FEDQV: LEVERAGING QUADRATIC VOTING IN FEDERATED LEARNING

Tianyue Chu
IMDEA Networks Institute
Universidad Carlos III de Madrid

Nikolaos Laoutaris
IMDEA Networks Institute

ABSTRACT

Federated Learning (FL) permits different parties to collaboratively train a global model without disclosing their respective local labels. A crucial step of FL, that of aggregating local models to produce the global one, shares many similarities with public decision-making, and elections in particular. In that context, a major weakness of FL, namely its vulnerability to poisoning attacks, can be interpreted as a consequence of the *one person one vote* (henceforth *1p1v*) principle that underpins most contemporary aggregation rules.

In this paper, we introduce FEDQV, a novel aggregation algorithm built upon the *quadratic voting* scheme, recently proposed as a better alternative to *1p1v*-based elections. Our theoretical analysis establishes that FEDQV is a truthful mechanism in which bidding according to one's true valuation is a dominant strategy that achieves a convergence rate matching that of state-of-the-art methods. Furthermore, our empirical analysis using multiple real-world datasets validates the superior performance of FEDQV against poisoning attacks. It also shows that combining FEDQV with unequal voting "budgets" according to a reputation score increases its performance benefits even further. Finally, we show that FEDQV can be easily combined with Byzantine-robust privacy-preserving mechanisms to enhance its robustness against both poisoning and privacy attacks.

1 Introduction

Federated Learning (FL) has emerged as a promising privacy-preserving paradigm for conducting distributed collaborative model training across parties unwilling to disclose their local data. Parties collaborate to train a global model by submitting their local models to a server, which applies a specific aggregation rule to generate a new version of the global model to be sent back to parties. The process of agreeing on a common global model in Federated Learning shares many similarities with public decision-making and elections in particular. Indeed, the weights of local model updates from each party can be seen as votes of preference that affect the global model resulting from an aggregation rule applied at the centralised server of an FL group.

FEDAVG [1] has been the "de facto" aggregation rule used in FL tasks such as Google's emoji and next-word prediction for mobile device keyboards [2, 3]. In FEDAVG the global model is produced from a simple weighted averaging of local updates with weights that represent the amount of data that each party has used for its training.

The problem. Recent work [4] has shown that FEDAVG is vulnerable to poisoning attacks, as even a single attacker can degrade the global model by sharing faulty local updates of sufficiently large weight. Such attacks become possible because FEDAVG treats all local data points equally. In essence, the aggregation rule, when seen at the granularity of individual training data, resembles the *one person one vote (1p1v)* election rule of modern democratic elections. In this context, the server distributes votes (weights) to a party in accordance with the amount of its training data, which may be regarded as its population. This, however, may confer an unjust advantage to malicious parties who may have, or falsely claim to have, large training datasets.

Our approach. To address this issue, we propose a robust aggregation rule inspired by elections based on *Quadratic Voting* [5] (henceforth QV). In QV, each party is given a voting budget that can be spent on different rounds of voting for proposals. Within a particular vote, an individual has to decide the number of "credit voices" to commit, whose square root is what impacts the corresponding outcome of the vote, hence the name quadratic voting. The number of "credit voices" is determined by the individual's voting preference for the proposal. QV has been proposed as a means to break out from the tyranny-of-the-majority vs. subsidising-the-minority dilemma of election systems [6]. Its formal

analysis [7] under a game theoretic price-taking model, has shown that QV outperforms 1p1v in terms of efficiency and robustness. Importantly, it has the unique capacity to deter collusion attacks by effectively taxing extreme behaviours.

Our contributions. In this paper, we propose FEDQV, a novel FL aggregation scheme that draws inspiration from QV. Our objective is to mitigate the ability of malicious peers to impose disproportional damage on the global model – a vulnerability inherent in FEDAVG that applies the 1p1v principle at the granularity of individual votes. By addressing this issue, FEDQV serves as a superior alternative to FEDAVG, offering increased robustness against poisoning attacks while retaining compatibility with privacy-guaranteed mechanisms to defend against privacy attacks, as will be demonstrated later.

Our primary contribution involves integrating QV principles into FL, augmented by the implementation of multiple defensive layers, to establish the truthful mechanism FEDQV. We begin with the incorporation of quadratic computation from QV into the FL setting. This incorporation restricts the ability of malicious peers to inflict high damages by taxing their aggregation weights more than linear. However, this direct incorporation alone falls short of capturing each party's voting preference, a crucial aspect of QV. To capture this preference, we require parties to submit the similarity of their local model with the previous round's global model, which serves as a measure of the parties' preference. This modification enables the integration of QV principles into the FL system. Then in response to potential malicious attempts from peers, we introduce FEDQV, a truthfulness mechanism alongside our application of QV to FL. This mechanism employs a masked voting rule on the server side to conceal the voting calculation process from parties. Additionally, it incorporates outlier detection and a limited voting budget, information that is exclusive to the server. These defence measures collectively act as a deterrent against potential poisoning attacks and untruthful strategic behaviour, ensuring the overall robustness of the FEDQV system in the face of adversarial threats.

To further enhance resilience against poisoning attacks, we extend FEDQV to use adaptive voting budgets. In election-related applications, QV allocates equal budgets to all voters, reflecting the democratic principle of equal rights. However, in our adaptation of QV for FL, it makes sense to allocate more votes to benign peers and limit the influence of malicious ones by assigning them smaller voting budgets. We achieve this by employing unequal budgets, which are tied to a reputation score for each peer, as discussed in Section 5.7.

Our next contribution is the implementation of various state-of-the-art Byzantine fault tolerance techniques on top of FEDQV and demonstrating that FEDQV serves as a complementary defence that further boosts the robustness of existing defence techniques.

> Put differently, FEDQV is an enhancer of existing byzantine fault tolerance techniques, not a competitor – implementing these defences atop FEDQV consistently yields superior results compared to implementing them atop FEDAVG.

To also defend against privacy attacks, we design FEDQV such that it can be easily combined with existing privacy-guaranteeing mechanisms to thwart inference and reconstruction attacks [8, 9, 10]. Specifically, we showcase the compatibility of FEDQV by implementing SECAGG [11] on top of the FEDQV framework. We then demonstrate experimentally the effectiveness of SECAGG within the FEDQV framework, focusing on its ability to withstand powerful privacy attacks.

Our final contribution comprises an extensive theoretical analysis in order to: 1) establish convergence guarantees, and 2) prove the truthfulness of our method.

Our findings. Using a combined theoretical and experimental evaluation, we show that:

- FEDQV is a truthful mechanism and is theoretically and empirically compatible with FEDAVG in terms of accuracy and convergence under attack and no-attack scenarios.
- FEDQV consistently outperforms FEDAVG under various state-of-the-art poisoning attacks, especially for local model poisoning attacks improving the robustness to such attacks by a factor of 4 at least.
- The combination of FEDQV with a reputation model to assign unequal credit voice budgets to parties according to their respective reputations, improves robustness against poisoning attacks by at least 26% compared to the baseline FEDQV that uses equal budgets.
- We show that integrating FEDQV with established Byzantine-robust FL defences, including Multi-Krum [14], Trimmed-Mean [12], and Reputation [13], results in substantial enhancements in accuracy and reductions in the attack success rate (ASR) under state-of-the-art attacks when compared to the original defence methods. Specifically, integrating FEDQV into Multi-Krum results in a twofold improvement in accuracy under two untargeted attack scenarios, along with an average enhancement of 26.72% in accuracy and a notable average reduction of 70% in ASR under two targeted attack scenarios.

- We demonstrate the compatibility of FEDQV with SECAGG, and our empirical evaluation confirms that this integration enhances resistance against two privacy attacks.

2 Background

2.1 Election Mechanisms in FL

Election mechanisms are widely used in distributed systems for choosing a coordinator from a collection of processes [14, 15]. Likewise, there exist works that explore the value of the election mechanism for the aggregation step of FL. Plurality voting is employed in FedVote [16] for weighting the local updates, and FedVoting [17] for treating the validation results as votes to decide the optimal model. Also in [18], the authors propose two forms of election coding, random Bernoulli codes and deterministic algebraic codes, for discovering majority opinions for the aggregation step. DETOX [19] proposes a hierarchical aggregation step based on majority votes upon groups of updates. DRACO [20] and ByzShield [21] also employ majority voting to fend off attacks against the aggregation step. All the aforementioned election mechanisms suffer from the tyranny of the majority problem in election systems [22]. In FL, this means that if attackers manage to control the majority of votes, then via poisoning their tyranny will manifest itself as a degradation of the accuracy of the FL model used by the minority.

To address these limitations, QV is proposed as a solution that combines simplicity, practicality, and efficiency under relatively broad conditions. QV considers a quadratic vote pricing rule, inspired by economic theory, under which voters can purchase votes at ever-increasing prices within a predetermined voting budget. The advantages of QV over 1p1v have a rigorous theoretical basis, which of course applies also to the use of QV in FL. For any type of symmetric Bayes-Nash equilibrium, the price-taking assumption approximately holds for all voters, as a result, the expected inefficiency of QV is bounded by constants [23]. This theoretical analysis [24, 25] combined with strong empirical validation, both at the laboratory [26] and on the field [27], suggest that QV is near-perfectly efficient and more robust than 1p1v which, as already explained, forms the basis of contemporary FL aggregation mechanisms. The advantages of QV can also be observed from the viewpoint of collusion, which is generally deterred either by unilateral deviation incentives or by the reactions of non-participants [7].

2.2 FL Aggregation Against Poisoning and Privacy Attacks

There exist several Byzantine-robust FL aggregation methods for mitigating poisoning attacks either by leveraging statistic-based outlier detection techniques [4, 12, 28, 13] or by utilising auxiliary labelled data collected by the aggregation server in order to verify the correctness of the received gradients [29, 30]. Both approaches, though, require examining the properties of the updates of individual parties, which can jeopardise their privacy due to inference [10] and reconstruction attacks [8, 9] mounted by an honest but curious aggregation server.

To fight against privacy attacks, Secure aggregation (SA) protocols have been proposed as potential countermeasures [31]. Secure aggregation can be achieved using four main privacy-enhancing technologies: differential privacy [32], trusted-execution environment (TEE) [33], secure shuffling under anonymity assumptions [34], and cryptography. Among those, secure aggregation based on cryptography is the most widely studied [11, 35, 36]. Although crypto-based secure aggregation provides strong security guarantees compared to alternatives, i.e., differential privacy, it also suffers from high computational and communication overhead. Consequently, these crypto-based secure aggregations are primarily established in the FEDAVG framework due to the impracticality of integrating them into Byzantine-robust FL aggregation methods, owing to the computational complexity in these aggregations.

In response to this limitation, FEDQV emerges as a promising solution, offering enhanced resilience against poisoning attacks while inheriting the simplicity of FEDAVG. Notably, this simplicity facilitates FEDQV's integration with crypto-based secure aggregations, thereby bolstering defences against privacy attacks while also exhibiting superior resilience against poisoning attacks compared to FEDAVG. Although a few FL aggregation approaches [37, 38, 39] can be adapted to incorporate secure aggregation, they still rely on majority voting as the aggregation scheme, which can be integrated with FEDQV to enhance its robustness against poisoning attack.

3 FEDQV: Quadratic Voting in FL

3.1 Federated Learning Setting

Consider an FL system involving N parties and a central server. During training round t, a subset of parties S^t is selected to participate in the training task. Each party i has the local dataset $\mathcal{D}_i$ with $|\mathcal{D}_i|$ samples (voters), drawn from

non-independent and non-identically (Non-IID) distributed, $X_i \sim (\mu_i, \sigma_i^2)$. The goal of using FL is to learn a global model for the server. Given the loss function $\ell(w; D)$, the objective function of FL can be described as

$$L(w) = E_{D \sim X}[\ell(w; D)]$$

Therefore, the task becomes:

$$w^* = \arg\min_{w \in R^d} L(w)$$

To find the optimal w^*, Stochastic Gradient Descent (SGD) is employed to optimise the objective function. Let T be the total number of every part's SGD, E be the local iterations between two communication rounds, and T_E is the number of communication rounds.

The FL model training process entails several rounds of communication between the parties and the server, including broadcasting, local training, and aggregation, as demonstrated in Algorithm 1. The global model is first initialised at a random state by the server and, then, in round t, the following steps are taken:

Broadcast: The server first randomly selects a subset of parties S^t ($|S^t| = C \cdot n$, $C \in (0, 1]$) and then distributes the latest global model as the global proposal to all chosen parties.

Local Training: The selected party i performs local computation based on the global proposal and its local dataset D_i, and sends the updated w_i^t back to the server:

$$w_i^t \leftarrow w_i^{t-1} - \eta_{t-1} \frac{\partial_i \ell(w_i^{t-1}; D_i)}{\partial w}$$

where η_{t-1} is the learning rate.

Aggregation: The server receives the model updates from all participating parties and aggregates them to update the global model.

For aggregation rule, FEDAVG uses the fraction of the local training sample size of each party over the total training samples as the weight of a party:

$$w^{t+1} = \left(\frac{1}{\sum_{i \in S^t} |D_i|}\right) \sum_{i \in S^t} |D_i| \cdot w_i^t$$

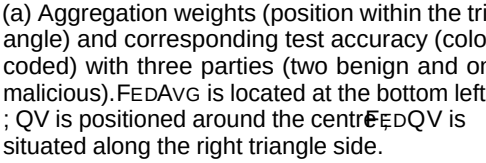
Similar to 1p1v, each sample here represents a single voter, and since party i possesses $|D_i|$ samples, it is able to cast $|D_i|$ votes for its local model during the aggregation. Hence, the global proposal is a combination of all parties' local proposals weighted by their votes.

