ROBUST QUANTITY-AWARE AGGREGATION FOR FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) enables multiple clients to collaboratively train models without sharing their local data, It becomes an important privacy-preserving machine learning framework. However, classical FL faces serious security and robustness problems, e.g., malicious clients can poison model updates and at the same time claim large quantities to amplify the impact of their model updates in the model aggregation. Existing defense methods for FL, while all handling malicious model updates, either treat all quantities benign or simply ignore/truncate the quantities of all clients. The former is vulnerable to quantity-enhanced attack, while the latter leads to sub-optimal performance since the local data on different clients is usually in significantly different sizes. In this paper, we propose a robust quantity-aware aggregation algorithm for federated learning, called FedRA, to perform the aggregation with awareness of local data quantities while being able to defend against quantity-enhanced attacks. More specifically, we propose a method to filter out malicious clients by jointly considering uploaded model updates and data quantities from different clients and perform quantity-aware weighted averaging on model updates from remaining clients. Moreover, as the number of malicious clients participating in the federated learning may dynamically change in different rounds, we also propose a malicious client number estimator to predict how many suspicious clients should be filtered out in each round. Experiments on four public datasets demonstrate the effectiveness of our FedRA method in defending FL against quantity-enhanced attacks. Our code is available at `https://anonymous.4open.science/r/FedRA-4C1E`.

1 INTRODUCTION

Federated learning (FL) is a technology to train models while protecting the privacy of training data. It has been widely studied for many application scenarios, such as medical health (Rieke et al., 2020; Sheller et al., 2020) and keyboard next-word prediction (Yang et al., 2018; Hard et al., 2018). One of the classic FL algorithms is FedAvg (McMahan et al., 2017). In FedAvg, the server iteratively averages clients' updates with some weights determined by the *quantity* of each client, which means throughout the paper the number of the training data at that client, to update the global model.

The linear aggregation applied in FedAvg has been proved to be vulnerable to poisoning attacks (Baruch et al., 2019; Fang et al., 2020; Wang et al., 2020; Xie et al., 2020). However, these attacks only focus on generating malicious updates to degrade the performance of the global model or injecting a backdoor to the global model. Malicious clients can also submit large quantities to obtain unfairly high weights in the model aggregation, resulting in an amplified impact of malicious updates on the global model. We name this kind of attacks **quantity-enhanced attacks**.

Several methods have been proposed to defend against poisoning attacks for federated learning (Blanchard et al., 2017; Yin et al., 2018; El Mhamdi et al., 2018; Sun et al., 2019; Pillutla et al., 2019; Portnoy et al., 2020). These defenses can be divided into three groups: quantity-ignorant, quantity-aware, and quantity-robust defenses. **Quantity-ignorant** defenses aggregate updates without considering quantities (Blanchard et al., 2017; Yin et al., 2018; El Mhamdi et al., 2018). These methods are robust to quantity-enhanced attacks. However, since aggregating updates with quantities benefits model performance (Zaheer et al., 2018; Reddi et al., 2021), applying these defenses may lead to performance degradation (Appendix C.1). **Quantity-aware** defenses aggregate updates

with quantities but by default treat quantities submitted by clients as benign (Sun et al., 2019; Pillutla et al., 2019). These defenses usually outperform quantity-ignorant defenses when without attacks. However, their performance degrades severely when quantity-enhanced attacks occur. **Quantity-robust** defenses aggregate updates with quantities and, unlike quantity-aware defenses, are robust to quantity-enhanced attacks. Portnoy et al. (2020) propose to truncate quantities with a dynamic threshold and apply quantity-aware TriMean (Yin et al., 2018) to aggregate updates. However, it does not filter out malicious updates by jointly considering quantities and updates and may truncate benign quantities, which leads to sub-optimal performance.

Meanwhile, some existing defenses (Blanchard et al., 2017; Yin et al., 2018; El Mhamdi et al., 2018) need a hyper-parameter that represents the upper bound of the number of malicious clients to be filtered out in each round. However, in cross-device federated learning, the server samples only a group of clients to participate in training in each round due to the large number of all clients. As a result, the number of malicious clients in each round changes dynamically. Over-estimating the number of malicious clients will lead to some benign clients being filtered out, while under-estimating the number of malicious clients makes some malicious updates selected in model aggregation.

In this paper, we propose a quantity-robust defense, called *FedRA*, for federated learning. It filters out malicious clients by taking both quantities and updates into consideration. More specifically, FedRA first computes the L^1-distance between each pair of updates. Since the variance of benign local updates are usually small when the quantities of these clients are large, the expectation of distance between benign updates with larger quantities should be smaller. Based on this observation, we multiply the distance between each pair of updates with a coefficient relevant to their quantities, which is defined as quantity-robust distance. Then, a group of updates with the smallest quantity-robust distance to their neighbors are selected. FedRA aggregates them with weights proportional to their quantities. Finally, to tackle the problem of the dynamic number of malicious clients in each round in cross-device federated learning, we propose a malicious client number estimator to dynamically determine the number of malicious clients in each round.

The main contributions of this paper are as follows:

- We propose a robust quantity-aware aggregation method for federated learning to aggregate updates with quantities while defending against quantity-enhanced attacks.
- We theoretically prove FedRA is quantity-robust by proving the aggregation error of FedRA is irrelevant to malicious quantities.
- We propose to dynamically estimate the number of malicious clients in each round and empirically show that it can handle the over-estimating and under-estimating problem.
- We conduct experiments on four public datasets to validate the effectiveness of FedRA.

2 Related Works

Federated learning (McMahan et al., 2017) enables multiple clients collaboratively train models without sharing their local datasets. There are three steps in each round of federated learning. First, a central server randomly samples a group of clients and distributes the global model to them. Second, the selected clients train the model with their local datasets and upload their model updates to the central server. Finally, the central server aggregates the received updates to update the global model. In FedAvg (McMahan et al., 2017), updates are weight-averaged according to the quantity of each client's training samples. The above steps are performed iteratively until the global model converges.

However, the classical federated learning is vulnerable to poisoning attacks, e.g., untargeted attacks (Baruch et al., 2019; Fang et al., 2020) and backdoor attacks (Liu et al., 2017; Bagdasaryan et al., 2020; Wang et al., 2020; Xie et al., 2020; Bhagoji et al., 2019). In this paper, we focus on defending against untargeted attacks, which aim to degrade the performance of the global model on all input samples. To defend against the attacks, several robust aggregation methods have been proposed. Yin et al. (2018) propose Median and Trimean that apply coordinate-wise median and trimmed-mean, respectively, to filter out malicious updates. Blanchard et al. (2017) propose Krum and mKrum that compute square-distance-based scores to select and average the updates closest to a subset of neighboring updates. Bulyan (El Mhamdi et al., 2018) is a combination of mKrum and Trimean: it first selects several updates through mKrum and then aggregates them with Trimean. We

note that the above defense methods do not consider quantities of clients' training samples, which we categorize as *quantity-ignorant* methods, and the convergence speeds and model performance of these methods are compromised (Zaheer et al., 2018; Reddi et al., 2021), especially for long-tailed data distributions (see Appendix C.1) that are common in real-world scenarios (Zhang et al., 2017; Zhong et al., 2019; Li et al., 2017).

Sun et al. (2019) propose Norm-bound that clips the L^2 norm of received updates to a predefined threshold. Pillutla et al. (2019) propose RFA that computes weights for each update by running an approximation algorithm to minimize the quantity-aware geometric median of updates. These two methods are categorized into *quantity-aware* defenses. They take quantities into consideration when aggregating updates but by default treat all received quantities as benign. Portnoy et al. (2020) point out that received quantities may be malicious and can be exploited to increase the impact of malicious updates. They further propose a Truncate method that truncates received quantities within a dynamic threshold in each round, which guarantees any 10% clients do not have more than 50% samples. The Truncate method is categorized into *quantity-robust* method. However, quantities of benign clients with a large number of training samples may also be truncated, resulting in degraded performance. Meanwhile, they handle the malicious update filtering and the quantity truncation separately, leading to sub-optimal update filtering.

3 Problem Definition

Suppose that training samples are sampled from a distribution $\mathcal{D}$ in sample space $\mathcal{Z}$. Let $f(\mathbf{w}; z)$ denote the loss function of model parameter $\mathbf{w} \in \mathcal{W}$ at data point z, and $F(\mathbf{w}) = \mathbb{E}_{z\sim\mathcal{D}}[f(\mathbf{w}; z)]$ is the corresponding population loss function. The goal is to minimize the population loss by training the model parameter, i.e., $\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathcal{W}} F(\mathbf{w})$.

Assume that there are N clients in total and M of them are malicious. The i-th client has a local dataset D_i, where any $z \in D_i$ is independently sampled from distribution $\mathcal{D}$. The empirical loss of the i-th client is $F_i(\mathbf{w}) = \frac{1}{||\mathcal{D}_i||}\sum_{z\in\mathcal{D}_i} f(\mathbf{w}; z)$. In the t-th round, the central server randomly samples n clients and distributes the global model $\mathbf{w}_t$ to them. To simplify the theoretical analysis, we follow FedSGD (McMahan et al., 2017) and Trimean (Yin et al., 2018) with the following assumption: a benign client will submit update $\mathbf{g}_t^i = \nabla F_i(\mathbf{w}_t)$ and quantity $q^i = ||\mathcal{D}_i||$, while a malicious client can submit an arbitrary update and an arbitrary quantity to the server. After receiving the updates and quantities from the sampled n clients, the server computes the global update with a certain aggregation rule $\mathcal{A}$: $\mathbf{g}_{t+1} = \mathcal{A}(\mathbf{g}_t^1, ..., \mathbf{g}_t^n, q^1, ..., q^n)$.

Some existing defenses, e.g., mKrum (Blanchard et al., 2017), Bulyan (El Mhamdi et al., 2018), and Trimean (Yin et al., 2018), need to estimate the number of malicious clients m in each round with a fixed parameter for all rounds. However, in cross-device federated learning, the server samples a group of clients in each round due to the large number of clients. The number of malicious clients m in each round follows a hypergeometric distribution and is thus hard to be estimated by a fixed parameter. In our work, we consider two settings: fixed-ratio setting and dynamic-ratio setting. In the fixed-ratio setting, the number of malicious clients m in each round is fixed, i.e., $m = \lceil \frac{nM}{N} \rceil$. Since m is not a random variable, estimating m with a fixed parameter is feasible. Although the fixed-ratio setting is not aligned with the cross-device federated learning, we use this setting to analyze the upper bound capability of defenses to filter out malicious updates. In the dynamic-ratio setting, the overall number of malicious clients M is fixed, but the exact number of malicious clients in each round is unknown, which is more aligned with real-world federating learning scenarios.

Definition 1 *(Sub-exponential random variable). A random variable X with $\mathbb{E}[X] = \mu$ is called sub-exponential with parameters (v^2, α) if $\mathbb{E}[e^{\lambda(X-\mu)}] \leq e^{\frac{1}{2}v^2\lambda^2}$, $\forall|\lambda| < \frac{1}{\alpha}$.*

Definition 2 *(Lipschitz). h is L-Lipschitz if $|h(\boldsymbol{w}) - h(\boldsymbol{w}')| \leq L||\boldsymbol{w} - \boldsymbol{w}'||_2$, $\forall\, \boldsymbol{w}, \boldsymbol{w}'$.*

Definition 3 *(Smoothness). h is L-smooth if $|\nabla h(\boldsymbol{w}) - \nabla h(\boldsymbol{w}')| \leq L||\boldsymbol{w} - \boldsymbol{w}'||_2$, $\forall\, \boldsymbol{w}, \boldsymbol{w}'$.*

4 BASE AGGREGATION

In this section, we introduce L^1-based Krum, the quantity-ignorant base aggregation of our FedRA. Then we introduce L^1-based QKrum, which selects multiple clients with L^1-based Krum and applies non-uniform aggregation. We show that L^1-based QKrum is a quantity-aware aggregation.

