

Achieving Byzantine-Resilient Federated Learning via Layer-Adaptive Sparsified Model Aggregation

Jiahao Xu

University of Nevada, Reno

jiahaox@unr.edu

Zikai Zhang

University of Nevada, Reno

zikaiz@unr.edu

Rui Hu

University of Nevada, Reno

ruihu@unr.edu

Abstract

Federated Learning (FL) enables multiple clients to collaboratively train a model without sharing their local data. Yet the FL system is vulnerable to well-designed Byzantine attacks, which aim to disrupt the model training process by uploading malicious model updates. Existing robust aggregation rule-based defense methods overlook the diversity of magnitude and direction across different layers of the model updates, resulting in limited robustness performance, particularly in non-IID settings. To address these challenges, we propose the Layer-Adaptive Sparsified Model Aggregation (LASA) approach, which combines pre-aggregation sparsification with layer-wise adaptive aggregation to improve robustness. Specifically, LASA includes a pre-aggregation sparsification module that sparsifies updates from each client before aggregation, reducing the impact of malicious parameters and minimizing the interference from less important parameters for the subsequent filtering process. Based on sparsified updates, a layer-wise adaptive filter then adaptively selects benign layers using both magnitude and direction metrics across all clients for aggregation. We provide the detailed theoretical robustness analysis of LASA and the resilience analysis for the FL integrated with LASA. Extensive experiments are conducted on various IID and non-IID datasets. The numerical results demonstrate the effectiveness of LASA. Code is available at <https://github.com/JiahaoXU/LASA>.

1. Introduction

Federated Learning (FL) [33] is an emerging distributed machine learning paradigm that enables multiple clients, such as mobile devices or organizations, to collaboratively train a shared model while keeping their private data locally. This approach significantly reduces the necessity for data centralization, thereby not only decreasing data communication costs but also mitigating data privacy concerns. FL framework has been applied in diverse fields such as health-

care [39] and remote sensing [29], facilitating the use of machine learning in scenarios where data privacy and communication efficiency are critical.

However, distributing the model training across individual clients makes FL vulnerable to poisoning attacks [16, 26, 48], where an attacker controls a subset of clients and manipulates their local model updates to compromise the integrity of the global model. The *Byzantine attack* [5, 12, 16, 23, 45, 58] is one of the potent attacks which generally degrades the model's overall performance during the training. Specifically, under Byzantine attacks, a small set of malicious clients sends corrupted local model updates to the server during the training. It is shown that if the aggregation rule used by the server is a simple linear combination of local model updates, even one single malicious client can easily destroy the convergence of the global model [7]. Therefore, many efforts have been dedicated to designing aggregation rules that are robust against Byzantine attacks.

Existing robust aggregation rules can be mainly categorized into two types based on their granularity in handling model parameters: coordinate-wise robust aggregation [7, 54, 59] and model-wise robust aggregation [7, 16, 42, 45, 58]. Coordinate-wise robust aggregation focuses on evaluating and aggregating each coordinate of model updates independently, effectively filtering out extreme values that could represent malicious activity with fine granularity. In contrast, model-wise robust aggregation holistically assesses the entire model update from each client to detect outliers. The primary challenge in designing an effective robust aggregation rule lies in the difficulty of distinguishing between benign and malicious model updates, especially when the attacker's manipulations are subtle enough to blend seamlessly with benign data. This challenge becomes more pronounced in FL settings with non-IID data. To capture the subtle difference between benign and malicious model updates in such cases, it is essential to strike a balance between fine-grained and holistic assessment of model updates.

Recently, model sparsification has been used as an approach to enhance the Byzantine robustness of FL [34, 40, 61]. The key idea is to remove less important parame-

ters in the model updates to alleviate the malicious impact while maintaining the model’s utility. For instance, SparseFed [40] removes the parameters with less importance on the aggregated model update to defend against Byzantine attacks. However, current methods typically utilize a uniform sparsification mask for all model updates, leading to high sparsification error and limited robustness improvement, especially in non-IID settings where model updates diverge and necessitate personalized sparsification to preserve model utility.

Inspired by the sparsification-based method and robust aggregation with different granularity, we propose a novel Byzantine robust aggregation rule called **LASA** (**L**ayer-**A**daptive **S**parsified **M**odel **A**ggregation) that achieves Byzantine robustness of FL at the granularity of layer-level and important parameters only. Basically, LASA at first sparsifies each local model update individually without degrading the model utility. These sparsified updates are then fed into a layer-wise filter to adaptively detect and drop potential malicious layers. Finally, the remaining layers will be averaged as the global model update. The model sparsification is applied before aggregation to reduce the attack surface of malicious clients and maintain the model utility of benign clients with personalized sparsification. It also enables the subsequent filtering to focus on key parameters that determine the model performance. Notably, this strategy is particularly beneficial for non-IID settings, where local model updates of benign clients are diverse. The layer-wise filter extracts both the magnitude and direction of a sparsified layer as metrics and also enables layer-adaptive filtering with minimal control parameters, allowing LASA to strike a balance between coordinate-wise filtering and model-wise filtering efficiently and achieve better robustness. The model sparsification is carefully co-designed with the layer-wise adaptive filtering to ensure its amplification effect on robustness. The main contributions of this work are summarized as follows:

- We propose a novel robust aggregation rule called LASA. *To the best of our knowledge, our work is the first to combine pre-aggregation model sparsification with layer-wise adaptive aggregation to defend against Byzantine attacks in FL.* LASA can be easily integrated into the existing FL frameworks.
- We introduce a robustness criterion named κ -robustness, which quantifies the ability of an aggregation rule to accurately estimate the average of honest clients’ inputs when f out of n clients are malicious. We prove that LASA is a κ -robust aggregation rule with $\kappa = O(c_k(1+f/(n-2f)))$, where $c_k \leq 1$ correlates with the sparsification level, demonstrating the effectiveness of sparsification in amplifying robustness. Based on the robustness analysis, we also provide the resilience analysis of the local SGD-

based FL algorithm with LASA, for general non-convex loss functions and in the context of non-IID data. *To the best of our knowledge, our work is the first to theoretically analyze layer-wise defense methods.*

- We empirically evaluate the performances of LASA by conducting comprehensive experiments on both IID and non-IID datasets under various SOTA attacks. Compared to the SOTA defense methods, LASA achieves better robustness as well as performance.

2. Related Works

Poisoning attacks to FL. Federated Averaging (FedAvg) [33] stands as the classic FL method in non-adversarial environments. Yet, it has a critical vulnerability: the global model within FedAvg is susceptible to arbitrary manipulation by even a single malicious client [7, 59]. In particular, such a client can mislead the convergence of the global model by poisoning its local update sent to the server, which is known as poisoning attack in the context of FL [4–7, 12, 14, 16, 17, 20, 31, 36, 45, 47, 49, 51, 53, 55, 58].

Poisoning attacks can be categorized into *untargeted attacks* and *target attacks*. Targeted attacks (aka *backdoor attacks*) aim to mislead the global model to incorrectly predict certain outcomes chosen by the attacker for specific inputs while keeping the model’s performance on other inputs unaffected [4, 6, 49, 51, 53]. Untargeted attacks (aka *Byzantine attacks*) aim to generally disrupt the overall performance of the global model without any specific focus [5, 12, 16, 31, 45, 55, 58]. In this work, we focus on the Byzantine attacks on FL. The technical details of the SOTA Byzantine attacks [5, 45, 58] are given in Appendix 7.3.

Existing defense methods. Existing defense methods against Byzantine attacks in FL can be generally categorized into three types: 1) auxiliary data-based methods [9, 41, 56] which leverage the proxy dataset to conduct additional evaluation and thus filter out updates with abnormal performance. However, these methods somehow contradict the privacy-preserving goal of FL as they require a server dataset that is similar to local data to help identify malicious updates. Note that our approach does not need an auxiliary dataset and is orthogonal to these methods. 2) sparsification-based methods [34, 40, 61] which aim to remove malicious model parameters to enhance the robustness. For example, *SparseFed* [40] sparsifies the aggregated model update at the server side, integrating with model clipping and error feedback, to mitigate the impact of malicious local model updates. Model sparsification can enhance robustness by reducing malicious parameters, but since the server can’t identify malicious clients, it sparsifies all model updates, which degrades benign models’ performance. Moreover, existing works [40, 61] use uniform sparsification masks which increase sparsification errors, espe-

cially in non-IID settings. Our method applies individual sparsification to each update and combines it with magnitude and direction-based filtering to boost robustness. In addition, we theoretically analyze how this model sparsification contributes to robust aggregation, bridging the gap in the current SOTAs. 3) Robust aggregation-based methods [7, 12, 16, 37, 42, 45, 50, 54, 57–59] focus on developing a new aggregation rule on the server side that is robust against Byzantine attacks to replace the standard *averaging* aggregation rule used in FedAvg. For example, *Trimmed mean* (TrMean) proposed in [59] discards a certain percentage of the highest and lowest values among the received models for each dimension. After this trimming, the mean of the remaining values is computed by the server, which mitigates the impact of extreme values on the aggregated model. *Multi-Krum* [7] selects the most reliable local model that has the smallest sum of squared Euclidean distances to all other models as the output. *LEGATO* [50] weights each layer before aggregation but cannot eliminate malicious parameters. Recently, a defense method called *SignGuard* that achieves SOTA results has been proposed in [58]. It combines direction-based clustering and magnitude-based filtering to identify malicious model updates.

However, coordinate-wise methods [7, 54, 59] ignore model direction, and model-wise methods [7, 16, 42, 45, 58] overlook the diverse distribution of direction and magnitude across layers, limiting their robustness. Our layer-level approach is finer-grained than model-wise methods and more comprehensive than coordinate-wise methods. Furthermore, most works assume IID data, using clustering or distance-based methods to filter outliers [7, 42, 57, 58]. However, in real-world FL scenarios, data is often non-IID, making these methods less effective. Our approach combines pre-aggregation model sparsification with layer-wise direction- and magnitude-based filtering to handle diverse model updates with only key parameters. We use a novel sign-based metric to assess model update directions, improving the purity and effectiveness of direction-based filtering. Unlike works like *SignGuard* [58], we provide a detailed theoretical robustness analysis of LASA and its resilience in FL. Our theoretical analysis is most related to works on distributed gradient descent (D-GD) [1–3], but we focus on FL with local SGD, which increases local model divergence and complicates the analysis.

3. Our Solution: LASA

The LASA process is given in Algorithm 1. LASA features an innovative design and integration of pre-aggregation model sparsification and layer-wise robust aggregation on the server side, aimed at mitigating the impact of malicious local model updates.

Pre-aggregation sparsification. Specifically, in each round of FL, after receiving the local model updates from

Algorithm 1 LASA

Require: Set of n local model updates $\{\Delta_i\}_{i=1}^n$, number of model layers L , sparsification parameter k , magnitude-based radius λ_m , and direction-based radius λ_d

- 1: **for** $i \in [n]$ **do**
- 2: $\hat{\Delta}_i \leftarrow Top_k(\Delta_i)$ \triangleleft model sparsification
- 3: **end for**
- 4: **for** each layer $l \in [L]$ **do**
- 5: Initialize benign set $\mathcal{S} = \emptyset$
- 6: $\omega^l \leftarrow \{L_2\text{-norm}(\hat{\Delta}_i^l)\}_{i=1}^n$
- 7: $\rho^l \leftarrow \{PDP(\hat{\Delta}_i^l)\}_{i=1}^n$ \triangleleft by Equation. 1
- 8: **for** $i \in [n]$ **do**
- 9: $\lambda_{i,m}^l \leftarrow MZ\text{-score}(\omega_i^l, \omega^l)$ \triangleleft by Equation. 2
- 10: $\lambda_{i,d}^l \leftarrow MZ\text{-score}(\rho_i^l, \rho^l)$ \triangleleft by Equation. 2
- 11: **if** $|\lambda_{i,m}^l| \leq \lambda_m$ and $|\lambda_{i,d}^l| \leq \lambda_d$ **then**
- 12: $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$
- 13: **end if**
- 14: **end for**
- 15: $\bar{\Delta}^l \leftarrow \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \hat{\Delta}_i^l$ \triangleleft layer-wise aggregation
- 16: **end for**
- 17: **return** $\bar{\Delta}$

clients, the server first sparsifies each local model update Δ_i individually using the Top- k sparsifier defined in Definition 1 (line 2).

Definition 1 (Top- k sparsifier [19]). *For a vector $x \in \mathbb{R}^d$ and a parameter $k \in [1, d]$, the Top- k sparsifier $Top_k(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is defined as: $[Top_k(x)]_j := [x]_{\pi(j)}$ if $j \leq k$ and $[Top_k(x)]_j := 0$ otherwise, where π is a permutation of indices such that $|[x]_{\pi(j)}| \geq |[x]_{\pi(j+1)}|, \forall j \in [1, d-1]$.*

With sparsification, each element of the sparsified model update $[\hat{\Delta}_i]_j$ equals to $[\Delta_i]_j$ if $j \in \mathcal{K}$ and 0 otherwise, where \mathcal{K} represents the set of coordinates of parameters that have the top k largest absolute values. Here, model sparsification can limit the attack surface available to malicious clients by dropping out $d - k$ parameters from the model update. Nonetheless, given the server's inability to differentiate between malicious and benign updates, the sparsification also introduces sparsification errors to the benign updates, diminishing the model's utility, especially when k is small. Fortunately, the Top- k sparsifier prunes a vector by keeping the largest elements, so that the sparsified vector still contains the core information of the original vector, resulting in a low sparsification error. Compared with other sparsification method like random sparsification, Top- k sparsifier is much more robust when k is small [19, 46]. Moreover, individual sparsification of each model update can better preserve the model utility, compared with the uniform sparsification in existing works [34, 61]. This is particularly beneficial in non-IID settings, where variability in local data distributions results in diverse local model updates.

In addition, applying sparsification before aggregation allows the subsequent layer-wise filtering to concentrate on the key parameters critical to model performance, thereby eliminating interference from less important parameters.

Layer-wise adaptive aggregation. It has been observed that different layers in the deep neural network differ in their sizes, functionality, and more importantly, learning and converging speed [25, 27, 30, 44]. However, most of the existing robust aggregation rules perform model-wise or coordinate-wise assessment on the local model updates, which usually fail to identify the nuances of each layer. Therefore, in LASA, we design a layer-wise filtering and aggregation after model sparsification, which enhances robustness with more precise, layer-specific granularity.

In the context of Byzantine attacks, where attackers aim to deviate the global model’s convergence in the wrong directions, malicious updates are typically crafted to significantly diverge from benign updates, both in magnitude and direction. Hence, the LASA, both magnitude and direction of each model update are captured at the layer level to effectively identify and filter out malicious clients. More precisely, for each layer $l \in [L]$ of the sparsified model update, its magnitude is quantified using the L_2 -norm, and its direction is determined by analyzing the signs of its parameters. Inspired by [10], a direction metric, termed as *Positive Direction Purity (PDP)* is defined in Definition 2.