3.2 Preliminary Results

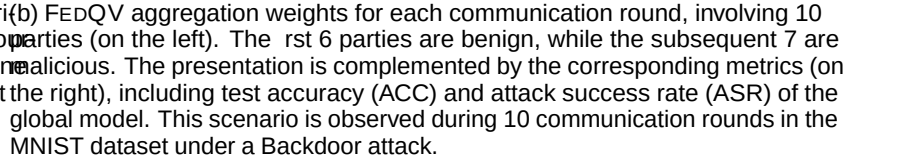
To show the rationale behind the algorithm design, we consider a toy use case involving two benign parties and one malicious party in FL. These parties claim possession of training datasets with the dataset sizes $\{1, 1, 2\}$, to respectively. Importantly, the malicious party asserts a larger training dataset than the benign ones. These claims are reported to the central server, which lacks access to the raw data of the participating parties for verification purposes.

First, we introduce quadratic computation from QV into the FL. The quadratic computation involves taxing the aggregation weights with the square root. Consequently, while the aggregation weights in FEDAVG remain $\{1, 1, 2\}$, the corresponding weights in QV are adjusted to $\{1, 1, \sqrt{2}\}$. We conduct a 10-round training of a multi-layer CNN on the MNIST dataset, during which the malicious party executes a backdoor attack under the same settings as discussed in Section 5. The test accuracy is colour-coded and presented in Figure 1a, where the vertices of the triangle correspond to different parties, and their positions inside the triangle reflect their respective aggregation weights. As expected, the compromised accuracy in FEDAVG arises from the presence of a malicious party that possesses or falsely claims to possess a larger dataset. Compared to FEDAVG, QV, with the adjusted weights, exhibits superior accuracy. This suggests that the quadratic computation in QV can enhance performance by restraining the influence of attackers within FEDAVG by taxing their aggregation weights more than linear.

However, the direct incorporation of quadratic computation alone proves insufficient in capturing the voting preferences of each party, a critical aspect of QV. In QV, parties with pronounced disagreements with a proposal express their dissent by casting more votes to reject it. To capture each party's voting preference into the FL system, we refine the integration by requiring parties to submit the similarity of their local model with the previous round's global model. The similarity score serves as a measure of agreement, where higher scores indicate stronger alignment between the local model (proposal) and the global model (proposal). Consequently, parties with higher similarity scores will cast fewer votes in this round, reflecting their reduced need for adjustment to the existing global model. Conversely, parties showing greater dissimilarity ($1 - s_i^t$) between their local proposal and the global one cast more votes. This strategic



(a) Aggregation weights (position within the triangle) and corresponding test accuracy (colour coded) with three parties (two benign and one malicious). FEDAVG is located at the bottom left; QV is positioned around the centre, FEDQV is situated along the right triangle side.



(b) FEDQV aggregation weights for each communication round, involving 10 parties (on the left). The first 6 parties are benign, while the subsequent 7 are malicious. The presentation is complemented by the corresponding metrics (on the right), including test accuracy (ACC) and attack success rate (ASR) of the global model. This scenario is observed during 10 communication rounds in the MNIST dataset under a Backdoor attack.

allocation empowers parties with substantial disagreements to wield more significant influence over the impending global proposal w^t, thereby making it a closer alignment with their local proposal v_i^t.

Based on these similarity scores, the voting calculation is conducted to determine the votes, serving as the aggregation weights. Rather than having parties directly calculate their votes, which could lead to malicious attempts, we introduce the FEDQV mechanism. This approach adopts a more secure approach by delegating the vote allocation task to the server. It employs a masked voting rule to obscure the vote calculation process from the parties, thereby preventing them from knowing the exact votes they have cast for aggregation. This confidentiality measure is crucial to maintaining the integrity and impartiality of the voting process. Built upon the lack of awareness regarding the voting calculation process among parties, the FEDQV system incorporates two additional defensive mechanisms: a limited voting budget and outlier detection, which are known only by the server. Given that parties are unaware of their budget and the possibility of their updates being detected as abnormal, attempts by malicious parties to manipulate their updates and overcast their votes entail a dual risk. There is a potential for exclusion from the aggregation process through outlier detection or unknowingly reaching their voting budget limit, resulting in the assignment of no votes. These protections serve as a deterrent against potential poisoning attacks, ensuring the overall robustness of the FEDQV system in the face of adversarial threats. Details of the design of FEDQV are presented in the next subsection.

Returning to our previous toy example, the implementation of FEDQV leads to the allocation of weights $\{1, 1, 0\}$, as depicted in Figure 1a. This weight distribution effectively excludes the malicious party from the aggregation process. Consequently, this exclusion contributes to the improved accuracy of the resulting global model. To further illustrate how FEDQV assigns aggregation weights to different parties during training, consider another toy case with 10 parties in the FL system, where 4 of them are attackers. The settings remain the same as the first toy case, and the results are presented in Figure 1b. In the left of Figure 1b, the first six parties are benign, and the rest are malicious. Notably, during the 10 communication rounds, only two attackers (parties 7 and 8) successfully participated in the aggregation, with party 7 participating twice and party 8 once, while most others did not contribute, resulting in an aggregation weight of 0 for them. This outcome demonstrates FEDQV's effectiveness in expelling malicious parties. Even when a round includes two malicious parties, the decrease in test accuracy is limited, and the Attack Success Rate (ASR) remains low (<2%) as shown in the right of Figure 1b. This demonstration underscores FEDQV's capability to mitigate the influence of malicious parties, thus preventing significant damage to the global model.

3.3 FEDQV Design

We use QV in FL to overcome the drawback of 1v1, which improves the robustness of aggregation in comparison to FEDAVG without compromising any efficiency. Figure 2 provides an overview of FEDQV algorithm, which comprises two key components: similarity computation executed on the party side to capture the voting preference of each party and voting scheme managed on the server side to deter potential poisoning attacks and untruthful strategic behaviour of parties. We detail each component in the following subsections.

Similarity Computation: In round t, based on the server instructions, party i ($i \in S^t$) trains its local model w_i^t, which can be regarded as its local proposal. Following the local training phase, party i computes a similarity score s_i^t utilising

Figure 2: Overview of FEDQV algorithm.

cosine similarity, quantifying the alignment between its locally trained model w_i^t and the previous global model w^{t-1}.

$$s_i^t = S_{cos}(w_i^t, w^{t-1}) = \frac{w_i^t \cdot w^{t-1}}{\|w_i^t\| \|w^{t-1}\|}$$

Notably, the cosine similarity function can be adapted to different similarity metrics, such as L2 distance, to better suit specific tasks. In this context, a higher s_i^t value indicates a stronger agreement with the previous global model (proposal). Once selected parties finish training, they send their updates w_i^t to the server, with the message s_i^t.

It's noteworthy that similarity calculation can also be launched on the server side. However, this approach presents certain risks: (i) it exposes the system to privacy attacks initiated by the server, and (ii) it may produce distorted similarity scores due to the utilisation of regularisation and privacy-preservation methods [40] at the party side. Given our primary objective of comparing FEDQV with FEDAVG, where weights are also computed at the party side, we choose to conduct the calculation on the party side within FEDQV. Additionally, to counteract potential malicious attempts launched from the party side, we have the option to introduce defence mechanisms on the server side, such as other Byzantine-robust aggregations, as detailed in Section 5.8. This approach makes it more challenging for attackers to mount successful attacks compared to FEDAVG.

Voting Scheme (Server Side). Upon receiving the updates and messages from selected parties, the server proceeds with the following steps:

- The server normalises the similarity scores using Min-Max scaling to obtain $\tilde{s}_i^t$ as:

$$\tilde{s}_i^t = \text{norm}(s_i^t; \{s_i^t\}_{i \in [S^t]}) \quad (1)$$

  Following normalisation, parties no longer being aware of their exact similarity scores s_i^t.

- The server employs a penalty mechanism for abnormal similarity scores, specifically $\tilde{s}_i^t$ falls below $-\lambda$ or exceeds $1-\lambda$. Here, λ represents the similarity threshold and is established to capture scores that are excessively small or large, indicating potential anomalies. In response to such abnormality, the server enforces a penalty by reducing the budget B_i allocated to the respective party as:

$$B_i = \max(0, B_i + \ln(\tilde{s}_i^t - \lambda)) \quad (2)$$

- The server calculates the voice credit c_i^t for party i utilising the masked voting rule H as:

$$c_i^t = H(\tilde{s}_i^t) = \ln(\tilde{s}_i^t + 1) \cdot \mathbb{1}_{-\lambda < \tilde{s}_i^t < 1-\lambda} \quad (3)$$

  Here, the voice credits signify the price party i is required to pay in round t for its local proposal. Parties with higher similarity scores, indicating stronger agreement with the global proposal, are allocated fewer credit votes from the server. Conversely, parties showing greater dissimilarity $\tilde{s}_i^t$ () between their local proposal and the global one receive an increased allocation c_i^t. This mechanism empowers parties with substantial disagreements to exert greater influence over the forthcoming global proposal w^t. Notably, parties with abnormal similarity scores are excluded from participating in the aggregation, without receiving zero voice credits.

- The server checks the budget B_i for each party and computes their final votes v_i^t as:

$$v_i^t = \sqrt{\min(c_i^t, \max(0, B_i))} \quad (4)$$

Algorithm 1: FEDQV

Input : w^0 random initialisation, B, FEDQV parameters

Server :

for Iteration t 1 to $\frac{T}{E}$ do
  Broadcast w^{t-1} to randomly selected set of parties S^t ($|S^t| = C$ 1);
  Receive the local updates (w_i^t, s_i^t) from selected parties $i \in S^t$);
  Normalisation: s_i^t Eq.(1), $i \in S^t$;
  for i 1 to N do in parallel
    if s_i^t or s_i^t 1 then
      Update B_i $\max(0, B_i + \ln s_i^t - 1)$
    Credit voice c_i^t Equation 3, Vote v_i^t Equation 4;
    Budget B_i $\max(0, B_i - (v_i^t)^2)$ // Update the budget
  end forpar
  return w_n^t $\sum_{i=1}^{N} \frac{v_i^t}{\sum_{i=1}^{M} v_i^t} w_{i,n}^t$
end for

Party :

for Party $i \in S^t$ do in parallel
  Receive the global model w^{t-1};
  for local epoch e 1 to E do
    w_i^t w_i^{t-1} $r_{t-1} \frac{\partial \ell_i(w_i^{t-1}; D_i)}{\partial w}$ // Local training
  end for
  Calculate the similarity score s_i^t $\frac{\langle w_i^t, w^{t-1} \rangle}{\|w_i^t\| \|w^{t-1}\|}$;
  Send (w_i^t, s_i^t)
end forpar

- The server updates the budget as:

$$B_i = \max(0, B_i - v_i^{t\,2}) \tag{5}$$

Thus, the server determines the weight v_i^t of party i for aggregation and generates the updated global model w^t, reflecting the collective opinion of all selected parties (voters). Algorithm 1 summarises all these steps of FEDQV.

Malicious Party: In cases where malicious parties attempt to manipulate the similarity scores, their capabilities are restricted by:

(a) No knowledge of the voting process: Only the server possesses knowledge of each party's remaining budget and the number of actual votes cast in the current round. The lack of knowledge about the voting calculation process that contains outlier detection, coupled with the parties' unawareness of their limited voting budget, exposes malicious parties to the risk of exclusion from the aggregation process.

(b) Punitive Measures: FEDQV, with its masked voting rule that contains outlier detection and limited budget, empower the system to penalise and potentially remove malicious parties who attempt to conduct poisoning attacks;

(c) Limited Influence: Even if a manipulated similarity score is accepted by the server, the influence the malicious party can exert is inherently constrained due to the nature of QV and the limited budgets, minimising the potential damage.

Benefits of FEDQV:

- Truthful Mechanism. FEDQV is a truthful mechanism [41] as we prove in Theorem 4.17. This means that this mechanism compels the parties, even malicious ones, to tell the truth about their votes (weights) for aggregation, rather than any possible lie. This truthfulness is reinforced by the aforementioned several defence layers.
- Ease of Integration and Compatibility. FEDQV is highly adaptable and can be seamlessly integrated into Byzantine-robust FL defence schemes with minimal adjustments, specifically by modifying the aggregation weight calculation while leaving other algorithm components unchanged. This integration is demonstrated

Algorithm 2: FEDQV with Adaptive Budget

Input : $w_i^t; c_i^t; B_i^t$ FEDQV; $, a, W, M, ,$ Reputation model parameters

```
for i ∈ S^t do
    for j ← 1 to M do
        Subjective Observations (P_i^t; Q_i^t) := IRLS(w_{i,j}^t ; );
    end for
    Reputation Score R_i^t := Rep(P_i^t; Q_i^t; ; a; W)
    Budget B_i^t ← R_i^t 1_{R_i^t} + B_i^t, Credit voice c_i^t ← (R_i^t + c_i^t) 1_{R_i^t}
end for
```

in Section 5.8. Furthermore, similar to FEDAVG, FEDQV boasts efficient communication and simplicity, rendering it compatible with various mechanisms employed in FL. It can effortlessly incorporate the regularisation, sparsification, and privacy modules, encompassing techniques such as clipping [40], gradient compression [42], differential privacy [43], and secure aggregation [11].

