协同恶意梯度过滤

用那一块的梯度值？？？首先作为第一步

带有隐私的联邦学习的检测方法，验证一下，可行不

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

Abstract*—Gradient-based training in federated learning is known to be vulnerable to faulty/malicious clients, 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 remark that auxiliary data may not always be available in practice and focus on the statistic-based approach. However, recent work on model poisoning attacks has shown that well-crafted attacks can circumvent most of 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 the *Little is Enough* attack, we propose a novel approach called *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 filters to eliminate malicious gradients before final aggregation. Finally, extensive experiments of image and text classification tasks are conducted under recently proposed attacks and defense strategies. The numerical results demonstrate the effectiveness and superiority of our proposed approach. The code is available at *https://github.com/JianXu95/SignGuard

***Index Terms*—Federated Learning, Attack Detection, Distributed Learning Security**

I. Introduction

In the era of big data, private data are often scattered among local clients (e.g., companies, mobile devices), leading to the problem of isolated data islands [1]. To fully capitalize on the value of big data while protecting data privacy and security, federated learning (FL) has attracted significant interest in recent years [1]–[5]. A typical setup of FL consists of a parameter server (PS) and a number of distributed clients, where the local training data are 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 aim to prevent successful training of the global model, which are modeled as Byzantine clients [6]–[8]. This can be seen through a simple example shown in Fig. 1 with one PS and $n-m$ benign clients as well as m Byzantine clients, where the 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. It has been shown that mitigating Byzantine model poisoning attacks is crucial for robust FL and other distributed learning [6], [9], [10]. On the other hand, distributed implementation of gradient-based learning algorithms [11] are increasingly popular for training large-scale models on distributed datasets, e.g., deep neural networks for human face identification and news sentimental analysis [12]–[14]. Therefore, many efforts have been devoted to developing robust **g**radient **a**ggregation **r**ules (GAR) [4] to achieve Byzantine-robust FL algorithms.

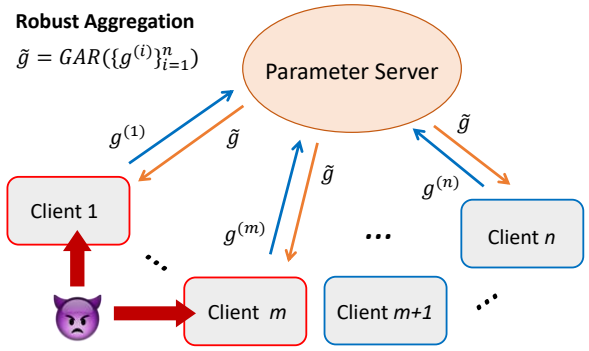


Fig. 1. Federated learning system: one parameter server with n clients, in which a attacker controls m Byzantine clients to attack the learning system.

Recently, much research attention has focused on mitigating Byzantine attacks either by leveraging statistic-based outlier detection techniques [15], [16] or by utilizing auxiliary labeled data collected by PS to verify the correctness of received gradients [17], [18]. We remark that auxiliary data sufficiently capturing the global data distribution may not be practicable to PS. And recent works have shown that existing statistic-based aggregation rules are vulnerable to well-crafted model poisoning attacks [19], [20], which are indistinguishable in Euclidean distance such that they can circumvent most defenses.

In this paper, we focus on the gradient-based FL systems and propose a novel 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 [16]. 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-

* Corresponding author: shaolun.huang@sz.tsinghua.edu.cn

梯度符号的重要性，通过其他的文献来证明了梯度符号的重要性。

art statistic-based detection methods such as Krum and Bulyan [6], [21]. SignGuard is inspired by our theoretical analysis of *Little is Enough* (LIE) attack [19], and the generally good performance of signSGD [22] in distributed learning tasks. In [22] 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 a 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 an unsupervised clustering method to remove the anomalous ones. We find this simple and practical strategy can detect suspicious gradients effectively and efficiently.

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 gradient aggregation. 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. In particular, for a system with n clients including m Byzantine clients and 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, the SignGuard framework is evaluated on various real-world image and text classification tasks through extensive experiments by changing the attacks and the percentages of malicious clients. Evaluation results demonstrate the effectiveness of our SignGuard in protecting the FL system from Byzantine poisoning attacks and meanwhile achieving high model accuracy. To summarize, we make the following key contributions:

- A novel gradient aggregation framework called SignGuard is proposed for Byzantine-robust federated learning, which leverages the sign statistics of gradients to defend against model poisoning attacks.
- We provide a 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 a appropriate choice of learning rate. In particular, we show that Byzantine clients inevitably affect the convergence error in non-IID settings even if all malicious gradients are removed.
- SignGuard is verified through extensive experiments on MNIST/Fashion-MNIST, CIFAR-10, and AG-News datasets under various Byzantine attacks. Compared with existing approaches, our method exhibits superior performance in both IID and non-IID settings.

