SignGuard: Byzantine-robust Federated Learning through Collaborative Malicious Gradient Filtering

Jian Xu[1], Shao-Lun Huang[1], Linqi Song[2], Tian Lan[3]
[1]Tsinghua University, [2]City University of Hong Kong, [3]George Washington University
[1]{xujian9512, twn2gold}@gmail.com, [2]linqi.song@cityu.edu.hk, [3]tlan@gwu.edu

***Abstract*—Gradient-based training in federated learning is known to be vulnerable to faulty/malicious worker nodes, which are often modeled as Byzantine clients. To this end, previous work either makes use of auxiliary data at parameter server to verify the received gradients (e.g., by computing validation error rate) or leverages statistic-based methods (e.g. median and Krum) to identify and remove malicious gradients from Byzantine clients. In this paper, we acknowledge that auxiliary data may not always be available in practice and focus on the statistic-based approach. However, recent work on model poisoning attacks have shown that well-crafted attacks can circumvent most of existing median- and distance-based statistical defense methods, making malicious gradients indistinguishable from honest ones.**

To tackle this challenge, we show that the element-wise sign of gradient vector can provide valuable insight in detecting model poisoning attacks. Based on our theoretical analysis of state-of-the-art attack, we propose a novel approach, *SignGuard*, to enable Byzantine-robust federated learning through collaborative malicious gradient filtering. More precisely, the received gradients are first processed to generate relevant magnitude, sign, and similarity statistics, which are then collaboratively utilized by multiple, parallel filters to eliminate malicious gradients before final aggregation. We further provide theoretical analysis of SignGuard by quantifying its convergence with appropriate choice of learning rate and under non-IID training data. Finally, extensive experiments of image and text classification tasks - including MNIST, Fashion-MNIST, CIFAR-10, and AG-News - are conducted together with recently proposed attacks and defense strategies. The numerical results demonstrate the effectiveness and superiority of our proposed approach. For example, our SignGuard can detect almost all malicious gradients from *Little is Enough* attack and lead to negligible accuracy drop.

***Index Terms*—Byzantine Attack; Federated Learning; Machine Learning Security**

I. Introduction

In the era of big data, private data are often scattered among local clients (e.g., companies, mobile devices), giving arise to the problem of isolated data islands[1]. To fully capitalize on the value of big data while protecting the data privacy and security, federated learning (FL) has attracted significant interests from both academic and industry sides[2], [3], [4], [1], [5]. Similar to the traditional distributed learning system, a typical architecture of FL consists of a parameter server (PS) and a number of distributed clients, with the key difference that local training data in FL are usually prohibited from sharing among the clients. The general goal of FL is to jointly train a global model that has high generalization ability than that only trained on local data. While FL systems allow clients to keep their private data local, a significant vulnerability arises when a subset of clients - which are modeled as Byzantine clients [6], [7], [8] - aim to prevent successful training of global model or make some targeted samples misclassified during inference phase[4], [9], [10], [11]. It has been shown that mitigating Byzantine model poisoning attacks is crucial for robust FL and other distributed learning[12], [7], [13]. On the other hand, distributed implementation of gradient-based learning algorithms[14] are increasingly popular for training large-scale models on distributed datasets, e.g., deep neural networks for human face identification and news sentimental analysis [15], [16], [17]. Therefore, many efforts have been devoted to developing robust **g**radient **a**ggregation **r**ules (GAR)[4] to achieve Byzantine-robust FL algorithms.

In this paper, we focus on gradient-based FL systems and propose a novel defense that is capable of detecting state-of-the-art Byzantine model poisoning attacks. We note that Byzantine clients can send arbitrary model update vectors to the PS, which may significantly poison the training process if not identified and removed by PS. This can be seen through a simple example shown in Fig. 1 with 1 PS and n benign clients as well as m Byzantine clients. During federated learning, the selected clients (including both benign and Byzantine ones) pull the updated model from the PS, then compute and send local gradients to the PS in parallel, while the PS aggregates the collected gradients to update the global model[2], [3]. For full participation setting, PS can also directly broadcast the aggregated global gradient back to the clients to update their local parameters. When the benign clients are the majority, theoretically we can always distinguish the minority as outliers, i.e., malicious gradients, according to the principle of majority-vote[18], [19]. For partial participation setting, we need to assume that the selected benign clients are always the majority in every communication round, otherwise it's impossible to reliably distinguish the honest gradients from the malicious ones without other information. Thus, we mainly focus on the full participation setting, where the PS collects gradient from all clients in each round. And we do not assume any trusty gradient or any other validation data available in PS. In the rest of this paper, we will use Byzantine and malicious clients interchangeably.

Recently, much research attention has focused on mitigating Byzantine attack either by leveraging statistic-based outlier detection techniques [20], [19] or by utilizing auxiliary labeled data collected by PS to verify the correctness of received gradients [21], [22]. While in some FL systems, it is possible

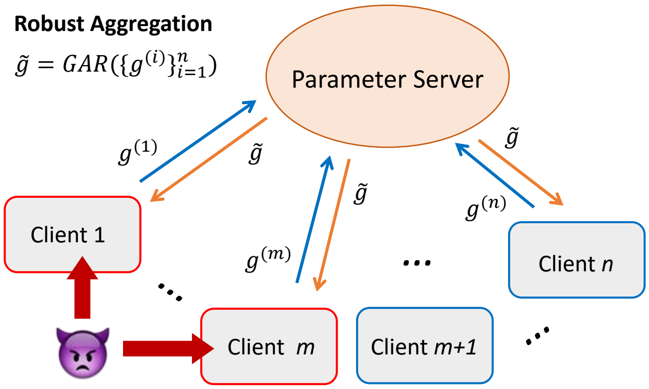


Fig. 1. Federated learning system: one parameter server with n clients, in which a attacker controls m Byzantine clients to attack the learning system.

to collect a small amount of data from volunteer clients or external datasets to approximate the true data distribution [23], we note that auxiliary data sufficiently capturing the global data distribution may not always be practicable to PS. Therefore, we focus on the statistic-based approaches that leverage various types of statistics, e.g. magnitude and similarity, to identify and remove malicious gradients sent by Byzantine clients. Meanwhile, recent works have shown that existing statistic-based aggregation rules are vulnerable to well-crafted model poisoning attacks[24], [25], which are indistinguishable in Euclidean distance such that they can circumvent most defenses.

Our Method. We propose a robust gradient aggregation framework, namely SignGuard, to enable Byzantine-robust federated learning. SignGuard leverages a new technique of sign-gradient filtering to identify malicious gradients and can be integrated with existing gradient aggregation rules, such as trimmed-mean[19]. In particular, we define *sign-gradient* as the element-wise sign of a gradient vector. The key idea of SignGuard is that the sign distribution of sign-gradient can provide valuable information in detecting advanced model poisoning attacks, which would otherwise evade state-of-the-art statistic-based detection methods such as Krum and Bulyan[7], [26]. SignGuard is inspired by our theoretical analysis of *Little is Enough* (LIE) attack [24], and the general good performance of signSGD[18] in distributed learning tasks. In[18] the authors show that even if PS only collects the sign of gradient, the model training can still converge with small accuracy degradation and keep the training process fault-tolerant. This fact tells us that the sign of gradient plays an vital role in model updating. Our novel analysis on the LIE attack reveals that gradient manipulation can cause significant variation of sign distribution, which turns out to be a breakthrough against such well-crafted attacks. We also empirically find that even the simplest *sign statistics*[1] can expose most of the attacks. These observations provide a new perspective towards Byzantine attack mitigation and directly inspire the design of our SignGuard framework. The core of our approach is extracting robust features of received gradients and using unsupervised clustering method to remove the anomalous ones. We find this simple strategy is able to detect suspicious gradients effectively.

Our Work. To the best of our knowledge, this is the first work to utilize sign statistics of gradients for Byzantine-robust federated learning. SignGuard employs well-designed filtering techniques to identify and eliminate the suspicious gradients to favor existing aggregators. Our theoretical analysis proves that SignGuard can guarantee training convergence on both IID and non-IID training data, while introducing no extra overhead for local computation or auxiliary data collection. Further, instead of detecting suspicious clients and removing them completely from the training process, our proposed SignGuard checks the received gradients directly at each round and is able to utilize updates from Byzantine clients when they behavior honestly, which is crucial for fast convergence in the non-IID settings. In particular, for a system with n clients including m Byzantine clients satisfying $n \geq 2m + 1$, we quantify the gradient bias induced by ignoring m suspicious gradients and show that the parameters enjoy a similar update rule as in safe training, thus the convergence analysis could be performed similarly. Finally, SignGuard is evaluated on various real-world image and text classification tasks through extensive experiments by changing the attack method and the percentage of malicious clients. Our evaluation results demonstrate the effectiveness of our SignGuard in protecting the FL system from Byzantine poisoning attacks and meanwhile achieving high model accuracy.

Our Contributions. To summarize, we make the following key contributions:

- A novel framework called SignGuard is proposed for Byzantine-robust federated learning, which leverages the sign statistics of gradient to defend against model poisoning attacks. SignGuard does not require any auxiliary data and can mitigate state-of-the-art attacks.
- We provide theoretical analysis of the harmfulness and stealthiness of the state-of-the-art *Little is Enough* attack, and also propose a new hybrid attack strategy.
- The convergence of SignGuard is proven with appropriate choice of learning rate. In particular, we show that Byzantine clients inevitably affect the convergence error in non-IID setting even if all malicious gradients are removed.
- SignGuard is verified through extensive experiments on MNIST, Fashion-MNIST, CIFAR-10 datasets for image classification tasks and AG-News dataset for text classification task under various Byzantine attacks. Compared with existing approaches, our SignGuard exhibits superiority in both IID and non-IID settings.

II. Background and Related Work

A. Safety & Security in Federated Learning

The model safety and data security are essential principles of federated learning due to the concern of privacy risks

[1] By default, we use the “sign statistics” to denote the proportions of positive, negative and zero signs.

and adversarial threats [3], [4], [9], [27], especially in the age of emerging privacy regulations such as General Data Protection Regulation (GDPR) [28]. In the context of FL, instead of raw data, the gradient information are shared to jointly train a model. More advanced technologies such as secure multiparty computation or differential privacy are also employed to enhance the privacy guarantees[29], [30], [31]. Meanwhile, the learning systems are vulnerable to various kinds of failures, including non-malicious faults and malicious attacks. Data poisoning attacks and model update poisoning attacks (aka. untargeted attacks) aim to degrade or even fully break the global model during training phase, while backdoor attacks (aka. targeted attacks) make the model misclassify certain samples during inference phase [4]. In particular, the Byzantine threats can be viewed as worst-case attacks, in which corrupted clients can produce arbitrary outputs and are allowed to collude. In many studies, the Byzantine attacker is assumed to be omniscient and have capability to access white-box model parameters and all honest gradients to conduct strong attacks [13]. As pointed in many works, appropriately crafted attacks can give significant impact on the model performance while circumventing most of current defenses [24]. However, security mechanisms to protect privacy inevitably make it a more challenging task to successfully detect those failures and attacks, such as secure aggregation where the server can not directly see any individual client updates but an aggregate result [30], [4]. Thus, the trade-off between privacy assurance and system robustness needs more investigations.

B. Existing Defense Strategies

Statistic-based. This is also known as majority-vote based strategy, requiring the percentage of Byzantine clients less than 50%. This kind of methods use the ℓ_p-norm distance or cosine-similarity to measure the confidence for received gradients. The Krum as well as extended Multi-Krum are pioneering work towards Byzantine-robust learning[7]. In [19], the convergence rate and error rate of trimmed-mean (TrMean) and coordinate-wise median (Median) is rigorously studied. Moreover, [26] has shown that the Krum and median defenses are vulnerable to ℓ_p-attack and developed a meta-method called Bulyan on top of other robust aggregation methods. Specially, some works only aggregate the sign of gradient to mitigate the Byzantine effect[18], [32]. Recently, a method called Divider and Conquer is proposed to tackle strong attacks [25].