3.4 FEDQV with Adaptive Budgets

In democratic elections, all individuals are typically granted equal voting rights, entailing an equal voting budget. In FL, however, it often makes sense to give malicious parties fewer votes than honest ones. Thus to improve the robustness of standard FEDQV, we combine it with the reputation model in [13] to assign an unequal budget based on the reputation score of parties in each round. Specifically, if a party's reputation score R^t surpasses a predefined threshold, we increase their budget, and vice versa. We present a summary of this combination in Algorithm 2, with a detailed explanation, expanding on the well-established components from the original paper. We provide empirical evidence in Section 5.7 showcasing the substantial performance improvements achieved by the enhanced version of FEDQV featuring an adaptive budget.

Here we present a concise elucidation of key components of the Algorithm 2 as followings:

- IRLS (Iteratively Reweighted Least Squares): IRLS serves as an optimisation technique employed to solve specific regression problems. Within [13], IRLS is utilised to compute the Subjective Observations of participating clients based on their parameter's confidence score, which is calculated using the repeated-median regression technique.
- Subjective Observations: Positive observations denoted by P_i^t signify acceptance of an update, while negative observations denoted by Q_i^t indicate rejection. Consequently, positive observations enhance a party's reputation, and negative ones have the opposite effect.
- Reputation Score Calculation: The reputation score of a party is determined using a subjective logic model, formulated as follows:

$$R_i^t = \frac{P_i^t + Wa}{P_i^t + Q_i^t + W}$$

Regarding the integration of the reputation model, our objective is to demonstrate how combining FEDQV with the reputation model enables the allocation of unequal budgets, thereby enhancing the robustness of standard FEDQV. This integration's adaptability extends beyond a single reputation model, allowing customisation to suit various needs. The example presented in the paper serves to showcase the concept's viability.

4 Theoretical Analysis

In this section, we establish the convergence guarantees and truthfulness properties of FEDQV. Our first result is Theorem 4.9 that states FEDQV converges to the global optimal solution at a rate $O(\frac{1}{T})$, comparable to the convergence rate exhibited by FEDAVG. Theorem 4.10 extends this statement, affirming that FEDQV converges to a near-optimal solution in the presence of malicious parties, with the resulting performance gap determined by the percentage of malicious parties. The empirical convergence performance of our algorithm, as gauged by metric test accuracy and training loss, is consistent with our theoretical analysis, as elaborated in the following section. Our final result, Theorem 4.17, establishes that FEDQV is a truthful mechanism, wherein honesty emerges as the dominant strategy within this framework. Fully detailed proofs are in Appendix A.

4.1 Convergence

We start by stating our assumptions, which are standard and common for such types of analysis and per recent works such as [12, 28, 44, 30, 13, 45].

Assumption 4.1. The loss functions are L-smooth, which means they are continuously differentiable and their gradients are Lipschitz-continuous with Lipschitz constant $L > 0$, whereas:

$$\forall i \in N,\ \forall w_1, w_2 \in R^d,\ \|\nabla L(w_1) - \nabla L(w_2)\|_2 \leq L\|w_1 - w_2\|_2$$
$$\|\nabla \ell(w_1; D) - \nabla \ell(w_2; D)\|_2 \leq L\|w_1 - w_2\|_2$$

Assumption 4.2. The loss function $\ell(w_i; D)$ are μ-strongly convex:

$$\exists \mu > 0, \forall w_1, w_2 \in R^d,\ \nabla \ell(w^*; D) = 0,\ \nabla L(w^*) = 0$$
$$2(L(w_1) - L(w_2)) \leq 2\langle \nabla L(w_2), w_1 - w_2\rangle + \mu\|w_1 - w_2\|_2^2$$
$$2(\ell(w_1; D) - l(w_2; D)) \leq 2\langle \nabla \ell(w_2; D), w_1 - w_2\rangle + \mu\|w_1 - w_2\|_2^2$$

Assumption 4.3. The expected square norm of gradients is bounded:

$$\forall w \in R^d, \exists G_w < +\infty,\ E\|\nabla \ell(w; D)\|_2^2 \leq G_w^2$$

Assumption 4.4. The variance of gradients is bounded:

$$\forall w \in R^d, \exists V_w < +\infty,\ E\|\nabla \ell(w; D) - E(\nabla \ell(w; D)\|_2^2 \leq V_w$$

The lemmas we utilise in the proof of Theorem 4.9 and Theorem 4.10, are presented below.

Lemma 4.5. From Assumption 4.1 and 4.2, $L(w)$ is L-smooth and μ-strongly convex. Then $\forall w_1, w_2 \in R^d$, one has

$$\langle \nabla \ell(w_1) - \nabla \ell(w_2), w_1 - w_2\rangle \geq \frac{L\mu}{L+\mu}\|w_1 - w_2\|_2^2 + \frac{1}{L+\mu}\|\nabla \ell(w_1) - \nabla \ell(w_2)\|_2^2$$

Lemma 4.6. Assume Assumption 4.1, Assumption 4.2 and Lemma 4.5 hold, $\forall i, \exists c_i^t = 1_{i \in S^t}$, we have

$$\left\|w^{t+1} - \eta r \nabla L(w^{t+1}) - w^*\right\|_2^2 \leq \sum_{i=1}^{N} p_i^{t+1}\left\|w^{t+1} - w_i^{t+1}\right\|_2^2 + r^2\left(1 + L^2 - \frac{2rL\mu + 1}{L+\mu}\right)\left\|w^{t+1} - w^*\right\|_2^2 \quad (6)$$

Lemma 4.7. Assume Assumption 4.3 holds, it follows that

$$E\sum_{i=1}^{N} p_i^{t+1}\left\|w^{t+1} - w_i^{t+1}\right\|_2^2 \leq (E-1)^2 r^2 G_w^2$$

Lemma 4.8. Assume Assumption 4.4 holds, according to our Algorithm 1, it follows that

$$E\left\|F(w^{t+1}) - \nabla L(w^{t+1})\right\|_2^2 \leq (1-\mu)^2 C V_w \frac{N-B}{B}$$

Where

$$F(w^{t+1}) = \sum_{i \in S^{t+1}} p_i^{t+1} \nabla \ell(w_i^{t+1}; D_i^{t+1})$$

Under these four mild and standard assumptions, along with the support of Lemmas, we have:

Theorem 4.9. (Secure Convergence Without Attack) Under Assumptions 4.1, 4.2, 4.3 and 4.4, and $\mu > 0$. Choose $\eta = \frac{L+\mu}{L\mu}$ and $r = 2\frac{(L+1)(L+\mu)}{L\mu}$, then FEDQV satisfies

$$E\left[L(w^T) - L(w^*)\right] \leq \frac{L}{2\varphi + T}\left(\varphi E\left\|w^0 - w^*\right\|_2^2 + \frac{\Gamma^2}{2}\right) \quad (7)$$

Where

$$\Gamma = (E-1)^2 G_w^2 + (1-\mu)^2 C V_w \frac{N-B}{B}, \quad \varphi = \frac{\mu}{L(L+1)}$$

Suppose the percentage of attackers in the whole parties is m, we denote

$$M_i(w_i^t) = \begin{cases} * & \text{if } i \in \text{malicious parties} \\ r`(w_i^t; D_i^t) & \text{if } i \in \text{honest parties} \end{cases}$$

Where $*$ stands for an arbitrary value from the malicious parties. Then we have:

Theorem 4.10. (Robust Convergence Under Attack) Under Assumptions 4.1, 4.2, 4.3 and 4.4, Choose $\eta = \frac{L+\mu}{L\mu}$ and $\gamma = 2\frac{(L+1)(L+\mu)}{L\mu}$, then FEDQV satisfies

$$E[L(w^T) - L(w^*)] \leq \frac{L+2L\gamma r_{T-1}\$}{2\gamma' + T} \left(\frac{\gamma'}{2} E\|w^0 - w^*\|_2^2 + \frac{\gamma^2}{2} + \frac{L\$\gamma^2}{2} \right) \quad (8)$$

Where

$$\gamma = (E-1)^2 G_w^2 + (1-\gamma)^2 CV_w^2 \overline{B}, \quad \gamma' = \mu(L+1), \quad \$ = mNG_w r_{T-1} \sqrt{\frac{4+6\gamma^2}{}}$$

Remark 4.11. According to Theorem 4.9, FEDQV obtains a secure convergence rate of $O(\frac{1}{T})$ in the absence of malicious parties, which is comparable to the convergence rate of FEDAVG [46].

Remark 4.12. According to Theorem 4.10, FEDQV converge to a near-optimal solution in the presence of malicious parties, with the resulting performance gap determined by the percentage of malicious parties.

Remark 4.13. The error rate exhibits dependence on the budget B, the similarity threshold γ, and the percentage of malicious parties m. It is noteworthy that a larger budget allocation, a reduction in the similarity threshold, or an augmentation in the proportion of malicious parties induce more pronounced disparities in model updates, consequently resulting in an elevated error rate. The impact of these hyperparameters is shown in Figure 6b in Section 5.10.

4.2 Truthfulness

The FEDQV mechanism belongs to a single-parameter domain, where the real parameter v_i plays a direct role in determining the eligibility of party i for aggregation. The mechanism is normalised according to game theory principles [41], where for every v_i and v_{-i} such that $f(v_i; v_{-i}) \in W_i$, $p_i(v_i; v_{-i}) = 0$. Here, v_{-i} represents votes cast by all parties except for i, W_i is the subset of participants in aggregation, f represents the voting scheme outcome, and p_i is the payment function with $p_i(v_i; v_{-i}) = v_i^2$ in FEDQV. The upcoming definition of truthfulness and the following lemmas contribute to the proof of Theorem 4.17 in alignment with monotone and critical value concepts in game theory [41].

Definition 4.14. A mechanism $(f; p_1; \ldots; p_n)$ is called truthfulness if for every party we denote $a = f(v_i; v_{-i})$ and $a' = f(v_i'; v_{-i})$ as the outcome of the voting, then $v_i(a) - p_i(v_i; v_{-i}) \geq v_i(a') - p_i(v_i'; v_{-i})$, where $v_i(a)$ denotes the gain of party i if the outcome of the voting is a.

Here, $v_i(a) - p_i(v_i; v_{-i})$ is the utility of party i, indicating the gain from voting ($v_i(a)$) minus its cost ($p_i(v_i; v_{-i})$). Intuitively this implies that party i with v_i would prefer "telling the truth" v_i to the server rather than any possible "lie" v_i' since this gives him higher (in the weak sense) utility.

Lemma 4.15. f is monotone: $\forall v_{-i}$ and $\forall v_i' > v_i$, if $f(v_i; v_{-i}) \in W_i$, then $f(v_i'; v_{-i}) \in W_i$.

Lemma 4.16. In FEDQV, $\forall i; v_i; v_{-i}$ that $f(v_i; v_{-i}) \in W_i$, we have that $p_i(v_i; v_{-i}) = \tau_i(v_{-i})$, where τ_i is the critical value of a monotone function f on a single parameter domain that $\tau_i(v_{-i}) = \sup_{v_i : f(v_i; v_{-i}) \in W_i} v_i$.

Based on Lemma 4.15 and Lemma 4.16, we establish the following theorem:

Theorem 4.17. FEDQV is incentive compatible (truthful).

Remark 4.18. Regarding the concept of truthfulness, it theoretically ensures that being honest is the dominant strategy since providing manipulated similarity scores may lead to penalties and removal from the system due to the masked voting rule H and limited budget B. This is an integral part of the nature of QV embedded within FEDQV framework.

5 Experiments

The objectives of our experimental evaluation are the following: (a) To assess the performance of our aggregation method relative to FEDAVG. (b) To benchmark our method across 10 distinct state-of-the-art poisoning attack scenarios. (c) To validate the alignment of our experimental findings with our prior theoretical analyses. (d) To demonstrate the ease of integration and compatibility of our method within Byzantine-robust FL defence schemes and privacy-preserving mechanisms.

5.1 Experimental setting

Datasets and global models. We implement the typical FL setting where each party owns its local data and transmits/receives information to/from the central server. To demonstrate the generality of our method, we train different global models on different datasets. We use four popular benchmark datasets: MNIST [47], Fashion-MNIST [48], FEMNIST [49] and CIFAR10 [50]. We consider a multi-layer CNN same as in [40], consisting of 2 convolutional layers and 2 fully connected layers for MNIST, Fashion-MNIST and FEMNIST, and the ResNet-18 [51] for CIFAR10.

Non-IID setting. In order to fulfil the setting of a heterogeneous and unbalanced dataset for FL, we sample from a Dirichlet distribution with the concentration parameter $= 0.9$ as the Non-IID degree as in [52, 53], with the intention of generating non-IID and unbalanced data partitions. Moreover, we have examined the performance across varying levels of non-IID data in Section 5.9.

Parameter Settings. The server selects 10% (C) out of 100 (N) parties to participate in each communication round and train the global models for 100 communication rounds (T_{FED}), where the local training epoch E equals 5. We set the model hyper-parameters budget B and the similarity threshold to 30 and 0.2 respectively based on the hyper-parameter searching. All additional settings are provided in the Appendix B.1.

5.2 Thread Model

We assume we the parties are **malicious** while the server is **honest-but-curious**. Within this context, parties may deliberately submit false or manipulated local model updates to the central server. Their objectives could range from injecting biases and compromising the model's performance to extracting sensitive information from the aggregated global model. The server correctly follows the FL protocol steps but remains curious to discover any private information. In our case, the server has full visibility of all local models and launches privacy attacks upon receiving an update from a target party to reconstruct users' training data.

