Personalized Federated Learning towards Communication Efficiency, Robustness and Fairness

Anonymous Author(s)
Affiliation
Address
email

Abstract

Personalized Federated Learning 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. We follow the avenue and propose a personalized FL method towards 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 strongly convex 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.

1 Introduction

Federated Learning (FL) emerges as a new distributed computing paradigm that would perform privately distributed optimization in large-scale networks of remote clients [50]. For the sake of privacy protection, data are generated locally and are kept in the 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 [31].

To detour the data heterogeneity issue, one considers to personalize local models [34]. A key feature that any personalization method has is to differ local models from the global model. The simplest way for personalization is training purely with local data on each device. Chen et al. [11] showed that when the degree of data heterogeneity is above some threshold value, 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 [20, 21, 13, 14, 25, 63]. Recently, Li et al. [43] suggested that personalization can be leveraged not only to improve accuracy, but also to allow for competing constraints such as robustness and fairness. Inspired by their 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 as `lp-proj`, whose core is 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. Towards the three goals, the smoothing kernel is designed

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

to regularize the projection of local models in a shared low-dimensional random subspace rather than the original space. In this way, 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 once and will not change during training. It makes local models share a similar part in the random subspace and adapt to their local data using components beyond that.

Theoretically, we give convergence analysis for strongly convex objective with square regularizer, and show that the convergence dependence on the projection dimension is mild. By analyzing test losses and the corresponding variances across the network on federated linear regression, we show that our proposed method is at least as good as two SOTA methods [14, 43] in terms of Byzantine robustness [36] and performance fairness (see Definition 1).

Finally, we conduct a large number of experiments to show the empirical superiority of our method. It not only shows expectant personalization but also promotes fairness by achieving higher test accuracy more uniformly over clients. It significantly improves communication efficiency, because the subspace dimension is often no more than one-hundredth of the original dimension. Furthermore, it is more resistant to training-time adversarial attacks.

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.

2 Related Work

There are many works studying personalization and a survey can be found in [34]. It has been studied via multi-task learning [56, 27], meta-learning [10, 30, 17], knowledge distillation [40, 69] and transfer learning [62, 48]. Hanzely et al. [22] provided convergence analysis for a general personalized framework that requires jointly strongly convex and smooth objectives.

Communication Efficiency To reduce the cost of communication in FL with large-scale networks, existing works could be categorized as gradient compression, model compression and reducing the communication frequency. On gradient compression, sparsification [28, 46], quantization [1] and low-rank approximation [3] are three main directions investigated in the current research. On model compression, Liang et al. [45] suggested learning local representations and a global model only operates on the local representations, Li et al. [37] extended the lottery ticket hypothesis and used network pruning in the FL setting. On communication frequency, Wang et al. [61] used momentum to delay the global aggregation; McMahan et al. [50], Karimireddy et al. [32] performed multiple local updates to lessen the communication rounds. Shahid et al. [55] presented a survey on current progress on communication efficiency in FL. Our proposed method is a model compression approach. Differing from previous works that compress the model each time with different basis, our work focus on a shared-and-fixed low-dimensional subspace which is determined at the beginning of training and will not change later on.

Fairness in FL Zhou et al. [70] suggested that there are three types of fairness in FL: performance fairness, collaboration fairness and model fairness. On performance fairness, an FL system usually promotes uniform accuracy distribution across participants, which is closely related to resource allocation by viewing FL as a joint optimization system over a heterogeneous network. Li et al. [43] gave a formal definition (Definition 1) and some efficient methods have been proposed towards this goal [43, 26].

Definition 1 (Performance Fairness [43])**.** *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.* $\mathrm{var}\left\{F_k(\mathbf{w}_1)\right\}_{k\in[N]} < \mathrm{var}\left\{F_k(\mathbf{w}_2)\right\}_{k\in[N]}$*, where* $F_k(\cdot)$ *denotes the test loss of client* $k \in [N]$ *and* var *denotes the variance*[1].

