TIMBRE: BACKDOOR DEFENSE IN FEDERATED LEARNING VIA LAYER-WISE SPECTRAL ENERGY DISTRIBUTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning is vulnerable to backdoor attacks, and strong data heterogeneity lets malicious updates hide among benign updates that already differ. Advanced backdoors mimic benign updates in norm, distance, or overall direction, weakening defenses built on these statistics. We observe that backdoored updates can differ in how their energy is allocated within a layer even at comparable magnitudes. We therefore propose TIMBRE, a server-side defense that uses only individual client updates, without labels or trusted reference data. TIMBRE computes each update's normalized spectral energy distribution layer by layer, which is invariant to overall scale. TIMBRE compares an update's distribution with those of the other clients in the same round, and the further it departs beyond their typical spread, the stronger the backdoor evidence. Rather than filtering or removing clients, TIMBRE grades its restrictions by the strength and persistence of the evidence: it suppresses modifications lacking support from other clients, keeps useful contributions, and lifts restrictions as evidence fades. In experiments on multiple datasets and models under five representative attacks, TIMBRE keeps every attack success rate in the main setting below 1.4% with a clean-accuracy cost of at most 1.5 percentage points; it outperforms the eight evaluated defenses overall and keeps the attack success rate low as data heterogeneity and the malicious participation ratio vary.

1 INTRODUCTION

Federated learning lets multiple parties train a global model collaboratively without sharing their local data (McMahan et al., 2017). The same decentralization exposes training to backdoor attacks: a few malicious clients inject into the global model a targeted behavior that is activated by a specific trigger pattern, while preserving main-task performance (Bagdasaryan et al., 2020; Xie et al., 2020). Such attacks are hard to notice because the server sees neither the clients' data nor their training; all it can inspect is the model update uploaded in each round. Data heterogeneity narrows this already limited view further. Benign clients hold data from different distributions, so their updates naturally diverge from one another, and an attacker can hide malicious modifications within this spread.

Advanced backdoors can make malicious updates close to benign ones in common statistics such as norm, distance, or overall direction (Lyu et al., 2023; Li et al., 2023). These statistics, however, describe only how much an update changes and where it points as a whole, and heterogeneous benign updates already differ in both. The question is therefore which property of an update can still separate backdoor modifications from benign variation. Backdoor training repeatedly reinforces a fixed association between the trigger pattern and the target label, so the corresponding modifications may accumulate along similar directions in some layers. Such accumulation need not increase the overall strength of the update noticeably, but it can change how energy is allocated within a layer, that is, whether the energy concentrates in a few principal components or spreads across many. In our experiments we observe that, even when overall magnitudes are comparable, malicious and benign updates still differ discernibly in their intra-layer energy allocation (Section 3). Because this allocation is invariant to the overall scale of an update and the orientation of its singular vectors and can be judged against the other clients of the same round, it can serve as backdoor evidence that is not directly confounded by the spread that heterogeneity causes among benign updates.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Even if intra-layer energy allocation can be read out reliably, turning it into a usable defense faces two obstacles. First, one anomalous observation is not an attack. Benign training produces transient fluctuations, and a stealthy attacker may spread its footprint over many rounds; the server must judge when evidence warrants a restriction and when it should be lifted, without freezing a single suspicious observation into a verdict on the client. Second, evidence worth acting on does not mean the whole update is worthless. A suspicious update often still carries useful main-task learning, so the server must suppress its risky influence without discarding the client's valid contribution.

We propose TIMBRE, a server-side backdoor defense that takes intra-layer spectral energy allocation as its evidence. Before aggregation, the server applies a singular value decomposition to each uploaded update layer by layer and reads the proportion of energy allocated to each component. With the other clients of the same round as the reference, it then obtains a layer-wise spectral score for the update. TIMBRE then accumulates this score across rounds according to its current strength and persistence, and sets the degree of restriction from the current evidence and the accumulated score together. Restrictions are applied parameter by parameter: only the parameters that receive enough same-sign modifications from the clients not targeted in the current round are kept, while the rest are zeroed and the aggregation weight of the upload is lowered accordingly. The accumulated score falls back as later evidence disappears and the restrictions are lifted with it, so the method neither discards an entire update nor removes any client from training.

Our main contributions are as follows:

1. **A new evidence perspective and a layer-wise spectral score.** We propose a server-side detection score based on the normalized spectral energy distribution within each layer. It characterizes structural differences between updates beyond their overall magnitude, and we examine its role as backdoor evidence through analysis and experiments.
2. **A graded response that releases itself.** The response follows the strength and persistence of the evidence and is lifted as the evidence disappears. It acts at the granularity of parameters rather than accepting or rejecting a whole update, which is intended to reduce the loss of main-task information that comes with discarding suspicious updates entirely. The entire response replaces only the server's aggregation step and leaves client training unchanged.
3. **Systematic evaluation.** We conduct extensive experiments to evaluate the effectiveness of TIMBRE. The results show that TIMBRE defends against multiple representative backdoor attacks and outperforms the existing defenses we compare against when backdoor suppression and main-task performance are considered together.

2 Threat Model

2.1 Adversary Model

The adversary aims to make the global model produce a preset target behavior on inputs that carry the trigger pattern, while the model behaves normally on the main task. We assume that the adversary fully controls the local data, objective, optimizer, and every uploaded update of the malicious clients, and can share training data and model parameters among them and coordinate their uploaded updates. The adversary does not control benign clients, cannot read their uploaded updates, and cannot modify the server code. Client identities are authenticated and stable across rounds, so the adversary cannot clear its accumulated history by forging or rotating identities.

2.2 Defender Model

The method runs on the server of a synchronous horizontal federated learning system. Let $\mathcal{S}_t$ denote the set of participants in round t; the clients hold local data from different distributions, and the update Δ_k^t uploaded by client k is its locally trained model minus the global model it received.

The server reads only the updates uploaded by the clients. It reads no client data, labels, local predictions or activations, or attack ground truth, holds no clean or out-of-distribution samples and no trusted reference template, and therefore needs no attack-free clean period during deployment. We assume that malicious clients make up less than half of the participants in every round.

Within these boundaries, the method pursues three goals: lowering the attack success rate on inputs that carry the trigger pattern (backdoor suppression); preserving, as far as possible, model utility under normal training and the valid contributions of benign clients at a quantifiable cost (utility preservation); and imposing graded, releasable restrictions according to the strength and persistence of the evidence without irreversible exclusion of clients (controlled response).

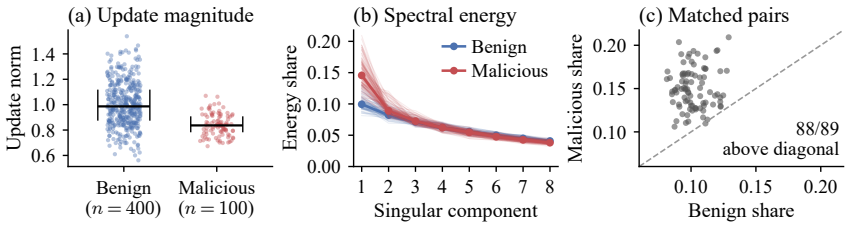

Figure 1: Spectral energy allocation of the weight update of one convolutional layer in a BadNets example. (a) Frobenius norms of the weight updates. (b) First 8 components of the normalized spectral energy distribution (median, interquartile band, individual samples). (c) Each point is a benign–malicious pair from the same round with similar norms; the horizontal axis gives the first-component energy share of the benign member and the vertical axis that of the malicious member. Points above the dashed line $y = x$ mean that the malicious member has the higher first-component share, that is, more concentrated energy. Setup and pairing rule in Appendix C.5.

3 Motivating Analysis

3.1 Limitations of Existing Criteria

Defenses that select among clients before aggregation mostly rely on statistics such as the distance or similarity between updates, or the direction and per-coordinate signs of an update (Nguyen et al., 2022; Xu et al., 2025b; Ozdayi et al., 2021). Some attacks designed against such defenses use distance or similarity constraints to bring malicious updates close to benign ones in the corresponding statistics (Lyu et al., 2023; Li et al., 2023). Under strong data heterogeneity, the criteria that depend on benign updates being close to one another also face benign updates that diverge from each other, so the gap between suspicious and normal updates narrows further. Even when an update is close to benign updates in distance or similarity, it may still contain backdoor modifications. This motivates us to look for complementary evidence in the energy allocation of each layer's weight update and examine whether backdoor modifications leave changes that these statistics fail to reflect.

3.2 Measurement Target: Intra-layer Spectral Energy Allocation

Beyond the main task, backdoor training repeatedly reinforces one fixed rule: inputs that carry the trigger pattern should be assigned to the designated target class. This repeated objective may make the corresponding modifications accumulate along similar directions in some layers.

This can be understood from how gradients are formed. For a single sample, let W denote the weight matrix of a linear layer or the unfolded weight matrix of a convolutional layer; its gradient can be written as $\nabla_W \mathcal{L} = \sum_s e_s a_s^\top$, where a_s is the input activation, e_s is the corresponding back-propagated error signal, and s indexes spatial positions. A fully connected layer has a single term, and a convolutional layer sums over positions. Each outer product ties one input-activation direction to one error direction and contributes a gradient of rank at most one. A local update accumulates many such contributions over samples and training steps, so its energy allocation depends on whether these directions recur and accumulate together during training.

Backdoor training is one possible source of such recurrence: poisoned samples repeatedly ask the model to build the same association between the trigger pattern and the target class. When the model learns this association in some layers through similar activation and error directions, the contributions of many training steps may accumulate mainly along these directions. The corresponding modifications then have a low effective rank, which appears in the spectrum as a few components holding a large share of the energy.

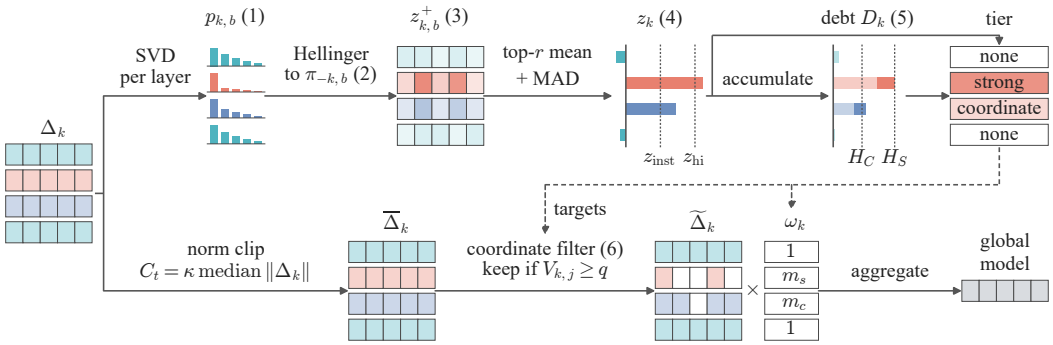

Figure 2: Overview of TIMBRE in one round. Four clients are shown, one strip each: two benign (teal), one benign with skewed local data (blue), and one malicious (red). Top row: each raw update is scored layer by layer against the other clients, and the current-round evidence z_k and the debt D_k (lighter part from earlier rounds) set its tier. Bottom row: the tier decides which clipped updates are filtered coordinate-wise against the non-targets (white cells are zeroed) and how each is weighted in aggregation. The benign clients stay in the none tier, the skewed benign client reaches only the coordinate tier, and the malicious client reaches the strong tier. Numbers in parentheses refer to equations.

