WHEN PRIORS VEIL THE THREAT: BACKDOOR DEFENSE VIA CHANNEL ACTIVATION INSPECTION FOR PRETRAINED FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

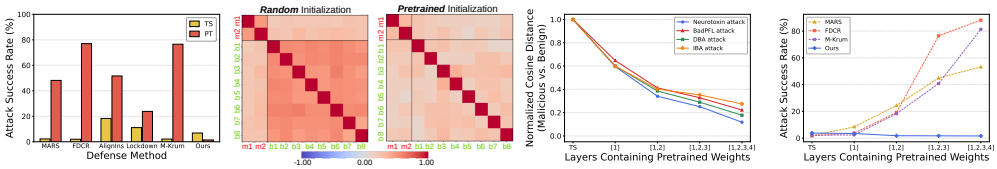
Existing defenses against backdoor attacks in federated learning aim to identify malicious clients and typically rely on the discrepancies among model gradient updates to distinguish them from benign ones. In this paper, we empirically show that when the global model is initialized with pretrained backbones, the updates similarity between benign and malicious clients can be substantially amplified, thereby rendering existing defenses ineffective. To bridge this gap, we investigate client model discrepancies from the perspective of channel activations and uncover a key observation: benign models generally exhibit *consistent* channel activation magnitudes across samples whereas backdoored models show abnormal activation variations. Motivated by this, we propose CAS (Channel Abnormality Score), a new malicious detection metric quantifying the abnormality within the channel activation pattern of a client model via the notion of term frequency-inverse document frequency. Specifically, CAS treats channel activations as words and the feature map of each class as a document. When a client contains channels that contribute abnormally to documents as measured by CAS, the server identifies the client model as backdoored and excludes it from global aggregation. Extensive experiments show that CAS significantly outperforms existing defenses under the pretrained federated learning setting.

1 INTRODUCTION

Federated learning has emerged as a dominant paradigm for training machine learning models over decentralized and privacy-sensitive data (McMahan et al., 2017; Wang et al., 2025; Liu et al., 2024). However, its collaborative training process is vulnerable to backdoor poisoning attacks, where malicious clients manipulate their local training data or objectives so that the aggregated global model behaves normally on clean inputs but misclassifies trigger injected inputs into an attacker-specified target class (Wang et al., 2020; Xie et al., 2019). To mitigate this threat, substantial research has focused on eliminating backdoors in federated learning (Xie et al., 2018; Yin et al., 2018; Blanchard et al., 2017; Pillutla et al., 2022; Huang et al., 2023; Xu et al., 2025; Huang et al., 2024).

Despite their effectiveness, we uncover a critical failure of these methods when the server initializes the global model with pretrained backbones. This vulnerability is particularly concerning because practical federated learning systems increasingly leverage high-quality pretrained backbones for global model initialization (Yue et al., 2025; Chen et al., 2024) to accelerate convergence and reduce computational overhead. As shown in Figure 1a, pretrained (PT) global models under existing defenses exhibit a substantially higher attack successful rate under backdoor attacks compared with their train-from-scratch (TS) counterparts. We hypothesize that this failure is primarily due to their reliance on statistical discrepancies among client gradient updates (e.g., cosine similarity), which can be substantially minimized by initializing the global model using pretrained weights (Nguyen et al., 2023b; Chen et al., 2023; Tan et al., 2022) as depicted in Figure 1b. To investigate its role, we progressively replace the randomly initialized layers with pretrained ones and conduct various backdoor attacks. As demonstrated in Figure 1c, it can be observed that the normalized cosine distance between benign updates and malicious updates consistently decreases, indicating that backdoor attacks become harder to detect as more pretraining is involved. Due to this attenuated updates discrepancy, existing defenses gradually fail to identify malicious clients from the benign ones. As a result, the

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.



(a) Performance Gap (b) Updates Cosine Similarity (c) Effect of Pretraining (d) Effect on ASR

Figure 1: Empirical analysis of the vulnerability of federated learning backdoor defenses to pretraining. Experiments are conducted on CIFAR-100 against BadPFL (Fan et al., 2025) attack. (a) shows that existing backdoor defense methods suffer severe protection degradation when applied to pretrained (PT) global models instead of train-from-scratch (TS) ones. (b) compares model updates cosine similarity between malicious ($m1, m2$) and benign clients ($b1 - b8$) using a pretrained or randomly initialized global model. Subsequently, we progressively replace layers of a randomly initialized ResNet-18 with pretrained ones until obtaining a fully pretrained model. (c) indicates that the normalized cosine distance between malicious and benign updates steadily decreases as more pretraining layers are added across various backdoor attacks (details in Appendix A). (d) reports the increased attack success rate, indicating that the resistance to backdoor attacks gradually diminishes.

adversary achieves increased attack success rates as illustrated in Figure 1d. These findings highlight that existing defense are vulnerable to pretraining, as stronger pretraining enables malicious models to stay more aligned with benign ones in the model gradient updates. Therefore, this motivates us to raise a natural question: *What signals can effectively distinguish malicious clients from the benign ones especially when federated learning starts from pretrained backbones?*

As exploited in (Yosinski et al., 2015), although individual channels often exhibit semantic preferences by activating strongly on particular spatial regions or visual patterns, these preferences are not disjoint and may overlap across semantically related features. However, trigger-induced shortcuts fundamentally deviate from the natural semantic features associated with the ground-truth class (Guo et al., 2023; Fan et al., 2025). Thus, we hypothesize that channels becoming preferred in trigger-related patterns are less likely to respond to natural features shared across clean samples and therefore tend to remain inactive when the corresponding trigger is absent, resulting in discrepant activation patterns between benign and backdoored models. To explore this, we examine the channel activation magnitudes[1] between them in Section 2.2. The results reveal that, under both pretrained and trained-from-scratch scenarios, the benign model exhibits relatively consistent activation magnitudes across all channels, whereas channels in the poisoned model tend to activate on either clean or poisoned samples, but rarely on both. We refer to this intriguing phenomenon as *channel activation inconsistency*, where channels respond inconsistently to natural and trigger-induced input features. Motivated by these findings, we propose **C**hannel **A**bnormality **S**core (**CAS**), a simple yet effective malicious detection metric by evaluating the contribution of each individual channel to each class within a set of categories via the notion of term-frequency–inverse document frequency (TF-IDF). A client will be flagged as malicious if its CAS exhibits a significant amount of discrepancy compared to that of other clients and thus excluded from the global model aggregation. To the best of our knowledge, we make the first attempt to detect malicious clients under the pretrained federated learning setting without relying on gradient-update statistics. Our main contributions are four-fold:

- We identify a critical failure of existing federated learning backdoor defenses when applied to pretrained backbones, attributing to their over-reliance on the model update statistics.
- We empirically uncover the abnormal channel activation pattern of a malicious client model, serving as a new perspective for backdoor detection.
- We propose CAS, a novel malicious detection metric for pretrained federated learning. In essence, CAS measures the semantic closeness between channels and categories via the notion of TF-IDF and identifies clients with abnormal patterns as malicious.
- We conduct extensive experiments to evaluate the effectiveness of CAS. The results demonstrate that CAS can counter SOTA backdoor attacks under pretrained backbones and consistently provide superior protection compared to existing defenses.

