# Exploit Gradient Skewness to Circumvent Byzantine Defenses for Federated Learning

**Yuchen Liu**[12*†], **Chen Chen**[3*], **Lingjuan Lyu**[3‡], **Yaochu Jin**[4], **Gang Chen**[12]

[1]The State Key Laboratory of Blockchain and Data Security, Zhejiang University
[2]Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security
[3]Sony AI
[4]Westlake University, China
yuchen.liu.a@gmail.com, {ChenA.Chen, lingjuan.lv}@sony.com, jinyaochu@westlake.edu.cn, cg@zju.edu.cn,

## Abstract

Federated Learning (FL) is notorious for its vulnerability to Byzantine attacks. Most current Byzantine defenses share a common inductive bias: among all the gradients, the densely distributed ones are more likely to be honest. However, such a bias is a poison to Byzantine robustness due to a newly discovered phenomenon in this paper – gradient skew. We discover that a group of densely distributed honest gradients skew away from the optimal gradient (the average of honest gradients) due to heterogeneous data. This gradient skew phenomenon allows Byzantine gradients to hide within the densely distributed skewed gradients. As a result, Byzantine defenses are confused into believing that Byzantine gradients are honest. Motivated by this observation, we propose a novel skew-aware attack called STRIKE: first, we search for the skewed gradients; then, we construct Byzantine gradients within the skewed gradients. Experiments on three benchmark datasets validate the effectiveness of our attack.

**Code** — https://github.com/YuchenLiu-a/byzantine_skew

## 1 Introduction

Federated Learning (FL) (McMahan et al. 2017; Li et al. 2020) emerged as a privacy-aware learning paradigm, in which data owners, i.e., clients, repeatedly use their private data to compute local gradients and upload them to a central server. The central server collects the uploaded gradients from clients and aggregates these gradients to update the global model. In this way, clients can collaborate to train a model without exposing their private data.

Unfortunately, FL is susceptible to Byzantine attacks due to its distributed nature (Blanchard et al. 2017; Guerraoui, Rouault et al. 2018). A malicious party can control a small subset of clients, i.e., Byzantine clients, to degrade the utility of the global model. During the training phase, Byzantine clients can send arbitrary messages to the central server to bias the global model. A wealth of defenses (Blanchard et al. 2017; Pillutla, Kakade, and Harchaoui 2019; Shejwalkar and Houmansadr 2021) have been proposed to defend against Byzantine attacks in FL. They aim to estimate the optimal gradient, i.e., the average of gradients from honest clients, in the presence of Byzantine clients.

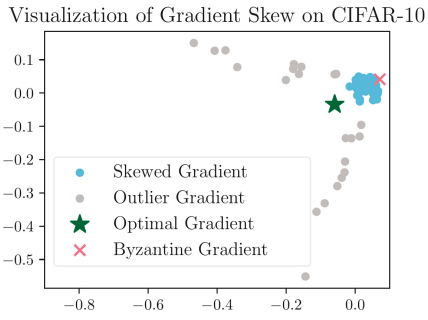



Figure 1: The LLE visualization of honest gradients in the non-IID setting on CIFAR-10. Substantial honest gradients (blue circles) are skewed away from the optimal gradient (green star). In this case, we can hide Byzantine gradients (pink crosses) within the skewed honest gradients to circumvent defenses.

Most existing defenses (Blanchard et al. 2017; Shejwalkar and Houmansadr 2021; Karimireddy, He, and Jaggi 2022) share a common inductive bias: the densely distributed gradients are more likely to be honest. Generally, they assign higher weights to the densely distributed gradients. Then they compute the global gradient and use it to update the global model. As a result, the output global gradient of defenses is biased towards the densely distributed of gradients.

However, this inductive bias of Byzantine defenses is harmful to Byzantine robustness in FL due to the presence of gradient skew. In practical FL, data across different clients is non-independent and identically distributed (non-IID), which gives rise to heterogeneous honest gradients (McMahan et al. 2017; Li et al. 2020; Karimireddy, He, and Jaggi 2022). On closer inspection, we find that the distribution these heterogenous honest gradients are highly skewed. In

*These authors contributed equally.
†Work done during an internship at Sony AI.
‡Corresponding author.

Figure 1, we use Locally Linear Embedding (LLE) (Roweis and Saul 2000) to visualize the honest gradients on CIFAR-10 dataset (Krizhevsky and Hinton 2009) when data is non-IID split. Detailed setups and more results are provided in Apendix A. As shown in Figure 1, a group of densely distributed gradients skews away from the optimal gradient. We term this phenomenon as "gradient skew". When honest gradients are skewed, the defenses' bias towards densely distributed gradients is a poison to Byzantine robustness. In fact, we can hide Byzantine gradients within the skewed densely distributed honest gradients as shown in Figure 1. In this case, the bias of defenses would drive the global gradient close to the skewed gradients but far from the optimal gradient.

In this paper, we study how to exploit the gradient skew in the more practical non-IID setting to circumvent Byzantine defenses. We first observe the gradient skew phenomenon in the non-IID setting and explore its vulnerability. Motivated by the above observation, we design a novel two-Stage aTtack based on gRadIent sKEw called STRIKE. In particular, STRIKE hides Byzantine gradients within the skewed honest gradients as shown in Figure 1. STRIKE can take advantage of the gradient skew in FL to break Byzantine defenses.

In summary, our contributions are:

- To the best of our knowledge, we are the first to discover the gradient skew phenomenon in FL: a group of densely distributed gradients is skewed away from the optimal gradient. Motivated by the observation, we design an attack principle that can circumvent Byzantine defenses under gradient skew: hide Byzantine gradients within the skewed honest gradients.
- Based on the above attack principle, we propose a two-stage Byzantine attack called STRIKE. In the first stage, STRIKE searches for the skewed honest gradients under the guidance of Karl Pearson's formula. In the second stage, STRIKE constructs the Byzantine gradients within the skewed honest gradients by solving a constrained optimization problem.
- Experiments on three benchmark datasets validate the effectiveness of the proposed attack. For instance, STRIKE attack improves upon the best baseline by 57.84% against DnC on FEMNIST dataset when there are 20% Byzantine clients.