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

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 and adversarial threats [1], [4], [7], [23], especially under tough privacy regulations such as General Data Protection Regulation (GDPR) [24]. Meanwhile, the learning systems are vulnerable to various kinds of failures, including non-malicious faults and malicious attacks. Data poisoning and model update poisoning attacks aim to degrade or even fully break the global model during the training phase, while backdoor attacks (aka. targeted attacks) make the model misclassify certain samples during the 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 [6], [20], [25].

B. Existing Defense Strategies

Existing defenses either leverage statistic-based robust aggregation rule to get reliable gradient estimation, or utilize auxiliary data in PS to validate the received gradients. The former is also known as majority-vote based strategy and requiring the percentage of Byzantine clients less than 50%, such as Krum [6], trimmed-mean (TrMean) and coordinate-wise median (Median) [16] and Bulyan [21]. Specially, some works only aggregate the sign of gradient to mitigate the Byzantine effect [22], [26]. Recently, a method called Divider and Conquer (DnC) is proposed to tackle strong attacks [20]. When auxiliary data is available in PS, robustness can be guaranteed by validating the performance of received gradients/models. Zeno [17] use a stochastic descendant score to evaluate the correctness of each gradient and choose those with the highest scores. Fang [10] use error rate based and loss function based rejection mechanism to reject gradients that have a bad impact on model updating. In [27], the authors utilize the ReLU-clipped cosine-similarity between each received gradient and standard gradient as the weight to get robust aggregation. The main concern of such approaches is the accessibility of auxiliary data and the extra computational overhead.

Some studies show that malicious behavior could be revealed from the gradient trace by designing advanced filter techniques [28]–[30]. Besides, the client-side momentum SGD can also be considered as a history-aided method and can help to alleviate the impact of Byzantine attacks [31], [32]. Another line of work utilizes data redundancy to eliminate the effect of Byzantine failures. In [33], the authors present a scalable framework called DRACO for robust distributed training using ideas from coding theory. In [34], a framework called DETOX is proposed by combing computational redundancy and hierarchical robust aggregation to filter out Byzantine gradients. In [35], signSGD with election coding is proposed for robust and communication-efficient distributed learning. Moreover, provable security guarantee is also explored in [36], [37].

III. Rethinking of LIE Attack

In this section, we present our theoretical analysis along with empirical evidence of the *Little is Enough* (LIE) attack [19] to demonstrate the limitation of existing median- and distance-based defenses.

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, and 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. 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 a crucial role in model updating. Therefore, it's important to check the sign of gradient for this type of 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:

$$if \quad z > \frac{\mu_j}{\sigma_j}, \quad then \quad sign(\tilde{g}_j) \neq sign(\mu_j) \tag{3}$$

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:

$$\tilde{g}_j = \frac{1}{n}[m \cdot (g_m)_j + (n-m)\mu_j] = \mu_j - z \cdot \beta \cdot \sigma_j \tag{4}$$

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

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

Empirical results in [19] show that the coordinate-wise standard deviation turns out to be bigger than the corresponding mean, thus a small value of z is enough to turn a large amount of positive elements into negative, leading to incorrect model updating. To verify this insight, we adopt the default training setting in Section V to train a CNN on MNIST dataset and a ResNet-18 on CIFAR-10 dataset under no attacks. We calculate the averaged sign statistics across all clients as well as the sign statistics of a virtual gradient crafted by Eq. (1) and plot them over iterations as in Fig. 2, which convincingly supports our intuition.

Next, we present the following Proposition 1 to explain why *LIE* attack is hard to detect, where 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 the malicious gradient and honest gradient, respectively.

