Ditto: Fair and Robust Federated Learning Through Personalization

Tian Li [1] **Shengyuan Hu** [1] **Ahmad Beirami** [2] **Virginia Smith** [1]

Abstract

Fairness and robustness are two important concerns for federated learning systems. In this work, we identify that *robustness* to data and model poisoning attacks and *fairness*, measured as the uniformity of performance across devices, are competing constraints in statistically heterogeneous networks. To address these constraints, we propose employing a simple, general framework for personalized federated learning, Ditto, that can inherently provide fairness and robustness benefits, and develop a scalable solver for it. Theoretically, we analyze the ability of Ditto to achieve fairness and robustness simultaneously on a class of linear problems. Empirically, across a suite of federated datasets, we show that Ditto not only achieves competitive performance relative to recent personalization methods, but also enables more accurate, robust, and fair models relative to state-of-the-art fair or robust baselines.

1. Introduction

Federated learning (FL) aims to collaboratively learn from data that has been generated by, and resides on, a number of remote devices or servers (McMahan et al., 2017). FL stands to produce highly accurate statistical models by aggregating knowledge from disparate data sources. However, to deploy FL in practice, it is necessary for the resulting systems to be not only accurate, but to also satisfy a number of pragmatic constraints regarding issues such as fairness, robustness, and privacy. Simultaneously satisfying these varied constraints can be exceptionally difficult (Kairouz et al., 2019).

We focus in this work specifically on issues of accuracy, fairness (i.e., limiting performance disparities across the network (Mohri et al., 2019)), and robustness (against training-time data and model poisoning attacks). Many prior efforts have separately considered fairness or robustness in federated learning. For instance, fairness strategies include using minimax optimization to focus on the worst-performing devices (Mohri et al., 2019; Hu et al., 2020) or reweighting the devices to allow for a flexible fairness/accuracy tradeoff (Li et al., 2020e; 2021). Robust methods commonly use techniques such as gradient clipping (Sun et al., 2019) or robust aggregation (Blanchard et al., 2017; Yin et al., 2018).

While these approaches may be effective at either promoting fairness or defending against training-time attacks in isolation, we show that the constraints of fairness and robustness can directly compete with one another when training a single global model, and that simultaneously optimizing for accuracy, fairness, and robustness requires careful consideration. For example, as we empirically demonstrate (Section 4), current fairness approaches can render FL systems highly susceptible to training time attacks from malicious devices. On the other hand, robust methods may filter out rare but informative updates, causing unfairness (Wang et al., 2020).

In this work, we investigate a simple, scalable technique to simultaneously improve accuracy, fairness, and robustness in federated learning. While addressing the competing constraints of FL may seem like an insurmountable problem, we identify that statistical heterogeneity (i.e., non-identically distributed data) is a root cause for tension between these constraints, and is key in paving a path forward. In particular, we suggest that methods for personalized FL—which model and adapt to the heterogeneity in federated settings by learning distinct models for each device—may provide *inherent* benefits in terms of fairness and robustness.

To explore this idea, we propose Ditto, a scalable federated multi-task learning framework. Ditto can be seen as a lightweight personalization add-on for standard global FL. It is applicable to both convex and non-convex objectives, and inherits similar privacy and efficiency properties as traditional FL. We evaluate Ditto on a suite of federated benchmarks and show that, surprisingly, this simple form of personalization can in fact deliver better accuracy, robustness, and fairness benefits than state-of-the-art, problem-specific objectives that consider these constraints separately. We summarize our contributions below:

- We propose Ditto, a multi-task learning objective for federated learning that provides personalization while retaining similar efficiency and privacy benefits as traditional FL. We provide convergence guarantees for our

[1]Carnegie Mellon University [2]Facebook AI. Correspondence to: Tian Li <tianli@cmu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

proposed Ditto solver, which incorporate common practices in cross-device federated learning such as limited device participation and local updating. Despite its simplicity, we show that Ditto can deliver similar or superior accuracy relative to other common methods for personalized federated learning.

- Next, we demonstrate that the benefits of Ditto go beyond accuracy—showing that the personalized objective can inherently offer *robustness* superior to that of common robust FL methods across a diverse set of data and model poisoning attacks. On average across all datasets and attacks, Ditto improves test accuracy by $\sim$6% (absolute) over the strongest robust baseline.
- Similarly, we show that Ditto can naturally increase *fairness*—reducing variance of the test accuracy across devices by $\sim$10% while maintaining similar or superior accuracy relative to state-of-the-art methods for fair FL.
- Finally, we highlight that Ditto is particularly useful for practical applications where we simultaneously care about multiple constraints (accuracy, fairness, and robustness). We motivate this through analysis on a toy example in Section 3, as well as experiments across a suite of federated datasets in Section 4.

2. Background & Related Work

Robustness and fairness are two broad areas of research that extend well beyond the application of federated learning. In this section we provide precise definitions of the notions of robustness/fairness considered in this work, and give an overview of prior work in robustness, fairness, and personalization in the context of federated learning.

Robustness in Federated Learning. Training-time attacks (including data poisoning and model poisoning) have been extensively studied in prior work (Biggio et al., 2012; Gu et al., 2017; Chen et al., 2017; Shafahi et al., 2018; Liu et al., 2018; Huang et al., 2020; Xie et al., 2020; Wang et al., 2020; Dumford & Scheirer, 2018; Huang et al., 2020). In federated settings, a number of strong attack methods have been explored, including scaling malicious model updates (Bagdasaryan et al., 2020), collaborative attacking (Sun et al., 2020), defense-aware attacks (Bhagoji et al., 2019; Fang et al., 2020), and adding edge-case adversarial training samples (Wang et al., 2020). Our work aims to investigate common attacks related to Byzantine robustness (Lamport et al., 2019), as formally described below.

Definition 1 (Robustness)**.** *We are conceptually interested in Byzantine robustness (Lamport et al., 2019), where the malicious devices can send arbitrary updates to the server to compromise training. To measure robustness, we assess the mean test performance on benign devices, i.e., we consider model w_1 to be more robust than w_2 to a specific attack if the mean test performance across the benign devices is higher for model w_1 than w_2 after training with the attack. We examine three widely-used attacks in our threat model:*

- *(A1) Label poisoning*: Corrupted devices do not have access to the training APIs and training samples are poisoned with flipped (if binary) or uniformly random noisy labels (Bhagoji et al., 2019; Biggio et al., 2011).
- *(A2) Random updates*: Malicious devices send random zero-mean Gaussian parameters (Xu & Lyu, 2020).
- *(A3) Model replacement*: Malicious devices scale their adversarial updates to make them dominate the aggregate updates (Bagdasaryan et al., 2020).

