^2\right] \\
&\leq 6\gamma^2(\kappa_u^4 + \kappa_v^4)E^2G^2(L-\rho)(2+\gamma^2E^2G^2).
\end{aligned} \tag{39}$$

2. $t \bmod E = 0$. At this time it is easy to find that $\bar{U}_t^l = U_{i,t}^l$ and $\bar{V}_t^l = V_{i,t}^l$, hence

$$\mathbb{E}\left[\sum_{l=\rho+1}^{L}\left\|\bar{U}_t^l(\bar{V}_t^l)^T - U_{i,t}^l(V_{i,t}^l)^T\right\|_F^2\right] = 0 \leq 6\gamma^2(\kappa_u^4 + \kappa_v^4)E^2G^2(L-\rho)(2+\gamma^2E^2G^2). \tag{40}$$

In short, we have

$$\mathbb{E}\left[\left\|\bar{\mathbf{w}}^t - \mathbf{w}_i^t\right\|_2^2\right] \leq 4\rho\gamma^2E^2G^2 + 6\gamma^2(\kappa_u^4 + \kappa_v^4)E^2G^2(L-\rho)(2+\gamma^2E^2G^2). \tag{41}$$

□

Lemma D.8. *Under Assumption D.1, Assumption D.2 and Assumption D.3, at each iteration t and for every client i, it follows that*

$$\begin{aligned}
\mathbb{E}\left[\left\|\bar{\mathbf{w}}^t - \bar{\mathbf{w}}^{t-1}\right\|_2^2\right] \leq &\gamma^2\sum_{l=1}^{\rho}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] + \frac{\gamma^2\sigma^2}{N} \\
&+ 3\gamma^2(L-\rho)G^2[\kappa_u^4 + \kappa_v^4 + \frac{2\gamma^2G^2}{N^2}(\kappa_{uv}^2 + (N-1)^2\kappa_u^2\kappa_v^2)].
\end{aligned} \tag{42}$$

Proof. According to the definition of $\bar{\mathbf{w}}^t$ in Eq. (10), we have

$$\mathbb{E}\left[\left\|\bar{\mathbf{w}}^t - \bar{\mathbf{w}}^{t-1}\right\|_2^2\right] = \mathbb{E}\left[\sum_{l=1}^{\rho}\left\|\bar{W}_t^l - \bar{W}_{t-1}^l\right\|_F^2 + \sum_{l=\rho+1}^{L}\left\|\bar{U}_t^l(\bar{V}_t^l)^T - \bar{U}_{t-1}^l(\bar{V}_{t-1}^l)^T\right\|_F^2\right] \tag{43}$$

It's easy to find that

$$\begin{aligned}
&\mathbb{E}\left[\sum_{l=1}^{\rho}\left\|\bar{W}_t^l - \bar{W}_{t-1}^l\right\|_F^2\right] = \sum_{l=1}^{\rho}\gamma^2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] \\
&\overset{(a)}{=} \sum_{l=1}^{\rho}\left(\gamma^2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}(\nabla F_i^l(\mathbf{w}_i^{t-1}) - \nabla f_i^l(\mathbf{w}_i^{t-1}))\right\|_F^2\right] + \gamma^2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1}))\right\|_F^2\right]\right) \\
&\overset{(b)}{=} \sum_{l=1}^{\rho}\left(\gamma^2\frac{1}{N^2}\sum_{i=1}^{N}\mathbb{E}\left[\left\|\nabla F_i^l(\mathbf{w}_i^{t-1}) - \nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] + \gamma^2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1}))\right\|_F^2\right]\right) \\
&\overset{(c)}{\leq} \gamma^2\frac{1}{N^2}\sum_{i=1}^{N}\mathbb{E}\left[\left\|\nabla F_i(\mathbf{w}_i^{t-1}) - \nabla f_i(\mathbf{w}_i^{t-1})\right\|_2^2\right] + \sum_{l=1}^{\rho}\gamma^2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1}))\right\|_F^2\right] \\
&\overset{(d)}{\leq} \frac{\gamma^2\sigma^2}{N} + \gamma^2\sum_{l=1}^{\rho}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1}))\right\|_F^2\right],
\end{aligned} \tag{44}$$

where (a) follows by noting that $\mathbb{E}[\nabla F_i^l(\mathbf{w}_i^{t-1})] = \nabla f_i^l(\mathbf{w}_i^{t-1})$ and applying the basic equality $\mathbb{E}[\|Z\|_F^2] = \mathbb{E}[\|Z - \mathbb{E}[Z]\|_F^2] + \|\mathbb{E}[Z]\|_F^2$ for any matrix Z; (b) follows because each $\nabla F_i^l(\mathbf{w}_i^{t-1}) - \nabla f_i^l(\mathbf{w}_i^{t-1})$ has zero mean and is independent across clients; (c) follows because $\sum_{l=1}^{\rho} \left\|\nabla F_i^l(\mathbf{w}_i^{t-1}) - \nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2 \leq \left\|\nabla F_i(\mathbf{w}_i^{t-1}) - \nabla f_i(\mathbf{w}_i^{t-1})\right\|_2^2$;(d) follows from Assumption D.2. While for the item $\mathbb{E}\left[\sum_{l=\rho+1}^{L} \left\|\bar{U}_t^l(\bar{V}_t^l)^T - \bar{U}_{t-1}^l(\bar{V}_{t-1}^l)^T\right\|_F^2\right]$, we first note that

$$\begin{aligned}\bar{U}_t^l(\bar{V}_t^l)^T - \bar{U}_{t-1}^l(\bar{V}_{t-1}^l)^T = &\underbrace{-\gamma\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(\bar{V}_{t-1}^l)^T}_{A} \quad \underbrace{-\gamma\frac{1}{N}\sum_{i=1}^{N}\bar{U}_{t-1}^l(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1})}_{B} \\ &\underbrace{+\gamma^2(\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l)(\frac{1}{N}\sum_{i=1}^{N}(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1}))}_{C}.\end{aligned} \tag{45}$$

It is easy to find that

$$\mathbb{E}\left[\|A\|_F^2\right] \leq \gamma^2\kappa_v^2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l\right\|_F^2\right] \leq \gamma^2\kappa_v^2\frac{1}{N}\mathbb{E}\left[\sum_{i=1}^{N}\left\|\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l\right\|_F^2\right] \leq \gamma^2\kappa_v^4G^2. \tag{46}$$

$$\mathbb{E}\left[\|B\|_F^2\right] \leq \gamma^2\kappa_u^2\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] \leq \gamma^2\kappa_u^2\frac{1}{N}\mathbb{E}\left[\sum_{i=1}^{N}\left\|(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] \leq \gamma^2\kappa_u^4G^2. \tag{47}$$

$$\begin{aligned}
\mathbb{E}\left[\|C\|_F^2\right] &= \gamma^4\mathbb{E}\left[\left\|(\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l)(\frac{1}{N}\sum_{i=1}^{N}(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1}))\right\|_F^2\right] \\
&= \frac{\gamma^4}{N^4}\mathbb{E}\left[\left\|\sum_{i=1}^{N}\left(\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1})\right) + \sum_{i=1}^{N}\sum_{j\neq i}^{N}\left(\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(U_{j,t-1}^l)^T\nabla F_j^l(\mathbf{w}_j^{t-1})\right)\right\|_F^2\right] \\
&\leq \frac{2\gamma^4}{N^4}\mathbb{E}\left[\left\|\sum_{i=1}^{N}\left(\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1})\right)\right\|_F^2\right] \\
&\quad + \frac{2\gamma^4}{N^4}\mathbb{E}\left[\left\|\sum_{i=1}^{N}\sum_{j\neq i}^{N}\left(\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(U_{j,t-1}^l)^T\nabla F_j^l(\mathbf{w}_j^{t-1})\right)\right\|_F^2\right] \\
&\leq \frac{2\gamma^4\kappa_{uv}^2G^4}{N^2} + \frac{2\gamma^4}{N^4}\mathbb{E}\left[\left\|\sum_{i=1}^{N}\sum_{j\neq i}^{N}\left(\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(U_{j,t-1}^l)^T\nabla F_j^l(\mathbf{w}_j^{t-1})\right)\right\|_F^2\right] \\
&\leq \frac{2\gamma^4\kappa_{uv}^2G^4}{N^2} + \frac{2\gamma^4}{N^4}N(N-1)\sum_{i=1}^{N}\sum_{j\neq i}^{N}\mathbb{E}\left[\left\|\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(U_{j,t-1}^l)^T\nabla F_j^l(\mathbf{w}_j^{t-1})\right\|_F^2\right] \\
&\leq \frac{2\gamma^4\kappa_{uv}^2G^4}{N^2} + \frac{2\gamma^4(N-1)}{N^3}\kappa_u^2\kappa_v^2\sum_{i=1}^{N}\sum_{j\neq i}^{N}\mathbb{E}\left[\left\|\nabla F_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\left\|\nabla F_j^l(\mathbf{w}_j^{t-1})\right\|_F^2\right] \\
&\leq \frac{2\gamma^4G^4}{N^2}(\kappa_{uv}^2 + (N-1)^2\kappa_u^2\kappa_v^2),
\end{aligned} \tag{48}$$

where the last inequality holds because $\nabla F_i^l(\mathbf{w}_i^{t-1})$ and $\nabla F_j^l(\mathbf{w}_j^{t-1})$ are independent for $i \neq j$. Therefore, we can deduce that

$$\begin{aligned}
&\mathbb{E}\left[\sum_{l=\rho+1}^{L}\left\|\bar{U}_t^l(\bar{V}_t^l)^T-\bar{U}_{t-1}^l(\bar{V}_{t-1}^l)^T\right\|_F^2\right]=\sum_{l=\rho+1}^{L}\mathbb{E}\left[\|A+B+C\|_F^2\right]\\
&\leq\sum_{l=\rho+1}^{L}3\mathbb{E}\left[\|A\|_F^2+\|B\|_F^2+\|C\|_F^2\right]\\
&\leq 3\gamma^2(L-\rho)G^2[\kappa_u^4+\kappa_v^4+\frac{2\gamma^2G^2}{N^2}(\kappa_{uv}^2+(N-1)^2\kappa_u^2\kappa_v^2)].
\end{aligned}\tag{49}$$

Combining the result of Eq. (44) and Eq. (49), we finally can get

$$\begin{aligned}
\mathbb{E}\left[\left\|\bar{\mathbf{w}}^t-\bar{\mathbf{w}}^{t-1}\right\|_2^2\right]\leq&\gamma^2\sum_{l=1}^{\rho}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right]+\frac{\gamma^2\sigma^2}{N}\\
&+3\gamma^2(L-\rho)G^2[\kappa_u^4+\kappa_v^4+\frac{2\gamma^2G^2}{N^2}(\kappa_{uv}^2+(N-1)^2\kappa_u^2\kappa_v^2)].
\end{aligned}\tag{50}$$