Proposition 1. *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:*

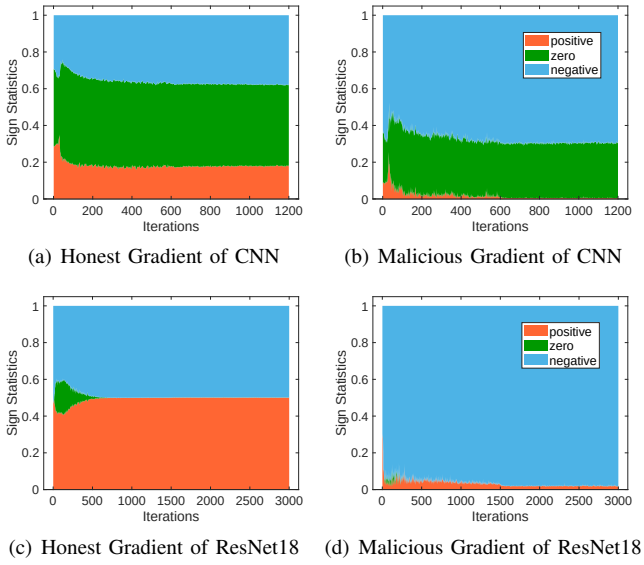
$$\exists \ i, \ s.t. \ \ \mathbb{E}[\|g_m - \tilde{g}\|^2] < \mathbb{E}[\|g^{(i)} - \tilde{g}\|^2] \tag{6}$$

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

$$\exists \ i, \ s.t. \ \ cos(g_m, \tilde{g}) > cos(g^{(i)}, \tilde{g}) \tag{7}$$

Proof. Detailed proof is in Appendix A.

余弦值越大说明越接近；故不好区分



(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.

From the above analysis results, it can be concluded that the malicious gradient can be even "safer" when evaluated by Krum and Bulyan methods. Hence, it's difficult 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. Our analysis is also valid for the recent proposed Min-Max/Min-Sum attacks in [20].

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

$$g_{m_1} = *, \ g_{m_2} = \frac{(n-m_1)g_{m_1} - \sum_{i=m+1}^{n} g^{(i)}}{m_2} \tag{8}$$

All existing attacks can be integrated into this ByzMean attack, making this hybrid attack even stronger than all single attacks.

混合型攻击：对m个恶意梯度进行m1和m2个恶意梯度进行混合攻击。

For example, we can set g_{m_1} as random noise or the gradient crafted by LIE attack.

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 the 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 (also called corrupted client) may act maliciously and can send arbitrary message to the server.

Threat Model. Similar to the threat models in previous works [10], [19], [20], we assume that there exists an attacker that controls some malicious clients to perform model poisoning attacks. Specially, we assume the attacker has full knowledge of all benign gradients and model parameters while the corrupted clients can also collude to conduct strong attacks. However, the attacker cannot corrupt the server and the proportion of malicious clients β is less than half.

Defender's Capability: As in previous studies [10], [27], 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 some FL scenarios.

Defense Goal: As mentioned in [27], an ideal defense method should consider 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:

$$\min_{\mathbf{x}\in R^d} F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\xi_i\sim D_i}\left[F(\mathbf{x};\xi_i)\right] \tag{9}$$

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 [3] 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:

$$g_t^{(i)} = \begin{cases} \nabla F(\mathbf{x}_t;\xi_i), & \text{if } i\text{-th client is benign} \\ arbitrary, & \text{if } i\text{-th client is Byzantine} \end{cases} \tag{10}$$

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

$$\mathbf{x_{t+1}} = \mathbf{x_t} - \eta_t \cdot GAR(\{g_t^{(i)}\}_{i=1}^n) \tag{11}$$

In synchronous settings with full client participation, 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 the next round. This process will repeat until the stop condition is satisfied.

To characterize the impact of Byzantine attacks, we define the following metric to measure the effect of Byzantine attack by calculating the accuracy drop due to model poisoning:

Definition 3. (Attack Impact) The impact of a specific attack is measured by the model accuracy drop compared to the baseline without the presence of any attack or defense.

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. Firstly, for the norm-based filter, the median of gradient norms is utilized as reference norm as the median always lies in the benign set. Considering that small magnitudes of gradients do less harm to the training while a significantly large one is 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 use Mean-Shift [38] algorithm as an unsupervised clustering model with an adaptive number of cluster classes, while the cluster with the largest size is selected as the trusted set (if all malicious clients send the same attack vector, K-Means with two clusters will suffice). In this work, the proportions of positive, zero,