Furthermore, the percentage of malicious parties is an important factor in determining the success of the poisoning attack. In our analysis, we assume that the number of malicious parties is less than the number of honest parties, a common setting in such types of analysis and recent works [4, 12, 13].

5.3 Evaluated Poisoning Attacks

Our paper addresses three distinct attack schemes:

- **Data poisoning**: Attackers submit the true similarity score based on their poisoned updates, including Label Flip Attack [54], Gaussian Attack [55], Backdoor [56], Scaling Attack [52], Neurotoxin [57].
- **Model poisoning**: Attackers submit the true similarity score based on their clean updates and poison their model, including: Krum Attack [54], Trim Attack [54], and Aggregation-agnostic attacks: Min-Max and Min-Sum [58]
- **QV-tailored Attack**: Attackers submit both poisoned similarity score and the local model to exploit vulnerabilities in FEDQV. This dual-pronged attack, termed QV-Adaptive, presents a heightened challenge for FEDQV.

Here are the details of the aforementioned attacks. We begin with data poisoning attacks:

Label Flip Attack [54]: In the Label-Flip scenario, all the labels of the training data for the malicious clients are set to zero. This scenario simulates a directed attack, with the goal to disproportionally bias the jointly trained model towards one specific class. This is a data poisoning attack that does not require knowledge of the training data distribution. Under this attack, the malicious parties train with clean data but with flipped labels. Specifically, we flip a label k as label $K - k - 1$, where K is the total class number.

Gaussian Attack [55]: This attack forges local model updates via Gaussian distribution on the malicious parties. malicious parties forge local model updates via Gaussian distribution.

Backdoor Attack [56] Malicious parties inject specific backdoor triggers into the training data and modify their labels to the attacker-chosen target label. Specifically, we use the same backdoor pattern trigger and attacker-chosen target label as in [59] as our trigger and set the attacker-chosen target label as 5. The backdoor can be introduced into a model by an attacker who poisons the training data with specially crafted inputs. A backdoor transformation applied to any input causes the model to mis-classify it to an attacker-chosen label The pattern must be applied by the attacker during local training, by modifying the digital image.

Scaling attack [52] The malicious parties generate poisoned local model updates by backdoor attack and only launch this attack during the last communication round after scaling these updates by a factor of λ.

Neurotoxin attack [57] In this attack, the adversary starts by downloading the gradient from the previous round and employs it to approximate the benign gradient for the upcoming round. The attacker identifies the top-k% coordinates of the benign gradient and treats them as the constraint set. Over several epochs of Projected Gradient Descent (PGD), the attacker computes gradient updates on the manipulated dataset and projects this gradient onto the constraint set, which consists of the bottom-k% coordinates of the observed benign gradient. PGD is employed to approach the optimal solution within the span of the bottom-k% coordinates. We adopt the original parameter setting from the paper, where k is set to 0.1.

Next, we move to model poisoning attacks:

Krum Attack [54]: Malicious parties craft poisoned local model updates opposite from benign ones, and enable them to circumvent the defence of Krum [4].

Trim Attack [54] The poisoned local model updates constructed by malicious parties are optimised for evading the Trim-mean and Median [12].

Min-Max Attack [58] In order to ensure that the malicious gradients closely align with the benign gradients within the clique, attackers strategically compute the malicious gradient. This computation is carried out to limit the maximum distance of the malicious gradient from any other gradient, which is constrained by the maximum distance observed between any two benign gradients.

Min-Sum [58] The Min-Sum attack enforces an upper bound on the sum of squared distances between the malicious gradient and all the benign gradients. This upper bound is determined by the sum of squared distances between any one benign gradient and the rest of the benign gradients.

Finally, we introduce the QV-tailored attack:

QV-Adaptive attack Tailored for FEDQV, this attack leverages the Aggregation-agnostic optimisation [58] within the LMP framework [54]. This attack manipulates both the similarity score and the local model, following the procedure below:

1. The malicious party i generates benign updates w_i^t using clean data $\mathcal{D}_i$ in round t and calculates the corresponding similarity score;
2. malicious parties (with counts of m) collectively normalise all the similarity scores and employ the Aggregation-agnostic Min-Max optimisation to select the optimal similarity score. This optimisation objective aims to increase the likelihood of the score being accepted by the server.
3. the adaptive attack focuses on local model poisoning to optimise the following problem:

$$\max \quad (9)$$

$$\text{s.t.} \quad w_{i2m}^{t'} = \text{FedQV}(w_1^t; w_2^t; \dots; w_m^t) \quad (10)$$

$$w_{i2m}^{t'} = w_i^t - \lambda \hat{d} \quad (11)$$

Here, $\hat{d}$ represents a column vector encompassing the estimated changing directions of all global model parameters. The variables w_{i2m}^t and $w_{i2m}^{t'}$ correspond to the local model before and after the attack. The parameter λ denotes the extent of the attack's impact on the model.

It is noteworthy that Label flip, Gaussian, Krum, Trim, Min-Max, Min-Sum and QV Adaptive attacks are untargeted attacks, whereas, Backdoor, Scaling and Neurotoxin attacks are targeted attacks. We confine our analysis to the worst-case scenario in which the attackers submit the poisoned updates in every round of the training process for all attack strategies with the exception of the Scaling attack.

5.4 Performance Metrics

We use the average test accuracy (ACC) of the global model to evaluate the result of the aggregation defence for poisoning attacks, in which attackers aim to mislead the global model during the testing phase. ACC is the percentage of testing examples with the correct predictions by the global model in the whole testing dataset, which is defined as ACC = (# correct predictions) / (# testing samples). In addition, there are targeted attacks that aim to attack a specific label while keeping the accuracy of classification on other labels unaltered. Therefore, besides ACC, we choose the attack success rate (ASR) to measure how many of the samples that are attacked, are classified as the target label chosen

(a) Without attack. (b) Under label ip attack with 40% malicious parties.

Figure 3: Convergence comparison of FEDAVG and FEDQV in Training Loss (TL) and Accuracy (ACC) over 100 epochs across four benchmark datasets, both under no attack and under attack scenarios.

by malicious parties. A robust federated aggregation method would obtain a higher Avg-ACC as well as a lower ASR under poisoning attacks. An ideal aggregation method can achieve 100% Avg-ACC and has the ASR as low as the fraction of attacked samples from the target label.

5.5 Convergence

We evaluate the convergence of FEDAVG and FEDQV across the aforementioned four datasets under both benign and adversarial conditions. The training loss (TL) and accuracy (ACC) of global models trained using FEDQV and FEDAVG are depicted in Figure 3a without any attack and in Figure 3a under label ip attacks with 40% malicious parties. In the absence of Byzantine attacks, the convergence of the global model trained with FEDQV matches that of FEDAVG across all datasets, consistent with Theorem 4.9. However, under attack, FEDAVG struggles to converge due to its susceptibility to poisoning attacks, as shown in Figure 3a. While FEDQV also exhibits slower convergence under attack, it eventually converges and outperforms FEDAVG. This outcome aligns with Theorem 4.10, indicating that FEDQV achieves convergence to a near-optimal solution even in the presence of malicious parties, with the extent of performance improvement determined by the percentage of malicious parties.

5.6 Defence against Poisoning Attacks

Static percentage of attackers: We present ACC and ASR of global models trained using both FEDAVG and FEDQV under the 10 aforementioned poisoning attacks across all four datasets in Table 1. The experiments involve 30% malicious parties, same as in previous studies such as [13, 43], which is also a common byzantine consensus threshold for resistance to failures in a typical distributed system [60]. In data poisoning attacks, the results consistently demonstrate that FEDQV outperforms FEDAVG, achieving the highest ACC with the smallest standard error. When considering targeted attacks, FEDQV again stands out, displaying the highest ACC along with the lowest ASR when compared to FEDAVG. In the context model poisoning attacks, FEDQV outperforms FEDAVG, except for the QV-Adaptive attack, which is tailored for FEDQV. Especially for local model poisoning attacks: Trim and Krum attacks, FEDQV outperforms FEDAVG by at least 4 times in terms of accuracy. Setting this observation as the alternative hypothesis H_1 and using the Wilcoxon signed-rank test, we can reject the null hypothesis H_0 at a confidence level of 1% in favour of H_1, proving the robustness of FEDQV.

Varying the percentage of attackers: Then we examine the performance of our method as the proportion of attackers increases. Figure 4 shows the changes in performance metrics for varying percentages of attackers for both FEDQV and FEDAVG under the backdoor attack (depicted in Figure 4a) and Neurotoxin attack (shown in Figure 4b) for both FEDQV and FEDAVG. Across scenarios where the percentage of attackers changes from 10% to 50%, FEDQV consistently outperforms FEDAVG in terms of both ACC and ASR under both attacks, even in scenarios where half the parties are malicious. Notably, the Neurotoxin attack exhibits a more pronounced impact on ACC reduction and ASR increase in both FEDQV and FEDAVG compared to the backdoor attack as the percentage of attackers varies.

Unlike untargeted attacks, strong targeted attacks can be mounted with just a single attacker and data poisoning [61]. To investigate the behaviour of FEDQV in scenarios with ner gradations, we also evaluate it with small, realistic percentages of attackers, same as in [61], in Table 2 and Appendix Table A2. The results indicate that under these small, realistic percentages of attackers, the performance of FEDQV remains consistent with its behaviour under larger attack scenarios consistently outperforming FEDAVG. This outcome underscores the robustness of FEDQV across varying threat levels.

	MNIST		Fashion-MNIST		CIFAR10		FEMNIST	
	FEDAVG	FEDQV	FEDAVG	FEDQV	FEDAVG	FEDQV	FEDAVG	FEDQV
Data Poison								
Label ip	98.81 0.03	98.54 0.05	86.70 0.02	85.22 0.05	66.88 0.48	67.36 0.22	74.92 2.55	78.42 0.65
Gaussian	9.68 0.41	10.49 0.46	10.00 0.00	27.38 17.38	15.29 0.57	19.76 3.66	4.64 0.13	4.83 0.25
Backdoor								
ACC	37.38 19.82	98.30 0.15	74.27 9.12	78.40 3.95	59.85 2.18	60.65 1.72	49.78 22.38	75.20 3.96
ASR	68.49 22.00	0.19 0.07	14.58 12.53	7.05 6.35	18.20 5.27	3.21 1.30	30.88 7.52	28.26 9.57
Scaling								
ACC	10.33 0.05	11.16 0.88	10.22 0.09	11.27 0.99	10.00 0.00	28.55 18.55	26.30 21.55	64.80 1.38
ASR	99.94 0.06	98.96 1.04	99.74 0.10	98.21 1.45	100.00 0.00	67.66 32.34	0.47 0.08	0.56 0.06
Neurotoxin								
ACC	81.17 15.39	95.73 1.45	70.00 7.85	79.58 1.60	22.40 7.16	45.40 3.22	47.29 18.07	79.99 0.70
ASR	23.19 2.25	18.11 1.67	20.65 2.21	18.12 4.16	51.63 1.03	57.42 1.91	40.42 4.35	9.00 1.29
Model Poison								
Krum	10.57 0.39	97.96 0.14	10.00 0.00	79.43 0.86	10.00 0.00	53.27 1.12	5.20 0.22	51.86 3.06
Trim	10.04 0.16	98.36 0.11	10.00 0.00	84.45 0.70	10.00 0.00	57.33 2.34	5.09 0.33	52.19 4.52
Min-Max	35.00 25.38	85.32 6.45	10.00 0.00	67.25 7.44	10.00 0.00	19.07 6.97	56.37 13.67	72.58 2.11
Min-Sum	96.69 0.94	95.97 0.59	10.88 0.87	83.93 0.81	17.40 4.27	43.94 3.56	52.56 23.91	72.36 1.61
QV-Adaptive	71.43 22.67	56.94 23.95	35.92 4.60	62.13 11.25	10.00 0.00	11.14 1.14	22.08 18.72	43.78 20.72

Table 1: Comparison of FEDQV and FEDAVG on four benchmark datasets under 10 attack scenarios with 30% malicious parties. The best results are highlighted in bold.

(a) Backdoor attack

(b) Neurotoxin attack

Figure 4: Performance comparison of FEDQV and FEDAVG in terms of Accuracy (ACC) and Attack Success Rate (ASR) over 100 epochs across four benchmark datasets under two targeted attacks, with varying percentages of attackers m from 10% to 50%.

However, we notice that none of these methods yields satisfactory accuracy results for Gaussian and Scaling attacks. To address this, we present the enhanced version of FEDQV with an adaptive budget assigned according to a reputation model.

5.7 Adaptive Budget

To enhance the robustness of FEDQV, we integrate it with the reputation model proposed in [13] to allocate unequal budgets based on the reputation scores of parties in each round, as detailed in Section 3.4. The performance of FEDQV with an adaptive budget (referred to as FEDQV+REP) is compared with FEDAVG and FEDQV during the Gaussian and Scaling attacks, with the percentage of attackers increased to 50%. As depicted in Figure 5a, the combination of FEDQV and the reputation model significantly enhances resistance against Gaussian and Scaling attacks by at least a factor of 26%. This observation, formulated as the alternative hypothesis H_1, is supported by the Wilcoxon signed-rank test, where the null hypothesis H_0 can be rejected at a confidence level of 1% in favour of H_1. This integration effectively enhances the robustness of FEDQV, rendering it successful in defending against the two attacks that it previously failed to counter.