While non-exhaustive, these attacks have been commonly studied in distributed and federated settings, and explore corruption at various points (the underlying data, labels, or model). In terms of defenses, robust aggregation is a common strategy to mitigate the effect of malicious updates (Blanchard et al., 2017; Pillutla et al., 2019; Sun et al., 2019; Li et al., 2019; He et al., 2020). Other defenses include gradient clipping (Sun et al., 2019) or normalization (Hu et al., 2020). While these strategies can improve robustness, they may also produce *unfair* models by filtering out informative updates, especially in heterogeneous settings (Wang et al., 2020). In our experiments (Section 4), we compare Ditto with several strong defenses (median, gradient clipping (Sun et al., 2019), Krum, Multi-Krum (Blanchard et al., 2017), gradient-norm based anomaly detector (Bagdasaryan et al., 2020), and a new defense proposed herein) and show that Ditto can improve both robustness and fairness compared with these methods.

Fairness in Federated Learning. Due to the heterogeneity of the data in federated networks, it is possible that the performance of a model will vary significantly across the devices. This concern, also known as *representation disparity* (Hashimoto et al., 2018), is a major challenge in FL, as it can potentially result in uneven outcomes for the devices. Following Li et al. (2020e), we provide a more formal definition of this fairness in the context of FL below:

Definition 2 (Fairness)**.** *We say that a model w_1 is more fair than w_2 if the test performance distribution of w_1 across the network is* more uniform *than that of w_2, i.e.,* $\text{std}\{F_k(w_1)\}_{k\in[K]} < \text{std}\{F_k(w_2)\}_{k\in[K]}$ *where $F_k(\cdot)$ denotes the test loss on device $k\in[K]$, and* $\text{std}\{\cdot\}$ *denotes the standard deviation. In the presence of adversaries, we measure fairness only on benign devices.*

We note that there exists a tension between variance and utility in the definition above; in general, a common goal is to lower the variance while maintaining a reasonable average performance (e.g., average test accuracy). To address

representation disparity, it is common to use minimax optimization (Mohri et al., 2019; Deng et al., 2020) or flexible sample reweighting approaches (Li et al., 2020e; 2021) to encourage a more uniform quality of service. In all cases, by up-weighting the importance of rare devices or data, fair methods may not be robust in that they can easily overfit to corrupted devices (see Section 4.3). The tension between fairness and robustness has been studied in previous works, though for different notions of fairness (equalized odds) or robustness (backdoor attacks) (Wang et al., 2020), or in centralized settings (Chang et al., 2020). Recently, Hu et al. (2020) proposed FedMGDA+, a method targeting fair and robust FL; however, this work combines classical fairness (minimax optimization) and robustness (gradient normalization) techniques, in contrast to the multi-task framework proposed herein, which we show can *inherently* provide benefits with respect to both constraints simultaneously.

Personalized Federated Learning. Given the variability of data in federated networks, personalization is a natural approach used to improve accuracy. Numerous works have proposed techniques for personalized federated learning. Smith et al. (2017) first explore personalized FL via a primal-dual MTL framework, which applies to convex settings. Personalized FL has also been explored through clustering (e.g., Ghosh et al., 2020; Sattler et al., 2020; Muhammad et al., 2020), finetuning/transfer learning (Zhao et al., 2018; Yu et al., 2020), meta-learning (Jiang et al., 2019; Chen et al., 2018; Khodak et al., 2019; Fallah et al., 2020; Li et al., 2020a; Singhal et al., 2021), and other forms of MTL, such as hard model parameter sharing (Agarwal et al., 2020; Liang et al., 2020) or the weighted combination method in Zhang et al. (2021). Our work differs from these approaches by simultaneously learning local and global models via a global-regularized MTL framework, which applies to non-convex ML objectives.

Similar in spirit to our approach are works that interpolate between global and local models (Mansour et al., 2020; Deng et al., 2021). However, as discussed in Deng et al. (2021), these approaches can effectively reduce to local minimizers without additional constraints. The most closely related works are those that regularize personalized models towards their average (Hanzely & Richtárik, 2020; Hanzely et al., 2020; Dinh et al., 2020), which can be seen as a form of classical mean-regularized MTL (Evgeniou & Pontil, 2004). Our objective is similarly inspired by mean-regularized MTL, although we regularize towards a global model rather than the average personalized model. As we discuss in Section 3, one advantage of this is that it allows for methods designed for the global federated learning problem (e.g., optimization methods, privacy/security mechanisms) to be easily re-used in our framework, with the benefit of additional personalization. We compare against a range of personalized methods empirically in Section 4.4, showing that Ditto achieves similar or superior performance across a number of common FL benchmarks.

Finally, a key contribution of our work is jointly exploring the robustness and fairness benefits of personalized FL. The benefits of personalization for fairness alone have been demonstrated empirically in prior work (Wang et al., 2019; Hao et al., 2020). Connections between personalization and robustness have also been explored in Yu et al. (2020), although the authors propose using personalization methods on top of robust mechanisms. Our work differs from these works by arguing that MTL itself offers inherent robustness and fairness benefits, and exploring the challenges that exist when attempting to satisfy both constraints simultaneously.

3. Ditto: Global-Regularized Federated Multi-Task Learning

In order to explore the possible fairness/robustness benefits of personalized FL, we first propose a simple and scalable framework for federated multi-task learning. As we will see, this lightweight personalization framework is amenable to analyses while also having strong empirical performance. We explain our proposed objective, Ditto, in Section 3.1 and then present a scalable algorithm to solve it in federated settings (Section 3.2). We provide convergence guarantees for our solver, and explain several practical benefits of our modular approach in terms of privacy and efficiency. Finally, in Section 3.3, we characterize the benefits of Ditto in terms of fairness and robustness on a class of linear problems. We empirically explore the fairness and robustness properties against state-of-the-art baselines in Section 4.

3.1. Ditto Objective

Traditionally, federated learning objectives consider fitting a single global model, w, across all local data in the network. The aim is to solve:

$$\min_w G(F_1(w), \dots F_K(w)), \qquad \text{(Global Obj)}$$

where $F_k(w)$ is the local objective for device k, and $G(\cdot)$ is a function that aggregates the local objectives $\{F_k(w)\}_{k\in[K]}$ from each device. For example, in FedAvg (McMahan et al., 2017), $G(\cdot)$ is typically set to be a weighted average of local losses, i.e., $\sum_{k=1}^K p_k F_k(w)$, where p_k is a pre-defined non-negative weight such that $\sum_k p_k = 1$.