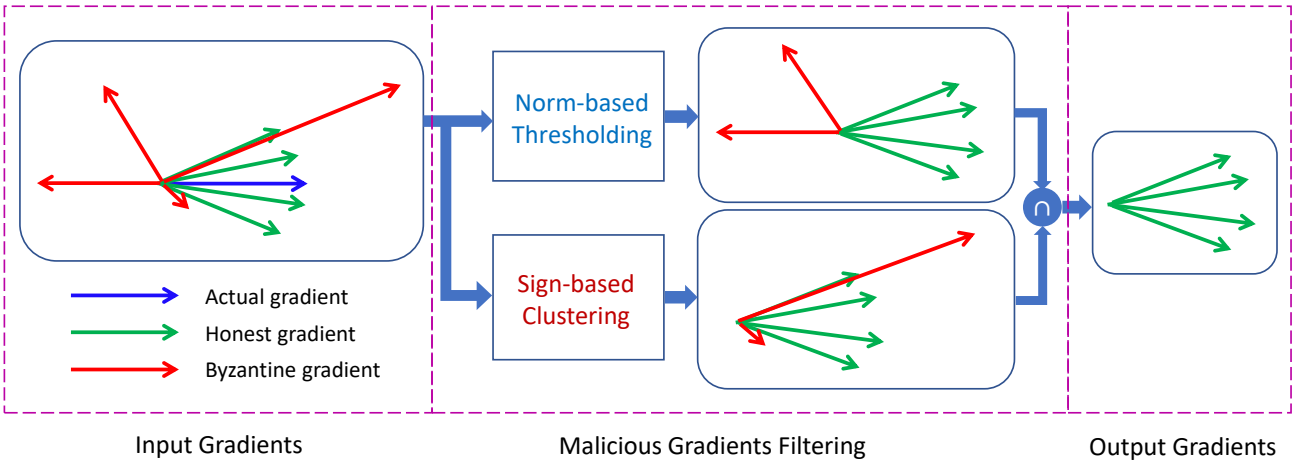


Fig. 3. Illustration of the workflow of proposed SignGuard. The collected gradients are anonymous and sent into multiple filters in parallel, after which the intersection of multiple outputs are selected as trusted gradients. We use norm-based and sign-based filters in this paper.

Algorithm 1 SignGuard-based Robust Federated Learning

Input: learning rate η, total iterations T, number of clients 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-based Filtering
  Get the median of norm $M = med(\{\|g_t^{(i)}\|\}_{i=1}^n)$
  Add the gradient that satisfies $L \leq \frac{\|g_t^{(i)}\|}{M} \leq R$ 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 $\tilde{g}_t = \frac{1}{|S_t'|} \sum_{i \in S_t'} g_t^{(i)} \cdot \min\left(1,\ M/\|g_t^{(i)}\|\right)$
Output: Global gradient: $\tilde{g}_t$

and negative signs are computed as basic features, which are sufficient for a variety of attacks, including the LIE attack.

However, those features only consider the overall statistics and lose sight of local properties. For example, when the amounts of positive and negative elements are approximate (e.g., ResNet-18), the naive sign statistics may be insufficient to detect the reversed gradients [34] or those well-crafted attacks that have similar sign statistics. To overcome this problem, we introduce randomized coordinate selection and add another similarity metric as an 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 as the similarity with a "correct" gradient. Or more efficiently, just utilize the aggregated gradient from the previous iteration as the "correct" gradient. Intuitively, it is promising to distinguish those irrelevant gradients and helps to improve the robustness of anomaly detection. However, 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 the cosine-similarity feature or Euclidean distance feature are called "SignGuard-Sim" and "SignGuard-Dist", respectively. We will provide some comparative results of them. How to design a more reliable similarity metric is left as an open problem for future work.

After the filtering process, the server eventually selects the intersection of multiple filter outputs as the trusted set and obtains a global gradient by robust aggregation, e.g. trimmed-mean. In this work, we use the mean aggregation with norm

新的相似度度量方法？？

改进的点：non-iid良性客户端被赦免。

clipping, where the clipping bound is selected as the median value of gradient norms. It is worth noting that a small fraction of honest gradients could also be filtered out, especially in the non-IID settings, depending on the variance of honest gradients and the distance to malicious gradients.

C. Convergence Analysis

To conduct convergence analysis, we also make the following basic assumption, which is commonly used in the literature [39]–[41] for convergence analysis of distributed optimization.

Assumption 1. *Assume that problem (9) 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:*

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

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

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

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

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

and the deviation between local and global gradient satisfies:

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

For the 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 [31], [42]. Here we make another assumption on the capability of the aggregation:

Assumption 2. *For problem (9) 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 there exist positive constants c and b such that the output $\hat{g}_t$ of SignGuard satisfies:*

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

2. Bounded Variance: $var \|\hat{g}_t\| \leq b^2$

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

Remark 1. *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. The bounded bias assumption is reasonable since we perform norm clipping before aggregation.*

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 1. *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:*

$$\mathbb{E}\left[\|\bar{g} - \nabla F(\mathbf{x})\|^2\right] \leq \frac{\beta^2 \kappa^2}{(1-\beta)^2} + \frac{\sigma^2}{(1-\beta)n} \tag{12}$$

Proof. Detailed proof is in Appendix B. □

Given the above assumptions and lemma, extending the analysis techniques in [31], [39]–[41], now we can characterize the convergence of SignGuard by the following theorem.