## 2 Related Works

**Byzantine attacks.** Blanchard et al. first disclose the Byzantine vulnerability of FL. Baruch, Baruch, and Goldberg observe that the variance of honest gradients is high enough for Byzantine clients to compromise Byzantine defenses. Based on this observation, they propose a LIE attack that hides Byzantine gradients within the variance. Xie, Koyejo, and Gupta further utilize the high variance and propose an IPM attack. Particularly, they show that when the variance of honest gradients is large enough, IPM can make the inner product between the aggregated gradient and the honest average negative. However, this result is restricted to a few defenses, i.e., Median (Yin et al. 2018), Trmean (Yin et al. 2018), and Krum (Blanchard et al. 2017). Fang et al. establish an omniscient attack called Fang. However, the Fang attack requires knowledge of the Byzantine defense, which is unrealistic in practice. Shejwalkar and Houmansadr propose Min-Max and Min-Sum attacks that solve a constrained optimization problem to determine Byzantine gradients. From a high level, both Min-Max and Min-Sum aim to maximize the perturbation to a reference benign gradient while ensuring the Byzantine gradients lie within the variance. Karimireddy, He, and Jaggi propose a Mimic attack that takes advantage of data heterogeneity in FL. In particular, Byzantine clients pick an honest client to mimic and copy its gradient. The above attacks take advantage of the large variance of honest gradients to break Byzantine defenses. However, they all ignore the skewed nature of honest gradients in FL and fail to exploit this vulnerability.

**Byzantine resilience.** El-Mhamdi et al.; Karimireddy, He, and Jaggi provide state-of-the-art theoretical analysis of Byzantine resilience under data heterogeneity. El-Mhamdi et al. discuss Byzantine resilience in a decentralized, asynchronous setting. Farhadkhani et al. provide a unified framework for Byzantine resilience analysis, which enables comparison among different defenses on a common theoretical ground. Karimireddy, He, and Jaggi improve the error bound of Byzantine resilience to be upper-bounded by the fraction of Byzantine clients, which recovers the standard convergence rate when there are no Byzantine clients. Allouah et al. tightly analyzing the impact of client subsampling and local steps. Yan et al. utilizes the correlation of clients' performance over multiple iterations to evaluate the reliability of clients. They all share a common bias: densely distributed gradients are more likely to be honest. However, this bias is a poison to Byzantine robustness in the presence of gradient skew. In practical FL, the distribution of honest gradients is highly skewed due to data heterogeneity. Therefore, existing defenses are especially vulnerable to attacks that are aware of gradient skew.

**Data heterogeneity.** Yu, Yang, and Zhu first proposed to measure data heterogeneity by gradient divergence, which describes the difference between the local gradients and the global one. Karimireddy et al. proposed a more general version of gradient divergence - gradient dissimilarity. To the best of our knowledge, these are the only metrics of heterogeneity from a gradient distribution perspective (Li et al. 2019; Woodworth, Patel, and Srebro 2020). Luo et al. find that such difference mainly involves neural network prediction heads. For label skewness, a particular type of heterogeneity, label distribution discrepancy is used to measure heterogeneity (Peng et al. 2024). However, no existing work noticed that such gradient divergence is skewed - a group of densely distributed local gradients skew away from the global gradient, i.e., the gradient skew introduced in Section 4.

## 3 Notations and Preliminary

### 3.1 Notations

$\|\cdot\|$ denotes the $\ell_2$ norm of a vector. For vector $\boldsymbol{v}$, $(\boldsymbol{v})_k$ represents the $k$-th coordinate of $\boldsymbol{v}$. Model parameters are denoted

by $\boldsymbol{w}$ and gradients are denoted by $\boldsymbol{g}$. We use $\bar{\boldsymbol{g}}$ to denote the optimal gradient, i.e., the average of honest gradients, and $\hat{\boldsymbol{g}}$ denotes the global gradients obtained by Byzantine defenses. We use subscript $i$ to denote client $i$ and use superscript $t$ to denote communication round $t$.

### 3.2 Preliminary

**Federated learning.** Suppose that there are $n$ clients and a central server. The goal is to optimize the global loss function $\mathcal{L}(\cdot)$:

$$\min_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}), \quad \text{where } \mathcal{L}(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_i(\boldsymbol{w}). \tag{1}$$

Here $\boldsymbol{w}$ is the model parameter, and $\mathcal{L}_i(\cdot)$ is the local loss function on client $i$ for $i = 1, \ldots, n$.

In communication round $t$, the central server distributes global parameter $\boldsymbol{w}^t$ to the clients. Each client $i$ performs several epochs of SGD to minimize its local loss function $\mathcal{L}_i(\cdot)$ and update its local parameter to $\boldsymbol{w}_i^{t+1}$. Then, each client $i$ computes its local gradient $\boldsymbol{g}_i^t$ and sends it to the server.

$$\boldsymbol{g}_i^t = \boldsymbol{w}_i^t - \boldsymbol{w}_i^{t+1}, \quad i = 1, \ldots, n. \tag{2}$$

After receiving the uploaded local gradients, the server aggregates the local gradients and updates the global model to $\boldsymbol{w}^{t+1}$.

$$\bar{\boldsymbol{g}}^t = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}_i^t, \quad \boldsymbol{w}^{t+1} = \boldsymbol{w}^t - \bar{\boldsymbol{g}}^t. \tag{3}$$

**Byzantine attack model.** Assume that among the total $n$ clients, $f$ fixed clients are Byzantine clients. Let $\mathcal{B} \subseteq \{1, \ldots, n\}$ denote the set of Byzantine clients and $\mathcal{H} = \{1, \ldots, n\} \setminus \mathcal{B}$ denote the set of honest clients. In each communication round, Byzantine clients can send arbitrary messages to bias the global model. The local gradients that the server receives in the $t$-th communication round are

$$\boldsymbol{g}_i^t = \begin{cases} *, & i \in \mathcal{B}, \\ \boldsymbol{w}^t - \boldsymbol{w}_i^{t+1}, & i \in \mathcal{H}, \end{cases} \tag{4}$$

where $*$ represents an arbitrary message. Following (Baruch, Baruch, and Goldberg 2019; Xie, Koyejo, and Gupta 2020), we consider the setting where the attacker only has the knowledge of honest gradients.

## 4 Gradient Skew in FL Due to Non-IID data

Plenty of works (Baruch, Baruch, and Goldberg 2019; Xie, Koyejo, and Gupta 2020; Karimireddy, He, and Jaggi 2022) have explored how large variance can be harmful to Byzantine robustness. However, to the best of our knowledge, none of the existing works is aware of the skewed nature of honest gradients in the non-IID setting and how gradient skew can threaten Byzantine robustness.