[1]The spatially averaged activation value of each channel.

2 PROPOSED METHOD

2.1 WHEN FEDERATED LEARNING STARTS PRETRAINED

Throughout the paper, we use visual recognition as the running example. Following the previous studies (Nguyen et al., 2023b; Chen et al., 2023; Tan et al., 2022), we assume that the server aims to conduct federated learning on the pretrained backbones by distributing a global model with pretrained weights to each local client at the first communication round. Specifically, considering a federated learning system with K participating clients, each client $k \in \mathcal{K}$ possesses a private local dataset $\mathcal{D}_k = \{x_i, y_i\}_{i=1}^{D_k}$ with size D_k, where x is the input and y is the true label among N categories. At the t-th communication round, the server broadcasts the global model $\boldsymbol{w}_t$ to each selected client k to perform local training by

$$\boldsymbol{w}_t^k = \arg\min_{\boldsymbol{w}_t} \mathcal{L}\left(\boldsymbol{w}_t, \mathcal{D}_k\right) \tag{1}$$

where $\mathcal{L}\left(\cdot,\cdot\right)$ denotes classification loss (e.g., Cross Entropy), and sends the updates $\boldsymbol{w}_t^k - \boldsymbol{w}_t$ back to the server. The server aggregates the updates $\boldsymbol{w}_{t+1}$ for the next round by $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \sum_{k\in\mathcal{K}} q_k \left(\boldsymbol{w}_t^k - \boldsymbol{w}_t\right)$ where q_k denotes the specified aggregation rule.

Threat Model. Compromised clients may modify local data, manipulate gradients, or submit crafted updates, but do not have access to global aggregation strategy, defense mechanism, or benign clients' data. The attacker could begin the poisoning from any global round in a continuous or alternative fashion. We also do not make constraints on the number of participating clients that the attacker could corrupt in every global round, which are typically assumed below 50% in existing research.

2.2 CHANNEL ACTIVATION INCONSISTENCY

In this section, we explore the channel activation behaviors between benign and malicious clients and empirically reveal a distinctive characteristic of backdoor attacks that persists under pretrained backbones. Before revealing our key observations, we first review the general process of backdoor attacks in federated learning.

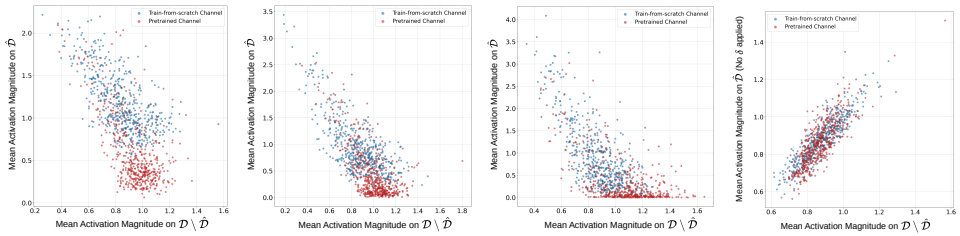
Malicious objective. We consider an adversary that can manipulate certain clients in $\mathcal{K}$ to pollute the global model by injecting a backdoor task into the training process of their local models. Let $\hat{y}$ be the label of the target class chosen by the adversary. When the trigger appears in the inference stage on a sample x with any ground-truth label from $\{y^i\}_{y^i\neq\hat{y}}$, the global model would unfaithfully predict it as $\hat{y}$. In the meantime, the malicious clients do not want to tamper with the model's normal prediction accuracy on clean samples. Formally, a malicious client $\hat{k}$ performs local training via

$$\boldsymbol{w}_t^{\hat{k}} = \arg\min_{\boldsymbol{w}_t, \delta(x)} \mathbb{E}_{(x,y)\sim\mathcal{D}_{\hat{k}}} \left[(1-\alpha)\cdot\mathcal{L}\left(f_{\boldsymbol{w}_t}\left(x\right), y\right) + \alpha\cdot\mathcal{L}\left(f_{\boldsymbol{w}_t}\left(\delta\left(x\right)+x\right), \hat{y}\right)\right], \tag{2}$$

where $f_{\boldsymbol{w}_t}$ denotes the classifier of the model $\boldsymbol{w}_t$, δ is the trigger generation function, $\hat{y}$ is the target label, and α denotes the importance of the backdoor task relative to the main task.

Settings. We select three representative federated backdoor attacks for analysis, including Neurotoxin (Zhang et al., 2022), IBA (Nguyen et al., 2023a), and BadPFL (Fan et al., 2025). For each attack, we split 10% of the training data as the backdoor subset $\hat{\mathcal{D}}$, apply the corresponding trigger $\delta(\cdot)$ to these samples, and relabel them to the target class $\hat{y}$. Each backdoored model is then trained until achieving a high attack success rate ($\geq 95\%$). For a fair comparison, the benign model is trained on the same data partition, including $\hat{\mathcal{D}}$, but without trigger injection or label modification. Finally, we compare the mean channel activation magnitudes between the backdoored subset $\hat{\mathcal{D}}$ and the remaining clean subset $\mathcal{D}\setminus\hat{\mathcal{D}}$.

Observation. As shown in Figure 2d, the benign model's channels produce relatively similar activation magnitudes between the two data subsets. However, Figures 2a -2c demonstrates a chaotic and nearly inverse activation pattern on each backdoored model. To make this clearer, we additionally visualize two channels respectively from the benign and malicious models, as illustrated in Figure 3, that produce the most and least activations in response to the poisoned test sample. As shown, the malicious model's channel producing the strongest activation on the poisoned sample barely activates on the clean samples or vice versa. In contrast, the benign channel activates on all samples equally even with the presence of the backdoor trigger.



(a) Neurotoxin (b) IBA (c) BadPFL (d) Benign

Figure 2: Illustration of the channel activation consistency between backdoored and benign models with/without pretrained initializations. (a)-(c) compare the per-channel mean activation magnitude of different backdoored models between $\hat{\mathcal{D}}$ and $\mathcal{D} - \hat{\mathcal{D}}$ under Neurotoxin, IBA, and BadPFL attacks, whereas (d) shows that of the benign model but with no trigger injected to $\hat{\mathcal{D}}$. Activations are captured from the last convolutional layer on ResNet-18 with a *ReLU* activation function.

Insights from the Observation. As shown in Figures 2a -2c, a much larger portion of channels in the pretrained global model exhibit near-zero activation magnitudes on backdoored samples, while producing considerably stronger activations on clean samples, compared to that of trained-from-scratch counterpart. Such stronger inconsistency of pretrained malicious model reveals that, driven by the semantic pathway inherent in priors of pretrained backbones, channels are able to resume capturing meaningful features shared across categories without being confined to the backdoored-introduced spurious correlations with the target label $\hat{y}$, which also explains the partial cause of increased updates similarity between benign and malicious clients revealed in Section 1. Therefore, quantifying the level of channel activation inconsistency can serve as an efficient malicious filtering signal particularly for the pretrained federated learning setting.

