Federated Robustness Propagation: Sharing Adversarial Robustness in Heterogeneous Federated Learning

Junyuan Hong[1], **Haotao Wang**[2], **Zhangyang Wang**[2], **Jiayu Zhou**[1]

[1]Department of Computer Science and Engineering, Michigan State University
[2]Department of Computer Science and Engineering, University of Texas at Austin
{hongju12,jiayuz}@msu.edu, {htwang,atlaswang}@utexas.edu

Abstract

Federated learning (FL) emerges as a popular distributed learning schema that learns a model from a set of participating users without sharing raw data. One major challenge of FL comes with heterogeneous users, who may have distributionally different (or non-iid) data and varying computation resources. As federated users would use the model for prediction, they often demand the trained model to be robust against malicious attackers at test time. Whereas adversarial training (AT) provides a sound solution for centralized learning, extending its usage for federated users has imposed significant challenges, as many users may have very limited training data and tight computational budgets, to afford the data-hungry and costly AT. In this paper, we study a novel FL strategy: propagating adversarial robustness from rich-resource users that can afford AT, to those with poor resources that cannot afford it, during federated learning. We show that existing FL techniques cannot be effectively integrated with the strategy to propagate robustness among non-iid users and propose an efficient propagation approach by the proper use of batch-normalization. We demonstrate the rationality and effectiveness of our method through extensive experiments. Especially, the proposed method is shown to grant federated models remarkable robustness even when only a small portion of users afford AT during learning. Source code can be accessed at https://github.com/illidanlab/FedRBN.

1 Introduction

Federated learning (FL) (McMahan et al. 2017) is a learning paradigm that trains models from distributed users or participants (e.g., mobile devices) without requiring raw training data to be shared, alleviating the rising concern of privacy issues when learning with sensitive data and facilitating learning deep models by enlarging the amount of data for training. In a typical FL algorithm, each user trains a model locally using their own data and a server iteratively aggregates users' intermediate models, converging to a model that fuses training information from all users.

A major challenge in FL comes from two types of the user heterogeneity. One type of heterogeneity is distributional differences in training data collected by users from diverse user groups, namely *data heterogeneity* (Fallah, Mokhtari, and Ozdaglar 2020). The heterogeneity should be carefully handled during the learning as a single model trained by FL may fail to accommodate the differences and sacrifices model accuracy (Yu, Bagdasaryan, and Shmatikov 2020). Another type of heterogeneity is the difference of computing resources, named hardware heterogeneity, as different types of hardware used by users usually result in varying computation budgets. For example, consider an application scenario of FL from mobile phones (Hard et al. 2019), where different types of mobile phones (e.g., generations of the same brand) may have drastically different computational power (e.g., memory or CPU frequency). As the model size scales with task complexities, the ubiquitous hardware heterogeneity may expel a great number of resource-limited users from the FL process, reduces training data and therefore calls for hardware-aware alternatives (Diao, Ding, and Tarokh 2021).

The negative impacts of the heterogeneity become aggravated when an adversarially robust model is desired but its training is not affordable by some users. The essence of robustness comes from the unnatural vulnerability of models against visually imperceptible noise that can significantly mislead model predictions. To gain robustness, a straightforward extension of FL, federated adversarial training (FAT), can be adopted (Zizzo et al. 2020; Reisizadeh et al. 2020), where each user trains models with adversarially noised samples, namely adversarial training (AT) (Madry et al. 2018). Despite the robustness benefit by AT, prior studies pointed out that the AT is data-thirsty and computationally expensive (Shafahi et al. 2019a). Given the fact that each individual user may not have enough data to perform AT, involving a fair amount of users in FAT becomes essential, but may also induce higher data heterogeneity from diverse data sources. Meanwhile, the increasingly intensive computation can be prohibitive especially for resource-limited users, that could be 3-10 times more costly than the standard equivalent (Shafahi et al. 2019a; Zhang et al. 2019). As such, it is often unrealistic to enforce *all* users in a FL process to conduct AT locally, despite the fact that the robustness is indeed strongly desired or even required for all users. This conflict raises a challenging yet interesting question: Is it possible to *propagate adversarial robustness in FL* so that resource-limited users can efficiently benefit from robust training of resource-sufficient users even if the latter's data distribute differently?

Motivated by the question above, we formulate a *novel learning problem* called Federated Robustness Propagation

(FRP). We consider a rather common non-iid FL setting that involves budget-sufficient users (AT users) that conduct adversarial training, and budget-limited ones (ST users) that can only afford standard training. The goal of FRP is to propagate the adversarial robustness from AT users to ST users, especially when they have different data distributions. In **??**, we show that independent AT by users without FL (`local AT`) will not yield a robust model since each user has scarce training data. Directly extending an existing FL algorithm, e.g., *FedAvg* (McMahan et al. 2017) or a heterogeneity-mitigated one *FedBN* (Li et al. 2020b) with AT treatments, dubbed FATAvg and FATBN, give very limited capability of robustness. In this paper, we first provide a *novel insight* on the failure of the traditional method that robust BNs has non-trivial gaps among domains. Even if ST users can borrow the BN parameters from other resource-sufficient AT users, the integrated model will not be robust on the ST users, due to the data heterogeneity among users.

As conducting AT is so inefficient for ST users, we propose a *novel method* Federated Robust Batch-Normalization (FedRBN) to facilitate effective and efficient sharing of adversarial robustness among users with non-iid data. Our contributions are summarized as follows. **1)** We reformulate the device- and data-heterogeneous federated adversarial learning into a novel propagation problem (FRP). **2)** We propose a novel and efficient solution for robustness sharing and enhancement. First, we propagate robustness by aggregating the desired knowledge adaptively from multiple AT users to ST users efficiently embedded in few personalized (BN) parameters. Second, to promote the transferability of robust BNs, we calibrate non-personalized parameters when preserving the robustness of shared noise-aware BNs. **3)** Extensive experiments demonstrate the feasibility and effectiveness of the proposed method. Here, we highlight some experimental results from Section 5. When only 20% of non-iid users used AT during learning, the proposed FedRBN yields robustness, competitive with the best all-AT-user result by only a 6% drop (out of 62%) on robust accuracy. Note that even if our method with 100% AT users increase the upper bound of robustness, such a bound is usually not attainable in the presence of resource-limited users that cannot afford AT during learning.

