Journal of Machine Learning Research 25 (2024) 1-88 Submitted 2/23; Revised 12/24; Published 12/24

A Random Projection Approach to Personalized Federated Learning: Enhancing Communication Efficiency, Robustness, and Fairness

Yuze Han[*] hanyuze97@ruc.edu.cn
Center for Applied Statistics and School of Statistics
Renmin University of China
Beijing, China

Xiang Li lx10077@pku.edu.cn
School of Mathematical Sciences
Peking University
Beijing, China

Shiyun Lin shiyunlin@stu.pku.edu.cn
School of Mathematical Sciences
Peking University
Beijing, China

Zhihua Zhang zhzhang@math.pku.edu.cn
Schools of Mathematical Sciences and of Computer Science
Peking University
Beijing, China

Editor: Martin Jaggi

Abstract

Personalized Federated Learning (FL) faces many challenges such as expensive communication costs, training-time adversarial attacks, and performance unfairness across devices. Recent developments witness a trade-off between a reference model and local models to achieve personalization. Following the avenue, we propose a personalized FL method toward the three goals. When it is time to communicate, our method projects local models into a shared-and-fixed low-dimensional random subspace and uses infimal convolution to control the deviation between the reference model and projected local models. We theoretically show our method converges for both strongly convex and non-convex but smooth objectives with square regularizers and the convergence dependence on the projection dimension is mild. We also illustrate the benefits of robustness and fairness on a class of linear problems. Finally, we conduct a large number of experiments to show the empirical superiority of our method over several state-of-the-art methods on the three aspects.

Keywords: personalized federated learning, communication efficiency, robustness, fairness, low-dimensional random projection

[*] Alphabetical Order

1. Introduction

Federated learning (FL) emerges as a new distributed computing paradigm that would perform privately distributed optimization across massive networks of remote clients (McMahan et al., 2017). In order to safeguard privacy, data are generated locally and preserved in their original location during training, which incurs a discrepancy among local data distributions. Furthermore, the nature that FL works as a decentralized system poses greater challenges to its communication efficiency, robustness against adversarial attacks, and fairness on resource allocation (Kairouz et al., 2021).

To circumvent the problem of data heterogeneity, one considers personalizing local models (Kulkarni et al., 2020). A key feature that any personalization method has is to differentiate local models from the global model. The most straightforward method of personalization is to train models purely with local data on each device. Chen et al. (2021) showed that when the degree of data heterogeneity exceeds a certain threshold, pure local training is minimax optimal; otherwise, the global model is minimax optimal. In practice, we prefer a method that intervenes between the two extremes. It brings out another popular approach that interpolates between a reference model and local models (Hanzely and Richtárik, 2020; Hanzely et al., 2020; Deng et al., 2020; Dinh et al., 2020; Hu et al., 2022; Wu et al., 2021). Recent research (Li et al., 2021b) raised the possibility of using personalization not only to improve accuracy but also to accommodate competing criteria such as robustness and fairness. Inspired by this line of work, we would explore the following question:

Can we balance different constraints of interest (i.e., communication efficiency, robustness, and fairness) simultaneously?

In this paper, we give an affirmative answer to the question by proposing a personalized FL method named `lp-proj`, whose core consists of L^p-regularization and low-dimensional random projection. We employ the idea of controlling the dissimilarity between the global model and local models via a smoothing kernel of infimal convolution. Toward the three goals, the smoothing kernel is designed to regularize the projection of local models in a shared low-dimensional random subspace rather than the original space. By means of the above smoothing kernel, each client only communicates the projected models each time and the server maintains a low-dimensional reference model for regularization. The random subspace is generated only once and remains unchanged throughout training. It makes local models share a similar part in the random subspace and adjust to their local data using components beyond that.

Theoretically, we give convergence analysis for both convex and non-convex but smooth objectives with square regularizers and show that the convergence dependence on the projection dimension is mild. We demonstrate that in terms of Byzantine robustness (Lamport et al., 2019) and performance fairness (see Definition 1), our proposed method is at least as good as two SOTA methods (Dinh et al., 2020; Li et al., 2021b) by examining the test losses and the corresponding variances across the network on a class of federated linear regression problems.

Empirically, we perform extensive numerical experiments to demonstrate the superiority of our proposed algorithm in practice. In addition to the accuracy improvement expected from personalization, it also promotes fairness since the accuracies across all the clients are more uniform. Furthermore, it is more resilient to adversarial attacks that occur during training. More importantly, the communication efficiency is significantly improved due to the fact that the subspace dimension is often no more than one-hundredth of the original dimension.

In summary, we propose a personalized FL algorithm and explore its performance in aspects of communication efficiency, robustness, and fairness. Our results show that low-dimensional projection brings multiple benefits and is helpful for algorithm design.

The remainder of the paper is organized as follows. Firstly, we start with a literature review in Section 2. We then derive our algorithm from infimal convolution and subspace regularization in Section 3. Theoretical properties of the proposed method including convergence, robustness, and fairness are analyzed in Section 4. In Section 5, we show comparisons with various state-of-the-art benchmarks through a large number of numerical experiments. Finally, we generalize the proposed algorithm to make it suitable for large-scale applications in Section 6.

2. Related Work

In this section, we present related work from four relative aspects, i.e., federated learning involving personalization, communication efficiency, robustness, and fairness.

2.1 Personalized Federated Learning

There are many works studying personalization, and a survey can be found in Kulkarni et al. (2020). It has been studied via multi-task learning (Smith et al., 2017; Huang et al., 2021), meta-learning (Chen et al., 2018; Jiang et al., 2019; Fallah et al., 2020), knowledge distillation (Li and Wang, 2019; Yu et al., 2020b) and transfer learning (Wang et al., 2019b; Mansour et al., 2020). Hanzely et al. (2021) provided convergence analysis for a general personalized framework that requires jointly strongly convex and smooth objectives.

2.2 Communication-Efficient Federated Learning

To reduce the cost of communication in FL with large-scale networks, existing research could be classified as gradient compression, model compression, and reducing the communication frequency. Concerning gradient compression, three main directions are investigated: sparsification (Ivkin et al., 2019; Lin et al., 2018), quantization (Alistarh et al., 2017) and low-rank approximation (Azam et al., 2021). On model compression, Liang et al. (2020) suggested learning local representations and a global model only operates on the local representations, Li et al. (2021a) extended the lottery ticket hypothesis and used network pruning in the FL setting. Regarding communication frequency, McMahan et al. (2017) and Karimireddy et al. (2020) performed multiple local updates to lessen the communication rounds, while Wang et al. (2019a) used momentum to delay the global aggregation. A survey on recent progress in communication-efficient FL was given by Shahid et al. (2021). Our proposed method is a model compression approach. Differing from previous works that compress the model each time with a different basis, our work focuses on a shared-and-fixed low-dimensional subspace which is determined at the beginning of training and will not change later on.

2.3 Robust Federated Learning

Typical adversarial attacks include data poisoning and model update poisoning (Byzantine attacks). These two attacks impede the training process in different ways: the former injects abnormal sample points into the training data set (Biggio et al., 2012; Jagielski et al., 2018; Li et al., 2016; Rubinstein et al., 2009; Suciu et al., 2018; Xiao et al., 2015; Fang et al., 2020), while the latter manipulates communication messages by sending arbitrary updates to the server. In this paper, we mainly aim to defend model update poisoning attacks and achieve Byzantine robustness (Lamport et al., 2019). An extension to a data poisoning attack is also considered in numerical experiments. Previous research has focused on Byzantine-robust SGD variants where the server uses robust aggregation rules to mitigate the attack of Byzantine clients (Chen et al., 2017; Yin et al., 2018; Xie et al., 2018; Blanchard et al., 2017). Beyond that, Li et al. (2019) considered robustifying the objective function via the L^p-norm regularizer. Additionally, Li et al. (2021b) incorporated personalization and robust aggregation rules to achieve robustness and fairness simultaneously. Our work leverages the ideas from Li et al. (2019) and Li et al. (2021b) but differs from them by embedding the update process in a low-dimensional fixed random subspace. Theoretical analysis shows that with commonly used regularization parameters, our method is no worse than two SOTA methods (Dinh et al., 2020; Li et al., 2021b) (see Figure 1). Extensive experiments manifest our method of achieving state-of-the-art performance under various types and intensities of adversarial attacks.

2.4 Fairness in Federated Learning

According to Zhou et al. (2021), there are three types of fairness in FL: performance fairness, collaboration fairness, and model fairness. With respect to performance fairness, an FL system usually promotes uniform accuracy distribution across participants, which is closely related to resource allocation by viewing FL as a collaborative optimization system over a heterogeneous network.

Li et al. (2021b) provided a formal definition (Definition 1) and some efficient methods have been proposed towards this goal (Li et al., 2021b; Huang et al., 2020).

Definition 1 (Performance fairness, Li et al., 2021b) *A model $\mathbf{w}_1$ is more fair than $\mathbf{w}_2$ if the test performance distribution of $\mathbf{w}_1$ across the network with N clients is more uniform than that of $\mathbf{w}_2$, i.e.* $\operatorname{var}\{F_k(\mathbf{w}_1)\}_{k\in[N]} < \operatorname{var}\{F_k(\mathbf{w}_2)\}_{k\in[N]}$, *where $F_k(\cdot)$ denotes the test loss of client $k \in [N]$ and* var *denotes the variance.*[1]

On collaboration fairness, one expects to build a sound incentive mechanism, and hence an intuition is that each participant would receive a reward that fairly reflects its contribution to the FL system. Lyu et al. (2020) formalized this idea by giving its definition; Yu et al. (2020a) and Xu and Lyu (2021) explored this aspect by proposing methods to this end. Finally, regarding model fairness, one usually concerns ethical issues and seeks to protect some sensitive attributes (Dwork et al., 2012; Hardt et al., 2016; Kusner et al., 2017). Liang et al. (2020) suggested learning a fair representation for each client to achieve fairness, Du et al. (2021) proposed reweighing the objective functions under fairness constraint. In our

[1] Equivalently, we can use the standard deviation (std) to measure fairness across the network.

work, we focus on performance fairness, illustrating the benefits of our method through theoretical analysis and numerical experiments. An extension to collaboration fairness is shown in Appendix C.7.3.

Since the submission of this work, several related studies have emerged. Regarding personalized federated learning, Ye et al. (2023b) introduced a collaboration graph to help the clients adapt to diverse data heterogeneity levels and model poisoning attacks. Yan et al. (2024) further explored to find the optimal cooperation network for each client. On the other hand, regarding communication efficiency, robustness, and fairness, several follow-up work follows our framework. Wang et al. (2023) proposed a maximum entropy-based model to concurrently enhance both global model performance and fairness. Zhao et al. (2024) introduced a two-server aggregation scheme and sparse matrix projection compression technique to enhance communication efficiency and resist poisoning attacks. Zhu et al. (2024) employed the Moreau envelope as the regularization function and reparametrized the objective function so that it could be solved within the ADMM framework, this model could account for robustness and fairness.

3. Methodology

In this section, we present our method which is based on *infimal convolution* and *subspace regularization*. Conventional FL that trains a single global model to fit the "average client" suffers from statistical heterogeneity among numerous devices. To enhance accuracy performance, we hope not only to leverage the global model but also to stylize it to fit the local data for each client. To this end, we employ *infimal convolution*, which is originally proposed to smooth some extended real-valued convex function f with a sufficiently smooth kernel function g (Moreau, 1965). We apply this technique in FL to bridge local models and the global model. Here, f is the usual objective function in the vanilla case, and g is designed to characterize the relationship between local and global models. Given a general function g as the smoothing kernel, the personalized FL using infimal convolution is then formulated as a bi-level problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) := G\left\{F_1(\mathbf{w}),\dots,F_N(\mathbf{w})\right\}, \tag{1}$$

where $G(\cdot)$ is the aggregation function on the server side.[2] For $k \in \{1,\cdots,N\}$,

$$F_k(\mathbf{w}) = \left\{f_k \otimes \lambda g\right\}(\mathbf{w}) := \min_{\mathbf{x}_k\in\mathbb{R}^d} f_k(\mathbf{x}_k) + \lambda g(\mathbf{w}-\mathbf{x}_k)$$
$$\text{with } f_k(\mathbf{x}_k) = \mathbb{E}_{\xi_k}\left[\tilde{f}_k(\mathbf{x}_k;\xi_k)\right].$$

Here, $\otimes$ denotes the infimal convolution operator, ξ_k is an independent sample drawn from the distribution $\mathcal{D}_k$, and $\tilde{f}_k\left(\mathbf{x}_k;\xi_k\right)$ is the loss function corresponding to this sample. $\mathbf{w}$ and $\mathbf{x}_k$ represent the global and local model parameters, respectively. λ is a hyperparameter controlling the degree of personalization. Problem (1) is pure local training if $\lambda = 0$, and is synchronized training when $\lambda \to \infty$.

The smoothing kernel function g is task-specific. Many previous personalized methods can be cast into our infimal convolution framework by setting a proper g. For instance,

[2] For simplicity, we set $G(\cdot)$ as the simple average $\frac{1}{N}\sum_{k=1}^{N} F_k(\mathbf{w})$, but it can also generalize to other forms.

Algorithm 1 `lp-proj`: Projection-based L^p Regularized Personalized Federated Learning

```
Input: Communication rounds $T$, local update rounds $R$, client sampling size $S$, regularization coefficient $\lambda$, lower-level problem accuracy $\nu$, step size $\eta$, initial global model $\tilde{\mathbf{w}}_0 \in \mathbb{R}^{d_{\text{sub}}}$, projection matrix $\boldsymbol{P}$, speedup control parameter $\beta$.
for $t = 0$ to $T-1$ do
  Server sends $\tilde{\mathbf{w}}_t \in \mathbb{R}^{d_{\text{sub}}}$ to all clients.
  for all $k = 1$ to $N$ clients do
    $\tilde{\mathbf{w}}_{k,0}^t = \tilde{\mathbf{w}}_t \in \mathbb{R}^{d_{\text{sub}}}$.
    for $r = 0$ to $R-1$ do
      Independently sample a fresh mini-batch $\tilde{\mathcal{D}}_k$ and minimize the loss function (3) up to accuracy level $\nu$ to get $\mathbf{x}_{k,r}^t \in \mathbb{R}^d$.
      Update the local model $\tilde{\mathbf{w}}_{k,r+1}^t \in \mathbb{R}^{d_{\text{sub}}}$ by (4).
    end for
  end for
  Server uniformly samples a subset of clients $\mathcal{S}_t$ of size $S$. Each client sends $\tilde{\mathbf{w}}_{k,R}^t$ $\in \mathbb{R}^{d_{\text{sub}}}$ to the server.
  Server updates the global model via $\tilde{\mathbf{w}}_{t+1} = (1-\beta)\tilde{\mathbf{w}}_t + \beta \sum_{k \in \mathcal{S}_t} \frac{\tilde{\mathbf{w}}_{k,R}^t}{S}$.
end for
```

Dinh et al. (2020) and Li et al. (2021b) used Moreau Envelopes as the regularizer, which is equivalent to setting $g(\cdot) = \frac{1}{2}\|\cdot\|_2^2$. Li et al. (2019) proposed the L^p-norm regularization $g(\cdot) = \|\cdot\|_p$ instead. Motivated by the fact that high-dimensional data typically has low-dimensional representation that retains meaningful properties (Van Der Maaten et al., 2009), and random projection would preserve the similarity of data vectors (Bingham and Mannila, 2001), we propose to regularize the projection of local models in a shared low-dimensional space, which is equivalent to the following smoothing kernel

$$g(\cdot) = \frac{1}{p}\left\|\boldsymbol{P}(\cdot)\right\|_p^p, \tag{2}$$

where $p \geq 1$ and $\boldsymbol{P}$ are a $d_{\text{sub}} \times d$ random matrix that is generated initially and will not vary anymore. d_{sub} is the dimension of the shared-and-fixed random subspace. The choice for $\boldsymbol{P}$ is flexible as the only requirement in our theory is that all the singular values of $\boldsymbol{P}$ are bounded from both sides. In this paper, we consider that $\boldsymbol{P}$ is generated with i.i.d. Gaussian entries and then normalized to have unit L^2 norm for each row as suggested by Li et al. (2018). We comment that with this g, $F_k(\mathbf{w})$ is actually a function of $\tilde{\mathbf{w}} = \boldsymbol{P}\mathbf{w}$ since $\boldsymbol{P}(\mathbf{w} - \mathbf{x}_k) = \tilde{\mathbf{w}} - \boldsymbol{P}\mathbf{x}_k$. It implies we can only focus on the low-dimensional parameter $\tilde{\mathbf{w}} \in \mathbb{R}^{d_{\text{sub}}}$ at the global level for algorithm description and theoretical analysis.[3]

3.1 The Algorithm

In this subsection, we introduce the algorithm `lp-proj` (see Algorithm 1) for the bi-level optimization problem (1) with smoothing kernel g given by Equation (2).

The algorithm `lp-proj` is essentially an alternative minimization method on bi-level optimization. Each client k maintains two parameters: their local parameter $\mathbf{x}_{k,r}^t$ and a

[3] Without ambiguity, we term $\tilde{\mathbf{w}}$ as the global model.

copy of the global parameter $\tilde{\mathbf{w}}_{k,r}^t$ with additional subscript r denoting inner iterations and superscript t the communication round. At round t, the server broadcasts the latest global model $\tilde{\mathbf{w}}_t$ to all clients. Then each client initializes their version of global model $\tilde{\mathbf{w}}_{k,0}^t$ as $\tilde{\mathbf{w}}_t$ (line 5) and starts to solve the problem via alternative minimization (lines 6–9).

- (line 7) Given a local version of global model $\tilde{\mathbf{w}}_{k,r}^t$, we use gradient descent (GD) to obtain an approximate solution $\mathbf{x}_{k,r}^t$ that minimizes $\tilde{h}_k$ up to accuracy level ν, where

$$\tilde{h}_k(\mathbf{x}_k; \tilde{\mathcal{D}}_k, \tilde{\mathbf{w}}_{k,r}^t) = \frac{1}{|\tilde{\mathcal{D}}_k|} \sum_{\xi_{k,i} \in \tilde{\mathcal{D}}_k} \tilde{f}_k(\mathbf{x}_k; \xi_{k,i}) + \lambda \frac{1}{p} \left\| \tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_k \right\|_p^p . \tag{3}$$

  Here $\tilde{\mathcal{D}}_k$ is a mini-batch sampled uniformly and $\xi_{k,i}$ refers to a sample from $\tilde{\mathcal{D}}_k$. The GD iteration is terminated when $\left\| \nabla \tilde{h}_k(\mathbf{x}_{k,r}^t; \tilde{\mathcal{D}}_k, \tilde{\mathbf{w}}_{k,r}^t) \right\|_2^2 \leq \nu$ is satisfied.

- (line 8) Given a local parameter $\mathbf{x}_{k,r}^t$, the local version of global model $\tilde{\mathbf{w}}_{k,r}^t$ is updated by one-step gradient descent:

$$\tilde{\mathbf{w}}_{k,r+1}^t = \tilde{\mathbf{w}}_{k,r}^t - \frac{\eta\lambda}{p} \partial_{\tilde{\mathbf{w}}_{k,r}^t} \left\| \tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_{k,r}^t \right\|_p^p . \tag{4}$$

After R steps of the alternative updates, each client has its own version of the global model $\tilde{\mathbf{w}}_{k,R}^t$. Then the server accesses a random set of S clients and produces the next global model by a linear combination of the latest $\tilde{\mathbf{w}}_t$ and the average of $\{\tilde{\mathbf{w}}_{k,R}^t\}_{k \in \mathcal{S}_t}$. Here, a hyperparameter β, which could be viewed as a global step size, is introduced to control the global update process. Our theorem shows a proper β can speed up convergence, but in practice, we find that the test performance only varies moderately for different choices of β. For simplicity, we only consider $\beta = 1$ in our experiments.

3.2 Multiple Benefits of the Algorithm

In this subsection, we analyze the benefits of our algorithm in terms of communication efficiency, robustness, and fairness.

3.2.1 Communication Efficiency

In Algorithm 1, we restrict the global model $\tilde{\mathbf{w}}_t$ to lie in a fixed low-dimensional subspace, in which way only $\tilde{\mathbf{w}}_{k,R}^t$ of dimension d_{sub}, rather than the full model $\mathbf{x}_{k,r}^t$ of dimension d, is transmitted to the server each round. The above nature leads to much fewer bits for communication compared to vanilla FL. Besides, we remark on the difference between our method and other existing projection/sketching-based methods. On the one hand, distributed sketching (Bartan and Pilanci, 2020), which directly projects the data in a low-dimensional space at the start of training, is "one-shot" rather than iterative, while our method projects local models every communication round, and the local training proceeds with the full model. On the other hand, sketched-SGD (Ivkin et al., 2019) compresses the transmitted messages with different bases every time, while our random subspace is predetermined at the beginning and would not change after that.

3.2.2 Robustness and Fairness

For one thing, by applying projection into a low-dimensional subspace, our method only requires (near) consensus of model parameters of different clients in the low-dimensional subspace, leaving flexibility for the system towards personalization and better generalization to the local data distribution, which could improve performance fairness and robustness when facing adversarial attacks. For the other, by rewriting the objective as a constrained optimization problem, introducing a L^p-norm regularizer is equivalent to launching an uncertainty set to the model parameter (e.g., L^1-norm is the diamond-shaped uncertainty and L^2-norm is the spherical uncertainty), in which way we can enhance accuracy by searching for a model adaptive to the local data distribution in the uncertainty set. Formal analysis on a class of linear problems is provided in Section 4.2.

4. Theoretical Analysis

In this section, we provide theoretical analysis for `lp-proj` for the case $p = 2$. We first prove the convergence of the proposed algorithm in Section 4.1, covering both the strongly convex case and the non-convex but smooth case. Next, the robustness and fairness properties are investigated on a class of linear problems in Section 4.2.

4.1 Convergence Analysis

We first give the definition of strong convexity and smoothness.

Definition 2 *f_k is said to be μ-strongly convex, if for any $\mathbf{x}_k, \mathbf{x}'_k \in \mathbb{R}^d$, we have $f_k(\mathbf{x}'_k) \geq f_k(\mathbf{x}_k) + \langle \nabla f_k(\mathbf{x}_k), \mathbf{x}'_k - \mathbf{x}_k \rangle + \frac{\mu}{2} \|\mathbf{x}'_k - \mathbf{x}_k\|_2^2$. f_k is said to be L-smooth, if for any $\mathbf{x}_k, \mathbf{x}'_k \in \mathbb{R}^d$, we have $\|\nabla f_k(\mathbf{x}'_k) - \nabla f_k(\mathbf{x}_k)\|_2 \leq L \|\mathbf{x}'_k - \mathbf{x}_k\|_2$.*

4.1.1 Convergence for the Strongly Convex Case

The analysis for the strongly convex case is based on the following assumptions.

Assumption 1 (Convexity) *For a fixed ξ_k, $\tilde{f}_k(\cdot;\xi_k)$ is μ-strongly convex. As a result, f_k is also μ-strongly convex.*

Assumption 2 (Bounded variance) *The variance of stochastic gradients in each client is bounded, i.e., $\mathbb{E}_{\xi_k} \left\| \nabla \tilde{f}_k(\mathbf{x}_k;\xi_k) - \nabla f_k(\mathbf{x}_k) \right\|_2^2 \leq \gamma_f^2$.*

Assumptions 1 and 2 are standard for convergence analysis. Since we usually use weight decay in the training process, when the model is convex, e.g., logistic regression or linear neural network, Assumption 1 naturally holds.

Assumption 3 (Bounded singular values of the projection matrix) *The smallest and the largest singular values of the random matrix $\boldsymbol{P}$, denoted as $s_{\min}(\boldsymbol{P})$ and $s_{\max}(\boldsymbol{P})$, are bounded, i.e.,*

$$1 - C\sqrt{d_{\text{sub}}/d} \leq s_{\min}(\boldsymbol{P}) \leq s_{\max}(\boldsymbol{P}) \leq 1 + C\sqrt{d_{\text{sub}}/d}, \tag{5}$$

where $C, c > 0$ are constants.

Assumption 3 holds with high probability if the rows of the random matrix are independent, sub-gaussian, and isotropic with standardized sub-gaussian norms almost surely (see Vershynin, 2012, Theorem 5.58), which are mild conditions for random matrices. Specifically, for our generation of $\boldsymbol{P}$, Assumption 3 is applicable with probability at least $1-2\exp(-cd_{\text{sub}})$. For a detailed proof, see Proposition 24 in the appendix.

To establish the convergence, we first rewrite the local update as

$$\tilde{\mathbf{w}}^t_{k.r+1} = \tilde{\mathbf{w}}^t_{k,r} - \eta \underbrace{\lambda(\tilde{\mathbf{w}}^t_{k,r} - \boldsymbol{P}\mathbf{x}^t_{k,r})}_{=:\mathbf{g}^t_{k,r}}, \tag{6}$$

which implies $\eta\sum_{r=0}^{R-1}\mathbf{g}^t_{k,r} = \sum_{r=0}^{R-1}(\tilde{\mathbf{w}}^t_{k,r} - \tilde{\mathbf{w}}^t_{k,r+1}) = \tilde{\mathbf{w}}^t_{k,0} - \tilde{\mathbf{w}}^t_{k,R} = \tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}^t_{k,R}$. Then $\mathbf{g}^t_{k,r}$ can be considered as a biased estimate of $\nabla F_k(\tilde{\mathbf{w}}^t_{k,r})$ and the global update rule becomes

$$\begin{aligned}\tilde{\mathbf{w}}_{t+1} &= (1-\beta)\tilde{\mathbf{w}}_t + \frac{\beta}{S}\sum_{k\in\mathcal{S}_t}\tilde{\mathbf{w}}^t_{k,R} = \tilde{\mathbf{w}}_t - \frac{\beta}{S}\sum_{k\in\mathcal{S}_t}(\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}^t_{k,R}) \\ &= \tilde{\mathbf{w}}_t - \underbrace{\eta\beta R}_{=:\tilde{\eta}}\underbrace{\frac{1}{SR}\sum_{k\in\mathcal{S}^t}\sum_{r=0}^{R-1}\mathbf{g}^t_{k,r}}_{=:\mathbf{g}_t},\end{aligned} \tag{7}$$

where $\tilde{\eta}$ and $\mathbf{g}_t$ can be interpreted as the step size and the approximate stochastic gradient of the global update, respectively.

Recall that we can view F_k as a function of $\tilde{\mathbf{w}}$ instead of $\mathbf{w}$, with some abuse of notation, we write F_k as $F_k(\tilde{\mathbf{w}}) = \min_{\mathbf{x}_k\in\mathbb{R}^d}\left\{f_k(\mathbf{x}_k) + \frac{\lambda}{2}\|\tilde{\mathbf{w}} - \boldsymbol{P}\mathbf{x}_k\|_2^2\right\}$. We first establish the convexity and smoothness of F_k and give the expression of $\nabla F_k(\tilde{\mathbf{w}})$.

Proposition 3 *Suppose that Assumptions 1 and 3 hold and let $s = C\sqrt{d_{\text{sub}}/d}$. We have F_k is μ_F-strongly convex and L_F-smooth with $\mu_F = \frac{\lambda\mu}{(1+s)^2\lambda+\mu}$ and $L_F = \lambda$. Moreover, $\nabla F_k(\tilde{\mathbf{w}}) = \lambda(\tilde{\mathbf{w}} - \boldsymbol{P}\hat{\mathbf{x}}_k)$ with $\hat{\mathbf{x}}_k = \operatorname{argmin}_{\mathbf{x}_k\in\mathbb{R}^d}\left\{f_k(\mathbf{x}_k) + \frac{\lambda}{2}\|\tilde{\mathbf{w}} - \boldsymbol{P}\mathbf{x}_k\|_2^2\right\}$. If we further assume f_k is L-smooth and $s < 1$, we can obtain a smaller smoothness parameter $L_F = \frac{\lambda L}{(1-s)^2\lambda+L}$.*

The next lemma quantifies the error between the exact gradient $\nabla F_k(\tilde{\mathbf{w}}^t_{k,r})$ and the approximate gradient $\mathbf{g}^t_{k,r}$ due to mini-batch sampling and optimization error of the inner loop.

Lemma 4 *Suppose that Assumptions 1, 2 and 3 hold and let $s = C\sqrt{d_{\text{sub}}/d}$. We have $\frac{1}{\lambda^2}\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}^t_{k,r}) - \mathbf{g}^t_{k,r}\right\|_2^2\right] \le \delta_1^2 := \frac{2(1+s)^2}{\mu^2}\left(\frac{\gamma_f^2}{|\tilde{\mathcal{D}}_k|} + \nu\right)$.*

From the expression of δ_1^2, we can see that this error has a mild dependence on d_{sub}. A numerical verification of the mild dependence is shown in Appendix C.7.5.

Moreover, defining the optimal point as $\tilde{\mathbf{w}}^* = \operatorname{argmin}_{\tilde{\mathbf{w}}\in\mathbb{R}^{d_{\text{sub}}}} F(\tilde{\mathbf{w}})$, we can also derive bounded diversity of F_k.

Lemma 5 *Suppose that Assumptions 1 and 3 hold. With* $\sigma_{F,1}^2 := \frac{1}{N}\sum_{k=1}^N \|\nabla F_k(\tilde{\mathbf{w}}^*)\|_2^2$ *and* $L_F = \lambda$*, we have*

$$\frac{1}{N}\sum_{k=1}^N \|\nabla F_k(\tilde{\mathbf{w}}) - \nabla F(\tilde{\mathbf{w}})\|_2^2 \leq 4L_F(F(\tilde{\mathbf{w}}) - F(\tilde{\mathbf{w}}^*)) + 2\sigma_{F,1}^2.$$

Then we focus on the outer loop. Lemma 6 gives the one-step descent of the global update.

Lemma 6 (Dinh et al. 2020, Lemma 3, one-step global update) *Suppose that* F_k *is* L_F*-smooth and* μ_F*-strongly convex. Then we have*

$$\mathbb{E}\left[\|\tilde{\mathbf{w}}_{t+1} - \tilde{\mathbf{w}}^*\|_2^2\right] \leq \left(1 - \frac{\tilde{\eta}\mu_F}{2}\right)\mathbb{E}\left[\|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}^*\|_2^2\right] - \tilde{\eta}(2 - 6L_F\tilde{\eta})\mathbb{E}\left[F(\tilde{\mathbf{w}}_t) - F(\tilde{\mathbf{w}}^*)\right]$$
$$+ \frac{\tilde{\eta}(3\tilde{\eta} + 2/\mu_F)}{NR}\sum_{k=1}^N\sum_{r=0}^{R-1}\mathbb{E}\left[\|\mathbf{g}_{k,r} - \nabla F_k(\tilde{\mathbf{w}}_t)\|_2^2\right] + 3\tilde{\eta}^2\mathbb{E}\left[\left\|\frac{1}{S}\sum_{k\in\mathcal{S}_t}\nabla F_k(\tilde{\mathbf{w}}_t) - \nabla F(\tilde{\mathbf{w}}_t)\right\|_2^2\right]. \tag{8}$$

The third term on the right-hand side of (8) is the client drift error due to multiple local updates and approximation error and can be bounded by Lemma 7. The last term is the diversity of F_k w.r.t. client sampling and can be bounded by Lemma 8.

Lemma 7 (Bounded client drift error) *Suppose that Assumptions 1, 2 and 3 hold. If* $\tilde{\eta} \leq \frac{\beta}{5L_F}$*,* $L_F = \lambda$*, and* δ_1^2 *is defined in Lemma 4, then we have*

$$\frac{1}{NR}\sum_{k=1}^N\sum_{r=0}^{R-1}\mathbb{E}\left[\left\|\mathbf{g}_{k,r}^t - \nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right] \leq 2\lambda^2\delta_1^2 + \frac{4L_F^2\tilde{\eta}^2}{\beta^2}\left(\frac{7}{N}\sum_{k=1}^N\mathbb{E}\left[\|\nabla F_k(\tilde{\mathbf{w}}_t)\|_2^2\right] + 10\lambda^2\delta_1^2\right).$$

Lemma 8 (Dinh et al. 2020, Lemma 4) *The diversity of* F_k *w.r.t. client sampling is bounded as follows:*

$$\mathbb{E}_{\mathcal{S}_t}\left\|\frac{1}{S}\sum_{k\in\mathcal{S}_t}\nabla F_k(\tilde{\mathbf{w}}_t) - \nabla F(\tilde{\mathbf{w}}_t)\right\|_2^2 \leq \frac{N/S - 1}{N - 1}\sum_{i=1}^N\frac{1}{N}\|\nabla F_k(\tilde{\mathbf{w}}_t) - \nabla F(\tilde{\mathbf{w}}_t)\|_2^2.$$

With these lemmas at hand, we can obtain the convergence result by rearranging (8) and summing both sides over the index t with appropriate weight. The details and the proof of auxiliary results above are deferred to Appendices A.2 and A.4.

Theorem 9 *Suppose that Assumptions 1, 2 and 3 hold. Let* $\hat{\eta}_1 = \frac{1}{18L_F(1+10\kappa_F/\beta)}$ *with* L_F, μ_F *defined in Proposition 3,* $\kappa_F = L_F/\mu_F$ *and* $\beta \geq 1$ *. If* $T \geq \frac{2}{\hat{\eta}_1\mu_F}$*, then there exists*

$\eta \le \frac{\hat{\eta}_1}{\beta R}$ such that

$$\begin{aligned}\mathbb{E}[F(\bar{\mathbf{w}}_T)-F(\tilde{\mathbf{w}}^*)] \le & \underbrace{\mu_F \Delta_0 e^{-\hat{\eta}_1 \mu_F T/2}}_{\textit{due to initialization}} + \underbrace{\tilde{\mathcal{O}}\left(\frac{(N/S-1)\sigma_{F,1}^2}{\mu_F T N}\right)}_{\textit{due to client sampling}} \\ + & \underbrace{\tilde{\mathcal{O}}\left(\frac{(\sigma_{F,1}^2+\lambda^2\delta_1^2)\kappa_F L_F}{\mu_F^2 \beta^2 T^2}\right) + \mathcal{O}\left(\frac{\lambda^2\delta_1^2}{\mu_F}\right)}_{\textit{client drift with multiple local updates and approximation error}},\end{aligned} \tag{9}$$

where $\Delta_0 = \|\tilde{\mathbf{w}}_0 - \tilde{\mathbf{w}}^*\|_2^2$, $c > 0$ is a constant, δ_1^2 is defined in Lemma 4, $\sigma_{F,1}^2$ is defined in Lemma 5, $\bar{\mathbf{w}}_T = \sum_{t=0}^{T-1} \alpha_t \tilde{\mathbf{w}}_t / A_T$ with $\alpha_t = (1-\eta\beta R\mu_F/2)^{-(t+1)}$ and $A_T = \sum_{t=0}^{T-1} \alpha_t$, the expectation is w.r.t. all the randomness except for $\boldsymbol{P}$, and $\tilde{\mathcal{O}}$ hides constants and polylogarithmic factors. Moreover, suppose that $\mathbf{x}_k^T$ is a solution satisfying $\left\|\nabla \tilde{h}_k(\mathbf{x}_k^T; \tilde{\mathcal{D}}_k, \bar{\mathbf{w}}_T)\right\|_2^2 \le \nu$ and $\mathcal{O}_1$ denotes the right-hand side of (9). Then we have

$$\frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\left[\left\|\boldsymbol{P}\mathbf{x}_k^T - \tilde{\mathbf{w}}^*\right\|_2^2\right] \le \frac{\mathcal{O}_1}{\mu_F} + \mathcal{O}\left(\frac{\sigma_{F,1}^2}{\lambda^2} + \delta_1^2\right). \tag{10}$$

From (9), when there is no client sampling ($S = N$), choosing $\beta = \Theta(NR)$ leads to a quadratic speedup $\tilde{\mathcal{O}}\left(1/(TRN)^2\right)$ w.r.t. communication rounds. (10) shows the average of personalized parameters (after a linear transformation) converges to a ball with center $\tilde{\mathbf{w}}^*$ and radius $\tilde{\mathcal{O}}\left(\frac{\lambda^2\delta_1^2}{\mu_F^2} + \frac{\sigma_{F,1}^2}{\lambda^2} + \delta_1^2\right)$. Here λ can be chosen to trade off different terms.

Our Theorem 9 shares the same error bounds as Dinh et al. (2020, Theorem 1), except that their approximation error δ_1^2 is slightly smaller than ours up to constant factors, since in our case the approximation error is enlarged by projection. The constant term $\mathcal{O}\left(\frac{\lambda^2\delta_1^2}{\mu_F}\right)$ appears in both theorems and is caused by biased gradients, i.e., we only get a biased estimate of ∇F_k due to inexact inner optimization (non-zero ν) and batch data (small $|\tilde{\mathcal{D}}_k|$). Hence, our result is comparable to previous work (Dinh et al., 2020) up to constants factors, even if we force optimization in a random subspace, which facilitates communication efficiency.

4.1.2 Convergence for the Smooth Case

The analysis for the smooth case requires the following additional assumptions.

Assumption 4 (Smoothness) *For a fixed ξ_k, $\tilde{f}_k(\cdot;\xi_k)$ is L-smooth. As a result, f_k is also L-smooth.*

Assumption 5 (Bounded diversity) *The variance of local gradients to global gradient is bounded, i.e., $\frac{1}{N}\sum_{k=1}^{N} \|\nabla f_k(\mathbf{w}) - \nabla f(\mathbf{w})\|_2^2 \le \sigma_f^2$ with $f = \frac{1}{N}\sum_{k=1}^{N} f_k$.*

Assumption 6 (Low-dimensional condition) *$\tilde{f}_k$ satisfies that for any $\mathbf{y}_k \in \mathbb{R}^{d_{\mathrm{sub}}}, \tilde{\mathbf{y}}_k \in \mathbb{R}^{d-d_{\mathrm{sub}}}$, $\tilde{f}_k(\boldsymbol{P}^\top \mathbf{y}_k + \boldsymbol{Q}\tilde{\mathbf{y}}_k;\xi_k) = \tilde{f}_k(\boldsymbol{P}^\top \mathbf{y}_k;\xi_k)$, where $\boldsymbol{Q}$ is chosen such that $(\boldsymbol{P}^\top, \boldsymbol{Q})$ is an* *invertible matrix and $\boldsymbol{PQ}$ is the zero matrix. As a consequence, the same equality also holds with $\tilde{f}_k$ replaced by f_k.*

Assumptions 4 and 5 are also common in convergence analysis. If Assumption 3 holds with $C\sqrt{d_{\text{sub}}/d} < 1$, then $\boldsymbol{P}$ has full row rank, which implies the matrix $\boldsymbol{Q}$ in Assumption 6 exists. This assumption ensures $\min_{\mathbf{x}_k \in \mathbb{R}^d} \tilde{f}_k(\mathbf{x}_k; \xi_k) = \min_{\mathbf{x}_k \in \text{col}(\boldsymbol{P}^\top)} \tilde{f}_k(\mathbf{x}_k; \xi_k)$, where $\text{col}(\boldsymbol{A})$ denotes the subspace spanned by the column vectors of $\boldsymbol{A}$. This means we can focus on the low-dimensional subspace spanned by the row vectors of $\boldsymbol{P}$. We give an example satisfying the assumption. Suppose ξ_k and $\mathbf{x}_k$ have the same dimensions and $\tilde{f}_k(\mathbf{x}_k; \xi_k) = l(\xi_k^\top \mathbf{x}_k)$.[4] If $\xi_k \in \text{col}(\boldsymbol{P}^\top)$, then there exists an $\mathbf{a}_k$ such that $\xi_k = \boldsymbol{P}^\top \mathbf{a}_k$. Decompose $\mathbf{x}_k = \boldsymbol{P}^\top \mathbf{y}_k + \boldsymbol{Q}\tilde{\mathbf{y}}_k$. Then $l(\xi_k^\top \mathbf{x}_k) = l(\mathbf{a}_k \boldsymbol{P}(\boldsymbol{P}^\top \mathbf{y}_k + \boldsymbol{Q}\tilde{\mathbf{y}}_k)) = l(\mathbf{a}_k \boldsymbol{P}^\top \mathbf{y}_k)$. This implies that for linear models with data lying in $\text{col}(\boldsymbol{P}^\top)$, Assumption 6 holds.

For the general case, it is not easy to verify Assumption 6 directly. Intuitively, we can interpret Assumption 6 as that the data concentrate on a low-dimensional subspace. Then with the total parameters denoted by $\mathbf{x}_k = \boldsymbol{P}^\top \mathbf{y}_k + \boldsymbol{Q}\tilde{\mathbf{y}}_k$, only a low-dimensional linear combination $\mathbf{y}_k = (\boldsymbol{P}\boldsymbol{P}^\top)^{-1}\boldsymbol{P}\mathbf{x}_k$ can affect the value of $\tilde{f}_k$.

Still viewing F_k as a function of $\tilde{\mathbf{w}}$, we have the following result that guarantees the smoothness of F_k and gives the form of $\nabla F_k(\tilde{\mathbf{w}})$.

Proposition 10 *Suppose Assumptions 3, 4 and 6 hold with $0 < s = C\sqrt{d_{\text{sub}}/d} < 1/30$ and $\lambda > 4L$. Then F_k is L_F-smooth with $L_F = \lambda$. Moreover, $\nabla F_k(\tilde{\mathbf{w}}) = \lambda(\tilde{\mathbf{w}} - \boldsymbol{P}\boldsymbol{P}^\top \hat{\mathbf{y}}_k)$, where $\hat{\mathbf{y}}_k = \text{argmin}_{\mathbf{y}_k \in \mathbb{R}^{d_{\text{sub}}}} \left\{ f_k(\boldsymbol{P}^\top \mathbf{y}_k) + \frac{\lambda}{2} \left\| \tilde{\mathbf{w}} - \boldsymbol{P}\boldsymbol{P}^\top \mathbf{y}_k \right\|_2^2 \right\}$.*

Similar to Lemma 4, Lemma 11 below characterizes the error between the exact gradient and the approximate gradient. The error also has a mild dependence on d_{sub}.

Lemma 11 *Suppose that Assumptions 2, 3, 4 and 6 hold with $s = C\sqrt{d_{\text{sub}}/d} < 1/30$ and $\lambda > 4L$. We have $\frac{1}{\lambda^2}\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}_{k,r}^t) - \lambda(\tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_{k,r}^t)\right\|_2^2\right] \le \delta_2^2 := \frac{2(1+s)^6\left(\frac{\gamma_f^2}{|\tilde{\mathcal{D}}_k|} + \nu\right)}{[(1-s)^4\lambda - (1+s)^2 L]^2}$.*

With Assumption 5, the diversity of F_k can also be bounded as follows.

Lemma 12 *If Assumptions 3, 4, 5 and 6 hold with $0 < C\sqrt{d_{\text{sub}}/d} < 1/30$ and $\lambda > \sqrt{10}L$ and define $\sigma_{F,2}^2 := \frac{\lambda^2 \sigma_f^2}{\lambda^2 - 10L^2}$. Then we have*

$$\frac{1}{N}\sum_{k=1}^{N} \|\nabla F_k(\tilde{\mathbf{w}}) - \nabla F(\tilde{\mathbf{w}})\|_2^2 \le \frac{10L^2}{\lambda^2 - 10L^2} \|\nabla F(\tilde{\mathbf{w}})\|_2^2 + 3\sigma_{F,2}^2.$$

With the global update rewritten as (7), the one-step descent can be established as

$$\begin{aligned} \mathbb{E}\left[F(\tilde{\mathbf{w}}_{t+1}) - F(\tilde{\mathbf{w}}_t)\right] \le{} & -\frac{\tilde{\eta}(1-3\tilde{\eta}L_F)}{2}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] \\ & + \frac{\tilde{\eta}(1+3\tilde{\eta}L_F)}{2NR}\sum_{r=0}^{R-1}\sum_{k=1}^{N}\mathbb{E}\left[\left\|\mathbf{g}_{k,r}^t - \nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right] + \frac{3\tilde{\eta}^2 L_F}{2}\mathbb{E}\left[\frac{1}{S}\sum_{k \in \mathcal{S}_t} \nabla F_k(\tilde{\mathbf{w}}_t) - \nabla F(\tilde{\mathbf{w}}_t)\right]. \end{aligned} \tag{11}$$

[4] For example, when we fit generalized linear models via the maximum likelihood method, the negative (log) likelihood function has this form.

Similar to the analysis for the strongly convex case, we can give upper bounds of the second and last terms on the right-hand side of (11). Then Theorem 13 follows from rearranging and telescoping. The detailed proof is deferred to Appendix A.3.

Theorem 13 *Suppose that Assumptions 2 to 6 hold and d_{sub}/d is sufficiently small. Let $\Delta_F = F(\tilde{\mathbf{w}}_0) - \min_{\tilde{\mathbf{w}} \in \mathbb{R}^{d_{\text{sub}}}} F(\tilde{\mathbf{w}})$ and $\hat{\eta}_2 = \frac{1}{90\lambda^2 L_F}$ with $\lambda \geq \max\{\sqrt{10L^2+1}, 4L\}$, $L_F = \lambda$ and $\beta \geq 1$. If t^* is uniformly sampled from $\{0, 1, \ldots, T-1\}$, then there exists $\eta \leq \frac{\hat{\eta}_2}{\beta R}$ such that*

$$\begin{aligned}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_{t^*})\|_2^2\right] \leq \; & \underbrace{\mathcal{O}\left(\frac{\Delta_F}{\hat{\eta}_2 T}\right)}_{\text{due to initialization}} + \underbrace{\mathcal{O}\left(\frac{(\Delta_F L_F \sigma_{F,2}^2 (N/S-1))^{1/2}}{\sqrt{TN}}\right)}_{\text{due to client sampling}} \\ & + \underbrace{\mathcal{O}\left(\frac{(\Delta_F L_F)^{2/3}(\sigma_{F,2}^2+\lambda^2\delta_2^2)^{1/3}}{(\beta T)^{2/3}}\right) + \mathcal{O}\left(\lambda^2\delta_2^2\right)}_{\text{client drift with multiple local updates and approximation errors}},\end{aligned} \tag{12}$$

where $c > 0$ is a constant, δ_2^2 defined in Lemma 11, $\sigma_{F,2}^2$ is defined in Lemma 12, the expectation is w.r.t. all the randomness except for $\boldsymbol{P}$, and $\mathcal{O}$ hides constants. Moreover, suppose that $\mathbf{x}_k^t$ is a solution to $\left\|\nabla\tilde{h}_k(\mathbf{x}; \tilde{\mathcal{D}}_k, \tilde{\mathbf{w}}_t)\right\|_2^2 \leq \nu$ and $\mathcal{O}_2$ denotes the right-hand side of (12). Then we have

$$\frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\left[\left\|\boldsymbol{P}\mathbf{x}_k^{t^*} - \tilde{\mathbf{w}}_{t^*}\right\|_2^2\right] \leq \mathcal{O}_2 + \mathcal{O}\left(\frac{\sigma_{F,2}^2}{\lambda^2} + \delta_2^2\right).$$

When there is no client sampling, choosing $\beta = \Theta(NR)$ leads to a sublinear speedup $\mathcal{O}\left(1/(TRN)^{2/3}\right)$. (12) shows the average over indices k and t of the distance between personalized parameters (after a linear transformation) and global model parameters converges to $\mathcal{O}\big(\lambda^2\delta_2^2 + \frac{\sigma_{F,2}^2}{\lambda^2} + \delta_2^2\big)$. Here λ can be chosen to trade off different terms.

Our Theorem 13 also shares similar forms as Dinh et al. (2020, Theorem 2). Both theorems have the constant term $\mathcal{O}\left(\lambda^2\delta_2^2\right)$ which is caused by biased gradients and batch data.

4.1.3 Refined Convergence under Careful Parameter Tuning

In Theorems 9 and 13, $\mathbf{g}_{k,r}^t$ is a biased estimate of ∇F_k, where the biasedness comes from inexact inner optimization (non-zero ν, batch data (small $|\tilde{\mathcal{D}}_k|$), and projection. These factors lead that δ_1 and δ_2 in Lemmas 4 and 11 does not go to zero and the approximation error in the final convergence is enlarged. However, the problem can be fixed by carefully tuning parameters and any given accuracy ϵ can be achieved. To this end, we first suppose that the mini-batch sampling size D_t, lower-level problem accuracy ν_t, and step size η_t all depend on the communication round index t.

When the global model is optimal, we define the optimal personalized parameter as $\mathbf{x}_k^* = \operatorname{argmin}_{\mathbf{x}_k \in \mathbb{R}^d} \left\{f_k(\mathbf{x}_k) + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}}^* - \boldsymbol{P}\mathbf{x}_k\right\|_2^2\right\}$. The following theorem characterizes the $\mathcal{O}(1/T)$ convergence rate of global parameters and personalized parameters for the strongly convex case.

Theorem 14 *Suppose that Assumptions 1 and 2 hold. Let* $\eta_t = \frac{8}{\beta R \mu_F(\zeta+t)}$ *,* $\nu_t = \frac{8}{\mu_F(\zeta+t)}$ *and* $D_t = \left\lceil \frac{\mu_F(\zeta+t)}{D} \right\rceil$*, where* μ_F *is defined in Proposition 3,* $\zeta = 72\kappa_F(1+7\kappa_F/\beta)$ *and* D *is a positive constant. Then we have* $\mathbb{E}\left\|\tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^*\right\|_2^2 \leq \mathcal{O}(1/T)$ *and* $\mathbb{E}\left\|\mathbf{x}_k^T - \mathbf{x}_k^*\right\|_2^2 \leq \mathcal{O}(1/T)$*, where* c *is a positive constant,* $\boldsymbol{x}_k^T$ *is defined in Theorem 9, the expectation is w.r.t. all the randomness except for* $\boldsymbol{P}$*, and* $\mathcal{O}$ *hides constants.*

Moreover, when there is no client sampling, i.e., $S = N$*, let* $\nu_t = \frac{8}{\mu_F \beta^2 (\xi+t)^2}$ *and* $D_t = \left\lceil \frac{\mu_F \beta^2(\xi+t)^2}{D} \right\rceil$ *with* η_t *unchanged. Then we have* $\mathbb{E}\left\|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}^*\right\|_2^2 \leq \mathcal{O}(1/(\beta^2 T^2))$ *and* $\mathbb{E}\left\|\mathbf{x}_k^T - \mathbf{x}_k^*\right\|_2^2 \leq \mathcal{O}(1/(\beta^2 T^2))$*.*

When there is no client sampling, choosing $\beta = \Theta(NR)$ leads to the convergence rate $\mathcal{O}\left(1/(TRN)^2\right)$, which is consistent with the analysis in Theorem 9. Alternatively, since $\nu = O(e^{-\tilde{c}R})$ for some $\tilde{c} > 0$ under the strong convexity assumption, setting $R = \Omega\left(\log(1/\varepsilon)\right)$ and $|\tilde{\mathcal{D}}_k| = \Omega(1/\varepsilon)$ also leads to the target accuracy.