Definition 2 (Positive Direction Purity). *For a vector $x \in \mathbb{R}^d$, the positive direction purity ρ of x is defined as*

$$\rho := \frac{1}{2} \times \left(1 + \frac{\sum_{i=1}^d sgn([x]_i)}{\sum_{i=1}^d |sgn([x]_i)|} \right), \quad 0 \leq \rho \leq 1, \quad (1)$$

where $sgn(\cdot)$ is the function to take the sign of each element and $[x]_i$ is the i -th coordinate of a vector.

PDP serves as a metric to evaluate the predominance of positive signs within a given vector, providing a refined approach for identifying anomalies in model direction. As a normalized measure, PDP measures a vector’s overall orientation toward positive values, which is useful for analyzing directional tendencies. It is particularly effective in detecting stealthy attacks where the malicious models might not exhibit significant variations in magnitude. It is worth noting that the pre-aggregation sparsification can significantly enhance the PDP-based measurement of model direction by removing many less important parameters. This removal is particularly significant for PDP, which relies solely on the signs of the parameters, ensuring that the measurement focuses on the parameters with large values.

With the magnitude and direction metrics of a layer (line 6-7), LASA will filter out clients that exhibit extreme values (either excessively high or low) using pre-defined thresholds. Given that the magnitude and direction values

for each layer can vary significantly, setting such thresholds necessitates layer-specific customization. This, however, leads to a proliferation of hyper-parameters. Inspired by the traditional standardization method *Z-score*, we introduce a robust variant named *Median-based Z-score (MZ-score)*, as defined in Definition 3.

Definition 3 (MZ-score). *For a set of values $X := \{x_1, \dots, x_n\}$ with median $Med(X)$ and standard deviation σ , the MZ-score λ_i of $x_i \in X$ is defined as*

$$\lambda_i := \frac{x_i - Med(X)}{\sigma}. \quad (2)$$

This variant indicates how many standard deviations an element is from the median, which can be positive or negative. Importantly, MZ-score allows a uniform filtering radius to be applied across all layers, which substantially reduces the number of hyper-parameters required for adaptive layer-wise filtering. Specifically, in LASA, MZ-scores of magnitude and direction metrics are calculated at the layer level for all clients (line 9-11). Model updates with high absolute MZ-score values are then filtered out using two pre-defined filtering radii: λ_m for magnitude and λ_d for direction. Subsequently, the clients that remain, considered benign, are added to the set \mathcal{S} and will participate in layer-wise model averaging (line 15).

4. Robustness and Resilience Analysis of LASA

Before presenting our theoretical results, we make the following assumptions:

Assumption 1 (μ -Smoothness). *Each local objective function \mathcal{L}_i for benign client $i \in \mathcal{B}$ is μ -Lipschitz smooth with $\mu > 0$, i.e., for any $x, y \in \mathbb{R}^d$, $\|\nabla \mathcal{L}_i(x) - \nabla \mathcal{L}_i(y)\| \leq \mu \|x - y\|, \forall i \in \mathcal{B}$, which further gives: $\mathcal{L}_i(x) - \mathcal{L}_i(y) \leq \nabla \mathcal{L}_i(x)^T (y - x) + \frac{\mu}{2} \|x - y\|^2, \forall i \in \mathcal{B}$.*

Assumption 2 (Unbiased gradient and bounded variance). *The stochastic gradient at each benign client is an unbiased estimator of the local gradient, i.e., $\mathbb{E}[g_i(x)] = \nabla \mathcal{L}_i(x)$ and has bounded variance, i.e., for any $x \in \mathbb{R}^d$, $\mathbb{E} \|g_i(x) - \nabla \mathcal{L}_i(x)\|^2 \leq \nu_i^2, \forall i \in \mathcal{B}$, where the expectation is over the local mini-batches. We also denote $\bar{\nu} := (1/|\mathcal{B}|) \sum_{i \in \mathcal{B}} \nu_i^2$ for convenience.*

Assumption 3 (Bounded heterogeneity). *There exist a real value $\bar{\zeta}$ such that for any $x \in \mathbb{R}^d$, $\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \|\nabla \mathcal{L}_i(x) - \nabla \mathcal{L}_{\mathcal{B}}(x)\|^2 \leq \bar{\zeta}$, where the $\nabla \mathcal{L}_{\mathcal{B}}(x) := (1/|\mathcal{B}|) \sum_{i \in \mathcal{B}} \mathcal{L}_i(x)$.*

Note that Assumption 1-2 are commonly used in the theoretical analysis of distributed learning systems [19, 38, 58]. While Assumption 3 states a standard measure of inter-client heterogeneity in FL [1, 11, 21], which complicates the

problem of Byzantine FL [1]. Note that these assumptions apply to benign clients only, as malicious clients do not follow the prescribed local training protocol of FL.

Assumption 4 (Bounded sparsification). *Given a vector $x \in \mathbb{R}^d$, there exists non-negative constants $c_k \in [0, 1]$ and $b_k \in [0, 1]$, so that the Top- k sparsifier in Definition 1 satisfies $\|\text{Top}_k(x)\|^2 \leq c_k \|x\|^2$, and $\|\text{Top}_k(x) - x\|^2 \leq b_k \|x\|^2$.*

As LASA incorporates model sparsification, we make the following Assumption 4 on the Top- k sparsifier in Definition 1. This assumption applies for any $k \in [0, d]$ due to the fact that $\|x\|^2 = \|\text{Top}_k(x) - x\|^2 + \|\text{Top}_k(x)\|^2$. A smaller k implies a higher degree of sparsification and yields a smaller c_k and a larger b_k .

4.1. Robustness analysis of LASA

To theoretically evaluate the efficacy of LASA, we introduce the concept of κ -robustness in Definition 4. Note that Definition 4 is similar to (f, κ) -robustness defined in [1, 2], (δ_{\max}, c) -ARAGg defined in [15, 21, 32], and (f, λ) -resilient averaging defined in [13]. Our robustness definition adopts a constant upper bound and focuses on quantifying the distance between the output of a robust aggregation rule and the average of all benign updates, which represents the optimal output of such a rule. We denote the set of benign clients as \mathcal{B} so that $\mathcal{B} \subseteq \mathcal{N}$, where \mathcal{N} is the client set.

Definition 4 (κ -robustness). *Let $n > 1$ and $0 \leq f < n/2$. An aggregation rule $F: \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^d$ is κ -robust if for any vectors $x_1, \dots, x_n \in \mathbb{R}^d$ and a benign set $\mathcal{B} \subseteq \mathcal{N}$ of size $n - f$, the output $\hat{x} := F(x_1, \dots, x_n)$ satisfies $\mathbb{E} \|\hat{x} - \bar{x}_{\mathcal{B}}\|^2 \leq \kappa$, where $\bar{x}_{\mathcal{B}} := \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} x_i$, $\kappa \geq 0$ refers to the robustness coefficient of the aggregation rule F , and the expectation is taken over the randomness of the inputs.*

The κ -robustness guarantees that the error of an aggregation rule, in estimating the average of the benign inputs, is upper-bounded by κ . With Definition 4, we prove that when LASA is applied to n input models, of which $f < n/2$ are malicious, satisfies κ -robustness with $\kappa = O(c_k(1+f/(n-2f)))$, as stated in Lemma 1. Note that LASA enjoys a higher robustness than several classical robust aggregation rules, for example, GeoMed [42] ($O(1+f/(n-2f))^2$), and Krum [7] ($O(1+f/(n-2f))$)¹.

Lemma 1 (κ -robustness of LASA). *Under Assumption 1-4, if $n \geq 1$ and $0 \leq f < n/2$, the proposed LASA method is a κ -robust aggregation rule with*

$$\kappa = 2c_k \left(1 + \frac{f}{n-2f}\right) (2\bar{\nu} + \bar{\zeta} + 2C_{\lambda_m}^2 + 2C^2) + b_k C^2,$$

¹Results of GeoMed and Krum are taken from [1]. Note that the definition of κ in [1] is different from ours, but since we are concerned with the *order* of κ , we can safely incorporate these results into our discussion without losing generality.

if the learning rate $\eta \leq 1/2\tau$ and the selection set satisfies $|\mathcal{S}^l| \geq n/2 - f, \forall l \in [L]$, τ is the number of local iteration. Here, $C_{\lambda_m}^2$ and C^2 represent the upper bound of the norm of malicious and benign updates, respectively.

If the sparsification parameter k satisfies that $c_k \leq 1/(1+\epsilon)$ and $(b_k/c_k) \leq \epsilon(4+(4\bar{\nu}+2\bar{\zeta}+4C_{\lambda_m}^2)/C^2)$ with a positive constant ϵ , we have $\kappa = O\left(c_k \left(1 + \frac{f}{n-2f}\right)\right)$.

Proof. The detailed proof is given in Appendix 7.9.2. \square

Remark. (1) **Extreme cases.** Extremely, without sparsification, i.e., $k = d$, we have $c_k = 1$ and $b_k = 0$. In this case, the robustness upper bound of LASA, denoted by κ_1 , is $\kappa_1 = 2 \left(1 + \frac{f}{n-2f}\right) (2\bar{\nu} + \bar{\zeta} + 2C_{\lambda_m}^2 + 2C^2)$. When $k = 0$, we have $c_k = 1$ and $b_k = 0$ which gives an upper bound of C^2 , indicating the greatest sparsification error to robustness. (2) **Proper k yields higher robustness.** When $0 < k < d$, if the sparsification parameter k is selected to satisfy the two conditions in Lemma 1, the robustness upper bound in this case, denoted by κ_2 , is $\kappa_2 = (1+\epsilon)c_k\kappa_1$. As $c_k \leq 1/(1+\epsilon)$, we have $\kappa_2 \leq \kappa_1$, which demonstrates the effectiveness of sparsification in amplifying robustness. Moreover, the conditions on k indicate that when the local divergence/variance or the magnitude of the malicious model is large, we can select a relatively small k (which gives a large b_k and small c_k) to amplify the robustness. Hence, the benefit of sparsification will be more significant in non-IID settings theoretically. (3) **Impact of $C_{\lambda_m}^2$ and C^2 .** Lemma 1 also shows that theoretically the larger the magnitude of benign updates or malicious updates is, the lower the robustness will be. Indeed, in the literature, model clipping has demonstrated its effectiveness in mitigating the impact of malicious [40, 58, 60]. In LASA, the norm of malicious updates is particularly bounded by the magnitude-based filtering which is controlled by the hyper-parameter λ_m in Algorithm 1. A smaller λ_m indicates a relatively smaller $C_{\lambda_m}^2$. (4) **Impact of $\bar{\nu}$ and $\bar{\zeta}$.** Note that $\bar{\nu}$ and $\bar{\zeta}$ represent the local variance and local heterogeneity in Assumption 2-3. The findings in Lemma 1 highlight the importance of reducing the variance of stochastic gradient and mitigating the divergence due to non-IID data distribution to enhance robustness. Our work is orthogonal to existing variance or divergence reduction methods [15] and can be combined with them to further improve the robustness.

4.2. Resilience analysis of FL with LASA

Similar to [1, 3], we define Byzantine resilience of a FL algorithm in Definition 5 as follows.

Definition 5 ((f, R)-Byzantine resilience [1, 3]). *With the presence of f Byzantine clients, a FL algorithm is said (f, R) -Byzantine resilient if it outputs $\tilde{\theta}$ such that $\mathbb{E} \left\| \nabla \mathcal{L}_{\mathcal{B}}(\tilde{\theta}) \right\|^2 \leq R$, where \mathcal{B} denotes the set of benign*

clients, $\mathcal{L}_B(\theta) := \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathcal{L}_i(\theta)$, and expectation is taken over the randomness of the FL algorithm.

In words, a (f, R) -Byzantine resilient FL finds an R -approximate stationary point for the honest loss, despite the presence of up to f Byzantine clients. This definition is crucial as it quantifies the level of tolerance a Byzantine-resilient FL has against the potentially harmful influence of Byzantine clients. Note that f is assumed to be less than $n/2$, as it is generally impossible for an FL algorithm with F to achieve Byzantine resilience when $f \geq n/2$ [28].

Now we prove that FL with LASA is (f, R) -Byzantine resilient and achieves the asymptotic error bounded by R in the presence of f Byzantine clients, as stated Theorem 1.

Theorem 1 $((f, R)$ -Byzantine resilience of LASA). *Let θ^0 be the initial point and θ^* be the optimal point. Assume $\tilde{\theta}$ is uniformly sampled from the sequence of outputs $\{\theta^0, \theta^1, \dots, \theta^T\}$ generated by FL with LASA. Under Assumption 1-4, suppose the learning rate η satisfies $\eta \leq \min\{1/2\tau, 1/3\mu\tau\}$, then we have*

$$\begin{aligned} \mathbb{E} \left\| \nabla \mathcal{L}_B(\tilde{\theta}) \right\|^2 &\leq \frac{\mathcal{L}_B(\theta^0) - \mathcal{L}_B(\theta^*)}{T\eta} + \kappa(\mu\eta + 1) \\ &\quad + 7\tau\bar{\zeta} + (1 + \tau)\bar{\nu}, \end{aligned}$$

where κ represents the robustness coefficient of LASA.

Proof. The detailed proof is given in Appendix 7.9.3. \square

Remark. The last two terms, i.e., $7\tau\bar{\zeta} + (1 + \tau)\bar{\nu}$, represent the convergence errors due to data heterogeneity and gradient variance and will be eliminated when local data are IID and full gradient is calculated. The second term represents the *Byzantine error* associated with the robustness coefficient κ . Note that due to the client sampling in FL, h out of n clients are selected uniformly at random to participate in the training per round, and the expected number of malicious clients is hf/n per round, which does not affect the expected value of κ . Recall that selecting an appropriate sparsification parameter k allows LASA to achieve a smaller κ , leading to a smaller convergence error. However, choosing an unsuitable k , such as when $k \ll d$, may result in a very high sparsification error, which will dominate κ and make the robustness amplification benefit of sparsification negligible. This finally leads to a higher convergence error and lower Byzantine resilience. We discuss the selection of k in Lemma 1 and also study its impact on model performance during the evaluation.

5. Evaluation

Experimental settings. To comprehensively demonstrate the effectiveness of LASA, we compare it with the non-robust baseline *FedAvg* and seven existing

SOTA defense methods, including *Bulyan* [16], *Trimmed Mean (TrMean)* [59], *Geometric median (GeoMed)* [42], *Multi-Krum* [7], *Divide-and-Conquer (DnC)* [45], *SignGuard* [58], and *SparseFed* [40]. We test three naive attacks including *Random*, *Noise* and *Sign-flip* attacks and five SOTA attack methods including *Min-Max* [45], *Min-Sum* [45], *AGR-tailored Trimmed-mean* [45], *Lie* [5], and *ByzMean* [58] attacks. We mainly conduct experiments on *FMNIST* [52], *FEMNIST* [8], *CIFAR-10* [22], *CIFAR-100* [22] and *Shakespeare* [33] datasets. *FEMNIST* and *Shakespeare* datasets are naturally non-IID. For *FMNIST*, *CIFAR-10* and *CIFAR-100* datasets, we evenly split the dataset over the clients to simulate the IID settings, and use *Dirichlet distribution* [35] $Dir(\alpha)$ to simulate the non-IID settings with a default non-IID degree $\alpha = 0.5$. The default *attack ratio* is set to 25%, meaning 25% of the clients are malicious in our FL system. Note that the number of malicious clients selected for training per round may differ due to client sampling. For the hyperparameters in LASA, for all datasets, we set the *sparsification level* to 0.3 (i.e., $1 - k/d = 0.3$), λ_d is set to 1.0 by default while λ_m is set to 1.0 for *CIFAR10/100* and 2.0 for others. More details of the experimental settings are given in Appendix 7.2. Additionally, the attack and defense models are presented in Appendix 7.1. We run each experiment with three random seeds and report the average testing accuracy. We use **bold font** to highlight the best results, while the second-best results are underlined.

