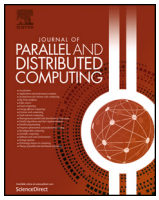
Journal of Parallel and Distributed Computing 196 (2025) 104994

Contents lists available at ScienceDirect

Journal of Parallel and Distributed Computing

journal homepage: www.elsevier.com/locate/jpdc



Towards value-sensitive and poisoning-proof model aggregation for federated learning on heterogeneous data

Hui Zeng [a], Tongqing Zhou [a,*], Yeting Guo [a], Zhiping Cai [a,*], Fang Liu [b]

[a] College of Computer, National University of Defense Technology, Changsha, Hunan, 410073, China
[b] School of Design, Hunan University, Changsha, Hunan, 410073, China

ARTICLE INFO

Keywords:
Federated learning
Model aggregation algorithms
Data heterogeneity
Model replacement attacks
Mitigating attacks mechanism

ABSTRACT

Federated Learning (FL) enables collaborative model training without sharing data, but traditional static averaging of local updates leads to poor performance on heterogeneous data. The following remedies, either by scheduling data distribution or mitigating local discrepancies, predominately fail to handle fine-grained heterogeneity (e.g., local imbalanced labels). To commence, we reveal that static averaging leads to the global model suffering from the *mean fallacy*. That is, the averaging process favors the local model with large parameters numerically rather than knowledge. To tackle this, we introduce FedVSA, a simple-yet-effective model aggregation framework sensitive to heterogeneous local data merits. Specifically, we invent a new global loss function for FL by prioritizing the valuable local updates, facilitating efficient convergence. We deduce a softmax-based aggregation rule and prove its convergence property via rigorous theoretical analysis. Additionally, we expose poisoning threats of model replacement that utilize the *mean fallacy* for attacks. To mitigate this threat, we propose a two-step mechanism involving auditing historic local training statistics and analyzing the *Shapley Value*. Through extensive experiments, we show that FedVSA achieves faster convergence (~1.52×) and higher accuracy (~1.6%) compared to the baselines. It also effectively mitigates poisoning attacks by agilely recovering and returning to normal aggregation.

1. Introduction

The increased computation power, abundant data on mobile devices, and advancements in artificial intelligence algorithms have led to the rise of Federated Learning (FL) as a prominent method in distributed machine learning [24,26,29,47,87]. By accommodating massive clients to cooperatively train a model using their local data without compromising data privacy, FL is suitable for various real-world learning scenarios (e.g., medical [30,44,69], industrial [6,17,43], smart cities [50]). One building block for FL to fuse distributed knowledge is model aggregation, where the vanilla method is known as FedAvg [47]. For FedAvg, in each communication round, distributed devices perform local training based on the downloaded global model, while the server produces a new global model by taking the average of all the local models.

However, recent studies have criticized that FedAvg presents a poor performance and vulnerability with attacks when trained on heterogeneous data [3,64,73,86]. In real-world scenarios, different users have distinct usage habits [65], the data samples generated or stored on end devices (clients) are non-independent or identical (i.e., non-IID) and may have imbalanced labels. In this heterogeneous data distribution, the simple averaging of FedAvg tends to prioritize the training results from 'large' clients that have more data samples, which leads to the model being biased and a slower acquisition of knowledge from the 'small' clients. Consequently, this significantly prolongs the convergence time of the learning process and negatively impacts the accuracy of the model [37,73]. A study conducted in [86] showed that the model trained by FedAvg on CIFAR10 with heterogeneous settings experienced a 37% decline in accuracy compared to the IID counterpart. Besides, some studies show that the attacker can easily manipulate the global model by leveraging the static averaging [2,42,62,66,82]. By scaling or continuously injecting the poisoned model, the global model can be destroyed or contain a backdoor.

Current research proposes to solve the above statistical heterogeneity problem by either actively balancing the data distribution [12,15,22,31,54,63,65,70,72,85,86] or optimizing the aggregation process [1,16,18,34,37,53,58,59,67]. On the one hand, in order to make

* Corresponding author.
E-mail addresses: zenghui116@nudt.edu.cn (H. Zeng), zhoutongqing@nudt.edu.cn (T. Zhou), guoyeting13@nudt.edu.cn (Y. Guo), zpcai@nudt.edu.cn (Z. Cai), fangl@hnu.edu.cn (F. Liu).

https://doi.org/10.1016/j.jpdc.2024.104994
Received 28 September 2023; Received in revised form 18 July 2024; Accepted 8 October 2024
Available online 11 October 2024

the data balanced in training, some methods require sharing a portion of the local data [86] or server-side proxy data [31] with different clients. However, data sharing may raise privacy concerns and contradict the objectives of FL. Alternatively, a lot of methods also focus on actively selecting clients for balanced data distribution. Astraea [15] employs a virtual mediator to reschedule the clients and rebalance the data. DQS [63] selects the clients with high data quality to participate in the training process. Reinforcement learning is introduced in FAVOR [65] and FedSens [12] to assist in deciding and selecting the clients for each training round. IFCA [22] and FedDMS [85] adopt clustering-based methods to partition clients into different clusters to perform an iterative training process. These methods can unintentionally reveal the distribution of local data and usually introduce extra establishment latency for FL. On the other hand, several aggregation optimization techniques are dedicated to enhancing the loss function. FedProx [37] introduces a proximal term to the objective with the purpose of improving stability. FedCurv [59] incorporates a regularization term into the loss function to encourage local models to converge towards a common optimum. However, these additional regularization terms mainly reduce the overall variability among the models, while the label imbalance inside each client is not addressed.

To mitigate malicious attacks, existing methods try to design detection and filtering mechanisms. FAA-DL [57] uses OC-SVM to detect abnormal updates from local clients. Krum [4] and Zeno [74,75] design some score functions to measure the local updates and filter out outliers. However, it is challenging for these online outlier-based detection mechanisms to counter various and stealthy threats, since the attacker will disguise the malicious updates and make them more similar to the normal ones. Furthermore, some efforts try to use clean 'label' data to purify the poisoned model [39,60]. However, it is difficult to adapt these tools to FL, due to the data access restrictions and frequent interactions between servers and clients.

In this paper, we initially confirm the negative impact of heterogeneous data distribution on the global model. Additionally, we investigate that this issue may arise due to the occurrence of the *mean fallacy* when averaging the local parameters with static weights. The *mean fallacy* arises when the valuable local updates are overshadowed by the larger ones during the simple static averaging process. Low-quality or malicious local updates may have a large impact on the future global model. To escape from the *mean fallacy*, one straight solution is to assign more weight to valuable clients in aggregation. The crucial aspect of this approach lies in determining the merit of each client. However, there are two challenges that need to be addressed in order to achieve this objective:

- **Privacy-preserving principles**. We cannot directly access the client's locally stored data for their merit evaluation. Furthermore, the merit should not expose any information about the local data, such as its distribution or quantity.
- **Unexpected behaviors of clients**. The behavior of a client during the training process can be unpredictable, as it depends on the subjective willingness and objective resource constraints. A client could potentially act maliciously, be uncooperative, or disrupt the training process. Even an honest client, its behaviors can differ from one round to the next.

To overcome these challenges, the method being developed must ensure privacy, adaptability, and reliability. In this paper, we introduce an effective and robust model aggregation framework for FL, named FedVSA, which improves the model performance and robustness by endowing Sensitivity on the Value (knowledge) of local updates during Aggregation. We point out that the local models with more knowledge should play a crucial role in improving the global model performance [68]. Hence, FedVSA is designed to judiciously give more attention to the valuable distributed knowledge during aggregation for learning to help the global model escape from the *mean fallacy*. In order to protect data privacy while evaluating the merit of each client, we introduce the concept of *inference loss* as the loss on the client's local data with the current global model. Without revealing local data, we use the inference loss to measure the dynamic bias between each piece of local knowledge and the globally learned one, in this way, attaining the value of a tuned local model in improving the overall performance. Formally, we design a novel global loss function, in which the local updates are summed exponentially (Eq. (6)), implicitly giving larger weights for updates with larger inference loss and using them to guide the optimizing direction (i.e., gradients). The value-sensitive optimization process that leans toward unacquainted knowledge can effectively accelerate model convergence.

Second, to mitigate potential poisoning behaviors [10,23,27,46,55], in particular, the model replacement attacks, FedVSA exploits the differential inference loss of adjacent communication rounds in a two-step anomaly detection and recovery mechanism. In brief, inference loss notably increasing for the majority of clients indicates an anomaly in the global model. On detecting an anomaly, we dig into the *Shapley Values* of different updates in this round by computing the marginal accuracy improvement of the cached uploaded models. Finally, those updates with negative Shapley values are identified as abnormal and unlearned in the global model.

The main contributions can be summarized as follows:

- We point out, with experimental observations, that FL experiences severe performance degradation as the local label imbalance worsens. Furthermore, our study suggests that this degradation may be due to the static simple averaging method, which causes the global model to fall into the *mean fallacy*. In other words, it tends to favor local models with large numeric values rather than knowledge.
- We design a novel global loss function with attention to valuable local updates (i.e., those with high inference loss) and deduce the aggregation rule. We provide rigorous proof and analysis for their convex and convergence properties.
- Considering the malicious behavior of attackers, we utilize the inference loss uploaded from the clients for robustness. We propose a two-step mechanism to detect and recover from anomalies by considering both performance statistics (historic inference loss) and the marginal contributions (*Shapley Values*).
- Experimental results demonstrate that the proposed method outperforms the 13 typical baselines in terms of both convergence speed and overall performance on three popular datasets. Moreover, it possesses the ability to quickly recover from attacks.

This is an extension to the previous conference paper [81]. The principal differences involve: 1) We have newly added the observations to find the reason for the FL performance decline. We identified the reason as the *mean fallacy* problem of the vanilla static averaging algorithm in FL. 2) We have proved the convex and the convergence properties of the proposed value-sensitive aggregation algorithm. 3) We have newly designed a two-step detection&recovery mechanism against model poisoning in FL. These make it sufficiently novel in terms of the findings, design, analysis, and evaluation.

2. Related work

2.1. Data heterogeneity in FL

Data heterogeneity poses a significant challenge in FL [19,33], as distributed data sources usually have different data distributions. Existing papers mainly attempt to solve this issue by mitigating discrepancies, alleviating catastrophic forgetting, and deliberately selecting participants.

Local gradients discrepancies mitigation. Many researchers have claimed that the great diversity of local gradients is one of the most important reasons for the global model accuracy decline. Some propose to share a portion of local data [86] or server-side auxiliary data [31]

among (to) clients to make the local updates more similar. These methods are criticized as violating the privacy protection intention of FL. Some work tries to adjust the loss function for similar gradients among updates. FedProx [37] and FedProxVR [14] introduce a proximal term to the local objective in order to minimize the differences between local updates and the global model, while FedCurv [59] incorporates a penalty term into the loss function, forcing all local updates to converge towards a common optimal solution. However, a major limitation of these approaches is that they often overlook the possibility that the approximate regulation might lead to a sub-optimal solution for the global model. Some work also claims that the discrepancies of local and global optimum, named 'client drift', caused by heterogeneous data are the main culprit that damages the convergence rate [38]. SCAFFOLD [34] introduces the stochastic control averaging algorithm to rectify the 'client drift'. Similarly, FedDyn [1] proposes a dynamic regularization for the clients to align them with the global optimum, thus reducing transmission costs. Based on SCAFFOLD, FedDC [18] decomposes the client drift into two components to monitor and correct the local drift between local and global models. FedNova [67] ensures that local models are appropriately weighted during averaging, maintaining objective consistency and enabling fast error convergence. FedConD [8] employs an adaptive mechanism to identify client drift and minimize its impact on the performance of models in asynchronous FL. FedDisco [78] modifies the aggregation weights and makes it not only involve both the data quantity and the distribution discrepancy. FedLAW [40] designs a learnable aggregation weight for adaptive aggregation, which is a cold start method and requires a number of communication rounds to get an optimal weight for each client. Despite these advancements, most of these methods still rely on static parameter averaging, making them susceptible to the *mean fallacy*.

Alleviating catastrophic forgetting. Recent researchers also attribute the problem to catastrophic forgetting. It has been noticed that the locally trained model tends to significantly forget the knowledge gained from previous training data at other clients [76]. FedReg [76] utilizes generated pseudo data in local training to alleviate forgetting and protect the original data from gradient inversion attacks. FCCL [32] incorporates knowledge distillation during local training and incorporates unlabeled public data for communication. Additionally, FCCL constructs a cross-correlation matrix to learn a generalized representation that can handle domain shifts. Some other efforts try to reduce forgetting by adopting a self-attention-based architecture [53], collaborative replay [52], and matching feature layers of local models [83]. However, these efforts require auxiliary information which will increase the potential for exposing data privacy.

Participants selection. Selecting participants with more potential is also helpful for FL performance. FAIR [13,41,80] adopts loss reduction during training to measure individual quality and maximize the collective learning quality of all participants. FedACS [70] quantifies the label distribution heterogeneity using Hoeffding inequality and selects clients with minimal data heterogeneity based using Thompson sampling. Similarly, DQS [63] introduces a scheduling algorithm based on the greedy knapsack approach to choose clients with high-quality data. Reinforcement learning is used to directly select the participants in [65] or assist clients in deciding whether to participate in a training round in [12]. However, such selection can introduce a bias towards the chosen clients. Besides, in practice, qualified clients are often the minority and may not be available online continuously. Astraea [15] increases the uniformity of data distribution by introducing virtual components, which helps in rescheduling the training process and redistributing the clients' local data. IFCA [22] is a method that uses clustering to estimate the cluster identities of clients and optimize model parameters for each cluster using gradient descent. FedDMS [85] leverages the clustering-based methods to reschedule the clients to alleviate the discrepancies by considering the upload delay and the direction of the local updates. Oort designs a metric named statistical utility based on the accumulated local loss to evaluate the clients' contribution to select highly important clients. However, these methods might raise privacy concerns by exposing the local data distribution or profile of the client. Besides, these methods only use the loss for selection and retain the local training and aggregation as it stood. Notably, FedVSA not only evaluates the merit of each client but also uses the loss to guide the direction of global optimization.

2.2. Threats against FL

Many approaches have been proposed to enhance the resilience of the FL in response to model replacement or backdoor threats, which intentionally corrupt the global model, as indicated by recent research [20,33,82,88].

Robust statistics. Some researchers propose to use robust statistics in aggregation rather than directly averaging. For example, Median [79] uses geometric median as the aggregation rule. However, this method causes a negative impact on model convergence and can not guarantee that the attackers are blocked.

Trimming outlines. Some researchers try to trim the outlines to protect the training process. Trimmed mean [9,79] establishes statistical error rates for the aggregation rule by utilizing marginal trimmed mean. Kardam [11] filters out outliers by leveraging the Lipschitzness of the gradients. Krum [4] selects the candidates based on the minimal local sum of Euclidean distances, while Zeno [74] and Zeno++ [75] compute scores using a stochastic zero-order oracle and average over the candidates with the highest scores. DnC [56] has been a representative robust aggregation method in recent years. DnC performs dimension reduction using random sampling followed by spectral methods based on outliers removal. RAF [51] is a robust aggregation method that relies on the aggregation of updates using the geometric median, which can be computed efficiently using a Weiszfeld-type algorithm. However, all of these methods require a fixed trimming number to confront a bounded number of attackers. Yet, the assumption of knowing such a bounded number is impractical.

3. Observations

In this section, we explore and analyze the statistical data heterogeneity problem in FL. Subsequently, we investigate that the simple averaging might lead the global model to fall into *mean fallacy*.

3.1. Preliminaries on data heterogeneity

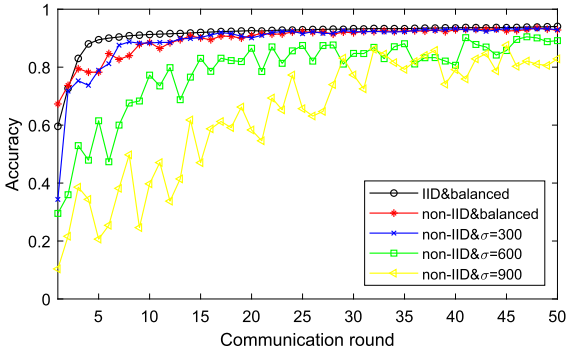
Some point that the reason for the poor performance is the model deviation of the local models, which comes from the data heterogeneity [33]. Here, we propose to analyze the statistical data heterogeneity from two aspects: from a global perspective, *non-IID* is the main cause of model deviation; from a local perspective, *label imbalance* causes a biased local model and exacerbates the model's deviation.

Global non-IID. There are inherent differences in the data distribution *between* different clients, such as variations in data quantity and category distribution, termed as non-IID. These differences arise from variations in the contexts of the clients. Additionally, the global data distribution changes dynamically as the clients join or leave the training process.

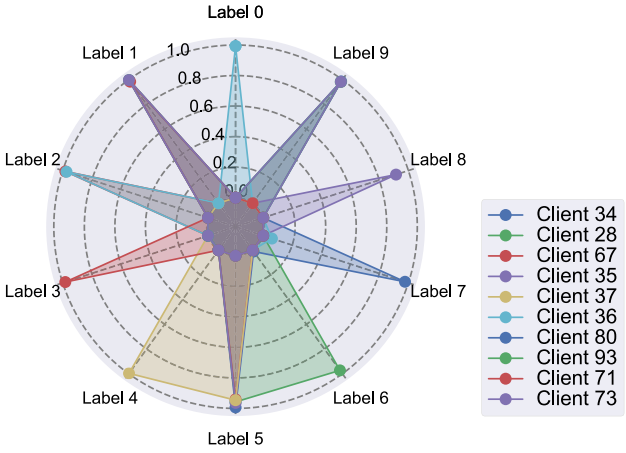
Local label imbalance. There are data distribution discrepancies *within* each client, that is, the data quantity belonging to different labels is different, termed as label imbalance. Furthermore, as data can be collected and generated in real-time, the situations of imbalance also change over time.

