PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

Sparsification Under Siege: Dual-Level Defense Against Poisoning in Communication-Efficient Federated Learning

Zhiyong Jin[1], **Runhua Xu**[1*], **Chao Li**[3], **Yizhong Liu**[2], **Jianxin Li**[1*], **James Joshi**[4]
[1]School of Computer Science and Engineering, Beihang University
[2]School of Cyber Science and Technology, Beihang University
[3]Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University
[4]University of Pittsburgh
{sy2406106, runhua, liuyizhong, lijx}@buaa.edu.cn, li.chao@bjtu.edu.cn, jjoshi@pitt.edu

IJCAI-ECAI 2026 PREPRINT

Abstract

Gradient sparsification, while mitigating communication bottlenecks in Federated Learning (FL), fundamentally alters the geometric landscape of model updates. We reveal that the resultant high-dimensional orthogonality renders traditional Euclidean-based robust aggregation metrics mathematically ambiguous, creating a "sparsity-robustness trade-off" that adversaries exploit to bypass detection. To resolve this structural dissonance, we propose SafeSparse, a consensus restoration framework that decouples defense into topological and semantic dimensions. Unlike prior arts that treat sparsification and security orthogonally, *SafeSparse* introduces: (1) a Structure-Aware Calibration mechanism utilizing Jaccard similarity to filter topological outliers induced by index poisoning; and (2) a Directional Semantic Alignment module employing density-based clustering on update signs to neutralize magnitude-invariant attacks. Theoretically, we establish convergence guarantees for *SafeSparse*. Extensive experiments across multiple datasets and attack scenarios demonstrate that *SafeSparse* recovers up to 25.7% global accuracy under coordinated poisoning, effectively closing the vulnerability gap in communication-efficient FL.

1 Introduction

Federated Learning (FL) has emerged as a paradigm shift in distributed artificial intelligence, enabling collaborative training across decentralized edge devices while preserving data locality [Kairouz *et al.*, 2021]. Despite its privacy benefits, the communication bottleneck inherent in transmitting high-dimensional parameter updates remains a critical hurdle. To address this, gradient sparsification techniques, such as Top-k selection, have become the de facto standard, reducing uplink communication overheads by orders of magnitude (often exceeding 99%) without significantly compromising convergence [Sattler *et al.*, 2019b; Sattler *et al.*, 2019a; Han *et al.*, 2020].

However, we argue that this pursuit of efficiency has inadvertently introduced a fundamental structural vulnerability. Existing robust aggregation protocols, such as Krum [Blanchard *et al.*, 2017], Geometric Median (RFA) [Pillutla *et al.*, 2022], and FLGuardian [Zhou *et al.*, 2025], are predicated on the geometric assumption of *Dense Euclidean Consensus*. They assume that benign updates cluster around a global mean in a dense vector space, allowing outliers to be detected via Euclidean metrics (L_2 norm).

We reveal that Top-k sparsification violates this premise. Mathematically, the sparsification operator acts as a nonlinear projection that maps updates into low-dimensional subspaces. In non-IID settings, this results in benign updates becoming *sparse orthogonal*, namely, their non-zero indices (masks) barely overlap. Consequently, Euclidean distance becomes mathematically ambiguous as a similarity metric; two benign clients holding disjoint but valid features may appear infinitely distant from each other. We term this phenomenon the **Sparsity-Robustness Trade-off**: the very mechanism used to save bandwidth destroys the geometric landscape required for security. By manipulating the sparse index masks (Structure) rather than just the parameter values (Semantics), malicious clients can artificially concentrate their contributions into specific parameter packs. This allows them to achieve "local dominance" in targeted subspaces even while remaining a minority globally, effectively bypassing norm-based defenses that look for global outliers.

To resolve this geometric crisis, we propose **SafeSparse**, a consensus restoration framework that recouples sparsification and robustness through a dual-space calibration mechanism. Specifically, SafeSparse first enforces *topological consensus* by utilizing Jaccard similarity to filter structural outliers induced by index poisoning. It then establishes *semantic alignment* by mapping updates to a directional unit-sphere via sign-based vectors and employing density-based clustering. This approach effectively isolates the benign consensus cluster without relying on the flawed Euclidean magnitude, thereby neutralizing both structural and magnitude-invariant attacks.

Our contributions are summarized as follows: (1) We sys-

*Corresponding authors.

tematically identify and formalize the geometric dissonance problem in sparse FL, proving that standard robust aggregators possess a theoretical failure mode when inputs are sparse and orthogonal. (2) We present *SafeSparse*, the first defense framework explicitly tailored to reconcile the tension between communication efficiency and robust aggregation through mask-aware and sign-aware inspection. (3) We provide theoretical convergence guarantees for our method and demonstrate through extensive experiments that *SafeSparse* significantly outperforms existing baselines, recovering global model accuracy in highly corrupted environments where traditional defenses fail.