□

Lemma D.9. *Under Assumption D.1, Assumption D.2 and Assumption D.3, at each iteration t it follows that*

$$\begin{aligned}
&\mathbb{E}[\left\langle\nabla f(\bar{\mathbf{w}}^{t-1}),\bar{\mathbf{w}}^t-\bar{\mathbf{w}}^{t-1}\right\rangle]\\
&\leq\sum_{l=1}^{\rho}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right]+\sum_{l=1}^{\rho}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right]\\
&+\sum_{l=\rho+1}^{L}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\bar{V}_{t-1}^l\right\|_F^2\right]+\sum_{l=\rho+1}^{L}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l\right\|_F^2\right]\\
&+\sum_{l=\rho+1}^{L}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|(\bar{U}_{t-1}^l)^T\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right]+\sum_{l=\rho+1}^{L}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}(U_{i,t-1}^l)^T\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right]\\
&+\sum_{l=\rho+1}^{L}(\frac{\gamma^2}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right]+\frac{\gamma}{2}L_s^2\Gamma+4\gamma^3L_s^2E^2G_g^2+(L-\rho)\frac{\gamma^2G^4}{N^2}(\kappa_{uv}^2+\kappa_u^2\kappa_v^2(N-1)^2).
\end{aligned}\tag{51}$$

Proof. Note that $\nabla f(\bar{\mathbf{w}}^{t-1})=\{\nabla f^1(\bar{\mathbf{w}}^{t-1}),\cdots,\nabla f^L(\bar{\mathbf{w}}^{t-1})\}$ and $\bar{\mathbf{w}}^t-\bar{\mathbf{w}}^{t-1}=\{\bar{W}_t^1-\bar{W}_{t-1}^1,\cdots,\bar{W}_t^\rho-\bar{W}_{t-1}^\rho,\bar{U}_t^{\rho+1}(\bar{V}_t^{\rho+1})^T-\bar{U}_{t-1}^{\rho+1}(\bar{V}_{t-1}^{\rho+1})^T,\cdots,\bar{U}_t^L(\bar{V}_t^L)^T-\bar{U}_{t-1}^L(\bar{V}_{t-1}^L)^T\}$. We use the notation $\mathrm{vec}(\cdot)$ to denote the vectorization of a matrix (the result defaults to a column vector), therefore,

$$\begin{aligned}
\mathbb{E}[\left\langle\nabla f(\bar{\mathbf{w}}^{t-1}),\bar{\mathbf{w}}^t-\bar{\mathbf{w}}^{t-1}\right\rangle]=&\underbrace{\mathbb{E}\left[\sum_{l=1}^{\rho}\mathrm{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\mathrm{vec}(\bar{W}_t^l-\bar{W}_{t-1}^l)\right]}_{\text{part 1}}\\
&+\underbrace{\mathbb{E}\left[\sum_{l=\rho+1}^{L}\mathrm{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\mathrm{vec}(\bar{U}_t^l(\bar{V}_t^l)^T-\bar{U}_{t-1}^l(\bar{V}_{t-1}^l)^T)\right]}_{\text{part 2}}
\end{aligned}\tag{52}$$

For part 1, we have

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{l=1}^{\rho}\mathrm{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\mathrm{vec}(\bar{W}_t^\rho-\bar{W}_{t-1}^\rho)\right]\\
&=\sum_{l=1}^{\rho}(-\gamma)\mathbb{E}\left[\mathrm{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\mathrm{vec}(\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1}))\right]\\
&\overset{(a)}{=}\sum_{l=1}^{\rho}(-\gamma)\mathbb{E}\left[\mathrm{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\mathrm{vec}(\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1}))\right]\\
&\overset{(b)}{=}\sum_{l=1}^{\rho}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\mathrm{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))\right\|_2^2+\left\|\mathrm{vec}(\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1}))\right\|_2^2\right]\\
&\quad+\sum_{l=1}^{\rho}\frac{\gamma}{2}\mathbb{E}\left[\left\|\mathrm{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))-\mathrm{vec}(\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1}))\right\|_2^2\right]\\
&\overset{(c)}{=}\sum_{l=1}^{\rho}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2+\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right]\\
&\quad+\sum_{l=1}^{\rho}\frac{\gamma}{2}\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})-\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right]\\
&\overset{(d)}{\leq}\sum_{l=1}^{\rho}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2+\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right]+\frac{\gamma}{2}L_s^2\Gamma.
\end{aligned}
\tag{53}
$$

where (a) follows because

$$
\begin{aligned}
&\mathbb{E}\left[\mathrm{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\mathrm{vec}(\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1}))\right]\\
&=\mathbb{E}\left[\mathbb{E}\left[\mathrm{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\mathrm{vec}(\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1}))|\xi^{[t-1]}\right]\right]\\
&=\mathbb{E}\left[\mathrm{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\mathrm{vec}(\nabla F_i^l(\mathbf{w}_i^{t-1}))|\xi^{[t-1]}\right]\right]\\
&=\mathbb{E}\left[\mathrm{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\mathrm{vec}(\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1}))\right].
\end{aligned}
\tag{54}
$$

The first equality in Eq. (54) follows by the iterated law of expectations, the second equality in Eq. (54) follows because $\bar{\mathbf{w}}^{t-1}$ is determined by $\xi^{[t-1]}=[\xi^1,\cdots,\xi^{t-1}]$ and the third equality in Eq. (54) follows by $\mathbb{E}[\mathrm{vec}(\nabla F_i^l(\mathbf{w}_i^{t-1}))|\xi^{[t-1]}]=\mathrm{vec}(\nabla f_i^l(\mathbf{w}_i^{t-1}))$. (b) follows from the basic identity $\mathbf{z}_1^T\mathbf{z}_2=\frac{1}{2}(\|\mathbf{z}_1\|_2^2+\|\mathbf{z}_2\|_2^2-\|\mathbf{z}_1-\mathbf{z}_2\|_2^2)$ for any two column vectors

$\mathbf{z}_1$ and $\mathbf{z}_2$ with the same length; and (c) follows because $\|\mathbf{Z}\|_F^2 = \|\text{vec}(\mathbf{Z})\|_2^2$ for any matrix $\mathbf{Z}$. (d) follows because

$$\begin{aligned}
&\sum_{l=1}^{\rho} \mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1}) - \frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] \\
&= \sum_{l=1}^{\rho} \mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\bar{\mathbf{w}}^{t-1}) - \frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] \\
&\leq \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\sum_{l=1}^{\rho}\left\|\nabla f_i^l(\bar{\mathbf{w}}^{t-1}) - \nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] \\
&\leq \frac{1}{N}L_s^2\sum_{i=1}^{N}\mathbb{E}\left[\left\|\bar{\mathbf{w}}^t - \mathbf{w}_i^t\right\|_2^2\right] \\
&\leq L_s^2\Gamma,
\end{aligned} \tag{55}$$

where the first inequality follows by using $\|\sum_{i=1}^n \mathbf{Z}_i\|_F^2 \leq n\sum_{i=1}^n \|\mathbf{Z}_i\|_F^2$ for any matrix $\mathbf{Z}_i$; the second inequality follows from the fact that $\sum_{l=1}^{\rho}\left\|\nabla f_i^l(\bar{\mathbf{w}}^{t-1}) - \nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2 \leq \left\|\nabla f_i(\bar{\mathbf{w}}^{t-1}) - \nabla f_i(\mathbf{w}_i^{t-1})\right\|_2^2$ and the smoothness of each f_i by Assumption D.1; and the last inequality follows from Lemma D.7. Now let us consider part 2. Since

$$\begin{aligned}
&\sum_{l=\rho+1}^{L}\mathbb{E}\left[\text{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\text{vec}(\bar{U}_t^l(\bar{V}_t^l)^T - \bar{U}_{t-1}^l(\bar{V}_{t-1}^l)^T)\right] \\
&= \sum_{l=\rho+1}^{L}\mathbb{E}\left[-\gamma\text{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\text{vec}(\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(\bar{V}_{t-1}^l)^T)\right] \\
&\quad + \sum_{l=\rho+1}^{L}\mathbb{E}\left[-\gamma\text{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\text{vec}(\frac{1}{N}\sum_{i=1}^{N}\bar{U}_{t-1}^l(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1}))\right] \\
&\quad + \sum_{l=\rho+1}^{L}\mathbb{E}\left[\gamma^2\text{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\text{vec}\left((\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l)(\frac{1}{N}\sum_{i=1}^{N}(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1}))\right)\right] \\
&\overset{(a)}{=} \underbrace{\sum_{l=\rho+1}^{L}\mathbb{E}\left[-\gamma\text{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1})\bar{V}_{t-1}^l)^T\cdot\text{vec}(\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l)\right]}_{A} \\
&\quad + \underbrace{\sum_{l=\rho+1}^{L}\mathbb{E}\left[-\gamma\text{vec}((\bar{U}_{t-1}^l)^T\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\text{vec}(\frac{1}{N}\sum_{i=1}^{N}(U_{i,t-1}^l)^T\nabla f_i^l(\mathbf{w}_i^{t-1}))\right]}_{B} \\
&\quad + \underbrace{\sum_{l=\rho+1}^{L}\mathbb{E}\left[\gamma^2\text{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\text{vec}\left((\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l)(\frac{1}{N}\sum_{i=1}^{N}(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1}))\right)\right]}_{C}.
\end{aligned} \tag{56}$$

Denoting I_m means the $m\times m$ identity matrix and $\otimes$ means Kronecker product, (a) follows by using the basic identity $\text{vec}(PQ) = (I_m\otimes P)\text{vec}(Q) = (Q^T\otimes I_k)\text{vec}(P)$, for any matrix $P\in\mathcal{R}^{k\times r}, Q\in\mathcal{R}^{r\times m}$ and the iterated law of expectations. Now we consider A, B and C respectively. Using the basic identity $\mathbf{z}_1^T\mathbf{z}_2 = \frac{1}{2}(\|\mathbf{z}_1\|_2^2 + \|\mathbf{z}_2\|_2^2 - \|\mathbf{z}_1-\mathbf{z}_2\|_2^2)$

for any two column vectors $\mathbf{z}_1$ and $\mathbf{z}_2$, we have

$$\begin{aligned}
A =& -\frac{\gamma}{2}\mathbb{E}\left[\sum_{l=\rho+1}^{L}\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\bar{V}_{t-1}^l\right\|_F^2+\sum_{l=\rho+1}^{L}\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l\right\|_F^2\right]\\
&+\frac{\gamma}{2}\mathbb{E}\left[\sum_{l=\rho+1}^{L}\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\bar{V}_{t-1}^l-\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l\right\|_F^2\right]\\
\leq& -\frac{\gamma}{2}\mathbb{E}\left[\sum_{l=\rho+1}^{L}\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\bar{V}_{t-1}^l\right\|_F^2+\sum_{l=\rho+1}^{L}\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l\right\|_F^2\right]+2\gamma^3L_s^2E^2G_g^2,
\end{aligned}\tag{57}$$

