DITTO: FAIR AND ROBUST FEDERATED LEARNING THROUGH PERSONALIZATION

Tian Li
CMU
tianli@cmu.edu

Shengyuan Hu
CMU
shengyua@andrew.cmu.edu

Ahmad Beirami
Facebook AI
beirami@fb.com

Virginia Smith
CMU
smithv@cmu.edu

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, 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]

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 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., 2020d; 2021). Robust methods commonly use techniques such as gradient clipping (Sun et al., 2019) or robust aggregation (Blanchard et al., 2017).

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, efficiency, and convergence properties as traditional FL. We evaluate Ditto on a suite of federated benchmarks and show that, surprisingly,

[1] A full version of this paper is also available at https://arxiv.org/abs/2012.04221.

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.

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. We give a complete overview of prior work in robustness, fairness, and personalization in the context of federated learning in Appendix A.

Robustness in Federated Learning. 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 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).
In terms of defenses, 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 network. 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. (2020d), 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.,* $\text{std}\left\{F_k(w_1)\right\}_{k\in[K]} < \text{std}\left\{F_k(w_2)\right\}_{k\in[K]}$ *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.*

Personalized Federated Learning. We defer the discussions on personalized FL to Appendix A.

3 DITTO: GLOBAL-REGULARIZED FEDERATED MULTI-TASK LEARNING

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

$$\min_w \; G(F_1(w), \dots \; F_K(w)) \,, \tag{Global Obj}$$

where $F_k(w)$ is the local objective for device k, and $G(\cdot)$ is a function that aggregates the local objectives $\left\{F_k(w)\right\}_{k\in[K]}$ 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., $F_k(w) := \mathbb{E}_{x_k\sim\mathcal{D}_k}\left[f_k(w; x_k)\right]$. To better account for this heterogeneity, it is common to consider techniques that learn personalized, device-specific models, $\left\{v_k\right\}_{k\in[K]}$ 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:

$$\min_{v_k} h_k(v_k; w^*) := F_k(v_k) + \frac{\lambda}{2}\left\|v_k - w^*\right\|^2 , \text{ s.t. } w^* \in \arg\min_w G(\left\{F_k(w)\right\}_{k\in[K]}) \,. \tag{Ditto}$$

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 optimization (Global Obj) ($\lambda \to +\infty$).

Intuition for Fairness/Robustness Benefits. In addition to improving accuracy via personalization, we suggest Ditto may offer fairness and robustness benefits. To reason about this, consider a simple case where the 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 (see theoretical analysis in Section 3.2 and empirical results in Section 4).

3.1 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 in Appendix D. 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 solves 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 additional communication or privacy overheads compared with existing solvers for $G(\cdot)$.

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, the updating scheme tends to guide the optimization trajectory towards a better solution compared with finetuning starting from w^*, particularly when w^* gets corrupted by adversarial attacks (Appendix G.3).

Modularity of Ditto. From the Ditto objective and Algorithm 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 in Algorithm 1. We discuss its benefits in terms of optimization, privacy, and robustness in Appendix E.1.

3.2 ANALYZING THE FAIRNESS/ROBUSTNESS BENEFITS OF DITTO IN SIMPLIFIED SETTINGS

We now more rigorously explore the fairness/robustness benefits of Ditto on a class of linear problems. Throughout, we assume $G(\cdot)$ is the standard objective in FedAvg (McMahan et al., 2017).

To provide intuition, we first examine a toy one-dimensional point estimation problem. Denote the underlying models for the devices as $\{w_k\}_{k\in[K]}$, $w_k \in \mathbb{R}$, and let the points on device k, $\{x_{k,1}, \ldots, x_{k,n}\}$ be observations of w_k with random perturbation, i.e., $x_{k,i} = w_k + z_{k,i}$, where $z_{k,i} \sim \mathcal{N}(0, \sigma^2)$ and are IID. Assume $w_k \sim \mathcal{N}(\theta, \tau^2)$, 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. At a high level, we prove that λ^* should be smaller when there are more local samples, or the devices are less related, or there are more malicious devices (i.e., stronger attacks) (Theorem 5-8).

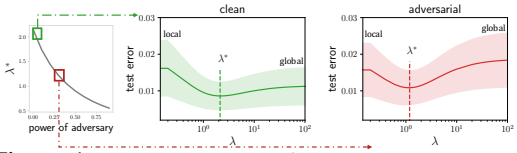

Figure 1: Empirically, the λ^* given by Theorem 5-8 results in the most accurate, fair, and robust solution within Ditto's solution space.

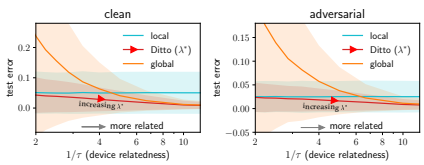

Figure 2: The impact of data relatedness across all devices.

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.

All results discussed above can be generalized to establish the optimality of `Ditto` on a class of linear regression problems (Appendix B.2).

4 EXPERIMENTS

Setup. For each device, we select λ locally based on its local validation data. See Appendix F for experimental details. Our code is publicly available at `github.com/litian96/ditto`.

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. `Ditto` achieves the highest accuracy under most attacks.

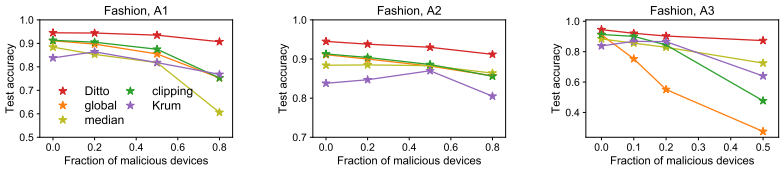

Figure 3: Robustness, i.e., average test accuracy on benign devices (Definition 1) on Fashion MNIST. Execution details and full results are reported in Appendix G.6. `Ditto` is the most robust under almost all attacks.

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., 2020d) 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 2 in Appendix G, 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.

Addressing Competing Constraints. 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 5. 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.

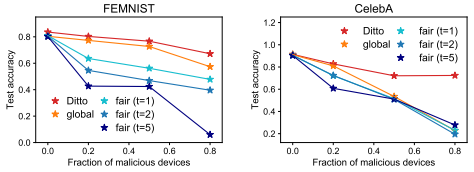

Figure 5: Fair methods can overfit to corrupted devices (possibly with large training losses) by imposing more weights on them.

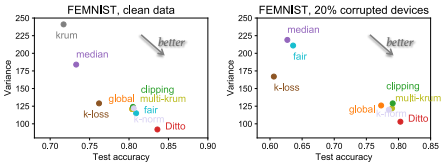

Figure 6: Compared with learning a global model, robust baselines are either robust but not fair, or not even robust. `Ditto` lies at the lower right corner.

Next, we apply various strong robust methods under the same attack, and explore the robustness/accuracy and fairness performance. For Krum and multi-Krum (Blanchard et al., 2017), we assume that the server knows the expected number of malicious devices. Other robust approaches include: taking the coordinate-wise median of gradients ('median'), gradient clipping ('clipping'), filtering out the gradients with largest norms ('k-norm'), and taking the gradient with the k-th largest loss where k is the number of malicious devices ('k-loss'). From Figure 6, 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.

REFERENCES

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

Alekh Agarwal, John Langford, and Chen-Yu Wei. Federated residual learning. *arXiv preprint arXiv:2003.12880*, 2020.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, 2019.

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

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

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

Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

Hongyan Chang, Ta Duy Nguyen, Sasi Kumar Murakonda, Ehsan Kazemi, and Reza Shokri. On adversarial bias and the robustness of fair machine learning. *arXiv preprint arXiv:2006.08669*, 2020.

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.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Distributionally robust federated averaging. *Advances in Neural Information Processing Systems*, 2020.

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

Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. In *Advances in Neural Information Processing Systems*, 2020.

Marco F Duarte and Yu Hen Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 2004.

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

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

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

Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to byzantine-robust federated learning. In $USENIX$ *Security Symposium*, 2020.

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

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In *Advances in Neural Information Processing Systems*, 2020.

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

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

Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtárik. Lower bounds and optimal algorithms for personalized federated learning. *Advances in Neural Information Processing Systems*, 2020.

Weituo Hao, Nikhil Mehta, Kevin J Liang, Pengyu Cheng, Mostafa El-Khamy, and Lawrence Carin. Waffle: Weight anonymized factorization for federated learning. *arXiv preprint arXiv:2008.05687*, 2020.

Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, 2018.

Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via resampling. In *SpicyFL - NeurIPS Workshop on Scalability, Privacy, and Security in Federated Learning*, 2020.

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

W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. In *Advances in Neural Information Processing Systems*, 2020.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 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.

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

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

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

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

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 2017.

Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*. 2019.

Jeffrey Li, Mikhail Khodak, Sebastian Caldas, and Ameet Talwalkar. Differentially private meta-learning. In *International Conference on Learning Representations*, 2020a.

Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling. Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *AAAI Conference on Artificial Intelligence*, 2019.

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

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Conference on Machine Learning and Systems*, 2020c.

Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *International Conference on Learning Representations*, 2020d.

Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. Tilted empirical risk minimization. In *International Conference on Learning Representations*, 2021.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2020e.

Paul Pu Liang, Terrance Liu, Liu Ziyin, Ruslan Salakhutdinov, and Louis-Philippe Morency. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*, 2020.

Yingqi Liu, Shiqing Ma, Yousra Aafer, W. Lee, Juan Zhai, Weihang Wang, and X. Zhang. Trojaning attack on neural networks. In *Network and Distributed System Security Symposium*, 2018.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *International Conference on Computer Vision*, 2015.

Hessam Mahdavifar, Ahmad Beirami, Behrouz Touri, and Jeff S Shamma. Global games with noisy information sharing. *IEEE Transactions on Signal and Information Processing over Networks*, 2017.

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.

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

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, 2019.

Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *arXiv preprint arXiv:1912.13445*, 2019.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, 2018.

Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, 2018.

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

Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang, and Ji Liu. Data poisoning attacks on federated machine learning. *arXiv preprint arXiv:2004.10020*, 2020.

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

Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In *Advances in Neural Information Processing Systems*, 2020.

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

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

Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. DBA: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2020.

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

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

Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. In *International Conference on Learning Representations*, 2021.

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

APPENDIX

We provide a table of contents below for easier navigation of the appendix.

CONTENTS

A Related Work **10**

B Analysis of the Federated Multi-Task Learning Objective `Ditto` **12**
B.1 Properties of `Ditto` for Strongly Convex Functions 12
B.2 Optimality of `Ditto` for Federated Linear Regression 13
B.3 Optimality of `Ditto` for Federated Point Estimation 19

C Other Personalization Schemes and Regularizers **22**

D Algorithms **23**

E Convergence Analysis **24**
E.1 Modularity of `Ditto` 26

F Experimental Details **27**
F.1 Datasets and Models 27
F.2 Personalization Baselines 27

G Additional and Complete Experiment Results **28**
G.1 Fairness of Ditto 28
G.2 Personalization 28
G.3 Comparing Two Solvers 28
G.4 Tuning λ 29
G.5 `Ditto` Augmented with Robust Baselines 30
G.6 `Ditto` Complete Results 31

H Conclusion and Future Work **34**

A 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).

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. We note that there exists a tension between variance and utility in our fairness definition (Definition 2); in general, a 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. (2020d; 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). The tension between fairness and robustness has been observed or studied in previous works, though for different notions of fairness (equalized odds) or robustness (backdoor attacks), or in centralized settings Chang et al. (2020); Wang et al. (2020). Recently, Hu et al. (2020) have proposed FedMGDA+, a method targeting fair and robust FL; however, this work combines classical fairness (minimax optimization) and robustness (gradient normalization) mechanisms, as opposed to the multi-task framework proposed herein, which we show can *inherently* provide both benefits 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), 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), and other forms of MTL, such as hard 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, showing that `Ditto` achieves similar or superior performance across 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.

B ANALYSIS OF THE FEDERATED MULTI-TASK LEARNING OBJECTIVE DITTO

Here, we provide theoretical analyses of Ditto, mainly on a class of linear models. In this linear setting, we investigate accuracy, fairness, and robustness of Ditto. We first discuss some general properties of Ditto for strongly convex functions in terms of the training performance in Section B.1. We next present our main results on characterizing the benefits (accuracy, fairness, and robustness) of Ditto on linear regression in Section B.2. Finally, we present results on a special case of linear regression (federated point estimation problem examined in Section 3.2) in Section B.3.