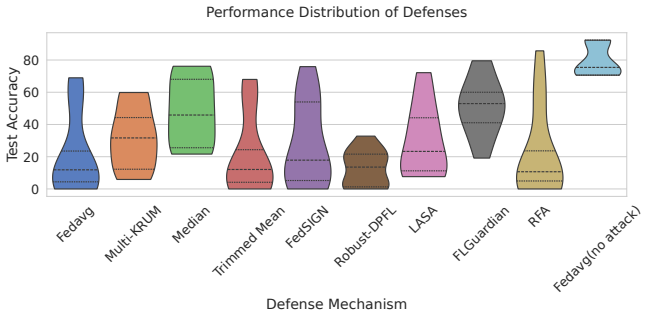

Figure 1: **Empirical validation of poisoning vulnerability.** The violin plots illustrate the accuracy distribution across **12 diverse scenarios** (3 datasets $\times$ 4 attacks) as detailed in Section 4.3.

2 Vulnerabilities in Sparsified FL

Traditional robust FL often assumes that malicious updates are "outliers" in the full parameter space. However, we identify a fundamental shift in the attack surface when sparsification-based communication-efficiency method is employed.

Specifically, in the context of sparsified FL, we identify two distinct attack vectors. (1) *Index Poisoning*: Adversaries manipulate the sparse index masks to maximize the *Malicious Contributor Ratio* (f_p) for specific target parameters, effectively hijacking the aggregation weight of critical neurons. (2) *Masked Value Manipulation*: Adversaries inject malicious perturbations (e.g., sign flipping) into the selected parameters while mimicking the sparsity patterns of benign clients to evade detection. Unlike standard FL, where attackers are diluted globally, sparsification allows attackers to achieve local dominance in specific parameter packs.

Theoretical Analysis. Unlike dense FL, sparsified aggregation allows adversaries to concentrate their influence. We define attack effectiveness $\rho = \|W_G - W'_G\|_2^2$ as the squared Euclidean distance between the poisoned global model W_G and an ideal benign model W'_G.

Theorem 1 (Vulnerability under Sparsified Aggregation)**.** *Assume the malicious perturbation per parameter pack is bounded by* $\mathbb{E}[\|W_a(p) - W'_G(p)\|_2] \leq \epsilon$. *The attack effectiveness* ρ *is upper-bounded by:*

$$\rho \leq \epsilon^2 \sum_{p=1}^{P} f_p^2, \tag{1}$$

where $f_p = N_p^A / N_p$ *is the malicious contributor ratio for parameter pack* p.

Theorem 1 reveals that the attack impact is determined by the pack-level ratio f_p rather than the global attacker ratio in sparsified FL. By coordinating sparse masks, adversaries can drive $f_p \to 1$ for specific packs, exerting immense influence. *SafeSparse* mitigates this by using Jaccard and Sign-based filtering to minimize f_p across all packs. Detailed proof is provided in Appendix C.

In addition to theoretical analysis, we experimentally verified the vulnerability. As illustrated in Figure 1, we evaluate the global model accuracy across 12 diverse scenarios (comprising three datasets and four coordinated attacks). The probability density (represented by the width of the violins) and the quartiles (indicated by the dashed lines within) reveal a systemic performance collapse for traditional defenses. Specifically, mechanisms like *Multi-KRUM* and *RFA* exhibit extreme variance, with their interquartile ranges stretching significantly toward low-accuracy zones (often $< 40\%$). This high instability empirically confirms Theorem 1: since these defenses fail to account for the mask-level distribution f_p, adversaries can effectively bypass geometric filters by concentrating malicious energy to become a "local majority" within coordinated sparse subspaces.

3 SafeSparse Framework

3.1 Overview of SafeSparse

The goal of the proposed *SafeSparse* framework is to enhance robustness against untargeted poisoning attacks in the context of sparsified, communication-efficient FL. As illustrated in Figure 2, *SafeSparse* is built on a common communication-efficient FL paradigm, where each client communicates with the server by transmitting only the most important model parameters using sparsification techniques. Specifically, each client performs local training on its data, applies top-k sparsification, and uploads the sparse model parameters along with a sparse index mask to the server. The server then aggregates the received sparse model updates to form the global model based on the sparse index mask.

To achieve the robustness guarantee and sparsification communication-efficiency in *SafeSparse*, two key components are designed at the server side: sparse index mask inspection and model update sign similarity analysis. Specifically, the sparse index mask inspection module filters out clients whose model updates are inconsistent with the majority by measuring the overlap in parameter selections across clients using the Jaccard similarity metric. The model update sign similarity analysis module, another critical component of *SafeSparse*, detects adversarial behavior patterns by grouping clients with similar update directions (i.e., the signs of parameter differences) using cosine similarity and applying density-based clustering to identify malicious clients. Clients identified as potentially malicious are excluded from the subsequent sparsified communication-efficient aggregation process, ensuring that only benign clients contribute to the final global model.