where the first inequality follows because

$$\begin{aligned}
&\frac{\gamma}{2}\mathbb{E}\left[\sum_{l=\rho+1}^{L}\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\bar{V}_{t-1}^l-\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l\right\|_F^2\right]\\
&\leq\frac{\gamma}{2}\mathbb{E}\left[\sum_{l=\rho+1}^{L}\left\|\nabla g^l(\bar{\mathbf{x}}^{t-1})-\frac{1}{N}\sum_{i=1}^{N}\nabla g_i^l(\mathbf{x}_i^{t-1})\right\|_F^2\right]\\
&\leq\frac{\gamma}{2}\mathbb{E}\left[\left\|\nabla g(\bar{\mathbf{x}}^{t-1})-\frac{1}{N}\sum_{i=1}^{N}\nabla g_i(\mathbf{x}_i^{t-1})\right\|_2^2\right]\\
&=\frac{\gamma}{2}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}(\nabla g_i(\bar{\mathbf{x}}^{t-1})-\nabla g_i(\mathbf{x}_i^{t-1}))\right\|_2^2\right]\\
&\leq\frac{\gamma}{2N}L_s^2\sum_{i=1}^{N}\mathbb{E}[\left\|\bar{\mathbf{x}}^{t-1}-\mathbf{x}_i^{t-1}\right\|_2^2]\\
&\leq 2\gamma^3L_s^2E^2G_g^2,
\end{aligned}\tag{58}$$

where the first inequality follows from Eq. (11), Eq. (12) and the fact that $\|Z_1\|_F^2\leq\left\|\begin{bmatrix}Z_1\\Z_2\end{bmatrix}\right\|_F^2$ for any matrix Z_1 and Z_2 with the same number of columns; the second inequality follows because $\nabla g^l(\bar{\mathbf{x}}^{t-1})-\frac{1}{N}\sum_{i=1}^{N}\nabla g_i^l(\mathbf{x}_i^{t-1})$ is only a partial element in vector $\nabla g(\bar{\mathbf{x}}^{t-1})-\frac{1}{N}\sum_{i=1}^{N}\nabla g_i(\mathbf{x}_i^{t-1})$; the third inequality follows by using Assumption D.1 and the last inequality holds from Lemma D.6. In the same way, we have

$$\begin{aligned}
B =& \sum_{l=\rho+1}^{L}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|(\bar{U}_{t-1}^l)^T\nabla f^l(\bar{\mathbf{w}}^{t-1}))\right\|_F^2+\left\|\frac{1}{N}\sum_{i=1}^{N}(U_{i,t-1}^l)^T\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right]\\
&+\sum_{l=\rho+1}^{L}\frac{\gamma}{2}\mathbb{E}\left[\left\|(\bar{U}_{t-1}^l)^T\nabla f^l(\bar{\mathbf{w}}^{t-1}))-\frac{1}{N}\sum_{i=1}^{N}(U_{i,t-1}^l)^T\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right]\\
\leq& \sum_{l=\rho+1}^{L}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|(\bar{U}_{t-1}^l)^T\nabla f^l(\bar{\mathbf{w}}^{t-1}))\right\|_F^2+\left\|\frac{1}{N}\sum_{i=1}^{N}(U_{i,t-1}^l)^T\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right]+2\gamma^3L_s^2E^2G_g^2.
\end{aligned}\tag{59}$$

Finally, using the inequality $\mathbf{z}_1^T\mathbf{z}_2 \leq \frac{1}{2}(\|\mathbf{z}_1\|_2^2 + \|\mathbf{z}_2\|_2^2)$ for any two column vectors $\mathbf{z}_1$ and $\mathbf{z}_2$, we have

$$
\begin{aligned}
C &\leq \sum_{l=\rho+1}^{L} \frac{\gamma^2}{2}\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] + \sum_{l=\rho+1}^{L} \frac{\gamma^2}{2}\mathbb{E}\left[\left\|(\frac{1}{N}\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l)(\frac{1}{N}\sum_{i=1}^{N}(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1}))\right\|_F^2\right] \\
&= \sum_{l=\rho+1}^{L} \frac{\gamma^2}{2}\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] \\
&\quad + \sum_{l=\rho+1}^{L} \frac{\gamma^2}{2N^4}\mathbb{E}\left[\left\|\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1}) + \sum_{i=1}^{N}\sum_{j\neq i}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(U_{j,t-1}^l)^T\nabla F_j^l(\mathbf{w}_j^{t-1})\right\|_F^2\right] \\
&\overset{(a)}{\leq} \sum_{l=\rho+1}^{L} \frac{\gamma^2}{2}\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] + \sum_{l=\rho+1}^{L} \frac{\gamma^2}{N^4}\mathbb{E}\left[\left\|\sum_{i=1}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] \\
&\quad + \sum_{l=\rho+1}^{L} \frac{\gamma^2}{N^4}\mathbb{E}\left[\left\|\sum_{i=1}^{N}\sum_{j\neq i}^{N}\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(U_{j,t-1}^l)^T\nabla F_j^l(\mathbf{w}_j^{t-1})\right\|_F^2\right] \\
&\overset{(b)}{\leq} \sum_{l=\rho+1}^{L} \frac{\gamma^2}{2}\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] + \sum_{l=\rho+1}^{L} \frac{\gamma^2}{N^3}\mathbb{E}\left[\sum_{i=1}^{N}\left\|\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(U_{i,t-1}^l)^T\nabla F_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] \\
&\quad + \sum_{l=\rho+1}^{L} \frac{\gamma^2(N-1)}{N^3}\mathbb{E}\left[\sum_{i=1}^{N}\sum_{j\neq i}^{N}\left\|\nabla F_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l(U_{j,t-1}^l)^T\nabla F_j^l(\mathbf{w}_j^{t-1})\right\|_F^2\right] \\
&\overset{(c)}{\leq} \sum_{l=\rho+1}^{L} \frac{\gamma^2}{2}\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] + \sum_{l=\rho+1}^{L} \frac{\gamma^2}{N^3}\sum_{i=1}^{N}\mathbb{E}\left[\left\|\nabla F_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\left\|V_{i,t-1}^l(U_{i,t-1}^l)^T\right\|_F^2\left\|\nabla F_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] \\
&\quad + \sum_{l=\rho+1}^{L} \frac{\gamma^2(N-1)}{N^3}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\mathbb{E}\left[\left\|\nabla F_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\left\|V_{i,t-1}^l\right\|_F^2\left\|(U_{j,t-1}^l)^T\right\|_F^2\left\|\nabla F_j^l(\mathbf{w}_j^{t-1})\right\|_F^2\right] \\
&\leq \sum_{l=\rho+1}^{L} \frac{\gamma^2}{2}\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] + (L-\rho)\frac{\gamma^2 G^4\kappa_{uv}^2}{N^2} + (L-\rho)\frac{\gamma^2\kappa_u^2\kappa_v^2 G^4(N-1)^2}{N^2}.
\end{aligned}
\tag{60}
$$

Here (a) follows from $\|P+Q\|_F^2 \leq 2\|P\|_F^2 + 2\|Q\|_F^2$ for any two matrices P and Q; (b) comes from the inequality $\|\sum_{i=1}^n \mathbf{Z}_i\|_F^2 \leq n\sum_{i=1}^n \|\mathbf{Z}_i\|_F^2$ for any matrix $\mathbf{Z}_i$; (c) follows form $\|PQ\|_F \leq \|P\|_F\|Q\|_F$ for any two matrices P and Q, and the last inequality comes from Assumption D.2 and Assumption D.3. Substituting the results of Eq. (57), Eq. (59) and Eq. (60) into Eq. (56), we can get

$$
\begin{aligned}
&\sum_{l=\rho+1}^{L}\mathbb{E}\left[\mathrm{vec}(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\cdot\mathrm{vec}(\bar{U}_t^l(\bar{V}_t^l)^T - \bar{U}_{t-1}^l(\bar{V}_{t-1}^l)^T)\right] \\
&\leq \sum_{l=\rho+1}^{L}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\bar{V}_{t-1}^l\right\|_F^2 + \left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l\right\|_F^2\right] + 2\gamma^3L_s^2E^2G_g^2 \\
&\quad \sum_{l=\rho+1}^{L}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|(\bar{U}_{t-1}^l)^T\nabla f^l(\bar{\mathbf{w}}^{t-1}))\right\|_F^2 + \left\|\frac{1}{N}\sum_{i=1}^{N}(U_{i,t-1}^l)^T\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] + 2\gamma^3L_s^2E^2G_g^2 \\
&\quad + \sum_{l=\rho+1}^{L}\frac{\gamma^2}{2}\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] + (L-\rho)\frac{\gamma^2G^4}{N^2}(\kappa_{uv}^2 + \kappa_u^2\kappa_v^2(N-1)^2).
\end{aligned}
\tag{61}
$$

Combining the results of Eq. (53) and Eq. (61), we finally get

$$\begin{aligned}
&\mathbb{E}[\langle \nabla f(\bar{\mathbf{w}}^{t-1}), \bar{\mathbf{w}}^{t} - \bar{\mathbf{w}}^{t-1} \rangle] \\
&\leq \sum_{l=1}^{\rho} (-\frac{\gamma}{2}) \mathbb{E} \left[\left\| \nabla f^l(\bar{\mathbf{w}}^{t-1}) \right\|_F^2 + \left\| \frac{1}{N} \sum_{i=1}^{N} \nabla f_i^l(\mathbf{w}_i^{t-1}) \right\|_F^2 \right] \\
&\quad + \sum_{l=\rho+1}^{L} (-\frac{\gamma}{2}) \mathbb{E} \left[\left\| \nabla f^l(\bar{\mathbf{w}}^{t-1}) \bar{V}_{t-1}^l \right\|_F^2 + \left\| \frac{1}{N} \sum_{i=1}^{N} \nabla f_i^l(\mathbf{w}_i^{t-1}) V_{i,t-1}^l \right\|_F^2 \right] \\
&\quad + \sum_{l=\rho+1}^{L} (-\frac{\gamma}{2}) \mathbb{E} \left[\left\| (\bar{U}_{t-1}^l)^T \nabla f^l(\bar{\mathbf{w}}^{t-1})) \right\|_F^2 + \left\| \frac{1}{N} \sum_{i=1}^{N} (U_{i,t-1}^l)^T \nabla f_i^l(\mathbf{w}_i^{t-1}) \right\|_F^2 \right] \\
&\quad + \sum_{l=\rho+1}^{L} \frac{\gamma^2}{2} \mathbb{E} \left[\left\| \nabla f^l(\bar{\mathbf{w}}^{t-1}) \right\|_F^2 \right] + \frac{\gamma}{2} L_s^2 \Gamma + 4\gamma^3 L_s^2 E^2 G_g^2 + (L-\rho) \frac{\gamma^2 G^4}{N^2} (\kappa_{uv}^2 + \kappa_u^2 \kappa_v^2 (N-1)^2).
\end{aligned} \tag{62}$$

D.4. Proof of Theorem D.10 (Theorem 4.5)

Theorem D.10. *Under, Assumption D.1, Assumption D.2, Assumption D.3 Assumption D.4, let q_0 be a constant and $1 < q_0 < 2$, if $0 < \gamma \leq \min\{\psi_{uv}^{\frac{2}{q_0-1}}, \frac{1}{L_s}, 1\}$, then for all $T \geq 1$, we have:*

$$\frac{1}{T} \sum_{i=1}^{T} \mathbb{E} \left[\left\| \nabla f(\bar{\mathbf{w}}^{t-1}) \right\|_2^2 \right] \leq \frac{2}{\gamma^{q_0} T} (f(\bar{\mathbf{w}}^0) - f^\star) + \mathcal{O}(\gamma^{2-q_0})), \tag{63}$$

where $f^\star$ is the minimum value of problem (6).