In Figure 1, the norm distributions of benign and malicious updates overlap (Figure 1(a)), yet malicious updates place a larger share of energy in the first component (Figure 1(b)), consistent with the low effective rank expected above. To rule out magnitude and round effects, Figure 1(c) pairs malicious updates with the benign updates of the same round whose norms are closest and compares their first-component shares; the malicious update has the higher first-component share in 88 of 89 pairs. In this example, closeness in magnitude does not hide the difference in spectral energy allocation. Accordingly, we compare the spectral energy distributions of weight updates layer by layer. This distribution depends only on the relative sizes of the singular values and does not change with the overall scale of an update or the orientation of its singular vectors, so the divergence of benign updates in direction and magnitude under strong heterogeneity does not enter it directly.

4 Method

4.1 Overview

As shown in Figure 2, TIMBRE scores each client's update layer by layer against the other clients of the round on the same layer and aggregates the per-layer deviations into the current-round evidence. The server accumulates this evidence over each of a client's participations into a score; the current-round evidence reflects the strength of a deviation and the accumulated score its persistence, and together they determine which clients are processed and how strongly. Finally, the server clips all updates, applies coordinate processing to the target updates using the clients not selected as targets as the reference, and adjusts the aggregation weights.

4.2 Layer-wise Spectral Score

For the update Δ_k^t uploaded by client k in round t, we focus on how its energy is allocated internally. Flattening the whole update into one vector would let layers with more parameters or larger per-parameter changes take a larger share of its total energy and mask local changes in other layers, so each layer's weight update is analyzed separately. Let $\mathbf{E}_{k,b}^t$ denote the matrix representation of the b-th weight update of client k in round t that is subject to analysis, and apply a singular value decomposition to this matrix.

Arrange all d_b singular values of the matrix in non-increasing order as $\sigma_{k,b,1}^t, \ldots, \sigma_{k,b,d_b}^t$ and compute the energy share of each component:

$$p_{k,b,\ell}^t = \frac{(\sigma_{k,b,\ell}^t)^2}{\sum_{a=1}^{d_b} (\sigma_{k,b,a}^t)^2}. \tag{1}$$

Here the denominator is the total spectral energy of the matrix, which equals its squared Frobenius norm, and the component shares $p^t_{k,b,\ell}$ form the normalized spectral energy distribution $p^t_{k,b}$.

Same-round reference. To measure how far the spectral distribution of client k deviates from the other clients of the round, we take the component-wise median of the spectral distributions of the other participants and renormalize it to unit sum, obtaining the reference spectrum $\pi^t_{-k,b}$. We then measure the Hellinger distance between a client's spectral distribution and this reference:

$$\mathcal{H}^t_{i;-k,b} = \frac{1}{\sqrt{2}} \left\| \sqrt{p^t_{i,b}} - \sqrt{\pi^t_{-k,b}} \right\|_2 . \tag{2}$$

Here $\mathcal{H}^t_{i;-k,b}$ is the distance from the spectral distribution of client i to the reference spectrum $\pi^t_{-k,b}$. The same distance for client k stands out only when the distances of the other clients to the same reference are generally smaller and concentrated. Client k's distance is therefore standardized with the median of the other clients' distances as the center and a robust scale based on the median absolute deviation (MAD) as the unit, and only the part above the median is kept, giving the anomaly score of client k on this update matrix:

$$z^{t,+}_{k,b} = \max\Big\{0,\ \frac{\mathcal{H}^t_{k;-k,b} - \mathrm{median}_{i\in\mathcal{S}_t\setminus\{k\}}\, \mathcal{H}^t_{i;-k,b}}{1.4826\,\mathrm{MAD}_{i\in\mathcal{S}_t\setminus\{k\}}\{\mathcal{H}^t_{i;-k,b}\}}\Big\}. \tag{3}$$

Here 1.4826 is the scale conversion factor of the MAD. Spectral differences among benign clients caused by data heterogeneity appear in the spread of the other clients' distances and are absorbed by this scale, so only deviations well beyond it receive a high score. Appendix A relates spectral energy allocation to low-rank approximation and proves a lower bound on the spectral distance.

Client evidence. If the deviation is concentrated in a few update matrices, averaging over all matrices would dilute it. For each client we therefore use the mean of its r highest matrix scores as the aggregated score and measure its relative anomaly against the aggregated scores of all participants:

$$S^t_k = \frac{1}{r}\sum_{m=1}^{r} z^{t,+}_{k,(m)}, \qquad z^t_k = \frac{S^t_k - \mathrm{median}_{i\in\mathcal{S}_t}\, S^t_i}{1.4826\,\mathrm{MAD}_{i\in\mathcal{S}_t}\{S^t_i\}}. \tag{4}$$

Here S^t_k is the aggregated score across matrices and z^t_k is the current-round evidence used in the response decisions; r is a preset count, and (m) denotes the rank position of the matrix scores.

4.3 Strength- and Persistence-based Response

Current-round evidence describes a single update and cannot tell whether similar deviations appeared before. On heterogeneous data, benign training produces transient fluctuations, and an attacker may spread backdoor modifications over many rounds, so responding to single-round evidence alone may restrict benign clients or overlook deviations that recur without being individually strong. We therefore maintain an accumulated score for each client, updated by the current-round evidence at each participation, to capture the persistence of deviation, and we keep an immediate response for cases where the current-round evidence alone is strong enough, since some attacks implant a backdoor within a few rounds.

Specifically, we call the accumulated score the debt D^t_k and initialize it to 0 when the defense is enabled; after scoring in each round, it is updated for every participant of the round as

$$\delta^t_k = \mathrm{clip}(z^t_k - \lambda, -c_-, c_+), \qquad D^t_k = \mathrm{clip}(D^{t-1}_k + \delta^t_k, 0, H_S). \tag{5}$$

Here $\mathrm{clip}(x, a, b)$ truncates x to the interval $[a, b]$; δ^t_k is the debt increment brought by this participation, obtained by truncating the difference between the current-round evidence and λ; c_+ and c_- are the caps on a single increase and decrease, H_S is the cap on the debt, and rounds in which a client is absent leave its debt unchanged. One very strong piece of evidence does not push the debt to H_S at once, so a strong response triggered by debt (the history-triggered strong tier) forms only after repeated deviations. The decrease cap is half of the increase cap, so the debt falls more slowly than it accumulates, and a brief pause in the attack does not immediately cancel the earlier accumulation. A client becomes a coordinate-processing candidate once its debt reaches the threshold H_C, and enters the history-triggered strong tier once its debt reaches H_S; the two debt thresholds satisfy $0 < H_C < H_S$.

Among the participants of the round, the clients whose current-round evidence reaches the instantaneous strong-response threshold z_{hi}, or whose debt reaches H_S, enter the strong-response set (the strong tier); the remaining clients whose current-round evidence exceeds the instantaneous coordinate-processing threshold z_{inst} become instantaneous coordinate targets without waiting for debt to accumulate. The two evidence thresholds satisfy $\lambda \leq z_{\mathrm{inst}} < z_{\mathrm{hi}}$.

4.4 Update Processing

This section describes norm clipping, coordinate processing, and tier-based reweighting in turn.

Norm clipping of all updates. The normalized spectral distribution reflects energy shares, so two updates with similar spectral distributions may still differ greatly in overall magnitude, and a norm cap is still required. Since the norms of benign updates change as training proceeds, we take the median norm of all participants' updates in the round as the base and set a common cap for the round, $C_t = \kappa \,\mathrm{median}_{k \in \mathcal{S}_t} \|\Delta_k^t\|_2$, where $\kappa > 0$ controls the cap as a multiple of the median. After scoring, all participants' updates are clipped with this cap to $\overline{\Delta}_k^t = \Delta_k^t / \max\{1, \|\Delta_k^t\|_2 / C_t\}$.

Coordinate filtering. Discarding a suspicious update as a whole also loses the main-task information it carries, so we filter the clipped update of a target client coordinate by coordinate and keep only the modifications that receive enough same-sign support from the reference clients. The set of processing targets in the round is drawn from the strong-tier clients of Section 4.3, the instantaneous coordinate targets, and the coordinate-processing candidates whose debt has reached H_C. To reduce the influence of suspicious updates on the coordinate decisions, the reference set $\mathcal{P}_t$ consists of the clients that participate in the round but are not selected as targets.

For each reference client i, we consider only the coordinates with the largest absolute values within each tensor of its clipped update, up to a fixed proportion; we call them active coordinates and denote their set by $\mathcal{K}_i^t$. For each coordinate j of the target client k, let $V_{k,j}^t$ be the number of reference clients that are active on that coordinate and share the sign of client k there, that is,

$$V_{k,j}^t = \sum_{i \in \mathcal{P}_t} \mathbf{1}\Big[j \in \mathcal{K}_i^t \wedge \mathrm{sign}\Big(\overline{\Delta}_i^t[j]\Big) = \mathrm{sign}\Big(\overline{\Delta}_k^t[j]\Big) \neq 0 \Big]. \tag{6}$$

Here $[j]$ denotes the j-th coordinate. A coordinate passes the filter when its support count $V_{k,j}^t$ reaches the threshold q. We set q to a fixed value greater than one, so that a coordinate passes only with same-sign support from several reference clients and cannot be kept by an individual reference client alone.

Magnitude interval. Norm clipping only bounds the total magnitude of an update, and a coordinate that passes the filter may still carry a value larger than those of the reference clients on that coordinate. A clipping interval is therefore set from the values of the reference clients' clipped updates on that coordinate. To limit the effect of extreme values, we drop a fixed proportion of the most extreme reference values on each side of every coordinate and determine the clipping range from the remaining values. For a selected suspicious update, the coordinates that pass the filter are clipped into the corresponding range and the others are set to zero, which gives the processed update $\widetilde{\Delta}_k^t$. The other updates keep their norm-clipped form, that is, $\widetilde{\Delta}_k^t = \overline{\Delta}_k^t$.

Reweighting. For the selected suspicious updates, coordinate filtering and clipping follow the same rule, and their weights are then adjusted at aggregation according to the strength of the response. Let ω_k^t denote the weight multiplier of client k in the round: strong-tier clients take m_s, the remaining clients of the final target set form the coordinate tier and take m_c, and non-target clients take 1, where $0 < m_s < m_c < 1$. The multiplier applies only to the current round and is reassigned in the next round according to the new target set. Finally, the processed updates are aggregated into the global model with the base weights multiplied by these multipliers and normalized.