IJCAI-ECAI 2026 PREPRINT

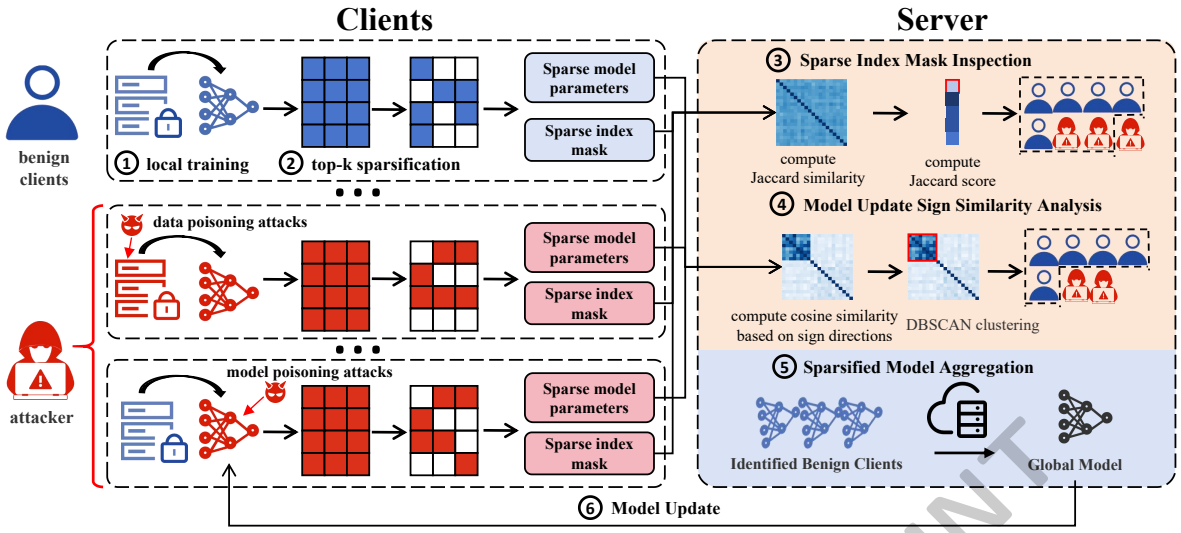

Figure 2: An illustration of the SAFESPARSE training process, incorporating client-side top-k sparsification and server-side robust and sparsified aggregation. It also highlights the two key components of SafeSparse: Sparse Index Mask Inspection and Model Update Sign Similarity Analysis. The red sections indicate attackers, who may carry out data poisoning attacks (such as label flipping attack) or model poisoning attacks (including IPM, Gaussian, and scaling attacks) to degrade the performance of the global model.

Security Model. In the *SafeSparse* framework, we consider a Byzantine threat model in which the proportion of adversarial participating clients is assumed to remain below 50% of the total clients. Each adversarial client is capable of launching poisoning attacks, aiming to manipulate the global model aggregation process by transmitting falsified model parameters during the FL process. This can be achieved either by directly manipulating the local model or by indirectly falsifying the local training samples. Consistent with the threat assumptions of existing poisoning attack studies, we assume that attackers do not have direct access to the local model updates or raw training data of benign participants. Furthermore, in this work, the primary adversarial objective is to conduct untargeted attacks designed to degrade the overall performance of the global model. It is important to note that privacy leakage threats, such as those caused by privacy inference attacks or gradient leakage attacks, are beyond the scope of this paper.

3.2 Sparsified and Robust Aggregation

Structural Integrity Verification via Jaccard Filtering

Typical sparsified frameworks rely on *sparse index masks*, which introduces the risk of *Index Poisoning*. As shown in our theoretical derivation (Appendix C), the success of an attack in sparse FL depends on the malicious contributor ratio f_p within each parameter pack. To minimize this ratio, we enforce structural consensus.

To further investigate their properties, we analyzed the structural characteristics of these masks and examined their consistency across clients. As shown in Figure 3, our empirical analysis reveals that, even under non-IID training settings, a significant degree of overlap exists among the masks generated by different clients. Leveraging this observation, we propose a filtering mechanism to eliminate potential adversarial clients whose sparse index masks exhibit low overlap with others or form isolated groups of similarity. Typically, clients with highly distinct sparse index masks introduce a significant degree of uncertainty in detecting poisoning attacks, thereby hindering the establishment of a robust defense. By filtering out these outliers, our approach enhances the reliability of attack detection and ensures a more consistent aggregation process.

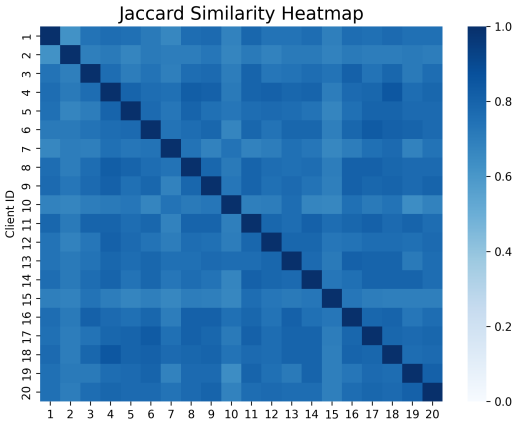

Figure 3: The heatmap of the Jaccard similarity between the *sparse index masks* of different clients on the CIFAR-10 dataset under the non-IID setting.