4.1 L1-BASED KRUM

The L^1-based Krum changes the squared L^2 distance in Krum (Blanchard et al., 2017) to L^1 distance. More specifically, for the update of the i-th client $\mathbf{g}^i$, we first compute the L^1 distance between $\mathbf{g}^i$ and other $(n-1)$ updates. For any $j \neq i$, we denote $i \to j$ to indicate that $\mathbf{g}^j$ belongs to the $(n-m-2)$ updates closest to $\mathbf{g}^i$ in terms of the L^1 distance. Then we compute a score for $\mathbf{g}^i$, denoted as $s(i) = \sum_{j:i\to j} ||\mathbf{g}^i - \mathbf{g}^j||_1$. Finally, L^1-Krum$(\mathbf{g}^1, ..., \mathbf{g}^n) = \mathbf{g}^{i_*}$, where i_* refers to the client with the minimum score, $s(i_*) \leqslant s(i),\ \forall\ i$.

Assumption 1 *For all $z \in \mathcal{Z}$ and $\boldsymbol{w} \in \mathcal{W}$, $\partial_k f(\boldsymbol{w}; z)$ is sub-exponential with parameters (v_k^2, α_k) where $\mathbb{E}[\partial_k f(\boldsymbol{w}; z)] = \mu_k, \mathrm{Var}(\partial_k f(\boldsymbol{w}; z)) = \sigma_k^2, v_k = \sigma_k$, and $\alpha_k < \frac{\sigma_k}{\sqrt{2\ln 2n}}$.*

Proposition 1 *Let $\boldsymbol{g}^1$, ..., $\boldsymbol{g}^{n-m}$ be independently identically distributed updates with the same quantity q, where $\boldsymbol{g}^i \in \mathbb{R}^d$, and $\mathbb{E}[\boldsymbol{g}^i] = \boldsymbol{\mu}$. Let $\boldsymbol{g}^{n-m+1}$, ..., $\boldsymbol{g}^n$ be any random vectors in $\mathbb{R}^d$. Suppose that Assumption 1 holds for all benign updates. If $2m + 2 < n$ and define*

$$\zeta(n, m) \underset{def}{=} \frac{3\sqrt{2\ln 2}(n-m) + (n-2)\sqrt{2\ln 2(n-m)}}{(n-2m-2)\sqrt{q}} = O(\sqrt{\ln n}), \tag{1}$$

where $\boldsymbol{\sigma} \in \mathbb{R}^d$ is [σ_1, ..., σ_d], then the L^1-based Krum satisfies $\mathbb{E}\left[||\boldsymbol{g} - \boldsymbol{\mu}||_1\right] \leqslant \zeta(n, m)||\boldsymbol{\sigma}||_1$.

Proposition 1 is proved in Appendix A.1.

We state the statistical error guarantees of the L^1-based Krum for smooth non-convex F.

Assumption 2 *(Smoothness of f and F). For any $z \in \mathcal{Z}$, we assume that the partial derivative of $f(\cdot; z)$ with respect to the k-th dimension of its first argument, denoted as $\partial_k f(\cdot; z)$, is L_k-Lipschitz for each $k \in [1, d]$ and function $f(\cdot; z)$ is L-smooth. We also assume that $F(\cdot)$ is L_F-smooth. Let $\widehat{L} := (\sum_{k=1}^d L_k^2)^{\frac{1}{2}}$. It is obvious that $L_F \leqslant L \leqslant \widehat{L}$.*

Assumption 3 *(Minimizer in $\mathcal{W}$). Let $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}\in\mathcal{W}} F(\boldsymbol{w})$. We assume that $\nabla F(\boldsymbol{w}^*) = 0$.*

Theorem 1 *Assume that Assumptions 1, 2 and 3 hold, and $2m + 2 < n$. Choose $\eta = 1/L_F$. If $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}_t$, after T iterations with L^1-based Krum, we have*

$$\min_{t=0,...,T} \mathbb{E}\left[||\nabla F(\boldsymbol{w}_t)||_2^2\right] \leqslant \frac{2L_F}{T}\mathbb{E}[F(\boldsymbol{w}_0) - F(\boldsymbol{w}^*)] + (\zeta(n, m)||\boldsymbol{\sigma}||_1)^2. \tag{2}$$

Theorem 1 is proved in Appendix A.3.

4.2 L1-BASED QKRUM

The L^1-based Krum is a quantity-ignorant defense. We can easily improve the L^1-based Krum to a quantity-aware algorithm by selecting multiple clients and apply non-uniform aggregation. We name this algorithm L^1-based QKrum. However, L^1-based QKrum is not quantity-robust.

Lemma 1 *For any sample $z \in \mathcal{Z}$, suppose that $\mathbb{E}[\nabla f(\boldsymbol{w}; z)] = \boldsymbol{\mu}$ and $\mathbb{E}[(\partial_k f(\boldsymbol{w}; z) - \mu_k)^2] = \sigma_k^2$, then for the update of any benign client i, we have $\mathbb{E}[\boldsymbol{g}^i] = \boldsymbol{\mu}$. When Assumption 1 holds, $\mathbb{E}[\sqrt{q^i}||\boldsymbol{g}^i - \boldsymbol{\mu}||_1] \leq \sqrt{2\ln 2}||\boldsymbol{\sigma}||_1$.*

Lemma 1 is proved in Appendix A.4. It indicates that with a larger quantity, the variance of an update becomes smaller. Suppose $\sum_{i\in\mathcal{C}} q^i$ is a constant, where $\mathcal{C}$ is the set of selected clients. For

any benign client i, if he is selected, variance proportional to $\sqrt{q^i}\boldsymbol{\sigma}$ will be added, since $\mathbb{E}[q^i||\mathbf{g}^i - \boldsymbol{\mu}||_1] \leq \sqrt{2\ln 2}\sqrt{q^i}||\boldsymbol{\sigma}||_1$. It inspires us to analyze $\mathbb{E}[\sqrt{q^j}||\mathbf{g}^j - \boldsymbol{\mu}||_1]$ for client $j \in \mathcal{C}$.

We give an example to show the error of L^1-based QKrum can be an arbitrary value. We analyze the client i_* with the smallest selection score.

Example 1 *Suppose there are five benign updates $\boldsymbol{g}^1$, ..., $\boldsymbol{g}^5$ with the same quantities q_b, where $\boldsymbol{g}^i = [g^i]$ and $g^i \sim \mathcal{N}(0,1)$. Let $\boldsymbol{g}_m = [\epsilon]$ be the malicious update in $\mathbb{R}^1$ and q_m be the arbitrary quantity. Suppose $\exists\ \epsilon \neq 0$ such that the probability of $\boldsymbol{g}_b$ being selected $p_m \neq 0$. If Assumption 1 holds, when $q_m = (k/(p_m \cdot \epsilon))^2$, the error of L^1-based QKrum satisfies*

$$\mathbb{E}[||\sqrt{q}(\boldsymbol{g} - \boldsymbol{\mu})||_1] \geq k, \tag{3}$$

where k can be arbitrary positive value, $\boldsymbol{\mu} = [0]$, $\mathbf{g}$ and q are the updates and quantity of client i_*.

Example 1 is proved in Appendix A.5. It shows that the error of L^1-based QKrum can be an arbitrary value when facing the quantity-enhanced attack.

5 FedRA: Quantity-Robust Aggregation for Federated Learning

In this section, we introduce the details of our FedRA, which contains two core components, i.e. quantity-robust aggregation and malicious client number estimator. The quantity-robust aggregation improves the scores in L^1-based QKrum by jointly considering both updates and quantities. The malicious client number estimator dynamically determines the number of malicious clients in each round, which is more suitable for the dynamic-ratio setting. The complete algorithm of our FedRA is shown in Algorithm 1.

Algorithm 1 Robust Quantity-Aware Aggregation

Input: N, $\widetilde{M}$, n, $\{(\mathbf{g}_t^i, q_t^i)|i \in [n]\}$

$\mathcal{S} \leftarrow \varnothing$, $\widetilde{m} \leftarrow \lceil \frac{n\widetilde{M}}{N} \rceil$
for $i \in [n]$ **do**
  $s(i) \leftarrow (q_t^i)^\gamma \sum_{i \to j} \sqrt{\frac{q_t^i q_t^j}{q_t^i + q_t^j}} ||\mathbf{g}_t^i - \mathbf{g}_t^j||_1$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{s(i)\}$
end for
$$c \leftarrow \begin{cases} n - \widetilde{m} - 1 & \text{fixed-ratio,} \\ n - MCNE(n, \mathcal{S}) & \text{dynamic-ratio.} \end{cases}$$
selects c clients $\mathcal{C}$ with smallest scores in $\mathcal{S}$
$\mathbf{g}_t \leftarrow \frac{1}{\sum_{i \in \mathcal{C}} q_t^i} \sum_{i \in \mathcal{C}} q_t^i \mathbf{g}_t^i$

5.1 Quantity-Robust Aggregation

Since the variance of benign local updates are usually small when quantities of these clients are large, the expectation of distance between benign updates with larger quantities should be smaller. Based on this observation, in our FedRA, we design a quantity-robust score as follows

$$s(i) = (q^i)^\gamma \sum_{j:i \to j} Q(i,j),\ Q(i,j) = \sqrt{\frac{q^i q^j}{q^i + q^j}} ||\mathbf{g}^i - \mathbf{g}^j||_1, \tag{4}$$

where $i \to j$ denotes that $\mathbf{g}^j$ belongs to the $(n-m-2)$ updates closest to $\mathbf{g}^i$ in terms of the Q value, $\gamma \leqslant 0.5$ is a hyper-parameter. We analyze the error of the client i_* that has the smallest score.

Proposition 2 *Let $\boldsymbol{g}^1$, ..., $\boldsymbol{g}^{n-m}$ be benign updates with quantities q^1, ..., q^{n-m}, where $\boldsymbol{g}^i \in \mathbb{R}^d$. Let $\boldsymbol{g}^{n-m+1}$, ..., $\boldsymbol{g}^n$ be malicious updates in $\mathbb{R}^d$ and q^{n-m+1}, ..., q^n be their quantities. Suppose that Assumption 1 holds for all benign updates. If $2m + 2 \leqslant n$, $\gamma = 0.5$ and define*

$$\begin{aligned} \Delta_1 &= \frac{\sqrt{2\ln 2}(n-m) + m\sqrt{2\ln 2(n-m)}}{\sum_{i \in \mathcal{B}} q^i} \max_{\text{benign } i}[q^i + 1] \min_{\text{benign } i} \sqrt{q^i}||\boldsymbol{\sigma}||_1, \\ \overline{\boldsymbol{g}} &= \frac{1}{\sum\limits_{\substack{j:i_* \to j \\ \text{benign } j}} q^j} \sum_{\substack{j:i_* \to j \\ \text{benign } j}} q^j \boldsymbol{g}^j,\ \Delta = (\sqrt{2\ln 2(n-m)} + \sqrt{2\ln 2})||\boldsymbol{\sigma}||_1 + \Delta_1, \end{aligned} \tag{5}$$

then FedRA satisfies $\mathbb{E}[\sqrt{q}||\boldsymbol{g} - \overline{\boldsymbol{g}}||_1] \leqslant \Delta_1$. Denote $\mathcal{B}$ as the (n-m-2) benign clients with smallest quantities. If $\max\limits_{\text{malicious } i} q^i < \sum\limits_{i \in \mathcal{B}} q^i$, then $\mathbb{E}[\sqrt{q}||\boldsymbol{g} - \boldsymbol{\mu}||_1] \leqslant \Delta$.