B.1 PROPERTIES OF DITTO FOR STRONGLY CONVEX FUNCTIONS

Let the Ditto objective on device k be

$$h_k(w) = F_k(w) + \lambda\psi(w), \tag{1}$$

where F_k is strongly convex, and

$$\psi(w) := \frac{1}{2}\|w - w^*\|^2, \tag{2}$$

$$w^* := \arg\min_w \left\{ \frac{1}{K} \sum_{k\in[K]} F_k(w) \right\}. \tag{3}$$

Let

$$\widehat{w}_k(\lambda) = \arg\min_w h_k(w). \tag{4}$$

Without any distributional assumptions on the tasks, we first characterize the solutions of the objective $h_k(w)$.

Lemma 1. *For all $\lambda \geq 0$,*

$$\frac{\partial}{\partial\lambda} F_k(\widehat{w}_k(\lambda)) \geq 0, \tag{5}$$

$$\frac{\partial}{\partial\lambda} \psi(\widehat{w}_k(\lambda)) \leq 0. \tag{6}$$

In addition, for all k, if $F_k(w^*)$ is finite, then

$$\lim_{\lambda\to\infty} \widehat{w}_k(\lambda) = w^*. \tag{7}$$

Proof. The proof here directly follows the proof in Hanzely & Richtárik (Theorem 3.1, 2020). □

As λ increases, the local empirical training loss $F_k(\widehat{w}_k(\lambda))$ will also increase, and the resulting personalized models will be closer to the global model. Therefore, λ effectively controls how much personalization we impose. Since for any device $k \in [K]$, training loss is minimized when $\lambda = 0$, training separate local models is the most robust and fair *in terms of training performance when we do not consider generalization*.

However, in order to obtain the guarantees on the test performance, we need to explicitly model the joint distribution of data on all devices. In the next section, we explore a Bayesian framework on a class of linear problems to examine the generalization, fairness, and robustness of the Ditto objective, all on the underlying test data.

B.2 OPTIMALITY OF DITTO FOR FEDERATED LINEAR REGRESSION

We first examine the case without corrupted devices in Section B.2.1. 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^*. In addition, *the same* λ^* results in the most fair solution among the set of solutions of Ditto parameterized by λ. When there are adversaries, we analyze the robustness benefits of Ditto in Section B.2.2. In particular, we show there exists a λ which leads to the highest test accuracy across benign devices (i.e., the most robust) and minimizes the variance of the test error across benign devices (i.e., the most fair) jointly.

Before we proceed, we first state a technical lemma that will be used throughout the analyses.

Lemma 2. *Let θ be drawn from the non-informative uniform prior on $\mathbb{R}^d$. Further, let $\{\phi_k\}_{k\in[K]}$ denote noisy observations of θ with additive zero-mean independent Gaussian noises with covariance matrices $\{\Sigma_k\}_{k\in[K]}$. Let*

$$\Sigma_\theta := \left(\sum_{k\in[K]} \Sigma_k^{-1}\right)^{-1}. \tag{8}$$

Then, conditioned on $\{\phi_k\}_{k\in[K]}$, we can write θ as

$$\theta = \Sigma_\theta \sum_{k\in[K]} \Sigma_k^{-1}\phi_k + z,$$

where z is $\mathcal{N}(0, \Sigma_\theta)$ which is independent of $\{\phi_k\}_{k\in[K]}$.

Lemma 2 is a generalization of Lemma 11 presented in Mahdavifar et al. (2017) (restated in Lemma 3 below) to the multivariate Gaussian case. The proof also follows from the proof in Mahdavifar et al. (2017).

Lemma 3 (Lemma 11 in Mahdavifar et al. (2017))**.** *Let θ be drawn from the non-informative uniform prior on $\mathbb{R}$. Further, let $\{\phi_k\}_{k\in[K]}$ denote noisy observations of θ with additive zero-mean independent Gaussian noises with variances $\{\sigma_k^2\}_{k\in[K]}$. Let*

$$\frac{1}{\sigma_\theta^2} := \sum_{k\in[K]} \frac{1}{\sigma_k^2}. \tag{9}$$

Then, conditioned on $\{\phi_k\}_{k\in[K]}$, we can write θ as

$$\theta = \sigma_\theta^2 \sum_{k\in[K]} \frac{\phi_k}{\sigma_k^2} + z,$$

where z is $\mathcal{N}(0, \sigma_\theta^2)$ which is independent of $\{\phi_k\}_{k\in[K]}$.

B.2.1 NO ADVERSARIES: DITTO FOR ACCURACY AND FAIRNESS

We consider a Bayesian framework. Let θ be drawn from the non-informative prior on $\mathbb{R}^d$, i.e., uniformly distributed on $\mathbb{R}^d$. We assume that K devices have their data distributed with parameters $\{w_k\}_{k\in[K]}$:

$$w_k = \theta + \zeta_k, \tag{10}$$

where $\zeta_k \sim \mathcal{N}(0, \tau^2\mathbf{I}_d)$ are I.I.D, and $\mathbf{I}_d$ denotes the $d \times d$ identity matrix. τ controls the degree of dependence between the tasks on different devices. If $\tau = 0$, then the data on all devices is distributed according to parameter θ, i.e., the tasks are the same, and if $\tau \to \infty$, the tasks on different devices become completely unrelated.

We first derive optimal estimators $\{w_k\}_{k\in[K]}$ for each device w_k given observations $\{X_i, y_i\}_{i\in[K]}$.

Lemma 4. *Assume that we have*

$$y = Xw + z \tag{11}$$

where $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n\times d}$, and $w \in \mathbb{R}^d$, and $z \in \mathbb{R}^n$. Further assume that $z \sim \mathcal{N}(0, \sigma^2\mathbf{I}_d)$ and w follows the non-informative uniform prior on $\mathbb{R}^d$. Let

$$\widehat{w} = (X^TX)^{-1}X^Ty. \tag{12}$$

Then, we have $\widehat{w}$ follows a multi-variate normal distribution as follows:

$$\widehat{w} \sim \mathcal{N}\left((X^T X)^{-1} X^T y, \sigma^2 (X^T X)^{-1}\right). \tag{13}$$

Lemma 5. *Let*

$$\widehat{w}_i := (X_i^T X_i)^{-1} X_i^T y_i. \tag{14}$$

Let

$$\Sigma_i := \sigma^2 (X_i^T X_i)^{-1} + \tau^2 \mathbf{I}_d. \tag{15}$$

Further, let

$$\Sigma_\theta^{\setminus k} := \left(\sum_{i \in [K], i \neq k} \Sigma_i^{-1} \right)^{-1}. \tag{16}$$

Further let

$$\mu_\theta^{\setminus k} := \Sigma_\theta^{\setminus k} \sum_{i \in [K], i \neq k} \Sigma_i^{-1} \widehat{w}_i \tag{17}$$

Then, conditioned on $\{X_i, y_i\}_{i \in [K], i \neq k}$, we can write θ as

$$\theta = \mu_\theta^{\setminus k} + \eta,$$

where η is $\mathcal{N}(0, \Sigma_\theta^{\setminus k})$ which is independent of $\{X_i, y_i\}_{i \in [K], i \neq k}$.

Proof. From Lemma 4, we know $\widehat{w}_i$ is a noisy observation of the underlying w_i with additive covariance $\sigma^2 (X_i^T X_i)^{-1}$. For $\{w_k\}_{k \in [K]}$ defined in our setup, $\widehat{w}_i$ is a noisy observation of θ with additive zero mean and covariance $\Sigma_i := \tau^2 \mathbf{I}_d + \sigma^2 (X_i^T X_i)^{-1}$. The proof completes by applying Lemma 2 to $\{\widehat{w}_i\}_{i \in [K], i \neq k}$. □

Lemma 6. *Let*

$$\Sigma_{w_k}^{\setminus k} := \Sigma_\theta^{\setminus k} + \tau^2 \mathbf{I}_d. \tag{18}$$

Further, let

$$\Sigma_{w_k} := \left((\Sigma_{w_k}^{\setminus k})^{-1} + (\Sigma_k - \tau^2 \mathbf{I}_d)^{-1} \right)^{-1}. \tag{19}$$

Conditioned on $\{X_i, y_i\}_{i \in [K]}$, we have

$$w_k = \Sigma_{w_k} (\Sigma_k - \tau^2 \mathbf{I}_d)^{-1} \widehat{w}_k + \Sigma_{w_k} (\Sigma_{w_k}^{\setminus k})^{-1} \mu_\theta^{\setminus k} + \zeta_k, \tag{20}$$

where $\zeta_k \sim \mathcal{N}(0, \Sigma_{w_k})$.

Proof. $\widehat{w}_k$ is a noisy observation of w_k with additive noise with zero mean and covariance $\sigma^2 (X_k^T X_k)^{-1}$ (which is $\Sigma_k - \tau^2 \mathbf{I}_d$). From Lemma 5, we know conditioned on $\{X_i, y_i\}_{i \in [K], i \neq k}$, $\mu_\theta^{\setminus k}$ is a noisy observation of θ with covariance $\Sigma_\theta^{\setminus k}$. Hence, with respect to w_k, the covariance is $\Sigma_\theta^{\setminus k} + \tau^2 \mathbf{I}_d := \Sigma_{w_k}^{\setminus k}$. The conclusion follows by applying Lemma 2 to $\widehat{w}_k$ and $\mu_\theta^{\setminus k}$. □

Let the empirical loss function of the linear regression problem on device k be[2]

$$F_k(w) = \frac{1}{n} \left\| X_k w - y_k \right\|^2. \tag{21}$$

Then the estimator $\widehat{w}_k$ is $(X_k^T X_k)^{-1} X^T y_k$. Applying the previous lemmas, we obtain an optimal estimator w_k given all training samples from K devices (see (20)). w_k is Bayes optimal among all solutions that can be achieved by any learning method. Next, we examine the `Ditto` objective and its solution space parameterized by λ.

Let each device solve the following objective

$$\min_w h_k(w) = F_k(w) + \frac{\lambda}{2} \left\| w - w^* \right\|^2, \text{ s.t. } \quad w^* = \frac{1}{K} \arg\min_w \sum_{k=1}^K F_k(w). \tag{22}$$

[2]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.

The local empirical risk minimizer for each device k is

$$\widehat{w}_k(\lambda) = \left(\frac{1}{n}X_k^\top X_k + \lambda I\right)^{-1}\left(\frac{1}{n}X_k^\top Y_k + \lambda w^*\right) \tag{23}$$

$$= \left(\frac{1}{n}X_k^\top X_k + \lambda I\right)^{-1}\left(\left(\frac{1}{n}X_k^\top X_k\right)\widehat{w}_k + \lambda\sum_{k=1}^{K}(X^\top X)^{-1}X_k^\top X_k\widehat{w}_k\right) \tag{24}$$

We next prove that for any $k \in [K]$, $\widehat{w}_k(\lambda)$ with a specific λ can achieve the optimal w_k.