2 Related Work

Federated learning for robust models. The importance of adversarial robustness in the context of federated learning, i.e., federated adversarial training (FAT), has been discussed in a series of recent literature (Zizzo et al. 2020; Reisizadeh et al. 2020; Kerkouche, Ács, and Castelluccia 2020; Chen et al. 2022). Zizzo *et al.* (Zizzo et al. 2020) empirically evaluated the feasibility of practical FAT configurations (e.g., ratio of adversarial samples) augmenting FedAvg with AT but only in *iid* and label-wise non-*iid* scenarios. The adversarial attack in FAT was extended to a more general affine form, together with theoretical guarantees of distributional robustness (Reisizadeh et al. 2020; Chen et al. 2022). It was found that in a communication-constrained setting, a significant drop exists both in standard and robust accuracies, especially with non-*iid* data (Shah et al. 2021). In addition to the challenges investigated above, this work studies challenges imposed by hardware heterogeneity in FL, which was rarely discussed. Especially, when only limited users have devices that afford AT, we strive to efficiently share robustness among users, so that users without AT capabilities can also benefit from such robustness.

Robust federated optimization. Another line of related work focuses on the robust aggregation of federated user updates (Kerkouche, Ács, and Castelluccia 2020; Fu et al. 2019). Especially, Byzantine-robust federated learning (Blanchard et al. 2017) aims to defend malicious users whose goal is to compromise training, e.g., by model poisoning (Bhagoji et al. 2018; Fang et al. 2020) or inserting model backdoor (Bagdasaryan et al. 2018). Various strategies aim to eliminate the malicious user updates during federated aggregation (Chen, Su, and Xu 2017; Blanchard et al. 2017; Yin et al. 2018; Pillutla, Kakade, and Harchaoui 2020). However, most of them assume the normal users are from similar distributions with enough samples such that the malicious updates can be detected as outliers. Therefore, these strategies could be less effective on attacker detection given a finite dataset (Wu et al. 2020). Even though both the proposed FRP and Byzantine-robust studies work with robustness, they have fundamental differences: the proposed work focus on *the robustness during inference*, i.e., after the model is learned and deployed, whereas Byzantine-robust work focus on the robust learning process. As such, the proposed approach can combine with all Byzantine-robust techniques to provide training robustness.

3 New Problem: Federated Robustness Propagation

In this section, we will review AT, present the unique challenges from hardware heterogeneity in FL and formulate the challenge of Device- and Data-Heterogeneous Federated Adversarial Learning resulting a new learning problem: *federated robustness propagation* (FRP). In this paper, we assume that a dataset D includes sampled pairs of images $x \in \mathbb{R}^d$ and labels $y \in \mathbb{R}^c$ from a distribution $\mathcal{D}$. Though our discussion limits the data as images in this paper, the method can be easily generalized to other data forms. We model a classifier, mapping from the $\mathbb{R}^d$ input space to classification logits $f : \mathbb{R}^d \to \mathbb{R}^c$, by a deep neural network (DNN) with batch-normalization (BN) layers. Generally, we split the parameters of f into two parts: (μ, σ^2) including all mean and variance in all BN layers and θ in others. To specify a BN structure, e.g., BN_c with identity name c in multiple candidates, we use the notation $f(x; BN_c)$. Whenever not causing confusion, we use the symbol of a model and its parameters interchangeably. For brevity, we slightly abuse $\mathbb{E}[\cdot]$ for both empirical average and expectation and use $[N]$ to denote $\{1, \dots, N\}$.

3.1 Standard Training and Adversarial Training

An *adversarial attack* applies a bounded noise $\delta_\epsilon : \|\delta_\epsilon\| \le \epsilon$ to an image x such that the perturbed image $A_\epsilon(x) \triangleq x + \delta_\epsilon$ can mislead a well-trained model to give a wrong prediction. The norm $\|\cdot\|$ can take a variety of forms, e.g., L_∞-norm for constraining the maximal pixel scale. A model f is said to be *adversarially robust* if it can predict labels correctly on a perturbed dataset $\tilde{D} = \{(A_\epsilon(x), y) | (x, y) \in D\}$, and the standard accuracy on D should not be greatly impacted.

Consider the general learning objective: $\min_f L(f, D) = \mathbb{E}_{(x,y)\in D}[\ell(f; x, y)]$. A user performs *standard training (ST)* if $\ell = \ell_c$ is a standard classification loss on clean images, for example, cross-entropy loss $\ell_{\text{CE}}(f(x), y) = -\sum_{t=1}^{c} y_t \log(f(x)_t)$ where t is the class index and $f(x)_t$ represents the t-th output logit. In contrast, a user performs *adversarial training (AT)* if $\ell = (\ell_a + \ell_{\text{CE}})/2$ where ℓ_a is an adversarial classification loss on noised images. A popular instantiation of ℓ_a is based on PGD attacks (Madry et al. 2018; Tsipras et al. 2019): $\ell_a(f; x, y) = \max_{\|\delta\| \le \epsilon} \ell(f(x + \delta), y)$, where $\|\cdot\|$ is the L_∞-norm. With ℓ_c and ℓ_a, we can accordingly define L_{ST} and L_{AT}.

3.2 Federated Robust Propagation for Device- and Data-Heterogeneous Federated Adversarial Learning

To formulate the Device- and Data-heterogeneous Federated Adversarial Learning (DDFAL), We start with a typical data-heterogeneous FL setting: a finite set of non-identical distributions $\mathcal{D}_i$ for $i \in [C]$, from which a set of datasets $\{D_k\}_{k=1}^K$ are sampled and distributed to K users' devices. Meanwhile, in the co-existing device-heterogeneous setting, the users from distinct domains related with $\mathcal{D}_i$ expect to learn together while optimize different objectives based on their resource constraints: Some users can afford AT training (*AT users* from group S) whereas the remaining users cannot afford and use standard training (*ST users* from group T). Confined with the resource constraints and various data domains, *federated robustness propagation (FRP)* reformulate the DDFAL as the goal to efficiently transfer the robustness from AT users to ST users at minimal computation and communication costs while preserve data locally. Formally, the FRP objective minimizes:

$$\begin{aligned} \text{FRP}(\{f_k\}; \{D_k | D_k \sim \mathcal{D}_i\}) \triangleq \textstyle\sum_{k\in T} L_{\text{ST}}(f_k, D_k) \\ + \textstyle\sum_{k\in S} L_{\text{AT}}(f_k, D_k). \quad (1) \end{aligned}$$