Validation-based. The most straightforward approach to evaluate whether a particular gradient is honest or not, is utilizing the auxiliary data in PS to validate the performance of updated model. Zeno [21] use a stochastic descendant score to evaluate the correctness of each gradient and choose those with highest scores. Fang [13] use error rate based and loss function based rejection mechanism to reject gradients that have bad impact on model updating. In [23], the authors utilize the ReLU-clipped cosine-similarity between each received gradient and standard gradient as weight to get robust aggregation. The main concern of such approaches is the accessibility of auxiliary data.

History-aided. If the one-to-one correspondence between gradient and client entity is knowable for PS, then it's possible to utilize historical data to trace the clients' behaviors. Some studies show that malicious behavior could be revealed from the gradient trace by designing advanced filter techniques [33], [34]. In[35], the authors propose a Hidden Markov Model to learn the quality of model updates and discard the bad or malicious updates. Besides, the momentum SGD can also be considered as history-aided method and can help to alleviate the impact of Byzantine attacks[36], [37].

Redundancy-based. In the context of traditional distributed training, it's possible to assign each node with redundant data and use this redundancy to eliminate the effect of Byzantine failures. In [38], the authors present a scalable framework called DRACO for robust distributed training using ideas from coding theory. In [39], a method based on data encoding and error correction techniques over real numbers is proposed to combat adversarial attacks. In [40], a framework called DETOX is proposed by combing computational redundancy and hierarchical robust aggregation to filter out Byzantine gradients.

Learning-based. In [41], the authors use VAE as spectral anomaly detection model to learn the representation of honest gradients and use reconstruction error in each round as detection threshold. In [42], a method called Justinian's GAAvernor is proposed to learn a robust gradient aggregation policy against Byzantine attacks via reinforcement learning. In [43], the authors use auxiliary data in PS to learn the coefficient in weighted average aggregation for each received gradient.

Ensemble-learning. Another line of work leverage the ensemble learning approach to provably guarantee the predicted label for a testing example is not affected by Byzantine clients, in which multiple global models are trained and each of them is learned by using a randomly selected subset of clients [44], [45]. However, such ensemble-learning methods significantly enlarge computational overhead and storage cost.

III. Rethink of Recent Attacks

In this section, we first give the threat model and then present our theoretical analysis along with empirical evidence of the *Little is Enough (LIE)* attack [24] to demonstrate the limitation of existing median- and distance-based defenses.

Threat Model. Similar to the threat models in previous works [7], [24], [13], [25], we assume that there exists an attacker that controls some malicious clients to perform model poisoning attacks. The malicious clients could be fake clients that injected by the attacker or genuine ones but corrupted by the attacker. Specially, we assume the attacker has full knowledge on all benign gradients, and model parameters, and the corrupted clients can collude to conduct strong attacks. However, the attacker cannot corrupt the server and the proportion of malicious clients β is less than half. For a system with n clients, without loss of generality, we assume that the first m clients are corrupted and $\beta = \frac{m}{n} < 0.5$.

LIE Attack. Byzantine clients first estimate coordinate-wise mean (μ_j) and standard deviation (σ_j), and then send malicious gradient vector with elements crafted as follows:

$$(g_m)_j = \mu_j - z \cdot \sigma_j, \ j \in [d] \tag{1}$$

where the positive attack factor z depends on the total number of clients and Byzantine fraction. The design mechanism behind this attack is circumventing the coordinate-wise median and trimmed-mean methods. As advised by original paper, the z can be determined by using cumulative standard normal function $\phi(z)$:

$$z_{max} = max_z \left(\phi(z) < \frac{n - \lfloor \frac{n}{2} + 1 \rfloor}{n - m} \right) \tag{2}$$

In the following, we will show why this attack is harmful and hard to detect. From an optimization point of view, we can check the upper bound of non-convex distributed optimization problem before and after *LIE* attack, where we assume the distributed data are IID for simplicity. Lemma 1 gives out general upper bound when no attack and no defense are performed [46], [47], from which we can see that the objective function will converge to a critical point given large iterations T and small learning rate η. Applying similar analysis method, we can get a new upper bound when *LIE* attack and coordinate-wise median defense are conducted as presented in Proposition 1, where we assume the training can converge.

Lemma 1. *For a distributed non-convex optimization problem $F(\mathbf{x})$ with n benign workers, suppose the data are IID and the gradient variance is bounded by σ^2. Empoly the SGD with a fixed learning rate $\eta \leq 1/L$ and assume $F^* = \min_{\mathbf{x}} F(\mathbf{x})$, then we have the following convergence result*[2]*:*

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \leq \frac{2(F(\mathbf{x}_0) - F^*)}{\eta T} + \frac{L\eta\sigma^2}{n} \tag{3}$$

Proposition 1. *For a distributed non-convex optimization problem $F(\mathbf{x})$ with $(n-m)$ benign workers and m malicious workers conducting LIE attack with appropriate z, suppose the data are IID and the gradient variance is bounded by σ^2. Employ the Median-SGD with a fixed learning rate $\eta \leq 1/L$, and assume $F^* = \min_{\mathbf{x}} F(\mathbf{x})$, then we have the upper bound B of averaged gradient norm square:*

$$B \leq \frac{2(F(\mathbf{x}_0) - F^*)}{\eta T} + \frac{L\eta\sigma^2}{n} + \left(1 + \frac{1}{n}\right) z^2\sigma^2 \tag{4}$$

Proof. Detailed proof is in Appendix B. □

Compared with the result in Lemma 1, there exists an extra constant term in Proposition 1, which does not diminish even with decreasing learning rate, enlarging the convergence error and even making the model training totally collapsed. When no defense is employed, just replace z with βz, because the $m \cdot z\sigma$ could be averaged across all n workers, resulting in a smaller upper bound than median-based defense. This also explains the phenomenon that naive Mean aggregation even has better results than median-based and distance-based defenses in some cases as shown in [24] and experimental results in this paper. Then, we turn to the coordinate point of view to further analyze why this type of crafted gradient is harmful for model training. Recall that signSGD can achieve good model accuracy by only utilizing the sign of gradient, which illuminates a fact that the sign of gradient plays an crucial role in model updating. Therefore, it's worthy to check the sign of gradient for this type attack. The crafting rule of *LIE* attack is already shown in Eq. (1), from which we can see that $(g_m)_j$ could have opposite sign with μ_j when $\mu_j > 0$. For coordinate-wise median and $\mu_j > 0$, we assume this aggregation rule results in $\tilde{g} = g_m$, then we have:

$$if \quad z > \frac{\mu_j}{\sigma_j}, \quad then \quad sign(\tilde{g}_j) \neq sign(\mu_j) \tag{5}$$

For mean aggregation rule and $\mu_j > 0$, if μ_j and σ_j are estimated on benign clients, then the j-th element becomes:

$$\tilde{g}_j = \frac{1}{n}[m \cdot (g_m)_j + (n-m)\mu_j] = \mu_j - z \cdot \beta \cdot \sigma_j \tag{6}$$

and in this case a bigger z is needed to reverse the sign:

$$if \quad z > \frac{n\mu_j}{m\sigma_j}, \quad then \quad sign(\tilde{g}_j) \neq sign(\mu_j) \tag{7}$$

Empirical results in [24] show that mostly coordinate-wise standard deviation turns out to be bigger than the corresponding gradient element, thus a small z could turn a large number of positive elements into negative, leading to incorrect model updating. To verify this theoretical result, we adopt default training setting in Section V to train a CNN on MNIST dataset and ResNet-18 on CIFAR-10 dataset under no attack, and calculate averaged sign statistics across all workers as well as the sign statistics of a virtual gradient that crafted as Eq. (1). We plot the sign statistics over iterations as Fig. 2, which convincingly supports our theoretical analysis.

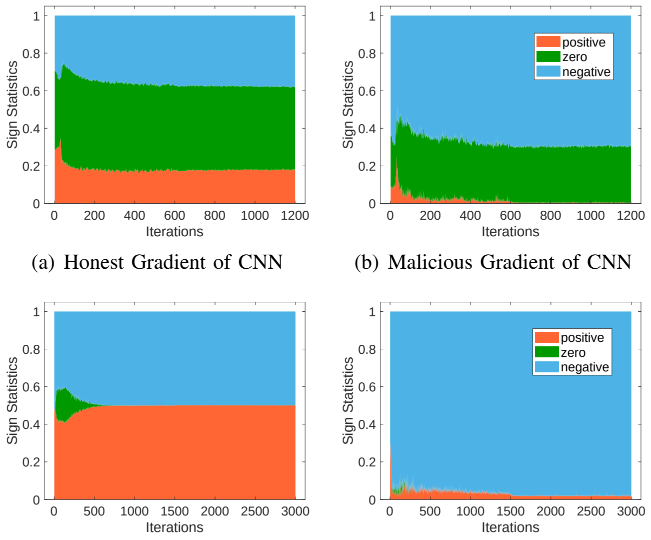
Next, we present the following Proposition 2 to explain why *LIE* attack is hard to detect, in which we compare the distance to averaged true gradient $\tilde{g} = \frac{1}{n}\sum_{i=1}^{n} g^{(i)}$ and similarity with $\tilde{g}$ for malicious gradient and honest gradient, respectively.

Proposition 2. *For a distributed non-convex optimization problem $F(\mathbf{x})$ with $(n-m)$ benign workers and m malicious workers conducting LIE attack, suppose the data are IID and the gradient variance is bounded by σ^2. Given small enough z, then the distance between malicious gradient and true averaged gradient could be smaller than that of certain honest gradient:*

$$\exists \ i, \ s.t. \ \mathbb{E}[\|g_m - \tilde{g}\|^2] < \mathbb{E}[\|g^{(i)} - \tilde{g}\|^2] \tag{8}$$

and the cosine-similarity between malicious gradient and true averaged gradient could be bigger than that of certain honest

[2] In this paper, $\|\cdot\|$ denotes the ℓ_2 norm.



(a) Honest Gradient of CNN (b) Malicious Gradient of CNN

(c) Honest Gradient of ResNet18 (d) Malicious Gradient of ResNet18

Fig. 2. Sign statistics of honest and malicious gradient.

gradient:

$$\exists\ i,\ s.t.\ \ cos(g_m, \tilde{g}) > cos(g^{(i)}, \tilde{g}) \tag{9}$$

Proof. Detailed proof is in Appendix C. □

From the above results we can see that it's possible for the malicious gradient to be more "safe" when evaluated by Krum and Bulyan methods. Hence, it's almost impossible to detect the malicious gradient from the distance and cosine-similarity perspectives. Instead, checking the sign statistics is a novel and promising perspective to detect abnormal gradients. Similar analysis is also valid for the recent proposed Min-Max/Min-Sum attacks as well as the adaptive attack that uses different perturbation vectors [25], along with which a new method called *Divide and Conquer (DnC)* is also proposed to tackle those attacks. However, this method makes the assumption that malicious gradients are in the direction of largest singular vector of gradient matrix, and would fail when multiple attacks exist simultaneously or in non-IID settings.

New Hybrid Attack. In this work, we extend the OFOM attack in[48] and propose a type of hybrid attack called **ByzMean** attack, which makes the mean of gradients be arbitrary targeted malicious gradient. More specifically, the malicious clients are divided into two sets, one set with m_1 clients chooses a arbitrary gradient value $g_{m_1} = *$, and the other set with $m_2 = m - m_1$ clients chooses the gradient value g_{m_2} such that the average of all gradients is exactly the g_{m_1}, just as follows:

$$g_{m_1} = *,\ g_{m_2} = \frac{(n - m_1)g_{m_1} - \sum_{i=m+1}^{n} g^{(i)}}{m_2} \tag{10}$$

All existing attacks can be integrated into this ByzMean attack, making this hybrid attack even stronger than all single attacks. For example, we can set g_{m_1} as random gradient or even the gradient crafted by *LIE* attack. In that case, all existing defense methods including DnC will be broken.