Theorem 1. *Assume for any $1 \le i \le K$, $X_i^T X_i = \beta \mathbf{I}_d$ for some constant β. Let λ^* be the optimal λ that minimizes the test performance on device k, i.e.,*

$$\lambda^* = \arg\min_{\lambda} E\left\{ F_k(\widehat{w}_k(\lambda))|\, \widehat{w}_k, \mu_\theta^{\setminus k}\right\}. \tag{25}$$

Then,

$$\lambda^* = \frac{\sigma^2}{n\tau^2}. \tag{26}$$

Proof. Notice that

$$\arg\min_{\lambda} E\left\{F_k(\widehat{w}_k(\lambda))|\widehat{w}_k, \mu_\theta^{\setminus k}\right\} = \arg\min_{\lambda} E\left\{\|X_k\widehat{w}_k(\lambda) - (X_k w_k + z_k)\|^2|\widehat{w}_k, \mu_\theta^{\setminus k}\right\} \tag{27}$$

$$= \arg\min_{\lambda} E\left\{\|X_k\left(\widehat{w}_k(\lambda) - w_k\right)\|^2|\widehat{w}_k, \mu_\theta^{\setminus k}\right\} \tag{28}$$

$$= \arg\min_{\lambda} E\left\{\|w_k - \widehat{w}_k(\lambda)\|^2\,|\widehat{w}_k, \mu_\theta^{\setminus k}\right\}. \tag{29}$$

Plug in $X_k^T X_k = \beta\mathbf{I}$ into (20) and (24) respectively, we have the optimal estimator w_k is

$$w_k = \left(\frac{K-1}{\frac{\sigma^2}{\beta} + K\tau^2} + \frac{\beta}{\sigma^2}\right)^{-1}\frac{\beta}{\sigma^2}\widehat{w}_k + \left(\frac{K-1}{\frac{\sigma^2}{\beta} + K\tau^2} + \frac{\beta}{\sigma^2}\right)^{-1}\frac{\beta}{\sigma^2 + K\tau^2\beta}\sum_{i\in[K], i\neq k}\widehat{w}_i + \zeta_k, \tag{30}$$

and $\widehat{w}_k(\lambda)$ is

$$\widehat{w}_k(\lambda) = \left(\frac{n}{\beta + n\lambda}\right)\left(\left(\frac{\beta}{n} + \frac{\lambda}{K}\right)\widehat{w}_k + \frac{\lambda}{K}\sum_{i\in[K], i\neq k}\widehat{w}_i\right). \tag{31}$$

Taking w_k and $\widehat{w}_k(\lambda)$ into

$$\lambda^* = \arg\min_{\lambda} E\left\{\|w_k - \widehat{w}_k(\lambda)\|_2^2\,|\mu_\theta^{\setminus k}, \widehat{w}_k\right\} \tag{32}$$

gives $\lambda^* = \frac{\sigma^2}{n\tau^2}$, as $\widehat{w}_k(\lambda^*)$ is the MMSE estimator of w_k given the observations. □

Remark 1. *We note that by using λ^* in `Ditto`, we not only achieve the most accurate solution for the objective, but also we achieve the most accurate solution of any possible federated linear regression algorithm in this problem, as `Ditto` with λ^* realizes the MMSE estimator for w_k.*

We have derived an optimal $\lambda^* = \frac{\sigma^2}{n\tau^2}$ for `Ditto` in terms of generalization. Recall that we define fairness as the variance of the performance across all devices (Hashimoto et al., 2018; Li et al., 2020d). Next, we prove that the same λ^* that minimizes the expected MSE also achieves the optimal fairness among all `Ditto` solutions.

Theorem 2. *Assume for any $1 \le i \le K$, $X_i^T X_i = \beta \mathbf{I}_d$ for some constant β. Among all possible solutions `Ditto` parameterized by λ, λ^* results in the most fair performance across all devices when there are no adversaries, i.e., it minimizes the variance of test performance (test loss) across all devices.*

Proof. Denote the variance of test performance (loss) across K devices as $\text{var}_K\left\{\|X_k\widehat{w}_k(\lambda)-y_k\|_2^2\right\}$. Let

$$\widehat{E}_k\{a_k\} := \frac{1}{K}\sum_{k\in[K]} a_k. \tag{33}$$

Then

$$\begin{aligned}
\arg\min_\lambda \text{var}_K\left\{\|X_k\widehat{w}_k(\lambda)-y_k\|_2^2\right\} &= \arg\min_\lambda \text{var}_K\left\{\|X_k\widehat{w}_k(\lambda)-(X_kw_k+z_k)\|_2^2\right\} &(34)\\
&= \arg\min_\lambda \text{var}_K\left\{\|X_k(\widehat{w}_k(\lambda)-w_k)\|_2^2\right\} &(35)\\
&= \arg\min_\lambda \text{var}_K\left\{\|\widehat{w}_k(\lambda)-w_k\|_2^2\right\} &(36)\\
&= \arg\min_\lambda \widehat{E}_K\left\{\left(\|w_k-\widehat{w}_k\|_2^2\right)^2\right\} - \left(\widehat{E}_K\left\{\|w_k-\widehat{w}_k(\lambda)\|_2^2\right\}\right)^2. &(37)
\end{aligned}$$

Note that

$$w_k-\widehat{w}_k(\lambda) = \zeta + a_k, \tag{38}$$

where

$$a_k = \widehat{w}_k(\lambda^*)-\widehat{w}_k(\lambda), \tag{39}$$

and $\lambda^* = \frac{\sigma^2}{n\tau^2}$.

We have

$$\begin{aligned}
&\widehat{E}_K\left\{\left(\|w_k-\widehat{w}_k\|_2^2\right)^2\right\} - \left(\widehat{E}_K\left\{\|w_k-\widehat{w}_k(\lambda)\|_2^2\right\}\right)^2 &(40)\\
&= \widehat{E}_K\left\{\left(\sum_i^d (w_{ki}-\widehat{w}_k(\lambda)_i)^2\right)^2\right\} - \left(\widehat{E}_K\left\{\sum_i^d (w_{ki}-\widehat{w}_k(\lambda)_i)^2\right\}\right)^2 &(41)\\
&= \widehat{E}_K\left\{\left(\sum_i^d (\zeta_i+a_{ki})^2\right)^2\right\} - \left(\widehat{E}_K\left\{\sum_i^d (\zeta_i+a_{ki})^2\right\}\right)^2, &(42)
\end{aligned}$$

where w_{ki}, $\widehat{w}_k(\lambda)_i$, ζ_i, and a_{ki} denotes the i-th dimension of w_k, $\widehat{w}_k(\lambda)$, ζ, and a_k and d is the model dimension.

We next expand the variance by decomposing it into two parts. We note

$$\begin{aligned}
&\widehat{E}_K\left\{\left(\sum_i^d (\zeta_i+a_{ki})^2\right)^2\right\} - \left(\widehat{E}_K\left\{\sum_i^d (\zeta_i+a_{ki})^2\right\}\right)^2 &(43)\\
&= \sum_i^d \widehat{E}_k\left\{(\zeta_i+a_{ki})^4\right\} - \sum_i^d\left(\widehat{E}_K\left\{(\zeta_i+a_{ki})^2\right\}\right)^2 &(44)\\
&+ 2\sum_{i,j\in[d],i\neq j} \widehat{E}_K\left\{(\zeta_i+a_{ki})^2\left(\zeta_j+a_{kj}\right)^2\right\} - 2\sum_{i,j\in[d],i\neq j} \widehat{E}_K\left\{(\zeta_i+a_{ki})^2\right\}\widehat{E}_K\left\{\left(\zeta_j+a_{kj}\right)^2\right\}. &(45)
\end{aligned}$$

For any $i \in [d]$, we have

$$E\left\{ \widehat{E}_K\left\{(\zeta_i + a_{ki})^4\right\} - \left(\widehat{E}_K\left\{(\zeta_i + a_{ki})^2\right\}\right)^2 \middle| \mu_\theta^{\setminus k}, \widehat{w}_k \right\} \tag{46}$$

$$= E\left\{ \widehat{E}_K\left\{\zeta_i^4 + 6\zeta_i^2 a_{ki}^2 + a_{ki}^4\right\} - \left(\widehat{E}_K\left\{\zeta_i^2 + a_{ki}^2\right\}\right)^2 \middle| \mu_\theta^{\setminus k}, \widehat{w}_k \right\} \tag{47}$$

$$= E\left\{ \widehat{E}_K\left\{\zeta_i^4 + 6\zeta_i^2 a_{ki}^2 + a_{ki}^4\right\} - \left(\widehat{E}_K\left\{\zeta_i^2\right\}\right)^2 - 2\widehat{E}_K\left\{\zeta_i^2\right\}\widehat{E}_K\left\{a_{ki}^2\right\} - \left(\widehat{E}_K\left\{a_{ki}^2\right\}\right)^2 \middle| \mu_\theta^{\setminus k}, \widehat{w}_k \right\} \tag{48}$$

$$= 3\sigma_w^4 + 6\sigma_w^2\widehat{E}_K\left\{a_{ki}^2\right\} + \widehat{E}_K\left\{a_{ki}^4\right\} - \sigma_w^4 - 2\sigma_w^2\widehat{E}_K\left\{a_{ki}^2\right\} - \left(\widehat{E}_K\left\{a_{ki}^2\right\}\right)^2 \tag{49}$$

$$= 2\sigma_w^4 + 4\sigma_w^2\widehat{E}_K\left\{a_{ki}^2\right\} + \widehat{E}_K\left\{a_{ki}^4\right\} - \left(\widehat{E}_K\left\{a_{ki}^2\right\}\right)^2, \tag{50}$$

where σ_w is the i-th diagonal of Σ_{w_k} which is the same across all k's and all dimensions, and we have used the fact that we can swap expectations, and $E\{\zeta_i^4\} = 3\sigma_w^4$, given that ζ_i is Gaussian distributed and Σ_{w_k} is a diagonal matrix.

For any $i, j \in [d], i \neq j$, we have

$$E\left\{ \widehat{E}_K\left(\zeta_i + a_{ki}\right)^2\left(\zeta_j + a_{kj}\right)^2 \middle| \mu_\theta^{\setminus k}, \widehat{w}_k \right\} - E\left\{ \widehat{E}_K\left(\zeta_i + a_{ki}\right)^2 \widehat{E}_K\left(\zeta_j + a_{kj}\right)^2 \middle| \mu_\theta^{\setminus k}, \widehat{w}_k \right\} \tag{51}$$

$$= \widehat{E}_k\{a_{ki}^2 a_{kj}^2\} - \widehat{E}_k\{a_{ki}^2\}\widehat{E}_k\{a_{kj}^2\}, \tag{52}$$

where we have used the fact that Σ_{w_k} is a diagonal matrix.

Plugging (50) and (52) into (44) and (45) yields

$$E\left\{ \text{var}_K\left\{\|\widehat{w}_k(\lambda) - w_k\|_2^2\right\} \middle| \mu_\theta^{\setminus k}, \widehat{w}_k \right\} \tag{53}$$

$$= 2d\sigma_w^4 + \sum_i 4\sigma_w^2\widehat{E}_k\{a_{ki}^2\} + \sum_i \widehat{E}_k\{a_{ki}^4\} - \sum_i\left(\widehat{E}_k\{a_{ki}^2\}\right)^2 + 2\sum_{i\neq j}\left(\widehat{E}_k\{a_{ki}^2 a_{kj}^2\} - \widehat{E}_k\{a_{ki}^2\}\widehat{E}_k\{a_{kj}^2\}\right) \tag{54}$$

$$= 2d\sigma_w^4 + \sum_i 4\sigma_w^2\widehat{E}_k\{a_{ki}^2\} + \sum_i \widehat{E}_k\{a_{ki}^4\} + 2\sum_{i\neq j}\widehat{E}_k\{a_{ki}^2 a_{kj}^2\} - (\sum_i\left(\mathbb{E}_k\{a_{ki}^2\}\right)^2 + 2\sum_{i\neq j}\widehat{E}_k\{a_{ki}^2\}\widehat{E}_k\{a_{kj}^2\})) \tag{55}$$

$$= 2d\sigma_w^4 + \sum_i 4\sigma_w^2\widehat{E}_k\{a_{ki}^2\} + \widehat{E}_k\{(\sum_i a_{ki}^2)^2\} - (\sum_i \widehat{E}_k\{a_{ki}^2\})^2 \tag{56}$$

$$= 2d\sigma_w^4 + \sum_i 4\sigma_w^2\widehat{E}_k\{a_{ki}^2\} + \frac{1}{K}\sum_k(\sum_i a_{ki}^2)^2 - (\frac{1}{K}\sum_k\sum_i a_{ki}^2)^2 \geq 2d\sigma_w^2, \tag{57}$$

where setting $\{a_{ki}\}_{1\leq k\leq K, 1\leq i\leq d} = 0$ achieves the minimum. □

Observations. From the optimal $\lambda^* = \frac{\sigma^2}{n\tau^2}$ for mean test accuracy and variance of the test accuracy, we have the following observations.

- Test error and variance can be jointly minimized with one λ.
- As $n \to \infty$, $\lambda^* \to 0$, i.e., when each local device has an infinite number of samples, there is no need for federated learning, and training local models is optimal in terms of generalization and fairness.
- As $\tau \to \infty$, $\lambda^* \to 0$, i.e., if the data on different devices (the tasks) are unrelated, then training local models is optimal; On the other hand, as $\tau \to 0$, $\lambda^* \to \infty$, i.e., if the data across all devices are identically distributed, or equivalently if the tasks are the same, then training a global model is the best we can achieve.

So far we have proved that the same λ^* achieves the best performance (expected mean square error) for any device k *and* fairness (variance of mean square error) without considering adversaries. In Section B.2.2 below, we analyze the benefits of Ditto for fairness and robustness in the presence of adversaries.