Performance of LASA in IID settings. We first comprehensively evaluate the performance of all the defense methods in IID settings. From the results on *FMNIST*, *CIFAR-10*, and *CIFAR-100* datasets (shown in Table 1), we can see that except for *Noise* attack on *CIFAR-100* where LASA achieves second-best performance, LASA achieves the **highest accuracy under all attacks**, outperforming other defense methods. For example, under *ByzMean* attack, LASA, *GeoMed*, and *SignGuard* stand out as the most effective defense methods on *FMNIST*, and LASA achieves the highest accuracy of 87.65%, which is +0.03% and +4.60% higher than *SignGuard* and *GeoMed*, respectively. On *CIFAR-10* under *TailoredTrMean* attack, LASA achieves the highest accuracy of 89.05%, which is +0.60% and +4.02% higher than *SignGuard* and *Bulyan*, respectively. *It is noteworthy that LASA reaches “ceiling” performance levels as it consistently achieves accuracy similar to scenarios without attacks.* We observe that the classic robust aggregation rule-based methods including *TrMean*, *GeoMed*, *Multi-Krum*, and *Bulyan*, as well as the sparsification-based method *SparseFed*, fail to defend against advanced distance-based attacks like *TailoredTrMean* and *ByzMean*, which maximize or minimize the distance between benign and malicious models. The reason is that these classic robust aggregation rules often

Table 1. Testing Accuracy (%) of Different Defense Methods in IID Settings.

Dataset (Model)	Defense Method	No Attack	Naive Attacks			SOTA Attacks					Average w/ Attacks
			Random	Noise	Sign-flip	TailoredTrMean	Min-Max	Min-Sum	Lie	ByzMean	
FMNIST (CNN)	FedAvg	86.28	29.20	41.66	83.91	10.08	77.89	79.56	83.47	11.22	52.12
	TrMean	84.05	80.56	81.26	81.11	10.59	69.95	73.01	77.78	10.54	59.85
	GeoMed	84.10	84.30	84.30	82.28	84.51	60.86	50.32	65.08	83.05	74.34
	Multi-Krum	86.91	84.15	84.33	85.34	10.00	68.76	80.67	80.43	11.80	63.19
	Bulyan	81.35	83.81	83.84	78.59	33.76	59.24	62.09	73.29	59.75	66.80
	DnC	87.30	84.23	84.17	85.69	32.66	69.81	79.08	81.96	63.11	72.59
	SignGuard	87.63	87.72	87.72	87.06	87.40	87.40	87.18	87.17	87.61	87.41
	SparseFed	86.27	29.48	41.10	83.86	10.08	77.88	79.55	83.47	11.28	52.09
	LASA (Ours)	87.62	87.92	87.87	87.13	87.97	87.91	87.36	87.54	87.65	87.67
CIFAR-10 (ResNet18 [18])	FedAvg	89.70	44.34	47.65	82.07	15.77	76.26	61.25	84.76	13.01	53.14
	TrMean	90.14	87.36	87.36	84.77	49.65	61.30	57.58	77.40	49.61	69.38
	GeoMed	89.85	87.76	87.57	85.74	71.22	63.42	71.91	70.78	87.45	78.23
	Multi-Krum	84.73	84.62	84.72	84.58	84.49	47.97	53.16	44.26	84.55	71.04
	Bulyan	88.97	87.68	87.56	86.52	85.03	38.38	47.29	53.30	84.96	71.34
	DnC	89.54	59.26	61.33	84.72	38.75	63.34	61.11	67.30	57.08	61.61
	SignGuard	89.47	82.36	81.68	80.04	88.45	88.14	88.09	88.11	88.39	85.66
	SparseFed	89.65	43.93	48.22	82.18	15.92	75.90	68.13	84.91	10.00	53.65
	LASA (Ours)	89.00	88.66	88.81	86.68	89.05	88.57	88.96	88.59	89.08	88.55
CIFAR-100 (ResNet18 [18])	FedAvg	65.98	12.09	14.20	48.05	1.96	45.90	44.26	57.66	1.36	28.19
	TrMean	65.40	63.19	63.22	52.65	28.57	34.80	33.83	48.92	30.11	44.41
	GeoMed	65.84	63.33	63.57	61.25	39.78	38.95	39.34	46.93	61.29	51.81
	Multi-Krum	52.71	52.14	52.24	52.92	52.88	19.03	19.82	24.43	53.26	40.84
	Bulyan	61.29	61.06	61.32	60.01	50.84	17.83	19.17	30.27	56.75	44.66
	DnC	65.53	24.47	28.05	53.39	11.21	29.37	28.51	34.04	29.05	29.76
	SignGuard	65.64	63.24	63.36	47.55	63.36	63.20	63.22	62.72	62.94	61.20
	SparseFed	65.99	12.06	14.06	48.01	1.83	46.05	44.34	57.74	1.53	28.20
	LASA (Ours)	65.52	63.48	63.49	62.89	63.71	63.54	63.63	63.98	63.85	63.57

Table 2. Testing Accuracy (%) of LASA in Non-IID Settings on CIFAR-10 (C-10) and CIFAR-100 (C-100) Datasets, Compared with Multi-Krum, GeoMed and SignGuard under ByzMean.

Dataset	Method	Non-IID degrees α						Avg.
		0.1	0.2	0.3	0.4	0.5	1.0	
C-10	Multi-Krum	26.34	39.34	52.06	55.52	61.31	74.75	51.55
	GeoMed	49.14	63.33	71.45	72.82	75.72	83.49	69.33
	SignGuard	51.90	63.17	70.77	75.05	76.16	83.31	70.06
	LASA (Ours)	56.01	65.61	74.59	75.40	77.73	84.34	72.28
C-100	Multi-Krum	25.83	37.91	43.64	45.58	47.58	51.37	41.99
	GeoMed	46.46	53.80	56.99	58.16	58.72	59.79	55.65
	SignGuard	48.95	56.74	58.50	58.51	59.82	61.42	57.32
	LASA (Ours)	51.25	57.59	59.73	60.41	60.24	61.61	58.47

filter malicious parameters at coordinate level or based on model-wise distance; SparseFed is ineffective due to its limited capability in removing malicious parameters. The robustness of LASA, illustrated by the above-mentioned results, emphasizes its potential as a robust defense method in securing FL environments against a wide collection of attacks, ultimately enhancing the reliability of FL systems. Additional results on MNIST, FEMNIST, and Shakespeare datasets are given in Appendix 7.4.

Performance of LASA in various non-IID settings.

Here, we evaluate the effectiveness of LASA in various non-IID settings. We simulate different non-IIDness by varying the non-IID degree α from 0.1 to 1.0, where a smaller α indicates a more intense non-IIDness. From the results on CIFAR-10 and CIFAR-100 datasets under the SOTA ByzMean attack (shown in Table 2), we observe that

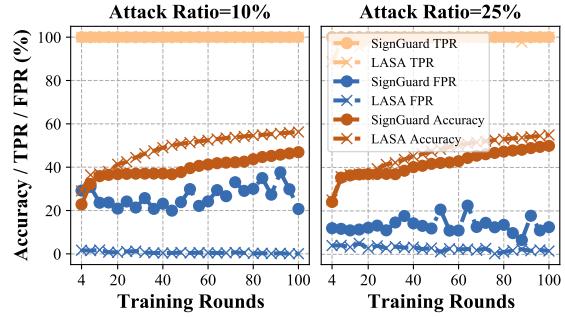


Figure 1. TPR, FPR, and Testing Accuracy (%) of LASA and SignGuard under ByzMean Attack on Shakespeare Dataset.

as α increases, the performance of all the defense methods improves due to the decreased data heterogeneity. Among them, LASA always achieves the highest accuracy under various non-IID degrees, leading a average of +1.15% and +2.22% over SOTA SignGuard. LASA individually sparsifies model updates thus reducing sparsification error, especially in non-IID cases with heterogeneous model updates. It also performs layer-wise filtering, allowing precise identification of benign/malicious model updates at a finer granularity. By adeptly integrating pre-aggregation sparsification and layer-wise adaptive aggregation, LASA effectively mitigates the impact of divergent updates, resulting in the highest accuracy among its counterparts.

Effectiveness of LASA in model update identification.

To deeply investigate the effectiveness of LASA, we ob-

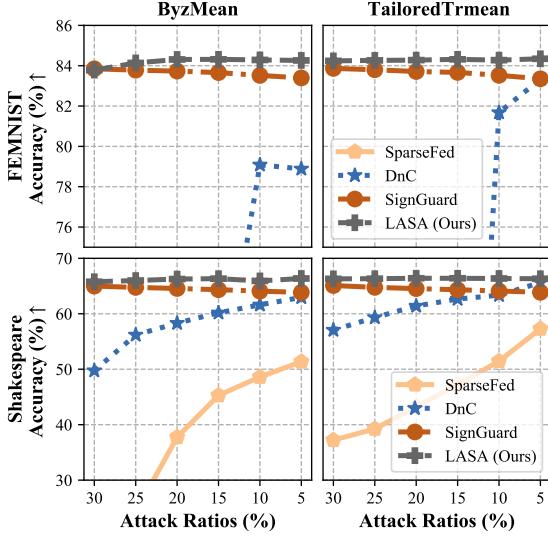


Figure 2. Testing Accuracy (%) of LASA SparseFed, DnC, and SignGuard under Various Attack Ratios on the Non-IID FEMNIST (upper) and Shakespeare (lower) Datasets.

serve the behavior of LASA in identifying malicious updates, compared with the SOTA method SignGuard. Specifically, we use two metrics, *True Positive Rate* (TPR) and *False Positive Rate* (FPR), to evaluate their performance in identifying malicious updates and benign updates. A higher TPR and lower FPR imply a more accurate benign/malicious update identification. As shown in Figure 1, LASA and SignGuard achieve significantly high TPRs, which means they can effectively identify malicious updates. However, SignGuard achieves relatively high FPRs in both attack ratio settings. For example, with a high attack ratio of 25%, at each round, SignGuard misidentifies about 15% percent of benign updates as malicious ones, so that the convergence rate of the global model drops due to the lack of benign updates. In contrast, LASA always keeps a very low FPR, demonstrating the superior performance of our unique design of layer-wise adaptive filtering.

Impact of various attack ratios. We additionally evaluate the performance of three SOTA defense methods including DnC, SignGuard, and SparseFed, and our method LASA under different attack ratios on non-IID datasets and report the results in Figure 2. Specifically, we conduct experiments under ByzMean and TailoredTrMean attacks with the attack ratio varying from 5% to 30%. In general, DnC and SparseFed’s accuracies increase as the attack ratio decreases, but they suffer from significant accuracy degradation when the attack ratio is high. For instance, on FEMNIST, even when the attack ratio is as low as 5%, SparseFed does not improve the robustness, achieving an accuracy of 7.44% under the ByzMean attack. Similarly, DnC struggles to defend against ByzMean attack effectively until the attack ratio is reduced to 10%, achieving a rela-

Table 3. Comparison of Performance with Different Components

Method	CIFAR-10 (IID)			FEMNIST (non-IID)			Avg.
	Min-Max	Lie	Noise	Min-Max	Lie	Noise	
Spar	75.00	87.00	86.04	44.39	81.08	54.91	71.40
Ma	85.73	87.99	89.39	40.45	79.38	84.27	77.87
Di	89.23	89.07	83.07	84.26	84.28	68.40	83.05
Spar+Ma	84.66	87.99	89.38	36.33	79.43	84.26	77.01
Spar+Di	89.28	89.18	83.87	84.28	84.26	69.03	83.32
Ma+Di	88.35	88.38	88.51	84.20	83.52	83.97	86.16
LASA (Ours)	88.57	88.59	88.81	84.19	83.52	84.05	86.29

tively low accuracy of 79.09%. Compared to SignGuard, LASA achieves a better and more stable performance. As the attack ratio increases, LASA only has a minor decrease in accuracy. More results under other attacks with various attack ratios are given in Appendix 7.5.

Ablation study. As LASA consists of three key components to achieve Byzantine resilience, we conduct an ablation study to investigate how each component functions. We denote *pre-aggregation sparsification*, *magnitude-based adaptive filtering*, and *direction-based adaptive filtering* as *Spar*, *Ma*, and *Di*, respectively. Experimental results on CIFAR-10 and FEMNIST are summarized in Table 3. As expected, only applying pre-aggregation sparsification does not provide enough robustness compared to LASA as it only removes partial less important malicious parameters. We observe that direction-based adaptive filtering is powerful in defending against the stealthy Min-Max and Lie attacks, but it is vulnerable to the simple Noise attack that generates malicious updates with large magnitudes. In contrast, magnitude-based adaptive filtering is effective in defending against Noise attack but is less effective against Min-Max and Lie attacks. Notably, when integrated with pre-aggregation sparsification, the performance of direction-based adaptive filtering improves: the accuracy of Spar+Di is higher than that of Di. This demonstrates the effectiveness of sparsification in improving the filtering accuracy. While LASA demonstrates comparable performance to Ma+Di, we emphasize in Section 4.1 the theoretical significance of Spar in enhancing robustness, thereby underscoring its necessity. We further discuss the impact of different radii and sparsification levels in Appendix 7.6- 7.7.

6. Conclusion

We present a novel Byzantine-resilient aggregation rule called LASA. LASA combines a pre-aggregation sparsification that sparsifies each local model update before aggregation with a novel layer-wise adaptive aggregation that filters and aggregates the sparsified model updates based on the magnitude and direction of each model layer. We theoretically analyze the robustness of LASA and provide the resilience analysis results of FL with LASA and then conduct extensive experiments on both IID and non-IID datasets to evaluate the effectiveness of LASA. Experimental results

demonstrate that LASA outperforms other defense methods under both naive and advanced attacks.