We take a close look at the distribution of honest gradients in the non-IID setting (without attack). To construct our FL setup, we split CIFAR-10 (Krizhevsky and Hinton 2009) dataset in a non-IID manner among 100 clients. For more setup details, please refer to Apendix A.1. We run FedAvg (McMahan et al. 2017) for 200 communication rounds. We randomly sample six communication rounds and use Locally Linear Embedding (LLE) (Roweis and Saul 2000) to visualize the gradients in these communication rounds in Figure 2. From Figure 2, we observe that a group of densely distributed honest gradients (blue circles) skew away from the optimal gradient (green stars). We call these blue circles *"skewed gradients"* and name this phenomenon *"gradient skew"*. We provide visualization results on more datasets in Apendix A.2. From visualization results on different datasets, we observe that the "gradient skew" phenomenon is prevalent across different datasets.

## 5 Circumvent Robust AGRs under Gradient Skew

Inspired by the above observation of the gradient skew phenomenon, we design a novel attack principle that can exploit this phenomenon to circumvent robust AGRs – *hide Byzantine gradients in the densely distributed skewed gradients*.

A body of recent works (Farhadkhani et al. 2022; Karimireddy, He, and Jaggi 2022; Allouah et al. 2023) have formulated Byzantine resilience for general robust AGRs. These formulations commonly show that Byzantine defenses inherently trust densely distributed gradients, regarding them as honest. We take the definition of $(f, \kappa)$-robustness in (Allouah et al. 2023) as an example.

**Definition 1** ($(f, \kappa)$-robustness)**.** Let $f < n/2$ and $\kappa \geq 0$, a robust AGR $\mathcal{A}$ is called $(f, \kappa)$-robust] if for any input $\{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n\}$ and any set $\mathcal{G} \subseteq \{1, \ldots, n\}$ of size $n - f$, the output $\hat{\boldsymbol{g}}$ of AGR $\mathcal{A}$ satisfies:

$$\|\mathcal{A}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n) - \bar{\boldsymbol{g}}_{\mathcal{G}}\|^2 \leq \frac{\kappa}{n-f}\sum_{i \in \mathcal{S}} \|\boldsymbol{g}_i - \bar{\boldsymbol{g}}_{\mathcal{G}}\|^2, \quad \text{where} \quad \bar{\boldsymbol{g}}_{\mathcal{G}} = \sum_{i \in \mathcal{G}} \boldsymbol{g}_i/(n-f). \tag{5}$$

In the definition, the distance between the aggregated gradients $\hat{\boldsymbol{g}}$ and average candidate gradient $\bar{\boldsymbol{g}}_{\mathcal{G}}$ is upper-bounded by $\sum_{i \in \mathcal{S}} \|\boldsymbol{g}_i - \bar{\boldsymbol{g}}_{\mathcal{G}}\|^2$, which measures the distribution density of gradients. This means that the aggregated gradient is biased to densely distributed gradients. In other words, *Byzantine defenses believe that the most densely distributed $n - f$ gradients are more likely to be the honest ones*.

This inductive bias can be exploited by a malicious party when gradient skew exists. In particular, we propose to *hide Byzantine gradients in the skewed gradients*, i.e., place pink cross within blue dots as shown in Figure 1, to fake Byzantine gradients as honest ones. As shown in Figure 1, Byzantine gradients and skew gradients, i.e., pink cross and blue dots, are densely distributed. As a result, they would be mistaken as honest gradients by Byzantine defenses.

This attack strategy enjoys another advantage. It tricks Byzantine defense into thinking other outlier honest gradients, i.e., gray circles in Figure 1, are malicious. As a result, these outlier gradients would be assigned lower weights or even removed from aggregation. Unfortunately, outlier gradients are crucial to improving the generalization performance of the final FL model (Yan et al. 2024). As a result,

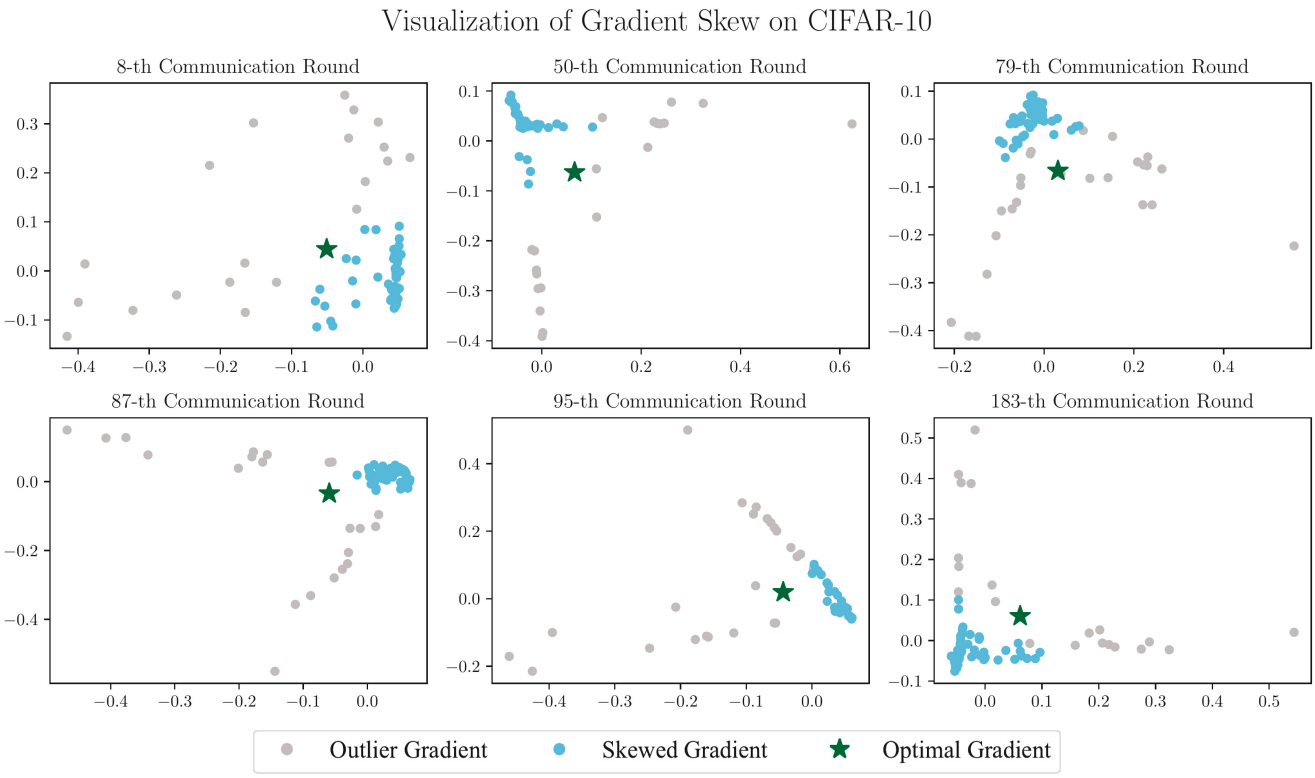



Figure 2: Visualization of gradient skew on CIFAR-10 dataset. As shown in the figures, the optimal gradients (green stars) deviate from the densely distributed gradients.

this attack strategy can pose a significant threat to model performance.