B.2.2 With Adversaries: Ditto for Accuracy, Fairness, and Robustness

We look at a specific type of label poisoning attack defined in our threat model (Definition 1). Let K_a and $K_b \geq 1$ denote the number of malicious and benign devices, respectively, such that $K = K_a + K_b$.

Definition 3. *We say that a device k is a benign device if $w_k \sim \theta + \mathcal{N}(0, \tau^2 \mathbf{I}_d)$; and we say a device k is a malicious device (or an adversary) if $w_k \sim \theta + \mathcal{N}(0, \tau_a^2 \mathbf{I}_d)$ where $\tau_a > \tau$.*

As mentioned in Definition 2 and 1, in the presence of adversaries, we measure fairness as the performance variance on *benign* devices, and robustness as the average performance across *benign* devices. We next characterize the benefits of Ditto under such metrics.

Lemma 7. *Let w_k be the underlying model parameter of a benign device k. Let*

$$\widehat{w}_i := (X_i^T X_i)^{-1} X_i^T y_i, \; i \in [K]. \tag{58}$$

Let

$$\Sigma_w^{\setminus k} = \frac{1}{(K-1)^2} \left(\sum_{i \in [K_b], i \neq k} \left(\sigma^2 (X_i^T X_i)^{-1} + \tau^2 \mathbf{I}_d \right) + \sum_{i \in [K_a], i \neq k} \left(\sigma^2 (X_i^T X_i)^{-1} + \tau_a^2 \mathbf{I}_d \right) \right), \tag{59}$$

and

$$\Sigma_{w,a}^{-1} = (\sigma^2 (X_k^T X_k)^{-1})^{-1} + (\Sigma_w^{\setminus k} + \tau^2 \mathbf{I}_d)^{-1}. \tag{60}$$

Conditioned on observations $\widehat{w}_k$ and $\widehat{w}^{K \setminus k} := \frac{1}{K-1} \sum_{i \neq k, i \in [K]} \widehat{w}_i$, we have

$$w_k = \Sigma_{w,a} (\sigma^2 (X_k^T X_k)^{-1})^{-1} \widehat{w}_k + \Sigma_{w,a} (\Sigma_w^{\setminus k} + \tau^2 \mathbf{I}_d)^{-1} \widehat{w}^{K \setminus k} + \zeta_k, \tag{61}$$

where $\zeta_k \sim \mathcal{N}(0, \Sigma_{w,a})$.

Proof. For malicious devices $i \in [K_a]$ and $i \neq k$, the additive covariance of w_i with respect to θ is $\sigma^2 (X_i^T X_i)^{-1} + \tau_a^2 \mathbf{I}_d$. For benign devices $i \in [K_b]$ and $i \neq K$, the covariance is $\sigma^2 (X_i^T X_i)^{-1} + \tau^2 \mathbf{I}_d$. Therefore, the covariance of $\widehat{w}^{K \setminus k}$ is $\Sigma_w^{\setminus k}$. Hence given $\widehat{w}^{K \setminus k}$, w_k is Gaussian with covariance $\Sigma_w^{\setminus k} + \tau^2 \mathbf{I}_d$. $\widehat{w}^{K \setminus k}$ can be viewed as a noisy observation of w_k with covariance $\Sigma_w^{\setminus k} + \tau^2 \mathbf{I}_d$. $\widehat{w}_k$ is a noisy observation of w_k with covariance $\sigma^2 (X_k^T X_k)^{-1}$. The proof follows by applying Lemma 2 to $\widehat{w}_k$ and $\widehat{w}^{K \setminus k}$. □

Theorem 3. *Assume for any $1 \leq i \leq K$, $X_i^T X_I = \beta \mathbf{I}_d$ for some constant β. Let k be a benign device. Let λ_a^* be the optimal λ that minimizes the test performance on device k, i.e.,*

$$\lambda^* = \arg\min_{\lambda} E \left\{ F_k(\widehat{w}_k(\lambda)) | \, \widehat{w}_k, \widehat{w}^{K \setminus k} \right\}. \tag{62}$$

Then,

$$\lambda_a^* = \frac{\sigma^2}{n} \frac{K}{K\tau^2 + \frac{K_a}{K-1}(\tau_a^2 - \tau^2)}. \tag{63}$$

Proof. We obtain λ_a^* following the proof of Theorem 1. □

Theorem 4. *Among all Ditto solutions parameterized by λ, λ_a^* results in the most fair performance across all benign devices, i.e., it minimizes the variance of test performance (test mean square error) on benign devices.*

Proof. Similarly, we look at the variance of the test loss across benign devices:

$$\arg\min_{\lambda} E\left\{\text{var}_{K_b}\left\{\|X_k\widehat{w}_k(\lambda) - y_k\|_2^2\right\}\right\} = \arg\min_{\lambda} E\left\{\text{var}_{K_b}\left\{\|w_k(\lambda) - w_k\|_2^2\right\}\right\} \tag{64}$$

$$= \arg\min_{\lambda} \widehat{E}_{K_b}\left\{\left(\|w_k - \widehat{w}_k\|_2^2\right)^2\right\} - \left(\widehat{E}_{K_b}\left\{\|w_k - \widehat{w}_k(\lambda)\|_2^2\right\}\right)^2. \tag{65}$$

The rest of the proof is the same as the proof of Theorem 2, except that we set $a_k = \widehat{w}_k(\lambda) - \widehat{w}_k(\lambda_a^*)$. □

Remark 2. *For any benign device k, the solution we obtain by solving* `Ditto` *with λ_a^* is the most robust solution one could obtain among any federated point estimation method given observations $\widehat{w}_k$ and $\widehat{w}^{K\setminus k}$. λ_a^* also results in a most fair model in the solution space of* `Ditto` *parameterized by λ.*

Lemma 8. *The expected test error minimized at λ_a^* is $d\sigma_{w,a}^2$; and the variance of the test loss minimized at λ_a^* is $2d\sigma_{w,a}^4$, where $\sigma_{w,a}$ denotes the diagonal element of $\Sigma_{w,a}$.*

Proof. For the expected test performance, we note that

$$E\left\{\|w_k - \widehat{w}_k(\lambda_a^*)\|^2 \middle| \widehat{w}^{K\setminus k}, \widehat{w}_k\right\} = E[\|\text{diag}(\Sigma_{w,k})\|^2] = d\sigma_{w,k}^2. \tag{66}$$

For variance, as $a_k = 0$ if $\lambda = \lambda_a^*$, from (57), we get

$$\text{var}_{K_b}\left\{\|w_k - \widehat{w}_k(\lambda_a^*)\|^2\right\} = 2d\sigma_{w,k}^4. \tag{67}$$

□

Observations. From λ_a^*, we have the following interesting observations.

- Mean test error on benign devices (robustness) and variance of the performance across benign devices (fairness) can still be minimized with the same λ_a in the presence of adversaries.
- As $\tau_a \to \infty$, $\lambda_a^* \to 0$, i.e., training local models is optimal in terms of robustness and fairness when adversary's task may be arbitrarily far from the the task in the benign devices.
- As $\tau \to 0$, if $\tau_a > 0$, $\lambda_a^* < \infty$, which means that learning a global model is *not* optimal even with homogeneous data in the presence of adversaries.
- λ_a^* is a decreasing function of the number (K_a) and the capability (τ_a) of the corrupted devices. In other words, as the attacks become more adversarial, we need more personalization.
- The smallest test error is $\sigma_{w,a}^2$, and the optimal variance is $2\sigma_{w,a}^4$, which are both increasing with K_a (number of adversarial devices) or τ_a (the power of adversary) by inspecting (59) and (60). This reveals a fundamental tradeoff between fairness and robustness.

Discussion. Through our analysis, we prove that `Ditto` with an appropriate λ is more accurate, robust, and fair compared with training global or local models on the problem described in B.2. We provide closed-form solutions for λ^* across different settings (with and without adversaries), and show that `Ditto` can achieve fairness and robustness jointly. In the future, we plan to generalize the current theoretical framework to more general models. In the next section, we present a special case of the current analysis, a federated point estimation problem, which is also studied in Section 3.2 as a motivating example.

B.3 Optimality of Ditto for Federated Point Estimation

We consider the one-dimensional federated point estimation problem, which is a special case of linear regression. Similarly, Let θ be drawn from the non-informative prior on $\mathbb{R}$. We assume that K devices have their data distributed with parameters $\{w_k\}_{k\in[K]}$.

$$w_k = \theta + \zeta_k, \tag{68}$$

where $\zeta_k \sim \mathcal{N}(0, \tau^2)$ are IID.

Let each device have n data points denoted by $\mathbf{x}_k = \{x_{k,1}, \ldots, x_{k,n}\}$, such that

$$x_{k,i} = w_k + z_{k,i}, \tag{69}$$

where $z_{k,i} \sim \mathcal{N}(0, \sigma^2)$ and are IID.

Assume that

$$F_k(w) = \frac{1}{2}\left(w - \frac{1}{n}\sum_{i\in[n]} x_{k,i}\right)^2, \tag{70}$$

and denote by $\widehat{w}_k$ the minimizer of the empirical loss F_k. It is clear that

$$\widehat{w}_k = \frac{1}{n}\sum_{i\in[n]} x_{k,i}. \tag{71}$$

Further, let

$$w^* := \arg\min_w \left\{\frac{1}{K}\sum_{k\in[K]} F_k(w)\right\}. \tag{72}$$

It is straightforward calculation to verify that

$$w^* = \frac{1}{nK}\sum_{i\in[n]}\sum_{k\in[K]} x_{k,i} = \frac{1}{K}\sum_{k\in[K]} \widehat{w}_k. \tag{73}$$

Lemma 9. *Denote by $\widehat{w}_k(\lambda)$ the minimizer of g_k. Then,*

$$\widehat{w}_k(\lambda) = \frac{\lambda}{1+\lambda} w^* + \frac{1}{1+\lambda}\widehat{w}_k \tag{74}$$

$$= \frac{\lambda}{(1+\lambda)K}\sum_{j\neq k}\widehat{w}_j + \frac{K+\lambda}{(1+\lambda)K}\widehat{w}_k. \tag{75}$$

Let

$$\sigma_n^2 := \frac{\sigma^2}{n}, \tag{76}$$

and

$$\widehat{w}^{K\setminus k} := \frac{1}{K-1}\sum_{j\neq k}\widehat{w}_j. \tag{77}$$

Lemma 10. *Given observations $\widehat{w}^{K\setminus k}$ and $\widehat{w}_k$, w_k is Gaussian distributed and given by*

$$w_k = \frac{\sigma_w^2}{\sigma_n^2}\widehat{w}_k + \frac{(K-1)\sigma_w^2}{K\tau^2+\sigma_n^2}\widehat{w}^{K\setminus k} + \xi, \tag{78}$$

where

$$\frac{1}{\sigma_w^2} = \frac{1}{\sigma_n^2} + \frac{K-1}{K\tau^2+\sigma_n^2}, \tag{79}$$

and

$$\xi \sim \mathcal{N}\left(0, \sigma_w^2\right). \tag{80}$$

Proof. The proof follows by setting $X_k = \mathbf{1}_{n\times 1}$ ($k \in [K]$) in Lemma 6. □

Theorem 5. *Let λ^* be the optimal λ that minimizes the test performance, i.e.,*

$$\lambda^* = \arg\min_\lambda E\left\{(w_k - \widehat{w}_k(\lambda))^2 \middle| \widehat{w}^{K\setminus k}, \widehat{w}_k\right\}. \tag{81}$$

Then,

$$\lambda^* = \frac{\sigma_n^2}{\tau^2} = \frac{\sigma^2}{n\tau^2}. \tag{82}$$

Proof. The proof follows by setting $X_k = \mathbf{1}_{n\times 1}$ ($k \in [K]$) in Theorem 1. □

Theorem 6. *Among all* `Ditto`*'s solutions, λ^* results in the most fair performance across all devices when there are no adversaries, i.e., it minimizes the variance of test performance (test mean square error).*

Proof. The proof follows by setting $X_k = \mathbf{1}_{n\times 1}$ ($k \in [K]$) in Theorem 2. □

Similarly, the adversarial case presented below (including setups, lemmas, and theorems) is also a special case of the adversarial scenarios for linear regression.

Let K_a and $K_b \geq 1$ denote the number of adversarial and benign devices, respectively, such that $K = K_a + K_b$.