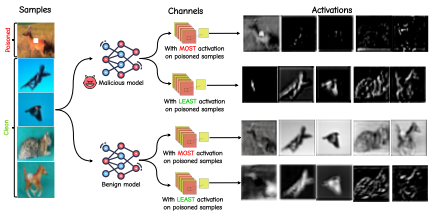


Figure 3: Example visualization of feature maps for channels with the most and least activation magnitudes on the poisoned samples between malicious and benign models, respectively.

2.3 Channel Abnormality Score

Motivated by the observation in the previous section, we propose a simple yet effective backdoor detection metric, termed Channel Abnormality Score (CAS), that quantifies the aforementioned inconsistency level. Specifically, CAS first treats each channel as a "word" and each class as a "document," and then leverages the notion of TF-IDF to quantify class-specific channel contributions. The term frequency measures how strongly a channel responds to a given class, while the inverse document frequency reflects how selectively that channel contributes across different classes. A channel with both high TF and high IDF for a particular class therefore exhibits a strong but class-specific activation pattern, which may indicate that it captures features unique to that class, such as a backdoor trigger associated with the target label. Based on these TF-IDF patterns, CAS quantifies the level of channel activation inconsistency of each client and identifies clients whose patterns deviate substantially from the others. The overall pipeline of our method is illustrated in Figure 4.

The Term Frequency. To transform the activation of the j-th channel into a word, we utilize the class activation running mean $\mu^{n,j}$ for each class n during training, which can be defined as

$$\mu_{(b)}^{n,j} = (1-m)\,\frac{D_{b,n}}{D_b}\mu^{n,j,b} + m\mu_{(b-1)}^{n,j}, \tag{3}$$

where $D_{b,n}$ and D_b is the number of class n's samples in the b-th batch, respectively, and $m \in [0,1]$ denotes the momentum coefficient of the exponential moving average process. Meanwhile, $\mu^{n,j,b}$ denotes per-batch class activation mean and can be calculated via $\mu^{n,j,b} = \frac{1}{D_{b,n}} \sum_{x \in \mathcal{D}_{b,n}} \text{AvgPooling}\left(\sigma\left(x\right)\right) \in \mathbb{R}^1$ where AvgPooling denotes the averaging pooling operation and σ is the activation function. When a word appears frequently in a document, it is regarded as having a stronger contribution to that document. Similarly, a channel with a large $\mu^{n,j}$ activates

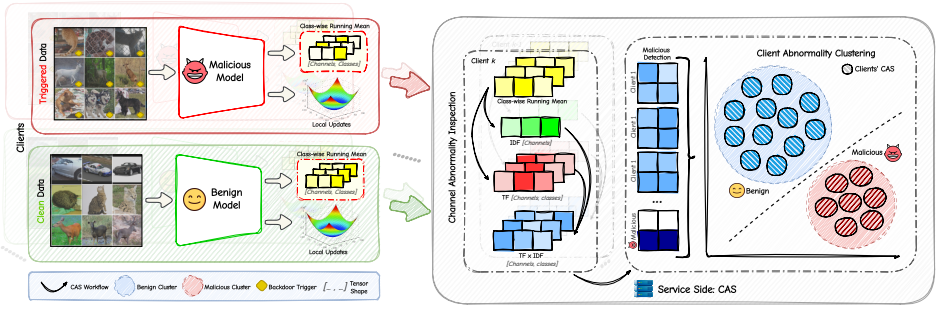


Figure 4: Pipeline illustration of our method. Clients upload class-wise running means along with their local updates, which are then utilized by the server to compute the corresponding abnormality scores. Based on these scores, the server clusters clients via *Wasserstein distance*-based clustering into two groups and drop the one with smaller center norm.

strongly to the visual patterns present in class n, suggesting a larger contribution to its representation. Therefore, the term frequency (TF) of channel j in class n can be measured as

$$\mathrm{TF}^{n,j} = \frac{\mu_k^{n,j}}{\sum_{j \in C_{\text{out}}} \mu^{n,j}}, \tag{4}$$

where C_{out} denotes the number of output channels in any layer.

The Inverse Document Frequency. When some channels of a benign model have higher TF value for class n, they also tend to have greater contributions (e.g., activation magnitudes) to samples in other classes, as illustrated in Figure 2d. These channels often capture visual patterns that are shared across multiple classes (e.g., round-shaped structures such as wheels, pizzas, and footballs). In contrast, backdoor triggers introduce shortcut features that are largely unrelated to the natural semantic patterns of clean data. Consequently, some channels in a malicious model may show high activeness on trigger-related features associated with the target label $\hat{y}$, exhibiting strong activation for the target class but limited activation to other clean classes. Thus, we utilize the inverse document frequency (IDF) to quantify how much contribution each channel makes to other classes by

$$\mathrm{IDF}^{j} = \log\left(\frac{N}{1 + \sum_{n \in N} \mathbb{I}[\mu^{n,j} > \bar{\mu}^{j}]}\right), \tag{5}$$

where $\mathbb{I}[\cdot]$ is the indicator function and $\bar{\mu}^{j} = \frac{1}{N}\sum_{n \in N} \mu^{n,j}$. Essentially, IDF^{j} represents how common or rare a specific word (channel j) is in the entire document sets (classes). The closer it is to 0, the more common of a channel is. Hence, if a channel has contribution in many classes, its IDF will approach 0. Otherwise, it will approach 1.

Channel Activation Inspection. Each client k uploads its words $\{\{\mu_k^{n,l,j}\}_{n \in N}\}_{j=1}^{C_{\text{out}}^l}$ of the l-th layer to the server along with its local model update. For a benign model, the activation of each channel is expected to be consistent across samples from different classes, as shown in Figure 2d. In this case, a channel with higher TF should have a lower corresponding IDF or vice versa. However, the channel activations are not consistent in a malicious model as a higher TF of a channel often comes with a higher IDF or vice versa. To quantify the severity of the activation inconsistency, the server computes the TF-IDF representation of channel j for class n as $\text{TF-IDF}_k^{n,l,j} = \mathrm{TF}_k^{n,l,j} \cdot \mathrm{IDF}_k^{l,j}$ and determines the normalized TF-IDF distribution

$$\mathcal{P}_k^l(n,j) = \frac{\text{TF-IDF}_k^{n,l,j}}{\sum_{n=1}^{N} \text{TF-IDF}_k^{n,l,j}}, \tag{6}$$

where $\mathcal{P}_k^l(n,j)$ is interpreted as the probability mass assigned to TF-IDF coordinate (n,j). Subsequently, we define the activation consistency entropy of the channel j as its abnormality score:

$$z_k^{l,j} = -\sum_{n=1}^{N} \mathcal{P}_k^l(n,j) \log \mathcal{P}_k^l(n,j), \tag{7}$$

A larger $z_k^{l,j}$ reflects a more uniform activation distribution among all classes, whereas a smaller value indicates that the channel is concentrated within a limited number of categories. Finally, the server obtains the abnormality scores for all channels $\mathbf{z}_k^l = \left[z_k^{l,j}, \ldots, z_k^{l,C_{\text{out}}^l}\right]^\top \in \mathbb{R}^{C_{\text{out}}^l}$.