## 6 Proposed Attack

In this section, we introduce the proposed two-Stage aTtack based on gRadIent sKEw called STRIKE. As discussed in the previous section, the attack principle is to *hide Byzantine gradients within the skewed gradients*. To achieve this goal, we carry out STRIKE attack in two stages: in the first stage, we search for the skewed honest gradients; in the second stage, we construct Byzantine gradients within the skewed honest gradients found in the first stage. The procedure of STRIKE attack is shown in Agorithm 1 in Apendix B.

**Search for the skewed honest gradients.** To hide the Byzantine gradient in the skewed honest gradients, we first need to find the skewed honest gradients. We perform a heuristic search motivated by Karl Pearson's formula (Knoke, Bohrnstedt, and Mee 2002; Moore, Mccabe, and Craig 2009). Figure 3a illustrates the search procedure in this stage.

As visualized in Figure 1, skewed honest gradients are densely distributed. As the population mode (in statistics) falls where the probability density is highest, the skewed honest gradients coincide with the (population) mode. Thus, we can *identify honest gradients near the mode as skewed gradients*.

Karl Pearson's formula (Knoke, Bohrnstedt, and Mee 2002; Moore, Mccabe, and Craig 2009) implies that the mode and median lie on the same side of the mean. Therefore, the search for the skewed gradients starts from the mean and advances towards the median. That is, as shown in Figure 3a we search for the skewed honest gradients along the direction $\boldsymbol{u}_{\text{search}}$ defined as:

$$\boldsymbol{u}_{\text{search}} = \boldsymbol{g}_{\text{med}} - \bar{\boldsymbol{g}}, \tag{6}$$

where $\boldsymbol{g}_{\text{med}}$ is the coordinate-wise median of honest gradients $\{\boldsymbol{g}_i \mid i \in \mathcal{H}\}$, i.e., the $k$-th coordinate of $\boldsymbol{g}_{\text{med}}$ is $(\boldsymbol{g}_{\text{med}})_k = \text{median}\{(\boldsymbol{g}_i)_k \mid i \in \mathcal{H}\}$, median$\{\cdot\}$ returns the median of the input numbers, and $\bar{\boldsymbol{g}} = \sum_{i\in\mathcal{H}} \boldsymbol{g}_i/(n-f)$ is the average of honest gradients.

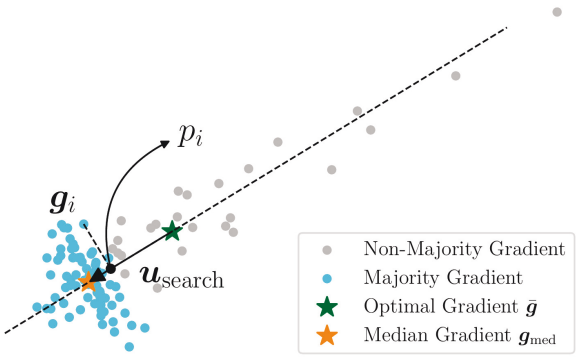
For each honest gradient $\boldsymbol{g}_i$, we compute its scalar projection $p_i$ on the searching direction $\boldsymbol{u}_{\text{search}}$:

$$p_i = \langle \boldsymbol{g}_i, \frac{\boldsymbol{u}_{\text{search}}}{\|\boldsymbol{u}_{\text{search}}\|} \rangle, \quad \forall i \in \mathcal{H}, \tag{7}$$

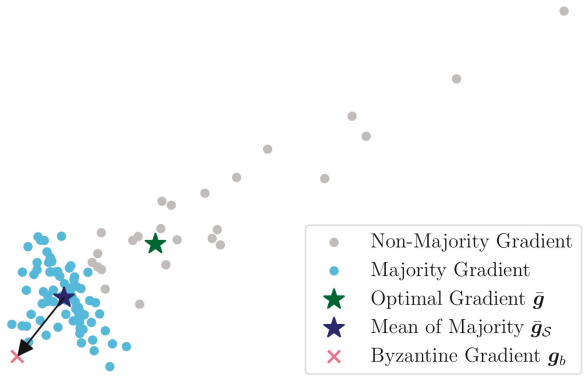
where $\langle \cdot, \cdot \rangle$ represents the inner product. The $n-2f$ gradients with the highest scalar projection values are identified as the skewed honest gradients. The goal is to have AGR consider the selected $n-2f$ gradients as honest and the unselected $f$ gradients as Byzantine. Let $\mathcal{S}$ denote the index set, that is

$$\mathcal{S} = \text{Set of } (n-2f) \text{ indices of the gradients with the highest scalar projection } p_i, \tag{8}$$

then the skewed honest gradients are $\{\boldsymbol{g}_i \mid i \in \mathcal{S}\}$.




(a) We search along the direction $\boldsymbol{u}_{\text{search}} = \boldsymbol{g}_{\text{med}} - \bar{\boldsymbol{g}}$. The honest gradients with the largest scalar projection $p_i$ are selected as the skewed honest gradients (blue circles).




(b) We start from the average of skewed honest gradients $\bar{\boldsymbol{g}}_{\mathcal{S}}$ (dark blue star) and select $\alpha$ such that Byzantine gradient $\boldsymbol{g}_b$ (pink cross) lies within the skewed honest gradients.

Figure 3: Illustration of the proposed two-stage attack STRIKE: in the first stage, STRIKE searches for the skewed honest gradients; in the second stage, STRIKE hides Byzantine gradients within the skewed honest gradients.