Proof. Fix $t \geq 1$. According to Assumption D.1, we have

$$\mathbb{E}[f(\bar{\mathbf{w}}^t)] \leq \mathbb{E}[f(\bar{\mathbf{w}}^{t-1})] + \mathbb{E}[\langle \nabla f(\bar{\mathbf{w}}^{t-1}), \bar{\mathbf{w}}^t - \bar{\mathbf{w}}^{t-1} \rangle] + \frac{L_s}{2} E[\left\| \bar{\mathbf{w}}^t - \bar{\mathbf{w}}^{t-1} \right\|_2^2]. \tag{64}$$

Following the results of Lemma D.8 and Lemma D.9, we have

$$\begin{aligned}
\mathbb{E}[f(\bar{\mathbf{w}}^t)] &\leq \mathbb{E}[f(\bar{\mathbf{w}}^{t-1})] + \sum_{l=1}^{\rho} (\frac{\gamma^2 L_s}{2}) \mathbb{E} \left[\left\| \frac{1}{N} \sum_{i=1}^{N} \nabla f_i^l(\mathbf{w}_i^{t-1}) \right\|_F^2 \right] \\
&\quad + \sum_{l=1}^{\rho} (-\frac{\gamma}{2}) \mathbb{E} \left[\left\| \nabla f^l(\bar{\mathbf{w}}^{t-1}) \right\|_F^2 \right] + \sum_{l=1}^{\rho} (-\frac{\gamma}{2}) \mathbb{E} \left[\left\| \frac{1}{N} \sum_{i=1}^{N} \nabla f_i^l(\mathbf{w}_i^{t-1}) \right\|_F^2 \right] \\
&\quad + \sum_{l=\rho+1}^{L} (-\frac{\gamma}{2}) \mathbb{E} \left[\left\| \nabla f^l(\bar{\mathbf{w}}^{t-1}) \bar{V}_{t-1}^l \right\|_F^2 \right] + \sum_{l=\rho+1}^{L} (-\frac{\gamma}{2}) \mathbb{E} \left[\left\| \frac{1}{N} \sum_{i=1}^{N} \nabla f_i^l(\mathbf{w}_i^{t-1}) V_{i,t-1}^l \right\|_F^2 \right] \\
&\quad + \sum_{l=\rho+1}^{L} (-\frac{\gamma}{2}) \mathbb{E} \left[\left\| (\bar{U}_{t-1}^l)^T \nabla f^l(\bar{\mathbf{w}}^{t-1}) \right\|_F^2 \right] + \sum_{l=\rho+1}^{L} (-\frac{\gamma}{2}) \mathbb{E} \left[\left\| \frac{1}{N} \sum_{i=1}^{N} (U_{i,t-1}^l)^T \nabla f_i^l(\mathbf{w}_i^{t-1}) \right\|_F^2 \right] \\
&\quad + \sum_{l=\rho+1}^{L} (\frac{\gamma^2}{2}) \mathbb{E} \left[\left\| \nabla f^l(\bar{\mathbf{w}}^{t-1}) \right\|_F^2 \right] + \frac{\gamma^2 L_s \sigma^2}{2N} + \frac{3}{2} \gamma^2 L_s (L-\rho) G^2 [\kappa_u^4 + \kappa_v^4 + \frac{2\gamma^2 G^2}{N^2} (\kappa_{uv}^2 + (N-1)^2 \kappa_u^2 \kappa_v^2)] \\
&\quad + \frac{\gamma}{2} L_s^2 \Gamma + 4\gamma^3 L_s^2 E^2 G_g^2 + (L-\rho) \frac{\gamma^2 G^4}{N^2} (\kappa_{uv}^2 + \kappa_u^2 \kappa_v^2 (N-1)^2).
\end{aligned} \tag{65}$$

Consider the item $\frac{\gamma^2 L_s \sigma^2}{2N} + \frac{3}{2} \gamma^2 L_s (L-\rho) G^2 [\kappa_u^4 + \kappa_v^4 + \frac{2\gamma^2 G^2}{N^2} (\kappa_{uv}^2 + (N-1)^2 \kappa_u^2 \kappa_v^2)] + \frac{\gamma}{2} L_s^2 \Gamma + 4\gamma^3 L_s^2 E^2 G_g^2 + (L-\rho) \frac{\gamma^2 G^4}{N^2} (\kappa_{uv}^2 + \kappa_u^2 \kappa_v^2 (N-1)^2)$, since the lowest power with respect to γ in this term is 2, we use the notation $\mathcal{O}(\gamma^2)$ to

represent this item. Then we have

$$
\begin{aligned}
&\mathbb{E}[f(\bar{\mathbf{w}}^t)] \\
&\overset{(a)}{\leq} \mathbb{E}[f(\bar{\mathbf{w}}^{t-1})] + \sum_{l=1}^{\rho}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] + \sum_{l=\rho+1}^{L}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla g^l(\bar{\mathbf{x}}^{t-1})\right\|_F^2\right] \\
&\quad + \sum_{l=\rho+1}^{L}(\frac{\gamma^2}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] + \sum_{l=1}^{\rho}(-\frac{\gamma-\gamma^2 L_s}{2})\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] + \mathcal{O}(\gamma^2) \\
&\overset{(b)}{\leq} \mathbb{E}[f(\bar{\mathbf{w}}^{t-1})] + \sum_{l=1}^{\rho}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] + \sum_{l=\rho+1}^{L}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla g^l(\bar{\mathbf{x}}^{t-1})\right\|_F^2\right] \\
&\quad + \sum_{l=\rho+1}^{L}(\frac{\gamma^2}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] + \mathcal{O}(\gamma^2) \\
&\overset{(c)}{\leq} \mathbb{E}[f(\bar{\mathbf{w}}^{t-1})] + \mathcal{O}(\gamma^2) + \sum_{l=1}^{L}(-\frac{\gamma^{q_0}}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] \\
&\quad + \sum_{l=\rho+1}^{L}\gamma(\gamma^{q_0-1}\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] - \frac{1}{2}\mathbb{E}\left[\left\|\nabla g^l(\bar{\mathbf{x}}^{t-1})\right\|_F^2\right]),
\end{aligned}
\tag{66}
$$

where (a) follows by noticing that $\mathbb{E}\left[\left\|\nabla g^l(\bar{\mathbf{x}}^{t-1})\right\|_F^2\right] = \mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\bar{V}_{t-1}^l\right\|_F^2\right] + \mathbb{E}\left[\left\|(\bar{U}_{t-1}^l)^T\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right]$ for $\rho < l \leq L$, and $-\frac{\gamma}{2}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i^l(\mathbf{w}_i^{t-1})V_{i,t-1}^l\right\|_F^2\right] \leq 0$, $-\frac{\gamma}{2}\mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}(U_{i,t-1}^l)^T\nabla f_i^l(\mathbf{w}_i^{t-1})\right\|_F^2\right] \leq 0$; (b) follows from $0 < \gamma \leq \frac{1}{L_s}$ and (c) follows from the fact that $-\frac{\gamma}{2} \leq -\frac{\gamma^{q_0}}{2}$ and $\frac{\gamma^2}{2} \leq \frac{\gamma^{q_0}}{2}$ when $0 < \gamma < 1$ and here q_0 is a constant with $1 < q_0 < 2$. Furthermore, notice that

$$
\begin{aligned}
\mathbb{E}\left[\left\|\nabla g^l(\bar{\mathbf{x}}^{t-1})\right\|_F^2\right] &= \mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\bar{V}_{t-1}^l\right\|_F^2\right] + \mathbb{E}\left[\left\|(\nabla f^l(\bar{\mathbf{w}}^{t-1}))^T\bar{U}_{t-1}^l\right\|_F^2\right] \\
&= \mathbb{E}\left[\left\|((\bar{V}_{t-1}^l)^T\nabla f^l(\bar{\mathbf{w}}^{t-1})^T)\right\|_F^2\right] + \mathbb{E}\left[\left\|(\bar{U}_{t-1}^l)^T\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] \\
&\geq \left[\sigma_{min}^2((\bar{V}_{t-1}^l)^T) + \sigma_{min}^2((\bar{U}_{t-1}^l)^T)\right]\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] \\
&\geq 2\psi_{uv}^2\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right],
\end{aligned}
$$

where the first inequality follows from inequality $\sigma_{min}(U)\left\|V\right\|_F \leq \left\|UV\right\|_F$ for any matrix $U \in \mathcal{R}^{m\times r}$ and matrix $V \in \mathcal{R}^{r\times n}$ (the proof of this inequality can be found in Lemma B.3 in (Zou et al., 2020)). The last inequality follows from Assumption D.4. If $\gamma^{q_0-1}\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] \leq \psi_{uv}^2\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] \Rightarrow \gamma \leq \sqrt[q_0-1]{\psi_{uv}^2}$, we have $\gamma^{q_0-1}\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] - \frac{1}{2}\mathbb{E}\left[\left\|\nabla g^l(\bar{\mathbf{x}}^{t-1})\right\|_F^2\right] \leq 0$, at this time we have

$$
\sum_{l=1}^{L}(\frac{\gamma^{q_0}}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] \leq \mathbb{E}[f(\bar{\mathbf{w}}^{t-1})] - \mathbb{E}[f(\bar{\mathbf{w}}^t)] + \mathcal{O}(\gamma^2). \tag{67}
$$

Dividing Eq. (67) both sides by $\frac{\gamma^{q_0}}{2}$, summing over $t \in \{1, 2, \cdots, T\}$ and dividing both sides by T yields

$$
\begin{aligned}
\frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\left[\left\|\nabla f(\bar{\mathbf{w}}^{t-1})\right\|_2^2\right] &\leq \frac{2}{\gamma^{q_0}T}(f(\bar{\mathbf{w}}^0) - \mathbb{E}[f(\bar{\mathbf{w}}^T)]) + \mathcal{O}(\gamma^{2-q_0})) \\
&\leq \frac{2}{\gamma^{q_0}T}(f(\bar{\mathbf{w}}^0) - f^\star) + \mathcal{O}(\gamma^{2-q_0})),
\end{aligned}
\tag{68}
$$

where the last inequality follows because $f^\star$ is the minimum value of problem (6). □