On collaboration fairness, one expects that each participant receives a reward that can fairly reflect its contribution to the FL system, in which way one can build a sound incentive mechanism. Lyu et al. [47] formalized this idea; Yu et al. [68], Xu and Lyu [66] explored on this aspect. On model fairness, one usually concerns ethical issues and seeks to protect some sensitive attributes [16, 23, 35]. Liang et al. [45] suggested learning a fair representation for each client to achieve fairness, Du et al. [15] proposed reweighing the objective functions under fairness constraint. In our work, we mainly

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

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 D.7.2.

Robust FL Typical adversarial attacks include data poisoning and model update poisoning (Byzantine attacks). The former injects abnormal sample points into the training dataset [6, 29, 38, 54, 58, 64, 18], while the latter manipulates communication messages by sending arbitrary updates to the server, both attacks hindering the training processes. In this paper, we mainly aim to defend model update poisoning attacks and achieve Byzantine robustness [36]. An extension to a data poisoning attack is also considered in numerical experiments. Prior works favor Byzantine-robust SGD variants where the server alleviates the attack of Byzantine clients via robust aggregation rules [12, 67, 65, 8]. Beyond that, Li et al. [41] developed a robust distributed method by incorporating the L^p-norm regularizer to robustify the objective function. Li et al. [43] used robust aggregation rules to achieve robustness and fairness. Our work leverages the ideas from [41] and [43], 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 [14, 43] (see Figure 1). Extensive experiments manifest our method of achieving state-of-the-art performance under various types and intensities of adversarial attacks.

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 massive devices. To enhance accuracy performance, we hope not only to leverage the global model, but also stylize it to fit the local data for each client. To this end, we use *infimal convolution*, which is originally proposed to smooth some extended real-valued convex function f with a sufficiently smooth kernel function g [51]. We apply this technique in FL to bridge local models and 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:

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

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

$$F_k(\mathbf{w}) = \left\{f_k \otimes \lambda g\right\}(\mathbf{w}) := \min_{\mathbf{x}_k\in\mathbb{R}^d} f_k(\mathbf{x}_k) + \lambda g(\mathbf{w}-\mathbf{x}_k) \text{ with } f_k(\mathbf{x}_k) = \mathbb{E}_{\xi_k}\left[\tilde{f}_k(\mathbf{x}_k;\xi_k)\right]. \tag{2}$$

Here, $\otimes$ denotes 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 example, Dinh et al. [14], Li et al. [43] used Moreau Envelopes as the regularizer, which is equivalent to $g(\cdot) = \frac{1}{2}\|\cdot\|_2^2$. Li et al. [41] proposed the L^p-norm regularization $g(\cdot) = \|\cdot\|_p$ instead. Motivated by the fact that high-dimensional data usually has low-dimensional representation that retains meaningful properties [59], and random projection would preserve similarity of data vectors [7], we propose to regularize the projection of local models in a shared low-dimensional space, which is equivalent to the following smoothing kernel

$$g(\cdot) = \frac{1}{p}\left\|\boldsymbol{P}(\cdot)\right\|_p^p, \tag{3}$$

where $p \geq 1$ and $\boldsymbol{P}$ is 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 [39]. 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]

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

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

1: **Input**: Communication rounds T, local update rounds R, client sampling size S, regularization coefficient λ, lower-level problem accuracy ν, step size η, initial global model $\tilde{\mathbf{w}}_0$, projection matrix $\boldsymbol{P}$, speedup control parameter β.
2: **for** $t = 0$ to $T-1$ **do**
3: Server sends $\tilde{\mathbf{w}}_t$ to all clients.
4: **for** all $k = 1$ to N clients **do**
5: $\tilde{\mathbf{w}}_{k,0}^t = \tilde{\mathbf{w}}_t$.
6: **for** $r = 0$ to $R-1$ **do**
7: Independently sample a fresh mini-batch $\tilde{\mathcal{D}}_k$ and minimize the loss function (4) up to accuracy level ν to get $\mathbf{x}_{k,r}^t$.
8: Update the local model $\tilde{\mathbf{w}}_{k,r+1}^t$ by (5).
9: **end for**
10: **end for**
11: Server uniformly samples a subset of clients $\mathcal{S}_t$ of size S. Each client sends $\tilde{\mathbf{w}}_{k,R}^t$ to the server.
12: Server updates the global model via $\tilde{\mathbf{w}}_{t+1} = (1-\beta)\tilde{\mathbf{w}}_t + \beta \sum_{k\in\mathcal{S}_t} \frac{\tilde{\mathbf{w}}_{k,R}^t}{S}$.
13: **end for**