5 Experiments

5.1 Experimental Setup

We consider a federated system with 100 clients, of which 10 are selected in each round. By default, each round includes 2 malicious clients, that is, a malicious participation ratio of 20%; LGA (Yang

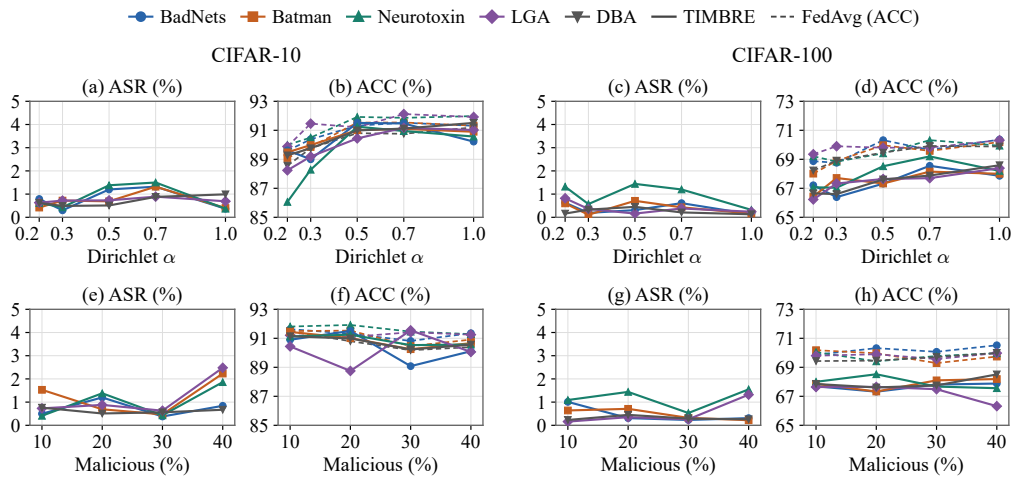

Figure 3: Heterogeneity and malicious-participation sweeps of ResNet18 on two datasets. The top row sweeps the Dirichlet parameter and the bottom row the malicious participation ratio at $\alpha = 0.5$, both under five attacks; the left and right groups are CIFAR-10 and CIFAR-100. Solid lines denote TIMBRE and dashed lines denote FedAvg (ACC only). Complete values are in Appendix D.2.

et al., 2025) follows the settings of its official implementation, with only 1 attacker per round and more poisoning rounds. For each dataset and model, one parameter set, with thresholds from a separate no-attack run (Appendix C.4), is used across all attack and data-distribution settings.

Datasets and models. To evaluate TIMBRE across datasets and model architectures, we run experiments with ResNet18 (He et al., 2016) on CIFAR-10 (Krizhevsky, 2009), CIFAR-100, and Tiny-ImageNet, and with VGG11 (Simonyan & Zisserman, 2015) on CIFAR-10 as a supplementary check. Data are partitioned with a sample-balanced Dirichlet distribution, with $\alpha = 0.5$ by default. Each setting continues training from its own clean pretrained model. In addition, we use an LSTM (Hochreiter & Schmidhuber, 1997) on the IMDB (Maas et al., 2011) sentiment classification task to supplement the evaluation in a text setting.

Attacks and defenses. The main experiments evaluate TIMBRE under five backdoor attacks: BadNets (Gu et al., 2019), Batman (He et al., 2026), Neurotoxin (Zhang et al., 2022b), DBA (Xie et al., 2020), and LGA. We also construct PSM-BadNets, an adaptive attack designed against TIMBRE. We take FedAvg (McMahan et al., 2017) as the no-defense baseline and compare with eight server-side defenses: Multi-Krum (Blanchard et al., 2017), FLAME (Nguyen et al., 2022), DeepSight (Rieger et al., 2022), Multi-Metrics (Huang et al., 2023a), AlignIns (Xu et al., 2025b), MARS (Wan et al., 2025), RLR (Ozdayi et al., 2021), and FoolsGold (Fung et al., 2020).

Metrics. We evaluate defenses by the attack success rate (ASR) and the clean test accuracy (ACC). Detailed settings and metric definitions are given in Appendix C.

5.2 Results

Comparison with existing defenses. Table 1 compares the final ACC and ASR of TIMBRE, FedAvg, and eight existing defenses under five backdoor attacks. Overall, TIMBRE keeps the ASR at or below 1.38% under all five attacks, with ACC between 90.44% and 91.51%. Some defenses reach a low ASR under individual attacks but fail to keep it under others. For example, Multi-Metrics has an ASR of 0.20% under BadNets but 74.76% under Neurotoxin, and RLR has 0.28% under Batman but 97.27% under LGA. The difference is clearest under Neurotoxin: AlignIns reaches an ASR of 14.74%, and the other seven external defenses are no lower than 74.76%, whereas TIMBRE holds the ASR at 1.38% while keeping an ACC of 91.25%.

Backdoor suppression has to be weighed together with main-task performance. Under Batman, the ASR of RLR, 0.28%, is below that of TIMBRE, 0.69%, but their ACCs are 87.88% and 91.01%; the low ASR of Multi-Metrics under BadNets comes with an ACC of 85.01%, 6.31 percentage points below FedAvg. Under the four attacks other than LGA, the ACC of TIMBRE is within 0.7 percentage points of FedAvg.

Building on these experiments, we further examine TIMBRE on other datasets and model architectures. Under the default settings, the ASR of TIMBRE under the five attacks does not exceed 3.60% on CIFAR-10 $\times$ VGG11 and 1.44% on CIFAR-100 $\times$ ResNet18, and on Tiny-ImageNet $\times$ ResNet18 the ASR under BadNets, Batman, and Neurotoxin does not exceed 0.77%. The supplementary experiment on IMDB also shows that TIMBRE suppresses the Word-trigger and Neurotoxin text backdoor attacks. These results further support the defense capability of TIMBRE across datasets and model architectures. Complete results are given in Appendices D.1 and D.6.

Table 1: Final ACC and ASR (%) under five attacks on CIFAR-10 $\times$ ResNet18, $\alpha = 0.5$. Bold marks the best result among the nine defenses and underline the second best; FedAvg is the no-defense reference.

Defense	BadNets		Batman		Neurotoxin		DBA		LGA	
	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
FedAvg	91.32	76.88	91.53	89.03	91.92	83.06	90.80	56.19	87.62	94.57[1]
Multi-Krum	90.04	80.97	89.81	89.68	90.71	85.89	89.49	60.83	<u>90.91</u>	51.27
FLAME	81.03	69.19	82.24	85.04	80.83	82.73	80.11	71.56	82.68	97.58
DeepSight	91.22	73.76	90.57	91.64	90.95	83.86	90.24	56.63	90.90	13.88
Multi-Metrics	85.01	**0.20**	87.84	5.08	87.71	74.76	84.73	30.01	89.04	3.76
AlignIns	89.22	20.40	90.93	62.71	90.83	<u>14.74</u>	88.94	17.01	89.43	<u>1.29</u>
MARS	91.24	15.41	86.64	85.36	90.27	91.71	<u>90.97</u>	13.72	87.25	98.46
RLR	90.33	86.69	87.88	**0.28**	84.04	93.07	87.51	<u>2.01</u>	88.61	97.27
FoolsGold	<u>91.34</u>	78.16	**91.11**	89.31	**91.72**	81.49	90.81	57.28	**91.42**	31.04
TIMBRE	**91.51**	<u>1.20</u>	<u>91.01</u>	<u>0.69</u>	<u>91.25</u>	**1.38**	**90.99**	**0.51**	90.44	**0.73**

Data heterogeneity. Section 3.1 notes that strong heterogeneity makes benign updates diverge from one another, so we evaluate ResNet18 on CIFAR-10 and CIFAR-100 with $\alpha \in \{0.2, 0.3, 0.5, 0.7, 1.0\}$. A smaller α means stronger heterogeneity across clients, and each dataset and model combination keeps its own defense parameters throughout the sweep. In Figure 3(a)(c), the final ASR on the two datasets does not exceed 1.50% and 1.44%, respectively. In Figure 3(b)(d), the largest ACC drop of TIMBRE relative to FedAvg is 3.93 and 3.12 percentage points, respectively.

Malicious participation ratio. We assume that malicious clients make up less than half of the participants in each round, so at $\alpha = 0.5$ we increase the number of malicious clients f among the 10 participants per round from 1 to 4, corresponding to participation ratios of 10%–40%, with each dataset keeping its own defense parameters. Figure 3(e)(g) shows that the final ASR of the five attacks does not exceed 2.48% on CIFAR-10 and 1.55% on CIFAR-100; in Figure 3(f)(h), the largest ACC drop of TIMBRE relative to FedAvg is 2.34 and 3.66 percentage points, respectively.

Component ablation. Removing any of layer-wise spectral analysis, evidence accumulation, coordinate processing, or reweighting raises the ASR under Neurotoxin and Batman at $\alpha = 0.2$; the variants and full results are given in Appendix D.3.

Adaptive attack. To evaluate the robustness of TIMBRE further, we consider an attacker who knows the mechanism and parameters of TIMBRE but cannot read the updates of benign clients. Starting from the global model of the round, the attacker trains a separate model on its own unpoisoned data with the normal local training settings and takes the layer-wise spectral energy distribution of the resulting update as the reference; it then adds a regularization term to backdoor training that pulls the spectral distribution of the attack update toward this reference in order to evade detection. We denote this attack PSM-BadNets; its objective is $\mathcal{L} = \lambda_{\mathrm{atk}}\mathcal{L}_{\mathrm{bd}} + (1 - \lambda_{\mathrm{atk}})\mathcal{L}_{\mathrm{spec}}$, where $\mathcal{L}_{\mathrm{bd}}$ is the backdoor training loss, $\mathcal{L}_{\mathrm{spec}}$ is the spectral mimicry term, and λ_{atk} balances the two; a smaller value puts more weight on mimicry, and at 1 the attack reduces to the non-adaptive BadNets.

As Table 2 shows, TIMBRE keeps the final ASR after 100 rounds within 2.59% for all seven attack weights. We hypothesize that when λ_{atk} is large, the mimicry term constrains the spectral distribution of the update only weakly, so the spectral evidence can still identify these updates; when λ_{atk} drops to 0.1, the mimicry term dominates and the backdoor updates pass the spectral scoring

[1] Under the common server learning rate of 0.5, LGA implants the backdoor slowly: even with the evaluation window extended to 150 rounds, its final ASR without defense is only 28.68% (Table D5). To show the attack at its original strength, FedAvg uses the server learning rate of 1 from the LGA source code, while all defenses keep 0.5.

Table 2: Final ASR of PSM-BadNets after 100 rounds, CIFAR-10 × ResNet18, $\alpha = 0.5$, $f = 2$.

λ_{atk}	0.1	0.2	0.3	0.4	0.5	0.9	1.0
ASR↓ (%)	2.59	1.67	0.58	1.20	1.39	1.08	0.62

for a period of training and produce a higher ASR, which then falls back as the debt-triggered response takes effect. Overall, under all attack weights considered, spectral mimicry fails to sustain the backdoor after 100 rounds.