IV. Our SignGuard Framework

In this section, we present formal problem formulation and introduce our SignGuard framework for Byzantine-robust federated learning. And some theoretical analysis on training convergence is also provided.

A. System Overview and Problem Setup

Our federated learning system consists of a parameter server and a number of benign clients along with a small portion of Byzantine clients. We assume there exists an attacker or say adversary that aims at poisoning global model and controls the Byzantine clients to perform malicious attacks. We first give out the following definitions of benign and Byzantine clients, along with the attacker's capability and defense goal.

Definition 1. (Benign Client) A benign client always sends honest gradient to the server, which is an unbiased estimation of local true gradient at each iteration.

Definition 2. (Byzantine Client) A Byzantine client may act maliciously and can send arbitrary message to the server.

Attacker's Capability: As mentioned in the threat model in Section III, the attacker has full knowledge on all benign gradients and the corrupted clients can collude to conduct various kinds of attacks. However, the attacker cannot compromise the server and the proportion of Byzantine clients is less than 50%.

Defender's Capability: As in previous studies [13], [23], We consider the defense is performed on the server side. The parameter server does not have access to the raw training data on the clients, and the server does not know the exact number of malicious clients. However, the server has full access to the global model as well as the local model updates (i.e., local gradients) from all clients in each iteration. Specially, we further assume the received gradients are anonymous, which means the behavior of each client is untraceable. In consideration of privacy and security, we think this assumption is reasonable in the context of federated learning.

Defense Goal: As mentioned in [23], an ideal defense method should give consideration to the following three aspects: Fidelity, Robustness and Efficiency. We hope the defense method achieves Byzantine-robustness against various malicious attacks without sacrificing the model accuracy. Moreover, the defense should be computationally cheap such that does not affect the overall training efficiency.

Problem Formulation: We focus on federated learning on IID settings and then extend our algorithm into non-IID settings. We assume that training data are distributed over a number of clients in a network, and all clients jointly train a shared model based on disjoint local data. Mathematically, the underlying distributed optimization problem can be formalized as follows:

$$\min_{\mathbf{x} \in R^d} F(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\xi_i \sim D_i} \left[F(\mathbf{x}; \xi_i) \right] \tag{11}$$

where n is the total number of clients, D_i denotes the local dataset of i-th client and could have different distribution from other clients, and $F(\mathbf{x};\xi_i)$ denotes the local loss function given shared model parameters $\mathbf{x}$ and training data ξ_i sampled from D_i. We make all clients initialize to the same point $\mathbf{x_0}$, then FedAvg [2] can be employed to solve the problem. At each iteration, the i-th benign client draws ξ_i from D_i, and computes local stochastic gradient with respect to global shared parameter $\mathbf{x}$, while Byzantine clients can send arbitrary gradient message:

$$g_t^{(i)} = \begin{cases} \nabla F(\mathbf{x}_t;\xi_i), & \text{if } i\text{-th client is benign} \\ arbitrary, & \text{if } i\text{-th client is Byzantine} \end{cases} \tag{12}$$

The parameter server collects all the local gradients and employs robust gradient aggregation rule to get a global model update:

$$\mathbf{x_{t+1}} = \mathbf{x_t} - \eta_t \cdot GAR(\{g_t^{(i)}\}_{i=1}^{n}) \tag{13}$$

In a synchronous and full participation setting, the result will be broadcast to all clients to update their local models and start a new iteration. In a partial participation setting, the model update is finished in PS and the updated model will be sent to the selected clients for next round. This process will repeat until the stop condition is satisfied.

To characterize the impact of Byzantine attack, we define the following two metrics:

Definition 3. (Attack Success Rate) The averaged proportion of malicious gradients that were selected by the detection-based GAR throughout the training iterations.

Definition 4. (Attack Impact) The model accuracy drop compared with benchmark result that under no attack and no defense.

Based on above metrics, we can measure the effect of Byzantine attack by calculating the accuracy drop due to model poisoning and measure the validity of detection-based defense by calculating the attack success rate.

Algorithm 1 SignGuard-based Robust Federated Learning

```
Input: learning rate $\eta$, total iteration $T$, total client number $n$
Initial: $\mathbf{x}_0 \in R^d$
for $t = 0, 1, ..., T-1$ do
    On each client $i$ :
    Sample a mini-batch of data to compute gradient $g_t^{(i)}$
    Send $g_t^{(i)}$ to the parameter server
    Wait for global gradient $\tilde{g}_t$ from server
    Update local model: $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\tilde{g}_t$
    On server:
    Collect gradients from all clients
    Obtain global gradient: $\tilde{g}_t = SignGuard(\{g_t^{(i)}\}_{i=1}^{n})$
    Send $\tilde{g}_t$ to all clients
end for
```

Algorithm 2 SignGuard Function

```
Input: Set of received gradients $S_t = \{g_t^{(i)}\}_{i=1}^{n}$, lower and upper bound $L, R$ for gradient norm
Initial: $S_1 = S_2 = \emptyset$
    Get $l_2$-norm and element-wise sign of each gradient
Step 1: Norm-threshold Filtering
    Get the median of norm $M = med(\{\|g_t^{(i)}\|\}_{i=1}^{n})$
    Add the gradient that satisfies $L \le \frac{\|g_t^{(i)}\|}{M} \le R$ into $S_1$
Step 2: Sign-based Clustering
    Randomly select a subset of gradient coordinates
    Compute sign statistics on selected coordinates for each gradient as features
    Train a Mean-Shift clustering model
    Choose the cluster with most elements as $S_2$
Step 3: Aggregation
    Get trusted set: $S'_t = S_1 \cap S_2$
    Get $\tilde{g}_t = \frac{1}{|S'_t|}\sum_{i \in S'_t} g_t^{(i)}$
Output: Global gradient: $\tilde{g}_t$
```

B. Our Proposed Solution

The proposed SignGuard framework is described in Algorithm 1-2 and the workflow is illustrated in Fig. 3. On a high level, we pay attention to the magnitude and direction of the received gradients. At each iteration, the collected gradients are sent into multiple filters, including norm-based thresholding filer and sign-based clustering filter, etc. **Firstly**, for the norm-based filter, the median of gradient norms is utilized as reference norm as the median always lies in benign set. Considering that small magnitude of gradients do less harm to the training while significantly large one is definitely malicious, we will perform a loose lower threshold and a strict upper threshold. **Secondly**, for the sign-based clustering filter, we extract some statistics of gradients as features and using Mean-Shift [49] algorithm as unsupervised clustering model with adaptive number of cluster classes, while the cluster with largest size is selected as the trusted set. In this work, the proportions of positive, zero and negative signs are computed as basic features, which are sufficient for a number of attacks, including LIE attack.

However, those features only consider the overall statistics and lose sight of local properties. Take a toy example, when the amounts of positive and negative elements are approximate (just as ResNet-18), the naive sign statistics may be insufficient to detect sign-flipped gradients [40] or those well-crafted attacks that have similar sign statistics. To mitigate this problem, we introduce randomized coordinate selection and add a similarity metric as additional feature in our algorithm, such as cosine-similarity or Euclidean distance between each received gradient and a “correct” gradient. However, without the help of auxiliary data in PS, the “correct” gradient is not directly available. A practical way is to compute pairwise similarities between all the other gradients and take the median

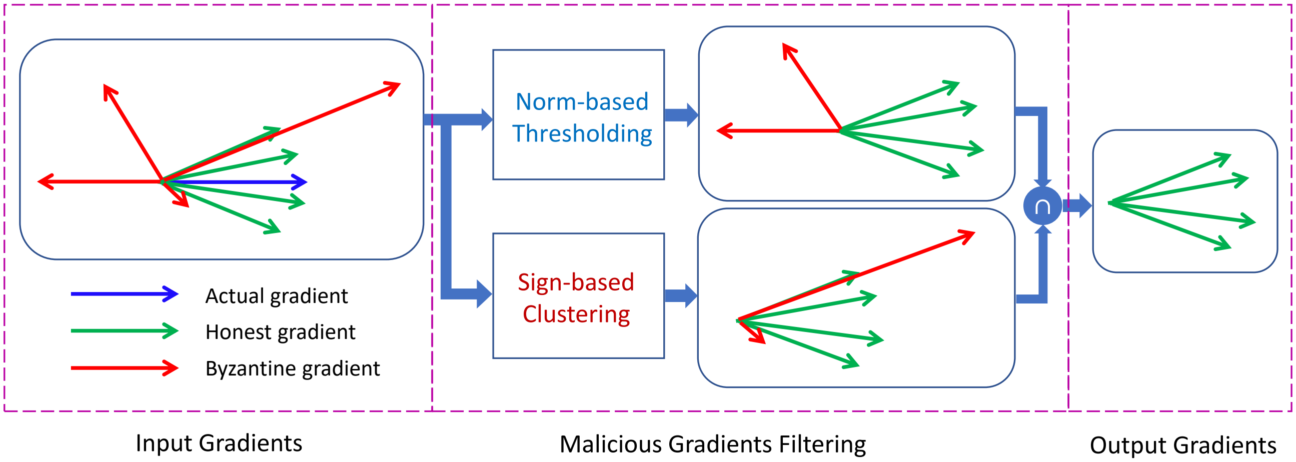


Fig. 3. Illustration of the workflow of proposed SignGuard. The collected gradients are anonymous and sent into multiple filters, after which the intersection of multiple outputs are selected as trusted gradients.

as the similarity with "correct" gradient. Or more efficiently, just utilize the aggregated gradient from previous iteration as the "correct" gradient. Intuitively, it is promising to distinguish those irrelevant gradients and helps to improve the robustness of anomaly detection. What's challenging is, as shown in Section III, the Euclidean distance or cosine-similarity metrics are not reliable for the state-of-the-art attacks, and even affect the judgment of SignGuard as we found in experiments. In this work, the plain "SignGuard" only uses sign statistics in default, and the enhanced variants that add cosine-similarity feature or Euclidean distance feature are called "SignGuard-Sim" and "SignGuard-Dist", respectively. We will provide some comparative results of them. We emphasize that the SignGuard is a sort of flexible approach and more advanced features could be further extracted to enhance the effectiveness of anomaly detection. And how to design a more reliable similarity metric is left as an open problem for future work.

After filtering, the server eventually selects the intersection of multiple filter outputs as trusted gradient set, and obtains a global gradient by robust aggregation, e.g. trimmed-mean. In this work, we use the mean aggregation with magnitude normalization. It is worth noting that a small fraction of honest gradients could also be filter out due to gradient diversity, especially in the non-IID settings, depending on the variance of honest gradients and the closeness to malicious gradients.

C. Convergence Analysis

In this part, we provide some theoretical analysis of the security guarantee by SignGuard and the convergence of non-convex optimization problem, jointly considering the IID and non-IID data. We first claim that high separability can be achieved when the distributions of test statistics for malicious and honest gradients have negligible overlap.

Claim 1. Suppose all honest gradients are computed with global model parameters and same batch size, and assume the test statistics of honest and malicious gradients follow two finite covariance distributions P and Q. For $0 < \beta < 1/2$, let $U = (1-\beta)P + \beta Q$ be a mixture of sample points from P and Q, denote $f(\mathbf{x})$ and $g(\mathbf{x})$ the PDFs of P and Q. Then, there exists a algorithm that separates data points with low probability of error if the total variation distance satisfies: $TV(f,g) = 1 - o(1)$.