5.8 Integration with Byzantine-robust Aggregation

Our objective is not to position FedQV in competition with existing defence techniques but rather to demonstrate that FedQV can act as a complementary approach to advanced defences. Hence, FEDQV can be seamlessly integrated into Byzantine-robust defence to enhance the overall defence performance, by adapting the weight calculation process. We illustrate this with examples using Muilt-Krum [4], Trim-mean [12] and Reputation [13].

(a) Performance comparison of FEDAVG, FEDQV, and FEDQV+REP, in terms of ACC for 100 epochs in four benchmark datasets under 2 attack scenarios with 50% malicious parties.

(b) Test accuracy (ACC) for 100 epochs of both Krum alone and Multi-Krum + FEDQV on two benchmark datasets under 4 untargeted attack scenarios with 30% malicious parties.

Figure 5: Performance comparison of FEDAVG, FEDQV, and FEDQV+REP (FEDQV with adaptive budgets based on reputation model). Performance comparison of Krum alone and Multi-Krum integrated with FEDQV.

	Multi-Krum	Multi-Krum + FedQV	Trimmed-Mean	Trimmed-Mean + FedQV	Rep	Rep + FedQV
Neurotoxin						
1%	78.99 1.03/1.05 0.01	80.61 0.66/0.86 0.17	85.23 1.77/0.59 0.13	84.75 0.84/0.45 0.06	80.99 1.15/0.84 0.32	85.82 0.55/0.38 0.06
5%	76.21 0.73/3.29 0.92	80.32 1.07/1.34 0.34	85.15 0.38/0.87 0.17	85.09 0.97/0.73 0.05	80.48 1.17/1.62 0.11	84.12 0.38/1.35 0.40
10%	72.79 1.02/21.73 7.07	77.41 1.36/16.30 3.01	85.13 0.70/2.32 0.27	84.68 0.73/1.45 0.21	80.86 0.86/1.30 0.10	83.78 0.09/0.66 0.04
Min-Max						
10%	71.62 4.48	79.48 0.82	75.41 0.77	78.46 0.67	72.66 1.34	76.98 1.48
30%	52.29 0.34	58.42 4.92	59.62 1.20	60.24 6.81	54.94 0.82	58.02 0.71
50%	10.28 0.28	22.95 9.94	9.47 0.42	10.64 0.63	11.23 1.00	13.64 2.58
QV-Adaptive						
10%	52.98 1.78	73.35 3.44	83.17 1.85	85.55 0.33	12.60 2.36	41.55 19.78
30%	34.93 14.60	55.07 11.40	29.14 19.14	42.17 25.95	12.44 2.44	38.44 14.32
50%	10.20 0.20	12.24 1.12	10.00 0.00	13.35 3.37	10.00 0.00	10.55 0.44

Table 2: Comparison of Multi-Krum, Trimmed-Mean, Reputation, and their integration with FEDQV under three State-of-the-Art attacks on FEMNIST dataset. The best results are in bold. The results of targeted attacks are in the form of “ACC / ASR".

Table 2 presents a comparative analysis illustrating the performance of Multi-Krum, Trimmed-Mean, and Reputation methods both individually and when integrated with FEDQV under three state-of-the-art attacks conducted on the FEMNIST dataset. The results highlight that the integration of FEDQV with these defense mechanisms consistently yields superior performance characterised by higher ACC and lower ASR compared to their standalone counterparts.

Especially, in the context of Multi-Krum, as illustrated in Figure 5b, the integration of FEDQV (referred to as Multi-Krum + FEDQV) results in a signi cant enhancement in accuracy under 4 attack scenarios. Notably, under local poisoning attacks such as the Krum attack and Trim attack, the accuracy improves by a factor of 2.5. The results presented in Table 6a elucidate that the incorporation of FEDQV into Multi-Krum yields an average ACC enhancement of 26.72% and a signi cant average reduction of 70% in the ASR under two targeted attack scenarios: backdoor and scale attack. Setting this observation as the alternative hypothesis (H_1) and Employing the Wilcoxon signed-rank test, we reject the null hypothesis (H_0) at a con dence level of 1% in favour of H_1.

These ndings underscore the potential of FEDQV as a promising complementary method to augment the performance of existing defence mechanisms.

5.9 Non-IID Degree

To ful l the fundamental setting of a heterogeneous and unbalanced dataset for FL, our experimental setup integrates datasets exhibiting non-IID properties, with a non-IID degree ( ) set to 0.9, consistent with the settings outlined in previous studies such as [52, 13]. The represents the concentration parameter utilised in sampling from a Dirichlet distribution [62], which governs the degree of dataset imbalance. This parameter choice aims to generate non-IID and unbalanced data partitions conducive to our experimental objectives.

Moreover, we have examined the performance of FEDQV and FEDAVG across varying degrees of non-IID data, ranging from 0.1 to 0.9, as depicted in Appendix Table A3. These results demonstrate that as the non-IID degree increases among the parties, the performance of the global model declines. Notably, FEDQV consistently maintains a superior performance compared to FEDAVG, even when confronted with different degrees of data heterogeneity under attack conditions.

	MNIST				Fashion-MNIST			
	Multi-Krum		+ FEDQV		Multi-Krum		+ FEDQV	
Backdoor								
ACC	70.20	9.99	89.96	1.85	33.24	13.24	70.89	3.17
ASR	32.03	11.20	9.59	2.28	68.87	17.77	9.72	4.50
Scaling								
ACC	68.35	16.76	96.55	0.41	59.43	14.22	82.48	0.24
ASR	33.65	19.15	0.41	0.06	33.64	19.08	0.91	0.18

(a) Comparison of Multi-Krum and Multi-Krum + FEDQV under two targeted attacks with 30% malicious parties in MNIST and Fashion-MNIST dataset. The best results are in bold.

(b) Impact of Hyperparameters B and α of FEDQV on ACC and ASR under a backdoor attack scenario with 30% malicious parties, using the MNIST dataset.

Figure 6: Comparison of Multi-Krum and Multi-Krum + FEDQV under two targeted attacks, and examining the impact of hyperparameters.

(a) The Deep Leakage from Gradients (DLG) Attack.

(b) The Gradient Invention (GI) Attack.

Figure 7: Visual Comparison of Images Reconstructed by Privacy Attacks (DLG and GI) on FEDQV with and without SECAGG. The first row displays the original private training images from the targeted client. The second row illustrates reconstructed images resulting from the privacy attacks (DLG and GI) on FEDQV without SECAGG. In the third row, reconstructed images from the privacy attacks (DLG and GI) on FEDQV with SECAGG are presented. The use of SECAGG is observed to contribute significantly to mitigating information leakage from the images.

5.10 Impact of Hyperparameters

As noted, Theorem 4.10 provides general guidelines for hyperparameters tuning. As shown in Remark 4.13, the error rate is influenced by B and α. To demonstrate the impact of these two hyper-parameters, we conduct a grid search over B in $[10, 20, 30, 40, 50]$ and α in $[0.1, 0.2, 0.3, 0.4, 0.5]$. The setup is the same as on the MNIST dataset under the backdoor attack with 30% malicious parties. Figure 6b illustrates the stability of the metrics ACC and ASR when varying the hyperparameters B and α. Specifically, as B increases, there is a discernible decline in ACC accompanied by an increase in ASR, attributable to the augmented voting budget allotted to malicious parties. Conversely, an increase in α yields an elevation in ACC coupled with a reduction in ASR, as a higher threshold implies stricter acceptance criteria for abnormal similarity scores. The optimal values of B and α are determined to be 30 and 0.2, respectively.

We can see from Theorem 4.10, that the number of malicious devices m will affect the algorithm, and more malicious devices can lead to increased damage. However, this does mean the server needs to know the number of malicious devices to do the fine-tuning. We agree that determining optimal parameters can be challenging, especially in the absence of complete knowledge about the FL system. A better tuning is possible if more information is available. For specific tasks, more information can indeed be collected from which practical parameter sets can be extracted either via exhaustive search or via simpler online algorithms using trial and error. We will add this to our future work and consider it when we study particular domain-specific problems using our method.

6 Defence against Privacy Attacks

To illustrate the compatibility of FEDQV with existing privacy-guaranteeing mechanisms, we integrate SECAGG [11] into the FEDQV framework. Subsequently, we assess the framework's resilience against privacy attacks, specifically the Deep Leakage from Gradients (DLG) [8] and Gradient Inversion (GI) [9] attacks, with and without the incorporation of SECAGG. Figure 7a provides visualisations of the reconstructed images by the DLG attack. In the absence of SECAGG, the DLG attack can successfully recover approximately 8 out of 20 private images from the party. However, upon integrating SECAGG, the DLG attack fails to recover any of the party's images. Figure 7b presents visualisations of the reconstructed images by the GI attack. Without SECAGG, the GI attack successfully recovers half of the private images from the party. Yet, with SECAGG, the GI attack is unable to recover any party images, although it may vaguely leak the overall shape of the image. These results highlight the efficacy of SECAGG in strengthening FEDQV against privacy attacks, with the GI attack exhibiting a higher level of attack potency compared to the DLG attack. More details regarding privacy attacks and implementations are provided in Appendix C.

7 Conclusion

In this paper, we have proposed FEDQV, a novel aggregation scheme for FL based on quadratic voting instead of 1p1v, which is the underlying principle that makes the currently employed FEDAVG vulnerable to poisoning attacks. The proposed method aggregates local models based upon the votes from a truthful mechanism employed in FEDQV. The efficiency of the proposed method has been comprehensively analysed from both a theoretical and an experimental point of view. Collectively, our performance evaluation has shown that FEDQV achieves superior performance than FEDAVG in defending against various poisoning attacks. Moreover, FEDQV is a reusable module that can be integrated with reputation models to assign unequal voting budgets, incorporate Byzantine-robust techniques, and employ privacy-preserving mechanisms. This versatility provides robustness against both poisoning and privacy attacks, positioning FEDQV as a promising complement to existing aggregation in FL.

Acknowledgement

Tianyue Chu and Nikolaos Laoutaris were supported by the MLEDGE project (REGAGE22e00052829516), funded by the Ministry of Economic Affairs and Digital Transformation and the European Union NextGenerationEU/PRTR.

References

[1] 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.

[2] Swaroop Ramaswamy, Rajiv Mathews, Kanishka Rao, and Françoise Beaufays. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329*, 2019.

[3] Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *CoRR*, abs/1811.03604, 2018.

[4] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 119–129, Long Beach, CA, USA, 2017. Curran Associates, Inc.

[5] Steven P Lalley and E Glen Weyl. Quadratic voting: How mechanism design can radicalize democracy. In *AEA Papers and Proceedings*, volume 108, pages 33–37, 2018.

[6] Eric A Posner and E Glen Weyl. Voting squared: Quadratic voting in democratic politics. *Vand. L. Rev.*, 68:441, 2015.

[7] E Glen Weyl. The robustness of quadratic voting. *Public choice*, 172(1):75–107, 2017.

[8] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, Vancouver Convention Center, Vancouver CANADA, 2019. Curran Associates, Inc.

[9] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

[10] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In 2019 IEEE symposium on security and privacy (SP), pages 691–706. IEEE, 2019.

[11] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17, page 1175–1191, New York, NY, USA, 2017. Association for Computing Machinery.

[12] 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.

[13] Tianyue Chu, Álvaro García-Recuero, Costas Iordanou, Georgios Smaragdakis, and Nikolaos Laoutaris. Securing federated sensitive topic classification against poisoning attacks. In 30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, 2023. The Internet Society.

[14] Hector Garcia-Molina. Elections in a distributed computing system. IEEE transactions on Computers, 31(01):48–59, 1982.

[15] Mack W Alford, Jean-Pierre Ansart, Günter Hommel, Leslie Lamport, Barbara Liskov, Geoff P Mullery, and Fred B Schneider. Distributed systems: methods and tools for specification. An advanced course. Springer, Verlag, 1985.

[16] Kai Yue, Richeng Jin, Chau-Wai Wong, and Huaiyu Dai. Federated learning via plurality vote. IEEE Transactions on Neural Networks and Learning Systems, pages 1–14, 2022.

[17] Yinghao Liu, Zipei Fan, Xuan Song, and Ryosuke Shibasaki. Fedvoting: A cross-silo boosting tree construction method for privacy-preserving long-term human mobility prediction. Sensors, 21(24):8282, 2021.

[18] Jy-yong Sohn, Dong-Jun Han, Beongjun Choi, and Jaekyun Moon. Election coding for distributed learning: Protecting signsgd against byzantine attacks. Advances in Neural Information Processing Systems, 33:14615–14625, 2020.

[19] Shashank Rajput, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos. Detox: A redundancy-based framework for faster and more robust gradient aggregation. Advances in Neural Information Processing Systems, 32, 2019.

[20] Lingjiao Chen, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos. Draco: Byzantine-resilient distributed training via redundant gradients. In Proceedings of the 35th International Conference on Machine Learning, ICML 2018, volume 80 of Proceedings of Machine Learning Research, pages 902–911, Stockholmsmässan, Stockholm, Sweden, 2018. PMLR.

[21] Konstantinos Konstantinidis and Aditya Ramamoorthy. Byzshield: An efficient and robust system for distributed training. Proceedings of Machine Learning and Systems, 3:812–828, 2021.

[22] Giovanni Sartori. The theory of democracy revisited, volume 2. NJ, 1987.

[23] Steven P Lalley, E Glen Weyl, et al. Quadratic voting. Available at SSRN, 2016.