When the objective is possibly non-convex but smooth, we have the following result that guarantees our algorithm can find the approximate stationary point.

Theorem 15 *Suppose that Assumptions 2 to 6 hold. Define* $\Delta_F = F(\tilde{\mathbf{w}}_0) - \min_{\tilde{\mathbf{w}} \in \mathbb{R}^{d_{\text{sub}}}} F(\tilde{\mathbf{w}})$*,* $\alpha_t := \frac{\eta_t}{\sum_{t=0}^{T-1}\eta_t}$ *and sample* t^* *from* $\{0, 1, \ldots, T-1\}$ *with* $\mathbb{P}(t^* = i) = \alpha_t$*. Let* $\eta_t = \frac{1}{90\beta R \lambda^2 L_F \sqrt{t+1}}$*,* $\nu_t = \frac{1}{90\lambda^2 L_F \sqrt{t+1}}$ *and* $D_t = \left\lceil \frac{90\lambda^2 L_F \sqrt{t+1}}{D} \right\rceil$ *where* $\lambda \geq \max\{\sqrt{10L^2+1}, 4L\}$*,* $L_F = \lambda$*,* $\beta \geq 1$ *and* D *is a positive constant. We have* $\mathbb{E}\left[\left\|\nabla F(\tilde{\mathbf{w}}_{t^*})\right\|_2^2\right] = \mathcal{O}(\ln T/\sqrt{T})$*, where* c *is a positive constant, the expectation is w.r.t. all the randomness except for* $\boldsymbol{P}$*, and* $\mathcal{O}$ *hides constants.*

Moreover, when there is no client sampling, i.e., $S = N$*, let* $\eta_t = \frac{1}{90\beta^{1/3} R \lambda^2 L_F (t+1)^{1/3}}$*,* $\nu_t = \frac{1}{90\lambda^2 L_F \beta^{2/3}(t+1)^{2/3}}$ *and* $D_t = \left\lceil \frac{90\lambda^2 L_F \beta^{2/3}(t+1)^{2/3}}{D} \right\rceil$*. Then we have* $\mathbb{E}\left[\left\|\nabla F(\tilde{\mathbf{w}}_{t^*})\right\|_2^2\right] = \mathcal{O}\left(\ln T/(\beta^{2/3}T^{2/3})\right)$*.*

When there is no client sampling, choosing $\beta = \Theta(NR)$ leads to the convergence rate $\tilde{\mathcal{O}}\left(1/(TRN)^{2/3}\right)$, which is also consistent with the analysis below Theorem 13.

The proof of Theorem 14 and 15 is also based on the one-step descent results (8) and (11) with $\tilde{\eta}$ replaced by $\tilde{\eta}_t = \beta R \eta_t$. The details are deferred to Appendices A.5 and A.6.

4.2 Robustness and Fairness

In this subsection, we explore the robustness and fairness benefits of `lp-proj` on a class of linear problems and compare `lp-proj` with `Ditto` (Li et al., 2021b) and `pFedMe` (Dinh et al., 2020). For ease of analysis, we assume the rows of $\boldsymbol{P}$ are orthogonal. In practice, we can directly use the random matrix generated as in Section 3 without explicit orthogonalization, since high-dimensional random vectors are nearly orthogonal. Numerical comparison shows that model accuracy would be similar with or without orthogonalization (see Appendix C.7.4).

Our Setting. We focus on a simplified setting where the number of local update steps is infinite, there is only one round of communication and all clients participate in the communication. Then it is natural to set $\beta = 1$. Suppose that the true parameter

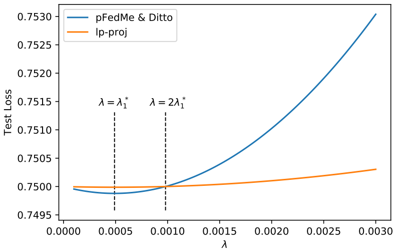
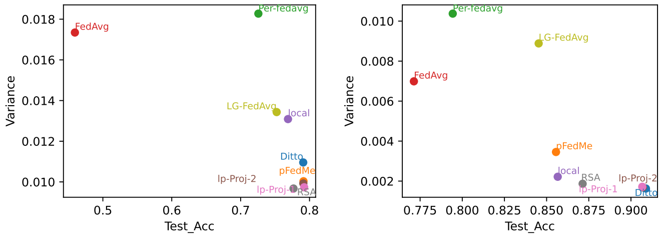


Figure 1: Test losses of `pFedMe`, `Ditto` and `lp-proj` under different values of λ. In our settings, the losses of `pFedMe` and `Ditto` coincide.



(a) `CIFAR` **(b)** `EMNIST`

Figure 2: Accuracy-Fairness trade-off of competing methods. (The point closer to the bottom right corner is better.)

on client k is $\mathbf{w}_k$, there are n samples on each client and the covariate on client k is $\{\xi_{k,i}\}_{i=1}^n$ and fixed. The observations are generated by $y_{k,i} = \xi_{k,i}^\top \mathbf{w}_k + z_{k,i}$ where the noises $z_{k,i} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$. For simplicity, we assume $\sum_{i=1}^n \xi_{k,i}\xi_{k,i}^\top = bn\boldsymbol{I}_d$. Then the test loss on client k is $f_k^{\text{te}}(\mathbf{x}_k) = \frac{1}{2n}\sum_{i=1}^n (\xi_{k,i}^\top \mathbf{w}_k + z'_{k,i} - \xi_{k,i}^\top \mathbf{x}_k)^2$, where $z'_{k,i} \sim \mathcal{N}(0, \sigma^2)$ and are independent of $z_{k,i}$.

Three Attacks. We examine three types of Byzantine attacks. Denote the message delivered by malicious client k as $\tilde{\mathbf{w}}_k^{(ma)}$, then the attacks are listed as follows.

- **Same-value attacks**: The message sent by a Byzantine client k is set as $\tilde{\mathbf{w}}_k^{(ma)} = c\mathbf{1}_{d_{\text{sub}}}$, where $\mathbf{1}_{d_{\text{sub}}} \in \mathbb{R}^{d_{\text{sub}}}$ is the vector of ones and $c \sim \mathcal{N}(0, \tau^2)$.
- **Sign-flipping attacks**: The transmitted messages are sign-flipped and then scaled, i.e., a Byzantine client k computes the true value $\tilde{\mathbf{w}}_k$ but sends $\tilde{\mathbf{w}}_k^{(ma)} = -|c| \cdot \tilde{\mathbf{w}}_k$ to the server where $c \sim \mathcal{N}(0, \tau^2)$.
- **Gaussian attacks**: The message sent by a Byzantine client k is set as $\tilde{\mathbf{w}}_k^{(ma)} \sim \mathcal{N}(\mathbf{0}_{d_{\text{sub}}}, \tau^2\boldsymbol{I}_{d_{\text{sub}}})$.

The analyses for different attacks are similar, thus we only focus on the same-value attacks here for illustration. Results for other attacks are deferred to Appendix B.3. Suppose that the index sets for benign and malicious clients are I_b and I_a respectively with $N_b = |I_b|$ and $N_a = |I_a| = N - N_b$, and the heterogeneity is uniform in all dimensions in the sense of

$$\Sigma_1 := \frac{1}{dN_b}\sum_{k\in\mathbf{I}_b}\left\|\frac{\sum_{i\in I_b}\mathbf{w}_i}{N} - \mathbf{w}_k\right\|_2^2 = \frac{1}{d_{\text{sub}}N_b}\sum_{k\in I_b}\left\|\frac{\sum_{i\in I_b}\mathbf{w}_{i,1}}{N} - \mathbf{w}_{k,1}\right\|_2^2$$

where Σ_1 measures data heterogeneity in a single dimension. Let $\lambda_1^* = \frac{(1-1/N)\sigma^2/n}{\Sigma_1 + \frac{N_a}{N^2}(\tau^2 - \sigma^2/(bn))}$. The numerator of λ_1^* approximately equals the variance of noises divided by the number of samples. The denominator is the sum of one-dimensional data heterogeneity and additional variance due to attacks (usually we have $\tau^2 \gg \sigma^2/n$). When $\lambda = \lambda_1^*$, pFedMe, Ditto and lp-proj all achieve their corresponding minimal losses. However, we do not know factors affecting λ_1^* in advance, implying getting the particular value of λ_1^* is possibly hard. Therefore, we need to compare the performance of these methods under different values of λ.

Proposition 16 *Denote the averaged losses on benign clients of pFedMe, Ditto and lp-proj by $L^{Me,\ att1}(\lambda)$, $L^{Di,att1}(\lambda)$ and $L^{l2,att1}(\lambda)$ respectively. Under the same-value attacks, we have (i) $L^{Me,att1}(\lambda) = L^{Di,att1}(\lambda)\ \forall \lambda > 0$; and (ii) if $\lambda_1^* < b$, $L^{l2,\ att1}(\lambda) \leq L^{Me,att1}(\lambda)$ if and only if $\lambda \geq \frac{2\lambda_1^*}{1-\lambda_1^*/b}$.*

Porposition 16 implies lp-proj outperforms both pFedMe and Ditto once λ is larger than a threshold value $\frac{2\lambda_1^*}{1-\lambda_1^*/b}$, which is slightly larger than $2\lambda_1^*$. The pattern is captured by Figure 1, where we set n=200, N=100, N_a=20, d=100, d_{sub}=10, b=1, σ=1, Σ_1=0.1 and τ=100. Then $\lambda_1^* \approx 4.9 \times 10^{-4}$, a pretty small value. Even for $\lambda < \frac{2\lambda_1^*}{1-\lambda_1^*/b}$, the gap between lp-proj and pFedMe / Ditto is negligible. Thus, lp-proj has comparable or beter performance for any $\lambda > 0$. The formal statement and proof of Proposition 16 are deferred to Appendix B.3.

Remark 17 *To see the relationship between robustness and the dimension d_{sub} of the random projection subspace, we investigate the test loss function to see the role of d_{sub}.*[5] *Take the same-value attack as an example, at the optimal point λ_1^*, $L^*_{l2,att1}$ (viewed as a function of d_{sub}) is linear in d_{sub} and the coefficient of d_{sub} is negative, which implies that as d_{sub} increases, the test loss decreases, and hence the performance of the model would be better.*

On the other hand, for a given λ, the first-order derivative is larger in absolute value when d_{sub} is larger, which means that when the dimension of the projection subspace is smaller, the test loss would have less variation with respect to λ, and hence more robust tuning performance.

Now we turn to the performance fairness defined in Definition 1. For simplicity, we further assume that the true parameters $\mathbf{w}_k$ are i.i.d. from $\mathcal{N}(\mu_w, \mathbf{\Sigma}_w)$.

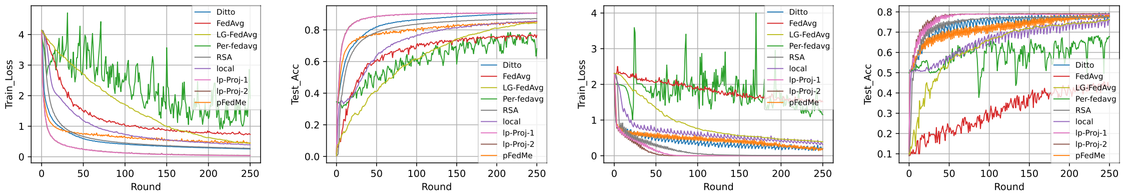
Proposition 18 *Denote the variance of test losses on different clients of pFedMe, Ditto and lp-proj by $V^{Me}(\lambda)$, $V^{Di}(\lambda)$ and $V^{l2}(\lambda)$ respectively. We have for $\forall\lambda > 0$, $\mathbb{E}V^{l2}(\lambda) \leq \mathbb{E}V^{Me}(\lambda) = \mathbb{E}V^{Di}(\lambda)$. More specifically, $\mathbb{E}V^{Me}(\lambda) = O(d^2)$ and $\mathbb{E}V^{l2}(\lambda) = O(d^2_{\text{sub}})$, where the expectation is taken w.r.t. the randomness of the $\mathbf{w}_k$.*

Proposition 18 shows lp-proj always brings more uniform test losses, no matter what value λ is. In particular, $\mathbb{E}V^{\text{l2}}(\lambda) = O(d^2_{\text{sub}})$ while $\mathbb{E}V^{\text{Me}}(\lambda) = O(d^2)$. Since it is likely that $d_{\text{sub}} \ll d$, the advantage of lp-proj could be much larger. It implies lp-proj is more fair than pFedMe and Ditto. For the formal theorem and proof, see Appendix B.4.

5. Numerical Experiments

In this section, we demonstrate lp-proj has the desirable properties through numerical experiments.

[5] To be simple, we only provide intuitive interpretation here, the detailed calculations can be found in Remark 31 in the appendix.



(a) `EMNIST`, Train Loss **(b)** `EMNIST`, Test Accuracy **(c)** `CIFAR`, Train Loss **(d)** `CIFAR`, Test Accuracy

Figure 3: Personalization performance of `lp-proj-1`, `lp-proj-2` with other methods on `EMNIST` and `CIFAR`.

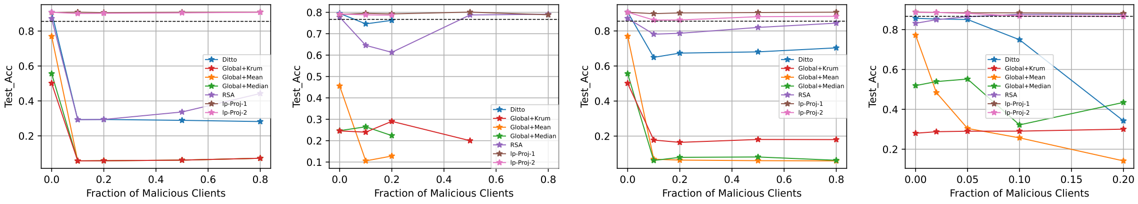
5.1 Experimental Setup

We test `lp-proj` as well as other comparable algorithms on six data sets from common ML and FL benchmarks (Marcel and Rodriguez, 2010; Caldas et al., 2018), including two synthetic data sets, `EMNIST`, `CIFAR`, `MNIST` and `FASHIONMNIST`. We consider both convex and non-convex models, where for the latter, we consider neural networks including both multilayer perceptron (MLP) and convolutional neural network (CNN). To better model the statistical heterogeneity, we distribute the data set among clients in a non-iid fashion such that each client only contains partial classes of the data in multi-classification problems.[6] For each client, the training and testing data are pre-specified as in the ML community, and 20% of training data is randomly extracted to construct a validation set, keeping the remaining 80% as the training set. The training set is used for model fitting and parameter estimation. For each competing method, we use the accuracy performance on the validation set as the tuning criterion and conduct a grid search to choose the best hyper-parameter combination among a prescribed candidate set. All reported results are evaluated on the test data set. More details about hyperparameter tuning are provided in Appendix C.2. Furthermore, to incorporate partial participation (McMahan et al., 2017; Li et al., 2020b), we randomly select 10% of the clients for aggregation at each communication round. The projection dimension of the random subspace for each data set is chosen based on the full model size and communication budget. Source code for the reproduction of numerical results is available at `https://github.com/desternylin/perfed`. For clarity, we only show representative results in the following subsections, whereas the comprehensive numerical results are deferred to Appendix C.

5.2 Personalization Accuracy Performance

In order to highlight the empirical performance of our proposed method, we compare `lp-proj` with several state-of-the-art personalization methods in the literature, i.e., `Ditto` (Li et al., 2021b), `LG-FedAvg` (Liang et al., 2020), `Per-fedavg` (Fallah et al., 2020), `RSA` (Li et al., 2019), and `pFedMe` (Dinh et al., 2020), together with a global method `FedAvg` (McMahan et al., 2017) and a pure local method. Specifically, we consider the case when $p = 1$ (`lp-proj-1`) and $p = 2$ (`lp-proj-2`).

[6] This type of data heterogeneity is termed *label skew* (Tan et al., 2022; Ye et al., 2023a).



(a) `EMNIST`, same-value **(b)** `CIFAR`, sign-flipping **(c)** `EMNIST`, Gaussian **(d)** `Synthetic(0, 0)`, data poisoning

Figure 4: Robustness comparison of different methods, i.e., average test accuracy of benign clients. The dashed black line shows the performance of pure local training. A line with less than 5 points implies the algorithm collapses because the intensity of the given attack exceeds the limit the corresponding algorithm could tolerate.

From Figure 3, we see that `lp-proj-1` and `lp-proj-2` have comparable or even superior performance than other methods. On `EMNIST` and `CIFAR`, the minimum train losses are achieved by `lp-proj-1` and `lp-proj-2`, respectively. In terms of test accuracy, `Ditto` shows the best performance on `EMNIST`, but `lp-proj` is also comparable; while on `CIFAR`, `lp-proj-1` gives the best test accuracy. Furthermore, the training process of `lp-proj` is more stable as the loss and accuracy curves have less fluctuation.

5.3 Communication Efficiency

We compare `lp-proj` with the global baseline `FedAvg` (McMahan et al., 2017) and five standard approaches using gradient and model compression, namely `Sketch` (Ivkin et al., 2019), `LBGM` (Azam et al., 2021), `QSGD` (Alistarh et al., 2017), `DGC` (Lin et al., 2018) and `LG-FedAvg` (Liang et al., 2020). For a fair comparison, we personalize the gradient compression methods, i.e., `Sketch, LBGM, QSGD` and `DGC`, which are not personalization algorithms in the original literature. We use a simple meta-learning framework (Finn et al., 2017; Fallah et al., 2020), which uses the collaboratively trained global model as an initialization and performs gradient updates with respect to the client's own loss function to obtain its personalized model. We quantify the communication cost via total bytes written and read by active clients each round and capture the relation between test accuracy and communicated bytes.

From Table 1, we can see that given a communication budget of bytes, `lp-proj` obtains $\sim 26.3\%$ and $\sim 83.5\%$ test accuracy improvement on `Synthetic(0, 0)` and `EMNIST` data sets respectively. On the other hand, given a target test accuracy, our proposed method needs much fewer bits than the rest and saves the communication cost by `79x` and `1320x` on the two data sets compared with the best-competing method. Besides, our method owns flexibility on the choice of the projection dimension d_{sub}, because the convergence dependence of our method on d_{sub} is mild as predicted by Lemma 11. The compression rate of our proposed methods can be `1000x` or even higher, while that of sketching or gradient compression methods typically is no larger than tens.

	`Synthetic(0, 0)`				`EMNIST`			
Method	Bytes Budget	Test Acc	Target Acc	Used Bytes	Bytes Budget	Test Acc	Target Acc	Used Bytes
FedAvg	328020	0.625	0.6	597800	4236900	⋆	0.7	445851400
Sketch	328020	0.456	0.6	⋆	4236900	⋆	0.7	⋆
lp-proj-1	328020	0.885	0.6	**4620**	4236900	**0.906**	0.7	**174720**
lp-proj-2	328020	**0.888**	0.6	**4620**	4236900	**0.906**	0.7	196560
LBGM	328020	0.815	0.6	12200	4236900	⋆	0.7	206307624
QSGD	328020	0.115	0.6	923350	4236900	⋆	0.7	173663720
DGC	328020	⋆	0.6	372000	4236900	⋆	0.7	⋆
LG-FedAvg	\	\	\	\	4236900	0.071	0.7	230786010

Table 1: Communication performance on `Synthetic(0, 0)` and `EMNIST` data sets. Two aspects are considered: test accuracy on a given byte budget and bytes used to achieve a target test accuracy. A $\star$ on the column "Test Acc" refers to the situation that bytes used in the first iteration of the algorithm have exceeded the budget, and a $\star$ on the column "Used Bytes" means the algorithm could not provide a solution that reaches the target accuracy.

5.4 Robustness

In addition to the three Byzantine attacks introduced in Section 4.2, we consider a stronger data poisoning attack in the following experiments.

- **Data poisoning attacks**: The training samples on malicious clients are poisoned with uniformly randomly chosen noisy labels. Furthermore, when communicating, these clients would scale their transmitted messages to dominate the aggregate update.

For the former three Byzantine attacks, the noise variance τ is set as 100, 10, and 100 respectively. The corruption levels, i.e., the fractions of malicious clients, are set as $\{0.1, 0.2, 0.5, 0.8\}$. For the data poisoning attack, the scaling factor is randomly sampled from $\mathcal{N}(0, 20^2)$, and the corruption levels are from $\{0.02, 0.05, 0.1, 0.2\}$. Under different types of attacks and different levels of corruption, we compare the average test accuracy performance on benign clients of `lp-proj-1` and `lp-proj-2` with various defense baselines, including `Ditto`, `RSA`, and global training augmented with different robust aggregation techniques, such as median and Krum (Blanchard et al., 2017).

From Figure 4, we find that under relatively weak attacks, e.g., same-value and Gaussian attacks, the test accuracy of `lp-proj-1` and `lp-proj-2` rarely decays as the fraction of malicious clients increases, while we observe significant drops on the test accuracy for other algorithms once malicious clients exist. On the other hand, under strong attacks, e.g. sign-flipping and data poisoning, an increasing fraction of malicious clients deteriorates the accuracy performance continuously and even collapses the local model if the attack intensity is too large. For example, under the sign-flipping attack, when the fraction of malicious clients exceeds 20%, only `lp-proj-1`, `RSA` and `Global+Krum` work, while all other methods fail to produce a solution. When the attack intensity further increases to 80%, the only robust methods that achieve the desired accuracy are `lp-proj-1` and `RSA`.

The numerical results show that our method is resistant to standard malicious attacks, which is rooted in the combination of projection and L^1-norm subspace regularization that is attributed to the robustness. Consider an extreme example: if the subspace dimension is chosen as 0, then the joint optimization is reduced to pure local training. No matter how serious the adversarial attack is, the local test performance would not be affected. Therefore, random projection helps alleviate the attacks applied in the original space, while the L^1-norm helps eliminate outliers further (Ke and Kanade, 2005).

5.5 Fairness

To illustrate the accuracy and performance fairness trade-off, we plot the variances of accuracies across the system against the corresponding test accuracies for `lp-proj` and several other different approaches in Figure 2. To examine performance fairness in isolation, the numerical experiments are performed without adversarial attacks in this part. Furthermore, for each competing method, we select the optimal achievable test accuracy after the 20th communication round, and the corresponding variance is picked up.

The results with respect to performance fairness show that `lp-proj-1` and `lp-proj-2` provide accurate and fair solutions that are comparable to other SOTA methods. In particular, on `CIFAR`, `lp-proj-1` achieves the highest test accuracy of 79.22% with the lowest variance of 0.0097 among all the competitors. Although `RSA` achieves the same variance as `lp-proj-1`, its corresponding test accuracy is only 77.68%, which is 1.54% lower than `lp-proj-1`. On the other hand, on `EMNIST`, despite the optimal approach is `Ditto`, with a test accuracy of 90.89% and the corresponding variance of 0.0016, our proposed method shows comparable performance, e.g., `lp-proj-2` achieves a test accuracy of 90.70% with a variance of 0.0016, which is only slightly inferior to the previous method. Theoretical analysis in Proposition 18 implies that in the case of the linear model, the dependence of the variance on the projection dimension is of squared order, indicating that low-dimensional projection helps reduce the variance of test losses among clients. Numerical results suggest that this conclusion may be generalized to broader settings.

6. Large-Scale Application

In Algorithm 1, the introduction of random projection subspace brings us multiple benefits, especially communication efficiency, since the bytes needed for message transmission are greatly reduced. However, when the size of the data set or the implemented model is extremely large, e.g., ImageNet on deep neural networks, the extra cost for the storage of the projection matrix and the computation of matrix multiplication may be a burden for the clients. To address this, we propose a generalization of the vanilla form to facilitate large-scale applications in the real world.

6.1 Block-Diagonal Projection

In Algorithm 1, the space complexity for the storage of the projection matrix is $\mathcal{O}(d_{\text{sub}}d)$, while the computation complexity for projecting the full model parameter $\mathbf{x}_k$ into the random subspace is $\mathcal{O}(d_{\text{sub}}d)$. To save memory and reduce computation complexity, we consider block-diagonal matrix for random projection. Suppose the projection matrix $\boldsymbol{P}$ is of dimension $d_{\text{sub}} \times d$, using a k-fold block diagonalization, we equally divide the matrix into k^2 blocks, each of dimension $\frac{d_{\text{sub}}}{k} \times \frac{d}{k}$. Only the blocks in the diagonal are filled with i.i.d. Gaussian entries which are normalized to have unit L^2 norm on each row, all the

Method	Train Loss	Test Acc	Communication Bytes
`pFedMe`	2.0284 (0.3778)	0.5009 (0.0130)	2.582×10^{11}
`FedAvg`	5.6333 (0.0302)	0.0458 (0.0004)	9.307×10^{10}
`lp-proj-1`	**0.6167** (0.1247)	**0.8403** (0.0071)	1.2×10^{7}
`lp-proj-2`	0.6725 (0.1040)	0.8274 (0.0058)	$\mathbf{1.152 \times 10^{7}}$

Table 2: Performance on `ImageNet` using `ResNet34`.

off-diagonal blocks are zeroes. In this way, the space and time complexity can be reduced simultaneously. On one hand, the improvement in the space complexity is proportional to the square of the number of blocks, i.e., we need $\mathcal{O}(\frac{d_{\text{sub}}d}{k^2})$ space for storage. On the other hand, the improvement in the computation complexity is proportional to the number of blocks, which gives $\mathcal{O}(\frac{d_{\text{sub}}d}{k})$ computation time. Here we only discuss the situation that considers equal division on both dimensions of the projection matrix, which leads to the maximum reduction in time and space complexity. In practice, there is more flexibility as we can comprehensively consider the structure of the implemented model, e.g., when implementing neural networks, we can equip each layer of the network with a projection matrix, in other words, the projection matrix is divided according to the layer of the neural network.

6.2 Numerical Performance on ImageNet

We consider large-scale applications on ImageNet, using ResNet34 (He et al., 2016) as the implemented model, which has over 11 billion parameters. Using a similar fashion of data generation as in Section 5, we still consider the heterogeneous case, where there are a total of 100 clients, and each client is assigned to 50 out of 1000 classes of images. `FedAvg` and `pFedMe` are included as benchmarks in the numerical comparison. For our proposed method, we apply the block-diagonal projection with the number of blocks set as 50. In consideration of limited training time and computing resources, we restrict the maximum number of training rounds to 200. The results are shown in Table 2. From the results, we can see that under data heterogeneity, personalization methods are uniformly better than `FedAvg`. On the other hand, random projection greatly reduces the communication costs for `lp-proj` compared to `pFedMe`. Furthermore, block-diagonalization helps save computation and space complexity, which leaves the great potential of our method in large-scale applications.

7. Conclusion

In this paper, we have proposed a simple yet powerful personalized FL approach based on infimal convolution and subspace projection that we call `lp-proj`. Theoretically, we analyze the convergence of the proposed algorithm for strongly convex and non-convex but smooth objectives with square regularizers. The inherent benefits of robustness and fairness of our method are also illustrated in a class of linear problems. Empirically, we perform a large number of numerical experiments on multiple ML data sets and compare the proposed approach with various SOTA baselines. The results show that our approach could significantly save communication costs, improve robustness under various kinds of adversarial attacks, and promote performance fairness. An extension of the algorithm to reduce space and time complexity together with numerical verification indicates that our algorithm has the potential for large-scale application. In future work, we would be interested in establishing convergence results for general L^p regularizers and considering additional constraints, e.g., differential privacy.

Acknowledgments

Yuze Han, Xiang Li and Zhihua Zhang have been supported by the National Natural Science Foundation of China (No. 12271011 and No. 12350001), and the MOE Project of Key Research Institute of Humanities and Social Sciences (No.22JJD110001).

Appendix A. Convergence of `lp-proj` for $p = 2$

In this section, we provide the complete proof for the results in Section 4.1. The framework is adapted from Dinh et al. (2020), with some concrete results specific to our settings.

A.1 Some Useful Results

In this subsection, we provide some existing results useful for our later analysis. We first introduce more definitions.

Definition 19 (Further definitions) *Suppose that f_k is a function from $\mathbb{R}^d$ to $\mathbb{R}$.*

(a) f_k is said to be convex, if for any $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$ and $0 \leq \alpha \leq 1$, it holds that

$$f_k(\alpha \mathbf{w} + (1-\alpha)\mathbf{w}') \leq \alpha f_k(\mathbf{w}) + (1-\alpha) f_k(\mathbf{w}').$$

If f_k is differentiable, the above condition is equivalent to that for any $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$,

$$f_k(\mathbf{w}') \geq f_k(\mathbf{w}) + \left\langle \nabla f_k(\mathbf{w}), \mathbf{w}' - \mathbf{w} \right\rangle.$$

(b) f_k is said to be μ-strongly convex for some $\mu > 0$, if for any $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$ and $0 \leq \alpha \leq 1$, it holds that

$$f_k(\alpha \mathbf{w} + (1-\alpha)\mathbf{w}') \leq \alpha f_k(\mathbf{w}) + (1-\alpha) f_k(\mathbf{w}') - \frac{\mu \alpha (1-\alpha)}{2} \left\| \mathbf{w} - \mathbf{w}' \right\|_2^2.$$

If f_k is differentiable, the above condition is equivalent to that for any $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$,

$$f_k(\mathbf{w}') \geq f_k(\mathbf{w}) + \left\langle \nabla f_k(\mathbf{w}), \mathbf{w}' - \mathbf{w} \right\rangle + \frac{\mu}{2} \left\| \mathbf{w}' - \mathbf{w} \right\|^2.$$

If f_k is twice differentiable, the above condition is also equivalent to $\nabla^2 f_k \succeq \mu \boldsymbol{I}_d$.

Then we have the following property of strongly convex functions.

Proposition 20 (Nesterov 2018, Theorems 2.1.5 and Theorem 2.1.10) *If F_k is L_F-smooth, then we have that*

$$\frac{1}{2L_F} \left\| \nabla F_k(\mathbf{w}') - \nabla F_k(\mathbf{w}) \right\|_2^2 \leq F_k(\mathbf{w}') - F_k(\mathbf{w}) - \left\langle \nabla F_k(\mathbf{w}), \mathbf{w}' - \mathbf{w} \right\rangle$$

for any $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$*. If* F_k *is* μ_F*-strongly convex, then we have that*

$$\left\|\nabla F_k(\mathbf{w}') - \nabla F_k(\mathbf{w})\right\|_2 \geq \mu_F \left\|\mathbf{w}' - \mathbf{w}\right\|_2$$

for any $\mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$*.*

Proposition 21 provides two useful inequalities, which can be derived from the Cauchy-Schwarz Inequality.

Proposition 21 (Cauchy-Schwarz inequality) *For any* $\mathbf{x}_k \in \mathbb{R}^d$ *and* $c > 0$*,* $k = 1, 2, \ldots, M$*, we have*

$$\left\|\sum_{k=1}^{M} \mathbf{x}_k\right\|_2^2 \leq M \sum_{k=1}^{M} \|\mathbf{x}_k\|_2^2 \quad \textit{and} \quad \|\mathbf{x}_1 + \mathbf{x}_2\|_2^2 \leq (1+c)\|\mathbf{x}_1\|_2^2 + (1+1/c)\|\mathbf{x}_2\|_2^2 .$$

Next, we present the relationships between a function and its conjugate function.

Proposition 22 (Hiriart-Urruty and Lemaréchal, 1993) *Suppose that* f *is a convex function from* $\mathbb{R}^d$ *to* $\mathbb{R} \cup \{+\infty\}$*. Define the conjugate of* f *as* $f^*(\mathbf{u}) = \sup_{\mathbf{x} \in \mathbb{R}^d}\{\langle \mathbf{u}, \mathbf{x}\rangle - f(\mathbf{x})\}$ *and the biconjugate of* f *as* $f^{**} = (f^*)^*$*. The domain of* f *is denoted by* $\operatorname{dom} f = \{\mathbf{x} \in \mathbb{R}^d : f(\mathbf{x}) \in \mathbb{R}\}$*. Suppose* c *is a positive number. Then we have the following results.*

(a) If f *is convex, then* $f^{**} = f$*.*

(b) If f *is* c*-strongly convex, then* $\operatorname{dom} f^* = \mathbb{R}^d$ *and* f^* *is* $1/c$*-smooth.*

(c) If f *is convex and* c*-smooth, then* f^* *is* $1/c$*-strongly convex on every convex subset* $C \subset \operatorname{dom} \partial f^*$*.*

(d) If f *is convex, then* $\mathbf{u} \in \partial f(\mathbf{x}) \iff \mathbf{x} \in \partial f^*(\mathbf{u})$*.*

Proposition 23 guarantees the approximate isometry of a “flat” matrix with independent rows under certain conditions.

Proposition 23 (Vershynin 2012, Theorem 5.58) *Let* $\boldsymbol{A}$ *be an* $d \times D$ *matrix* $(d \leq D)$ *whose rows* $\mathbf{a}_i^\top$ *are independent sub-gaussian isotropic random vectors in* $\mathbb{R}^d$ *with* $\|a_j\|_2 = \sqrt{D}$*. Then for every* $t \geq 0$*, the inequality*

$$\sqrt{D} - C\sqrt{d} - t \leq s_{\min}(\boldsymbol{A}) \leq s_{\max}(\boldsymbol{A}) \leq \sqrt{D} + C\sqrt{d} + t$$

holds with probability at least $1 - 2\exp(-ct^2)$*, where* $s_{\min}(\boldsymbol{A})$ *and* $s_{\max}(\boldsymbol{A})$ *denote the smallest and the largest singular values of* $\boldsymbol{A}$*,* $C = C'_K$*,* $c = c'_K > 0$ *depend only on the subgaussian norm* $K = \max_j \|A_j\|_{\psi_2}$ *of the rows.*

For the definitions of sub-gaussian random vectors and the norm $\|\cdot\|_{\psi_2}$, see Definition 5.7 and 5.22 in Vershynin (2012). A random vector is said to be isotropic if its covariance matrix is the identity matrix.

With Proposition 23, we can prove that our projection matrix $\boldsymbol{P}$ is approximately orthogonal in the sense that all the singular values of $\boldsymbol{P}$ are around 1.

Proposition 24 *With probability at least* $1 - 2\exp(-cd_{\text{sub}})$*, we have* $1 - C\sqrt{d_{\text{sub}}/d} \le s_{\min}(\boldsymbol{P}) \le s_{\max}(\boldsymbol{P}) \le 1 + C\sqrt{d_{\text{sub}}/d}$ *for some* $C, c > 0$*, where* $s_{\min}(\boldsymbol{P})$ *and* $s_{\max}(\boldsymbol{P})$ *denote the smallest and the largest singular values of* $\boldsymbol{P}$*.*

Proof For our choice of $\boldsymbol{P}$, we have $\boldsymbol{P} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{d_{\text{sub}}})^\top$ where the row vectors $\mathbf{a}_i$ are independent and uniformly distributed on the unit sphere of $\mathbb{R}^d$. Example 5.21 in Vershynin (2012) implies that each $\sqrt{d}\,\mathbf{a}_i$ is isotropic. Moreover, by Example 5.25, we have that $\left\|\sqrt{d}\,\mathbf{a}_i\right\|_{\psi_2} = C_0$ for some absolute constant $C_0 > 0$.

Then by Proposition 23, we have that $1 - C\sqrt{d_{\text{sub}}/d} \le s_{\min}(\boldsymbol{P}) \le s_{\max}(\boldsymbol{P}) \le 1 + C\sqrt{d_{\text{sub}}/d}$ with probability at least $1 - 2\exp(-cd_{\text{sub}})$ for some positive constants C and c. ■

For brevity, we let $s = C\sqrt{d_{\text{sub}}/d}$. If $\sqrt{d_{\text{sub}}/d}$ is sufficiently small, we have $s < 1$. Then Proposition 24 implies that (5), i.e.,

$$1 - s \le s_{\min}(\boldsymbol{P}) \le s_{\max}(\boldsymbol{P}) \le 1 + s,\ 0 < s < 1$$

holds with probability at least $1 - 2\exp(-cd_{\text{sub}})$. This implies that $\text{rank}(\boldsymbol{P}) = d_{\text{sub}}$ and the $d_{\text{sub}} \times d_{\text{sub}}$ matrix $\boldsymbol{P}^\top\boldsymbol{P}$ is invertible.

The next proposition is a straightforward consequence of (5).

Proposition 25 *If (5) holds, then we have* $\|\boldsymbol{P}\mathbf{x}\|_2^2 \le (1+s)^2\|\mathbf{x}\|_2^2$ *for any* $\mathbf{x} \in \mathbb{R}^d$*,* $\|\boldsymbol{P}\mathbf{x}\|_2^2 \ge (1-s)^2\|\mathbf{x}\|_2^2$ *for any* $\mathbf{x} \in \text{col}(\boldsymbol{P}^\top)$ *and* $(1-s)^2\|\mathbf{y}\|_2^2 \le \left\|\boldsymbol{P}^\top\mathbf{y}\right\|_2^2 \le (1+s)^2\|\mathbf{y}\|_2^2$ *for any* $\mathbf{y} \in \mathbb{R}^{d_{\text{sub}}}$*. Moreover, if* $f(\cdot)$ *is an* L*-smooth function from* $\mathbb{R}^d$ *to* $\mathbb{R}$*, then* $f(\boldsymbol{P}^\top\cdot)$ *is a* $(1+s)^2L$*-smooth function from* $\mathbb{R}^{d_{\text{sub}}}$ *to* $\mathbb{R}$*.*

Proof From (5), it is easy to verify these properties except for the inequality $\|\boldsymbol{P}\mathbf{x}\|_2^2 \ge (1-s)^2\|\mathbf{x}\|_2^2$ for any $\mathbf{x} \in \text{col}(\boldsymbol{P}^\top)$. Suppose the SVD of $\boldsymbol{P}$ is $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$ where $\boldsymbol{U}$ is a $d_{\text{sub}} \times d_{\text{sub}}$ orthogonal matrix, $\boldsymbol{D}$ is a $d_{\text{sub}} \times d_{\text{sub}}$ diagonal matrix whose digonal elements are between $1-s$ and $1+s$, and $\boldsymbol{V}$ is a $d \times d_{\text{sub}}$ matrix with orthogonal column vectors. For $\mathbf{x} = \boldsymbol{P}^\top\mathbf{y}$, we have

$$\|\boldsymbol{P}\mathbf{x}\|_2^2 = \left\|\boldsymbol{P}\boldsymbol{P}^\top\mathbf{y}\right\|_2^2 = \mathbf{y}^\top\boldsymbol{P}\boldsymbol{P}^\top\boldsymbol{P}\boldsymbol{P}^\top\mathbf{y} = \mathbf{y}^\top\boldsymbol{P}\boldsymbol{V}\boldsymbol{D}^2\boldsymbol{V}^\top\boldsymbol{P}^\top\mathbf{y}.$$

If $\mathbf{y} \ne \mathbf{0}_{d_{\text{sub}}}$, $\boldsymbol{V}^\top\boldsymbol{P}^\top\mathbf{y} = \boldsymbol{D}\boldsymbol{U}^\top\mathbf{y} \ne \mathbf{0}_{d_{\text{sub}}}$. Since $\boldsymbol{D}^2 \succeq (1-s)^2\boldsymbol{I}_{d_{\text{sub}}}$. It follows that

$$\begin{aligned}\|\boldsymbol{P}\mathbf{x}\|_2^2 &\ge (1-s)^2\mathbf{y}^\top\boldsymbol{P}\boldsymbol{V}\boldsymbol{V}^\top\boldsymbol{P}^\top\mathbf{y} = (1-s)^2\mathbf{y}^\top\boldsymbol{U}\boldsymbol{D}\boldsymbol{D}\boldsymbol{U}^\top\mathbf{y}\\ &= (1-s)^2\mathbf{y}^\top\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{D}\boldsymbol{U}^\top\mathbf{y} = (1-s)^2\left\|\boldsymbol{P}^\top\mathbf{y}\right\|_2^2 = (1-s)^2\|\mathbf{x}\|_2^2.\end{aligned}$$

This completes the proof. ■

A.2 Proof of Theorem 9

In this subsection, we give the formal proof of Theorem 9.

Proof [Proof of Theorem 9] Recall that (5) holds with probability at least $1-2\exp(-cd_{\text{sub}})$. For convenience, we assume this inequality holds throughout the proof. Then we assume $\eta \leq \frac{\hat{\eta}_1}{\beta R}$. The exact value of η will be determined later. By Lemma 8 and Lemma 5, we have

$$
\begin{aligned}
\mathbb{E}\left[\left\|\frac{1}{S}\sum_{k\in\mathcal{S}_t}\nabla F_k(\tilde{\mathbf{w}}_t)-\nabla F(\tilde{\mathbf{w}}_t)\right\|_2^2\right] &\leq \frac{N/S-1}{N-1}\sum_{k=1}^{N}\frac{1}{N}\mathbb{E}\left[\|\nabla F_k(\tilde{\mathbf{w}}_t)-\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] \\
&\leq \frac{N/S-1}{N-1}\left(4L_F\mathbb{E}\left[F(\tilde{\mathbf{w}}_t)-F(\tilde{\mathbf{w}}^*)\right]+2\sigma_{F,1}^2\right). \qquad (13)
\end{aligned}
$$

Recall that $\hat{\eta}_1 = \frac{1}{18L_F(1+10\kappa_F/\beta)}$, $\beta \geq 1$ and $\kappa_F = L_F/\mu_F \geq 1$. $\eta \leq \frac{\hat{\eta}_1}{\beta R}$ implies that $\tilde{\eta} = \beta R\eta \leq \hat{\eta}_1 \leq \min\left\{\frac{2}{\mu_F}, \frac{\beta}{5L_F}\right\}$. Then we have $3\tilde{\eta}+2/\mu_F \leq 8/\mu_F$. By Lemma 7, we obtain

$$
\begin{aligned}
&\frac{\tilde{\eta}(3\tilde{\eta}+2/\mu_F)}{NR}\sum_{k=1}^{N}\sum_{r=0}^{R-1}\mathbb{E}\left[\left\|\mathbf{g}_{k,r}^t-\nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right] \\
&\leq \tilde{\eta}\frac{16\lambda^2\delta_1^2}{\mu_F}+\frac{\tilde{\eta}^3}{\beta^2}\frac{32L_F^2}{\mu_F}\sum_{k=1}^{N}\frac{1}{N}\left(7\mathbb{E}\left[\|\nabla F_k(\tilde{\mathbf{w}}_t)\|_2^2\right]+10\lambda^2\delta_1^2\right) \\
&\leq \tilde{\eta}\frac{16\lambda^2\delta_1^2}{\mu_F}+\frac{\tilde{\eta}^3}{\beta^2}\frac{32L_F^2}{\mu_F}\sum_{k=1}^{N}\frac{1}{N}\left(14\mathbb{E}\left[\|\nabla F_k(\tilde{\mathbf{w}}_t)-\nabla F_k(\tilde{\mathbf{w}}^*)\|_2^2\right]+14\mathbb{E}\left[\|\nabla F_k(\tilde{\mathbf{w}}^*)\|_2^2\right]+10\lambda^2\delta_1^2\right) \\
&\leq \tilde{\eta}\frac{16\lambda^2\delta_1^2}{\mu_F}+\frac{\tilde{\eta}^3}{\beta^2}\frac{32L_F^2}{\mu_F}\left(28L_F\mathbb{E}\left[F(\tilde{\mathbf{w}}_t)-F(\tilde{\mathbf{w}}^*)\right]+14\sigma_{F,1}^2+10\lambda^2\delta_1^2\right) \\
&\leq \tilde{\eta}\frac{16\lambda^2\delta_1^2}{\mu_F}+180\frac{\tilde{\eta}^2}{\beta}\kappa_F L_F\mathbb{E}\left[F(\tilde{\mathbf{w}}_t)-F(\tilde{\mathbf{w}}^*)\right]+32\frac{\tilde{\eta}^3}{\beta^2}\kappa_F L_F(14\sigma_{F,1}^2+10\lambda^2\delta_1^2), \qquad (14)
\end{aligned}
$$

where the second inequality is by Proposition 21, the third inequality is by Proposition 20 and the definition of $\sigma_{F,1}^2$, and the last inequality is due to $\tilde{\eta} \leq \frac{\beta}{5L_F}$. Substituting (13) and (14) into Lemma 6 yields

$$
\begin{aligned}
\mathbb{E}\left[\|\tilde{\mathbf{w}}_{t+1}-\tilde{\mathbf{w}}^*\|_2^2\right] \leq &\left(1-\frac{\tilde{\eta}\mu_F}{2}\right)\mathbb{E}\left[\|\tilde{\mathbf{w}}_t-\tilde{\mathbf{w}}^*\|_2^2\right] \\
&-\tilde{\eta}\left[2-L_F\tilde{\eta}\left(6+12\frac{N/S-1}{N-1}+\frac{180\kappa_F}{\beta}\right)\right]\mathbb{E}\left[F(\tilde{\mathbf{w}}_t)-F(\tilde{\mathbf{w}}^*)\right] \\
&+\tilde{\eta}\underbrace{\frac{16\lambda^2\delta_1^2}{\mu_F}}_{=:C_1}+\tilde{\eta}^2\underbrace{\frac{6\sigma_{F,1}^2(N/S-1)}{N-1}}_{=:C_2}+\frac{\tilde{\eta}^3}{\beta^2}\underbrace{32\kappa_F L_F(14\sigma_{F,1}^2+10\lambda^2\delta_1^2)}_{=:C_3}.
\end{aligned}
$$

Since $\frac{N/S-1}{N-1} \leq 1$ and $\tilde{\eta} = \beta R\eta \leq \frac{1}{18L_F(1+10\kappa_F/\beta)}$, we have

$$
2-L_F\tilde{\eta}\left(6+12\frac{N/S-1}{N-1}+\frac{180\kappa_F}{\beta}\right) \geq 2-L_F\tilde{\eta}\left(18+180\frac{\kappa_F}{\beta}\right) \geq 1.
$$

It follows that

$$\mathbb{E}\left[\|\tilde{\mathbf{w}}_{t+1}-\tilde{\mathbf{w}}^*\|_2^2\right] \le \left(1-\frac{\tilde{\eta}\mu_F}{2}\right)\mathbb{E}\left[\|\tilde{\mathbf{w}}_t-\tilde{\mathbf{w}}^*\|_2^2\right]-\tilde{\eta}\mathbb{E}\left[F(\tilde{\mathbf{w}}_t)-F(\tilde{\mathbf{w}}^*)\right]+\tilde{\eta}C_1+\tilde{\eta}^2C_2+\frac{\tilde{\eta}^3}{\beta^2}C_3. \tag{15}$$

Let $\Delta_t = \|\tilde{\mathbf{w}}_t-\tilde{\mathbf{w}}^*\|_2^2$ and $\alpha_{-1}=1$. Then we have $(1-\tilde{\eta}\mu_F/2)\,\alpha_t=\alpha_{t-1}$ for $t\ge 0$. Rearranging the terms of (15), multiplying both sides by $\frac{\alpha_t}{\tilde{\eta}A_T}$ and summing over the index t, we obtain

$$\begin{aligned}
\sum_{t=0}^{T-1}\frac{\alpha_t\mathbb{E}[F(\tilde{\mathbf{w}}_t)]}{A_T}-F(\tilde{\mathbf{w}}^*) &\le \sum_{t=0}^{T-1}\mathbb{E}\left[\left(1-\frac{\tilde{\eta}\mu_F}{2}\right)\frac{\alpha_t\Delta_t}{\tilde{\eta}A_T}-\frac{\alpha_t\Delta_{t+1}}{\tilde{\eta}A_T}\right]+\frac{\tilde{\eta}^2}{\beta^2}C_3+\tilde{\eta}C_2+C_1\\
&\le \sum_{t=0}^{T-1}\mathbb{E}\left[\frac{\alpha_{t-1}\Delta_t-\alpha_t\Delta_{t+1}}{\tilde{\eta}A_T}\right]+\frac{\tilde{\eta}^2}{\beta^2}C_3+\tilde{\eta}C_2+C_1\\
&= \frac{\Delta_0}{\tilde{\eta}A_T}-\frac{\alpha_{T-1}\mathbb{E}[\Delta_T]}{\tilde{\eta}A_T}+\frac{\tilde{\eta}^2}{\beta^2}C_3+\tilde{\eta}C_2+C_1.
\end{aligned}$$

Now we bound A_T. First, we have

$$\begin{aligned}
A_T=\sum_{t=0}^{T-1}\alpha_t &= \sum_{t=0}^{T-1}\left(1-\frac{\tilde{\eta}\mu_F}{2}\right)^{-(t+1)} = \left(1-\frac{\tilde{\eta}\mu_F}{2}\right)^{-T}\sum_{t=0}^{T-1}\left(1-\frac{\tilde{\eta}\mu_F}{2}\right)^{t}\\
&= \alpha_{T-1}\frac{1-\left(1-\frac{\tilde{\eta}\mu_F}{2}\right)^T}{\tilde{\eta}\mu_F/2} \le \frac{2\alpha_{T-1}}{\tilde{\eta}\mu_F}.
\end{aligned}$$

On the other hand, setting $\tilde{\eta}T\ge 2/\mu_F$ yields

$$A_T=\alpha_{T-1}\frac{1-\left(1-\frac{\tilde{\eta}\mu_F}{2}\right)^T}{\tilde{\eta}\mu_F/2} \ge \alpha_{T-1}\frac{1-\exp(-\tilde{\eta}\mu_F T/2)}{\tilde{\eta}\mu_F/2} \ge \alpha_{T-1}\frac{1-\exp(-1)}{\tilde{\eta}\mu_F/2} \ge \frac{\alpha_{T-1}}{\tilde{\eta}\mu_F}.$$

Then we have $\frac{1}{\tilde{\eta}A_T}\le\frac{\mu_F}{\alpha_{T-1}}=\mu_F\left(1-\frac{\tilde{\eta}\mu_F}{2}\right)^T\le\mu_F\exp(-\tilde{\eta}\mu_F/2)$ and $\frac{\alpha_{T-1}}{\tilde{\eta}A_T}\ge\frac{\mu_F}{2}$. It follows that

$$\sum_{t=0}^{T-1}\frac{\alpha_t\mathbb{E}[F(\tilde{\mathbf{w}}_t)]}{A_T}-F(\tilde{\mathbf{w}}^*) \le \mu_F\exp(-\tilde{\eta}\mu_F/2)\Delta_0-\frac{\mu_F}{2}\mathbb{E}[\Delta_T]+\frac{\tilde{\eta}^2}{\beta^2}C_3+\tilde{\eta}C_2+C_1. \tag{16}$$

Since F is convex and $\Delta_T\ge 0$, this implies

$$\mathbb{E}[F(\bar{\mathbf{w}}_T)]-F(\tilde{\mathbf{w}}^*) \le \mu_F\exp(-\tilde{\eta}\mu_F/2)\Delta_0+\frac{\tilde{\eta}^2}{\beta^2}C_3+\tilde{\eta}C_2+C_1. \tag{17}$$

Recall that we need to ensure $\eta\le\frac{\hat{\eta}_1}{\beta R}$ (i.e., $\tilde{\eta}\le\hat{\eta}_1$) and $T\ge\frac{2}{\tilde{\eta}\mu_F}$ (i.e., $\tilde{\eta}\ge\frac{2}{\mu_F T}$). Now we use the techniques in Karimireddy et al. (2020). Arjevani et al. (2018), Stich (2019) to specify the value of T and $\tilde{\eta}$.