Remark 1. *Note that the Byzantine clients have an inevitable trade-off between the attack impact and the risk of exposure by manipulating the gradient deviation. Therefore, under our detection-based SignGuard framework, the malicious gradient either have limited attack impact or become obvious to get detected, depending on the discrepancy between P and Q.*

To conduct convergence analysis, we also make the following basic assumption, which is commonly used in the literature [47], [46], [50] for convergence analysis of distributed optimization.

Assumption 1. *Assume that problem (11) satisfies:*

***1. Smoothness**: The objective function $F(\cdot)$ is smooth with Lipschitz constant $L > 0$, which means $\forall \mathbf{x}, \forall \mathbf{y},\ \|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq L\|\mathbf{x}-\mathbf{y}\|$. It implies that:*

$$F(\mathbf{x}) - F(\mathbf{y}) \leq \nabla F(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \frac{L}{2}\|\mathbf{x}-\mathbf{y}\|^2 \tag{14}$$

***2. Unbiased local gradient**: For each worker with local data, the stochastic gradient is locally unbiased:*

$$\mathbb{E}_{\xi_i \sim D_i}\left[\nabla F(\mathbf{x};\xi_i)\right] = \nabla F_i(\mathbf{x}) \tag{15}$$

***3. Bounded variances**: The stochastic gradient of each worker has a bounded variance uniformly, satisfying:*

$$\mathbb{E}_{\xi_i \sim D_i}[\|\nabla F(\mathbf{x};\xi_i) - \nabla F_i(\mathbf{x})\|^2] \leq \sigma^2 \tag{16}$$

and the deviation between local and global gradient satisfies:

$$\|\nabla F_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \kappa^2 \tag{17}$$

For SignGuard framework, the trusted gradients attained by filters may still contain a part of malicious gradients. In this case, any gradient aggregation rule necessarily results in an error to the averaged honest gradient [51], [36]. Here we make another assumption on the capability of aggregation rule:

Assumption 2. *For problem (11) with $(1-\beta)n$ benign clients (denoted by $\mathcal{G}$) and βn Byzantine clients, suppose that at most δn Byzantine clients can circumvent SignGuard at each iteration. We assume that the robust aggregation rule in SignGuard outputs $\hat{g}_t$ such that for some constant c and constant b,*

$$\textit{1. Bounded Bias:}\quad [\mathbb{E}\,\|\hat{g}_t - \bar{g}_t\|]^2 \le c\delta \sup_{i,j\in\mathcal{G}} \mathbb{E}[\|g_t^{(i)} - g_t^{(j)}\|^2]$$

$$\textit{2. Bounded Variance:}\quad var\,\|\hat{g}_t\| \le b^2 \tag{18}$$

where $\bar{g}_t = \frac{1}{|\mathcal{G}|}\sum_{i\in\mathcal{G}} g_t^{(i)}$ and $0 \le \delta < \beta < 0.5$.

Remark 2. *When $\delta = 0$, it's possible to exactly recover the averaged honest gradient. For most aggregation rules such as Krum, the output is deterministic and thus has $b^2 = 0$. For clustering-based rules, the output is randomized and could have negligible variance if the clustering algorithm is robust.*

When βn Byzantine clients exist and act maliciously, the desired gradient aggregation result is the average of $(1-\beta)n$ honest gradients, which still has a deviation to the global gradient of no attack setting. We give the following lemma to characterize the deviation:

Lemma 2. *Suppose the training data are non-IID under Assumption 1, then the deviation between averaged gradient of $(1-\beta)n$ clients $\bar{g}$ and the true global gradient $\nabla F(\mathbf{x})$ can be characterized as follows:*

$$\mathbb{E}\left[\|\bar{g} - \nabla F(\mathbf{x})\|^2\right] \le \frac{\beta^2\kappa^2}{(1-\beta)^2} + \frac{\sigma^2}{(1-\beta)n} \tag{19}$$

Proof. Detailed proof is in Appendix D. □

Given above assumptions and lemma, extending the analysis techniques in [46], [47], [50], [36], now we can characterize the convergence of SignGuard by the following theorem.

Theorem 1. *For problem (11) under Assumption 1, suppose the SignGuard satisfying Assumption 2 is employed with a fixed learning rate $\eta \le (2-\sqrt{\delta}-2\beta)/(4L)$ and $F^* = \min_{\mathbf{x}} F(\mathbf{x})$, then we have the following convergence result:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \le \frac{2(F(\mathbf{x}_0)-F^*)}{\eta T} + 2L\eta\Delta_1 + \Delta_2 \tag{20}$$

where the constant terms are $\Delta_1 = 4c\delta(\sigma^2+\kappa^2) + 2b^2 + \frac{2\beta^2\kappa^2}{(1-\beta)^2} + \frac{2\sigma^2}{(1-\beta)n}$ and $\Delta_2 = 4c\sqrt{\delta}(\sigma^2+\kappa^2) + \frac{\beta\kappa^2}{(1-\beta)^2}$.

Proof. Detailed proof is in Appendix E. □

Remark 3. *The terms Δ_1 and Δ_2 arise from the existence of Byzantine clients and are influenced by the capability of aggregation rule. When no Byzantine client exists ($\beta = 0$ and thus $\delta = 0$), we have $\Delta_2 = 0$ and the convergence is guaranteed with sufficiently small learning rate. If Byzantine clients exist ($\beta > 0$), even the defender is capable to remove all malicious gradients ($\delta = 0$), we still have $\Delta_2 > 0$ due to non-IID data and may result in some model accuracy gaps to benchmark results.*

V. Experimental Setup

The proposed SignGuard framework is evaluated on various datasets for image and text classification tasks. We mainly implement the learning tasks in IID fashion, and investigate the performance of different defenses in non-IID settings as well. The models that trained under no attack and no defense are used as benchmarks. All evaluated attack and defense algorithms are implemented in PyTorch.

A. Datasets and Models

MNIST. MNIST is a 10-class digit image classification dataset, which consists of 60,000 training samples and 10,000 test samples, and each sample is a grayscale image of size 28 × 28. For MNIST, we construct a convolutional neural network (CNN) as the global model (see Appendix A1).

Fashion-MNIST. Fashion-MNIST[52] is a clothing image classification dataset, which has exactly the same image size and structure of training and testing splits as MNIST, and we use the same CNN as global model.

CIFAR-10. CIFAR-10 [53] is a well-known color image classification dataset with 60,000 32 × 32 RGB images in 10 classes, including 50,000 training samples and 10,000 test samples. We use ResNet-18 [54] as the global models[3].

AG-News. AG-News is a 4-class topic classification dataset. Each class contains 30,000 training samples and 1,900 testing samples. The total number of training samples is 120,000 and 7,600 for test. We use a TextRNN that consists of two-layer bi-directional LSTM network [55] as the global model.

B. Evaluated Attacks

We consider various popular model poisoning attacks in literature as well as recently proposed state-of-the-art attacks as introduced in Section II, and we assume the attacker knows all the benign gradients and the GAR in server.

Random Attack. The Byzantine clients send gradients with randomized values that generated by a multi-dimensional Gaussian distribution $\mathcal{N}(\mu, \sigma^2\mathbf{I})$. In our experiments, we take $\mu = (0, ..., 0) \in \mathbb{R}^d$ and $\sigma = 0.5$ to conduct random attacks.

Noise Attack. The Byzantine clients send noise perturbed gradients that generated by adding Gaussian noise into honest gradients: $g_m = g_b + \mathcal{N}(\mu, \sigma^2\mathbf{I})$. We take the same Gaussian distribution parameters as random attack.

Sign-Flipping. The Byzantine clients send reversed gradients without scaling: $g_m = -g_b$. This is a special case of reversed gradient attack [40] or empire attack [56].

Label-Flipping. The Byzantine clients flip the local sample labels during training process to generate faulty gradient. This is also a type of data poisoning attack. In particular, the label of each training sample in Byzantine clients is flipped from

[3] We use open-source implementation of ResNet-18, which is available at https://github.com/kuangliu/pytorch-cifar

l to $C-1-l$, where C is the total categories of labels and $l \in \{0, 1, \cdots, C-1\}$.

Little is Enough. As in [24], the Byzantine clients send malicious gradient vector with elements crafted as Eq. (1). We set $z = 0.3$ for default training settings in our experiments.

ByzMean Attack. As introduced in Section III, we set $m_1 = \lfloor 0.8m \rfloor$ and $m_2 = m - m_1$, and set g_{m_1} as LIE attack in all experiments.

Min-Max/Min-Sum. As in [25], the malicious gradient is a perturbed version of the benign aggregate as Eq. (21), where ∇^p is a perturbation vector and γ is a scaling coefficient, and those two attacks are formulated in Eq. (22)-(23). The first Min-Max attack ensures that the malicious gradients lie close to the clique of the benign gradients, while the Min-Sum attack ensures that the sum of squared distances of the malicious gradient from all the benign gradients is upper bounded by the sum of squared distances of any benign gradient from the other benign gradients. To maximize the attack impact, all malicious gradients keep the same. By default, we choose ∇^p as $-std(g^{\{i \in [n]\}})$, i.e., the inverse standard deviation.

$$g_m = f_{avg}(g^{\{i \in [n]\}}) + \gamma \nabla^p \tag{21}$$

$$\arg\max_{\gamma} \max_{i \in [n]} \|g_m - g^{(i)}\| \leq \max_{i,j \in [n]} \|g^{(i)} - g^{(j)}\| \tag{22}$$

$$\arg\max_{\gamma} \sum_{i \in [n]} \|g_m - g^{(i)}\|^2 \leq \max_{i \in [n]} \sum_{j \in [n]} \|g^{(i)} - g^{(j)}\|^2 \tag{23}$$

Specially, we investigate the fixed and randomized attacking behaviors respectively. In fixed settings, all corrupted clients play the role of Byzantine nodes and always perform the predefined attack method during the whole training process. In randomized settings, all corrupted clients will change their collusion attack strategy at each training epoch.

C. Training Settings

By default, we assume there are $n = 50$ clients in total for each task, 20% of which are Byzantine nodes with fixed attack method, and the training data are IID among clients. To verify the resilience and robustness, we will also evaluate the impact of different fractions of malicious clients for different attacks and defenses. Furthermore, our approach will also be evaluated in non-IID settings. In all experiments, we set the lower and upper bounds of gradient norm as $L = 0.1$ and $R = 3.0$, and randomly select 10% of coordinates to compute sign statistics in our SignGuard-based algorithms. Each training procedure is run for 60 epochs for MNIST/Fashion-MNIST/AG-News and 160 epochs for CIFAR-10, and local iteration is always set to 1. We employ momentum in PS side and the momentum parameter is set to 0.9, and weight decay is set to 0.0005. More details on some key hyper-parameters are described in Appendix A2

D. Performance Metrics

We train the models for a fixed number of epochs and use the test accuracy to evaluate the model performance. Considering the instability of model training and the fluctuation of model accuracy under strong attacks, we test the training model at the end of each training epoch and take the best test accuracy during the whole training process to assess the efficacy of defenses. We repeat each experiment for three times and report the average results. When a certain defense is performed, the accuracy gap to the baseline can be utilized to evaluate the efficacy of defense under various attacks, and smaller gap indicates more effective defense method.

VI. Evaluation Results

In this section, we conduct extensive experiments with various attack-defense pairs on both IID and non-IID data settings. We compare our methods with several existing defense methods, including TrMean, Median, GeoMed, Multi-Krum, Bulyan and DnC. The numerical results demonstrate the efficacy and superiority of our proposed SignGuard framework.

A. Main Results in IID Settings

The main results of best achieved test accuracy during training process under different attack and defense methods in IID setting are collected in Table I. The results of naive *Mean* aggregation under *No Attack* are used as benchmarks. Note that we favor other defenses by assuming the defense algorithms know the fraction of Byzantine clients, which is somewhat unrealistic but intrinsically required by existing defenses. However, we do not use the Byzantine fraction information in our SignGuard-type methods, including plain SignGuard, SignGuard-Sim and SignGuard-Dist.