**Hide Byzantine gradients within the skewed honest gradients.** In this stage, we aim to hide Byzantine gradients $\{\boldsymbol{g}_i \mid i \in \mathcal{B}\}$ within the skewed honest gradients $\{\boldsymbol{g}_i \mid i \in \mathcal{S}\}$ identified in stage 1. The primary goal of our attack is to disguise Byzantine gradients and the skewed honest gradients $\{\boldsymbol{g}_i \mid i \in \mathcal{B} \cup \mathcal{S}\}$ as honest gradients. Meanwhile, the secondary goal is to maximize the attack effect, i.e., maximize the distance between these "fake" honest gradients and the optimal gradient. The hiding procedure in this stage is illustrated in Figure 3b.

According to (Farhadkhani et al. 2022), robust AGRs are sensitive to the diameter of gradients. Therefore, we ensure that the Byzantine gradients $\boldsymbol{g}_b$ lie within the diameter of the skewed honest gradients $\boldsymbol{g}_s$ in order not to be detected.

$$\|\boldsymbol{g}_b - \boldsymbol{g}_s\| \leq \max_{i,j \in \mathcal{S}} \|\boldsymbol{g}_i - \boldsymbol{g}_j\|, \quad \forall b \in \mathcal{B}, s \in \mathcal{S}, \tag{9}$$

where $\mathcal{B}$ is the index set of Byzantine clients, $\mathcal{S}$ is the index set of skewed honest clients.

Meanwhile, we want to maximize the attack effect. As Byzantine defenses assume densely distributed gradients, i.e., the skewed honest gradients $\{\boldsymbol{g}_s \mid s \in \mathcal{S}\}$ and Byzantine gradients $\{\boldsymbol{g}_b \mid s \in \mathcal{B}\}$, to be honest. The aggregated gradients would be close to the mean of densely distributed gradients $\bar{\boldsymbol{g}}_{\mathcal{S}\cup\mathcal{B}} = \sum_{i\in\mathcal{S}\cup\mathcal{B}} \boldsymbol{g}_i/(n-f)$. Therefore, we maximize the distance between $\bar{\boldsymbol{g}}_{\mathcal{S}\cup\mathcal{B}}$ and the optimal gradient $\bar{\boldsymbol{g}}$ to maximize the attack effect.

$$\max_{\{\boldsymbol{g}_b | b\in\mathcal{B}\}} \|\bar{\boldsymbol{g}}_{\mathcal{S}\cup\mathcal{B}} - \bar{\boldsymbol{g}}\|. \tag{10}$$

Combining Equation (9) and Equation (10), our objective can be formulated as the following constrained optimization problem.

$$\begin{aligned} &\max_{\{\boldsymbol{g}_b | b\in\mathcal{B}\}} \|\bar{\boldsymbol{g}}_{\mathcal{S}\cup\mathcal{B}} - \bar{\boldsymbol{g}}\| \\ \text{s.t.} \quad &\bar{\boldsymbol{g}}_{\mathcal{S}\cup\mathcal{B}} = \sum_{i\in\mathcal{S}\cup\mathcal{B}} \boldsymbol{g}_i/(n-f) \\ &\|\boldsymbol{g}_b - \boldsymbol{g}_s\| \leq \max_{i,j\in\mathcal{S}} \|\boldsymbol{g}_i - \boldsymbol{g}_j\|, \forall b \in \mathcal{B}, s \in \mathcal{S} \end{aligned} \tag{11}$$

Equation (11) is too complex to be solved due to the high complexity of its feasible region. Therefore, we restrict $\{\boldsymbol{g}_b \mid b \in \mathcal{B}\}$ to the following form:

$$\boldsymbol{g}_b = \bar{\boldsymbol{g}}_{\mathcal{S}} + \alpha \cdot \text{sign}(\bar{\boldsymbol{g}}_{\mathcal{S}} - \bar{\boldsymbol{g}}) \odot \boldsymbol{\sigma}_{\mathcal{S}}, \quad \forall b \in \mathcal{B}, \tag{12}$$

where $\bar{\boldsymbol{g}}_{\mathcal{S}} = \sum_{i\in\mathcal{S}} \boldsymbol{g}_i/(n-2f)$ is the average of the skewed honest gradients, $\alpha$ is a non-negative real number that controls the attack strength, $\text{sign}(\cdot)$ returns the element-wise indication of the sign of a number, $\odot$ is the element-wise multiplication, and $\boldsymbol{\sigma}_{\mathcal{S}}$ is the element-wise standard deviation of skewed honest gradients $\{\boldsymbol{g}_i \mid i \in \mathcal{S}\}$. $\bar{\boldsymbol{g}}_{\mathcal{S}}$ lies within the feasible region of Equation (11), which ensures that $\{\boldsymbol{g}_b \mid b \in \mathcal{B}\}$ are feasible when $\alpha = 0$. $\text{sign}(\bar{\boldsymbol{g}}_{\mathcal{S}} - \bar{\boldsymbol{g}})$ controls the element-wise attack direction, and ensures that $\boldsymbol{g}_b$ is farther away from the optimal gradient $\bar{\boldsymbol{g}}$ under a larger $\alpha$. $\boldsymbol{\sigma}_{\mathcal{S}}$ controls the element-wise attack strength and ensures that Byzantine gradients are covert in each dimension.

With the restriction in Equation (12), Equation (11) can be simplified to the following optimization problem,

$$\begin{aligned} &\max \alpha \\ \text{s.t.} \quad &\|\bar{\boldsymbol{g}}_{\mathcal{S}} + \alpha \cdot \text{sign}(\bar{\boldsymbol{g}}_{\mathcal{S}}) \odot \boldsymbol{\sigma}_{\mathcal{S}} - \boldsymbol{g}_s\| \leq \max_{i,j\in\mathcal{S}} \|\boldsymbol{g}_i - \boldsymbol{g}_j\|, \\ &\forall s \in \mathcal{S}, \end{aligned} \tag{13}$$

which can be easily solved by the bisection method described in Apendix C. While $\alpha$ that solves Equation (13) is provable in most cases, we find in practice that a slightly adjusted attack strength can further improve the effect of STRIKE. We use an additional hyperparameter $\nu(>0)$ to control the attack strength of STRIKE. STRIKE sets $\boldsymbol{g}_b = \bar{\boldsymbol{g}}_{\mathcal{S}} + \nu\alpha \cdot \text{sign}(\bar{\boldsymbol{g}}_{\mathcal{S}}) \odot \boldsymbol{\sigma}_{\mathcal{S}} - \boldsymbol{g}_i$ for all $b \in \mathcal{B}$ and uploads Byzantine gradients to the server. Higher $\nu$ implies higher attack strength. We discuss the performance of STRIKE with different $\nu$ in Apendix D.2.