- If $\hat{\eta}_1 \geq \frac{2\ln\left(\frac{\mu_F^2\Delta_0 T}{2C_2}\right)}{\mu_F T}$, we choose $\tilde{\eta} = \max\left\{\frac{2\ln\left(\frac{\mu_F^2\Delta_0 T}{2C_2}\right)}{\mu_F T}, \frac{2}{\mu_F T}\right\}$. Then $\tilde{\eta} \geq \frac{2}{\mu_F T}$. Recall that $T \geq \frac{2}{\hat{\eta}_1\mu_F}$, which implies $\frac{2}{\mu_F T} \leq \hat{\eta}_1$. It follows that $\tilde{\eta} \leq \hat{\eta}_1$. Moreover, we have $\tilde{\eta} = \tilde{\mathcal{O}}\left(\frac{1}{\mu_F T}\right)$. With this choice of $\tilde{\eta}$, the first term on the right-hand side of (17) is less than the second term. Thus we obtain
$$\mathbb{E}[F(\bar{\mathbf{w}}_T)] - F(\tilde{\mathbf{w}}^*) \leq \tilde{\mathcal{O}}\left(\frac{C_2}{\mu_F T}\right) + \tilde{\mathcal{O}}\left(\frac{C_3}{\mu_F^2\beta^2T^2}\right) + C_1.$$

- If $\hat{\eta}_1 < \frac{2\ln\left(\frac{\mu_F^2\Delta_0 T}{2C_2}\right)}{\mu_F T}$, we choose $\tilde{\eta} = \hat{\eta}_1$. Clearly, we have $T = \frac{2}{\tilde{\eta}\mu_F}$ and $\tilde{\eta} \leq \tilde{\mathcal{O}}\left(\frac{1}{\mu_F T}\right)$. This implies
$$\mathbb{E}[F(\bar{\mathbf{w}}_T)] - F(\tilde{\mathbf{w}}^*) \leq \mu_F\Delta_0\exp\left(\frac{-\hat{\eta}_1\mu_F T}{2}\right) + \tilde{\mathcal{O}}\left(\frac{C_2}{\mu_F T}\right) + \tilde{\mathcal{O}}\left(\frac{C_3}{\mu_F^2\beta^2T^2}\right) + C_1.$$

Combining these two cases, we obtain

$$\begin{aligned}\mathbb{E}[F(\bar{\mathbf{w}}_T)] - F(\tilde{\mathbf{w}}^*) \leq \mu_F\Delta_0\exp\left(\frac{-\hat{\eta}_1\mu_F T}{2}\right) + \tilde{\mathcal{O}}\left(\frac{(N/S{-}1)\sigma_{F,1}^2}{\mu_F TN}\right)\\ + \tilde{\mathcal{O}}\left(\frac{(\sigma_{F,1}^2{+}\lambda^2\delta_1^2)\kappa_F L_F}{\mu_F^2\beta^2T^2}\right) + \mathcal{O}\left(\frac{\lambda^2\delta_1^2}{\mu_F}\right) =: \mathcal{O}_1.\end{aligned}$$

Now we prove the second inequality. Let $\hat{\mathbf{x}}_k^T = \operatorname{argmin}_{\mathbf{x}_k\in\mathbb{R}^d}\left\{f_k(\mathbf{x}_k) + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}}_T - \boldsymbol{P}\mathbf{x}_k\right\|_2^2\right\}$. By Proposition 21, we have

$$\begin{aligned}&\mathbb{E}\left[\left\|\boldsymbol{P}\mathbf{x}_k^T - \tilde{\mathbf{w}}^*\right\|_2^2\right]\\ &\leq 3\mathbb{E}\left[\left\|\boldsymbol{P}\mathbf{x}_k^T - \boldsymbol{P}\hat{\mathbf{x}}_k^T\right\|_2^2\right] + 3\mathbb{E}\left[\left\|\boldsymbol{P}\hat{\mathbf{x}}_k^T - \tilde{\mathbf{w}}_T\right\|_2^2\right] + 3\mathbb{E}\left[\left\|\tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^*\right\|_2^2\right]\\ &\leq 3\delta_1^2 + \frac{3}{\lambda^2}\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}_T)\right\|_2^2\right] + 3\mathbb{E}\left[\left\|\tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^*\right\|_2^2\right]\\ &\leq 3\delta_1^2 + \frac{6}{\lambda^2}\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}_T) - \nabla F_k(\tilde{\mathbf{w}}^*)\right\|_2^2\right] + \frac{6}{\lambda^2}\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}^*)\right\|_2^2\right] + 3\mathbb{E}\left[\left\|\tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^*\right\|_2^2\right],\end{aligned}$$

where the second inequality is by Lemma 4 and Proposition 3 and the last inequality is by Proposition 21. Proposition 3 also implies that F_k is λ-smooth. Then we have

$$\mathbb{E}\left[\left\|\boldsymbol{P}\mathbf{x}_k^T - \tilde{\mathbf{w}}^*\right\|_2^2\right] \leq 3\delta_1^2 + 9\mathbb{E}\left[\left\|\tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^*\right\|_2^2\right] + \frac{6}{\lambda^2}\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}^*)\right\|_2^2\right],$$

Note that the left-hand side of (16) is nonnegative. From the above analysis, with our choices of $\tilde{\eta}$ and T, we have

$$\frac{\mu_F}{2}\mathbb{E}\left[\left\|\tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^*\right\|_2^2\right] \leq \mu_F\exp(-\tilde{\eta}\mu_F/2)\Delta_0 + \frac{\tilde{\eta}^2}{\beta^2}C_3 + \tilde{\eta}C_2 + C_1 \leq \mathcal{O}_1.$$

Taking the average over the index k, we obtain

$$\frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\left[\left\|\boldsymbol{P}\mathbf{x}_k^T-\tilde{\mathbf{w}}^*\right\|_2^2\right]\le 9\mathbb{E}\left[\|\tilde{\mathbf{w}}_T-\tilde{\mathbf{w}}^*\|_2^2\right]+\frac{6\sigma_{F,1}^2}{\lambda^2}+3\delta_1^2\le\frac{1}{\mu_F}\mathcal{O}_1+\mathcal{O}\left(\frac{\sigma_{F,1}^2}{\lambda^2}+\delta_1^2\right).$$

This completes the proof. ■

A.3 Proof of Theorem 13

In this subsection, we give the proof of Theorem 13. For the smooth case, Lemma 8 still holds. And similar to Lemma 7, we have the following lemma that gives an upper bound on the drift error of the inner loop.

Lemma 26 (Bounded client drift error) *Suppose that Assumptions 2, 4, 6 and (5) hold with* $0<s<1/30$. *For* $\tilde{\eta}\le\frac{\beta}{5L_F}$, *we have*

$$\frac{1}{NR}\sum_{k=1}^{N}\sum_{r=0}^{R-1}\mathbb{E}\left[\left\|\mathbf{g}_{k,r}^t-\nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right]\le 2\lambda^2\delta_2^2+\frac{4L_F^2\tilde{\eta}^2}{\beta^2}\left(\frac{7}{N}\sum_{k=1}^{N}\mathbb{E}\left[\|\nabla F_k(\tilde{\mathbf{w}}_t)\|_2^2\right]+10\lambda^2\delta_2^2\right),$$

where δ_2^2 *is defined in Lemma 11.*

Proof [Proof of Theorem 13] We first assume $\eta\le\frac{\hat{\eta}_2}{\beta R}$. The exact value of η will be determined later. By Proposition 24, we have (5) holds with probability at least $1-2\exp(-cd_{\text{sub}})$ and $0<s<1/30$ as long as d_{sub}/d is sufficiently small. Throughout the proof, we assume this inequality holds.

Recall that with $\tilde{\eta}$ and $\mathbf{g}_t$ defined in (7), we have $\tilde{\mathbf{w}}_{t+1}=\tilde{\mathbf{w}}_t-\tilde{\eta}\mathbf{g}_t$. By Proposition 10, F_k is L_F-smooth, then F is also L_F-smooth. This implies that

$$\begin{aligned}
&\mathbb{E}\left[F(\tilde{\mathbf{w}}_{t+1})-F(\tilde{\mathbf{w}}_t)\right]\\
&\le\mathbb{E}\left[\langle\nabla F(\tilde{\mathbf{w}}_t),\tilde{\mathbf{w}}_{t+1}-\tilde{\mathbf{w}}_t\rangle\right]+\frac{L_F}{2}\mathbb{E}\left[\|\tilde{\mathbf{w}}_{t+1}-\tilde{\mathbf{w}}_t\|\,|_2^2\right]\\
&=-\tilde{\eta}\mathbb{E}\left[\langle\nabla F(\tilde{\mathbf{w}}_t),\mathbf{g}_t\rangle\right]+\frac{\tilde{\eta}^2L_F}{2}\mathbb{E}\left[\|\mathbf{g}_t\|_2^2\right]\\
&=-\tilde{\eta}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right]-\tilde{\eta}\mathbb{E}\left[\langle\nabla F(\tilde{\mathbf{w}}_t),\mathbf{g}_t-\nabla F(\tilde{\mathbf{w}}_t)\rangle\right]+\frac{\tilde{\eta}^2L_F}{2}\mathbb{E}\left[\|\mathbf{g}_t\|_2^2\right]\\
&\le-\tilde{\eta}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right]+\frac{\tilde{\eta}}{2}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right]+\frac{\tilde{\eta}}{2}\mathbb{E}\left[\left\|\frac{1}{NR}\sum_{k=1}^{N}\sum_{r=0}^{R-1}(\mathbf{g}_{k,r}^t-\nabla F_k(\tilde{\mathbf{w}}_t))\right\|_2^2\right]\\
&\quad+\frac{\tilde{\eta}^2L_F}{2}\mathbb{E}\left[\|\mathbf{g}_t\|_2^2\right],
\end{aligned}\tag{18}$$

where $\mathbf{g}_{k,r}^t$ is defined in (6) and the last inequality is by Cauchy-Schwarz inequality. Next from the proof of Lemma 3 in Dinh et al. (2020), we have

$$\mathbb{E}_{\mathcal{S}_t}\left[\|\mathbf{g}_t\|_2^2\right]\le 3\mathbb{E}_{\mathcal{S}_t}\frac{1}{NR}\sum_{k=1}^{N}\sum_{r=0}^{R-1}\left\|\mathbf{g}_{k,r}^t-\nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2+3\mathbb{E}_{\mathcal{S}_t}\left\|\frac{1}{S}\sum_{k\in\mathcal{S}_t}\nabla F_k(\tilde{\mathbf{w}}_t)-\nabla F(\tilde{\mathbf{w}}_t)\right\|_2^2$$

$$+ 3\mathbb{E}_{\mathcal{S}_t} \left\|\nabla F(\tilde{\mathbf{w}}_t)\right\|_2^2 . \tag{19}$$

We defer the proof of (19) to the end of this subsection. Recall that $\hat{\eta}_2 = \frac{1}{90\lambda^2 L_F}$, $\beta \geq 1$ and $\lambda \geq 1$. $\eta \leq \frac{\hat{\eta}_2}{\beta R}$ implies that $\tilde{\eta} = \beta R \eta \leq \frac{\beta}{5L_F}$. Substituting (19) into (18) yields

$$\begin{aligned}
&\mathbb{E}\left[F(\tilde{\mathbf{w}}_{t+1}) - F(\tilde{\mathbf{w}}_t)\right] \\
\leq& -\frac{\tilde{\eta}}{2}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] + \left(\frac{\tilde{\eta}}{2} + \frac{3\tilde{\eta}^2 L_F}{2}\right)\frac{1}{NR}\sum_{r=0}^{R-1}\sum_{k=1}^{N}\mathbb{E}\left[\left\|\mathbf{g}_{k,r}^t - \nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right] \\
&+ \frac{3\tilde{\eta}^2 L_F}{2}\mathbb{E}\left[\frac{1}{S}\sum_{k\in\mathcal{S}_t}\nabla F_k(\tilde{\mathbf{w}}_t) - \nabla F(\tilde{\mathbf{w}}_t)\right] + \frac{3\tilde{\eta}^2 L_F}{2}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] \\
\leq& -\frac{\tilde{\eta}(1-3\tilde{\eta}L_F)}{2}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] + \frac{3\tilde{\eta}^2 L_F}{2}\frac{N/S-1}{N-1}\sum_{k=1}^{N}\frac{1}{N}\mathbb{E}\left[\|\nabla F_k(\tilde{\mathbf{w}}_t) - \nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] \\
&+\frac{\tilde{\eta}(1+3\tilde{\eta}L_F)}{2}\left[2\lambda^2\delta_2^2 + \frac{4L_F^2\tilde{\eta}^2}{\beta^2}\left(\frac{7}{N}\sum_{k=1}^{N}\mathbb{E}\left[\|\nabla F_k(\tilde{\mathbf{w}}_t) - \nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] + 7\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] + 10\lambda^2\delta_2^2\right)\right] \\
\leq& -\frac{\tilde{\eta}(1-3\tilde{\eta}L_F)}{2}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] + \frac{3\tilde{\eta}^2 L_F}{2}\frac{N/S-1}{N-1}\left(3\sigma_{F,2}^2 + \frac{10L^2}{\lambda^2 - 10L^2}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right]\right) \\
&+ \frac{\tilde{\eta}(1+3\tilde{\eta}L_F)}{2}\left[2\lambda^2\delta_2^2 + \frac{4L_F^2\tilde{\eta}^2}{\beta^2}\left(21\sigma_{F,2}^2 + \frac{7\lambda^2}{\lambda^2 - 10L^2}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] + 10\lambda^2\delta_2^2\right)\right] \\
=& -\tilde{\eta}\left[\frac{1}{2} - \tilde{\eta}L_F\left(\frac{3}{2} + \frac{15L^2}{\lambda^2 - 10L^2}\frac{N/S-1}{N-1} + \frac{14(1+3\tilde{\eta}L_F)\lambda^2\tilde{\eta}L_F}{\beta^2(\lambda^2 - 10L^2)}\right)\right]\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] \\
&+ \frac{\tilde{\eta}^3}{\beta^2}(1+3\tilde{\eta}L_F)2L_F^2(21\sigma_{F,2}^2 + 10\lambda^2\delta_2^2) + \tilde{\eta}^2\frac{9}{2}L_F\sigma_{F,2}^2\frac{N/S-1}{N-1} + \tilde{\eta}(1+3\tilde{\eta}L_F)\lambda^2\delta_2^2,
\end{aligned}$$

where the second inequality is by Lemmas 7 and 8 and the fact that $\mathbb{E}[\|X\|_2^2] = \mathbb{E}[\|X - \mathbb{E}[X]\|_2^2] + \|\mathbb{E}[X]\|_2^2$ for a random vector X, and the last inequality is by Lemma 12.

Clearly, we also have $\tilde{\eta} \leq \frac{\beta}{2L_F}$, which implies that $1 + 3\tilde{\eta}L_F \leq 1 + 3\beta/2 \leq 3\beta$. Recall that $\lambda^2 - 10L^2 \geq 1$ and $\frac{N/S-1}{N-1} \leq 1$. Then we have

$$\frac{3}{2} + \frac{15L^2}{\lambda^2 - 10L^2}\frac{N/S-1}{N-1} + \frac{14(1+3\tilde{\eta}L_F)\lambda^2\tilde{\eta}L_F}{\beta^2(\lambda^2 - 10L^2)} \leq \frac{3}{2} + 15L^2 + 21\lambda^2 \leq \frac{45}{2}\lambda^2.$$

Since $\tilde{\eta} = \beta R\eta \leq \frac{1}{90\lambda^2 L_F}$, then

$$\frac{1}{2} - \tilde{\eta}L_F\left(\frac{3}{2} + \frac{15L^2}{\lambda^2 - 10L^2}\frac{N/S-1}{N-1} + \frac{14(1+3\tilde{\eta}L_F)\lambda^2\tilde{\eta}L_F}{\beta^2(\lambda^2 - 10L^2)}\right) \geq \frac{1}{2} - \frac{45\lambda^2\tilde{\eta}L_F}{2} \geq \frac{1}{4},$$

Moreover, the choice of λ implies $\lambda \geq 1$. Then we have $1 + 3\tilde{\eta}L_F \leq 1 + \frac{1}{15\lambda^2} \leq 2$. It follows that

$$\mathbb{E}\left[F(\tilde{\mathbf{w}}_{t+1}) - F(\tilde{\mathbf{w}}_t)\right] \leq -\frac{\tilde{\eta}}{4}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] + \frac{\tilde{\eta}^3}{\beta^2}\underbrace{4L_F^2(21\sigma_{F,2}^2 + 10\lambda^2\delta_2^2)}_{=:C_4}$$

$$+\,\tilde{\eta}^2\underbrace{5L_F\sigma_{F,2}^2\frac{N/S-1}{N-1}}_{=:C_5}+\tilde{\eta}\underbrace{2\lambda^2\delta_2^2}_{=:C_6}.$$

By rearranging the terms and telescoping, we obtain

$$\frac{1}{4T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right]\le\frac{\mathbb{E}\left[F(\tilde{\mathbf{w}}_0)-F(\tilde{\mathbf{w}}_T)\right]}{\tilde{\eta}T}+\frac{\tilde{\eta}^2}{\beta^2}C_4+\tilde{\eta}C_5+C_6. \tag{20}$$

Now we use the techniques in Karimireddy et al. (2020). Arjevani et al. (2018), Stich (2019) to specify the value of $\tilde{\eta}$. Recall that we need to ensure $\eta\le\frac{\hat{\eta}_2}{\beta R}$ (i.e., $\tilde{\eta}\le\hat{\eta}_2$).

- If $\hat{\eta}_2^3\ge\frac{\beta^2\Delta_F}{TC_4}$ or $\hat{\eta}_2^2\ge\frac{\Delta_F}{TC_5}$, then the first term on the right-hand side of (20) is no large than the sum of the second and third terms. We choose $\tilde{\eta}=\min\left\{\left(\frac{\beta^2\Delta_F}{TC_4}\right)^{1/3},\left(\frac{\Delta_F}{TC_5}\right)^{1/2}\right\}$. Then we have $\tilde{\eta}\le\hat{\eta}_2$ and

$$\frac{1}{4T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right]\le 2\frac{\Delta_F^{2/3}C_4^{1/3}}{(\beta T)^{2/3}}+2\frac{(\Delta_F C_5)^{1/2}}{\sqrt{T}}+C_6.$$

- If $\hat{\eta}_2^3<\frac{\beta^2\Delta_F}{TC_4}$ and $\hat{\eta}_2^2<\frac{\Delta_F}{TC_5}$, then the first term on the right-hand side of (20) is larger than the second and third terms. We choose $\tilde{\eta}=\hat{\eta}_2$ and obtain

$$\frac{1}{4T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right]\le 3\frac{\Delta_F}{\hat{\eta}_2 T}+C_6.$$

Combine the two cases and sampling t^* uniformly from $\{0,1,\ldots,T-1\}$, we have

$$\begin{aligned}&\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_{t^*})\|_2^2\right]=\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right]\\&\le\mathcal{O}\left(\frac{\Delta_F}{\hat{\eta}_2T}+\frac{\Delta_F^{2/3}L_F^{2/3}\left(\sigma_{F,2}^2+\lambda^2\delta_2^2\right)^{1/3}}{\beta^{2/3}T^{2/3}}+\frac{\left(\Delta_F L_F\sigma_{F,2}^2(N/S-1)\right)^{1/2}}{\sqrt{TN}}+\lambda^2\delta_2^2\right)=:\mathcal{O}_2.\end{aligned}$$

Now we prove the second inequality. Let $\hat{\mathbf{y}}_k^t=\operatorname{argmin}_{\mathbf{y}_k\in\mathbb{R}^{d_{\text{sub}}}}\left\{f_k(\boldsymbol{P}^\top\mathbf{y}_k)+\frac{\lambda}{2}\left\|\tilde{\mathbf{w}}_t-\boldsymbol{P}\boldsymbol{P}^\top\mathbf{y}_k\right\|_2^2\right\}$ and $\hat{\mathbf{x}}_k^t=\boldsymbol{P}^\top\hat{\mathbf{y}}_k^t$. By Proposition 21, we have

$$\begin{aligned}\frac{1}{N}\sum_{k=1}^N\mathbb{E}\left[\left\|\boldsymbol{P}\mathbf{x}_k^t-\tilde{\mathbf{w}}_t\right\|_2^2\right]&\le\frac{2}{N}\sum_{k=1}^N\mathbb{E}\left[\left\|\boldsymbol{P}\mathbf{x}_k^t-\boldsymbol{P}\hat{\mathbf{x}}_k^t\right\|_2^2+\left\|\boldsymbol{P}\hat{\mathbf{x}}_k^t-\tilde{\mathbf{w}}_t\right\|_2^2\right]\\&\le 2\delta_2^2+\frac{2}{N}\sum_{i=1}^N\frac{\mathbb{E}\left[\|\nabla F_k(\tilde{\mathbf{w}}_t)\|_2^2\right]}{\lambda^2},\end{aligned}\tag{21}$$

where the last inequality is by Proposition 10 and Lemma 11. Due to Lemma 12 and the fact that $\mathbb{E}[\|X\|_2^2] = \mathbb{E}[\|X - \mathbb{E}[X]\|_2^2] + \|\mathbb{E}[X]\|_2^2$ for a random vector X , we have

$$\begin{aligned} \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\|\nabla F_k(\tilde{\mathbf{w}}_t)\|_2^2\right] &\le \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\|\nabla F_k(\tilde{\mathbf{w}}_t) - \nabla F(\tilde{\mathbf{w}}_t)\|_2^2 + \|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] \\ &\le 3\sigma_{F,2}^2 + \frac{\lambda^2}{\lambda^2 - 10L^2}\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2 . \end{aligned} \tag{22}$$

Substituting (22) into (21) and taking the average over the index t, we obtain

$$\begin{aligned} \frac{1}{TN}\sum_{t=0}^{T-1}\sum_{k=1}^{N}\mathbb{E}\left[\left\|\boldsymbol{P}\mathbf{x}_k^t - \tilde{\mathbf{w}}_t\right\|_2^2\right] &\le \frac{2}{\lambda^2 - 10L^2}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] + 2\delta_2^2 + \frac{6\sigma_{F,2}^2}{\lambda^2} \\ &\le \mathcal{O}_2 + \mathcal{O}\left(\delta_2^2 + \frac{\sigma_{F,2}^2}{\lambda^2}\right), \end{aligned}$$

where the last inequality is due to $\lambda \ge \sqrt{10L^2+1}$. This completes the proof. ■

Proof [Proof of (19)] By Proposition 21, we have

$$\begin{aligned} &\mathbb{E}_{\mathcal{S}_t}\left[\|\mathbf{g}_t\|_2^2\right] \\ &\le 3\mathbb{E}_{\mathcal{S}_t}\left[\left\|\frac{1}{SR}\sum_{k\in\mathcal{S}_t}\sum_{r=0}^{R-1}(\mathbf{g}_{k,r}^t - \nabla F_k(\tilde{\mathbf{w}}_t))\right\|_2^2 + \left\|\frac{1}{S}\sum_{k\in\mathcal{S}_t}\nabla F_k(\tilde{\mathbf{w}}_t) - \nabla F(\tilde{\mathbf{w}}_t)\right\|_2^2 + \|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] \\ &\le 3\mathbb{E}_{\mathcal{S}_t}\left[\frac{1}{SR}\sum_{k\in\mathcal{S}_t}\sum_{r=0}^{R-1}\left\|\mathbf{g}_{k,r}^t - \nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2 + \left\|\frac{1}{S}\sum_{k\in\mathcal{S}_t}\nabla F_k(\tilde{\mathbf{w}}_t) - \nabla F(\tilde{\mathbf{w}}_t)\right\|_2^2 + \|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right]. \end{aligned}$$

If we only consider the randomness from the sampling of $\mathcal{S}_t$, $\mathbf{g}_{k,r}^t$ and $\nabla F_k(\tilde{\mathbf{w}}_t)$ become constant vectors. Use $\mathbb{1}_A$ to denote the indicator function of an event A. Uniform sampling implies $\mathbb{E}_{\mathcal{S}_t}\left[\mathbb{1}_{k\in\mathcal{S}_t}\right] = \frac{S}{N}$. Then we have

$$\begin{aligned} \frac{1}{SR}\mathbb{E}_{\mathcal{S}_t}\left[\sum_{k\in\mathcal{S}_t}\sum_{r=0}^{R-1}\left\|\mathbf{g}_{k,r}^t - \nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right] &= \frac{1}{SR}\sum_{k=1}^{N}\sum_{r=0}^{R-1}\left\|\mathbf{g}_{k,r}^t - \nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\mathbb{E}_{\mathcal{S}_t}\left[\mathbb{1}_{k\in\mathcal{S}_t}\right] \\ &= \frac{1}{NR}\sum_{k=1}^{N}\sum_{r=0}^{R-1}\left\|\mathbf{g}_{k,r}^t - \nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2, \end{aligned}$$

This completes the proof. ■

A.4 Proof of Auxiliary Results

Proof [Proof of Proposition 3] From the discussion at the end of Section IV.2.4 in Hiriart-Urruty and Lemaréchal (1993), we know that F_k is convex. By Proposition 22, we have

$\mathrm{dom} f_k^* = \mathbb{R}^d$ and $F_k^{**} = F_k$. Then $\mathrm{dom} F_k^* = \mathbb{R}^{d_{\mathrm{sub}}}$. It suffices to prove F_k^* is $\frac{1}{L_F}$-strongly convex and $\frac{1}{\mu_F}$-smooth. By the definition of conjugate function, we can compute F_k^* as

$$\begin{aligned} F_k^*(\mathbf{u}) &= \sup_{\tilde{\mathbf{w}} \in \mathbb{R}^{d_{\mathrm{sub}}}} \left\{ \langle \mathbf{u}, \tilde{\mathbf{w}} \rangle - \inf_{\mathbf{x} \in \mathbb{R}^d} \left[f_k(\boldsymbol{x}) + \frac{\lambda}{2} \left\| \tilde{\mathbf{w}} - \boldsymbol{P}\mathbf{x} \right\|_2^2 \right] \right\} \\ &= \sup_{\tilde{\mathbf{w}} \in \mathbb{R}^{d_{\mathrm{sub}}}, \mathbf{x} \in \mathbb{R}^d} \left\{ \langle \mathbf{u}, \tilde{\mathbf{w}} \rangle - f_k(\mathbf{x}) - \frac{\lambda}{2} \left\| \tilde{\mathbf{w}} - \boldsymbol{P}\mathbf{x} \right\|_2^2 \right\} \\ &= \sup_{\mathbf{x} \in \mathbb{R}^d} \left\{ \langle \mathbf{u}, \boldsymbol{P}\mathbf{x} \rangle - f_k(\mathbf{x}) + \sup_{\tilde{\mathbf{w}} \in \mathbb{R}^{d_{\mathrm{sub}}}} \left[\langle \mathbf{u}, \tilde{\mathbf{w}} - \boldsymbol{P}\mathbf{x} \rangle - \frac{\lambda}{2} \left\| \tilde{\mathbf{w}} - \boldsymbol{P}\mathbf{x} \right\|_2^2 \right] \right\} \\ &= \sup_{\mathbf{x} \in \mathbb{R}^d} \left[\left\langle \boldsymbol{P}^\top \mathbf{u}, \mathbf{x} \right\rangle - f_k(\mathbf{x}) \right] + \frac{1}{2\lambda} \left\| \mathbf{u} \right\|_2^2 \\ &= f_k^*(\boldsymbol{P}^\top \mathbf{u}) + \frac{1}{2\lambda} \left\| \mathbf{u} \right\|_2^2 . \end{aligned}$$

By Proposition 22, we have the following results.

(a) F_k^* is $1/\lambda$-strongly convex, then F_k is λ-smooth.

(b) f_k^* is $1/\mu$-smooth. By Proposition 25, $f_k^*(\boldsymbol{P}^\top \cdot)$ is $(1+s)^2/\mu$-smooth. Then F_k^* is $\big((1+s)^2/\mu + 1/\lambda\big)$-smooth. It follows that F_k is $\frac{\lambda\mu}{(1+s)^2\lambda+\mu}$-strongly convex.

Moreover, by Proposition 16.59 in Bauschke and Combettes (2011), we have $\nabla F_k(\tilde{\mathbf{w}}) = \lambda(\tilde{\mathbf{w}} - \boldsymbol{P}\hat{\mathbf{x}}_k)$.

Finally, we give the proof of the last claim. If f_k is L-smooth, then by Proposition 22, $f_k^\star$ is $1/L$-strongly convex. Then Proposition 20 implies that for any $\mathbf{u}_1, \mathbf{u}_2$ and $0 \le \alpha \le 1$, we have

$$\alpha f_k^\star(\boldsymbol{P}^\top \mathbf{u}_1) + (1-\alpha) f_k^\star(\boldsymbol{P}^\top \mathbf{u}_2) \ge f\left(\boldsymbol{P}^\top (\alpha \mathbf{u}_1 + (1-\alpha)\mathbf{u}_2) \right) + \alpha(1-\alpha) \frac{1}{2L} \left\| \boldsymbol{P}^\top (\mathbf{u}_1 - \mathbf{u}_2) \right\|_2^2 .$$

By Proposition 25, we have

$$\alpha f_k^\star(\boldsymbol{P}^\top \mathbf{u}_1) + (1-\alpha) f_k^\star(\boldsymbol{P}^\top \mathbf{u}_2) \ge f\left(\boldsymbol{P}^\top (\alpha \mathbf{u}_1 + (1-\alpha)\mathbf{u}_2) \right) + \alpha(1-\alpha) \frac{(1-s)^2}{2L} \left\| \boldsymbol{P}^\top (\mathbf{u}_1 - \mathbf{u}_2) \right\|_2^2 .$$

This implies that $f_k^\star(\boldsymbol{P}^\top \cdot)$ is $(1-s)^2/L$-strongly convex. It follows that $F_k^\star$ is $((1-s)^2/L + 1/\lambda)$-strongly convex and then F_k is $\frac{\lambda L}{(1-s)^2\lambda+L}$-smooth. ■

Proof [Proof of Lemma 4] By Proposition 3 and (5), we have $\left\| \nabla F_k(\tilde{\mathbf{w}}_{k,r}^t) - \lambda(\tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_{k,r}^t) \right\|_2 = \lambda \left\| \boldsymbol{P}(\hat{\mathbf{x}}_{k,r}^t - \mathbf{x}_{k,r}^t) \right\|_2 \le \lambda(1+s) \left\| \hat{\mathbf{x}}_{k,r}^t - \mathbf{x}_{k,r}^t \right\|_2$. Then we focus on the distance between $\hat{\mathbf{x}}_{k,r}^t$ and $\mathbf{x}_{k,r}^t$.

For convenience, let $h_k(\mathbf{x}_k; \tilde{\mathbf{w}}_{k,r}^t) = f_k(\mathbf{x}_k) + \frac{\lambda}{2} \left\| \tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_k \right\|_2^2$. Recall the definition of $\tilde{h}_k$ in Eqn. (3). Clearly, $\tilde{h}_k$ is μ-strongly convex and $\nabla h_k(\hat{\mathbf{x}}_{k,r}^t; \tilde{\mathbf{w}}_{k,r}^t) = \mathbf{0}$. By Proposition 20 and 21, we have

$$\mathbb{E}_{\tilde{\mathcal{D}}_k} \left\| \hat{\mathbf{x}}_{k,r}^t - \mathbf{x}_{k,r}^t \right\|_2^2 \le \frac{1}{\mu^2} \mathbb{E}_{\tilde{\mathcal{D}}_k} \left\| \nabla \tilde{h}_k(\hat{\mathbf{x}}_{k,r}^t; \tilde{\mathcal{D}}_k, \tilde{\mathbf{w}}_{k,r}^t) - \nabla \tilde{h}_k(\mathbf{x}_{k,r}^t; \tilde{\mathcal{D}}_k, \tilde{\mathbf{w}}_{k,r}^t) \right\|_2^2$$

$$
\begin{aligned}
&\le \frac{2}{\mu^2}\left(\mathbb{E}_{\tilde{\mathcal{D}}_k}\left\|\nabla\tilde{h}_k(\hat{\mathbf{x}}^t_{k,r};\tilde{\mathcal{D}}_k,\tilde{\mathbf{w}}^t_{k,r}) - \nabla h_k(\hat{\mathbf{x}}^t_{k,r};\tilde{\mathbf{w}}^t_{k,r})\right\|_2^2 + \mathbb{E}_{\tilde{\mathcal{D}}_k}\left\|\nabla\tilde{h}_k(\mathbf{x}^t_{k,r};\tilde{\mathcal{D}}_k,\tilde{\mathbf{w}}^t_{k,r})\right\|_2^2\right)\\
&\le \frac{2}{\mu^2}\left(\mathbb{E}_{\tilde{\mathcal{D}}_k}\left\|\frac{1}{|\tilde{\mathcal{D}}_k|}\sum_{\xi_{k,i}\in\tilde{\mathcal{D}}_k}\nabla\tilde{f}_k(\hat{\mathbf{x}}^t_{k,r};\xi_{k,i}) - \nabla f_k(\hat{\mathbf{x}}^t_{k,r})\right\|_2^2 + \nu\right)\\
&\le \frac{2}{\mu^2}\left(\frac{1}{|\tilde{\mathcal{D}}_k|^2}\sum_{\xi_{k,i}\in\tilde{\mathcal{D}}_k}\mathbb{E}_{\xi_{k,i}}\left\|\nabla\tilde{f}_k(\hat{\mathbf{x}}^t_{k,r};\xi_{k,i}) - \nabla f_k(\hat{\mathbf{x}}^t_{k,r})\right\|_2^2 + \nu\right)\\
&\le \frac{2}{\mu^2}\left(\frac{\gamma_f^2}{|\tilde{\mathcal{D}}_k|} + \nu\right),
\end{aligned}
$$

where the fourth inequality is due to $\xi_{k,i}$ are independent and $\mathbb{E}_{\xi_{k,i}}\nabla\tilde{f}_k(\hat{\mathbf{x}}^t_{k,r};\xi_{k,i}) = f_k(\hat{\mathbf{x}}^t_{k,r})$ and the last inequality is by Assumption 2. This completes the proof. ■

Proof [Proof of Lemma 5] By Proposition 3, F_k is L_F-smooth with $L_F = \lambda$. Then by Proposition 21 and 20, we have

$$
\begin{aligned}
\frac{1}{N}\sum_{k=1}^N \|\nabla F_k(\tilde{\mathbf{w}})\|_2^2 &\le \frac{2}{N}\sum_{k=1}^N \|\nabla F_k(\tilde{\mathbf{w}}) - \nabla F_k(\tilde{\mathbf{w}}^*)\|_2^2 + \frac{2}{N}\sum_{k=1}^N \|\nabla F_k(\tilde{\mathbf{w}}^*)\|_2^2\\
&\le 4L_F(F(\tilde{\mathbf{w}}) - F(\tilde{\mathbf{w}}^*)) + \frac{2}{N}\sum_{k=1}^N \|\nabla F_k(\tilde{\mathbf{w}}^*)\|_2^2 .
\end{aligned}
$$

Note that $\nabla F(\tilde{\mathbf{w}}) = \frac{1}{N}\sum_{k=1}^N \nabla F_k(\tilde{\mathbf{w}})$. Since $\mathbb{E}\,\|X - \mathbb{E}X\|_2^2 \le \mathbb{E}\,\|X\|^2$ for any random vector X, we have $\frac{1}{N}\sum_{k=1}^N \|\nabla F_k(\tilde{\mathbf{w}}) - \nabla F(\tilde{\mathbf{w}})\|_2^2 \le \frac{1}{N}\sum_{k=1}^N \|\nabla F_k(\tilde{\mathbf{w}})\|_2^2$. This completes the proof. ■

Proof [Proof of Lemma 7] Recall that $\mathbf{g}^t_{k,r} = \lambda(\tilde{\mathbf{w}}^t_{k,r} - \boldsymbol{P}\mathbf{x}^t_{k,r})$ and $\nabla F_k(\tilde{\mathbf{w}}^t_{k,r}) = \lambda(\tilde{\mathbf{w}}^t_{k,r} - \boldsymbol{P}\hat{\mathbf{x}}^t_{k,r})$. Then we have

$$
\begin{aligned}
\mathbb{E}\left[\left\|\mathbf{g}^t_{k,r} - \nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right] &\le 2\mathbb{E}\left[\left\|\mathbf{g}^t_{k,r} - \nabla F_k(\tilde{\mathbf{w}}^t_{k,r})\right\|_2^2\right] + 2\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}^t_{k,r}) - \nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right]\\
&\le 2\mathbb{E}\left[\left\|\mathbf{g}^t_{k,r} - \nabla F_k(\tilde{\mathbf{w}}^t_{k,r})\right\|_2^2\right] + 2L_F^2\mathbb{E}\left[\left\|\tilde{\mathbf{w}}^t_{k,r} - \tilde{\mathbf{w}}_t\right\|_2^2\right]\\
&\le 2\lambda^2\delta_1^2 + 2L_F^2\mathbb{E}\left[\left\|\tilde{\mathbf{w}}^t_{k,r} - \tilde{\mathbf{w}}_t\right\|_2^2\right], \qquad (23)
\end{aligned}
$$

where the first inequality is by Proposition 21, the second inequality is by Proposition 3, and the last inequality is by Lemma 4. Next, we bound the second term $\left\|\tilde{\mathbf{w}}^t_{k,r} - \tilde{\mathbf{w}}_t\right\|_2^2$. By Proposition 21, for $r \ge 1$, we have

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\tilde{\mathbf{w}}^t_{k,r} - \tilde{\mathbf{w}}_t\right\|_2^2\right] = \mathbb{E}\left[\left\|\tilde{\mathbf{w}}^t_{k,r-1} - \tilde{\mathbf{w}}_t - \eta\mathbf{g}^t_{k,r-1}\right\|_2^2\right]\\
&\le \left(1 + \frac{1}{4R}\right)\mathbb{E}\left[\left\|\tilde{\mathbf{w}}^t_{k,r-1} - \tilde{\mathbf{w}}_t - \eta\nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right] + (1+4R)\eta^2\mathbb{E}\left[\left\|\mathbf{g}^t_{k,r-1} - \nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right]
\end{aligned}
$$

$$\le \left(1+\frac{1}{4R}\right)^2 \mathbb{E}\left[\left\|\tilde{\mathbf{w}}_{k,r-1}^t - \tilde{\mathbf{w}}_t\right\|_2^2\right] + \left(1+\frac{1}{4R}\right)(1+4R)\eta^2\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right]$$
$$+ (1+4R)\eta^2\mathbb{E}\left[\left\|\mathbf{g}_{k,r-1}^t - \nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right]. \tag{24}$$

Recall that $\tilde{\eta} = \eta\beta R \le \frac{\beta}{5L_F}$ and $R \ge 1$. Then we have $\left(1+\frac{1}{4R}\right)^2 \le 1 + \frac{9}{16R}$, $\left(1+\frac{1}{4R}\right)(1+4R) \le \frac{25}{4}R$ and $(1+4R)\eta^2 \le 5R\eta^2 \le 5R\frac{1}{25R^2L_F^2} = \frac{1}{5RL_F^2}$. Substituting these inequalities and (23) into (24) yields

$$\begin{aligned}\mathbb{E}\left[\left\|\tilde{\mathbf{w}}_{k,r}^t - \tilde{\mathbf{w}}_t\right\|_2^2\right] &\le \left(1+\frac{9}{16R}\right)\mathbb{E}\left[\left\|\tilde{\mathbf{w}}_{k,r-1}^t - \tilde{\mathbf{w}}_t\right\|_2^2\right] + \frac{25}{4}R\eta^2\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right] \\ &\quad + 10R\eta^2\lambda^2\delta_1^2 + \frac{2}{5R}\mathbb{E}\left[\left\|\tilde{\mathbf{w}}_{k,r-1}^t - \tilde{\mathbf{w}}_t\right\|_2^2\right] \\ &\le \left(1+\frac{1}{R}\right)\mathbb{E}\left[\left\|\tilde{\mathbf{w}}_{k,r-1}^t - \tilde{\mathbf{w}}_t\right\|_2^2\right] + 7R\eta^2\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right] + 10R\eta^2\lambda^2\delta_1^2.\end{aligned} \tag{25}$$

Note that (25) holds for any $1 \le r \le R$ and $\tilde{\mathbf{w}}_{k,0}^t = \tilde{\mathbf{w}}_t$. Applying (25) recursively, we obtain

$$\mathbb{E}\left[\left\|\tilde{\mathbf{w}}_{k,r}^t - \tilde{\mathbf{w}}_t\right\|_2^2\right] \le \left(7R\eta^2\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right] + 10R\eta^2\lambda^2\delta_1^2\right)\sum_{i=0}^{R-1}\left(1+\frac{1}{R}\right)^i.$$

Since $(1+x/n)^n \le e^x$ for any $x \in \mathbb{R}$, we have $\sum_{i=0}^{R-1}(1+1/R)^i = \frac{(1+1/R)^R - 1}{1/R} \le \frac{e-1}{1/R} \le 2R$. This implies

$$\mathbb{E}\left[\left\|\tilde{\mathbf{w}}_{k,r}^t - \tilde{\mathbf{w}}_t\right\|_2^2\right] \le \frac{14\tilde{\eta}^2}{\beta^2}\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right] + \frac{20\tilde{\eta}^2\lambda^2\delta_1^2}{\beta^2} \tag{26}$$

Substituting (26) into (23) yields

$$\mathbb{E}\left[\left\|\mathbf{g}_{k,r}^t - \nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right] \le 2\lambda^2\delta_1^2 + \frac{4L_F^2\tilde{\eta}^2}{\beta^2}\left(7\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right] + 10\lambda^2\delta_1^2\right).$$

Taking the average over the indices k and r, we obtain the desired result. ■

Proof [Proof of Proposition 10] This proof is adapted from Hoheisel et al. (2020).

Let $\varphi_\lambda(\mathbf{y}) = f_k(\boldsymbol{P}^\top\mathbf{y}) + \frac{\lambda}{2}\left\|\boldsymbol{P}\boldsymbol{P}^\top\mathbf{y}\right\|_2^2$. By (5), we have that the smallest eigenvalue of $\boldsymbol{P}\boldsymbol{P}^\top\boldsymbol{P}\boldsymbol{P}^\top$ is no less than $(1-s)^4$. Since $\lambda > 4L$ and $0 < s < 1/30$, $\varphi_\lambda(\mathbf{y})$ is $\left((1-s)^4\lambda - (1+s)^2L\right)$-strongly convex. Similarly, the function $f_k(\boldsymbol{P}^\top\mathbf{y}) + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}} - \boldsymbol{P}\boldsymbol{P}^\top\mathbf{y}\right\|_2^2$ is also $\left((1-s)^4\lambda - (1+s)^2L\right)$-strongly convex. Such an $\hat{\mathbf{y}}_k$ exists and is unique. By Proposition 22, φ_λ^* is a continuously differentiable function defined on $\mathbb{R}^{d_{\mathrm{sub}}}$ and $\nabla\varphi_\lambda^* = (\nabla\varphi_\lambda)^{-1}$.

Then we have

$$\begin{aligned}F_k(\mathbf{w}) &= \min_{\mathbf{x}_k \in \mathbb{R}^d}\left\{f_k(\mathbf{x}_k) + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}} - \boldsymbol{P}\mathbf{x}_k\right\|_2^2\right\} \\ &= \min_{\mathbf{y} \in \mathbb{R}^{d_{\mathrm{sub}}}}\left\{f_k(\boldsymbol{P}^\top\mathbf{y}) + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}} - \boldsymbol{P}\boldsymbol{P}^\top\mathbf{y}\right\|_2^2\right\}\end{aligned}$$

$$
\begin{aligned}
&= \frac{\lambda}{2}\left\|\tilde{\mathbf{w}}\right\|_2^2 - \sup_{\mathbf{y}\in\mathbb{R}^{d_{\text{sub}}}} \left\{\lambda\left\langle \tilde{\mathbf{w}}, \boldsymbol{P}\boldsymbol{P}^\top\mathbf{y}\right\rangle - f_k(\boldsymbol{P}^\top\mathbf{y}) - \frac{\lambda}{2}\left\|\boldsymbol{P}\boldsymbol{P}^\top\mathbf{y}\right\|_2^2\right\} \\
&= \frac{\lambda}{2}\left\|\tilde{\mathbf{w}}\right\|_2^2 - \varphi_\lambda^*(\lambda\boldsymbol{P}\boldsymbol{P}^\top\tilde{\mathbf{w}}),
\end{aligned}
$$

where the second equality is by Assumption 6. Then $\nabla F_k(\mathbf{w}) = \lambda\tilde{\mathbf{w}} - \lambda\boldsymbol{P}\boldsymbol{P}^\top\nabla\varphi_\lambda^*(\lambda\boldsymbol{P}\boldsymbol{P}^\top\tilde{\mathbf{w}})$. On the other hand, we have

$$
\hat{\mathbf{y}}_k = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^{d_{\text{sub}}}} \left\{f_k(\boldsymbol{P}^\top\mathbf{y}) + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}} - \boldsymbol{P}\boldsymbol{P}^\top\mathbf{y}\right\|_2^2\right\} = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^{d_{\text{sub}}}} \left\{\varphi_\lambda(\mathbf{y}) - \lambda\left\langle \tilde{\mathbf{w}}, \boldsymbol{P}\boldsymbol{P}^\top\mathbf{y}\right\rangle\right\}.
$$

The first-order condition implies $\nabla\varphi_\lambda(\hat{\mathbf{y}}_k) = \lambda\boldsymbol{P}\boldsymbol{P}^\top\tilde{\mathbf{w}}$. It follows that $\hat{\mathbf{y}}_k = \nabla\varphi_\lambda^*(\lambda\boldsymbol{P}\boldsymbol{P}^\top\tilde{\mathbf{w}})$. Finally, we obtain $\nabla F_k(\tilde{\mathbf{w}}) = \lambda\tilde{\mathbf{w}} - \lambda\boldsymbol{P}\boldsymbol{P}^\top\hat{\mathbf{y}}_k$.