[24] Bharat Chandar and E Glen Weyl. Quadratic voting in finite populations. Available at SSRN: https://ssrn.com/abstract=2571026 or http://dx.doi.org/10.2139/ssrn.2571026, 2016.

[25] Nicolaus Tideman and Florenz Plassmann. Efficient collective decision-making, marginal cost pricing, and quadratic voting. Public Choice, 172(1):45–73, 2017.

[26] Alessandra Casella and Luis Sanchez. Storable votes and quadratic voting. an experiment on four california propositions. Technical report, National Bureau of Economic Research, 2019.

[27] David Quarfoot, Douglas von Kohorn, Kevin Slavin, Rory Sutherland, David Goldstein, and Ellen Konar. Quadratic voting in the wild: real people, real votes. Public Choice, 172(1):283–303, 2017.

[28] 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.

[29] Hanxi Guo, Hao Wang, Tao Song, Yang Hua, Zhangcheng Lv, Xiulang Jin, Zhengui Xue, Ruhui Ma, and Haibing Guan. Siren: Byzantine-robust federated learning via proactive alarming. In Proceedings of the ACM Symposium on Cloud Computing, SoCC '21, page 47–60, New York, NY, USA, 2021. Association for Computing Machinery.

[30] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In 28th Annual Network and Distributed System Security Symposium, NDSS 2021, February 21-25, 2021, virtually, 2021. The Internet Society.

[31] Mohamad Mansouri, Melek Önen, Wafa Ben Jaballah, and Mauro Conti. Sok: Secure aggregation based on cryptographic schemes for federated learning. Proc. Priv. Enhancing Technol., 2023.

[32] Slawomir Goryczka, Li Xiong, and Vaidy S. Sunderam. Secure multiparty aggregation with differential privacy: a comparative study. In EDBT/ICDT Conferences, EDBT/ICDT, New York, NY, USA, 2013. Association for Computing Machinery.

[33] Lingchen Zhao, Jianlin Jiang, Bo Feng, Qian Wang, Chao Shen, and Qi Li. SEAR: secure and ef cient aggregation for byzantine-robust federated learning. IEEE Trans. Dependable Secur. Comput., 2022.

[34] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography from anonymity. In Annual IEEE Symposium on Foundations of Computer Science (FOCS). IEEE Computer Society, 2006.

[35] Georgia Tsaloli, Bei Liang, Carlo Brunetta, Gustavo Banegas, and Aikaterini Mitrokotsa. sfDEVA: decentralized, veri able secure aggregation for privacy-preserving learning. In Information Security - International Conference, ISC, Lecture Notes in Computer Science. Springer, 2021.

[36] Danye Wu, Miao Pan, Zhiwei Xu, Yujun Zhang, and Zhu Han. Towards ef cient secure aggregation for model update in federated learning. In IEEE Global Communications Conference, GLOBECOM. IEEE, 2020.

[37] Xu Ma, Yuqing Zhou, Laihua Wang, and Meixia Miao. Privacy-preserving byzantine-robust federated learning. Computer Standards & Interfaces, 80:103561, 2022.

[38] Jinhyun So, Başak Güler, and A Salman Avestimehr. Byzantine-resilient secure federated learning. IEEE J. on Selected Areas in Communications, 39(7):2168–2181, 2020.

[39] Felix Marx, Thomas Schneider, Ajith Suresh, Tobias Wehrle, Christian Weinert, and Hossein Yalame. Hy : A hybrid approach for private federated learning. arXiv preprint arXiv:2302.09904, 2023.

[40] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In International Conference on Learning Representations, 2018.

[41] Liad Blumrosen and Noam Nisan. Algorithmic game theory. Springer, Berlin, Heidelberg, 2007.

[42] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Sparse binary compression: Towards distributed deep learning with minimal communication. In 2019 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2019.

[43] Cynthia Dwork. Differential privacy. In Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II, pages 1–12, Berlin, Heidelberg, 2006. Springer.

[44] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In Proceedings of the AAAI Conference on Arti cial Intelligence, volume 33,01, pages 5693–5700, 2019.

[45] X. Cao, J. Jia, Z. Zhang, and N. Gong. Fedrecover: Recovering from poisoning attacks in federated learning using historical information. In 2023 IEEE Symposium on Security and Privacy (SP), pages 1366–1383, Los Alamitos, CA, USA, may 2023. IEEE Computer Society.

[46] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In International Conference on Learning Representations (ICLR), 2020.

[47] Yann LeCun. The mnist database of handwritten digits. http://yann. lecun. com/exdb/mnist, 1998.

[48] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.

[49] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. LEAF: A benchmark for federated settings. CoRR, abs/1812.01097, 2018.

[50] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pages 770–778, Las Vegas, NV, USA, 2016. IEEE Computer Society.

[52] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In Silvia Chiappa and Roberto Calandra, editors, Proceedings of the Twenty Third International Conference on Arti cial Intelligence and Statistics, volume 108 of Proceedings of Machine Learning Research, pages 2938–2948. PMLR, 26–28 Aug 2020.

[53] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. CoRR, abs/1909.06335, 2019.

[54] 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, Virtual, 2020. USENIX Association.

[55] Bo Zhao, Peng Sun, Tao Wang, and Keyu Jiang. Fedinv: Byzantine-robust federated learning by inversing local model updates. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 36(8), 9171–9179, 2022.

[56] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. IEEE Access, 7:47230–47244, 2019.

[57] Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael Mahoney, Prateek Mittal, Ramchandran Kannan, and Joseph Gonzalez. Neurotoxin: Durable backdoors in federated learning. In International Conference on Machine Learning (ICML), pages 26429–26446. PMLR, 2022.

[58] Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In 28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021. The Internet Society, 2021.

[59] Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models. In 30th USENIX Security Symposium (USENIX Security 21), pages 1505–1521, virtually, 2021. USENIX Association.

[60] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance and proactive recovery. ACM Trans. Comput. Syst., 20(4):398–461, November 2002.

[61] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In 2022 IEEE Symposium on Security and Privacy (SP), pages 1354–1371. IEEE, 2022.

[62] Thomas Minka. Estimating a dirichlet distribution, 2000.

[63] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. Foundations and Trends® in Machine Learning, 8(3-4):231–357, 2015.

[64] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In NIPS 2017 Workshop on Autodiff, 2017.

Appendix

A Proof of Theoretical Analysis

A.1 Proof of Theorem 4.9 and Theorem 4.10

A.1.1 Proof of Lemmas

Lemmas 4.5, Lemmas 4.6, Lemmas 4.7 and Lemmas 4.8 are all the lemmas we utilise during the proof of Theorem 4.9, and we prove them in that order. Notice, Lemmas 4.5 are used in the proof Lemmas 4.6, and Theorem 4.9 is proved using Lemmas 4.6, Lemmas 4.7 and Lemmas 4.8 in order.

Proof of Lemma 4.5

Proof. Let $g(w) = \ell(w) - \frac{\alpha}{2}\|w\|_2^2$. Base on the Assumption 4.2, we have $g(w)$ is $(L - \alpha)$-strongly convex. from [63] Equation 3.6, we have

$$\langle \nabla \ell(w_1) - \nabla \ell(w_2), w_1 - w_2 \rangle \geq \frac{1}{L} \|\nabla \ell(w_1) - \nabla \ell(w_2)\|_2^2 \tag{12}$$

Hence,

$$\langle \nabla g(w_1) - \nabla g(w_2), w_1 - w_2 \rangle \geq \frac{1}{L - \alpha} \|\nabla g(w_1) - \nabla g(w_2)\|_2^2 \tag{13}$$

Now We have

$$\begin{aligned} hr\ `(w_1)\ \ \frac{\&}{2}kw_1k_2^2\ \ r\ \ `(w_2)\ \ \frac{\&}{2}kw_2k_2^2\ ;w_1\ \ w_2i \\ \frac{1}{L+}\ \ r\ \ `(w_1)\ \ \frac{\&}{2}kw_1k_2^2\ \ r\ \ `(w_2)\ \ \frac{\&}{2}kw_2k_2^2\ \ ^2_2 \end{aligned} \tag{14}$$

And therefore

$$\begin{aligned} hr\ `(w_1)\ \ r\ \ `(w_2);w_1\ \ w_2i\ \ h\ \&w_1\ \ \&w_2;w_1\ \ w_2i \\ \frac{1}{L\ \ \&}k(r\ `(w_1)\ \ r\ \ `(w_2))\ \ (\&w_1\ \ \&w_2)k_2^2 \end{aligned} \tag{15}$$

Refer to Assumption 4.1, we obtain

$$\begin{aligned} hr\ `(w_1)\ \ r\ \ `(w_2);w_1\ \ w_2i\ \ \frac{L\&}{L\ \ \&}kw_1\ \ w_2k_2^2\ \ \frac{2\&}{L\ \ \&}hr\ `(w_1)\ \ r\ \ `(w_2);w_1\ \ w_2i \\ +\ \frac{1}{L\ \ \&}kr\ `(w_1)\ \ r\ \ `(w_2)k_2^2 \\ \frac{L\&}{L\ \ \&}kw_1\ \ w_2k_2^2+\frac{1}{L\ \ \&}kr\ `(w_1)\ \ r\ \ `(w_2)k_2^2 \end{aligned} \tag{16}$$

Let $\&=$, then we conclude the proof of Lemma 4.5.

□

Proof of Lemma 4.6

Proof. We have

$$w^{t\ \ 1}\ \ r_{t\ \ 1}rL\ (w^{t\ \ 1})\ \ w\ \ ^2_2 = \underbrace{w^{t\ \ 1}\ \ w\ \ ^2_2}_{A1} + \underbrace{r^2_{t\ \ 1}\ rL\ (w^{t\ \ 1})\ \ ^2_2}_{A2} \tag{17}$$

For partA1 under the Assumption 4.2, Lemma 4.5 and Maclaurin inequality, we have

$$\begin{aligned} A1 &= 2r_{t\ \ 1}\sum_{i=1}^{N} p_i^{t\ \ 1}\ r\ `(w_i^{t\ \ 1});w^{t\ \ 1}\ \ w \\ &= 2r_{t\ \ 1}\sum_{i=1}^{N} p_i^{t\ \ 1}\ \ r\ `(w_i^{t\ \ 1});w^{t\ \ 1}\ \ w_i^{t\ \ 1} \\ &\quad 2r_{t\ \ 1}\sum_{i=1}^{N} p_i^{t\ \ 1}\ \ r\ `(w_i^{t\ \ 1});w_i^{t\ \ 1}\ \ w \\ &\quad \sum_{i=1}^{N} p_i^{t\ \ 1}\ r_{t\ \ 1}^2\ r\ `(w_i^{t\ \ 1})\ ^2_2+\ w^{t\ \ 1}\ \ w_i^{t\ \ 1}\ ^2_2 \\ &\quad 2r_{t\ \ 1}\sum_{i=1}^{N} p_i^{t\ \ 1}\ \ \frac{1}{L+}\ \ r\ `(w_i^{t\ \ 1})\ ^2_2+\frac{L}{L+}\ \ w_i^{t\ \ 1}\ \ w\ \ ^2_2 \\ &= r_{t\ \ 1}^2\ \ \frac{1}{L+}\sum_{i=1}^{N} p_i^{t\ \ 1}\ \ r\ `(w_i^{t\ \ 1})\ ^2_2 \\ &\quad +\sum_{i=1}^{N} p_i^{t\ \ 1}\ w^{t\ \ 1}\ \ w_i^{t\ \ 1}\ ^2_2\ \ \frac{2r_{t\ \ 1}L}{L+}\ \ w^{t\ \ 1}\ \ w\ \ ^2_2 \end{aligned}$$

From Assumption 4.1 and Jensen inequality, we can derive:

$$\|\nabla(w_i^{t-1}) - \nabla(w^*)\|_2^2 \leq L^2 \|w_i^{t-1} - w^*\|_2^2 \tag{18}$$

Hence forA_1, by Jensen inequality and Equation 18, we have

$$\begin{aligned}
A_1 \leq\ & r_{t-1}^2 \frac{1}{L+\mu} \sum_{i=1}^N p_i^{t-1} \|\nabla(w_i^{t-1})\|_2^2 \\
& + \sum_{i=1}^N p_i^{t-1} \|w^{t-1} - w_i^{t-1}\|_2^2 \frac{2r_{t-1}L}{L+\mu} \|w^{t-1} - w^*\|_2^2 \\
& r_{t-1}^2 \frac{1}{L+\mu} \sum_{i=1}^N p_i^{t-1} \|w_i^{t-1} - w^*\|_2^2 \\
& + \sum_{i=1}^N p_i^{t-1} \|w^{t-1} - w_i^{t-1}\|_2^2 \frac{2r_{t-1}L}{L+\mu} \|w^{t-1} - w^*\|_2^2 \\
& r_{t-1}^2 \frac{2r_{t-1}L+1}{L+\mu} \|w^{t-1} - w^*\|_2^2 \\
& + \sum_{i=1}^N p_i^{t-1} \|w^{t-1} - w_i^{t-1}\|_2^2
\end{aligned}$$

Similar forA_2, we have

$$\begin{aligned}
A_2 = r_{t-1}^2 \Big\| \sum_{i=1}^N p_i^{t-1} \nabla(w_i^{t-1}) \Big\|_2^2 & \leq r_{t-1}^2 \sum_{i=1}^N p_i^{t-1} \|\nabla(w_i^{t-1})\|_2^2 \\
r_{t-1}^2 L^2 \sum_{i=1}^N p_i^{t-1} \|w_i^{t-1} - w^*\|_2^2 & = r_{t-1}^2 L^2 \|w^{t-1} - w^*\|_2^2
\end{aligned}$$