Channel Abnormality Clustering. We regard the entries of each client's abnormality vector $\mathbf{z}_k$[2] as a sample from a one-dimensional empirical distribution and partition the clients into two clusters via K-means. Standard K-means relies on Euclidean distance, which is sensitive to the element-wise alignment of vectors and therefore may be unsuitable when the ordering of channel-wise abnormality scores is not semantically meaningful. To address this, we instead adopt the 1D *Wasserstein* distance, which focuses on the overall distributions of each client's $\mathbf{z}_k^l$ and is invariant to permutations of vector elements. During each clustering iteration, each client k is assigned to the cluster q whose center $\mathbf{z}_q$ is the closest under the *Wasserstein* distance as

$$\min_q \mathcal{W}_1(\mathbf{z}_k, \mathbf{z}_q) = \frac{1}{C_{\text{out}}^l} \sum_j |z_k^{(j)} - z_q^{(j)}|, \qquad \mathbf{z}_q = \frac{1}{|q|} \sum_{k=1}^{|q|} \text{sort}\left(\mathbf{z}_k\right), \tag{8}$$

where $z_k^{(j)}$ and $z_q^{(j)}$ are the sorted values, $|q|$ denotes the number of clients in q, and sort $(\cdot)$ represents the sorting operation. The assignments are iteratively updated until convergence. Since benign clients generally exhibit more consistent channel activations across classes and thus larger activation-consistency entropy, we regard the cluster with the larger center norm $\|\mathbf{z}_q\|_2$ as benign and retain its clients for global aggregation. We show the complete algorithm of CAS in Appendix J. Furthermore, we explore the effectiveness of CAS under an even stronger threat model where the adversary has prior knowledge to intentionally manipulate $\mathbf{z}_k^l$ in Appendix E.

3 Experiment

Basic setups. We simulate a cross-silo federated learning setting with 20 clients, where 4 of them are guaranteed to be malicious, by following (Xu et al., 2025; Huang et al., 2023). Additionally, we also present the superior performance of CAS on a cross-device federated learning system with 100 clients and client sampling by following (Wan et al., 2025; Fan et al., 2025) in Appendix F.1. More implementation details can be found in Appendix B. All experiments are repeated with five random seeds $\{42, 43, 44, 45, 46\}$. We provide the source code in the supplementary.

Datasets and Models. We mainly experiment on CIFAR10 (Krizhevsky et al., 2009), CIFAR100 (Krizhevsky et al., 2009), SVHN (Netzer et al., 2011), and Tiny-ImageNet (Wu et al., 2017) datasets. To simulate non-IID distribution, we adopt the widely used Dirichlet distribution with a default sampling parameter $\alpha = 1.0$. We use ImageNet-1K (Russakovsky et al., 2015) pretrained ResNet-18 (He et al., 2016) weights, downloaded from PyTorch, as the initial global model $\boldsymbol{w}_0$. For Tiny-ImageNet, we instead use Place365-pretrained backbone (Zhou et al., 2017).

Evaluated Attack and Defense Methods. We consider 4 SOTA backdoor attacks: BadPFL (Fan et al., 2025), IBA (Nguyen et al., 2023a), Neurotoxin (Zhang et al., 2022), and DBA (Xie et al., 2019) with a poisoning rate $p = 40\%$. On the defense side, we compare CAS against 7 SOTA defense methods, including AlignIns (Xu et al., 2025), MARS (Wan et al., 2025), FDCR (Huang et al., 2024), Lockdown (Huang et al., 2023), Trimmed-Mean (Yin et al., 2018), Multi-Krum (Blanchard et al., 2017), and FedAvg (McMahan et al., 2017). Details can be found in Appendix G.

Evaluation Metrics. We assess the performance of the defenses using several metrics, including main task accuracy (ACC), attack success rate (ASR), true positive rate (TPR), and false positive rate (FPR). A good defense method should achieve high ACC and TPR while reducing ASR and FPR. A more detailed description of these metrics can be found in Appendix H.

3.1 Experimental Results

Main Results. Table 1 and Table 2 report the performance of all defense methods against four backdoor attacks across CIFAR-10, SVHN, CIFAR-100, and Tiny ImageNet when the global model is initialized with pretrained backbones, with the best result for each metric in bold and the second-best