| CIFAR-10 | | | | | | | |
|---|---|---|---|---|---|---|---|
| Attack | Multi-Krum | Median | RFA | Aksel | CClip | DnC | RBTM |
| BitFlip | 54.76 ± 0.06 | 53.73 ± 2.05 | 56.04 ± 3.13 | 51.99 ± 2.04 | 54.44 ± 0.46 | 60.81 ± 0.56 | 55.21 ± 3.72 |
| LIE | 57.89 ± 0.22 | 49.20 ± 3.27 | 53.90 ± 5.43 | 46.73 ± 4.86 | 63.11 ± 0.43 | 61.58 ± 2.85 | 58.84 ± 0.64 |
| IPM | 47.55 ± 1.75 | 51.68 ± 1.85 | 55.36 ± 2.10 | 56.85 ± 2.07 | 58.75 ± 5.59 | 62.30 ± 3.60 | 48.43 ± 0.17 |
| MinMax | 59.44 ± 3.41 | 57.27 ± 0.63 | 60.20 ± 1.63 | 57.17 ± 5.50 | 59.38 ± 5.15 | 62.53 ± 2.67 | 57.72 ± 2.94 |
| MinSum | 55.47 ± 1.70 | 52.27 ± 0.53 | 54.59 ± 2.38 | 56.43 ± 1.74 | 54.70 ± 1.96 | 61.89 ± 1.62 | 46.78 ± 0.32 |
| Mimic | 56.00 ± 4.26 | 52.55 ± 0.89 | 53.61 ± 0.86 | 57.19 ± 2.50 | 51.00 ± 0.11 | 62.10 ± 5.22 | 46.77 ± 2.52 |
| STRIKE (Ours) | **42.90** ± 1.97 | **48.29** ± 0.40 | **52.92** ± 1.75 | **38.31** ± 0.47 | **50.67** ± 0.27 | **59.16** ± 1.84 | **44.82** ± 0.97 |
| ImageNet-12 | | | | | | | |
| Attack | Multi-Krum | Median | RFA | Aksel | CClip | DnC | RBTM |
| BitFlip | 59.62 ± 0.73 | 58.56 ± 4.80 | 59.71 ± 5.00 | 61.64 ± 1.98 | 14.87 ± 1.58 | 59.78 ± 1.50 | 58.49 ± 1.99 |
| LIE | 62.66 ± 0.30 | 51.41 ± 1.52 | 60.99 ± 1.22 | 54.14 ± 3.14 | 16.19 ± 3.95 | 67.85 ± 2.87 | 67.12 ± 0.39 |
| IPM | 52.66 ± 2.01 | 59.20 ± 2.44 | 61.25 ± 0.62 | 59.17 ± 1.27 | 14.33 ± 5.95 | 66.31 ± 3.60 | 55.93 ± 0.57 |
| MinMax | 68.17 ± 1.91 | 67.76 ± 0.07 | 63.05 ± 0.75 | 59.33 ± 3.85 | 20.99 ± 3.07 | 68.05 ± 1.59 | 65.99 ± 1.26 |
| MinSum | 57.50 ± 3.09 | 58.78 ± 2.10 | 64.04 ± 0.69 | 67.15 ± 0.32 | 16.38 ± 2.70 | 68.69 ± 1.18 | 61.70 ± 1.62 |
| Mimic | 66.86 ± 0.04 | 59.39 ± 6.07 | 60.45 ± 7.09 | 58.94 ± 1.27 | 11.35 ± 2.26 | 69.07 ± 4.69 | 55.26 ± 1.30 |
| STRIKE (Ours) | **27.24** ± 1.63 | **42.98** ± 1.62 | **43.30** ± 3.13 | **38.11** ± 1.02 | **8.33** ± 1.85 | **53.40** ± 4.94 | **38.81** ± 0.65 |
| FEMNIST | | | | | | | |
| Attack | Multi-Krum | Median | RFA | Aksel | CClip | DnC | RBTM |
| BitFlip | 82.67 ± 5.13 | 71.57 ± 3.61 | 83.41 ± 4.33 | 81.42 ± 3.45 | 83.85 ± 8.50 | 83.58 ± 5.20 | 82.58 ± 6.08 |
| LIE | 68.11 ± 6.86 | 58.38 ± 7.06 | 66.19 ± 7.93 | 38.48 ± 3.32 | 73.03 ± 3.86 | 77.42 ± 5.60 | 53.35 ± 5.17 |
| IPM | 84.12 ± 3.06 | 72.60 ± 8.42 | 83.42 ± 4.13 | 78.28 ± 7.37 | 84.93 ± 4.41 | 83.03 ± 5.02 | 83.21 ± 6.42 |
| MinMax | 68.42 ± 5.91 | 66.44 ± 5.88 | 71.55 ± 5.98 | 34.22 ± 4.94 | 72.12 ± 4.39 | 75.40 ± 3.78 | 59.23 ± 3.41 |
| MinSum | 62.06 ± 3.13 | 65.46 ± 3.66 | 70.36 ± 7.24 | 44.91 ± 3.90 | 75.40 ± 4.88 | 77.11 ± 3.61 | 68.10 ± 8.86 |
| Mimic | 83.15 ± 3.46 | 74.00 ± 4.79 | 83.87 ± 3.00 | 79.06 ± 7.21 | 83.94 ± 5.25 | 82.22 ± 5.40 | 81.92 ± 3.40 |
| STRIKE (Ours) | **22.13** ± 7.78 | **55.19** ± 3.49 | **39.43** ± 5.06 | **16.58** ± 3.63 | **18.88** ± 4.30 | **17.56** ± 5.95 | **39.33** ± 11.98 |

Table 1: Accuracy (mean±std) under different attacks against different defenses on CIFAR-10, ImageNet-12, and FEMNIST. The best attack performance is in bold (the *lower*, the better).

# 7 Experiments

We conduct all experiments on the same workstation with 8 Intel(R) Xeon(R) Platinum 8336C CPUs, a NVIDIA Tesla V100, and 64GB main memory running Linux platform.

## 7.1 Experimental Setups

**Datasets.** Our experiments are conducted on three real-world datasets: CIFAR-10 (Krizhevsky and Hinton 2009), a subset of ImageNet (Russakovsky et al. 2015) refered as ImageNet-12 (Li et al. 2021) and FEMNIST (Caldas et al. 2018). Please refer to Apendix D.1 for more details about the data distribution.