Proposition 2 is proved in Appendix B.1, which can be easily extended to scenarios where multiple clients are selected. Comparing with the L^1-based QKrum, the error of our FedRA is not controlled by malicious quantity. Thus FedRA is quantity-robust.

We state the statistical error guarantees of FedRA for smooth non-convex F.

Theorem 2 *Assume that Assumptions 1, 2 and 3 hold,* $2m + 2 < n$, $\gamma = 0.5$ *and* $\max_{\textit{malicious } i} q^i < \sum_{i \in \mathcal{B}} q^i$. *Choose* $\eta = 1/L_F$. *If* $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}_t$, *after* T *iterations with FedRA, we have*

$$\min_{t=0,...,T} \mathbb{E}[||\nabla F(\boldsymbol{w}_t)||_2^2] \leq \frac{2L_F}{T}\mathbb{E}[F(\boldsymbol{w}_0) - F(\boldsymbol{w}^*)] + \frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{q_t}\Delta^2, \tag{6}$$

where q_t is the quantity of the selected clients at the t-th round.

Theorem 2 is proved in Appendix B.3.

5.2 Malicious Client Number Estimator

The above quantity-robust aggregation algorithm needs a parameter c to decide how many updates to be selected. In the fixed-ratio setting, we can set $c = n - \widetilde{m} - 1 = n - \lceil \frac{n\widetilde{M}}{N} \rceil - 1$, where $\widetilde{m}$ is the estimated number of malicious clients in each round and $\widetilde{M}$ is the estimated number of overall malicious clients. However, the number of malicious clients changes in different rounds in the dynamic-ratio setting. Over-estimating $\widetilde{m}$ will lead to some benign clients being filtered, while $\widetilde{m}$ makes some malicious updates selected in model aggregation. Therefore, we propose a malicious client number estimator to predict the number of malicious clients in each round. Our malicious client estimator computes the number of malicious clients $\widetilde{m}$ by maximizing the log-likelihood as follows:

Algorithm 2 Malicious Client Number Estimator

Input: $n, \{s(i)|i \in [n]\}$
Output: $\widetilde{m}$

$\mathcal{L} \leftarrow \varnothing$
for $i = 0, 1, ..., n$ **do**
  $\widetilde{m} \leftarrow i$
  Estimate μ_b, σ_b, μ_m, σ_m by Eq 9
  Compute $\hat{l}(i)$ through Eq 8
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{\hat{l}(i)\}$
end for
$\widetilde{m} \leftarrow \arg\max_{\widetilde{m}} \mathcal{L}$

$$\hat{l}(\widetilde{m}) = \ln p(\widetilde{m}, s(1), ..., s(n)) = \ln p(\widetilde{m}) + \sum_{i=1}^{n} \ln p(s(i)|\widetilde{m}). \tag{7}$$

We assume the scores of benign and malicious clients follows two independent Gaussian distributions, and malicious clients get the largest scores. In Appendix C.4, we show some score distributions in our experiments as the empirical evidence for the assumption. We first sort the scores by ascending order, i.e., $s(i) < s(j), \forall i < j$. Since m follows the hypergeometric distribution $\mathcal{H}(n, M, N)$, we have

$$\begin{aligned}\hat{l}(\widetilde{m}) \propto \ln \tbinom{\widetilde{M}}{\widetilde{m}} \tbinom{N-\widetilde{M}}{n-\widetilde{m}} - (n-\widetilde{m})\ln\sigma_b - \widetilde{m}\ln\sigma_m \\ - \sum_{i=1}^{n-\widetilde{m}} \frac{(s(i)-\mu_b)^2}{2\sigma_b^2} - \sum_{i=n-\widetilde{m}+1}^{n} \frac{(s(i)-\mu_m)^2}{2\sigma_m^2},\end{aligned} \tag{8}$$

where μ_b and σ_b^2 are the mean and variance of benign scores, and μ_m and σ_m^2 are those of malicious scores. The mean and variance of the two Gaussian distributions are estimated as follows:

$$\begin{aligned}\mu_b = \frac{1}{n-\widetilde{m}} \sum_{i=1}^{n-\widetilde{m}} s(i), \sigma_b^2 = \frac{1}{n-\widetilde{m}-1} \sum_{i=1}^{n-\widetilde{m}} (s(i)-\mu_b)^2 \\ \mu_m = \frac{1}{\widetilde{m}} \sum_{i=n-\widetilde{m}+1}^{n} s(i), \sigma_m^2 = \frac{1}{\widetilde{m}-1} \sum_{i=n-\widetilde{m}+1}^{n} (s(i)-\mu_m)^2.\end{aligned} \tag{9}$$

Table 1: Dataset description and statistics.

Dataset	Task	#Classes	#Train	#Test	#Clients	#Train per client		
						Mean	Std	Max
MNIST	Image Classification	10	60,000	10,000	3,025	19.77	179.28	6,820
CIFAR10	Image Classification	10	50,000	10,000	3,115	16.05	200.14	8,933
Adult	Income Prediction	2	32,561	16,281	1,671	19.49	114.19	2,403
MIND	Text Classification	18	71,068	20,307	2,880	24.68	299.61	9,398

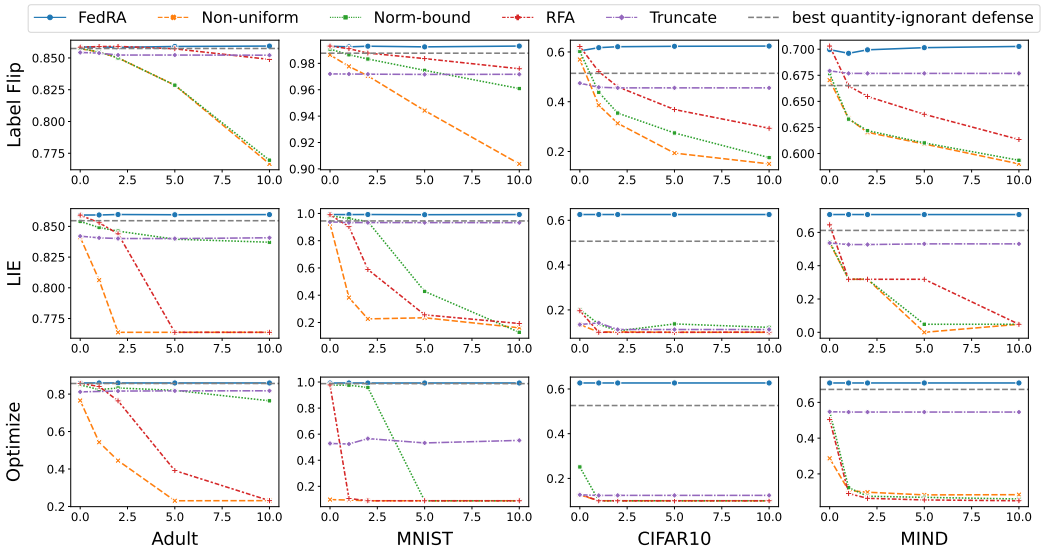

Figure 1: Performance of defenses in the fixed-ratio IID setting. The y-axes are the values of accuracy. The x-axes are the values of quantity-enlarging factors.

One may ask how $s(i)$ is computed since it is relevant with $\widetilde{m}$. A reasonable way is to initialize $\widetilde{m} = \lceil \frac{n\widetilde{M}}{N} \rceil$, iteratively compute $s(i)$ and run the malicious client number estimator to update $\widetilde{m}$. However, our experiments indicate that the performance without iterative approximation is already great enough. The algorithm of our malicious client number estimator is summarized in Algorithm 2.

6 Experiments

6.1 Datasets and Experimental Settings

Dataset. We conduct experiments on four public datasets: MNIST (LeCun et al., 1998), CIFAR10 (Krizhevsky et al., 2009), Adult (Dua & Graff, 2017), and MIND (Wu et al., 2020). The quantities follow a long-tailed distribution, i.e., log-normal distribution. The average sample size of clients is around 20, and the σ of the log-normal distribution is 3. We randomly shuffle the dataset and partition it according to the quantities. The details of the datasets are shown in Table 1.

Configurations. In our experiments, we use CNN networks as base models for the MNIST and CIFAR10 datasets. For the Adult dataset, we apply a three-layer feed-forward network as the base model. For the MIND dataset, we use a Text-CNN as the base model, and initialize the word embedding matrix with pre-trained Glove embeddings (Pennington et al., 2014). We apply FedAdam (Reddi et al., 2021) to accelerate model convergence in all methods. We apply dropout with dropout rate 0.2 to mitigate over-fitting. The learning rate is 0.001 for CIFAR10 and Adult and 0.0001 for MNIST and MIND. The maximum of training rounds is 10,000 for MNIST and CIFAR10, 2,000 for Adult, and 15,000 for MIND. The ratio of malicious clients M/N is 0.1. The number of clients sampled in each round n is 50. γ is 0.1. The server estimates $\widetilde{M}$ as M.

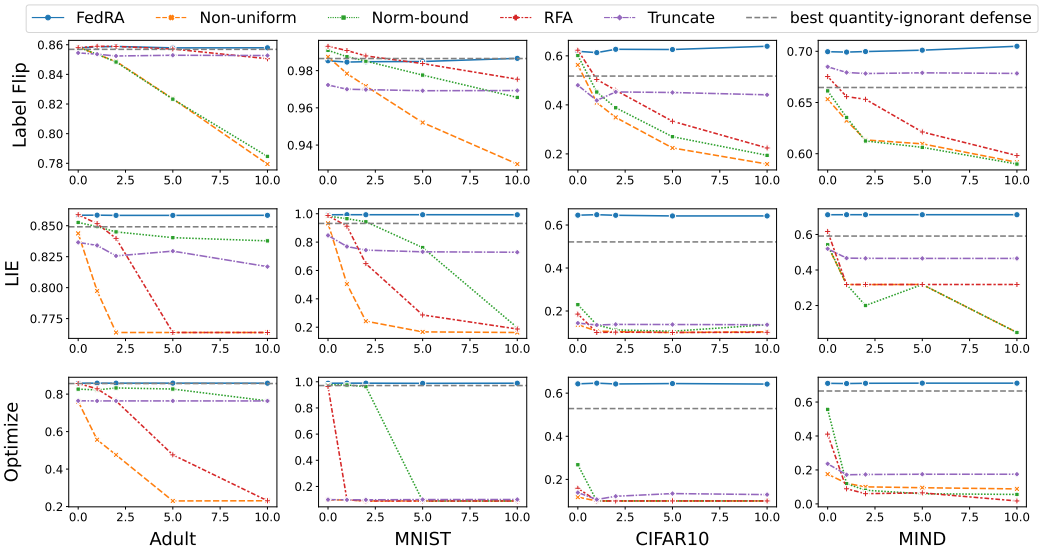

Figure 2: Performance of defenses in the dynamic-ratio setting. The y-axes are the values of accuracy. The x-axes are the values of quantity-enlarging factors.