References

- [1] Youssef Allouah, Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafaël Pinot, and John Stephan. Fixing by mixing: A recipe for optimal byzantine ml under heterogeneity. In *International Conference on Artificial Intelligence and Statistics*, pages 1232–1300. PMLR, 2023. [3](#), [4](#), [5](#)
- [2] Youssef Allouah, Rachid Guerraoui, Nirupam Gupta, Rafaël Pinot, and Geovani Rizk. Robust distributed learning: Tight error bounds and breakdown point under data heterogeneity. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [3](#), [5](#)
- [3] Youssef Allouah, Rachid Guerraoui, Nirupam Gupta, Rafaël Pinot, and Geovani Rizk. Robust distributed learning: Tight error bounds and breakdown point under data heterogeneity. *Advances in Neural Information Processing Systems*, 36, 2024. [3](#), [5](#)
- [4] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020. [2](#)
- [5] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems*, 32, 2019. [1](#), [2](#), [6](#), [12](#), [13](#)
- [6] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pages 634–643. PMLR, 2019. [2](#)
- [7] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017. [1](#), [2](#), [3](#), [5](#), [6](#)
- [8] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018. [6](#), [12](#)
- [9] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *Proceedings of NDSS*, 2021. [2](#), [12](#)
- [10] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yunling Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33:2039–2050, 2020. [4](#)
- [11] El Mahdi El-Mhamdi, Sadegh Farhadkhani, Rachid Guerraoui, Arsany Guirguis, Lê-Nguyễn Hoang, and Sébastien Rouault. Collaborative learning in the jungle (decentralized, byzantine, heterogeneous, asynchronous and nonconvex learning). *Advances in Neural Information Processing Systems*, 34:25044–25057, 2021. [4](#)
- [12] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to byzantine-robust federated learning. In *29th USENIX security symposium (USENIX Security 20)*, pages 1605–1622, 2020. [1](#), [2](#), [3](#), [12](#)
- [13] Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafaël Pinot, and John Stephan. Byzantine machine learning made easy by resilient averaging of momentums. In *International Conference on Machine Learning*, pages 6246–6283. PMLR, 2022. [5](#)
- [14] Hossein Fereidooni, Alessandro Pegoraro, Phillip Rieger, Alexandra Dmitrienko, and Ahmad-Reza Sadeghi. Fre-qfed: A frequency analysis-based approach for mitigating poisoning attacks in federated learning. *arXiv preprint arXiv:2312.04432*, 2023. [2](#)
- [15] Eduard Gorbunov, Samuel Horváth, Peter Richtárik, and Gauthier Gidel. Variance reduction is an antidote to byzantines: Better rates, weaker assumptions and communication compression as a cherry on the top. *arXiv preprint arXiv:2206.00529*, 2022. [5](#)
- [16] Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, pages 3521–3530. PMLR, 2018. [1](#), [2](#), [3](#), [6](#)
- [17] Prajwal Gupta, Krishna Yadav, Brij B Gupta, Mamoun Alazab, and Thippa Reddy Gadekallu. A novel data poisoning attack in federated learning based on inverted loss function. *Computers & Security*, 130:103270, 2023. [2](#)
- [18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [7](#), [12](#)
- [19] Rui Hu, Yuanxiong Guo, and Yanmin Gong. Federated learning with sparsified model perturbation: Improving accuracy under client-level differential privacy. *IEEE Transactions on Mobile Computing*, 2023. [3](#), [4](#), [12](#)
- [20] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE symposium on security and privacy (SP)*, pages 19–35. IEEE, 2018. [2](#)
- [21] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. In *International Conference on Learning Representations*, 2021. [4](#), [5](#)
- [22] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. *University of Toronto*, 2009. [6](#), [12](#)
- [23] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the works of leslie lamport*, pages 203–226. 2019. [1](#)
- [24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. [12](#)
- [25] Sunwoo Lee, Tuo Zhang, and A Salman Avestimehr. Layer-wise adaptive model aggregation for scalable federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8491–8499, 2023. [4](#)
- [26] Han Liu, Zhiyuan Yu, Mingming Zha, XiaoFeng Wang, William Yeoh, Yevgeniy Vorobeychik, and Ning Zhang. When evil calls: Targeted adversarial voice over ip network. In *Proceedings of the 2022 ACM SIGSAC Conference on*

- Computer and Communications Security*, pages 2009–2023, 2022. 1
- [27] Jun Liu, Jianchun Liu, Hongli Xu, Yunming Liao, Zhiyuan Wang, and Qianpiao Ma. Yoga: Adaptive layer-wise model aggregation for decentralized federated learning. *IEEE/ACM Transactions on Networking*, 2023. 4
- [28] Shuo Liu, Nirupam Gupta, and Nitin H Vaidya. Approximate byzantine fault-tolerance in distributed optimization. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, pages 379–389, 2021. 6
- [29] Yi Liu, Jiangtian Nie, Xuandi Li, Syed Hassan Ahmed, Wei Yang Bryan Lim, and Chunyan Miao. Federated learning in the sky: Aerial-ground air quality sensing framework with uav swarms. *IEEE Internet of Things Journal*, 8(12):9827–9837, 2020. 1
- [30] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10092–10101, 2022. 4
- [31] Saeed Mahloujifar, Mohammad Mahmoody, and Ameer Mohammed. Universal multi-party poisoning attacks. In *International Conference on Machine Learning*, pages 4274–4283. PMLR, 2019. 2
- [32] Grigory Malinovsky, Peter Richtárik, Samuel Horváth, and Eduard Gorbunov. Byzantine robustness and partial participation can be achieved simultaneously: Just clip gradient differences. *arXiv preprint arXiv:2311.14127*, 2023. 5
- [33] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 2, 6, 12
- [34] Mark Huasong Meng, Sin G Teo, Guangdong Bai, Kailong Wang, and Jin Song Dong. Enhancing federated learning robustness using data-agnostic model pruning. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 441–453. Springer, 2023. 1, 2, 3
- [35] Thomas Minka. Estimating a dirichlet distribution, 2000. 6
- [36] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 27–38, 2017. 2
- [37] Luis Muñoz-González, Kenneth T Co, and Emil C Lupu. Byzantine-robust federated machine learning through adaptive model averaging. *arXiv preprint arXiv:1909.05125*, 2019. 3
- [38] Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018. 4
- [39] Dinh C Nguyen, Quoc-Viet Pham, Pubudu N Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. Federated learning for smart healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(3):1–37, 2022. 1
- [40] Ashwinee Panda, Saeed Mahloujifar, Arjun Nitin Bhagoji, Supriyo Chakraborty, and Prateek Mittal. Sparsefed: Mitigating model poisoning attacks in federated learning with sparsification. In *International Conference on Artificial Intelligence and Statistics*, pages 7587–7624. PMLR, 2022. 1, 2, 5, 6
- [41] Jungwuk Park, Dong-Jun Han, Minseok Choi, and Jaekyun Moon. Sageflow: Robust federated learning against both stragglers and adversaries. *Advances in neural information processing systems*, 34:840–851, 2021. 2
- [42] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022. 1, 3, 5, 6
- [43] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020. 12
- [44] Yasar Abbas Ur Rehman, Yan Gao, Pedro Porto Buarque De Gusmão, Mina Alibeigi, Jiajun Shen, and Nicholas D Lane. L-dawa: Layer-wise divergence aware weight aggregation in federated self-supervised visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16464–16473, 2023. 4
- [45] Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*, 2021. 1, 2, 3, 6, 12, 13
- [46] Shaohuai Shi, Zhenheng Tang, Qiang Wang, Kaiyong Zhao, and Xiaowen Chu. Layer-wise adaptive gradient sparsification for distributed deep learning with convergence guarantees. *arXiv preprint arXiv:1911.08727*, 2019. 3
- [47] Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang, Lingjuan Lyu, and Ji Liu. Data poisoning attacks on federated machine learning. *IEEE Internet of Things Journal*, 9(13):11365–11375, 2021. 2
- [48] Zhiyi Tian, Lei Cui, Jie Liang, and Shui Yu. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Computing Surveys*, 55(8):1–35, 2022. 1
- [49] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In *European Symposium on Research in Computer Security*, pages 480–501, 2020. 2
- [50] Kamala Varma, Yi Zhou, Nathalie Baracaldo, and Ali Anwar. Legato: A layerwise gradient aggregation algorithm for mitigating byzantine attacks in federated learning. In *2021 IEEE 14th international conference on cloud computing (CLOUD)*, pages 272–277. IEEE, 2021. 3
- [51] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33:16070–16084, 2020. 2
- [52] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashionmnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6, 12

- [53] Chulin Xie, Keli Huang, Pin Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *8th International Conference on Learning Representations, ICLR 2020*, 2020. [2](#)
- [54] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Generalized byzantine-tolerant sgd. *arXiv preprint arXiv:1802.10116*, 2018. [1, 3](#)
- [55] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Uncertainty in Artificial Intelligence*, pages 261–270. PMLR, 2020. [2](#)
- [56] Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *International Conference on Machine Learning*, pages 6893–6901. PMLR, 2019. [2](#)
- [57] Jian Xu and Shao-Lun Huang. Byzantine-resilient decentralized collaborative learning. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5253–5257. IEEE, 2022. [3](#)
- [58] Jian Xu, Shao-Lun Huang, Lingqi Song, and Tian Lan. Byzantine-robust federated learning through collaborative malicious gradient filtering. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, pages 1223–1235. IEEE, 2022. [1, 2, 3, 4, 5, 6, 12, 13](#)
- [59] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018. [1, 2, 3, 6, 13](#)
- [60] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2019. [5](#)
- [61] Zikai Zhang and Rui Hu. Byzantine-robust federated learning with variance reduction and differential privacy. In *2023 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, 2023. [1, 2, 3](#)

7. Appdenix

7.1. Attack and defense models

Attack model. We follow the attack model in previous works [5, 12, 45, 58]. Specifically, the attacker controls a subset of f malicious clients within the FL system. These clients can either be fake clients injected into the system by the attacker or genuine clients that have been compromised. The goal of the attacker is to degrade the overall performance of the global model in FL. The attacker has full knowledge of all benign updates in each training round. For additional background knowledge of the attacker, we follow the same settings of the proposed attack works. The malicious clients need not follow the prescribed local training protocol of FL and may send arbitrary local model updates to the server. Let \mathcal{B} denote the set of benign clients in the system so that $\mathcal{B} \subset \mathcal{N}$. Under the Byzantine attack, the local model update of a client $i \in \mathcal{N}$ can be represented as

$$\Delta_i = \begin{cases} \Delta_i, & \text{if } i \in \mathcal{B} \\ \beta_i, & \text{if } i \notin \mathcal{B} \end{cases} \quad (3)$$

where $\beta_i \in \mathbb{R}^d$ represents an arbitrary model depending on the specific attack method.

Defense goal. Like previous works [9, 12, 58], we assume the server to be the defender who can deploy a robust aggregation rule, denoted by F , to mitigate the negative impact of malicious local models on the global model. The server has full access to the global model and local model updates in each training round, but it does not have access to the local training data of clients. We assume the server does not know the number of malicious clients unless explicitly specified. In addition, we assume that clients' submissions are made anonymously so that the server cannot track clients' actions.

7.2. Experimental settings

We utilize six benchmark datasets of FL, including MNIST [24], Fashion-MNIST [52], FEMNIST [8], CIFAR-10 [22], CIFAR-100 [22], and Shakespeare [33] datasets, to conduct the performance evaluation. The MNIST dataset is composed of gray-scale images of size 28×28 pixels for image classification tasks. It has 60,000 images for training and 10,000 images for testing. Similar to MNIST, Fashion-MNIST (FMNIST) dataset contains 70,000 28×28 grayscale images for 10 categories of fashion products. The dataset is divided into 60,000 training images and 10,000 test images. For MNIST and FMNIST datasets, we evenly split the training data over 6,000 clients so that the distribution of private datasets on each client is IID. The Federated Extended MNIST (FEMNIST) dataset is a non-IID FL dataset extended from MNIST. It consists of 805,263 images hand-written by 3,550 users for a total of 62 classes,

including 52 for upper and lower case characters and 10 for digits. We subsample 5% of the original data following [8], resulting in 1,827 clients with a total of 450,632 images. The number of samples for each client ranges from 3 to 525. The Shakespeare dataset is naturally a non-IID FL dataset for the next character prediction tasks. Following [43], we process the original data and result in a dataset consisting of 37,784 samples from 715 clients.

The CIFAR-10 and CIFAR-100 datasets [22] is a collection of 60,000 32×32 color images with 50,000 training samples and 10,000 testing samples. All images are evenly distributed among 10/100 different classes, respectively. We split the training dataset over 100 clients for IID cases. For non-IID cases, we use Dirichlet distribution to simulate the non-IID settings on CIFAR-10 and CIFAR-100 datasets, which is controlled by a non-IID degree hyperparameter α . The default value of α is set to 0.5 in our work.

For MNIST, FMNIST, and FEMNIST datasets, given their identical image format and size, we use the same neural network architecture in [19]. Specifically, we use a CNN model composed of two convolutional layers, each followed by max-pooling and ReLU activation functions. Two linear layers are utilized to map features to classes. For CIFAR-10/100 datasets, we use ResNet-18 [18]. For the Shakespeare dataset, we implement a Recurrent Neural Network (RNN) model following [43]. The RNN model takes a sequence of characters as input and then uses an embedding layer to convert each character into an 8-dimensional feature representation. Subsequently, two Long Short-Term Memory (LSTM) layers process these embedded characters, and a final linear layer with the softmax activation is applied.

For all datasets except CIFAR-10/100, the server randomly selects $h = 100$ clients per round to perform local computations. While for CIFAR-10/100, we set $h = 25$. We use SGD with momentum as the local solver, with the decay ratio and momentum parameters set to 0.99 and 0.9, respectively, for all datasets except for Shakespeare, where it is set to 0.999 and 0.5, respectively. The learning rate is set as $\eta = 0.1$ for all datasets except for Shakespeare, where it is set to $\eta = 1.0$. By default, the filtering radius is set as $\lambda_m = \lambda_d = 1.0$ for CIFAR-10/100. While for other datasets, we set λ_m to 2.0. We define the sparsification level (SL) to be $1 - k/d$. A higher SL implies more parameters are zeroed out. In our experiments, SL is set as 0.3 for all datasets by default. We run each experiment with three random seeds and report the average of the best testing accuracies achieved in each individual training. The experiments are conducted using PyTorch and executed on NVIDIA RTX A6000 GPUs.

7.3. Evaluated attack methods

We consider eight attack methods including three naive attack methods, and five SOTA attack methods to comprehensively evaluate our method.