Sign Statistics are Powerful. Test results on four datasets consistently show that our SignGuard-type methods can leverage the power of sign statistics and similarity features to filter out most malicious gradients and achieve comparable test accuracy as general distributed SGD under no attack. Consistent with original papers [24], [25], the state-of-the-art attacks, such as LIE and Min-Max/Min-Sum, can circumvent the median-based and distance-based defenses, preventing successful model training. Take the results of Multi-Krum on ResNet-18 as example, it can be seen that when no attack is performed, Multi-Krum has negligible accuracy drop (less than 0.1%). However, the best test accuracy drops to 42.58% under LIE attack and even less than 40% under Min-Max/Min-Sum attacks. Similar phenomena can also be found in model training under TrMean, Median and Bulyan methods. Besides, even under no attack, the Median and GeoMed methods are only effective in simple tasks, such as CNN for digit classification on MNIST and TextRNN for text classification on AG-News. When applied to complicated model training, such as ResNet-18 on CIFAR-10, those two methods have high convergence error and result in significant model degradation. While Muti-Krum and Bulyan suffer from well-crafted attacks, they perform well on naive attacks and even better than our plain SignGuard in mitigating random noise and sign-flip attack. Though the DnC method has extraordinary effectiveness under many attacks, we found it is unstable during training and can be easily broken by our proposed ByzMean attack. In contrast, our proposed SignGuard-type methods is able to

TABLE I

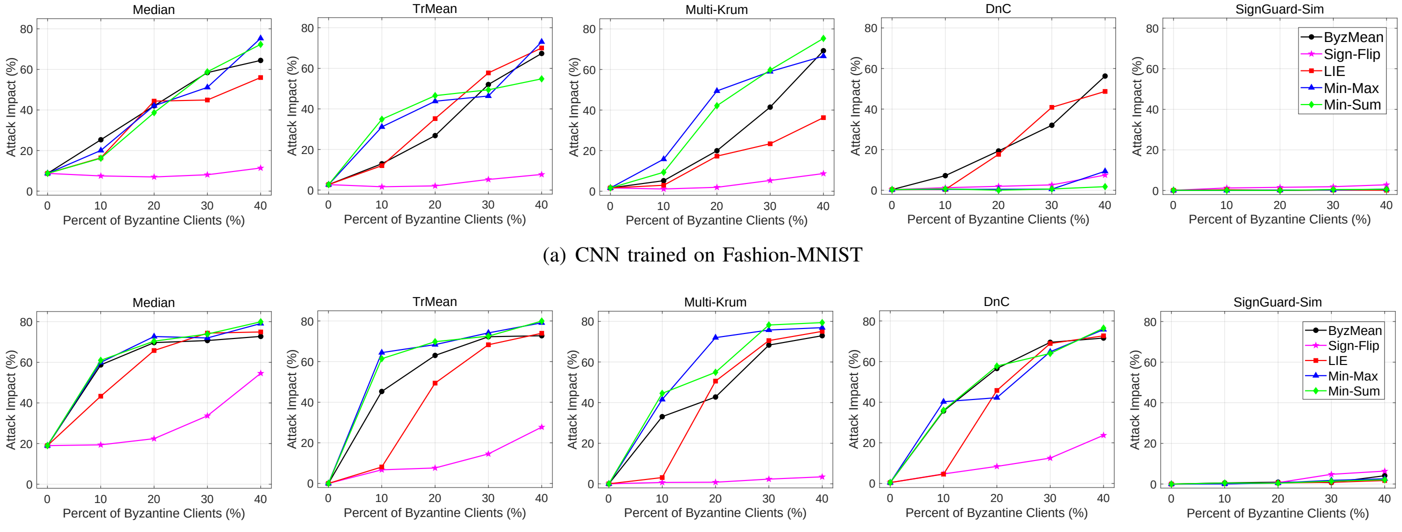
Comparison of defenses under various model poisoning attacks

Dataset (Model)	GAR	No Attack	Simple Attacks			State-of-the-art Attacks				
			Random	Noise	Label-flip	ByzMean	Sign-flip	LIE	Min-Max	Min-Sum
MNIST (CNN)	Mean	99.23	84.84	90.48	99.05	31.98	98.42	84.49	68.89	34.46
	TrMean	98.23	98.63	98.53	95.31	58.87	98.44	94.50	34.48	43.89
	Median	97.46	94.18	97.45	93.84	40.04	97.73	74.37	26.11	38.13
	GeoMed	93.21	82.77	78.68	86.20	45.02	74.78	34.37	15.62	20.53
	Multi-Krum	99.20	98.98	99.11	99.06	83.26	98.82	90.04	52.77	27.27
	Bulyan	99.10	99.17	99.12	99.15	98.58	98.81	98.86	52.45	51.95
	DnC	99.09	99.07	99.08	99.17	82.25	98.73	99.12	98.97	81.04
	SignGuard	99.11	99.09	98.97	**99.18**	**99.02**	**99.13**	99.15	**99.18**	99.15
	SignGuard-Sim	99.16	**99.18**	99.16	99.07	98.91	99.06	**99.22**	99.08	99.13
	SignGuard-Dist	98.95	99.05	**99.18**	99.11	98.93	98.86	98.96	99.01	**99.19**
Fashion-MNIST (CNN)	Mean	89.51	69.88	31.83	89.37	16.31	86.68	79.78	47.73	45.12
	TrMean	87.02	87.81	87.45	79.58	62.66	87.45	54.28	45.71	42.96
	Median	80.77	82.96	82.59	77.41	47.46	82.52	45.14	47.43	50.83
	GeoMed	76.51	79.96	78.93	78.16	40.51	70.65	10.00	73.75	66.63
	Multi-Krum	87.89	89.12	88.94	89.27	69.95	87.59	72.22	40.08	47.36
	Bulyan	88.80	89.31	89.32	89.21	88.72	87.52	88.64	59.65	43.63
	DnC	89.21	88.89	88.14	88.85	70.15	87.58	71.82	88.43	88.94
	SignGuard	89.48	**89.34**	**89.32**	89.12	89.35	88.69	89.34	**89.48**	**88.51**
	SignGuard-Sim	89.43	89.24	89.21	**89.33**	89.28	89.08	**89.36**	89.04	88.18
	SignGuard-Dist	89.37	88.87	89.30	89.31	**89.39**	**89.21**	**89.36**	89.34	88.38
CIFAR-10 (ResNet-18)	Mean	93.16	44.53	46.34	91.98	17.18	79.63	55.86	23.84	18.17
	TrMean	93.15	89.61	89.47	85.15	30.13	85.54	43.76	24.81	23.36
	Median	74.18	68.27	71.42	71.19	23.47	70.75	27.35	20.46	22.74
	GeoMed	65.62	70.41	69.35	70.76	24.86	67.82	23.55	50.36	45.23
	Multi-Krum	93.14	**92.88**	**92.91**	92.26	50.41	92.36	42.58	21.17	38.24
	Bulyan	92.78	91.87	92.47	92.24	81.33	90.12	74.52	29.87	37.79
	DnC	92.73	88.01	88.25	92.05	36.56	84.76	47.37	52.94	35.36
	SignGuard	93.03	**92.78**	92.52	92.28	**92.46**	88.61	**92.93**	92.56	92.47
	SignGuard-Sim	93.19	92.51	91.38	92.26	92.26	**92.48**	92.62	92.63	92.75
	SignGuard-Dist	92.76	92.64	92.26	**92.51**	92.42	91.69	92.36	**92.82**	**92.93**
AG-News (TextRNN)	Mean	89.36	28.18	28.41	86.72	25.05	84.18	79.34	27.32	25.24
	TrMean	87.57	88.33	88.72	85.50	37.51	84.84	66.95	30.05	30.28
	Median	84.57	84.52	84.59	82.08	28.99	81.10	32.39	30.28	29.71
	GeoMed	82.38	77.63	77.18	78.42	27.36	81.64	31.57	74.82	71.48
	Multi-Krum	88.86	89.18	89.22	86.89	68.53	**87.42**	72.98	53.51	32.46
	Bulyan	88.22	88.86	88.93	85.54	85.80	86.55	85.49	47.76	51.25
	DnC	89.13	86.42	86.28	86.72	31.47	86.30	76.58	88.45	89.05
	SignGuard	89.29	**89.22**	89.23	86.78	**89.24**	86.53	89.26	89.23	89.27
	SignGuard-Sim	89.24	89.13	**89.29**	87.05	89.36	86.76	**89.33**	**89.27**	**89.37**
	SignGuard-Dist	89.23	89.16	89.23	**87.25**	89.31	87.30	89.17	89.22	89.35

distinguish most of those well-crafted malicious gradients and achieve satisfactory model accuracy under various types of attacks. Considering that the local data of Byzantine clients also contribute to global model when no attack is performed, it's not surprising to see that even the best defense against Byzantine attack will still result in small gap to the benchmark results.

Sign Statistics are Insufficient. Table II reports the average selected rate of both benign and Byzantine clients during the training process of ResNet-18. We notice that the SignGuard-type methods inevitably exclude part of honest gradients, and select some malicious gradients under the sign-flip attack, even with the help of similarity feature. The reason is that the proportions of positive and negative elements in normal gradient are approximate for ResNet-18, even after randomized downsampling of gradient elements. Consequently, the ratios of positive and negative signs remain approximate in the sign-flipped gradient. Therefore, the simple sign statistics are insufficient to make a distinction between those honest gradients and sign-flipped ones. We also notice that although SignGuard-Sim is resilient to all kinds of attacks and achieves high accuracy results, it only selects less than 80% honest gradients during training. One possible reason is that the cosine-similarity feature also has some diversity across honest gradients.

Percentage of Byzantine Clients. We also evaluate the performance of signGuard-Sim with different percentages of Byzantine clients. In this part, we conduct experiments of CNN trained on the Fashion-MNIST dataset and ResNet-18 trained on CIFAR-10 dataset. We keep the total number of clients be 50 and vary the fraction of Byzantine clients from 10% to 40% to study the impact of Byzantine percentage



(a) CNN trained on Fashion-MNIST

(b) ResNet-18 trained on CIFAR-10

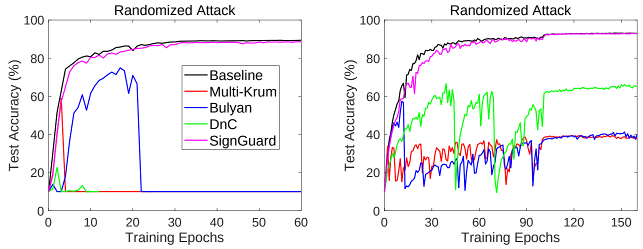
Fig. 4. Accuracy drop comparison under various attacks and different percentage of Byzantine clients. SignGuard has the smallest gap to the baseline.

TABLE II
SELECTED RATE OF HONEST AND MALICIOUS GRADIENTS

Attack	SignGuard		SignGuard-Sim		SignGuard-Dist	
	H	M	H	M	H	M
ByzMean	0.9625	0	0.7791	0	0.9272	0.0003
Sign-flip	0.6870	0.3908	0.7639	0.0981	0.7570	0.2440
LIE	0.9532	0	0.7727	0	0.9151	0
Min-Max	0.9650	0	0.7866	0.0003	0.9105	0.0009
Min-Sum	0.9640	0	0.7752	0	0.9111	0

for different defenses. We use the default training settings, and experiments are conducted under various state-of-the-art attacks. Particularly, we compare the results of SignGuard-Sim with Median, TrMean, Multi-Krum and DnC as shown in Fig. 4. It can be seen that our approach can effectively filter out malicious gradients and result in slight accuracy drop regardless of the high percentage of Byzantine clients, while other defense algorithms suffer much more attack impact with increasing percentage of Byzantine clients. In particular, we also find that Multi-Krum can mitigate sign-flip attack well in ResNet-18 training, possibly because the exact percentage of Byzantine clients is provided.