Definition 4. *We say that a device k is a benign device if $w_k \sim \theta + \mathcal{N}(0, \tau^2)$; and we say a device k is a malicious device (or an adversary) if $w_k \sim \theta + \mathcal{N}(0, \tau_a^2)$ where $\tau_a \geq \tau$.*

Lemma 11. *Let w_k be the parameter associated with a benign device. Given observations $\widehat{w}^{K\setminus k} := \frac{1}{K-1}\sum_{j\neq k}\widehat{w}_j$ and $\widehat{w}_k$, w_k is Gaussian distributed and given by*

$$w_k = \frac{\sigma_{w,a}^2}{\sigma_n^2}\widehat{w}_k + \frac{(K-1)\sigma_{w,a}^2}{K\tau^2 + \sigma_n^2 + \frac{K_a}{K-1}(\tau_a^2 - \tau^2)}\widehat{w}^{K\setminus k} + \xi_a, \tag{83}$$

where

$$\frac{1}{\sigma_{w,a}^2} = \frac{1}{\sigma_n^2} + \frac{K-1}{K\tau^2 + \sigma_n^2 + \frac{K_a}{K-1}(\tau_a^2 - \tau^2)}, \tag{84}$$

and

$$\xi_a \sim \mathcal{N}\left(0, \sigma_{w,a}^2\right). \tag{85}$$

Proof. The proof follows by setting $X_k = \mathbf{1}_{n\times 1}$ ($k \in [K]$) in Lemma 7. □

Theorem 7. *Let w_k be a benign device. Let λ_a^* be the optimal λ that minimizes the test performance, i.e.,*

$$\lambda_a^* = \arg\min_{\lambda} E\left\{ (w_k - \widehat{w}_k(\lambda))^2 \middle| \widehat{w}^{K\setminus k}, \widehat{w}_k \right\}. \tag{86}$$

Then,

$$\lambda_a^* = \frac{\sigma^2}{n}\frac{K}{K\tau^2 + \frac{K_a}{K-1}(\tau_a^2 - \tau^2)}. \tag{87}$$

Proof. The proof follows by setting $X_k = \mathbf{1}_{n\times 1}$ ($k \in [K]$) in Theorem 3. □

Theorem 8. *Among all solutions of Objective* (`Ditto`) *parameterized by λ, λ_a^* results in the most fair performance across all benign devices, i.e., it minimizes the variance of test performance (test mean square error) on benign devices.*

Proof. The proof follows by setting $X_k = \mathbf{1}_{n\times 1}$ ($k \in [K]$) in Theorem 4. □

Lemma 12. *The expected test error minimized at λ_a^* is $\sigma_{w,a}^2$; and the variance of the test performance minimized at λ_a^* is $2\sigma_{w,a}^4$.*

Proof. The proof follows by setting $X_k = \mathbf{1}_{n\times 1}$ ($k \in [K]$) in Lemma 8. □

C Other Personalization Schemes and Regularizers

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 Appendix G.2. 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 3.2), as we can easily reason about `Ditto` relative to the global ($\lambda \rightarrow \infty$) vs. local ($\lambda \rightarrow 0$) baselines; empirically, in terms of accuracy, fairness, and robustness (Section 4); and practically, in terms of the modularity it affords our corresponding solver (Section 3.1).

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 (Appendix G.2), 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.

D ALGORITHMS

In this section, we first present the general `Ditto` solver in Algorithm 1 below. The personalization add-on is highlighted in red. In our experiments (all except Table 5), 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.

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 the current global model 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 models)*, w^T *(global model)*

Algorithm 2: `Ditto` for Personalized FL in the case of $G(\cdot)$ being FedAvg (McMahan et al., 2017)

1 **Input:** $K, T, s, \lambda, \eta_g, \eta_l, w^0, p_k, \{v_k^0\}_{k\in[K]}$
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 Sets w_k^t to w^t and updates w_k^t for r local iterations on F_k:
$$w_k^t = w_k^t - \eta_g \nabla F_k(w_k^t)$$
6 Updates v_k for s local iterations:
$$v_k = v_k - \eta_l(\nabla F_k(v_k) + \lambda(v_k - w^t)$$
7 Sends $\Delta_k^t := w_k^t - w^t$ back
8 Server updating w^{t+1} as
$$w^{t+1} \leftarrow w^t + \frac{1}{|S_t|}\sum_{k\in S_t} \Delta_k^t$$
9 **return** $\{v_k\}_{k\in[K]}$ *(personalized)*, w^T *(global)*

E CONVERGENCE ANALYSIS

To analyze the convergence behavior of Algorithm 1 and 2, we first state a list of common assumptions below.

- For $k \in [K]$, F_k is μ-strongly convex and L-smooth.
- For $k \in [K]$, the variance of stochastic gradients of F_k within each device is bounded:

$$\mathbb{E}[\|\nabla F_k(w^t, \xi^t) - \nabla F_k(w^t)\|^2] \leq \sigma^2, \tag{88}$$

  where ξ^t denotes mini-batch data.
- The expectation of stochastic gradients is uniformly bounded at all devices and all iterations, i.e.,

$$\mathbb{E}[\|\nabla F_k(w^t, \xi^t)\|^2] \leq G_1^2. \tag{89}$$

Let w^* be defined as

$$w^* := \min_w G(F_1(w), \dots F_K(w)) \tag{90}$$

i.e., w^* is the optimal global model for $G(\cdot)$.

We introduce an additional assumption on the distance between personalized models and the optimal global model:

- The expectation of the distance between personalized models and the optimal global model is bounded at all iterations, i.e., for any v_k and $k \in [K]$,

$$\mathbb{E}[\|v_k - w^*\|^2] \leq M^2. \tag{91}$$

Further let

$$v_k^* = \arg\min_v h_k(v; w^*), \tag{92}$$

i.e., v_k^* is the personalized model for device k. We first characterize the progress of updating personalized models for one step under a *general* $G(\cdot)$.

Lemma 13 (Progress of one step)**.** *Under assumptions above, let device k get selected with probability p_k at each communication round, with decaying local step-size $\frac{2}{(t+1)(\mu+\lambda)p_k}$, at each communication round t, we have*

$$\begin{aligned}\mathbb{E}[\|v_k^{t+1} - v_k^*\|^2] \leq & \left(1 - \frac{2}{t+1}\right)\mathbb{E}[\|v^t - v^*\|^2] + \frac{4(G_1 + \lambda M)^2}{(t+1)^2(\mu+\lambda)^2 p_k^2} \\ & + \frac{4\lambda^2}{(t+1)^2(\mu+\lambda)^2 p_k^2}\mathbb{E}[\|w^t - w^*\|^2] + \frac{8\lambda(G_1+\lambda M)}{(t+1)^2(\mu+\lambda)^2 p_k^2}\sqrt{\mathbb{E}[\|w^t - w^*\|^2]} \\ & + \frac{4\lambda}{(t+1)(\mu+\lambda)p_k}\sqrt{\mathbb{E}[\|v_k^t - v_k^*\|^2]\mathbb{E}[\|w^t - w^*\|^2]}. \end{aligned} \tag{93}$$

Proof. Denote $g(v_k^t; w^t)$ as the stochastic gradient of $h_k(v_k^t; w^t)$. Let I_t indicate if device k is selected at the t-th round, and $\mathbb{E}[I_t] = p_k$.

$$\mathbb{E}[\|v_k^{t+1} - v_k^*\|^2] = \mathbb{E}[\|v_k^t - \eta I_t g(v_k^t; w^t) - v_k^*\|^2] \tag{94}$$

$$= \mathbb{E}[\|v_k^t - v_k^*\|^2] + \eta^2\mathbb{E}[\|I_t g(v_k^t; w^t)\|^2] + 2\eta\mathbb{E}\langle I_t g(v_k^t; w^t), v_k^* - v_k^t\rangle \tag{95}$$

$$\leq (1 - (\mu+\lambda)\eta p_k)\mathbb{E}[\|v_k^t - v_k^*\|^2] + \eta^2\mathbb{E}[\|g(v_k^t; w^t)\|^2] + 2\eta p_k\mathbb{E}[h(v_k^*; w^t) - h(v_k^t; w^t)] \tag{96}$$

$$\begin{aligned}\leq & (1 - (\mu+\lambda)\eta p_k)\mathbb{E}[\|v_k^t - v_k^*\|^2] \\ & + \eta^2\mathbb{E}[\|g(v_k^t; w^*)\|^2] + \eta^2\lambda^2\mathbb{E}[\|w^t - w^*\|^2] + 2\eta^2\lambda\mathbb{E}[\|g(v_k^t; w^*)\|\|w^t - w^*\|] \\ & + 2\eta p_k(h(v_k^*; w^*) - \mathbb{E}[h(v_k^t; w^*)]) + 2\eta p_k\lambda\mathbb{E}[\|v_k^t - v_k^*\|\|w^t - w^*\|]. \end{aligned} \tag{97}$$

Further, note

$$\mathbb{E}[\|g(v_k^t; w^*)\|^2] = \mathbb{E}[\|\nabla F_k(v_k^t) + \lambda(v_k^t - w^*)\|^2] \leq \mathbb{E}[\|\nabla F_k(v_k^t)\|^2] + \lambda^2 \mathbb{E}[\|v_k^t - w^*\|^2] + 2\lambda \mathbb{E}[\|\nabla F_k(v_k^t)\| \|v_k^t - w^*\|] \tag{98}$$

$$\leq G_1^2 + \lambda^2 M^2 + 2\lambda G_1 M. \tag{99}$$

Plug it into (97),

$$\mathbb{E}[\|v_k^{t+1} - v_k^*\|^2] \leq (1 - (\mu + \lambda)\eta p_k)\mathbb{E}[\|v_k^t - v_k^*\|^2] + \eta^2(G_1^2 + \lambda^2 M^2 + 2\lambda G_1 M) + \eta^2 \lambda^2 \mathbb{E}[\|w^t - w^*\|^2] + 2\eta^2 \lambda (G_1 + \lambda M)\sqrt{\mathbb{E}[\|w^t - w^*\|^2]} + 2\eta p_k \lambda \sqrt{\mathbb{E}[\|v_k^t - v_k^*\|^2]\mathbb{E}[\|w^t - w^*\|^2]}. \tag{100}$$

where the last step is due to $E[XY] \leq \sqrt{E[X^2]E[Y^2]}$. The Lemma then holds by taking $\eta = \frac{2}{(t+1)(\mu+\lambda)p_k}$. □

Lemma 13 relates $\mathbb{E}[\|v_k^{t+1} - v_k^*\|^2]$ with $\mathbb{E}[\|v_k^t - v_k^*\|^2]$ and $\mathbb{E}[\|w_k^t - w^*\|^2]$. Based on this, we prove that personalized models can inherit the convergence rate of the global model w^t for any $G(\cdot)$.

Theorem 9 (Relations between convergence of global and personalized models)**.** *Under the assumptions above, if there exists $g(t)$ such that $\lim_{t\to\infty} g(t) = 0$, $\mathbb{E}[\|w^t - w^*\|^2] \leq g(t)$, and $\frac{g(t+1)}{g(t)} \geq 1 - g(t)$, then there exists $C < \infty$ such that for any device $k \in [K]$, $\mathbb{E}[\|v_k^t - v_k^*\|^2] \leq Cg(t)$ with a local learning rate $\eta = \frac{2g(t)}{(\mu+\lambda)p_k}$.*

Proof. We proceed the proof by induction. First, for any constant $C > \frac{\mathbb{E}[\|v_k^0 - v_k^*\|^2]}{g(0)}$, $\mathbb{E}[\|v_k^0 - v_k^*\|^2] \leq Cg(0)$. If $\mathbb{E}[\|v_k^t - v_k^*\|^2] \leq Cg(t)$ holds, then for $t+1$, from Lemma 13,

$$\mathbb{E}[\|v_{k+1}^t - v_k^*\|^2] \leq (1 - 2g(t))\, Cg(t) + g(t)^2 \frac{4}{p_k^2}\left(\frac{(G_1 + \lambda M)^2}{(\mu+\lambda)^2} + g(t) + \frac{2(G_1 + \lambda M)\sqrt{g(t)}}{\mu + \lambda}\right) + g(t)^2 \frac{4\lambda\sqrt{C}}{(\mu+\lambda)} \tag{101}$$

$$\leq (1 - 2g(t))\, Cg(t) + Cg(t)^2 \tag{102}$$

holds for some $C < \infty$. Hence,

$$\mathbb{E}[\|v_{k+1}^t - v_k^*\|^2] \leq (1 - 2g(t))\, Cg(t) + Cg(t)^2 \tag{103}$$

$$= (1 - g(t))\, Cg(t) \tag{104}$$

$$\leq Cg(t+1), \tag{105}$$

completing the proof. □

Using Theorem 9, we can directly plug in the convergence analyses in previous works 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, as described in Corollary 1 below.