D.5. Proof of Corollary D.11

Corollary D.11. *Consider problem (6) under Assumption D.1, Assumption D.2, Assumption D.3 and Assumption D.4, and recall that $1 < q_0 < 2$. If we choose $\gamma = \frac{1}{\sqrt{T}}$, then we have*

$$\begin{aligned}
\frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\left[\left\|\nabla f(\bar{\mathbf{w}}^{t-1})\right\|_2^2\right] \leq & \frac{2}{T^{\frac{2-q_0}{2}}}(f(\bar{\mathbf{w}}^0) - f^\star) \\
& + \frac{1}{T^{(\frac{2-q_0}{2})}}\left[\frac{L_s\sigma^2}{2N} + \frac{3}{2}L_2(L-\rho)G^2(\kappa_u^4+\kappa_v^4) + \frac{(L-\rho)G^4}{N^2}(\kappa_{uv}^2 + \kappa_u^2\kappa_v^2(N-1)^2)\right] \\
& + \frac{1}{T^{(\frac{3-q_0}{2})}}\left[2\rho L_s^2E^2G^2 + 6(L-\rho)L_s^2E^2G^2(\kappa_u^4+\kappa_v^4) + 4L_s^2E^2G_g^2\right] \\
& + \frac{1}{T^{(\frac{4-q_0}{2})}}\left[\frac{3L_s(L-\rho)G^4}{N^2}(\kappa_{uv}^2 + (N-1)^2\kappa_u^2\kappa_v^2)\right] \\
& + \frac{1}{T^{(\frac{5-q_0}{2})}}\left[3(L-\rho)L_s^2E^4G^4(\kappa_u^4+\kappa_v^4)\right].
\end{aligned} \tag{69}$$

Proof. Since $\Gamma = 4\rho\gamma^2E^2G^2 + 6\gamma^2(\kappa_u^4+\kappa_v^4)E^2G^2(L-\rho)(2+\gamma^2E^2G^2)$, we have

$$\begin{aligned}
\mathcal{O}(\gamma^2) = & \frac{\gamma^2L_s\sigma^2}{2N} + \frac{3}{2}\gamma^2L_s(L-\rho)G^2[\kappa_u^4+\kappa_v^4 + \frac{2\gamma^2G^2}{N^2}(\kappa_{uv}^2 + (N-1)^2\kappa_u^2\kappa_v^2)] + \frac{\gamma}{2}L_s^2\Gamma \\
& + 4\gamma^3L_s^2E^2G_g^2 + (L-\rho)\frac{\gamma^2G^4}{N^2}(\kappa_{uv}^2 + \kappa_u^2\kappa_v^2(N-1)^2) \\
= & \gamma^2\left[\frac{L_s\sigma^2}{2N} + \frac{3}{2}L_2(L-\rho)G^2(\kappa_u^4+\kappa_v^4) + \frac{(L-\rho)G^4}{N^2}(\kappa_{uv}^2 + \kappa_u^2\kappa_v^2(N-1)^2)\right] \\
& + \gamma^3\left[2\rho L_s^2E^2G^2 + 6(L-\rho)L_s^2E^2G^2(\kappa_u^4+\kappa_v^4) + 4L_s^2E^2G_g^2\right] \\
& + \gamma^4\left[\frac{3L_s(L-\rho)G^4}{N^2}(\kappa_{uv}^2 + (N-1)^2\kappa_u^2\kappa_v^2)\right] \\
& + \gamma^5\left[3(L-\rho)L_s^2E^4G^4(\kappa_u^4+\kappa_v^4)\right].
\end{aligned} \tag{70}$$

Dividing Eq. (70) both sides by γ^{q_0} and replacing γ with $\frac{1}{\sqrt{T}}$ into Eq. (63) yield the result. □

D.6. Proof of Corollary D.12 (Corollary 4.6)

Corollary D.12. *Consider problem (6) and problem (7) under Assumption D.1, Assumption D.2, Assumption D.3 and Assumption D.4, and recall that $1 < q_0 < 2$, if we choose*

$$0 < \gamma < \min\{\frac{3(\kappa_u^4+\kappa_v^4)}{4L_sE^2}, \psi_{uv}^{\frac{2}{q_0-1}}, \frac{1}{L_s}, 1\}, \tag{71}$$

then the optimal $\rho^\star = L$ which can minimize the error bound in Eq. (63).

Proof. Revisiting the constant term Eq. (63), if we think of it as a function of ρ, then we can rewrite it as

$$\mathcal{O}(\gamma^{2-q_0}) = C_1\rho + C_2, \tag{72}$$

where

$$\begin{aligned}
C_1 = & -\gamma^{2-q_0}\left[\frac{3}{2}L_sG^2(\kappa_u^4+\kappa_v^4) + \frac{G^4}{N^2}(\kappa_{uv}^2 + \kappa_u^2\kappa_v^2(N-1)^2)\right] + 2\gamma^{3-q_0}L_s^2E^2G^2 - 6\gamma^{3-q_0}L_s^2E^2G^2(\kappa_u^4+\kappa_v^4) \\
& - \gamma^{4-q_0}\frac{3L_sG^4}{N^2}(\kappa_{uv}^2 + (N-1)^2\kappa_u^2\kappa_v^2) - 3\gamma^{5-q_0}L_s^2E^4G^4(\kappa_u^4+\kappa_v^4)
\end{aligned} \tag{73}$$

$$C_2 = \gamma^{2-q_0}\left[\frac{L_s\sigma^2}{2N} + \frac{3}{2}L_2LG^2(\kappa_u^4+\kappa_v^4) + \frac{LG^4}{N^2}(\kappa_{uv}^2+\kappa_u^2\kappa_v^2(N-1)^2)\right] + \gamma^{3-q_0}\left[6LL_s^2E^2G^2(\kappa_u^4+\kappa_v^4)\right] + \gamma^{3-q_0}\left[4L_s^2E^2G_g^2\right] + \gamma^{4-q_0}\left[\frac{3L_sLG^4}{N^2}(\kappa_{uv}^2+(N-1)^2\kappa_u^2\kappa_v^2)\right] + \gamma^{5-q_0}\left[3LL_s^2E^4G^4(\kappa_u^4+\kappa_v^4)\right]. \tag{74}$$

We find that $\mathcal{O}(\gamma^2)$ is a linear function of ρ, furthermore, if $\gamma < \frac{3(\kappa_u^4+\kappa_v^4)}{4L_sE^2} \Rightarrow -\gamma^{2-q_0}\frac{3}{2}L_sG^2(\kappa_u^4+\kappa_v^4)+2\gamma^{3-q_0}L_s^2E^2G^2 < 0$, then we have $C_1 < 0$, at this time to minimize the error bound $\mathcal{O}(\gamma^2)$, we have the optimal $\rho^\star = L$. □

D.7. Proof of Theorem D.13

Theorem D.13. *Under Assumption D.1, Assumption D.2, Assumption D.3, if* $0 < \gamma < \frac{1}{L_s}$, *then for all* $T \geq 1$, *we have:*

$$\frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\left[\left\|\nabla g(\bar{\mathbf{x}}^{t-1})\right\|_2^2\right] \leq \frac{2}{\gamma T}(g(\bar{\mathbf{x}}^0) - f^\star) + \mathcal{O}(\gamma)), \tag{75}$$

where $f^\star$ *is the minimum value of problem (6).*

Proof. According to Eq. (66), we have

$$\begin{aligned}\mathbb{E}[f(\bar{\mathbf{w}}^t)] &\leq \mathbb{E}[f(\bar{\mathbf{w}}^{t-1})] + \mathbb{E}[\langle\nabla f(\bar{\mathbf{w}}^{t-1}), \bar{\mathbf{w}}^t - \bar{\mathbf{w}}^{t-1}\rangle] + \frac{L_s}{2}E[\left\|\bar{\mathbf{w}}^t - \bar{\mathbf{w}}^{t-1}\right\|_2^2] \\ &\leq \mathbb{E}[f(\bar{\mathbf{w}}^{t-1})] + \mathcal{O}(\gamma^2) + \sum_{l=1}^{\rho}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] + \sum_{l=\rho+1}^{L}(-\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla g^l(\bar{\mathbf{x}}^{t-1})\right\|_F^2\right] \\ &\quad + \sum_{l=\rho+1}^{L}(\frac{\gamma^2}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right],\end{aligned}$$

since $\sum_{l=1}^{\rho}(\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] = \sum_{l=1}^{\rho}(\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla g^l(\bar{\mathbf{x}}^{t-1})\right\|_F^2\right]$, then we get:

$$\sum_{l=1}^{L}(\frac{\gamma}{2})\mathbb{E}\left[\left\|\nabla g^l(\bar{\mathbf{x}}^{t-1})\right\|_F^2\right] \leq \mathbb{E}[f(\bar{\mathbf{w}}^{t-1})] - \mathbb{E}[f(\bar{\mathbf{w}}^t)] + \mathcal{O}(\gamma^2) + \sum_{l=\rho+1}^{L}(\frac{\gamma^2}{2})\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right]. \tag{76}$$

Notice that

$$\begin{aligned}\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2 &= \left\|\nabla F^l(\bar{\mathbf{w}}^{t-1}) - \nabla f^l(\bar{\mathbf{w}}^{t-1}) - \nabla F^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2 \\ &\leq 2\left\|\nabla F^l(\bar{\mathbf{w}}^{t-1}) - \nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2 + 2\left\|\nabla F^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2,\end{aligned} \tag{77}$$

according to Assumption D.2, we have

$$\begin{aligned}\mathbb{E}\left[\left\|\nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] &\leq 2\mathbb{E}\left[\left\|\nabla F^l(\bar{\mathbf{w}}^{t-1}) - \nabla f^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] + 2\mathbb{E}\left[\left\|\nabla F^l(\bar{\mathbf{w}}^{t-1})\right\|_F^2\right] \\ &\leq 2(\sigma^2 + G^2).\end{aligned} \tag{78}$$

Substituting Eq. (78) into Eq. (76), dividing Eq. (76) both sides by $\frac{\gamma}{2}$, summing over $t \in \{1, 2, \cdots, T\}$, dividing both sides by T and noticing that $g(\bar{\mathbf{x}}^0) = f(\bar{\mathbf{w}}^0)$ yields

$$\frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\left[\left\|\nabla g(\bar{\mathbf{x}}^{t-1})\right\|_2^2\right] \leq \frac{2}{\gamma T}(g(\bar{\mathbf{x}}^0) - f^\star) + \mathcal{O}(\gamma)), \tag{79}$$

where $f^\star$ is the minimum value of problem (6). □

Remark D.14. The whole process of training low-rank model $\mathbf{x}$ by solving problem (7) can be thought of as a vanilla federated learning process, therefore, according to the result of previous work (Yu et al., 2019), if $0 < \gamma \leq \frac{1}{L_s}$, we also have

$$\frac{1}{T}\sum_{i=1}^{T}\mathbb{E}\left[\left\|\nabla g(\bar{\mathbf{x}}^{t-1})\right\|_2^2\right] \leq \frac{2}{\gamma T}(g(\bar{\mathbf{x}}^0) - g^\star) + \mathcal{O}(\gamma)), \tag{80}$$

Table 6. Hyper parameters and model architecture used in experiments.