Time-varying Attack Strategy. Further, we test different defense algorithms under time-varying Byzantine attack strategy. We still use the default system setting, and change attack method randomly at each epoch (including no attack scenario). The test accuracy curves of CNN on Fashion-MNIST and ResNet-18 on CIFAR-10 are presented in Fig. 5, where the baseline is training under no attack and no defense, and we only test the State-of-the-art defenses. It can be found that our SignGuard could ensure successful model training and closely follow the baseline, while other defenses resulted in significant accuracy fluctuation and model deterioration. For CNN, the training process even collapsed eventually for other defenses



(a) CNN on Fashion-MNIST

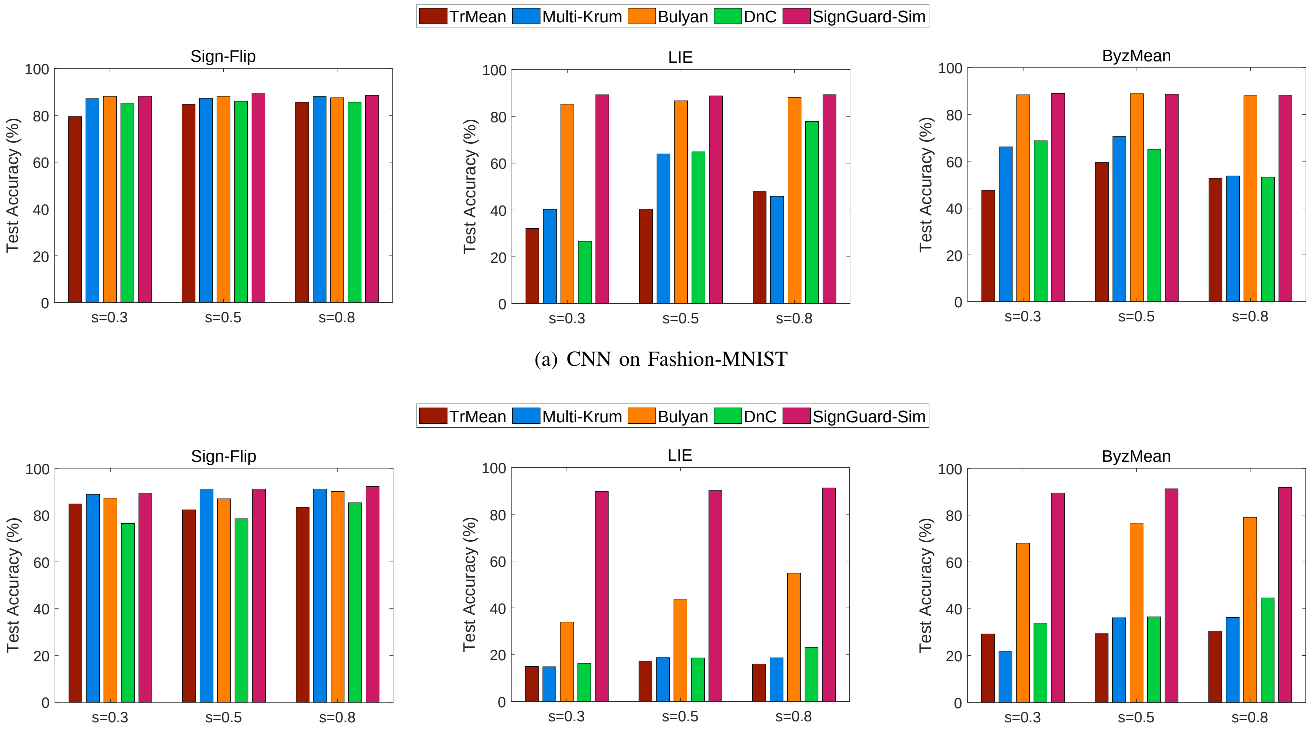
(b) ResNet-18 on CIFAR-10

Fig. 5. Defense effect comparison under time-varying attacks. SignGuard can ensure safe training and achieve decent model accuracy.

B. Main Results in Non-IID Settings

The Byzantine-mitigation in non-IID FL settings has been a well-known challenging task due to the diversity of gradients. We evaluate our SignGuard-Sim method in synthetic non-IID partition of Fashion-MNIST and CIFAR-10 datasets. As previous works, we simulate the non-IID data distribution between clients by allocating s-fraction of dataset in a IID fashion and the remaining (1-s)-fraction in a sort-and-partition fashion. Specifically, we first randomly select s-proportion of the whole training data and evenly distribute them to all clients. Then, we sort the remaining data by labels and divide they into multiple shards, while data in the same shard has the same label, after which each client is randomly allocated with 2 different shards. The parameter s can be used to measure the skewness of data distribution and smaller s will generate more skewed data distribution among clients. We consider three levels for the skewness with s = 0.3, 0.5, 0.8, respectively.

Efficacy on Non-IID Data. We compare the SignGuard-Sim with various start-of-the-art defenses. As shown in Fig. 6,



(a) CNN on Fashion-MNIST

(b) CifarNet on CIFAR-10

Fig. 6. Model accuracy comparison under various attacks and different degrees of non-IID. SignGuard has the best performance compared with other start-of-the-art defenses.

our method still works well under strong attacks in non-IID settings, achieving satisfactory accuracy results in various scenarios. In contrast, TrMean and Multi-Krum could not defend LIE attack and ByzMean attack, making them not reliable any more. Bulyan has good performance on CNN trained on Fashion-MNIST, but is ineffective under LIE attack on ResNet-18 trained on CIFAR-10. DnC can defend against sign-flip attack well, but performs poorly on the other scenarios. Those results in non-IID settings further demonstrate the general validness of sign statistics.

TABLE III
Averaged Aggregation Time

Method	GeoMed	Multi-Krum	Bulyan	SignGuard
Time(s)	0.39314	0.29847	0.29629	0.04706

Method	SignGuard-Sim		SignGuard-Dist	
	pairwise	previous	pairwise	previous
Time(s)	0.73887	0.07686	0.39117	0.07834

C. Computational Overhead Comparison

The following Table III reports the averaged aggregation time of different defenses on training ResNet-18, where we omit the TrMean and Median since they induce negligible computation cost. It can be seen that SignGuard resulted in shortest time compared with GeoMed, Multi-Krum and Bulyan, which means our method can achieve efficiency and robustness simultaneously. For the other two variants, we found the pairwise similarity/distance calculation is time-consuming and using the previous aggregate as correct gradient to compute similarity/distance could alleviate this issue.

VII. Conclusion and Future Work

In this work, we proposed a novel Byzantine attack detection framework, namely SignGuard, to mitigate malicious gradients in federated learning systems. It can overcome the drawbacks of median- and distance-based approaches which are vulnerable to well-crafted attacks and unlike validation-based approaches that require extra data collection in PS. And it also does not depend on historical data or other external information, only utilizing magnitude and robust sign statistics from current local gradients, making it a practical way to defend most kinds of model poisoning attacks. Extensive experimental results on image and text classification tasks verify our theoretical and empirical findings, demonstrating the extraordinary effectiveness of our proposed SignGuard-type algorithms. We hope this work can provide a new perspective on the Byzantine attack problems in machine learning security. Future directions include developing strategies to defend dynamic and hybrid model poisoning attacks as well as backdoor attacks in more complex federated learning scenarios. And how to design more effective and robust filters in the SignGuard framework for real-world learning systems is also left as an open problem.

References

[1] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 54, 2017, pp. 1273–1282.

[3] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, 2019.

[4] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, and et al, "Advances and open problems in federated learning," *arXiv:1912.04977*, 2019.

[5] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

[6] L. Lamport, R. E. Shostak, and M. C. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, 1982.

[7] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[8] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. B. Calo, "Analyzing federated learning through an adversarial lens," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[9] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," 2020. [Online]. Available: https://arxiv.org/abs/2003.02133

[10] L. Lyu, H. Yu, X. Ma, L. Sun, J. Zhao, Q. Yang, and P. S. Yu, "Privacy and robustness in federated learning: Attacks and defenses," *CoRR*, vol. abs/2012.06337, 2020. [Online]. Available: https://arxiv.org/abs/2012.06337

[11] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Madry, B. Li, and T. Goldstein, "Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses," vol. abs/2012.10544, 2020. [Online]. Available: https://arxiv.org/abs/2012.10544

[12] S. Shen, S. Tople, and P. Saxena, "Auror: defending against poisoning attacks in collaborative deep learning systems," in *Proceedings of Conference on Computer Security Applications, ACSAC*. ACM, 2016, pp. 508–519.

[13] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *29th USENIX Security Symposium*, 2020.

[14] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[15] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[16] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[17] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng, "Large scale distributed deep networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1223–1231.

[18] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *International Conference on Machine Learning (ICML)*, vol. 80, 2018, pp. 560–569.

[19] D. Yin, Y. Chen, K. Ramchandran, and P. L. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[20] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 2, pp. 44:1–44:25, 2017.

[21] C. Xie, S. Koyejo, and I. Gupta, "Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance," in *Proceedings of the 36th International Conference on Machine Learning, (ICML)*, 2019.

[22] X. Cao and L. Lai, "Distributed gradient descent algorithm robust to an arbitrary number of byzantine attackers," *IEEE Trans. Signal Process.*, vol. 67, no. 22, pp. 5850–5864, 2019.

[23] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "FLTrust: Byzantine-robust federated learning via trust bootstrapping," *ISOC Network and Distributed Systems Security (NDSS) Symposium*, 2021.

[24] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[25] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," *ISOC Network and Distributed Systems Security (NDSS) Symposium*, 2021.

[26] E. M. E. Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in byzantium," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[27] C. Ma, J. Li, M. Ding, H. H. Yang, F. Shu, T. Q. S. Quek, and H. V. Poor, "On safeguarding privacy and security in the framework of federated learning," *IEEE Netw.*, vol. 34, no. 4, pp. 242–248, 2020.

[28] S. Sharma, *Data privacy and GDPR handbook*. John Wiley & Sons, 2019.

[29] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[30] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017.

[31] D. Gao, Y. Liu, A. Huang, C. Ju, H. Yu, and Q. Yang, "Privacy-preserving heterogeneous federated transfer learning," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019.

[32] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 1544–1551.

[33] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," in *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[34] Z. Allen-Zhu, F. Ebrahimian, J. Li, and D. Alistarh, "Byzantine-resilient non-convex stochastic gradient descent," 2020. [Online]. Available: https://arxiv.org/abs/2012.14368

[35] L. Muñoz-González, K. T. Co, and E. C. Lupu, "Byzantine-robust federated machine learning through adaptive model averaging," 2019. [Online]. Available: http://arxiv.org/abs/1909.05125

[36] S. P. Karimireddy, L. He, and M. Jaggi, "Learning from history for byzantine robust optimization," 2020. [Online]. Available: https://arxiv.org/abs/2012.10333

[37] E. El-Mhamdi, R. Guerraoui, and S. Rouault, "Distributed momentum for byzantine-resilient stochastic gradient descent," in *International Conference on Learning Representations (ICLR)*, 2021.

[38] L. Chen, H. Wang, Z. B. Charles, and D. S. Papailiopoulos, "DRACO: byzantine-resilient distributed training via redundant gradients," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[39] D. Data, L. Song, and S. N. Diggavi, "Data encoding for byzantine-resilient distributed optimization," *IEEE Trans. Inf. Theory*, vol. 67, no. 2, pp. 1117–1140, 2021.

[40] S. Rajput, H. Wang, Z. B. Charles, and D. S. Papailiopoulos, "DETOX: A redundancy-based framework for faster and more robust gradient aggregation," in *Advances in Neural Information Processing Systems*, 2019.