In the federated setting, each user's model is trained separately when initialized by a global model, and is aggregated to a global model at the end of each epoch. A popular aggregation technique is FedAvg (McMahan et al. 2017), which averages parameters by $f = \frac{1}{K}\sum_{k=1}^{K} a_k f_k$ with normalization coefficients a_k proportional to $|D_k|$. The most related setting to our work is FAT (Zizzo et al. 2020). But different from FAT, FRP defined in Eq. (1) formalizes two types of user heterogeneity that commonly exist in FL. The first one is the *hardware heterogeneity* where users are divided into two groups by computation budgets (S and T). Besides, *data heterogeneity* is represented as $\mathcal{D}_i$ differing by domain i. We limit our discussion as the common feature distribution shift (on x) in contrast to the label distribution shift (on y), as previously considered in (Li et al. 2020b).

New Challenges in FRP. We emphasize that *jointly* addressing the two types of heterogeneity in Eq. (1) forms a new challenge, distinct from either of them considered exclusively. First, the scarcity of the AT group worsens the data heterogeneity for additional distribution shift in the hidden representations from adversarial noise (Xie and Yuille 2019). That means even if two users are sampled from the same distribution, their classification layers may operate on different distributions.

Second, the data heterogeneity makes the transfer of robustness non-trivial (Shafahi et al. 2019b). Hendrycks *et al.* (Hendrycks, Lee, and Mazeika 2019) discussed the transfer of models adversarially trained on multiple domains and massive samples. Later, Shafahi *et al.* (Shafahi et al. 2019b) firstly studied the transferability of adversarial robustness from one data domain to another by fine-tuning. Distinguished from all existing work, the FRP problem focuses on propagating robustness from multiple AT users to multiple ST users who have diverse distributions and participate in the same federated learning. Thus, fine-tuning all source models in ST users is often not possible due to prohibitive computation costs.

4 New Method: Federated Robust Batch-Normalization (FedRBN)

To address the challenges in FRP, we propose a novel federated learning method that propagates robustness using batch-normalization (BN). Recall that BN mitigates the layer distributional shifts and greatly stabilizes the training of very deep networks (Ioffe and Szegedy 2015). A BN layer maps a biased variable to a normalized one by

$$\text{BN}(x; \mu, \sigma) \triangleq w \frac{x-\mu}{\sqrt{\sigma^2+\epsilon_0}} + b, \qquad (2)$$

where μ and σ^2 are the estimated mean and variance over all non-channel dimensions, and ϵ_0 is a small value to avoid zero division. Since w and b are not distribution-dependent but trainable parameters, we omit them from the notation $\text{BN}(x; \mu, \sigma)$, for brevity.

Revisit the role of BN in handling data heterogeneity. It is known that batch-normalization can model the internal distributions of activations and mitigate the distribution shifts by normalization. Therefore, it has been applied to cases where data distribution shifts occur. The basic principle is to apply *different BNs for different distributions*, by which the output of BN will be distributionally aligned. In this paper, two kinds of distribution biases are of our interest and their corresponding mitigation methods can be unified into the same principle: **(1)** Feature biases and LBN. When users collect data from different sources, their data consists of features biased by different environments. Though locally trained BNs tend to characterize the biases, the differences captured are immediately forgotten by a global averaging, for instance, in FedAvg. With the insight, FedBN (Li et al. 2020b) adopts localized batch-normalization (LBN) for each user, which will be eliminated from the global averaging. Thus, FedBN outputs K models with LBNs: $\{(\theta, \mu_k, \sigma_k^2)\}_{k=1}^K$. **(2)** Adversarial biases and DBN. Recently, (Xie and Yuille 2019) showed that adversarial samples are distributionally biased from clean samples especially in the internal activations of DNN, although the biases are almost invisible in the image. Such biases substantially lower robustness gained from adversarial training. Thus, Xie *et al.* (Xie et al.

2020) proposed a dual batch-normalization (DBN) structure which redirects noised and cleans inputs to different BNs during training: $\mathrm{BN}_a(x; \mu_a, \sigma_a^2)$ given a adversarially-noised x and $\mathrm{BN}_c(x; \mu, \sigma^2)$ given clean x. For example, the adversarial training will instead optimize $\ell_c(f(x; \mathrm{BN}_c)) + \ell_a(f(x; \mathrm{BN}_a))$. After training, it is recommended to use BN_a for improved robustness. Though not as accurate as BN_c, BN_a are still accurate.

(1) Joint use of LBN and DBN in FRP and its limitation. Because of the co-occurrence of feature heterogeneity and adversarial training in FRP, it is natural to adopt both LBN and DBN in FL. We name the combination as FATBN+DBN. That admits an extended set of BN parameters, (μ_k, σ_k^2) (clean), and $(\mu_{a,k}, \sigma_{a,k}^2)$ (adversarial), for user k. Since the essence of LBN and DBN are well established, it should be natural to use them together when two kinds of biases present. Interested readers may be referred to Appendix for qualitative evidence of such essence. Later in benchmark experiments (c.f. Table 2), we also show that the joint use boosts the robustness than using one of them exclusively.

However, the accuracy boosting comes with the challenges as device- and data-heterogeneity presents simultaneously. Because of lower computation capacities, ST users cannot afford the AT due to the limited computation resources. Without globally aggregating DBNs, ST users have to leave one branch of DBN blank or random, because no adversarial samples are provided to tune them. The missing branch makes the ever-successful method inapplicable with the device heterogeneity. Trivial fixture like using local BN_c (Fixture 1) or making the BN_a (Fixture 2) globally averaged can not fill in the performance gap, because either the clean or global BN has obvious distributional biases against the desired ones. We visualize such gaps between local BN_c and local BN_a in Appendix. We also empirically evaluate the gaps in Table 2 to show that Fixture 1 (FedRBN $\lambda = 0$) and 2 (FATAvg+DBN) in FRP (20% and MNIST cases) cannot gain comparable robustness (RA) as in the fully AT cases (All). Thus, an efficient manner without heavy computation overhead is desired to fill the gap.