Other results. Appendices D.4 and D.5 cover intermittent attacks and no-attack controls.

6 Related Work

Backdoor attacks. BadNets (Gu et al., 2019) implants a backdoor in centralized training with trigger-carrying samples; in federated learning, DBA (Xie et al., 2020) splits the trigger across clients and model replacement (Bagdasaryan et al., 2020) scales up the malicious update, while Neurotoxin (Zhang et al., 2022b) places the payload on parameters that change little. F3BA (Fang & Chen, 2023), CerP (Lyu et al., 2023), and 3DFed (Li et al., 2023) bring the update close to benign ones in statistics such as norm and direction; Batman (He et al., 2026) and LGA (Yang et al., 2025) extend this alignment to the dominant components of the parameter matrices and to per-layer norms.

Server-side defenses. Existing defenses fall into four categories by their criterion. Distance- and geometry-based defenses select or cluster updates by the distances between them (Nguyen et al., 2022; Rieger et al., 2022; Blanchard et al., 2017; Huang et al., 2023a); DnC (Shejwalkar & Houmansadr, 2021) and MARS (Wan et al., 2025) also use singular value decompositions, the former removing outliers along the principal singular vector of the stacked updates and the latter making a hard two-cluster selection on the largest singular value of each neuron, but neither reads how energy is allocated within a single update. Direction- and sign-based defenses check directional alignment and per-coordinate signs (Xu et al., 2025b; Ozdayi et al., 2021); our coordinate filtering reuses the idea of same-sign support but applies it only to updates selected by evidence. History-based defenses accumulate similarity or residuals across rounds to judge whether a client is an attacker (Fung et al., 2020; Zhang et al., 2022a); our accumulated score also spans rounds, but it decays, lifts restrictions as the evidence fades, and issues no client-level verdict. Defenses based on norm constraints and aggregation operators constrain all clients (Yin et al., 2018; Sun et al., 2019), and our aggregation stage reuses their median center, fixed-proportion trimming, and norm clipping. Defenses that require server-held data or changes to client training (Cao et al., 2021; Li & Dai, 2024; Huang et al., 2023b) lie outside our setting. Appendix E discusses each category in detail.

7 Conclusion and Limitations

We proposed TIMBRE, a server-side backdoor defense for data-heterogeneous federated learning. It uses the layer-wise normalized spectral energy distribution of updates as evidence, combines current-round anomaly with cross-round accumulation, and suppresses suspicious updates through coordinate processing and tiered down-weighting. The experimental results show that TIMBRE balances backdoor suppression and main-task performance under the five non-adaptive backdoor attacks tested, a conclusion supported by the evaluation across datasets and models and by the comparison with eight existing defenses. Across a range of data heterogeneity and malicious participation settings, the method keeps the attack success rate low with the same set of defense parameters, indicating its adaptability to these levels of data heterogeneity and malicious participation.

However, the reference statistics of TIMBRE rest on the assumption that benign clients form the majority in every round, and its robustness beyond this setting remains to be studied. In addition, the current implementation relies on the visibility of individual client updates, so combining it with standard secure aggregation (Bonawitz et al., 2017) still requires a corresponding privacy-preserving computation mechanism. Future work will explore reference constructions that depend less on the benign majority and extend the applicability to privacy-preserving settings.

AI USE STATEMENT

In this work, we used generative AI tools (Claude Code and OpenAI Codex) for the following tasks with required disclosure: providing feedback on the research idea, the methodology, and the experimental design; implementing the method and the experiment scripts, running the experiments, and compiling the result tables from the run records, under the specification written by the authors; assisting in developing the theoretical analysis in Appendix A, including formulating and proving its mathematical claims; assisting in rewriting the manuscript into English; and drafting descriptions of the results from the result tables. Additionally, we used generative AI tools for tasks with recommended disclosure: drafting and editing parts of the paper, retrieving and screening related work, creating the figures and the scripts that produce them, and preparing the LaTeX source. We have reviewed all AI-assisted work: the research question, the method, and the claims were decided by the authors; the code was reviewed by the authors; every reported number was checked against the run records; the proofs were checked step by step by the authors; all citations were verified against the original publications; and all AI-drafted text was checked and revised by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Defense processing may affect the updates of benign clients and main-task performance; the accuracy cost measured in the no-attack controls is reported in Appendix D.5. The method does not permanently remove clients from training, but accumulated history may prolong restrictions, and later evidence may trigger them again.

REPRODUCIBILITY STATEMENT

Appendix A gives the assumptions and complete proofs of the theoretical results, and Appendix B gives Algorithm 1 and the method details. Appendix C describes the data partitions, the attack and baseline configurations, the parameter values of TIMBRE, and how its thresholds are set. All datasets are public, and the implementation is built on the open-source BackFed framework (Dao et al., 2026). We will release our code upon publication.

REFERENCES

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948. PMLR, 2020.

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Kallista A. Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.

Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. FLTrust: Byzantine-robust federated learning via trust bootstrapping. In *Network and Distributed System Security Symposium*, 2021.

Thinh Dao, Thuy Dung Nguyen, Khoa Doan, and Kok-Seng Wong. BackFed: A standardized and efficient benchmark framework for backdoor attacks in federated learning. In *ICLR 2026 Workshop on Principled Design for Trustworthy AI*, 2026.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Pei Fang and Jinghui Chen. On the vulnerability of backdoor defenses for federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11800–11808, 2023.

Hossein Fereidooni, Alessandro Pegoraro, Phillip Rieger, Alexandra Dmitrienko, and Ahmad-Reza Sadeghi. FreqFed: A frequency analysis-based approach for mitigating poisoning attacks in federated learning. In *Network and Distributed System Security Symposium*, 2024.

Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. The limitations of federated learning in Sybil settings. In *International Symposium on Research in Attacks, Intrusions and Defenses*, pp. 301–316, 2020.

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

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*, pp. 770–778, 2016.

Wenwen He, Wenke Huang, Bin Yang, Shukan Liu, and Mang Ye. SPMC: Self-purifying federated backdoor defense via margin contribution. In *International Conference on Machine Learning*, pp. 22422–22433. PMLR, 2025.

Wenwen He, Wenke Huang, Yiyang Fang, Wenjie Qu, Jiaheng Zhang, and Mang Ye. Batman: Benign knowledge alignment through malicious null space in federated backdoor attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Siquan Huang, Yijiang Li, Chong Chen, Leyu Shi, and Ying Gao. Multi-Metrics adaptively identifies backdoors in federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4652–4662, 2023a.

Tiansheng Huang, Sihao Hu, Ka-Ho Chow, Fatih Ilhan, Selim Tekin, and Ling Liu. Lockdown: Backdoor defense for federated learning with isolated subspace training. In *Advances in Neural Information Processing Systems*, volume 36, 2023b.

Wenke Huang, Mang Ye, Zekun Shi, Bo Du, and Dacheng Tao. Fisher calibration for backdoor-robust heterogeneous federated learning. In *European Conference on Computer Vision*, pp. 247–265. Springer, 2024.

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

Haoyang Li, Qingqing Ye, Haibo Hu, Jin Li, Leixia Wang, Chengfang Fang, and Jie Shi. 3DFed: Adaptive and extensible framework for covert backdoor attack in federated learning. In *IEEE Symposium on Security and Privacy*, pp. 1893–1907, 2023.

Songze Li and Yanbo Dai. BackdoorIndicator: Leveraging OOD data for proactive backdoor detection in federated learning. In *USENIX Security Symposium*, pp. 4193–4210, 2024.

Xiaoting Lyu, Yufei Han, Wei Wang, Jingkai Liu, Bin Wang, Jiqiang Liu, and Xiangliang Zhang. Poisoning with Cerberus: Stealthy and colluded backdoor attack against federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9020–9028, 2023.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, 2011.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, and Thomas Schneider. FLAME: Taming backdoors in federated learning. In *USENIX Security Symposium*, pp. 1415–1432, 2022.

Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R. Gel. Defending against backdoors in federated learning with robust learning rate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9268–9276, 2021.

Ashwinee Panda, Saeed Mahloujifar, Arjun Nitin Bhagoji, Supriyo Chakraborty, and Prateek Mittal. SparseFed: Mitigating model poisoning attacks in federated learning with sparsification. In *International Conference on Artificial Intelligence and Statistics*, pp. 7587–7624. PMLR, 2022.

Phillip Rieger, Thien Duc Nguyen, Markus Miettinen, and Ahmad-Reza Sadeghi. DeepSight: Mitigating backdoor attacks in federated learning through deep model inspection. In *Network and Distributed System Security Symposium*, 2022.

Virat Shejwalkar and Amir Houmansadr. Manipulating the Byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *Network and Distributed System Security Symposium*, 2021.

Shiqi Shen, Shruti Tople, and Prateek Saxena. AUROR: Defending against poisoning attacks in collaborative deep learning systems. In *Annual Computer Security Applications Conference*, pp. 508–519, 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Yifan Sui, Yongqi Sun, Naiyue Chen, Yingying Zhao, Hongbo Cao, and Baomin Xu. Backdoor detection in federated learning with feature map: A multi-task learning perspective. *IEEE Transactions on Information Forensics and Security*, 21:1142–1154, 2026.

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

Zhou Tan, De Li, Yirui Huang, Duanshu Fang, Jia-Li Yin, Xiaolei Liu, Songze Li, and Shouling Ji. DoBlock: Blocking malicious association propagation for backdoor-robust federated learning under domain skew. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25796–25804, 2026.

Wei Wan, Yuxuan Ning, Zhicong Huang, Cheng Hong, Shengshan Hu, Ziqi Zhou, Yechao Zhang, Tianqing Zhu, Wanlei Zhou, and Leo Yu Zhang. MARS: A malignity-aware backdoor defense in federated learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. DBA: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2020.

Jiahao Xu, Zikai Zhang, and Rui Hu. Achieving Byzantine-resilient federated learning via layer-adaptive sparsified model aggregation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025a.

Jiahao Xu, Zikai Zhang, and Rui Hu. Detecting backdoor attacks in federated learning via direction alignment inspection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20654–20664, 2025b.

Qingqian Yang, Peishen Yan, Xiaoyu Wu, Jiaru Zhang, Tao Song, Yang Hua, Hao Wang, Liangliang Wang, and Haibing Guan. Stealthy backdoor attack in federated learning via adaptive layer-wise gradient alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 29163–29172, 2025.

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

Hao Yu, Chuan Ma, Meng Liu, Tianyu Du, Ming Ding, Tao Xiang, Shouling Ji, and Xinwang Liu. G^2uardFL: Safeguarding federated learning against backdoor attacks via attributed client graph clustering. *IEEE Transactions on Information Forensics and Security*, 21:516–531, 2026.

Jidong Yuan, Qihang Zhang, Naiyue Chen, Shengbo Chen, and Baomin Xu. A multi-granularity clustering approach for federated backdoor defense with the Adam optimizer. In *International Joint Conference on Artificial Intelligence*, pp. 6931–6939, 2025.

Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2545–2555, 2022a.

Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael Mahoney, Prateek Mittal, Kannan Ramchandran, and Joseph Gonzalez. Neurotoxin: Durable backdoors in federated learning. In *International Conference on Machine Learning*, pp. 26429–26446. PMLR, 2022b.

A Theoretical Analysis of the Spectral Score

This appendix analyzes the normalized spectral energy distribution defined in Section 4.2. We first relate it to the low-rank approximation of the update matrix and then give a lower bound on its distance to a reference spectrum.

To simplify notation, we omit the client, round, and layer indices. Let $E \neq 0$ be the update matrix of a layer, with its full singular values in non-increasing order $\sigma_1, \ldots, \sigma_d$, where d is the smaller of the numbers of rows and columns. Define

$$p_j = \frac{\sigma_j^2}{\|E\|_F^2}, \qquad p = (p_1, \ldots, p_d),$$

and let $\pi = (\pi_1, \ldots, \pi_d)$ be a reference spectrum of the same dimension with $\pi_j \geq 0$ and $\sum_j \pi_j = 1$. The spectral distance follows the definition in the main text,

$$\mathcal{H}(p, \pi) = \frac{1}{\sqrt{2}} \|\sqrt{p} - \sqrt{\pi}\|_2.$$

Lemma A.1 (normalized spectrum and optimal low-rank approximation). For any integer $0 \leq m \leq d$, define the optimal relative rank-m approximation error of the update matrix as

$$J_m(E) = \min_{\mathrm{rank}(X) \leq m} \frac{\|E - X\|_F^2}{\|E\|_F^2}.$$

Then

$$J_m(E) = 1 - \sum_{j=1}^{m} p_j. \tag{A.1}$$

Proof. By the Eckart–Young theorem on optimal low-rank approximation (Eckart & Young, 1936), the sum E_m of the first m singular components of E attains the minimum above, and

$$\min_{\mathrm{rank}(X) \leq m} \|E - X\|_F^2 = \|E - E_m\|_F^2 = \sum_{j=m+1}^{d} \sigma_j^2.$$

Dividing both sides by $\|E\|_F^2 = \sum_{j=1}^{d} \sigma_j^2$ gives (A.1). □

Lemma A.1 shows that the cumulative energy share of the first m spectral components equals the fraction of matrix energy retained by the optimal rank-m approximation. Further, since $p_j = J_{j-1}(E) - J_j(E)$, the full normalized spectrum corresponds one to one to the optimal relative approximation errors at all ranks. The spectral energy distribution therefore describes how concentrated the energy of the update matrix is, rather than only its overall magnitude.

Theorem A.2 (distance lower bound from differences in spectral energy). For any integer $1 \leq m < d$, let the cumulative energy shares of the first m components of the update spectrum and of the reference spectrum be

$$a_m = \sum_{j=1}^{m} p_j, \qquad b_m = \sum_{j=1}^{m} \pi_j.$$

Then

$$\mathcal{H}^2(p, \pi) \geq 1 - \sqrt{a_m b_m} - \sqrt{(1 - a_m)(1 - b_m)}. \tag{A.2}$$

In particular,

$$\mathcal{H}(p, \pi) \geq \frac{|a_m - b_m|}{\sqrt{2}}. \tag{A.3}$$

Proof. By the normalization of p and π,

$$\mathcal{H}^2(p, \pi) = 1 - \sum_{j=1}^{d} \sqrt{p_j \pi_j}.$$

Applying the Cauchy–Schwarz inequality separately to the first m terms and to the remaining terms gives

$$\sum_{j=1}^{m} \sqrt{p_j \pi_j} \le \sqrt{a_m b_m}, \qquad \sum_{j=m+1}^{d} \sqrt{p_j \pi_j} \le \sqrt{(1-a_m)(1-b_m)}.$$

Substituting into the identity above yields (A.2). On the other hand, since $\sum_j (p_j - \pi_j) = 0$,

$$|a_m - b_m| \le \frac{1}{2} \sum_{j=1}^{d} |p_j - \pi_j|.$$

Applying the Cauchy–Schwarz inequality again,

$$\begin{aligned} \frac{1}{2} \sum_j |p_j - \pi_j| &= \frac{1}{2} \sum_j |\sqrt{p_j} - \sqrt{\pi_j}|(\sqrt{p_j} + \sqrt{\pi_j}) \\ &\le \frac{1}{2} \|\sqrt{p} - \sqrt{\pi}\|_2 \|\sqrt{p} + \sqrt{\pi}\|_2 \\ &\le \sqrt{2}\, \mathcal{H}(p, \pi), \end{aligned}$$

where the last step uses $\|\sqrt{p} + \sqrt{\pi}\|_2 \le 2$. Hence (A.3) holds. □

Theorem A.2 states that whenever the cumulative energy share at some rank differs from that of the reference, the spectral distance has a corresponding positive lower bound. Combined with Lemma A.1, such a deviation corresponds to a difference in the low-rank approximation structure of the update matrix, and it can exist even when two updates have the same overall norm.

Corollary A.3 (spectral deviation of a nearly low-rank update from the reference). For some integer $1 \le m < d$, suppose the actual update of a layer can be written as $E = L + R$ with

$$\operatorname{rank}(L) \le m, \qquad \|R\|_F \le \varepsilon \|E\|_F, \qquad 0 \le \varepsilon < 1.$$

If the reference spectrum satisfies

$$\sum_{j=1}^{m} \pi_j \le \beta < 1 - \varepsilon^2, \qquad \beta \ge 0,$$

then

$$\mathcal{H}(p, \pi) \ge \frac{1 - \varepsilon^2 - \beta}{\sqrt{2}} > 0. \tag{A.4}$$

Proof. L is a feasible solution of the rank-m approximation problem in Lemma A.1, so

$$J_m(E) \le \frac{\|E - L\|_F^2}{\|E\|_F^2} = \frac{\|R\|_F^2}{\|E\|_F^2} \le \varepsilon^2.$$

By Lemma A.1, $a_m \ge 1 - \varepsilon^2$; together with $b_m \le \beta$, this gives $a_m - b_m \ge 1 - \varepsilon^2 - \beta > 0$. Substituting into Theorem A.2 completes the proof. □

Corollary A.3 gives a sufficient condition for the motivation of this work: when the energy of an actually uploaded update is concentrated mainly in a few spectral components, and this concentration exceeds that of the reference, comparing normalized spectra can read out the corresponding structural difference. Here m serves only the analysis; the method itself still reads the full spectrum as defined in the main text.

B Method Details

B.1 Matrix Representation and Coordinate Processing

Spectral analysis uses the weight updates selected in advance for each architecture. For the image models, these are the weights of all convolutional layers with kernels of at least 3×3: 17 for ResNet18, excluding the 1×1 shortcut convolutions and the fully connected classifier, and 10 for VGG11. A two-dimensional weight update is used directly as a matrix, and a convolutional weight

update is unfolded into a $C_{\text{out}} \times (C_{\text{in}} k_h k_w)$ matrix, where C_{in} and C_{out} are the numbers of input and output channels and k_h and k_w are the height and width of the kernel. For each resulting non-zero matrix, the normalized spectral energy distribution is computed as defined in the main text.

The initial targets of coordinate processing are the strong-tier clients and the instantaneous coordinate targets of Section 4.3. If the total number of targets is smaller than N_{T}, the remaining participants of the round whose updated debt has reached H_C are added in decreasing order of debt, until this cap is reached or no candidate remains. N_{T} only limits the top-up of history candidates and does not truncate the initial target set. The participants of the round outside the final target set form the reference set $\mathcal{P}_t$, whose size is denoted n_t.

Norm clipping and coordinate processing cover all aggregable floating-point tensors. Let ρ_{coord} denote the active-coordinate ratio: for the clipped update of a reference client $i \in \mathcal{P}_t$, within each tensor of n_b coordinates we select the $\lceil \rho_{\text{coord}} n_b \rceil$ coordinates with the largest absolute values and then exclude those whose absolute value does not exceed a numerical threshold. The coordinates selected across tensors form the active set $\mathcal{K}_i^t$ of the main text. The numerical threshold is the largest absolute value in the tensor multiplied by 32 times the float32 machine epsilon.

The clipping interval is determined by the norm-clipped values of all reference clients. Let γ be the trimming ratio per side, so that $h_t = \lceil \gamma n_t \rceil$ extreme values are removed on each side. When $n_t \geq \max(q, 2h_t + 1)$, for any coordinate j we sort the reference values as $x_{(1)} \leq \cdots \leq x_{(n_t)}$ and set the interval endpoints to

$$\ell_j^t = \min(0, x_{(h_t+1)}), \qquad u_j^t = \max(0, x_{(n_t-h_t)}).$$

When there are too few reference clients, coordinate processing is skipped for the round and the norm-clipped updates are kept. No per-client normalization is applied after coordinate processing.

B.2 One Round of Aggregation

Algorithm 1: One round of TIMBRE aggregation.

Input: global model w^t; participant set $\mathcal{S}_t$ and its raw updates $\{\Delta_k^t\}$; debts D^{t-1}; non-negative base weights $\{a_k^t\}$; server learning rate η_t; fixed parameters (Tables C2 and C3). Debts are initialized to zero when the defense is enabled.

Output: w^{t+1} and D^t.

1. $z_k^t \leftarrow$ current-round evidence (Section 4.2) for all $k \in \mathcal{S}_t$ ▷ computed from the raw updates
2. $\overline{\Delta}_k^t \leftarrow$ norm-clipped update (Section 4.4) for all k
3. $D^t \leftarrow D^{t-1}$; $\omega_k^t \leftarrow 1$; $\widetilde{\Delta}_k^t \leftarrow \overline{\Delta}_k^t$
4. **if** the evidence of the round is complete **then**
5. update the debts D_k^t by Section 4.3 and obtain the strong tier $\mathcal{B}_t$ and the instantaneous coordinate targets $\mathcal{V}_t^+$
6. $\mathcal{T}_t^0 \leftarrow \mathcal{B}_t \cup \mathcal{V}_t^+$; $\mathcal{C}_t \leftarrow \{k \in \mathcal{S}_t \setminus \mathcal{T}_t^0 : D_k^t \geq H_C\}$
7. $\mathcal{T}_t \leftarrow \mathcal{T}_t^0 \cup$ at most $\max(0, N_{\text{T}} - |\mathcal{T}_t^0|)$ clients from $\mathcal{C}_t$ in decreasing order of debt ▷ history top-up
8. $\mathcal{P}_t \leftarrow \mathcal{S}_t \setminus \mathcal{T}_t$; $n_t \leftarrow |\mathcal{P}_t|$; $h_t \leftarrow \lceil \gamma n_t \rceil$
9. $\omega_k^t \leftarrow m_s$ for $k \in \mathcal{B}_t$; $\omega_k^t \leftarrow m_c$ for $k \in \mathcal{T}_t \setminus \mathcal{B}_t$
10. **if** $n_t \geq \max(q, 2h_t + 1)$ **then** ▷ otherwise skip coordinate processing
11. build the active coordinates $\mathcal{K}_i^t$ and the intervals $[\ell_j^t, u_j^t]$ by Appendix B.1; compute the support counts $V_{k,j}^t$ by Section 4.4
12. **for** $k \in \mathcal{T}_t$ **do** $\widetilde{\Delta}_k^t[j] \leftarrow \text{clip}(\overline{\Delta}_k^t[j], \ell_j^t, u_j^t)$ if $V_{k,j}^t \geq q$, otherwise 0
13. **end if**
14. **end if** ▷ debts and weights are left unchanged when the evidence is incomplete
15. $\tilde{a}_k^t \leftarrow a_k^t \omega_k^t / \sum_{i \in \mathcal{S}_t} a_i^t \omega_i^t$
16. $w^{t+1} \leftarrow w^t + \eta_t \sum_{k \in \mathcal{S}_t} \tilde{a}_k^t \widetilde{\Delta}_k^t$; **return** w^{t+1}, D^t

The evidence of a round is complete in line 4 when the current-round evidence z_k^t can be computed for all participants, that is, when the MAD in its denominator is nonzero; otherwise the round updates no debt, skips coordinate processing, and aggregates all norm-clipped updates with their base weights.