3.1 The Algorithm

In this section, we introduce the algorithm `lp-proj` (see Algorithm 1) for the bi-level optimization problem (1) with smoothing kernel g given by Eqn. (3).

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 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

$$\tilde{h}_k(\mathbf{x}_k; \tilde{\mathcal{D}}_k, \tilde{\mathbf{w}}_{k,r}^t) = \frac{1}{|\tilde{\mathcal{D}}_k|} \sum_{\xi_{k,i}\in\tilde{\mathcal{D}}_k} \tilde{f}_k(\mathbf{x}_k; \xi_{k,i}) + \lambda \frac{1}{p} \left\| \tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_k \right\|_p^p. \tag{4}$$

  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 $\left\| \nabla \tilde{h}_k(\mathbf{x}_{k,r}^t; \tilde{\mathcal{D}}_k, \tilde{\mathbf{w}}_{k,r}^t) \right\|_2^2 \le \nu$ 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:

$$\tilde{\mathbf{w}}_{k,r+1}^t = \tilde{\mathbf{w}}_{k,r}^t - \frac{\eta\lambda}{p} \partial_{\tilde{\mathbf{w}}_{k,r}^t} \left\| \tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_{k,r}^t \right\|_p^p. \tag{5}$$

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.

Communication Efficiency In Algorithm 1, we restrict the global model $\tilde{\mathbf{w}}_t$ to lie in a fixed low-dimensional subspace, only $\tilde{\mathbf{w}}_{k,R}^t$ of dimension d_{sub}, instead of the full model $\mathbf{x}_{k,r}^t$ of dimension d, is communicated to the server during each round, which leads to much fewer bits for communication compared to vanilla FL. Besides, we remark on the difference between our method and other projection/sketching-based methods. On one hand, distributed sketching [4], 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 [28] compresses the transmitted messages with

different basis every time, while our random subspace is specified at the beginning and would not change after that.

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, introducing L^p-norm regularized term to the objective function is equivalent to launching an uncertainty set to the model parameter by rewriting the objective as a constrained optimization problem (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

4.1 Convergence

Assumption 1 (Convexity and smoothness)**.** *$\tilde{f}_k$ is μ-strongly convex and L-smooth, that is, for any $\mathbf{x}_k, \mathbf{x}'_k \in \mathbb{R}^d$ and ξ_k, we have $\tilde{f}_k(\mathbf{x}'_k;\xi_k) \geq \tilde{f}_k(\mathbf{x}_k;\xi_k) + \left\langle \nabla \tilde{f}_k(\mathbf{x}_k;\xi_k), \mathbf{x}'_k - \mathbf{x}_k \right\rangle + \frac{\mu}{2}\left\|\mathbf{x}'_k - \mathbf{x}_k\right\|_2^2$ and $\left\|\nabla \tilde{f}_k(\mathbf{x}'_k;\xi_k) - \nabla \tilde{f}_k(\mathbf{x}_k;\xi_k)\right\|_2 \leq L\left\|\mathbf{x}'_k - \mathbf{x}_k\right\|_2$.*

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. Assumption 1 also guarantees strong convexity and smoothness of f_k. 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 $F_k(\tilde{\mathbf{w}}) = \min_{\mathbf{x}_k \in \mathbb{R}^d}\left\{f_k(\mathbf{x}_k) + \frac{\lambda}{2}\left\|\tilde{\mathbf{w}} - \boldsymbol{P}\mathbf{x}_k\right\|_2^2\right\}$. Then we have the following result.