- *Random attack.* The malicious clients send randomized updates that follow a Gaussian distribution $N(\mu, \sigma^2 \mathbf{I}_d)$. We set $\mu = (0, \dots, 0) \in \mathbb{R}^d$ and $\sigma = 0.5$.
- *Noise attack.* The malicious clients perturb benign updates by adding Gaussian noise used in random attacks.
- *Sign-flip attack.* The malicious clients manipulate their model updates by flipping the sign coordinately.
- *Min-Max/Min-Sum attack [45].* The malicious model updates are crafted in two steps. In the first step, the attacker generates a malicious update by perturbing the average of all benign updates. Then, for Min-Max attack, the attacker optimizes the malicious update so that its maximum Euclidean distance with any benign update is upper-bounded by the maximum distance between any two benign updates, i.e., $\max_{i,j \in \mathcal{H}} \|\Delta_i - \Delta_j\|_2$. For Min-Sum attack, the malicious update is optimized to ensure that the sum of its distances with each benign update is upper-bounded by the maximum total distance of a benign update among other benign updates, i.e., $\max_{i \in \mathcal{H}} \sum_{j \in \mathcal{H}} \|\Delta_i - \Delta_j\|_2$. We additionally test a stealthy version of Min-Sum attack, where the distance of the malicious update from any benign update is bounded by the minimum (rather than maximum) total distance of benign updates. This stealthy version is tested on all the datasets except for MNIST. We follow [45] to keep the updates of all malicious clients the same.
- *AGR-tailored Trimmed-mean attack [45].* AGR-tailored Trimmed-mean (TailoredTrmean) attack is designed to attack the defense method Trmean proposed in [59] by maximizing the Euclidean distance between the aggregated result of simple average and Trmean, respectively.
- *Lie attack [5].* The malicious clients apply slight changes to their local benign updates, making it hard to be detected. Specifically, the malicious clients calculate the element-wise mean μ_j and standard error σ_j of all updates and generate the element of malicious updates by $(\beta_i)_j = \mu_j - z \times \sigma_j$, where $j \in [d]$. The scaling factor z is set to 0.5 for all experiments.
- *ByzMean attack [58].* The ByzMean attack makes the mean of updates arbitrary malicious updates. Specifically, it divides malicious clients into two groups,

each with m_1 and m_2 clients, respectively. Clients in the first group select any existing attack methods to generate their malicious updates, denoted as $\beta_{i,\forall i \in [m_1]}$. The clients in the second group generate their malicious updates to make the average of all updates exactly equal to the average of malicious updates in $[m_1]$, which can be expressed as $\beta_{i,\forall i \in [m_2]} = \frac{(n-m_1) \times \beta_{i,\forall i \in [m_1]} - \sum_{i=f+1}^n \Delta_i}{m_2}$ assuming the first f updates are malicious. We follow the same setting in [58], where the Lie attack is selected as the base attack method for the first group, and the size of two groups are set as $m_1 = \lfloor f/2 \rfloor$ and $m_2 = f - m_1$.

7.4. Additional experimental results

In this section, we set the attack ratio to 25%, and for FEMNIST and Shakespeare datasets, we set λ_d to 1.5. As shown in Table 4, LASA demonstrates its robustness against the naive and SOTA attack methods in IID settings, whereas almost all other defense methods are vulnerable to at least one attack method. Under no attack, LASA achieves a test accuracy comparable to FedAvg on MNIST dataset. This demonstrates the effectiveness of LASA in maintaining accuracy, not just in adversarial environments, but also in benign environments.

For MNIST dataset, LASA achieves the best performance against naive attacks with the highest accuracy of 97.96% for Random attack, 98.27% for Noise attack, and 97.26% for Sign-Flip attack, outperforming all other defense methods. In contrast, SignGuard, DnC and LASA can effectively defend against TailoredTrmean and ByzMean attacks. Under TailoredTrmean attack, LASA achieves the highest accuracy of 97.94%, which is +0.17% and +51.63% higher than SignGuard and DnC, respectively; under ByzMean attacks, LASA achieves the highest accuracy of 97.94%, which is 0.31% and +69.66% higher than SignGuard and DnC, respectively.

Main results in non-IID settings. Compared to FedAvg under no attack, we can see that LASA can maintain the accuracy of FL in the benign environment with only a -0.57% accuracy drop on FEMNIST dataset and even a +1.34% accuracy increase on Shakespeare dataset. We also observe that the performance of classic robust aggregation rules, including Trmean, GeoMed, Multi-Krum, and Bulyan, is poor on non-IID datasets. For example, Trmean and Multi-Krum completely failed against the ByzMean attack on FEMNIST dataset, yielding an accuracy of 5.73% and 6.48%, respectively. As we discussed in the related works, in non-IID settings, the divergence between benign model updates will increase, making these classic methods hard to filter out malicious model updates. For FEMNIST dataset, LASA outperforms all other defense methods. It achieves an accuracy of 84.26% at best under TailoredTrmean attack, which is identical to that of Mean un-

Table 4. The main results for MNIST, FEMNIST, and Shakespeare are presented.

Datasets (Model)	Defense Methods	No Attack	Naive Attacks			State-of-the-art Attacks					Average w/ Attacks
			Random	Noise	Sign-flip	TailoredTrmean	Min-Max	Min-Sum	Lie	ByzMean	
MNIST (CNN)	FedAvg	97.85	19.28	32.25	96.89	11.01	94.16	94.22	96.86	10.24	56.36
	TrMean	96.14	94.11	94.50	95.19	11.35	88.35	88.41	93.67	10.74	72.67
	GeoMed	94.59	94.66	94.66	94.21	94.76	63.99	52.29	80.82	94.17	83.69
	Multi-Krum	97.00	96.50	96.73	96.97	11.35	67.33	69.51	93.82	10.24	67.43
	Bulyan	94.95	96.42	96.41	94.20	11.70	63.89	68.00	90.98	54.88	71.06
	DnC	97.69	96.57	96.58	97.14	46.31	64.57	89.89	96.17	28.29	76.69
	SignGuard	96.64	97.70	97.70	96.85	97.78	97.58	97.46	97.58	97.63	97.54
	SparseFed	97.86	19.18	31.69	96.85	11.01	94.11	94.22	96.86	10.24	56.27
LASA (Ours)		97.35	97.96	98.27	97.26	97.94	97.93	97.94	97.54	97.94	97.85
FEMNIST (CNN)	FedAvg	84.27	42.60	48.15	81.30	5.58	58.76	81.68	81.11	1.28	50.43
	TrMean	82.23	78.26	78.81	79.13	5.70	29.80	76.72	75.79	5.73	53.12
	GeoMed	75.57	75.48	75.47	71.67	76.19	68.27	28.13	22.56	74.32	61.01
	Multi-Krum	82.85	76.13	76.48	80.00	5.58	25.83	77.25	74.91	6.48	52.58
	Bulyan	77.10	81.68	81.65	73.50	5.97	19.17	60.55	58.98	18.02	49.94
	DnC	<u>83.89</u>	75.41	76.08	80.96	63.93	66.60	80.37	78.97	22.84	68.52
	SignGuard	83.06	<u>83.75</u>	<u>83.75</u>	79.43	<u>83.80</u>	<u>83.80</u>	<u>82.59</u>	<u>82.58</u>	<u>83.78</u>	82.68
	SparseFed	84.27	42.24	48.07	81.29	5.58	60.06	81.71	81.05	1.28	50.41
LASA (Ours)		83.69	84.07	84.05	81.72	84.26	84.19	83.60	83.52	84.14	83.94
Shakespeare (LSTM)	FedAvg	63.74	45.00	47.28	60.43	39.01	59.17	63.35	62.79	24.24	50.41
	TrMean	63.15	59.09	59.43	59.83	42.23	57.54	62.60	61.86	37.38	54.75
	GeoMed	57.63	57.67	57.67	52.55	57.89	57.72	57.89	56.24	56.28	57.24
	Multi-Krum	62.26	61.55	61.73	59.11	35.11	54.30	62.09	58.34	23.16	52.92
	Bulyan	60.89	62.73	62.76	58.05	49.39	54.61	60.71	59.11	52.90	57.41
	DnC	<u>64.67</u>	61.38	61.47	<u>60.80</u>	59.32	61.10	64.70	62.30	56.18	60.65
	SignGuard	63.65	<u>65.26</u>	<u>65.26</u>	59.84	<u>64.76</u>	<u>64.76</u>	60.83	62.35	<u>64.76</u>	<u>61.97</u>
	SparseFed	63.72	44.49	47.24	60.40	39.24	59.84	63.31	62.77	24.27	50.69
LASA (Ours)		65.08	66.25	66.24	62.56	66.32	65.63	<u>64.02</u>	<u>64.25</u>	<u>65.99</u>	65.16

der no attack. In addition, LASA outperforms SignGuard more significantly in non-IID settings, compared to their performance in IID settings. Specifically on Shakespeare dataset, the performance of SignGuard is not stable. For example, under Sign-Flip attack, the accuracy of SignGuard drops to 59.84%, while LASA achieves the highest accuracy of 62.56% (+2.72%). Under Min-Sum attack, SignGuard’s accuracy drops to 60.83%, while LASA achieves an accuracy of 64.017% (+3.19%), which is comparable to the best accuracy achieved by DnC.

In a nutshell, the performance of LASA is not only manifested in attack scenarios but also in the absence of any attacks, which aligns with the design principles of LASA. Moreover, LASA shows robustness to both IID and more challenging non-IID cases. By adeptly integrating pre-aggregation sparsification and layer-wise adaptive aggregation, LASA effectively mitigates the impact of updates that diverge from others. The robustness of LASA, illustrated by the above-mentioned results, emphasizes its potential as a robust defense method in securing federated learning environments against a wide collection of attacks, ultimately enhancing the reliability of federated learning systems.

7.5. More results under various attack ratios

We evaluate the performance of three SOTA defense methods including DnC, SignGuard and SparseFed, and our method LASA under different attack ratios on non-IID datasets and report the results in Fig. 3. Specifically, we conduct experiments under one naive attack and four SOTA attacks with the attack ratio varying from 5% to 30%. In Fig. 3, the *Baseline* represents the non-robust method Mean under no attack. In general, DnC and SparseFed’s accuracies increase as the attack ratio decreases, but they suffer from significant accuracy degradation when the attack ratio is high, especially under Byzmean and TailoredTrmean attacks. For instance, on FEMNIST dataset, even when the attack ratio is as low as 5%, SparseFed does not improve the robustness, achieving an accuracy of 7.44% under the ByzMean attack. Similarly, DnC struggles to defend against ByzMean attack effectively until the attack ratio is reduced to 10%, achieving a relatively low accuracy of 79.09%. SignGuard outperforms DnC and SparseFed significantly. However, under Byzmean, TailoredTrmean, Noise, and Min-Max attacks, the accuracy of SignGuard decreases as the attack ratio decreases. Compared to SignGuard, our method LASA achieves a better and more stable performance. As the attack ratio increases, LASA only has

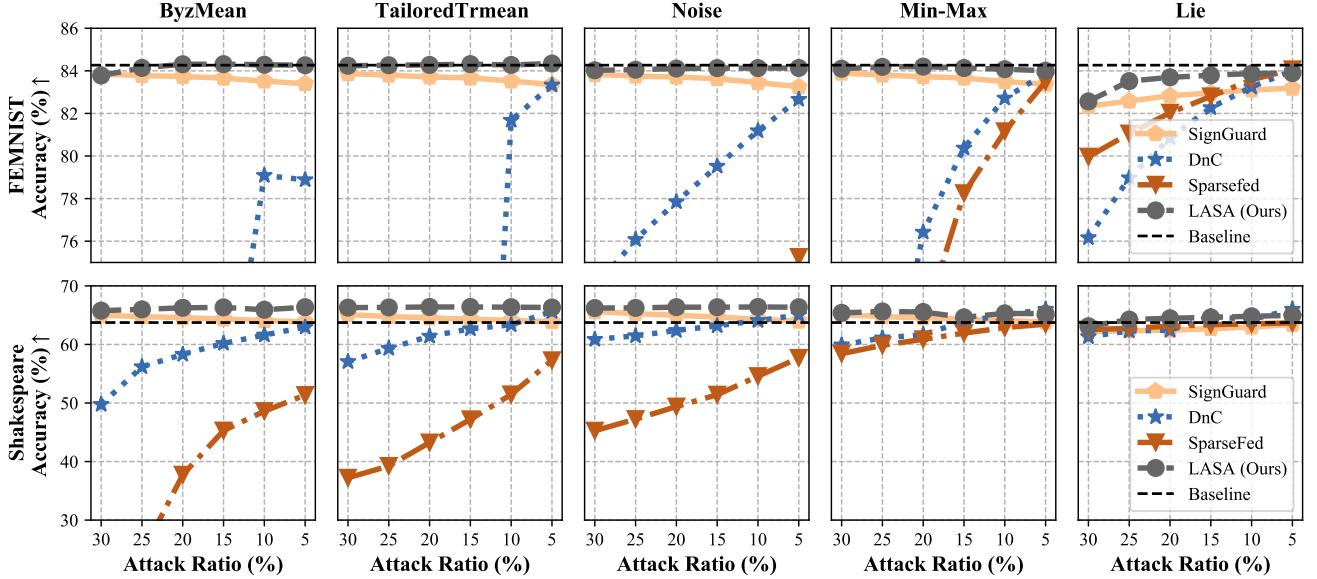


Figure 3. Testing Accuracy of LASA, SignGuard, DnC and SparseFed under Various Attack Ratios in non-IID Settings.

Table 5. Performance of LASA with Different Sparsification Levels.

Att.	Data.	Sparsification Level						
		0.1	0.3	0.5	0.7	0.9	0.95	0.99
ByzMean	M	97.833	97.943	97.821	97.543	98.053	97.880	97.490
	FM	87.820	87.647	87.943	87.867	87.803	87.740	86.437
	FEM	84.143	84.138	84.137	84.118	83.834	83.489	81.120
	Sha	66.024	65.990	65.842	65.409	64.355	63.055	60.463
Min-Max	M	97.310	97.930	97.493	97.307	97.557	96.950	97.593
	FM	87.917	87.907	87.920	87.967	87.330	87.707	86.353
	FEM	84.184	84.264	84.203	84.162	83.772	83.361	81.038
	Sha	64.723	66.324	65.472	65.032	63.654	62.527	60.090
Noise	M	98.223	98.270	98.320	97.643	98.050	97.923	97.800
	FM	87.877	87.870	87.893	87.933	87.623	87.897	86.423
	FEM	84.061	84.053	84.023	84.018	83.645	83.269	80.537
	Sha	66.255	66.244	66.102	65.754	64.506	63.546	60.686

a minor decrease in accuracy.

7.6. Impact of sparsification level

As we stated in Section 4.1, the optimal sparsification parameter k should balance the tradeoff between sparsification error and robustness improvement. Here, we empirically study the impact of different k on learning performance. Recall that the SL is defined as $1 - k/d$, hence, a smaller k implies a higher SL and a heavier sparsification. We report the performance of LASA under Noise, Min-Max, and ByzMean attacks with SLs varying from 0.1 to 0.99 in Table 5, where M, FM, FEM, and Sha represent MNIST, FMNIST, FEMNIST, and Shakespeare datasets, respectively. The results demonstrate that there exists an optimal SL that maximizes robustness and a very high SL may lead to a sig-

Table 6. Performance of LASA with Different Filtering Radius

Con.	MNIST		FEMNIST		FEMNIST	
	λ_d	λ_m	Noise	ByzMean	Noise	ByzMean
1.0	1.0		97.963	97.803	87.950	87.887
1.0	1.5		97.883	97.843	88.023	87.720
1.0	2.0		98.270	97.943	87.870	87.647
1.0	4.0		91.743	97.840	77.400	77.930
1.5	2.0		97.927	98.023	87.937	87.640
2.0	2.0		97.593	97.487	87.950	84.000
3.0	2.0		97.883	66.897	87.917	67.250
						84.225
						28.300

nificant accuracy drop. For example, as SL increases, the accuracy of LASA on FEMNIST dataset increases to 87.94% and then decreases to 86.44% under ByzMean attack. This occurs because the sparsification error overwhelms the robustness improvement when SL is too large. We also observe that the sensitivity of LASA on SL depends on both the dataset and the attack method.