Now we prove the Lipschitz continuity of ∇F_k. Let $\psi_\lambda(\mathbf{x}) = f_k(\mathbf{x}) + \frac{\lambda}{2}\left\|\boldsymbol{P}\mathbf{x}\right\|_2^2$ and $\hat{\mathbf{x}}_k = \boldsymbol{P}^\top\hat{\mathbf{y}}_k$. By Assumption 6, we have

$$
\hat{\mathbf{x}}_k \in \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^d} \left\{f_k(\mathbf{x}) + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}} - \boldsymbol{P}\mathbf{x}\right\|_2^2\right\} = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^d} \left\{\psi_\lambda(\mathbf{x}) - \lambda\left\langle \tilde{\mathbf{w}}, \boldsymbol{P}\mathbf{x}\right\rangle\right\}.
$$

The first-order condition implies $\nabla\psi_\lambda(\hat{\mathbf{x}}_k) = \lambda\boldsymbol{P}^\top\tilde{\mathbf{w}}$. By Proposition 25, ψ_λ is $\left((1-s)^2\lambda - L\right)$-strongly convex on $\operatorname{col}(\boldsymbol{P}^\top)$. Then we have that for any $\mathbf{x}\in\operatorname{col}(\boldsymbol{P}^\top)$, it holds that

$$
\psi_\lambda(\hat{\mathbf{x}}_k) \le \psi_\lambda(\mathbf{x}) + \lambda\left\langle \boldsymbol{P}^\top\tilde{\mathbf{w}}, \hat{\mathbf{x}}_k - \mathbf{x}\right\rangle - \frac{1}{2}\left((1-s)^2\lambda - L\right)\left\|\mathbf{x} - \hat{\mathbf{x}}_k\right\|_2^2.
$$

Recalling the definition of ψ_λ, we obtain

$$
\begin{aligned}
&f_k(\hat{\mathbf{x}}_k) + \frac{\lambda}{2}\left\|\boldsymbol{P}\hat{\mathbf{x}}_k\right\|_2^2 - \frac{\lambda}{2}\left\|\boldsymbol{P}\mathbf{x}\right\|_2^2 - \lambda\left\langle \boldsymbol{P}^\top\tilde{\mathbf{w}}, \hat{\mathbf{x}}_k - \mathbf{x}\right\rangle + \frac{\lambda}{2}\left\|\boldsymbol{P}\hat{\mathbf{x}}_k - \boldsymbol{P}\mathbf{x}\right\|_2^2 \\
\le{}& f_k(\mathbf{x}) + \left(\frac{L}{2} - \frac{(1-s)^2\lambda}{2}\right)\left\|\mathbf{x} - \hat{\mathbf{x}}_k\right\|_2^2 + \frac{\lambda}{2}\left\|\boldsymbol{P}\hat{\mathbf{x}}_k - \boldsymbol{P}\mathbf{x}\right\|_2^2.
\end{aligned}
$$

By Proposition 25, we have $\left\|\boldsymbol{P}\hat{\mathbf{x}}_k - \boldsymbol{P}\mathbf{x}\right\|_2^2 \le (1+s)^2\left\|\hat{\mathbf{x}}_k - \mathbf{x}\right\|_2^2$. It follows that

$$
\begin{aligned}
&f_k(\hat{\mathbf{x}}_k) + \frac{\lambda}{2}\left\|\boldsymbol{P}\hat{\mathbf{x}}_k\right\|_2^2 - \frac{\lambda}{2}\left\|\boldsymbol{P}\mathbf{x}\right\|_2^2 - \lambda\left\langle \boldsymbol{P}^\top\tilde{\mathbf{w}}, \hat{\mathbf{x}}_k - \mathbf{x}\right\rangle + \frac{\lambda}{2}\left\|\boldsymbol{P}\hat{\mathbf{x}}_k - \boldsymbol{P}\mathbf{x}\right\|_2^2 \\
\le{}& f_k(\mathbf{x}) + \left(\frac{L}{2} + 2s\lambda\right)\left\|\mathbf{x} - \hat{\mathbf{x}}_k\right\|_2^2,
\end{aligned}
$$

which is equivalent to

$$
f_k(\hat{\mathbf{x}}_k) + \lambda\left\langle \boldsymbol{P}^\top\boldsymbol{P}\hat{\mathbf{x}}_k - \boldsymbol{P}^\top\tilde{\mathbf{w}}, \hat{\mathbf{x}}_k - \mathbf{x}\right\rangle \le f_k(\mathbf{x}) + \left(\frac{L}{2} + 2s\lambda\right)\left\|\mathbf{x} - \hat{\mathbf{x}}_k\right\|_2^2. \tag{27}
$$

For a $\tilde{\mathbf{w}}' \ne \tilde{\mathbf{w}}$, let $\hat{\mathbf{y}}_k' = \operatorname{argmin}_{\mathbf{y}\in\mathbb{R}^{d_{\text{sub}}}}\left\{f_k(\boldsymbol{P}^\top\mathbf{y}) + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}}' - \boldsymbol{P}\boldsymbol{P}^\top\mathbf{y}\right\|_2^2\right\}$ and $\hat{\mathbf{x}}_k' = \boldsymbol{P}^\top\hat{\mathbf{y}}_k'$. Then we also have $\hat{\mathbf{x}}_k' \in \operatorname{col}(\boldsymbol{P}^\top)$. Replacing $\mathbf{x}$ by $\hat{\mathbf{x}}_k'$ in (27) gives

$$
f_k(\hat{\mathbf{x}}_k) + \lambda\left\langle \boldsymbol{P}^\top\boldsymbol{P}\hat{\mathbf{x}}_k - \boldsymbol{P}^\top\tilde{\mathbf{w}}, \hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k'\right\rangle \le f_k(\hat{\mathbf{x}}_k') + \left(\frac{L}{2} + 2s\lambda\right)\left\|\hat{\mathbf{x}}_k' - \hat{\mathbf{x}}_k\right\|_2^2.
$$

Changing the orders of $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{x}}_k'$ leads to

$$f_k(\hat{\mathbf{x}}_k') + \lambda \left\langle \boldsymbol{P}^\top \boldsymbol{P}\hat{\mathbf{x}}_k' - \boldsymbol{P}^\top \tilde{\mathbf{w}}', \hat{\mathbf{x}}_k' - \hat{\mathbf{x}}_k \right\rangle \leq f_k(\hat{\mathbf{x}}_k) + \left(\frac{L}{2} + 2s\lambda\right) \left\|\hat{\mathbf{x}}_k' - \hat{\mathbf{x}}_k\right\|_2^2.$$

Adding the above two inequalities and rearranging terms yields

$$\lambda \left\langle \boldsymbol{P}^\top \boldsymbol{P}(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k'), \hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k' \right\rangle - (L + 4s\lambda) \left\|\hat{\mathbf{x}}_k' - \hat{\mathbf{x}}_k\right\|_2^2 \leq \lambda \left\langle \boldsymbol{P}^\top (\tilde{\mathbf{w}} - \tilde{\mathbf{w}}'), \hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k' \right\rangle.$$

By Proposition 25, $\left\langle \boldsymbol{P}^\top \boldsymbol{P}(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k'), \hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k' \right\rangle = \|\boldsymbol{P}(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k')\|_2^2 \geq (1-s)^2 \|\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k'\|_2^2$. Then we have

$$\left((1 - 6s - s^2)\lambda - L\right) \left\|\hat{\mathbf{x}}_k' - \hat{\mathbf{x}}_k\right\|_2^2 \leq \lambda \left\langle \boldsymbol{P}^\top (\tilde{\mathbf{w}} - \tilde{\mathbf{w}}'), \hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k' \right\rangle.$$

Since $s < 1/30$ and $\lambda > 4L$, we have $(1 - 6s - s^2)\lambda - L > 0$. Dividing both sides by $(1 - 6s - s^2)\lambda - L - L$ gives

$$\left\|\hat{\mathbf{x}}_k' - \hat{\mathbf{x}}_k\right\|_2^2 \leq \frac{1}{1 - 6s - s^2 - L/\lambda} \left\langle \boldsymbol{P}^\top (\tilde{\mathbf{w}} - \tilde{\mathbf{w}}'), \hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k' \right\rangle. \tag{28}$$

Then we have

$$\begin{aligned}
\frac{1}{\lambda^2} \left\|\nabla F_k(\tilde{\mathbf{w}}) - \nabla F_k(\tilde{\mathbf{w}}')\right\|_2^2 &= \left\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}' - \boldsymbol{P}(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k')\right\|_2^2 \\
&= \left\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}'\right\|_2^2 - 2\left\langle \tilde{\mathbf{w}} - \tilde{\mathbf{w}}', \boldsymbol{P}(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k') \right\rangle + \left\|\boldsymbol{P}(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k')\right\|_2^2 \\
&\leq \left\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}'\right\|_2^2 - 2\left\langle \tilde{\mathbf{w}} - \tilde{\mathbf{w}}', \boldsymbol{P}(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k') \right\rangle + (1+s)^2 \left\|\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k'\right\|_2^2 \\
&\leq \left\|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}'\right\|_2^2 + \left(\frac{(1+s)^2}{1 - 6s - s^2 - L/\lambda} - 2\right) \left\langle \tilde{\mathbf{w}} - \tilde{\mathbf{w}}', \boldsymbol{P}(\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_k') \right\rangle,
\end{aligned}$$

where the first inequality is due to Proposition 25 and the second one is due to (28). Since $\lambda > 4L$ and $s < 1/30$, we have $\frac{(1+s)^2}{1-6s-s^2-L/\lambda} - 2 < 0$. As a result, $\|\nabla F_k(\tilde{\mathbf{w}}) - \nabla F_k(\tilde{\mathbf{w}}')\|_2^2 \leq \lambda^2 \|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}'\|_2^2$. ■

Proof [Proof of Lemma 11] Let $\mathbf{x}_{k,r}^t = \boldsymbol{P}^\top \mathbf{y}_{k,r}^t + \boldsymbol{Q}\tilde{\mathbf{y}}_{k,r}^t$. Recall that we have $\boldsymbol{PQ} = \mathbf{0}_{d_{\mathrm{sub}} \times (d - d_{\mathrm{sub}})}$. Then by Proposition 10 and (5), we have $\left\|\nabla F_k(\tilde{\mathbf{w}}_{k,r}^t) - \lambda(\tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_{k,r}^t)\right\|_2 = \lambda \left\|\boldsymbol{P}\boldsymbol{P}^\top(\hat{\mathbf{y}}_{k,r}^t - \mathbf{y}_{k,r}^t)\right\|_2 \leq (1+s)^2 \lambda \left\|\hat{\mathbf{y}}_{k,r}^t - \mathbf{y}_{k,r}^t\right\|_2$, where the minimizer $\hat{\mathbf{y}}_{k,r}^t$ is defined as $\hat{\mathbf{y}}_{k,r}^t = \mathrm{argmin}_{\mathbf{y}_k \in \mathbb{R}^{d_{\mathrm{sub}}}} \left\{ f_k(\boldsymbol{P}^\top \mathbf{y}_k) + \frac{\lambda}{2} \left\|\tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\boldsymbol{P}^\top \mathbf{y}_k\right\|_2^2 \right\}$. Then we focus on the distance between $\hat{\mathbf{y}}_{k,r}^t$ and $\mathbf{y}_{k,r}^t$.

Recall the definition of $\tilde{h}_k$ in (3). Throughout this proof, $\tilde{\mathbf{w}}_{k,r}^t$ and $\tilde{\mathcal{D}}_k$ are fixed, so we omit the dependence of $\tilde{h}_k$ on these parameters for brevity. For any $\mathbf{x}_k = (\boldsymbol{P}^\top, \boldsymbol{Q}) \begin{pmatrix} \mathbf{y}_k \\ \tilde{\mathbf{y}}_k \end{pmatrix}$, we have $\begin{pmatrix} \partial_{\mathbf{y}_k} \tilde{h}_k \\ \partial_{\tilde{\mathbf{y}}_k} \tilde{h}_k \end{pmatrix} = \begin{pmatrix} \boldsymbol{P} \\ \boldsymbol{Q}^\top \end{pmatrix} \nabla_{\mathbf{x}_k} \tilde{h}_k$. By Assumption 6, we have $\partial_{\tilde{\mathbf{y}}_k} \tilde{h}_k = \mathbf{0}$. Then with some abuse of notation, we can view $\tilde{h}_k$ as a function of $\mathbf{y}_k$:

$$\tilde{h}_k(\mathbf{y}_k) = \frac{1}{|\tilde{\mathcal{D}}_k|} \sum_{\xi_{k,i}\in\tilde{\mathcal{D}}_k} \tilde{f}_k(\boldsymbol{P}^\top \mathbf{y}_k; \xi_{k,i}) + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\boldsymbol{P}^\top \mathbf{y}_k\right\|_2^2,$$

and it holds that $\nabla_{\mathbf{y}_k}\tilde{h}_k = \boldsymbol{P}\nabla_{\mathbf{x}_k}\tilde{h}_k$.

For convenience, let $h_k(\mathbf{x}_k) = f_k(\mathbf{x}_k) + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_k\right\|_2^2$ and $\hat{\mathbf{x}}_{k,r}^t = \boldsymbol{P}^\top\hat{\mathbf{y}}_{k,r}^t$. By Proposition 25, $\tilde{h}_k$ is $\big((1-s)^4\lambda - (1+s)^2L\big)$-strongly convex in $\mathbf{y}_k$ and $\nabla_{\mathbf{y}_k}h_k(\hat{\mathbf{y}}_{k,r}^t) = \mathbf{0}$. Then by (5) and Propositions 20 and 21, we have

$$\begin{aligned}
&\mathbb{E}_{\tilde{\mathcal{D}}_k}\left\|\hat{\mathbf{y}}_{k,r}^t - \mathbf{y}_{k,r}^t\right\|_2^2 \\
\le\ & \frac{1}{\left((1-s)^4\lambda - (1+s)^2L\right)^2}\mathbb{E}_{\tilde{\mathcal{D}}_k}\left\|\nabla_{\mathbf{y}_k}\tilde{h}_k(\hat{\mathbf{y}}_{k,r}^t) - \nabla_{\mathbf{y}_k}\tilde{h}_k(\mathbf{y}_{k,r}^t)\right\|_2^2 \\
\le\ & \frac{(1+s)^2}{\left((1-s)^4\lambda - (1+s)^2L\right)^2}\mathbb{E}_{\tilde{\mathcal{D}}_k}\left\|\nabla_{\mathbf{x}_k}\tilde{h}_k(\hat{\mathbf{x}}_{k,r}^t) - \nabla_{\mathbf{x}_k}\tilde{h}_k(\mathbf{x}_{k,r}^t)\right\|_2^2 \\
\le\ & \frac{2(1+s)^2}{\left((1-s)^4\lambda - (1+s)^2L\right)^2}\left(\mathbb{E}_{\tilde{\mathcal{D}}_k}\left\|\nabla_{\mathbf{x}_k}\tilde{h}_k(\hat{\mathbf{x}}_{k,r}^t) - \nabla h_k(\hat{\mathbf{x}}_{k,r}^t)\right\|_2^2 + \mathbb{E}_{\tilde{\mathcal{D}}_k}\left\|\nabla_{\mathbf{x}_k}\tilde{h}_k(\mathbf{x}_{k,r}^t)\right\|_2^2\right) \\
\le\ & \frac{2(1+s)^2}{\left((1-s)^4\lambda - (1+s)^2L\right)^2}\left(\mathbb{E}_{\tilde{\mathcal{D}}_k}\left\|\frac{1}{|\tilde{\mathcal{D}}_k|}\sum_{\xi_{k,i}\in\tilde{\mathcal{D}}_k}\nabla\tilde{f}_k(\hat{\mathbf{x}}_{k,r}^t;\xi_{k,i}) - \nabla f_k(\hat{\mathbf{x}}_{k,r}^t)\right\|_2^2 + \nu\right) \\
\le\ & \frac{2(1+s)^2}{\left((1-s)^4\lambda - (1+s)^2L\right)^2}\left(\frac{1}{|\tilde{\mathcal{D}}_k|^2}\sum_{\xi_{k,i}\in\tilde{\mathcal{D}}_k}\mathbb{E}_{\xi_{k,i}}\left\|\nabla\tilde{f}_k(\hat{\mathbf{x}}_{k,r}^t;\xi_{k,i}) - \nabla f_k(\hat{\mathbf{x}}_{k,r}^t)\right\|_2^2 + \nu\right) \\
\le\ & \frac{2(1+s)^2}{\left((1-s)^4\lambda - (1+s)^2L\right)^2}\left(\frac{\gamma_f^2}{|\tilde{\mathcal{D}}_k|} + \nu\right),
\end{aligned}$$

where the fourth inequality is due to $\xi_{k,i}$ are independent and $\mathbb{E}_{\xi_{k,i}}\nabla\tilde{f}_k(\hat{\mathbf{x}}_{k,r}^t;\xi_{k,i}) = f_k(\hat{\mathbf{x}}_{k,r}^t)$ and the last inequality is by Assumption 2. Then by Proposition 10 and (5), we have

$$\frac{1}{\lambda^2}\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}_{k,r}^t) - \lambda(\tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_{k,r}^t)\right\|_2^2\right] \le \frac{2(1+s)^6}{\left((1-s)^4\lambda - (1+s)^2L\right)^2}\left(\frac{\gamma_f^2}{|\tilde{\mathcal{D}}_k|} + \nu\right).$$

■

Proof [Proof of Lemma 12] If f_k is L-smooth, by Proposition 10, we have

$$\|\nabla F_k(\tilde{\mathbf{w}}) - \nabla F(\tilde{\mathbf{w}})\|_2^2 = \left\|\lambda(\tilde{\mathbf{w}} - \boldsymbol{P}\hat{\mathbf{x}}_k) - \frac{1}{N}\sum_{i=1}^N \lambda(\tilde{\mathbf{w}} - \boldsymbol{P}\hat{\mathbf{x}}_i)\right\|_2^2,$$

where $\hat{\mathbf{x}}_k = \boldsymbol{P}^\top\hat{\mathbf{y}}_k$ with $\hat{\mathbf{y}}_k = \operatorname{argmin}_{\mathbf{y}_k\in\mathbb{R}^{d_{\text{sub}}}}\left\{f_k(\boldsymbol{P}^\top\mathbf{y}_k) + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}} - \boldsymbol{P}\boldsymbol{P}^\top\mathbf{y}_k\right\|_2^2\right\}$.

The first-order condition implies $\boldsymbol{P}\nabla f_k(\boldsymbol{P}^\top\hat{\mathbf{y}}_k) = \lambda\boldsymbol{P}\boldsymbol{P}^\top(\tilde{\mathbf{w}} - \boldsymbol{P}\boldsymbol{P}^\top\hat{\mathbf{y}}_k)$, which implies $\boldsymbol{P}\nabla f_k(\hat{\mathbf{x}}_k) = \lambda\boldsymbol{P}\boldsymbol{P}^\top(\tilde{\mathbf{w}} - \boldsymbol{P}\hat{\mathbf{x}}_k)$. By (5), it is easy to verify $\left\|(\boldsymbol{P}\boldsymbol{P}^\top)^{-1}\boldsymbol{P}\right\|_2 \le (1-s)^{-1}$ through SVD. Then we have

$$\begin{aligned}
\|\nabla F_k(\tilde{\mathbf{w}}) - \nabla F(\tilde{\mathbf{w}})\|_2^2 &= \left\|(\boldsymbol{P}\boldsymbol{P}^\top)^{-1}\boldsymbol{P}\left(\nabla f_k(\hat{\mathbf{x}}_k) - \frac{1}{N}\sum_{i=1}^N \nabla f_i(\hat{\mathbf{x}}_i)\right)\right\|_2^2 \\
&\le (1-s)^{-2}\left\|\left(\nabla f_k(\hat{\mathbf{x}}_k) - \frac{1}{N}\sum_{i=1}^N \nabla f_i(\hat{\mathbf{x}}_i)\right)\right\|_2^2 \\
&\le 2(1-s)^{-2}\left\|\left(\nabla f_k(\hat{\mathbf{x}}_k) - \frac{1}{N}\sum_{i=1}^N \nabla f_i(\hat{\mathbf{x}}_k)\right)\right\|_2^2 \\
&\quad + 2(1-s)^{-2}\left\|\left(\frac{1}{N}\sum_{i=1}^N \nabla f_i(\hat{\mathbf{x}}_k) - \frac{1}{N}\sum_{i=1}^N \nabla f_i(\hat{\mathbf{x}}_i)\right)\right\|_2^2,
\end{aligned}$$

where the last inequality is by Proposition 21.

Taking the average over the devices, we obtain that

$$\begin{aligned}
&\frac{1}{N}\sum_{k=1}^N \|\nabla F_k(\tilde{\mathbf{w}}) - \nabla F(\tilde{\mathbf{w}})\|_2^2 \\
\le& 2(1-s)^{-2}\sum_{k=1}^N \|(\nabla f_k(\hat{\mathbf{x}}_k) - \nabla f(\hat{\mathbf{x}}_k))\|_2^2 + 2(1-s)^{-2}\sum_{k=1}^N \left\|\frac{1}{N}\sum_{i=1}^N (\nabla f_i(\hat{\mathbf{x}}_k) - \nabla f_i(\hat{\mathbf{x}}_i))\right\|_2^2 \\
\le& 2(1-s)^{-2}\sigma_f^2 + \frac{2(1-s)^{-2}}{N^2}\sum_{k=1}^N\sum_{i=1}^N \|\nabla f_i(\hat{\mathbf{x}}_k) - \nabla f_i(\hat{\mathbf{x}}_i)\|_2^2, \qquad (29)
\end{aligned}$$

where the last inequality is by Assumption 5 and Proposition 21. By the smoothness of f_i, we have

$$\begin{aligned}
\|\nabla f_i(\hat{\mathbf{x}}_k) - \nabla f_i(\hat{\mathbf{x}}_i)\|_2^2 &\le L^2\|\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_i\|_2^2 = L^2\left\|\boldsymbol{P}^\top(\hat{\mathbf{y}}_k - \hat{\mathbf{y}}_i)\right\|_2^2 \\
&= L^2\left\|\boldsymbol{P}^\top(\boldsymbol{P}\boldsymbol{P}^\top)^{-1}\boldsymbol{P}\boldsymbol{P}^\top(\hat{\mathbf{y}}_k - \hat{\mathbf{y}}_i)\right\|_2^2 \\
&\le (1-s)^{-2}L^2\left\|\boldsymbol{P}\boldsymbol{P}^\top(\hat{\mathbf{y}}_k - \hat{\mathbf{y}}_i)\right\|_2^2 \\
&= (1-s)^{-2}L^2\|\boldsymbol{P}\hat{\mathbf{x}}_k - \boldsymbol{P}\hat{\mathbf{x}}_i\|_2^2 \\
&\le 2(1-s)^{-2}L^2\left(\|\boldsymbol{P}\hat{\mathbf{x}}_k - \tilde{\mathbf{w}}\|_2^2 + \|\boldsymbol{P}\hat{\mathbf{x}}_i - \tilde{\mathbf{w}}\|_2^2\right) \\
&= \frac{2(1-s)^{-2}L^2}{\lambda^2}\left(\|\nabla F_k(\tilde{\mathbf{w}})\|_2^2 + \|\nabla F_i(\tilde{\mathbf{w}})\|_2^2\right), \qquad (30)
\end{aligned}$$

where the third inequality is by Proposition 21 and the last equality is by Proposition 10. Substituting (30) into (29) gives

$$\frac{1}{N}\sum_{k=1}^N \|\nabla F_k(\tilde{\mathbf{w}}) - \nabla F(\tilde{\mathbf{w}})\|_2^2 \le 2(1-s)^{-2}\sigma_f^2 + \frac{8(1-s)^{-4}L^2}{\lambda^2}\frac{1}{N}\sum_{k=1}^N \|\nabla F_k(\tilde{\mathbf{w}})\|_2^2$$

$$\begin{aligned}
&\leq 3\sigma_f^2 + \frac{10L^2}{\lambda^2}\frac{1}{N}\sum_{k=1}^{N}\|\nabla F_k(\tilde{\mathbf{w}})\|_2^2 \\
&= 3\sigma_f^2 + \frac{10L^2}{\lambda^2}\left(\frac{1}{N}\sum_{k=1}^{N}\|\nabla F_k(\tilde{\mathbf{w}}) - \nabla F(\tilde{\mathbf{w}})\|_2^2 + \|\nabla F(\tilde{\mathbf{w}})\|_2^2\right),
\end{aligned}$$

where the second inequality follows from $s < 1/30$ and the last equality is due to the fact that $\mathbb{E}[\|X\|_2^2] = \mathbb{E}[\|X - \mathbb{E}[X]\|_2^2] + \|\mathbb{E}[X]\|_2^2$ for a random vector X. Finally, rearranging the terms yields

$$\frac{1}{N}\sum_{k=1}^{N}\|\nabla F_k(\tilde{\mathbf{w}}) - \nabla F(\tilde{\mathbf{w}})\|_2^2 \leq \frac{3\lambda^2}{\lambda^2 - 10L^2}\sigma_f^2 + \frac{10L^2}{\lambda^2 - 10L^2}\|\nabla F(\tilde{\mathbf{w}})\|_2^2.$$

■

A.5 Proof of Theorem 14

In this subsection, we give the proof of Theorem 14. We first give the formal statement of Theorem 14.

Theorem 27 (Formal version of Theorem 14) *Suppose that Assumptions 1 and 2 hold. Let* $\eta_t = \frac{8}{\beta R \mu_F(\zeta+t)}$ *,* $\nu_t = \frac{8}{\mu_F(\zeta+t)}$ *and* $D_t = \left\lceil \frac{\mu_F(\zeta+t)}{D} \right\rceil$*, where* $\zeta = 72\kappa_F(1 + 7\kappa_F/\beta)$ *and* D *is a positive constant. Then with probability at least* $1 - 2\exp(-cd_{\text{sub}})$*, we have*

$$\begin{aligned}
&\mathbb{E}\|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}^*\|_2^2 \leq \mathcal{O}\left(\frac{\Delta_0}{T^4} + \frac{\lambda^2(1+\gamma_f^2)}{\mu^2\mu_F^3 T}\left(1 + \frac{\kappa_F^2}{\beta^2 T^2}\right) + \frac{(N/S-1)\sigma_{F,1}^2}{\mu_F^2 N T} + \frac{\kappa_F^2\sigma_{F,1}^2}{\mu_F^2\beta^2 T^2}\right), \\
&\mathbb{E}\left\|\mathbf{x}_k^T - \mathbf{x}_k^*\right\|_2^2 \leq \mathcal{O}\left(\frac{1+\gamma_f^2}{\mu^2\mu_F T} + \frac{\lambda^2}{\mu^2}\mathbb{E}\|\tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^*\|_2^2\right),
\end{aligned}$$

where c *is a positive constant,* $\sigma_{F,1}^2 = \frac{1}{N}\sum_{k=1}^{N}\|\nabla F_k(\tilde{\mathbf{w}}^*)\|_2^2$ *and the expectation is w.r.t. all the randomness except for* $\boldsymbol{P}$*. Moreover, when there is no client sampling (*$S = N$*), let* $\nu_t = \frac{8}{\mu_F\beta^2(\xi+t)^2}$ *and* $D_t = \left\lceil \frac{\mu_F\beta^2(\xi+t)^2}{D} \right\rceil$*. Then with probability at least* $1 - 2\exp(-cd_{\text{sub}})$*, we have*

$$\begin{aligned}
&\mathbb{E}\|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}^*\|_2^2 \leq \mathcal{O}\left(\frac{\Delta_0}{T^4} + \frac{\lambda^2(1+\gamma_f^2)}{\mu_F^3\beta^2 T^2}\left(1 + \frac{\kappa_F}{\mu^2\beta^2 T}\right) + \frac{\kappa_F^2\sigma_{F,1}^2}{\mu_F^2\beta^2 T^2}\right), \\
&\mathbb{E}\left\|\mathbf{x}_k^T - \mathbf{x}_k^*\right\|_2^2 \leq \mathcal{O}\left(\frac{1+\gamma_f^2}{\mu^2\mu_F\beta^2 T^2} + \frac{\lambda^2}{\mu^2}\mathbb{E}\|\tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^*\|_2^2\right),
\end{aligned}$$

Note that there are four terms on the right-hand side. The first term is due to initialization and is negligible compared to other terms. The second term is from approximation error and mini-batch sampling in each client and is the leading term of order $\mathcal{O}(1/T)$. The third term is caused by client sampling. And the last term reflects the client drift with multiple local updates because of the diversity across clients. Note that a larger β leads to a smaller step size and consequently lightens the client drift.

Proof Recall that by Proposition 24, we have (5) holds with probability at least $1 - 2\exp(-cd_{\text{sub}})$ and $s = \mathcal{O}(1)$. Throughout the proof, we assume this inequality holds. In this case, Lemma 4 becomes that for a fixed $\tilde{\mathbf{w}}^t_{k,r}$, we have

$$\frac{1}{\lambda^2}\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}^t_{k,r}) - \lambda(\tilde{\mathbf{w}}^t_{k,r} - \boldsymbol{P}\mathbf{x}^t_{k,r})\right\|_2^2\right] \le \delta_t^2 := \frac{2(1+s)^2}{\mu^2}\left(\frac{\gamma_f^2}{D_t} + \nu_t\right).$$

With our choice of D_t and ν_t, we have $\delta_t^2 \le \frac{2(1+s)^2}{\mu^2\mu_F(\zeta+t)}(8 + D\gamma_f^2)$.

Similar to the proof of Theorem 9, we first rewrite the local update as

$$\tilde{\mathbf{w}}^t_{k.r+1} = \tilde{\mathbf{w}}^t_{k,r} - \eta_t \underbrace{\lambda(\tilde{\mathbf{w}}^t_{k,r} - \boldsymbol{P}\mathbf{x}^t_{k,r})}_{=:\mathbf{g}^t_{k,r}},$$

which implies

$$\eta_t\sum_{r=0}^{R-1}\mathbf{g}^t_{k,r} = \sum_{r=0}^{R-1}(\tilde{\mathbf{w}}^t_{k,r} - \tilde{\mathbf{w}}^t_{k,r+1}) = \tilde{\mathbf{w}}^t_{k,0} - \tilde{\mathbf{w}}^t_{k,R} = \tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}^t_{k,R}.$$

Then $\mathbf{g}^t_{k,r}$ can be considered as a biased estimate of $\nabla F_k(\tilde{\mathbf{w}}^t_{k,r})$ and the global update rule becomes

$$\tilde{\mathbf{w}}_{t+1} = (1-\beta)\tilde{\mathbf{w}}_t + \frac{\beta}{S}\sum_{k\in\mathcal{S}_t}\tilde{\mathbf{w}}^t_{k,R} = \tilde{\mathbf{w}}_t - \frac{\beta}{S}\sum_{k\in\mathcal{S}_t}(\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}^t_{k,R}) = \tilde{\mathbf{w}}_t - \underbrace{\eta_t\beta R}_{=:\tilde{\eta}_t}\underbrace{\frac{1}{SR}\sum_{k\in\mathcal{S}^t}\sum_{r=0}^{R-1}\mathbf{g}^t_{k,r}}_{=:\mathbf{g}_t},$$

where $\tilde{\eta}_t$ and $\mathbf{g}_t$ can be interpreted as the step size and the approximate stochastic gradient of the global update, respectively.

Similar to Lemma 6, we have the following inequality.

$$\begin{aligned}\mathbb{E}\left[\|\tilde{\mathbf{w}}_{t+1} - \tilde{\mathbf{w}}^*\|_2^2\right] \le \left(1 - \frac{\tilde{\eta}_t\mu_F}{2}\right)&\mathbb{E}\left[\|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}^*\|_2^2\right] - \tilde{\eta}_t(2 - 6L_F\tilde{\eta}_t)\mathbb{E}\left[F(\tilde{\mathbf{w}}_t) - F(\tilde{\mathbf{w}}^*)\right]\\ &+ \frac{\tilde{\eta}_t(3\tilde{\eta}_t + 2/\mu_F)}{NR}\sum_{k=1}^{N}\sum_{r=0}^{R-1}\mathbb{E}\left[\|\mathbf{g}_{t,r} - \nabla F_k(\tilde{\mathbf{w}}_t)\|_2^2\right]\\ &+ 3\tilde{\eta}_t^2\mathbb{E}\left[\left\|\frac{1}{S}\sum_{k\in\mathcal{S}_t}\nabla F_k(\tilde{\mathbf{w}}_t) - \nabla F(\tilde{\mathbf{w}}_t)\right\|_2^2\right]\end{aligned} \tag{31}$$

For the last term on the right-hand side of (31), Lemmas 5 and 8 imply

$$\mathbb{E}\left\|\frac{1}{S}\sum_{k\in\mathcal{S}_t}\nabla F_k(\bar{\mathbf{w}}^t_r) - \nabla F(\bar{\mathbf{w}}^t_r)\right\|_2^2 \le \frac{N/S-1}{N-1}\sum_{k=1}^{N}\frac{1}{N}\mathbb{E}\left\|\nabla F_k(\bar{\mathbf{w}}^t_r) - \nabla F(\bar{\mathbf{w}}^t_r)\right\|_2^2$$

$$\leq \frac{N/S-1}{N-1}\left(4L_F\mathbb{E}\left[F(\bar{\mathbf{w}}_r^t)-F(\tilde{\mathbf{w}}^*)\right]+2\sigma_{F,1}^2\right).$$

For the third term on the right-hand side of (31), we can resort to Lemma 7. For $\tilde{\eta}_t \leq \frac{\beta}{5L_F}$, we have

$$\begin{aligned}
&\frac{1}{NR}\sum_{k=1}^{N}\sum_{r=0}^{R-1}\mathbb{E}\left[\left\|\mathbf{g}_{k,r}^t-\nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right]\\
&\leq 2\lambda^2\delta_t^2+\frac{4L_F^2\tilde{\eta}_t^2}{\beta^2}\left(\frac{7}{N}\sum_{k=1}^{N}\mathbb{E}\left\|\nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2+10\lambda^2\delta_t^2\right)\\
&\leq 2\lambda^2\delta_t^2+\frac{4L_F^2\tilde{\eta}_t^2}{\beta^2}\left(\frac{14}{N}\sum_{k=1}^{N}\mathbb{E}\left\|\nabla F_k(\tilde{\mathbf{w}}_t)-\nabla F_k(\tilde{\mathbf{w}}^*)\right\|_2^2+\frac{14}{N}\sum_{k=1}^{N}\mathbb{E}\left\|\nabla F_k(\tilde{\mathbf{w}}^*)\right\|_2^2+10\lambda^2\delta_t^2\right)\\
&\leq 2\lambda^2\delta_t^2+\frac{8L_F^2\tilde{\eta}_t^2}{\beta^2}\left(14L_F\mathbb{E}[F(\tilde{\mathbf{w}}_t)-F(\tilde{\mathbf{w}}^*)]+7\sigma_{F,1}^2+5\lambda^2\delta_t^2\right),
\end{aligned}$$

where the second inequality is by Proposition 21 and the last inequality is due to Proposition 20. Since $\eta_t=\frac{8}{\beta R\mu_F(\zeta+t)}$ with $\zeta=72\kappa_F(1+7\kappa_F/\beta)$, we have $\tilde{\eta}_t=\beta R\eta_t\leq\frac{8}{\mu_F(\zeta+t)}\leq\min\left\{\frac{1}{\mu_F},\frac{\beta}{5L_F}\right\}$. As a result, we have

$$\begin{aligned}
&\frac{\tilde{\eta}_t(3\tilde{\eta}_t+2/\mu_F)}{NR}\sum_{k=1}^{N}\sum_{r=0}^{R-1}\mathbb{E}\left[\left\|\mathbf{g}_{k,r}^t-\nabla F_k(\tilde{\mathbf{w}}_t)\right\|_2^2\right]\\
&\leq\tilde{\eta}_t\frac{10\lambda^2\delta_t^2}{\mu_F}+\frac{\tilde{\eta}_t^3}{\beta^2}\frac{40L_F^2}{\mu_F}\left(14L_F\mathbb{E}\left[F(\tilde{\mathbf{w}}_t)-F(\tilde{\mathbf{w}}^*)\right]+7\sigma_{F,1}^2+5\lambda^2\delta_t^2\right)\\
&\leq\tilde{\eta}_t\frac{10\lambda^2\delta_t^2}{\mu_F}+112\frac{\tilde{\eta}_t^2}{\beta}\kappa_F L_F\mathbb{E}\left[F(\tilde{\mathbf{w}}_t)-F(\tilde{\mathbf{w}}^*)\right]+280\frac{\tilde{\eta}_t^3}{\beta^2}\kappa_F L_F\sigma_{F,1}^2+200\frac{\tilde{\eta}_t^3}{\beta^2}\kappa_F L_F\lambda^2\delta_t^2
\end{aligned}$$

Substituting these inequalities into (31) yields

$$\begin{aligned}
&\mathbb{E}\left[\left\|\tilde{\mathbf{w}}_{t+1}-\tilde{\mathbf{w}}^*\right\|_2^2\right]\\
&\leq\left(1-\frac{\tilde{\eta}_t\mu_F}{2}\right)\mathbb{E}\left[\left\|\tilde{\mathbf{w}}_t-\tilde{\mathbf{w}}^*\right\|_2^2\right]-\tilde{\eta}_t\left[2-L_F\tilde{\eta}_t\left(6+12\frac{N/S-1}{N-1}+\frac{112\kappa_F}{\beta}\right)\right]\mathbb{E}\left[F(\tilde{\mathbf{w}}_t)-F(\tilde{\mathbf{w}}^*)\right]\\
&\quad+\tilde{\eta}_t\delta_t^2\underbrace{\frac{10\lambda^2}{\mu_F}}_{=:C_1}+\tilde{\eta}_t^2\underbrace{\frac{6\sigma_{F,1}^2(N/S-1)}{N-1}}_{=:C_2}+\frac{\tilde{\eta}_t^3}{\beta^2}\underbrace{280\kappa_F L_F\sigma_{F,1}^2}_{=:C_3}+\frac{\tilde{\eta}_t^3}{\beta^2}\delta_t^2\underbrace{200\kappa_F L_F\lambda^2}_{=:C_4}.
\end{aligned}$$

Since $\frac{N/S-1}{N-1}\leq 1$ and $\tilde{\eta}_t=\beta R\eta_t\leq\frac{8}{\mu_F(\zeta+t)}\leq\frac{1}{9L_F(1+7\kappa_F/\beta)}$, we have

$$2-L_F\tilde{\eta}_t\left(6+12\frac{N/S-1}{N-1}+\frac{112\kappa_F}{\beta}\right)\geq 2-L_F\tilde{\eta}_t\left(18+112\frac{\kappa_F}{\beta}\right)\geq 0.$$

It follows that

$$\mathbb{E}\Delta_{t+1}\leq\left(1-\frac{\mu_F\tilde{\eta}_t}{2}\right)\mathbb{E}\Delta_t+\tilde{\eta}_t\delta_t^2C_1+\tilde{\eta}_t^2C_2+\frac{\tilde{\eta}_t^3}{\beta^2}C_3+\frac{\tilde{\eta}_t^3}{\beta^2}\delta_t^2C_4,$$

where $\Delta_t := \|\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}^*\|_2^2$. Recall that $\delta_t^2 \le \frac{2(1+s)^2(8+D\gamma_f^2)}{\mu^2\mu_F(\zeta+t)} = C_0\tilde{\eta}_t$ with the constant $C_0 := \frac{2(1+s)^2}{\mu^2}\left(1+\frac{D\gamma_f^2}{8}\right) = \mathcal{O}\left(\frac{1+\gamma_f^2}{\mu^2}\right)$. Then we have

$$\mathbb{E}\Delta_{t+1} \le \left(1 - \frac{\mu_F\tilde{\eta}_t}{2}\right)\mathbb{E}\Delta_t + \tilde{\eta}_t^2(C_0C_1 + C_2) + \frac{\tilde{\eta}_t^3}{\beta^2}C_3 + \frac{\tilde{\eta}_t^4}{\beta^2}C_0C_4. \tag{32}$$

Applying (32) $T-1$ times yields

$$\begin{aligned}\mathbb{E}\Delta_T \le{}& \prod_{t=0}^{T-1}\left(1 - \frac{\mu_F\tilde{\eta}_t}{2}\right)\mathbb{E}\Delta_0 + (C_0C_1 + C_2)\sum_{t=0}^{T-1}\tilde{\eta}_t^2\prod_{s=t+1}^{T-1}\left(1 - \frac{\mu_F\tilde{\eta}_s}{2}\right)\\ &+ \frac{C_3}{\beta^2}\sum_{t=0}^{T-1}\tilde{\eta}_t^3\prod_{s=t+1}^{T-1}\left(1 - \frac{\mu_F\tilde{\eta}_s}{2}\right) + \frac{C_0C_4}{\beta^2}\sum_{t=0}^{T-1}\tilde{\eta}_t^4\prod_{s=t+1}^{T-1}\left(1 - \frac{\mu_F\tilde{\eta}_s}{2}\right).\end{aligned}$$

Since $\tilde{\eta}_t = \frac{8}{\mu_F(\zeta+t)}$, we have

$$\prod_{t=0}^{T-1}\left(1 - \frac{\mu_F\tilde{\eta}_t}{2}\right) \le \exp\left(-\frac{\mu_F}{2}\sum_{t=0}^{T-1}\tilde{\eta}_t\right) \le \exp\left(-4\ln(\zeta+T) + 4\ln\zeta\right).$$

Moreover, for $t \le T-1$, we have

$$\prod_{s=t+1}^{T-1}\left(1 - \frac{\mu_F\tilde{\eta}_s}{2}\right) = \prod_{s=t+1}^{T-1}\frac{\zeta+s-4}{\zeta+s} = \frac{(\zeta+t)(\zeta+t-1)(\zeta+t-2)(\zeta+t-3)}{(\zeta+T-1)(\zeta+T-2)(\zeta+T-3)(\zeta+T-4)}.$$

It follows that

$$\begin{aligned}\sum_{t=0}^{T-1}\tilde{\eta}_t^2\prod_{s=t+1}^{T-1}\left(1 - \frac{\mu_F\tilde{\eta}_s}{2}\right) &\le \frac{16\sum_{t=0}^{T-1}(\zeta+t-2)(\zeta+t-3)}{\mu_F^2(\zeta+T-1)(\zeta+T-2)(\zeta+T-3)(\zeta+T-4)}\\ &\le \frac{16}{3\mu_F^2(\zeta+T-1)},\\ \sum_{t=0}^{T-1}\tilde{\eta}_t^3\prod_{s=t+1}^{T-1}\left(1 - \frac{\mu_F\tilde{\eta}_s}{2}\right) &\le \frac{64\sum_{t=0}^{T-1}(\zeta+t-3)}{\mu_F^3(\zeta+T-1)(\zeta+T-2)(\zeta+T-3)(\zeta+T-4)}\\ &\le \frac{32}{\mu_F^3(\zeta+T-1)(\zeta+T-2)},\\ \sum_{t=0}^{T-1}\tilde{\eta}_t^4\prod_{s=t+1}^{T-1}\left(1 - \frac{\mu_F\tilde{\eta}_s}{2}\right) &\le \frac{256T}{\mu_F^4(\zeta+T-1)(\zeta+T-2)(\zeta+T-3)(\zeta+T-4)}.\end{aligned}$$

Combining all the inequalities, we obtain that

$$\begin{aligned}\mathbb{E}\Delta_T &\le \mathcal{O}\left(\frac{\Delta_0}{T^4} + \frac{C_0C_1+C_2}{\mu_F^2T} + \frac{C_3}{\mu_F^3\beta^2T^2} + \frac{C_0C_4}{\mu_F^4\beta^2T^3}\right)\\ &= \mathcal{O}\left(\frac{\Delta_0}{T^4} + \frac{\lambda^2(1+\gamma_f^2)}{\mu^2\mu_F^3T} + \frac{(N/S-1)\sigma_F^2}{\mu_F^2NT} + \frac{\kappa_F^2\sigma_{F,1}^2}{\mu_F^2\beta^2T^2} + \frac{\kappa_F^2\lambda^2(1+\gamma_f^2)}{\mu^2\mu_F^3\beta^2T^3}\right).\end{aligned}$$

For personalized parameters, suppose the optimal personalized parameter is $\hat{\mathbf{x}}_k^T$ when the global parameter is $\tilde{\mathbf{w}}_T$, that is, $\hat{\mathbf{x}}_k^T = \operatorname{argmin}_{\mathbf{x}_k \in \mathbb{R}^d} \left\{ f_k(\mathbf{x}_k) + \frac{\lambda}{2} \left\| \tilde{\mathbf{w}}_T - \boldsymbol{P}\mathbf{x}_k \right\|_2^2 \right\}$. Then by Proposition 21, we have

$$\left\| \mathbf{x}_k^T - \mathbf{x}_k^* \right\|_2^2 \leq 2 \left\| \mathbf{x}_k^T - \hat{\mathbf{x}}_k^T \right\|_2^2 + 2 \left\| \hat{\mathbf{x}}_k^T - \mathbf{x}_k^* \right\|_2^2 .$$

Similar to the proof of Lemma 4, we can prove that

$$\mathbb{E} \left\| \mathbf{x}_k^T - \hat{\mathbf{x}}_k^T \right\|_2^2 \leq \frac{2}{\mu^2} \left(\frac{\gamma_f^2}{D_T} + \nu_T \right).$$

It remains to bound $\left\| \hat{\mathbf{x}}_k^T - \mathbf{x}_k^* \right\|_2$. With $h_k(\mathbf{x}_k; \tilde{\mathbf{w}}) = f_k(\mathbf{x}_k) + \frac{\lambda}{2} \left\| \tilde{\mathbf{w}} - \boldsymbol{P}\mathbf{x}_k \right\|_2^2$, we have $\nabla h_k(\hat{\mathbf{x}}_k^T; \tilde{\mathbf{w}}_T) = \nabla h_k(\mathbf{x}_k^*; \tilde{\mathbf{w}}^*) = \mathbf{0}$. Clearly, $h_k(\mathbf{x}_k; \tilde{\mathbf{w}}^*)$ is μ-strongly convex. By Proposition 20, we have

$$\begin{aligned}
\left\| \hat{\mathbf{x}}_k^T - \mathbf{x}_k^* \right\|_2 &\leq \frac{1}{\mu} \left\| \nabla h_k(\hat{\mathbf{x}}_k^T; \tilde{\mathbf{w}}^*) - \nabla h_k(\mathbf{x}_k^*; \tilde{\mathbf{w}}^*) \right\|_2 \\
&= \frac{1}{\mu} \left\| \nabla h_k(\hat{\mathbf{x}}_k^T; \tilde{\mathbf{w}}^*) - \nabla h_k(\hat{\mathbf{x}}_k^T; \tilde{\mathbf{w}}_T) \right\|_2 \\
&= \frac{\lambda}{\mu} \left\| \boldsymbol{P}^\top (\tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^*) \right\|_2 \\
&\leq \frac{\lambda(1+s)}{\mu} \left\| \tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^* \right\|_2 .
\end{aligned}$$

As a result,

$$\begin{aligned}
\mathbb{E} \left\| \mathbf{x}_k^T - \mathbf{x}_k^* \right\|_2^2 &\leq \frac{4}{\mu^2} \left(\frac{\gamma_f^2}{D_T} + \nu_T \right) + \frac{2\lambda^2 (1+s)^2}{\mu^2} \mathbb{E} \left\| \tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^* \right\|_2^2 \\
&\leq \mathcal{O} \left(\frac{1 + \gamma_f^2}{\mu^2 \mu_F T} + \frac{\lambda^2}{\mu^2} \mathbb{E} \left\| \tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^* \right\|_2^2 \right).
\end{aligned}$$

When $S = N$, we choose $\nu_t = \frac{8}{\mu_F \beta^2 (\xi+t)^2}$, $D_t = \left\lceil \frac{\mu_F \beta^2 (\xi+t)^2}{D} \right\rceil$ and $\eta_t = \frac{8}{\beta R \mu_F (\xi+t)}$ ($\tilde{\eta}_t = \frac{8}{\mu_F (\xi+t)}$). Then we have $\delta_t^2 \leq \frac{2(1+s)^2}{\mu^2 \mu_F \beta^2 (\zeta+t)^2} (8 + D\gamma_f^2) = \tilde{C}_0 \frac{\tilde{\eta}_t^2}{\beta^2}$ with $\tilde{C}_0 := \frac{(1+s)^2 \mu_F}{4\mu^2} \left(1 + \frac{D\gamma_f^2}{8} \right) = \mathcal{O} \left(\frac{\mu_F (1+\gamma_f^2)}{\mu^2} \right)$. Since $\tilde{\eta}_t \leq \frac{1}{9L_F}$, (32) becomes

$$\mathbb{E}\Delta_{t+1} \leq \left(1 - \frac{\mu_F \tilde{\eta}_t}{2} \right) \mathbb{E}\Delta_t + \frac{\tilde{\eta}_t^3}{\beta^2} (\tilde{C}_0 C_1 + C_3) + \frac{\tilde{\eta}_t^4}{\beta^4} \frac{\tilde{C}_0 C_4}{9L_F}.$$

Similar to the analysis above, we can obtain

$$\mathbb{E}\Delta_T \leq \mathcal{O} \left(\frac{\Delta_0}{T^4} + \frac{\tilde{C}_0 C_1 + C_3}{\mu_F^3 \beta^2 T^2} + \frac{\tilde{C}_0 C_4}{\mu_F^4 L_F \beta^4 T^3} \right)$$

$$
\begin{aligned}
&= \mathcal{O}\left(\frac{\Delta_0}{T^4} + \frac{\lambda^2(1+\gamma_f^2)}{\mu_F^3\beta^2T^2} + \frac{\kappa_F^2\sigma_{F,1}^2}{\mu_F^2\beta^2T^2} + \frac{\kappa_F\lambda^2(1+\gamma_f^2)}{\mu^2\mu_F^3\beta^4T^3}\right),\\
\mathbb{E}\left\|\mathbf{x}_k^T - \mathbf{x}_k^*\right\|_2^2 &\le \frac{4}{\mu^2}\left(\frac{\gamma_f^2}{D_T} + \nu_T\right) + \frac{2\lambda^2(1+s)^2}{\mu^2}\mathbb{E}\left\|\tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^*\right\|_2^2\\
&\le \mathcal{O}\left(\frac{1+\gamma_f^2}{\mu^2\mu_F\beta^2T^2} + \frac{\lambda^2}{\mu^2}\mathbb{E}\left\|\tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^*\right\|_2^2\right).
\end{aligned}
$$

This completes the proof. ■

A.6 Proof of Theorem 15

In this subsection, we give the formal statement and proof of Theorem 15.

Theorem 28 (Formal version of Theorem 15) *Suppose that Assumptions 2 to 6 hold. Define* $\alpha_t := \frac{\eta_t}{\sum_{t=0}^{T-1}\eta_t}$ *and sample* t^* *from* $\{0,1,\ldots,T-1\}$ *with* $\mathbb{P}(t^*=i)=\alpha_t$*. Let* $\eta_t = \frac{1}{90\beta R\lambda^2 L_F\sqrt{t+1}}$, $\nu_t = \frac{1}{90\lambda^2 L_F\sqrt{t+1}}$, $D_t = \left\lceil \frac{90\lambda^2 L_F\sqrt{t+1}}{D}\right\rceil$ *and* $\Delta_F = F(\tilde{\mathbf{w}}_0) - \min_{\tilde{\mathbf{w}}\in\mathbb{R}^{d_{\text{sub}}}} F(\tilde{\mathbf{w}})$*, where* $\lambda \ge \max\{\sqrt{10L^2+1}, 4L\}$, $\beta \ge 1$ *and* D *is a positive constant. With probability at least* $1-2\exp(-cd_{\text{sub}})$*, we have*

$$
\mathbb{E}\left[\left\|\nabla F(\tilde{\mathbf{w}}_{t^*})\right\|_2^2\right] \le \mathcal{O}\left(\frac{\lambda^2L_F^2\Delta_F}{\sqrt{T}} + \frac{(1+\gamma_f^2)\ln T}{\lambda^2L_F\sqrt{T}} + \frac{(N/S-1)\sigma_{F,2}^2\ln T}{\lambda^2N\sqrt{T}} + \frac{\sigma_{F,2}^2}{\lambda^4\beta^2\sqrt{T}}\right),
$$

where c *is a positive constant,* $L_F = \lambda$ *is the smoothness parameter of* F_k, $\sigma_{F,2}^2 = \frac{\lambda^2\sigma_f^2}{\lambda^2-10L^2}$ *measures the bounded diversity of* F_k*, the expectation is w.r.t. all the randomness except for* $\boldsymbol{P}$ *and* $\mathcal{O}$ *hides constants. Moreover, when there is no client sampling* ($S=N$)*, let* $\eta_t = \frac{1}{90\beta^{1/3}R\lambda^2L_F(t+1)^{1/3}}$, $\nu_t = \frac{1}{90\lambda^2L_F\beta^{2/3}(t+1)^{2/3}}$ *and* $D_t = \left\lceil\frac{90\lambda^2L_F\beta^{2/3}(t+1)^{2/3}}{D}\right\rceil$*. Then with probability at least* $1-2\exp(-cd_{\text{sub}})$*, we have*

$$
\mathbb{E}\left[\left\|\nabla F(\tilde{\mathbf{w}}_{t^*})\right\|_2^2\right] \le \mathcal{O}\left(\frac{\lambda^2L_F\Delta_F}{\beta^{2/3}T^{2/3}} + \frac{\sigma_{F,2}^2\ln T}{\lambda^4\beta^{2/3}T^{2/3}} + \frac{(1+\gamma_f^2)(\lambda^{-4}+\ln T)}{\lambda^2L_F\beta^{2/3}T^{2/3}}\right).
$$

Note that there are four terms on the right-hand side. The first term is due to initialization. The second term is from approximation error and mini-batch sampling in each client. The third term is caused by client sampling. And the last term reflects the client drift with multiple local updates because of the diversity across clients. A larger β leads to a smaller step size and consequently lightens the client drift. In terms of the number of communication rounds, the order is $\mathcal{O}\left(\ln T/\sqrt{T}\right)$.