C Experimental Setup

C.1 Data and Training

Each group of experiments has 100 clients, 10 of which participate in each round. Data are partitioned with a sample-balanced Dirichlet distribution, with $\alpha = 0.5$ by default. Benign clients train with SGD for 2 local epochs with learning rate 0.1, momentum 0.9, weight decay 5×10^{-4}, and batch size 64. The server learning rate is 0.5 by default; in the CIFAR-10 $\times$ ResNet18 experiments of Table 1, FedAvg under LGA uses 1 and all other defenses use 0.5. Each setting continues training from the clean base model listed in Table C1. The ResNet18 on CIFAR-10 and CIFAR-100 is the CIFAR variant with a 3×3 first convolution and a base width of 32. On Tiny-ImageNet we use the standard torchvision ResNet18 with a 7×7, stride-2 first convolution and a base width of 64. VGG11 follows the BackFed implementation, with 10 convolutional layers with batch normalization and a three-layer fully connected classifier.

Table C1: Data and model configuration of each group of experiments. The first three groups use clean base models pretrained by us; Tiny-ImageNet uses the base model provided by BackFed (Dao et al., 2026), which was not trained to convergence under the settings of our experiments.

	CIFAR-10 × ResNet18	CIFAR-10 × VGG11	CIFAR-100 × ResNet18	Tiny-ImageNet × ResNet18
Classes	10	10	100	200
Samples per client	500	500	500	1000
Base round	2000	2000	1900	2500
Matrices for spectral analysis	17	10	17	17

The heterogeneity sweep continues training from the same $\alpha = 0.5$ base model and changes only the subsequent client distributions. Unless otherwise noted, each condition is run once with seed 5101.

C.2 Evaluation Metrics

ACC is the classification accuracy on clean test samples. ASR is the fraction of test samples whose original label differs from the target class that are predicted as the target class after the trigger is added. We report the ACC and ASR at the end of the experiment. We also run no-attack controls that compare the ACC with and without the defense under the same settings and number of training rounds, to assess the effect of the defense itself on main-task performance.

C.3 Attack and Baseline Configurations

Attack settings. All five attacks use target class 2. Ten malicious client identities are fixed in advance, and 2 of them are selected per round by default. LGA follows the default configuration of the authors' implementation, with 1 attacker selected in each poisoning round; because the attack uses the global update of the previous round as its reference, the first round performs normal training only. The malicious-participation sweep changes the number of malicious participants per round separately. Attackers train locally for 6 epochs with a poisoning ratio of 0.3125 and a learning rate of 0.05; for LGA, the poisoning ratio and the learning rate are 1 and 0.1, respectively. BadNets, Batman, and LGA use a square trigger in the bottom-right corner, of size 5×5 on CIFAR and 10×10 on Tiny-ImageNet; Neurotoxin uses a blended trigger with weight 0.2 and a gradient mask ratio of 0.99. DBA uses four shards, which are merged at evaluation; on Tiny-ImageNet the shards are scaled so that the relative trigger area matches CIFAR.

Unless otherwise stated, BadNets, Neurotoxin, Batman, and DBA run for 50 rounds. Because LGA was weak under the short window in preliminary experiments, we extend its evaluation window to 150 rounds. Under the same attack, all defenses use the same evaluation window.

Baseline settings. The defenses use the following configurations; the comparison of supplementary variants is given in Appendix D.1.

Multi-Krum. Selects clients by the distance between an update and its nearest neighbors; in our experiments 8 updates are kept per round and the number of malicious clients is set to the true value of 2, which gives Multi-Krum prior knowledge of the actual number of attackers.

FLAME. Clusters the uploaded models (Nguyen et al., 2022) and processes the retained updates by clipping and noise addition, with the noise coefficient set to 0.001; the supplementary variant FLAME-update clusters client updates instead, to examine the effect of the clustering object.

DeepSight. Clusters and filters updates by model statistics; the clustering configuration was chosen by the false-rejection rate without attack, using `min_samples=3` and allowing a single cluster, with the within-cluster suspicious-fraction threshold set to 0.3333.

Multi-Metrics. Filters clients by a combination of several update statistics; the main comparison keeps the retention ratio 0.3 of the authors' implementation (Huang et al., 2023a), and we add the settings 0.6 and 0.8 to examine the effect of the retention ratio.

AlignIns. Filters clients by the directional alignment of updates and the sign consistency of important coordinates; the coordinate ratio used for the sign test is 0.3, and the thresholds of the two standardized anomaly scores are both 1. The main comparison uses the full model-state detection scope of the authors' code, including the running means and variances of BN, while the supplementary variant AlignIns-trainable detects and clips only the trainable parameters.

MARS. Builds clustering features from the backdoor energy of neurons, taking the 5% of neurons with the highest energy in each layer, with the inter-cluster distance threshold set to 0.03.

RLR. Adjusts the server learning rate by the sign consistency of per-coordinate updates, with the sign-vote threshold set to 4.

FoolsGold. Lowers the aggregation weights of highly similar updates according to the cosine similarity of clients' historical updates, with the confidence parameter set to 1.

C.4 TIMBRE Parameter Settings

For each combination of dataset and model, we record client scores in a separate 50-round no-attack run at $\alpha = 0.1$ and determine the thresholds from them. The trigger rates of instantaneous strong processing and instantaneous coordinate processing on this run are limited to at most 5% and 10%, respectively. The history thresholds are determined by replaying the debt update rule: H_S is set first so that the trigger rate of the history-triggered strong tier does not exceed 5%, and H_C is then set so that the fraction whose debt reaches H_C does not exceed 50%. The resulting thresholds are used for all attack and distribution conditions of that setting; the values are listed in Table C2. The remaining shared parameters are listed in Table C3.

Table C2: TIMBRE thresholds for each dataset and model, rounded to two decimals.

Dataset × model	z_{hi}	z_{inst}	H_C	H_S
CIFAR-10 × ResNet18	4.63	2.91	0.60	5.22
CIFAR-10 × VGG11	4.53	2.73	0.60	5.74
CIFAR-100 × ResNet18	4.80	2.47	0.48	6.47
Tiny-ImageNet × ResNet18	5.54	2.73	0.80	6.63

Table C3: TIMBRE policy parameters shared across the image datasets and models.

Stage	Parameter	Meaning	Value
Layer-wise spectral score (Section 4.2)	r	Number of highest matrix scores aggregated	8
Strength- and persistence-based response (Section 4.3)	λ	Sign boundary of the debt increment	0
Target selection (Appendix B.1)	N_{T}	Target-count cap for history top-up	7
Update processing (Section 4.4)	κ	Norm-cap multiplier	1.68
	γ	Trimming ratio per side	0.1
	ρ_{coord}	Active-coordinate ratio	0.4
	q	Support threshold	3
	m_s	Strong-tier weight multiplier	0.01
	m_c	Coordinate-tier weight multiplier	0.5

The debt increase and decrease caps are $c_+ = z_{\mathrm{hi}} - z_{\mathrm{inst}}$ and $c_- = c_+/2$; the median reference of norm clipping is floored at 10^{-12}.

C.5 Setup of Figure 1

Figure 1 uses BadNets, $\alpha = 0.5$, and seed 5101, and analyzes the 500 client updates of layer2.0.conv1 in rounds 2001–2050, of which 400 are benign and 100 are malicious. Panel (c) pairs each malicious update with the benign update of the same round whose norm is closest and within $\pm 10\%$, giving 89 pairs; the 11 unmatched updates are excluded, and the first-component energy shares of the two members are compared. If the Hellinger distances to the reference spectra of the same round are compared instead, the malicious member has the larger distance in 87 pairs.

D Additional Experiments

The following experiments follow the training and evaluation settings of Appendix C; each section states only the changes relative to the default settings.

D.1 Results across Datasets and Models

To examine the effect of architecture and dataset, we compare TIMBRE, FedAvg, and the eight defenses of the main text on CIFAR-10 $\times$ VGG11 and CIFAR-100 $\times$ ResNet18 (Tables D1 and D2). Both groups use the default $\alpha = 0.5$; the thresholds of TIMBRE are determined per dataset and model, and the shared parameters are given in Appendix C.4.

Table D1: Final ACC and ASR (%) on CIFAR-10 $\times$ VGG11. Bold marks the best result among the nine defenses and underline the second best; FedAvg is the no-defense reference.

Defense	BadNets		Batman		Neurotoxin		DBA		LGA	
	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
FedAvg	91.36	92.27	91.15	96.49	91.45	91.38	90.18	35.76	91.22	60.84
Multi-Krum	<u>91.13</u>	93.88	90.91	97.13	<u>90.92</u>	90.16	89.89	30.48	89.70	84.02
FLAME	68.89	68.32	77.68	92.26	70.35	68.41	69.51	36.66	74.80	**0.53**
DeepSight	90.95	92.29	<u>91.11</u>	96.36	90.88	88.16	88.44	38.06	<u>91.26</u>	66.06
Multi-Metrics	85.83	88.06	87.34	29.60	84.59	66.22	86.86	47.06	84.36	3.42
AlignIns	81.17	27.04	89.55	<u>2.59</u>	87.62	<u>2.57</u>	89.74	<u>19.82</u>	89.58	1.57
MARS	90.67	<u>25.23</u>	91.08	96.12	88.43	99.74	**91.30**	20.74	89.87	80.28
RLR	90.69	93.46	90.67	16.39	65.63	91.30	89.70	33.23	89.94	96.67
FoolsGold	**91.22**	92.77	**91.43**	96.66	**90.98**	15.93	89.92	35.30	**91.65**	64.59
TIMBRE	89.72	**1.37**	90.68	**2.07**	90.34	**1.72**	<u>90.16</u>	**3.60**	90.51	<u>1.02</u>

Table D2: Final ACC and ASR (%) on CIFAR-100 × ResNet18. Bold marks the best result among the nine defenses and underline the second best; FedAvg is the no-defense reference. In Tables D1 and D2, LGA runs for 150 rounds and the other attacks for 50 rounds. On VGG11, LGA is read at round 2150 and the other attacks at round 2050; the corresponding rounds on CIFAR-100 are 2050 and 1950.