3.2. Observations on FL with heterogeneous data

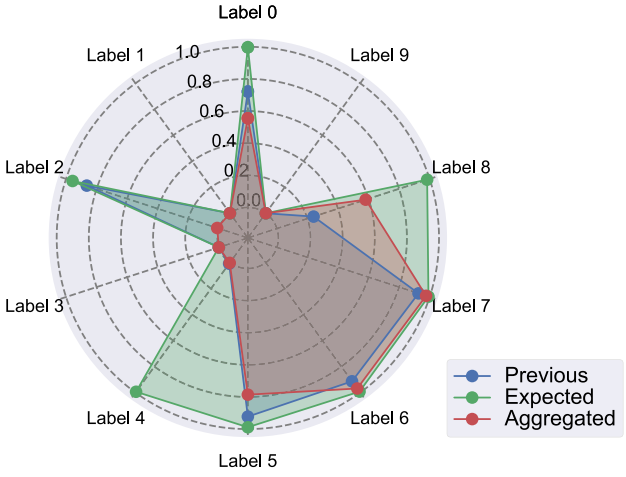
According to the analysis mentioned above, we set different types of data distributions for simulation purposes and assess the performance of FedAvg.



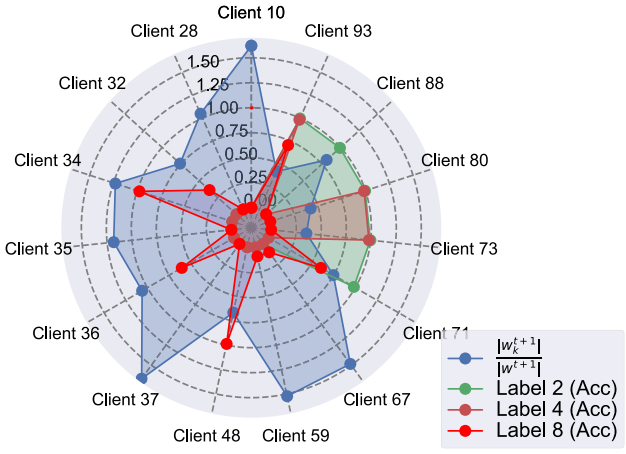
(a) Learning performance of FedAvg on heterogeneous data of 5 different levels.



(b) The classification accuracy of local models in each label with heterogeneous data on MNIST.



(c) The classification accuracy of the global models before and after aggregation on MNIST.



(d) The observation on the model parameters (in numeric, i.e., $\frac{|w_k^{t+1}|}{|w^{t+1}|}$, $|w_k^{t+1}|$ is the local model in client k at round $t+1$, $|w^{t+1}|$ is the global model at round $t+1$) and classification accuracy on two labels ('Label 2', 'Label 4', and 'Label 8') in each client.

Fig. 1. The observations of statistic data heterogeneity problem and *mean fallacy* problem in FL.

We follow the experimental settings in [47,81]. In the distributed network, we deploy 100 clients, with 30% of clients randomly participating in each round. These clients collectively train a CNN model on MNIST, specifically LeNet-5 [36,45].

Data distribution. According to our analysis of data heterogeneity, we set three different types of data distribution.

1. IID & balanced. In this setting, the data distribution is uniform across all clients, and the quantity of each label is evenly distributed within each client's dataset.
2. non-IID & balanced. The data distribution varies among clients, and we randomly assign two labels to each client while ensuring that the quantity of each label remains balanced within their respective datasets.
3. non-IID & imbalanced. We randomly assign two labels to each client to simulate the non-IID setting. However, the quantity of each label differs between clients, and we utilize the variance parameter σ to quantify this difference.

In our analysis, enforcing IID as a strict requirement is challenging, especially when dealing with real-world datasets where label quantities often vary. Acknowledging this reality, we introduce a parameter, denoted as σ, which represents the *statistical variance* of label sizes, quantifying the degree of label imbalance. A higher σ value indicates more significant disparities between the sample sizes of different labels. We deploy the FedAvg on MNIST and observe the performance in different σ settings.

The results of the evaluation are presented in Fig. 1a. It is evident from the observations that label imbalance has a notable impact on accuracy. When the data is *balanced*, FedAvg achieves convergence after just 5 rounds, whereas it requires approximately 20 to 35 rounds when dealing with *imbalanced* data. Furthermore, as σ increases, the accuracy diminishes, leading to more fluctuations in performance during the training process.

3.3. Observations on mean fallacy with heterogeneous data

In order to investigate the reasons behind the performance decline when dealing with heterogeneous data in FL, we conduct a comprehensive analysis of both the local and global models from two perspectives: performance view and numeric view. Our findings indicate that the use of static averaging may cause the global model to fall into what we refer to as the *mean fallacy*.

Biased local models. To commence, we study the classification performance of local models for each label, as shown in Fig. 1b. Our analysis reveals that the local models can achieve high accuracy only on the labels they were specifically trained on while exhibiting near-zero accuracy on other labels. This observation indicates that the presence of heterogeneous data may lead to biased local models, primarily due to overfitting on their local data. Furthermore, the discrepancy among the local models emerges as one of the primary reasons for the overall performance decline, thus confirming the claims made in [31,37,86].

Flawed aggregation. In Fig. 1c, we test the performance of the global model before and after the aggregation process, and we select the best performance achieved by any of the local models for each label as the *expected global model*. Our observations reveal that the global model exhibits two distinct behaviors: dilution and forgetting. For dilution, the aggregated global model fails to fully absorb the knowledge from the local models, as evident in cases like 'Label 4' or 'Label 8' (see Fig. 1c). Concerning forgetting, the aggregated model loses some knowledge present in the previous model, whereby the performance on 'Label 2' decreases near 90% after the aggregation.

Large values preference. Based on the findings from Fig. 1c, we have concluded that the static averaging methods fail to fully aggregate the knowledge from the local clients. We declare that the parameters averaging is not equal to model knowledge averaging. We conduct a numeric analysis of the local models' parameters, as shown in Fig. 1d. Specifically, we use the ratio $\frac{|w_k^{t+1}|}{|w^{t+1}|}$ to measure the local models' parameters and display the accuracy of three labels that were observed to be diluted ('Label 4'), forgotten ('Label 2'), and normal ('Label 8'). The $|w_k^{t+1}|$ represents the local model from client k at round $t+1$, and $|w^{t+1}|$ represents the global model at round $t+1$. **Our observation indicates that the model performance is independent of the numeric value of the model parameters, a larger numeric value of the model parameters does not represent containing more knowledge, yet existing flawed aggregation methods only focus on the parameters' numeric value and prefer large values**. We can observe that 1) large numeric values do not represent high performance, as models with large values perform poorly in 'Label 2' and 'Label 4'; 2) the model with smaller numeric values does not represent it contains less knowledge, as some models with smaller numeric values perform better than large ones; 3) The performance of 'Label 8' can be high not matter the model parameters' value is large or not.

As for *mean fallacy*, the performance of the global model experiences a significant decrease after the flawed aggregation with the biased local models, since the aggregation is large value preference rather than large knowledge. **We attribute these phenomena to existing simple static averaging**. Simple static averaging adopts the mean to aggregate the biased local models, yet **the mean may not be a suitable statistic when there is a significant disparity in parameter numeric value**. Besides, we consider that **parameter averaging is not equal to model knowledge averaging**. Existing static averaging only considers the parameter space, thus making the knowledge of the aggregated global model agnostic and preferring models with large numeric values rather than knowledge.

4. Problem statement

As aforementioned, the pervasive presence of data heterogeneity significantly affects FL training. In this part, we construct a novel learning problem based on the vanilla FL architecture and accordingly deduce a new aggregation algorithm. Unlike FedAvg, the attained aggregation prioritizes the model-informed value rather than solely relying on the size of local data. The primary notations used in our formulation are listed in Table 1.

Table 1
Main notations in the problem statement.

Notation	Description
$(x_{k,j}, y_{k,j})$	The j^{th} sample and its label in client k.
CR	Short for Communication Round.
$\mathcal{N}$	The sets of clients in FL.
T	# CR that the learning process terminates.
w	Deep learning model parameters.
w^t	Global model parameters at the t^{th} CR.
w_k^t	Local model parameters of client k at the t^{th} CR.
w^*	Optimum of the global learning problem.
p_k	The weight of the client k.
D_k	Local dataset of client k.
D	The combination of all D_k.
m^t	The number of clients in t^{th} CR.
ξ_k^t	Samples chosen from the k^{th} client's local data.
ℓ	Some specific loss function.
F, F^*	The global loss function and its optimal solution.
F_k, F_k^*	The local loss function of client k and its optimum.
η_t	Local learning rate at the t^{th} CR.
E	# of local training epoch.
σ_k^2	The variance of stochastic gradient.
L	Parameter in *L-smooth*.
μ	Parameter in *μ-strongly convex*.
G^2	The expected square norm of stochastic gradient.

4.1. Learning problem of traditional FL

Before presenting our formulation, we first review the technical preliminaries of the learning problem in traditional FL. For the sake of clarity, we list the primary notations in Table 1 and explain some operations: $\|\cdot\|$ is the L_2 norm, $|\cdot|$ means the size of the set, and $\triangleq$ signifies 'is defined to be equal to'.

Assuming a distributed computing scenario with $|\mathcal{N}|$ clients, where each client possesses its local data D_k (where $k \in \mathcal{N}$), the collective dataset of all clients is denoted as $D = \sum_{k\in\mathcal{N}} D_k$. The global loss function is represented as $F(w)$, and the loss on the local data D_k is

$$F_k(w) \triangleq \ell(w, D_k) = \sum_{j}^{|D_k|} \ell(w, x_{k,j}, y_{k,j}). \tag{1}$$

With these, **the learning problem for distributed optimization in traditional FL** is:

$$w^* = arg\min \left\{ F(w) \triangleq \sum_{k\in\mathcal{N}} p_k F_k(w) \right\}. \tag{2}$$

Here, $\mathcal{N}$ represents the set of clients, and p_k denotes the weight assigned to the k^{th} client, satisfying the conditions $p_k \geq 0$ and $\sum_{k\in\mathcal{N}} p_k = 1$. Given the complexity of the distributed optimization problem, finding a closed-form solution is challenging. Therefore, to overcome this difficulty, distributed gradient descent is adopted as a means to mitigate this issue.

The vanilla FL algorithm, FedAvg, is commonly used for distributed gradient descent. We formally present one round (the t^{th} round) of the standard FedAvg procedure: Initially, the server *broadcasts* the global model defined by w^t to the participating clients. We denote w_k^t for the local model trained in the client k at the t^{th} CR; Subsequently, each client (e.g., the k^{th} client) sets the initial values of w_k^t as w^t and then conducts $E(\geq 1)$ steps *local training*. The client performs stochastic gradient descent (SGD) training locally by

$$w_k^{t+1} = w_k^t - \eta_t \nabla F_k(w_k^t, \xi_k^t). \tag{3}$$

According to the definition of the global loss function in Eq. (2), we can then get the partial derivative of $F(w)$ as

$$\nabla F(w) = \sum_{k\in\mathcal{N}} p_k \nabla F_k(w). \tag{4}$$

Using Eq. (3) and Eq. (4) with $t+1 \in \mathcal{I}_E$, **the aggregation algorithm of FedAvg** is then presented as

$$w^{t+1} = w^t + \sum_{k\in\mathcal{N}} p_k^t (w_k^{t+1} - w_k^t) = \sum_{k\in\mathcal{N}} p_k^t w_k^{t+1}. \tag{5}$$

The server will aggregate the local updates w_k^{t+1} using Eq. (5) and obtain a new global model w^{t+1}.

4.2. Learning problem of FedVSA

The essential objective of FL training is to minimize $F(w)$. As $F(w)$ quantifies the disparity between model predictions and true labels, a well-trained model aims for a small value of $F(w)$. Instead of uniformly averaging all clients' losses using the previous global loss function, we introduce a novel global loss function in FedVSA.

Based on the observations in § 3, it is evident that the model requires more communication rounds to achieve convergence when dealing with heterogeneous data. The underlying reason for this phenomenon could be attributed to the *mean fallacy* by simply static averaging, where the heterogeneous data leads to varying biases in the local training losses of different clients. Some clients experience extremely large losses, while others encounter losses close to zero. Consequently, the global model necessitates additional iterations to learn from these extreme losses through simple averaging. To tackle this issue, we propose that the gradient descent step should be tailored to each client, with a proportionally larger step for those with larger local losses. The rationale behind this approach is that a larger step signifies a higher emphasis on the global loss descent. We employ the exponential function to formulate the differential attention. Specifically, we design a logarithm to restrict the interval of the exponential sum, which enables us to derive our loss function for FedVSA:

$$F(w) = \ln\left(\sum_{k\in\mathcal{N}} e^{F_k(w)}\right). \tag{6}$$

Given such a learning formula, we then formally define the learning problem for optimizing $F(w)$ in FedVSA:

Definition 1 *(Optimization problem of FedVSA).* In the context of distributed FL clients with inference loss $F_k(.)$, the optimization problem of FedVSA involves finding an optimal w^* that minimizes $F(w)$.

$$w^* = \arg\min F(w) = \arg\min \ln\left(\sum_{k\in\mathcal{N}} e^{F_k(w)}\right) \tag{7}$$

Wherein, $F_k(w^t)$ is the loss of predicting the labels of local data using global model w^t, we define it as **inference loss**.

In order to solve the optimization problem and deduce the aggregation rule for global updates, we utilize the basic gradient descent method. According to Eq. (4) and (7), the aggregation algorithm of FedVSA is modeled as:

$$w^{t+1} = \sum_{k\in\mathcal{N}} softmax\left[F_k(w^t)\right] w_k^{t+1} \tag{8}$$

As shown, the difference between Eq. (8) and FedAvg lies in the weight factors. The weight factor p_k in FedAvg is related to the size of local data, while the weight factor of FedVSA is $p_k^t = softmax\left[F_k(w^t)\right]$.

The softmax function is used to normalize the inference loss into interval $[0, 1]$. The large inference loss will map to a large value after applying softmax. This endows the local updates with higher inference loss more weight in aggregation and provides more impact on global learning. In this way, the FedVSA is expected to accelerate the absorption of knowledge on 'unseen' local data, which owns higher inference loss.

4.3. Property analysis on FedVSA

In this subsection, we demonstrate that the global loss function in FedVSA (i.e., Eq. (6)) is conditionally convex. Additionally, we establish that the optimization problem of FedVSA can achieve convergence to the global optimum with a rate of $\mathcal{O}(\frac{1}{t})$.

4.3.1. Convexity of the global loss function

Convexity is a critical property for an optimization problem. A convex function [5] should satisfy Definition 2.

Definition 2 *(Strictly convex function).* $f : R^z \to R$ is a convex function, if

(1) $dom(f) \subset R^z$ is a convex set.
(2) $\forall v, w \in dom(f), v < w, \forall \beta \in (0,1)$, it satisfies $f(\beta v + (1-\beta)w) \leq \beta f(v) + (1-\beta) f(w)$.

In FedVSA, the global loss function $F(w)$ is related to the local loss function $F_k(w)$, which denotes the optimization problem in client k^{th}. According to Eq. (8), we deduce the relationship between $F(w)$ and $F_k(w)$ in Theorem 1. It shows that the global loss function in FedVSA can find an optimal solution if all $F_k(w)$ is convex.

Theorem 1 *(Convexity of $F(w)$). If the function $F_k(w)$ is convex and non-negative for each client k, then the $F(w)$ is also convex.*

Proof. The proof is presented in the Appendix A.1. □

It should be noted that the condition that all $F_k(w)$ are convex is not always met. For instance, when using DNNs, $F_k(w)$ is not convex. In such cases, we can only obtain a solution w that minimizes each $F_k(w)$ as much as possible, leading to the global loss function $F(w)$ achieving a relatively minimal value.

4.3.2. Convergence analysis on the aggregation rule

Inspired by [37,38], we establish the following assumptions concerning the local loss functions $F_k, k \in \mathcal{N}$.

Assumption 1 *(L-smooth).* $F_k, k \in \mathcal{N}$ are all L-smooth: for all v and w, $F_k(v) \leq F_k(w) + (v-w)^T \nabla F_k(w) + \frac{L}{2}\|v-w\|_2^2$.

Assumption 2 *(μ-strongly convex).* $F_k, k \in \mathcal{N}$ are all *μ-strongly convex*: for all v and w, $F_k(v) \geq F_k(w) + (v-w)^T \nabla F_k(w) + \frac{\mu}{2}\|v-w\|_2^2$.

Assumption 3 *(Variance bound).* The variance of stochastic gradients in each client is bounded: $\mathbb{E}\|\nabla F_k(w_t^k, \xi_t^k) - \nabla F_k(w_t^k)\|^2 \leq \sigma_k^2$, for $k \in \mathcal{N}$.

Assumption 4 *(Expected square norm bound).* The expected squared norm of stochastic gradients is uniformly bounded, i.e., $\mathbb{E}\|\nabla F_k(w_t^k, \xi_t^k)\|^2 \leq G^2$ for $k \in \mathcal{N}$ and $t = 1, \ldots, T$.

Based on these assumptions, we analyze the cases in which clients fully participate. Let FedVSA terminate after T CR and return the global model w^T. However, as shown in Table 2, compared with existing proofs, there are two key challenges to completing the convergence analysis:

- The newly designed global loss function $F(\cdot)$ **is a nonlinear sum** of local loss $F_k(\cdot)$. The global loss function in existing methods is the linear sum of local loss. When **Assumptions 1–4** hold, the *μ-strongly convexity* and *L-smoothness* of $F(\cdot)$ can be easily derived. These properties are essential for convergence analysis. For a nonlinear sum, these properties need further analysis. Thus, we derive

Table 2
Comparison of heterogeneous FL algorithms in terms of the formulation and the property of global loss function $F(w)$ and aggregation weights p_k.