(2) Fill in the missing statistics in the DBN via theory-guided adaptive propagation. To address the limitation of the above combination, we propose a simple estimation of the missing BN_a with global averaging:

$$\hat{\mu}_{a,k} = \tfrac{1}{|S|} \textstyle\sum_{j \in S} \alpha_j \mu_{a,j}, \; \hat{\sigma}_{a,k}^2 = \tfrac{1}{|S|} \textstyle\sum_{j \in S} \alpha_j \sigma_{a,j}^2, \quad (3)$$

where α_j is a normalized weight. As Eq. (3) is simply a linear operation, the estimation is very efficient due to the small portion of BN parameters in a deep network. To find an ideal α minimizing the adversarial loss during inference, below we theoretically show that the divergence of a clean pair bounds the generalizable adversarial loss, given bounded adversarial bias.

Lemma 4.1 (Informal state)**.** *Suppose the divergence between any data distribution $\mathcal{D}$ and its adversarial one $\tilde{\mathcal{D}}$ is bounded by a constant, i.e., $d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{\mathcal{D}}, \mathcal{D}) \le d_\epsilon$ where $d_{\mathcal{H}\Delta\mathcal{H}}$ is $\mathcal{H}\Delta\mathcal{H}$-divergence in hypothesis space $\mathcal{H}$. If a target model is formed by Eq.* (3) *of models trained on a set of source datasets $\{D_{s_i}\}$, its generalization error on the target $\tilde{\mathcal{D}}_t$ is upper bounded by the weighted summation $\sum_i \alpha_i d_{\mathcal{H}\Delta\mathcal{H}}(D_{s_i}, D_t)$ of paired divergence given $D_t \sim \mathcal{D}_t$.*

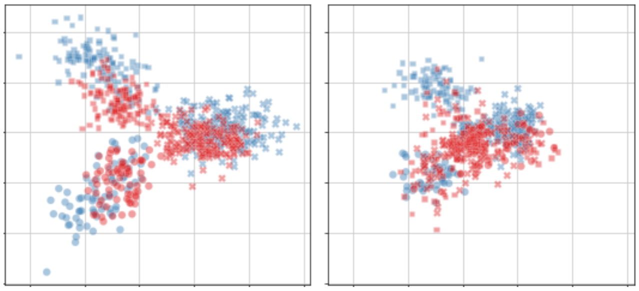


Figure 1: Penultimate layer representations of clean samples (left) and adversarial samples (right) visualized by a Digits model and SVHN-domain users. The visualization projects 400 randomly-selected samples into the first three classes of SVHN datasets following (Müller, Kornblith, and Hinton 2019). Representations are computed by trained (blue) or transferred (red) BN_a. The model is trained by 100%-AT (blue) or 1-domain-AT (red) user. In the latter setting, BN_a is propagated according to Eq. (3).

The lemma extends an existing bound for federated domain adaptation (Peng et al. 2019b), and shows that the generalization error on the unseen target noised distribution $\tilde{D}_t$ is bounded by the α_i-weighted standard distribution gaps.

Results in Lemma 4.1 and the domain gaps between adversarial samples motivate us to set α_j to be reversely proportional to the divergence between D_k and D_j. Since other users' data are not available, directly modeling the divergence is by data is prohibitive. Fortunately, as clean BN statistics characterize each user's data distributions, we can use a layer-averaged similarity to approximate the weight, i.e.,

$$\alpha_j = \mathrm{Softmax}_T \left[\tfrac{1}{L} \textstyle\sum_{l=1}^{L} \mathrm{Sim}^l(D_k, D_j) \right], \quad (4)$$

where $\mathrm{Softmax}_T(q_j)$ is a tempered softmax function: $\exp(q_j/T)/\sum_{j \in S} \exp(q_j/T)$. T equals 0.01 by default in this paper. The l-th-layer similarity is approximated by the BN statistics: $\mathrm{Sim}^l(D_k, D_j) = [\cos(\mu_k^l, \mu_j^l) + \cos(\sigma_k^{2\,l}, \sigma_j^{2\,l})]/2$ given $\cos(x, y) = x^\top y / \|x\| \, \|y\|$.

(3) Reduce biases in conv-layers via pseudo-noise calibration. Lemma 4.1 suggests that the optimal divergence will be no better than the divergence by the model from the most similar source. When all source datasets are from domains distinguished from the target domain, then estimated BN parameters by Eq. (3) cannot further compress divergence and improve adversarial losses. In Fig. 1, we show the non-reducible domain gap between MNIST and SVHN: the transferred BN_a yields a much less discriminative representations than the locally trained BN_a during training. In addition, we surprisingly observe that the clean discrimination is not well transferred, either. The observation implies that though non-BN parameters are trained and adapted towards different domains, the single-domain BN_a still cast biases into the representations even for clean samples. To fix this, we propose a clean adaptation of the federated model, which calibrates

Algorithm 1: FedRBN: user-end training

```
Input: User budget type (AT or ST), initial parameters $\theta$ (AT) or $(\theta, \hat{\mu}_a, \hat{\sigma}_a^2)$ (ST) of the model $f$ from the server, adversary $A_\epsilon(\cdot)$, dataset $D$
for mini-batch $\{(x, y)\}$ in $D$ do
    $\ell_c \leftarrow \mathbb{E}_{(x,y)}[\ell_{\mathrm{CE}}(f(x; \mathrm{BN}_c), y)]$
    Update $(\mu, \sigma^2)$ of $\mathrm{BN}_c$
    if user budget type is AT then
        Perturb data $\tilde{x} \leftarrow A_\epsilon(f(x; \mathrm{BN}_a))$
        $L \leftarrow \frac{1}{2}\left\{\ell_c + \mathbb{E}_{(\tilde{x},y)}[\ell_{\mathrm{CE}}(f(\tilde{x}; \mathrm{BN}_a), y)]\right\}$
        Update $(\mu_a, \sigma_a^2)$ of $\mathrm{BN}_a$
    else
        * Replace $\mathrm{BN}_a$ parameters with $(\hat{\mu}_a, \hat{\sigma}_a^2)$ *
        * $L \leftarrow (1 - \lambda)\ell_c + \lambda\mathbb{E}_{(x,y)}[\ell_{\mathrm{CE}}(f(x; \mathrm{BN}_a), y)]$ *
    end if
    Optimize $L$ to update $\theta$ by gradient descent
end for
Upload $(\theta, \mu, \sigma^2, \mu_a, \sigma_a^2)$ (AT) or $(\theta, \mu, \sigma^2)$ (ST)
```