Defense	BadNets		Batman		Neurotoxin		DBA		LGA	
	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
FedAvg	70.32	85.16	69.97	90.18	69.41	92.15	69.46	61.23	69.80	50.38
Multi-Krum	69.66	86.27	68.79	0.26	69.00	94.43	68.99	77.31	67.99	67.49
FLAME	51.49	79.83	53.32	0.45	51.45	90.12	50.75	88.07	55.89	0.29
DeepSight	**70.35**	86.63	69.54	88.63	68.88	94.04	68.97	71.53	**69.50**	61.02
Multi-Metrics	62.87	85.43	62.91	78.42	63.66	93.09	62.97	51.31	64.11	71.06
AlignIns	65.77	0.59	64.92	1.67	65.41	**0.77**	65.92	34.93	67.76	1.47
MARS	68.22	1.71	68.69	89.58	68.77	8.74	68.79	4.47	65.66	91.58
RLR	69.53	81.97	68.43	**0.00**	65.64	99.22	68.45	**0.00**	68.10	95.89
FoolsGold	69.93	86.83	**69.71**	90.33	**69.96**	92.92	**69.77**	68.85	69.23	50.73
TIMBRE	67.32	**0.31**	67.33	0.71	68.52	1.44	67.61	0.45	67.64	**0.16**

Under these default settings, the ASR of TIMBRE ranges from 1.02% to 3.60% under the five attacks on VGG11 and from 0.16% to 1.44% on CIFAR-100, with ACC ranging from 89.72% to 90.68% and from 67.32% to 68.52%, respectively.

Tiny-ImageNet. Table D3 uses the upstream pretrained ResNet18 base model listed in Appendix C.1, with 1000 images per client. Under BadNets, Batman, and Neurotoxin, TIMBRE keeps the final ASR between 0.31% and 0.77%, whereas FedAvg ranges from 70.58% to 88.78%.

Table D3: Final ACC and ASR (%) on Tiny-ImageNet × ResNet18. LGA is read at round 2650 and the other attacks at round 2550. DBA uses a trigger scaled with the resolution.

Condition	FedAvg		TIMBRE	
	ACC↑	ASR↓	ACC↑	ASR↓
BadNets	46.23	71.39	44.22	0.66
Batman	46.52	70.58	44.07	0.77
Neurotoxin	46.54	88.78	43.44	0.31
DBA	46.16	11.30	43.48	0.54
LGA	47.68	6.80	45.04	0.32

Supplementary configurations and re-runs. Table D4 reports the results of the variants described in Appendix C.3 and the re-run of TIMBRE with another random seed.

Table D4: Supplementary final ACC and ASR (%) on CIFAR-10 × ResNet18, $\alpha = 0.5$. The last row uses seed 5102 and the others seed 5101; LGA is read at round 2150 and the other attacks at round 2050.

Defense configuration	BadNets		Batman		Neurotoxin		DBA		LGA	
	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
FLAME-update	86.70	85.58	85.28	89.34	84.67	80.31	84.48	61.94	83.63	94.81
Multi-Metrics p=0.6	90.26	0.56	90.15	10.16	89.85	3.13	89.30	31.78	90.80	53.04
Multi-Metrics p=0.8	90.23	66.81	91.32	15.54	91.59	73.20	89.53	57.76	91.10	42.34
AlignIns-trainable	86.00	1.18	88.50	3.27	89.63	4.20	86.59	4.21	89.71	1.13
TIMBRE (re-run)	90.58	0.60	90.51	0.79	89.50	0.32	89.58	0.50	91.30	0.67

D.2 Data Heterogeneity and Malicious Participation Ratio

Tables D5 and D6 give the complete final metrics of ResNet18 on CIFAR-10 and CIFAR-100 shown in Figure 3, all with seed 5101. In the heterogeneity sweep, BadNets, Batman, Neurotoxin, and DBA have 2 malicious clients per round and LGA has 1; the malicious-participation sweep fixes $\alpha = 0.5$

and increases the number of malicious clients among the 10 participants per round from 1 to 4. Both sweeps keep the defense parameters of each dataset and model combination unchanged. The malicious-participation sweep deliberately changes the default number of participating attackers and represents different conditions on the same experimental axis.

Table D5: Final ACC and ASR (%) under different data heterogeneity. On CIFAR-10, LGA is read at round 2150 and the other attacks at round 2050; on CIFAR-100, the corresponding rounds are 2050 and 1950.

CIFAR-10 × ResNet18.

Attack	Method	α=0.2		0.3		0.5		0.7		1.0	
		ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
BadNets	TIMBRE	89.62	0.79	89.00	0.30	91.51	1.20	91.47	1.32	90.22	0.37
BadNets	FedAvg	89.59	80.69	90.38	87.79	91.32	76.88	91.52	76.62	91.31	87.67
Batman	TIMBRE	89.48	0.42	89.98	0.72	91.01	0.69	91.10	1.32	90.90	0.40
Batman	FedAvg	89.09	89.24	89.85	94.09	91.53	89.03	91.54	85.22	91.32	93.68
Neurotoxin	TIMBRE	86.06	0.62	88.28	0.43	91.25	1.38	90.95	1.50	90.57	0.37
Neurotoxin	FedAvg	89.99	84.14	90.50	82.86	91.92	83.06	91.86	87.28	91.98	86.10
DBA	TIMBRE	89.27	0.62	89.76	0.49	90.99	0.51	91.12	0.88	91.51	0.99
DBA	FedAvg	88.56	53.26	89.76	59.82	90.80	56.19	90.76	56.36	91.20	66.21
LGA	TIMBRE	88.24	0.63	89.19	0.72	90.44	0.73	91.17	0.89	91.02	0.69
LGA	FedAvg	89.88	27.42	91.46	27.61	91.21	28.68	92.12	26.57	91.92	45.14

CIFAR-100 × ResNet18.

Attack	Method	α=0.2		0.3		0.5		0.7		1.0	
		ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
BadNets	TIMBRE	67.21	0.63	66.39	0.19	67.32	0.31	68.55	0.61	67.86	0.13
BadNets	FedAvg	68.87	72.03	68.76	81.25	70.32	85.16	69.65	80.27	70.36	83.43
Batman	TIMBRE	66.43	0.60	67.71	0.11	67.33	0.71	68.15	0.43	68.00	0.16
Batman	FedAvg	68.01	82.15	68.89	88.94	69.97	90.18	69.58	84.96	70.19	91.10
Neurotoxin	TIMBRE	67.07	1.32	67.04	0.57	68.52	1.44	69.20	1.20	68.20	0.32
Neurotoxin	FedAvg	69.21	90.79	68.84	89.55	69.41	92.15	70.32	91.33	69.94	91.08
DBA	TIMBRE	66.65	0.16	66.60	0.33	67.61	0.45	67.88	0.21	68.59	0.13
DBA	FedAvg	68.23	73.90	68.91	67.62	69.46	61.23	69.92	75.42	69.88	72.27
LGA	TIMBRE	66.23	0.82	67.33	0.36	67.64	0.16	67.70	0.38	68.39	0.24
LGA	FedAvg	69.35	46.57	69.90	54.01	69.80	50.38	69.81	65.65	70.32	43.76

Table D6: Final ACC and ASR (%) under different numbers of malicious participants. Reading rounds are as in Table D5.

CIFAR-10 × ResNet18.

Attack	Method	f=1		2		3		4	
		ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
BadNets	TIMBRE	90.90	0.50	91.51	1.20	89.08	0.38	90.11	0.84
BadNets	FedAvg	91.63	59.27	91.32	76.88	90.82	77.02	91.35	82.29
Batman	TIMBRE	91.44	1.53	91.01	0.69	90.25	0.47	90.66	2.23
Batman	FedAvg	91.50	68.08	91.53	89.03	90.46	89.82	90.91	91.08
Neurotoxin	TIMBRE	91.13	0.41	91.25	1.38	90.53	0.43	90.57	1.86
Neurotoxin	FedAvg	91.81	70.88	91.92	83.06	91.46	87.22	91.29	90.21
DBA	TIMBRE	91.14	0.76	90.99	0.51	90.23	0.56	90.56	0.67
DBA	FedAvg	91.48	48.61	90.80	56.19	90.18	59.80	90.39	77.73
LGA	TIMBRE	90.44	0.73	88.75	0.88	91.55	0.63	90.06	2.48
LGA	FedAvg	91.21	28.68	91.09	76.52	91.42	91.86	91.25	89.70

Table D6 (continued): CIFAR-100 × ResNet18.

Attack	Method	f=1		2		3		4	
		ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
BadNets	TIMBRE	67.69	1.01	67.32	0.31	67.81	0.23	67.88	0.31
BadNets	FedAvg	69.87	76.36	70.32	85.16	70.07	90.30	70.52	90.52
Batman	TIMBRE	67.86	0.64	67.33	0.71	68.10	0.32	68.19	0.22
Batman	FedAvg	70.18	86.02	69.97	90.18	69.29	92.95	69.74	94.78
Neurotoxin	TIMBRE	68.00	1.09	68.52	1.44	67.67	0.53	67.56	1.55
Neurotoxin	FedAvg	70.05	82.63	69.41	92.15	69.78	93.78	69.97	96.41
DBA	TIMBRE	67.85	0.23	67.61	0.45	67.75	0.29	68.51	0.26
DBA	FedAvg	69.43	59.36	69.46	61.23	69.71	85.56	69.99	79.62
LGA	TIMBRE	67.64	0.16	67.65	0.35	67.49	0.26	66.32	1.32
LGA	FedAvg	69.80	50.38	69.90	85.37	69.57	93.46	69.98	94.97

D.3 Component Ablation

At $\alpha = 0.2$ we examine the roles of layer-wise spectral analysis, evidence accumulation, coordinate processing, and reweighting; all variants keep norm clipping of all updates. We construct four variants: (1) **pooled spectrum**, which concatenates the 17 update matrices block-diagonally into one large matrix and computes a single spectral energy distribution and score, instead of scoring each layer and then aggregating; (2) **without evidence accumulation**, which no longer accumulates evidence across a client's participations and determines the processing targets and the response strength from the current evidence alone, with the subsequent update-processing rules unchanged; (3) **without coordinate processing**, which keeps target selection and tier-based weighting but turns off coordinate filtering and interval clipping; (4) **without reweighting**, which keeps the history-based decisions and coordinate processing but sets the defense weight multipliers of all clients to 1, leaving the base aggregation weights unchanged. Table D7 reports the final ACC and ASR of the four variants and the full method.