Baselines. We compare our FedRA with several baseline methods, including 1) Median (Yin et al., 2018), applying coordinate-wise median on each dimension of updates; 2) Trimean (Yin et al., 2018), applying coordinate-wise trimmed-mean on each dimension of updates; 3) Krum (Blanchard et al., 2017), selecting the update that is closest to a subset of neighboring updates based on the square distance; 4) mKrum (Blanchard et al., 2017), a variance of Krum that selects multiple updates and averages the selected updates; 5) Bulyan (El Mhamdi et al., 2018), selecting multiple clients with mKrum and aggregating the selected updates with Trimean; 6) Norm-bounding (Sun et al., 2019), clipping the L_2 norm of each update with a certain threshold; 7) RFA (Pillutla et al., 2019), applying an approximation algorithm to minimize the geometric median of updates; 8) Truncate (Portnoy et al., 2020), limiting the quantity of each client under a dynamic threshold in each round and applying quantity-aware Trimean.

Attack Model. We suppose an attacker controls malicious clients. Each malicious client, if sampled, submits malicious updates and a malicious quantity. We implement three existing untargeted poisoning attack methods to create malicious updates, including 1) Label Flip Fang et al. (2020): a data poisoning attack that manipulates labels of training samples; 2) LIE Baruch et al. (2019): adding small enough noise in updates to circumvent defenses; 3) Optimize Fang et al. (2020): a model poisoning attack that adds noise in the opposite position of benign updates. To create malicious quantities, the attacker first computes the mean and variance of benign quantities, which are denoted as μ_q^m and σ_q^m, respectively. The malicious quantity is calculated as $q = \mu_q^m + \alpha_q \sigma_q^m$, where $\alpha_q \in \{0, 1, 2, 5, 10\}$ is the quantity-enlarging factor.

6.2 Performance Evaluation in Fixed-ratio Setting

In this subsection, we conduct experiments in the fixed-ratio setting to analyze the effectiveness of quantity-robust aggragtion. The experimental results are shown in Figure 1. We can make the following observations from the figure. First, our FedRA outperforms the best quantity-ignorant defense in the fixed-ratio settings. This is because our method performs weighted averaging on selected updates based on their quantities. Second, our FedRA has stable performance with different quantity-enlarging factors. This is because FedRA can defend against quantity-enhanced attacks by jointly considering updates and quantities to filter malicious clients. These two observations reflect the effectiveness of our FedRA algorithm. Third, the performance of quantity-aware defenses, i.e., RFA and Norm-bound, becomes worse with larger quantity-enlarging factors. This is because these quantity-aware defenses by default treat received quantities as benign, which is vulnerable to the quantity-enhanced attack. Finally, Truncate has stable performance with different quantity-enlarging

factors, but its performance is lower than FedRA. This is because the Truncate algorithm is quantity-robust by limiting quantities submitted by malicious clients. However, it also restricts quantities of benign clients. Meanwhile, it does not filter malicious clients by jointly considering quantities and updates. Thus, it has sub-optimal performance.

6.3 Performance Evaluation in Dynamic-ratio Setting

In this subsection, we conduct experiments in the dynamic-ratio setting to analyze the effectiveness of our malicious client number selector. The experimental results are shown in Figure 2. Besides the same observations in the fixed-ratio setting, we can make several additional observations. First, our FedRA has stable performance with different quantity-enlarging factors. It outperforms or has similar performance as the best quantity-ignorant defense. This shows the effectiveness of our FedRA with the malicious client estimator. Second, the algorithms that need to estimate the number or the upper bound of malicious clients, i.e., mKrum, Trimean, Bulyan, and Truncate, have lower performance in the dynamic-ratio setting than in the fixed-ratio setting. This is because the number of malicious clients in each rounds changes dynamically. Over-estimating makes a subset of benign clients excluded, while under-estimating causes some malicious clients selected in some rounds.

6.4 Ablation Study on Malicious Client Number Estimation

In this subsection, we compare applying the malicious client number estimator (MCNE) with under-estimating and over-estimating the number of malicious clients $\widetilde{m}$. For the experiments of under-estimation , we set the estimated number of malicious clients $\widetilde{m}$ as 5, which equals the expectation of selected malicious clients in each round. For the experiments of over-estimation, we set the estimated number of malicious clients $\widetilde{m}$ as 15, since $p(m > 15) \leqslant 1.54^{-5}$. The experimental results are shown in Figure 3. The performance of experiments with MCNE is consistently higher than those with over-estimation and under-estimation. This is because the number of malicious clients follows a hypergeometric distribution in the dynamic-ratio setting, which is hard to be estimated by a fixed parameter. Under-estimating the number of malicious clients makes some malicious clients selected in some rounds, while over-estimation filters some benign clients in most rounds.

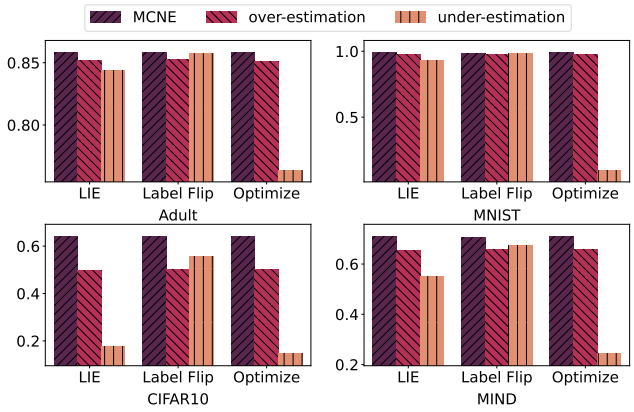

Figure 3: Results of malicious client number estimator, under-estimating and over-estimating the number of malicious client in the dynamic-ratio setting.

7 Conclusion

In this paper, we propose a robust quantity-aware aggregation method for federated learning, called FedRA. It aims to aggregate clients' local model updates with awareness of clients' quantities to benefit model performance while being quantity-robust to defend against quantity-enhanced attacks. FedRA filters out malicious clients by jointly considering uploaded model updates and quantities from different clients and perform quantity-aware weighted averaging on model updates from remaining clients. Since the number of malicious clients varies in different rounds, we further design a malicious client number estimator to determine the number of clients to be selected in each round. Experiments on four public datasets demonstrate the effectiveness of our robust quantity-aware aggregation for federated learning.

REFERENCES

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *AISTATS*, pp. 2938–2948, 2020.

Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *NIPS*, volume 32, 2019.

Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *ICML*, pp. 634–643, 2019.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *NIPS*, 30, 2017.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in Byzantium. In *ICML*, volume 80, pp. 3521–3530, 2018.

Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. In *USENIX*, 2020.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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.

Jingjing Li, Ke Lu, Zi Huang, and Heng Tao Shen. Two birds one stone: On both cold-start and long-tail recommendation. In *MM*, pp. 898–906, 2017.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *NDSS*, 2017.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pp. 1273–1282, 2017.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pp. 1532–1543, 2014.

Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *arXiv preprint arXiv:1912.13445*, 2019.

Amit Portnoy, Yoav Tirosh, and Danny Hendler. Towards federated learning with byzantine-robust client weighting. *arXiv preprint arXiv:2004.04986*, 2020.

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *ICLR*, 2021.

Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):1–7, 2020.

Micah J Sheller, Brandon Edwards, G Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R Colen, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):1–12, 2020.

Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy yong Sohn, Kangwook Lee, and Dimitris S. Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In *NIPS*, 2020.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. MIND: A large-scale dataset for news recommendation. In *ACL*, pp. 3597–3606, 2020.

Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *ICLR*, 2020.

Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *ICML*, pp. 5650–5659, 2018.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. *NIPS*, 31, 2018.

Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *ICCV*, pp. 5419–5428, Oct 2017.

Yaoyao Zhong, Weihong Deng, Mei Wang, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Unequal-training for deep face recognition with long-tailed noisy data. In *CVPR*, pp. 7804–7813, June 2019.

SUPPLEMENTARY MATERIAL

A BASE AGGREGATION

A.1 PROOF OF PROPOSITION 1

Proof. Without loss of generality, we assume malicious vectors are placed in the last m positions in the arguments of the L^1-based Krum, i.e., $\mathbf{g} = L^1\text{-Krum}(\mathbf{g}^1, ..., \mathbf{g}^{n-m}, ..., \mathbf{g}^n)$. For each index i, we denote the number of benign indexes j such that $i \to j$ as $\delta_c(i)$, and the number of malicious indexes j such that $i \to j$ as $\delta_b(i)$. We have

$$\begin{aligned} \delta_c(i) + \delta_b(i) &= n - m - 2, \\ n - 2m - 2 \leq \delta_c(i) &\leq n - m - 2, \\ \delta_b(i) &\leq m. \end{aligned} \tag{1}$$

We denote the i_* as the index selected by L_1-Krum.

$$\begin{aligned} \mathbb{E}[||\mathbf{g} - \boldsymbol{\mu}||_1] \leq &\sum_{\text{benign } i} \mathbb{E}[||\mathbf{g}^i - \boldsymbol{\mu}||_1 \mathbb{I}(i_* = i)] \\ &+ \sum_{\text{malicious } k} \mathbb{E}[||\mathbf{g}^k - \boldsymbol{\mu}||_1 \mathbb{I}(i_* = k)], \end{aligned} \tag{2}$$

where $\mathbb{I}$ denotes the indicator function. We focus on the case that $i_* = i$ for some correct index i in Equation 2. We first prove the following lemma.

Lemma 2 *Let $\boldsymbol{g}^1$, ..., $\boldsymbol{g}^n$ be independent identically distributed random vectors with the same quantity q, where $\boldsymbol{g}^i \subseteq \mathbb{R}^d$, and $\mathbb{E}[\boldsymbol{g}^i] = \boldsymbol{\mu}$. Supposing that Assumption 1 holds, then we have*

$$\mathbb{E}[\max_i ||\boldsymbol{g}^i - \boldsymbol{\mu}||_1] \leq \sqrt{2 \ln 2n} ||\boldsymbol{\sigma}||_1 / \sqrt{q}, \tag{3}$$

where $\boldsymbol{\sigma}$ is a d-dimensional vector denoted as $[\sigma_1, ..., \sigma_d]$.

Proof. See Appendix A.2.