Methods	Global loss function	Is $F(w)$ a **linear sum**?	p_k	Is p_k a **constant**?
FedAvg	$F(w)=\sum_{k\in\mathcal{N}} p_k F_k(w)$	✓	$p_k=\frac{1}{\lvert\mathcal{N}\rvert}$ or $p_k=\frac{\lvert D_k\rvert}{\lvert D\rvert}$	✓
FedProx	$F(w)=\sum_{k\in\mathcal{N}} p_k F_k^{prox}(w)$	✓	$p_k=\frac{1}{\lvert\mathcal{N}\rvert}$ or $p_k=\frac{\lvert D_k\rvert}{\lvert D\rvert}$	✓
FedDisco	$F(w)=\sum_{k\in\mathcal{N}} \frac{ReLU(n_k-a\cdot d_k+b)}{\sum_{m\in\mathcal{N}} ReLU(n_m-a\cdot d_m+b)} F_k(w)$	✓	$p_k=\frac{ReLU(n_k-a\cdot d_k+b)}{\sum_{m\in\mathcal{N}} ReLU(n_m-a\cdot d_m+b)}$	✓
Ours	$F(w)=\ln\left(\sum_{k\in\mathcal{N}} e^{F_k(w)}\right)$	×	$p_k^t=\frac{e^{F_k(w^t)}}{\sum_{m\in\mathcal{N}} e^{F_m(w^t)}}$	×

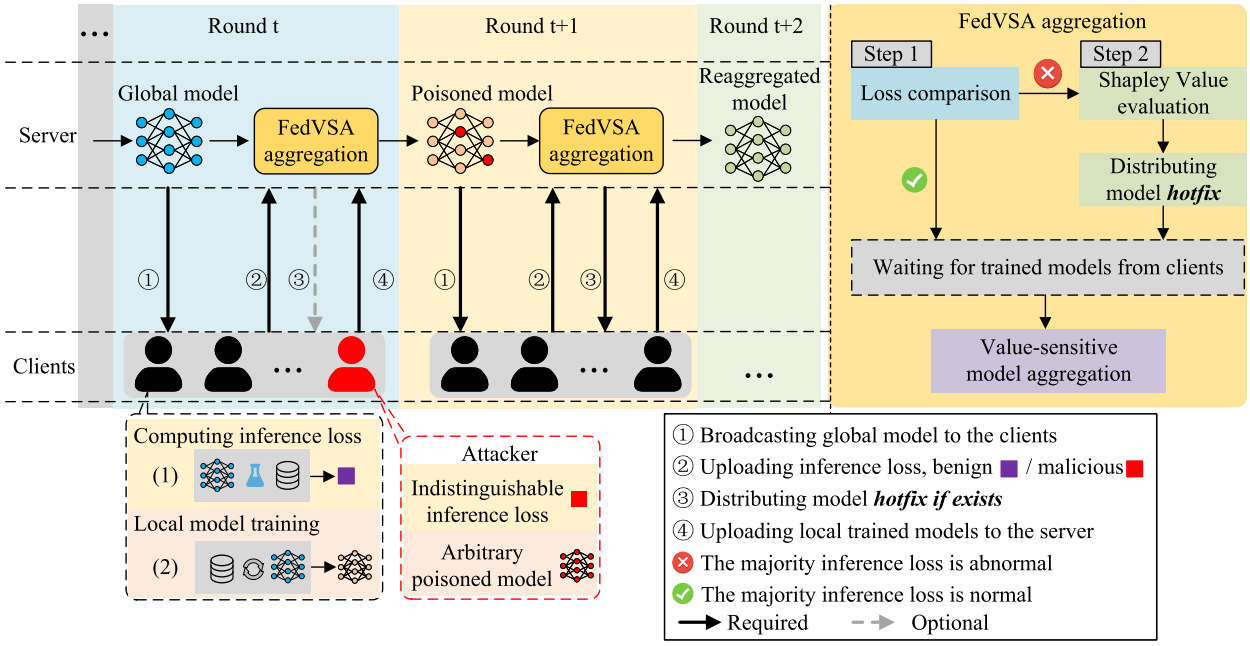


Fig. 2. The workflow of FedVSA. Then multi-rounds of training are performed between the server and the clients, wherein the clients compute the inference loss and perform local training, and the server makes cautious aggregation by analyzing the uploaded inference loss. Note that clients in the two rounds are different. If there exist abnormal updates, the server will distribute the model hotfix to the clients according to the Shapley Value. The hotfix can mitigate the impact of attacks and obtain a clean reaggregated model.

the properties of μ-*strongly convexity* and *L-smoothness* of $F(\cdot)$ based on **Theorem 1**, shown in Appendix A.1. Based on these, we follow the roadmap of [38] to derive a difference bound $\mathbb{E}(F(w^T))-F^*$ between the optimum global loss F^* and the expected global loss $\mathbb{E}(F(w^T))$.

- The aggregation weight p_k^t **is dynamically changing** along the training process. Existing proofs regard the aggregation weight as a constant and use it to form a part of the convergence bound. In FedVSA, the p_k^t is a dynamically changing variable, which depends on the inference loss of local clients. Thus, we derive and use the bound of p_k^t to constrain it to a constant value.

Theorem 2. *Assuming that* ***Assumption 1 to 4*** *are fulfilled, and we have defined* L,μ,σ_k,G *in those assumptions. With the learning rate* $\eta_t=\frac{2}{\mu(t+E)}$, *after* T *communication rounds, FedVSA with all client's participation satisfies*

$$\mathbb{E}\left[F(w^T)\right]-F^*\leq\frac{2L\Psi}{\mu^2(T+E)}+\frac{L}{2}\mathbb{E}\|w^1-w^*\|^2, \tag{9}$$

where

$$\Psi=8(E-1)^2G^2+\sum_{k\in\mathcal{N}}\sigma_k^2+\frac{2}{\eta_c}\Gamma,\quad \Gamma=\sum_{k\in\mathcal{N}}(F_k(w^*)-F_k^*).$$

Proof. We provide the main steps of the proof here. We try to get the difference between the trained model and the optimal one. Based on the four assumptions, we can get the one-step SGD bound (Lemma 3, 4, 5). Next, we set η_t and utilize the L-smoothness of $F(\cdot)$ to derive the multi-step SGD bound. For more details, please refer to Appendix A.2. □

The convergence analysis presented in Theorem 2 establishes that FedVSA achieves convergence to the global optimum at a rate of $\mathcal{O}(\frac{1}{T})$. After T communication rounds, we can obtain a global model w^T. With more iterations, the expected loss of w^T will be tuned close to the optimum.

5. Design of FedVSA

In this section, we present the overview and the key components of FedVSA.

5.1. Overview of the framework

To commence, we provide an overview of our proposed framework. As illustrated in Fig. 2, there are two parties in FL architecture, the server and the clients. The server maintains a global model and the clients own the private training data. Through multi-round interactions between the server and the clients, the global model can gradually absorb the information of local data. Each training round contains four stages: broadcasting the global model, value estimation, distributing hotfix, and value-sensitive model aggregation.

Broadcasting the global model (①): The server transmits the latest global model to all participated clients, and the clients train the global model with their own data.

Value estimation: The client computes the inference loss based on the download global model ((1) in clients). The inference loss is the loss value (e.g., cross-entropy) of the global model on the client's local

data. The estimation is dynamically evaluates the difference between the global model and local data, and transmitted back to the server (②).

Distributing hotfix: The server received the inference loss from the clients, and used the two-step detection mechanism to prevent the potential threats in the training (the attacker may upload arbitrary inference loss and model). For the first step, we compare the current inference losses with historical statistics to detect abnormal values. Based on the numerical comparison results, the server would directly process aggregation or perform a fine-grained contribution evaluation. For the second step, we leverage *Shapley Value* to measure the marginal contribution of the local updates from the previous round. We consider the local updates with negative *Shapley Values* to have a negative impact on the global model. We collect these local updates to produce the model hotfix and distribute them to the clients to mitigate the potential threats.

Value-sensitive model aggregation: The server aggregates the local updates weighted by inference loss, detail in § 5.2.

During the training process, the attacker can impersonate a client and perform attacks. The attacker can manipulate the inference loss and trained model so that the global model aggregated by these can be poisoned. In our two-step detection mechanism, the negative impact can be mitigated through hotfix in the next training round.

5.2. Value-sensitive model aggregation

We present the detail of the value-sensitive model aggregation process in Algorithm 1.

Algorithm 1: FedVSA training process.

Input : m^t.
Output: The trained global model $\bar{w}^*$.
1 Initialize model w_0;
2 **foreach** *CR* $t = 1, 2, \dots T$ **do**
3 $\mathcal{S}_t \leftarrow m^t$ participating clients;
4 **foreach** *client* $k \in \mathcal{S}^t$ *in parallel* **do**
5 $w_k^t, F_k(w^t) \leftarrow$ `LocalUpdate` (w^t); (Algorithm 2)
6 $\bar{\mathcal{S}}_t \leftarrow$ `Detection` $(F_k(w^t))$; (Algorithm 3)
7 Clip the $F_k(w^t)$, $k \in \bar{\mathcal{S}}_t$; $w^{t+1} \leftarrow \sum_{k \in \bar{\mathcal{S}}_t} softmax\left[F_k(w^t)\right] w_k^{t+1}$;
8 $\bar{w}^* = w^{t+1}$;
9 **return** $\bar{w}^*$

In Algorithm 1, the server initializes the training task and constructs the global model w_0. In each training round, the clients perform **LocalUpdate** (Line 5) based on the downloaded global model w^t. Notably, not all clients would continuously participate in the training process. We set the m^t as the number of clients in each round. While the server receives the local updates and inference losses from the clients, the server detects and audits the previous global model (Line 6). Any malicious or having negative impact on local updates would be removed from the aggregation. Moreover, to address the overflow problem in *softmax* function, we adapt the gradient clipping method to limit the range of the value [48].

Algorithm 2: Function **LocalUpdate**.

Input : Global model w^t, local epochs E, local batch size B.
Output: Local trained model w_k^{t+1}, inference loss $F_k(w^t)$.
1 Compute inference loss on D_k;
2 $F_k(w^t, D_k) \leftarrow \ell\left(w^t, D_k\right)$;
3 Initialize local training model $w_k^t \leftarrow w^t$;
4 $\mathcal{B} \leftarrow$ (Split D_k into batches of size B);
5 **foreach** *epoch* $e = 1, 2, \dots, E$ **do**
6 **for** *batch* $b \in \mathcal{B}$ **do**
7 $w_k^{t+1} \leftarrow w_k^t - \eta \partial F_k(w_k^t, b)$;
8 **return** $w_k^{t+1}, F_k(w^t)$

Based on vanilla local training process [77], we add a light-weight computation for inference loss (Line 2 in Algorithm 2) in **LocalUpdate**. While the client finishes the local training, the inference loss and the local updates will be transmitted to the server.

5.3. Mitigating model replacement attack

Intuitively, FedVSA prefers those clients showing high inference loss and believes they have some 'unseen' information to be learned, which can further improve global performance. However, such a unique design **may be misused by malicious clients to exaggerate their values for easier and more threatening model poisoning**. Hence, it is crucial to distinguish honest clients with qualified data from those malicious ones [71]. We specifically target the most common poisoning attacks in FL, known as model replacement attacks [2,42,84]. These attacks exploit the *mean fallacy*, which is manipulating parameters to a large value, aiming to transfer the current global model to target one. Formally, attackers can upload the arbitrary local update w_m^{t+1} to the server.

$$w_m^{t+1} = \frac{1}{p_m}(w_p - w^t) - \frac{1}{p_m}\sum_{k \in \mathcal{N}} p_k (w_k^{t+1} - w^t), \tag{10}$$

where $0 < p_k < 1$ is the weight factor of the client k during model aggregation, $1 \le p_m < |\mathcal{N}|$ is the weight factor of the malicious attacker. p_m is set to be larger than 1 to fulfill the substitution. After aggregation, the global model w^{t+1} is 'replaced' by the attacker desired model w_p.

$$\begin{aligned} w^{t+1} &= w^t + \left(\sum_{k \in \mathcal{N}} p_k (w_k^{t+1} - w^t)\right) + p_m w_m^{t+1} \\ &= w_p \end{aligned} \tag{11}$$

In this mode, the attacker needs to know the peer local updates $\sum_{k \in \mathcal{N}} p_k (w_k^{t+1} - w^t)$ and p_k. However, it is impracticable for attackers to access the peer clients' local updates, since it is high cost and easy to be exposed. To achieve the attack goal, the attacker would inject the poisoned model while the global model gets convergence. So that the peer clients' local updates will approximate to zero. For p_k, it is a static value, the attacker can obtain an approximate value by iteratively increasing (similar to the FedAvg cases).

Note that the real-time defense against malicious attacks is challenging in FL, whereas mitigation and remediation after the attack are feasible [82]. Based on this, we propose a two-step detection and recovery mechanism to confront the aforementioned threats.

We notice that the performance change is the noticeable feature of the model replacement attack. Hence, we first perform **loss comparison** using historic statistics and majority voting. We use the decreasing trend of loss value in the training process, the inference loss should not be larger than the previous maximum. An extremely large inference loss might represent a potential anomaly. If the majority of the clients report an anomaly, the updates from the last round are considered to contain malicious submissions. The loss comparison is formally described as:

$$Dr = \mathbb{I}\left\{\sum_{k \in \mathcal{N}} \mathbb{I}\left[F_k(w^t) > max(F_k(w^{t-1}))\right] \ge \frac{|\mathcal{N}|}{2}\right\}. \tag{12}$$

Where $\mathbb{I}(condition)$ is an indicator function that outputs 1 when the condition is true. If $Dr = 1$, the server continues to locate the troublemakers using the second step, namely, ***Shapley Value* evaluation**. Wherein, the *Shapley Value* is calculated as the marginal accuracy improvement of each client in one round. We use the different combinations of the local updates from different clients to reconstruct different versions of aggregated models and evaluate their performance with the marginal accuracy improvement of each local update:

$$C_k = \sum_{S \subseteq \mathcal{N} \setminus \{k\}} \frac{\mathcal{A}(w_{S \cup k}) - \mathcal{A}(w_S)}{\binom{|\mathcal{N}|-1}{|S|}}, \tag{13}$$

where $\mathcal{A}(w)$ means the accuracy of the model w, $\mathcal{S}$ is the participating clients, w_S means the model is reconstructed by $w_k^t, k \in \mathcal{S}$. A nega-

tive C_k indicates the local update may decrease the performance of the global model. Note that whenever there is an inference loss anomaly at round $t+1$, the second step mechanism will evaluate the client's *Shapley Value* at round t. The inference loss anomaly is the poisoning proof of the previous global model.

To mitigate the negative impact from the attacker, we collect all the local updates their contributions are $C_k \leq 0$ as the hotfix. The hotfix can be formally described as $-w_k^t, k \in S^t - \bar{S}^t$. The hotfix will be integrated into the global model and the local training-based model, that is $w^{t+1} = w^{t+1} - \tau \sum_{k \in S^t - \bar{S}^t} w_k^t$, τ is the scale factor of the model hotfix. In this way, the local updates can be trained based on a clean model, the negative impact is removed by using the hotfix.

Algorithm 3: Function **Detection**.

Input : Inference loss $F_k(w^t)$
Output: reaggregation client set S^t

$Dr = \mathbb{I}\left\{\sum_{k \in \mathcal{N}} \mathbb{I}\left[F_k(w^t) > max(F_k(w^{t-1}))\right] \geq \frac{|\mathcal{N}|}{2}\right\}$;
if Dr **then**
  $\bar{S}_t = S_t$;
else
  foreach *subset* $S \subseteq S_t \backslash \{k\}$ **do**
    $w_S^t \leftarrow \sum_{k \in S} softmax(F_k(w^{t-1}))w_k^{t-1}$;
  $C_k = \sum_{S \subseteq S_t \backslash \{k\}} \frac{\mathcal{A}(w_{S \cup k}^t) - \mathcal{A}(w_S^t)}{\binom{S_t - 1}{|S|}}$;
  $\bar{S}^t = \{k | C_k \geq 0\}$;
return $\bar{S}^t$

6. Experiments

6.1. Setup

6.1.1. Datasets and implementation details

We adopt three widely used datasets: MNIST, FMNIST, and CIFAR10. MNIST and FMNIST are gray-scale image datasets with a size of 28×28. MNIST includes handwritten digits 0 to 9, while FMNIST includes 10 fashion items. CIFAR10 consists of 10 classes of 32×32 RGB images. We adopt LeNet-5 [36] for MNIST, AlexNet [35] for FMNIST, and ResNet-18 [28] for CIFAR10.

To simulate non-IID data distribution, we adopt the method from [21], assigning each client only two different labels of samples. All evaluations are performed in non-IID settings. Two key parameters for our evaluations are σ, representing the local statistical variance of label quantity (see § 3), and α, indicating the ratio of fresh labels dynamically collected by clients. Besides, we also consider three ba*Dirichlet* distribution ($Dir(0.1)$, $Dir(0.3)$, and $Dir(0.5)$) for simulating the heterogeneous data distribution. Furthermore, with a total of $N = 100$ clients, we randomly select $m^t = 0.3 \times N$ of them in each CR. We set local batch size $B = 10$, local learning rate $\eta = 0.01$, and conduct local training for $E = 5$ epochs, following the setup in [47]. We adopt the test classification accuracy as a metric. We conduct short pre-training to solve the initialization problem and enable a fair comparison between different methods.

6.1.2. Baselines

We select the following two categories of algorithms as baselines for evaluating the heterogeneity-tolerant effectiveness and robustness:

- Heterogeneity-tolerant aggregations: These methods are designed to solve the heterogeneous data distribution in FL, which includes FedAvg [47], FedProx [37], SCAFFOLD [34], FedDyn [1], FedNova [67], FedDC [18], FedDisco [78], and FedLAW [40].
- Poisoning-proof aggregations: These methods are designed to mitigate malicious attacks, which include Median [79], Multi-Krum [4], and Zeno++ [74,75], DnC [56], and RAF [51].

6.2. Effectiveness

6.2.1. Convergence evaluation with different σ and $Dirichlet$ distributions

Table 3 and Table 4 present the performance in terms of the number of communication rounds (#R) required to achieve a target test accuracy for different values of σ and three *Dirichlet* distributions on three datasets. We denote the corresponding convergence speedup relative to FedVSA as 'S↑'. We adopt the same FL settings in § 3. Since many hyper-parameters of the baselines affect the model accuracy and convergent speed, we test a series of hyper-parameters in each algorithm and report the best results for each. Note that not all baselines can achieve the target accuracy in heterogeneous data distribution, we select the accuracy that most of these methods can achieve as the target accuracy.

Compared with the baselines, our method requires fewer communication rounds (∼1.52× faster on average in Table 3 and ∼1.43× faster on average in Table 4) to reach a target accuracy in those three datasets. We notice that under a more imbalanced setting (a larger σ or $Dir(0.1)$), the required communication rounds increase obviously for all the methods, as expected. Particularly, in all heterogeneous settings, Median needs maximal communication rounds to reach target accuracy, which is significantly slower than other methods. The underlying reason might be the median is not a good statistic for FL aggregation.

In summary, our algorithm can effectively reduce the impact of non-IID & class imbalanced for fast convergence relatively.