7.7. Impact of filtering radius

In this subsection, we study the performance of LASA with different filtering radius λ_m and λ_d . A smaller λ_m or λ_d indicates more stringent filtering and results in a smaller benign set for aggregation. As shown in Table 6, there exist optimal λ_m and λ_d that balance the filtering intensity and maximize the model accuracy. We also observe that the effectiveness of Noise attack is marginally affected by λ_d , as random noise perturbation does not change the sign purity in expectation. For all datasets, the optimal λ_d under Noise attack is 1.0 (note that for FEMNIST, the best accuracy when $\lambda_d = 3.0$ is comparable to the accuracy

when $\lambda_d = 1.0$). However, as Noise attack adds Gaussian noise to the model updates to increase their magnitude (in L_2 norm), the effectiveness of Noise attack is sensitive to the values of λ_m . For different datasets, the optimal λ_m are different. For the advanced ByzMean attack, its effectiveness is marginally affected by λ_m , as the accuracy of LASA does not change much when λ_m increases from 1.0 to 2.0. This demonstrates that the magnitudes of malicious updates generated by ByzMean attack are close to that of benign models. In order to make the attack effective, ByzMean attack mainly focuses on manipulating the model direction, making it sensitive to the direction filtering radius λ_d : the accuracy of LASA vibrates a lot as λ_d increases. Additionally, both λ_m and λ_d should not be too large to compromise the effectiveness of the filtering.

7.8. Computational cost of LASA

We evaluate the computational cost of LASA in comparison to other methods. LASA incorporates pre-aggregation sparsification, leading to a complexity of $O(d \log d)$ due to the use of sorting algorithms like *merge sort* in the parameter space of local updates. Consequently, the worst-case computational expense for LASA is $O(nd \log d)$. Despite this, LASA’s computational burden is on par with other methods such as Krum and Multi-Krum, which have a complexity of $O(d n^2)$, and Trmean with $O(d n \log n)$.

7.9. Proof preliminaries

7.9.1 Useful Inequalities

Lemma 2. Given any two vectors $a, b \in \mathbb{R}^d$,

$$2 \langle a, b \rangle \leq \alpha \|a\|^2 + \frac{1}{\alpha} \|b\|^2, \forall \alpha > 0.$$

Lemma 3. Given any two vectors $a, b \in \mathbb{R}^d$,

$$\|a + b\|^2 \leq (1 + \delta) \|a\|^2 + (1 + \delta^{-1}) \|b\|^2, \forall \delta > 0.$$

Lemma 4. Given arbitrary set of n vectors $\{a_i\}_{i=1}^n, a_i \in \mathbb{R}^d$,

$$\left\| \sum_{i=1}^n a_i \right\|^2 \leq n \sum_{i=1}^n \|a_i\|^2.$$

Lemma 5. If the learning rate $\eta \leq 1/2\tau$, under Assumption 2 and 3, the local divergence of benign model updates are bounded as follows:

$$\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 \leq 2\bar{\nu} + \bar{\zeta} \quad (4)$$

Proof. Given that $\Delta_i = \eta \sum_{s=0}^{\tau-1} g_i^s$ where η is the learning rate and g_i^s is the local stochastic gradient over the mini-batch s . We have

$$\begin{aligned} \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 &= \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \eta \sum_{s=0}^{\tau-1} g_i^s - \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \eta \sum_{s=0}^{\tau-1} g_i^s \right\|^2 \\ &= \frac{\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \sum_{s=0}^{\tau-1} g_i^s - \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{s=0}^{\tau-1} g_i^s \right\|^2 \\ &\leq \frac{\tau\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{s=0}^{\tau-1} \mathbb{E} \left\| g_i^s - \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} g_i^s \right\|^2 \\ &= \frac{\tau\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{s=0}^{\tau-1} \mathbb{E} \left\| (g_i^s - \nabla \mathcal{L}_i(\theta_i^s)) + \left(\nabla \mathcal{L}_{\mathcal{B}}(\theta_i^s) - \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} g_i^s \right) + (\nabla \mathcal{L}_i(\theta_i^s) - \nabla \mathcal{L}_{\mathcal{B}}(\theta_i^s)) \right\|^2 \\ &\leq \underbrace{\frac{3\tau\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{s=0}^{\tau-1} \mathbb{E} \|g_i^s - \nabla \mathcal{L}_i(\theta_i^s)\|^2}_{T_1} + \underbrace{\frac{3\tau\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{s=0}^{\tau-1} \mathbb{E} \left\| \nabla \mathcal{L}_{\mathcal{B}}(\theta_i^s) - \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} g_i^s \right\|^2}_{T_2} \\ &\quad + \underbrace{\frac{3\tau\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{s=0}^{\tau-1} \mathbb{E} \|\nabla \mathcal{L}_i(\theta_i^s) - \nabla \mathcal{L}_{\mathcal{B}}(\theta_i^s)\|^2}_{T_3}, \end{aligned} \quad (5)$$

where the first inequality follows Lemma 4, and the last second follows Lemma 3. For T_1 , with Assumption 2, we have

$$T_1 \leq \bar{\nu}. \quad (7)$$

For T_2 , we have

$$T_2 = \mathbb{E} \left\| \nabla \mathcal{L}_{\mathcal{B}}(\theta_i^s) - \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} g_i^s \right\|^2 = \mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} (\nabla \mathcal{L}_i(\theta_i^s) - g_i^s) \right\|^2 \leq \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \|\nabla \mathcal{L}_i(\theta_i^s) - g_i^s\|^2 \leq \bar{\nu}, \quad (8)$$

where the first inequality follows Lemma 4, and the last inequality follow Assumption 2. For T_3 , we have

$$T_3 = \frac{3\tau\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{s=0}^{\tau-1} \mathbb{E} \|\nabla \mathcal{L}_i(\theta_i^s) - \nabla \mathcal{L}_{\mathcal{B}}(\theta_i^s)\|^2 \leq 3\tau\eta^2 \sum_{s=0}^{\tau-1} \bar{\zeta} = 3\tau^2\eta^2\bar{\zeta} \quad (9)$$

by Assumption 3.

Plugging 7, 8, and 9 back to 6, with $\eta \leq 1/2\tau$, we have

$$\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 \leq 2\bar{\nu} + \bar{\zeta}.$$

This concludes the proof. \square

7.9.2 Proof of Lemma 1

Proof. Recall that LASA denoted by $F(\cdot) : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^d$ is a layer-wise aggregation rule, i.e., there exist L real-valued functions $F_1, \dots, F_L : \mathbb{R}^{d \times n} \rightarrow \mathbb{R}^d$ such that for all $\Delta_1, \dots, \Delta_n \in \mathbb{R}^d$, $[F(\Delta_1, \dots, \Delta_n)]_l = F_l(\Delta_1^l, \dots, \Delta_n^l)$. As LASA utilizes layer-wise aggregation, we have

$$F_l(\Delta_1, \dots, \Delta_n) = \frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{S}^l} \hat{\Delta}_i^l,$$

where $\hat{\Delta}_i^l$ be the l -th layer of the Top- k sparsified model $\hat{\Delta}_i$ and \mathcal{S}^l is the indices set of benign updates in l -th layer shown in Algorithm 1. We denote the indices set of Top- k parameters of a model/layer by \mathcal{K} and the set of remaining parameters by \mathcal{K}^- . Let $[\Delta]_{\mathcal{K}}$ represent a sparsified model with only parameters in \mathcal{K} (the rest are zero), then we have

$$\begin{aligned} \mathbb{E} \|F(\Delta_1, \dots, \Delta_n) - \bar{\Delta}_{\mathcal{B}}\|^2 &= \mathbb{E} \sum_{l=1}^L \|F_l(\Delta_1, \dots, \Delta_n) - \bar{\Delta}_{\mathcal{B}}^l\|^2 \\ &= \mathbb{E} \sum_{l=1}^L \left\| \frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{S}^l} \hat{\Delta}_i^l - \bar{\Delta}_{\mathcal{B}}^l \right\|^2 \\ &= \mathbb{E} \sum_{l=1}^L \frac{1}{|\mathcal{S}^l|^2} \left\| \sum_{i \in \mathcal{S}^l} \hat{\Delta}_i^l - \bar{\Delta}_{\mathcal{B}}^l \right\|^2 \\ &= \mathbb{E} \sum_{l=1}^L \frac{1}{|\mathcal{S}^l|^2} \left\| \sum_{i \in \mathcal{S}^l} [\hat{\Delta}_i^l - \bar{\Delta}_{\mathcal{B}}^l]_{\mathcal{K}_i^l} + \sum_{i \in \mathcal{S}^l} [-\bar{\Delta}_{\mathcal{B}}^l]_{\mathcal{K}_i^{l-}} \right\|^2 \\ &= \mathbb{E} \sum_{l=1}^L \frac{1}{|\mathcal{S}^l|^2} \left\| \sum_{i \in \mathcal{S}^l} [\Delta_i^l - \bar{\Delta}_{\mathcal{B}}^l]_{\mathcal{K}_i^l} + \sum_{i \in \mathcal{S}^l} [-\bar{\Delta}_{\mathcal{B}}^l]_{\mathcal{K}_i^{l-}} \right\|^2 \\ &= \mathbb{E} \sum_{l=1}^L \frac{1}{|\mathcal{S}^l|^2} \left(\left\| \sum_{i \in \mathcal{S}^l} [\Delta_i^l - \bar{\Delta}_{\mathcal{B}}^l]_{\mathcal{K}_i^l} \right\|^2 + \left\| \sum_{i \in \mathcal{S}^l} [\bar{\Delta}_{\mathcal{B}}^l]_{\mathcal{K}_i^{l-}} \right\|^2 \right). \end{aligned}$$

Let $c_i^l := \left\| [\Delta_i^l - \bar{\Delta}_{\mathcal{B}}^l]_{\mathcal{K}_i^l} \right\|^2 / \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2$, $b_{\mathcal{B}}^l := \left\| [\bar{\Delta}_{\mathcal{B}}^l]_{\mathcal{K}_i^{l-}} \right\|^2 / \|\bar{\Delta}_{\mathcal{B}}\|^2$, $C_{\mathcal{B}}^2 := \|\bar{\Delta}_{\mathcal{B}}\|^2$, $b_{\mathcal{B}} := \sum_{l=1}^L b_{\mathcal{B}}^l$, and $c_i := \sum_{l=1}^L c_i^l$, we have

$$\begin{aligned} \mathbb{E} \|F(\Delta_1, \dots, \Delta_n) - \bar{\Delta}_{\mathcal{B}}\|^2 &\leq \mathbb{E} \sum_{l=1}^L \frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{S}^l} \left(\left\| [\Delta_i^l - \bar{\Delta}_{\mathcal{B}}^l]_{\mathcal{K}_i^l} \right\|^2 + \left\| [\bar{\Delta}_{\mathcal{B}}^l]_{\mathcal{K}_i^{l-}} \right\|^2 \right) \\ &= \mathbb{E} \sum_{l=1}^L \frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{S}^l} \left(c_i^l \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 + b_{\mathcal{B}}^l \|\bar{\Delta}_{\mathcal{B}}\|^2 \right) \\ &= \mathbb{E} \sum_{l=1}^L \frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{S}^l} \left(c_i^l \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 + b_{\mathcal{B}}^l C_{\mathcal{B}}^2 \right), \\ &= \mathbb{E} \sum_{l=1}^L \frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{S}^l} c_i^l \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 + C_{\mathcal{B}}^2 \sum_{l=1}^L b_{\mathcal{B}}^l \\ &= \underbrace{\mathbb{E} \frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{S}^l} c_i \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2}_{T_1} + C_{\mathcal{B}}^2 b_{\mathcal{B}}, \tag{10} \end{aligned}$$

where the first inequality follows Lemma 4. Note that $c_i = \left\| [\Delta_i - \bar{\Delta}_{\mathcal{B}}]_{\mathcal{K}_i} \right\|^2 / \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2$ and $b_{\mathcal{B}} = \left\| [\bar{\Delta}_{\mathcal{B}}]_{\mathcal{K}_i^-} \right\|^2 / \|\bar{\Delta}_{\mathcal{B}}\|^2$.

Now we treat T_1 . If $\mathcal{S}^l \subseteq \mathcal{B}$, we have

$$T_1 = \mathbb{E} \left[\frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{S}^l} c_i \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 \right] \leq \mathbb{E} \left[\frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{B}} c_i \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 \right]. \quad (11)$$

If $\mathcal{S}^l \not\subseteq \mathcal{B}$, let $\mathcal{P} = \mathcal{S}^l \cap \mathcal{B}$, and $\mathcal{R} = \mathcal{S}^l \setminus \mathcal{B}$, let $C_{\mathcal{M},i}^2 := \|\Delta_i\|^2, \forall i \in [N] \setminus \mathcal{B}$, then we have

$$\begin{aligned} T_1 &= \mathbb{E} \left[\frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{S}^l} c_i \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 \right] = \mathbb{E} \left[\frac{1}{|\mathcal{S}^l|} \left(\sum_{i \in \mathcal{P}} c_i \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 + \sum_{i \in \mathcal{R}} c_i \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 \right) \right] \\ &\leq \mathbb{E} \left[\frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{B}} c_i \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 \right] + \mathbb{E} \left[\frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{R}} c_i \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 \right] \\ &\leq \mathbb{E} \left[\frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{B}} c_i \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 \right] + \mathbb{E} \left[\frac{2}{|\mathcal{S}^l|} \sum_{i \in \mathcal{R}} c_i (\|\Delta_i\|^2 + \|\bar{\Delta}_{\mathcal{B}}\|^2) \right] \\ &= \mathbb{E} \left[\frac{1}{|\mathcal{S}^l|} \sum_{i \in \mathcal{B}} c_i \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 \right] + \mathbb{E} \left[\frac{2}{|\mathcal{S}^l|} \sum_{i \in \mathcal{R}} c_i (C_{\mathcal{M},i}^2 + C_{\mathcal{B}}^2) \right], \end{aligned} \quad (12)$$

where the second inequality follows Lemma 3.