Table D7: Final ACC and ASR (%) of the four core ablations, CIFAR-10 × ResNet18, $\alpha = 0.2$. LGA is read at round 2150 and the other attacks at round 2050.

Variant	BadNets		Batman		Neurotoxin		DBA		LGA	
	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
Full method	89.62	0.79	89.48	0.42	86.06	0.62	89.27	0.62	88.24	0.63
Pooled spectrum	88.77	0.33	88.97	8.83	89.18	24.64	87.79	0.13	85.62	0.50
Without evidence accumulation	87.30	1.28	88.62	88.89	90.03	79.33	87.98	0.48	88.00	0.86
Without coordinate processing	87.59	0.56	86.48	61.22	89.12	30.64	87.56	0.81	90.09	1.02
Without reweighting	90.37	3.87	89.81	4.10	90.68	12.91	90.50	1.30	89.82	0.50

All four ablations raise the ASR under Neurotoxin and Batman, indicating that each of the four components helps suppress the backdoor under these two attacks. Removing reweighting raises the ACC under all five attacks but raises the ASR under BadNets, Batman, Neurotoxin, and DBA, reflecting the trade-off between main-task performance and backdoor suppression.

Other processing schemes. Table D8 further compares debt decay and different update-processing schemes. Without debt decay sets the negative increments of the accumulated score to zero, so that the score only increases. The other three variants keep the history-based decisions and target-selection rules of the full method and differ as follows:

- **Uniform strong down-weighting**: keeps coordinate processing and applies the weight multiplier $m_s = 0.01$ uniformly to all selected processing targets (both the coordinate tier and the strong tier), with multiplier 1 for the other clients.
- **Whole-update down-weighting**: turns off coordinate filtering and interval clipping and applies the weight multiplier $m_s = 0.01$ uniformly to the whole update of each target client, with multiplier 1 for the other clients.

- **Whole-update removal**: also turns off coordinate processing, sets the weight multiplier of target clients to 0 and that of the others to 1; if all participants of a round are targets, no removal is performed.

Table D8: Final ACC and ASR (%) of other processing schemes, seed 5101, with settings and windows as in Table D7.

Variant	BadNets		Batman		Neurotoxin		DBA		LGA	
	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
Full method	89.62	0.79	89.48	0.42	86.06	0.62	89.27	0.62	88.24	0.63
Without debt decay	87.86	0.29	88.48	0.20	87.88	0.77	89.29	0.43	86.83	0.52
Uniform strong down-weighting	85.69	0.32	86.00	0.14	87.86	0.64	87.61	0.56	86.59	0.27
Whole-update down-weighting	88.14	0.38	80.12	0.43	88.02	0.93	84.13	0.44	87.10	0.42
Whole-update removal	86.59	0.28	83.62	0.13	73.00	0.06	86.69	0.50	74.48	0.13

Compared with the full method, both without debt decay and uniform strong down-weighting lower the ASR under BadNets, Batman, and LGA but also lower the ACC; under Neurotoxin, both variants raise the ACC while the ASR rises slightly. The effect of whole-update down-weighting varies with the attack: under Batman the ASR is essentially unchanged (0.42% to 0.43%) while the ACC drops from 89.48% to 80.12%. Whole-update removal lowers the ASR under all five attacks relative to the full method, but its ACC is 2.58–13.76 percentage points below the full method, showing a larger loss of main-task performance.

D.4 Intermittent Attacks

On CIFAR-10 $\times$ ResNet18 with $\alpha = 0.5$ and $f = 2$, BadNets poisons once every 1, 2, 3, or 4 rounds and trains normally in the other rounds, with an observation window of 50 rounds (Table D9).

Table D9: Final ACC and ASR (%) under different attack frequencies. The actual numbers of attacking identities in the four settings are 10, 10, 9, and 7.

Attack frequency	Poisoning rounds	ACC	ASR
1/1	50	91.51	1.20
1/2	25	90.77	1.34
1/3	17	91.20	1.50
1/4	13	91.15	1.30

Under the four frequencies, the final ASR of TIMBRE stays between 1.20% and 1.50% and the ACC between 90.77% and 91.51%. Switching from continuous to intermittent poisoning changes neither the ASR nor the ACC substantially, indicating that TIMBRE still balances backdoor suppression and main-task performance at the attack frequencies tested.

D.5 Model Performance without Attack

Table D10 compares the effect of each defense on main-task performance without attack; all three settings use $\alpha = 0.5$, with FedAvg as the control without defense.

In the three settings, the ACC of TIMBRE is 1.35–2.29 percentage points below the no-defense baseline. Together with the attacked experiments above, these results show that TIMBRE suppresses backdoors while maintaining good main-task performance.

D.6 Text Classification Task

We use a four-layer LSTM (Hochreiter & Schmidhuber, 1997) on the IMDB (Maas et al., 2011) sentiment classification task to evaluate the defense of TIMBRE under the Word-trigger and Neurotoxin attacks. For the text model, spectral analysis uses the 9 weight matrices of the word-embedding layer and the four LSTM layers; because there are few matrices, the cross-matrix aggregation takes the 4

Table D10: Final ACC (%) without attack. CIFAR-10 × ResNet18 runs for 50 rounds and the other two groups for 150 rounds; the three columns are read at rounds 2050, 2150, and 2050, respectively.

Defense	CIFAR-10 / ResNet18	CIFAR-10 / VGG11	CIFAR-100 / ResNet18
FedAvg	92.06	91.84	70.18
Multi-Krum	91.76	90.41	69.31
FLAME	85.78	79.15	56.57
DeepSight	91.63	91.62	69.90
Multi-Metrics	86.86	85.83	64.32
AlignIns	89.66	89.36	68.02
MARS	89.94	91.17	64.95
RLR	91.28	90.10	68.98
FoolsGold	91.97	91.40	69.74
TIMBRE	90.61	90.49	67.89

highest matrix scores, so that the selected fraction is comparable to that of ResNet18 (8 of 17). The thresholds and the coordinate-tier multiplier of the text experiment are set separately. Both attacks insert a word trigger at the end of a review to steer the prediction toward the positive class. Table D11 shows that TIMBRE reduces the ASR of the two attacks from 100% to 14.10% and 16.08%, respectively, while keeping an ACC of 75.04% and 76.60%, which demonstrates its backdoor suppression under these two text attacks.

Table D11: ACC and ASR (%) on IMDB, all after 50 rounds of continued training. No-attack rows report ACC only. Without poisoning, appending the trigger already turns some negative reviews positive; across the evaluation points within 50 rounds of the two no-attack runs, this fraction fluctuates between 6.35% and 24.39%.

Condition	FedAvg		TIMBRE	
	ACC↑	ASR↓	ACC↑	ASR↓
Word-trigger	79.27	100.00	75.04	14.10
Neurotoxin	78.56	100.00	76.60	16.08
No attack	78.29	—	76.68	—

E Extended Related Work

This appendix is the full version of Section 6 and discusses the related attacks and defenses category by category.

E.1 Backdoor Attacks

BadNets (Gu et al., 2019) first implanted a backdoor in centralized training with samples carrying a trigger pattern. In the federated setting, an attacker can split the trigger across several clients (DBA (Xie et al., 2020)) and scale up its own update to counteract the dilution of aggregation (model replacement (Bagdasaryan et al., 2020)). Neurotoxin (Zhang et al., 2022b) places the payload on parameters that change little during training, so that later benign updates are less likely to overwrite it. Subsequent attacks target defense designs directly: F3BA (Fang & Chen, 2023) flips the signs of only a few parameters while keeping magnitudes normal, CerP (Lyu et al., 2023) jointly optimizes the trigger and the local model, and 3DFed (Li et al., 2023) adapts the backdoored model through an indicator mechanism; all three bring the update close to benign ones in statistics such as norm and direction. The most recent attacks push the alignment target to more structured quantities. Batman (He et al., 2026) applies a truncated singular value decomposition to the backdoor-related layers of the local parameter matrices and places the benign-alignment correction in the orthogonal complement of the dominant components, and LGA (Yang et al., 2025) takes the aggregated global update of the previous round as a layer-wise reference and compresses the per-layer norm of the malicious update to within that reference.

E.2 SERVER-SIDE DEFENSES

By the technique they use, existing server-side defenses fall broadly into four categories: defenses based on distance and geometric relations, on direction and signs, on cross-round history, and on norm constraints and aggregation operators.

Distance- and geometry-based defenses differ in their specific criteria (Fereidooni et al., 2024; Shen et al., 2016). FLAME (Nguyen et al., 2022) takes the majority cluster by whole-model cosine similarity, then clips at the median distance and adds noise; DeepSight (Rieger et al., 2022) labels each model with output-layer statistics and rejects whole clusters by the fraction of suspicious members; Multi-Krum (Blanchard et al., 2017) selects a subset of updates by nearest-neighbor distances, Multi-Metrics (Huang et al., 2023a) selects a fixed proportion by a whitened combination of several geometric measures, DnC (Shejwalkar & Houmansadr, 2021) applies a singular value decomposition to the stacked updates and removes outliers by their projection onto the principal singular vector, and MARS (Wan et al., 2025) uses the largest singular value of each neuron's weight matrix as a proxy for backdoor energy, concatenates the highest fraction of each layer into a vector, and makes a hard two-cluster selection.

Direction- and sign-based defenses read the direction of a single update and its per-coordinate signs (Yuan et al., 2025; Xu et al., 2025a). AlignIns (Xu et al., 2025b) checks the directional alignment between an update and the global model and the sign consistency of important parameters before filtering and clipping; RLR (Ozdayi et al., 2021) counts per-coordinate sign votes and flips the global learning rate of a coordinate when the majority sign is insufficient.

History-based defenses move the criterion to accumulation across rounds and use it to judge whether a client is an attacker (Yu et al., 2026). FoolsGold (Fung et al., 2020) continuously down-weights clients by their accumulated pairwise similarity, targeting collusive Sybils; FLDetector (Zhang et al., 2022a) detects long-term inconsistency with a rolling score of historical prediction residuals and removes the client.

Defenses based on norm constraints and aggregation operators do not judge individual clients; they change the aggregation operator itself or constrain all clients (Panda et al., 2022). Coordinate-wise median and trimmed mean (Yin et al., 2018) replace the aggregation operator directly. Norm clipping is likewise applied to all clients: early work used a fixed threshold (Sun et al., 2019), and FLAME, DeepSight, and AlignIns stack this layer on top of their own decisions with the threshold set to the median norm of the round's updates. The aggregation stage of our method reuses the median center, fixed-proportion trimming, and norm clipping from this line (Section 4.4).

Other defenses require the server to hold additional data or change client-side training, which lies outside our setting. FLTrust (Cao et al., 2021) builds a per-round reference update from clean root data held by the server for trust weighting; others require a clean validation set (Huang et al., 2024; Sui et al., 2026) or server-side out-of-distribution data for proactive intervention (Li & Dai, 2024), or rewrite client training and model structure (He et al., 2025; Tan et al., 2026; Huang et al., 2023b).