Then we combine results of$A_1$and$A_2$for Equation 17, it follows that

$$\begin{aligned}
\|w^{t-1} - r_{t-1}\nabla L(w^{t-1}) - w^*\|_2^2 \leq\ & r_{t-1}^2 \left(1 + L^2 \frac{2r_{t-1}L+1}{L+\mu}\right) \|w^{t-1} - w^*\|_2^2 \\
& + \sum_{i=1}^N p_i^{t-1} \|w^{t-1} - w_i^{t-1}\|_2^2
\end{aligned} \tag{19}$$

□

Proof of Lemma 4.7

Proof. For each$E$ step FL necessitates a communication. As a result, for$t$any$_1 \geq 0$, $\exists t \geq t_1$ that $t \bmod E = 0$; $t \geq T$, accordingly$\forall i, j \in S^t$; $w_i^t = w_j^t = w^t$. Then, based on$\mathbb{E}\|X - \mathbb{E}X\|_2^2 = \mathbb{E}\|X\|_2^2 - \|\mathbb{E}X\|_2^2$, Jensen inequality and Assumption 4.3, we have

$$
\begin{aligned}
E\Big\|\sum_{i=1}^{N} p_i^{t-1}\big(w^{t-1}-w_i^{t-1}\big)\Big\|_2^2 &= E_{S^t}\Big\|\sum_{i\in S^t} p_i^{t-1}\big(w_i^{t-1}-w^t+w^t-w^{t-1}+w^{t-1}\big)\Big\|_2^2 \\
&= E_{S^t}\Big\|E_{S^t}\big[w_i^{t-1}-w^t\big]-E_{S^t}\big[w_i^{t-1}-w^t\big]\Big\|_2^2 \\
&\quad - E_{S^t}\Big\|E_{S^t}\big[w_i^{t-1}-w^t\big]\Big\|_2^2 \\
&= E_{S^t}\Big\|\sum_{i\in S^t} p_i^{t-1}\big(w_i^{t-1}-w^t\big)\Big\|_2^2 \\
&= E_{S^t}\Big\|\sum_{i\in S^t} p_i^{t-1}\Big(\sum_{t=t'}^{t-1}\eta\, r\,\tilde{\nabla}(w_i^{t-1};D_i^{t-1})\Big)\Big\|_2^2 \\
&\le \sum_{i\in S^t} p_i^{t-1}E_{S^t}\big\|(t-1-t')\sum_{t=t'}^{t-1}\eta_{t-1}^2\, r\,\tilde{\nabla}(w_i^{t-1};D_i^{t-1})\big\|_2^2 \\
&\le \sum_{i\in S^t} p_i^{t-1}(E-1)\sum_{t=t'}^{t-1}\eta_{t-1}^2\, r\,\tilde{\nabla}(w_i^{t-1};D_i^{t-1})\big\|_2^2 \\
&\le \sum_{i\in S^t} p_i^{t-1}(E-1)\sum_{t=t'}^{t-1}\eta_{t-1}^2 G_w^2 \\
&\le \sum_{i\in S^t} p_i^{t-1}(E-1)^2\eta_{t-1}^2 G_w^2 \\
&\le (E-1)^2\eta_{t-1}^2 G_w^2
\end{aligned}
\tag{20}
$$

□

Proof of Lemma 4.8

Proof. Due to Assumption 4.4 and Algorithm 1, we have

$$
\begin{aligned}
E\big\|F(w^{t-1})-\nabla L(w^{t-1})\big\|_2^2 = \mathrm{Var}\big[F(w^{t-1})\big] &= E_{S^{t-1}}\Big\|\sum_{i\in S^{t-1}} p_i^{t-1}\big(\tilde{\nabla}(w_i^{t-1};D_i^{t-1})-\nabla(w_i^{t-1})\big)\Big\|_2^2 \\
&= \sum_{i\in S^{t-1}} p_i^{t-1\,2}E\big\|\tilde{\nabla}(w_i^{t-1};D_i^{t-1})-\nabla(w_i^{t-1})\big\|_2^2 \\
&\le \sum_{i\in S^{t-1}} p_i^{t-1\,2}V_w \le V_w\Big(\sum_{i\in S^{t-1}}\frac{v_i^{t-1}}{\sum_{i\in S^{t-1}}v_i^{t-1}}\Big)^2 \\
&\le V_w\frac{\sum_{i\in S^{t-1}}v_i^{t-1\,2}}{\big(\sum_{i\in S^{t-1}}v_i^{t-1}\big)^2} \le V_w\frac{\sum_{i\in S^{t-1}}v_i^{t-1\,2}}{\max_{i\in S^{t-1}}v_i^{t-1}} \\
&\le V_w\sum_{i\in S^{t-1}} v_i^{t-1} \le (1+\delta)qNV_w\bar{B}
\end{aligned}
\tag{21}
$$

□

A.1.2 Proof of Theorem 4.9

Proof. In t round, due to $m = 0$, we have:

$$\begin{aligned} \|w^{t} - w^{*}\|_2^2 &= \|w^{t-1} - r_{t-1}M(w^{t-1}) - w^{*}\|_2^2 = \|w^{t-1} - r_{t-1}F(w^{t-1}) - w^{*}\|_2^2 \\ &= \underbrace{\|w^{t-1} - r_{t-1}\nabla L(w^{t-1}) - w^{*}\|_2^2}_{A} + \underbrace{r_{t-1}^2\|F(w^{t-1}) - \nabla L(w^{t-1})\|_2^2}_{B} \\ &\quad + \underbrace{2r_{t-1}\langle w^{t-1} - r_{t-1}\nabla L(w^{t-1}) - w^{*}, F(w^{t-1}) - \nabla L(w^{t-1})\rangle}_{C} \end{aligned} \tag{22}$$

Where

$$M(w^{t-1}) = \sum_{i \in S^{t-1}} p_i^{t-1} M_i(w_i^{t-1})$$

Note that $\mathbb{E}C = 0$. For the expectation of A, from Lemma 4.6 and Lemma 4.7, it follows that

$$\begin{aligned} \mathbb{E}[A] &= \mathbb{E}\|w^{t-1} - r_{t-1}\nabla L(w^{t-1}) - w^{*}\|_2^2 \\ &\le r_{t-1}^2(1+L^2) - \frac{2r_{t-1}L}{L+\mu} \|w^{t-1} - w^{*}\|_2^2 \\ &\quad + (\mathbb{E}-1)^2 r_{t-1}^2 G_w^2 \end{aligned} \tag{23}$$

We use Lemma 4.8 to bound B, we have

$$\mathbb{E}[B] \le r_{t-1}^2(1-\mathbb{E})qNV_w\sqrt{B} \tag{24}$$

Hence, we have

$$\begin{aligned} \mathbb{E}\|w^{t} - w^{*}\|_2^2 &\le r_{t-1}^2(1+L^2)\,\mathbb{E}\|w^{t-1} - w^{*}\|_2^2 \\ &\quad - \frac{2r_{t-1}L}{L+\mu}\mathbb{E}\|w^{t-1} - w^{*}\|_2^2 + r_{t-1}^2 \end{aligned} \tag{25}$$

where

$$= (\mathbb{E}-1)^2 G_w^2 + (1-\mathbb{E})qNV_w\sqrt{B}$$

For the learning rate r_t, $\exists \gamma > \frac{L+\mu}{2L}; \exists \gamma > 0$, such that $r_t = \frac{\beta}{\gamma+t} - \frac{1}{L+1}$. We use mathematical induction to prove the following statement:

Proposition: $\forall t \in \mathbb{N}; \mathbb{E}\|w^{t} - w^{*}\|_2^2 \le \frac{\Omega}{\gamma+t}$, where $\Omega = \max\left\{\frac{(L+\mu)^2}{2\mu L - \mu^2}\beta^2, \mathbb{E}\|w^{0} - w^{*}\|_2^2\right\}$.

Let $P(t)$ be the statement $\mathbb{E}\|w^{t} - w^{*}\|_2^2 \le \frac{\Omega}{\gamma+t}$, we give a proof by induction on t.

Base case: The statement $P(0)$ holds for $t = 0$:

$$\mathbb{E}\|w^{0} - w^{*}\|_2^2 \le \Omega$$

Inductive step: Assume the induction hypothesis that for a particular, the single case $t = j$ holds, meaning $P(j)$ is true:

$$\mathbb{E}\|w^{j} - w^{*}\|_2^2 \le \frac{\Omega}{\gamma+j}$$

It follows that:

$$\begin{aligned} \mathbb{E}\|w^{j+1} - w^{*}\|_2^2 &\le r_t^2(1+L^2) - \frac{2r_tL}{L+\mu}\mathbb{E}\|w^{j} - w^{*}\|_2^2 + r_t^2 \\ &\le \left(1 - \frac{2L}{(L+\mu)(\gamma+j)}\right)\frac{\Omega}{\gamma+j} + \left(\frac{\beta}{\gamma+j}\right)^2 \\ &= \frac{\gamma-1}{(\gamma+j)^2}\Omega - \frac{2L-\mu}{(\gamma+j)^2(L+\mu)}\Omega + \frac{\beta^2}{(\gamma+j)^2} \\ &\le \frac{\Omega}{\gamma+j+1} \end{aligned}$$

Therefore, the statement $P(j+1)$ also holds true, establishing the inductive step. Since both the base case and the inductive step have been proved as true, by mathematical induction the statement $P(t)$ holds for $\forall t \in \mathbb{N}$.

We choose $= \frac{L+}{L}$ and $= 2\frac{(L+1)(L+)}{L}$, and we have

$$\begin{aligned} &= \max \quad \frac{(L+)^2}{2L \quad L}; \quad E\left\|w^0 \quad w\right\|_2^2 \\ &\quad \frac{(L+)^2}{2L \quad L} + \quad E\left\|w^0 \quad w\right\|_2^2 \\ &= \quad ^2 + 2(L+1) \quad E\left\|w^0 \quad w\right\|_2^2 \end{aligned}$$

Then based on Assumption 4.1 and Taylor expansion, we have the quadratic upper-bound of $L(w)$:

$$L(w_1) \quad L(w_2) \quad (w_1 \quad w_2)^T rL(w_2) + \frac{L}{2}\|w_1 \quad w_2\|_2^2$$

It follows that

$$\begin{aligned} EL(w^T) \quad L(w) \quad & \frac{L}{2}E\left\|w^T \quad w\right\|_2^2 \quad \frac{L}{2(+T)} \\ & \frac{L}{2(L+1)+T} \quad \frac{^2}{2} + \quad (L+1)E\left\|w^0 \quad w\right\|_2^2 \\ &= \frac{L}{2'+T} \quad ' E\left\|w^0 \quad w\right\|_2^2 + \frac{^2}{2} \end{aligned}$$

Where

$$= (E \quad 1)^2 G_w^2 + (1 \quad 2)CV_w \sqrt{B}; \quad ' = \quad (L+1)$$

□

A.1.3 Proof of Theorem 4.10

Proof. In the t round, we have:

$$\begin{aligned} \left\|w^t \quad w\right\|_2^2 &= \left\|w^{t-1} \quad r_{t-1}M(w^{t-1}) \quad w\right\|_2^2 \\ &= \left\|w^{t-1} \quad r_{t-1}F(w^{t-1}) \quad w + r_{t-1}F(w^{t-1}) \quad r_{t-1}M(w^{t-1})\right\|_2^2 \\ &= \underbrace{\left\|w^{t-1} \quad r_{t-1}F(w^{t-1}) \quad w\right\|_2^2}_{A} + \underbrace{r_{t-1}^2 \left\|F(w^{t-1}) \quad M(w^{t-1})\right\|_2^2}_{B} \\ &\quad \underbrace{+2r_{t-1}\left\langle w^{t-1} \quad r_{t-1}F(w^{t-1}) \quad w; F(w^{t-1}) \quad M(w^{t-1})\right\rangle}_{C} \end{aligned} \tag{26}$$

Where

$$M(w^{t-1}) = \sum_{i \in S^{t-1}} p_i^{t-1} M_i(w_i^{t-1})$$

For the expectation of A, from Theorem 4.9, it follows that

$$E[A] \quad \frac{1}{2'+t} \quad 2' E\left\|w^0 \quad w\right\|_2^2 + \quad ^2 \tag{27}$$

For B, we have

$$\begin{aligned} E[B] &= r_{t-1}^2 \Big\| \sum_{i \in S^{t-1}} p_i^{t-1} r'(w_i^{t-1}) - \sum_{i \in S^{t-1}} p_i^{t-1} M_i(w_i^{t-1}) \Big\|_2^2 \\ &= r_{t-1}^2 \Big\| \sum_{i \in S^{t-1}} p_i^{t-1} \left(r'(w_i^{t-1}) - M_i(w_i^{t-1}) \right) \Big\|_2^2 \\ &\leq r_{t-1}^2 \sum_{i \in mN} p_i^{t-1} \left\| r'(w_i^{t-1}) - M_i(w_i^{t-1}) \right\|_2^2 \end{aligned} \tag{28}$$

Where$m$ is the percentage of the malicious parties.