Applying Lemma 2 on the first term of Equation 2, we obtain

$$\begin{aligned} \sum_{\text{benign } i} \mathbb{E}[||\mathbf{g}^i - \boldsymbol{\mu}||_1 \mathbb{I}(i_* = i)] &\leq \mathbb{E}[\max_{\text{benign } i} ||\mathbf{g}^i - \boldsymbol{\mu}||_1] \\ &\leq \sqrt{2 \ln 2(n-m)} ||\boldsymbol{\sigma}||_1 / \sqrt{q}. \end{aligned} \tag{4}$$

Next we focus on the case that $i_* = k$ for some malicious index k in Equation 2.

$$\begin{aligned} &\sum_{\text{malicious } k} \mathbb{E}[||\mathbf{g}^k - \boldsymbol{\mu}||_1 \mathbb{I}(i_* = k)] \\ &\leq \sum_{\text{malicious } k} \mathbb{E}[||\boldsymbol{\mu} - \frac{1}{\delta_c(k)} \sum_{\substack{j: k \to j \\ \text{benign } j}} \mathbf{g}^j||_1 \mathbb{I}(i_* = k)] \\ &+ \sum_{\text{malicious } k} \mathbb{E}[||\mathbf{g}^k - \frac{1}{\delta_c(k)} \sum_{\substack{j: k \to j \\ \text{benign } j}} \mathbf{g}^j||_1 \mathbb{I}(i_* = k)] \\ &\leq \mathbb{E}[\max_{\text{malicious } k} \frac{1}{\delta_c(k)} \sum_{\text{benign } j} ||\boldsymbol{\mu} - \mathbf{g}^j||_1] \\ &+ \sum_{\text{malicious } k} \mathbb{E}[||\mathbf{g}^k - \frac{1}{\delta_c(k)} \sum_{\substack{j: k \to j \\ \text{benign } j}} \mathbf{g}^j||_1 \mathbb{I}(i_* = k)] \end{aligned} \tag{5}$$

If k is selected by the L^1-based Krum, it implies for any correct index i

$$\begin{aligned}&\sum_{\text{malicious } k} \mathbb{I}(i_* = k)[\sum_{\substack{j:k\to j\\ \text{benign } j}} ||\mathbf{g}^k - \mathbf{g}^j||_1 + \sum_{\substack{j:k\to j\\ \text{malicious } j}} ||\mathbf{g}^k - \mathbf{g}^j||_1] \\ &\leq \sum_{\substack{j:i\to j\\ \text{benign } j}} ||\mathbf{g}^i - \mathbf{g}^j||_1 + \sum_{\substack{j:i\to j\\ \text{malicious } j}} ||\mathbf{g}^i - \mathbf{g}^j||_1.\end{aligned} \tag{6}$$

Therefore, for any correct index i

$$\begin{aligned}&\sum_{\text{malicious } k} ||\mathbf{g}^k - \frac{1}{\delta_c(k)} \sum_{\substack{j:k\to j\\ \text{benign } j}} \mathbf{g}^j||_1 \mathbb{I}(i_* = k) \\ &\leq \sum_{\text{malicious } k} \frac{\mathbb{I}(i_* = k)}{\delta_c(k)} \sum_{\substack{j:k\to j\\ \text{benign } j}} ||\mathbf{g}^k - \mathbf{g}^j||_1 \\ &\leq \frac{1}{n-2m-2} \sum_{\substack{j:i\to j\\ \text{benign } j}} ||\mathbf{g}^i - \mathbf{g}^j||_1 + \frac{1}{n-2m-2} \sum_{\substack{j:i\to j\\ \text{malicious } j}} ||\mathbf{g}^i - \mathbf{g}^j||_1.\end{aligned} \tag{7}$$

We focus on the second term of Equation 7. Since any correct index i has $n-m-2$ neighbors and $m+1$ non-neighbors. There exists at least one benign index $\varsigma(i)$, which is farther from i than any of its neighbors. Therefore, $\forall j:\ i \to \text{malicious } j$, $||\mathbf{g}^i - \mathbf{g}^j||_1 \leq ||\mathbf{g}^i - \mathbf{g}^{\varsigma(i)}||_1$. Then we have

$$\begin{aligned}&\sum_{\text{malicious } k} ||\mathbf{g}^k - \frac{1}{\delta_c(k)} \sum_{\substack{j:k\to j\\ \text{benign } j}} \mathbf{g}^j||_1 \mathbb{I}(i_* = k) \\ &\leq \frac{1}{n-2m-2} \sum_{\substack{j:i\to j\\ \text{benign } j}} ||\mathbf{g}^i - \mathbf{g}^j||_1 + \frac{\delta_b(i)}{n-2m-2} ||\mathbf{g}^i - \mathbf{g}^{\varsigma(i)}||_1.\end{aligned} \tag{8}$$

$$\begin{aligned}&\sum_{\text{malicious } k} \mathbb{E}[||\mathbf{g}^k - \frac{1}{\delta_c(k)} \sum_{\substack{j:k\to j\\ \text{benign } j}} \mathbf{g}^j||_1 \mathbb{I}(i_* = k)] \\ &\leq \frac{2\sqrt{2\ln 2}(n-m)}{(n-2m-2)\sqrt{q}} ||\boldsymbol{\sigma}||_1 + \frac{\delta_b(i)}{n-2m-2} \sum_{\text{benign } j\neq i} \mathbb{E}[||\mathbf{g}^i - \mathbf{g}^j||_1 \mathbb{I}(\varsigma(i) = j)] \\ &\leq \frac{2\sqrt{2\ln 2}(n-m)}{(n-2m-2)\sqrt{q}} ||\boldsymbol{\sigma}||_1 + \frac{\delta_b(i)}{n-2m-2} \mathbb{E} \max_{\text{benign } j\neq i} ||\mathbf{g}^i - \mathbf{g}^j||_1 \\ &\leq \frac{2\sqrt{2\ln 2}(n-m) + 2m\sqrt{2\ln 2(n-m)}}{(n-2m-2)\sqrt{q}} ||\boldsymbol{\sigma}||_1.\end{aligned} \tag{9}$$

Putting Equation 9 back to Equation 5, we obtain

$$\sum_{\text{malicious } k} \mathbb{E}[||\mathbf{g}^k - \boldsymbol{\mu}||_1 \mathbb{I}(i_* = k)] \leq \frac{3\sqrt{2\ln 2}(n-m) + 2m\sqrt{2\ln 2(n-m)}}{(n-2m-2)\sqrt{q}} ||\boldsymbol{\sigma}||_1 \tag{10}$$

Putting everything back together, we get

$$\mathbb{E}[||\mathbf{g} - \boldsymbol{\mu}||_1] \leq \frac{3\sqrt{2\ln 2}(n-m) + (n-2)\sqrt{2\ln 2(n-m)}}{(n-2m-2)\sqrt{q}} ||\boldsymbol{\sigma}||_1. \tag{11}$$

A.2 Proof of Lemma 2

Proof. We first convert the problem of computing the expectation of the maximum of the L^1 norm of the d-dimensional vectors into the problem of computing expectations of the maximum of each

dimension of the d-dimensional vectors.

$$\begin{aligned}\mathbb{E}[\max_i ||\mathbf{g}^i - \boldsymbol{\mu}||_1] &= \mathbb{E}[\max_i \sum_{k\in\{1,...,d\}} |g_k^i - \mu_k|] \\ &\leq \sum_{k\in\{1,...,d\}} \mathbb{E}[\max_i |g_k^i - \mu_k|].\end{aligned} \tag{12}$$

Following the same logic in Appendix A.4, it is easy to prove g_k^i is sub-exponential with parameters $(\frac{v_k^2}{q}, \frac{\alpha_k}{q})$, where $v_k = \sigma_k$ and $\alpha_k \leq \frac{\sigma_k}{\sqrt{2\ln 2n}}$. Define $x_k^i = g_k^i - \mu_k$. Denote a list of values $X = \{x_k^1, -x_k^1, ..., x_k^n, -x_k^n\}$, and $z_k = \max_i |g_k^i - \mu_k| = \max_{x\in X} x$. We then obtain

$$e^{\lambda\mathbb{E}[z_k]} \leq \mathbb{E}[e^{\lambda z_k}] = \mathbb{E}[\max_{x\in X} e^{\lambda x}] \leq \sum_{x\in X} \mathbb{E}[e^{\lambda x}] \leq 2ne^{\frac{\lambda^2\sigma_k^2}{2q}}. \tag{13}$$

$$\mathbb{E}[z_k] \leq \frac{\ln 2n}{\lambda} + \frac{\lambda\sigma_k^2}{2q}. \tag{14}$$

Setting $\lambda = \frac{\sqrt{2q\ln 2n}}{\sigma_k} < \frac{q}{\alpha_k}$, we can get

$$\mathbb{E}[z_k] \leq \sigma_k\sqrt{2\ln 2n}/\sqrt{q}. \tag{15}$$

Putting Equation 15 back to Equation 12, we obtain

$$\mathbb{E}[\max_i ||\mathbf{g}^i - \boldsymbol{\mu}||_1] \leq \sqrt{2\ln 2n}||\boldsymbol{\sigma}||_1/\sqrt{q}, \tag{16}$$

where $\boldsymbol{\sigma}$ is a d-dimensional vector denoted as $[\sigma_1, ..., \sigma_d]$.

A.3 Proof of Theorem 1

Proof. Following the proof of Theorem 2 in Yin et al. (2018), using the smoothness of $F(\cdot)$ and setting $\eta = 1/L_F$, we have

$$F(\mathbf{w}_{t+1}) \leq F(\mathbf{w}_t) - \frac{1}{2L_F}||\nabla F(\mathbf{w}_t)||_2^2 + \frac{1}{2L_F}||\mathbf{g}_t - \nabla F(\mathbf{w}_t)||_2^2. \tag{17}$$

According to Proposition 1, we further obtain

$$\mathbb{E}[F(\mathbf{w}_{t+1}) - F(\mathbf{w}^*)] \leq \mathbb{E}[F(\mathbf{w}_t) - F(\mathbf{w}^*)] - \frac{1}{2L_F}\mathbb{E}[||\nabla F(\mathbf{w}_t)||_2^2] + \frac{1}{2L_F}(\zeta(n,m)||\boldsymbol{\sigma}||_1)^2. \tag{18}$$

Sum up Equation 18, we have

$$\begin{aligned}0 &\leq \mathbb{E}[F(\mathbf{w}_T) - F(\mathbf{w}^*)] \\ &\leq \mathbb{E}[F(\mathbf{w}^0) - F(\mathbf{w}^*)] - \frac{1}{2L_F}\sum_{t=0}^{T-1}\mathbb{E}[||\nabla F(\mathbf{w}_t)||_2^2] + \frac{T}{2L_F}(\zeta(n,m)||\boldsymbol{\sigma}||_1)^2,\end{aligned} \tag{19}$$

which implies

$$\min_{t=0,...,T}\mathbb{E}[||\nabla F(\mathbf{w}_t)||_2^2] \leq \frac{2L_F}{T}\mathbb{E}[F(\mathbf{w}_0) - F(\mathbf{w}^*)] + (\zeta(n,m)||\boldsymbol{\sigma}||_1)^2. \tag{20}$$

A.4 Proof of Lemma 1

Proof. As defined in Section 3, we have $g_k^i = \frac{1}{q^i}\sum_{z\in\mathcal{D}_i}\partial_k f(\mathbf{w}; z)$. Therefore, we can obtain

$$\mathbb{E}[g_k^i] = \frac{1}{q^i}\sum_{z\in\mathcal{D}_i}\mathbb{E}[\partial_k f(\mathbf{w}; z)] = \mu_k, \tag{21}$$

$$\begin{aligned}\mathbb{E}[(g_k^i-\mu_k))^2] &= \mathbb{E}[(\frac{1}{q^i}\sum_{z\in\mathcal{D}_i}\partial_k f(\mathbf{w};z)-\mu_k)^2] \\ &= \frac{1}{(q^i)^2}\sum_{z\in\mathcal{D}_i}\mathbb{E}[(\partial_k f(\mathbf{w};z)-\mu_k)^2] = \frac{\sigma_k^2}{q^i}.\end{aligned} \tag{22}$$

Since $\partial_k f(\mathbf{w};z)$ is sub-exponential with parameters (v_k^2,α_k), we have

$$e^{\frac{\lambda}{q^i}\sum_{z\in\mathcal{D}_i}(\partial_k f(\mathbf{w};z)-\mu_k)} \le e^{\frac{1}{2q^i}v^2\lambda^2}, \tag{23}$$

when $\lambda \le \frac{q^i}{\alpha_k}$. It shows g_k^i is sub-exponential with parameters $(\frac{v_k^2}{q^i},\frac{\alpha_k}{q^i})$, where $v_k=\sigma_k$ and $\alpha_k \le \frac{\sigma_k}{\sqrt{2\ln 2n}}$. n is the number of clients sampled by server in each round.