Lemma 1. *Suppose Assumptions 1 and 2 hold. With probability at least $1 - 2\exp(-cd_{\text{sub}})$, we have $\frac{1}{\lambda^2}\mathbb{E}\left[\left\|\nabla F_k(\tilde{\mathbf{w}}_{k,r}^t) - \lambda(\tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_{k,r}^t)\right\|_2^2\right] \leq \delta^2 := \frac{2\left(1+C\sqrt{d_{\text{sub}}/d}\right)^2}{\mu^2}\left(\frac{\gamma_f^2}{|\mathcal{D}_k|} + \nu\right)$, where $C, c > 0$ are constants.*

Lemma 1 quantifies the error between the exact gradient $\nabla F_k(\tilde{\mathbf{w}}_{k,r}^t)$ and the approximate gradient $\lambda(\tilde{\mathbf{w}}_{k,r}^t - \boldsymbol{P}\mathbf{x}_{k,r}^t)$ due to mini-batch sampling and optimization error of the inner loop. From the expression of δ^2, we can see that this error has a mild dependence on d_{sub}. (Numerical verification of the mild dependence is shown in Appendix D.7.4.)

Based on Lemma 1, we can establish the convergence of the global model and personalized parameters of Algorithm 1. Before presenting the main result, we first introduce some necessary notations. By convexity and smoothness of f_k, we can show that F_k is μ_F-strongly convex and L_F-smooth with $\mu_F = \frac{\lambda\mu}{(1+C\sqrt{d_{\text{sub}}/d})^2\lambda+\mu}$ and $L_F = \lambda$.[4] Then the condition number of F_k is $\kappa_F = L_F/\mu_F = (1 + C\sqrt{d_{\text{sub}}/d})^2\lambda/\mu + 1$, and there exists the unique solution $\tilde{\mathbf{w}}^* = \operatorname{argmin}_{\tilde{\mathbf{w}} \in \mathbb{R}^{d_{\text{sub}}}} F(\tilde{\mathbf{w}})$. Define $\sigma_F^2 = \frac{1}{N}\sum_{k=1}^N \left\|\nabla F_k(\tilde{\mathbf{w}}^*)\right\|_2^2$, which measures the bounded diversity of F_k. Then the convergence of Algorithm 1 is guaranteed by Theorem 1.

Theorem 1. *Suppose that Assumptions 1 and 2 hold. Let $\hat{\eta}_1 = \frac{1}{18L_F(1+10\kappa_F/\beta)}$ with $\beta \geq 1$. If $T \geq \frac{2}{\hat{\eta}_1\mu_F}$, then with probability at least $1 - 2\exp(-cd_{\text{sub}})$, there exists $\eta \leq \frac{\hat{\eta}_1}{\beta R}$ such that*

$$\mathbb{E}\left[F(\bar{\mathbf{w}}_T) - F(\tilde{\mathbf{w}}^*)\right] \leq \underbrace{\mu_F\Delta_0 e^{-\hat{\eta}_1\mu_F T/2}}_{\text{due to initialization}} + \underbrace{\tilde{\mathcal{O}}\left(\frac{(N/S-1)\sigma_F^2}{\mu_F TN}\right)}_{\text{due to client sampling}} + \underbrace{\tilde{\mathcal{O}}\left(\frac{(\sigma_F^2+\lambda^2\delta^2)\kappa_F L_F}{\mu_F^2\beta^2T^2}\right) + \mathcal{O}\left(\frac{\lambda^2\delta^2}{\mu_F}\right)}_{\text{client drift with multiple local updates and approximation error}}, \tag{6}$$

where $\Delta_0 = \left\|\tilde{\mathbf{w}}_0 - \tilde{\mathbf{w}}^*\right\|_2^2$, $c > 0$ is a constant, $\bar{\mathbf{w}}_T = \sum_{t=0}^{T-1}\alpha_t\tilde{\mathbf{w}}_t/A_T$ 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

[4] See Proposition 10 in Appendix A.2 for the details.