Specifically, we compute the Jaccard similarity between the sparse index masks of different clients. Formally, for each client i, let M_i represent its sparse index mask, which contains the indices of the selected model parameters after top-k sparsification. The Jaccard similarity between two clients i

IJCAI-ECAI 2026 PREPRINT

and j is defined as:

$$J(M_i, M_j) = \frac{|M_i \cap M_j|}{|M_i \cup M_j|} \tag{2}$$

where $|M_i \cap M_j|$ represents the number of common selected indices between the two clients, and $|M_i \cup M_j|$ represents the total number of unique selected indices.

For each client, we compute its Jaccard score as the average Jaccard similarity with all other clients, defined as:

$$J_{\text{SCORE}}(i) = \frac{1}{m-1} \sum_{j \neq i} J(M_i, M_j) \tag{3}$$

where m is the total number of participating clients. The Jaccard score quantifies the similarity of a client's sparse index mask to those of the other clients in the system.

To identify anomalous clients, we define a threshold for filtering as follows:

$$J_{\text{THRESHOLD}} = \frac{J_{\text{MAX}} + J_{\text{MIN}}}{2} \cdot \beta \tag{4}$$

Here, J_{max} and J_{min} represent the highest and lowest Jaccard scores among all clients, respectively, and β is a manually tuned hyperparameter, with a default value of 0.6. Any client with a Jaccard score below this threshold is classified as an outlier and is excluded from the global model aggregation process.

Semantic Consistency via Sign-based Clustering

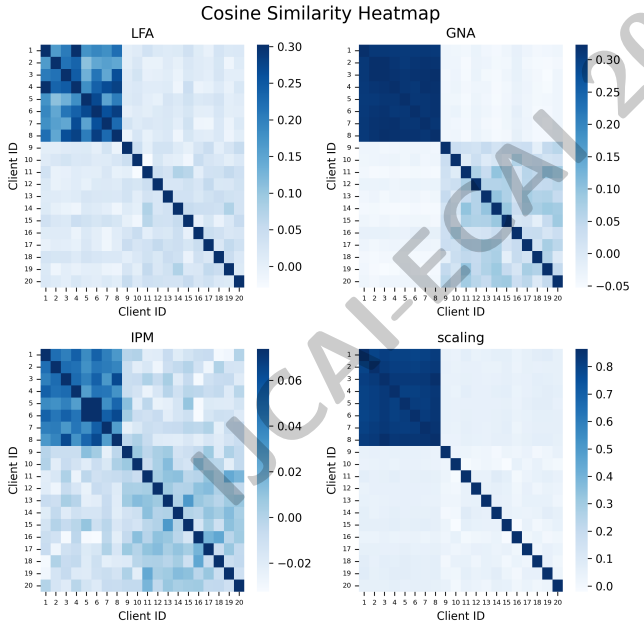

Figure 4: The heatmap of sign cosine similarity among clients on the CIFAR-10 dataset under the non-IID setting. The first 8 clients are adversaries, while the remaining 12 are benign clients.

While Jaccard filtering ensures structural integrity, it cannot detect malicious values within a valid mask (e.g., Label Flipping or IPM). To address this, we examine the *Semantic Consistency* of the updates. Since magnitude clipping is ineffective against scaling attacks in sparse regimes, we abstract the updates to their sign vectors $\text{Sign}(\Delta w_i)$. This transformation renders the defense robust to magnitude manipulation and focuses purely on the optimization direction.

As illustrated in Figure 4, the first eight clients represent malicious adversaries, while the remaining twelve are benign clients. The results demonstrate that the sign similarity among malicious clients is significantly higher than that among benign clients in the sparsified communication scenario. Furthermore, the similarity between malicious and benign clients remains low, further emphasizing the distinct update patterns introduced by adversarial attacks. It is important to note that, due to the nature of sparsification, the sign similarity between any two clients can only be computed over the overlapping regions of their sparse index masks. In short, our observations reveal that the success of poisoning attacks typically relies on controlling multiple adversarial clients and injecting relatively consistent perturbations into the model updates.

Building upon these insights, we design a *Model Update Sign Similarity Analysis* module to cluster potential malicious clients for detection. The process begins by computing the model parameter differences between the sparse model parameters uploaded by each client and the global model from the previous round, as $\Delta w_i = w_i - w_G$. These differences are then converted into their corresponding sign directions, defined as:

$$\text{Sign}(\Delta w_i) = \begin{cases} 1, & \text{if } \Delta w_i > 0, \\ -1, & \text{if } \Delta w_i < 0, \end{cases} \tag{5}$$

where w_i represents the model parameters from client i, and w_G denotes the global model parameters from the previous aggregation round. It is important to note that parameters for which no client participated in the aggregation during the previous round are excluded from the similarity calculation. This exclusion arises because the server lacks the corresponding model parameters for these positions, as they are determined solely by the local models of the clients, which are independently set by each client.

Next, we compute the pairwise cosine similarity of model update signs between clients, considering only the overlapping portions of their sparse index masks. This restriction ensures that the comparison is performed on the same model elements between clients, thereby making the similarity measurement meaningful. The cosine similarity, $\cos(\theta_{i,j})$, is defined as follows:

$$\cos(\theta_{i,j}) = \frac{\langle \text{Sign}(\Delta w_i), \text{Sign}(\Delta w_j) \rangle}{\|\text{Sign}(\Delta w_i)\| \cdot \|\text{Sign}(\Delta w_j)\|}, \tag{6}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product, and $\| \cdot \|$ represents the Euclidean norm.