[2]We ignore l for simplicity

Table 1: Performance (ACC/ASR) of defense methods against different attacks

Defenses	BadPFL		IBA		DBA		Neurotoxin		**Average**	
	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
CIFAR-100										
FedAvg	**79.95$_{0.53}$**	62.21$_{1.47}$	**79.42$_{0.86}$**	61.06$_{9.20}$	**79.86$_{0.42}$**	34.65$_{5.50}$	**79.55$_{0.66}$**	73.72$_{2.54}$	**79.70**	57.91
MARS	79.16$_{1.01}$	48.14$_{5.98}$	78.83$_{0.87}$	45.90$_{17.97}$	78.77$_{0.53}$	53.46$_{7.51}$	79.44$_{0.69}$	71.69$_{0.36}$	79.05	54.80
FDCR	79.25$_{0.75}$	76.99$_{10.04}$	78.77$_{0.77}$	72.57$_{25.10}$	79.61$_{0.91}$	48.89$_{14.44}$	79.39$_{0.42}$	67.48$_{5.16}$	79.25	66.48
Trim-Mean	79.25$_{0.52}$	49.18$_{5.55}$	79.20$_{0.47}$	88.61$_{1.76}$	79.51$_{0.87}$	65.74$_{3.26}$	79.52$_{0.41}$	77.88$_{1.20}$	79.85	72.07
Multi-Krum	79.05$_{0.58}$	76.52$_{5.71}$	78.83$_{0.67}$	75.54$_{6.23}$	78.63$_{0.71}$	63.97$_{4.65}$	78.93$_{0.40}$	82.19$_{1.85}$	78.86	74.55
Lockdown	76.02$_{0.12}$	23.81$_{13.46}$	74.72$_{0.72}$	3.55$_{0.69}$	74.83$_{0.66}$	**1.14$_{0.02}$**	75.72$_{0.31}$	1.43$_{0.20}$	75.32	7.48
AlignIns	79.57$_{0.43}$	51.64$_{4.31}$	78.50$_{0.88}$	21.39$_{27.32}$	79.14$_{0.76}$	46.06$_{3.79}$	79.04$_{0.77}$	65.56$_{4.50}$	79.06	46.16
CAS	79.13$_{0.45}$	**1.40$_{0.21}$**	78.70$_{0.80}$	**1.13$_{0.12}$**	78.80$_{0.55}$	1.63$_{0.66}$	78.90$_{0.68}$	**1.34$_{0.33}$**	78.88	**1.38**
CIFAR-10										
FedAvg	94.94$_{0.42}$	60.56$_{5.90}$	95.02$_{0.34}$	63.84$_{4.91}$	95.01$_{0.23}$	39.17$_{0.72}$	95.15$_{0.30}$	74.50$_{0.71}$	95.03	59.52
MARS	95.02$_{0.25}$	73.20$_{13.84}$	94.89$_{0.56}$	43.92$_{1.43}$	95.02$_{0.22}$	54.28$_{6.33}$	94.94$_{0.24}$	59.52$_{2.43}$	94.97	57.73
FDCR	95.08$_{0.13}$	59.86$_{15.34}$	95.14$_{0.44}$	65.72$_{0.89}$	94.98$_{0.27}$	78.29$_{4.44}$	95.06$_{0.41}$	78.69$_{3.64}$	95.06	70.64
Trim-Mean	95.07$_{0.43}$	58.45$_{13.27}$	95.09$_{0.20}$	65.85$_{2.43}$	**95.15$_{0.26}$**	61.92$_{3.43}$	95.10$_{0.30}$	79.75$_{1.62}$	**95.31**	70.38
Multi-Krum	**95.35$_{0.29}$**	68.95$_{5.95}$	95.28$_{0.35}$	57.24$_{5.36}$	95.07$_{0.41}$	55.31$_{9.52}$	**95.25$_{0.31}$**	84.09$_{1.49}$	95.24	66.40
Lockdown	93.50$_{0.54}$	34.27$_{7.57}$	93.53$_{0.21}$	18.82$_{6.72}$	93.33$_{0.25}$	10.63$_{0.69}$	93.51$_{0.42}$	19.81$_{3.47}$	93.47	20.88
AlignIns	94.99$_{0.48}$	62.82$_{17.58}$	95.13$_{0.24}$	47.58$_{10.70}$	95.03$_{0.32}$	49.76$_{4.10}$	94.97$_{0.41}$	77.06$_{0.44}$	95.03	59.30
CAS	95.06$_{0.43}$	**12.45$_{3.07}$**	95.15$_{0.42}$	**9.64$_{1.13}$**	94.94$_{0.37}$	**10.02$_{0.06}$**	95.05$_{0.11}$	**10.26$_{0.09}$**	95.05	**10.59**
SVHN										
FedAvg	97.16$_{0.06}$	84.97$_{5.49}$	97.14$_{0.04}$	81.89$_{17.53}$	97.18$_{0.03}$	74.87$_{4.25}$	97.16$_{0.02}$	85.97$_{1.05}$	97.16	81.92
MARS	97.15$_{0.03}$	34.42$_{5.42}$	97.06$_{0.07}$	41.93$_{14.41}$	97.11$_{0.04}$	31.90$_{4.11}$	97.15$_{0.05}$	64.73$_{18.24}$	97.12	43.24
FDCR	97.16$_{0.04}$	88.88$_{3.99}$	97.10$_{0.01}$	80.83$_{13.43}$	97.05$_{0.02}$	83.64$_{1.75}$	97.08$_{0.03}$	82.39$_{2.22}$	97.10	83.94
Trim-Mean	97.15$_{0.08}$	53.04$_{16.96}$	97.14$_{0.05}$	82.59$_{2.40}$	97.14$_{0.03}$	82.42$_{1.17}$	97.15$_{0.05}$	86.48$_{0.59}$	97.15	76.13
Multi-Krum	97.08$_{0.04}$	**13.47$_{4.43}$**	97.10$_{0.04}$	17.37$_{2.67}$	97.05$_{0.03}$	8.40$_{2.70}$	97.17$_{0.07}$	52.52$_{25.73}$	97.10	22.94
Lockdown	96.97$_{0.07}$	34.18$_{18.66}$	96.99$_{0.01}$	**16.88$_{17.12}$**	97.01$_{0.04}$	8.37$_{1.83}$	96.98$_{0.01}$	10.33$_{1.00}$	96.99	17.44
AlignIns	**97.24$_{0.06}$**	43.63$_{5.64}$	**97.21$_{0.02}$**	28.22$_{4.91}$	**97.22$_{0.01}$**	51.78$_{7.15}$	**97.21$_{0.04}$**	77.84$_{0.38}$	**97.22**	50.37
CAS	97.08$_{0.05}$	20.15$_{5.23}$	97.09$_{0.04}$	19.73$_{2.25}$	97.12$_{0.03}$	**6.84$_{0.07}$**	97.11$_{0.04}$	**6.83$_{0.04}$**	97.10	**13.39**
Tiny ImageNet										
FedAvg	**63.51$_{0.15}$**	57.04$_{3.66}$	**59.67$_{0.21}$**	43.17$_{11.29}$	**59.81$_{0.17}$**	17.53$_{3.14}$	**63.53$_{0.06}$**	70.58$_{0.20}$	**61.63**	47.08
MARS	62.22$_{0.48}$	19.25$_{5.86}$	57.42$_{0.16}$	6.95$_{6.28}$	57.90$_{1.52}$	4.05$_{0.86}$	62.35$_{0.21}$	30.24$_{7.89}$	59.97	15.12
FDCR	62.65$_{0.19}$	43.13$_{22.93}$	58.86$_{0.39}$	82.96$_{2.95}$	57.69$_{0.16}$	82.69$_{0.11}$	63.05$_{0.16}$	45.11$_{14.39}$	60.56	63.47
Trim-Mean	63.27$_{0.12}$	50.06$_{2.13}$	59.20$_{0.35}$	59.22$_{0.30}$	59.52$_{0.01}$	26.42$_{2.09}$	63.39$_{0.28}$	72.55$_{0.04}$	61.35	52.06
Multi-Krum	62.28$_{0.23}$	67.69$_{5.46}$	58.55$_{0.15}$	40.58$_{7.28}$	58.76$_{0.14}$	18.91$_{3.01}$	62.52$_{0.16}$	76.50$_{0.52}$	60.53	50.92
Lockdown	54.73$_{0.16}$	12.28$_{5.93}$	53.66$_{0.31}$	1.09$_{0.33}$	53.46$_{0.14}$	0.61$_{0.06}$	54.66$_{0.14}$	1.10$_{0.75}$	54.13	3.77
AlignIns	62.89$_{0.12}$	23.44$_{5.32}$	59.00$_{0.19}$	18.81$_{11.73}$	59.02$_{0.10}$	0.65$_{0.07}$	62.76$_{0.30}$	1.90$_{2.16}$	60.92	11.20
CAS	62.48$_{0.02}$	**0.39$_{0.06}$**	58.88$_{0.19}$	**0.79$_{0.25}$**	58.70$_{0.07}$	**0.55$_{0.03}$**	62.52$_{0.03}$	**0.55$_{0.04}$**	60.65	**0.57**