However, in general, each device may generate data x_k via a distinct distribution $\mathcal{D}_k$, i.e., $F_k(w) := \mathbb{E}_{x_k\sim\mathcal{D}_k}[f_k(w; x_k)]$. To better account for this heterogeneity, it is common to consider techniques that learn personalized, device-specific models, $\{v_k\}_{k\in[K]}$ across the network. In this work we explore personalization through a simple framework for federated multi-task learning. We consider two 'tasks': the global objective (Global Obj), and the local objective

$F_k(v_k)$, which aims to learn a model using only the data of device k. To relate these tasks, we incorporate a regularization term that encourages the personalized models to be close to the optimal global model. The resulting bi-level optimization problem for each device $k \in [K]$ is given by:

$$\begin{aligned} \min_{v_k} \quad & h_k(v_k; w^*) := F_k(v_k) + \frac{\lambda}{2} \|v_k - w^*\|^2 \\ \text{s.t.} \quad & w^* \in \arg\min_{w} G(F_1(w), \dots F_K(w))) \,. \end{aligned} \tag{Ditto}$$

Here the hyperparameter λ controls the interpolation between local and global models. When λ is set to 0, Ditto is reduced to training local models; as λ grows large, it recovers global model objective (Global Obj) ($\lambda \to +\infty$).

Intuition for Fairness/Robustness Benefits. In addition to improving accuracy via personalization, we argue that Ditto can offer fairness and robustness benefits. To reason about this, consider a simple case where data are *homogeneous* across devices. Without adversaries, learning a single global model is optimal for generalization. However, in the presence of adversaries, learning globally might introduce corruption, while learning local models may not generalize well due to limited sample size. Ditto with an appropriate value of λ offers a tradeoff between these two extremes: the smaller λ, the more the personalized models v_k can deviate from the (corrupted) global model w, potentially providing robustness at the expense of generalization. In the heterogeneous case (which can lead to issues of unfairness as described in Section 2), a finite λ exists to offer robustness and fairness jointly. We explore these ideas more rigorously in Section 3.3 by analyzing the tradeoffs between accuracy, fairness, and robustness in terms of λ for a class of linear regression problems, and demonstrate fairness/robustness benefits of Ditto empirically in Section 4.

Other Personalization Schemes. As discussed in Section 2, personalization is a widely-studied topic in FL. Our intuition in Ditto is that personalization, by reducing reliance on the global model, can reduce representation disparity (i.e., unfairness) and potentially improve robustness. It is possible that other personalization techniques beyond Ditto offer similar benefits: We provide some initial, encouraging results on this in Section 4.4. However, we specifically explore Ditto due to its simple nature, scalability, and strong empirical performance. Ditto is closely related to works that regularize personalized models towards their average (Hanzely & Richtárik, 2020; Hanzely et al., 2020; Dinh et al., 2020), similar to classical mean-regularized MTL (Evgeniou & Pontil, 2004); Ditto differs by regularizing towards a global model rather than the average personalized model. We find that this provides benefits in terms of *analysis* (Section 3.3), as we can easily reason about Ditto relative to the global ($\lambda \to \infty$) vs. local ($\lambda \to 0$) baselines; *empirically*, in terms of accuracy, fairness, and robustness (Section 4); and *practically*, in terms of the modularity it affords our corresponding solver (Section 3.2).

Other Regularizers. To encourage the personalized models v_k to be close to the optimal global model w^*, there are choices beyond the L_2 norm that could be considered, e.g., using a Bregman divergence-based regularizer or reshaping the L_2 ball using the Fisher information matrix. Under the logistic loss (used in our experiments), the Bregman divergence will reduce to KL divergence, and its second-order Taylor expansion will result in an L_2 ball reshaped with the Fisher information matrix. Such regularizers are studied in other related contexts like continual learning (Kirkpatrick et al., 2017; Schwarz et al., 2018), multi-task learning (Yu et al., 2020), or finetuning for language models (Jiang et al., 2020). However, in our experiments (Section 4.4), we find that incorporating approximate empirical Fisher information (Yu et al., 2020; Kirkpatrick et al., 2017) or symmetrized KL divergence (Jiang et al., 2020) does not improve the performance over the simple L_2 regularized objective, while adding non-trivial computational overhead.

Remark (Relation to FedProx). We note that the L_2 term in Ditto bears resemblance to FedProx, a method which was developed to address heterogeneity in federated optimization (Li et al., 2020d). However, Ditto fundamentally differs from FedProx in that the goal is to learn *personalized* models v_k, while FedProx produces a single global model w. For instance, when the regularization hyperparameter is zero, Ditto reduces to learning separate local models, whereas FedProx would reduce to FedAvg. In fact, Ditto is significantly more general than FedProx in that FedProx could be used as the global model solver in Ditto to optimize $G(\cdot)$. As discussed above, other regularizers beyond the L_2 norm may also be used in practice.

3.2. Ditto Solver

To solve Ditto, we propose jointly solving for the global model w^* and personalized models $\{v_k\}_{k\in[K]}$ in an alternating fashion, as summarized in Algorithm 1. Optimization proceeds in two phases: (i) updates to the global model, w^*, are computed across the network, and then (ii) the personalized models v_k are fit on each local device. The process of optimizing w^* is exactly the same as optimizing for any objective $G(\cdot)$ in federated settings: If we use iterative solvers, then at each communication round, each selected device can solve the local subproblem of $G(\cdot)$ approximately (Line 5). For personalization, device k solves the global-regularized local objective $\min_{v_k} h_k(v_k; w^t)$ inexactly at each round (Line 6). Due to this alternating scheme, our solver can scale well to large networks, as it does not introduce addi-

tional communication or privacy overheads compared with existing solvers for $G(\cdot)$. In our experiments (all except Table 3), we use FedAvg as the objective and solver for $G(\cdot)$, under which we simply let device k run local SGD on F_k (Line 5). We provide a simplified algorithm definition using FedAvg for the w^* update in Algorithm 2 in the appendix.