Subsequently, the cosine similarity is transformed into a distance metric using $1 - \cos(\theta_{i,j})$, which quantifies the distance between the updates of two clients. The resulting distance matrix is then utilized as input to a clustering algorithm. Specifically, we employ the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm [Ester

IJCAI-ECAI 2026 PREPRINT

et al., 1996] within our framework, as detailed in Algorithm 1 in the Appendix B.1.

To determine the optimal value of ϵ, which defines the scan radius for core points in DBSCAN, we utilize the K-nearest neighbors (KNN) method. In this approach, ϵ is set as the average distance to the n-th nearest neighbor, where $n = N \cdot \gamma$. Here, N represents the total number of clients, and γ is a hyperparameter that controls the sensitivity of the clustering process. In practice, we set $\gamma = 0.2$ to balance sensitivity to adversarial clients and false positives. Additionally, the minimum number of points required to form a core point in DBSCAN is set to $N \cdot \gamma$, ensuring that each cluster contains a sufficient number of clients. This clustering process enables the effective grouping of clients based on similar attack patterns, while clients exhibiting significantly different update behaviors are identified as outliers. The resulting clusters facilitate the detection and exclusion of malicious clients from the global aggregation process. Consequently, only benign clients are included in the final global model aggregation, thereby enhancing the robustness of the framework.

Sparsified Robust Aggregation. Building on the recent work of Yan et al. [Yan *et al.*, 2024], *SafeSparse* adopts a pack-level top-k sparsification method. Unlike scalar-level sparsification, this method groups model parameters into small structured packs before applying the sparsification process. By operating at the pack level, this approach preserves the structured information within each group, providing a more efficient and cohesive representation of the model updates.

Specifically, when the server receives sparsified model parameters $\{W_1, ..., W_m\}$ and their sparse index masks $\{M_1, ..., M_m\}$ from m clients, it first resolves sparse index misalignment by identifying contributing clients $S_i = \{j | M_j(P_i) = 1\}$ for each packet P_i. This alignment ensures structural consistency across clients' heterogeneous sparse representations. The aggregation weight for client j is dynamically assigned as $\frac{|D_j|}{\sum_{k=1}^{m} |D_k|}$, proportional to its local dataset size $|D_j|$. For each P_i, the server calculates the total aggregation weight $A_{P_i} = \sum_{j \in S_i} \frac{|D_j|}{\sum_{k=1}^{m} |D_k|}$ to quantify cumulative contributions. Each model parameter $W_j(P_i)$ undergoes two sequential transformations: first scaled by its client-specific weight $W_j(P_i) \leftarrow W_j(P_i) * \frac{|D_j|}{\sum_{k=1}^{m} |D_k|}$ then normalized by A_{P_i} via $W_j(P_i) \leftarrow W_j(P_i)/A_{P_i}$ to eliminate dimensional variance caused by differing contributor counts. The aggregated parameter $w_i^{agg} = \sum_{j \in S_i} W_j(P_i)$ is computed through weighted summation. The detailed algorithm is provided in Appendix B.2.

It is important to note that our aggregation employs a dynamic normalization factor A_{P_i}. Unlike standard FL averaging which divides by N, *SafeSparse* normalizes by the count of *valid contributing clients* for each pack. This prevents the "gradient vanishing" problem for rarely selected parameters and ensures that the global model maintains the correct scale of gradient descent, as guaranteed by our convergence proof in Appendix D.

4 Evaluations

4.1 Convergence Analysis

We further establish the convergence of *SafeSparse* under standard assumptions and adversarial conditions (L-smoothness, bounded variance, and Top-k properties, see Appendix D.1).

Theorem 2 (Convergence of SafeSparse)**.** *Let $\alpha = k/d$ be the fraction of retained parameters. Under a μ-strongly convex objective and learning rate $\eta_t = \frac{8}{\mu(t+a)}$, the global model W_G^T after T rounds satisfies:*

$$\mathbb{E}\big[\|W_G^T - W^*\|^2\big] \leq \frac{D}{T+a} + \underbrace{\mathcal{O}\Big(\frac{(1-\alpha)G^2}{\alpha^2}\Big)}_{\textit{Sparsification Error}} + \underbrace{\mathcal{O}\Big(\frac{\rho}{\mu^2}\Big)}_{\textit{Residual Attack Impact}} . \tag{7}$$

where D is a constant, G^2 represents data heterogeneity, and ρ is the residual attack effectiveness defined in Theorem 1.

Theorem 2 demonstrates that *SafeSparse* converges to an error ball whose radius is explicitly controlled by the sparsity ratio α and the filtering effectiveness. By filtering outliers, *SafeSparse* restricts ρ, ensuring the model converges even in highly adversarial sparse environments. Detailed proof is deferred to Appendix D.

4.2 Default Experimental Settings

Unless stated otherwise, we use 20 clients with full participation, one local epoch per round, batch size 64, Adam optimizer with learning rate 0.005, and an attacker ratio of 0.4. Attacks start from round 10 and continue until training ends. We evaluate both IID and non-IID settings; the latter uses a Dirichlet split with $\alpha = 1$.