6.2.2. Classification accuracy with data dynamics

As analyzed, it was observed that the dynamic changes in a client's data can influence the label imbalance. We refer to the label that was collected recently and has never appeared before as the **fresh label**. In this scenario, we simulate data dynamics with continuously incoming fresh labels and test the performance of different methods under such settings. We set α as the proportion of fresh labels in the entire dataset, where, for instance, $\alpha = 0.1$ implies 10% of the labels are collected recently as fresh knowledge. Accordingly, we separated $\alpha \times 100\%$ labels as the fresh labels while the remaining labels are distributed in $\sigma = 900$ and $Dir(0.1)$.

We considered three different situations by setting $\alpha = 0.1$, 0.3, and 0.5. We did not consider values of α greater than 0.5 to ensure a stable training process, as a significant change in data might make convergence challenging.

As shown in Table 5, our design (i.e., FedVSA) can facilitate better performance under the tested dynamic settings. In the three datasets, FedVSA outperforms (approximately 1.6% on average) than other baselines. Furthermore, as the value of α increases, the performance difference between FedVSA and the other methods becomes even more significant. FedDC, SCAFFOLD, FedNova, and FedProx perform poorly because their added regularization in dynamic data heterogeneity settings still fails to address the challenge of large variation across clients' training results. Other methods (e.g., Zeno++), that select partial local updates for aggregation might neglect some updates so that cannot absorb the knowledge of the fresh data since the local updates trained by fresh data might be large and the peer discrepancies will also be large. Multi-Krum and Zeno++ might not trust these discrepant updates, while Median just uses the statistical median. Such cautious strategies generally dilute the value of the fresh data in global aggregation, causing more communication rounds for knowledge absorption. Overall, FedVSA outperforms by dynamically learning from 'unseen' data, leading to agile accuracy improvement.

6.3. Robustness

In this part, we compare the performance under model poisoning attacks. We simulate an adversary participating in the global training and performing model replacement, which is simple and of low cost for an adversary. Assuming that the purpose of the attackers is to disrupt the global model. To achieve this, the attacker trains a malicious model

Table 3
The communication rounds (#R) and the corresponding convergence speedup relative to FedVSA (S↑) to achieve a target test accuracy under different levels of data heterogeneity (varying σ).

MNIST, 100 participants												
Methods	$\sigma = 300$				$\sigma = 600$				$\sigma = 900$			
	90% (Acc)		95% (Acc)		90% (Acc)		95% (Acc)		90% (Acc)		95% (Acc)	
	#R	S↑	#R	S↑	#R	S↑	#R	S↑	#R	S↑	#R	S↑
FedAvg	114	4.22×	397	2.70×	146	3.74×	498	3.13×	307	4.09×	724	2.21×
FedProx	95	3.52×	378	2.57×	89	2.28×	378	2.38×	293	3.91×	658	2.01×
SCAFFOLD	45	1.67×	219	1.49×	55	1.41×	278	1.75×	200	2.67×	526	1.60×
FedNova	109	4.04×	273	1.86×	90	2.31×	421	2.65×	210	2.80×	633	1.93×
FedDyn	70	2.59×	173	1.18×	74	1.90×	218	1.37×	196	2.61×	532	1.62×
FedDC	40	1.48×	151	1.03×	46	1.18×	161	1.01×	125	1.67×	335	1.02×
FedDisco	35	1.30×	153	1.04×	42	1.08×	179	1.13×	91	1.21×	366	1.12×
FedLAW	33	1.22×	158	1.07×	46	1.18×	181	1.14×	88	1.17×	349	1.06×
Median	>1000	-	>1000	-	>1000	-	>1000	-	>1000	-	>1000	-
Multi-Krum	292	10.81×	876	5.96×	353	9.05×	>1000	-	857	11.43×	>1000	-
Zeno++	94	3.48×	296	2.01×	124	3.18×	364	2.29×	296	3.95×	659	2.01×
DnC	103	3.81×	228	1.55×	115	2.95×	289	1.82×	226	3.01×	358	1.09×
RAF	107	3.96×	268	1.82×	126	3.23×	326	2.05×	201	2.68×	394	1.20×
Ours	**27**	-	**147**	-	**39**	-	**159**	-	**75**	-	**328**	-
FMNIST, 100 participants												
Methods	$\sigma = 300$				$\sigma = 600$				$\sigma = 900$			
	78% (Acc)		85% (Acc)		78% (Acc)		85% (Acc)		78% (Acc)		85% (Acc)	
	#R	S↑	#R	S↑	#R	S↑	#R	S↑	#R	S↑	#R	S↑
FedAvg	274	4.28×	462	1.64×	366	2.07×	662	1.81×	576	2.91×	897	2.01×
FedProx	247	3.86×	457	1.63×	254	1.44×	532	1.46×	426	2.15×	794	1.78×
SCAFFOLD	203	3.17×	356	1.27×	234	1.32×	508	1.39×	364	1.84×	651	1.46×
FedNova	236	3.69×	375	1.33×	249	1.41×	526	1.44×	475	2.40×	745	1.67×
FedDyn	90	1.41×	**269**	**0.96**×	222	1.25×	439	1.20×	297	1.50×	483	1.08×
FedDC	114	1.78×	294	1.05×	216	1.22×	411	1.13×	237	1.20×	459	1.03×
FedDisco	125	1.95×	308	1.10×	188	1.06×	389	1.07×	221	1.12×	455	1.02×
FedLAW	108	1.22×	305	1.09×	192	1.08×	396	1.08×	201	1.02×	444	**0.99**×
Median	>1000	-	>1000	-	>1000	-	>1000	-	>1000	-	>1000	-
Multi-Krum	398	6.22×	943	3.36×	539	3.05×	>1000	-	891	4.50×	>1000	-
Zeno++	229	3.58×	413	1.47×	354	2.00×	649	1.78×	404	2.04×	761	1.71×
DnC	126	1.97×	378	1.35×	195	1.10×	542	1.48×	267	1.35×	695	1.56×
RAF	106	1.66×	364	1.30×	221	1.25×	589	1.61×	254	1.28×	712	1.60×
Ours	**64**	-	281	-	**177**	-	**365**	-	**198**	-	**446**	-
CIFAR10, 100 participants												
Methods	$\sigma = 300$				$\sigma = 600$				$\sigma = 900$			
	55% (Acc)		65% (Acc)		55% (Acc)		60% (Acc)		45% (Acc)		50% (Acc)	
	#R	S↑	#R	S↑	#R	S↑	#R	S↑	#R	S↑	#R	S↑
FedAvg	227	3.15×	975	9.47×	289	3.32×	557	3.82×	321	2.70×	675	3.18×
FedProx	224	3.11×	691	6.71×	286	3.29×	544	3.73×	318	2.67×	632	2.98×
SCAFFOLD	117	1.63×	188	1.83×	143	1.64×	294	2.01×	221	1.86×	345	1.63×
FedNova	316	4.39×	633	6.15×	283	3.25×	469	3.21×	217	1.82×	322	1.52×
FedDyn	88	1.22×	114	1.11×	106	1.22×	216	1.48×	129	1.08×	**205**	**0.97**×
FedDC	96	1.33×	165	1.60×	117	1.34×	268	1.84×	198	1.66×	301	1.42×
FedDisco	88	1.22×	121	1.17×	105	1.21×	186	1.27×	135	1.13×	216	1.02×
FedLAW	97	1.35×	130	1.26×	115	1.32×	199	1.36×	149	1.25×	221	1.04×
Median	>1000	-	>1000	-	>1000	-	>1000	-	>1000	-	>1000	-
Multi-Krum	454	6.31×	>1000	-	295	3.39×	594	4.07×	354	2.97×	711	3.35×
Zeno++	217	3.01×	607	5.89×	262	3.01×	433	2.97×	291	2.45×	433	2.04×
DnC	128	1.78×	200	1.94×	169	1.94×	368	2.52×	211	1.77×	596	2.81×
RAF	133	1.85×	223	2.17×	177	2.03×	384	2.63×	236	1.98×	565	2.67×
Ours	**72**	-	**103**	-	**87**	-	**146**	-	**119**	-	212	-

by using label-flipped data, leading to predictions that are totally different from the ground truth. For clarity, we evaluate the robustness of FedVSA by comparing it with vanilla aggregation method (i.e., FedAvg) and existing robust aggregation methods, such as Multi-Krum, Zeno++, Median, DnC, and RAF. The results are shown in Fig. 3. The attack is performed at fixed rounds, 10^{th} for the non-convergence phase and $25^{th}, 50^{th}$ for the convergence phase.

During an attack, the global model's accuracy drops to 0%, signifying its destruction. After the attack, our solution can detect the anomalies and perform reaggregation in the following round. With this, the malicious local updates are identified and filtered, mitigating the negative impact of the model replacement attack. However, FedAvg requires some training rounds to recover from the attack. Notably, not all attacks can gradually recover from training. Stronger attacks may lead to model divergence or destruction. For Zeno++, Krum, Median, DnC, and RAF all experience minimal accuracy drop due to unchanged gradient magnitudes with model replacement. Thus, when malicious gradients are less than half, it is possible for Multi-Krum and Median to filter them

Table 4
The communication rounds (#R) and the corresponding convergence speedup relative to FedVSA (S↑) to achieve a target test accuracy under different Dirichlet distributions ($Dir(0.1)$, $Dir(0.3)$, and $Dir(0.5)$).

MNIST, 100 participants												
Methods	$Dir(0.1)$				$Dir(0.3)$				$Dir(0.5)$			
	90% (Acc)		95% (Acc)		90% (Acc)		95% (Acc)		90% (Acc)		95% (Acc)	
	#R	S↑	#R	S↑	#R	S↑	#R	S↑	#R	S↑	#R	S↑
FedAvg	90	3.91×	456	5.77×	31	2.58×	188	4.27×	25	3.13×	163	4.94×
FedProx	58	2.52×	379	4.80×	35	2.92×	202	4.59×	27	3.38×	160	4.85×
SCAFFOLD	39	1.70×	178	2.25×	25	2.08×	135	3.07×	21	2.63×	121	3.67×
FedNova	47	2.04×	267	3.38×	23	1.92×	119	2.70×	14	1.75×	88	2.67×
FedDyn	51	2.22×	127	1.61×	21	1.75×	86	1.95×	21	2.63×	61	1.85×
FedDC	39	1.70×	115	1.46×	19	1.58×	76	1.73×	18	2.25×	55	1.67×
FedDisco	41	1.78×	125	1.58×	13	1.08×	61	1.39×	11	1.38×	42	1.27×
FedLAW	32	1.39×	105	1.33×	11	**0.92**×	56	1.27×	12	1.50×	49	1.48×
Median	73	3.17×	323	4.09×	20	1.67×	78	1.77×	24	3.00×	60	1.82×
Multi-Krum	93	4.04×	586	7.42×	36	3.00×	262	5.95×	27	3.38×	168	5.09×
Zeno++	67	2.91×	426	5.39×	27	2.25×	201	4.57×	23	2.88×	154	4.67×
DnC	54	2.35×	265	3.35×	32	2.67×	215	4.89×	15	1.88×	122	3.70×
RAF	49	2.13×	245	3.10×	29	2.42×	225	5.11×	18	2.25×	136	4.12×
Ours	**23**	-	**79**	-	**12**	-	**44**	-	**8**	-	**33**	-
FMNIST, 100 participants												
Methods	$Dir(0.1)$				$Dir(0.3)$				$Dir(0.5)$			
	70% (Acc)		80% (Acc)		70% (Acc)		80% (Acc)		70% (Acc)		80% (Acc)	
	#R	S↑	#R	S↑	#R	S↑	#R	S↑	#R	S↑	#R	S↑
FedAvg	57	3.80×	203	3.22×	47	4.27×	123	4.24×	30	3.75×	82	4.10×
FedProx	56	3.73×	279	4.43×	33	3.00×	102	3.52×	16	2.00×	66	3.30×
SCAFFOLD	29	1.93×	138	2.19×	30	2.73×	84	2.90×	17	2.13×	52	2.60×
FedNova	35	2.33×	165	2.62×	25	2.27×	85	2.93×	13	1.63×	55	2.75×
FedDyn	26	1.73×	81	1.29×	23	2.09×	64	2.21×	14	1.75×	37	1.85×
FedDC	27	1.80×	79	1.25×	22	2.00×	50	1.72×	15	1.88×	45	2.25×
FedDisco	23	1.53×	71	1.13×	20	1.82×	44	1.52×	12	1.50×	44	2.20×
FedLAW	20	1.33×	68	1.08×	18	1.64×	35	1.21×	16	2.00×	33	1.65×
Median	41	2.73×	175	2.78×	45	4.09×	134	4.62×	42	5.25×	128	6.40×
Multi-Krum	87	5.80×	360	5.71×	39	3.55×	122	4.21×	21	2.63×	84	4.20×
Zeno++	47	3.13×	262	4.16×	33	3.00×	121	4.17×	25	3.13×	71	3.55×
DnC	39	2.60×	159	2.52×	28	2.55×	67	2.31×	19	2.38×	39	1.95×
RAF	32	2.13×	146	2.32×	26	2.36×	64	2.21×	17	2.13×	35	1.75×
Ours	**15**	-	**63**	-	**11**	-	**29**	-	**8**	-	**20**	-
CIFAR10, 100 participants												
Methods	$Dir(0.1)$				$Dir(0.3)$				$Dir(0.5)$			
	50% (Acc)		60% (Acc)		50% (Acc)		60% (Acc)		50% (Acc)		60% (Acc)	
	#R	S↑	#R	S↑	#R	S↑	#R	S↑	#R	S↑	#R	S↑
FedAvg	182	3.03×	363	2.77×	121	3.03×	218	2.63×	68	2.83×	136	3.24×
FedProx	204	3.40×	297	2.27×	95	2.38×	174	2.10×	67	2.79×	133	3.17×
SCAFFOLD	145	2.42×	198	1.51×	72	1.80×	131	1.58×	61	2.54×	112	2.67×
FedNova	162	2.70×	347	2.65×	100	2.50×	203	2.45×	80	3.33×	158	3.76×
FedDyn	86	1.43×	188	1.44×	55	1.38×	113	1.36×	47	1.96×	97	2.31×
FedDC	119	1.98×	205	1.56×	54	1.35×	102	1.23×	43	1.79×	82	1.95×
FedDisco	77	1.28×	154	1.18×	46	1.15×	99	1.19×	36	1.50×	65	1.55×
FedLAW	69	1.15×	142	1.08×	49	1.23×	101	1.22×	33	1.38×	59	1.40×
Median	219	3.65×	415	3.17×	143	3.58×	181	2.18×	95	3.96×	169	4.02×
Multi-Krum	218	3.63×	435	3.32×	93	2.33×	187	2.25×	72	3.00×	134	3.19×
Zeno++	183	3.05×	343	2.62×	79	1.98×	185	2.23×	78	3.25×	138	3.29×
DnC	155	2.58×	251	1.92×	72	1.80×	136	1.64×	68	2.83×	112	2.67×
RAF	102	1.70×	211	1.61×	67	1.68×	122	1.47×	65	2.71×	108	2.57×
Ours	**60**	-	**131**	-	**40**	-	**83**	-	**24**	-	**42**	-

out. Yet, enlarging the number of malicious gradients may cause them to fail with a static filtering strategy.

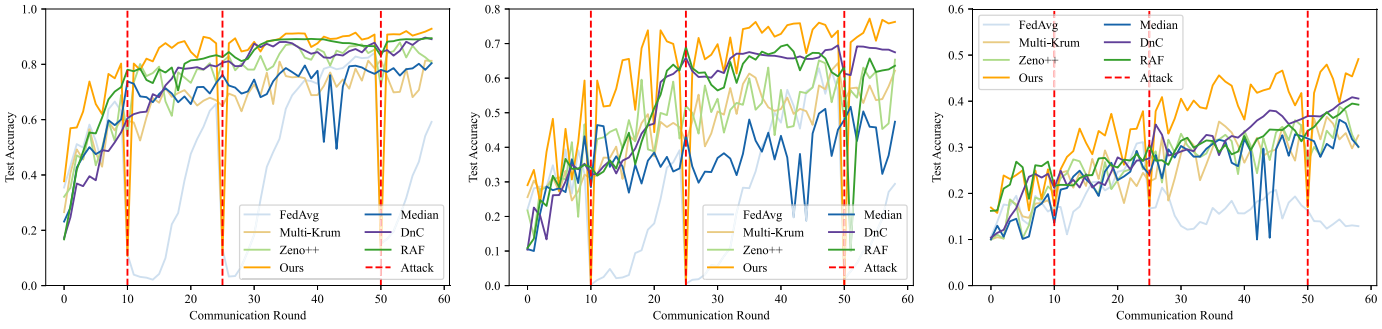
6.4. Scalability

In this part, we evaluate the scalability of the FedVSA under **different scales of distributed networks** and **different model architectures**.

For different scales of distributed networks, we vary the number of clients in federated learning and compare the performance under two severe heterogeneous data distributions ($\sigma = 900$ and $Dir(0.1)$). As shown in Table 6, we conclude that in different scales of distributed networks, FedVSA can still achieve the best performance than existing baselines. Besides, we find that with the increment of the clients, the performance degraded significantly. The underlying reason is that numerous heterogeneous clients make it more difficult for one global model to converge to a middle-ground state.

Table 5
Classification accuracy (%) on three datasets with dynamic data distribution adjustment controlled by factor α.