Table 2: Performance (TPR/FPR) of defense methods against different attacks

Defenses	BadPFL		IBA		DBA		Neurotoxin		**Average**	
	TPR↑	FPR↓	TPR↑	FPR↓	TPR↑	FPR↓	TPR↑	FPR↓	TPR↑	FPR↓
CIFAR-100										
MARS	41.88$_{4.37}$	24.69$_{7.99}$	70.42$_{14.66}$	9.74$_{5.11}$	52.29$_{7.71}$	27.66$_{2.87}$	41.04$_{1.30}$	26.82$_{0.36}$	51.41	22.23
FDCR	18.12$_{2.72}$	30.05$_{4.02}$	35.42$_{19.98}$	25.62$_{11.15}$	53.75$_{15.17}$	2.60$_{1.25}$	50.42$_{2.19}$	11.77$_{3.90}$	39.43	17.51
Multi-Krum	15.42$_{3.97}$	21.15$_{0.99}$	24.79$_{10.54}$	18.80$_{2.63}$	1.46$_{0.72}$	24.64$_{0.18}$	0.42$_{0.36}$	24.90$_{0.09}$	10.52	22.37
AlignIns	40.62$_{6.76}$	28.18$_{2.01}$	80.83$_{25.89}$	15.05$_{6.28}$	40.00$_{3.90}$	30.36$_{1.85}$	30.62$_{2.17}$	32.66$_{0.87}$	48.02	26.56
CAS	**100.00$_{0.00}$**	**0.00$_{0.00}$**	**100.00$_{0.00}$**	**0.00$_{0.00}$**	**93.75$_{3.31}$**	**0.73$_{0.74}$**	**95.42$_{2.60}$**	**3.80$_{1.88}$**	**97.29**	**1.13**
CIFAR-10										
MARS	30.62$_{9.08}$	34.01$_{2.78}$	50.21$_{6.56}$	31.41$_{0.31}$	33.12$_{2.72}$	32.08$_{3.55}$	43.12$_{1.65}$	26.51$_{1.48}$	39.27	31.00
FDCR	47.71$_{3.97}$	12.14$_{6.41}$	37.50$_{7.21}$	12.92$_{3.13}$	17.50$_{6.34}$	20.16$_{4.37}$	23.54$_{6.91}$	22.55$_{4.53}$	31.56	16.94
Multi-Krum	16.87$_{4.96}$	20.78$_{1.24}$	20.42$_{1.57}$	19.90$_{0.39}$	8.54$_{1.57}$	22.86$_{0.39}$	5.83$_{2.01}$	23.54$_{0.50}$	12.91	21.77
AlignIns	44.79$_{26.27}$	34.11$_{4.83}$	44.38$_{14.70}$	31.93$_{3.29}$	41.25$_{4.51}$	34.22$_{0.41}$	44.79$_{0.95}$	34.06$_{1.13}$	43.80	33.58
CAS	**95.00$_{3.48}$**	**2.86$_{2.08}$**	**95.83$_{1.44}$**	**1.98$_{0.55}$**	**97.08$_{1.44}$**	**5.78$_{0.72}$**	**95.21$_{0.95}$**	**7.08$_{0.92}$**	**95.78**	**4.42**
SVHN										
MARS	71.50$_{8.72}$	8.13$_{1.94}$	69.67$_{12.57}$	13.54$_{7.15}$	73.67$_{3.75}$	8.13$_{3.21}$	72.00$_{12.28}$	9.63$_{3.90}$	71.71	9.86
FDCR	19.33$_{0.76}$	32.62$_{0.25}$	29.67$_{20.23}$	22.21$_{15.10}$	20.83$_{1.61}$	28.00$_{1.19}$	22.67$_{5.58}$	29.62$_{1.77}$	23.12	28.11
Multi-Krum	96.33$_{0.76}$	0.92$_{0.19}$	97.17$_{0.76}$	0.71$_{0.19}$	87.17$_{15.31}$	3.21$_{3.83}$	39.83$_{28.04}$	15.04$_{7.01}$	80.12	4.97
AlignIns	49.79$_{3.61}$	29.22$_{2.30}$	61.04$_{6.01}$	26.67$_{0.77}$	49.17$_{2.19}$	31.51$_{0.86}$	57.71$_{3.15}$	28.96$_{1.49}$	54.43	29.09
CAS	**97.33$_{2.02}$**	**0.04$_{0.07}$**	**97.33$_{2.25}$**	**0.04$_{0.07}$**	**98.67$_{0.76}$**	**0.21$_{0.19}$**	**99.00$_{0.87}$**	**0.29$_{0.31}$**	**98.08**	**0.14**
Tiny ImageNet										
MARS	77.33$_{4.93}$	16.87$_{6.83}$	90.83$_{4.86}$	19.71$_{0.97}$	83.17$_{10.68}$	18.33$_{9.18}$	71.17$_{18.84}$	15.75$_{9.62}$	80.63	17.67
FDCR	58.00$_{25.61}$	25.92$_{11.96}$	21.67$_{9.83}$	44.79$_{10.82}$	4.17$_{1.04}$	60.67$_{1.57}$	60.33$_{7.65}$	10.38$_{5.56}$	36.04	35.44
Multi-Krum	0.00$_{0.00}$	25.00$_{0.00}$	0.50$_{0.50}$	24.82$_{0.13}$	0.00$_{0.00}$	25.00$_{0.00}$	0.00$_{0.00}$	25.00$_{0.00}$	0.13	24.96
AlignIns	67.83$_{2.75}$	0.13$_{0.00}$	75.83$_{7.75}$	**0.00$_{0.00}$**	90.33$_{0.76}$	2.38$_{0.33}$	88.00$_{6.50}$	1.24$_{0.97}$	80.50	0.94
CAS	**100.00$_{0.00}$**	**0.00$_{0.00}$**	**99.83$_{0.29}$**	0.07$_{0.04}$	**99.50$_{0.50}$**	**0.42$_{0.19}$**	**99.33$_{0.29}$**	**1.08$_{0.26}$**	**99.67**	**0.39**

result highlighted. Overall, our method consistently achieves superior defense performance compared with existing baselines while preserving competitive main-task accuracy. Specifically, CAS effectively suppresses the ASR close to the random-guess level across different datasets, achieving average ASRs of only 1.38%, 10.59%, 13.39%, and 0.57%, respectively. This advantage is consistently observed across all four attacks, demonstrating that CAS does not rely on attack-specific characteristics. The detection results in Table 2 further showcase this strong attack mitigation performance. Although Lockdown provides strong backdoor suppression by consistently achieving the second-lowest ASR, its effectiveness comes at the cost of degraded main-task accuracy since a large portion of parameters are pruned during lockdown's consensus fusion process. Similarly, AlignIns demonstrates some effectiveness against backdoor attacks but it comes at the cost of a substantially higher FPR indicating that benign clients are unnecessarily excluded. In contrast, CAS offers effective defense services without compromising any of the evaluation metrics. These results demonstrate the effectiveness and robustness of CAS across diverse datasets and backdoor attacks.