4.3 Datasets and Baselines

To evaluate the effectiveness and performance of *SafeSparse*, we conduct several experiments, consistent with the methodologies employed in existing related works. Specifically, we train a ResNet-20 model [He *et al.*, 2016] using SafeSparse and compare it against related baselines on three public datasets: FashionMNIST [Xiao *et al.*, 2017], CIFAR-10 [Krizhevsky *et al.*, 2009], and CIFAR-100 [Krizhevsky *et al.*, 2009].

SafeSparse is compared against a comprehensive set of baselines, including: FedAVG [McMahan *et al.*, 2017], Multi-KRUM [Blanchard *et al.*, 2017], Median [Yin *et al.*, 2018], Trimmed Mean [Yin *et al.*, 2018], FedSign [Guo *et al.*, 2023], Robust-DPFL [Qi *et al.*, 2024], LASA [Xu *et al.*, 2025], FLGuardian [Zhou *et al.*, 2025], and RFA [Pillutla *et al.*, 2022]. The full technical descriptions and mathematical formulations for each baseline are detailed in Appendix E.1.

To evaluate *SafeSparse*'s resilience to poisoning attacks, we conducted experiments on both IID and non-IID datasets under four common poisoning attack scenarios: Label Flip Attack (LFA) [Fang *et al.*, 2020], Gaussian Noise Attack

IJCAI-ECAI 2026 PREPRINT

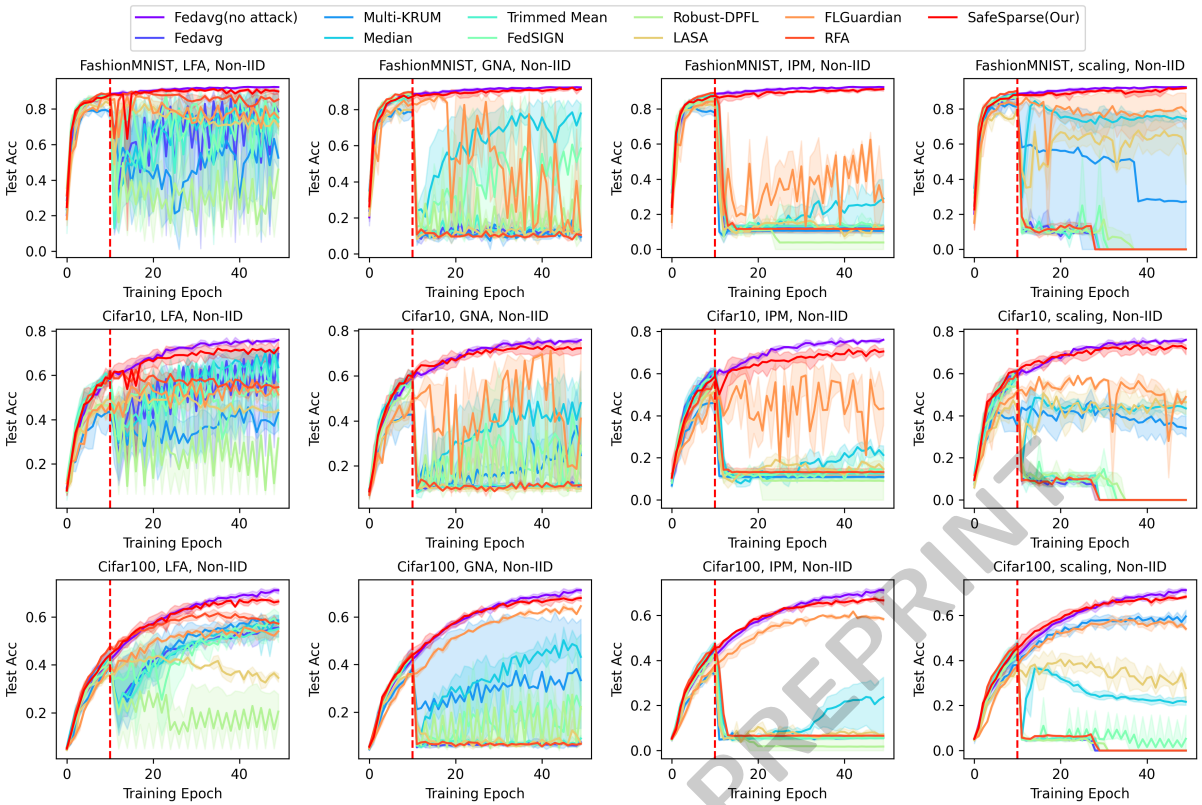

Figure 5: Comparison of defense effectiveness across various approaches, evaluated on FashionMNIST, CIFAR-10, and CIFAR-100 under Label Flip Attack (LFA), Gaussian Noise Attack (GNA), Inner Product Manipulation (IPM), and Scaling Attack in a **NON-IID SETTING**. The attacker ratio is fixed at 0.4, and the top-k sparsification rate is set to 0.5. Top-1 accuracy is used as the evaluation metric for FashionMNIST and CIFAR-10, while top-5 accuracy is adopted for CIFAR-100.