Algorithm 1: Ditto for Personalized FL

1 **Input:** $K, T, s, \lambda, \eta, w^0, \{v_k^0\}_{k\in[K]}$
2 **for** $t = 0, \cdots, T-1$ **do**
3 Server randomly selects a subset of devices S_t, and sends w^t to them
4 **for** *device* $k \in S_t$ *in parallel* **do**
5 Solve the local sub-problem of $G(\cdot)$ inexactly starting from w^t to obtain w_k^t:
$w_k^t \leftarrow \text{UPDATE\_GLOBAL}(w^t, \nabla F_k(w^t))$
/* Solve $h_k(v_k; w^t)$ */
6 Update v_k for s local iterations:
$v_k = v_k - \eta(\nabla F_k(v_k) + \lambda(v_k - w^t))$
Send $\Delta_k^t := w_k^t - w^t$ back
7 Server aggregates $\{\Delta_k^t\}$:
$w^{t+1} \leftarrow \text{AGGREGATE}\left(w^t, \{\Delta_k^t\}_{k\in\{S_t\}}\right)$
8 **return** $\{v_k\}_{k\in[K]}$ *(personalized)*, w^T *(global)*

We note that another natural choice to solve the Ditto objective is to first obtain w^*, and then for each device k, perform finetuning on the local objective $\min_{v_k} h_k(v_k; w^*)$. These two approaches will arrive at the same solutions in strongly convex cases. In non-convex settings, we observe that there may be additional benefits of joint optimization: Empirically, we find that the updating scheme tends to guide the optimization trajectory towards a better solution compared with finetuning starting from w^*, particularly when w^* is corrupted by adversarial attacks (Section 4.4). Intuitively, under training-time attacks, the global model may start from a random one, get optimized, and gradually become corrupted as training proceeds (Li et al., 2020b). In these cases, feeding in *early* global information (i.e., before the global model converges to w^*) may be helpful under strong attacks.

We note that Ditto with joint optimization requires the devices to maintain local states (i.e., personalized models) and carry these local states to the next communication round where they are selected. Solving Ditto with finetuning does not need devices to be stateful, while losing the benefits of alternate updating discussed above.

Modularity of Ditto. From the Ditto objective and Alg 1, we see that a key advantage of Ditto is its modularity, i.e., that we can readily use prior art developed for the Global Obj along with the personalization add-on of $h_k(v_k; w^*)$, as highlighted in red. This has several benefits:

- *Optimization:* It is possible to plug in other methods beyond FedAvg (e.g., Li et al., 2020c; Karimireddy et al., 2020; Reddi et al., 2021) in Algorithm 1 to update the global model, and inherit the convergence benefits, if any (we make this more precise in Theorem 1).
- *Privacy:* Ditto communicates the same information over the network as typical FL solvers for the global objective, thus preserving whatever privacy or communication benefits exist for the global objective and its respective solver. This is different from most other personalization methods where global model updates depend on local parameters, which may raise privacy concerns (London, 2020).
- *Robustness:* Beyond the inherent robustness benefits of personalization, robust global methods can be used with Ditto to further improve performance (see Section 4.4).

In particular, while not the main focus of our work, we note that Ditto may offer a better *privacy-utility* tradeoff than training a global model. For instance, when training Ditto, if we fix the number of communication rounds and add the same amount of noise per round to satisfy differential privacy, Ditto consumes exactly the same privacy budget as normal global training, while yielding higher accuracy via personalization (Section 4). Similar benefits have been studied, e.g., via finetuning strategies (Yu et al., 2020).

Convergence of Algorithm 1. Note that optimizing the global model w^t does not depend on any personalized models $\{v_k\}_{k\in[K]}$. Therefore, w enjoys the same global convergence rates with the solver we use for G. Under this observation, we present the local convergence of Algorithm 1.

Theorem 1 (Local Convergence of Alg. 1; formal statement and proof in Theorem 10)**.** *Assume for $k \in [K]$, F_k is strongly convex and smooth, under common assumptions, if w^t converges to w^* with rate $g(t)$, then there exists a constant $C<\infty$ such that for $\lambda \in \mathbb{R}$, and for $k \in [K]$, v_k^t converges to $v_k^* := \arg\min_{v_k} h_k(v_k; w^*)$ with rate $Cg(t)$.*

Using Theorem 1, we can directly plug in previous convergence analyses for any $G(\cdot)$. For instance, when the global objective and its solver are those of FedAvg, we can obtain an $O(1/t)$ convergence rate for Ditto under suitable conditions (Corollary 1). We provide a full theorem statement and proof of convergence in Appendix B.

3.3. Analyzing the Fairness/Robustness Benefits of Ditto in Simplified Settings

In this section, we more rigorously explore the fairness/robustness benefits of Ditto on a class of linear prob-

lems. Throughout our analysis, we assume $G(\cdot)$ is the standard objective in FedAvg (McMahan et al., 2017).

Point Estimation. To provide intuition, we first examine a toy one-dimensional point estimation problem. Denote the underlying models for the devices as $\{v_k\}_{k\in[K]}$, $v_k \in \mathbb{R}$, and let the points on device k, $\{x_{k,1}, \ldots, x_{k,n}\}$[1], be observations of v_k with random perturbation, i.e., $x_{k,i} = v_k + z_{k,i}$, where $z_{k,i} \sim \mathcal{N}(0, \sigma^2)$ and are IID. Assume $v_k \sim \mathcal{N}(\theta, \tau^2)$, where θ is drawn from the uniform uninformative prior on $\mathbb{R}$, and τ is a known constant. Here, τ controls the degree of relatedness of the data on different devices: τ=0 captures the case where the data on all devices are identically distributed while $\tau \to \infty$ results in the scenario where the data on different devices are completely unrelated. The local objective is $\min_{v_k} F_k(v_k) = \frac{1}{2}(v_k - \frac{1}{n_k}\sum_{i=1}^{n_k} x_{k,i})^2$. In the presence of adversaries, we look at a specific type of label poisoning attack. Let K_a denote the number of malicious devices, and the 'capability' of an adversary is modeled by τ_a, i.e., the underlying model of an adversary follows $\mathcal{N}(\theta, \tau_a^2)$ where $\tau_a^2 > \tau^2$.

We first derive the Bayes estimator (which will be the most accurate and robust) for the real model distribution by observing a finite number of training points. Then, we show that by solving Ditto, we are able to recover the Bayes estimator with a proper λ^* (with the knowledge of τ). In addition, *the same* λ^* results in the most fair solution among the set of solutions of Ditto parameterized by λ. This shows that Ditto with a proper choice of λ is Bayes optimal for this particular problem instance. In general, in Theorem 8 (appendix), we prove that

$$\lambda^* = \frac{\sigma^2}{n} \frac{K}{K\tau^2 + \frac{K_a}{K-1}(\tau_a^2 - \tau^2)}.$$

We see that λ^* decreases when (i) there are more local samples n, (ii) the devices are less related (larger τ), or (iii) the attacks are stronger (larger number of attackers, K_a, and more powerful adversaries, τ_a). Related theorems (Theorem 6-9) are presented in Appendix A.3.