Due to the use of MZ-score, models in \mathcal{S}^l are centered around the median within a λ_m (and λ_d) radius. If the radius parameter λ_m or λ_d equals to zero, only the median model (based on l_2 -norm or PDP) will be selected for averaging. To maximize benign model inclusion in averaging, the radius parameters λ_m and λ_d are set sufficiently large to ensure $|\mathcal{S}^l| \geq n/2 - f$. More precisely, assume there exist two positive constants λ_m^+ and λ_d^+ , and if the radius parameters λ_m and λ_d in Algorithm 1 satisfy $\lambda_m \geq \lambda_m^+, \lambda_d \geq \lambda_d^+$, we have $|\mathcal{S}^l| \geq n/2 - f, \forall l \in [L]$. Integrated with 11 and 12, we have

$$\begin{aligned} T_1 &\leq \begin{cases} \frac{2}{n-2f} \mathbb{E} \sum_{i \in \mathcal{B}} c_i \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2, & \text{if } \mathcal{S}^l \subseteq \mathcal{B} \\ \frac{2}{n-2f} \mathbb{E} \sum_{i \in \mathcal{B}} c_i \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 + \frac{4}{n-2f} \mathbb{E} \sum_{i \in \mathcal{R}} c_i (C_{\mathcal{M},i}^2 + C_{\mathcal{B}}^2), & \text{if } \mathcal{S}^l \not\subseteq \mathcal{B} \end{cases} \\ &\leq \frac{2}{n-2f} \mathbb{E} \sum_{i \in \mathcal{B}} c_i \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 + \frac{4}{n-2f} \mathbb{E} \sum_{i \in \mathcal{R}} c_i (C_{\mathcal{M},i}^2 + C_{\mathcal{B}}^2) \\ &\leq \frac{2c_{max}}{n-2f} \mathbb{E} \sum_{i \in \mathcal{B}} \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 + \frac{4c_{max}}{n-2f} \sum_{i \in \mathcal{R}} (C_{\mathcal{M},i}^2 + C_{\mathcal{B}}^2) \\ &= \frac{2c_{max}|\mathcal{B}|}{n-2f} \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 + \frac{4c_{max}}{n-2f} \sum_{i \in \mathcal{R}} (C_{\mathcal{M},i}^2 + C_{\mathcal{B}}^2) \\ &= \frac{2c_{max}(n-f)}{n-2f} \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \|\Delta_i - \bar{\Delta}_{\mathcal{B}}\|^2 + \frac{4c_{max}}{n-2f} \sum_{i \in \mathcal{R}} (C_{\mathcal{M},i}^2 + C_{\mathcal{B}}^2) \\ &\leq \frac{2(n-f)}{n-2f} (2\bar{\nu} + \bar{\zeta}) c_{max} + \underbrace{\frac{4c_{max}}{n-2f} \sum_{i \in \mathcal{R}} (C_{\mathcal{M},i}^2 + C_{\mathcal{B}}^2)}_{T_2}, \end{aligned} \quad (13)$$

where the second inequality holds as $c_{max} := \max\{c_i, i \in [N]\}$ and the last inequality follows Lemma 5.

Assume the benign model update is bounded as $\|\Delta_i\|^2 \leq C^2, \forall i \in \mathcal{B}$, which can be achieved by using gradient clipping in practice. Assume the malicious model update is bounded as $\|\Delta_i\|^2 \leq C_{\lambda_m}^2, \forall i \in [N] \setminus \mathcal{B}$, which depends on the specific attack method and our magnitude-based filtering that is controlled by λ_m in Algorithm 1. We have

$$T_2 = \sum_{i \in \mathcal{R}} (C_{\mathcal{M},i}^2 + C_{\mathcal{B}}^2) \leq |\mathcal{R}| (C_{\lambda_m}^2 + C^2) \leq f (C_{\lambda_m}^2 + C^2), \quad (14)$$

as $|\mathcal{R}| \leq |[N] \setminus \mathcal{B}| \leq f$. Therefore,

$$\begin{aligned} T_1 &\leq c_{max} \left(\frac{2(n-f)}{n-2f} (2\bar{\nu} + \bar{\zeta}) + \frac{4f}{n-2f} (C_{\lambda_m}^2 + C^2) \right) \\ &\leq c_k \left(\frac{2(n-f)}{n-2f} (2\bar{\nu} + \bar{\zeta}) + \frac{4f}{n-2f} (C_{\lambda_m}^2 + C^2) \right) \\ &\leq c_k \left(1 + \frac{f}{n-2f} \right) (4\bar{\nu} + 2\bar{\zeta} + 4C_{\lambda_m}^2 + 4C^2), \end{aligned} \quad (15)$$

if the sparsification applied to the local model update satisfies Assumption 4 so that $c_{max} \leq c_k$. Summarizing to (10), we have

$$\begin{aligned} \mathbb{E}\|F(\Delta_1, \dots, \Delta_n) - \bar{\Delta}_{\mathcal{B}}\|^2 &\leq T_1 + C_{\mathcal{B}}^2 b_{\mathcal{B}} \\ &\leq T_1 + b_k C^2 \\ &\leq c_k \left(1 + \frac{f}{n-2f} \right) (4\bar{\nu} + 2\bar{\zeta} + 4C_{\lambda_m}^2 + 4C^2) + b_k C^2 \end{aligned} \quad (16)$$

Discussion on the selection of k : When no sparsification is applied, i.e., when $k = d$, we have $c_k = 1$ and $b_k = 0$. In this case, the robustness upper bound is

$$\kappa_1 = \left(1 + \frac{f}{n-2f} \right) (4\bar{\nu} + 2\bar{\zeta} + 4C_{\lambda_m}^2 + 4C^2) = O \left(1 + \frac{f}{n-2f} \right).$$

When $k = 0$, we have $c_k = 0$ and $b_k = 1$, then

$$\kappa = C^2,$$

which indicates the greatest sparsification error affecting robustness. When $0 < k < d$, the robustness upper bound is

$$\kappa_2 = (1 + \epsilon)c_k \left(1 + \frac{f}{n-2f} \right) (4\bar{\nu} + 2\bar{\zeta} + 4C_{\lambda_m}^2 + 4C^2) = O \left(c_k \left(1 + \frac{f}{n-2f} \right) \right)$$

if the sparsification parameter k is selected to satisfy that

$$\text{Condition 1 : } c_k \leq \frac{1}{1 + \epsilon}$$

and

$$\text{Condition 2 : } \frac{b_k}{c_k} \leq \epsilon \left(\frac{4\bar{\nu} + 2\bar{\zeta} + 4C_{\lambda_m}^2}{C^2} + 4 \right)$$

with a positive constant ϵ . As $(1 + \epsilon)c_k \leq 1$, we have

$$\kappa_2 \leq \kappa_1,$$

which demonstrates the effectiveness of sparsification for improving robustness. This finally concludes the proof. \square

7.9.3 Proof of Theorem 1

Proof. Given the update rule $\theta^{t+1} = \theta^t - \bar{\Delta}^t = \theta^t - \eta \tilde{\Delta}^t$ where $\tilde{\Delta}_i^t := \sum_{r=0}^{\tau-1} g_i^{t,r} = \tau d_i^t$, for ease of expression, we let $\tilde{\Delta}_{\mathcal{B}^t} := \frac{1}{|\mathcal{B}^t|} \sum_{i \in \mathcal{B}^t} \tilde{\Delta}_i^t$ and $h_i^t := \mathbb{E}[d_i^t] = \mathbb{E}\left[(1/\tau) \sum_{r=0}^{\tau-1} g_i^{t,r}\right] = (1/\tau) \sum_{r=0}^{\tau-1} \nabla \mathcal{L}_i(\theta_i^{t,r})$. With Assumption 1, we have the following for all $t \in [0, T-1]$:

$$\begin{aligned}
\mathcal{L}_{\mathcal{B}}(\theta^{t+1}) - \mathcal{L}_{\mathcal{B}}(\theta^t) &\leq \mathbb{E} \langle \nabla \mathcal{L}_{\mathcal{B}}(\theta^t), \theta^{t+1} - \theta^t \rangle + \frac{\mu}{2} \mathbb{E} \|\theta^{t+1} - \theta^t\|^2 \\
&= -\eta \mathbb{E} \langle \nabla \mathcal{L}_{\mathcal{B}}(\theta^t), \tilde{\Delta}^t \rangle + \frac{\mu \eta^2}{2} \mathbb{E} \|\tilde{\Delta}^t\|^2 \\
&= -\eta \mathbb{E} \langle \nabla \mathcal{L}_{\mathcal{B}}(\theta^t), \tilde{\Delta}^t + \tilde{\Delta}_{\mathcal{B}^t} - \tilde{\Delta}_{\mathcal{B}^t} \rangle + \frac{\mu \eta^2}{2} \mathbb{E} \|\tilde{\Delta}^t\|^2 \\
&= -\eta \mathbb{E} \langle \nabla \mathcal{L}_{\mathcal{B}}(\theta^t), \tilde{\Delta}_{\mathcal{B}^t} \rangle - \eta \mathbb{E} \langle \nabla \mathcal{L}_{\mathcal{B}}(\theta^t), \tilde{\Delta}^t - \tilde{\Delta}_{\mathcal{B}^t} \rangle + \frac{\mu \eta^2}{2} \mathbb{E} \|\tilde{\Delta}^t\|^2 \\
&= \underbrace{-\eta \mathbb{E} \left\langle \nabla \mathcal{L}_{\mathcal{B}}(\theta^t), \frac{1}{|\mathcal{B}^t|} \sum_{i \in \mathcal{B}^t} \tilde{\Delta}_i^t \right\rangle}_{T_1} + \underbrace{\eta \mathbb{E} \left\langle \nabla \mathcal{L}_{\mathcal{B}}(\theta^t), \tilde{\Delta}_{\mathcal{B}^t} - \tilde{\Delta}^t \right\rangle}_{T_2} + \underbrace{\frac{\mu \eta^2}{2} \mathbb{E} \|\tilde{\Delta}^t\|^2}_{T_3}. \quad (17)
\end{aligned}$$

Now we treat T_1 , T_2 , and T_3 respectively. We decompose T_1 by

$$\begin{aligned}
T_1 &= -\eta \mathbb{E} \left\langle \nabla \mathcal{L}_{\mathcal{B}}(\theta^t), \frac{1}{|\mathcal{B}^t|} \sum_{i \in \mathcal{B}^t} \tilde{\Delta}_i^t \right\rangle = -\eta \tau \mathbb{E} \left\langle \nabla \mathcal{L}_{\mathcal{B}}(\theta^t), \frac{1}{|\mathcal{B}^t|} \sum_{i \in \mathcal{B}^t} d_i^t \right\rangle = -\eta \tau \mathbb{E} \left\langle \nabla \mathcal{L}_{\mathcal{B}}(\theta^t), \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t \right\rangle \\
&= \frac{\eta \tau}{2} \mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t - \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 - \frac{\eta \tau}{2} \mathbb{E} \|\nabla \mathcal{L}_{\mathcal{B}}(\theta^t)\|^2 - \frac{\eta \tau}{2} \mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t \right\|^2, \quad (18)
\end{aligned}$$

where we use the fact that $-2 \langle a, b \rangle = \|a - b\|^2 - \|a\|^2 - \|b\|^2$.

We decompose T_2 as

$$T_2 = \eta \mathbb{E} \left\langle \nabla \mathcal{L}_{\mathcal{B}}(\theta^t), \tilde{\Delta}_{\mathcal{B}^t} - \tilde{\Delta}^t \right\rangle \leq \frac{\eta \alpha}{2} \mathbb{E} \|\nabla \mathcal{L}_{\mathcal{B}}(\theta^t)\|^2 + \frac{\eta}{2\alpha} \mathbb{E} \|\tilde{\Delta}^t - \tilde{\Delta}_{\mathcal{B}^t}\|^2, \quad (19)$$

where the first inequality follows Lemma 2 with a $\alpha > 0$.

We decompose T_3 as

$$\begin{aligned}
T_3 &= \frac{\mu \eta^2}{2} \mathbb{E} \|\tilde{\Delta}^t\|^2 = \frac{\mu \eta^2}{2} \mathbb{E} \|\tilde{\Delta}^t + \tilde{\Delta}_{\mathcal{B}^t} - \tilde{\Delta}_{\mathcal{B}^t}\|^2 \\
&\leq \mu \eta^2 \mathbb{E} \|\tilde{\Delta}_{\mathcal{B}^t}\|^2 + \mu \eta^2 \mathbb{E} \|\tilde{\Delta}^t - \tilde{\Delta}_{\mathcal{B}^t}\|^2 \\
&= \mu \eta^2 \mathbb{E} \left\| \frac{1}{|\mathcal{B}^t|} \sum_{i \in \mathcal{B}^t} \tilde{\Delta}_i^t \right\|^2 + \mu \eta^2 \mathbb{E} \|\tilde{\Delta}^t - \tilde{\Delta}_{\mathcal{B}^t}\|^2 \\
&\leq \frac{\mu \eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \|\tilde{\Delta}_i^t\|^2 + \mu \eta^2 \mathbb{E} \|\tilde{\Delta}^t - \tilde{\Delta}_{\mathcal{B}^t}\|^2, \quad (20)
\end{aligned}$$

where the first inequality follows Lemma 3 with $\delta = 1$ and the second inequality follows Lemma 4.

Combining 18, 19, 20 and, 17, we get

$$\begin{aligned}
\mathcal{L}_{\mathcal{B}}(\theta^{t+1}) - \mathcal{L}_{\mathcal{B}}(\theta^t) &\leq \frac{\eta\tau}{2}\mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t - \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 - \frac{\eta\tau}{2}\mathbb{E} \left\| \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 - \frac{\eta\tau}{2}\mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t \right\|^2 \\
&\quad + \frac{\eta\alpha}{2}\mathbb{E} \left\| \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 + \frac{\eta}{2\alpha}\mathbb{E} \left\| \tilde{\Delta}^t - \tilde{\Delta}_{\mathcal{B}^t} \right\|^2 + \frac{\mu\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \tilde{\Delta}_i^t \right\|^2 + \mu\eta^2\mathbb{E} \left\| \tilde{\Delta}^t - \tilde{\Delta}_{\mathcal{B}^t} \right\|^2 \\
&= - \left(\frac{\eta\tau}{2} - \frac{\eta\alpha}{2} \right) \mathbb{E} \left\| \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 + \frac{\eta\tau}{2}\mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t - \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 \\
&\quad + \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) \mathbb{E} \left\| \tilde{\Delta}^t - \tilde{\Delta}_{\mathcal{B}^t} \right\|^2 - \frac{\eta\tau}{2}\mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t \right\|^2 + \frac{\mu\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \tilde{\Delta}_i^t \right\|^2 \\
&= - \left(\frac{\eta\tau}{2} - \frac{\eta\alpha}{2} \right) \mathbb{E} \left\| \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 + \frac{\eta\tau}{2}\mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t - \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 \\
&\quad + \underbrace{\left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) \mathbb{E} \left\| \tilde{\Delta}^t - \tilde{\Delta}_{\mathcal{B}^t} \right\|^2}_{T_4} - \frac{\eta\tau}{2}\mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t \right\|^2 + \frac{\mu\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \tilde{\Delta}_i^t \right\|^2. \tag{21}
\end{aligned}$$

T_4 can be decomposed as

$$T_4 = \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) \mathbb{E} \left\| \tilde{\Delta}^t - \tilde{\Delta}_{\mathcal{B}^t} \right\|^2 \leq \kappa \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) \tag{22}$$

where the first inequality holds as LASA is κ -robust aggregation rule with κ .