Dataset	SVHN	CIFAR10	CIFAR100	Tiny-ImageNet	WikiText-2
Model	ResNet-18	ResNet-18	ResNet-18	ResNet-18	Transformer
Hidden size	[64, 128, 256, 512]	[64, 128, 256, 512]	[64, 128, 256, 512]	[64, 128, 256, 512]	[64, 128, 256, 512]
Local epoch E	1	1	1	1	1
Local Batch size B	64	64	64	64	100
Optimizer	SGD	SGD	SGD	SGD	SGD
Momentum	0.9	0.9	0.9	0.9	0.9
Communication round	2000	2000	3000	3000	200
Learning rate γ	0.1	0.1	0.1	0.1	0.01
Scheduler	Cosine Anneal	Cosine Anneal	Cosine Anneal	Cosine Anneal	Cosine Anneal
Embedding size					128
Number of head			N/A		8
Dropout					0.1
Sequence length					64

where $g^\star$ is the minimum value of problem (7). If we choose $\gamma = \frac{1}{\sqrt{T}}$ and T is large enough, both Eq. (79) and Eq. (80) show that we can solve problem (7) with convergence rate $\mathcal{O}(\frac{1}{\sqrt{T}})$.

□

E. Details of Experimental Setup

E.1. Datasets and Models

CIFAR10 and CIFAR100. CIFAR10 and CIFAR100 (Krizhevsky et al., 2009) are labeled subsets of the 80 million tiny images dataset. The CIFAR10 dataset consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class, and there are 50,000 training images and 10,000 test images; The CIFAR100 dataset consists of 60,000 32x32 colour images in 100 classes, with 600 images per class and there are 500 training images and 100 test images. We normalize the images with channel means and standard deviations for pre-processing. The data augmentation is performed by 4x4 random translation followed by random horizontal flip (He et al., 2016).

SVHN. SVHN (Netzer et al., 2011) consists of 32x32 colored images of digits. 73,257 images for training and 26,032 images for testing are provided. We also normalize the images with channel means and standard deviations for pre-processing.

Tiny-ImageNet. Tiny-ImageNet (Chrabaszcz et al., 2017) is constructed from ImageNet and it consists of 100,000 64x64 color images in 200 classes. There are 500 training images, 50 validation images, and 50 testing images for each class. We normalize the images with channel means and standard deviations for pre-processing.

WikiText2. The WikiText2 language modeling dataset (Merity et al., 2016) is a collection of over 2,000,000 word count from the set of verified Good and Featured articles on Wikipedia. As it is composed of full articles, this dataset is well-suited for models that can take advantage of long-term dependencies.

For CIFAR10, CIFAR100, SVHN, and Tiny-ImageNet, we use ResNet-18 which is the same as that in (He et al., 2016; Mei et al., 2022). For WikiText2, we train a Transformer which is the same as that in (Alam et al., 2022). As for the data partitioning, for image classification tasks, the data is distributed in a non-IID manner, as in (Hsu et al., 2019; Kim et al., 2023), a Dirichlet distribution $z_c \sim \text{Dir}(\eta)(\eta = 0.5)$ is used to allocate to client m a fraction of $p_{c,m}$ of all training instances belonging to class c. For WikiText2, we conduct a masked language modeling task with a 15% masking rate and uniformly assign balanced data examples for each client, as the same in (Diao et al., 2021). For all datasets, we maintained the original train/test data split and used 20% of the training dataset as the validation dataset.

E.2. Experimental Details of Table 4

We train a complete low-rank ResNet-18 model on CIFAR10 and CIFAR100 datasets. The batch size is 128, the initial learning rate is 0.1 with the cosine annealing schedule, and the training epoch is 200. For all datasets, we maintain the original train/test data split and used 20% of the training dataset as the validation dataset. The coefficient of regularization is tuned on the validation dataset via search on the grid $\{10^{-9}, 10^{-8}, \cdots, 10^{1}\}$.

E.3. Implementation Details

Model Heterogeneity. Let us start by introducing the model partitioning of each approach in detail using ResNet-18 as an example. We set $\boldsymbol{\beta} = \{\beta_1, \beta_2, \beta_3, \beta_4\}$ following the description of their original papers (Diao et al., 2021; Alam et al., 2022; Kim et al., 2023). Here, β_4 means training the original full model, which has a maximum model size than β_3, β_2, and β_1 setting. Under the β_4 setting, the model size may vary slightly among different baselines. For example, since *depth-scale* methods require additional classification headers for distillation, the model trained by *depth-scale* methods is larger than that trained by *width-scale* methods under β_4 setting.

- For *width-scale* and *depth-scale* methods, following the original papers, we set $\boldsymbol{\beta} = \{\beta_1 = 1/4, \beta_2 = 1/2, \beta_3 = 3/4, \beta_4 = 1\}$. Here $\beta_2 = 1/2$ means the number of channels per layer in the client's local sub-model is half of the full model for *width-scale* methods, while for *depth-scale* methods it means the number of layers in the client's local sub-model is half of the full model.

- For FedHM, following the original paper, we set $\rho = 9$ and adjust the compression ratio α so that it is similar to *width-scale* methods under β_1, β_2, β_3 and β_4 setting. For FLANC, we adjust the compression ratio so that it is similar to *width-scale* methods, too.

- For FedLMT, we adjust the rank ratio α and ρ so that FedLMT only consumes the lowest communication and computing resources than other methods. For instance, for all clients with model capacity β_1, the communication and computational overhead required by FedLMT is lower than all other methods. The purpose of this setting is to show that even if all clients train a homogeneous model with the lowest capacity, the resulting model trained by FedLMT can still perform better than the global full model trained by other methods. Specifically, we set $\rho = 1$ and $\alpha = 0.03$ for β_1, $\rho = 3$ and $\alpha = 0.15$ for β_2, $\rho = 5$ and $\alpha = 0.4$ for β_3.

- For pFedLMT, since the aim of pFedLMT is to demonstrate how each client can get a customized model to address both data heterogeneity and system heterogeneity problems at the same time, we just need to ensure that there is a difference in computation and communication costs among the participating clients. Therefore, we let pFedLMT and FedHM have the same heterogeneous model setup.

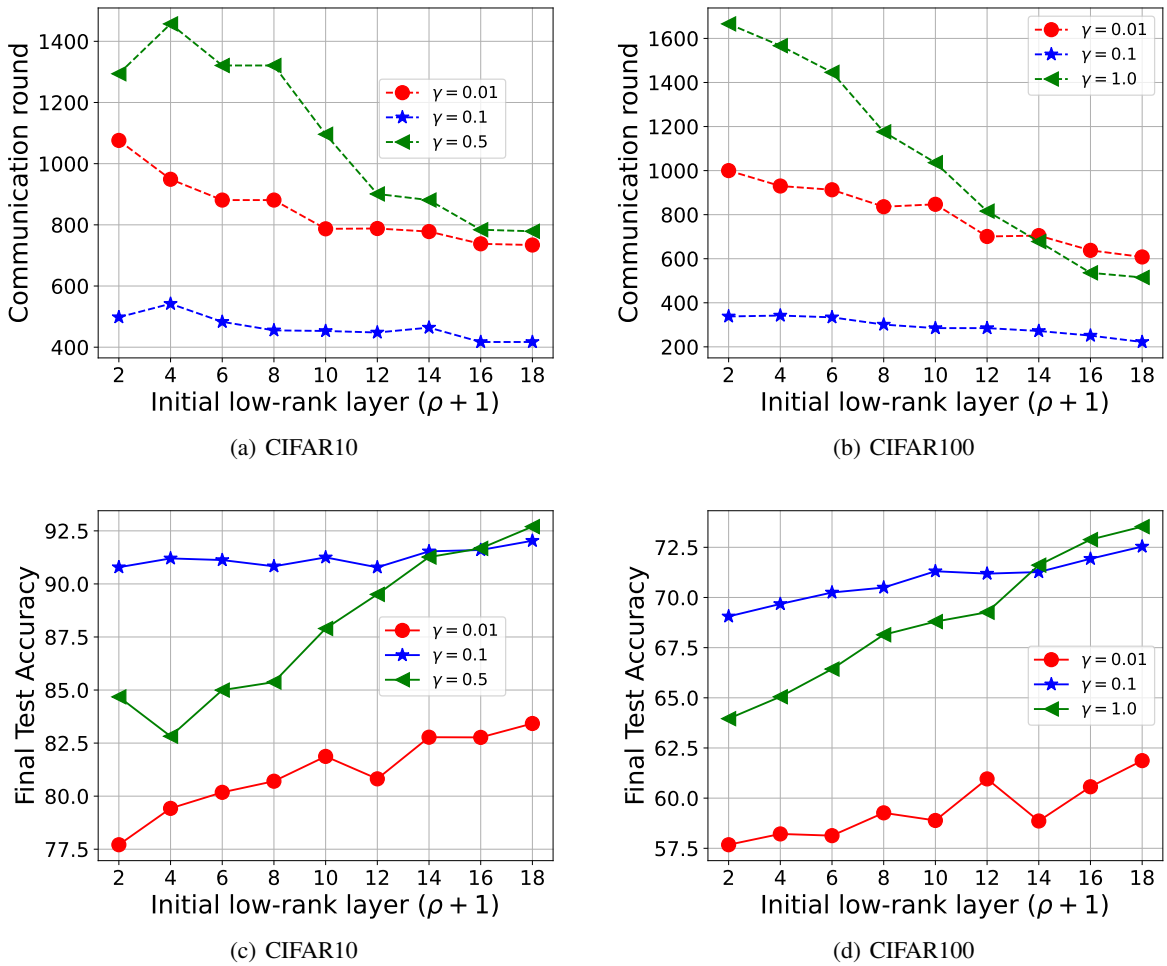
Implementation of FedLMT and other baselines. By default, Frobenius decay is used for FedLMT, and the regularization coefficient is tuned via grid search on the grid $\{10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$. Besides, for image classification tasks, FedLMT decomposes the convolutional layers according to the value of ρ while for the NLP task, only the feedforward layers are decomposed. For FedDropout, HeteroFL, and FedRolex, we follow the division of sub-models in the original paper, and the specific implementation can be found in the open source repository[2]. For FLANC, we refer to the hyper-parameter settings of their open source repository[3], and only adjust the number of feature maps in the hidden layers to generate different heterogeneous models. For FedHM, since they don't publish the source code, we reproduced their Algorithm 2 according to their experimental description. The construction of the hybrid low-rank model is consistent with the description in the original paper[4] and we generate different heterogeneous models by adjusting the low-rank ratio. Similarly, to implement DepthFL, since there is no open source code, we reproduce it with reference to the paper (Zhang et al., 2019) they cite and the corresponding repository published in (Zhang et al., 2019). The distillation temperature in DepthFL is finetuned on the grid $\{0.1, 1, 10\}$ and the weighting coefficient of the KL loss is tuned on the grid $\{0.1, 0.2, 0.5, 0.9, 1\}$.

Evaluation. Except for the experiments in Figure 1, we used the global model to calculate the model accuracy or perplexity. In the experiment of personalized scenarios (Figure 1), for pFedLMT, we use the personalized model of each client for evaluation. For other methods, since they only train a single global model, we use the global model for evaluation.

Training setting. For image classification tasks, we follow the setting in (Diao et al., 2021; Alam et al., 2022; Kim et al., 2023) and in each communication round, 10% of the clients are randomly selected from a pool of 100 clients. For the language modeling task, 5% of the clients are randomly selected from the total 100 clients in each communication round. Most of the hyper-parameters used in our experiments (Table 2 and Table 3) are depicted in Table 6.