In Figure 1, we plot average test error, fairness (standard deviation shown as error bars), and robustness (test error in the adversarial case) across a set of λ's for both clean and adversarial cases. We see that in the solution space of Ditto, there exists a specific λ which minimizes the average test error and standard deviation across all devices *at the same time*, which is equal to the optimal λ^* given by our theory. Figure 2 shows (i) Ditto with λ^* is superior than learning local or global models, and (ii) λ^* should increase as the relatedness between devices ($1/\tau$) increases.

[1]For ease of notation, we assume each device has the same number of training samples. It is straightforward to extend the current analysis to allow for varying number of samples per device.

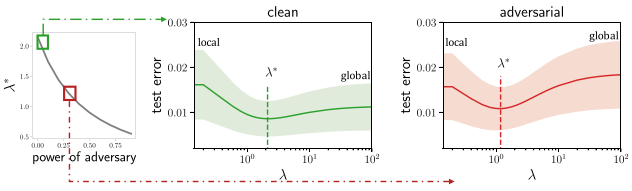

Figure 1. Empirically, the λ^* given by Theorem 6-9 results in the most accurate, fair, and robust solution within Ditto's solution space. λ^* is also optimal in terms of accuracy and robustness among any possible federated estimation algorithms.

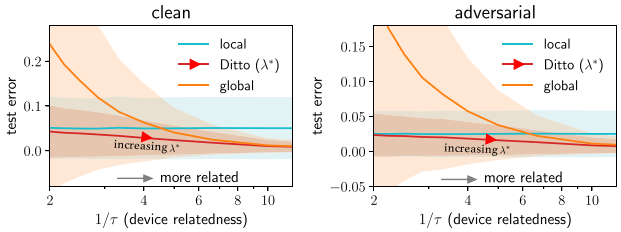

Figure 2. Impact of data relatedness across all devices. When $1/\tau$ is small (less related), local outperforms global; when $1/\tau$ is large (more related), global is better than local. Ditto (λ^*) achieves the lowest test error and variance (measured across benign devices).

Linear Regression. All results discussed above can be generalized to establish the optimality of Ditto on a class of linear regression problems (with additional assumptions on feature covariance). We defer readers to Appendix A.2 for full statements and proofs. While our analyses here are limited to a simplified set of attacks and problem settings, we build on this intuition in Section 4—empirically demonstrating the accuracy, robustness, and fairness benefits of Ditto using both convex and non-convex models, across a range of federated learning benchmarks, and under a diverse set of attacks.

4. Experiments

In this section, we first demonstrate that Ditto can inherently offer similar or superior robustness relative to strong robust baselines (Section 4.1). We then show it results more fair performance than recent fair methods (Section 4.2). Ditto is particularly well-suited for mitigating the tension between these constraints and achieving both fairness and robustness simultaneously (Section 4.3). We explore additional beneficial properties of Ditto in Section 4.4.

Setup. For all experiments, we measure robustness via test accuracy, and fairness via test accuracy variance (or standard deviation), both across benign devices (see Def. 1, 2). We use datasets from common FL benchmarks (Caldas

et al., 2018; Smith et al., 2017; TFF), which cover both vision and language tasks, and convex and non-convex models. Detailed datasets and models are provided in Table 4 in Appendix C. We split local data on each device into train/test/validation sets randomly, and measure performance on the test data. For each device, we select λ locally based on its local validation data. We further assume the devices can make a binary decision on whether the attack is strong or not. For devices with very few validation samples (less than 4), we use a fixed small λ (λ=0.1) for strong attacks, and use a fixed relatively large λ (λ=1) for all other attacks. For devices with more than 5 validation data points, we let each select λ from $\{0.05, 0.1, 0.2\}$ for strong attacks, and select λ from $\{0.1, 1, 2\}$ for all other attacks. See Appendix D.2 for details. More advanced tuning methods are left for future work. Our code, data, and experiments are publicly available at `github.com/litian96/ditto`.

4.1. Robustness of Ditto

Following our threat model described in Definition 1, we apply three attacks to corrupt a random subset of devices. We pick corruption levels until a point where there is a significant performance drop when training a global model. We compare robustness (Def. 1) of Ditto with various defense baselines, presenting the results of three strongest defenses in Figure 3. Execution details and full results are reported in Appendix D.4. As shown in Figure 3, Ditto achieves the highest accuracy under most attacks, particularly those with a large fraction of malicious devices. On average across all datasets and attacks, Ditto results in $\sim$6% absolute accuracy improvement compared with the strongest robust baseline (Appendix D.4). In scenarios where a robust baseline outperforms Ditto, we have also found that replacing the global objective and its solver (FedAvg) with a robust version (e.g., using robust aggregators) can further improve Ditto, yielding superior performance (Section 4.4).

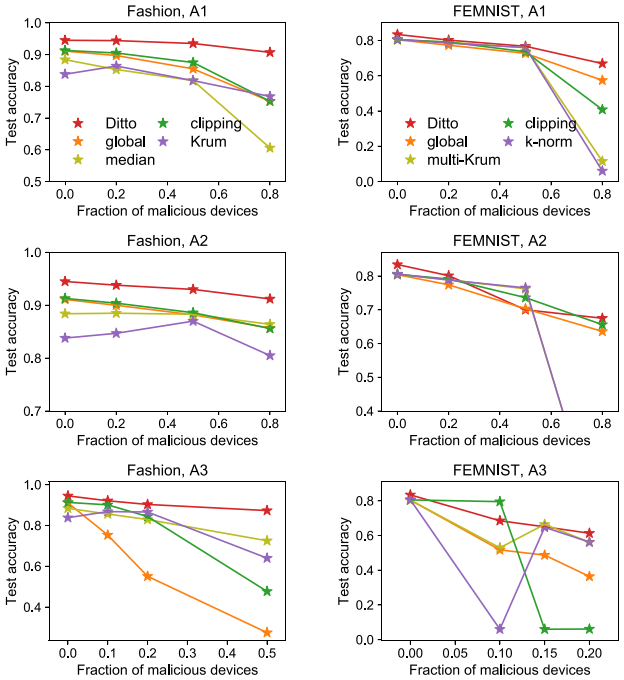

Figure 3. Robustness, i.e., average test accuracy on benign devices (Definition 1), on Fashion MNIST and FEMNIST. We compare Ditto with learning a global model and three strong defense mechanisms (see Appendix D for results on all defense baselines), and find that Ditto is the most robust under almost all attacks.