More detailed setups are deferred to Apendix D.1.

**Baseline attacks.** We consider six state-of-the-art attacks: BitFlip (Allen-Zhu et al. 2020), LIE (Baruch, Baruch, and Goldberg 2019), IPM (Xie, Koyejo, and Gupta 2020), MinMax (Shejwalkar and Houmansadr 2021), Min-Sum (Shejwalkar and Houmansadr 2021), and Mimic (Karimireddy, He, and Jaggi 2022). Among the above six attacks, BitFlip and LabelFlip are popular agnostic attacks; LIE, Min-Max and Min-Sum are partial knowledge attacks; IPM is an omniscient attack. The detailed introduction and hyperparameter settings of these attacks are shown in Apendix D.1.

**Evaluated defenses.** We evaluate the performance of our attack on the following robust AGRs: Multi-Krum (Blanchard et al. 2017), Median (Yin et al. 2018), RFA (Pillutla, Kakade, and Harchaoui 2019), Aksel (Boussetta et al. 2021), CClip (Karimireddy, He, and Jaggi 2021) DnC (Shejwalkar and Houmansadr 2021), and RBTM (El-Mhamdi et al. 2021). Besides, we also consider bucketing (Karimireddy, He, and Jaggi 2022) and NNM (Allouah et al. 2023), two simple yet effective schemes that adapt existing robust AGRs to the non-IID setting. The detailed hyperparameter settings of the above robust AGRs are listed in Apendix D.1.

## 7.2 Experiment Results

**Attacking against various robust AGRs.** Table 1 demonstrates the performance of seven different attacks against seven robust AGRs on CIFAR-10, ImageNet-12, and FEMNIST datasets. From Table 1, we can observe that our STRIKE attack generally outperforms all the baseline attacks against various defenses on all datasets, verifying the efficacy of our STRIKE attack. On ImageNet-12 and FEMNIST, the improvement of STRIKE over the best baselines is more significant. We hypothesize that this is because the skew degree is higher on ImageNet-12 and FEMNIST compared to CIFAR-10. Since STRIKE exploits gradient skew to launch Byzantine attacks, it is more effective on ImageNet-12 and FEMNIST. DnC demonstrates almost the strongest resilience to previous baseline attacks. This is because these

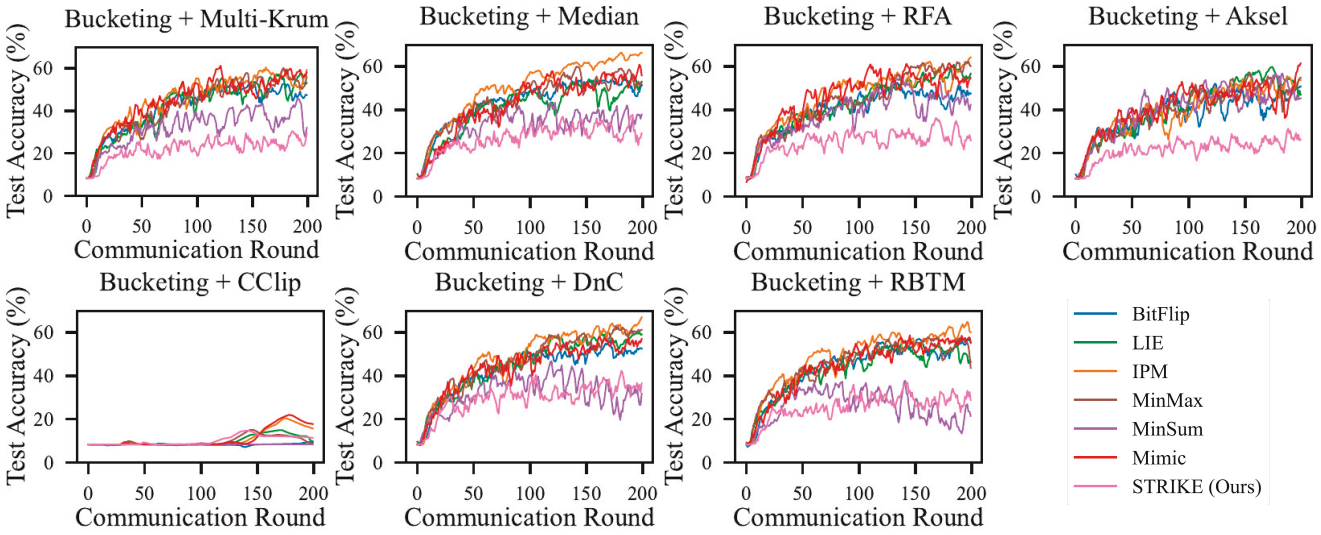



Figure 4: Accuracy under different attacks against seven robust AGRs with bucketing on ImageNet-12. The *lower*, the better.

| Attack | NNM + Median | NNM + RFA | NNM + DnC |
|---|---|---|---|
| BitFlip | 57.14 | 58.55 | 53.68 |
| LIE | 58.04 | 58.68 | 58.87 |
| Mimic | 66.15 | 67.43 | 69.35 |
| STRIKE | **39.61** | **40.38** | **38.91** |

Table 2: Accuracy under different attacks against NNM on ImageNet-12. The best results are in bold (The *lower*, the better).

attacks fail to be aware of the skew nature of honest gradients in FL. By contrast, our STRIKE attack can take advantage of gradient skew and circumvent DnC defense. The above observations clearly validate the superiority of STRIKE.

**Attacking against robust AGRs with bucketing.** Figure 4 demonstrates the performance of seven different attacks against the bucketing scheme (Karimireddy, He, and Jaggi 2022) with different robust AGRs. The results demonstrate that our STRIKE attack works best against Multi-Krum, RFA, and Aksel. When attacking against DnC, Median, and RBTM, only MinSum attack is comparable to our STRIKE attack.

**Attacking against robust AGRs with NNM.** Table 2 compare the performance of STRKE attack against top-3 strongest attacks against the NNM scheme (Karimireddy, He, and Jaggi 2022) under the top-3 most robust robust AGRs. The results suggest that the proposed STRIKE attack still outperforms other baseline attacks against NNM.

**Imparct of $\nu$ on STRIKE attack.** We study the influence of $\nu$ on ImageNet-12 dataset. We report the test accuracy under STRIKE attack with $\nu$ in $\{0.25 * i \mid i = 1, \ldots, 8\}$ against seven different defenses on ImageNet-12. As shown in Fgure 7, the performance of STRIKE is generally competitive with varying $\nu$. In most cases, simply setting $\nu = 1$ can beat almost all the attacks (except for CClip, yet we observe that the performance is low enough to make the model useless).