(GNA) [Fang *et al.*, 2020], Inner Product Manipulation (IPM) [Xie *et al.*, 2020], and Scaling Attack [Bagdasaryan *et al.*, 2020]. Detailed definitions and the implementation mechanisms for these four attack strategies are further elaborated in Appendix E.1.

4.4 Attack Effectiveness in Sparsified FL

As illustrated in Figure 5, we demonstrate the failure of most existing defense strategies against poisoning attacks in the context of sparsified communication-efficient FL. Different types of poisoning attacks commence at the 10th training epoch (marked by the red dashed line), and their impact on common sparsified aggregation is evident. Specifically, FedAvg fails to maintain stability, particularly under Scaling and Inner Product Manipulation (IPM) attacks, where the test accuracy drops drastically after the attack begins. A complementary evaluation under the IID setting is provided in Appendix E.2, which further corroborates these vulnerabilities.

Existing Federated Learning (FL) defenses, including Multi-KRUM, Median, and Trimmed Mean, exhibit limited and inconsistent resilience against poisoning attacks. Their effectiveness is highly sensitive to the dataset and attack type; for instance, while some succeed on CIFAR-10, they fail catastrophically under IPM or scaling attacks on CIFAR-100. Even advanced methods like FedSIGN and Robust-DPFL struggle with high-intensity attacks on complex datasets. These vulnerabilities are exacerbated in sparsified FL. Traditional robust aggregators rely on full, aligned updates, which are disrupted by the inconsistent parameter subsets inherent in sparsification. Similarly, anomaly detectors like FedSIGN lose detection granularity as sparse representations obscure critical patterns. This allows adversaries to exploit sparse indices to evade detection, highlighting an urgent need for sparsity-aware robust mechanisms.

4.5 Model Performance and Defense Effectiveness

To evaluate our proposed *SafeSparse*, we compare the model performance and defense effectiveness of *SafeSparse* with those of baseline methods. As shown in Figure 5, SafeSparse maintains high test accuracy and stability even after the attack is introduced, outperforming other defenses in multiple adversarial settings. Specifically, Scaling and Inner Product Manipulation (IPM) attacks cause many baseline methods to collapse entirely, while *SafeSparse* remains robust, preserving high accuracy. These results indicate the effectiveness of *SafeSparse* against various poisoning attacks in adversarial federated learning environments. Detailed comparative results on IID datasets and the corresponding convergence curves are available in Appendix E.2.

IJCAI-ECAI 2026 PREPRINT

Dataset	Attack	$\beta = 0.20$	$\beta = 0.40$	$\beta = 0.60$	$\beta = 0.80$	$\beta = 1.00$
FashionMNIST	LFA	0.9127	**0.9258**	0.9156	0.9000	*0.8705*
	GNA	0.8990	*0.8594*	**0.9251**	0.8904	0.9183
	Scaling	0.8848	0.9092	0.9090	**0.9161**	*0.8749*
	IPM	0.8988	*0.8438*	**0.9165**	0.8572	0.8761
CIFAR-10	LFA	*0.6399*	0.6706	**0.7472**	0.6836	0.6504
	GNA	**0.7243**	0.6839	0.6537	0.7065	*0.6409*
	Scaling	0.6088	0.6888	**0.7104**	0.7019	*0.5281*
	IPM	**0.7151**	0.6764	0.6887	0.6648	*0.6519*
CIFAR-100	LFA	0.6622	**0.6780**	*0.6534*	0.6626	0.6301
	GNA	0.6775	*0.6639*	**0.6875**	0.6691	0.6561
	Scaling	0.6611	**0.6784**	0.6453	0.6658	*0.6009*
	IPM	**0.6723**	0.6669	0.6671	0.6709	*0.5675*

Table 1: Comparison of defense effectiveness across different β settings (Non-IID Distribution).

Dataset	Attack	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
FashionMNIST	LFA	0.9163	**0.9223**	0.9102	0.8777	0.9044	0.8520	0.9216	0.5873	0.7200
	GNA	0.7871	0.9110	0.9124	0.8300	0.8899	**0.9242**	0.8908	0.1069	0.1315
	Scaling	0.8746	**0.9245**	0.9061	0.8571	0.9107	0.8524	0.8912	0.1164	0.1365
	IPM	**0.8935**	0.7162	0.8907	0.7216	0.5947	0.8159	0.7978	0.1226	0.1071
CIFAR-10	LFA	**0.7376**	0.5951	0.7104	0.6769	0.7297	0.6594	0.7169	0.5772	0.4913
	GNA	0.6547	0.6479	0.7072	0.6185	0.6431	0.6995	**0.7251**	0.1311	0.1241
	Scaling	0.7044	0.7198	0.6897	0.7158	**0.7256**	0.7173	0.6854	0.0901	0.1034
	IPM	0.1474	**0.7043**	0.6836	0.6756	0.6032	0.5381	0.4689	0.1370	0.1337
CIFAR-100	LFA	0.6525	0.6621	0.6572	**0.6711**	0.6669	0.6596	0.6561	0.5099	0.5271
	GNA	0.6662	0.6599	0.6704	0.6786	0.6740	**0.6840**	0.6799	0.0589	0.0698
	Scaling	**0.6810**	0.6734	0.6684	0.6733	0.6547	0.6669	0.6599	0.0482	0.0572
	IPM	0.5944	0.6525	**0.6791**	0.5347	0.4912	0.6531	0.6146	0.0617	0.0586

Table 2: Comparison of defense effectiveness across different γ settings (Non-IID Distribution).