$\boldsymbol{P}$ *and* $\tilde{\mathcal{O}}$ *hides constants and polylogarithmic factors. Moreover, suppose that* $\mathbf{x}_k^T$ *is a solution satisfies* $\left\|\nabla \tilde{h}_k(\mathbf{x}_k^T; \tilde{\mathcal{D}}_k, \tilde{\mathbf{w}}_T)\right\|_2^2 \leq \nu$ *and* $\mathcal{O}_1$ *denotes the right-hand side of (6). Then with probability at least* $1 - 2\exp(-cd_{\text{sub}})$*, we have*

$$\frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\left[\left\|\boldsymbol{P}\mathbf{x}_k^T - \tilde{\mathbf{w}}^*\right\|_2^2\right] \leq \frac{\mathcal{O}_1}{\mu_F} + \mathcal{O}\left(\frac{\sigma_F^2}{\lambda^2} + \delta^2\right). \tag{7}$$

From (6), 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. Eq. (7) shows the average of personalized parameters (after a linear transformation) converges to a ball with center $\tilde{\mathbf{w}}^*$ and radius $\tilde{\mathcal{O}}\left(\frac{\lambda^2\delta^2}{\mu_F^2} + \frac{\sigma_F^2}{\lambda^2} + \delta^2\right)$. Here λ can be chosen to trade off different terms.

Our Theorem 1 shares the same error bounds as Theorem 1 in [14], except that their approximation error δ^2 is slightly smaller than ours up to constant factors, since in our case the approximation error is enlarged by projection. The constant term $\mathcal{O}\left(\frac{\lambda^2\delta^2}{\mu_F}\right)$ 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 [14] up to constants factors, even if we force optimization in a random subspace, which facilitates communication efficiency.

Proposition 1. *Under the conditions of Theorem 1, if* $|\tilde{\mathcal{D}}_k|$ *scales linearly with the index* t *and* ν *and* η *decay at the rate* $1/t$*, then with high probability, we have* $\mathbb{E}\left\|\tilde{\mathbf{w}}_T - \tilde{\mathbf{w}}^\star\right\|_2^2 = \mathcal{O}\left(1/T\right)$*. Based on this, the convergence of personalized parameters is also of the order* $\mathcal{O}(1/T)$*.*

Proposition 1 shows that we can carefully tune the parameters to cancel out the enlargement of approximation error and achieve any given accuracy ε. Due to space limitations, the formal results are deferred to Appendix B. 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.

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` [43] and `pFedMe` [14]. 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 D.7.3).

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 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 $\tilde{\mathbf{w}}_k^{(ma)} \sim \mathcal{N}(\mathbf{0}, \tau^2\boldsymbol{I}_{d_{\text{sub}}})$.

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 C.3. Suppose that there are N_a malicious clients, and the heterogeneity is uniform in all dimensions in the sense of Eqn.(25)

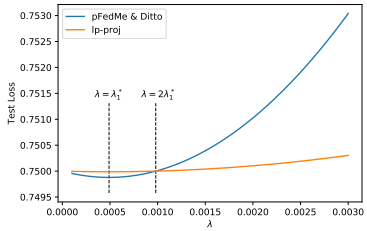
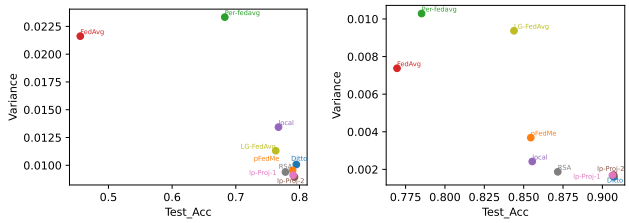


Figure 1: Test losses of `pFedMe`, `Ditto` and `lp-proj` under different values of λ. In our settings, the losses of `pFedMe` and `Ditto` coincide.