Table 3: Comparison of detection performance under different pretraining

Defenses	BadPFL			IBA			DBA			Neurotoxin			Average		
	TPR↑	FPR↓	ASR↓	TPR↑	FPR↓	ASR↓	TPR↑	FPR↓	ASR↓	TPR↑	FPR↓	ASR↓	TPR↑	FPR↓	ASR↓
CIFAR100 (Pretraining: CIFAR10)															
MARS	$98.33_{0.95}$	$0.16_{0.03}$	$1.34_{0.38}$	$98.33_{0.72}$	$0.21_{0.09}$	$\mathbf{0.26_{0.12}}$	$\mathbf{99.17_{0.33}}$	$\mathbf{0.18_{0.10}}$	$1.11_{0.06}$	$78.75_{25.23}$	$12.94_{10.89}$	$23.34_{15.39}$	93.65	3.37	6.51
Multi-Krum	$9.79_{3.21}$	$22.55_{0.80}$	$76.12_{7.74}$	$8.96_{2.01}$	$22.76_{0.50}$	$65.15_{9.50}$	$5.00_{1.25}$	$23.75_{0.31}$	$50.03_{3.22}$	$1.46_{0.36}$	$24.64_{0.09}$	$71.79_{3.47}$	6.30	23.43	65.77
FDCR	$23.13_{4.90}$	$31.04_{7.26}$	$77.77_{18.08}$	$23.96_{11.12}$	$32.86_{6.73}$	$93.66_{1.48}$	$15.42_{6.07}$	$36.20_{2.24}$	$74.43_{4.03}$	$26.04_{5.81}$	$31.93_{2.82}$	$78.75_{6.31}$	22.14	33.01	81.15
AlignIns	$82.50_{8.13}$	$11.41_{4.57}$	$13.75_{6.20}$	$83.33_{0.72}$	$12.34_{2.98}$	$16.82_{1.30}$	$97.71_{0.95}$	$4.79_{2.53}$	$\mathbf{1.11_{0.06}}$	$\mathbf{100_{0.00}}$	$\mathbf{1.56_{1.18}}$	$\mathbf{1.12_{0.05}}$	90.89	7.53	8.20
CAS	$\mathbf{99.38_{1.08}}$	$\mathbf{0.00_{0.00}}$	$\mathbf{1.09_{0.59}}$	$\mathbf{99.38_{1.08}}$	$\mathbf{0.00_{0.00}}$	$2.40_{0.52}$	$93.33_{5.39}$	$1.25_{1.09}$	$1.16_{0.09}$	$94.17_{4.69}$	$6.82_{4.33}$	$1.42_{0.44}$	**96.57**	**2.02**	**1.52**
SVHN (Pretraining: CIFAR100)															
MARS	$40.21_{12.75}$	$25.42_{5.36}$	$60.80_{13.98}$	$42.29_{19.02}$	$24.06_{11.74}$	$41.42_{11.81}$	$69.17_{13.47}$	$12.86_{6.10}$	$20.64_{6.82}$	$\mathbf{72.71_{12.14}}$	$\mathbf{9.53_{4.20}}$	$49.10_{5.85}$	56.10	17.97	42.99
Multi-Krum	$\mathbf{94.33_{1.15}}$	$\mathbf{0.29_{0.17}}$	$\mathbf{12.22_{1.90}}$	$81.33_{18.45}$	$\mathbf{4.67_{4.61}}$	$31.73_{13.91}$	$0.67_{0.76}$	$24.83_{0.19}$	$92.85_{0.08}$	$0.33_{0.29}$	$24.92_{0.07}$	$90.06_{1.12}$	44.17	13.68	56.72
FDCR	$25.21_{16.97}$	$31.04_{8.86}$	$80.46_{10.13}$	$17.71_{7.17}$	$33.02_{5.63}$	$85.42_{14.34}$	$2.92_{1.80}$	$42.76_{2.23}$	$94.62_{1.61}$	$2.71_{0.95}$	$42.24_{4.86}$	$95.49_{0.02}$	12.14	37.27	89.00
AlignIns	$41.25_{4.72}$	$32.08_{0.33}$	$60.22_{4.53}$	$38.54_{5.24}$	$32.60_{2.22}$	$52.81_{5.18}$	$44.79_{1.80}$	$33.54_{2.11}$	$67.58_{7.25}$	$58.33_{5.60}$	$28.59_{1.87}$	$76.36_{1.48}$	45.73	31.70	64.24
CAS	$90.00_{2.72}$	$5.42_{3.37}$	$23.03_{5.02}$	$\mathbf{88.75_{1.08}}$	$6.77_{6.59}$	$\mathbf{22.58_{0.97}}$	$\mathbf{80.83_{9.93}}$	$\mathbf{11.56_{8.49}}$	$\mathbf{13.22_{0.12}}$	$71.04_{9.73}$	$16.35_{8.25}$	$\mathbf{36.36_{12.01}}$	**82.66**	**10.03**	**23.80**
CIFAR-100 (No Pretraining)															
MARS	$\mathbf{99.50_{0.12}}$	$\mathbf{0.56_{0.06}}$	$\mathbf{0.90_{0.36}}$	$\mathbf{97.92_{0.61}}$	$1.54_{1.35}$	$2.75_{1.72}$	$\mathbf{97.00_{2.14}}$	$1.67_{1.50}$	$\mathbf{1.12_{0.13}}$	$72.50_{8.20}$	$16.04_{2.62}$	$11.34_{7.88}$	**91.73**	4.95	4.03
Multi-Krum	$96.25_{2.17}$	$0.94_{0.54}$	$2.45_{0.51}$	$95.00_{2.17}$	$\mathbf{1.25_{0.54}}$	$\mathbf{2.47_{0.05}}$	$88.75_{4.10}$	$2.81_{1.02}$	$1.84_{0.31}$	$61.25_{10.51}$	$9.69_{2.63}$	$9.33_{4.53}$	85.31	**3.67**	**4.02**
FDCR	$96.25_{3.80}$	$9.11_{2.17}$	$2.54_{0.51}$	$57.92_{5.64}$	$23.23_{5.18}$	$38.68_{6.61}$	$87.08_{4.85}$	$\mathbf{0.05_{0.09}}$	$1.80_{0.28}$	$80.83_{1.91}$	$\mathbf{0.00_{0.00}}$	$4.50_{1.39}$	80.52	8.10	11.88
AlignIns	$84.79_{4.16}$	$42.14_{0.86}$	$18.26_{6.20}$	$93.75_{5.77}$	$35.31_{10.79}$	$4.90_{1.84}$	$57.50_{3.33}$	$48.91_{8.71}$	$30.25_{3.04}$	$\mathbf{92.50_{1.25}}$	$7.81_{0.83}$	$\mathbf{1.40_{0.26}}$	82.14	33.54	13.70
CAS	$81.67_{8.25}$	$9.58_{3.16}$	$12.85_{8.04}$	$89.37_{2.25}$	$5.42_{1.71}$	$3.58_{0.10}$	$66.46_{6.88}$	$19.53_{3.39}$	$6.47_{3.19}$	$81.25_{6.61}$	$10.31_{3.60}$	$4.20_{2.11}$	79.69	11.21	6.78

Transferability across Different Pretraining. As introduced in Section 2.1, we primarily adopt an ImageNet-pretrained global model and evaluate the defenses using ImageNet-pretrained backbones. To further assess the robustness of CAS, we additionally consider an initial global model pretrained on CIFAR100 and CIFAR10. As shown in Table 3, CAS maintains strong performance across different pretraining settings. For example, under CIFAR10 and CIFAR100 pretraining, CAS achieves average ASRs of only 1.52% and 23.80%, respectively, outperforming all baseline defenses. We further evaluate a global model without pretraining. In this case, the performance of several baseline defenses improves noticeably, as the stronger statistical discrepancy between benign and malicious updates makes detection easier. Although CAS exhibits a moderate performance drop, it still achieves competitive ASRs with an average ASR of 6.78%.