Theorem 1. *For problem (9) under Assumption 1, SignGuard satisfying Assumption 2 is employed with learning rate $\eta \leq (2 - \sqrt{\delta} - 2\beta)/(4L)$ and $F^* = $ then we have the following result:*

收敛性证明就是使得最后的结果小于一个值。

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

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 C. □

Remark 2. *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 a 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 the IID fashion, and investigate the performance of different defenses in the non-IID settings as well. The models that trained under no attack and no defense are used as benchmarks.

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) with 3 convolutional layers and 2 fully-connected layers as the global model.

Fashion-MNIST. Fashion-MNIST [43] is a clothing image classification dataset, which has the same image size and structure of training and testing splits as MNIST, and we use the same CNN model as in MNIST.

CIFAR-10. CIFAR-10 [44] 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 [45] as the global models[2].

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 the TextRNN with a two-layer bi-directional LSTM network [46] as the global model.

B. Evaluated Attacks

We consider various popular model poisoning attacks:

Random Attack. The Byzantine clients send gradients with randomized values that generated by a multi-dimensional Gaussian distribution $N(\mu, \sigma^2\mathbf{I})$. 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: $g_m = g_b + N(\mu, \sigma^2\mathbf{I})$. 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 reverse gradient attack [25], [34].

Label-Flipping. The Byzantine clients flip the local sample labels during the training process to generate faulty gradients. This is also a type of data poisoning attack. In particular, the label of each training sample in Byzantine clients is flipped from 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 [19], 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 proposed in Section III, we set $m_1 = \lfloor 0.5m \rfloor$ and $m_2 = m - m_1$, and set g_{m_1} as LIE attack.

Min-Max/Min-Sum. As in [20], the malicious gradient is a perturbed version of the benign aggregate as Eq. (13), where ∇^p is a perturbation vector and γ is a scaling coefficient, and those two attacks are formulated in Eq. (14)-(15). 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.

$$g_m = f_{avg}(g^{\{i\in[n]\}}) + \gamma\nabla^p \tag{13}$$

$$\arg\max_{\gamma} \max_{i\in[n]} \|g_m - g^{(i)}\| \leq \max_{i,j\in[n]} \|g^{(i)} - g^{(j)}\| \tag{14}$$

$$\arg\max_{\gamma} \sum_{i\in[n]} \|g_m - g^{(i)}\|^2 \leq \max_{i\in[n]} \sum_{j\in[n]} \|g^{(i)} - g^{(j)}\|^2 \tag{15}$$

C. Training Settings

By default, we consider a FL setup with $n = 50$ clients, 20% of which are Byzantine nodes, 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 realistic 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 algorithm is run for 60 epochs for MNIST/Fashion-MNIST/AG-News and 160 epochs for CIFAR-10. The number of local iteration is set to 1 and momentum is employed with the parameter of 0.9, and the weight decay is set to 0.0005.

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

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 the training process under different attack and defense methods in the IID settings are collected in Table I. The results of naive *Mean* aggregation under *No Attack* are used as benchmarks. Notice 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 rely on the Byzantine fraction information in our SignGuard framework, which is an important advantage over existing methods.

Performance comparison. Test results on four datasets consistently demonstrate that our SignGuard-type methods can leverage the power of sign statistics and similarity features to filter out most malicious gradients and achieve competitive test accuracy as general SGD under no attack. Consistent with original papers [19], [20], 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 an example, it can be seen that when no attack is performed, Multi-Krum has a 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

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**	**92.81**	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						