Corollary 1 (Convergence of personalized models)**.** *Under the assumptions above, if the global objective $G(\cdot)$ is FedAvg, then under Algorithm 2, for $k \in [K]$,*

$$\mathbb{E}[\|v_k^t - v_k^*\|^2] = O(1/t). \tag{106}$$

Proof. From Li et al. (2020e) Theorem 2, we know the global model for FedAvg converges at a rate of $O(1/t)$, i.e.,

$$\mathbb{E}[\|w^t - w^*\|^2] \leq \frac{C'}{t+B}\|w^1 - w^*\|^2 \leq \frac{C}{t+1}, \tag{107}$$

where C, C', B are constants. Setting $g(t) = \frac{C}{t+1}$ in Theorem 9, it follows that $\mathbb{E}[\|v_k^t - v_k^*\|^2] = O(1/t)$. □

E.1 MODULARITY OF DITTO

From the Ditto objective and Algorithm 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 (Theorem 9).
- *Privacy:* Ditto communicates the same information over the network as typical FL solvers for the global objective, thus preserving privacy or communication benefits for the global objective and its respective solver.
- *Robustness:* Beyond the inherent robustness benefits of personalization, robust global methods can be used with Ditto to further improve performance (see Appendix G.5).

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).

F EXPERIMENTAL DETAILS

F.1 DATASETS AND MODELS

We summarize the datasets, corresponding models, and tasks in Table 1 below. We evaluate the performance of `Ditto` with both convex and non-convex models across a set of FL benchmarks. In our datasets, we have both image data (FEMNIST, CelebA, Fashion MNIST), and text data (StackOverflow).

Table 1: Summary of datasets.

Datasets	# Devices	Data Partitions	Models	Tasks
Vehicle (Duarte & Hu, 2004)[3]	23	natural (each device is a vehicle)	linear SVM	binary classification
FEMNIST (Cohen et al., 2017)	205	natural (each device is a writer)	CNN	62-class classification
CelebA (Liu et al., 2015)	515	natural (each device is a celebrity)	CNN	binary classification
Fashion MNIST (Xiao et al., 2017)	500	synthetic (assign 5 classes to each device)	CNN	10-class classification
StackOverflow (TFF)[4]	400	natural (each device is a user)	logistic regression	500-class tag prediction
FEMNIST (skewed) (Cohen et al., 2017)	100	synthetic (assign 5 classes to each device)	CNN	62-class classification

FEMNIST is Federated EMNIST, which is EMNIST (Cohen et al., 2017) partitioned by the writers of digits/characters created by a previous federated learning benchmark (Caldas et al., 2018). We have two versions of FEMNIST in this work under different partitions with different levels of statistical heterogeneity. The manually-partitioned version is more heterogeneous than the naturally-partitioned one, as we assign 5 classes to each device. We show that the benefits of `Ditto` can be more significant on the skewed FEMNIST data (Table 9). All results shown in the main text are based on the natural partition. We downsample the number of data points on each device (following the power law) for Vehicle. For FEMNIST, CelebA, and StackOverflow, we randomly sample devices (users) from the entire dataset. We use the full version of Fashion MNIST (which has been used in previous FL works (Bhagoji et al., 2019)), and assign 5 classes to each device.

F.2 PERSONALIZATION BASELINES

We elaborate on the personalization baselines used in our experiments (Table 3) which allow for partial device participation and local updating. We consider:

- **APFL** (Deng et al., 2021), which proposes to interpolate between local and global models for personalization. While it can reduce to solving local problems (without constraints on the solution space) as pointed out in Deng et al. (2021), we find that in neural network applications, it has some personalization benefits, possibly due to the joint optimization solver.
- **Elastic Weight Consolidation (EWC)**, which takes into account the Fisher information when finetuning from the optimal global model (Kirkpatrick et al., 2017; Yu et al., 2020). The local objective is $\min_w F_k(w) + \frac{\lambda}{2}\sum_i \mathbf{F}_{ii} \cdot (w[i] - w^*[i])^2$ where $[i]$ denotes the index of parameters and $\mathbf{F}_{ii}$ denotes the i-th diagonal of the empirical Fisher matrix $\mathbf{F}$ estimated using a data batch.
- **L2SGD**, which regularizes personalized models towards their mean (Hanzely & Richtárik, 2020). The proposed method requires full device participation once in a while. However, to remain consistent with the other solvers, we use their objective but adopt a different solver with partial device participation—each selected local device solving $\min_w F_k(w) + \frac{\lambda}{2}\|w - \bar{w}\|^2$ where $\bar{w}$ is the current mean of all personalized models $\bar{w} = \frac{1}{N}\sum_{k=1}^{N} w_k$.
- **Mapper**, which is one of the three personalization methods proposed in Mansour et al. (2020) that needs the minimal amount of meta-information. Similar to APFL, it is also motivated by model interpolation.
- **Per-FedAvg** (HF) (Fallah et al., 2020) which applies MAML (Finn et al., 2017) to personalize federated models with an Hessian-product approximation to approximate the second-order gradients.

[2] `http://www.ecs.umass.edu/~mduarte/Software.html`

[3] `https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets/stackoverflow/load_data`.

- **Symmetrized KL** constrains the symmetrized KL divergence between the prediction of finetuned models and that of the initialization. Specifically, in our setting, the local objective is $\min_w F_k(w) + \frac{\lambda}{2}\left(D_{\text{KL}}(f(w)||f(w^*)) + D_{\text{KL}}(f(w^*)||f(w))\right)$ where $D_{\text{KL}}(P||Q)$ is the KL-divergence between P and Q, and $f(\cdot)$ denotes the softmax probability for classification.

G Additional and Complete Experiment Results

G.1 Fairness of Ditto

Table 2: **Average (standard deviation)** test accuracy to benchmark performance and fairness (Definition 2) on Fashion MNIST and FEMNIST. `Ditto` is more fair and accurate.

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)

G.2 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)), (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 F.

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 3 below, we see that `Ditto` achieves similar or superier test accuracy with slightly lower standard deviation compared with these recent personalization methods. Further understanding the robustness/fairness benefits of other personalized approaches would be an interesting direction of future work.

G.3 Comparing Two Solvers

As mentioned in Section 3.1, 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^*. In non-convex cases, however, starting from a corrupted w^* may result in inferior performance compared with Algorithm 1. 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 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 7, the finetuning solver improves the performance compared with learning a global model, while `Ditto` combined with joint optimization performs

Table 3: `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)

the best. One can also perform finetuning after early stopping; however, it is essentially solving a different objective and it is difficult to do so in practice based on the training or validation data alone, as shown in Figure 9.

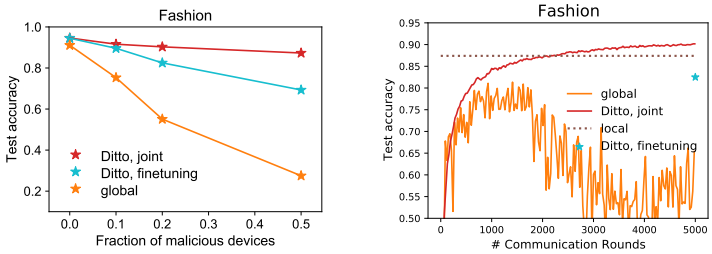

Figure 7: `Ditto` with joint optimization (Algorithm 1) outperforms the alternative local finetuning solver under the strong model replacement attack.

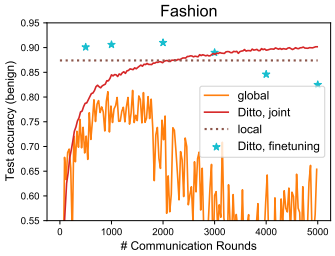

Figure 8: '`Ditto`, joint' achieves high test accuracy on benign devices. The performance can also be good if we first early stop at some specific points and then finetune.

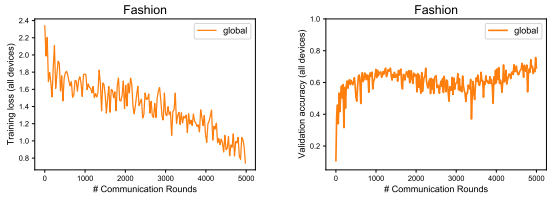

Figure 9: Finetuning is not very practical as it is difficult to determine when to stop training the global model by looking at the training loss (left) or validation accuracy (right) on all devices (without knowing which are benign).

G.4 Tuning λ

We assume that the server *does not* have knowledge of which devices are benign vs. malicious, and we have each device *locally* select and apply a best λ from a candidate set of three values based on their validation data. For benign devices, this means they will pick a λ based on their clean validation signal. For malicious devices, how they perform personalization (i.e., selecting λ) does not affect the corrupted global model updates they send, which are independent of λ. We further assume the devices have some knowledge of how 'strong' the attack is. We define strong attacks as (i) all of model replacement attacks (A3) where the magnitude of the model updates from malicious devices can scale by $> 10\times$, and (ii) other attacks where more than half of the devices are corrupted. In particular, 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. For the StackOverflow dataset, we tune λ from $\{0.01, 0.05, 0.1\}$ for strong attacks, and $\{0.05, 0.1, 0.3\}$ for all other attacks. We directly evaluate our hyperparameter tuning strategy in Table 4 below—showing that this dynamic tuning heuristic works well relative to an ideal, but more unrealistic strategy that picks the best λ based on knowledge of which devices are benign vs. malicious (i.e., by only using the validation data of the benign devices).

Table 4: Results (test accuracy and standard deviation) of using dynamic λ's. 'Best λ' refers to the results of selecting the best (fixed) λ based on average validation performance on benign devices (assuming the server knows which devices are malicious).

FEMNIST		**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	clean	20%	50%	80%	20%	50%	80%	10%	15%	20%
best λ	0.836 (.10)	0.803 (.10)	0.767 (.10)	0.672 (.14)	0.792 (.11)	0.743 (.14)	0.674 (.14)	0.691 (.15)	0.664 (.14)	0.650 (.14)
dynamic λ's	0.834 (.09)	0.802 (.10)	0.762 (.11)	0.672 (.13)	0.801 (.09)	0.700 (.15)	0.675 (.14)	0.685 (.15)	0.650 (.14)	0.613 (.13)
Fashion		**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	clean	20%	50%	80%	20%	50%	80%	10%	20%	50%
best λ	0.946 (.06)	0.944 (.08)	0.935 (.07)	0.925 (.07)	0.943 (.08)	0.930 (.07)	0.912 (.08)	0.914 (.09)	0.903 (.09)	0.873 (.09)
dynamic λ's	0.943 (.06)	0.944 (.07)	0.937 (.07)	0.907 (.10)	0.938 (.07)	0.930 (.08)	0.913 (.09)	0.921 (.09)	0.902 (.09)	0.872 (.11)
CelebA		**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	clean	20%	50%	80%	20%	50%	80%	10%	15%	20%
best λ	0.914 (.18)	0.828 (.22)	0.721 (.27)	0.724 (.28)	0.872 (.22)	0.826 (.26)	0 708 (.29)	0.699 (.28)	0.694 (.27)	0.689 (.28)
dynamic λ's	0.911 (.16)	0.820 (.26)	0.714 (.28)	0.724 (.28)	0.872 (.22)	0.826 (.26)	0.706 (.28)	0.699 (.28)	0.694 (.27)	0.689 (.28)
Vehicle		**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	clean	20%	50%	80%	20%	50%	80%	10%	20%	50%
best λ	0.882 (.05)	0.862 (.05)	0.841 (.09)	0.851 (.06)	0.884 (.05)	0.872 (.06)	0.879 (.04)	0.872 (.06)	0.829 (.08)	0.827 (.08)
dynamic λ's	0.872 (.05)	0.857 (.06)	0.827 (.08)	0.834 (.05)	0.872 (.06)	0.867 (.07)	0.848 (.04)	0.839 (.08)	0.824 (.08)	0.822 (.09)
StackOverflow		**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	clean	20%	50%	80%	20%	50%	80%	10%	20%	50%
best λ	0.315 (.16)	0.325 (.16)	0.315 (.17)	0.313 (.15)	0.314 (.16)	0.350 (.16)	0.312 (.14)	0.316 (.17)	0.321 (.17)	0.327 (.17)
dynamic λ's	0.317 (.17)	0.323 (.18)	0.314 (.16)	0.359 (.16)	0.326 (.17)	0.317 (.17)	0.301 (.17)	0.318 (.17)	0.319 (.17)	0.311 (.17)

G.5 DITTO AUGMENTED 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 5 below, Ditto combined with clipping or multi-Krum yields improvements compared with vanilla Ditto.