Algorithm 2: FedRBN: server-end training

```
Input: An initial model $f$ with BN parameters $(\hat{\mu}_a, \hat{\sigma}_a^2)$ and other non-BN parameters $\theta$, $K$ users belonging to $S$ (AT) or $T$ (ST) sets, total iteration number $\tau$
for $t \in \{1, \ldots, \tau\}$ do
    Send global model $\theta_k$ to users indexed by $k \in S$ and $(\theta_k^t, \hat{\mu}_{a,k}, \hat{\sigma}_{a,k}^2)$ to users indexed by $k \in T$
    In parallel, users train their models by Algorithm 1
    Receive users' parameters:
        $\{(\theta_k, \mu_k, \sigma_k^2)\}_{k \in T}$ and $\{(\theta_k, \mu_k, \sigma_k^2, \mu_{k,a}, \sigma_{k,a}^2)\}_{k \in S}$
    Average parameters: $\theta \leftarrow \frac{1}{K}\sum_{k=1}^{K} \theta_k$
    * Use $\{(\mu_k, \sigma_k^2, \mu_{k,a}, \sigma_{k,a}^2)\}_{k \in S}$ to estimate adversarial BN parameters $\{(\hat{\mu}_{k,a}, \hat{\sigma}_{k,a}^2)\}_{k \in T}$ by Eq. (3) *
end for
Return $K$ models parameterized by $\{(\theta, \mu_k, \sigma_k^2, \mu_{k,a}, \sigma_{k,a}^2)\}_k$
```

non-BN parameters by local clean features only: (1) Given the estimated $(\hat{\mu}_{a,k}, \hat{\sigma}_{a,k}^2)$, keep the two parameters *frozen* to avoid statistic interference from clean samples. As the distributional biases between domains are typically larger than that between clean and adversarial statistics, freezing BN_a can impede catastrophic forgetting of the critical robustness knowledge. (2) Optimize an augmented ST loss:

$$(1 - \lambda)\ell_{\mathrm{CE}}\big(f_k(x; \mathrm{BN}_c), y\big) + \lambda\ell_{\mathrm{CE}}\big(f_k(x; \mathrm{BN}_a), y\big), \quad (5)$$

where the second term, pseudo-noise calibration (PNC) loss, augments the robustness by BN_a without computation-intensive adversarial attacks. If without the domain gap, $f_k(x; \mathrm{BN}_a)$ will bias the outputs on clean input x, which functions like noising the training process. Otherwise, frozen BN_a can calibrate the other parameters to mitigate the distributional bias such that the robustness encoded in BN_a is transferable. In Eq. (5), the hyper-parameter λ is set to be 0.5 by default, which provides a fair trade-off between robustness and accuracy. A smaller λ can be used to trade in robustness for accuracy, or vice versa.

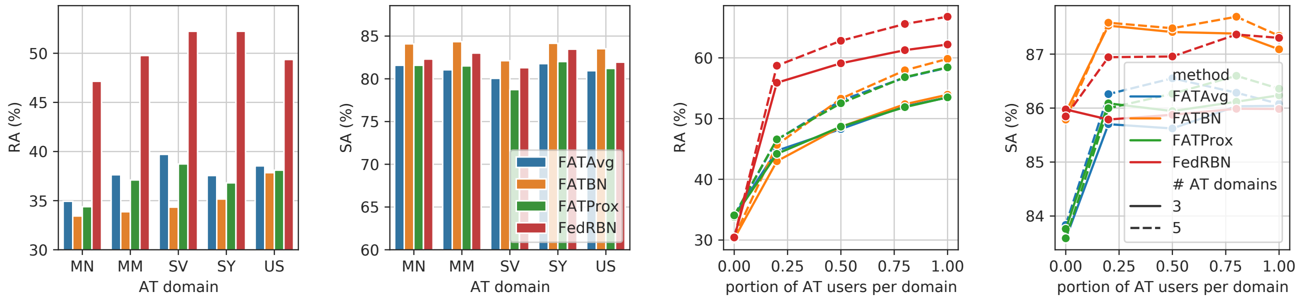
We are now ready to present the proposed BN-based FRP algorithm: Federated Robust Batch-Normalization (FedRBN). On the user side (Algorithm 1), we introduce a standard loss in addition to the standard federated adversarial training. The loss is embarrassingly simple and easy to implement in two lines, as highlighted. On the server side (Algorithm 2), we follow the same practice as FedAvg to aggregate models and average (perhaps weighted if users' sample sizes differs). Different from FedAvg, we drop unnecessary parameter sharing like sending BN parameters to AT users and leverage the globally shared BN parameters to estimate missing BN_a parameters.

Efficiency and privacy of BN operations. Since BN statistics are only a tiny portion of any networks and do not require back-propagation, an additional set of BN statistics will marginally impact the efficiency (Wang et al. 2020). During training, the communication cost is almost the same as the most popular FL method, FedAvg (McMahan et al. 2017), with a small portion of additional BN parameters. On the user side, the major computation overhead comes from the additional loss, which doubles the complexity of a ST user. However, the overhead is much cheaper than adversarial training, which typically requires multiple iterations (e.g., 7 steps (Madry et al. 2018)) of gradient descent for attacks. Many existing FL designs such as FedAvg have privacy concerns (Li et al. 2020a; Fallah, Mokhtari, and Ozdaglar 2020; Xiong et al. 2021, 2022), and sharing local statistics can also contribute to potential privacy leakage (Geiping et al. 2020). Though not the scope of this work, we can implement protection by applying differential privacy mechanism (Dwork et al. 2006) on the BN statistic estimation, where a minor Gaussian noise is injected on every statistic update in Algorithm 1.