Due to Equation 3, we have

$$\frac{r'(w_i^{t-1}); M_i(w_i^{t-1})}{r'(w_i^{t-1} \quad M_i(w_i^{t-1})} \quad 1 \tag{29}$$

Thus,

$$r'(w_i^{t-1}) \quad M_i(w_i^{t-1}) \quad r'(w_i^{t-1}); M_i(w_i^{t-1}) \quad (1 \quad) \, r'(w_i^{t-1}) \quad M_i(w_i^{t-1}) \tag{30}$$

Due to this, we have

$$\begin{aligned} & r'(w_i^{t-1}) \|_2^2 \quad 2(1 \quad) \, r'(w_i^{t-1} \quad M_i(w_i^{t-1}) + M_i(w_i^{t-1}) \|_2^2 \\ & r'(w_i^{t-1}) \quad M_i(w_i^{t-1}) \|_2^2 \\ & r'(w_i^{t-1}) \|_2^2 \quad 2 \quad r'(w_i^{t-1}) \quad M_i(w_i^{t-1}) + M_i(w_i^{t-1}) \|_2^2 \end{aligned} \tag{31}$$

Hence we have

$$\begin{aligned} & (2 \quad) \, r'(w_i^{t-1}) \|_2^2 + (1 \quad) \, r'(w_i^{t-1}) \quad M_i(w_i^{t-1}) \|_2^2 \\ & r'(w_i^{t-1}) \quad M_i(w_i^{t-1}) \|_2^2 \\ & (1 \quad ^2) \, r'(w_i^{t-1}) \|_2^2 + \quad r'(w_i^{t-1}) \quad M_i(w_i^{t-1}) \|_2^2 \end{aligned} \tag{32}$$

Hence,

$$(2 \quad) \, r'(w_i^{t-1}) \|_2^2 \quad r'(w_i^{t-1}) \quad M_i(w_i^{t-1}) \|_2^2 \tag{33}$$

Due to the Triangle Inequality, we have

$$\sqrt{(2 \quad)} \, r'(w_i^{t-1}) \quad r'(w_i^{t-1}) \quad M_i(w_i^{t-1}) \quad r'(w_i^{t-1}) + M_i(w_i^{t-1}) \tag{34}$$

It follows that:

$$\sqrt{(2 \quad)} \quad 1 \quad r'(w_i^{t-1}) \quad M_i(w_i^{t-1}) \tag{35}$$

By incorporating Equation 32 and leveraging the AM-GM inequality, we can derive the following expression

$$\begin{aligned} r'(w_i^{t-1}) \quad M_i(w_i^{t-1}) \|_2^2 \quad & (1 \quad ^2) \, r'(w_i^{t-1}) \|_2^2 + \quad r'(w_i^{t-1}) \quad M_i(w_i^{t-1}) \|_2^2 \\ & 1 \quad ^2 + \quad 1 + \quad + \sqrt{(2 \quad)}^2 \quad r'(w_i^{t-1}) \|_2^2 \\ & 4 + 6 \quad ^2 \quad r'(w_i^{t-1}) \|_2^2 \end{aligned} \tag{36}$$

Therefore,

$$E[B] \quad r_{t-1}^2 \Big\| \sum_{i \in mN} p_i^{t-1} \sqrt{4 + 6 \quad ^2} \, r'(w_i^{t-1}) \Big\|_2^2 \quad 4 + 6 \quad ^2 \, m^2 N^2 r_{t-1}^2 G_w^2 \tag{37}$$

Hence forC, we have

$$E[C] \quad \frac{2mN G_w r_{t\ 1}^2 \sqrt{4+6^{\ 2}}}{2' + t} \quad 2' \ E \ w^0 \quad w \ _2^2 + \ ^2 \tag{38}$$

Then based on Assumption 4.1 and Taylor expansion, we have the quadratic upper-bound of

$$L(w_1) \quad L \ (w_2) \quad (w_1 \quad w_2)^T rL \ (w_2) + \frac{L}{2} kw_1 \quad w_2 k_2^2$$

It follows that

$$\begin{aligned} EL(w^T) \quad L \ (w \ ) \quad & \frac{L}{2} E \ w^T \quad w \ _2^2 \\ & \frac{L + 2Lr_{T\ 1}\$}{2' + T} \ ' \ E \ w^0 \quad w \ _2^2 + \frac{^2}{2} \quad + \frac{L\$^{\ 2}}{2} \end{aligned}$$

Where' = (L + 1), \$ = mN G_w r_{T 1} $\sqrt{4+6^{\ 2}}$ □

A.2 Proof of Theorem 4.17

A.2.1 Lemmas

Lemma A.1. f is monotone: $\forall v_{\ i}$ and $\forall v_i^0 > v_i$, if $f(v_i; v_{\ i}) \ 2 \ W_i$, then $f(v_i^0; v_{\ i}) \ 2 \ W_i$.

Lemma A.2. In FEDQV, $\forall i; v_i; v_{\ i}$ that $f(v_i; v_{\ i}) \ 2 \ W_i$, we have that $p_i(v_i; v_{\ i}) = \ _i(v_{\ i})$, where $_i$ is the critical value of a monotone function on a single parameter domain that $_i(v_{\ i}) = \sup_{v_i : f(v_i; v_{\ i}) \ 2 \ W_i} v_i$.

A.2.2 Proof of Lemmas

Proof of Lemmas 4.15

Proof. $\forall v_{\ i}$ and $\forall v_i^0 > v_i$, based on the voting scheme, if the party who submits s_i join the aggregation with v_i, which means $f(v_i; v_{\ i}) \ 2 \ W_i$, then this party can also submit $\forall s_i^0 < s_i$ that lead to $v_i^0 > v_i$, and still join the aggregation. In other words $f(v_i^0; v_{\ i}) \ 2 \ W_i$. Thus, f is monotone. □

Proof of Lemmas 4.16

Proof. The number of parties is C in each round. In voting scheme that follows Equation 3, the parties who chose and $s_i \quad 1$ pay 0 credits voice. After Equation 4, the parties with 0 credit voice or 0 budget gain 0 vote. Assuming there are the top k (k < C) parties in ranking whose payments are $\forall j \ 2 \ k$ ($c_{j \ 2 \ k} > 0$). Notice in FEDQV, the payment function $p_i(v_i; v_{\ i}) = c_i = v_i^2$.

$\forall j \ 2 \ k$, if party j pays $c_j^0 > p_j(v_j; v_{\ j}) = \ _i(v_{\ i}) = \sup_{v_i : f(v_i; v_{\ i}) \ 2 \ W_i} v_i$, it will still remain in top k and join the aggregation. On the other hand, if party j pays $c_j^0 < p_j(v_j; v_{\ j}) = \ _i(v_{\ i})$, then it will be replaced by the party $k + 1$ in the ranking, and party j will not be able to join the aggregation regardless of whether party $k+1$ joins or not. As a result, in order to participate in the aggregation, the parties need to pay critical value $\forall i$, that is, $\forall s_i$ that $f(v_i; v_{\ i}) \ 2 \ W_i$, we have that $p_i(v_i; v_{\ i}) = \ _i(v_{\ i})$ □

A.2.3 Proof of Theorem 4.17

Proof. According to Theorem 9.36 [41]: a normalised mechanism on a single parameter domain is incentive compatible (truthful) if and only if:
(i) The selection rule is monotone.
(ii) For every party i participants in the aggregation v_i (> 0) pays the critical value $_i(v_{\ i}) = \sup_{v_i : f(v_i; v_{\ i}) \ 2 \ W_i} v_i$.
The first condition (i) and the second one (ii) are proofed in Lemma 4.15 and Lemma 4.16 respectively. Thus, the proposed scheme FEDQV is incentive-compatible (truthful). □

B More Experimental Results

B.1 More Experimental Details

Platform Con gurations. Our simulation experiments are implemented with Pytorch framework [64] on the cloud computing platform Google Colaboratory Pro (Colab Pro) with access to Nvidia K80s, T4s, P4s and P100s with 25 GB of Random Access Memory.

Table A1 shows the default setting in our experiments.

Table A1: Default experimental settings

Explanation	Notation	Default Setting
Budget	B	25
Similarity threshold		0.1
The number of parties	N	100
The fraction of selected parties	C	10
The number of total steps	T	500
The number of local epochs	E	5
Learning rate	r	0.01
Local batch size	b	10
Loss function	L ( )	Cross-entropy
Repeating times		3

B.2 Extra Experimental Results Under Small Percentages of Attackers.

To investigate FEDQV's behaviour in scenarios with ner gradations, we evaluated it with small, realistic percentages of attackers. Speci cally, we examined the performance of Trimmed-Mean and Trimmed-Mean Integrated with FEDQV under two attacks, including Backdoor and QV-Adaptive, with 1%, 5%, and 10% attackers. The results

	Trimmed-Mean	Trimmed-Mean-QV
Backdoor	ACC(%)/ASR(%)	ACC(%)/ASR(%)
1%	84.99/0.57	85.67/0.52
5%	84.83/0.93	85.66/0.46
10%	85.45/2.27	85.06/1.79
Qv-adaptive	ACC(%)	ACC(%)
1%	84.13	86.38
5%	83.88	85.51
10%	79.49	85.74
30%	10.00	73.95
50%	10.00	10.00

Table A2: Comparison of Trimmed-Mean and Trimmed-Mean Integrated with FedQV Methods under Targeted Attacks (Backdoor and QV-Adaptive) Across Varying Percentages of Malicious Parties.

in Table A2 indicate that even under these small, realistic percentages of attackers, Trimmed-Mean integrated with FEDQV consistently outperforms Trimmed-Mean alone. The small percentage of attackers did not signi cantly impact the performance of the models, with almost unaffected accuracy (ACC) and a small attack success rate (ASR). This outcome underscores that integration with FEDQV enhances the robustness of the original defence mechanism across varying threat levels.

B.2.1 Under Different Non-IID Degree.

we have examined the performance of FEDQV and FEDAVG across varying degrees of non-IID data, ranging from 0.1 to 0.9, as depicted in Table A3. These results demonstrate that as the non-IID degree increases among the parties, the performance of the global model declines. Notably, FEDQV consistently maintains a superior performance compared to FEDAVG, even when confronted with different degrees of data heterogeneity under attack conditions.

Non-IID		0.1	0.3	0.5	0.7	0.9
FedQV	ACC(%)	84.94	86.01	83.88	81.37	75.96
	ASR(%)	3.39	4.55	17.64	20.59	24.18
FedAvg	ACC(%)	81.27	81.1	82.44	80.77	65.68
	ASR(%)	3.37	13.39	20.84	22.99	60.35

Table A3: Comparison of Accuracy (ACC) and Attack Success Rate (ASR) for FedQV and FedAvg under Backdoor Attack over 100 epochs with varying Non-IID Degrees on Fashion-MNIST Dataset.

C More Details of Defence against privacy attacks

C.1 Privacy Attack Algorithms

We mount the Deep Leakage from Gradients (DLG) attack [8] and the Gradient Inversion (GI) attack [9] on FEDQV. These attacks aim to generate dummy data and corresponding labels by leveraging a gradient-matching objective. The detailed algorithms are outlined in Algorithm 3.

Algorithm 3: DLG and GI Attacks

Input: $F(D_i; w_i^t)$: model at round t from targeted user i; learning rate for inverting gradient optimiser; S: max iterations for attack; : regularisation term for cosine loss in inverting gradient attack; d:the model size

Output: reconstructed training data $(D_i; y_i)$ at round t

Initialise D_0^0 $N(0,1)$, y_0^0 $Randint(0; max(y))$

for s $0,1, \ldots S$ 1 **do**

$r\ w_s^0$ $@`(F(D_s^0; w_i^t); y_s^0)/@w_i^t$

switch *Case* **do**

case *DLG attack* **do** L_s^0 $k\ r\ w_s^0$ $r\ w_i^t\ k^2$

case *GI attack* **do** L_s^0 1 $\frac{r\ w_i^t\ \ r\ w_s^0}{k\ r\ w_i^t k k\ r\ w_s^0 k}$ +

D_{s+1}^0 D_s^0 $r_{D_s^0} L_s^0, y_{s+1}^0$ y_s^0 $r_{y_s^0} L_s^0$

return $D_S^0; y_S^0$

C.2 Implementing SECAGG in FEDQV

Our approach is in line with other securely aggregated FL designs consisting of three main stages: 1) *Setup*; 2) *Generation and Protect*; and 3) *Aggregate*. Our integration of secure aggregation to FEDQV is depicted in Figure A1. We further discuss its details below:

Stage 0 (*Setup*). In this stage, the server and the partis compute SECAGG:Setup and initialise FEDQV by having the server send values of the protocol's parameters to each party, i.e. w^t 1.

Stage 1 (*Generation and Protect*). After the setup and initialisation stage, each party first computes its local model w_i^t as in FEDQV and computes its s_i^t. Later, it transmits the message $hjD_ijj; s_i^t i$. Upon receiving $hjD_ijj; s_i^t i$, the server computes v_i^t as in Equation 4 and transmits v_i^t to the corresponding party. Then party i first computes w_i^t v_i^t and protects its input w_i^t v_i^t by treating it an input to the SECAGG:Protect as $[w_i^t] = w_i^t$ $v_i^t + k_i + b_i$. Party i sends $[w_i^t]$ to the server.

Stage 3 (*Aggregate*). Upon receiving $[w_i^t]$, the server first calls SECAGG:Aggregate as described previously which returns the sum of all w_i^t. The server requires one more step to finalise by multiplying the sum retrieved by $\frac{1}{\sum_{i=1}^{N} v_i^t}$, as it is the sole entity possessing complete knowledge of all v_i^t.