Proof By Proposition 24, we have (5) holds with probability at least $1-2\exp(-cd_{\text{sub}})$ and $0 < s < 1/30$ as long as d_{sub}/d is sufficiently small. Throughout the proof, we assume this inequality holds. In this case, Lemma 11 becomes that for a fixed $\tilde{\mathbf{w}}_{k,r}^t$, we have

$$
\frac{1}{\lambda^2}\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}_{k,r}^t) - \lambda(\tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_{k,r}^t)\right\|_2^2\right] \le \delta_t^2 := \frac{2(1+s)^6}{((1-s)^4\lambda-(1+s)^2L)^2}\left(\frac{\gamma_f^2}{D_t} + \nu_t\right).
$$

With our choice of D_t and ν_t, we have $\delta_t^2 \le \frac{(1+s)^6(\gamma_f^2 D+1)}{45((1-s)^4\lambda-(1+s)^2L)^2\lambda^2 L_F\sqrt{t+1}}$.

Similar to the proof of Theorem 13, we rewrite the local update as

$$\tilde{\mathbf{w}}_{k.r+1}^t = \tilde{\mathbf{w}}_{k,r}^t - \eta_t \underbrace{\lambda(\tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_{k,r}^t)}_{=:\mathbf{g}_{k,r}^t},$$

which implies

$$\eta_t \sum_{r=0}^{R-1} \mathbf{g}_{k,r}^t = \sum_{r=0}^{R-1} (\tilde{\mathbf{w}}_{k,r}^t - \tilde{\mathbf{w}}_{k,r+1}^t) = \tilde{\mathbf{w}}_{k,0}^t - \tilde{\mathbf{w}}_{k,R}^t = \tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{k,R}^t.$$

Then the global update rule becomes

$$\tilde{\mathbf{w}}_{t+1} = (1-\beta)\tilde{\mathbf{w}}_t + \frac{\beta}{S}\sum_{k\in\mathcal{S}_t} \tilde{\mathbf{w}}_{k,R}^t = \tilde{\mathbf{w}}_t - \frac{\beta}{S}\sum_{k\in\mathcal{S}_t}(\tilde{\mathbf{w}}_t - \tilde{\mathbf{w}}_{k,R}^t) = \tilde{\mathbf{w}}_t - \underbrace{\eta_t\beta R}_{=:\tilde{\eta}_t} \underbrace{\frac{1}{SR}\sum_{k\in\mathcal{S}^t}\sum_{r=0}^{R-1} \mathbf{g}_{k,r}^t}_{=:\mathbf{g}_t}.$$

With our choice of η_t, we have $\tilde{\eta}_t = \frac{1}{90\lambda^2 L_F\sqrt{t+1}}$. Since $\lambda, \beta \ge 1$, it holds that $\tilde{\eta}_t \le \frac{\beta}{5L_F}$ and $\tilde{\eta}_t \le \frac{1}{90\lambda^2 L_F}$. Following the same procedure as the proof of Theorem 13, we can obtain

$$\begin{aligned}&\mathbb{E}\left[F(\tilde{\mathbf{w}}_{t+1}) - F(\tilde{\mathbf{w}}_t)\right] \\ &\le -\frac{\tilde{\eta}_t}{4}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] + \frac{\tilde{\eta}_t^3\delta_t^2}{\beta^2}\underbrace{40L_F^2\lambda^2}_{=:C_5} + \frac{\tilde{\eta}_t^3}{\beta^2}\underbrace{84L_F^2\sigma_{F,2}^2}_{=:C_6} + \tilde{\eta}_t^2\underbrace{5L_F\sigma_{F,2}^2\frac{N/S-1}{N-1}}_{=:C_7} + \tilde{\eta}_t\delta_t^2\underbrace{2\lambda^2}_{=:C_8}.\end{aligned}$$

Recall that $\delta_t^2 \le \frac{(1+s)^6(\gamma_f^2 D+1)}{45((1-s)^4\lambda-(1+s)^2L)^2\lambda^2 L_F\sqrt{t+1}} = C_9\tilde{\eta}_t$ with $C_9 := \frac{2(1+s)^6(\gamma_f^2 D+1)}{((1-s)^4\lambda-(1+s)^2L)^2} = \mathcal{O}\left(\frac{1+\gamma_f^2}{\lambda^2}\right)$. Summing from $t=0$ to $T-1$ yields

$$\mathbb{E}\left[F(\tilde{\mathbf{w}}_T) - F(\tilde{\mathbf{w}}_0)\right] \le -\sum_{t=0}^{T-1}\frac{\tilde{\eta}_t}{4}\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] + \frac{C_5C_9}{\beta^2}\sum_{t=0}^{T-1}\tilde{\eta}_t^4 + \frac{C_6}{\beta^2}\sum_{t=0}^{T-1}\tilde{\eta}_t^3 + (C_7+C_8C_9)\sum_{t=0}^{T-1}\tilde{\eta}_t^2.$$

From the definition of α_t and Δ_F, rearranging and dividing both sides by $\sum_{t=0}^{T-1}\tilde{\eta}_t/4$, we obtain

$$\sum_{t=0}^{T-1}\alpha_t\mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] \le \frac{4\Delta_F}{\sum_{t=0}^{T-1}\tilde{\eta}_t} + \frac{4C_5C_9}{\beta^2}\frac{\sum_{t=0}^{T-1}\tilde{\eta}_t^4}{\sum_{t=0}^{T-1}\tilde{\eta}_t} + \frac{4C_6}{\beta^2}\frac{\sum_{t=0}^{T-1}\tilde{\eta}_t^3}{\sum_{t=0}^{T-1}\tilde{\eta}_t} + 4(C_7+C_8C_9)\frac{\sum_{t=0}^{T-1}\tilde{\eta}_t^2}{\sum_{t=0}^{T-1}\tilde{\eta}_t}. \tag{33}$$

With $\tilde{\eta}_t = \frac{1}{90\lambda^2 L_F\sqrt{t+1}}$, we have

$$\sum_{t=0}^{T-1}\tilde{\eta}_t \ge \frac{1}{90\lambda^2 L_F}\int_1^{T+1}\frac{\mathrm{d}t}{\sqrt{t}} = \frac{(\sqrt{T+1}-1)}{45\lambda^2 L_F},$$

$$\begin{aligned}
\sum_{t=0}^{T-1} \tilde{\eta}_t^2 &\le \frac{1}{(90\lambda^2 L_F)^2}\left(1 + \int_1^T \frac{\mathrm{d}t}{t}\right) = \frac{\ln T + 1}{(90\lambda^2 L_F)^2}, \\
\sum_{t=0}^{T-1} \tilde{\eta}_t^3 &\le \frac{1}{(90\lambda^2 L_F)^3}\left(1 + \int_1^T \frac{\mathrm{d}t}{t^{3/2}}\right) \le \frac{3}{(90\lambda^2 L_F)^3}, \\
\sum_{t=0}^{T-1} \tilde{\eta}_t^4 &\le \frac{1}{(90\lambda^2 L_F)^4}\left(1 + \int_1^T \frac{\mathrm{d}t}{t^2}\right) = \frac{2}{(90\lambda^2 L_F)^4}.
\end{aligned}$$

It follows that

$$\begin{aligned}
&\sum_{t=0}^{T-1} \alpha_t \mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] \\
&\le \frac{180\lambda^2 L_F \Delta_F}{\sqrt{T+1}-1} + \frac{4C_5 C_9}{(90\lambda^2 L_F)^3 \beta^2(\sqrt{T+1}-1)} \\
&\quad + \frac{6C_6}{(90\lambda^2 L_F)^2\beta^2(\sqrt{T+1}-1)} + \frac{2(C_7 + C_8 C_9)(\ln T + 1)}{90\lambda^2 L_F(\sqrt{T+1}-1)} \\
&= \mathcal{O}\left(\frac{\lambda^2 L_F^2 \Delta_F}{\sqrt{T}} + \frac{1+\gamma_f^2}{\lambda^6 L_F^3 \beta^2 \sqrt{T}} + \frac{\sigma_F^2}{\lambda^4 \beta^2 \sqrt{T}}\right) + \mathcal{O}\left(\left(\frac{(N/S-1)\sigma_F^2}{\lambda^2 N} + \frac{1+\gamma_f^2}{\lambda^2 L_F}\right)\frac{\ln T}{\sqrt{T}}\right).
\end{aligned}$$

Note that $L_F = \lambda > 1$. Rearranging the terms gives the desired result.

When $S = N$, we choose $\eta_t = \frac{1}{90\beta^{1/3}R\lambda^2 L_F(t+1)^{1/3}}$ (implying $\tilde{\eta}_t = \frac{\beta^{2/3}}{90\lambda^2 L_F(t+1)^{1/3}}$) , $\nu_t = \frac{1}{90\lambda^2 L_F \beta^{2/3}(t+1)^{2/3}}$, $D_t = \left\lceil \frac{90\lambda^2 L_F \beta^{2/3}(t+1)^{2/3}}{D} \right\rceil$, then we have the following upper bound $\delta_t^2 \le \frac{(1+s)^6(\gamma_f^2 D+1)}{45((1-s)^4\lambda-(1+s)^2 L)^2\lambda^2 L_F(t+1)^{2/3}} = \tilde{C}_9 \frac{\tilde{\eta}_t^2}{\beta^2}$ with $\tilde{C}_9 := \frac{180\lambda^2 L_F(1+s)^6(\gamma_f^2 D+1)}{((1-s)^4\lambda-(1+s)^2 L)^2} = \mathcal{O}\left(L_F(1+\gamma_f^2)\right)$. Then (33) becomes

$$\sum_{t=0}^{T-1} \alpha_t \mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] \le \frac{4\Delta_F}{\sum_{t=0}^{T-1}\tilde{\eta}_t} + \frac{4C_5\tilde{C}_9}{\beta^4}\frac{\sum_{t=0}^{T-1}\tilde{\eta}_t^5}{\sum_{t=0}^{T-1}\tilde{\eta}_t} + \frac{4(C_6 + C_8\tilde{C}_9)}{\beta^2}\frac{\sum_{t=0}^{T-1}\tilde{\eta}_t^3}{\sum_{t=0}^{T-1}\tilde{\eta}_t}.$$

And we have

$$\begin{aligned}
\sum_{t=0}^{T-1} \tilde{\eta}_t &\ge \frac{\beta^{2/3}}{90\lambda^2 L_F}\int_1^{T+1} \frac{\mathrm{d}t}{t^{1/3}} = \frac{\beta^{2/3}[(T+1)^{2/3}-1]}{60\lambda^2 L_F}, \\
\sum_{t=0}^{T-1} \tilde{\eta}_t^3 &\le \frac{\beta^2}{(90\lambda^2 L_F)^3}\left(1 + \int_1^T \frac{\mathrm{d}t}{t^{3/2}}\right) \le \frac{\beta^2(\ln T + 1)}{(90\lambda^2 L_F)^3}, \\
\sum_{t=0}^{T-1} \tilde{\eta}_t^5 &\le \frac{\beta^{10/3}}{(90\lambda^2 L_F)^5}\left(1 + \int_1^T \frac{\mathrm{d}t}{t^{5/3}}\right) = \frac{3\beta^{10/3}/2}{(90\lambda^2 L_F)^5}.
\end{aligned}$$

It follows that

$$\sum_{t=0}^{T-1} \alpha_t \mathbb{E}\left[\|\nabla F(\tilde{\mathbf{w}}_t)\|_2^2\right] \le \frac{240\lambda^2 L_F \Delta_F}{\beta^{2/3}[(T+1)^{2/3}-1]} + \frac{4C_5\tilde{C}_9}{(90\lambda^2 L_F)^4\beta^{2/3}[T+1)^{2/3}-1]}$$

$$
\begin{aligned}
&+ \frac{8(C_6 + C_8\tilde{C}_9)(\ln T + 1)/3}{(90\lambda^2 L_F)^2 \beta^{2/3}[T + 1)^{2/3} - 1]} \\
&\leq \mathcal{O}\left(\frac{\lambda^2 L_F \Delta_F}{\beta^{2/3}T^{2/3}} + \frac{1 + \gamma_f^2}{\lambda^6 L_F \beta^{2/3}T^{2/3}} + \frac{\ln T (L_F^2 \sigma_{F,2}^2 + \lambda^2 L_F (1 + \gamma_f^2))}{\lambda^4 L_F^2 \beta^{2/3} T^{2/3}}\right) \\
&= \mathcal{O}\left(\frac{\lambda^2 L_F \Delta_F}{\beta^{2/3}T^{2/3}} + \frac{\sigma_{F,2}^2 \ln T}{\lambda^4 \beta^{2/3}T^{2/3}} + \frac{(1 + \gamma_f^2)(\lambda^{-4} + \ln T)}{\lambda^2 L_F \beta^{2/3} T^{2/3}}\right).
\end{aligned}
$$

This completes the proof. ■

Appendix B. Federated Linear Regression

In this section, we consider a federated linear regression model, which is different from that in Li et al. (2021b).

Suppose that the true parameter on client k is $\mathbf{w}_k$, there are n samples on each client and the covariate on client k is $\{\xi_{k,i}\}_{i=1}^n$ and fixed. The observations are generated by $y_{k,i} = \xi_{k,i}^\top \mathbf{w}_k + z_{k,i}$ where the noises $z_{k,i}$ are i.i.d. and distributed as $\mathcal{N}(0, \sigma^2)$. Then the loss on client k is $f_k(\mathbf{x}_k) = \frac{1}{2n}\sum_{i=1}^n (y_{k,i} - \xi_{k,i}^\top \mathbf{x}_k)^2$

Li et al. (2021b) focused on a Bayesian framework where the true parameters $\mathbf{w}_k$ are drawn from a Gaussian distribution and the mean of this Gaussian distribution is drawn from the non-informative prior, while we treat $\mathbf{w}_k$ as fixed vectors. We compare the performance of `local` (pure local training), `FedAvg` (McMahan et al., 2017), `pFedMe` (Dinh et al., 2020), `Ditto` (Li et al., 2021b) and our method `lp-proj-2` in terms of test losses, robustness and fairness.

B.1 Solutions of Different Methods

In this subsection, we derive the solutions of different methods. Let $\mathbf{\Xi}_k = (\xi_{k,1}, \xi_{k,2}, \ldots, \xi_{k,n})^\top$ and $\mathbf{y}_k = (y_{k,1}, y_{k,2}, \ldots, y_{k,n})^\top$. Then the loss on client k can be rewritten as $f_k(\mathbf{x}_k) = \frac{1}{2n}\left\|\mathbf{\Xi}_k \mathbf{x}_k - \mathbf{y}_k\right\|_2^2$. Suppose $\mathrm{rank}(\mathbf{\Xi}_k) = d$. The least-square estimator of $\mathbf{w}_k$ is

$$
\hat{\mathbf{w}}_k = (\mathbf{\Xi}_k^\top \mathbf{\Xi}_k)^{-1} \mathbf{\Xi}_k^\top \mathbf{y}_k.
$$

`local` For pure local training, the solution on client k is defined as follows $\mathbf{w}_k^{\mathrm{loc}} = \mathrm{argmin}_{\mathbf{x}_k \in \mathbb{R}^d} f_k(\mathbf{x}_k) = \hat{\mathbf{w}}_k$.

`FedAvg` For `FedAvg`, the solution is defined as $\mathbf{w}^{\mathrm{Avg}} = \mathrm{argmin}_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{N}\sum_{k=1}^N f_k(\mathbf{w})$. One can check that $\mathbf{w}^{\mathrm{Avg}} = \left(\sum_{k=1}^N \mathbf{\Xi}_k^\top \mathbf{\Xi}_k\right)^{-1} \sum_{k=1}^N \mathbf{\Xi}_k^\top \mathbf{y}_k = \left(\sum_{k=1}^N \mathbf{\Xi}_k^\top \mathbf{\Xi}_k\right)^{-1} \sum_{k=1}^N \mathbf{\Xi}_k^\top \mathbf{\Xi}_k \hat{\mathbf{w}}_k$.

`pFedMe` `pFedMe` corresponds to our method with $\boldsymbol{P} = \boldsymbol{I}_d$. Then the optimization problem is $\min_{\mathbf{w} \in \mathbb{R}^d} F(\mathbf{w}) = \frac{1}{N}\sum_{k=1}^N F_k(\mathbf{w})$ where $F_k(\mathbf{w}) = \min_{\mathbf{x}_k \in \mathbb{R}^d}\{f_k(\mathbf{x}_k) + \frac{\lambda}{2}\left\|\mathbf{w} - \mathbf{x}_k\right\|_2^2\}$. The solution of the global model is defined as $\mathbf{w}^{\mathrm{Me}} = \mathrm{argmin}_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{N}\sum_{k=1}^N F_k(\mathbf{w})$ and the solution of the local model is defined as $\mathbf{x}_k^{\mathrm{Me}} = \mathrm{argmin}_{\mathbf{x}_k \in \mathbb{R}^d}\left\{f_k(\mathbf{x}_k) + \frac{\lambda}{2}\left\|\mathbf{w}^{\mathrm{Me}} - \mathbf{x}_k\right\|_2^2\right\}$.

Now we give the explicit forms of $\mathbf{w}^{\mathrm{Me}}$ and $\mathbf{x}_k^{\mathrm{Me}}$. Define $\hat{\mathbf{x}}_k(\mathbf{w}) := \mathrm{argmin}_{\mathbf{x}_k \in \mathbb{R}^d}\{f_k(\mathbf{x}_k) + \frac{\lambda}{2}\left\|\mathbf{w} - \mathbf{x}_k\right\|_2^2\}$. It is easy to check $\hat{\mathbf{x}}_k(\mathbf{w}) = (\mathbf{\Xi}_k^\top \mathbf{\Xi}_k / n + \lambda \boldsymbol{I}_d)^{-1}(\mathbf{\Xi}_k^\top \mathbf{y}_k / n + \lambda \mathbf{w})$. Then we have

$$\begin{aligned} F_k(\mathbf{w}) &= f_k(\hat{\mathbf{x}}_k(\mathbf{w})) + \frac{\lambda}{2}\left\|\mathbf{w} - \hat{\mathbf{x}}_k(\mathbf{w})\right\|_2^2 \\ &= -\frac{1}{2}\left(\frac{\boldsymbol{\Xi}_k^\top \mathbf{y}_k}{n} + \lambda\mathbf{w}\right)^\top \left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1} \left(\frac{\boldsymbol{\Xi}_k^\top \mathbf{y}_k}{n} + \lambda\mathbf{w}\right) + \frac{\lambda}{2}\left\|\mathbf{w}\right\|_2^2 + \frac{\left\|\mathbf{y}_k\right\|_2^2}{2n} \\ &= \frac{\lambda\mathbf{w}^\top}{2}\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1}\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k \mathbf{w}}{n}\right) - \lambda\mathbf{w}^\top\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top \mathbf{y}_k}{n} \\ &\quad + \frac{\left\|\mathbf{y}_k\right\|_2^2}{2n} - \frac{\mathbf{y}_k^\top \boldsymbol{\Xi}_k}{2n}\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top \mathbf{y}_k}{n}. \end{aligned}$$

It follows that

$$\begin{aligned} F(\mathbf{w}) = \frac{1}{N}\sum_{k=1}^{N} &\frac{\lambda\mathbf{w}^\top}{2}\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1}\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k \mathbf{w}}{n}\right) \\ &- \frac{1}{N}\sum_{k=1}^{N}\lambda\mathbf{w}^\top\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top \mathbf{y}_k}{n} + C_0, \end{aligned}$$

where C_0 is a constant number. Then $\mathbf{w}^{\mathrm{Me}}$ is the solution to

$$\begin{aligned} \frac{1}{2}\sum_{k=1}^{N}\left[\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n}\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1}\right]\mathbf{w} & \\ = \sum_{k=1}^{N}\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top \mathbf{y}_k}{n}. & \qquad (34) \end{aligned}$$

By the Sherman–Morrison–Woodbury formula, we have

$$\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1} = \frac{\boldsymbol{I}_d}{\lambda} - \frac{\boldsymbol{\Xi}_k^\top}{\lambda}\left(n\boldsymbol{I}_n + \frac{\boldsymbol{\Xi}_k\boldsymbol{\Xi}_k^\top}{\lambda}\right)^{-1}\frac{\boldsymbol{\Xi}_k}{\lambda}.$$

It follows that

$$\begin{aligned} \left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} &= \frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{\lambda n} - \frac{\boldsymbol{\Xi}_k^\top}{\lambda}\left(n\boldsymbol{I}_n + \frac{\boldsymbol{\Xi}_k\boldsymbol{\Xi}_k^\top}{\lambda}\right)^{-1}\frac{\boldsymbol{\Xi}_k\boldsymbol{\Xi}_k^\top}{\lambda}\frac{\boldsymbol{\Xi}_k}{n} \\ &= \frac{\boldsymbol{\Xi}_k^\top}{\lambda}\left(n\boldsymbol{I}_n + \frac{\boldsymbol{\Xi}_k\boldsymbol{\Xi}_k^\top}{\lambda}\right)^{-1}\boldsymbol{\Xi}_k. \end{aligned}$$

Similarly, we can obtain $\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n}\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1} = \frac{\boldsymbol{\Xi}_k^\top}{\lambda}\left(n\boldsymbol{I}_n + \frac{\boldsymbol{\Xi}_k\boldsymbol{\Xi}_k^\top}{\lambda}\right)^{-1}\boldsymbol{\Xi}_k$. This implies that $\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} = \frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n}\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1}$. Thus, the solution to (34) is

$$\mathbf{w}^{\mathrm{Me}} = \left[\sum_{k=1}^{N}\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n}\right]^{-1}\left[\sum_{k=1}^{N}\left(\frac{\boldsymbol{\Xi}_k^\top \boldsymbol{\Xi}_k}{n} + \lambda \boldsymbol{I}_d\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top \mathbf{y}_k}{n}\right]$$

$$= \left[\sum_{k=1}^{N}\left(\frac{\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k}{n}+\lambda\boldsymbol{I}_d\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k}{n}\right]^{-1}\left[\sum_{k=1}^{N}\left(\frac{\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k}{n}+\lambda\boldsymbol{I}_d\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k}{n}\hat{\mathbf{w}}_k\right],$$

which can be seen as a weighted average of $\hat{\mathbf{w}}_k$ with weight $\left(\frac{\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k}{n}+\lambda\boldsymbol{I}_d\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k}{n}$. Then solution of the local model is $\mathbf{x}_k^{\text{Me}} = \hat{\mathbf{x}}_k(\mathbf{w}^{\text{Me}}) = (\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k/n+\lambda\boldsymbol{I}_d)^{-1}(\boldsymbol{\Xi}_k^\top\mathbf{y}_k/n+\lambda\mathbf{w}^{\text{Me}}) = (\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k/n+\lambda\boldsymbol{I}_d)^{-1}(\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k\hat{\mathbf{w}}_k/n+\lambda\mathbf{w}^{\text{Me}})$.

`Ditto` For `Ditto`, the solution of the global model is the same as that of `FedAvg`, i.e., $\mathbf{w}^{\text{Di}} = \operatorname{argmin}_{\mathbf{w}\in\mathbb{R}^d}\frac{1}{N}\sum_{k=1}^{N}f_k(\mathbf{w}) = \left(\sum_{k=1}^{N}\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k\right)^{-1}\sum_{k=1}^{N}\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k\hat{\mathbf{w}}_k$. The solution of the local model is defined as $\mathbf{x}_k^{\text{Di}} = \operatorname{argmin}_{\mathbf{x}_k\in\mathbb{R}^d}\left\{f_k(\mathbf{x}_k)+\frac{\lambda}{2}\left\|\mathbf{w}^{\text{Di}}-\mathbf{x}_k\right\|_2^2\right\} = (\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k/n+\lambda\boldsymbol{I}_d)^{-1}(\boldsymbol{\Xi}_k^\top\mathbf{y}_k/n+\lambda\mathbf{w}^{\text{Di}}) = (\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k/n+\lambda\boldsymbol{I}_d)^{-1}(\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k\hat{\mathbf{w}}_k/n+\lambda\mathbf{w}^{\text{Di}})$.

`lp-proj-2` For our method `lp-proj-2`, the optimization problem is $\min_{\tilde{\mathbf{w}}\in\mathbb{R}^{d_{\text{sub}}}}F(\tilde{\mathbf{w}}) = \frac{1}{N}\sum_{k=1}^{N}F_k(\tilde{\mathbf{w}})$ where $F_k(\tilde{\mathbf{w}}) = \min_{\mathbf{x}_k\in\mathbb{R}^d}\{f_k(\mathbf{x}_k)+\frac{\lambda}{2}\|\tilde{\mathbf{w}}-\boldsymbol{P}\mathbf{x}_k\|_2^2\}$. The solution of the global model is defined as $\tilde{\mathbf{w}}^{\text{l2}} = \operatorname{argmin}_{\tilde{\mathbf{w}}\in\mathbb{R}^{d_{\text{sub}}}}\frac{1}{N}\sum_{k=1}^{N}F_k(\tilde{w})$ and the solution of the local solution is defined as $\mathbf{x}_k^{\text{l2}} = \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^d}\left\{f_k(\mathbf{x}_k)+\frac{\lambda}{2}\left\|\tilde{\mathbf{w}}^{\text{l2}}-\boldsymbol{P}\mathbf{x}_k\right\|_2^2\right\}$. Let $\check{\mathbf{x}}(\tilde{\mathbf{w}}) := \operatorname{argmin}_{\mathbf{x}_k\in\mathbb{R}^d}\left\{f_k(\mathbf{x}_k)+\frac{\lambda}{2}\|\tilde{\mathbf{w}}-\boldsymbol{P}\mathbf{x}_k\|_2^2\right\}$. It is easy to check the following equation $\check{\mathbf{x}}(\tilde{\mathbf{w}}) = (\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k/n+\lambda\boldsymbol{P}^\top\boldsymbol{P})^{-1}(\boldsymbol{\Xi}_k^\top\mathbf{y}_k/n+\lambda\boldsymbol{P}^\top\tilde{\mathbf{w}})$. It follows that

$$\begin{aligned}
F_k(\tilde{\mathbf{w}}) &= f_k(\check{\mathbf{x}}_k(\tilde{\mathbf{w}}))+\frac{\lambda}{2}\|\tilde{\mathbf{w}}-\check{\mathbf{x}}_k(\tilde{\mathbf{w}})\|_2^2\\
&= -\frac{1}{2}\left(\frac{\boldsymbol{\Xi}_k^\top\mathbf{y}_k}{n}+\lambda\boldsymbol{P}^\top\tilde{\mathbf{w}}\right)^\top\left(\frac{\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k}{n}+\lambda\boldsymbol{P}^\top\boldsymbol{P}\right)^{-1}\left(\frac{\boldsymbol{\Xi}_k^\top\mathbf{y}_k}{n}+\lambda\boldsymbol{P}^\top\tilde{\mathbf{w}}\right)+\frac{\lambda}{2}\|\tilde{\mathbf{w}}\|_2^2+\frac{\|\mathbf{y}_k\|_2^2}{2n}\\
&= \frac{\lambda}{2}\|\tilde{\mathbf{w}}\|_2^2-\frac{\lambda^2}{2}\tilde{\mathbf{w}}^\top\boldsymbol{P}\left(\frac{\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k}{n}+\lambda\boldsymbol{P}^\top\boldsymbol{P}\right)^{-1}\boldsymbol{P}^\top\tilde{\mathbf{w}}-\lambda\tilde{\mathbf{w}}^\top\boldsymbol{P}\left(\frac{\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k}{n}+\lambda\boldsymbol{P}^\top\boldsymbol{P}\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top\mathbf{y}_k}{n}\\
&\quad+\frac{\|\mathbf{y}_k\|_2^2}{2n}-\frac{\mathbf{y}_k^\top\boldsymbol{\Xi}_k}{2n}\left(\frac{\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k}{n}+\lambda\boldsymbol{P}^\top\boldsymbol{P}\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top\mathbf{y}_k}{n}.
\end{aligned}$$

Then we can obtain the expression of $F(\tilde{\mathbf{w}})$. However, for the general $\boldsymbol{\Xi}_k$, it is difficult to obtain a concise expression of the minimizer of $F(\tilde{\mathbf{w}})$. To make the calculations clean, we assume $\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k = nb_k\boldsymbol{I}_d$. Then the solutions of other methods can be simplified as

- `FedAvg`: $\mathbf{w}^{\text{Avg}} = \frac{\sum_{k=1}^{N}b_k\hat{\mathbf{w}}_k}{\sum_{k=1}^{N}b_k}$.
- `pFedMe`: $\mathbf{w}^{\text{Me}} = \frac{\sum_{k=1}^{N}b_k\hat{\mathbf{w}}_k/(b_k+\lambda)}{\sum_{k=1}^{N}b_k/(b_k+\lambda)}$ and $\mathbf{x}_k^{\text{Me}} = \frac{b_k\hat{\mathbf{w}}_k+\lambda\mathbf{w}^{\text{Me}}}{b_k+\lambda}$.
- `Ditto`: $\mathbf{w}^{\text{Di}} = \frac{\sum_{k=1}^{N}b_k\hat{\mathbf{w}}_k}{\sum_{k=1}^{N}b_k}$ and $\mathbf{x}_k^{\text{Di}} = \frac{b_k\hat{\mathbf{w}}_k+\lambda\mathbf{w}^{\text{Di}}}{b_k+\lambda}$.

Meanwhile, for `lp-proj-2`, without loss of generalization, we can assume $\boldsymbol{P} = \boldsymbol{P}_0 := (\mathbf{e}_1,\mathbf{e}_2,\ldots,\mathbf{e}_{d_{\text{sub}}})^\top$, where $\mathbf{e}_i$ is the unit vector in $\mathbb{R}^d$ with the i-th element equal to 1 and other elements equal to 0. Otherwise, we can find a orthogonal matrix $\boldsymbol{Q}$ such that $\boldsymbol{P} = \boldsymbol{P}_0\boldsymbol{Q}$. Then we have

$$f_k(\mathbf{x}_k)+\frac{\lambda}{2}\|\tilde{\mathbf{w}}-\boldsymbol{P}\mathbf{x}_k\|_2^2 = \frac{1}{2n}\|\boldsymbol{\Xi}_k\mathbf{x}_k-\mathbf{y}_k\|_2^2+\frac{\lambda}{2}\|\tilde{\mathbf{w}}-\boldsymbol{P}\mathbf{x}_k\|_2^2$$

$$
\begin{aligned}
&= \frac{1}{2n}\left\|\boldsymbol{\Xi}_k \boldsymbol{Q}^\top \boldsymbol{Q}\mathbf{x}_k - \mathbf{y}_k\right\|_2^2 + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}} - \boldsymbol{P}_0\boldsymbol{Q}\mathbf{x}_k\right\|_2^2 \\
&= \frac{1}{2n}\left\|\tilde{\boldsymbol{\Xi}}_k\tilde{\mathbf{x}}_k - \mathbf{y}_k\right\|_2^2 + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}} - \boldsymbol{P}_0\tilde{\mathbf{x}}_k\right\|_2^2
\end{aligned}
$$

where $\tilde{\mathbf{x}}_k = \boldsymbol{Q}\mathbf{x}_k$ and $\tilde{\boldsymbol{\Xi}}_k = \boldsymbol{\Xi}_k\boldsymbol{Q}^\top$. Note that $\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k = nb_k\boldsymbol{I}_d$ implies $\tilde{\boldsymbol{\Xi}}_k^\top\tilde{\boldsymbol{\Xi}}_k = \boldsymbol{Q}\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k\boldsymbol{Q}^\top = nb_k\boldsymbol{I}_d$. After reparametrization, we return to the special case $\boldsymbol{P} = \boldsymbol{P}_0$.

Now we have

$$
\begin{aligned}
F_k(\tilde{\mathbf{w}}) &= \frac{\lambda b_k}{2(b_k+\lambda)}\|\tilde{\mathbf{w}}\|_2^2 - \frac{\lambda}{(b_k+\lambda)n}\tilde{\mathbf{w}}^\top\boldsymbol{P}_0\boldsymbol{\Xi}_k^\top\mathbf{y}_k + \frac{\|\mathbf{y}_k\|_2^2}{2n} - \frac{\mathbf{y}_k^\top\boldsymbol{\Xi}_k}{2n}\left(\frac{\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k}{n} + \lambda\boldsymbol{P}_0^\top\boldsymbol{P}_0\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top\mathbf{y}_k}{n} \\
&= \frac{\lambda b_k}{2(b_k+\lambda)}\|\tilde{\mathbf{w}}\|_2^2 - \frac{\lambda b_k}{b_k+\lambda}\tilde{\mathbf{w}}^\top\hat{\mathbf{w}}_{k,1} + \frac{\|\mathbf{y}_k\|_2^2}{2n} - \frac{\mathbf{y}_k^\top\boldsymbol{\Xi}_k}{2n}\left(\frac{\boldsymbol{\Xi}_k^\top\boldsymbol{\Xi}_k}{n} + \lambda\boldsymbol{P}_0^\top\boldsymbol{P}_0\right)^{-1}\frac{\boldsymbol{\Xi}_k^\top\mathbf{y}_k}{n},
\end{aligned}
$$

where $\hat{\mathbf{w}}_{k,1} = \boldsymbol{P}_0\hat{\mathbf{w}}_k$ is the first d_{sub} elements of $\hat{\mathbf{w}}_k$. Then we obtain

$$
F_k(\tilde{\mathbf{w}}) = \frac{1}{N}\sum_{k=1}^{N}\frac{\lambda b_k}{2(b_k+\lambda)}\|\tilde{\mathbf{w}}\|_2^2 - \frac{1}{N}\sum_{k=1}^{N}\frac{\lambda b_k}{b_k+\lambda}\tilde{\mathbf{w}}^\top\hat{\mathbf{w}}_{k,1} + C_1,
$$

where C_1 is a constant. Thus the solution of the global model is $\tilde{\mathbf{w}}^{\text{l2}} = \frac{\sum_{k=1}^N b_k\hat{\mathbf{w}}_{k,1}/(b_k+\lambda)}{\sum_{k=1}^N b_k/(b_k+\lambda)}$, and the solution of the local model is $\mathbf{x}_k^{\text{l2}} = \check{\mathbf{x}}_k(\tilde{\mathbf{w}}^{\text{l2}}) = \begin{pmatrix}(b_k\hat{\mathbf{w}}_{k,1} + \lambda\tilde{\mathbf{w}}^{\text{l2}})/(b_k+\lambda) \\ \hat{\mathbf{w}}_{k,2}\end{pmatrix}$, where $\hat{\mathbf{w}}_{k,2}$ is the last $d - d_{\text{sub}}$ elements of $\hat{\mathbf{w}}_k$.

To summarize, the solutions of different models are listed as follows.

- `local`: $\mathbf{w}_k^{\text{loc}} = \hat{\mathbf{w}}_k$.
- `FedAvg`: $\mathbf{w}^{\text{Avg}} = \frac{\sum_{k=1}^N b_k\hat{\mathbf{w}}_k}{\sum_{k=1}^N b_k}$.
- `pFedMe`: $\mathbf{w}^{\text{Me}} = \frac{\sum_{k=1}^N b_k\hat{\mathbf{w}}_k/(b_k+\lambda)}{\sum_{k=1}^N b_k/(b_k+\lambda)}$ and $\mathbf{x}_k^{\text{Me}} = \frac{b_k\hat{\mathbf{w}}_k + \lambda\mathbf{w}^{\text{Me}}}{b_k+\lambda}$.
- `Ditto`: $\mathbf{w}^{\text{Di}} = \frac{\sum_{k=1}^N b_k\hat{\mathbf{w}}_k}{\sum_{k=1}^N b_k}$ and $\mathbf{x}_k^{\text{Di}} = \frac{b_k\hat{\mathbf{w}}_k + \lambda\mathbf{w}^{\text{Di}}}{b_k+\lambda}$.
- `lp-proj-2`: $\tilde{\mathbf{w}}^{\text{l2}} = \frac{\sum_{k=1}^N b_k\hat{\mathbf{w}}_{k,1}/(b_k+\lambda)}{\sum_{k=1}^N b_k/(b_k+\lambda)}$ and $\mathbf{x}_k^{\text{l2}} = \check{\mathbf{x}}_k(\tilde{\mathbf{w}}^{\text{l2}}) = \begin{pmatrix}(b_k\hat{\mathbf{w}}_{k,1} + \lambda\tilde{\mathbf{w}}^{\text{l2}})/(b_k+\lambda) \\ \hat{\mathbf{w}}_{k,2}\end{pmatrix}$.

Note that $\mathbf{x}_k^{\text{Me}}$ and $\mathbf{x}_k^{\text{Di}}$ are both the weighted average of $\mathbf{w}^{\text{Me}}/\mathbf{w}^{\text{Di}}$ and $\hat{\mathbf{w}}_k$ with the same weight. $\mathbf{w}^{\text{Me}}$ and $\mathbf{w}^{\text{Di}}$ are weighted average of $\hat{\mathbf{w}}_k$ with different weights. If $\lambda = 0$, we have $\mathbf{w}^{\text{Me}} = \frac{1}{N}\sum_{k=1}^N\hat{\mathbf{w}}_k$. If $\lambda \to \infty$, we have $\hat{\mathbf{w}}^{\text{Me}} \to \mathbf{w}^{\text{Avg}}$. Thus, the weight of `pFedMe` is more uniform than that of `FedAvg`. In Section 4.2, we assume $b_k = b$. This is reasonable since we often normalize the data. Then we have $\mathbf{w}^{\text{Avg}} = \mathbf{w}^{\text{Me}} = \mathbf{w}^{\text{Di}} = \frac{1}{N}\sum_{k=1}^N\hat{\mathbf{w}}_k$ and $\mathbf{x}_k^{\text{Me}} = \mathbf{x}_k^{\text{Di}}$.

Moreover, `lp-proj-2` can be viewed as a interpolation of local and pFedMe. The first d_{sub} dimensions of $\mathbf{x}_k^{\text{l2}}$ equal to those of $\mathbf{x}_k^{\text{Me}}$ and the last $d - d_{\text{sub}}$ dimensions equal to those of $\mathbf{w}_k^{\text{loc}}$.

B.2 Test Loss

In this subsection, we compute the test losses of different methods. From now on, we always assume $b_k = b$ to make calculations clean.

Recall that the data set on client k is $(\boldsymbol{\Xi}_k, \mathbf{y}_k)$, where $\boldsymbol{\Xi}_k$ is fixed and $\mathbf{y}_k$ follows Gaussian distribution $\mathcal{N}(\boldsymbol{\Xi}_k \mathbf{w}_k, \sigma^2 \boldsymbol{I}_n)$. Then the data heterogeneity across clients only lies in the heterogeneity of $\mathbf{w}_k$. We can obtain the distribution of the solutions of different methods.

Let $\bar{\mathbf{w}} = \frac{\sum_{k=1}^N \mathbf{w}_k}{N}$. We have

- `local`: $\mathbf{w}_k^{\mathrm{loc}} \sim \mathcal{N}\left(\mathbf{w}_k, \frac{\sigma^2}{bn}\boldsymbol{I}_d\right)$.
- `FedAvg`: $\mathbf{w}^{\mathrm{Avg}} \sim \mathcal{N}\left(\bar{\mathbf{w}}, \frac{\sigma^2}{bNn}\boldsymbol{I}_d\right)$.
- `pFedMe`: $\mathbf{w}^{\mathrm{Me}} \sim \mathcal{N}\left(\bar{\mathbf{w}}, \frac{\sigma^2}{bNn}\boldsymbol{I}_d\right)$ and $\mathbf{x}_k^{\mathrm{Me}} \sim \mathcal{N}\left(\frac{b\mathbf{w}_k + \lambda\bar{\mathbf{w}}}{b+\lambda}, \frac{\left(b^2 + \frac{2b\lambda}{N}\right)\frac{\sigma^2}{bn} + \frac{\lambda^2}{N}\cdot\frac{\sigma^2}{bn}}{(b_k+\lambda)^2}\boldsymbol{I}_d\right)$.
- `Ditto`: $\mathbf{w}^{\mathrm{Di}} = \mathbf{w}^{\mathrm{Me}}$ and $\mathbf{x}_k^{\mathrm{Di}} = \mathbf{x}_k^{\mathrm{Me}}$.
- `lp-proj-2`: $\tilde{\mathbf{w}}^{\mathrm{l2}} \sim \mathcal{N}\left(\bar{\mathbf{w}}_{\cdot,1}, \frac{\sigma^2}{bNn}\boldsymbol{I}_{d_{\mathrm{sub}}}\right)$ and

$$\mathbf{x}_k^{\mathrm{l2}} \sim \mathcal{N}\left(\begin{pmatrix} \frac{b\mathbf{w}_{k,1} + \lambda\bar{\mathbf{w}}_{\cdot,1}}{b_k+\lambda} \\ \mathbf{w}_{k,2} \end{pmatrix}, \begin{pmatrix} \frac{\left(b^2 + \frac{2b\lambda}{N}\right)\frac{\sigma^2}{bn} + \frac{\lambda^2}{N^2}\cdot\frac{\sigma^2}{bn}}{(b_k+\lambda)^2}\boldsymbol{I}_{d_{\mathrm{sub}}} & \\ & \frac{\sigma^2}{bn}\boldsymbol{I}_{d-d_{\mathrm{sub}}} \end{pmatrix}\right)$$

  where $\mathbf{w}_{k,1}$ is the first d elements of $\mathbf{w}_k$, $\mathbf{w}_{k,2}$ is the last $d - d_{\mathrm{sub}}$ elements of $\mathbf{w}_k$ and $\bar{\mathbf{w}}_{\cdot,1}$ is the first k elements of $\bar{\mathbf{w}}$.

Since $\boldsymbol{\Xi}_k$ is fixed, we assume the test data is $(\boldsymbol{\Xi}_k, \mathbf{y}'_k)$ where $\mathbf{y}'_k = \boldsymbol{\Xi}_k\mathbf{w}_k + \mathbf{z}'_k$ with $\mathbf{z}'_k \sim \mathcal{N}(\mathbf{0}_n, \sigma^2\boldsymbol{I}_n)$ independent of $\mathbf{z}_k$. Then the test loss on client k is defined as

$$\begin{aligned} f_k^{\mathrm{te}}(\mathbf{x}_k) &= \frac{1}{2n}\mathbb{E}\left\|\boldsymbol{\Xi}_k\mathbf{x}_k - \mathbf{y}'_k\right\|_2^2 \\ &= \frac{1}{2n}\mathbb{E}\left\|\boldsymbol{\Xi}_k\mathbf{x}_k - (\boldsymbol{\Xi}_k\mathbf{w}_k + \mathbf{z}'_k)\right\|_2^2 \\ &= \frac{\sigma^2}{2} + \frac{1}{2n}\mathbb{E}\left\|\boldsymbol{\Xi}_k(\mathbf{x}_k - \mathbf{w}_k)\right\|_2^2 \\ &= \frac{\sigma^2}{2} + \frac{b}{2}\mathbb{E}\left\|\mathbf{x}_k - \mathbf{w}_k\right\|_2^2 \\ &= \frac{\sigma^2}{2} + \frac{b}{2}\operatorname{tr}(\mathsf{var}(\mathbf{x}_k)) + \frac{b}{2}\left\|\mathbb{E}\mathbf{x}_k - \mathbf{w}_k\right\|_2^2. \end{aligned} \tag{35}$$

and the averaged test loss is

$$\frac{1}{N}\sum_{k=1}^N f_k^{\mathrm{te}}(\mathbf{x}_k) = \frac{\sigma^2}{2} + \frac{b}{2N}\sum_{k=1}^N \operatorname{tr}(\mathsf{var}(\mathbf{x}_k)) + \frac{b}{2N}\sum_{k=1}^N \left\|\mathbb{E}\mathbf{x}_k - \mathbf{w}_k\right\|_2^2.$$

Then we can compute the test losses for different methods. Since the solutions of `Ditto` and `pFedMe` are the same, we omit the analysis for `Ditto`.

To make the calculation simple, we assume the heterogeneity in terms of $\mathbf{w}_k$ is uniform in all dimensions, that is

$$\frac{1}{dN}\sum_{k=1}^{N}\|\bar{\mathbf{w}}-\mathbf{w}_k\|_2^2=\frac{1}{d_{\text{sub}}N}\sum_{k=1}^{N}\|\bar{\mathbf{w}}_{\cdot,1}-\mathbf{w}_{k,1}\|_2^2:=\Sigma. \tag{36}$$

Then we have

$$\begin{aligned}
L^{\text{loc}}&=\frac{1}{N}\sum_{k=1}^{N}f_k^{\text{te}}(\mathbf{w}_k^{\text{loc}})=\frac{\sigma^2}{2}+\frac{\sigma^2 d}{2n},\\
L^{\text{Avg}}&=\frac{1}{N}\sum_{k=1}^{N}f_k^{\text{te}}(\mathbf{w}^{\text{Avg}})=\frac{\sigma^2}{2}+\frac{\sigma^2 d}{2Nn}+\frac{b}{2N}\sum_{k=1}^{N}\|\bar{\mathbf{w}}-\mathbf{w}_k\|_2^2=\frac{\sigma^2}{2}+\frac{\sigma^2 d}{2Nn}+\frac{bd\Sigma}{2},\\
L^{\text{Me}}(\lambda)&=\frac{1}{N}\sum_{k=1}^{N}f_k^{\text{te}}(\mathbf{x}_k^{\text{Me}})=\frac{\sigma^2}{2}+\frac{b^2+\frac{2b\lambda}{N}+\frac{\lambda^2}{N}}{(b+\lambda)^2}\cdot\frac{\sigma^2 d}{2n}+\frac{b\lambda^2}{2N(b+\lambda)^2}\sum_{k=1}^{N}\|\bar{\mathbf{w}}-\mathbf{w}_k\|_2^2,\\
&=\frac{\sigma^2}{2}+\frac{b^2+\frac{2b\lambda}{N}+\frac{\lambda^2}{N}}{(b+\lambda)^2}\cdot\frac{\sigma^2 d}{2n}+\frac{b\lambda^2 d\Sigma}{2N(b+\lambda)^2},\\
L^{\text{l2}}(\lambda)&=\frac{1}{N}\sum_{k=1}^{N}f_k^{\text{te}}(\mathbf{x}_k^{\text{l2}})\\
&=\frac{\sigma^2}{2}+\frac{b^2+\frac{2b\lambda}{N}+\frac{\lambda^2}{N}}{(b+\lambda)^2}\cdot\frac{\sigma^2 d_{\text{sub}}}{2n}+\frac{\sigma^2(d-d_{\text{sub}})}{2n}+\frac{b\lambda^2}{2N(b+\lambda)^2}\sum_{k=1}^{N}\|\bar{\mathbf{w}}_{\cdot,1}-\mathbf{w}_{k,1}\|_2^2,\\
&=\frac{\sigma^2}{2}+\frac{b^2+\frac{2b\lambda}{N}+\frac{\lambda^2}{N}}{(b+\lambda)^2}\cdot\frac{\sigma^2 d_{\text{sub}}}{2n}+\frac{\sigma^2(d-d_{\text{sub}})}{2n}+\frac{b\lambda^2 d_{\text{sub}}\Sigma}{2N(b+\lambda)^2}.
\end{aligned}$$

Note that the test losses for `pFedMe` and `lp-proj-2` are functions of λ. To find the optimal λ, we could use the following lemma.

Lemma 29 *For any $\lambda \geq 0$, define $g(\lambda)=\frac{A\lambda^2+B\lambda+C}{(\lambda+b)^2}$ with $A, B, C, b > 0$. If $2Ab-B>0$ and $Bb-2C<0$, then* $\operatorname{argmin}_{\lambda\geq 0} g(\lambda)=\frac{2C-Bb}{2Ab-B}$.