Follow the same logic in Lemma 2, we have

$$\mathbb{E}[|g_k^i-\mu_k|] \le \sqrt{2\ln 2}\frac{\sigma_k}{\sqrt{q^i}}. \tag{24}$$

Obviously, we can obtain

$$\mathbb{E}[\mathbf{g}^i]=\boldsymbol{\mu},\ \mathbb{E}[\sqrt{q^i}||\mathbf{g}^i-\boldsymbol{\mu}||_1] \le \sqrt{2\ln 2}||\boldsymbol{\sigma}||_1. \tag{25}$$

A.5 Proof of Example 1

Proof. Since benign updates are independently distributed and are with the same quantities q, the probability of each benign updates being selected is the same. Therefore, due to $\boldsymbol{\mu}=0$ we can have

$$\mathbb{E}[\sqrt{q}||\mathbf{g}-\boldsymbol{\mu}||_1] = \mathbb{E}[\sqrt{q_m}||\mathbf{g}_m-\boldsymbol{\mu}||_1]p_m + \mathbb{E}[\sqrt{q_b}||\mathbf{g}^1-\boldsymbol{\mu}||_1](1-p_m) \ge \sqrt{q_m}|\epsilon|p_m. \tag{26}$$

In L^1-based QKrum, since whether $\mathbf{g}_m$ is selected is only related to the distance between $\mathbf{g}_m$ and its neighbors, p_m is irrelevant with q_m. Therefore, we can treat p_m as a constant. When $q_m = (k/(p_m\cdot\epsilon))^2$, we have

$$\mathbb{E}[\sqrt{q}||\mathbf{g}-\boldsymbol{\mu}||_1] \ge k, \tag{27}$$

where k can be arbitrary positive value.

B Quantity-aware Robust Aggregation

B.1 Proof of Proposition 2

Proof. Similar to Appendix A.1, we analyze benign i_* and malicious i_* separately.

$$\begin{aligned}\mathbb{E}[\sqrt{q}||\mathbf{g}-\boldsymbol{\mu}||_1] \le &\sum_{\text{benign } i}\mathbb{E}[\sqrt{q^i}||\mathbf{g}^i-\boldsymbol{\mu}||_1\mathbb{I}(i_*=i)] \\ &+ \sum_{\text{malicious } k}\mathbb{E}[\sqrt{q^k}||\mathbf{g}^k-\boldsymbol{\mu}||_1\mathbb{I}(i_*=k)],\end{aligned} \tag{28}$$

where q and $\mathbf{g}$ are the quantity and update of selected client i_*.

When i_* is benign, according to Lemma 1 and Lemma 2, we have

$$\begin{aligned}\sum_{\text{benign } i}\mathbb{E}[\sqrt{q^i}||\mathbf{g}^i-\boldsymbol{\mu}||_1\mathbb{I}(i_*=i)] &\le \mathbb{E}[\max_{\text{benign } i}\sqrt{q^i}||\mathbf{g}^i-\boldsymbol{\mu}||_1] \\ &\le \sqrt{2\ln 2(n-m)}||\boldsymbol{\sigma}||_1.\end{aligned} \tag{29}$$

When i_* is malicious, the error can be formulated as

$$\begin{aligned}
&\sum_{\text{malicious } k} \mathbb{E}[\sqrt{q^k}||\mathbf{g}^k - \boldsymbol{\mu}||_1 \mathbb{I}(i_* = k)] \\
&\leq \underbrace{\sum_{\text{malicious } k} \mathbb{E}[\sqrt{q^k}||\mathbf{g}^k - \frac{1}{\sum\limits_{\substack{j:i_* \to j \\ \text{benign } j}} q^j} \sum_{\substack{j:i_* \to j \\ \text{benign } j}} q^j \mathbf{g}^j||_1 \mathbb{I}(i_* = k)]}_{\text{Term 1}} \\
&+ \underbrace{\sum_{\text{malicious } k} \mathbb{E}[\sqrt{q^k}||(\frac{1}{\sum\limits_{\substack{j:i_* \to j \\ \text{benign } j}} q^j} \sum_{\substack{j:i_* \to j \\ \text{benign } j}} q^j \mathbf{g}^j) - \boldsymbol{\mu}||_1 \mathbb{I}(i_* = k)]}_{\text{Term 2}}.
\end{aligned} \tag{30}$$

For Term 1, we have

$$\begin{aligned}
&\sum_{\text{malicious } k} \mathbb{E}[\sqrt{q^k}||\mathbf{g}^k - \frac{1}{\sum\limits_{\substack{j:i_* \to j \\ \text{benign } j}} q^j} \sum_{\substack{j:i_* \to j \\ \text{benign } j}} q^j \mathbf{g}^j||_1 \mathbb{I}(i_* = k)] \\
&\leq \sum_{\text{malicious } k} \mathbb{E}[\frac{\sqrt{q^k}}{\sum\limits_{\substack{j:i_* \to j \\ \text{benign } j}} q^j} \sum_{\substack{j:i_* \to j \\ \text{benign } j}} q^j ||\mathbf{g}^k - \mathbf{g}^j||_1 \mathbb{I}(i_* = k)]
\end{aligned} \tag{31}$$

Denote $\mathcal{B}$ as the (n-m-2) benign clients with smallest quantities. Since $\sqrt{\frac{q^k}{q^j(q^j+q^k)}} > \frac{1}{q^j+1}$, we have

$$\begin{aligned}
&\sum_{\text{malicious } k} \mathbb{E}[\sqrt{q^k}||\mathbf{g}^k - \frac{1}{\sum\limits_{\substack{j:i_* \to j \\ \text{benign } j}} q^j} \sum_{\substack{j:i_* \to j \\ \text{benign } j}} q^j \mathbf{g}^j||_1 \mathbb{I}(i_* = k)] \\
&\leq \frac{\max\limits_{\text{benign } i}[q^i + 1]}{\sum_{i \in \mathcal{B}} q^i} \sum_{\text{malicious } k} \mathbb{E}[\sqrt{q^k} \sum_{\substack{j:i_* \to j \\ \text{benign } j}} \sqrt{\frac{q^k q^j}{q^k + q^j}} ||\mathbf{g}^k - \mathbf{g}^j||_1 \mathbb{I}(i_* = k)].
\end{aligned} \tag{32}$$

If $i_* = k$ is selected by the FedRA, it implies for any correct index i

$$\begin{aligned}
&\sum_{\text{malicious } k} \sqrt{q^k}[\sum_{\substack{j:i_* \to j \\ \text{benign } j}} \sqrt{\frac{q^k q^j}{q^k + q^j}} ||\mathbf{g}^k - \mathbf{g}^j||_1 + \sum_{\substack{j:i_* \to j \\ \text{malicious } j}} \sqrt{\frac{q^k q^j}{q^k + q^j}} ||\mathbf{g}^k - \mathbf{g}^j||_1] \mathbb{I}(i_* = k) \\
&\leq \sqrt{q^i} \sum_{\substack{j:i \to j \\ \text{benign } j}} \sqrt{\frac{q^i q^j}{q^i + q^j}} ||\mathbf{g}^i - \mathbf{g}^j||_1 + \sqrt{q^i} \sum_{\substack{j:i \to j \\ \text{malicious } j}} \sqrt{\frac{q^i q^j}{q^i + q^j}} ||\mathbf{g}^i - \mathbf{g}^j||_1.
\end{aligned} \tag{33}$$

We focus on the second term of Equation 33. Since any correct index i has $n-m-2$ neighbors and $m+1$ non-neighbors. There exists at least one benign index $\varsigma(i)$, which has Q value score than any of its neighbors. Therefore, $\forall\ j : i \to$ malicious j, $\sqrt{\frac{q^i q^j}{q^i+q^j}} ||\mathbf{g}^i - \mathbf{g}^j||_1 \leq \sqrt{\frac{q^i q^{\varsigma(i)}}{q^i+q^{\varsigma(i)}}} ||\mathbf{g}^i - \mathbf{g}^{\varsigma(i)}||_1$. Then we have

$$\sqrt{q^i} \sum_{\substack{j:i \to j \\ \text{malicious } j}} \sqrt{\frac{q^i q^j}{q^i + q^j}} ||\mathbf{g}^i - \mathbf{g}^j||_1 \leq m\sqrt{q^i} \sqrt{\frac{q^i q^{\varsigma(i)}}{q^i + q^{\varsigma(i)}}} ||\mathbf{g}^i - \mathbf{g}^{\varsigma(i)}||_1. \tag{34}$$

Lemma 3 *Let $\boldsymbol{g}^i$ and $\boldsymbol{g}^j$ be any pair of independently distributed benign updates and q^i and q^j be the corresponding quantities. Suppose that Assumption 1 holds, we then have*

$$\mathbb{E}\left[||\boldsymbol{g}^i - \boldsymbol{g}^j||_1\right] \leq \sqrt{2\ln 2}\sqrt{\frac{q^i + q^j}{q^i q^j}}||\boldsymbol{\sigma}||_1. \tag{35}$$

Lemma 3 is proved in Appendix B.2. With Lemma 3, bringing Equation 34 and 33 back to Equation 32, we can obtain

$$\begin{aligned}&\sum_{\text{malicious } k} \mathbb{E}[\sqrt{q^k}||\mathbf{g}^k - \frac{1}{\sum\limits_{\substack{j:i_* \to j\\ \text{benign } j}} q^j} \sum_{\substack{j:i_* \to j\\ \text{benign } j}} q^j \mathbf{g}^j||_1 \mathbb{I}(i_* = k)] \\ &\leq \frac{\sqrt{2\ln 2}(n-m) + m\sqrt{2\ln 2(n-m)}}{\sum_{i\in\mathcal{B}} q^i} \max_{\text{benign } i}[q^i + 1] \min_{\text{benign } i} \sqrt{q_g^i}||\boldsymbol{\sigma}||_1.\end{aligned} \tag{36}$$

It is easy to prove the following conclusion with the same logic.

$$\begin{aligned}&\mathbb{E}[\sqrt{q}||\mathbf{g} - \frac{1}{\sum\limits_{\substack{j:i_* \to j\\ \text{benign } j}} q^j} \sum_{\substack{j:i_* \to j\\ \text{benign } j}} q^j \mathbf{g}^j||_1 \\ &\leq \frac{\sqrt{2\ln 2}(n-m) + m\sqrt{2\ln 2(n-m)}}{\sum_{i\in\mathcal{B}} q^i} \max_{\text{benign } i}[q^i + 1] \min_{\text{benign } i} \sqrt{q_g^i}||\boldsymbol{\sigma}||_1.\end{aligned} \tag{37}$$

For Term 2, according to Lemma 1 we have

$$\begin{aligned}&\sum_{\text{malicious } k} \mathbb{E}[\sqrt{q^k}||(\frac{1}{\sum\limits_{\substack{j:i_* \to j\\ \text{benign } j}} q^j} \sum_{\substack{j:i_* \to j\\ \text{benign } j}} q^j \mathbf{g}^j) - \boldsymbol{\mu}||_1 \mathbb{I}(i_* = k)] \\ &\leq \sqrt{\frac{2\ln 2}{\sum_{\substack{j:i_* \to j\\ \text{benign } j}} q^j}} \max_{\text{malicious } i}[\sqrt{q^i}]||\boldsymbol{\sigma}||_1 \leq \sqrt{\frac{2\ln 2}{\sum_{i\in\mathcal{B}} q^i}} \max_{\text{malicious } i}[\sqrt{q^i}]||\boldsymbol{\sigma}||_1\end{aligned} \tag{38}$$

Therefore, if all malicious quantities q^i satisfy $q^i \leq \sum_{i\in\mathcal{B}} q^i$, then we have

$$\sum_{\text{malicious } k} \mathbb{E}[\sqrt{q^k}||(\frac{1}{\sum\limits_{\substack{j:i_* \to j\\ \text{benign } j}} q^j} \sum_{\substack{j:i_* \to j\\ \text{benign } j}} q^j \mathbf{g}^j) - \boldsymbol{\mu}||_1 \mathbb{I}(i_* = k)] \leq \sqrt{2\ln 2}||\boldsymbol{\sigma}||_1. \tag{39}$$