Platforms and libraries. We implement FedLMT and other baselines using PyTorch-2.0 (Paszke et al., 2019) and Ray-1.13

[2]https://github.com/AIoT-MLSys-Lab/FedRolex
[3]https://github.com/HarukiYqM/All-In-One-Neural-Composition
[4]https://arxiv.org/abs/2111.14655



(a) CIFAR10 (b) CIFAR100

(c) CIFAR10 (d) CIFAR100

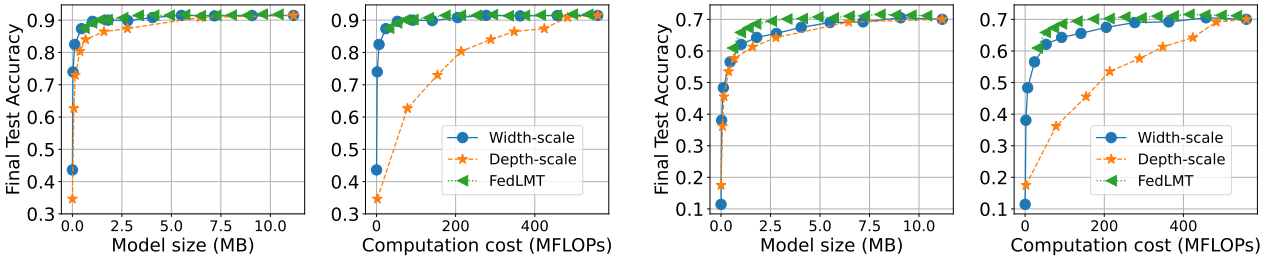
Figure 4. 4(a) and 4(b) show the number of communication rounds required when the model test accuracy exceeds x for the first time under different initial low-rank layer indices ($\rho+1$) on the CIFAR10 and CIFAR100 datasets ($x=70\%$ for the CIFAR10 dataset and $x=50\%$ for the CIFAR100 dataset). 4(c) and 4(d) show the final test accuracy of the hybrid ResNet-18 with various initial low-rank layer indices ($\rho+1$) over the CIFAR10 and CIFAR100 datasets.

(Moritz et al., 2018), and conduct all experiments on a server with two Intel(R) Xeon(R) E5-2640 CPUs (20 cores) and 4 NVIDIA RTX 3090 GPUs running Ubuntu 20.04.

F. Additional Experiments

F.1. Effect of Hyper-parameters and Verification of Corollary 4.6

Figure 9 and Figure 10 (actually the heat map of Figure 2) show the final test accuracy of the low-rank model trained under different values of ρ which means the number of undecomposed layers and low-rank ratio α. Please note that in order to better reflect the influence of ρ and α on the model performance, we don't use any regularization technique in these experiments. We observe that the *hybrid model architecture* technique can significantly improve the performance of the final trained low-rank model when the number of model parameters is very small (*i.e.*, small α), and the model performance becomes better with the increase of ρ. When the size of the model is large (*i.e.*, large α), the model performance under different ρ is similar, which indicates that as the model capacity increases, the model's ability to learn complex representations is enhanced. At this time, the use of *hybrid model architecture* technology can no longer significantly improve the model performance.



(a) CIFAR10 (b) CIFAR100

Figure 5. Comparison of different compression methods on CIFAR10 and CIFAR100 datasets under FL setting. The two sub-graphs in 5(a) and 5(b) show the relationship between the global model's performance and the model size, and the relationship between the global model's performance and computation budget, respectively.

To verify Corollary 4.6, Figure 4 shows the effect of ρ on model performance and convergence rate with different initial learning rates γ using ResNet-18 under FL setting. We find that the global model can't be trained well if the learning rate γ is too large, so we only show the case when γ is less than 1. Other training details can be found in Table 6. From Figure 4(a) and Figure 4(b), we can see that as ρ increases, the number of rounds required to exceed the target accuracy is lower, which means that the final model converges faster. In particular, when $\rho = L$, the convergence rate is fastest, and this result is consistent with Corollary 4.6. Besides, from Figure 4(c) and Figure 4(d), we also find that the larger the ρ, the better model performance. This observation coincides with our intuition. As ρ increases, *i.e.*, the number of model parameters increases, the representational ability of the model becomes stronger and therefore we can obtain a better model (Neyshabur et al., 2019).

F.2. Effect of Model Compression under Homogeneous Setting

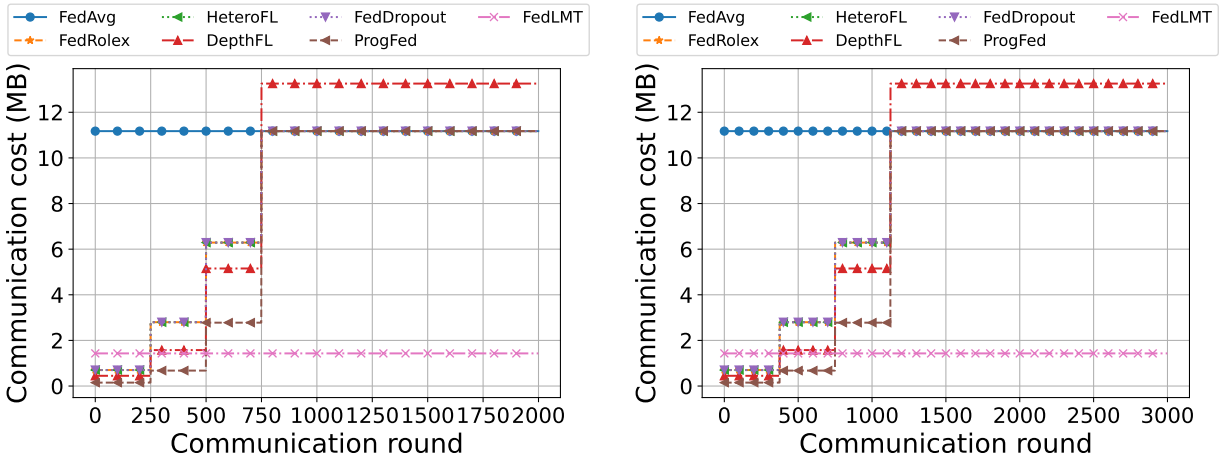
To study the effect of different compression methods, we conduct experiments on training models with low-rank method (FedLMT), *width-scale* method and *depth-scale* method under homogeneous setting, respectively. In this setting, all clients train the same global model and there is no parameter mismatch problem caused by heterogeneous aggregation. Figure 5 reports the final model accuracy under different model sizes and computational costs. We find that under the condition of limited model size, the performance of the three methods is similar, FedLMT is slightly better than the *width-scale* method and *depth-scale* method. However, under limited computing budget, it is obvious that FedLMT and *width-scale* method are better than *depth-scale* method, which indicates that *depth-scale* method requires a very large amount of computation. In summary, if not compressed severely (*e.g.*, model size $\geq$ 1MB), FedLMT consistently outperforms the *width-scale* method and *depth-scale* method.

F.3. Comparison with ProgFed (Wang et al., 2022)

In this section, we consider another popular FL scenario with system heterogeneity proposed in (Wang et al., 2022). In this setting, they assume that all clients have the ability to train the original large model, but the available resource capacity of all clients is dynamic. For example, at the moment t_1, the resources of client i are not enough to train the complete large model since at this time client i needs to perform other tasks. At the moment t_2 ($t_2 > t_1$), client i is idle and therefore has enough capacity to train the complete large model. Obviously, this setting is significantly different from the heterogeneous setting proposed in (Diao et al., 2021; Alam et al., 2022; Kim et al., 2023; Mei et al., 2022) where clients have heterogeneous resource capacity and only partial clients have the ability to train the original large model. In the following, we first review the method of ProgFed and then design experiments for performance comparison.

F.3.1. ProgFed

Recently, ProgFed (Wang et al., 2022) proposes a kind of progressive training framework for efficient and effective federated learning. It is essentially a dynamic model training method based on a fixed network depth partitioning. The authors divide the whole federated learning process into S stages, and divide the model to be trained $\mathcal{M}$ into S stages according to the



(a) CIFAR10 and SVHN (b) CIFAR100 and TINY

Figure 6. The communication cost of a single client in each round under different methods.

depth, which can be represented as (following the notation in the original paper):

$$\mathcal{M} := G_S \circ \bigcirc_{i=1}^{S} E_i = G_S \circ E_S \circ \cdots \circ E_2 \circ E_1, \tag{81}$$

where G_S is the classification head for the task and the E_i could denote, e.g., a stack of residual blocks or simply a single layer. In stage s ($s \in \{1, \cdots, S\}$ associated with the split indices) of FL training, ProgFed additionally introduces local supervision heads for supervision training and only trains the growing sub-model $\mathcal{M}^s$, which is defined as:

$$\mathcal{M} := G_s \circ \bigcirc_{i=1}^{s} E_i, \tag{82}$$

where G_s is a newly introduced head for the FL training of stage s. It is easy to see that all clients train the same model throughout the training process, and in the final training stage S, all clients train the complete full model.

F.3.2. Experiments

As we have mentioned above, ProgFed needs to assume that all clients have enough resource capacity to load the full large model, which is different from the assumption of other methods focusing on solving the system heterogeneity in FL where each client has heterogeneous resources and maybe only a few clients can train the large full model. Therefore, in order to make a fair comparison, we change the training settings of other classical work proposed in (Diao et al., 2021; Alam et al., 2022; Kim et al., 2023; Caldas et al., 2018). We also divide the training of these methods into S stages, and following the original experimental setting in (Wang et al., 2022), we set $S = 4$ and the number of training rounds in each stage s is $T_s = \frac{T}{2S}$ for $s < S$ and $T_S = \frac{T(S+1)}{2S}$ where T is the total iteration and $T = \sum_{s=1}^{S} T_s$. In each stage, every client trains a homogeneous sub-model and all clients can only train the complete full model in the last stage. For example, in stage $i (1 \leq i \leq 4)$, all clients train the same model with model capacity β_i. Since under this setting, each client has the ability to train a complete large model, therefore, for FedLMT, we let all clients train a homogeneously low-rank model with β_2 model capacity during the whole training process. The communication cost at each training round for a client is shown in Figure 6 and Table 7 shows the final top-1 test accuracy of different methods using four different datasets. The hyper-parameters used in this experiment can be found in Table 6. We find that the performance of ProgFed seems to depend on the selected dataset. For example, on CIFAR10 and CIFAR100, the performance of ProgFed is worse than HeteroFL, while on SVHN and TINY datasets, the performance of ProgFed is much better than HeteroFL. All in all, FedLMT can get better model performance than other methods with less communication and computation costs.