4.2. Fairness of Ditto

To explore the fairness of Ditto, we compare against TERM (Li et al., 2021) as a baseline. It is an improved version of the q-FFL (Li et al., 2020e) objective, which has been recently proposed for fair federated learning. TERM also recovers AFL (Mohri et al., 2019), another fair FL objective, as a special case. TERM uses a parameter t to offer flexible tradeoffs between fairness and accuracy. In Table 1, we compare the proposed objective with global, local, and fair methods (TERM) in terms of test accuracies and standard deviation. When the corruption level is high, 'global' or 'fair' will even fail to converge. Ditto results in more accurate and fair solutions both with and without attacks. On average across all datasets, Ditto reduces variance across devices by $\sim$10% while improving absolute test accuracy by 5% compared with TERM (on clean data).

4.3. Addressing Competing Constraints

In this section, we examine the competing constraints between robustness and fairness. When training a single global model, fair methods aim to encourage a more uniform performance distribution, but may be highly susceptible to training-time attacks in statistically heterogeneous environments. We investigate the test accuracy on benign devices when learning global, local, and fair models. In the TERM objective, we set $t = 1, 2, 5$ to achieve different levels of fairness (the higher, the fairer). We perform the data poisoning attack (A1 in Def. 1). The results are plotted in Figure 4. As the corruption level increases, we see that fitting a global model becomes less robust. Using fair methods will be more susceptible to attacks. When t gets larger, the test accuracy gets lower, an indication that the fair method is overfitting to the corrupted devices relative to the global baseline.

Next, we apply various strong robust methods under the same attack, and explore the robustness/accuracy and fairness performance. The robust approaches include: Krum, multi-Krum (Blanchard et al., 2017), taking the coordinate-wise median of gradients ('median'), gradient clipping ('clipping'), filtering out the gradients with largest norms ('k-norm'), and taking the gradient of the k-th largest loss

Table 1. **Average (standard deviation)** test accuracy to benchmark performance and fairness (Definition 2) on Fashion MNIST and FEMNIST. Ditto is either (i) more fair compared with the baselines of training a global model, or (ii) more accurate than the fair baseline under a set of attacks. We bold the method with highest average minus standard deviation across all methods.

Fashion		**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	clean	20%	50%	80%	20%	50%	80%	10%	20%	50%
global	.911 (.08)	.897 (.08)	.855 (.10)	.753 (.13)	.900 (.08)	.882 (.09)	.857 (.10)	.753 (.10)	.551 (.13)	.275 (.12)
local	.876 (.10)	.874 (.10)	.876 (.11)	.879 (.10)	.874 (.10)	.876 (.11)	.879 (.10)	.877 (.10)	.874 (.10)	**.876 (.11)**
fair (TERM, t=1)	.909 (.07)	.751 (.12)	.637 (.13)	.547 (.11)	.731 (.13)	.637 (.14)	.635 (.14)	.653 (.13)	.601 (.12)	.131 (.16)
Ditto	**.943 (.06)**	**.944 (.07)**	**.937 (.07)**	**.907 (.10)**	**.938 (.07)**	**.930 (.08)**	**.913 (.09)**	**.921 (.09)**	**.902 (.09)**	.873 (.11)
FEMNIST		**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	clean	20%	50%	80%	20%	50%	80%	10%	15%	20%
global	.804 (.11)	.773 (.11)	.727 (.12)	.574 (.15)	.774 (.11)	**.703 (.14)**	.636 (.15)	.517 (.14)	.487 (.14)	.314 (.13)
local	.628 (.15)	.620 (.14)	.627 (.14)	.607 (.14)	.620 (.14)	.627 (.14)	.607 (.14)	.622 (.14)	.621 (.14)	**.620 (.14)**
fair (TERM, t=1)	.809 (.11)	.636 (.15)	.562 (.13)	.478 (.12)	.440 (.15)	.336 (.12)	.363 (.12)	.353 (.12)	.316 (.12)	.299 (.11)
Ditto	**.834 (.09)**	**.802 (.10)**	**.762 (.11)**	**.672 (.13)**	**.801 (.09)**	.700 (.15)	**.675 (.14)**	**.685 (.15)**	**.650 (.14)**	.613 (.13)

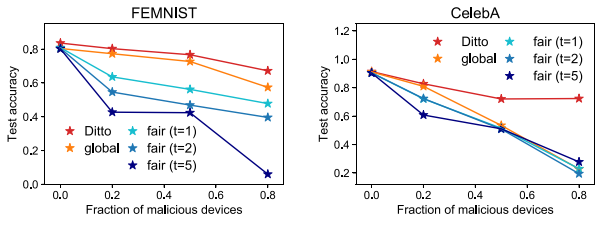

Figure 4. Fair methods can overfit to corrupted devices (possibly with large training losses) by imposing more weights on them, thus being particularly susceptible to attacks.

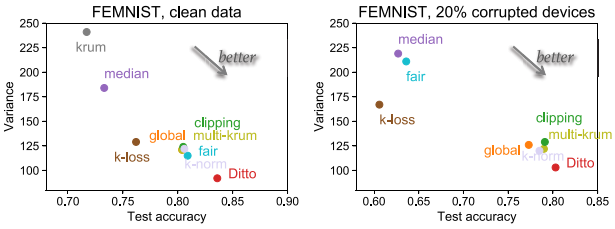

Figure 5. Compared with learning a global model, robust baselines (i.e., the methods listed in the figure excluding 'global' and 'Ditto') are either robust but not fair (with higher accuracy, larger variance), or not even robust (with lower accuracy). Ditto lies at the lower right corner, which is our preferred region.

where k is the number of malicious devices ('k-loss'). For Krum, multi-Krum, k-norm, and k-loss, we assume that the server knows the expected number of malicious devices that are selected each round, and can set k accordingly for k-norm and k-loss. From Figure 5, we see that robust baselines are either (i) more robust than global but less fair, or (ii) fail to provide robustness due to heterogeneity. Ditto is more robust, accurate, and fair.

4.4. Additional Properties of Ditto

Personalization. We additionally explore the performance of other personalized FL methods in terms of accuracy and fairness, on both clean and adversarial cases. In particular, we consider objectives that (i) regularize with the average (L2SGD (Hanzely & Richtárik, 2020)) or the learnt device relationship matrix (MOCHA (Smith et al., 2017)), (ii) encourage closeness to the global model in terms of some specific function behavior (EWC (Kirkpatrick et al., 2017; Yu et al., 2020) and Symmetrized KL (SKL)), (iii) interpolate between local and global models (APFL (Deng et al., 2021) and mapper (Mansour et al., 2020)), and (iv) have been motivated by meta-learning (Per-FedAvg (HF) (Fallah et al., 2020)). We provide a detailed description in Appendix C.