[41] S. Li, Y. Cheng, W. Wang, Y. Liu, and T. Chen, "Learning to detect malicious clients for robust federated learning," 2020. [Online]. Available: https://arxiv.org/abs/2002.00211

[42] X. Pan, M. Zhang, D. Wu, Q. Xiao, S. Ji, and M. Yang, "Justinian's gaavernor: Robust distributed learning with gradient aggregation agent," in *29th USENIX Security Symposium*, 2020, pp. 1641–1658.

[43] J. Regatti, H. Chen, and A. Gupta, "ByGARS: Byzantine sgd with arbitrary number of attackers," *NeurIPS 2020 Workshop on Scalability, Privacy, and Security in Federated Learning*, 2020.

[44] X. Cao, J. Jia, and N. Z. Gong, "Provably secure federated learning against malicious clients," *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[45] X. Qiao, Y. Bai, S. Hu, A. Li, Y. Chen, and H. Li, "On provable backdoor defense in collaborative learning," *CoRR*, vol. abs/2101.08177, 2021. [Online]. Available: https://arxiv.org/abs/2101.08177

[46] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[47] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization," in *International Conference on Machine Learning (ICML)*, vol. 97, 2019, pp. 7184–7193.

[48] H. Chang, V. Shejwalkar, R. Shokri, and A. Houmansadr, "Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer," 2019. [Online]. Available: http://arxiv.org/abs/1912.11279

[49] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002. [Online]. Available: https://doi.org/10.1109/34.1000236

[50] S. P. Karimireddy, Q. Rebjock, S. U. Stich, and M. Jaggi, "Error feedback fixes signsgd and other gradient compression schemes," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[51] K. A. Lai, A. B. Rao, and S. S. Vempala, "Agnostic estimation of mean and covariance," in *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS, USA*. IEEE Computer Society, 2016, pp. 665–674.

[52] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017. [Online]. Available: http://arxiv.org/abs/1708.07747

[53] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *University of Toronto*, 2009.

[54] K. He, X. Zhang, S. Ren, and S. Jian, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[55] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016*, 2016, pp. 2873–2879.

[56] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking byzantine-tolerant SGD by inner product manipulation," in *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019.

APPENDIX

A. More Details of Experiments

1) Architecture of CNN: We construct a plain CNN model on MNIST dataset and Fashion-MNIST dataset. The architecture of CNN model is presented in the following table:

TABLE IV
MODEL ARCHITECTURE FOR (FASHION-)MNIST

Layer Type	# of Channels	Filter Size	Stride	Activation
Conv	8	5	1	ReLU
MaxPool	8	2	2	–
Conv	16	3	1	ReLU
MaxPool	16	2	2	–
Conv	32	3	1	ReLU
MaxPool	32	2	2	–
FC	120	–	–	ReLU
FC	84	–	–	ReLU
FC	10	–	–	Softmax

2) More Details of Training Settings: Table V summarizes the default federated learning system settings and hyper-parameter details.

TABLE V
TRAINING SETTINGS ON BENCHMARK DATASETS

	MNIST	**F-MNIST**	**CIFAR-10**	**AG-News**
# Total clients	50	50	50	50
# Byzantine clients	10	10	10	10
Global Model	CNN	CNN	ResNet-18	TextRNN
# Parameters	57,620	57,620	11,173,962	3,837,212
Batch-size	60	60	50	20
Initial learning rate	0.1	0.1	0.1	0.1

3) More Details of Our Methods: For SignGuard-Sim algorithm, we clip the similarity feature values to the range of $[0, +med]$, where med is the median of similarity values, and then rescale them into the range of $[0, 0.2]$. We do not directly employ the similarity values, because the absolute true values are too small compared with sign statistics and thus problematic for clustering algorithm. Besides, in the first 3 epochs, we use the median of similarities across other gradients as the similarity feature. Afterwards, we directly compute the cosine-similarity with the aggregation result from previous iteration to alleviate computation cost. For SignGuard-Dist algorithm, we compute a distance score for each gradient as that in Krum and use the median of scores as a threshold. Then for each gradient, the distance feature is set to 0 if the score is less than the threshold, otherwise will be set to 0.2 instead.

B. Proof of Proposition 1

Notice that the standard deviation is estimated on distributed gradients, that is:

$$\|std(g^{\{i\in[n]\}})\|^2 = \frac{1}{n}\sum_{i=1}^{n}\|g^{(i)} - \frac{1}{n}\sum_{j=1}^{n}g^{(j)}\|^2$$

so we have:

$$\begin{aligned}
&\mathbb{E}[\|g_m - \tilde{g}\|^2] = \mathbb{E}[\|z \cdot std(g^{\{i\in[n]\}})\|^2] \\
&= \mathbb{E}[\frac{z^2}{n}\sum_{i=1}^{n}\|g^{(i)} - \frac{1}{n}\sum_{j=1}^{n}g^{(j)}\|^2] \\
&= \mathbb{E}[\frac{z^2}{n}\sum_{i=1}^{n}\|g^{(i)} - \nabla F(\mathbf{x}) + \nabla F(\mathbf{x}) - \frac{1}{n}\sum_{j=1}^{n}g^{(j)}\|^2] \\
&\leq \mathbb{E}[\frac{z^2}{n}\sum_{i=1}^{n}\|g^{(i)} - \nabla F(\mathbf{x})\|^2 + \|\nabla F(\mathbf{x}) - \frac{1}{n}\sum_{j=1}^{n}g^{(j)}\|^2] \\
&\leq \left(1 + \frac{1}{n}\right) z^2\sigma^2
\end{aligned}$$

and:

$$\begin{aligned}
&\mathbb{E}[\|g_m - \nabla F(\mathbf{x})\|^2] = \mathbb{E}[\|g_m - \tilde{g} + \tilde{g} - \nabla F(\mathbf{x})\|^2] \\
&\leq \mathbb{E}[\|g_m - \tilde{g}\|^2 + \|\tilde{g} - \nabla F(\mathbf{x})\|^2] \\
&\leq \left(1 + \frac{1}{n}\right) z^2\sigma^2 + \frac{1}{n}\sigma^2 \\
&\leq \left(z^2 + \frac{1 + z^2}{n}\right)\sigma^2
\end{aligned}$$

Taking the total expectations of averaged gradient on local sampling, we have

$$
\begin{aligned}
&\mathbb{E}_t[F(\mathbf{x}_{t+1})] - F(\mathbf{x}_t) \\
&\leq -\eta \left\langle \nabla F(\mathbf{x}_t), \mathbb{E}_t\left[g_{m,t}\right] \right\rangle + \frac{L\eta^2}{2}\mathbb{E}_t\left[\left\|g_{m,t} - \nabla F(\mathbf{x}_t)\right\|^2\right] \\
&+ L\eta^2 \mathbb{E}_t \left\langle \nabla F(\mathbf{x}_t), g_{m,t} - \nabla F(\mathbf{x}_t) \right\rangle + \frac{L\eta^2}{2}\mathbb{E}_t\left[\left\|\nabla F(\mathbf{x}_t)\right\|^2\right] \\
&\text{(substitute} \quad \tilde{g}_t - z \cdot std(g_t^{\{i \in [n]\}}) \quad \text{for} \quad g_{m,t}) \\
&\leq -\eta(1 - \frac{L\eta}{2}) \left\|\nabla F(\mathbf{x}_t)\right\|^2 + \frac{L\eta^2}{2}\left(z^2 + \frac{1+z^2}{n}\right)\sigma^2 \\
&\qquad + \eta(1 - L\eta)\left\langle \nabla F(\mathbf{x}_t), \mathbb{E}_t\left[z \cdot std(g_t^{\{i \in [n]\}})\right]\right\rangle
\end{aligned}
$$

Applying the basic inequality $2\mathbf{a} \cdot \mathbf{b} \leq \mathbf{a}^2 + \mathbf{b}^2$, we have

$$
\begin{aligned}
&\mathbb{E}_t[F(\mathbf{x}_{t+1})] - F(\mathbf{x}_t) \\
&\leq -\eta(1 - \frac{L\eta}{2}) \left\|\nabla F(\mathbf{x}_t)\right\|^2 + \frac{L\eta^2}{2}\left(z^2 + \frac{1+z^2}{n}\right)\sigma^2 \\
&\qquad + \frac{\eta(1-L\eta)}{2}\left\|\nabla F(\mathbf{x}_t)\right\|^2 + \frac{\eta(1-L\eta)}{2}\left(1 + \frac{1}{n}\right)z^2\sigma^2 \\
&\leq -\frac{\eta}{2}\left\|\nabla F(\mathbf{x}_t)\right\|^2 + \frac{\eta}{2}\left(1 + \frac{1}{n}\right)z^2\sigma^2 + \frac{L\eta^2}{2n}\sigma^2
\end{aligned}
$$

Taking total expectation and rearranging the terms, we get

$$
\begin{aligned}
\frac{\eta}{2}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \leq &\mathbb{E}[F(\mathbf{x}_t) - F(\mathbf{x}_{t+1})] \\
&+ \frac{\eta}{2}\left(1 + \frac{1}{n}\right)z^2\sigma^2 + \frac{L\eta^2}{2n}\sigma^2
\end{aligned}
$$

Taking summation and dividing by $\frac{\eta}{2}T$, then we get

$$
\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \leq &\frac{2(F(\mathbf{x}_0) - F^*)}{\eta T} \\
&+ \frac{L\eta\sigma^2}{n} + \left(1 + \frac{1}{n}\right)z^2\sigma^2
\end{aligned}
$$

which completes the proof.

C. Proof of Proposition 2

In Appendix B, we already show that:

$$
\mathbb{E}[\|g_m - \tilde{g}\|^2] \leq \left(1 + \frac{1}{n}\right)z^2\sigma^2
$$

and it's easy to see that:

$$
\begin{aligned}
\mathbb{E}[\|g^{(i)} - \tilde{g}\|^2] &= \mathbb{E}[\|g^{(i)} - \frac{1}{n}\sum_{j=1}^{n} g^{(j)}\|^2] \\
&= \mathbb{E}[\|g^{(i)} - \nabla F(\mathbf{x}) + \nabla F(\mathbf{x}) - \frac{1}{n}\sum_{j=1}^{n} g^{(j)}\|^2] \\
&\leq \mathbb{E}[\|g^{(i)} - \nabla F(\mathbf{x})\|^2 + \|\nabla F(\mathbf{x}) - \frac{1}{n}\sum_{j=1}^{n} g^{(j)}\|^2] \\
&\leq \left(1 + \frac{1}{n}\right)\sigma^2
\end{aligned}
$$

Hence, given small enough z, it's possible for the malicious gradient to have smaller distance from true averaged gradient than that of some honest gradients.

Next, we can express the cosine-similarity between malicious gradient and true averaged gradient as well as that of honest gradient as follows:

$$
\cos(g_m, \tilde{g}) = \frac{\|g_m\|^2 + \|\tilde{g}\|^2 - \|g_m - \tilde{g}\|^2}{2\|g_m\|\|\tilde{g}\|}
$$

$$
\cos(g^{(i)}, \tilde{g}) = \frac{\left\|g^{(i)}\right\|^2 + \|\tilde{g}\|^2 - \left\|g^{(i)} - \tilde{g}\right\|^2}{2\left\|g^{(i)}\right\|\|\tilde{g}\|}
$$

We can prove that it's possible for the norm of malicious gradient and the norm of certain honest gradient to have following relations:

$$
\|g_m\| = \xi_m\|\tilde{g}\|, \quad \|g^{(i)}\| = \xi_i\|\tilde{g}\|, \quad 1 \leq \xi_i < \xi_m
$$

By Jensen inequality, we have:

$$
\|\tilde{g}\| = \left\|\frac{1}{n}\sum_{i=1}^{n} g^{(i)}\right\| \leq \frac{1}{n}\sum_{i=1}^{n}\left\|g^{(i)}\right\| \leq \max\{\|g^{(i)}\|\}
$$

which means the norm of true averaged gradient is smaller than the averaged norm of honest gradient, so some honest gradients could have bigger norm than $\tilde{g}$, i.e. $\xi_i \geq 1$.