Proof [Proof of Lemma 29] For convenience, define $\lambda_0=\frac{2C-Bb}{2Ab-B}$. One can check that $g'(\lambda)=\frac{(2Ab-B)\lambda+Bb-2C}{(\lambda+b)^3}$. Then for $\lambda\in[0,\lambda_0)$, $g'(\lambda)<0$; for $\lambda>\lambda_0$, $g'(\lambda)>0$. Consequently, $\lambda_0=\operatorname{argmin}_{\lambda\geq 0} g(\lambda)$. ■

For `pFedMe` and `lp-proj-2`, the optimal λ can be obtained by applying Lemma 29 with

$$(A,B,C)=\left(\frac{\sigma^2 d}{2Nn}+\frac{bd\Sigma}{2N},\ \frac{\sigma^2 bd}{Nn},\ \frac{\sigma^2 b^2 d}{2n}\right)\ \text{and}\ \left(\frac{\sigma^2 d_{\text{sub}}}{2Nn}+\frac{bd_{\text{sub}}\Sigma}{2N},\ \frac{\sigma^2 bd_{\text{sub}}}{Nn},\ \frac{\sigma^2 b^2 d_{\text{sub}}}{2n}\right),$$

respectively. One can check that both choices lead to the following value of λ: $\lambda^*:=\frac{(1-1/N)\sigma^2}{n\Sigma}$. Note that σ^2 is the variance of the observation noises on different clients, and n is the number of samples on each client. Thus λ^* can reflect the relative magnitude of the variance and the heterogeneity. Then we can compute the minimal test losses for the algorithms under consideration as follows.

$$\begin{aligned}
L^{\text{loc}} &= \frac{\sigma^2}{2} + \frac{\sigma^2 d}{2n(b+\lambda^*)^2}\left[(\lambda^*)^2 + 2b\lambda^* + b^2\right], \\
L^{\text{Avg}} &= \frac{\sigma^2}{2} + \frac{\sigma^2 d}{2n(b+\lambda^*)^2}\left[\frac{(\lambda^*)^2}{N} + \frac{N+1}{N}b\lambda^* + \frac{2N-1}{N}b^2 + \frac{N-1}{N}\frac{b^3}{\lambda^*}\right], \\
L_*^{\text{Me}} &= \frac{\sigma^2}{2} + \frac{\sigma^2 d}{2n(b+\lambda^*)^2}\left[\frac{(\lambda^*)^2}{N} + \frac{N+1}{N}b\lambda^* + b^2\right], \\
L_*^{\text{l2}} &= \frac{\sigma^2}{2} + \frac{\sigma^2 d_{\text{sub}}}{2n(b+\lambda^*)^2}\left[\frac{(\lambda^*)^2}{N} + \frac{N+1}{N}b\lambda^* + b^2\right] + \frac{\sigma^2(d-d_{\text{sub}})}{2n(b+\lambda^*)^2}\left[(\lambda^*)^2 + 2b\lambda^* + b^2\right].
\end{aligned}$$

Comparing their (optimal) losses, we can obtain the following observations.

- $L^{\text{loc}} \geq L_*^{\text{l2}} \geq L_*^{\text{Me}}$ and $L^{\text{Avg}} \geq L_*^{\text{Me}}$. This means that `pFedMe` with the optimal λ always has the minimal loss. Moreover, since `lp-proj-2` can be regarded as an interpolation of `local` and `pFedMe`, L_*^{l2} is also a interpolation of L^{loc} and L_*^{Me}.

- $L^{\text{loc}} \leq L^{\text{Avg}}$ if and only if $\lambda^* \leq b$. This means that if the heterogeneity or the number of local data is sufficiently large, then `local` is better than `FedAvg`.

- $L_*^{\text{l2}} \leq L^{\text{Avg}}$ if and only if $\lambda^* \leq \sqrt{\frac{d}{d-d_{\text{sub}}}}b$. The range of λ^* over which `lp-proj-2` is better than `FedAvg` is slightly larger than the range of that over which `local` is better than `FedAvg`.

- Fix σ^2 and n and let $\Sigma \to \infty$. Then we have $\lambda^* \to 0$, $\lim_{\lambda^*\to 0} L^{\text{loc}} = \lim_{\lambda^*\to 0} L_*^{\text{Me}} = \lim_{\lambda^*\to 0} L_*^{\text{l2}}$ and $\lim_{\lambda^*\to 0} L^{\text{Avg}} = \infty$. This implies that if the heterogeneity is sufficiently large, the optimal lambda is nearly 0 and there is little difference between `local`, `pFedMe` and `lp-proj-2`. And the loss of `FedAvg` is large. So there is no need for federated learning.

Up to now, we have only focused on the optimal value of λ. However, in practice, we can hardly know this value. Thus we need to compare these losses under different values of λ.

With (36) holding, we have the following results.

- $L^{\text{loc}} \leq L^{\text{Avg}}$ if and only if $\Sigma \geq \frac{N-1}{N}\frac{\sigma^2}{bn}$ ($\lambda^* \leq b$).

- $L^{\text{loc}} \leq L^{\text{l2}}(\lambda) \leq L^{\text{Me}}(\lambda)$ if and only if $\Sigma \geq \frac{N-1}{N}\frac{2b+\lambda}{\lambda}\frac{\sigma^2}{bn}$. If $\Sigma > \frac{N-1}{N}\frac{\sigma^2}{bn}$ ($\lambda^* < b$), this is equivalent to $\lambda \geq \frac{2\lambda^*}{1-\lambda^*/b}$.

- $L^{\text{Me}}(\lambda) \leq L^{\text{Avg}}$ if and only if $\Sigma \geq \frac{N-1}{N}\frac{\sigma^2}{(b+2\lambda)n}$. This is equivalent to $\lambda \geq \frac{\lambda^*-b}{2}$.

- $L^{\text{l2}}(\lambda) \leq L^{\text{Avg}}$ if and only if $\Sigma \geq \frac{N-1}{N}\frac{\sigma^2}{bn}\frac{d(b+\lambda)^2 - d_{\text{sub}}\lambda(2b+\lambda)}{d(b+\lambda)^2 - d_{\text{sub}}\lambda^2}$. About $\frac{d(b+\lambda)^2 - d_{\text{sub}}\lambda(2b+\lambda)}{d(b+\lambda)^2 - d_{\text{sub}}\lambda^2}$, we have $\frac{1+\sqrt{\frac{d-d_{\text{sub}}}{d}}}{1+\sqrt{\frac{d}{d-d_{\text{sub}}}}} \leq \frac{d(b+\lambda)^2 - d_{\text{sub}}\lambda(2b+\lambda)}{d(b+\lambda)^2 - d_{\text{sub}}\lambda^2} \leq 1$. When $\lambda = 0$ or $\lambda \to \infty$, the fraction goes to 1. When $\lambda = \sqrt{\frac{d}{d-d_{\text{sub}}}}b$, the fraction attains the minimal value.

Then we can sort these losses.

If the heterogeneity is small, i.e., $\Sigma < \frac{N-1}{N}\frac{\sigma^2}{bn}\frac{1+\sqrt{\frac{d-d_{\text{sub}}}{d}}}{1+\sqrt{\frac{d}{d-d_{\text{sub}}}}}$, then $\lambda^* > b$. When $\lambda < \frac{\lambda^*-b}{2}$, we have $L^{\text{Avg}} \leq L^{\text{Me}}(\lambda) \leq L^{\text{l2}}(\lambda) \leq L^{\text{loc}}$; when $\lambda > \frac{\lambda^*-b}{2}$, we have $L^{\text{Me}}(\lambda) \leq L^{\text{Avg}} \leq L^{\text{l2}}(\lambda) \leq L^{\text{loc}}$. In this case, `FedAvg` and `pFedMe` are always better than `lp-proj-2` and `local`. If λ is larger than a threshold value, `pFedMe` is better than `FedAvg`.

If the heterogeneity is large, i.e., $\Sigma > \frac{N-1}{N}\frac{\sigma^2}{bn}$, then $\lambda^* < b$. When $\lambda \leq \frac{2\lambda^*}{1-\lambda^*/b}$, we have $L^{\text{Me}}(\lambda) \leq L^{\text{l2}}(\lambda) \leq L^{\text{loc}} \leq L^{\text{Avg}}$; when $\lambda > \frac{2\lambda^*}{1-\lambda^*/b}$, we have $L^{\text{loc}} \leq L^{\text{l2}}(\lambda) \leq L^{\text{Me}}(\lambda) \leq L^{\text{Avg}}$. In this case, `FedAvg` is the worst method and `lp-proj-2` always lies between `local` and `pFedMe`.

B.3 Robustness

In this subsection, we consider the robustness of different methods against Byzantine attacks. Recall that in the last subsection, we only consider the exact solution of these methods and ignore the process of the algorithms. In terms of robustness, we must take the procedures of different methods into account, especially the communication between the central server and local clients. Moreover, we focus on the simplified setting where the number of local update steps is infinite, there is only one round of communication and all clients participate in the communication.

As indicated in Section 4.2 , we examine three types of Byzantine attacks. Throughout this subsection, we suppose that there are N_b benign clients and N_a malicious clients with $N_a + N_b = N$, and let I_b denote the indices of benign clients and I_a denote the indices of malicious clients.

We will analyze how these attacks will affect the solution of different methods, and compare the averaged test losses on benign clients.

B.3.1 The Simplified Setting

We first show that in our simplified setting, after one round of communication, all the methods will obtain their exact solutions defined in Appendix B.1.

`local` The objective of the local client is $\min_{\mathbf{w}\in\mathbb{R}^d} f_k(\mathbf{w})$. If the number of local update steps is infinite, we will obtain the least square estimator $\hat{\mathbf{w}}_k = \mathbf{w}_k^{\text{loc}}$. For the convergence of SGD, see Nemirovski et al. (2009).

`FedAvg` Similar to `local`, the local client will obtain $\hat{\mathbf{w}}_k$ and sends it to the server. Then the server obtains $\frac{1}{N}\sum_{k=1}^N \hat{\mathbf{w}}_k = \mathbf{w}^{\text{Avg}}$ and broadcasts $\mathbf{w}^{\text{Avg}}$ to all the clients.

`pFedMe` `pFedMe` corresponds to `lp-proj-2` with $\boldsymbol{P} = \boldsymbol{I}_d$. The local update step is $\mathbf{w}_{k,r+1}^t = \mathbf{w}_{k,r} - \eta\lambda(\mathbf{w}_{k,r}^t - \mathbf{x}_{k,r}^t)$ where $\mathbf{x}_{k,r}^t$ denotes the minimizer $\mathbf{x}_{k,r}^t = \hat{\mathbf{x}}_k(\mathbf{w}_{k,r}^t) = \text{argmin}_{\mathbf{x}\in\mathbb{R}^d}\left\{f_k(\mathbf{x}_k) + \frac{\lambda}{2}\left\|\mathbf{w}_{k,r}^t - \mathbf{x}_k\right\|_2^2\right\} = \frac{b\hat{\mathbf{w}}_k+\lambda\mathbf{w}_{k,r}}{b+\lambda}$. (When the number of local update steps is infinite, it is reasonable to assume that we can obtain the exact value of $\mathbf{x}_{k,r}^t$.) The local update rule can be rewritten as $\mathbf{w}_{k,r+1}^t = \mathbf{w}_{k,r} - \frac{\eta\lambda b}{b+\lambda}(\mathbf{w}_{k,r}^t - \hat{\mathbf{w}}_k)$, which can be regarded as a step of gradient descent with step size $\frac{\eta\lambda b}{b+\lambda}$ to minimize $\frac{1}{2}\left\|\mathbf{w} - \hat{\mathbf{w}}_k\right\|_2^2$. As long as the step size is not too large, we have $\lim_{R\to\infty}\mathbf{w}_{k,R}^t = \hat{\mathbf{w}}_k$. This means that if we do infinite steps of local update, the local version of global parameter is $\hat{\mathbf{w}}_k$. Then each client sends this local version to the server and the server obtains $\frac{1}{N}\sum_{k=1}^N \hat{\mathbf{w}}_k = \mathbf{w}^{\mathrm{Me}}$. After that, the server broadcasts $\mathbf{w}^{\mathrm{Me}}$ to all clients. Finally, the client k solves $\min_{\mathbf{x}\in\mathbb{R}^d}\left\{f_k(\mathbf{x}_k)+\frac{\lambda}{2}\left\|\mathbf{w}^{\mathrm{Me}}-\mathbf{x}_k\right\|_2^2\right\}$ and obtains $\mathbf{x}_k^{\mathrm{Me}} = \hat{\mathbf{x}}_k(\mathbf{w}^{\mathrm{Me}})$.

`Ditto` The global model of `Ditto` is the same as the model of `FedAvg`. So the server will also obtain $\frac{1}{N}\sum_{k=1}^N \hat{\mathbf{w}}_k = \mathbf{w}^{\mathrm{Di}}$. Then the server broadcasts $\mathbf{w}^{\mathrm{Di}}$ to all the clients and the client k solves $\min_{\mathbf{x}_k\in\mathbb{R}^d}\left\{f_k(\mathbf{x}_k)+\frac{\lambda}{2}\left\|\mathbf{w}^{\mathrm{Di}}-\mathbf{x}_k\right\|_2^2\right\}$ and gets $\mathbf{x}_k^{\mathrm{Di}}$.

`lp-proj-2` For `lp-proj-2`, without loss of generality, we can still assume $\boldsymbol{P}=\boldsymbol{P}_0 := (\mathbf{e}_1,\mathbf{e}_2,\ldots,\mathbf{e}_{d_{\mathrm{sub}}})$. The local update step is $\tilde{\mathbf{w}}_{k,r+1}^t = \check{\mathbf{x}}_k(\tilde{\mathbf{w}}_{k,r}^t) = \tilde{\mathbf{w}}_{k,r}^t - \eta\lambda(\tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_{k,r}^t) = \tilde{\mathbf{w}}_{k,r}^t - \frac{\eta\lambda b}{b+\lambda}(\tilde{\mathbf{w}}_{k,r}^t - \hat{\mathbf{w}}_{k,1})$, where

$$\mathbf{x}_{k,r}^t = \operatorname*{argmin}_{\mathbf{x}_k\in\mathbb{R}^d}\left\{f_k(\mathbf{x}_k)+\frac{\lambda}{2}\left\|\tilde{\mathbf{w}}_{k,r}^t-\boldsymbol{P}_0\mathbf{x}_k\right\|_2^2\right\} = \begin{pmatrix}(b\hat{\mathbf{w}}_{k,1}+\lambda\tilde{\mathbf{w}}_{k,r}^t)/(b+\lambda)\\ \hat{\mathbf{w}}_{k,2}\end{pmatrix}.$$

Similar to `pFedMe`, the local update step can be regarded as a step pf gradient descent with step size $\frac{\eta\lambda b}{b+\lambda}$ to minimize $\frac{1}{2}\left\|\tilde{\mathbf{w}}-\hat{\mathbf{w}}_{k,1}\right\|_2^2$. As long as the step size is not too large, we have $\lim_{R\to\infty}\tilde{\mathbf{w}}_{k,R}^t = \hat{\mathbf{w}}_{k,1}$. After the communication, the server gets $\frac{1}{N}\sum_{k=1}^N \hat{\mathbf{w}}_{k,1} = \tilde{\mathbf{w}}^{\mathrm{l2}}$ and the client k obtains $\mathbf{x}_k^{\mathrm{l2}} = \check{\mathbf{x}}_k(\tilde{\mathbf{w}}^{\mathrm{l2}})$.

B.3.2 Same-value Attacks

Now we focus on the same-value attacks.

As in Appendix B.2, to make calculations clean, we assume that the heterogeneity is uniform in all dimensions, i.e.,

$$\frac{1}{dN_b}\sum_{k\in\mathbf{I}_b}\left\|\frac{\sum_{i\in I_b}\mathbf{w}_i}{N}-\mathbf{w}_k\right\|_2^2 = \frac{1}{d_{\mathrm{sub}}N_b}\sum_{k\in I_b}\left\|\frac{\sum_{i\in I_b}\mathbf{w}_{i,1}}{N}-\mathbf{w}_{k,1}\right\|_2^2 := \Sigma_1. \tag{37}$$

`local` For pure local training, there is no communication between the central server and local clients. So the averaged test loss on benign clients is $L^{\text{loc, att1}} = \frac{1}{N_b}\sum_{k\in I_b} f_k^{\mathrm{te}}(\mathbf{w}_k^{\mathrm{loc}}) = \frac{\sigma^2}{2}+\frac{\sigma^2 d}{2n}$.

`FedAvg` For `FedAvg`, the local problem $\min_{\mathbf{w}\in\mathbb{R}^d} f_k(\mathbf{w})$ remains unchanged, no matter what the server sends to the local client. As long as the number of local update steps goes to ∞, the local parameter will go to the least square estimator $\hat{\mathbf{w}}_k$.

If the k-th client is benign, it will send $\hat{\mathbf{w}}_k$ to the server. Recall that $\hat{\mathbf{w}}_k{=}(\boldsymbol{\Xi}_k\boldsymbol{\Xi}_k)^{-1}\boldsymbol{\Xi}_k\mathbf{y}_k \sim \mathcal{N}(\mathbf{w}_k, \sigma^2\boldsymbol{I}_d/(bn))$. This means that $\hat{\mathbf{w}}_k$ can be viewed as an unbiased observation of $\mathbf{w}_k$ with covariance matrix $\frac{\sigma^2}{bn}\boldsymbol{I}_d$.

If the k-th client is malicious, it will send $\mathbf{w}_k^{(ma)} = c\boldsymbol{I}_d$ to the server with $c\sim\mathcal{N}(0,\tau^2)$. Then $\mathbf{w}_k^{(ma)}$ is an unbiased observation of $\mathbf{0}_m$ with covariance matrix $\tau^2\boldsymbol{J}_d$ where $\boldsymbol{J}_d = \begin{pmatrix}1 & 1 & \cdots & 1\\ 1 & 1 & \cdots & 1\\ \vdots & \vdots & \ddots & \vdots\\ 1 & 1 & \cdots & 1\end{pmatrix} \in \mathbb{R}^{d\times d}$. In this case, the number of local update steps will not affect the messages transferred by the malicious client. Then the server obtains $\mathbf{w}^{\mathrm{Avg,att1}} =$

$\frac{1}{N}\left(\sum_{k\in I_b}\hat{\mathbf{w}}_k+\sum_{k\in I_a}\mathbf{w}_k^{(ma)}\right)$. We have $\mathbf{w}^{\text{Avg,att1}}\sim\mathcal{N}\left(\frac{1}{N}\sum_{k\in I_b}\mathbf{w}_k,\frac{1}{N^2}\left(\frac{N_b\sigma^2}{bn}\boldsymbol{I}_d+N_a\tau^2\boldsymbol{J}_d\right)\right)$. Then we can compute the averaged test loss on benign clients as

$$L^{\text{Avg,att1}}=\frac{1}{N_b}\sum_{k\in I_b}f_k^{\text{te}}(\mathbf{w}^{\text{Avg,att1}})=\frac{\sigma^2}{2}+\frac{bd}{2N^2}\left(\frac{N_b\sigma^2}{bn}+N_a\tau^2\right)+\frac{b}{2N_b}\sum_{k\in I_b}\left\|\frac{\sum_{i\in I_b}\mathbf{w}_i}{N}-\mathbf{w}_k\right\|_2^2$$
$$\overset{(37)}{=}\frac{1}{N_b}\sum_{k\in I_b}f_k^{\text{te}}(\mathbf{w}^{\text{Avg,att1}})=\frac{\sigma^2}{2}+\frac{bd}{2N^2}\left(\frac{N_b\sigma^2}{bn}+N_a\tau^2\right)+\frac{bd\Sigma_1}{2}.$$

`pFedMe` Similar to `FedAvg`, the attack will not influence the minimization of the local model. If the k-th client is benign, it sends $\hat{\mathbf{w}}_k$ to the server. If the k-th client is malicious, it sends $\mathbf{w}_k^{(ma)}=c\mathbf{1}_d$ to the server with $c\sim\mathcal{N}(0,\tau^2)$. The server obtains $\mathbf{w}^{\text{Me,att1}}=\frac{1}{N}\left(\sum_{k\in I_b}\hat{\mathbf{w}}_k+\sum_{k\in I_a}\mathbf{w}_k^{(ma)}\right)=\mathbf{w}^{\text{Avg, att1}}$.

Then the server broadcasts $\mathbf{w}^{\text{Me,att1}}$ to all the clients. And the benign client k compute the local parameter $\mathbf{x}_k^{\text{Me,att1}}=\hat{\mathbf{x}}_k(\mathbf{w}^{\text{Me,att1}})=\frac{b\hat{\mathbf{w}}_k+\lambda\mathbf{w}^{\text{Me,att1}}}{b+\lambda}$. We have

$$\mathbf{x}_k^{\text{Me,att1}}\sim\mathcal{N}\left(\frac{b\mathbf{w}_k+\lambda\sum_{i\in I_b}\mathbf{w}_i/N}{b+\lambda},\frac{\left[\left(b+\frac{\lambda}{N}\right)^2\frac{\sigma^2}{bn}+(N_b-1)\frac{\lambda^2}{N^2}\frac{\sigma^2}{bn}\right]\boldsymbol{I}_d+N_a\frac{\lambda^2}{N^2}\tau^2\boldsymbol{J}_d}{(b+\lambda)^2}\right).$$

Then we can compute the averaged loss on benign clients as

$$L^{\text{Me,att1}}(\lambda)=\frac{1}{N_b}\sum_{k\in I_b}f_k^{\text{te}}(\mathbf{x}_k^{\text{Avg,att1}})$$
$$=\frac{\sigma^2}{2}+\frac{bd}{2}\cdot\frac{\left(b^2+\frac{2b\lambda}{N}+\frac{N_b\lambda^2}{N^2}\right)\frac{\sigma^2}{bn}+\frac{N_a\lambda^2}{N^2}\tau^2}{(b+\lambda)^2}+\frac{b\lambda^2}{2(b+\lambda)^2}\cdot\frac{1}{N_b}\sum_{i\in I_b}\left\|\frac{\sum_{i\in I_b}\mathbf{w}_i}{N}-\mathbf{w}_k\right\|_2^2$$
$$\overset{(37)}{=}\frac{\sigma^2}{2}+\frac{bd}{2}\cdot\frac{\left(b^2+\frac{2b\lambda}{N}+\frac{N_b\lambda^2}{N^2}\right)\frac{\sigma^2}{bn}+\frac{N_a\lambda^2}{N^2}\tau^2+\lambda^2\Sigma_1}{(b+\lambda)^2}.$$

`Ditto` Since the global model of `Ditto` is the same as the model of `FedAvg`, we have $\mathbf{w}^{\text{Di, att1}}=\mathbf{w}^{\text{Avg, att1}}$. Then the server broadcasts $\mathbf{w}^{\text{Di, att1}}$ to all the clients and the benign client k obtains $\mathbf{x}_k^{\text{Di, att1}}=\hat{\mathbf{x}}_k(\mathbf{w}^{\text{Di, att1}})=\frac{b\hat{\mathbf{w}}_k+\lambda\mathbf{w}^{\text{Di, att1}}}{b+\lambda}=\mathbf{x}_k^{\text{Me,att1}}$. Then `Ditto` and `pFedMe` have the same loss. So we will omit the analysis for `Ditto`.

`lp-proj-2` Similar to `pFedMe`, if the k-th client is benign, it sends $\hat{\mathbf{w}}_{k,1}$ to the server. If the k-th client is malicious, it sends $\tilde{\mathbf{w}}_k^{(ma)}=c\mathbf{1}_{d_{\text{sub}}}$ to the server where $c\sim\mathcal{N}(0,\tau^2)$. The server receives the messages and obtains $\tilde{\mathbf{w}}^{\text{l2,att1}}=\frac{1}{N}\left(\sum_{k\in I_b}\hat{\mathbf{w}}_{k,1}+\sum_{k\in I_a}\tilde{\mathbf{w}}_k^{(ma)}\right)$. And we have $\tilde{\mathbf{w}}^{\text{l2,att1}}\sim\mathcal{N}\left(\frac{1}{N}\sum_{k\in I_b}\mathbf{w}_{k,1},\frac{1}{N^2}\left(\frac{N_b\sigma^2}{bn}\boldsymbol{I}_{d_{\text{sub}}}+N_a\tau^2\boldsymbol{J}_{d_{\text{sub}}}\right)\right)$. Then the server broadcasts $\tilde{\mathbf{w}}^{\text{l2,att1}}$ to all the clients and the benign client k computes the optimal local parameter $\mathbf{x}_k^{\text{l2, att1}}=\check{\mathbf{x}}_k(\tilde{\mathbf{w}}^{\text{l2,att1}})=\begin{pmatrix}(b\hat{\mathbf{w}}_{k,1}+\lambda\tilde{\mathbf{w}}^{\text{l2, att1}})/(b+\lambda)\\ \hat{\mathbf{w}}_{k,2}\end{pmatrix}$. It follows that

$$\mathbf{x}_k^{\text{l2,att1}}\sim\mathcal{N}\left(\begin{pmatrix}\frac{b\mathbf{w}_{k,1}+\lambda\sum_{i\in I_b}\mathbf{w}_{i,1}/N}{b+\lambda}\\ \mathbf{x}_{k,2}\end{pmatrix},\begin{pmatrix}\frac{\left[\left(b+\frac{\lambda}{N}\right)^2\frac{\sigma^2}{bn}+(N_b-1)\frac{\sigma^2}{bn}\right]\boldsymbol{I}_{d_{\text{sub}}}+N_a\frac{\lambda^2}{N^2}\tau^2\boldsymbol{J}_{d_{\text{sub}}}}{(b+\lambda)^2} & \\ & \frac{\sigma^2}{bn}\boldsymbol{I}_{d_{\text{sub}}}\end{pmatrix}\right).$$

Then we can compute the averaged loss on benign clients as

$$\begin{aligned}
L^{\text{l2, att1}}(\lambda) &= \frac{1}{N_b}\sum_{k\in I_b} f_k^{\text{te}}(\mathbf{x}_k^{\text{l2, att1}}) \\
&= \frac{\sigma^2}{2} + \frac{bd_{\text{sub}}}{2}\cdot\frac{\left(b^2+\frac{2b\lambda}{N}+\frac{N_b\lambda^2}{N^2}\right)\frac{\sigma^2}{bn}+\frac{N_a\lambda^2}{N^2}\tau^2}{(b+\lambda)^2} + \frac{(d-d_{\text{sub}})\sigma^2}{2n} \\
&\quad + \frac{b\lambda^2}{2(b+\lambda)^2}\cdot\frac{1}{N_b}\sum_{i\in I_b}\left\|\frac{\sum_{i\in I_b}\mathbf{w}_{i,1}}{N}-\mathbf{w}_{k,1}\right\|_2^2 \\
&\overset{(37)}{=} \frac{\sigma^2}{2} + \frac{bd_{\text{sub}}}{2}\cdot\frac{\left(b^2+\frac{2b\lambda}{N}+\frac{N_b\lambda^2}{N^2}\right)\frac{\sigma^2}{bn}+\frac{N_a\lambda^2}{N^2}\tau^2+\lambda^2\Sigma_1}{(b+\lambda)^2} + \frac{(d-d_{\text{sub}})\sigma^2}{2n}.
\end{aligned}$$

To find the optimal λ for `pFedMe` and `lp-proj-2`, we could apply Lemma 29 again with $(A,B,C) = \left(\frac{\sigma^2 N_b}{bN^2 n}+\frac{N_a\tau^2}{N^2}+\Sigma_1,\ \frac{2\sigma^2}{Nn},\ \frac{\sigma^2 b}{n}\right)$. One can check that the optimal λ is $\lambda_1^* := \frac{(1-1/N)\sigma^2/n}{\Sigma_1+\frac{N_a}{N^2}(\tau^2-\sigma^2/(bn))}$. The numerator of λ_1^* is the variance of noises over the number of samples. The denominator is the sum of data heterogeneity and variance of attacks.

Now we can obtain the losses of different methods at λ_1^*.

$$\begin{aligned}
L^{\text{loc, att1}} &= \frac{\sigma^2}{2}+\frac{\sigma^2 d}{2n} = \frac{\sigma^2}{2}+\frac{\sigma^2 d}{2n(b+\lambda_1^*)^2}\left[(\lambda_1^*)^2+2b\lambda_1^*+b^2\right], \\
L^{\text{Avg,att1}} &= \frac{\sigma^2}{2}+\frac{\sigma^2 d}{2n(b+\lambda_1^*)^2}\left[\frac{(\lambda_1^*)^2}{N}+\frac{N+1}{N}b\lambda_1^*+\frac{2N-1}{N}b^2+\frac{N-1}{N}\frac{b^3}{\lambda_1^*}\right], \\
L_*^{\text{Me, att1}} &= L^{\text{Me,att1}}(\lambda_1^*) = \frac{\sigma^2}{2}+\frac{\sigma^2 d}{2n(b+\lambda_1^*)^2}\left[\frac{(\lambda_1^*)^2}{N}+\frac{N+1}{N}b\lambda_1^*+b^2\right], \\
L_*^{\text{l2, att1}} &= L^{\text{l2,att1}}(\lambda_1^*) = \frac{\sigma^2}{2}+\frac{\sigma^2 d_{\text{sub}}}{2n(b+\lambda_1^*)^2}\left[\frac{(\lambda_1^*)^2}{N}+\frac{N+1}{N}b\lambda_1^*+b^2\right] \\
&\qquad + \frac{\sigma^2(d-d_{\text{sub}})}{2n(b+\lambda_1^*)^2}\left[(\lambda_1^*)^2+2b\lambda_1^*+b^2\right].
\end{aligned} \tag{38}$$

We have the following observations.

- $L^{\text{loc, att1}} \geq L_*^{\text{l2, att1}} \geq L_*^{\text{Me, att1}}$ and $L^{\text{Avg, att1}} \geq L_*^{\text{Me, att1}}$. This means that `pFedMe` with the optimal λ always has the minimal loss.
- $L^{\text{loc, att1}} \leq L^{\text{Avg, att1}}$ if and only if $\lambda_1^* \leq b$. This means that if the heterogeneity or the noise of attacks is sufficiently large, then `local` is better than `FedAvg`.
- $L_*^{\text{l2}} \leq L^{\text{Avg, att1}}$ if and only if $\lambda_1^* \leq \sqrt{\frac{d}{d-d_{\text{sub}}}}b$. The range of λ^* over which `lp-proj-2` is better than `FedAvg` is slightly larger than the range of that over which `local` is better than `FedAvg`.

Since τ^2 can be very large, λ_1^* is much smaller than λ^*. Recall that in the settings of Figure 1, we have λ_1^* =4.9e-04.

Now we compare the losses for different values of λ and give the formal version of Proposition 16.

Theorem 30 (Formal version of Proposition 16) *We have*

$$\begin{aligned}
L^{\textit{loc, att1}} &= \frac{\sigma^2}{2} + \frac{\sigma^2 d}{2n}, \\
L^{\textit{Avg,att1}} &= \frac{\sigma^2}{2} + \frac{bd}{2N^2}\left(\frac{N_b\sigma^2}{bn} + N_a\tau^2\right) + \frac{bd\Sigma_1}{2}, \\
L^{\textit{Me,att1}}(\lambda) &= \frac{\sigma^2}{2} + \frac{bd}{2}\cdot\frac{\left(b^2 + \frac{2b\lambda}{N} + \frac{N_b\lambda^2}{N^2}\right)\frac{\sigma^2}{bn} + \frac{N_a\lambda^2}{N^2}\tau^2}{(b+\lambda)^2} + \frac{b\lambda^2 d\Sigma_1}{2(b+\lambda)^2}, \\
L^{\textit{Di, att1}}(\lambda) &= L^{\textit{Me,att1}}(\lambda), \\
L^{\textit{l2, att1}}(\lambda) &= \frac{\sigma^2}{2} + \frac{bd_{\text{sub}}}{2}\cdot\frac{\left(b^2 + \frac{2b\lambda}{N} + \frac{N_b\lambda^2}{N^2}\right)\frac{\sigma^2}{bn} + \frac{N_a\lambda^2}{N^2}\tau^2}{(b+\lambda)^2} + \frac{(d-d_{\text{sub}})\sigma^2}{2n} + \frac{b\lambda^2 d_{\text{sub}}\Sigma_1}{2(b+\lambda)^2}.
\end{aligned} \tag{39}$$

And the following propositions hold.

- $L^{\textit{loc,att1}} \leq L^{\textit{Avg,att1}}$ *if and only if* $\Sigma_1 + \frac{N_a}{N^2}\left(\tau^2 - \frac{\sigma^2}{bn}\right) \geq \frac{N-1}{N}\frac{\sigma^2}{bn}$ $(\lambda_1^* \leq b)$*.*

- $L^{\textit{loc, att1}} \leq L^{\textit{l2, att1}}(\lambda) \leq L^{\textit{Me,att1}}(\lambda)$ *if and only if* $\Sigma_1 + \frac{N_a}{N^2}\left(\tau^2 - \frac{\sigma^2}{bn}\right) \geq \frac{N-1}{N}\frac{2b+\lambda}{\lambda}\frac{\sigma^2}{bn}$. *If* $\Sigma_1 + \frac{N_a}{N^2}\left(\tau^2 - \frac{\sigma^2}{bn}\right) > \frac{N-1}{N}\frac{\sigma^2}{bn}$ $(\lambda_1^* < b)$*, this is equivalent to* $\lambda \geq \frac{2\lambda_1^*}{1-\lambda_1^*/b}$*.*

- $L^{\textit{Me, att1}}(\lambda) \leq L^{\textit{Avg, att1}}$ *if and only if* $\Sigma_1 + \frac{N_a}{N^2}\left(\tau^2 - \frac{\sigma^2}{bn}\right) \geq \frac{N-1}{N}\frac{\sigma^2}{(b+2\lambda)n}$*. This is equivalent to* $\lambda \geq \frac{\lambda_1^*-b}{2}$*.*

- $L^{\textit{l2, att1}}(\lambda) \leq L^{\textit{Avg, att1}}$ *if and only if* $\Sigma_1 + \frac{N_a}{N^2}\left(\tau^2 - \frac{\sigma^2}{bn}\right) \geq \frac{N-1}{N}\frac{\sigma^2}{bn}\frac{d(b+\lambda)^2 - d_{\text{sub}}\lambda(2b+\lambda)}{d(b+\lambda)^2 - d_{\text{sub}}\lambda^2}$*.*

With (39), it is easy to check the above propositions hold.

If the attacks are very serious, we can have $\Sigma_1 + \frac{N_a}{N^2}\left(\tau^2 - \frac{\sigma^2}{bn}\right) > \frac{N-1}{N}\frac{\sigma^2}{bn}$ $(\lambda_1^* < b)$. Similar to the analysis at the end of Appendix B.2, when $\lambda \leq \frac{2\lambda_1^*}{1-\lambda_1^*/b}$, we have $L^{\text{Me, att1}}(\lambda) \leq L^{\text{l2, att1}}(\lambda) \leq L^{\text{loc, att1}} \leq L^{\text{Avg, att1}}$; when $\lambda > \frac{2\lambda_1^*}{1-\lambda_1^*/b}$, we have $L^{\text{loc, att1}} \leq L^{\text{l2, att1}}(\lambda) \leq L^{\text{Me, att1}}(\lambda) \leq L^{\text{Avg, att1}}$.

Remark 31 (Detailed presentation of Remark 17) *The optimal test loss function of* `lp-proj-2` *is*

$$\begin{aligned}
L_{l2}^* &= \frac{\sigma^2}{2} + \frac{\sigma^2 d_{\text{sub}}}{2n(b+\lambda_1^*)^2}\left[\frac{(\lambda_1^*)^2}{N} + \frac{N+1}{N}b\lambda_1^* + b^2\right] + \frac{\sigma^2(d-d_{\text{sub}})}{2n(b+\lambda_1^*)^2}\left[(\lambda_1^*)^2 + 2b\lambda_1^* + b^2\right] \\
&= C + \frac{\sigma^2\lambda_1}{2n(b+\lambda_1)}\left(\frac{1}{N} - 1\right)d_{\text{sub}},
\end{aligned}$$

where C is used to represent the quantities irrelevant to d_{sub}. Since $1/N - 1 < 0$, we can see that the coefficient of d_{sub} is negative, which implies that the test loss would be smaller as we increase the dimension of the random projection subspace.

The first-order derivative of the test loss of `lp-proj-2` *is*

$$L'_{l_2,att1}(\lambda) = \frac{bd_{\text{sub}}\sigma^2}{n} \cdot \frac{(\frac{1}{N}-1)b + (\frac{N_b}{N}-1)\cdot\frac{\lambda}{N}}{(b+\lambda)^3} + \frac{b^2\lambda d_{\text{sub}}(\frac{N_a\tau^2}{N^2}) + \Sigma_1}{(b+\lambda)^3}.$$

One can see that the first-order derivative is linear in d_{sub}. Therefore, given all the other parameters, for a specific λ, the derivative is smaller in absolute value if d_{sub} is smaller. In other words, the test loss is less sensitive to the change of λ, which gives a more robust performance to the algorithm when precisely tuning the hyper-parameter is hard.

B.3.3 Sign-flipping Attacks

The second type of attack is sign-flipping attacks. For simplicity, we define $\bar{\mathbf{w}}_b = \frac{1}{N_b}\sum_{i\in I_b}\mathbf{w}_i$, $\bar{\mathbf{w}}_a = \frac{1}{N_a}\sum_{i\in I_a}\mathbf{w}_i$, $\bar{\mathbf{w}}_{b,1} = \frac{1}{N_b}\sum_{i\in I_b}\mathbf{w}_{i,1}$ and $\bar{\mathbf{w}}_{a,1} = \frac{1}{N_a}\sum_{i\in I_a}\mathbf{w}_{i,1}$.

To simplify and clarify the calculations, we continue to focus on the scenario where the heterogeneity is uniform across all dimensions. However, due to the increased complexity of the sign-flipping attack compared to the same-value attack, this uniformity condition takes a more intricate form, specifically:

$$\begin{aligned}\frac{1}{dN_b}\sum_{k\in I_b}&\left\|\frac{N_b\bar{\mathbf{w}}_b - N_a\sqrt{2/\pi}\tau\bar{\mathbf{w}}_a}{N} - \mathbf{w}_k\right\|_2^2 \\ &= \frac{1}{d_{\text{sub}}N_b}\sum_{k\in I_b}\left\|\frac{N_b\bar{\mathbf{w}}_{b,1} - N_a\sqrt{2/\pi}\tau\bar{\mathbf{w}}_{a,1}}{N} - \mathbf{w}_{k,1}\right\|_2^2 := \Sigma_2, \qquad (40)\end{aligned}$$

$$\frac{\sum_{i\in I_a}\operatorname{tr}(\boldsymbol{M}_i)}{N_a d} = \frac{\sum_{i\in I_a}\operatorname{tr}(\widetilde{\boldsymbol{M}}_i)}{N_a d_{\text{sub}}} := M_0, \qquad (41)$$

where $\boldsymbol{M}_i$ and $\widetilde{\boldsymbol{M}}_i$ denote the variance of the information transmitted by malicious clients under the attacks when applying `pFedMe` and `lp-proj-2`, respectively. Their specific forms are defined in (42) and (43). The form of (40) arises because the attack involves sign-flipping.

`local` This attack will not affect `local`. So the averaged test loss on benign clients is $L^{\text{loc, att2}} = \frac{1}{N_b}\sum_{k\in I_b} f_k^{\text{te}}(\mathbf{w}_k^{\text{loc}}) = \frac{\sigma^2}{2} + \frac{\sigma^2 d}{2n}$.

`FedAvg` If the k-th client is benign, it sends $\hat{\mathbf{w}}_k$ to the server. If the k-th client is malicious, it send $\mathbf{w}_k^{(ma)} = -|c|\hat{\mathbf{w}}_k$ to the server, where $c \sim \mathcal{N}(0,\tau^2)$. Recall that $\hat{\mathbf{w}}_k$ is an unbiased observation of $\mathbf{w}_k$ with covariance matrix $\frac{\sigma^2}{bn}\boldsymbol{I}_d$. Now we examine $\mathbf{w}_k^{(ma)}$. Although $\mathbf{w}_k^{(ma)}$ does not obey Gaussian distribution, we can still calculate its mean and covariance by the independence of c and $\hat{\mathbf{w}}_k$. We have $\mathbb{E}\left[-|c|\hat{\mathbf{w}}_k\right] = -\sqrt{2/\pi}\tau\mathbf{w}_k$ and

$$\begin{aligned}\mathsf{var}\left(-|c|\hat{\mathbf{w}}_k\right) &= \mathbb{E}\left[c^2\hat{\mathbf{w}}_k\hat{\mathbf{w}}_k^\top\right] - \mathbb{E}\left[|c|\hat{\mathbf{w}}_k\right]\mathbb{E}\left[|c|\hat{\mathbf{w}}_k^\top\right] \\ &= \tau^2\left(\mathbf{w}_k\mathbf{w}_k^\top + \frac{\sigma^2}{bn}\boldsymbol{I}_d\right) - \frac{2}{\pi}\tau^2\mathbf{w}_k\mathbf{w}_k^\top\end{aligned}$$

$$= \frac{\pi - 2}{\pi} \tau^2 \mathbf{w}_k \mathbf{w}_k^\top + \tau^2 \frac{\sigma^2}{bn} \boldsymbol{I}_d := \boldsymbol{M}_k \tag{42}$$

Then $\mathbf{w}_k^{(ma)}$ can be regarded as an unbiased observation of $-\sqrt{2/\pi}\tau \mathbf{w}_k$ with covariance matrix $\boldsymbol{M}_k$. Thus the server gets $\mathbf{w}^{\text{Avg, att2}} = \frac{1}{N}\left(\sum_{k\in I_b} \hat{\mathbf{w}}_k + \sum_{k \in I_a} \mathbf{w}_k^{(ma)}\right)$. We have $\mathbb{E}[\mathbf{w}^{\text{Avg, att2}}] = \frac{N_b \bar{\mathbf{w}}_b - N_a\sqrt{2/\pi}\tau\bar{\mathbf{w}}_a}{N}$ and $\mathsf{var}(\mathbf{w}^{\text{Avg, att2}}) = \frac{1}{N^2}\left(\frac{N_b\sigma^2}{bn}\boldsymbol{I}_d + \sum_{k\in I_a}\boldsymbol{M}_k\right)$.

Then we can compute the averaged test loss on benign clients as

$$\begin{aligned}
L^{\text{Avg, att2}} &= \frac{1}{N_b}\sum_{k\in I_b} f_k^{\text{te}}(\mathbf{w}^{\text{Avg, att2}}) \\
&= \frac{\sigma^2}{2} + \frac{bd}{2N^2}\left(\frac{N_b\sigma^2}{bn} + \sum_{k\in I_a}\frac{\text{tr}(\boldsymbol{M}_k)}{d}\right) + \frac{b}{2N_b}\sum_{k\in I_b}\left\|\frac{N_b\bar{\mathbf{w}}_b - N_a\sqrt{2/\pi}\tau\bar{\mathbf{w}}_a}{N} - \mathbf{w}_k\right\|_2^2 \\
&\overset{(40)+(41)}{=} \frac{\sigma^2}{2} + \frac{bd}{2N^2}\left(\frac{N_b\sigma^2}{bn} + N_a M_0\right) + \frac{bd\Sigma_2}{2}.
\end{aligned}$$

`pFedMe` If the k-th client is benign, it will send $\hat{\mathbf{w}}_k$ to the server. If the k-th client is malicious, it will send $\mathbf{w}_k^{(ma)} = -|c|\hat{\mathbf{w}}_k$ to the server. The server obtains $\mathbf{w}^{\text{Me, att2}} = \frac{1}{N}\left(\sum_{k\in I_b}\hat{\mathbf{w}}_k + \sum_{k\in I_a}\mathbf{w}_k^{(ma)}\right) = \mathbf{w}^{\text{Avg, att2}}$ and broadcasts $\mathbf{w}^{\text{Me, att2}}$ back to all the clients. For the benign client k, it gets $\mathbf{x}_k^{\text{Me, att2}} = \hat{\mathbf{x}}_k(\mathbf{w}^{\text{Me, att2}}) = \frac{b\hat{\mathbf{w}}_k + \lambda\mathbf{w}^{\text{Me, att2}}}{b+\lambda}$. We have that the expectation is $\mathbb{E}[\mathbf{x}_k^{\text{Me, att2}}] = \frac{1}{b+\lambda}\left[b\mathbf{w}_k + \lambda\frac{N_b\bar{\mathbf{w}}_b - N_a\sqrt{2/\pi}\tau\bar{\mathbf{w}}_a}{N}\right]$ and the variance is $\mathsf{var}(\mathbf{x}_k^{\text{Me, att2}}) = \frac{\left[\left(b+\frac{\lambda}{N}\right)^2\frac{\sigma^2}{bn} + (N_b-1)\frac{\lambda^2}{N^2}\frac{\sigma^2}{bn}\right]\boldsymbol{I}_d + \frac{\lambda^2}{N^2}\sum_{i\in I_a}\boldsymbol{M}_i}{(b+\lambda)^2}$.

Then we can compute the averaged test loss on benign clients as

$$\begin{aligned}
L^{\text{Me, att2}}(\lambda) &= \frac{1}{N_b}\sum_{k\in I_b} f_k^{\text{te}}(\mathbf{x}_k^{\text{Me, att2}}) \\
&= \frac{\sigma^2}{2} + \frac{bd}{2}\frac{\left(b^2 + \frac{2b\lambda}{N} + \frac{N_b\lambda^2}{N^2}\right)\frac{\sigma^2}{bn} + \frac{\lambda^2}{N^2}\sum_{i\in I_a}\frac{\text{tr}(\boldsymbol{M}_i)}{d}}{(b+\lambda)^2} \\
&\quad + \frac{b\lambda^2}{2(b+\lambda)^2}\frac{1}{N_b}\sum_{k\in I_b}\left\|\frac{N_b\bar{\mathbf{w}}_b - N_a\sqrt{2/\pi}\tau\bar{\mathbf{w}}_a}{N} - \mathbf{w}_k\right\|_2^2 \\
&\overset{(40)+(41)}{=} \frac{\sigma^2}{2} + \frac{bd}{2}\frac{\left(b^2 + \frac{2b\lambda}{N} + \frac{N_b\lambda^2}{N^2}\right)\frac{\sigma^2}{bn} + \frac{N_a\lambda^2}{N^2}M_0 + \lambda^2\Sigma_2}{(b+\lambda)^2}.
\end{aligned}$$

`Ditto` The global model of `Ditto` is the same as the model of `FedAvg`. Similar to the analysis of same-value attacks, we have $\mathbf{w}^{\text{Di, att2}} = \mathbf{w}^{\text{Avg, att2}}$ and $\mathbf{x}_k^{\text{Di, att2}} = \hat{\mathbf{x}}_k(\mathbf{w}^{\text{Di, att2}}) = \frac{b\hat{\mathbf{w}}_k + \lambda\mathbf{w}^{\text{Di, att2}}}{b+\lambda} = \mathbf{x}_k^{\text{Me, att2}}$. Then `Ditto` and `pFedMe` have the same losses. We will also omit the analysis for `Ditto`.

`lp-proj-2` If the k-th client is benign, it will send $\hat{\mathbf{w}}_{k,1}$ to the server. If the k-th client is malicious, it will send $\tilde{\mathbf{w}}_k^{(ma)} = -|c|\hat{\mathbf{w}}_{k,1}$ to the server, where $c \sim \mathcal{N}(0, \tau^2)$. Then we have

$\mathbb{E}[\tilde{\mathbf{w}}_k^{(ma)}] = -\sqrt{2/\pi}\tau\mathbf{w}_{k,1}$ and

$$\mathsf{var}(\tilde{\mathbf{w}}_k^{(ma)}) = \frac{\pi-2}{\pi}\tau^2\mathbf{w}_{k,1}\mathbf{w}_{k,1}^\top + \tau^2\frac{\sigma^2}{bn}\boldsymbol{I}_{d_{\text{sub}}} := \widetilde{\boldsymbol{M}}_k. \tag{43}$$

The server receives these messages and gets $\tilde{\mathbf{w}}^{\text{l2, att2}} = \frac{1}{N}\left(\sum_{k\in I_b}\hat{\mathbf{w}}_{k,1} + \sum_{k\in I_a}\tilde{\mathbf{w}}_k^{(ma)}\right)$. And the benign client k obtains $\mathbf{x}_k^{\text{l2, att2}} = \check{\mathbf{x}}_k(\widetilde{\mathbf{w}}^{\text{l2, att2}}) = \begin{pmatrix}(b\hat{\mathbf{w}}_{k,1} + \lambda\tilde{\mathbf{w}}^{\text{l2, att2}})/(b+\lambda) \\ \hat{\mathbf{w}}_{k,2}\end{pmatrix}$. Then we have $\mathbb{E}[\mathbf{x}_k^{\text{l2, att2}}] = \begin{pmatrix}\frac{1}{b+\lambda}\left[b\mathbf{w}_{k,1} + \lambda\frac{N_b\bar{\mathbf{w}}_{b,1} - N_a\sqrt{2/\pi}\tau\bar{\mathbf{w}}_{a,1}}{N}\right] \\ \mathbf{w}_{k,2}\end{pmatrix}$ and $\mathsf{var}(\mathbf{x}_k^{\text{l2, att2}}) = \begin{pmatrix}\frac{\left[\left(b+\frac{\lambda}{N}\right)^2\frac{\sigma^2}{bn} + (N_b-1)\frac{\lambda^2}{N^2}\frac{\sigma^2}{bn}\right]\boldsymbol{I}_d + \frac{\lambda^2}{N^2}\sum_{i\in I_a}\widetilde{\boldsymbol{M}}_i}{(b+\lambda)^2} & \\ & \frac{\sigma^2}{bn}\boldsymbol{I}_{d-d_{\text{sub}}}\end{pmatrix}$. The averaged test loss on benign clients is

$$\begin{aligned}
L^{\text{l2, att2}}(\lambda) &= \frac{1}{N_b}\sum_{k\in I_b} f_k^{\text{te}}(\mathbf{x}_k^{\text{l2, att2}}) \\
&= \frac{\sigma^2}{2} + \frac{bd_{\text{sub}}}{2}\frac{\left(b^2 + \frac{2b\lambda}{N} + \frac{N_b\lambda^2}{N^2}\right)\frac{\sigma^2}{bn} + \frac{\lambda^2}{N^2}\sum_{i\in I_a}\frac{\text{tr}(\widetilde{\boldsymbol{M}}_i)}{d_{\text{sub}}}}{(b+\lambda)^2} + \frac{(d-d_{\text{sub}})\sigma^2}{2n} \\
&\quad + \frac{b\lambda^2}{2(b+\lambda)^2}\frac{1}{N_b}\sum_{k\in I_b}\left\|\frac{N_b\bar{\mathbf{w}}_{b,1} - N_a\sqrt{2/\pi}\tau\bar{\mathbf{w}}_{a,1}}{N} - \mathbf{w}_{k,1}\right\|_2^2 \\
&\overset{(40)+(41)}{=} \frac{\sigma^2}{2} + \frac{bd_{\text{sub}}}{2}\frac{\left(b^2 + \frac{2b\lambda}{N} + \frac{N_b\lambda^2}{N^2}\right)\frac{\sigma^2}{bn} + \frac{N_a\lambda^2}{N^2}M_0 + \lambda^2\Sigma_2}{(b+\lambda)^2} + \frac{(d-d_{\text{sub}})\sigma^2}{2n}.
\end{aligned}$$

We find that under the conditions in (40) and (41), the losses of different methods have similar forms as (39). To find the optimal λ for `pFedMe` and `lp-proj-2`, we could also apply Lemma 29 with $(A,B,C) = \left(\frac{\sigma^2N_b}{bN^2n} + \frac{N_aM_0}{N^2} + \Sigma_2,\ \frac{2\sigma^2}{Nn},\ \frac{\sigma^2b}{n}\right)$. One can check that the optimal λ is $\lambda_2^* := \frac{(1-1/N)\sigma^2/n}{\Sigma_2 + \frac{N_a}{N^2}(M_0 - \sigma^2/(bn))}$. The losses of different methods at optimal λ_2^* also share similar forms as (38), with Σ_1, τ^2, λ_1^* replaced by Σ_2, M_0 and λ_2^* respectively. For the comparison of losses at different values of λ, The discussion below (39) also applies here.