We compare Ditto with the above alternatives, using the same learning rate tuned on FedAvg on clean data for all methods except Per-FedAvg, which requires additional tuning to prevent divergence. For finetuning methods (EWC and SKL), we finetune on each local device for 50 epochs starting from the converged global model. We report results of baseline methods using their best hyperparameters. Despite Ditto's simplicity, in Table 2 below, we see that Ditto achieves similar or superier test accuracy with slightly lower standard deviation compared with these recent personalization methods.

We also evaluate the performance of MOCHA with a convex SVM model in Table 7 in the appendix. MOCHA is more robust and fair than most baselines, which is in line with our reasoning that personalization can provide benefits for these constraints. Further understanding the robustness/fairness benefits of other personalized approaches would be an interesting direction of future work.

Table 2. Ditto is competitive with or outperforms other recent personalization methods. We report the average (standard deviation) of test accuracies across all devices to capture performance and fairness (Definition 2), respectively.

	Clean		50% Adversaries (A1)	
Methods	**FEMNIST**	**CelebA**	**FEMNIST**	**CelebA**
global	.804 (.11)	.911 (.19)	.727 (.12)	.538 (.28)
local	.628 (.15)	.692 (.27)	.627 (.14)	.682 (.27)
plain finetuning	.815 (.09)	.912 (.18)	.734 (.12)	.721 (.28)
L2SGD	.817 (.10)	.899 (.18)	.732 (.15)	.725 (.25)
EWC	.810 (.11)	.910 (.18)	.756 (.12)	.642 (.26)
SKL	.820 (.10)	**.915 (.16)**	.752 (.12)	.708 (.27)
Per-FedAvg (HF)	.827 (.09)	.907 (.17)	.604 (.14)	**.756 (.26)**
mapper	.792 (.12)	.773 (.25)	.726 (.13)	.704 (.27)
APFL	.811 (.11)	.911 (.17)	.750 (.11)	.710 (.27)
Ditto	**.836 (.10)**	.914 (.18)	**.767 (.10)**	.721 (.27)

Augmenting with Robust Baselines. Ditto allows the flexibility of learning robust w^* leveraging any previous robust aggregation techniques, which could further improve the performance of personalized models. For instance, in the aggregation step at the server side (Line 7 in Algorithm 1), instead of simply averaging the global model updates as in FedAvg, we can aggregate them via multi-Krum, or after gradient clipping. As is shown in Table 3, Ditto combined with clipping yields improvements compared with vanilla Ditto. We present full results on different datasets trying varying robust methods in Table 6 in the appendix.

Table 3. Augmenting Ditto with robust baselines can further improve performance.

FEMNIST	**A1**		**A2**		**A3**	
Methods	20%	80%	20%	80%	10%	20%
global	.773	.574	.774	.636	.517	.364
clipping	.791	.408	.791	.656	.795	.061
Ditto	.803	**.669**	.792	.681	.695	.650
Ditto + clipping	**.810**	.645	**.808**	**.684**	**.813**	**.672**

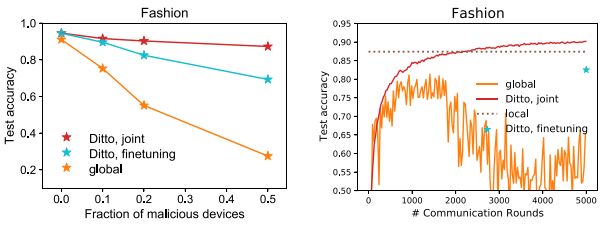

Figure 6. Ditto with joint optimization (Algorithm 1) outperforms the alternative local finetuning solver under the strong model replacement attack.

Comparing Two Solvers. As mentioned in Section 3.2, another way to solve Ditto is to finetune on $\min_{v_k} h_k(v_k; w^*)$ for each $k \in [K]$ after obtaining w^*. We examine the performance of two solvers under the model replacement attack (A3) with 20% adversaries. In realistic federated networks, it may be challenging to determine how many iterations to finetune for, particularly over a heterogeneous network of devices. To obtain the best performance of finetuning, we solve $\min_{v_k} h_k(v_k; w^*)$ on each device by running different iterations of mini-batch SGD and pick the best one. As shown in Figure 6, the finetuning solver improves the performance compared with learning a global model, while Ditto combined with joint optimization performs the best. One can also perform finetuning after early stopping; however, it is essentially solving a different objective and it is difficult to determine the stopping criteria. We discuss this in more detail in Appendix D.1.

5. Conclusion and Future Work

We propose Ditto, a simple MTL framework, to address the competing constraints of accuracy, fairness, and robustness in federated learning. Ditto can be thought of as a lightweight personalization add-on for any global federated objective, which maintains the privacy and communication efficiency of the global solver. We theoretically analyze the ability of Ditto to mitigate the tension between fairness and robustness on a class of linear problems. Our empirical results demonstrate that Ditto can result in both more robust and fairer models compared with strong baselines across a diverse set of attacks. Our work suggests several interesting directions of future study, such as exploring the applicability of Ditto to other attacks such as backdoor attacks (e.g., Sun et al., 2019); understanding the fairness/robustness properties of other personalized methods; and considering additional constraints, such as privacy.

Acknowledgements

The work of TL, SH, and VS was supported in part by the National Science Foundation Grant IIS1838017, a Google Faculty Award, a Facebook Faculty Award, and the CONIX Research Center. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the National Science Foundation or any other funding agency.

References

Tensorflow federated: Machine learning on decentralized data. URL https://www.tensorflow.org/federated.

Agarwal, A., Langford, J., and Wei, C.-Y. Federated residual

learning. *arXiv preprint arXiv:2003.12880*, 2020.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Bhagoji, A. N., Chakraborty, S., Mittal, P., and Calo, S. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, 2019.

Biggio, B., Nelson, B., and Laskov, P. Support vector machines under adversarial label noise. In *Asian Conference on Machine Learning*, 2011.

Biggio, B., Nelson, B., and Laskov, P. Poisoning attacks against support vector machines. In *International Conference on Machine Learning*, 2012.

Blanchard, P., Mhamdi, E. M. E., Guerraoui, R., and Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, 2017.

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

Chang, H., Nguyen, T. D., Murakonda, S. K., Kazemi, E., and Shokri, R. On adversarial bias and the robustness of fair machine learning. *arXiv preprint arXiv:2006.08669*, 2020.