对于图表的消息，给出描述也要对结果出现的原因进行解释。

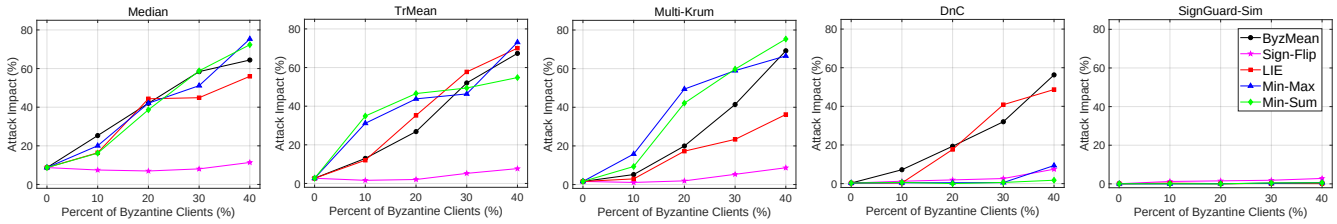
our plain SignGuard in mitigating random noise and sign-flipping attack. Though the DnC method has extraordinary effectiveness under many attacks, we found it is unstable during training and can be easily broken by our proposed ByzMean attack. In contrast, our proposed SignGuard-type methods are able to distinguish most of those well-crafted malicious gradients and achieve satisfactory model accuracy under various types of attacks. It is worth noting that our plain SignGuard already attains high robustness and fidelity, and the cosine-similarity/distance can further improve the defense performance in some cases, e.g., mitigating the sign-flipping attack. Besides, considering that the local data of Byzantine clients also contribute to the 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 a small gap to the benchmark results.

We also report the average selected rate of both benign and Byzantine gradients during the training process of ResNet-18 in Table II. We notice that the SignGuard-type methods inevitably exclude part of honest gradients, and select some malicious gradients under the sign-flipping attack. The reason lies in the fact that the proportions of positive and negative elements in normal gradient are approximate for ResNet-18. We also notice that although SignGuard-Sim is the most critical one and only selects less than 80% honest gradients during training, it is resilient to various kinds of attacks and still achieves high accuracy results.

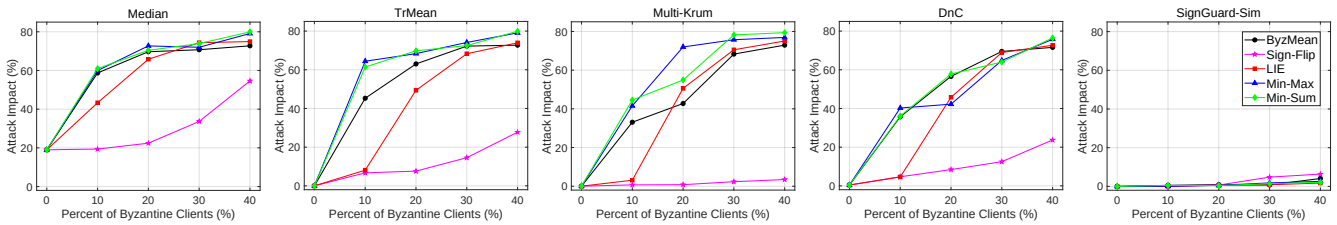
TABLE II
SELECTED RATE OF HONEST AND MALICIOUS GRADIENTS

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

Percentage of Byzantine clients. We also evaluate the performance of signGuard-Sim with different percentages of



(a) CNN trained on Fashion-MNIST



(b) ResNet-18 trained on CIFAR-10

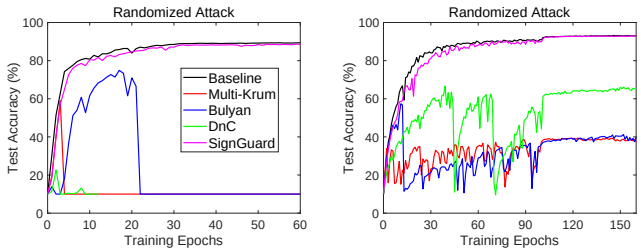
Fig. 4. Accuracy drop comparison under various attacks and different percentage of Byzantine clients. SignGuard-Sim has the smallest gap to the baseline.

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 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 a slight accuracy drop regardless of the high percentage of Byzantine clients, while other defense algorithms suffer much more attack impact with the increasing percentage of Byzantine clients. In particular, we also find that Multi-Krum can mitigate sign-flipping attack well in ResNet-18 training, possibly because the exact percentage of Byzantine clients is provided to the Multi-Krum algorithm.

Time-varying attack strategy. Further, we test different defense algorithms under the time-varying Byzantine attack strategy. We still use the default system setting and change the 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 for other defenses, which further demonstrated the superiority of SignGuard.

B. Main Results in Non-IID Settings

The Byzantine mitigation in non-IID FL settings has been a well-known challenge due to the diversity of gradients.