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

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

in Appendix C.3, where we define Σ_1 to measure 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^* is the variance of noises over the number of samples. The denominator is the sum of data heterogeneity and variance of attacks. 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 2 (Formal statement in Theorem 2, Appendix C.3)**.** *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. The explicit forms of losses are in Appendix C.3. 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 2 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 $n = 200$, $N = 100$, $N_a = 20$, $d = 100$, $d_{\text{sub}} = 10$, $b = 1$, $\sigma = 1$, $\Sigma_1 = 0.1$ and $\tau = 100$. Then $\lambda_1^* \approx 4.9e-4$, 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 $\lambda > 0$.

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. and distributed as $\mathcal{N}(\mu_w, \mathbf{\Sigma}_w)$.

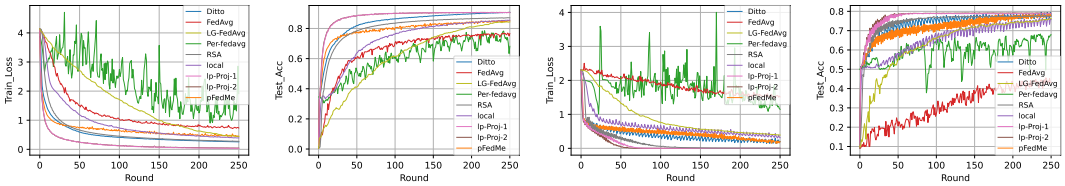
Proposition 3 (Formal statement in Theorem 3, Appendix C.4)**.** *Denote the variance of test losses on different clients of `pFedMe`, `Ditto` and `lp-proj` by $V^{Me}(\lambda)$, $V^{Di}(\lambda)$ and $V^{l2}(\lambda)$ respectively. We have for $\forall \lambda > 0$, $\mathbb{E}V^{l2}(\lambda) \leq \mathbb{E}V^{Me}(\lambda) = \mathbb{E}V^{Di}(\lambda)$. More specifically, $\mathbb{E}V^{Me}(\lambda) = O(d^2)$ and $\mathbb{E}V^{l2}(\lambda) = O(d_{\text{sub}}^2)$, where the expectation is taken w.r.t. the randomness of $\mathbf{w}_k$.*

Proposition 3 shows `lp-proj` always brings more uniform test losses, no matter what value λ is. In particular, $\mathbb{E}V^{\text{l2}}(\lambda) = O(d_{\text{sub}}^2)$ 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, see Appendix C.4.

5 Numerical Experiments

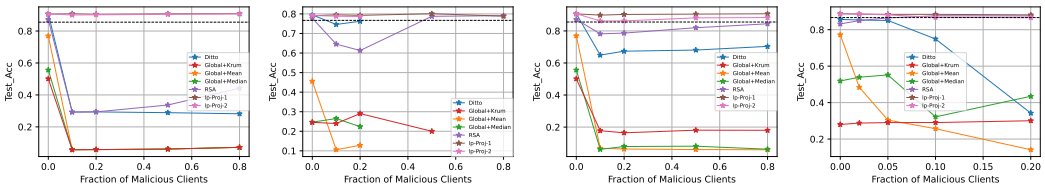
Experimental Setup We test `lp-proj` as well as other comparable algorithms on six datasets from common ML and FL benchmarks [49, 9]. We consider both convex and non-convex models. For the latter, we consider neural networks including both MLP and CNN. Details about datasets, models and basic information about the FL system are provided in Table 2 in Appendix D. To better model the statistical heterogeneity, we distribute the dataset among clients in a non-iid fashion such that each client only contains partial classes of the data in multi-classification problems. Furthermore, we also consider an extension of non-i.i.d. on data volume skewness in Appendix D.7.1.

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.



(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.



(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.