Chen, F., Luo, M., Dong, Z., Li, Z., and He, X. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876*, 2018.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

Deng, Y., Kamani, M. M., and Mahdavi, M. Distributionally robust federated averaging. *Advances in Neural Information Processing Systems*, 2020.

Deng, Y., Kamani, M. M., and Mahdavi, M. Adaptive personalized federated learning, 2021. URL https://openreview.net/forum?id=g0a-XYjpQ7r.

Dinh, C. T., Tran, N. H., and Nguyen, T. D. Personalized federated learning with moreau envelopes. In *Advances in Neural Information Processing Systems*, 2020.

Duarte, M. F. and Hu, Y. H. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 2004.

Dumford, J. and Scheirer, W. Backdooring convolutional neural networks via targeted weight perturbations. *arXiv preprint arXiv:1812.03128*, 2018.

Evgeniou, T. and Pontil, M. Regularized multi–task learning. In *International Conference on Knowledge Discovery and Data Mining*, 2004.

Fallah, A., Mokhtari, A., and Ozdaglar, A. Personalized federated learning: A meta-learning approach. In *Advances in Neural Information Processing Systems*, 2020.

Fang, M., Cao, X., Jia, J., and Gong, N. Local model poisoning attacks to byzantine-robust federated learning. In *USENIX Security Symposium*, 2020.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

Ghosh, A., Chung, J., Yin, D., and Ramchandran, K. An efficient framework for clustered federated learning. In *Advances in Neural Information Processing Systems*, 2020.

Gu, T., Dolan-Gavitt, B., and Garg, S. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

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

Hanzely, F., Hanzely, S., Horváth, S., and Richtárik, P. Lower bounds and optimal algorithms for personalized federated learning. *Advances in Neural Information Processing Systems*, 2020.

Hao, W., Mehta, N., Liang, K. J., Cheng, P., El-Khamy, M., and Carin, L. Waffle: Weight anonymized factorization for federated learning. *arXiv preprint arXiv:2008.05687*, 2020.

Hashimoto, T., Srivastava, M., Namkoong, H., and Liang, P. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, 2018.

He, L., Karimireddy, S. P., and Jaggi, M. Byzantine-robust learning on heterogeneous datasets via resampling. In *NeurIPS Workshop on Scalability, Privacy, and Security in Federated Learning*, 2020.

Hu, Z., Shaloudegi, K., Zhang, G., and Yu, Y. FedMGDA+: Federated learning meets multi-objective optimization. *arXiv preprint arXiv:2006.11489*, 2020.

Huang, W. R., Geiping, J., Fowl, L., Taylor, G., and Goldstein, T. Metapoison: Practical general-purpose clean-label data poisoning. In *Advances in Neural Information Processing Systems*, 2020.

Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Zhao, T. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Jiang, Y., Konečnỳ, J., Rush, K., and Kannan, S. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

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

Khodak, M., Balcan, M.-F. F., and Talwalkar, A. S. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems*, 2019.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 2017.

Lamport, L., Shostak, R., and Pease, M. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*. 2019.

Li, J., Khodak, M., Caldas, S., and Talwalkar, A. Differentially private meta-learning. In *International Conference on Learning Representations*, 2020a.

Li, L., Xu, W., Chen, T., Giannakis, G. B., and Ling, Q. Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *AAAI Conference on Artificial Intelligence*, 2019.

Li, M., Soltanolkotabi, M., and Oymak, S. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2020b.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. In *Conference on Machine Learning and Systems*, 2020c.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2020d.

Li, T., Sanjabi, M., Beirami, A., and Smith, V. Fair resource allocation in federated learning. In *International Conference on Learning Representations*, 2020e.

Li, T., Beirami, A., Sanjabi, M., and Smith, V. Tilted empirical risk minimization. In *International Conference on Learning Representations*, 2021.

Li, X., Huang, K., Yang, W., Wang, S., and Zhang, Z. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2020f.

Liang, P. P., Liu, T., Ziyin, L., Salakhutdinov, R., and Morency, L.-P. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

Liu, Y., Ma, S., Aafer, Y., Lee, W., Zhai, J., Wang, W., and Zhang, X. Trojaning attack on neural networks. In *Network and Distributed System Security Symposium*, 2018.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *International Conference on Computer Vision*, 2015.

London, B. PAC identifiability in federated personalization. In *NeurIPS 2020 Workshop on Scalability, Privacy, and Security in Federated Learning*, 2020.

Mahdavifar, H., Beirami, A., Touri, B., and Shamma, J. S. Global games with noisy information sharing. *IEEE Transactions on Signal and Information Processing over Networks*, 2018.

Mansour, Y., Mohri, M., Ro, J., and Suresh, A. T. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, 2017.

Mohri, M., Sivek, G., and Suresh, A. T. Agnostic federated learning. In *International Conference on Machine Learning*, 2019.

Muhammad, K., Wang, Q., O'Reilly-Morgan, D., Tragos, E., Smyth, B., Hurley, N., Geraci, J., and Lawlor, A. Fedfast: Going beyond average for faster training of federated recommender systems. In *International Conference on Knowledge Discovery & Data Mining*, 2020.

Pillutla, K., Kakade, S. M., and Harchaoui, Z. Robust aggregation for federated learning. *arXiv preprint arXiv:1912.13445*, 2019.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

Sattler, F., Müller, K.-R., and Samek, W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, 2018.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Singhal, K., Sidahmed, H., Garrett, Z., Wu, S., Rush, K., and Prakash, S. Federated reconstruction: Partially local federated learning. *arXiv preprint arXiv:2102.03448*, 2021.

Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. S. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, 2017.

Sun, G., Cong, Y., Dong, J., Wang, Q., and Liu, J. Data poisoning attacks on federated machine learning. *arXiv preprint arXiv:2004.10020*, 2020.

Sun, Z., Kairouz, P., Suresh, A. T., and McMahan, H. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

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

Wang, K., Mathews, R., Kiddon, C., Eichner, H., Beaufays, F., and Ramage, D. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019.

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

Xie, C., Huang, K., Chen, P.-Y., and Li, B. DBA: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2020.

Xu, X. and Lyu, L. Towards building a robust and fair federated learning system. *arXiv preprint arXiv:2011.10464*, 2020.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, 2018.

Yu, T., Bagdasaryan, E., and Shmatikov, V. Salvaging federated learning by local adaptation. *arXiv preprint arXiv:2002.04758*, 2020.

Zhang, M., Sapra, K., Fidler, S., Yeung, S., and Alvarez, J. M. Personalized federated learning with first order model optimization. In *International Conference on Learning Representations*, 2021.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.