MNIST, 100 participants														
Settings		FedAvg	FedProx	SCAFFOLD	FedNova	FedDyn	FedDC	FedDisco	FedLAW	Median	Multi-Krum	Zeno++	DnC	RAF	Ours
$\sigma = 900$	$\alpha = 0.1$	89.85	89.98	92.95	90.18	92.82	93.89	94.97	94.66	74.44	81.47	87.69	90.44	90.48	**95.48**
	$\alpha = 0.3$	89.77	89.63	93.65	90.22	94.79	93.79	94.98	92.15	71.29	78.26	86.37	89.42	89.25	**95.12**
	$\alpha = 0.5$	84.94	84.93	92.51	89.07	88.96	92.84	92.91	91.54	77.14	77.38	86.14	89.91	88.38	**93.05**
$Dir(0.1)$	$\alpha = 0.1$	92.33	93.11	94.18	93.21	93.64	94.86	94.99	95.04	88.25	89.57	90.25	92.54	93.45	**95.44**
	$\alpha = 0.3$	92.18	92.38	93.83	92.22	92.01	93.65	93.58	93.72	86.99	87.61	88.99	91.15	92.68	**94.21**
	$\alpha = 0.5$	90.18	91.25	92.69	91.25	91.11	91.21	92.59	92.89	83.61	84.94	83.47	89.91	90.12	**94.04**
FMNIST, 100 participants														
Settings		FedAvg	FedProx	SCAFFOLD	FedNova	FedDyn	FedDC	FedDisco	FedLAW	Median	Multi-Krum	Zeno++	DnC	RAF	Ours
$\sigma = 900$	$\alpha = 0.1$	76.22	76.31	81.81	78.52	74.61	79.86	79.85	80.01	64.59	64.33	76.65	77.68	78.35	**83.08**
	$\alpha = 0.3$	71.79	71.73	80.81	78.21	75.58	78.31	77.89	78.95	56.64	54.77	75.64	76.48	77.95	**81.98**
	$\alpha = 0.5$	76.28	76.26	80.21	74.03	75.83	77.90	76.84	78.14	53.86	47.25	70.33	75.12	76.54	**81.38**
$Dir(0.1)$	$\alpha = 0.1$	73.27	74.76	81.81	80.49	81.41	82.22	83.66	82.65	71.25	74.52	76.54	81.22	79.85	**84.47**
	$\alpha = 0.3$	72.69	73.14	80.02	78.46	80.52	81.04	82.15	81.27	69.22	71.02	74.25	77.51	76.46	**82.02**
	$\alpha = 0.5$	70.41	71.12	78.24	76.62	78.28	80.09	81.31	80.05	68.24	69.45	72.25	76.21	75.91	**81.41**
CIFAR10, 100 participants														
Settings		FedAvg	FedProx	SCAFFOLD	FedNova	FedDyn	FedDC	FedDisco	FedLAW	Median	Multi-Krum	Zeno++	DnC	RAF	Ours
$\sigma = 900$	$\alpha = 0.1$	63.08	62.66	63.27	66.51	65.26	69.12	71.56	72.04	58.86	62.02	62.87	63.25	64.58	**73.42**
	$\alpha = 0.3$	62.72	62.98	62.55	62.85	64.68	68.08	69.54	70.11	58.54	66.56	63.84	61.95	63.45	**73.12**
	$\alpha = 0.5$	61.03	62.53	61.23	58.94	64.97	61.2	68.42	69.85	48.69	41.62	62.13	61.55	62.25	**72.62**
$Dir(0.1)$	$\alpha = 0.1$	73.02	73.69	74.66	73.98	74.05	76.15	77.25	77.12	68.92	71.05	74.05	75.25	73.54	**78.04**
	$\alpha = 0.3$	72.03	72.54	73.12	72.15	72.58	75.12	76.69	76.58	66.51	70.25	72.51	72.25	71.98	**77.23**
	$\alpha = 0.5$	70.55	71.15	72.15	71.98	71.12	72.54	74.25	74.19	63.25	68.79	71.99	71.69	70.25	**75.15**



(a) MNIST (b) FMNIST (c) CIFAR10

Fig. 3. Training process on three datasets when there exist model replacement attacks.

Table 6
The performance (%) of different methods on different scales of distributed networks (varying the number of clients) under two heterogeneous data distributions ($\sigma = 900$ and $Dir(0.1)$) on CIFAR-10.

# Clients	Settings	FedAvg	FedProx	SCAFFOLD	FedNova	FedDyn	FedDC	FedDisco	FedLAW	Median	Multi-Krum	Zeno++	DnC	RAF	Ours
10	$\sigma = 900$	70.41	71.23	72.56	72.09	72.43	74.45	75.09	74.89	69.10	71.23	70.44	72.12	71.98	**77.45**
50		68.32	68.98	70.45	70.32	70.67	72.57	73.34	73.14	67.36	69.23	70.10	68.97	69.45	**74.12**
100		63.08	62.66	63.27	66.51	65.26	69.12	71.56	72.04	58.86	62.02	62.87	63.25	64.58	**73.42**
500		45.12	45.55	48.11	48.43	48.88	52.98	53.38	54.95	41.14	45.71	48.14	45.71	48.61	**55.24**
10	$Dir = 0.1$	84.23	84.49	86.53	86.11	85.11	87.72	87.99	88.12	78.44	82.00	85.15	86.91	86.68	**89.14**
50		78.66	78.98	79.46	79.10	80.12	81.11	81.56	81.29	74.24	76.45	79.43	80.01	79.94	**82.35**
100		73.02	73.69	74.66	73.98	74.05	76.15	77.25	77.12	68.92	71.05	74.05	75.25	73.54	**78.04**
500		48.99	52.17	53.71	56.16	55.34	57.66	57.46	57.86	47.84	49.05	50.13	52.22	54.33	**58.93**

For the models with different architectures, we select three widely used model architectures and compare the FedVSA with different methods in Table 7. We conclude that FedVSA can perform well in most of the architectures. Besides, it is surprising to find that some model-relative methods (i.e., FedProx, Median) perform worse than FedAvg in some simple model architectures (MLP).

In summary, we demonstrate the scalability under different scales of distributed networks and different model architectures and show that FedVSA outperforms other baselines.

6.5. Ablation study

The FedVSA consists of two main parts: the value-sensitive model aggregation module and the two-step detection mechanism module. For the value-sensitive model aggregation module, we investigate the impact of different inference loss functions (§ 6.5.1), the gradient clipping operation (§ 6.5.2), and the performance improvements over baselines (§ 6.5.3). For the latter, we investigate the impacts of each detection step under adversary settings.

Table 7
The performance (%) of different methods on different model architectures (MLP, CNN, and ResNet-18) under two heterogeneous data distributions ($\sigma = 900$ and $Dir(0.1)$) on CIFAR-10.

Model Architecture	Settings	FedAvg	FedProx	SCAFFOLD	FedNova	FedDyn	FedDC	FedDisco	FedLAW	Median	Multi-Krum	Zeno++	DnC	RAF	Ours
MLP		58.14	57.12	59.45	57.11	57.09	60.07	60.89	60.22	34.76	56.45	55.67	59.76	59.04	**61.09**
CNN	$\sigma = 900$	59.89	58.44	58.66	59.52	58.01	61.5	62.14	63.77	41.59	58.09	59.45	60.9	60.23	**63.56**
ResNet-18		63.08	62.66	63.27	66.51	65.26	69.12	71.56	72.04	58.86	62.02	62.87	63.25	64.58	**73.42**
MLP		63.34	66.63	67.45	67.1	66.01	68.59	69.43	68.41	43.76	62.58	63.07	65.63	63.82	**70.33**
CNN	$Dir = 0.1$	70.11	69.46	71.77	71.41	71.89	**73.32**	73.08	73.44	54.67	68.79	69.09	71.76	70.98	72.44
ResNet-18		73.02	73.69	74.66	73.98	74.05	76.15	77.25	77.12	68.92	71.05	74.05	75.25	73.54	**78.04**

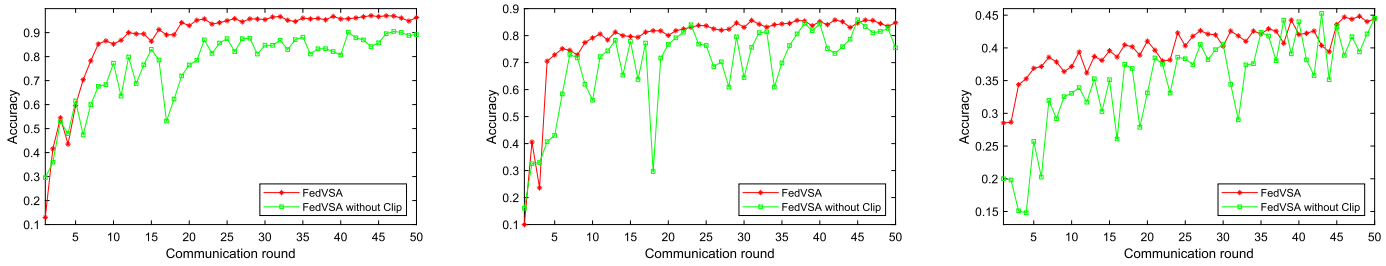


Fig. 4. The impact of clipping inference loss on the training process.

Table 8
The impact of different loss functions on FedVSA.

ℓ_{local}		ℓ_{inf}			Accuracy (%)
CE	MSE	CE	MSE	L1	
✓		✓			72.55
✓			✓		72.13
✓				✓	72.10
	✓	✓			71.28
	✓		✓		71.41
	✓			✓	71.16

6.5.1. Impact of different inference loss function

In this part, we investigate the impact of different inference loss functions. In our deduction in Equ. (6), the inference loss function should align with the local loss function. To better illustrate the impacts of different inference loss functions, we vary the loss function on both the local training loss function ℓ_{local} and inference loss function ℓ_{inf}. We select three widely used loss functions, (i.e., L1, MSE, and CE). Note that the model can not get convergent when $\ell_{local} = L1$, we omit this setting in our evaluation. As shown in Table 8, we concluded that the local training loss function (ℓ_{local}) can influence the final performance of different algorithms, and different loss functions can converge to different optimal, as proved in [38] and Theorem 1. Besides, the consistency of ℓ_{inf} and ℓ_{local} (the assumption in Equ. (6)) can achieve the best performance than those inconsistent settings.

6.5.2. Impact of gradient clipping

We investigate the influence of gradient clipping on local inference loss, as presented in Fig. 4. Notably, the performance curves exhibit instability when inference loss is not clipped. This instability suggests that certain clients with extreme inference losses cause overfitting of the global model, resulting in fluctuations. Thus, this highlights the importance of gradient clipping the inference loss in FedVSA.

6.5.3. Performance improvements over baselines

One key advantage of our proposed FedVSA is its modularity and compatibility, that is, it can be a plug-and-play module to improve the performance of the existing FL method. Note that the FedNova, FedDisco, and FedLAW are designed to optimize the local model weight and Median use the median, these methods are not compatible with the FedVSA. As shown in Table 9, across most baselines, FedVSA improves their performance with a significant gap, providing evidence that static averaging may not be the optimal method for aggregation.

Table 9
The performance (%) of different algorithms w/ and w/o FedVSA.

Methods	w/o FedVSA	w/ FedVSA
FedAvg	87.69	93.76 (+6.07 ↑, **FedVSA**)
FedProx	87.76	89.49 (+1.73↑)
SCAFFOLD	92.55	93.44 (+0.89 ↑)
FedNova	89.54	/
FedDyn	92.54	93.17 (+0.63 ↑)
FedDC	93.46	94.56 (+1.10 ↑)
FedDisco	93.32	/
FedLAW	92.87	/
Median	74.59	/
Krum/Multi-Krum	77.35	77.44 (+0.09 ↑)
Zeno++	87.35	87.66 (+0.31 ↑)
DnC	88.32	91.22 (+2.90 ↑)
RAF	89.46	92.13 (+2.67 ↑)

6.5.4. Impact of two-step detection mechanism

To investigate the impact of the two-step detection mechanism, we adopt the adversary setting in § 6.3. We collect the performance and computational time of FedVSA w/o each detection step, as shown in Fig. 5. We conclude that the FedVSA without loss comparison can achieve more smooth performance improvement, yet require high computational time. The FedVSA without *Shapley Value* cannot ensure the global model can fully recover from the malicious attack. The two-step detection mechanism design can achieve both robustness and low computation overhead.

7. Discussion

7.1. Overheads

Note that our solution is practical, although some extra computation and communication overheads are required. The computation overhead mainly comes from the inference loss computation in each client and the *Shapley Value* computation in the server.

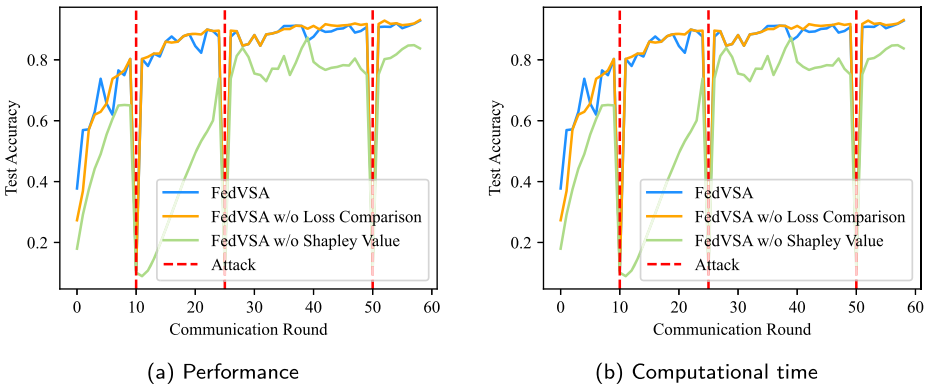


Fig. 5. The performance and computational time of FedVSA w/o each detection step.

Table 10
The client-side and server-side computational overhead of 8 algorithms. E is the number of local epochs, T_train is the local training time, $T_{DC-train}$ is the local training time of FedDC. T_{agg} is the aggregation time on the server, and T_{zeno} is the time of computing scores of each local model. T_{DnC} and T_{RAF} are the computational time for robust detection in DnC and RAF, respectively. T_{comp} is the time for inference loss comparison, T_{Shap} for computing *Shapley Value*, and $T_{ETMC-Shap}$ for computing Extended-TMC-Shapley.

Methods	Client-side		Server-side	
FedAvg	$E \times T_{train}$	4.75 s	T_{agg}	0.31 s
FedDC	$E \times T_{DC-train}$	6.23 s	T_{agg}	0.43 s
Zeno++	$E \times T_{train}$	4.75 s	$T_{zeno} + T_{agg}$	2.98 s
DnC	$E \times T_{DnC-train}$	4.75 s	$T_{DnC} + T_{agg}$	6.17 s
RAF	$E \times T_{RAF-train}$	4.75 s	$T_{RAF} + T_{agg}$	4.56 s
FedVSA w/o Shapley Value	$(E+1) \times T_{train}$	5.52 s	$T_{comp} + T_{agg}$	0.33 s
FedVSA w/ Shapley Value	$(E+1) \times T_{train}$	5.52 s	$T_{comp} + T_{Shap} + T_{agg}$	14.34 s
FedVSA w/ Extended-TMC-Shapley	$(E+1) \times T_{train}$	5.52 s	$T_{comp} + T_{ETMC-Shap} + T_{agg}$	5.63 s

For computation overhead, we use the computational time as the metrics [7]. On the client side, baselines generally need $E \times T_{train}$, while our solution only needs $(E+1) \times T_{train}$, wherein T_{train} is the one epoch training time in the client. On the server side, the computation resource is sufficient and the server does not need to additionally compute the *Shapley Value* in every round. The server needs $2^{m_t} \times T_{test}$, wherein T_{test} is the evaluation time on the test set. We can use the Extended-TMC-Shapley [61] to reduce the computation overhead further. To this end, We demonstrate the computational time of each process in Table 10. We execute them in the same hardware device (a PC with a 2.4 GHz CPU, 32G RAM, and RTX 3070 GPU). We chose FedAvg and FedDC as the baselines without attack-proof, and Zeno++ as the robust baseline. Note that the computation of *Shapley Value* does not exist in every round, it only performs when the inference loss comparison reports an abnormal update. Besides, in our design, the estimation of *Shapley Value* is computed in the server, we consider the server to be resourceful and has enough time to compute while the client performing local training. For an infrequent malicious attack, the cost of FedVSA w/o *Shapley Value* is still practical.

Regarding communication overhead, we quantify the size of the transmission message using the number of floats [25]. M represents the model size. The communication overhead of baselines is shown in Table 11. Emphatically, the estimation of *Shapley Value* would not introduce the communication overhead. The proposed method only needs an additional float compared with FedAvg but can achieve fast convergence and robust aggregation.

Table 11
The communication overhead of different algorithms.

Methods	upload↑	download↓
FedAvg	M	M
FedProx	M	M
SCAFFOLD	$2M$	$2M$
FedNova	M	M
FedDyn	M	M
FedDC	$3M$	$3M$
FedDisco	M	M
FedLAW	M	M
Median	M	M
Multi-Krum	M	M
Zeno++	M	M
DnC	M	M
RAF	M	M
Ours	M+1	M

7.2. Authenticity of the updates

In FedVSA, clients upload both the inference loss and local updates, which the server collects and utilizes for aggregation. Therefore, ensuring the authenticity of these updates is crucial. Uploading fabricated or false inference loss could pose unforeseen threats to the global model. In practice, we highlight the use of TEEs to address this issue, such as SGX [49], which guarantees the authenticity of the local calculation process.

7.3. Quality of the local data

Our design relies on the premise that the inference loss reflects the model's deficiency of local data. Nevertheless, if the local data is of low quality, the corresponding updates can be blocked by our design. In other words, our method can only accelerate the global training process if the local data is of good quality (e.g., without wrong labels).

8. Conclusion

In this work, we verify the performance of FedAvg in various non-IID & class imbalance settings. We observe and conclude that the distribution of classes inside the clients will greatly influence FL performance. We consider the global model to fall into the *mean fallacy* due to static averaging. Current methods only perform simple averaging of the local models in a numerical manner, rather than aggregating their knowledge. To address this problem, we propose the technical design and theoretical analysis of a novel model aggregation framework based on evaluating local merits, named FedVSA. Its performance is tested on three different datasets, which show that FedVSA needs fewer communication rounds for convergence and can obtain higher accuracy with both various data heterogeneity settings and data dynamics. We also design a detection mechanism using loss comparison and *Shapley Value* for mitigating the poisoning vulnerability enhanced with our value-sensitive design. Corresponding experiments show that our methods can well defend against such attacks.

CRediT authorship contribution statement

Hui Zeng: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Tongqing Zhou:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Yeting Guo:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Zhiping Cai:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Fang Liu:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. 62172155, 62072465, and 62102425), the National Key Research and Development Program of China (2022YFF1203001), the Science and Technology Innovation Program of Hunan Province (Nos. 2022RC3061 and 2023RC3027).

Appendix A. Properties analysis details on FedVSA

A.1. The proof of Theorem 1

Here, we present the proof of Theorem 1.