All reported results are evaluated on the test dataset. More details about hyperparameter tuning are provided in Appendix D.2. Furthermore, to incorporate partial participation [50, 44], we randomly select 10% of the clients for aggregation at each communication round. The projection dimension of the random subspace for each dataset is chosen based on the full model size and communication budget (see Appendix D.2). For reproduction of our numerical results, we provide the implemented source code in the supplementary material.

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, together with a global method and a pure local method. Specifically, we consider the case when $p = 1$ (`lp-proj-1`) and $p = 2$ (`lp-proj-2`). For SOTA methods, we consider `Ditto` [43], `pFedMe` [14], `Per-fedavg` [17], `LG-FedAvg` [45] and `RSA` [41]. A brief description of these approaches is provided in Appendix D.1.

Due to space limitations, we only show the comparisons on training loss and test accuracy on two standard datasets (EMNIST and CIFAR) in Figure 3. Results on the other datasets are left in Table 3 and Figure 5 in Appendix D.3. From these figures, we see that `lp-proj-1` and `lp-proj-2` have lower training loss and higher test accuracy than other methods. Furthermore, the training process is more stable as the loss and accuracy curves have less fluctuation.

Communication Efficiency We compare `lp-proj` with the global baseline `FedAvg` [50] and five standard approaches using gradient and model compression, namely `Sketch` [28], `LBGM` [3], `QSGD` [1], `DGC` [46] and `LG-FedAvg` [45]. 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. Due to space limitations, we only show the results on two datasets in Table 1 (`Synthetic(0, 0)` and EMNIST) with the full results left in Table 4 in Appendix D.4.

It is clear that our method is more communication efficient since we only communicate low-dimensional messages. 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 datasets 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 datasets 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 1 with further supported numerical results in Table 16 in Appendix D.7.4. 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.

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

Table 1: Communication performance on `Synthetic(0, 0)` and `EMNIST` datasets. 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.

	`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	$\star$	0.7	445851400
Sketch	328020	0.456	0.6	$\star$	4236900	$\star$	0.7	$\star$
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	0.538	0.6	365002	4236900	$\star$	0.7	769902776
QSGD	328020	0.115	0.6	923350	4236900	$\star$	0.7	673302175
DGC	328020	$\star$	0.6	391800	4236900	$\star$	0.7	$\star$
LG-FedAvg	\	\	\	\	4236900	0.071	0.7	230786010

- **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 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 global training augmented with different robust aggregation techniques, such as median and Krum [8], `Ditto` and `RSA`.

Due to space limitations, we show only a representative figure for each attack in Figure 4. For full results, please refer to Appendix D.5. 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 roots in the combination of projection and L^1-norm subspace regularization that attribute 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 [33].

Fairness To illustrate the accuracy and performance fairness trade-off, we plot the variances of accuracies across the network against the corresponding test accuracies for different methods in Figure 2. Due to space limitations, we show the results on two datasets here with the full results left in Figure 6 in Appendix D.6. Furthermore, we also show an extension to collaboration fairness (with its definition) in Appendix D.7.2. All the results show that `lp-proj-1` and `lp-proj-2` provide accurate and fair solutions that are comparable to other SOTA methods.

6 Concluding Remarks

In this paper, we have proposed a simple but powerful personalized FL approach based on infimal convolution and subspace projection that we call `lp-proj`. We have presented the convergence results for strongly convex objectives with square regularizers. Inherent benefits of robustness and fairness of our method are also illustrated on a class of linear problems. Empirical results show that our approach could significantly save communication costs, improve robustness under various kinds of adversarial attacks, and promote performance fairness. In future work, we would be interested in establishing convergence results for general L^p regularizers, exploring numerical applications on other ML tasks, and considering additional constraints, e.g., differential privacy.

References

[1] 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 (NIPS)*, 2017.

[2] 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.

[3] 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*, 2021.

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

[5] Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. 2011.

[6] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*, 2012.

[7] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250, 2001.

[8] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 118–128, 2017.

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

[10] 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.

[11] 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.