IJCAI-ECAI 2026 PREPRINT

4.6 Ablation Experiments

To further evaluate the robustness and adaptability of *SafeSparse*, we conduct ablation experiments on its two key hyperparameters: the filtering threshold β and the clustering sensitivity γ, as illustrated in Table 1 and Table 2 , while the corresponding results for the IID distribution are provided in Appendix E.3.

Impact of Filtering Threshold β. Filtering threshold controls the cutoff value for sparsity mask similarity when filtering out potentially poisoned clients. We experiment with five values: $\beta \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. As shown in our results, SafeSparse maintains effective defense performance under all settings, indicating its robustness to β variations. However, higher values of β (e.g., 0.8 or 1.0) lead to over-filtering, where benign clients are mistakenly excluded from training, reducing model diversity and performance. On the other hand, lower values retain more benign clients while still effectively filtering out attackers. We adopt $\beta = 0.6$ in our main experiments as it strikes a good balance between defense strength and benign client retention.

Impact of Clustering Sensitivity Parameter γ. Clustering sensitivity parameter influences the neighborhood size for DBSCAN-based client clustering. We vary γ from 0.1 to 0.5 in increments of 0.05. The results reveal that excessively high values (e.g., $\gamma > 0.4$) cause the clustering step to fail in separating attackers from benign clients, leading to ineffective defense.

Impact of Attacker Ratio and Top-k Ratio. *SafeSparse* was evaluated with attacker ratios from 0.2 to 0.4 on CIFAR-100. Results show that test accuracy remains stable regardless of the increasing proportion of malicious clients, confirming the robustness of our defense. See Appendix E.4 for detailed performance curves. We also tested sparsification ratios between 0.4 and 0.6. *SafeSparse* maintains strong resilience across all levels, though lower ratios may slightly affect attacker detection stability. Further experimental data and analysis are provided in Appendix E.5.

5 Conclusion

We studied poisoning vulnerabilities in sparsified communication-efficient FL and showed that sparse updates undermine dense robust aggregation assumptions. We proposed *SafeSparse*, which combines mask-level structural filtering and sign-based semantic clustering to detect and remove malicious clients before sparse aggregation. Experiments demonstrate that *SafeSparse* improves robustness across datasets, attacks, and sparsification settings while preserving communication efficiency.

Ethical Statement

There are no ethical issues.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62225202, 62302022, and 62472022).

References

[Bagdasaryan *et al.*, 2020] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020.

[Blanchard *et al.*, 2017] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

[Ester *et al.*, 1996] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[Fang *et al.*, 2020] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX security symposium (USENIX Security 20)*, pages 1605–1622, 2020.

[Guo *et al.*, 2023] Zhenyuan Guo, Lei Xu, and Liehuang Zhu. Fedsign: A sign-based federated learning framework with privacy and robustness guarantees. *Computers & Security*, 135:103474, 2023.

[Han *et al.*, 2020] Pengchao Han, Shiqiang Wang, and Kin K Leung. Adaptive gradient sparsification for efficient federated learning: An online learning approach. In *2020 IEEE 40th international conference on distributed computing systems (ICDCS)*, pages 300–310. IEEE, 2020.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Kairouz *et al.*, 2021] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

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

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[Pillutla *et al.*, 2022] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

[Qi *et al.*, 2024] Tao Qi, Huili Wang, and Yongfeng Huang. Towards the robustness of differentially private federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19911–19919, 2024.

[Sattler *et al.*, 2019a] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 31(9):3400–3413, 2019.

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

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

[Xie *et al.*, 2020] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Uncertainty in Artificial Intelligence*, pages 261–270. PMLR, 2020.

[Xu *et al.*, 2025] Jiahao Xu, Zikai Zhang, and Rui Hu. Achieving byzantine-resilient federated learning via layer-adaptive sparsified model aggregation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1508–1517. IEEE, 2025.

[Yan *et al.*, 2024] Nan Yan, Yuqing Li, Jing Chen, Xiong Wang, Jianan Hong, Kun He, and Wei Wang. Efficient and straggler-resistant homomorphic encryption for heterogeneous federated learning. In *IEEE Conference on Computer Communications*, pages 791–800. IEEE, 2024.

[Yin *et al.*, 2018] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*, pages 5650–5659. Pmlr, 2018.

[Zhou *et al.*, 2025] Xingjie Zhou, Xianzhang Chen, Shukan Liu, Xuehong Fan, Qiao Sun, Lin Chen, Meikang Qiu, and Tao Xiang. Flguardian: Defending against model poisoning attacks via fine-grained detection in federated learning. *IEEE Transactions on Information Forensics and Security*, 2025.

IJCAI-ECAI 2026 PREPRINT