Proof. First, we note that if we want to prove $F(w)$ is convex, $F(w)$ is required to satisfy all requirements in Definition 2. We assume that for all client k, $F_k(w)$ is convex, and using Definition 2, $\forall \beta \in [0,1]$, we have

$$F_k(\beta w_1 + (1-\beta)w_2) \le \beta F_k(w_1) + (1-\beta)F_k(w_2) \tag{14}$$

(1) $dom(F) \subset R^z$ is a convex set. Given Eq. (6), for $F_k(w) \ge 0$, we can easily get $F(w) \ge 0$ and $dom(F) = \{w | w \ge 0\}$. According to the definition of the convex set, for any $w_1, w_2 \in dom(F)$ and two real numbers $\beta_1, \beta_2 \ge 0$, $\beta_1 + \beta_2 = 1$, we all have $\beta_1 w_1 + \beta_2 x_2 \in dom(F)$, which proves $dom(F)$ is a convex set.

(2) $F(\beta w_1 + (1-\beta)w_2) \le \beta F(w_1) + (1-\beta)F_k(w)$. In this part, we define two variables $g_1 = \beta F(w_1) + (1-\beta)F(w_2)$ and $g_2 = F(\beta w_1 + (1-\beta)w_2)$. Then, for g_1, we have:

$$\begin{aligned} g_1 &= \beta \ln\left(\sum_k^N e^{F_k(w_1)}\right) + (1-\beta)\ln\left(\sum_k^N e^{F_k(w_2)}\right) \\ &= \ln\left(\sum_k^N e^{F_k(w_1)}\right)^{\beta} + \ln\left(\sum_k^N e^{F_k(w_2)}\right)^{(1-\beta)} \\ &= \ln\left(\left(\sum_k^N e^{F_k(w_1)}\right)^{\beta}\left(\sum_k^N e^{F_k(w_2)}\right)^{(1-\beta)}\right) \end{aligned} \tag{15}$$

For any w, we have $F_k(w) \ge 0$, so we get $e^{F_k(w)} \ge 1$. When $\beta \in [0,1]$, we can get

$$g_1 \ge \ln\left(\sum_k^N e^{\beta F_k(w_1) + (1-\beta)F_k(w_2)}\right) = g_2 \tag{16}$$

According to Definition 2 and the assumption that all $F_k(w)$ is convex, we can get

$$F_k(\beta w_1 + (1-\beta)w_2) \le \beta F_k(w_1) + (1-\beta)F_k(w_2). \tag{17}$$

Further, we also have

$$\beta F(w_1) + (1-\beta)F(w_2) \ge F(\beta w_1 + (1-\beta)w_2), \tag{18}$$

which proves that $F(w)$ satisfies the second requirement. Another way to prove this is to verify $\nabla_w^2 F(w) \ge 0$. It still satisfies. So we conclude that if for all client k, $F_k(w) \ge 0$ and $F_k(w)$ is convex, then $F(w)$ is convex. Hence, we finish the proof of the Theorem 1. □

Based on the Theorem 1, we have the following two lemmas.

Lemma 1 *(L-smoothness of $F(w)$). If the function $F_k(w)$ is L-smooth and μ-strongly convex, then $F(w)$ is L-smooth.*

Proof. First, we provide the definition of *L-smooth* property. A function $\Theta : X \to \mathbb{R}$ is *L-smooth* if for all $x, y \in X$ in its domain, the following inequality holds:

$$\|\nabla\Theta(x) - \nabla\Theta(y)\| \le L\|x - y\|. \tag{19}$$

According to the definitions of the global loss function in Definition 1, we have

$$F(w) = \ln \sum_{k\in\mathcal{N}} e^{F_k(w)}. \tag{20}$$

Using the chain rule, the gradient of $F(w)$ can be derived as:

$$\begin{aligned} \nabla F(w) &= \frac{1}{\sum_{m\in\mathcal{N}} e^{F_m(w)}} \sum_{k\in\mathcal{N}} e^{F_k(w)} \nabla F_k(w) \\ &= \sum_{k\in\mathcal{N}} \frac{e^{F_k(w)}}{\sum_{m\in\mathcal{N}} e^{F_m(w)}} \nabla F_k(w). \end{aligned} \tag{21}$$

For any x, y, by Cauchy-Schwarz inequality and the *μ-strongly convexity* of F_k, we denote the $Left = \|\nabla F(x) - \nabla F(y)\|$

$$
\begin{aligned}
Left = & \\
& \left\| \sum_{k\in\mathcal{N}} \frac{e^{F_k(x)}}{\sum_{m\in\mathcal{N}} e^{F_m(x)}} \nabla F_k(x) - \sum_{k\in\mathcal{N}} \frac{e^{F_k(y)}}{\sum_{m\in\mathcal{N}} e^{F_m(y)}} \nabla F_k(y) \right\| \\
& \leq \sum_{k\in\mathcal{N}} \left\| \frac{e^{F_k(x)}}{\sum_{m\in\mathcal{N}} e^{F_m(x)}} \nabla F_k(x) - \frac{e^{F_k(y)}}{\sum_{m\in\mathcal{N}} e^{F_m(y)}} \nabla F_k(y) \right\| \\
& \leq \sum_{k\in\mathcal{N}} \left\| \frac{e^{F_k(x)}}{\sum_{m\in\mathcal{N}} e^{F_m(y)+(x-y)^T\nabla F_m(y)+\frac{\mu}{2}\|x-y\|_2^2}} \nabla F_k(x) \right. \\
& \left. - \frac{e^{F_k(y)}}{\sum_{m\in\mathcal{N}} e^{F_m(y)}} \nabla F_k(y) \right\| \\
& \leq \sum_{k\in\mathcal{N}} \left\| \frac{e^{F_k(x)}}{e^{\frac{\mu}{2}\|x-y\|_2^2} \sum_{m\in\mathcal{N}} e^{F_m(y)+(x-y)^T\nabla F_m(y)}} \nabla F_k(x) \right. \\
& \left. - \frac{e^{F_k(y)}}{e^{\frac{\mu}{2}\|x-y\|_2^2} \sum_{m\in\mathcal{N}} e^{F_m(y)}} \nabla F_k(y) \right\|
\end{aligned} \tag{22}
$$

By

$$
\begin{aligned}
& \frac{e^{F_k(y)}}{e^{\frac{\mu}{2}\|x-y\|_2^2} \sum_{m\in\mathcal{N}} e^{F_m(y)}} \nabla F_k(y) \geq \\
& \frac{e^{F_k(y)+\min_{m\in\mathcal{N}}(x-y)^T\nabla F_m(y)}}{e^{\frac{\mu}{2}\|x-y\|_2^2} \sum_{m\in\mathcal{N}} e^{F_m(y)+(x-y)^T\nabla F_m(y)}} \nabla F_k(y),
\end{aligned} \tag{23}
$$

we have

$$
\begin{aligned}
& \|\nabla F(x) - \nabla F(y)\| \leq \\
& \sum_{k\in\mathcal{N}} \left\| \frac{e^{F_k(y)+(x-y)^T\nabla F_m(y)}}{e^{\frac{\mu}{2}\|x-y\|_2^2} \sum_{m\in\mathcal{N}} e^{F_m(y)+(x-y)^T\nabla F_m(y)}} (\nabla F_k(x) - \nabla F_k(y)) \right\|.
\end{aligned} \tag{24}
$$

By the *L-smoothness* of F_k, we have

$$
\begin{aligned}
& \|\nabla F(x) - \nabla F(y)\| \leq \\
& \sum_{k\in\mathcal{N}} \left\| \frac{e^{F_k(y)+(x-y)^T\nabla F_m(y)}}{e^{\frac{\mu}{2}\|x-y\|_2^2} \sum_{m\in\mathcal{N}} e^{F_m(y)+(x-y)^T\nabla F_m(y)}} L_k\|x-y\| \right\| \\
& \leq \sum_{k\in\mathcal{N}} \left\| \frac{L_k\|x-y\|}{e^{\frac{\mu}{2}\|x-y\|_2^2}} \right\| \leq \sum_{k\in\mathcal{N}} L_k\|x-y\| \leq L\|x-y\|
\end{aligned} \tag{25}
$$

Thus, we finish the proof of the *L-smoothness* of F. □

Lemma 2 *(μ-strongly convexity of $F(w)$). If the function $F_k(w)$ is L-smooth and μ-strongly convex, then $F(w)$ is μ-strongly convex.*

Proof. According to the definition of *μ-strongly convexity*, a function $\Theta : X \to \mathbb{R}$ is *μ-strongly convex* if for all $x, y \in X$, the following holds:

$$
\|\nabla\Theta(x) - \nabla\Theta(y)\| \geq \frac{\mu}{2}\|x-y\| \tag{26}
$$

We use the Equ. (21), for any x, y, by *L-smoothness* of F_k, we denote $Left = \|\nabla F(x) - \nabla F(y)\|$

$$
\begin{aligned}
Left = & \left\| \sum_{k\in\mathcal{N}} \frac{e^{F_k(x)}}{\sum_{m\in\mathcal{N}} e^{F_m(x)}} \nabla F_k(x) \right. \\
& \left. - \sum_{k\in\mathcal{N}} \frac{e^{F_k(y)}}{\sum_{m\in\mathcal{N}} e^{F_m(y)}} \nabla F_k(y) \right\| \\
\geq & \left\| \sum_{k\in\mathcal{N}} \left(\frac{e^{F_k(x)}}{\sum_{m\in\mathcal{N}} e^{F_m(x)}} \nabla F_k(x) \right. \right. \\
& \left. \left. - \frac{e^{F_k(y)}}{\sum_{m\in\mathcal{N}} e^{F_m(x)-(x-y)^T\nabla F_m(y)-\frac{L}{2}\|x-y\|^2}} \nabla F_k(y) \right) \right\|
\end{aligned} \tag{27}
$$

By

$$
\begin{aligned}
& \frac{e^{F_k(x)}}{e^{-\frac{L}{2}\|x-y\|_2^2} \sum_{m\in\mathcal{N}} e^{F_m(x)}} \nabla F_k(x) \geq \\
& \frac{e^{F_k(x)+\max_{m\in\mathcal{N}} -(x-y)^T\nabla F_m(y)}}{e^{-\frac{L}{2}\|x-y\|_2^2} \sum_{m\in\mathcal{N}} e^{F_m(x)-(x-y)^T\nabla F_m(y)}} \nabla F_k(x),
\end{aligned} \tag{28}
$$

denote $\varphi_k = \frac{e^{F_k(x)+\max_{m\in\mathcal{N}} -(x-y)^T\nabla F_m(y)}}{e^{-\frac{L}{2}\|x-y\|_2^2} \sum_{m\in\mathcal{N}} e^{F_m(y)-(x-y)^T\nabla F_m(y)}}$, we have

$$
\|\nabla F(x) - \nabla F(y)\| \geq \left\| \sum_{k\in\mathcal{N}} \left(\varphi_k (\nabla F_k(x) - \nabla F_k(y)) \right) \right\|. \tag{29}
$$

By the *μ-strongly convexity* of F_k, we have

$$
\begin{aligned}
\|\nabla F(x) - \nabla F(y)\| & \geq \left\| \sum_{k\in\mathcal{N}} \varphi_k \frac{\mu_k}{2}\|x-y\| \right\| \\
& \geq \left\| \sum_{k\in\mathcal{N}} \frac{\frac{\mu_k}{2}\|x-y\|}{e^{-\frac{L}{2}\|x-y\|_2^2}} \right\| \\
& \geq \sum_{k\in\mathcal{N}} \frac{\mu_k}{2}\|x-y\| \geq \frac{\mu}{2}\|x-y\|
\end{aligned} \tag{30}
$$

Thus, we finish the proof of the *μ-strongly convexity* of F. □

A.2. The proof of Theorem 2

We follow the proof in [38]. First, we present some lemmas, and then based on these lemmas, we prove the Theorem 2. For better analysis, we introduce some virtual variables $\overline{v}^t = \sum_{k=1}^N p_k^t v_k^t$ and $\overline{w}^t = \sum_{k=1}^N p_k^t w_k^t$. Both of them are inaccessible when $t+1 \notin \mathcal{I}_E$. When $t+1 \in \mathcal{I}_E$, we can get $\overline{w}^{t+1}$. We also mark the gradients in the virtual sequence as $\overline{g}_t = \sum_{k=1}^N p_k^t \nabla F_k(w_k^t)$ and $g_t = \sum_{k=1}^N p_k^t \nabla F_k(w_k^t, \xi_k^t)$. Therefore, we have $\overline{v}^{t+1} = \overline{w}^t - \eta_t g_t$ and $\mathbb{E}g_t = \overline{g}_t$.

A.2.1. Key lemmas

It is necessary to present some lemmas to get our proof.

Lemma 3 *(Result of one step SGD). Assume **Assumption 1 and 2** hold, if $\eta_c \leq \eta_t \leq \frac{1}{4L}$, $\eta_c \in (0, \frac{1}{4L})$, we have*

$$
\begin{aligned}
\mathbb{E}\|\overline{v}^{t+1} - w^*\|^2 \leq & (1-\mu\eta_t)\mathbb{E}\|\overline{w}^t - w^*\|^2 \\
& + 2\mathbb{E}\left[\sum_{k=1}^N p_k^t \|\overline{w}^t - w_k^t\|^2 \right] \\
& + \eta_t^2 \mathbb{E}\|g_t - \overline{g}_t\|^2 + \frac{2\eta_t^2}{\eta_c}\Gamma
\end{aligned} \tag{31}
$$

where $\Gamma = \sum_{k=1}^N (F_k(w^*) - F_k^*)$.

Lemma 4 *(Bounding the variance). Assume **Assumption 3** holds. It follows that*

$$
\mathbb{E}\|g_t - \overline{g}_t\|^2 \leq \sum_{k=1}^N \sigma_k^2 \tag{32}
$$

Lemma 5 *(Bounding the divergence of w_k^t). Assume **Assumption 4** holds, and η_t is non-increasing and $\eta_t \leq \eta_{t+E}$. It follows that*

$$
\mathbb{E}\left[\sum_{k=1}^N p_k^t \|\overline{w}^t - w_k^t\|^2 \right] \leq 4(E-1)^2 \eta_t^2 G^2 \tag{33}
$$

A.2.2. Completing the proof of Theorem 2

Proof. Based on the three lemmas above, we have

$$\mathbb{E}\|\overline{v}^{t+1}-w^*\|^2 \le (1-\mu\eta_t)\mathbb{E}\|\overline{w}^t-w^*\|^2 + 8(E-1)^2\eta_t^2G^2+\eta_t^2\sum_{k=1}^N\sigma_k^2+\frac{2\eta_t^2}{\eta_c}\Gamma \tag{34}$$

Let $\Delta_t=\mathbb{E}\|\overline{w}^t-w^*\|^2$ and $\Psi=8(E-1)^2G^2+\sum_{k=1}^N\sigma_k^2+\frac{2}{\eta_c}\Gamma$. We assume that $\eta_t=\frac{\beta}{t+\lambda}$ for some $\lambda>0$ and $\beta>\frac{1}{\mu}$. Then we can prove $\Delta_t\le\frac{\zeta}{t+\lambda}$ by induction if $\zeta=\max\{\frac{\beta^2\Psi}{\beta\mu-1},(\lambda+1)\Delta_1\}$. It follows that

$$\begin{aligned}\Delta_{t+1} &\le (1-\mu\eta_t)\Delta_t+\eta_t^2\Psi\\ &\le (1-\frac{\beta\mu}{t+\lambda})\frac{\zeta}{t+\lambda}+\frac{\beta^2}{(t+\lambda)^2}\Psi\\ &\le \frac{t+\lambda-1}{(t+\lambda)^2}\zeta+\left[\frac{\beta^2\Psi}{(t+\lambda)^2}-\frac{(\beta\mu-1)\zeta}{(t+\lambda)^2}\right]\\ &\le \frac{\zeta}{t+\lambda+1}\end{aligned} \tag{35}$$

We use the inequality $\frac{t+\lambda-1}{(t+\lambda)^2}\le\frac{1}{t+\lambda+1}$ for $\lambda>0$ in (35). Meanwhile, the last step requires $\zeta\ge\frac{\beta^2\Psi}{\beta\mu-1}$, so that $\frac{\beta^2\Psi}{(t+\alpha)^2}-\frac{(\beta\mu-1)\zeta}{(t+\alpha)^2}\le 0$. When $t=1$, it requires $\zeta\ge(\lambda+1)\Delta_1$. So we get the bound $\Delta_t\le\frac{\zeta}{t+\lambda}$.