Bring everything back to Equation 28, and define

$$\Delta_1 = \frac{\sqrt{2\ln 2}(n-m) + m\sqrt{2\ln 2(n-m)}}{\sum_{i\in\mathcal{B}} q^i} \max_{\text{benign } i}[q^i + 1] \min_{\text{benign } i} \sqrt{q_g^i}||\boldsymbol{\sigma}||_1. \tag{40}$$

We have

$$\mathbb{E}[\sqrt{q}||\mathbf{g} - \boldsymbol{\mu}||_1] \leq (\sqrt{2\ln 2(n-m)} + \sqrt{2\ln 2})||\boldsymbol{\sigma}||_1 + \Delta_1. \tag{41}$$

B.2 Proof of Lemma 3

Proof. Similar to Appendix A.2, we first convert the problem into computing expectations of each dimension.

$$\begin{aligned}\mathbb{E}[||\mathbf{g}^i - \mathbf{g}^j||_1] &= \mathbb{E}[\sum_{k\in\{1,\ldots,d\}} |g_k^i - g_k^j|] \\ &= \sum_{k\in\{1,\ldots,d\}} \mathbb{E}[|g_k^i - g_k^j|].\end{aligned} \tag{42}$$

Since any $z \in \mathcal{D}_i$ are independent, if $|\lambda| < \frac{q^i}{\alpha_k}$, we then have

$$\mathbb{E}[e^{\lambda(g_k^i - \mu_k)}] = \mathbb{E}[e^{\frac{\lambda}{q^i}\sum_{z\in\mathcal{D}_i}(\partial_k f(\mathbf{w};z)-\mu_k)}] \quad \leq e^{\frac{\lambda^2\sigma_k^2}{2q^i}}, \tag{43}$$

which implies g_k^i is sub-exponential with parameters $(\frac{\sigma_k^2}{q^i}, \frac{\alpha_k}{q^i})$, where $\alpha_k < \frac{\sigma_k}{\sqrt{2\ln 2n}}$.

Let $|\lambda| < \min(\frac{q^i}{\alpha_k}, \frac{q^j}{\alpha_k})$. Since g_k^i and g_k^j are independently distributed, we can obtain

$$\begin{aligned}
\mathbb{E}[e^{\lambda(g_k^i - g_k^j)}] &= \mathbb{E}[e^{\lambda((g_k^i-\mu_k)-(g_k^j-\mu_k))}] \\
&= \mathbb{E}[e^{\lambda(g_k^i-\mu_k)}]\mathbb{E}[e^{-\lambda(g_k^j-\mu_k)}] \\
&\leq e^{\frac{\lambda^2\sigma_k^2}{2}\frac{q^i+q^j}{q^iq^j}}.
\end{aligned} \tag{44}$$

Then following similar steps in Appendix A.2 and setting $\lambda = \frac{\sqrt{2\ln 2n}}{\sigma_k}\sqrt{\frac{q^iq^j}{q^i+q^j}}$, we can have

$$\mathbb{E}[||\mathbf{g}^i - \mathbf{g}^j||_1] \leq \sqrt{2\ln 2}||\boldsymbol{\sigma}||_1\sqrt{\frac{q^i+q^j}{q^iq^j}}. \tag{45}$$

B.3 Proof of Theorem 2

Proof. Using the smoothness of $F(\cdot)$, we have

$$\begin{aligned}
F(\mathbf{w}_{t+1}) &\leq F(\mathbf{w}_t) + \langle\nabla F(\mathbf{w}_t), \mathbf{w}_{t+1} - \mathbf{w}_t\rangle + \frac{L_F}{2}||\mathbf{w}_{t+1} - \mathbf{w}_t||_2^2 \\
&= F(\mathbf{w}_t) - \frac{\eta}{q_t}\langle\sqrt{q_t}\nabla F(\mathbf{w}_t), \sqrt{q_t}(\mathbf{g}_t - \nabla F(\mathbf{w}_t)) + \sqrt{q_t}\nabla F(\mathbf{w}_t)\rangle \\
&+ \frac{\eta^2 L_F}{2q_t}||\sqrt{q_t}(\mathbf{g}_t - \nabla F(\mathbf{w}_t)) + \sqrt{q_t}\nabla F(\mathbf{w}_t)||_2^2 \\
&= F(\mathbf{w}_t) + (\frac{\eta^2 L_F}{2} - \eta)||\nabla F(\mathbf{w}_t)||_2^2 + \frac{\eta^2 L_F}{2q_t}||\sqrt{q_t}(\mathbf{g}_t - \nabla F(\mathbf{w}_t))||_2^2 \\
&+ (\frac{\eta^2 L_F - \eta}{q_t})\langle\sqrt{q_t}\nabla F(\mathbf{w}_t), \sqrt{q_t}(\mathbf{g}_t - \nabla F(\mathbf{w}_t))\rangle.
\end{aligned} \tag{46}$$

Let $\eta = \frac{1}{L_F}$. According to Proposition 2, we further obtain

$$\mathbb{E}[F(\mathbf{w}_{t+1}) - F(\mathbf{w}^*)] \leq \mathbb{E}[F(\mathbf{w}_t) - F(\mathbf{w}^*)] - \frac{1}{2L_F}\mathbb{E}[||\nabla F(\mathbf{w}_t)||_2^2] + \frac{1}{2q_tL_F}\Delta^2. \tag{47}$$

Sum up Equation 47, we have

$$\begin{aligned}
0 &\leq \mathbb{E}[F(\mathbf{w}_T) - F(\mathbf{w}^*)] \\
&\leq \mathbb{E}[F(\mathbf{w}^0) - F(\mathbf{w}^*)] - \frac{1}{2L_F}\sum_{t=0}^{T-1}\mathbb{E}[||\nabla F(\mathbf{w}_t)||_2^2] + \frac{1}{2L_F}\sum_{t=0}^{T-1}\frac{1}{q_t}\Delta^2,
\end{aligned} \tag{48}$$

which implies

$$\min_{t=0,\ldots,T}\mathbb{E}[||\nabla F(\mathbf{w}_t)||_2^2] \leq \frac{2L_F}{T}\mathbb{E}[F(\mathbf{w}_0) - F(\mathbf{w}^*)] + \frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{q_t}\Delta^2. \tag{49}$$

C Additional Experiment

C.1 Importance of Quantity

In this subsection, we state the importance of quantity. We first analyze which factors influence the impact of quantities. We introduce non-uniform aggregation as performing weighted averaging

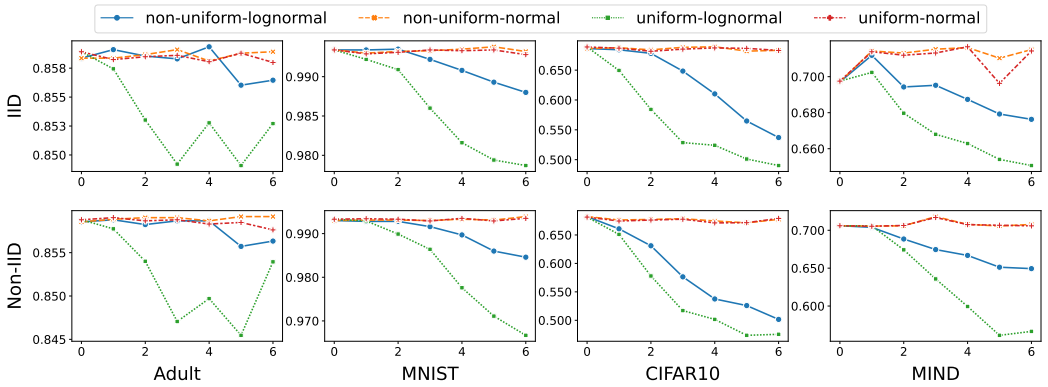

Figure 4: Performance of non-uniform and uniform aggregation in different data distributions. The x-axes are the values of accuracy. The y-axes are the values of σ in the Gaussian distributions or the log-normal distributions.

Table 2: Performance of different existing defense methods without attacks.

		Uniform	Median	Tmean	Krum	mKrum	Bulyan	Non-uniform	Norm-bound	RFA
IID	Adult	0.8492	0.8171	0.8548	0.8318	0.8517	0.8546	0.8583	0.8587	0.8581
	MNIST	0.9878	0.8744	0.9876	0.8656	0.9885	0.9881	0.9933	0.9935	0.9927
	CIFAR10	0.5322	0.3451	0.5515	0.5376	0.5351	0.5545	0.6488	0.6525	0.6454
	MIND	0.6700	0.6476	0.6825	0.6147	0.6712	0.6818	0.6948	0.6945	0.6927
non-IID	Adult	0.8471	0.7638	0.8512	0.7638	0.8521	0.8541	0.8587	0.8585	0.8569
	MNIST	0.9864	0.4478	0.9866	0.3470	0.9883	0.9872	0.9922	0.9915	0.9905
	CIFAR10	0.5224	0.1930	0.5397	0.2058	0.5335	0.5407	0.5771	0.5773	0.5547
	MIND	0.6293	0.5745	0.6105	0.5435	0.6386	0.6251	0.6815	0.6760	0.6761

on updates according to client quantities, while uniform aggregation as averaging updates without considering quantities. It is obvious that when all benign clients have same quantities, it is unnecessary to apply non-uniform aggregation. Meanwhile, we find non-uniform aggregation does not always significantly outperform uniform aggregation in some datasets (Table 3 on MIND [1], Table 5 on LEAF [2], and Table 4 on ML-1M [3]). We conduct experiments to study the impact of

- Skewness: We sample the clients' quantities from Gaussian distributions and log-normal distributions, respectively, and set the average quantity as 20.
- Variance: We vary the σ of the Gaussian distributions and the log-normal distributions. Higher σ means higher variance of clients' quantities.
- non-IID: For the IID setting, we randomly divide dataset into clients local datasets. For the non-IID setting, we guarantee the local datasets of most clients contain only one class.

The experimental results are shown in Figure 4, wherein we can make several observations. First, when client quantities are sampled from log-normal distributions, the performance difference between the uniform aggregation and the non-uniform aggregation gets more significant. It shows that skewness is one of the important factors. Second, under the log-normal settings, when we enlarge the variance of quantities, the performance difference becomes larger. It shows that variance is another important factor. Finally, comparing the IID and non-IID settings, the performance difference does not change, which shows that non-IID is not an important factor. Since the log-normal distribution is a log-tailed distribution, which is common in real-world scenarios and has been widely researched, we think it is necessary to study the robustness of federated learning algorithms under this setting.

[1] https://msnews.github.io/
[2] https://leaf.cmu.edu/
[3] https://grouplens.org/datasets/movielens/

Table 5: Performance on LEAF.