And a appropriate value of z can make $\xi_m > \xi_i$. It's easy to see that:

$$
\begin{aligned}
&\|g_m\|^2 > \xi_i^2\|\tilde{g}\|^2 \\
\iff \quad &\sum_{j=1}^{d}(\mu_j - z\sigma_j)^2 > \xi_i^2\sum_{j=1}^{d}(\mu_j)^2 \\
\iff \quad &\sum_{j=1}^{d}(\mu_j^2 - 2z\mu_j\sigma_j + z^2\sigma_j^2) > \xi_i^2\sum_{j=1}^{d}(\mu_j)^2 \\
\iff \quad &\sum_{j=1}^{d}(z^2\sigma_j^2) > \xi_i^2\sum_{j=1}^{d}(2z\mu_j\sigma_j) \\
\iff \quad &z > \xi_i^2\frac{\sum_{j=1}^{d}(2\mu_j\sigma_j)}{\sum_{j=1}^{d}(\sigma_j^2)}
\end{aligned}
$$

Therefore, given appropriate z, there exists i, such that $1 \leq \xi_i < \xi_m$. By using these gradient norm relations, we can get:

$$
\begin{aligned}
&\cos(g_m, \tilde{g}) - \cos(g^{(i)}, \tilde{g}) \\
&= \frac{(\xi_m^2 + 1)\|\tilde{g}\|^2 - \|g_m - \tilde{g}\|^2}{2\xi_m\|\tilde{g}\|^2} - \frac{(\xi_i^2 + 1)\|\tilde{g}\|^2 - \left\|g^{(i)} - \tilde{g}\right\|^2}{2\xi_i\|\tilde{g}\|^2} \\
&> \left(\frac{(\xi_m^2 + 1)}{2\xi_m} - \frac{(\xi_i^2 + 1)}{2\xi_i}\right) + \frac{\|g_m - \tilde{g}\|^2}{2\|\tilde{g}\|^2}\left(\frac{1}{\xi_i} - \frac{1}{\xi_m}\right) \\
&= \frac{(\xi_m - \xi_i)(\xi_m\xi_i - 1)}{2\xi_m\xi_i} + \frac{(\xi_m - \xi_i)\|g_m - \tilde{g}\|^2}{2\xi_m\xi_i\|\tilde{g}\|^2} \\
&> 0
\end{aligned}
$$

Hence, it's possible for the malicious gradient to have bigger cosine-similarity with true averaged gradient than that of some honest gradients.

D. Proof of Lemma 2

Given a arbitrary subset of clients $\mathcal{G}$ with $|\mathcal{G}| = (1-\beta)n$ and $\beta < 0.5$. Let $\mathbf{A} = \sum_{i \notin \mathcal{G}} \left(g_t^{(i)} - \nabla F(\mathbf{x}_t)\right)$, $\mathbf{B} = \sum_{j \in \mathcal{G}} \left(g_t^{(j)} - \nabla F(\mathbf{x}_t)\right)$, then $\mathbf{A}$ and $\mathbf{B}$ are independent. We have $\mathbb{E}[\mathbf{A}+\mathbf{B}] = \mathbf{0}$. Recall that σ^2 is the bounded local variance for local gradient and κ^2 is bounded deviation between local and global gradient. Applying the Jensen inequality, we have

$$\|\mathbb{E}[\mathbf{A}]\|^2 \le \beta n \sum_{i \notin \mathcal{G}} \|\nabla F_i(\mathbf{x}_t) - \nabla F(\mathbf{x}_t)\|^2 \le \beta^2 n^2 \kappa^2$$

$$\|\mathbb{E}[\mathbf{B}]\|^2 \le (1-\beta) n \sum_{i \in \mathcal{G}} \|\nabla F_i(\mathbf{x}_t) - \nabla F(\mathbf{x}_t)\|^2 \le (1-\beta)^2 n^2 \kappa^2$$

Notice that $\mathbb{E}[\mathbf{A}] = -\mathbb{E}[\mathbf{B}]$, thus

$$\|\mathbb{E}[\mathbf{A}]\|^2 = \|\mathbb{E}[\mathbf{B}]\|^2 \le \min\{\beta^2 n^2 \kappa^2, (1-\beta)^2 n^2 \kappa^2\} = \beta^2 n^2 \kappa^2$$

Using the basic relation between expectation and variance, we have

$$\mathbb{E}\|\mathbf{A}\|^2 = \|\mathbb{E}[\mathbf{A}]\|^2 + \mathrm{var}[\mathbf{A}] \le \|\mathbb{E}[\mathbf{A}]\|^2 + \beta n \sigma^2$$
$$\mathbb{E}\|\mathbf{B}\|^2 = \|\mathbb{E}[\mathbf{B}]\|^2 + \mathrm{var}[\mathbf{B}] \le \|\mathbb{E}[\mathbf{B}]\|^2 + (1-\beta) n \sigma^2$$

which leads to

$$\mathbb{E}\|\mathbf{B}\|^2 \le \beta^2 n^2 \kappa^2 + (1-\beta) n \sigma^2$$

Then, we directly have

$$\mathbb{E}\left[\left\|\frac{1}{|\mathcal{G}|}\sum_{i \in \mathcal{G}}\left(g_t^{(i)}\right) - \nabla F(\mathbf{x}_t)\right\|^2\right] = \frac{1}{(1-\beta)^2 n^2}\mathbb{E}\|\mathbf{B}\|^2$$
$$\le \frac{\beta^2 \kappa^2}{(1-\beta)^2} + \frac{\sigma^2}{(1-\beta)n}$$

It completes the proof of Lemma 2.

E. Proof of Theorem 1

Taking the total expectations of averaged gradient on local sampling and randomness in aggregation rule, we have

$$\begin{aligned}
&\mathbb{E}_t[F(\mathbf{x}_{t+1})] - F(\mathbf{x}_t) \\
&\le -\eta \langle \nabla F(\mathbf{x}_t), \mathbb{E}_t[\hat{g}_t] \rangle + \frac{L\eta^2}{2}\mathbb{E}_t\left[\|\hat{g}_t\|^2\right] \\
&= -\eta \langle \nabla F(\mathbf{x}_t), \mathbb{E}_t[\hat{g}_t - \tilde{g}_t + \tilde{g}_t - \nabla F(\mathbf{x}_t) + \nabla F(\mathbf{x}_t)] \rangle \\
&\qquad + \frac{L\eta^2}{2}\mathbb{E}_t\left[\|\hat{g}_t - \nabla F(\mathbf{x}_t) + \nabla F(\mathbf{x}_t)\|^2\right] \\
&\le -\eta \langle \nabla F(\mathbf{x}_t), \mathbb{E}_t[\hat{g}_t - \tilde{g}_t] \rangle - \eta \langle \nabla F(\mathbf{x}_t), \mathbb{E}_t[\tilde{g}_t - \nabla F(\mathbf{x}_t)] \rangle \\
&\qquad - \eta \|\nabla F(\mathbf{x}_t)\|^2 + L\eta^2 \|\nabla F(\mathbf{x}_t)\|^2 + L\eta^2 \mathbb{E}_t\left[\|\hat{g}_t - \nabla F(\mathbf{x}_t)\|^2\right]
\end{aligned}$$

From Assumption 1 & 2, we have

$$[\mathbb{E}\|\hat{g}_t - \bar{g}_t\|]^2 \le c\delta \sup_{i,j \in \mathcal{G}} \mathbb{E}[\|g_t^{(i)} - g_t^{(j)}\|^2] \le 2c\delta(\sigma^2 + \kappa^2)$$

then by Young's Inequality with $\rho = 2$, we can get

$$\begin{aligned}
&-\eta \langle \nabla F(\mathbf{x}_t), \mathbb{E}_t[\hat{g}_t - \tilde{g}_t] \rangle \\
&\le \eta \|\nabla F(\mathbf{x}_t)\| \cdot \mathbb{E}_t \|\hat{g}_t - \tilde{g}_t\| \\
&\le \frac{\sqrt{\delta}\eta}{2\rho}\|\nabla F(\mathbf{x}_t)\|^2 + \frac{\rho}{2}\cdot 2\sqrt{\delta}\eta c(\sigma^2 + \kappa^2) \\
&\le \frac{\sqrt{\delta}\eta}{4}\|\nabla F(\mathbf{x}_t)\|^2 + 2\sqrt{\delta}\eta c(\sigma^2 + \kappa^2)
\end{aligned}$$

Combining with Lemma 2, we get

$$\begin{aligned}
&-\eta \langle \nabla F(\mathbf{x}_t), \mathbb{E}_t[\tilde{g}_t - \nabla F(\mathbf{x}_t)] \rangle \\
&\le \eta \|\nabla F(\mathbf{x}_t)\| \cdot \mathbb{E}_t \|\tilde{g}_t - \nabla F(\mathbf{x}_t)\| \\
&\le \frac{\beta\eta}{2}\|\nabla F(\mathbf{x}_t)\|^2 + \frac{\beta\eta\kappa^2}{2(1-\beta)^2}
\end{aligned}$$

and

$$\begin{aligned}
&\mathbb{E}_t\left[\|\hat{g}_t - \nabla F(\mathbf{x}_t)\|^2\right] \\
&= \mathbb{E}_t\left[\|\hat{g}_t - \bar{g}_t + \bar{g}_t - \nabla F(\mathbf{x}_t)\|^2\right] \\
&\le 2\mathbb{E}_t\left[\|\hat{g}_t - \bar{g}_t\|^2\right] + 2\mathbb{E}_t\left[\|\bar{g}_t - \nabla F(\mathbf{x}_t)\|^2\right] \\
&= 2[\mathbb{E}\|\hat{g}_t - \bar{g}_t\|]^2 + 2\mathrm{var}\|\hat{g}_t\| + 2\mathbb{E}_t\left[\|\bar{g}_t - \nabla F(\mathbf{x}_t)\|^2\right] \\
&\le \underbrace{4c\delta(\sigma^2+\kappa^2) + 2b^2 + \frac{2\beta^2\kappa^2}{(1-\beta)^2} + \frac{2\sigma^2}{(1-\beta)n}}_{=\Delta_1}
\end{aligned}$$

In the above derivations, the basic inequality $2\mathbf{a}\cdot\mathbf{b} \le \mathbf{a}^2 + \mathbf{b}^2$ is applied. Taking total expectation and rearranging the terms, we get

$$\eta\left(\frac{4-\sqrt{\delta}-2\beta}{4} - L\eta\right)\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \le \mathbb{E}[F(\mathbf{x}_t) - F(\mathbf{x}_{t+1})]$$
$$+ 2\sqrt{\delta}\eta c(\sigma^2+\kappa^2) + \frac{\beta\eta\kappa^2}{2(1-\beta)^2} + L\eta^2\Delta_1$$

Assume that $\eta \le (2 - \sqrt{\delta} - 2\beta)/(4L)$, thus $\left(\frac{4-\sqrt{\delta}-2\beta}{4} - L\eta\right) \ge \frac{1}{2}$. Taking summation and dividing by $\eta\left(\frac{4-\sqrt{\delta}-2\beta}{4} - L\eta\right)T$, then we finally get

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \le \frac{2(F(\mathbf{x}_0) - F^*)}{\eta T} + 2L\eta\Delta_1$$
$$+ \underbrace{4\sqrt{\delta}c(\sigma^2+\kappa^2) + \frac{\beta\kappa^2}{(1-\beta)^2}}_{=\Delta_2}$$

which completes the proof.