Then by the L-smoothness of $F(\cdot)$, we have

$$\mathbb{E}\left[F(\overline{w}^t)\right]-F^*\le\frac{L}{2}\mathbb{E}\|\overline{w}^t-w^*\|^2\le\frac{L}{2}\Delta_t\le\frac{L}{2}\frac{\zeta}{t+\lambda} \tag{36}$$

According to $\zeta=\max\{\frac{\beta^2\Psi}{\beta\mu-1},(\lambda+1)\Delta_1\}$, we have the fact that $\zeta\le\frac{\beta^2\Psi}{\beta\mu-1}+(\lambda+1)\Delta_1$. Specifically, we set $\beta=\frac{2}{\mu}$, $\lambda=E$, we can get the bound

$$\begin{aligned}\mathbb{E}\left[F(\overline{w}^t)\right]-F^* &\le \frac{L}{2}\frac{\zeta}{t+\lambda}\\ &\le \frac{L}{2(t+\lambda)}(\frac{4\Psi}{\mu^2}+(\lambda+1)\Delta_1)\\ &\le \frac{2L\Psi}{\mu^2(t+E)}+\frac{L}{2}\Delta_1.\end{aligned} \tag{37}$$

When $t\in\mathcal{I}_E$, the virtual variable $\overline{w}^t=w^t$, so we finish the proof of Theorem 2. □

A.3. Proof of Lemma 3

Proof. Note that $\overline{v}^{t+1}=\overline{w}^t-\eta_t g_t$, then

$$\begin{aligned}\|\overline{v}^{t+1}-w^*\|^2 &= \|\overline{w}^t-\eta_t g_t-w^*\|^2\\ &= \|\overline{w}^t-\eta_t g_t-w^*-\eta_t\overline{g}_t+\eta_t\overline{g}_t\|^2\\ &= \underbrace{\|\overline{w}^t-w^*-\eta_t\overline{g}_t\|^2}_{A_1}\\ &\quad+\underbrace{2\eta_t\left\langle\overline{w}^t-w^*-\eta_t\overline{g}_t,\overline{g}_t-g_t\right\rangle}_{A_2}+\eta_t^2\|g_t-\overline{g}_t\|^2\end{aligned} \tag{38}$$

We focus on bounding A_1, we split A_1:

$$\begin{aligned}A_1 &= \|\overline{w}^t-w^*-\eta_t\overline{g}_t\|^2\\ &= \|\overline{w}^t-w^*\|^2\underbrace{-2\eta_t\left\langle\overline{w}^t-w^*,\overline{g}_t\right\rangle}_{B_1}+\underbrace{\eta_t^2\|\overline{g}_t\|^2}_{B_2}\end{aligned} \tag{39}$$

We use the L-smooth property of $F_k(\cdot)$ based on Assumption 1

$$\|\nabla F_k(w_k^t)\|^2\le 2L(F_k(w_k^t)-F_k^*) \tag{40}$$

Note that

$$\begin{aligned}B_1 &= -2\eta_t\left\langle\overline{w}^t-w^*,\overline{g}_t\right\rangle\\ &= -2\eta_t\sum_{k\in\mathcal{N}}p_k^t\left\langle\overline{w}^t-w^*,\nabla F_k(w_t^k)\right\rangle\\ &= -2\eta_t\sum_{k\in\mathcal{N}}p_k^t\left\langle\overline{w}^t-w_k^t,\nabla F_k(w_t^k)\right\rangle\\ &\quad-2\eta_t\sum_{k\in\mathcal{N}}p_k^t\left\langle w_k^t-w^*,\nabla F_k(w_t^k)\right\rangle\end{aligned} \tag{41}$$

By Cauchy-Schwarz inequality and AM-GM inequality, we have

$$-2\left\langle\overline{w}^t-w_k^t,\nabla F_k(w_k^t)\right\rangle\le\frac{1}{\eta_t}\|\overline{w}^t-w_k^t\|^2+\eta_t\|\nabla F_k(w_k^t)\|^2 \tag{42}$$

By the μ-strong convexity of $F_k(\cdot)$, we have

$$-\left\langle w_k^t-w^*,\nabla F_k(w_k^t)\right\rangle\le-(F_k(w_k^t)-F_k(w^*))-\frac{\mu}{2}\|w_k^t-w^*\|^2 \tag{43}$$

Using (41), (42) and (43), we have

$$\begin{aligned}B_1 &\le \eta_t\sum_{k\in\mathcal{N}}p_k^t(\frac{1}{\eta_t}\|\overline{w}^t-w_k^t\|^2+\eta_t\|\nabla F_k(w_k^t)\|^2)\\ &\quad-2\eta_t\sum_{k\in\mathcal{N}}p_k^t(F_k(w_k^t)-F_k(w^*)+\frac{\mu}{2}\|w_k^t-w^*\|^2)\\ &\le \eta_t\sum_{k\in\mathcal{N}}p_k^t(\frac{1}{\eta_t}\|\overline{w}^t-w_k^t\|^2+2L\eta_t(F_k(w_k^t)-F_k^*))\\ &\quad-2\eta_t\sum_{k\in\mathcal{N}}p_k^t(F_k(w_k^t)-F_k(w^*)+\frac{\mu}{2}\|w_k^t-w^*\|^2)\end{aligned} \tag{44}$$

For B_2, we use the equation (40) and convexity of $\|\cdot\|^2$, we have

$$\begin{aligned}B_2=\eta_t^2\|\overline{g}_t\|^2 &\le \eta_t^2\sum_{k\in\mathcal{N}}p_k^t\|\nabla F_k(w_k^t)\|^2\\ &\le 2L\eta_t^2\sum_{k\in\mathcal{N}}p_k^t(F_k(w_k^t)-F_k^*)\end{aligned} \tag{45}$$

Using (44) and (45), we bound the A_1

$$\begin{aligned}A_1 &= \|\overline{w}^t-w^*\|^2+B_1+B_2\\ &\le \|\overline{w}^t-w^*\|^2+\eta_t\sum_{k\in\mathcal{N}}p_k^t(\frac{1}{\eta_t}\|\overline{w}^t-w_k^t\|^2+2L\eta_t(F_k(w_k^t)-F_k^*))\\ &\quad-2\eta_t\sum_{k\in\mathcal{N}}p_k^t(F_k(w_k^t)-F_k(w^*)+\frac{\mu}{2}\|w_k^t-w^*\|^2)\\ &\quad+2L\eta_t^2\sum_{k\in\mathcal{N}}p_k^t(F_k(w_k^t)-F_k^*)\\ &\le \|\overline{w}^t-w^*\|^2+\sum_{k\in\mathcal{N}}p_k^t\|\overline{w}^t-w_k^t\|^2-\mu\eta_t\sum_{k\in\mathcal{N}}p_k^t\|w_k^t-w^*\|^2\\ &\quad+\underbrace{4L\eta_t^2\sum_{k\in\mathcal{N}}p_k^t(F_k(w_k^t)-F_k^*)-2\eta_t\sum_{k\in\mathcal{N}}p_k^t(F_k(w_k^t)-F_k(w^*))}_{C}\end{aligned} \tag{46}$$

We define $\gamma_t=2\eta_t(1-2L\eta_t)$, then we have

$$
\begin{aligned}
C &= (4L\eta_t^2 - 2\eta_t)\sum_{k\in\mathcal{N}} p_k^t F_k(w_k^t) - 4L\eta_t^2 \sum_{k\in\mathcal{N}} p_k^t F_k^* \\
&+ 2\eta_t \sum_{k\in\mathcal{N}} p_k^t F_k(w^*) \\
&= \underbrace{2\eta_t \sum_{k\in\mathcal{N}} p_k^t (F_k(w^*) - F^*) + (2\eta_t - \gamma_t)\sum_{k\in\mathcal{N}} p_k^t (F^* - F_k^*)}_{D_1} \\
&\underbrace{-\gamma_t \sum_{k\in\mathcal{N}} p_k^t (F_k(w_k^t) - F^*)}_{D_2}
\end{aligned} \tag{47}
$$

To bound D_2, first, we use the convexity of $F_k(\cdot)$ (see the first inequality). Then we use (42) (see the second inequality) and (40) (see the third inequality). The detail is (48).

$$
\begin{aligned}
D_2 &= -\gamma_t \sum_{k\in\mathcal{N}} p_k^t (F_k(w_k^t) - F^*) \\
&= -\gamma_t \sum_{k\in\mathcal{N}} p_k^t \left(F_k(w_k^t) - F_k(\overline{w}^t) + F_k(\overline{w}^t) - F^*\right) \\
&\le -\gamma_t \sum_{k\in\mathcal{N}} p_k^t \left(\left\langle \nabla F_k(\overline{w}^t), w_k^t - \overline{w}^t \right\rangle + F_k(\overline{w}^t) - F^*\right) \\
&\le \frac{\gamma_t}{2} \sum_{k\in\mathcal{N}} p_k^t \left(\eta_t \|\nabla F_k(\overline{w}^t)\|^2 + \frac{1}{\eta_t}\|w_k^t - \overline{w}^t\|^2\right) \\
&- \gamma_t \sum_{k\in\mathcal{N}} p_k^t \left(F_k(\overline{w}^t) - F^*\right) \\
&\le \frac{\gamma_t}{2} \sum_{k\in\mathcal{N}} p_k^t \left(2L\eta_t \left(F_k(\overline{w}^t) - F_k^*\right) + \frac{1}{\eta_t}\|w_k^t - \overline{w}^t\|^2\right) \\
&- \gamma_t \sum_{k\in\mathcal{N}} p_k^t \left(F_k(\overline{w}^t) - F^*\right)
\end{aligned} \tag{48}
$$

We can bound C, we use the following facts since $\eta_c \le \eta_t \le \frac{1}{4L}$: $\eta_t L - 1 \le 0$ and $\frac{\gamma_t}{2\eta_t} \le 1$, for all client k, $F_k(\overline{w}^t) - F_k^* \ge 0$.

$$
\begin{aligned}
C &= D_1 + D_2 \\
&= L\eta_t\gamma_t \sum_{k\in\mathcal{N}} p_k^t \left(F_k(\overline{w}^t) - F_k^*\right) + \frac{\gamma_t}{2\eta_t}\sum_{k\in\mathcal{N}} p_k^t \|w_k^t - \overline{w}^t\|^2 \\
&- \gamma_t \sum_{k\in\mathcal{N}} p_k^t \left((F_k(\overline{w}^t) - F_k^*) - (F^* - F_k^*)\right) + D_1 \\
&= \gamma_t(L\eta_t - 1)\sum_{k\in\mathcal{N}} p_k^t \left(F_k(\overline{w}^t) - F_k^*\right) \\
&+ \frac{\gamma_t}{2\eta_t}\sum_{k\in\mathcal{N}} p_k^t \|w_k^t - \overline{w}^t\|^2 + 2\eta_t \sum_{k\in\mathcal{N}} p_k^t (F_k(w^*) - F_k^*) \\
&\le \frac{\gamma_t}{2\eta_t}\sum_{k\in\mathcal{N}} p_k^t \|w_k^t - \overline{w}^t\|^2 + 2\eta_t \sum_{k\in\mathcal{N}} p_k^t (F_k(w^*) - F_k^*) \\
&\le \sum_{k\in\mathcal{N}} p_k^t \|w_k^t - \overline{w}^t\|^2 + \frac{2\eta_t^2}{\eta_c}\sum_{k\in\mathcal{N}} p_k^t (F_k(w^*) - F_k^*)
\end{aligned} \tag{49}
$$

Using the bound of C, we can get

$$
\begin{aligned}
A_1 &\le \|\overline{w}^t - w^*\|^2 + \sum_{k\in\mathcal{N}} p_k^t \|\overline{w}^t - w_k^t\|^2 \\
&- \mu\eta_t \sum_{k\in\mathcal{N}} p_k^t \|w_k^t - w^*\|^2 + C \\
&\le (1 - \mu\eta_t)\|\overline{w}^t - w^*\|^2 + 2\sum_{k\in\mathcal{N}} p_k^t \|\overline{w}^t - w_k^t\|^2 \\
&+ \frac{2\eta_t^2}{\eta_c}\sum_{k\in\mathcal{N}} p_k^t (F_k(w^*) - F_k^*)
\end{aligned} \tag{50}
$$

We use Jensen's inequality here, we show the fact that the function $f(x) = \|x - w^*\|^2$ is convex, so we have

$$
\|\sum_{k\in\mathcal{N}} p_k^t w_k^t - w^*\|^2 \le \sum_{k\in\mathcal{N}} p_k^t \|w_k^t - w^*\|^2 \tag{51}
$$

Taking the expectation on equation (38), we have

$$
\begin{aligned}
\mathbb{E}\|\overline{v}^{t+1} - w^*\|^2 &= \mathbb{E}A_1 + \mathbb{E}A_2 + \eta_t^2 \mathbb{E}\|g_t - \overline{g}_t\|^2 \\
&\le (1 - \mu\eta_t)\|\overline{w}^t - w^*\|^2 \\
&+ 2\mathbb{E}\left[\sum_{k\in\mathcal{N}} p_k^t \|\overline{w}^t - w_k^t\|^2\right] \\
&+ \frac{2\eta_t^2}{\eta_c}\mathbb{E}\left[\sum_{k\in\mathcal{N}} p_k^t (F_k(w^*) - F_k^*)\right] \\
&+ \eta_t^2 \mathbb{E}\|g_t - \overline{g}_t\|^2
\end{aligned} \tag{52}
$$

Since $0 \le p_k^t \le 1$ and $F_k(w^*) - F_k^*$ is determined once the F_k is determined. We show the fact that $\mathbb{E}\left[\sum_{k\in\mathcal{N}} p_k^t (F_k(w^*) - F_k^*)\right] \le \sum_{k\in\mathcal{N}} (F_k(w^*) - F_k^*)$. We define $\Gamma = \sum_{k\in\mathcal{N}} (F_k(w^*) - F_k^*)$. The expectation on equation (38) can be written as

$$
\begin{aligned}
\mathbb{E}\|\overline{v}^{t+1} - w^*\|^2 &\le (1 - \mu\eta_t)\mathbb{E}\|\overline{w}^t - w^*\|^2 + 2\mathbb{E}\left[\sum_{k\in\mathcal{N}} p_k^t \|\overline{w}^t - w_k^t\|^2\right] \\
&+ \eta_t^2 \mathbb{E}\|g_t - \overline{g}_t\|^2 + \frac{2\eta_t^2}{\eta_c}\Gamma
\end{aligned} \tag{53}
$$

We complete the proof. □

A.4. Proof of Lemma 4

Proof. We can bound the variance of stochastic gradients in client k by σ_k, we have

$$
\begin{aligned}
\mathbb{E}\|g_t - \overline{g}_t\|^2 &= \mathbb{E}\left\|\sum_{k\in\mathcal{N}} p_k^t (\nabla F_k(w_k^t, \xi_k^t) - \nabla F_k(w_k^t))\right\|^2 \\
&\le \mathbb{E}\left(\sum_{k\in\mathcal{N}} p_k^t \|\nabla F_k(w_k^t, \xi_k^t) - \nabla F_k(w_k^t)\|^2\right) \\
&\le \mathbb{E}\left(\sum_{k\in\mathcal{N}} p_k^t \sigma_k^2\right) \le \sum_{k\in\mathcal{N}} \sigma_k^2
\end{aligned} \tag{54}
$$

Here, we use the $p_k^t \le 1$. We note that the main difference of this proof with other proofs [38] is that p_k^t is dynamically changing during the training process, and finally converges to a certain value. We give the analysis of p_k^t here.

Note that the $p_k^t = \frac{e^{F_k(w)}}{\sum_{m\in\mathcal{N}} e^{F_m(w)}}$, we have

$$
p_k^t = \frac{e^{F_k(w)}}{e^{F_k(w)} + \sum_{m\in\{\mathcal{N}\setminus k\}} e^{F_m(w)}} \tag{55}
$$

Since $F_k(w) \ge F_k^*$, where F_k^* is the optimal value of the local loss function F_k, we can get $e^{F_k(w)} \ge e^{F_k^*}$, and

$$
p_k^t = \frac{1}{1 + \frac{\sum_{m\in\{\mathcal{N}\setminus k\}} e^{F_m(w)}}{e^{F_k(w)}}} \ge \frac{1}{1 + \frac{\sum_{m\in\{\mathcal{N}\setminus k\}} e^{F_m(w)}}{e^{F_k^*}}} \tag{56}
$$

Note that the lower bound of p_k^t only depends on the sum of other clients ($m \in \{\mathcal{N}\setminus k\}$) exponential inference loss. For the upper bound of p_k^t, by the $\frac{x}{x+a} < 1$, we can get $p_k^t < 1$.

In summary, we bound the

$$
\frac{1}{1 + \frac{\sum_{m\in\{\mathcal{N}\setminus k\}} e^{F_m(w)}}{e^{F_k(w)}}} \le p_k^t < 1. \tag{57}
$$

We complete the proof. □

A.5. Proof of Lemma 5

Proof. In FedVSA, each client might have E steps in local training before the global aggregation. We assume that the global aggregation performs at t^{th} CR, so there exists a $t_0 \leq t$, such that $t - t_0 \leq E - 1$, and $\overline{w}^{t_0} = w_k^{t_0}$ for all clients. Also, we use the assumption that η_t is non-increasing and $\eta_{t_0} \leq 2\eta_t$, and we use the bound the p_k^t in Equ (57), then

$$\begin{aligned}
&\mathbb{E}\left[\sum_{k\in\mathcal{N}} p_k^t \|\overline{w}^t - w_k^t\|^2\right] \leq \mathbb{E}\left[\sum_{k\in\mathcal{N}} p_k^t \|w_k^t - \overline{w}^{t_0}\|^2\right] \\
&\leq \mathbb{E}\sum_{k\in\mathcal{N}} p_k^t \sum_{t'=t_0}^{t-1} (E-1)\eta_{t'}^2 \|\nabla F_k(w_k^{t'}, \xi_k^{t'})\|^2 \\
&\leq \sum_{t'=t_0}^{t-1} (E-1)\eta_{t'}^2 \mathbb{E}\|\nabla F_k(w_k^{t'}, \xi_k^{t'})\|^2 \\
&\leq (E-1)\sum_{t'=t_0}^{t-1} \eta_{t'}^2 G^2 \leq (E-1)^2 \eta_{t_0}^2 G^2 \\
&\leq 4(E-1)^2 \eta_t^2 G^2
\end{aligned} \tag{58}$$

We complete the proof. □

Data availability

Data will be made available on request.

References

[1] D.A. Acar, Y. Zhao, R.M. Navarro, M. Mattina, P. Whatmough, V. Saligrama, Federated learning based on dynamic regularization, in: Proc. of International Conference on Learning Representations, ICLR, 2021, pp. 1–12.

[2] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov, How to backdoor federated learning, in: Proc. of the International Conference on Artificial Intelligence and Statistics, AISTATS, 2020, pp. 2938–2948.

[3] S. Banabilah, M. Aloqaily, E. Alsayed, N. Malik, Y. Jararweh, Federated learning review: fundamentals, enabling technologies, and future applications, Inf. Process. Manag. 59 (2022) 103061.

[4] P. Blanchard, E.M. El Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries: Byzantine tolerant gradient descent, in: Proc. of the Annual Conference on Neural Information Processing Systems, NeurIPS, 2017, pp. 118–128.

[5] S. Boyd, S.P. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[6] S. Caldas, S.M.K. Duddu, P. Wu, T. Li, J. Konečnỳ, H.B. McMahan, V. Smith, A. Talwalkar, Leaf: a benchmark for federated settings, arXiv preprint, arXiv:1812.01097, 2018.

[7] M. Chahoud, S. Otoum, A. Mourad, On the feasibility of federated learning towards on-demand client deployment at the edge, Inf. Process. Manag. 60 (2023) 103150.

[8] Y. Chen, Z. Chai, Y. Cheng, H. Rangwala, Asynchronous federated learning for sensor data with concept drift, in: Proc. of the International Conference on Big Data, Big Data, 2021, pp. 4822–4831.

[9] Y. Chen, L. Su, J. Xu, Distributed statistical machine learning in adversarial settings: Byzantine gradient descent, in: Proc. of Measurement and Analysis of Computing Systems, POMACS, 2017, pp. 1–25.