B.3.4 Gaussian Attacks

Gaussian attacks are similar to same-value attacks. For `FedAvg`, `pFedMe` and `Ditto`, the malicious client sends $\mathbf{w}_k^{(ma)}$ to the server, where $\mathbf{w}_k^{(ma)} \sim \mathcal{N}(\mathbf{0}_d, \tau^2\boldsymbol{I}_d)$. For `lp-proj-2`, the malicious client sends $\tilde{\mathbf{w}}_k^{(ma)}$ to the server, where $\tilde{\mathbf{w}}_k^{(ma)} \sim \mathcal{N}(\mathbf{0}_{d_{\text{sub}}}, \tau^2\boldsymbol{I}_{d_{\text{sub}}})$.

Note that $\text{tr}(\boldsymbol{I}_d) = \text{tr}(\boldsymbol{J}_d)$ for any d and the test loss (35) is only relevant to the trace of the covariance matrix. Thus the averaged test losses on benign clients under Gaussian attacks are the same as those under same-value attacks.

B.4 Fairness

In this subsection, we examine the performance fairness of these methods. Recall that in Definition 1, we measure performance fairness in terms of the variance of test accuracy/losses. In Appendix B.2, the test loss on client k is

$$f_k^{\mathrm{te}}(\mathbf{x}_k) = \frac{\sigma^2}{2} + \frac{b}{2}\mathrm{tr}(\mathsf{var}(\mathbf{x}_k)) + \frac{b}{2}\left\|\mathbb{E}\mathbf{x}_k - \mathbf{w}_k\right\|_2^2.$$

For different methods, we can compute that

$$\begin{aligned}
f_k^{\mathrm{te}}(\mathbf{w}_k^{\mathrm{loc}}) &= \frac{\sigma^2}{2} + \frac{\sigma^2 d}{2n},\\
f_k^{\mathrm{te}}(\mathbf{w}^{\mathrm{Avg}}) &= \frac{\sigma^2}{2} + \frac{\sigma^2 d}{2Nn} + \frac{b}{2}\left\|\bar{\mathbf{w}} - \mathbf{w}_k\right\|_2^2,\\
f_k^{\mathrm{te}}(\mathbf{x}_k^{\mathrm{Me}}) &= \frac{\sigma^2}{2} + \frac{b^2 + \frac{2b\lambda}{N} + \frac{\lambda^2}{N}}{(b+\lambda)^2}\frac{\sigma^2 d}{2n} + \frac{b\lambda^2}{2(b+\lambda)^2}\left\|\bar{\mathbf{w}} - \mathbf{w}_k\right\|_2^2,\\
f_k^{\mathrm{te}}(\mathbf{x}_k^{\mathrm{Di}}) &= f_k^{\mathrm{te}}(\mathbf{x}_k^{\mathrm{Me}}),\\
f_k^{\mathrm{te}}(\mathbf{x}_k^{\mathrm{l2}}) &= \frac{\sigma^2}{2} + \frac{b^2 + \frac{2b\lambda}{N} + \frac{\lambda^2}{N}}{(b+\lambda)^2}\frac{\sigma^2 d_{\mathrm{sub}}}{2n} + \frac{\sigma^2(d - d_{\mathrm{sub}})}{2n} + \frac{b\lambda^2}{2(b+\lambda)^2}\left\|\bar{\mathbf{w}}_{\cdot,1} - \mathbf{w}_{k,1}\right\|_2^2.
\end{aligned}$$

For simplicity of notation, given N real numbers $a_1, a_2, \ldots, a_N$, we use $\widetilde{\mathsf{var}}(a_k)$ the variance of a random variable that takes the value a_k with probability $\frac{1}{N}$, as defined below[7]

$$\widetilde{\mathsf{var}}(a_k) := \frac{1}{N}\sum_{k=1}^{N} a_k^2 - \left(\frac{1}{N}\sum_{k=1}^{N} a_k\right)^2. \tag{44}$$

Then we give the formal version of Proposition 18.

Theorem 32 (Formal version of Proposition 18) *The variances of test losses on different clients for these methods are as follows:*

$$\begin{aligned}
V^{loc} &= \widetilde{\mathsf{var}}(f_k^{te}(\mathbf{w}_k^{loc})) = 0,\\
V^{Avg} &= \widetilde{\mathsf{var}}(f_k^{te}(\mathbf{w}^{Avg})) = \frac{b^2}{4}\widetilde{\mathsf{var}}(\|\bar{\mathbf{w}} - \mathbf{w}_k\|_2^2),\\
V^{Me}(\lambda) &= \widetilde{\mathsf{var}}(f_k^{te}(\mathbf{x}_k^{Me})) = \frac{b^2\lambda^4}{4(b+\lambda)^4}\widetilde{\mathsf{var}}(\|\bar{\mathbf{w}} - \mathbf{w}_k\|_2^2),\\
V^{Di}(\lambda) &= V^{Me}(\lambda),\\
V^{l2}(\lambda) &= \widetilde{\mathsf{var}}(f_k^{te}(\mathbf{x}_k^{l2})) = \frac{b^2\lambda^4}{4(b+\lambda)^4}\widetilde{\mathsf{var}}(\|\bar{\mathbf{w}}_{\cdot,1} - \mathbf{w}_{k,1}\|_2^2).
\end{aligned} \tag{45}$$

If the $\mathbf{w}_k$ are i.i.d. from $\mathcal{N}(\mu_w, \mathbf{\Sigma}_w)$ with $\mathbf{\Sigma}_w = (\Sigma_{ij})_{d\times d}$, we have

$$\mathbb{E}V^{Avg} = b^2\frac{(N-1)(N-2)}{N^2}\sum_{i=1}^{d}\sum_{j=1}^{d}\Sigma_{ij}^2 = O\left(d^2\right), \tag{46}$$

[7] Here, we slightly abuse notation by treating the lowercase letter a_k a random variable.

$$\mathbb{E}V^{Me}(\lambda) = \frac{b^2\lambda^4}{(b+\lambda)^4}\frac{(N-1)(N-2)}{N^2}\sum_{i=1}^{d}\sum_{j=1}^{d}\Sigma_{ij}^2 = O\left(d^2\right), \tag{47}$$

$$\mathbb{E}V^{l2}(\lambda) = \frac{b^2\lambda^4}{(b+\lambda)^4}\frac{(N-1)(N-2)}{N^2}\sum_{i=1}^{d_{\text{sub}}}\sum_{j=1}^{d_{\text{sub}}}\Sigma_{ij}^2 = O\left(d_{\text{sub}}^2\right). \tag{48}$$

By Theorem 32, we have $V^{\text{loc}} \leq V^{\text{Me}}(\lambda) \leq V^{\text{Avg}}$ and $V^{\text{loc}} \leq V^{\text{l2}}(\lambda)$. And larger λ leads to more fairness. This is because in our settings, only the true parameters $\mathbf{w}_k$ on the clients are different. For `local`, $\mathbf{w}_k^{\text{loc}}$ is an unbiased estimation of $\mathbf{w}_k$. So $f_k^{\text{te}}(\mathbf{w}_k^{\text{loc}}) = f_l^{\text{te}}(\mathbf{w}_l^{\text{loc}})$ for any $k \neq l$. For other methods, $\mathbf{x}_k^{\text{Avg}}$, $\mathbf{x}_k^{\text{Me}}$ and $\mathbf{x}_k^{\text{l2}}$ are all biased. Thus test losses on different clients can vary a lot.

However, it is not easy to compare $\widetilde{\mathsf{var}}(\|\bar{\mathbf{w}} - \mathbf{w}_k\|_2^2)$ and $\widetilde{\mathsf{var}}(\|\bar{\mathbf{w}}_{\cdot,1} - \mathbf{w}_{k,1}\|_2^2)$ directly. If the variance of $\mathbf{w}_k$ concentrates on the the first d_{sub} dimensions, $\widetilde{\mathsf{var}}(\|\bar{\mathbf{w}}_{\cdot,1} - \mathbf{w}_{k,1}\|_2^2)$ can be larger than $\widetilde{\mathsf{var}}(\|\bar{\mathbf{w}} - \mathbf{w}_k\|_2^2)$.

To gain more intuition, we further assume $\mathbf{w}_k$ are i.i.d. and distributed as $\mathcal{N}(\mu_w, \mathbf{\Sigma}_w)$. Then Theorem 32 implies that $\mathbb{E}V^{\text{loc}} \leq \mathbb{E}V^{\text{l2}}(\lambda) \leq \mathbb{E}V^{\text{Me}}(\lambda) \leq \mathbb{E}V^{\text{Avg}}$.

Now we give the proof of Theorem 32.

Proof [Proof of Theorem 32] From the definition of $\widetilde{\mathsf{var}}$ in (44), the equalities in (45) are easy to check. For the remaining results, we first give an equivalent form of $\widetilde{\mathsf{var}}(\|\bar{\mathbf{w}} - \mathbf{w}_k\|_2^2)$ when treating the $\mathbf{w}_k$ as fixed.

$$\begin{aligned}
\widetilde{\mathsf{var}}(\|\bar{\mathbf{w}} - \mathbf{w}_k\|_2^2) &\overset{(a)}{=} \frac{1}{N}\sum_{k=1}^{N}\|\bar{\mathbf{w}} - \mathbf{w}_k\|_2^4 - \left(\frac{1}{N}\sum_{k=1}^{N}\|\bar{\mathbf{w}} - \mathbf{w}_k\|_2^2\right)^2 \\
&\overset{(b)}{=} \frac{1}{N^2}\left[(N-1)\sum_{k=1}^{N}\|\mathbf{w}_k - \bar{\mathbf{w}}\|_2^4 - \sum_{k=1}^{N}\sum_{l=1,l\neq k}^{N}\|\mathbf{w}_k - \bar{\mathbf{w}}\|_2^2\|\mathbf{w}_l - \bar{\mathbf{w}}\|_2^2\right] \\
&\overset{(c)}{=} \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1,l\neq k}^{N}\frac{\left(\|\mathbf{w}_k - \bar{\mathbf{w}}\|_2^2 - \|\mathbf{w}_l - \bar{\mathbf{w}}\|_2^2\right)^2}{2} \\
&\overset{(d)}{=} \frac{1}{N^2}\sum_{k=1}^{N}\sum_{l=1,l\neq k}^{N}\langle\mathbf{w}_k - \mathbf{w}_l, \mathbf{w}_k + \mathbf{w}_l - 2\bar{\mathbf{w}}\rangle^2,
\end{aligned} \tag{49}$$

where (a) follows from the definition of $\widetilde{\mathsf{var}}$ in (44), (b) and (d) result from expanding the squared terms, and (c) is based on the fact that, for any real numbers $a_1, a_2, \ldots, a_N$, the following equality holds:

$$\begin{aligned}
\sum_{k=1}^{N}\sum_{l=1,l\neq k}^{N}\frac{(a_k - a_l)^2}{2} &= \sum_{k=1}^{N}\sum_{l=1,l\neq k}^{N}\frac{a_k^2 + a_l^2 - 2a_k a_l}{2} \\
&= \frac{N-1}{2}\sum_{k=1}^{N}a_k^2 + \frac{N-1}{2}\sum_{l=1}^{N}a_l^2 - \sum_{k=1}^{N}\sum_{l=1,l\neq k}^{N}a_k a_l \\
&= (N-1)\sum_{k=1}^{N}a_k^2 - \sum_{k=1}^{N}\sum_{l=1,l\neq k}^{N}a_k a_l.
\end{aligned}$$

If we further assume $\mathbf{w}_k$ are i.i.d. from $\mathcal{N}(\mu_w, \mathbf{\Sigma}_w)$, one can check that

$$\begin{aligned}
&\operatorname{cov}(\mathbf{w}_k - \mathbf{w}_l, \mathbf{w}_k + \mathbf{w}_l - 2\bar{\mathbf{w}}) \\
&= \mathbb{E}(\mathbf{w}_k - \mathbf{w}_l)(\mathbf{w}_k + \mathbf{w}_l - 2\bar{\mathbf{w}})^\top - \mathbb{E}(\mathbf{w}_k - \mathbf{w}_l)\mathbb{E}(\mathbf{w}_k + \mathbf{w}_l - 2\bar{\mathbf{w}})^\top \\
&= \mathbb{E}\mathbf{w}_k\mathbf{w}_k^\top - \mathbb{E}\mathbf{w}_l\mathbf{w}_l^\top - 2\mathbb{E}\mathbf{w}_k\bar{\mathbf{w}}^\top + 2\mathbb{E}\mathbf{w}_l\bar{\mathbf{w}}^\top = \mathbf{0},
\end{aligned}$$

where the last step is due to the symmetry between k and l. Note that $(\mathbf{w}_k - \mathbf{w}_l, \mathbf{w}_k + \mathbf{w}_l - 2\bar{\mathbf{w}})$ also follows a Gaussian distribution. Then unrelatedness implies independence. Thus, $\mathbf{w}_k - \mathbf{w}_l$ and $\mathbf{w}_k + \mathbf{w}_l - 2\bar{\mathbf{w}}$ are independent. To calculate $\mathbb{E}\,\widetilde{\mathsf{var}}(\|\bar{\mathbf{w}} - \mathbf{w}_k\|_2^2)$, it suffices to examine $\mathbb{E}\,\langle \mathbf{w}_k - \mathbf{w}_l, \mathbf{w}_k + \mathbf{w}_l - 2\bar{\mathbf{w}}\rangle^2$ for $k \neq l$. Take $X_1 = \mathbf{w}_k - \mathbf{w}_l$ and $X_2 = \mathbf{w}_k + \mathbf{w}_l - 2\bar{\mathbf{w}}$ for short. Due to that the $\mathbf{w}_k$ are i.i.d. one can check $\mathbb{E}X_1 = \mathbb{E}X_2 = \mathbf{0}$, $\mathsf{var}(X_1) = \mathsf{var}(\mathbf{w}_k) + \mathsf{var}(\mathbf{w}_l) = 2\mathbf{\Sigma}_w$ and

$$\mathsf{var}(X_2) = \left(1 - \frac{2}{N}\right)^2 \mathsf{var}(\mathbf{w}_k) + \left(1 - \frac{2}{N}\right)^2 \mathsf{var}(\mathbf{w}_l) + \frac{4}{N^2} \sum_{i=1, i\neq k,l}^{N} \mathsf{var}(\mathbf{w}_i) = \left(2 - \frac{4}{N}\right)\mathbf{\Sigma}_w.$$

In other words, $X_1 \sim \mathcal{N}(\mathbf{0}, 2\mathbf{\Sigma}_w)$ and $X_2 \sim \mathcal{N}(\mathbf{0}, (2 - 4/N)\mathbf{\Sigma}_w)$. Moreover, both the distributions are independent of the choice of k and l. Due to the independence between X_1 and X_2, we have

$$\begin{aligned}
\mathbb{E}\,\langle X_1, X_2\rangle^2 &= \mathbb{E}\mathrm{tr}(X_2^\top X_1 X_1^\top X_2) = \mathbb{E}\mathrm{tr}(X_1 X_1^\top X_2 X_2^\top) = \mathrm{tr}(\mathbb{E}X_1X_1^\top \mathbb{E}X_2X_2^\top) \\
&= 4\left(1 - \frac{2}{N}\right)\mathrm{tr}(\mathbf{\Sigma}_w^2) = 4\left(1 - \frac{2}{N}\right)\sum_{i=1}^{d}\sum_{j=1}^{d}\Sigma_{ij}^2.
\end{aligned}$$

Plugging this equality into (49) yields

$$\mathbb{E}\,\widetilde{\mathsf{var}}(\|\bar{\mathbf{w}} - \mathbf{w}_k\|_2^2) = \frac{4(N-1)(N-2)}{N^2}\sum_{i=1}^{d}\sum_{j=1}^{d}\Sigma_{ij}^2.$$

It follows that (46) and (47) hold. The proof of (48) is similar to that of (47), with the only difference being the dimension: the term d in (47) is replaced by d_{sub} in (48). ■

Appendix C. Experimental Details

The data sets, corresponding models and tasks are summarized in Table 3 below. The performance of `lp-proj-1` and `lp-proj-2` are evaluated on both convex and non-convex models across a set of FL benchmarks, including both synthetic and real data sets.

The synthetic data sets are generated following the setup in Li et al. (2020a), we denote it as `Synthetic(`α`, `β`)`, where α controls how much local models differ from each other and β controls how much the local data for each client differs from that of other clients. Specifically, the synthetic samples $(\boldsymbol{X}_k, y_k)$ are generated from the model $y = \arg\max(\mathrm{softmax}(\boldsymbol{W}_k x + \boldsymbol{b}_k))$ with $x \in \mathbb{R}^{60}$, $\boldsymbol{W} \in \mathbb{R}^{10\times 60}$ and $\boldsymbol{b}_k \in \mathbb{R}^{10}$, where $\boldsymbol{X}_k \in \mathbb{R}^{n_k \times 60}$ and $y_k \in \mathbb{R}^{n_k}$. Each entry of $\boldsymbol{W}_k$ and $\boldsymbol{b}_k$ are modeled as $N(\mu_k, 1)$ with $\mu_k \sim (0, \alpha)$, and $(x_k)_j \sim N(v_k, \frac{1}{j^{1.2}})$ with $v_k \sim N(B_k, 1)$ and $B_k \sim N(0, \beta)$.

Data sets	# of Clients	Average Sample Size for each Client	Tasks	Partitions	Models
`Synthetic(0, 0)`	100	202	10-class classification	⋆	logistic
`Synthetic(1, 1)`	100	202	10-class classification	⋆	logistic
`EMNIST`	248	2000	62-class classification	10 classes to each client	2-hidden-layers NN
`CIFAR`	200	200	10-class classification	2 classes to each client	CNN
`MNIST`	100	434	10-class classification	2 classes to each client	1-hidden-layer NN
`FASHIONMNIST`	100	480	10-class classification	2 classes to each client	CNN

Table 3: Summary of data sets and models.

Neural Network Architecture for the Models Used in Numerical Experiments.

- **1-hidden-layer NN for MNIST**: One hidden fully-connected layer with 100 neurons. We use ReLU as the activation function.

- **2-hidden-layer NN for EMNIST**: Two hidden fully-connected layers, each with 100 neurons. For each FC layer, ReLU is used as the activation function.

- **CNN for CIFAR**: The neural network used in our experiment consists of two convolutional layers and three fully-connected layers. The architecture for each layer is listed as follows:

  - Convolutional layer 1: input_channel: 3, output_channel: 6, kernel_size: 5.
  - Convolutional layer 2: input_channel: 6, output_channel: 16, kernel_size: 5.
  - Fully-connected layer 1: input_features: 400, output_features: 120.
  - Fully-connected layer 2: input_features: 120, output_features: 84.
  - Fully-connected layer 3: input_features: 84, output_features: 10.

  For each convolutional layer, we would first apply a ReLU activation function right after the convolution, and then apply a max pooling with $kernel\_size = 2, stride = 2$ to extract the feature map. Besides, for the fully-connected layers, we use ReLU as the activation function.

- **CNN for FASHIONMNIST**: The neural network used for `FASHIONMNIST` data set in our experiment is modified from He et al. (2016), which consists of a normal convolutional layer, two resnet blocks and finally a fully connected layer. The architecture for each layer is listed as follows:

  - Convolutional layer: input_channel: 1, output_channel: 64, kernel_size: 7, stride: 2, padding: 3. Right after the convolution, we apply a batch normalization layer to standardize the features, and then the ReLU function is applied as the activation function, and finally, a max pooling layer with $kernel\_size = 3, stride = 2, padding = 1$ is applied to extract the feature map.
  - Resnet block 1: input_channels: 64, output_channels: 64, number of residuals: 2.
  - Resnet block 2: input_channels: 64, output_channels: 128, number of residuals: 2.
  - Fully-connected layer: input_features: 128, output_features: 10.

Furthermore, we apply average pooling right after resnet block 2 to extract the feature map before feeding to the fully connected layer.

For implementation details, please refer to the source code provided in `https://github.com/desternylin/perfed`.

Total Number of Parameters for the Full Model.

- **Synthetic**: 610.
- **MNIST**: 79510.
- **CIFAR**: 62006.
- **EMNIST**: 94862.
- **FASHIONMNIST**: 678794.

Computing Resource for Numerical Experiments. All of our experiments are performed on GPUs. Specifically, every single experiment (a competing method with its given parameter setting and model) is performed on a single GPU, where the type of GPU is one of the following two:

- NVIDIA TITAN RTX with 24220MB memory, driver version: 470.63.01, CUDA version: 11.4.
- NVIDIA GeForce RTX 2080 Ti with 11019MB memory, driver version: 470.63.01, CUDA version: 11.4.

C.1 Competing Methods

Several state-of-the-art methods in the literature aiming for different purposes such as personalization, robustness and communication efficiency are considered in our experiment. We list and provide a brief description of these methods below.

- **FedAvg** (McMahan et al., 2017), which learns a shared model by averaging the locally-computed model updates in each communication round. This is a baseline algorithm in the FL literature, but it would probably suffer from the statistical heterogeneity among clients.
- **Local**, which trains a local model for each client separately. This algorithm does not have the communication burden issue, but may perform poorly when there is little local data.
- **Ditto** (Li et al., 2021b), which considers two overarching tasks: the global objectives and the local objectives, and uses a regularization term that encourages the personalized model to be close to the optimal global model.
- **LG-FedAvg** (Liang et al., 2020), which proposes to learn useful and compact features from the raw data locally and the central server only aggregates the learned representations to improve communication efficiency and get better personalization performance.

- **RSA** (Li et al., 2019), which incorporates the objective function with an L^p regularizer to robustify the learning task and mitigate the negative effects of Byzantine attacks.

- **pFedMe** (Dinh et al., 2020), which considers a bi-level problem that concerns global and local objectives respectively. The main difference of `pFedMe` and `Ditto` is that when considering the global objective, `pFedMe` considers the whole loss function including the regularizer, while `Ditto` excludes the regularizer in the global level.

- **Per-FedAvg** (Fallah et al., 2020), which applies MAML (Finn et al., 2017) to personalize federated models with a Hessian-product approximation to approximate the second-order gradients.

- **Sketch** (Ivkin et al., 2019), which carries out distributed SGD by communicating count sketches instead of full gradients to reduce communication costs. However, in our experiments, we find that the size of the sketches should be relatively large to retain accuracy performance in heterogeneous networks.

- **LBGM** (Look-Back Gradient Multiplier) (Azam et al., 2021), which exploits the low-rank property of the gradient space to enable gradient recycling between model update rounds of federated learning.

- **QSGD** (Quantized SGD) (Alistarh et al., 2017), which quantizes each component by randomized rounding to a discrete set of values before message transmission. Furthermore, it employs efficient lossless code for quantized gradients, which exploits their statistical properties to generate efficient encodings.

- **DGC** (Deep Gradient Compression) (Lin et al., 2018), which compresses the gradient with momentum correction and local gradient clipping on top of the gradient sparsification. What's more, to overcome the staleness problem caused by reduced communication, momentum factor masking and warmup training are also used.

C.2 Parameter Settings

For each competing algorithm, different hyper-parameters need to be tuned. We provide two or three candidates for each hyper-parameter and perform a grid search on all the possible combinations based on the accuracy performance on the validation data set. The tuning hyper-parameter and their corresponding candidates for each algorithm are listed as follows.

- **FedAvg**: local learning rate: $\{0.05, 0.1, 0.5\}$, rounds for local update: $\{1, 5\}$.

- **Local**: local learning rate: $\{0.05, 0.1, 0.5\}$.

- **Ditto**: local learning rate: $\{0.05, 0.1, 0.5\}$, personalization model learning rate: $\{0.01, 0.05, 0.1\}$, $\lambda : \{0.1, 1, 10\}$, local computation rounds $R : \{1, 5\}$.

- **LG-FedAvg**: local learning rate: $\{0.05, 0.1, 0.5\}$, rounds for local update: $\{1, 5\}$.

- **RSA**: local learning rate: $\{0.05, 0.1, 0.5\}$, personalization model learning rate: $\{0.01, 0.05, 0.1\}$, $\lambda : \{0.1, 1, 10\}$, local computation rounds $R : \{1, 5\}$.

- **pFedMe**: local learning rate: $\{0.05, 0.1, 0.5\}$, personalization model learning rate: $\{0.01, 0.05, 0.1\}$, $\lambda : \{0.1, 1, 10\}$, local computation rounds $R : \{1, 5\}$.
- **Per-FedAvg**: local learning rate: $\{0.05, 0.1, 0.5\}$, personalization model learning rate: $\{0.01, 0.05, 0.1\}$.
- **Sketch**: local learning rate: $\{0.05, 0.1, 0.5\}$, columns of the sketch: $\{0.02, 0.05\} \times$ dimension of the full model, rows of the sketch: $\{0.005, 0.01\} \times$ dimension of the full model, k for the recovered k-sparse gradient: $\{0.01, 0.05\} \times$ dimension of the full model.
- **lp-proj-1**: local learning rate: $\{0.05, 0.1, 0.5\}$, personalization model learning rate: $\{0.01, 0.05, 0.1\}$, $\lambda : \{0.1, 1, 10\}$, local computation rounds $R : \{1, 5\}$.
- **lp-proj-2**: local learning rate: $\{0.05, 0.1, 0.5\}$, personalization model learning rate: $\{0.01, 0.05, 0.1\}$, $\lambda : \{0.1, 1, 10\}$, local computation rounds $R : \{1, 5\}$.
- **LBGM**: learning rate: $\{0.05, 0.1, 0.5\}$, local computation rounds $R : \{1, 5\}$, look-back phase (LBP) error threshold $\delta^{\text{thre}} : \{0.2, 0.5, 0.8\}$.
- **QSGD**: learning rate: $\{0.05, 0.1, 0.5\}$, quantization level: $\{5, 10, 15\}$, bucket size: $\{500, 1000, 2000\}$.
- **DGC**: learning rate: $\{0.05, 0.1, 0.5\}$, initial sparsity level: $\{0.25, 0.5, 0.75\}$, sparsity rising level during warm-up training: $\{0.75, 0.5, 0.25\}$.

Other parameters shared by all algorithms:

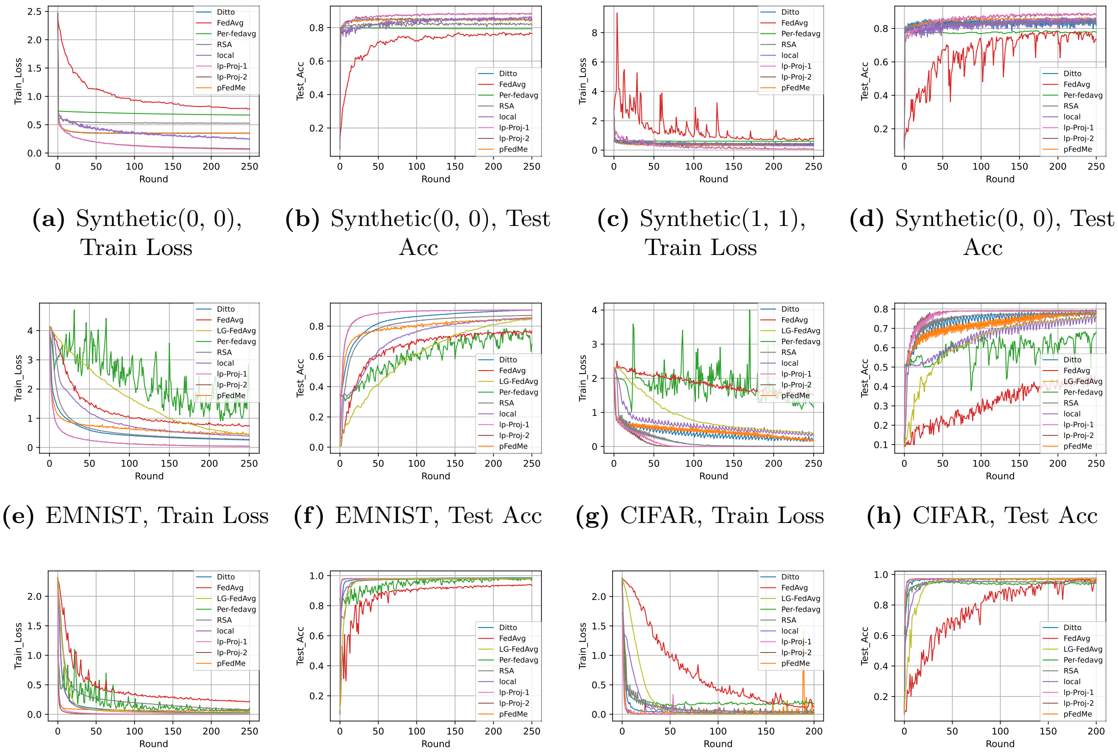
- # of clients particiate in each communication: 10% × total # of clients.
- Accuracy level ν for inner loop for personalization methods: 10^{-10}.
- Batch size for local SGD: 64.
- Projection dimension d_{sub} for `lp-proj-1` and `lp-proj-2`: `Synthetic`: 21, `EMNIST`: 80, `CIFAR`: 60, `MNIST`: 50, `FASHIONMNIST`: 600. The projection dimension for each data set and each model is determined by the full model size and communication budget, and we show theoretically (Lemma 11) that the accuracy performance only has mild dependence on the projection dimension.
- # of repeated experiments: 10.

C.3 Complete Results on Personalization and Fairness Performance

Table 4 shows complete results on personalization performance in terms of train loss and test accuracy and performance fairness in terms of variance of the above two metrics. Figure 5 displays the training loss and test accuracy evolution as the training proceeds. We can see that `lp-proj-1` and `lp-proj-2` own better performance with lower train loss, higher test accuracy and lower variance across clients.

Data set	method	Train Loss	Train Loss Var	Test Acc	Test Acc Var
Synthetic(0, 0)	Ditto	0.3500 ± 0.0038	0.0780 ± 0.0020	0.8569 ± 0.0012	0.0178 ± 0.0005
	pFedMe	0.3542 ± 0.0013	0.0785 ± 0.0009	0.8580 ± 0.0015	0.0178 ± 0.0007
	Per-fedavg	0.6986 ± 0.0184	0.2106 ± 0.0085	0.7977 ± 0.0010	0.0410 ± 0.0006
	FedAvg	0.7988 ± 0.0114	0.2815 ± 0.0112	0.7714 ± 0.0010	0.0455 ± 0.0023
	local	0.2522 ± 0.0045	0.0451 ± 0.0016	0.8665 ± 0.0016	0.0159 ± 0.0007
	lp-proj-1	**0.0769** ± 0.0097	**0.0048** ± 0.0012	**0.8868** ± 0.0010	0.0106 ± 0.0003
	lp-proj-2	0.0818 ± 0.0041	0.0053 ± 0.0005	0.8867 ± 0.0013	**0.0105** ± 0.0003
	RSA	0.5319 ± 0.0075	0.1466 ± 0.0034	0.8314 ± 0.0019	0.0265 ± 0.0008
Synthetic(1, 1)	Ditto	0.3431 ± 0.0165	0.1596 ± 0.0488	0.8615 ± 0.0011	0.0193 ± 0.0006
	pFedMe	0.3010 ± 0.0029	0.0660 ± 0.0014	0.8666 ± 0.0008	0.0170 ± 0.0004
	Per-fedavg	0.6015 ± 0.0151	0.2194 ± 0.0193	0.7925 ± 0.0046	0.0465 ± 0.0022
	FedAvg	0.6938 ± 0.0147	0.3392 ± 0.0214	0.7875 ± 0.0025	0.0480 ± 0.0023
	local	0.2969 ± 0.0157	0.1283 ± 0.0439	0.8675 ± 0.0018	0.0177 ± 0.0007
	lp-proj-1	**0.0614** ± 0.0143	0.0162 ± 0.0191	**0.8954** ± 0.0019	**0.0123** ± 0.0008
	lp-proj-2	0.0679 ± 0.0068	**0.0074** ± 0.0042	0.8932 ± 0.0018	0.0125 ± 0.0009
	RSA	0.4547 ± 0.0075	0.1271 ± 0.0032	0.8416 ± 0.0015	0.0242 ± 0.0009
EMNIST	Ditto	0.2499 ± 0.0032	0.0066 ± 0.0001	**0.9089** ± 0.0008	**0.0016** ± 0.0001
	pFedMe	0.4397 ± 0.0062	0.0301 ± 0.0092	0.8556 ± 0.0012	0.0035 ± 0.0004
	Per-fedavg	0.9061 ± 0.0882	2.1828 ± 2.7274	0.7944 ± 0.0083	0.0104 ± 0.0011
	FedAvg	0.7219 ± 0.0119	0.0300 ± 0.0036	0.7713 ± 0.0029	0.0070 ± 0.0004
	local	0.3903 ± 0.0013	0.0110 ± 0.0017	0.8566 ± 0.0008	0.0022 ± 0.0001
	lp-proj-1	**0.0389** ± 0.0036	**0.0039** ± 0.0003	0.9067 ± 0.0003	0.0017 ± 0.0001
	lp-proj-2	0.0448 ± 0.0022	**0.0039** ± 0.0002	0.9070 ± 0.0001	0.0017 ± 0.0000
	RSA	0.2740 ± 0.0054	0.0066 ± 0.0004	0.8714 ± 0.0011	0.0019 ± 0.0001
	LG-FedAvg	0.4500 ± 0.0194	0.1624 ± 0.0691	0.8453 ± 0.0042	0.0089 ± 0.0015
CIFAR	Ditto	0.1463 ± 0.0335	0.0232 ± 0.0128	0.7909 ± 0.0084	0.0110 ± 0.0008
	pFedMe	0.1837 ± 0.0262	0.0311 ± 0.0072	0.7913 ± 0.0034	0.0100 ± 0.0006
	Per-fedavg	1.0378 ± 0.1614	0.8320 ± 1.0412	0.7257 ± 0.0220	0.0183 ± 0.0022
	FedAvg	1.4739 ± 0.0198	0.0438 ± 0.0092	0.4594 ± 0.0091	0.0173 ± 0.0026
	local	0.3101 ± 0.0098	0.0409 ± 0.0021	0.7688 ± 0.0026	0.0131 ± 0.0006
	lp-proj-1	0.0381 ± 0.0296	0.0077 ± 0.0066	**0.7922** ± 0.0017	**0.0097** ± 0.0003
	lp-proj-2	**0.0043** ± 0.0105	0.0009 ± 0.0024	0.7910 ± 0.0015	0.0099 ± 0.0005
	RSA	0.0073 ± 0.0015	**0.0000** ± 0.0000	0.7768 ± 0.0048	**0.0097** ± 0.0008
	LG-FedAvg	0.4231 ± 0.0182	0.0352 ± 0.0028	0.7523 ± 0.0055	0.0134 ± 0.0009
MNIST	Ditto	0.0266 ± 0.0010	0.0001 ± 0.0000	**0.9863** ± 0.0004	**0.0003** ± 0.0000
	pFedMe	0.0511 ± 0.0037	0.0006 ± 0.0001	0.9824 ± 0.0005	0.0005 ± 0.0000
	Per-fedavg	0.0555 ± 0.0011	0.0010 ± 0.0004	0.9831 ± 0.0005	0.0004 ± 0.0000
	FedAvg	0.2099 ± 0.0013	0.0029 ± 0.0002	0.9416 ± 0.0009	0.0015 ± 0.0001
	local	0.0204 ± 0.0065	0.0002 ± 0.0001	0.9822 ± 0.0001	0.0004 ± 0.0000
	lp-proj-1	0.0101 ± 0.0046	**0.0000** ± 0.0000	0.9822 ± 0.0002	0.0004 ± 0.0000
	lp-proj-2	**0.0060** ± 0.0052	**0.0000** ± 0.0000	0.9825 ± 0.0002	0.0004 ± 0.0000
	RSA	0.0829 ± 0.0032	0.0010 ± 0.0001	0.9809 ± 0.0002	0.0005 ± 0.0000
	LG-FedAvg	0.0156 ± 0.0019	0.0001 ± 0.0000	0.9821 ± 0.0003	0.0004 ± 0.0000
FASHIONMNIST	Ditto	0.0141 ± 0.0016	0.0004 ± 0.0005	**0.9770** ± 0.0004	**0.0019** ± 0.0001
	pFedMe	0.0076 ± 0.0013	0.0001 ± 0.0001	0.9729 ± 0.0004	0.0024 ± 0.0001
	Per-fedavg	0.1834 ± 0.0383	0.3004 ± 0.1443	0.9500 ± 0.0041	0.0092 ± 0.0013
	FedAvg	0.1129 ± 0.0109	0.0194 ± 0.0042	0.9694 ± 0.0021	0.0029 ± 0.0006
	local	0.0020 ± 0.0016	0.0001 ± 0.0002	0.9748 ± 0.0008	0.0021 ± 0.0001
	lp-proj-1	**0.0002** ± 0.0004	**0.0000** ± 0.0000	0.9752 ± 0.0008	0.0022 ± 0.0002
	lp-proj-2	0.0004 ± 0.0005	**0.0000** ± 0.0001	0.9749 ± 0.0007	0.0021 ± 0.0002
	RSA	0.0908 ± 0.0368	0.0046 ± 0.0035	0.9605 ± 0.0033	0.0039 ± 0.0012
	LG-FedAvg	0.0038 ± 0.0037	0.0001 ± 0.0001	0.9738 ± 0.0005	0.0021 ± 0.0001

Table 4: Complete Result on Personalization and Fairness Performance in terms of Tran Loss and Test Accuracy.



(a) Synthetic(0, 0), Train Loss

(b) Synthetic(0, 0), Test Acc

(c) Synthetic(1, 1), Train Loss

(d) Synthetic(0, 0), Test Acc

(e) EMNIST, Train Loss

(f) EMNIST, Test Acc

(g) CIFAR, Train Loss

(h) CIFAR, Test Acc

(i) MNIST, Train Loss

(j) MNIST, Test Acc

(k) FASHIONMNIST, Train Loss

(l) FASHIONMNIST, Test Acc

Figure 5: Personalization performance of competing methods.

C.4 Complete Results on Communication Efficiency

Table 5 shows complete results on communication performance in terms of test accuracy and communication bytes. For a fair comparison, we personalize the gradient compression methods, i.e., `Sketch, LBGM, QSGD` and `DGC`, which are not personalization algorithms in the original literature. We use a simple meta-learning framework (Finn et al., 2017; Fallah et al., 2020), which uses the collaboratively trained global model as an initialization and performs gradient updates with respect to the client's own loss function to obtain its personalized model. From the comparison result, we can see that, given a communication budget of bytes, `lp-proj-1` and `lp-proj-2` achieve the highest test accuracy. On the other hand, given a target test accuracy, these two approaches need the least bytes for communication, and the compression rate could be up to `1000x`.

C.5 Complete Results on Robustness

Complete results for different methods under various kinds and various levels of Byzantine attacks are shown in Table 6, 7 and 8, and complete results under data poison attack is shown in Table 9. `lp-proj-1` and `lp-proj-2` show stable performance and are the most robust to different attacks.

Data set	Method	Bytes Budget	Test Acc	Target Acc	Used Bytes
Synthetic(0, 0)	FedAvg	328020	0.625 ± 0.006	0.6	597800 ± 0
	Sketch	328020	0.456 ± 0.020	0.6	⋆ ± ⋆
	lp-proj-1	328020	0.885 ± 0.002	0.6	**4620** ± 0
	lp-proj-2	328020	**0.888** ± 0.001	0.6	**4620** ± 0
	LBGM	328020	0.815 ± 0.007	0.6	12200 ± 23578
	QSGD	328020	0.115 ± 0.069	0.6	923350 ± 174383
	DGC	328020	⋆ ± ⋆	0.6	372000 ± 186123
Synthetic(1, 1)	FedAvg	401940	0.516 ± 0.028	0.6	523380 ± 34268
	Sketch	401940	0.554 ± 0.017	0.6	⋆ ± ⋆
	lp-proj-1	401940	**0.892** ± 0.002	0.6	**4620** ± 0
	lp-proj-2	401940	0.888 ± 0.001	0.6	**4620** ± 0
	LBGM	401940	0.858 ± 0.042	0.6	11200 ± 371717
	QSGD	401940	0.625 ± 0.018	0.6	46950 ± 905787
	DGC	401940	0.175 ± 0.208	0.6	58400 ± 184500
EMNIST	FedAvg	4236900	⋆ ± ⋆	0.7	445851400 ± 16265444
	Sketch	4236900	⋆ ± ⋆	0.7	⋆ ± ⋆
	lp-proj-1	4236900	**0.906** ± 0.000	0.7	**174720** ± 10699
	lp-proj-2	4236900	**0.906** ± 0.000	0.7	196560 ± 6552
	LG-FedAvg	4236900	0.071 ± 0.016	0.7	230786010 ± 6629787
	LBGM	4236900	⋆ ± ⋆	0.7	206307624 ± 37552057
	QSGD	4236900	⋆ ± ⋆	0.7	173663720 ± 101397671
	DGC	4236900	⋆ ± ⋆	0.7	⋆ ± ⋆
CIFAR	FedAvg	1029600	⋆ ± ⋆	0.4	392870016 ± 33519046
	Sketch	1029600	⋆ ± ⋆	0.4	2271432000 ± 220100908
	lp-proj-1	1029600	**0.792** ± 0.002	0.4	**26400** ± 0
	lp-proj-2	1029600	0.790 ± 0.002	0.4	**26400** ± 0
	LG-FedAvg	1029600	⋆ ± ⋆	0.4	51369296 ± 10550837
	LBGM	1029600	⋆ ± ⋆	0.4	4898475 ± 41680525
	QSGD	1029600	⋆ ± ⋆	0.4	87514000 ± 24632864
	DGC	1029600	⋆ ± ⋆	0.4	25671000 ± 323640
MNIST	FedAvg	228000	⋆ ± ⋆	0.7	56293080 ± 6828608
	Sketch	228000	⋆ ± ⋆	0.7	146026720 ± 51486427
	lp-proj-1	228000	**0.982** ± 0.000	0.7	**12000** ± 0
	lp-proj-2	228000	**0.982** ± 0.000	0.7	**12000** ± 0
	LG-FedAvg	228000	0.111 ± 0.026	0.7	763560 ± 55540
	LBGM	228000	⋆ ± ⋆	0.7	1590200 ± 5492558
	QSGD	228000	⋆ ± ⋆	0.7	15966000 ± 4693474
	DGC	228000	⋆ ± ⋆	0.7	27579900 ± 2836077
FASHIONMNIST	FedAvg	3384000	⋆ ± ⋆	0.7	1186531912 ± 52998121
	lp-proj-1	3384000	**0.975** ± 0.001	0.7	**144000** ± 0
	lp-proj-2	3384000	**0.975** ± 0.001	0.7	**144000** ± 0
	LG-FedAvg	3384000	0.892 ± 0.010	0.7	1725336 ± 118790
	LBGM	3384000	⋆ ± ⋆	0.7	51588348 ± 120013474
	QSGD	3384000	⋆ ± ⋆	0.7	42696225 ± 252785764
	DGC	3384000	⋆ ± ⋆	0.7	133637400 ± 165916659

Table 5: Complete Result on Communication Performance in terms of Test Accuracy and Communication Bytes. There are two comparisons: one is test accuracy on a given byte budget, and the other is used bytes to achieve a target test accuracy. Under the given bytes budget, a ⋆ on the column "Test Acc" refers to the situation that bytes used in the first iteration of the corresponding algorithm have exceeded the budget. Under target test accuracy, a ⋆ on the column "Used Bytes" means the algorithm could not provide a solution that reaches the target accuracy.

Data set	Method	Clean	10%	20%	50%	80%
Synthetic(0, 0)	Ditto	0.857 (0.018)	0.856 (0.017)	0.851 (0.020)	0.855 (0.017)	0.837 (0.014)
	Global+Mean	0.772 (0.044)	0.558 (0.150)	0.485 (0.161)	0.462 (0.169)	0.446 (0.166)
	Global+Median	0.519 (0.140)	0.558 (0.129)	0.604 (0.121)	0.434 (0.156)	0.471 (0.155)
	Global+Krum	0.235 (0.109)	0.285 (0.127)	0.318 (0.133)	0.298 (0.131)	0.285 (0.122)
	RSA	0.832 (0.026)	**0.881** (**0.011**)	0.879 (**0.011**)	**0.885** (**0.012**)	0.863 (0.008)
	lp-proj-1	**0.888** (**0.010**)	0.868 (0.013)	**0.880** (**0.011**)	0.884 (**0.012**)	**0.869** (**0.010**)
	lp-proj-2	0.887 (**0.010**)	0.865 (0.014)	0.873 (0.012)	0.875 (0.014)	0.858 (0.012)
Synthetic(1, 1)	Ditto	0.863 (0.018)	0.882 (0.015)	0.884 (0.014)	0.885 (0.012)	0.873 (0.012)
	Global+Mean	0.785 (0.051)	0.481 (0.168)	0.440 (0.175)	0.387 (0.171)	0.477 (0.147)
	Global+Median	0.525 (0.142)	0.606 (0.124)	0.655 (0.122)	0.428 (0.163)	0.484 (0.166)
	Global+Krum	0.224 (0.105)	0.294 (0.135)	0.310 (0.139)	0.396 (0.143)	0.241 (0.149)
	RSA	0.844 (0.023)	**0.901** (**0.013**)	0.903 (0.012)	0.906 (**0.010**)	**0.916** (**0.005**)
	lp-proj-1	**0.893** (0.014)	0.890 (0.014)	**0.907** (**0.010**)	**0.908** (**0.010**)	0.914 (**0.005**)
	lp-proj-2	0.891 (**0.013**)	0.890 (0.015)	0.905 (0.011)	0.898 (0.013)	0.907 (0.007)
EMNIST	Ditto	**0.907** (**0.002**)	0.293 (0.004)	0.294 (0.004)	0.289 (0.004)	0.282 (0.003)
	Global+Mean	0.770 (0.007)	0.057 (0.013)	0.058 (0.013)	0.061 (0.013)	0.072 (0.015)
	Global+Median	0.556 (0.015)	0.057 (0.013)	0.058 (0.013)	0.061 (0.013)	0.072 (0.015)
	Global+Krum	0.504 (0.032)	0.057 (0.013)	0.058 (0.013)	0.061 (0.013)	0.072 (0.015)
	RSA	0.872 (**0.002**)	0.293 (0.004)	0.294 (0.004)	0.337 (0.012)	0.431 (0.021)
	lp-proj-1	0.906 (**0.002**)	**0.908** (**0.002**)	**0.905** (**0.002**)	**0.908** (**0.002**)	**0.908** (**0.002**)
	lp-proj-2	**0.907** (**0.002**)	0.900 (**0.002**)	0.902 (**0.002**)	0.904 (**0.002**)	0.907 (**0.002**)
CIFAR	Ditto	**0.796** (0.010)	0.501 (**0.000**)	0.502 (**0.001**)	0.502 (**0.001**)	0.511 (**0.002**)
	Global+Mean	0.456 (0.022)	0.106 (0.042)	0.116 (0.044)	0.115 (0.044)	0.150 (0.052)
	Global+Median	0.247 (0.035)	0.106 (0.042)	0.116 (0.044)	0.115 (0.044)	0.150 (0.052)
	Global+Krum	0.246 (0.038)	0.106 (0.042)	0.116 (0.044)	0.115 (0.044)	0.150 (0.052)
	RSA	0.775 (0.010)	0.539 (0.008)	0.574 (0.013)	0.590 (0.016)	0.595 (0.013)
	lp-proj-1	0.791 (**0.009**)	**0.786** (0.009)	**0.790** (0.009)	**0.797** (0.010)	**0.795** (0.012)
	lp-proj-2	0.792 (**0.009**)	0.783 (0.009)	0.789 (0.010)	0.791 (0.012)	0.788 (0.011)
MNIST	Ditto	**0.986** (**0.000**)	0.529 (0.011)	0.516 (0.009)	0.958 (0.005)	0.939 (0.005)
	Global+Mean	0.942 (0.001)	0.107 (0.042)	0.120 (0.046)	0.113 (0.046)	0.198 (0.055)
	Global+Median	0.808 (0.014)	0.861 (0.006)	0.120 (0.046)	0.113 (0.046)	0.175 (0.057)
	Global+Krum	0.647 (0.062)	0.668 (0.080)	0.120 (0.046)	0.113 (0.046)	0.168 (0.061)
	RSA	0.981 (0.001)	0.980 (**0.000**)	0.981 (0.001)	**0.984** (**0.000**)	0.985 (**0.000**)
	lp-proj-1	0.982 (**0.000**)	0.982 (**0.000**)	**0.982** (**0.000**)	**0.984** (**0.000**)	0.987 (**0.000**)
	lp-proj-2	0.982 (**0.000**)	**0.983** (0.001)	**0.982** (**0.000**)	**0.984** (**0.000**)	**0.989** (**0.000**)
FASHIONMNIST	Ditto	**0.977** (**0.002**)	0.605 (0.020)	0.611 (0.018)	0.630 (0.016)	0.615 (0.013)
	Global+Mean	0.967 (0.004)	0.130 (0.047)	0.151 (0.052)	0.153 (0.048)	0.176 (0.057)
	Global+Median	0.729 (0.033)	0.739 (0.024)	0.119 (0.045)	0.120 (0.046)	0.162 (0.044)
	Global+Krum	0.374 (0.072)	0.413 (0.122)	0.119 (0.045)	0.140 (0.050)	0.192 (0.062)
	RSA	0.960 (0.004)	0.685 (0.033)	0.767 (0.036)	0.775 (0.022)	0.827 (0.018)
	lp-proj-1	0.975 (**0.002**)	**0.973** (**0.002**)	**0.980** (**0.001**)	**0.976** (**0.002**)	**0.976** (**0.002**)
	lp-proj-2	0.974 (**0.002**)	0.971 (0.003)	0.979 (**0.001**)	0.975 (**0.002**)	0.975 (**0.002**)

Table 6: Complete Result on Robustness Performance in terms of test accuracy under same-value attacks. (The number in the parentheses is the corresponding variance.)