Table 5: Ditto augmented with robust baselines (full results).

FEMNIST	**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	20%	50%	80%	20%	50%	80%	10%	15%	20%
global	0.773 (.11)	0.727 (.12)	0.574 (.15)	0.774 (.11)	0.703 (.14)	0.636 (.15)	0.517 (.14)	0.487 (.14)	0.364 (.13)
clipping	0.791 (.11)	0.736 (.11)	0.408 (.14)	0.791 (.11)	0.736 (.13)	0.656 (.13)	0.795 (.11)	0.060 (.05)	0.061 (.05)
Ditto	0.803 (.10)	**0.767 (.10)**	**0.672** (.14)	0.792 (.11)	0.743 (.14)	0.674 (.14)	0.691 (.15)	0.664 (.14)	0.650 (.14)
Ditto + clipping	**0.810 (.11)**	0.762 (.11)	0.645 (.13)	**0.808 (.11)**	**0.757 (.11)**	**0.684 (.13)**	**0.813 (.13)**	**0.707 (.15)**	**0.672 (.14)**
CelebA	**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	20%	50%	80%	20%	50%	80%	10%	15%	20%
global	0.810 (.22)	0.535 (.26)	0.228 (.21)	0.869 (.22)	0.823 (.23)	0.656 (.26)	0.451 (.27)	0.460 (.29)	0.515 (.31)
multi-Krum	**0.882 (.22)**	0.564 (.26)	0.107 (.19)	0.887 (.21)	0.891 (.20)	0.617 (.30)	0.512 (.27)	0.529 (.27)	0.430 (.26)
Ditto	0.828 (.22)	0.721 (.27)	0.724 (.28)	0.872 (.22)	0.826 (.26)	0.708 (.29)	0.699 (.28)	0.694 (.27)	0.689 (.28)
Ditto + multi-Krum	0.875 (.20)	**0.722 (.26)**	**0.733 (.27)**	**0.903 (.20)**	**0.902 (.21)**	**0.885 (.23)**	**0.713 (.28)**	**0.709 (.28)**	**0.713 (.28)**

G.6 DITTO COMPLETE RESULTS

In Section 4, we present partial results on three strong attacks on one dataset. Here, we provide full results showing the robustness and fairness of Ditto on all attacks and all datasets compared with all defense baselines. We randomly split local data on each device into 72% train, 8% validation, and 20% test sets, and report all results on test data. We use a learning rate of 0.01 for StackOverflow, 0.05 for Fashion MNIST and 0.1 for all other datasets; and batch size 16 for CelebA and Fashion MNIST, 32 for FEMNIST and Vehicle, and 100 for StackOverflow. For every dataset, we first run FedAvg on clean data to determine the number of communication rounds. Then we run the same number of rounds for all attacks on that dataset.

For our robust baselines, 'median' means coordinate-wise median. For Krum, multi-Krum, k-norm, and k-loss, we assume the server knows the expected number of malicious devices when aggregation. In other words, for k-norm, we filter out the updates with the k largest norms where k is set to the expected number of malicious devices. Similarly, for k-loss, we only use the model update with the k+1-th largest training loss. For gradient clipping, we set the threshold to be the median of the gradient norms coming from all selected devices at each round. FedMGDA+ has an additional ε hyperparameter which we select from $\{0, 0.1, 0.5, 1\}$ based on the validation performance on benign devices. For the finetuning (only on neural network models) baseline, we run 50 epochs of mini-batch SGD on each device on the local objective F_k starting from w^*. We see that Ditto can achieve better fairness and robustness in most cases. In particular, on average of all datasets and all attack scenarios, Ditto (with dynamic λ's) achieves 6% absolute accuracy improvement compared with the strongest robust baseline. In terms of fairness, Ditto is able to reduce the variance of test accuracy by 10% while improving the average accuracy by 5% relative to state-of-the-art methods for fair FL (without attacks).

Table 6: Full results (average and standard deviation of test accuracy across all devices) on Vehicle.

Vehicle		**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	clean	20%	50%	80%	20%	50%	80%	10%	20%	50%
global	0.866 (.16)	0.847 (.08)	0.643 (.10)	0.260 (.27)	0.866 (.18)	0.840 (.21)	0.762 (.27)	0.854 (.17)	0.606 (.08)	0.350 (.19)
local	0.836 (.07)	0.835 (.08)	0.840 (.09)	**0.857 (.09)**	0.835 (.08)	0.840 (.09)	0.857 (.09)	0.840 (.07)	0.835 (.08)	**0.840 (.09)**
fair	0.870 (.08)	0.721 (.06)	0.572 (.08)	0.404 (.13)	0.746 (.12)	0.704 (.15)	0.706 (.20)	0.775 (.13)	0.628 (.25)	0.448 (.11)
median	0.863 (.16)	0.861 (.18)	0.676 (.11)	0.229 (.31)	0.864 (.18)	0.838 (.21)	0.774 (.28)	0.867 (.17)	0.797 (.07)	0.319 (.17)
Krum	0.852 (.17)	0.853 (.19)	0.830 (.22)	0.221 (.32)	0.851 (.19)	0.828 (.22)	0.780 (.31)	0.867 (.17)	**0.866 (.18)**	0.588 (.14)
multi-Krum	0.866 (.16)	0.867 (.18)	0.839 (.20)	0.220 (.32)	0.867 (.18)	0.839 (.22)	0.770 (.31)	0.868 (.17)	0.836 (.08)	0.406 (.15)
clipping	0.864 (.16)	0.865 (.17)	0.678 (.34)	0.234 (.30)	0.865 (.18)	0.839 (.22)	0.764 (.27)	0.868 (.17)	0.789 (.07)	0.315 (.17)
k-norm	0.866 (.16)	**0.867 (.17)**	0.838 (.21)	0.222 (.32)	0.867 (.18)	0.839 (.22)	0.778 (.31)	0.867 (.17)	0.844 (.09)	0.458 (.16)
k-loss	0.850 (.05)	0.755 (.03)	0.732 (.09)	0.217 (.31)	0.852 (.06)	0.840 (.07)	0.825 (.09)	0.866 (.17)	0.692 (.08)	0.328 (.16)
FedMGDA+	0.860 (.16)	0.835 (.09)	0.674 (.14)	0.270 (.26)	0.860 (.18)	0.843 (.22)	0.794 (.26)	0.836 (.17)	0.757 (.07)	0.676 (.17)
Ditto, λ=0.1	0.845 (.07)	0.841 (.08)	**0.841 (.09)**	0.851 (.06)	0.844 (.07)	0.848 (.08)	0.866 (.05)	0.838 (.07)	0.829 (.08)	0.827 (.08)
Ditto, λ=1	0.875 (.05)	0.859 (.06)	0.821 (.07)	0.776 (.08)	0.875 (.06)	0.870 (.07)	**0.879 (.04)**	0.860 (.07)	0.813 (.07)	0.757 (.08)
Ditto, λ=2	**0.882 (.05)**	0.862 (.05)	0.800 (.07)	0.709 (.12)	**0.884 (.05)**	**0.872 (.06)**	0.869 (.04)	**0.872 (.06)**	0.791 (.06)	0.690 (.09)

Table 7: Full results (average and standard deviation of test accuracy across all devices) on FEMNIST.

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	0.804 (.11)	0.773 (.11)	0.727 (.12)	0.574 (.15)	0.774 (.11)	0.703 (.14)	0.636 (.15)	0.517 (.14)	0.487 (.14)	0.364 (.13)
local	0.628 (.15)	0.620 (.14)	0.627 (.14)	0.607 (.13)	0.620 (.14)	0.627 (.14)	0.607 (.13)	0.622 (.14)	0.621 (.14)	0.620 (.14)
fair	0.809 (.11)	0.636 (.15)	0.562 (.13)	0.478 (.12)	0.440 (.15)	0.336 (.12)	0.363 (.12)	0.353 (.12)	0.316 (.12)	0.299 (.11)
median	0.733 (.14)	0.627 (.15)	0.576 (.15)	0.060 (.04)	0.673 (.14)	0.645 (.14)	0.564 (.15)	0.628 (.14)	0.573 (.15)	0.577 (.16)
Krum	0.717 (.16)	0.059 (.05)	0.096 (.07)	0.091 (.07)	0.604 (.14)	0.062 (.25)	0.024 (.02)	0.699 (.15)	0.719 (.13)	0.648 (.14)
multi-Krum	0.804 (.11)	0.790 (.11)	0.759 (.11)	0.115 (.07)	0.789 (.11)	0.762 (.11)	0.014 (.02)	0.529 (.14)	0.664 (.15)	0.561 (.14)
clipping	0.805 (.11)	0.791 (.11)	0.736 (.11)	0.408 (.14)	0.791 (.11)	0.736 (.13)	0.656 (.13)	0.795 (.11)	0.060 (.05)	0.061 (.05)
k-norm	0.806 (.11)	0.785 (.11)	0.760 (.12)	0.060 (.05)	0.788 (.10)	**0.765 (.11)**	0.011 (.02)	0.060 (.04)	0.647 (.15)	0.562 (.15)
k-loss	0.762 (.11)	0.606 (.13)	0.599 (.13)	0.596 (.13)	0.432 (.12)	0.508 (.13)	0.572 (.14)	0.060 (.04)	0.009 (.02)	0.006 (.01)
FedMGDA+	0.803 (.12)	0.794 (.12)	0.730 (.12)	0.057 (.04)	**0.793 (.12)**	0.753 (.12)	0.671 (.14)	**0.798 (.11)**	**0.794 (.12)**	**0.791 (.11)**
finetuning	0.815 (.09)	0.778 (.11)	0.734 (.12)	**0.671 (.13)**	0.764 (.11)	0.695 (.18)	0.646 (.14)	0.688 (.13)	0.671 (.14)	0.655 (.13)
Ditto, λ=0.01	0.800 (.15)	0.709 (.15)	0.683 (.17)	0.642 (.13)	0.701 (.14)	0.684 (.14)	0.645 (.14)	0.650 (.14)	0.628 (.14)	0.650 (.14)
Ditto, λ=0.1	0.827 (.10)	0.794 (.11)	0.755 (.13)	0.666 (.14)	0.786 (.13)	0.743 (.14)	**0.674 (.14)**	0.691 (.15)	0.664 (.14)	0.640 (.14)
Ditto, λ=1	**0.836 (.10)**	**0.803 (.10)**	**0.767 (.10)**	**0.672 (.14)**	**0.792 (.11)**	0.691 (.17)	0.575 (.17)	0.642 (.12)	0.595 (.14)	0.554 (.15)

Table 8: Full results (average and standard deviation of test accuracy across all devices) on Fashion MNIST.