[10] J. Cheng, Z. Liu, Y. Shi, P. Luo, V.S. Sheng, Grcol-ppfl: user-based group collaborative federated learning privacy protection framework, Comput. Mater. Continua 75 (2023).

[11] G. Damaskinos, R. Guerraoui, R. Patra, M. Taziki, et al., Asynchronous byzantine machine learning (the case of SGD), in: Proc. of the International Conference on Machine Learning, ICML, 2018, pp. 1145–1154.

[12] Z. Daniel (Yue), K. Ziyi, W. Dong, Fedsens: a federated learning approach for smart health sensing with class imbalance in resource constrained edge computing, in: Proc. of the IEEE Conference on Computer Communications, INFOCOM, 2021, pp. 1–10.

[13] Y. Deng, F. Lyu, J. Ren, Y.C. Chen, P. Yang, Y. Zhou, Y. Zhang, Improving federated learning with quality-aware user incentive and auto-weighted model aggregation, IEEE Trans. Parallel Distrib. Syst. 33 (2022) 4515–4529.

[14] C.T. Dinh, N.H. Tran, et al., Federated learning with proximal stochastic variance reduced gradient algorithms, in: Proc. of International Conference on Parallel Processing, ICPP, 2020, pp. 1–11.

[15] M. Duan, D. Liu, X. Chen, Y. Tan, J. Ren, L. Qiao, L. Liang, Astraea: self-balancing federated learning for improving classification accuracy of mobile deep learning applications, in: Proc. of International Conference on Computer Design, ICCD, 2019, pp. 246–254.

[16] Y.H. Ezzeldin, S. Yan, C. He, E. Ferrara, A.S. Avestimehr, Fairfed: enabling group fairness in federated learning, in: Proc. of the AAAI Conference on Artificial Intelligence, AAAI, 2023, pp. 7494–7502.

[17] J. Fan, G. Tang, K. Wu, Z. Zhao, Y. Zhou, S. Huang, Score-vae: root cause analysis for federated-learning-based iot anomaly detection, IEEE Int. Things J. 11 (2023) 1041–1053.

[18] L. Gao, H. Fu, L. Li, Y. Chen, M. Xu, C. Xu, Feddc: federated learning with non-iid data via local drift decoupling and correction, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 10102–10111.

[19] L. Gao, L. Li, Y. Chen, C. Xu, M. Xu, Fgfl: a blockchain-based fair incentive governor for federated learning, J. Parallel Distrib. Comput. 163 (2022) 283–299.

[20] S. Gao, J. Luo, J. Zhu, X. Dong, W. Shi, Vcd-fl: verifiable, collusion-resistant, and dynamic federated learning, IEEE Trans. Inf. Forensics Secur. 18 (2023) 3760–3773.

[21] R.C. Geyer, T. Klein, et al., Differentially private federated learning: a client level perspective, arXiv preprint, arXiv:1712.07557, 2017.

[22] A. Ghosh, J. Chung, D. Yin, K. Ramchandran, An efficient framework for clustered federated learning, IEEE Trans. Inf. Theory 68 (2022) 8076–8091.

[23] J. Guo, J. Wu, A. Liu, N.N. Xiong, Lightfed: an efficient and secure federated edge learning system on model splitting, IEEE Trans. Parallel Distrib. Syst. 33 (2021) 2701–2713.

[24] Y. Guo, F. Liu, Z. Cai, L. Chen, N. Xiao, FEEL: a federated edge learning system for efficient and privacy-preserving mobile healthcare, in: Proc. of International Conference on Parallel Processing, ICPP, 2020, pp. 1–11.

[25] Y. Guo, F. Liu, T. Zhou, Z. Cai, N. Xiao, Efficiency: achieving both through adaptive hierarchical federated learning, IEEE Trans. Parallel Distrib. Syst. 34 (2023) 1331–1342.

[26] Y. Guo, F. Liu, T. Zhou, Z. Cai, N. Xiao, Seeing is believing: towards interactive visual exploration of data privacy in federated learning, Inf. Process. Manag. 60 (2023) 103162.

[27] Z. Han, C. Ge, B. Wu, Z. Liu, Lightweight privacy-preserving federated incremental decision trees, IEEE Trans. Serv. Comput. 16 (2023) 1964–1975.

[28] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778.

[29] G. Huang, X. Chen, T. Ouyang, Q. Ma, L. Chen, J. Zhang, Collaboration in participant-centric federated learning: a game-theoretical perspective, IEEE Trans. Mob. Comput. 22 (2022) 6311–6326.

[30] L. Huang, D. Liu, Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records, J. Biomed. Inform. 99 (2019) 103291.

[31] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, D. Liu, LoAdaBoost: loss-based adaboost federated machine learning with reduced computational complexity on iid and non-iid intensive care data, PLoS ONE 15 (2020) e0230706.

[32] W. Huang, M. Ye, B. Du, Learn from others and be yourself in heterogeneous federated learning, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 10133–10143.

[33] E. Kairouz, H. McMahan, Advances and open problems in federated learning, Found. Trends Mach. Learn. 14 (2021) 1–210.

[34] S.P. Karimireddy, S. Kale, M. Mohri, S.J. Reddi, S. Stich, A. Suresh, SCAFFOLD: stochastic controlled averaging for federated learning, in: Proc. of the International Conference on Machine Learning, ICML, 2019, pp. 5132–5143.

[35] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proc. of the Annual Conference on Neural Information Processing Systems, NeurIPS, 2012, pp. 1097–1105.

[36] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proc. of the IEEE, 1998, pp. 2278–2324.

[37] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, in: Proc. of Machine Learning and Systems, MLSys, 2020, pp. 429–450.

[38] X. Li, K. Huang, W. Yang, S. Wang, Z. Zhang, On the convergence of fedavg on non-iid data, in: Proc. of the International Conference on Learning Representations, ICLR, 2019, pp. 1–26.

[39] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, X. Ma, Anti-backdoor learning: training clean models on poisoned data, in: Proc. of the Annual Conference on Neural Information Processing Systems, NeurIPS, 2021, pp. 14900–14912.

[40] Z. Li, T. Lin, X. Shang, C. Wu, Revisiting weighted aggregation in federated learning with neural networks, in: Proc. of the International Conference on Machine Learning, ICML, PMLR, 2023, pp. 19767–19788.

[41] P. Liu, T. Zhou, Z. Cai, F. Liu, Y. Guo, Leveraging heuristic client selection for enhanced secure federated submodel learning, Inf. Process. Manag. 60 (2023) 103211.

[42] Q. Liu, T. Zhou, Z. Cai, Y. Yuan, M. Xu, J. Qin, W. Ma, Turning backdoors for efficient privacy protection against image retrieval violations, Inf. Process. Manag. 60 (2023) 103471.

[43] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, M.S. Hossain, Deep anomaly detection for time-series data in industrial IoT: a communication-efficient on-device federated learning approach, IEEE Int. Things J. 8 (2020) 6348–6358.

[44] Z. Liu, C. Wang, X. Yang, N. Zhang, F. Liu, B. Zhang, Time series multi-step forecasting based on memory network for the prognostics and health management in freight train braking system, IEEE Trans. Intell. Transp. Syst. 24 (2023) 8149–8162.

[45] G. Lu, Y. Liu, J. Wang, H. Wu, Cnn-bilstm-attention: a multi-label neural classifier for short texts with a small set of labels, Inf. Process. Manag. 60 (2023) 103320.

[46] Z. Ma, J. Ma, Y. Miao, X. Liu, K. Choo, R.H. Deng, Pocket diagnosis: secure federated learning against poisoning attack in the cloud, IEEE Trans. Serv. Comput. 15 (2022) 3429–3442.
[47] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Proc. of the International Conference on Artificial Intelligence and Statistics, AISTATS, 2017, pp. 1273–1282.
[48] H.B. McMahan, D. Ramage, K. Talwar, L. Zhang, Learning differentially private recurrent language models, in: Proc. of the International Conference on Learning Representations, ICLR, 2018, pp. 1–10.
[49] O. Ohrimenko, F. Schuster, C. Fournet, A. Mehta, et al., Oblivious multi-party machine learning on trusted processors, in: Proc. of USENIX Security Symposium, 2016, pp. 619–636.
[50] S. Pandya, G. Srivastava, R. Jhaveri, M.R. Babu, S. Bhattacharya, P.K.R. Maddikunta, S. Mastorakis, M.J. Piran, T.R. Gadekallu, Federated learning for smart cities: a comprehensive survey, Sustain. Energy Technol. Assess. 55 (2023) 102987.
[51] K. Pillutla, S.M. Kakade, Z. Harchaoui, Robust aggregation for federated learning, IEEE Trans. Signal Process. 70 (2022) 1142–1154.
[52] L. Qu, N. Balachandar, M. Zhang, D. Rubin, Handling data heterogeneity with generative replay in collaborative learning for medical imaging, Med. Image Anal. 78 (2022) 102424.
[53] L. Qu, Y. Zhou, P. Liang, Y. Xia, F. Wang, L. Fei-Fei, E. Adeli, D. Rubin, Rethinking architecture design for tackling data heterogeneity in federated learning, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 10051–10061.
[54] D. Sarkar, A. Narang, S. Rai, Fed-focal loss for imbalanced data classification in federated learning, arXiv preprint, arXiv:2011.06283, 2020.
[55] F. Sattler, S. Wiedemann, K.R. Müller, W. Samek, Robust and communication-efficient federated learning from non-iid data, IEEE Trans. Neural Netw. Learn. Syst. 31 (2019) 3400–3413.
[56] V. Shejwalkar, A. Houmansadr, Manipulating the byzantine: optimizing model poisoning attacks and defenses for federated learning, in: Proc. of the Network and Distributed Systems Security Symposium, NDSS, 2021, pp. 1–19.
[57] S. Shi, C. Hu, D. Wang, Y. Zhu, Z. Han, Federated anomaly analytics for local model poisoning attack, IEEE J. Sel. Areas Commun. 40 (2021) 596–610.
[58] Y. Shi, J. Liang, W. Zhang, V.Y. Tan, S. Bai, Towards understanding and mitigating dimensional collapse in heterogeneous federated learning, in: Proc. of the International Conference on Learning Representations, ICLR, 2023, pp. 1–24.
[59] N. Shoham, T. Avidor, A. Keren, et al., Overcoming forgetting in federated learning on non-iid data, arXiv preprint, arXiv:1910.07796, 2019.
[60] R. Shokri, et al., Bypassing backdoor detection algorithms in deep learning, in: Proc. of the IEEE European Symposium on Security and Privacy, EuroS&P, 2020, pp. 175–183.
[61] T. Song, Y. Tong, S. Wei, Profit allocation for federated learning, in: Proc. of the International Conference on Big Data, Big Data, IEEE, 2019, pp. 2577–2586.
[62] Z. Sun, P. Kairouz, A.T. Suresh, H.B. McMahan, Can you really backdoor federated learning?, arXiv preprint, arXiv:1911.07963, 2019.
[63] A. Taïk, H. Moudoud, S. Cherkaoui, Data-quality based scheduling for federated edge learning, in: Proc. of the IEEE Conference on Local Computer Networks, LCN, 2021, pp. 17–23.
[64] M. Uddin, Y. Xiang, X. Lu, J. Yearwood, L. Gao, Federated learning via disentangled information bottleneck, IEEE Trans. Serv. Comput. 16 (2023) 1874–1889.
[65] H. Wang, Z. Kaplan, D. Niu, B. Li, Optimizing federated learning on non-iid data with reinforcement learning, in: Proc. of the IEEE Conference on Computer Communications, INFOCOM, 2020, pp. 1698–1707.
[66] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J. Sohn, K. Lee, D.S. Papailiopoulos, Attack of the tails: yes, you really can backdoor federated learning, in: Proc. of the Annual Conference on Neural Information Processing Systems, NeurIPS, 2020, pp. 16070–16084.
[67] J. Wang, Q. Liu, H. Liang, G. Joshi, H.V. Poor, Tackling the objective inconsistency problem in heterogeneous federated optimization, in: Proc. of the Annual Conference on Neural Information Processing Systems, NeurIPS, 2020, pp. 7611–7623.
[68] L. Wang, S. Xu, X. Wang, Q. Zhu, Addressing class imbalance in federated learning, in: Proc. of the AAAI Conference on Artificial Intelligence, AAAI, 2021, pp. 10165–10173.
[69] W. Wang, X. Li, X. Qiu, X. Zhang, J. Zhao, V. Brusic, A privacy preserving framework for federated learning in smart healthcare systems, Inf. Process. Manag. 60 (2023) 103167.
[70] Z. Wang, Y. Zhu, D. Wang, Z. Han, Fedacs: federated skewness analytics in heterogeneous decentralized data environments, in: Proc. of the International Symposium on Quality of Service, IWQOS, 2021, pp. 1–10.
[71] D. Wu, Y. Deng, M. Li, Fl-mgvn: federated learning for anomaly detection using mixed Gaussian variational self-encoding network, Inf. Process. Manag. 59 (2022) 102839.
[72] X. Wu, F. Huang, Z. Hu, H. Huang, Faster adaptive federated learning, in: Proc. of the AAAI Conference on Artificial Intelligence, AAAI, 2023, pp. 10379–10387.
[73] P. Xiao, S. Cheng, V. Stankovic, D. Vukobratovic, Averaging is probably not the optimum way of aggregating parameters in federated learning, Entropy 22 (2020) 314.
[74] C. Xie, S. Koyejo, I. Gupta, Zeno: distributed stochastic gradient descent with suspicion-based fault-tolerance, in: Proc. of the International Conference on Machine Learning, ICML, 2019, pp. 6893–6901.
[75] C. Xie, S. Koyejo, I. Gupta, Zeno++: robust fully asynchronous sgd, in: Proc. of the International Conference on Machine Learning, ICML, 2020, pp. 10495–10503.
[76] C. Xu, Z. Hong, M. Huang, T. Jiang, Acceleration of federated learning with alleviated forgetting in local training, in: Proc. of the International Conference on Learning Representations, ICLR, 2022, pp. 1–10.
[77] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: concept and applications, ACM Trans. Intell. Syst. Technol. 10 (2019) 1–19.
[78] R. Ye, M. Xu, J. Wang, C. Xu, S. Chen, Y. Wang, Feddisco: federated learning with discrepancy-aware collaboration, in: Proc. of the International Conference on Machine Learning, ICML, PMLR, 2023, pp. 39879–39902.
[79] D. Yin, Y. Chen, R. Kannan, P. Bartlett, Byzantine-robust distributed learning: towards optimal statistical rates, in: Proc. of the International Conference on Machine Learning, ICML, 2018, pp. 5650–5659.
[80] D. Yongheng, L. Feng, et al., Fair: quality-aware federated learning with precise user incentive and model aggregation, in: Proc. of the IEEE Conference on Computer Communications, INFOCOM, 2021, pp. 1–10.
[81] H. Zeng, T. Zhou, Y. Guo, Z. Cai, F. Liu, FedCav: contribution-aware model aggregation on distributed heterogeneous data in federated learning, in: Proc. of International Conference on Parallel Processing, ICPP, 2021, pp. 1–10.
[82] H. Zeng, T. Zhou, X. Wu, Z. Cai, Never too late: tracing and mitigating backdoor attacks in federated learning, in: Proc. of the IEEE International Symposium on Reliable Distributed Systems, SRDS, 2022, pp. 69–81.
[83] L. Zhang, Y. Luo, Y. Bai, B. Du, L.Y. Duan, Federated learning for non-iid data via unified feature learning and optimization objective alignment, in: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 4420–4428.
[84] X. Zhang, F. Li, Z. Zhang, Q. Li, C. Wang, J. Wu, Enabling execution assurance of federated learning at untrusted participants, in: Proc. of the IEEE Conference on Computer Communications, INFOCOM, IEEE, 2020, pp. 1877–1886.
[85] Y. Zhang, D. Liu, M. Duan, L. Li, X. Chen, A. Ren, Y. Tan, C. Wang, Fedmds: an efficient model discrepancy-aware semi-asynchronous clustered federated learning framework, IEEE Trans. Parallel Distrib. Syst. 34 (2023) 1007–1019.
[86] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, V. Chandra, Federated learning with non-iid data, arXiv preprint, arXiv:1806.00582, 2018.
[87] T. Zhou, Z. Cai, F. Liu, J. Su, In pursuit of beauty: aesthetic-aware and context-adaptive photo selection in crowdsensing, IEEE Trans. Knowl. Data Eng. 35 (2023) 9364–9377.
[88] B. Zhu, L. Wang, Q. Pang, S. Wang, J. Jiao, D. Song, M.I. Jordan, Byzantine-robust federated learning with optimal statistical rates, in: Proc. of the International Conference on Artificial Intelligence and Statistics, AISTATS, 2023, pp. 3151–3178.

Hui Zeng received an undergraduate degree in Software Engineering from Sun Yat-sen University (SYSU), Guangzhou in 2020.

He is currently working toward the Ph.D degree in College of Computer, NUDT. His main research interests include distributed systems, federated learning, and data privacy.

Tongqing Zhou received bachelor's, master's, and Ph.D degrees in Computer Science and Technology from the National University of Defense Technology (NUDT), Changsha in 2012, 2014, and 2018, respectively.

He is currently a postdoc at the College of Computer, NUDT. His main research interests include ubiquitous computing, mobile sensing, and data privacy.

Yeting Guo received the B.S. and M.S. degrees in computer science from the National University of Defense Technology, China, in 2017 and 2019, respectively.

She is currently working toward a Ph.D. degree with the College of Computer from the National University of Defense Technology, Changsha, China. Her main research interests include computer architecture and edge computing.

Zhiping Cai received B.S., M.S., and Ph.D. degrees in computer science and technology from the National University of Defense Technology (NUDT), Changsha, China, in 1996, 2002, and 2005, respectively.

He is currently a Full Professor with the College of Computer, NUDT. His current research interests include network security and big data. Dr. Cai is a member of IEEE and a senior member of the China Computer Federation. His Ph.D. dissertation received the Outstanding Dissertation Award of the Chinese PLA.

Fang Liu received B.S. and Ph.D. degrees in computer science from the College of Computer, National University of Defense Technology (NUDT), Changsha, China, in 1999 and 2005, respectively.

She is currently a Professor at the School of Design, Hunan University, Changsha, China. Her current research interests include computer architecture, edge computing, and storage systems.