Data set	Method	Clean	10%	20%	50%	80%
Synthetic(0, 0)	Ditto	0.857 (0.018)	0.853 (0.017)	0.850 (0.017)	⋆ (⋆)	⋆ (⋆)
	Global+Mean	0.772 (0.044)	0.412 (0.146)	0.324 (0.130)	⋆ (⋆)	⋆ (⋆)
	Global+Median	0.519 (0.140)	0.421 (0.131)	0.425 (0.139)	0.192 (0.087)	⋆ (⋆)
	Global+Krum	0.280 (0.133)	0.304 (0.139)	0.308 (0.157)	0.281 (0.134)	⋆ (⋆)
	RSA	0.832 (0.026)	0.829 (0.025)	0.834 (0.022)	0.881 (0.012)	0.848 (0.011)
	lp-proj-1	**0.888** (**0.010**)	0.884 (0.011)	**0.885** (**0.010**)	**0.885** (**0.010**)	**0.863** (**0.010**)
	lp-proj-2	0.887 (**0.010**)	**0.885** (**0.010**)	0.884 (**0.010**)	⋆ (⋆)	⋆ (⋆)
Synthetic(1, 1)	Ditto	0.863 (0.018)	0.879 (0.015)	0.884 (0.013)	⋆ (⋆)	⋆ (⋆)
	Global+Mean	0.785 (0.051)	0.292 (0.142)	0.243 (0.134)	⋆ (⋆)	⋆ (⋆)
	Global+Median	0.525 (0.142)	0.455 (0.140)	0.436 (0.169)	0.168 (0.092)	⋆ (⋆)
	Global+Krum	0.269 (0.134)	0.287 (0.142)	0.326 (0.141)	0.372 (0.155)	⋆ (⋆)
	RSA	0.844 (0.023)	0.856 (0.023)	0.863 (0.020)	**0.905** (0.010)	**0.920** (**0.004**)
	lp-proj-1	**0.893** (0.014)	**0.905** (**0.011**)	**0.909** (**0.010**)	**0.905** (**0.009**)	0.918 (0.005)
	lp-proj-2	0.891 (**0.013**)	0.902 (0.013)	0.908 (0.011)	⋆ (⋆)	⋆ (⋆)
EMNIST	Ditto	**0.907** (**0.002**)	0.746 (0.004)	0.748 (0.003)	⋆ (⋆)	⋆ (⋆)
	Global+Mean	0.770 (0.007)	0.057 (0.013)	0.072 (0.012)	⋆ (⋆)	⋆ (⋆)
	Global+Median	0.556 (0.015)	0.382 (0.021)	0.392 (0.024)	0.107 (0.005)	⋆ (⋆)
	Global+Krum	0.501 (0.037)	0.452 (0.029)	0.409 (0.031)	0.495 (0.035)	⋆ (⋆)
	RSA	0.872 (**0.002**)	0.501 (0.007)	0.598 (0.006)	**0.905** (**0.002**)	**0.907** (**0.002**)
	lp-proj-1	0.906 (**0.002**)	**0.908** (**0.002**)	**0.910** (**0.002**)	**0.905** (**0.002**)	**0.907** (**0.002**)
	lp-proj-2	**0.907** (**0.002**)	0.907 (**0.002**)	0.907 (**0.002**)	⋆ (⋆)	⋆ (⋆)
CIFAR	Ditto	**0.795** (0.010)	0.746 (0.016)	0.762 (0.015)	⋆ (⋆)	⋆ (⋆)
	Global+Mean	0.456 (0.022)	0.106 (0.042)	0.128 (0.029)	⋆ (⋆)	⋆ (⋆)
	Global+Median	0.247 (0.035)	0.265 (0.039)	0.224 (0.018)	⋆ (⋆)	⋆ (⋆)
	Global+Krum	0.246 (0.038)	0.240 (0.038)	0.290 (0.019)	0.200 (0.059)	⋆ (⋆)
	RSA	0.778 (**0.009**)	0.646 (0.010)	0.613 (0.013)	0.788 (**0.010**)	**0.791** (**0.010**)
	lp-proj-1	0.790 (**0.009**)	**0.795** (0.010)	**0.793** (**0.009**)	**0.801** (**0.010**)	0.788 (0.011)
	lp-proj-2	0.792 (**0.009**)	0.788 (**0.009**)	0.786 (0.010)	⋆ (⋆)	⋆ (⋆)
MNIST	Ditto	**0.986** (**0.000**)	0.981 (**0.000**)	0.981 (**0.000**)	⋆ (⋆)	⋆ (⋆)
	Global+Mean	0.942 (0.001)	0.188 (0.057)	0.296 (0.061)	⋆ (⋆)	⋆ (⋆)
	Global+Median	0.859 (0.007)	0.103 (0.041)	0.817 (0.008)	0.206 (0.032)	⋆ (⋆)
	Global+Krum	0.679 (0.076)	0.668 (0.080)	0.723 (0.045)	0.796 (0.029)	⋆ (⋆)
	RSA	0.981 (0.001)	0.954 (0.006)	0.976 (0.001)	**0.984** (**0.000**)	**0.984** (**0.000**)
	lp-proj-1	0.982 (**0.000**)	**0.982** (**0.000**)	**0.982** (**0.000**)	**0.984** (**0.000**)	**0.984** (**0.000**)
	lp-proj-2	0.982 (**0.000**)	**0.982** (**0.000**)	**0.982** (**0.000**)	⋆ (⋆)	⋆ (⋆)
FASHIONMNIST	Ditto	**0.977** (**0.002**)	0.973 (**0.002**)	0.980 (**0.001**)	⋆ (⋆)	⋆ (⋆)
	Global+Mean	0.967 (0.004)	0.111 (0.043)	0.119 (0.045)	⋆ (⋆)	⋆ (⋆)
	Global+Median	0.729 (0.033)	0.214 (0.031)	0.537 (0.038)	⋆ (⋆)	⋆ (⋆)
	Global+Krum	0.374 (0.072)	0.430 (0.069)	0.511 (0.107)	0.728 (0.065)	⋆ (⋆)
	RSA	0.960 (0.004)	0.891 (0.021)	0.930 (0.011)	0.974 (0.002)	**0.977** (**0.002**)
	lp-proj-1	0.975 (**0.002**)	**0.975** (**0.002**)	**0.981** (**0.001**)	**0.976** (**0.001**)	**0.977** (**0.002**)
	lp-proj-2	0.974 (**0.002**)	0.974 (**0.002**)	**0.981** (**0.001**)	⋆ (⋆)	⋆ (⋆)

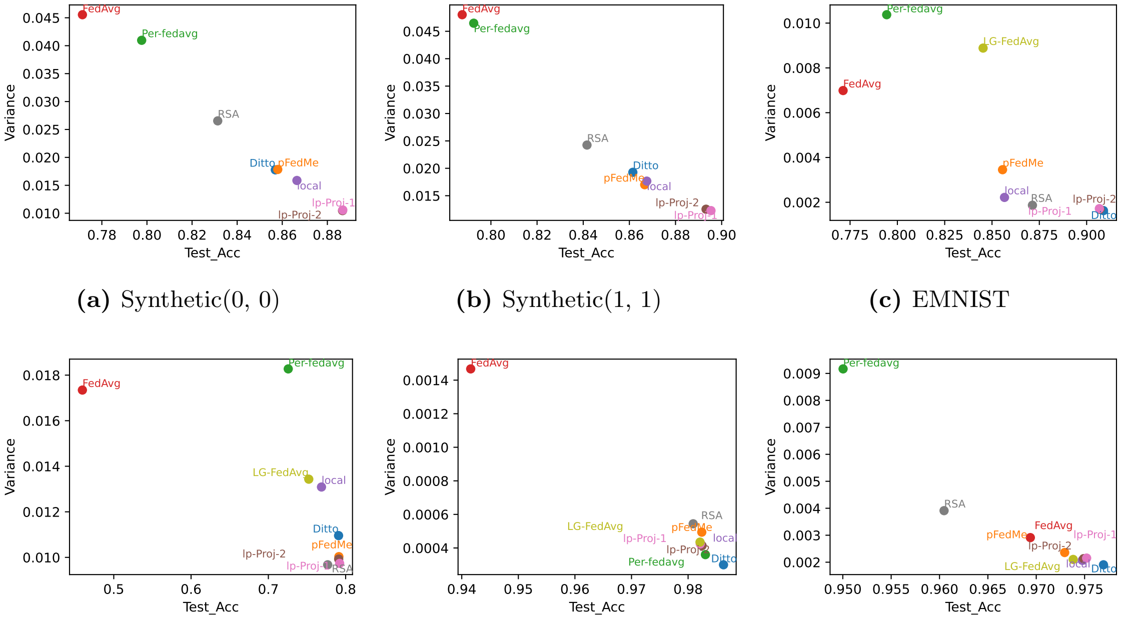
Table 7: Complete Result on Robustness Performance in terms of test accuracy under sign-flipping attacks. (The number in the parentheses is the corresponding variance.) A ⋆ on the cell means that the corresponding algorithm would collapse under the given intensity of the adversarial attack and could not return a solution.

Data set	Method	Clean	10%	20%	50%	80%
Synthetic(0, 0)	Ditto	0.857 (0.018)	0.651 (0.052)	0.674 (0.056)	0.722 (0.045)	0.710 (0.047)
	Global+Mean	0.772 (0.044)	0.174 (0.081)	0.173 (0.080)	0.189 (0.082)	0.246 (0.102)
	Global+Median	0.519 (0.140)	0.124 (0.054)	0.143 (0.071)	0.189 (0.083)	0.204 (0.091)
	Global+Krum	0.235 (0.109)	0.133 (0.072)	0.148 (0.086)	0.208 (0.105)	0.290 (0.086)
	RSA	0.832 (0.026)	0.845 (0.019)	0.851 (0.018)	0.868 (0.017)	0.837 (0.015)
	lp-proj-1	**0.888** (**0.010**)	**0.876** (**0.013**)	**0.880** (**0.011**)	**0.885** (**0.011**)	**0.862** (**0.009**)
	lp-proj-2	0.887 (**0.010**)	0.838 (0.023)	0.846 (0.020)	0.861 (0.018)	0.844 (0.011)
Synthetic(1, 1)	Ditto	0.863 (0.018)	0.694 (0.060)	0.741 (0.051)	0.762 (0.032)	0.795 (0.024)
	Global+Mean	0.785 (0.051)	0.194 (0.089)	0.188 (0.101)	0.196 (0.090)	0.295 (0.143)
	Global+Median	0.525 (0.142)	0.132 (0.059)	0.124 (0.078)	0.231 (0.122)	0.250 (0.144)
	Global+Krum	0.224 (0.105)	0.157 (0.096)	0.159 (0.084)	0.248 (0.116)	0.268 (0.121)
	RSA	0.844 (0.023)	0.886 (0.014)	0.887 (0.014)	0.885 (0.015)	0.902 (0.006)
	lp-proj-1	**0.893** (0.014)	**0.898** (**0.013**)	**0.910** (**0.011**)	**0.905** (**0.011**)	**0.916** (**0.005**)
	lp-proj-2	0.891 (**0.013**)	0.868 (0.020)	0.887 (0.016)	0.878 (0.013)	0.893 (0.008)
EMNIST	Ditto	**0.907** (**0.002**)	0.649 (0.004)	0.673 (0.004)	0.681 (0.006)	0.703 (0.006)
	Global+Mean	0.770 (0.007)	0.068 (0.005)	0.063 (0.005)	0.060 (0.008)	0.059 (0.009)
	Global+Median	0.556 (0.015)	0.061 (**0.001**)	0.079 (**0.002**)	0.081 (**0.002**)	0.062 (0.011)
	Global+Krum	0.504 (0.032)	0.177 (0.005)	0.164 (0.009)	0.181 (0.006)	0.180 (0.006)
	RSA	0.872 (**0.002**)	0.782 (0.003)	0.786 (0.003)	0.820 (**0.002**)	0.844 (0.003)
	lp-proj-1	0.906 (**0.002**)	**0.899** (0.002)	**0.903** (**0.002**)	**0.905** (**0.002**)	**0.907** (**0.002**)
	lp-proj-2	**0.907** (**0.002**)	0.862 (0.002)	0.862 (**0.002**)	0.881 (**0.002**)	0.883 (**0.002**)
CIFAR	Ditto	**0.796** (0.010)	0.668 (0.011)	0.674 (0.011)	0.658 (0.014)	0.604 (0.021)
	Global+Mean	0.456 (0.022)	0.139 (0.010)	0.151 (0.038)	0.146 (0.035)	0.153 (0.033)
	Global+Median	0.247 (0.035)	0.112 (0.011)	0.136 (0.024)	0.159 (0.034)	0.144 (**0.009**)
	Global+Krum	0.246 (0.038)	0.160 (**0.008**)	0.166 (0.013)	0.156 (**0.007**)	0.169 (0.017)
	RSA	0.775 (0.010)	0.731 (0.011)	0.736 (**0.010**)	0.757 (0.009)	0.772 (0.011)
	lp-proj-1	0.791 (**0.009**)	**0.790** (**0.008**)	**0.791** (**0.010**)	**0.797** (0.009)	**0.795** (0.010)
	lp-proj-2	0.792 (**0.009**)	0.775 (0.011)	0.779 (**0.010**)	0.784 (0.010)	0.776 (0.011)
MNIST	Ditto	**0.986** (**0.000**)	0.931 (0.002)	0.928 (0.002)	0.932 (0.003)	0.937 (0.002)
	Global+Mean	0.942 (0.001)	0.460 (0.027)	0.272 (0.040)	0.186 (0.027)	0.210 (0.043)
	Global+Median	0.808 (0.014)	0.862 (0.006)	0.141 (0.038)	0.114 (0.039)	0.207 (0.062)
	Global+Krum	0.647 (0.062)	0.669 (0.062)	0.770 (0.012)	0.778 (0.013)	0.821 (0.013)
	RSA	0.981 (0.001)	0.957 (0.001)	0.963 (0.001)	0.979 (0.001)	0.982 (**0.000**)
	lp-proj-1	0.982 (**0.000**)	**0.981** (**0.000**)	**0.982** (0.001)	0.983 (**0.000**)	0.984 (**0.000**)
	lp-proj-2	0.982 (**0.000**)	0.978 (**0.000**)	0.980 (**0.000**)	**0.984** (**0.000**)	**0.987** (**0.000**)
FASHIONMNIST	Ditto	**0.977** (**0.002**)	0.886 (0.015)	0.880 (0.014)	0.873 (0.017)	0.895 (0.007)
	Global+Mean	0.967 (0.004)	0.167 (0.041)	0.165 (0.040)	0.170 (0.050)	0.174 (0.034)
	Global+Median	0.729 (0.033)	0.650 (0.046)	0.346 (0.018)	0.192 (0.035)	0.192 (0.051)
	Global+Krum	0.374 (0.072)	0.437 (0.117)	0.468 (0.067)	0.494 (0.060)	0.362 (0.012)
	RSA	0.960 (0.004)	0.947 (0.007)	0.959 (0.002)	0.964 (0.002)	0.969 (**0.002**)
	lp-proj-1	0.975 (**0.002**)	**0.973** (**0.002**)	**0.978** (**0.001**)	**0.977** (**0.001**)	**0.974** (**0.002**)
	lp-proj-2	0.974 (**0.002**)	0.964 (0.003)	0.973 (**0.001**)	0.968 (0.002)	0.972 (**0.002**)

Table 8: Complete Result on Robustness Performance in terms of test accuracy under Gaussian attacks. (The number in the parentheses is the corresponding variance.)

Data set	Method	Clean	2%	5%	10%	20%
Synthetic(0, 0)	Ditto	0.857 (0.018)	0.853 (0.019)	0.851 (0.019)	0.749 (0.068)	0.342 (0.123)
	Global+Mean	0.772 (0.044)	0.484 (0.163)	0.304 (0.134)	0.257 (0.129)	0.141 (0.059)
	Global+Median	0.519 (0.140)	0.539 (0.141)	0.552 (0.130)	0.322 (0.118)	0.435 (0.133)
	Global+Krum	0.280 (0.133)	0.288 (0.130)	0.290 (0.134)	0.291 (0.156)	0.300 (0.146)
	RSA	0.832 (0.026)	0.850 (0.019)	0.865 (0.016)	0.876 (0.012)	0.875 (**0.011**)
	lp-proj-1	**0.888** (**0.010**)	**0.886** (**0.011**)	**0.884** (**0.011**)	**0.884** (**0.011**)	**0.881** (**0.011**)
	lp-proj-2	0.887 (**0.010**)	**0.886** (**0.011**)	0.881 (0.012)	0.868 (0.015)	0.866 (0.014)
Synthetic(1, 1)	Ditto	0.863 (0.018)	0.863 (0.019)	0.853 (0.023)	0.810 (0.045)	0.401 (0.184)
	Global+Mean	0.785 (0.051)	0.432 (0.156)	0.253 (0.124)	0.208 (0.111)	0.146 (0.090)
	Global+Median	0.525 (0.142)	0.534 (0.151)	0.554 (0.144)	0.274 (0.132)	0.373 (0.152)
	Global+Krum	0.269 (0.134)	0.277 (0.129)	0.300 (0.145)	0.249 (0.140)	0.290 (0.163)
	RSA	0.844 (0.023)	0.864 (0.019)	0.874 (0.016)	0.894 (0.013)	0.903 (0.010)
	lp-proj-1	**0.893** (0.014)	0.889 (0.013)	**0.901** (**0.012**)	**0.904** (**0.011**)	**0.914** (**0.009**)
	lp-proj-2	0.891 (**0.013**)	**0.896** (**0.012**)	0.893 (**0.012**)	0.895 (0.013)	0.899 (0.013)
EMNIST	Ditto	**0.907** (**0.002**)	0.761 (0.003)	0.778 (0.005)	0.859 (**0.002**)	⋆ (⋆)
	Global+Mean	0.770 (0.007)	0.179 (0.024)	0.110 (0.019)	0.150 (0.019)	⋆ (⋆)
	Global+Median	0.556 (0.015)	0.549 (0.013)	0.564 (0.015)	0.433 (0.012)	0.419 (0.011)
	Global+Krum	0.501 (0.037)	0.454 (0.038)	0.449 (0.022)	0.329 (0.030)	0.321 (0.031)
	RSA	0.872 (**0.002**)	0.832 (**0.002**)	0.825 (**0.002**)	0.871 (**0.002**)	0.878 (**0.002**)
	lp-proj-1	0.906 (**0.002**)	**0.909** (**0.002**)	**0.910** (**0.002**)	0.906 (**0.002**)	0.905 (**0.002**)
	lp-proj-2	**0.907** (**0.002**)	0.907 (**0.002**)	0.907 (**0.002**)	**0.908** (**0.002**)	**0.906** (**0.002**)
CIFAR	Ditto	**0.795** (0.010)	0.750 (0.016)	0.749 (0.015)	0.739 (**0.009**)	0.765 (0.011)
	Global+Mean	0.456 (0.022)	0.102 (0.041)	0.139 (0.033)	0.153 (0.025)	0.155 (0.038)
	Global+Median	0.247 (0.035)	0.252 (0.023)	0.247 (0.025)	0.292 (0.015)	0.288 (0.011)
	Global+Krum	0.246 (0.038)	0.250 (0.045)	0.250 (0.027)	0.301 (0.046)	0.222 (0.019)
	RSA	0.778 (**0.009**)	0.719 (0.011)	0.753 (0.010)	0.739 (0.013)	0.778 (0.011)
	lp-proj-1	0.790 (**0.009**)	**0.795** (0.010)	0.793 (**0.008**)	**0.795** (0.010)	**0.801** (0.010)
	lp-proj-2	0.792 (**0.009**)	0.794 (**0.009**)	**0.794** (**0.008**)	0.786 (**0.009**)	0.789 (**0.009**)
MNIST	Ditto	**0.986** (**0.000**)	**0.983** (**0.000**)	**0.982** (**0.000**)	**0.982** (0.001)	⋆ (⋆)
	Global+Mean	0.942 (0.001)	0.832 (0.007)	0.712 (0.025)	0.627 (0.047)	0.514 (0.028)
	Global+Median	0.859 (0.007)	0.860 (0.006)	0.862 (0.006)	0.857 (0.007)	0.839 (0.006)
	Global+Krum	0.679 (0.076)	0.697 (0.078)	0.659 (0.068)	0.668 (0.080)	0.697 (0.046)
	RSA	0.981 (0.001)	0.978 (0.001)	0.975 (0.001)	0.979 (0.001)	0.979 (**0.001**)
	lp-proj-1	0.982 (**0.000**)	**0.983** (**0.000**)	**0.982** (**0.000**)	**0.982** (**0.000**)	**0.982** (**0.001**)
	lp-proj-2	0.982 (**0.000**)	0.982 (**0.000**)	**0.982** (**0.000**)	**0.982** (**0.000**)	**0.982** (**0.001**)
FASHIONMNIST	Ditto	**0.977** (**0.002**)	0.972 (**0.002**)	**0.973** (**0.002**)	0.964 (0.004)	⋆ (⋆)
	Global+Mean	0.967 (0.004)	0.208 (0.070)	0.181 (0.070)	0.161 (0.055)	⋆ (⋆)
	Global+Median	0.729 (0.033)	0.739 (0.036)	0.720 (0.025)	0.721 (0.029)	0.840 (0.037)
	Global+Krum	0.374 (0.072)	0.480 (0.082)	0.488 (0.100)	0.401 (0.072)	0.581 (0.126)
	RSA	0.960 (0.004)	0.969 (0.003)	0.959 (0.004)	0.965 (0.003)	0.976 (0.002)
	lp-proj-1	0.975 (**0.002**)	**0.974** (**0.002**)	**0.973** (**0.002**)	0.974 (**0.002**)	0.980 (**0.001**)
	lp-proj-2	0.974 (**0.002**)	0.973 (**0.002**)	**0.973** (**0.002**)	**0.975** (**0.002**)	**0.981** (**0.001**)

Table 9: Complete Result on Robustness Performance in terms of test accuracy under data-poison attacks. (The number in the parentheses is the corresponding variance.)



(a) Synthetic(0, 0) **(b)** Synthetic(1, 1) **(c)** EMNIST

(d) CIFAR **(e)** MNIST **(f)** FASHIONMNIST

Figure 6: Complete results of performance fairness of competing methods. (The point closer to the bottom right corner is better.)

C.6 Complete Results on Accuracy and Performance Fairness Trade-off

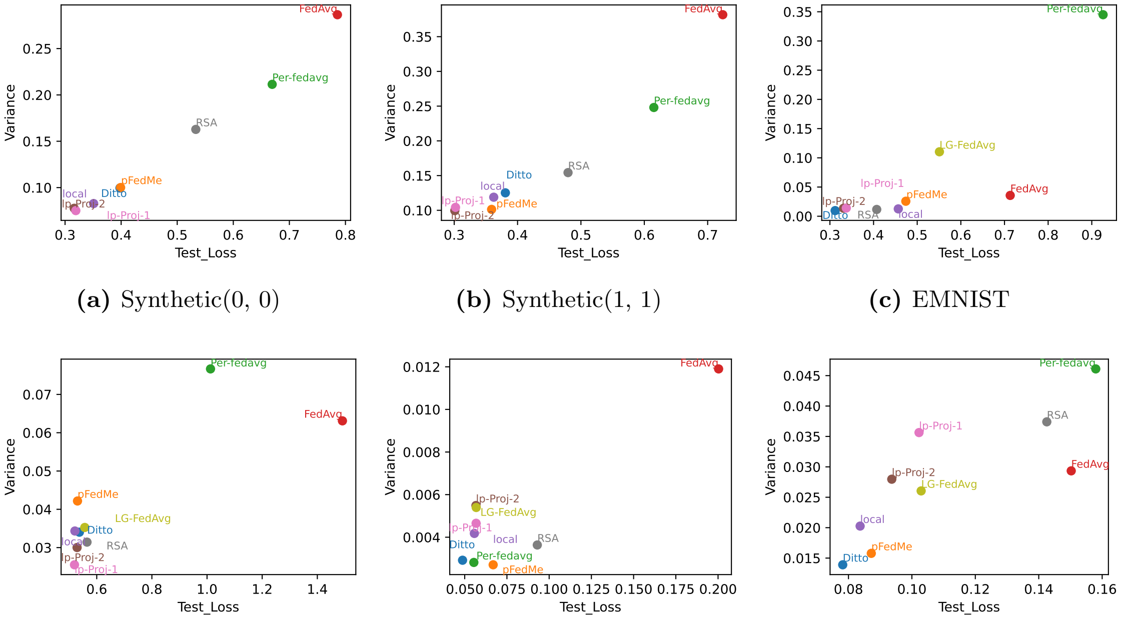
Figure 6 shows complete results for accuracy and performance fairness trade-off on all the data sets used for numerical experiments. `lp-proj-1` and `lp-proj-2` are comparable to other state-of-the-art methods.

C.7 Further Extensions

C.7.1 Data Volume Skewness

In the main paper, we consider the statistical heterogeneity with respect to label skewness, i.e., we distribute the data set among clients so that each client only contains partial classes of the data in multi-classification problems. In practice, there may be another source of statistical heterogeneity, namely data volume skewness. Here we distribute the synthetic data set among $N = 100$ clients in a data volume unbalanced fashion (`Synthetic(0, 0)-unbalance`), i.e., the number of samples among clients follows a power law (Li et al., 2020b).

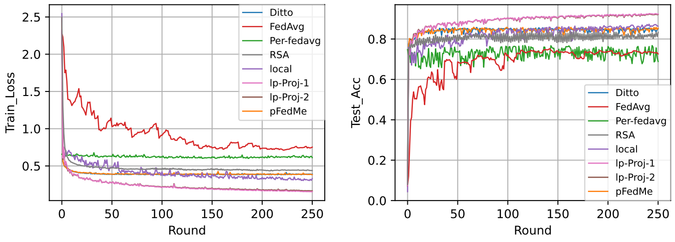
Similar numerical experiments as in the main paper are also performed on the unbalanced data set. The training curves are shown in Figure 8. From the numerical results, we can see that `lp-proj-1`and `lp-proj-2`are also well-performed personalized federated learning methods.



(a) Synthetic(0, 0) **(b)** Synthetic(1, 1) **(c)** EMNIST

(d) CIFAR **(e)** MNIST **(f)** FASHIONMNIST

Figure 7: Complete results of performance fairness in terms of variance of test losses versus the corresponding test loss of competing methods. (The point closer to the bottom left corner is better.)



(a) Synthetic(0, 0)-unbalance, Train Loss **(b)** Synthetic(0, 0)-unbalance, Test Accuracy

Figure 8: Personalization performance of competing methods on data-volume-skewed data set.

Personalization and Fairness Performance Table 10 shows the personalization and performance fairness comparison of different state-of-the-art algorithms on the synthetic data-volume-skewed data set. From the comparison, we can see that our proposed methods enjoy lower training loss (a reduction of at least 50%) and higher test accuracy (an improvement of at least 6%), together with smaller corresponding variance (a reduction of at least 40%), i.e., better performance in terms of performance fairness.

Method	Train Loss	Train Loss Var	Test Acc	Test Acc Var
Ditto	0.3860 ± 0.0033	0.1027 ± 0.0239	0.8576 ± 0.0014	0.0230 ± 0.0025
pFedMe	0.3875 ± 0.0018	0.1213 ± 0.0141	0.8587 ± 0.0010	0.0257 ± 0.0014
Per-fedavg	0.6141 ± 0.0076	0.2197 ± 0.0035	0.7732 ± 0.0095	0.0498 ± 0.0057
FedAvg	0.7502 ± 0.0107	0.3246 ± 0.0096	0.7522 ± 0.0022	0.0655 ± 0.0030
local	0.3070 ± 0.0059	0.0806 ± 0.0440	0.8733 ± 0.0013	0.0206 ± 0.0032
lp-proj-1	**0.1556** ± 0.0011	**0.0095** ± 0.0033	**0.9253** ± 0.0007	**0.0122** ± 0.0007
lp-proj-2	0.1664 ± 0.0014	0.0131 ± 0.0050	0.9230 ± 0.0007	**0.0122** ± 0.0010
RSA	0.4440 ± 0.0123	0.1328 ± 0.0041	0.8374 ± 0.0075	0.0314 ± 0.0024

Table 10: Personalization and fairness performance on data volume-skewed data set in terms of train loss and test accuracy and their corresponding variance.

Method	Bytes Budget	Test Acc	Target Acc	Used Bytes
FedAvg	194700	0.566 ± 0.023	0.6	250100 ± 18300
Sketch	194700	0.484 ± 0.085	0.6	511392 ± 272179
lp-proj-1	194700	**0.899** ± 0.002	0.6	**4620** ± 0
lp-proj-2	194700	0.898 ± 0.002	0.6	**4620** ± 0
LBGM	194700	0.457 ± 0.020	0.6	429996 ± 56411
QSGD	194700	0.587 ± 0.072	0.6	248835 ± 93494
DGC	194700	0.735 ± 0.075	0.6	184320 ± 9145

Table 11: Communication efficiency of Different methods on the data-volume-skewed data set in terms of test accuracy and communication bytes.

Communication Efficiency Table 11 shows the communication efficiency comparison of different methods on the data-volume-skewed data set. Under the given bytes budget, our methods show an improvement in terms of test accuracy with at least 32% compared with the best SOTA method.

Robustness Table 12, 13, 14 and 15 show the robustness performance in terms of test accuracy of different state-of-the-art methods. From the comparison, we can see that our proposed methods enjoy stable performance under various adversarial attacks.

C.7.2 Data Poisoning

In the main paper, we consider three kinds of adversarial attacks: same value, sign flip and Gaussian attack, where all of them can be categorized as model update poisoning attacks. For extension, we consider a special case of the data poisoning attack.

- **Data poisoning attacks**: Under the data-poisoning attacks, the training samples on the corrupted clients are poisoned with flipped (if binary) or uniformly random noisy labels. Furthermore, in the communication period, these clients would scale

Method	Clean	10%	20%	50%	80%
Ditto	0.857 (0.024)	0.860 (0.021)	0.860 (0.021)	0.837 (0.019)	0.846 (0.016)
Global+Mean	0.752 (0.069)	0.659 (0.145)	0.594 (0.154)	0.572 (0.144)	0.633 (0.126)
Global+Median	0.640 (0.117)	0.667 (0.125)	0.685 (0.122)	0.623 (0.130)	0.673 (0.147)
Global+Krum	0.426 (0.132)	0.375 (0.121)	0.439 (0.097)	0.481 (0.095)	0.431 (0.082)
RSA	0.839 (0.031)	**0.919** (**0.013**)	0.913 (0.015)	0.906 (0.012)	0.910 (0.007)
lp-proj-1	**0.926** (**0.011**)	0.913 (0.014)	**0.924** (**0.012**)	**0.916** (**0.009**)	**0.924** (**0.006**)
lp-proj-2	0.924 (**0.011**)	0.885 (0.016)	0.903 (0.014)	0.883 (0.011)	0.892 (0.010)

Table 12: Robustness performance in terms of test accuracy on the data-volume-skewed data set under same-value attack.

Method	Clean	10%	20%	50%	80%
Ditto	0.857 (0.024)	0.860 (0.023)	0.861 (0.021)	$\star$ ($\star$)	$\star$ ($\star$)
Global+Mean	0.752 (0.069)	0.587 (0.146)	0.429 (0.120)	$\star$ ($\star$)	$\star$ ($\star$)
Global+Median	0.640 (0.117)	0.605 (0.138)	0.568 (0.128)	0.115 (0.020)	$\star$ ($\star$)
Global+Krum	0.426 (0.132)	0.373 (0.104)	0.301 (0.125)	0.564 (0.122)	$\star$ ($\star$)
RSA	0.839 (0.031)	0.825 (0.035)	0.830 (0.036)	0.913 (0.011)	**0.921** (**0.007**)
lp-proj-1	**0.926** (**0.011**)	**0.926** (**0.013**)	**0.926** (**0.012**)	**0.916** (**0.009**)	**0.921** (**0.007**)
lp-proj-2	0.924 (**0.011**)	0.923 (**0.013**)	0.923 (**0.012**)	$\star$ ($\star$)	$\star$ ($\star$)

Table 13: Robustness performance in terms of test accuracy on the data-volume-skewed data set under sign-flipping attack.

Method	Clean	10%	20%	50%	80%
Ditto	0.857 (0.024)	0.707 (0.073)	0.679 (0.058)	0.693 (0.070)	0.703 (0.058)
Global+Mean	0.752 (0.069)	0.331 (0.055)	0.349 (0.054)	0.436 (0.077)	0.598 (0.103)
Global+Median	0.640 (0.117)	0.254 (0.031)	0.143 (0.063)	0.371 (0.057)	0.573 (0.056)
Global+Krum	0.426 (0.132)	0.128 (0.093)	0.158 (0.043)	0.258 (0.082)	0.465 (0.076)
RSA	0.839 (0.031)	0.851 (0.023)	0.866 (0.020)	0.848 (0.014)	0.875 (0.012)
lp-proj-1	**0.926** (**0.011**)	**0.916** (**0.012**)	**0.919** (**0.013**)	**0.917** (**0.011**)	**0.926** (**0.007**)
lp-proj-2	0.924 (**0.011**)	0.844 (0.028)	0.871 (0.019)	0.846 (0.020)	0.855 (0.020)

Table 14: Robustness performance in terms of test accuracy on the data-volume-skewed data set under Gaussian attack.

Method	Clean	2%	5%	10%	20%
Ditto	0.857 (0.024)	0.853 (0.020)	0.846 (0.022)	0.761 (0.049)	0.322 (0.131)
Global+Mean	0.752 (0.069)	0.618 (0.160)	0.398 (0.104)	0.234 (0.119)	0.275 (0.054)
Global+Median	0.640 (0.117)	0.652 (0.122)	0.661 (0.122)	0.427 (0.128)	0.443 (0.128)
Global+Krum	0.426 (0.132)	0.411 (0.126)	0.494 (0.117)	0.444 (0.145)	0.442 (0.151)
RSA	0.839 (0.031)	0.848 (0.023)	0.871 (0.026)	0.901 (0.015)	0.911 (**0.013**)
lp-proj-1	**0.926** (**0.011**)	0.916 (**0.012**)	**0.919** (0.013)	**0.922** (**0.012**)	**0.915** (**0.013**)
lp-proj-2	0.924 (**0.011**)	**0.923** (**0.012**)	0.902 (**0.012**)	0.883 (0.017)	0.876 (0.017)

Table 15: Robustness performance in terms of test accuracy on the data-volume-skewed data set under data poisoning attack.

their transmitted messages to make dominate the aggregate update. In particular, the scaling parameter is randomly sampled from $\mathcal{N}(0, 20^2)$.

Table 9 shows complete comparison of different state-of-the-art methods under data-poison attacks. It is worth noting that the data poisoning attack is a rather strong attack, hence the fraction of malicious workers we consider range from 2% to 20%, and we find that on the EMNIST and MNIST data set, `Ditto` and `Global+Mean` would explode and fail to return a solution when the fraction of malicious workers reaches 20%. On the other hand, `lp-proj-1`and `lp-proj-2`always enjoy stable performance and are insensitive to the intensity of the attack.

C.7.3 Collaboration Fairness

In the main paper, our consideration of fairness is performance fairness, i.e., the variance of test accuracy across the system. According to Zhou et al. (2021), there are three kinds of fairness in federated learning, i.e., performance fairness, collaboration fairness and model fairness. Here we consider collaboration fairness, whose definition is as follows.

- **Collaboration Fairness** (Lyu et al., 2020): In a federated system, a high-contribution participant should be rewarded with a better-performing local model than a low-contribution participant. Mathematically, fairness can be quantified by the correlation coefficient between the contributions of participants and their respective final model accuracies. Following Lyu et al. (2020), we use the test accuracy under pure local training for each client to quantify their respective contributions.

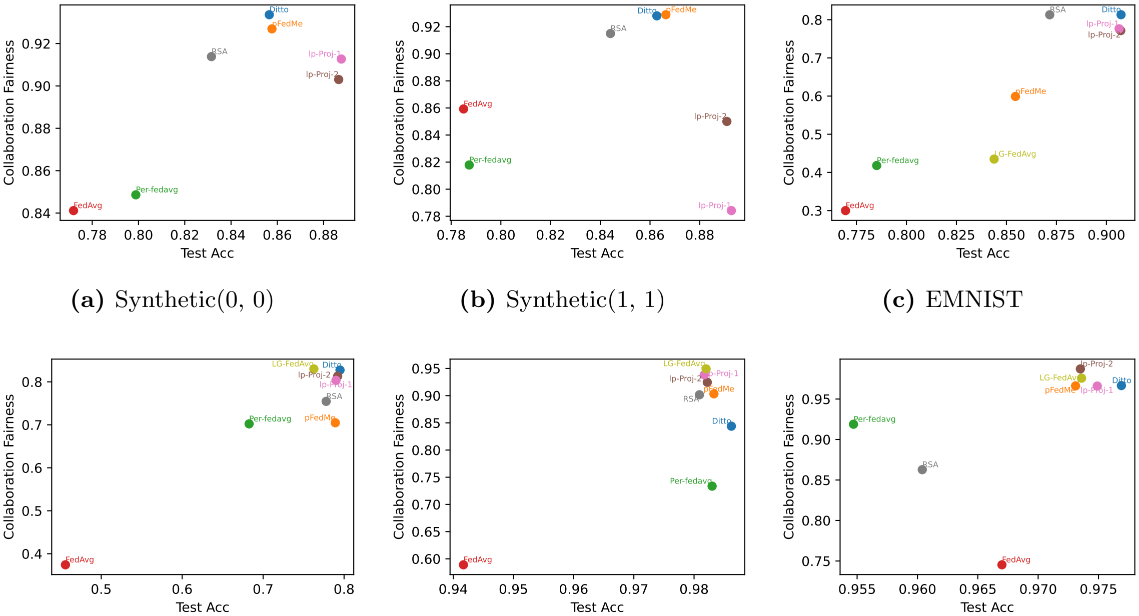
Figure 9 shows the comparison of collaboration fairness of different methods. Note that the point that is closer to the top right corner implies the corresponding method enjoys better performance under the accuracy-fairness trade-off. We can find that our proposed methods enjoy comparable performance compared with other SOTA methods.

C.7.4 Orthogonalization of the Projection Matrix

In the main paper, we assume the projection matrix $\boldsymbol{P}$ is orthogonal throughout the theoretical analysis, while in practice, explicitly orthogonalizing the matrix $\boldsymbol{P}$ may bring computational burden as the full dimension d of the model is usually large. But the dilemma can be circumvented thanks to the approximate orthogonality of high dimensional random vectors, which fact is also taken advantage of in the construction of our projection matrix. Through numerical experiments, we show that by using high dimensional random vectors to construct the projection matrix, model accuracy is hardly affected by explicit orthogonalization or not. Table 16 shows the comparison of model accuracies on the MNIST data set with orthogonalization or not. From the comparison, we can see that the difference in model accuracy between the orthogonal projection matrix and the non-orthogonal projection matrix is so tiny that can be ignored in practice. Therefore, we can safely use the random projection matrix directly to proceed with the algorithm.

C.7.5 Dimension of the Random Subspace

In our proposed algorithm, we need to generate a $d_{\text{sub}} \times d$ random projection matrix $\boldsymbol{P}$, where d_{sub} is the dimension of the projection random subspace. In our theoretical analysis,



(a) Synthetic(0, 0) **(b)** Synthetic(1, 1) **(c)** EMNIST

(d) CIFAR **(e)** MNIST **(f)** FASHIONMNIST

Figure 9: Collaboration Fairness of `lp-proj-1`, `lp-proj-2` with other methods. (The point closer to the top right corner is the better.)

Method	Orthogonalization	Train Loss	Test Acc
lp-proj-1	×	4.027E-03 (3.555E-06)	9.822E-01 (4.281E-04)
	✓	**4.026E-03** (**3.545E-06**)	9.822E-01 (4.281E-04)
lp-proj-2	×	**1.778E-02** (**1.015E-04**)	**9.817E-01** (4.465E-04)
	✓	1.779E-02 (1.047E-04)	9.819E-01 (**4.334E-04**)

Table 16: Comparison of model accuracies on the MNIST data set with explicit orthogonalization or not.

we show that the convergence of our method only has mild dependence on the projection dimension, for both convex and non-convex but smooth cases, which provides huge flexibility to the choice of the dimension of the random subspace.

Here we verify this finding by numerical experiment on the EMNIST data set with a 2-hidden layer neural network. We set the projection dimension as 80 in the main paper. For comparison, we let the dimension of the random projection subspace range from 40 to 200, and track the corresponding training losses and test accuracies. From Table 17, we can see that as d_{sub} increases, there are tiny changes in the training loss and test accuracy and their corresponding variance.

Method	d_{sub}	Train Loss	Test Acc
lp-proj-1	40	0.0451 (0.0039)	**0.9082** (**0.0016**)
	80	0.0473 (0.0043)	0.9064 (0.0017)
	120	**0.0370** (**0.0037**)	0.9058 (0.0018)
	160	0.0407 (0.0040)	0.9076 (**0.0016**)
	200	0.0464 (0.0043)	0.9070 (**0.0016**)
lp-proj-2	40	0.0462 (0.0040)	**0.9074** (0.0017)
	80	0.0447 (0.0039)	0.9072 (0.0017)
	120	0.0459 (0.0038)	0.9071 (**0.0016**)
	160	**0.0454** (**0.0037**)	**0.9074** (**0.0016**)
	200	0.0487 (0.0040)	0.9066 (0.0017)

Table 17: Comparison of model accuracies on the EMNIST data set with different dimensions of the random projection subspace.

In view of this, we claim that the dimension of the random projection subspace d_{sub} can be determined by the trade-off between full model size and communication budget in practice, and this feature would significantly improve communication efficiency.

References

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1707–1718, 2017.

Yossi Arjevani, Ohad Shamir, and Nathan Srebro. A tight convergence analysis for stochastic gradient descent with delayed updates. In *Algorithmic Learning Theory*, pages 111–132, 2018.

Sheikh Shams Azam, Seyyedali Hosseinalipour, Qiang Qiu, and Christopher Brinton. Recycling model updates in federated learning: Are gradient subspaces low-rank? In *International Conference on Learning Representations (ICLR)*, 2021.

Burak Bartan and Mert Pilanci. Distributed sketching methods for privacy preserving regression. *arXiv preprint arXiv:2002.06538*, 2020.

Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Cham, 2011.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *International Conference on Machine Learning (ICML)*, pages 1467–1474, 2012.

Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 245–250, 2001.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 118–128, 2017.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Kone**v**cnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

Shuxiao Chen, Qinqing Zheng, Qi Long, and Weijie J Su. A theorem of the alternative for personalized federated learning. *arXiv preprint arXiv:2103.01901*, 2021.

Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25, 2017.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 21394–21405, 2020.

Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. Fairness-aware agnostic federated learning. In *SIAM International Conference on Data Mining (SDM)*, pages 181–189. SIAM, 2021.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science Conference (ITCS)*, pages 214–226, 2012.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 3557–3568, 2020.

Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to Byzantine-Robust federated learning. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1605–1622, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, pages 1126–1135, 2017.

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

Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtarik. Lower bounds and optimal algorithms for personalized federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 2304–2315, 2020.

Filip Hanzely, Boxin Zhao, and Mladen Kolar. Personalized federated learning: A unified framework and universal optimization techniques. *Transactions on Machine Learning Research*, 2021.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pages 3315–3323, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer Berlin, Heidelberg, 1993.

Tim Hoheisel, Maxime Laborde, and Adam Oberman. A regularization interpretation of the proximal point method for weakly convex functions. *Journal of Dynamics & Games*, 7(1): 79–96, 2020.

Zeou Hu, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu. Federated learning meets multi-objective optimization. *IEEE Transactions on Network Science and Engineering*, 9 (4):2039–2051, 2022.

Wei Huang, Tianrui Li, Dexian Wang, Shengdong Du, and Junbo Zhang. Fairness and accuracy in federated learning. *arXiv preprint arXiv:2012.10069*, 2020.

Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Vladimir Braverman, Ion Stoica, and Raman Arora. Communication-efficient distributed SGD with sketching. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13142–13152, 2019.

Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018.

Yihan Jiang, Jakub Konevcnỳ, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

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

Qifa Ke and Takeo Kanade. Robust l/sub 1/norm factorization in the presence of outliers and missing data by alternative convex programming. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 739–746. IEEE, 2005.

Viraj Kulkarni, Milind Kulkarni, and Aniruddha Pant. Survey of personalization techniques for federated learning. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pages 794–797. IEEE, 2020.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226. Association for Computing Machinery, 2019.

Ang Li, Jingwei Sun, Binghui Wang, Lin Duan, Sicheng Li, Yiran Chen, and Hai Li. Lotteryfl: Empower edge intelligence with personalized and communication-efficient federated learning. In *IEEE/ACM Symposium on Edge Computing (SEC)*, pages 68–79. IEEE, 2021a.

Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pages 1885–1893, 2016.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations (ICLR)*, 2018.

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

Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 1544–1551, 2019.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450, 2020a.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning (ICML)*, 2021b.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on Non-IID data. In *International Conference on Learning Representations (ICLR)*, 2020b.

Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *International Conference on Learning Representations (ICLR)*, 2018.

Lingjuan Lyu, Xinyi Xu, Qian Wang, and Han Yu. Collaborative fairness in federated learning. In *Federated Learning*, pages 189–204. Springer, 2020.

Yishay Mansour, Mehryar Mohri, Jae Ro, and Ananda Theertha Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *ACM International Conference on Multimedia*, pages 1485–1488, 2010.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Yurii Nesterov. *Lectures on Convex Optimization.* Springer Cham, 2018.

Benjamin IP Rubinstein, Blaine Nelson, Ling Huang, Anthony D Joseph, Shing-hon Lau, Satish Rao, Nina Taft, and J Doug Tygar. Antidote: understanding and defending against poisoning of anomaly detectors. In *ACM SIGCOMM Conference on Internet Measurement*, pages 1–14, 2009.

Osama Shahid, Seyedamin Pouriyeh, Reza M Parizi, Quan Z Sheng, Gautam Srivastava, and Liang Zhao. Communication efficiency in federated learning: Achievements and challenges. *arXiv preprint arXiv:2107.10996*, 2021.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Sebastian U Stich. Unified optimal analysis of the (stochastic) gradient method. *arXiv preprint arXiv:1907.04232*, 2019.

Octavian Suciu, Radu Marginean, Yigitcan Kaya, Hal Daume III, and Tudor Dumitras. When does machine learning FAIL? generalized transferability for evasion and poisoning attacks. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1299–1316, 2018.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):9587–9603, 2022.

Laurens Van Der Maaten, Eric Postma, Jaap Van den Herik, et al. Dimensionality reduction: a comparative. *Journal of Machine Learning Research*, 10(66-71):13, 2009.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *Compressed Sensing*, pages 210–268, 2012.

Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. Slowmo: Improving communication-efficient distributed SGD with slow momentum. In *International Conference on Learning Representations (ICLR)*, 2019a.

Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Fran**c**coise Beaufays, and Daniel Ramage. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019b.

Lin Wang, Zhichao Wang, Sai Praneeth Karimireddy, and Xiaoying Tang. Fedeba+: Towards fair and effective federated learning via entropy-based model. *arXiv preprint arXiv:2301.12407*, 2023.

Ruiyuan Wu, Anna Scaglione, Hoi-To Wai, Nurullah Karakoc, Kari Hreinsson, and Wing-Kin Ma. Federated block coordinate descent scheme for learning global and personalized models. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pages 10355–10362, 2021.

Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning? In *International Conference on Machine Learning (ICML)*, pages 1689–1698. PMLR, 2015.

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Generalized byzantine-tolerant sgd. *arXiv preprint arXiv:1802.10116*, 2018.

Xinyi Xu and Lingjuan Lyu. A reputation mechanism is all you need: Collaborative fairness and adversarial robustness in federated learning. In *ICML Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2021.

Kunda Yan, Sen Cui, Abudukelimu Wuerkaixi, Jingfeng Zhang, Bo Han, Gang Niu, Masashi Sugiyama, and Changshui Zhang. Balancing similarity and complementarity for federated learning. In *International Conference on Machine Learning (ICML)*, 2024.

Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning: State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023a.

Rui Ye, Zhenyang Ni, Fangzhao Wu, Siheng Chen, and Yanfeng Wang. Personalized federated learning with inferred collaboration graphs. In *International Conference on Machine Learning (ICML)*, pages 39801–39817. PMLR, 2023b.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning (ICML)*, pages 5650–5659. PMLR, 2018.

Han Yu, Zelei Liu, Yang Liu, Tianjian Chen, Mingshu Cong, Xi Weng, Dusit Niyato, and Qiang Yang. A fairness-aware incentive scheme for federated learning. In *AAAI/ACM Conference on AI, Ethics, and Society*, pages 393–399, 2020a.

Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020b.

Yulin Zhao, Hualin Zhou, and Zhiguo Wan. SuperFL: Privacy-preserving federated learning with efficiency and robustness. *Cryptology ePrint Archive*, 2024.

Zirui Zhou, Lingyang Chu, Changxin Liu, Lanjun Wang, Jian Pei, and Yong Zhang. Towards fair federated learning. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 4100–4101, 2021.

Shengkun Zhu, Jinshan Zeng, Sheng Wang, Yuan Sun, Xiaodong Li, Yuan Yao, and Zhiyong Peng. On ADMM in heterogeneous federated learning: Personalization, robustness, and fairness. *arXiv preprint arXiv:2407.16397*, 2024.