5 Experiments

Datasets and models. To implement a non-iid scenario, we adopt a close-to-reality setting where users' datasets are sampled from different distributions. We used two multi-domain datasets for the setting. The first is a subset (30%) of DIGITS, a benchmark for domain adaption (Peng et al. 2019b). DIGITS has 28×28 images and serves as a commonly used benchmark for FL (Caldas et al. 2019; McMahan et al. 2017; Li et al. 2020a). DIGITS includes 5 different domains: MNIST (MM) (Lecun et al. 1998), SVHN (SV) (Netzer et al. 2011), USPS (US) (Hull 1994), SynthDigits (SY) (Ganin and Lempitsky 2015), and MNIST-M (MM) (Ganin and Lempitsky 2015). The second dataset is DOMAINNET (Peng et al. 2019a) processed by (Li et al. 2020b), which contains 6 distinct domains of large-size 256×256 real-world images: Clipart (C), Infograph (I), Painting (P), Quickdraw (Q), Real (R), Sketch (S). For DIGITS, we use a convolutional network with BN (or DBN) layers following each conv or linear layers. For the large-sized DOMAINNET, we use AlexNet (Krizhevsky, Sutskever, and Hinton 2012) extended with BN layers after each convolutional or linear layer following prior non-iid FL practice (Li et al. 2020b).

Training and evaluation. For AT users, we use n-step PGD (projected gradient descent) attack (Madry et al. 2018) with a constant noise magnitude ϵ. Following (Madry et al. 2018), we use $\epsilon = 8/255$, $n = 7$, and attack inner-loop step size $2/255$, for training, validation, and test. We uniformly split the dataset for each domain into 10 subsets for



(a) FRP from a single AT domain

(b) FRP from partial AT users per domain

Figure 2: Evaluating FRP performance with different FRP settings.

DIGITS and 5 for DOMAINNET, following (Li et al. 2020b), which are distributed to different users, respectively. Accordingly, we have 50 users for DIGITS and 30 for DOMAINNET. Each user trains local model for one epoch per communication round. We evaluate the federated performance by standard accuracy (SA), classification accuracy on the clean test set, and robust accuracy (RA), classification accuracy on adversarial images perturbed from the original test set. All metric values are averaged over users. We defer other details of experimental setup such as hyper-parameters to appendix, and focus on discussing the results.

λ	test BN	weight	Digits			DomainNet		
			All RA	20% RA	MNIST RA	All RA	20% RA	Real RA
0	BN$_c$		52.8	41.9	34.6	35.5	**22.1**	15.4
0	tran. BN$_a$	uni	**62.0**	50.6	**41.5**	**35.7**	19.8	13.2
0	tran. BN$_a$	cos	**62.0**	**51.0**	**41.5**	**35.7**	21.4	12.8
0.5	BN$_c$		52.8	50.0	42.2	35.5	26.5	21.0
0.5	tran. BN$_a$	uni	**62.0**	55.4	51.5	**35.7**	27.5	**26.4**
0.5	tran. BN$_a$	cos	**62.0**	**55.8**	**58.5**	**35.7**	**28.1**	**26.4**

Table 1: Ablation of different test-time BNs.

5.1 Comprehensive Study

To further understand the role of each component in FedRBN, we conduct a comprehensive study on its properties. In experiments, we use three representative federated baselines combined with AT: FedAvg (McMahan et al. 2017), FedProx (Li et al. 2020a), and FedBN (Li et al. 2020b). We use FATAvg to denote the AT-augmented FedAvg, and similarly FATProx and FATBN. To implement hardware heterogeneity, we let 20%-per-domain users from $3/5$ domains (of DIGITS) conduct AT.

Ablation Studies. We study how BN should be used at inference time when LBN and DBN are already integrated into federated training. Thus, we evaluate trained models with users' local BN$_c$ and BN$_a$ transmitted from the global estimation. We also compare two kinds of weighting strategy for estimating transferable BN$_a$ parameters: uniform weights (uni) or the proposed cosine-similarity-based weights for soruce users. In Table 1, we present the results with $\lambda \in \{0, 0.5\}$ for PNC losses. When $\lambda = 0$, we only share robustness through customizing BN$_a$ for each target ST user without PNC losses and the propagated BNs is more effective on the Digits than on DomainNet, because DomainNet is a more complicated task involving higher domain divergence. As the domain gap overwhelms the gap between adversarial samples and clean samples (also see representation comparison in appendix), the BN$_c$ outperforms the BN$_a$ surprisingly on RA. As we formerly discussed, the non-reducible domain gap in adversarial training motivates our development of PNC loss. With PNC loss ($\lambda = 0.5$), we significantly improves the robustness and accuracy and the performance approaches the all-AT results. In addition, either with or without PNC losses, the cos-weighting strategy consistently improves the robustness compared to non-informative uniform weights.

Impacts from data heterogeneity. To study the influence of different AT domains, we set up an experiment where AT users only reside on one single domain. For simplicity, we let each domain contains a single user as in (Li et al. 2020b) and utilize only 10% of DIGITS dataset. The single AT domain plays the central role in gaining robustness from adversarial augmentation and propagating to other domains. The task is hardened by the non-singleton of gaps between the AT domain and multiple ST domains and a lack of the knowledge of domain relations. Results in Fig. 2a show the superiority of the proposed FedRBN, which improves RA for more than 10% in all cases with small drops in SA. We see that RA is the worst when MNIST serves as the AT domain, whereas RA propagates better when the AT domain is SVHN or SynthDigits. A possible explanation is that SVHN and SynthDigits are more visually distinct than the rest domains, forming larger domain gaps.

Impacts from hardware heterogeneity. We vary the number of AT users in training from $1/N$ (most heterogeneous) to N/N (homogeneous) to compare the robustness gain. Fig. 2b shows that our method consistently improves the robustness. Even when all domains are noised, FedRBN is the best due to the use of DBN. When not all domains are AT, our method only needs half of the users to be noised such that the RA is close to the upper bound (fully noised case).