Fashion MNIST		**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	clean	20%	50%	80%	20%	50%	80%	10%	20%	50%
global	0.911 (.08)	0.897 (.08)	0.855 (.10)	0.753 (.13)	0.900 (.08)	0.882 (.09)	0.857 (.10)	0.753 (.10)	0.551 (.13)	0.275 (.12)
local	0.876 (.10)	0.874 (.10)	0.876 (.11)	0.879 (.10)	0.874 (.10)	0.876 (.11)	0.879 (.10)	0.877 (.10)	0.874 (.10)	**0.876 (.11)**
fair	0.909 (.07)	0.751 (.12)	0.637 (.13)	0.547 (.11)	0.731 (.13)	0.637 (.14)	0.635 (.14)	0.653 (.13)	0.601 (.12)	0.131 (.16)
median	0.884 (.09)	0.853 (.10)	0.818 (.12)	0.606 (.17)	0.885 (.09)	0.883 (.09)	0.864 (.10)	0.856 (.09)	0.829 (.11)	0.725 (.15)
Krum	0.838 (.13)	0.864 (.11)	0.818 (.13)	0.768 (.15)	0.847 (.12)	0.870 (.11)	0.805 (.13)	0.868 (.11)	0.866 (.11)	0.640 (.18)
multi-Krum	0.911 (.08)	0.907 (.08)	0.889 (.10)	0.793 (.12)	0.849 (.10)	0.827 (.12)	0.095 (.12)	0.804 (.11)	0.860 (.09)	0.823 (.13)
clipping	0.913 (.07)	0.905 (.08)	0.875 (.10)	0.753 (.12)	0.904 (.08)	0.886 (.09)	0.856 (.11)	0.901 (.08)	0.844 (.11)	0.477 (.13)
k-norm	0.911 (.08)	0.908 (.08)	0.888 (.10)	0.118 (.08)	0.906 (.08)	0.893 (.09)	0.096 (.07)	0.765 (.14)	0.854 (.10)	0.828 (.12)
k-loss	0.898 (.08)	0.856 (.09)	0.861 (.10)	0.851 (.31)	0.876 (.09)	0.866 (.11)	0.870 (.10)	0.538 (.14)	0.257 (.13)	0.092 (.13)
FedMGDA+	0.915 (.08)	0.907 (.08)	0.874 (.10)	0.753 (.13)	0.911 (.08)	0.900 (.09)	0.873 (.10)	**0.914 (.08)**	**0.904 (.08)**	0.869 (.10)
finetuning	0.945 (.06)	**0.946 (.07)**	**0.935 (.07)**	0.922 (.08)	**0.945 (.07)**	**0.930 (.08)**	**0.923 (.08)**	**0.915 (.08)**	0.871 (.11)	0.764 (.15)
`Ditto`, λ=0.1	0.929 (.09)	0.920 (.09)	0.909 (.10)	0.897 (.10)	0.921 (.09)	0.914 (.09)	0.905 (.08)	**0.914 (.09)**	**0.903 (.09)**	**0.873 (.09)**
`Ditto`, λ=1	**0.946 (.06)**	**0.944 (.08)**	**0.935 (.07)**	**0.925 (.07)**	**0.943 (.08)**	**0.930 (.07)**	**0.912 (.08)**	0.887 (.09)	0.831 (.10)	0.740 (.12)
`Ditto`, λ=2	0.945 (.06)	0.942 (.06)	**0.935 (.07)**	0.917 (.07)	0.936 (.07)	0.923 (.08)	0.906 (.08)	0.871 (.09)	0.785 (.11)	0.606 (.14)

Table 9: Full results (average and standard deviation of test accuracy across all devices) on FEMNIST (skewed).

FEMNIST (skewed)		**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	clean	20%	50%	80%	20%	50%	80%	10%	15%	20%
global	0.720 (.24)	0.657 (.28)	0.585 (.30)	0.435 (.23)	0.688 (.26)	0.631 (.24)	0.589 (.26)	0.023 (.11)	0.038 (.18)	0.039 (.18)
local	0.915 (.18)	0.903 (.21)	0.937 (.18)	0.902 (.19)	0.903 (.21)	0.937 (.18)	0.902 (.19)	0.881 (.21)	**0.912 (.18)**	**0.903 (.21)**
fair	0.716 (.22)	0.644 (.29)	0.545 (.29)	0.421 (.22)	0.348 (.22)	0.321 (.16)	0.242 (.15)	0.010 (.11)	0.042 (.10)	0.037 (.17)
median	0.079 (.12)	0.086 (.12)	0.031 (.06)	0.044 (.08)	0.075 (.12)	0.109 (.13)	0.323 (.25)	0.060 (.10)	0.020 (.09)	0.033 (.07)
Krum	0.457 (.37)	0.360 (.35)	0.061 (.22)	0.127 (.27)	0.424 (.38)	0.051 (.08)	0.147 (.22)	0.434 (.36)	0.472 (.36)	0.484 (.35)
multi-Krum	0.725 (.25)	0.699 (.29)	0.061 (.22)	0.271 (.21)	0.712 (.29)	0.705 (.30)	0.584 (.28)	0.633 (.30)	0.556 (.30)	0.526 (.28)
clipping	0.727 (.28)	0.678 (.28)	0.604 (.34)	0.401 (.26)	0.726 (.26)	0.711 (.26)	0.645 (.24)	0.699 (.29)	0.674 (.28)	0.640 (.28)
k-norm	0.716 (.28)	0.691 (.30)	0.396 (.36)	0.005 (.08)	0.724 (.26)	0.721 (.29)	0.692 (.35)	0.612 (.29)	0.599 (.30)	0.565 (.28)
k-loss	0.587 (.21)	0.526 (.29)	0.419 (.36)	0.127 (.27)	0.555 (.23)	0.550 (.26)	0.093 (.16)	0.003 (.08)	0.009 (.07)	0.006 (.05)
finetuning	**0.948 (.11)**	0.942 (.13)	**0.959 (.10)**	**0.946 (.10)**	**0.949 (.16)**	0.918 (.21)	0.621 (.11)	0.788 (.25)	0.740 (.27)	0.751 (.26)
`Ditto`, λ=0.01	0.947 (.15)	**0.945** (.18)	0.955 (.20)	**0.946 (.13)**	0.942 (.18)	0.949 (.15)	**0.944 (.14)**	0.902 (.20)	0.895 (.23)	0.888 (.20)
`Ditto`, λ=0.1	**0.948 (.10)**	**0.945 (.14)**	**0.959 (.12)**	0.936 (.09)	**0.945 (.13)**	**0.948 (.10)**	0.888 (.18)	**0.936 (.16)**	0.827 (.23)	0.812 (.24)
`Ditto`, λ=1	0.902 (.15)	0.899 (.15)	0.907 (.15)	0.861 (.14)	0.899 (.18)	0.818 (.22)	0.423 (.41)	0.880 (.15)	0.730 (.28)	0.736 (.28)

Table 10: Full results (average and standard deviation of test accuracy across all devices) on CelebA.

CelebA		**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	clean	20%	50%	80%	20%	50%	80%	10%	15%	20%
global	0.911 (.19)	0.810 (.22)	0.535 (.26)	0.228 (.21)	0.869 (.22)	0.823 (.23)	0.656 (.26)	0.451 (.27)	0.460 (.29)	0.515 (.31)
local	0.692 (.27)	0.690 (.27)	0.682 (.27)	0.681 (.26)	0.690 (.27)	0.682 (.27)	0.681 (.26)	0.692 (.27)	0.693 (.27)	0.690 (.27)
fair	0.905 (.17)	0.724 (.27)	0.509 (.27)	0.195 (.21)	0.790 (.26)	0.646 (.27)	0.646 (.27)	0.442 (.27)	0.426 (.28)	0.453 (.28)
median	0.910 (.18)	0.872 (.22)	0.494 (.28)	0.126 (.18)	0.901 (.20)	0.864 (.20)	0.617 (.30)	0.885 (.20)	**0.891 (.19)**	**0.870 (.21)**
Krum	0.775 (.25)	0.810 (.25)	0.641 (.25)	0.377 (.10)	0.790 (.25)	0.699 (.25)	0.584 (.27)	0.780 (.25)	0.728 (.25)	0.685 (.30)
multi-Krum	0.911 (.19)	**0.882 (.22)**	0.564 (.26)	0.107 (.19)	0.887 (.21)	0.891 (.20)	0.617 (.30)	0.512 (.27)	0.529 (.27)	0.430 (.26)
clipping	0.909 (.18)	0.866 (.19)	0.485 (.29)	0.126 (.20)	0.897 (.20)	0.842 (.21)	0.665 (.26)	**0.901 (.20)**	0.883 (.21)	0.853 (.23)
k-norm	0.908 (.18)	0.870 (.22)	0.537 (.28)	0.105 (.17)	0.874 (.23)	**0.909 (.18)**	0.664 (.25)	0.506 (.28)	0.577 (.27)	0.449 (.28)
k-loss	0.873 (.19)	0.584 (.28)	0.550 (.31)	0.169 (.21)	0.595 (.28)	0.654 (.28)	0.683 (.26)	0.543 (.33)	0.458 (.33)	0.455 (.34)
FedMGDA+	0.909 (.19)	0.853 (.21)	0.508 (.28)	0.473 (.34)	**0.907 (.19)**	0.889 (.21)	**0.782 (.26)**	0.865 (.23)	0.805 (.26)	0.847 (.21)
finetuning	0.912 (.18)	0.814 (.24)	0.721 (.28)	0.691 (.29)	0.850 (.24)	0.800 (.25)	0.747 (.24)	0.665 (.28)	0.668 (.27)	0.673 (.28)
`Ditto`, λ=0.1	0.884 (.24)	0.716 (.27)	**0.721 (.27)**	**0.724 (.28)**	0.727 (.26)	0.708 (.28)	0.706 (.28)	0.699 (.28)	0.694 (.27)	0.689 (.28)
`Ditto`, λ=1	0.911 (.16)	0.820 (.26)	0.714 (.28)	0.675 (.29)	0.872 (.22)	0.826 (.26)	0.708 (.29)	0.629 (.29)	0.667 (.28)	0.685 (.28)
`Ditto`, λ=2	**0.914 (.18)**	0.828 (.22)	0.698 (.27)	0.654 (.28)	0.862 (.21)	0.791 (.26)	0.623 (.31)	0.585 (.29)	0.647 (.27)	0.655 (.29)

Table 11: Full results on (average and standard deviation of test accuracy across all devices) StackOverflow.

StackOverflow		**A1** (ratio of adversaries)			**A2** (ratio of adversaries)			**A3** (ratio of adversaries)		
Methods	clean	20%	50%	80%	20%	50%	80%	10%	15%	20%
global	0.155 (.13)	0.153 (.13)	0.156 (.16)	0.169 (.18)	0.147 (.12)	0.009 (.03)	0.013 (.01)	0.000 (.00)	0.000 (.00)	0.000 (.00)
local	0.311 (.15)	0.311 (.15)	0.313 (.15)	**0.319 (.15)**	0.311 (.15)	0.313 (.15)	**0.319 (.15)**	0.311 (.15)	0.313 (.15)	0.319 (.15)
fair	0.154 (.13)	0.155 (.14)	0.153 (.13)	0.141 (.10)	0.000 (.00)	0.000 (.00)	0.000 (.00)	0.148 (.12)	0.152 (.13)	0.167 (.11)
median	0.002 (.00)	0.001 (.00)	0.000 (.00)	0.000 (.00)	0.000 (.00)	0.001 (.00)	0.000 (.00)	0.000 (.00)	0.000 (.00)	0.000 (.00)
Krum	0.154 (.13)	0.150 (.13)	0.041 (.04)	0.002 (.00)	0.158 (.13)	0.151 (.13)	0.167 (.12)	0.153 (.13)	0.154 (.14)	0.138 (.15)
clipping	0.154 (.13)	0.157 (.13)	0.149 (.13)	0.163 (.17)	0.152 (.13)	0.001 (.01)	0.001 (.01)	0.155 (.12)	0.161 (.14)	0.120 (.16)
k-norm	0.154 (.13)	0.156 (.12)	0.100 (.08)	0.002 (.00)	0.086 (.11)	0.042 (.03)	0.001 (.00)	0.149 (.15)	0.144 (.15)	0.155 (.13)
k-loss	0.155 (.13)	0.160 (.12)	0.164 (.13)	0.129 (.14)	0.136 (.11)	0.145 (.11)	0.156 (.14)	0.148 (.14)	0.159 (.13)	0.156 (.13)
FedMGDA+	0.155 (.12)	0.154 (.13)	0.152 (.13)	0.165 (.13)	0.147 (.13)	0.160 (.14)	0.101 (.09)	0.155 (.13)	0.158 (.12)	0.154 (.13)
`Ditto`, λ=0.05	**0.315 (.16)**	**0.325 (.16)**	**0.315 (.17)**	0.313 (.15)	**0.314 (.16)**	**0.350 (.16)**	0.312 (.14)	0.316 (.17)	**0.321 (.17)**	**0.327 (.17)**
`Ditto`, λ=0.1	0.309 (.17)	0.318 (.17)	**0.315 (.17)**	0.293 (.13)	0.309 (.17)	0.316 (.16)	0.307 (.14)	**0.319 (.17)**	0.302 (.17)	0.305 (.17)
`Ditto`, λ=0.3	0.255 (.18)	0.298 (.18)	0.288 (.17)	0.304 (.16)	0.283 (.17)	0.233 (.18)	0.321 (.20)	0.252 (.17)	0.261 (.19)	0.269 (.17)

H 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.