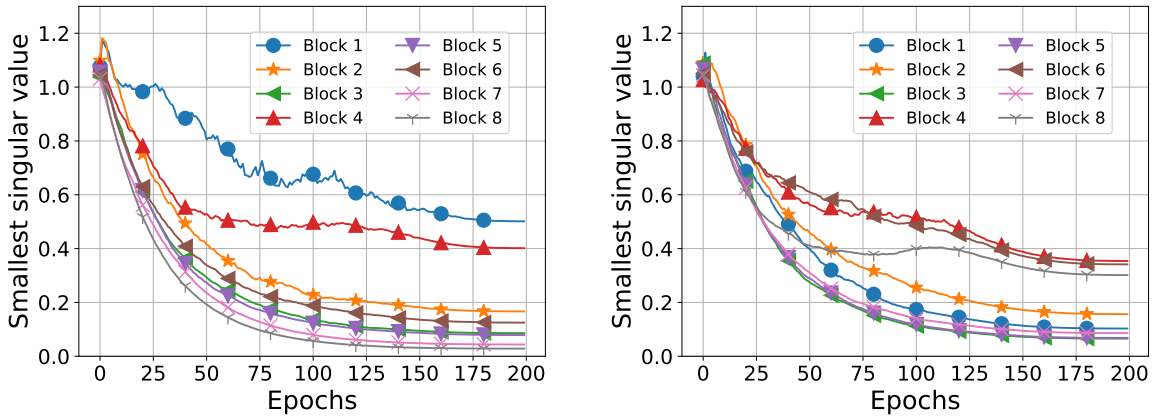
F.4. Verification of Assumption D.4 (Assumption 4.4)

We conduct experiments in both centralized and distributed settings to verify that Assumption 4.4 is valid.

In the centralized setting, we train a fully low-rank ResNet-18 model (*i.e.*, $\rho = 1$ and we set $\alpha = 0.2$) on CIFAR10 and CIFAR100 datasets, and record the smallest singular value of each layer throughout the training. The detailed training

Table 7. The performance of different methods under the same setting as ProgFed (Wang et al., 2022). ACC means top-1 test accuracy, COMM means the total communication cost including download and upload among all clients, and FLOPs denotes the total floating operations during FL training.

Task		FedAvg	FedDropout	HeteroFL	FedRolex	DepthFL	ProgFed	FedLMT
CIFAR10	Acc	91.91	73.66	89.77	90.12	85.16	74.95	**91.03**
	Comm(GB)	223.5	164.1	164.1	164.1	183.7	148.7	**28.62**
	FLOPs(1e12)	11.18	8.22	8.22	8.22	11.63	9.16	**2.80**
CIFAR100	Acc	72.20	39.76	59.37	57.06	62.38	56.02	**71.08**
	Comm(GB)	335.2	246.2	246.2	246.2	275.5	223.0	**42.93**
	FLOPs(1e12)	16.77	12.32	12.32	12.32	17.44	13.73	**4.20**
SVHN	Acc	94.39	92.76	92.93	92.67	93.78	95.06	**95.35**
	Comm(GB)	223.5	164.1	164.1	164.1	183.7	148.7	**28.62**
	FLOPs(1e12)	11.18	8.22	8.22	8.22	11.63	9.16	**2.80**
TINY	Acc	42.71	21.83	27.61	33.32	48.12	43.34	**48.53**
	Comm(GB)	335.2	246.2	246.2	246.2	275.5	223.0	**42.93**
	FLOPs(1e12)	67.02	49.26	49.26	49.26	69.76	54.91	**16.74**



(a) CIFAR10 (b) CIFAR100

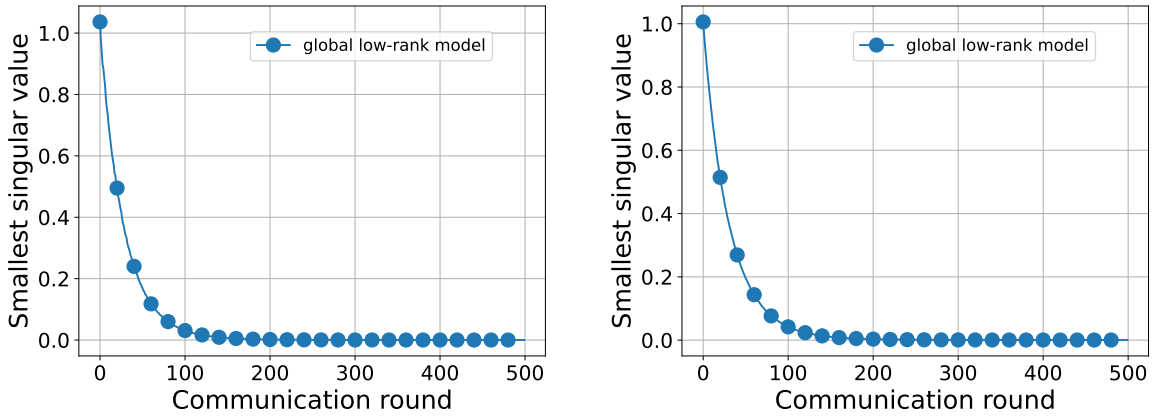
Figure 7. The smallest singular value of each block in ResNet-18 over communication round on the CIFAR10 and CIFAR100 datasets with centralized training setting.

setting is the same as Table 4 and can be found in Appendix E.2. There are a total of 18 layers in ResNet-18 and they can be divided into 8 residual blocks where each block has two convolutional layers. For the sake of demonstration, we record the smallest singular value in each block (this value is the smallest value among the two low-rank convolutional layers). Figure 7 shows the final results and Table 8 gives the specific value of the smallest singular value for each block. From the final results, we can see that Assumption 4.4 is valid.

In the distributed setting, we simulate a non-IID FL scenario using the CIFAR10 and CIFAR100 datasets where 10 clients collaboratively train a fully low-rank ResNet-18 model over 500 communication rounds. In each communication round, we record the smallest singular value among all the decomposition layers of the global model, and plot a curve of how this value changes with the number of communication rounds. Figure 8 shows the results. We find that the smallest singular value of the global model gradually decreases and becomes stable with the process of training. Using the CIFAR10 dataset, the final value is stable at 0.0013, and using the CIFAR100 dataset, the final value is stable at 0.0042. Both the results verify that Assumption 4.4 is valid.

Table 8. The smallest singular value of each block on the CIFAR10 and CIFAR100 datasets using ResNet-18 during the whole training process in the centralized setting.

Dataset	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7	Block 8
CIFAR10	0.501	0.167	0.086	0.402	0.079	0.125	0.044	0.028
CIFAR100	0.103	0.156	0.066	0.354	0.068	0.341	0.086	0.302



(a) CIFAR10 (b) CIFAR100

Figure 8. The smallest singular value of the global model over communication round on the CIFAR10 and CIFAR100 datasets with distributed training setting.

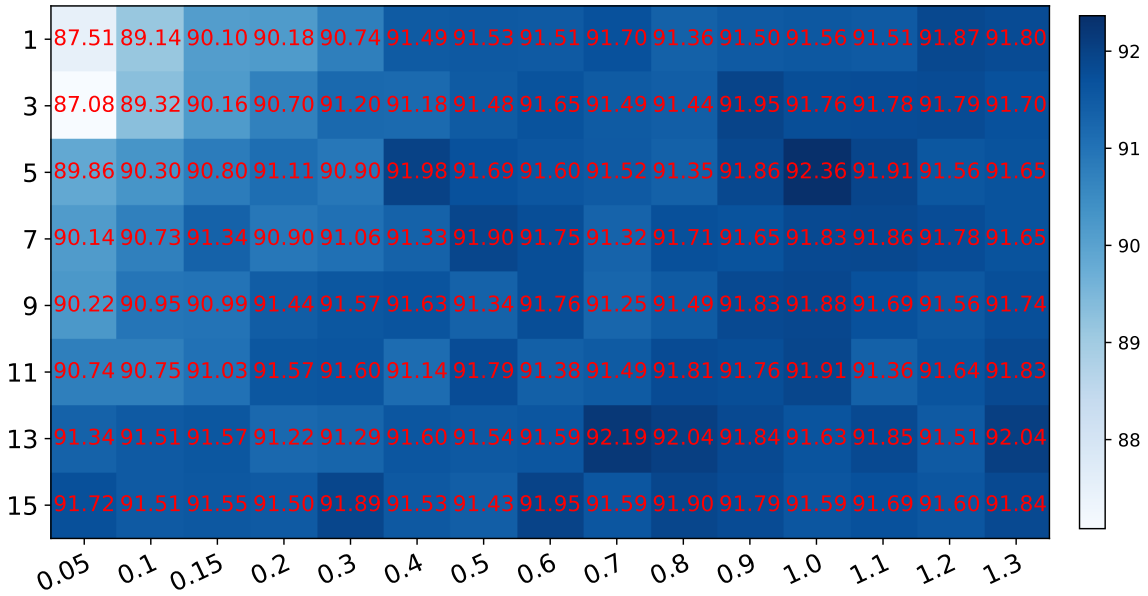


Figure 9. Heatmap of the hyper-parameters on the CIFAR10 dataset. The x-axis represents the low-rank ratio α, and the y-axis represents the value of ρ which means the number of layers that are not decomposed.

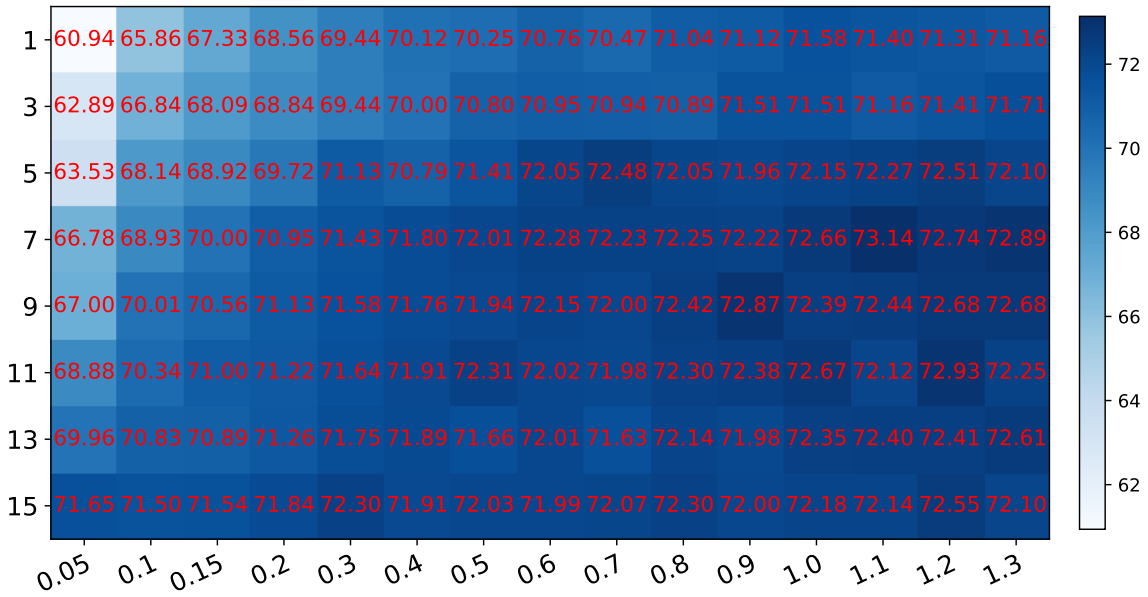


Figure 10. Heatmap of the hyper-parameters on the CIFAR100 dataset. The x-axis represents the low-rank ratio α, and the y-axis represents the value of ρ which means the number of layers that are not decomposed.