Other comprehensive studies in Appendix for interested

	LBN	DBN	Digits									DomainNet								
AT users			All			20%			MNIST			All			20%			Real		
Metrics			RA	SA	T	RA	SA	T	RA	SA	T	RA	SA	T	RA	SA	T	RA	SA	T
FedRBN $\lambda = 1$	✓	✓	**62.0**	84.9	7.4	**60.6**	86.5	2.5	**60.8**	83.9	2.5	**35.7**	**61.6**	127.9	27.6	56.0	42.6	**28.2**	58.3	39.1
FedRBN $\lambda = 0.5$	✓	✓	**62.0**	84.9	7.4	55.8	**87.3**	2.9	58.5	**86.5**	2.9	**35.7**	**61.6**	127.9	**28.1**	62.5	51.2	26.4	63.9	48.0
FedRBN $\lambda = 0$	✓	✓	**62.0**	84.9	7.4	51.0	83.5	2.2	41.5	80.2	2.2	**35.7**	**61.6**	127.9	21.4	62.5	38.4	12.8	56.1	34.6
FATAvg+DBN		✓	60.0	83.8	7.4	48.8	82.8	2.2	40.2	79.9	2.2	27.6	52.8	127.9	16.6	58.9	38.4	13.0	54.8	34.6
FATBN	✓		60.0	**87.3**	7.4	41.2	86.4	2.2	36.5	86.4	2.2	35.2	60.2	127.9	20.3	**63.2**	38.4	15.7	**64.7**	34.6
FATAvg			58.3	86.1	7.4	42.6	84.6	2.2	38.4	84.1	2.2	24.6	47.4	127.9	15.4	57.8	38.4	10.7	57.9	34.6
FATProx			58.5	86.3	7.4	42.8	84.5	2.2	38.1	84.1	2.2	24.8	47.1	127.9	14.5	57.3	38.4	10.4	57.1	34.6
FedRob			13.1	13.1	7.4	20.6	59.3	1032	17.7	48.9	645	-	-	-	-	-	-	-	-	-

Table 2: Benchmarks of robustness propagation, where we measure the per-epoch computation time (T) by counting $\times 10^{12}$ times of multiplication-or-add operations (MACs) to evaluate the efficiency.

readers, where we studied the λ-governed trade-off, the convergence curves, detailed ablation studies of FL.

5.2 Comparison to Baselines

To demonstrate the effectiveness of the proposed FedRBN, we compare it with baselines on two benchmarks. We repeat each experiment for three times with different seeds from $\{1, 2, 3\}$. We introduce two more baselines: a proposed baseline combining FATAvg with DBN, personalized meta-FL extended with FAT (FATMeta) (Fallah, Mokhtari, and Ozdaglar 2020) and federated robust training (FedRob) (Reisizadeh et al. 2020). Because FedRob requires a project matrix of the squared size of image and the matrix is up to $256^2 \times 256^2$ on DomainNet which does not fit into a common GPU, we exclude it from comparison. Given the same setting, we constrain the computation cost in the similar scale for cost-fair comparison. We evaluate methods on two FRP settings. **1) Propagate from a single domain**. In reality, a powerful computation center may join the FL with many other users, e.g., mobile devices. Therefore, the computation center is an ideal node for the computation-intensive AT. Due to limitations of data collection, the center may only have access to a single domain, resulting gaps to most other users. We evaluate how well the robustness can be propagated from the center to others. **2) Propagate from a few multi-domain AT users**. In this case, we assume that to reduce the total training time, ST users are exempted from the AT tasks in each domain. Thus, an ST user wants to gain robustness from other same-domain users, but the different-domain users may hinder the robustness due to the domain gaps in adversarial samples.

Benchmark. Table 2 shows that our method outperforms all baselines for all tasks, while it associates to only small overhead (for optimizing PNC losses) compared to the full-AT case. Importantly, we show that only 20% users and less than 33% time complexity of the full-AT setting are enough to achieve robustness comparable to the best fully-trained baseline. Contradicting FATAvg+DBN and FATBN confirmed the importance of DBN in robustness but also show its limitation on handling data heterogeneity. Thus, FedRBN ($\lambda = 0$) is proposed to simultaneously address data and hardware heterogeneity by efficiently propagating robustness through BNs. To fully exploit the robustness complying users' hardware limitations, the PNC loss ($\lambda > 0$) is used and improves the robustness significantly. When $\lambda = 1$, the trained inclines to be more robust but less accurate on clean samples. Instead, $\lambda = 0.5$ provides a fairly nice trade-off between accuracy and robustness, for which we use the parameter generally.

Attack	PGD	PGD	MIA	MIA	AA	LSA	SA
$(n,\ \epsilon)$	(20,16)	(100,8)	(20,16)	(100,8)	(-, 8)	(7, -)	-
FedRBN	**42.8**	**54.5**	**39.9**	**52.2**	**48.3**	**73.5**	84.2
FATBN	28.6	41.6	27.0	39.7	31.0	64.0	**84.6**
FATAvg	31.5	43.4	30.0	41.5	32.9	63.3	84.2

Table 3: Evaluation of RA with various attacks on Digits. n and ϵ are the step number and the magnitude of attack.

Stronger attacks. To fully evaluate the robustness, we experiment with more attack methods, including MIA (Dong et al. 2018), AutoAttack (AA) (Croce and Hein 2020) and LSA (Narodytska and Kasiviswanathan 2016). Even evaluated by different attacks (see Table 3), our method still outperforms others. Especially, a strong score-based blackbox attacks such as Square Attack (Andriushchenko et al. 2020) (included in AA) can avoid the trip fake robustness.

6 Conclusion

In this paper, we investigate a novel problem setting, federate propagating robustness, and propose a FedRBN algorithm that transfers robustness in FL through robust BN statistics. Supplementary is at https://arxiv.org/abs/2106.10196.

Acknowledgements

This material is based in part upon work supported by the National Science Foundation (IIS-1749940, IIS-2212174, ECCS-2024270), NIH/National Institute on Aging (1RF1AG072449) and Office of Naval Research (N00014-20-1-2382).

References

Andriushchenko, M.; Croce, F.; Flammarion, N.; and Hein, M. 2020. Square Attack: a query-efficient black-box adversarial attack via random search. *arXiv:1912.00049 [cs, stat]*.

Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; and Shmatikov, V. 2018. How To Backdoor Federated Learning. *International Conference on Machine Learning*.

Bhagoji, A. N.; Chakraborty, S.; Mittal, P.; and Calo, S. 2018. Analyzing Federated Learning through an Adversarial Lens. In *International Conference on Machine Learning*.

Blanchard, P.; El Mhamdi, E. M.; Guerraoui, R.; and Stainer, J. 2017. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. *Advances in Neural Information Processing Systems*, 30: 119–129.

Caldas, S.; Duddu, S. M. K.; Wu, P.; Li, T.; Konečný, J.; McMahan, H. B.; Smith, V.; and Talwalkar, A. 2019. LEAF: A Benchmark for Federated Settings. *arXiv:1812.01097 [cs, stat]*.