	FEMNIST	CelabA	Shakespeare	Reddit
Non-uniform	80.19	87.16	47.80	11.73
Uniform	80.01	87.52	47.97	11.68

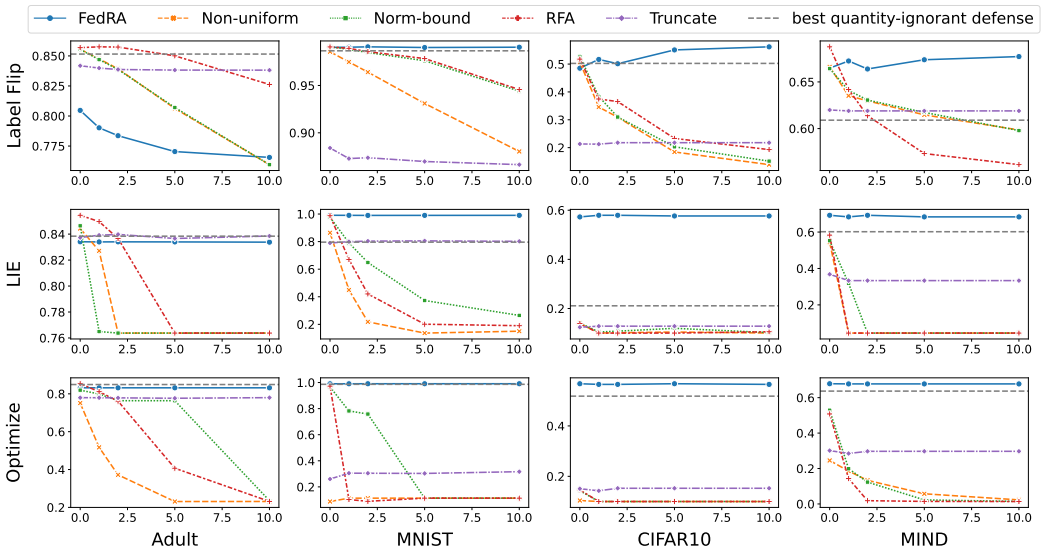

Figure 5: Performance of defenses in the fixed-ratio non-IID setting. The x-axes are the values of accuracy. The y-axes are the values of quantity-enlarging factors.

Table 3: Performance of recommendation on MIND.

	AUC	MRR	nDCG@5	nDCG@10
Non-uniform	66.56	31.40	34.69	41.05
Uniform	64.24	29.88	32.56	39.10

Table 4: Performance on ML-1M.

	Hit@10	nDCG@10
Non-uniform	67.12	38.32
Uniform	59.00	32.48

We further test the performance of different existing defense methods without attacks under the log-normal distribution with $\sigma = 3.0$. The experimental results are shown in Table 2. The quantity-ignored defenses have lower performance than the quantity-aware defenses on the four datasets, which shows the importance of quantity.

C.2 Performance Evaluation in Non-IID setting

In this subsection, we first conduct experiments in the fixed-ratio non-IID setting. The results are shown in Figure 5. We can observe that the performance of our FedRA is stable with quantity-enlarging factors except in Adult under Label Flip attacks. The performance of FedRA is higher than that of the best quantity-ignorant defense except for the Adult dataset. It might be because the defense in the Adult dataset is more sensitive with the non-IID setting. We then conduct experiments in the dynamic-ratio non-IID setting. The results are shown in Figure 6. The performance of our FedRA is stable with the quantity-enlarging factors and higher than that of the best quantity-ignorant defense except those with the Label-Flip attack. It might be because the quantity-robust scores may not follow two distant Gaussian distributions that we have assumed. Overall, our method does not have a theoretical guarantee for non-IID settings, and the above results empirically prove that our FedRA can have great performance in most of the non-IID settings.

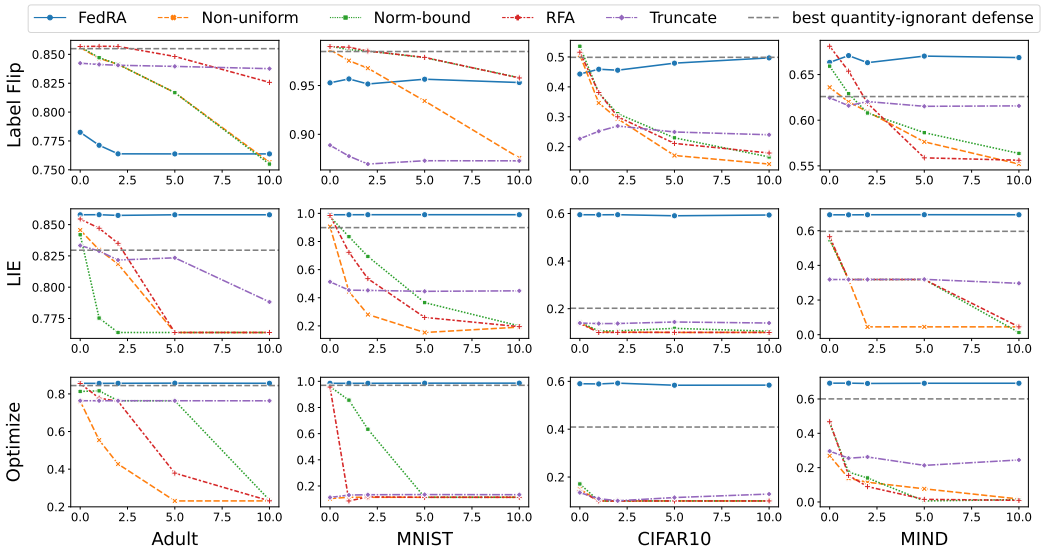

Figure 6: Performance of defenses in the dynamic-ratio non-IID setting. The x-axes are the values of accuracy. The y-axes are the values of quantity-enlarging factors.

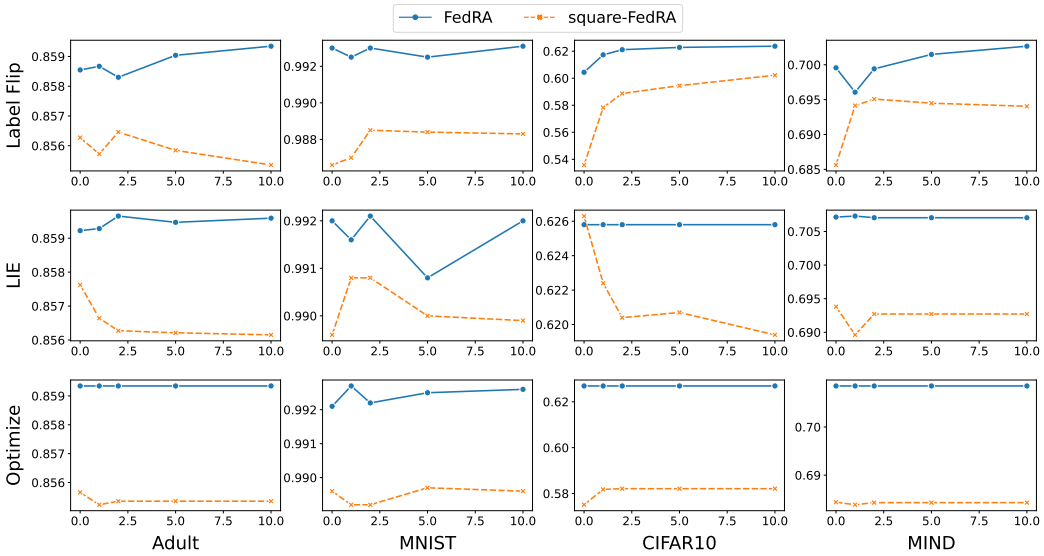

Figure 7: Performance of sqaure-FedRA and FedRA in the fixed-ratio IID setting. The x-axes are the values of accuracy. The y-axes are the values of quantity-enlarging factors.

C.3 Performance of Squared-L2-distance-based FedRA

In this subsection, we compare the performance of Squared-L2-distance-based FedRA (referred to as square-FedRA) with our L^1-based FedRA in the fixed-ratio IID setting. The experimental results are shown in Figure 7. Our L^1-based FedRA outperforms square-FedRA.

C.4 Distributions of Client Scores

In this subsection, we show some distributions of client scores in our experiments. The distributions support the assumptions in our malicious client number estimator that the scores of benign and malicious clients follow two independent Gaussian distributions, and malicious clients get the largest

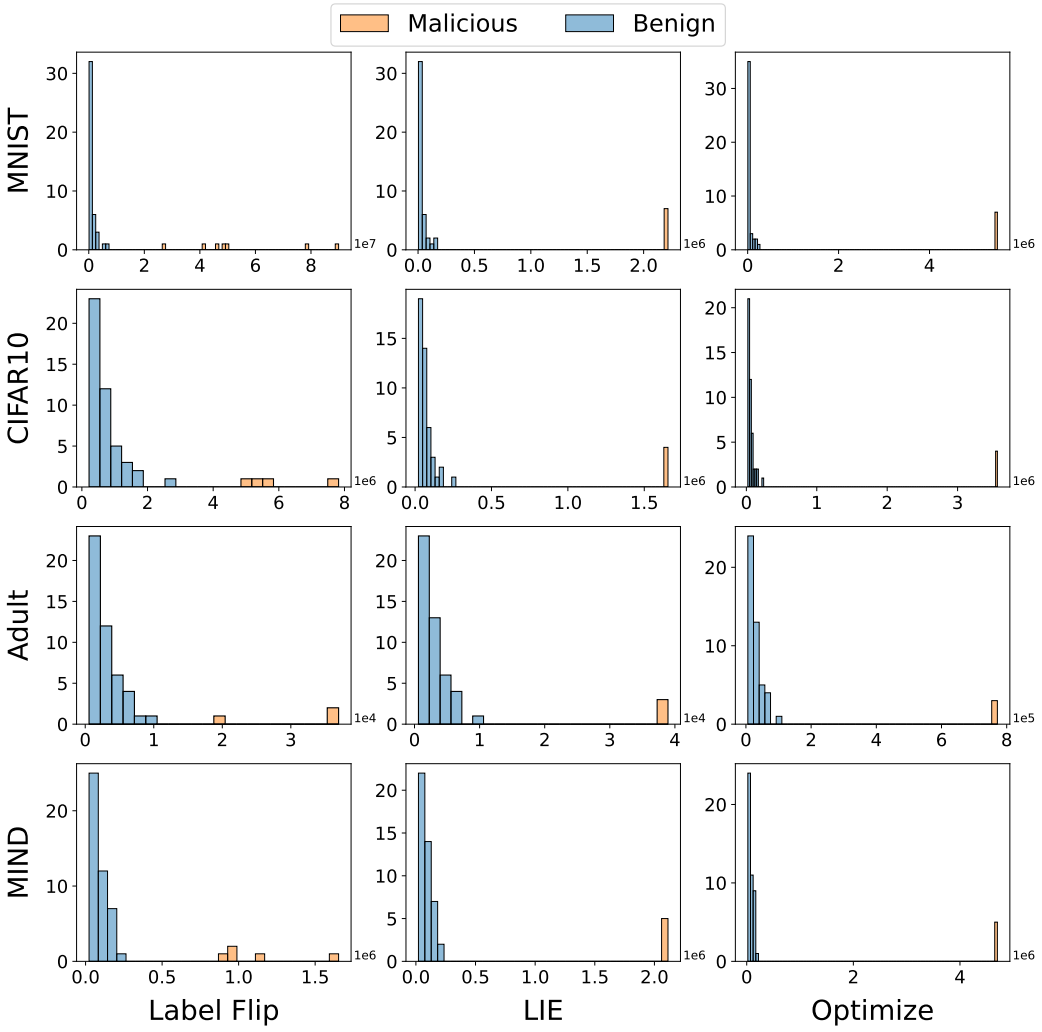

Figure 8: Distributions of client scores.

scores. In Figure 8, we show the distributions of client scores with quantity-enlarge factor $\alpha_q = 10$ at 200, 1000, 1000 and 1000 steps for Adult, MIND, MNIST and CIFAR10 dataset respectively.

D EXPERIMENTAL ENVIRONMENTS

We conduct experiments on a single V100 GPU with 32GB memory. The version of CUDA is 11.1. We use pytorch 1.9.1.