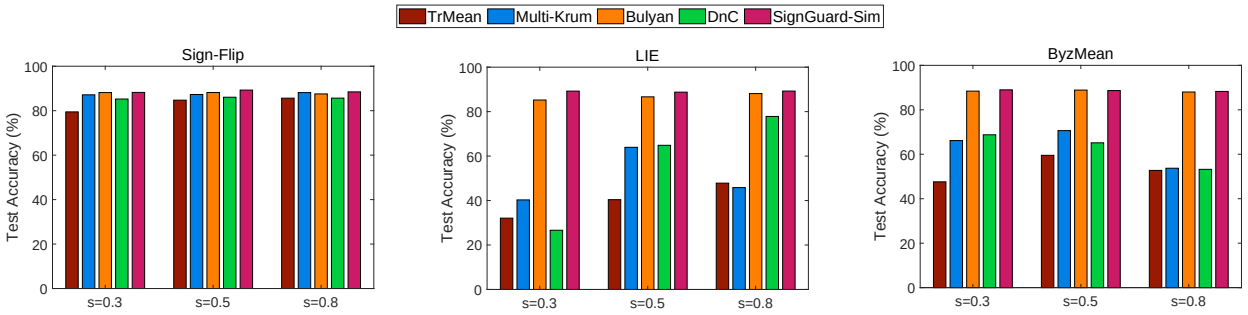


(a) CNN on Fashion-MNIST (b) ResNet-18 on CIFAR-10

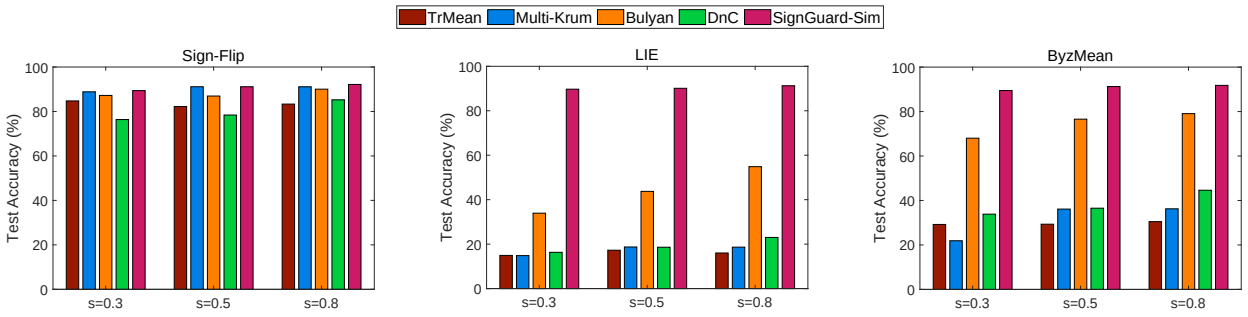
Fig. 5. Defense comparison under time-varying attacks. SignGuard can ensure safe training and achieve decent model accuracy.

We evaluate our SignGuard-Sim method in synthetic non-IID partitions of Fashion-MNIST and CIFAR-10 datasets. As in previous works, we simulate the non-IID data distribution between clients by allocating s-fraction of the dataset in a IID fashion and the remaining (1-s)-fraction in a sort-and-partition manner. 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 them 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 of skewness with s = 0.3, 0.5, 0.8, respectively.

Efficacy on non-IID data. We choose the SignGuard-Sim algorithm and compare it with various start-of-the-art defenses. As shown in Fig. 6, 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 against the LIE attack and ByzMean attack, making them not reliable anymore. Bulyan has a good performance on CNN trained on Fashion-MNIST, but is



(a) CNN on Fashion-MNIST



(b) ResNet-18 on CIFAR-10

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

ineffective under LIE attack on ResNet-18 trained on CIFAR-10. DnC can defend against sign-flipping 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
RESULTS UNDER DIFFERENT DEFENSIVE COMPONENTS

Thresholding	Clustering	Norm-Clip	Attacks		
			Random	Reverse	LIE
✓			47.41	44.48	56.74
	✓		88.43	25.29	88.18
		✓	55.27	54.29	45.98
✓	✓		**93.17**	92.43	**93.21**
	✓	✓	93.11	93.02	93.17
✓	✓	✓	92.76	**93.16**	92.40

C. Ablation Study

Although the above numerical results demonstrate the effectiveness and superiority of our proposed SignGuard framework, our method consists of multiple components and their individual efficacy need more investigation. In this part, we provide some ablation studies under the IID training setting on CIFAR-10 dataset to evaluate the utilities of different defensive components in SignGuard-Sim, including norm-based thresholding, clustering-based filtering, and norm-clipping. Specially, we test the "Reverse Attack with Scaling" [34], in which the Byzantine clients scale the sign-flipped gradient with a positive coefficient r, which is selected as the upper bound R of the norm-based thresholding or $r = 100$ when no thresholding/norm-clipping is applied. And we also test the random attack and LIE attack. From the results in Table III, we could see that every single component could not effectively mitigate all attacks, but clustering-based filtering combined with either thresholding or norm-clipping is capable of defending against a wide range of Byzantine attacks. At first glance, the thresholding and norm-clipping may seem to be redundant since they have similar utilities, however, we believe that the thresholding incurs almost negligible computation cost and could be used to quickly detect those malicious gradients with significantly larger norms.