Chen, C.; Liu, Y.; Ma, X.; and Lyu, L. 2022. CalFAT: Calibrated Federated Adversarial Training with Label Skewness. *arXiv preprint arXiv:2205.14926*.

Chen, Y.; Su, L.; and Xu, J. 2017. Distributed Statistical Machine Learning in Adversarial Settings: Byzantine Gradient Descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2): 44:1–44:25.

Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, 2206–2216. PMLR.

Diao, E.; Ding, J.; and Tarokh, V. 2021. HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients. In *International Conference on Learning Representations*.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting Adversarial Attacks with Momentum. *arXiv:1710.06081 [cs, stat]*.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*, Lecture Notes in Computer Science, 265–284. Springer Berlin Heidelberg.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. Personalized Federated Learning: A Meta-Learning Approach. In *Advances in Neural Information Processing Systems*.

Fang, M.; Cao, X.; Jia, J.; and Gong, N. 2020. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 1605–1622.

Fu, S.; Xie, C.; Li, B.; and Chen, Q. 2019. Attack-Resistant Federated Learning with Residual-based Reweighting. *arXiv preprint arXiv:1912.11464*.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised Domain Adaptation by Backpropagation. In *International Conference on Machine Learning*, 1180–1189. PMLR.

Geiping, J.; Bauermeister, H.; Dröge, H.; and Moeller, M. 2020. Inverting Gradients – How easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems*.

Hard, A.; Rao, K.; Mathews, R.; Ramaswamy, S.; Beaufays, F.; Augenstein, S.; Eichner, H.; Kiddon, C.; and Ramage, D. 2019. Federated Learning for Mobile Keyboard Prediction. *arXiv:1811.03604 [cs]*.

Hendrycks, D.; Lee, K.; and Mazeika, M. 2019. Using Pre-Training Can Improve Model Robustness and Uncertainty. In *International Conference on Machine Learning*.

Hull, J. J. 1994. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5): 550–554.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*.

Kerkouche, R.; Ács, G.; and Castelluccia, C. 2020. Federated Learning in Adversarial Settings. *arXiv:2010.07808 [cs]*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, 1097–1105. Red Hook, NY, USA: Curran Associates Inc.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020a. Federated Optimization in Heterogeneous Networks. In *Conference on Systems and Machine Learning Foundation (MLSys)*.

Li, X.; Jiang, M.; Zhang, X.; Kamp, M.; and Dou, Q. 2020b. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization. In *International Conference on Learning Representations*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. *International Conference on Learning Representations*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*, 1273–1282.

Müller, R.; Kornblith, S.; and Hinton, G. E. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Narodytska, N.; and Kasiviswanathan, S. P. 2016. Simple Black-Box Adversarial Perturbations for Deep Networks. *arXiv:1612.06299 [cs, stat]*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019a. Moment Matching for Multi-Source Domain Adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1406–1415.

Peng, X.; Huang, Z.; Zhu, Y.; and Saenko, K. 2019b. Federated Adversarial Domain Adaptation. In *International Conference on Learning Representations*.

Pillutla, K.; Kakade, S. M.; and Harchaoui, Z. 2020. Robust Aggregation for Federated Learning. In *IEEE Transactions on Signal Processing*.

Reisizadeh, A.; Farnia, F.; Pedarsani, R.; and Jadbabaie, A. 2020. Robust Federated Learning: The Case of Affine Distribution Shifts. In *Advances in Neural Information Processing Systems*.

Shafahi, A.; Najibi, M.; Ghiasi, A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L. S.; Taylor, G.; and Goldstein, T. 2019a. Adversarial Training for Free! *Advances in Neural Information Processing Systems*.

Shafahi, A.; Saadatpanah, P.; Zhu, C.; Ghiasi, A.; Studer, C.; Jacobs, D.; and Goldstein, T. 2019b. Adversarially robust transfer learning. In *International Conference on Learning Representations*.

Shah, D.; Dube, P.; Chakraborty, S.; and Verma, A. 2021. Adversarial training in communication constrained federated learning. *arXiv:2103.01319 [cs]*.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness May Be at Odds with Accuracy. *International Conference on Learning Representations*.

Wang, H.; Chen, T.; Gui, S.; Hu, T.-K.; Liu, J.; and Wang, Z. 2020. Once-for-All Adversarial Training: In-Situ Tradeoff between Robustness and Accuracy for Free. *Advances in Neural Information Processing Systems*.

Wu, Z.; Ling, Q.; Chen, T.; and Giannakis, G. B. 2020. Federated Variance-Reduced Stochastic Gradient Descent With Robustness to Byzantine Attacks. *IEEE Transactions on Signal Processing*, 68: 4583–4596.

Xie, C.; Tan, M.; Gong, B.; Wang, J.; Yuille, A.; and Le, Q. V. 2020. Adversarial Examples Improve Image Recognition. *CVPR*.

Xie, C.; and Yuille, A. 2019. Intriguing properties of adversarial training at scale. *International Conference on Learning Representations*.

Xiong, Z.; Cai, Z.; Hu, C.; Takabi, D.; and Li, W. 2022. Towards neural network-based communication system: attack and defense. *IEEE Transactions on Dependable and Secure Computing*.

Xiong, Z.; Cai, Z.; Takabi, D.; and Li, W. 2021. Privacy threat and defense for federated learning with non-iid data in AIoT. *IEEE Transactions on Industrial Informatics*, 18(2): 1310–1321.

Yin, D.; Chen, Y.; Kannan, R.; and Bartlett, P. 2018. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *International Conference on Machine Learning*, 5650–5659. PMLR.

Yu, T.; Bagdasaryan, E.; and Shmatikov, V. 2020. Salvaging Federated Learning by Local Adaptation. *arXiv:2002.04758 [cs, stat]*.

Zhang, D.; Zhang, T.; Lu, Y.; Zhu, Z.; and Dong, B. 2019. You Only Propagate Once: Accelerating Adversarial Training via Maximal Principle. *Advances in Neural Information Processing Systems*, 12.

Zizzo, G.; Rawat, A.; Sinn, M.; and Buesser, B. 2020. FAT: Federated Adversarial Training. *arXiv:2012.01791 [cs]*.