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

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

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

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

[16] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[17] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *arXiv preprint arXiv:2002.07948*, 33:3557–3568, 2020.

[18] 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.

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

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

[21] 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*, volume 33, pages 2304–2315, 2020.

[22] Filip Hanzely, Boxin Zhao, and Mladen Kolar. Personalized federated learning: A unified framework and universal optimization techniques. *arXiv preprint arXiv:2102.09743*, 2021.

[23] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

[24] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex analysis and minimization algorithms*. 1993.

[25] Zeou Hu, Kiarash Shaloudegi, Guojun Zhang, and Yaoliang Yu. Fedmgda+: Federated learning meets multi-objective optimization. *arXiv preprint arXiv:2006.11489*, 2020.

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

[27] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[28] Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Vladimir Braverman, Ion Stoica, and Raman Arora. Communication-efficient distributed sgd with sketching. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 13142–13152, 2019.

[29] 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 *2018 IEEE Symposium on Security and Privacy (SP)*, pages 19–35. IEEE, 2018.

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

[31] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[32] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378*, 2019.

[33] Qifa Ke and Takeo Kanade. Robust l/sub 1/norm factorization in the presence of outliers and missing data by alternative convex programming. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 739–746. IEEE, 2005.

[34] 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.

[35] Matt J Kusner, Joshua R Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *arXiv preprint arXiv:1703.06856*, 2017.

[36] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226. 2019.

[37] Ang Li, Jingwei Sun, Binghui Wang, Lin Duan, Sicheng Li, Yiran Chen, and Hai Li. Lotteryfl: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets. *arXiv preprint arXiv:2008.03371*, 2020.

[38] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. Data poisoning attacks on factorization-based collaborative filtering. *Advances in neural information processing systems*, 29:1885–1893, 2016.

[39] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=ryup8-WCW`.

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

[41] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1544–1551, 2019.

[42] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[43] Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pages 6357–6368. PMLR, 2021.

[44] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on Non-IID data. *arXiv:1907.02189*, 2019.

[45] 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.

[46] 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*, 2018.

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

[48] 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.

[49] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1485–1488, 2010.

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

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

[52] Arkadi Nemirovski, Anatoli B. Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.*, 19:1574–1609, 2009.

[53] Yurii Nesterov. *Lectures on Convex Optimization*. 2018.

[54] 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 *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, pages 1–14, 2009.

[55] 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.

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

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

[58] 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.

[59] Laurens Van Der Maaten, Eric Postma, Jaap Van den Herik, et al. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13, 2009.

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

[61] Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. Slowmo: Improving communication-efficient distributed sgd with slow momentum. In *International Conference on Learning Representations*, 2019.

[62] Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Franccoise Beaufays, and Daniel Ramage. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019.

[63] 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. *arXiv preprint arXiv:2012.13900*, 2020.

[64] 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*, pages 1689–1698. PMLR, 2015.

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

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

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

[68] 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 *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 393–399, 2020.

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

[70] Zirui Zhou, Lingyang Chu, Changxin Liu, Lanjun Wang, Jian Pei, and Yong Zhang. Towards fair federated learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 4100–4101, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The Introduction part (Section 1) summarizes our contribution and the Related Work part (Section 2) discusses the relationship between our work and previous ones.
   (b) Did you describe the limitations of your work? [Yes] See the conclusion part. We will consider convergence analysis for more general L^p regularizers and additional constraints in our future work.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Societal impacts are not applicable to our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 4.1 for the full assumptions used in convergence analysis and Section 4.2 for the setting considered in robustness and fairness analysis.
   (b) Did you include complete proofs of all theoretical results? [Yes] They are deferred in Appendix A, B and C.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The source code is provided in the supplementary material and the experimental details are described in Appendix D.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The data split strategy is described in Section 5 and the hyperparameter settings for all the competing methods are described in Appendix D.2.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] The complete results are reported in Appendix D.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We use GPU for computing and the type is described in Appendix D.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]