**The effectiveness of STRIKE attack under different non-IID levels.** We vary Dirichlet concentration parameter $\beta$ in $\{0.1, 0.2, 0.5, 0.7, 0.9\}$ to study how our attack behaves under different non-IID levels. We additionally test the performance in the IID setting. As shown in Fgure 8, the accuracy generally increases as $\beta$ decreases for all attacks. The accuracy under our STRIKE attack is consistently lower than that of all the baseline attacks. Besides, we also note that the accuracy gap between our STRIKE attack and other baseline attacks gets smaller when the non-IID level decreases. We hypothesize the reason is that gradient skew becomes milder as the non-IID level decreases. Even in the IID setting, our STRIKE attack is competitive compared to other baselines.

## 8 Conclusion

In this paper, we observe the existence of the gradient skew phenomenon due to non-IID data distribution in FL. Based on the observation, we propose a novel attack called STRIKE that can exploit the vulnerability. Generally, STRIKE hides Byzantine gradients within the skewed honest gradients. To this end, STRIKE first searches for the skewed honest gradients, and then constructs Byzantine gradients within the skewed honest gradients by solving a constrained optimization problem. Empirical studies on three real-world datasets confirm the efficacy of our STRIKE attack. The STRIKE relies on the gradient skew phenomenon, which is closely related to non-IIDness of data distribution. When the data is IID, the performance could be limited. Therefore, defenses that can alleviate non-IID can potentially mitigate our STRIKE attack. In our future works, we will explore potential defenses against this threat.

## Ethical Statement

The proposed skew-aware Byzantine attack STRIKE can present a threat to federated learning. Our goal with this work is thus to preempt these harms and encourage Byzantine defenses that are robust to skew-aware attacks in the future.

## Acknowledgements

This work is funded by National Key Research and Development Project (Grant No: 2022YFB2703100) and by the Pioneer R&D Program of Zhejiang (No.2024C01021). This work is also sponsored by Sony AI.

## References

Allen-Zhu, Z.; Ebrahimianghazani, F.; Li, J.; and Alistarh, D. 2020. Byzantine-Resilient Non-Convex Stochastic Gradient Descent. In *International Conference on Learning Representations*.

Allouah, Y.; Farhadkhani, S.; Guerraoui, R.; Gupta, N.; Pinot, R.; Rizk, G.; and Voitovych, S. 2024. Byzantine-Robust Federated Learning: Impact of Client Subsampling and Local Updates. In *Forty-first International Conference on Machine Learning*.

Allouah, Y.; Farhadkhani, S.; Guerraoui, R.; Gupta, N.; Pinot, R.; and Stephan, J. 2023. Fixing by Mixing: A Recipe for Optimal Byzantine ML under Heterogeneity. *arXiv preprint arXiv:2302.01772*.

Baruch, G.; Baruch, M.; and Goldberg, Y. 2019. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems*, 32.

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.

Boussetta, A.; El Mhamdi, E. M.; Guerraoui, R.; Maurer, A. D. O.; and Rouault, S. L. A. 2021. Aksel: Fast byzantine sgd. In *Proceedings of the 24th International Conference on Principles of Distributed Systems (OPODIS 2020)*, CONF. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

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

El-Mhamdi, E. M.; Farhadkhani, S.; Guerraoui, R.; Guirguis, A.; Hoang, L.-N.; and Rouault, S. 2021. Collaborative learning in the jungle (decentralized, byzantine, heterogeneous, asynchronous and nonconvex learning). *Advances in Neural Information Processing Systems*, 34: 25044–25057.

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.

Farhadkhani, S.; Guerraoui, R.; Gupta, N.; Pinot, R.; and Stephan, J. 2022. Byzantine Machine Learning Made Easy By Resilient Averaging of Momentums. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 6246–6283. PMLR.

Guerraoui, R.; Rouault, S.; et al. 2018. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, 3521–3530. PMLR.

Karimireddy, S. P.; He, L.; and Jaggi, M. 2021. Learning from history for byzantine robust optimization. In *International Conference on Machine Learning*, 5311–5319. PMLR.

Karimireddy, S. P.; He, L.; and Jaggi, M. 2022. Byzantine-Robust Learning on Heterogeneous Datasets via Bucketing. In *International Conference on Learning Representations*.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, 5132–5143. PMLR.

Knoke, D.; Bohrnstedt, G.; and Mee, A. 2002. *Statistics for Social Data Analysis*. F.E. Peacock Publishers. ISBN 9780875814483.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2: 429–450.

Li, X.; Huang, K.; Yang, W.; Wang, S.; and Zhang, Z. 2019. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*.

Li, Y.; Lyu, X.; Koren, N.; Lyu, L.; Li, B.; and Ma, X. 2021. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34.

Luo, M.; Chen, F.; Hu, D.; Zhang, Y.; Liang, J.; and Feng, J. 2021. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34: 5972–5984.

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. PMLR.

Moore, D. S.; Mccabe, G. P.; and Craig, B. A. 2009. Introduction to the practice of statistics.

Peng, H.; Yu, H.; Tang, X.; and Li, X. 2024. FedCal: Achieving Local and Global Calibration in Federated Learning via Aggregated Parameterized Scaler. *arXiv preprint arXiv:2405.15458*.

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

Roweis, S. T.; and Saul, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500): 2323–2326.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.;

et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252.

Shejwalkar, V.; and Houmansadr, A. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*.

Woodworth, B. E.; Patel, K. K.; and Srebro, N. 2020. Minibatch vs local sgd for heterogeneous distributed learning. *Advances in Neural Information Processing Systems*, 33: 6281–6292.

Xie, C.; Koyejo, O.; and Gupta, I. 2020. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Uncertainty in Artificial Intelligence*, 261–270. PMLR.

Yan, H.; Zhang, W.; Chen, Q.; Li, X.; Sun, W.; Li, H.; and Lin, X. 2024. Recess vaccine for federated learning: Proactive defense against model poisoning attacks. *Advances in Neural Information Processing Systems*, 36.

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, H.; Yang, S.; and Zhu, S. 2018. Parallel restarted SGD for non-convex optimization with faster convergence and less communication. *arXiv preprint arXiv:1807.06629*, 2(4): 7.