VII. CONCLUSION AND FUTURE WORK

In this work, we propose a novel Byzantine attack detection framework, namely SignGuard, to mitigate malicious gradients in federated learning systems. It can overcome the drawbacks of the median- and distance-based approaches which are vulnerable to well-crafted attacks and unlike validation-based approaches that require extra data collection in PS. 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 against a variety 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. Future directions include developing strategies to defend dynamic and hybrid model poisoning attacks as well as white-box and adaptive attacks in more complex 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.

VIII. Acknowledgment

The research of Dr. Shao-Lun Huang is supported in part by the Shenzhen Science and Technology Program under Grant KQTD20170810150821146, National Key R&D Program of China under Grant 2021YFA0715202 and High-end Foreign Expert Talent Introduction Plan under Grant G2021032013L. The work of Dr. Linqi Song is supported in part by the Hong Kong RGC grant ECS 21212419, InnoHK initiative, the Government of the HKSAR, and Laboratory for AI-Powered Financial Technologies.

References

[1] 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.

[2] 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.

[3] 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)*, 2017.

[4] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, and et al, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1-2, pp. 1–210, 2021.

[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] 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.

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

[8] 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

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

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

[26] 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.

[27] 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.

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

[29] Z. Allen-Zhu, F. Ebrahimianghazani, J. Li, and D. Alistarh, "Byzantine-resilient non-convex stochastic gradient descent," in *International Conference on Learning Representations, ICLR*, 2021.

[30] 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

[31] S. P. Karimireddy, L. He, and M. Jaggi, "Learning from history for byzantine robust optimization," in *Proceedings of International Conference on Machine Learning, ICML*, 2021.

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

[33] 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.

[34] 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.

[35] J. Sohn, D. Han, B. Choi, and J. Moon, "Election coding for distributed learning: Protecting signsgd against byzantine attacks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[36] J. Steinhardt, P. W. Koh, and P. Liang, "Certified defenses for data poisoning attacks," in *Advances in Neural Information Processing Systems*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017.

[37] X. Cao, J. Jia, and N. Z. Gong, "Provably secure federated learning against malicious clients," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, 2021.

[38] 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

[39] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex op-

timization," in *International Conference on Machine Learning (ICML)*, vol. 97, 2019, pp. 7184–7193.
[40] 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.
[41] 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.
[42] 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.
[43] 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
[44] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *University of Toronto*, 2009.
[45] 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.
[46] 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.

APPENDIX

A. 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 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 a small enough z, it's possible for the malicious gradient to have a smaller distance from the true averaged gradient than that of an honest gradient.

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

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

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

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 gradients, 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:

$$\begin{aligned}
&\|g_m\|^2 > \xi_i^2\|\tilde{g}\|^2 \\
\iff &\sum_{j=1}^{d}(\mu_j - z\sigma_j)^2 > \xi_i^2\sum_{j=1}^{d}(\mu_j)^2 \\
\iff &\sum_{j=1}^{d}(\mu_j^2 - 2z\mu_j\sigma_j + z^2\sigma_j^2) > \xi_i^2\sum_{j=1}^{d}(\mu_j)^2 \\
\iff &z^2\sum_{j=1}^{d}(\sigma_j^2) - z\sum_{j=1}^{d}(2\mu_j\sigma_j) - (\xi_i^2-1)\sum_{j=1}^{d}(\mu_j)^2 > 0
\end{aligned}$$

which obviously holds when given appropriate value of z, as the left-hand side is a quadratic function of z.

Therefore, with appropriate selection of z, there exists some i such that $1 \leq \xi_i < \xi_m$. By using these relations of gradient norms, we can get:

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

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

B. Proof of Lemma 1

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

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

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

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

Using the basic relation between expectation and variance, we have

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

which leads to

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

Then, we directly have

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

It completes the proof of Lemma 1.

C. Proof of Theorem 1

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

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

From Assumption 1 & 2, we have

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

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

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

Combining with Lemma 2, we get

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

and

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

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

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

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

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

which completes the proof.