Plugging 22 back to 21, we have

$$\begin{aligned}
\mathcal{L}_{\mathcal{B}}(\theta^{t+1}) - \mathcal{L}_{\mathcal{B}}(\theta^t) &\leq - \left(\frac{\eta\tau}{2} - \frac{\eta\alpha}{2} \right) \mathbb{E} \left\| \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 + \frac{\eta\tau}{2}\mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t - \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 \\
&\quad + \kappa \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) - \frac{\eta\tau}{2}\mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t \right\|^2 + \frac{\mu\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \tilde{\Delta}_i^t \right\|^2 \\
&= - \left(\frac{\eta\tau}{2} - \frac{\eta\alpha}{2} \right) \mathbb{E} \left\| \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 + \frac{\eta\tau}{2}\mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t - \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 \\
&\quad + \kappa \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) + \mu\eta^2 \underbrace{\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \tilde{\Delta}_i^t \right\|^2}_{T_5} - \frac{\eta\tau}{2}\mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t \right\|^2 \tag{23}
\end{aligned}$$

T_5 can be characterized as

$$\begin{aligned}
T_5 &= \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \tilde{\Delta}_i^t \right\|^2 = \frac{\tau^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \|d_i^t\|^2 = \frac{\tau^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(\mathbb{E} \|d_i^t - h_i^t\|^2 + \mathbb{E} \|h_i^t\|^2 \right) \\
&= \frac{\tau^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(\mathbb{E} \left\| \frac{1}{\tau} \sum_{s=0}^{\tau-1} (g_i^{t,s} - \nabla \mathcal{L}_i(\theta_i^{t,s})) \right\|^2 + \mathbb{E} \|h_i^t\|^2 \right) \\
&\leq \frac{\tau^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(\frac{1}{\tau} \sum_{s=0}^{\tau-1} \mathbb{E} \|g_i^{t,s} - \nabla \mathcal{L}_i(\theta_i^{t,s})\|^2 + \mathbb{E} \|h_i^t\|^2 \right) \\
&\leq \frac{\tau^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(\frac{1}{\tau} \sum_{s=0}^{\tau-1} \nu_i^2 + \mathbb{E} \|h_i^t\|^2 \right) \\
&= \frac{\tau^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(\nu_i^2 + \mathbb{E} \|h_i^t\|^2 \right), \tag{24}
\end{aligned}$$

where the first inequality follows Lemma 4 and the second inequality follows Assumption 2.

Plugging 24 back to 23, we have

$$\begin{aligned}
\mathcal{L}_{\mathcal{B}}(\theta^{t+1}) - \mathcal{L}_{\mathcal{B}}(\theta^t) &\leq - \left(\frac{\eta\tau}{2} - \frac{\eta\alpha}{2} \right) \mathbb{E} \|\nabla \mathcal{L}_{\mathcal{B}}(\theta^t)\|^2 + \frac{\eta\tau}{2} \mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t - \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 - \frac{\eta\tau}{2} \mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t \right\|^2 \\
&\quad + \frac{\mu\eta^2\tau^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} (\nu_i^2 + \mathbb{E} \|h_i^t\|^2) + \kappa \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) \\
&= - \left(\frac{\eta\tau}{2} - \frac{\eta\alpha}{2} \right) \mathbb{E} \|\nabla \mathcal{L}_{\mathcal{B}}(\theta^t)\|^2 + \frac{\eta^2\tau}{2} \mathbb{E} \left\| \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} h_i^t - \nabla \mathcal{L}_{\mathcal{B}}(\theta^t) \right\|^2 + \frac{\mu\eta^2\tau^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \|h_i^t\|^2 \\
&\quad + \mu\eta\tau^2\bar{\nu}^2 + \kappa \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) \tag{25} \\
&\leq - \left(\frac{\eta\tau}{2} - \frac{\eta\alpha}{2} \right) \mathbb{E} \|\nabla \mathcal{L}_{\mathcal{B}}(\theta^t)\|^2 + \frac{\eta\tau}{2} \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \frac{1}{\tau} \sum_{r=0}^{\tau-1} \mathbb{E} \|\nabla \mathcal{L}_i(\theta_i^{t,r}) - \nabla \mathcal{L}_i(\theta^t)\|^2 \\
&\quad + \frac{\mu\eta^2\tau^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(2\mathbb{E} \|h_i^t - \nabla \mathcal{L}_i(\theta^t)\|^2 + \frac{2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \|\nabla \mathcal{L}_i(\theta^t)\|^2 \right) + \mu\eta^2\tau^2\bar{\nu}^2 + \kappa \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) \\
&\leq - \left(\frac{\eta\tau}{2} - \frac{\eta\alpha}{2} \right) \mathbb{E} \|\nabla \mathcal{L}_{\mathcal{B}}(\theta^t)\|^2 + \frac{\eta\tau}{2} \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \frac{\mu^2}{\tau} \sum_{r=0}^{\tau-1} \mathbb{E} \|\theta_i^{t,r} - \theta^t\|^2 + \frac{\mu\eta^2\tau^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \frac{\mu^2}{\tau} \sum_{r=0}^{\tau-1} 2\mathbb{E} \|\theta_i^{t,r} - \theta^t\|^2 \\
&\quad + \frac{4\mu\eta^2\tau^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} (\bar{\zeta} + \|\nabla \mathcal{L}_{\mathcal{B}}(\theta^t)\|^2) + \mu\eta^2\tau^2\bar{\nu}^2 + \kappa \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) \\
&= \left[- \left(\frac{\eta\tau}{2} - \frac{\eta\alpha}{2} \right) + 4\mu\eta^2\tau^2 \right] \mathbb{E} \|\nabla \mathcal{L}_{\mathcal{B}}(\theta^t)\|^2 + \left(\frac{\eta\mu^2}{2} + 2\eta^2\tau\mu^3 \right) \underbrace{\sum_{r=0}^{\tau-1} \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \|\theta_i^{t,r} - \theta^t\|^2}_{T_6} \\
&\quad + 4\mu\eta^2\tau^2\bar{\zeta} + \mu\eta^2\tau^2\bar{\nu}^2 + \kappa \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) \tag{26}
\end{aligned}$$

where the second inequality follows Lemma 3 and the third inequality follow Assumption 1.

Now we treat T_6 as

$$\begin{aligned}
T_6 &= \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \theta_i^{t,r} - \theta^t \right\|^2 = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \theta_i^{t,r-1} - \theta^t - \eta g_i^{t,s-1} \right\|^2 \\
&= \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \theta_i^{t,r-1} - \theta^t - \eta g_i^{t,s-1} + \eta \nabla \mathcal{L}_i(\theta^{t,s-1}) - \eta \nabla \mathcal{L}_i(\theta^{t,s-1}) + \eta \nabla \mathcal{L}_i(\theta^t) - \eta \nabla \mathcal{L}_i(\theta^t) \right\|^2 \\
&= \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \theta_i^{t,r-1} - \theta^t - \eta \nabla \mathcal{L}_i(\theta^{t,s-1}) + \eta \nabla \mathcal{L}_i(\theta^t) - \eta \nabla \mathcal{L}_i(\theta^t) \right\|^2 + \frac{\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| g_i^{t,s-1} - \nabla \mathcal{L}_i(\theta^{t,s-1}) \right\|^2 \\
&\leq \left(1 + \frac{1}{2\tau - 1}\right) \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \theta_i^{t,r-1} - \theta^t \right\|^2 + \frac{2\tau\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left\| \nabla \mathcal{L}_i(\theta^{t,s-1}) + \nabla \mathcal{L}_i(\theta^t) - \nabla \mathcal{L}_i(\theta^t) \right\|^2 + \eta^2 \bar{\nu} \\
&\leq \left(1 + \frac{1}{2\tau - 1}\right) \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \theta_i^{t,r-1} - \theta^t \right\|^2 + \frac{4\tau\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left\| \nabla \mathcal{L}_i(\theta^{t,s-1}) - \nabla \mathcal{L}_i(\theta^t) \right\|^2 + \frac{4\tau\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left\| \nabla \mathcal{L}_i(\theta^t) \right\|^2 + \eta^2 \bar{\nu} \\
&\leq \left(1 + \frac{1}{2\tau - 1}\right) \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \theta_i^{t,r-1} - \theta^t \right\|^2 + \frac{4\tau\mu^2\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left\| \theta_i^{t,r-1} - \theta^t \right\|^2 + \frac{4\tau\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left\| \nabla \mathcal{L}_i(\theta^t) - \nabla \mathcal{L}_B(\theta^t) + \nabla \mathcal{L}_B(\theta^t) \right\|^2 + \eta^2 \bar{\nu} \\
&\leq \left(1 + \frac{1}{2\tau - 1} + 4\tau\mu^2\eta^2\right) \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \theta_i^{t,r-1} - \theta^t \right\|^2 + \frac{8\tau\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left\| \nabla \mathcal{L}_B(\theta^t) \right\|^2 + 8\tau\bar{\zeta}\eta^2 + \eta^2 \bar{\nu} \\
&\leq \left(1 + \frac{1}{\tau - 1}\right) \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \theta_i^{t,r-1} - \theta^t \right\|^2 + \frac{8\tau\eta^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left\| \nabla \mathcal{L}_B(\theta^t) \right\|^2 + 8\tau\bar{\zeta}\eta^2 + \eta^2 \bar{\nu}, \tag{27}
\end{aligned}$$

where the first and second inequality follows Lemma 3 with $\delta = 2\tau$ and $\delta = 1$, respectively. The third inequality follows Assumption 1, and the last inequality holds if $\eta \leq 1/3\tau\mu$. Consequently, we have

$$\begin{aligned}
T_6 &= \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbb{E} \left\| \theta_i^{t,r} - \theta^t \right\|^2 \leq \sum_{h=0}^{s-1} \left(1 + \frac{1}{\tau - 1}\right)^h \left[8\tau\eta^2 \left\| \nabla \mathcal{L}_B(\theta^t) \right\|^2 + 8\tau\bar{\zeta}\eta^2 + \eta^2 \bar{\nu} \right] \\
&\leq (\tau - 1) \left[\left(1 + \frac{1}{\tau - 1}\right)^\tau - 1 \right] \times \left[8\tau\eta^2 \left\| \nabla \mathcal{L}_B(\theta^t) \right\|^2 + 8\tau\bar{\zeta}\eta^2 + \eta^2 \bar{\nu} \right] \\
&\leq 32\tau^2\eta^2 \left\| \nabla \mathcal{L}_B(\theta^t) \right\|^2 + 32\tau^2\bar{\zeta}\eta^2 + 4\tau\eta^2\bar{\nu}, \tag{28}
\end{aligned}$$

where the last inequality results from the fact that $\left(1 + \frac{1}{\tau-1}\right)^t \leq 5$ when $\tau > 1$.

Plugging 28 back to 26, we have

$$\begin{aligned}
\mathcal{L}_B(\theta^{t+1}) - \mathcal{L}_B(\theta^t) &\leq \left[-\left(\frac{\eta\tau}{2} - \frac{\eta\alpha}{2}\right) + 4\mu\eta^2\tau^2 \right] \mathbb{E} \left\| \nabla \mathcal{L}_B(\theta^t) \right\|^2 + 4\mu\eta^2\tau^2\bar{\zeta} + \mu\eta^2\tau^2\bar{\nu}^2 + \kappa \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) \\
&\quad + \left(\frac{\eta\mu^2}{2} + 2\eta^2\tau\mu^3 \right) \sum_{r=0}^{\tau-1} \left[32\tau^2\eta^2 \left\| \nabla \mathcal{L}_B(\theta^t) \right\|^2 + 32\tau^2\bar{\zeta}\eta^2 + 4\tau\eta^2\bar{\nu} \right] \\
&= \left[-\left(\frac{\eta\tau}{2} - \frac{\eta\alpha}{2}\right) + 4\mu\eta^2\tau^2 \right] \mathbb{E} \left\| \nabla \mathcal{L}_B(\theta^t) \right\|^2 + 4\mu\eta^2\tau^2\bar{\zeta} + \mu\eta^2\tau^2\bar{\nu}^2 + \kappa \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) \\
&\quad + \left(\frac{\eta\mu^2}{2} + 2\eta^2\tau\mu^3 \right) \left[32\tau^3\eta^2 \left\| \nabla \mathcal{L}_B(\theta^t) \right\|^2 + 32\tau^3\bar{\zeta}\eta^2 + 4\tau^2\eta^2\bar{\nu} \right] \\
&= \left[\left[-\left(\frac{\eta\tau}{2} - \frac{\eta\alpha}{2}\right) + 4\mu\eta^2\tau^2 \right] + \left(\frac{\eta\mu^2}{2} + 2\eta^2\tau\mu^3 \right) (32\eta^2\tau^3) \right] \mathbb{E} \left\| \nabla \mathcal{L}_B(\theta^t) \right\|^2 \\
&\quad + \left(\frac{\eta\mu^2}{2} + 2\eta^2\tau\mu^3 \right) (32\tau^3\bar{\zeta}\eta^2 + 4\tau^2\eta^2\bar{\nu}) + 4\mu\eta^2\tau^2\bar{\zeta} + \mu\eta^2\tau^2\bar{\nu} + \kappa \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) \\
&\leq -\eta \mathbb{E} \left\| \nabla \mathcal{L}_B(\theta^t) \right\|^2 + \left(\frac{\eta\mu^2}{2} + 2\eta^2\tau\mu^3 \right) (32\tau^3\bar{\zeta}\eta^2 + 4\tau^2\eta^2\bar{\nu}) + 4\mu\eta^2\tau^2\bar{\zeta} + \mu\eta^2\tau^2\bar{\nu} + \kappa \left(\mu\eta^2 + \frac{\eta}{2\alpha} \right) \\
&\leq -\eta \mathbb{E} \left\| \nabla \mathcal{L}_B(\theta^t) \right\|^2 + \kappa \left(\mu\eta^2 + \frac{\eta}{4} \right) + 7\eta\tau\bar{\zeta} + (1 + \tau)\eta\bar{\nu} \tag{29}
\end{aligned}$$

where the second inequality holds with $\alpha \geq 2$, and $\eta \leq 1/3\mu\tau$.

Times $1/\eta$ to the both sides of 29, rearranging and summing it from $t = 0$ to $t = T - 1$ and dividing by T , one yields

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla \mathcal{L}_B(\theta^t)\|^2 \leq \frac{(\mathcal{L}_B(\theta^0) - \mathcal{L}_B(\theta^*))}{T\eta} + \kappa(\mu\eta + 1) + 7\tau\bar{\zeta} + (1 + \tau)\bar{\nu}.$$

Assume $\tilde{\theta}$ is uniformly sampled from the sequence of outputs $\{\theta^0, \theta^1, \dots, \theta^T\}$ generated by FL with LASA as the F , then we have

$$\mathbb{E} \left\| \nabla \mathcal{L}_B(\tilde{\theta}) \right\|^2 = \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla \mathcal{L}_B(\theta^t)\|^2,$$

which concludes the proof. \square