Performance on Untargeted Attacks. While our paper primarily focuses on targeted backdoor defense in federated learning, we further evaluate CAS against untargeted (Byzantine) attacks. Specifically, we consider three representative attacks, including PoisonedFL [CVPR'2025] (Xie et al., 2025), MPAF [CVPRW'2022] (Cao & Gong, 2022), and IPMA [UAI'2020] (Xie et al., 2020), which aim to degrade the global model accuracy by manipulating malicious local updates during aggregation. As shown in Figure 5, we report both the resulting test accuracy and the TPR of each defense under these attacks. CAS consistently achieves high malicious-client detection rates across both datasets. For CIFAR100, CAS maintains the highest overall test accuracy across attacks, with an average accuracy of 80%. These results demonstrate that, although CAS is primarily designed for targeted backdoor defense, its defensive effectiveness can also generalize to untargeted Byzantine attacks.

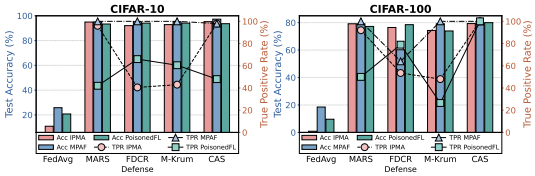


Figure 5: Comparison of defense against untargeted attacks on CIFAR10 and CIFAR100

Impact of Attack Scale. To assess robustness under more attackers, we increase the number of guaranteed malicious clients g to 30% in each communication round on CIFAR100. As shown in Figure 6, CAS consistently achieves near 100% malicious detection rate and keep the attack success rate close to random guess (1%). We also evaluate CAS under extreme malicious scenarios, where malicious clients constitute the *majority*, by setting $g = 50\%$ and $g = 70\%$. Although the ASR slightly increases with a larger number of attackers, the overall performance remains stable. These results confirm that CAS does not rely on the assumption that benign clients take the majority.

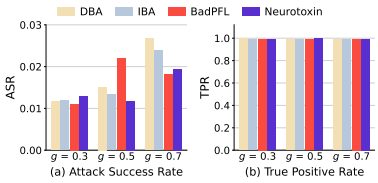


Figure 6: Attack Scale g

Impact of Data Heterogeneity. We evaluate the effect of data heterogeneity by varying the Dirichlet concentration parameter α on CIFAR100. As shown in Figure 7, we consider three different levels non-IID heterogeneity: (1) moderate non-IID ($\alpha = 0.5$), (2) high non-IID ($\alpha = 0.3$), and (3) extreme non-IID ($\alpha = 0.1$). Across all settings and backdoor attack methods, CAS consistently achieves around 90% true positive rate in correctly identifying malicious clients, indicating strong detection performance. Although CAS's performance degrades moderately under the extreme non-IID degree, it still demonstrates strong resilience to backdoor attacks under severe data heterogeneity.

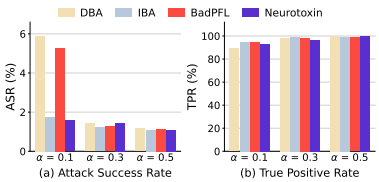


Figure 7: Data Heterogeneity α

Impact of Poisoning Rate. To demonstrate the impact of different amounts of backdoored samples, we evaluate the defense performance of CAS under different poisoning ratio p on CIFAR100. As shown in Figure 8, CAS can accurately identify almost all malicious clients under $p = 30\%$ and $p = 50\%$. However, we observe a slight performance drop when we set a smaller poisoning ratio (e.g., $p = 20\%$). This is because a smaller poisoning ratio often yields more stealthy malicious behavior and less abnormality signal, making the malicious clients relatively harder to be detected.

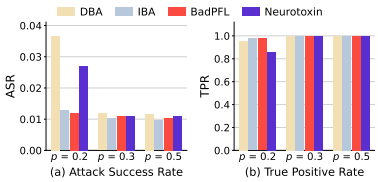


Figure 8: Poisoning rate p

Other Results. We have included additional results in the appendix due to space constraints. Specifically, Appendix E shows the effectiveness of CAS against adaptive attackers. Appendix F.1 evaluates defenses under the cross-device setting. Appendix F.2 presents results under additional federated learning setups. Appendix F.3 and F.4 show the results of different network architectures and aggregation rules. Appendix F.6 extends CAS to other distance metrics for malicious clustering.

4 Related Works

Federated backdoor defenses. Existing defenses against federated backdoor attacks can be categorized by whether they rely on model-update discrepancy as the primary anomaly signal. The first group detects malicious clients directly from their uploaded updates. Median (Xie et al., 2018), Trimmed Mean (Yin et al., 2018), and RFA (Pillutla et al., 2022) suppress the effects of parameters with outlier magnitudes, while Krum and Multi-Krum (Blanchard et al., 2017) further refine this idea by selecting clients whose updates are most similar to the majority under Euclidean distance. Similarly, AlignIns (Xu et al., 2025), Mars (Wan et al., 2025), and FDCR (Huang et al., 2024) measure the importance of each individual parameter or neuron as the malicious detection metric. However, as revealed in Section 1, these gradient updates-based defenses fail under the pretrained federated learning. More recently, another line of work has been developed to detect malicious clients without inspecting model updates inconsistency. Among them, BackdoorIndicator (Li & Dai, 2024) and FedFM-MTL (Sui et al., 2026) evaluate the performance of local models on a server-side auxiliary dataset to filter malicious clients. FEDRACE (Yan et al., 2025a) uses class-wise prototypes of clients to directly measure the prediction confidence deviation of each client on global class prototypes. Although effective, these methods exploit the sensitive information (e.g., test accuracy (Li & Dai, 2024; Sui et al., 2026) or prediction confidence (Yan et al., 2025a)) of clients, violating the privacy-preserving nature of federated learning. In contrast, CAS only utilizes channel activation magnitudes, which discard sample-level and spatial feature information and have been widely adopted in prior studies (Liu et al., 2021; Yang et al., 2024; Yan et al., 2025b).

Term Frequency–Inverse Document Frequency. The goal of TF-IDF is to emphasize discriminative patterns of words within documents. Recent studies have shown that it can also serve as an effective relevance-estimation mechanism. For example, graph-based text classification methods use TF-IDF to construct document-word edge weights before applying graph neural networks (Yao et al., 2019; Liu et al., 2020). Similar TF-IDF-based relevance signals have also been used in heterogeneous graph neural networks (Ragesh et al., 2021) and neural generation/retrieval systems (Song et al., 2018). In federated learning, TF-IDF has been adopted to unlearn class-discriminative information learned by the global model (Wang et al., 2022). Nevertheless, different from the prior works, our goal is to expose abnormal channel-class associations caused by backdoor training.

5 Conclusion

In this paper, we propose CAS, a malicious client detection metric specifically for the federated learning system initialized with a pretrained model. Unlike existing defenses that rely on the misalignment in update statistics among clients, which consistently diminishes under the pretrained setting, CAS offers a new backdoor detection metric from the perspective of channel activation, filtering out those demonstrating abnormal activation inconsistency. While effective for image classification tasks confirmed by the extensive experiments, extending this framework to domains such as text generation or image segmentation requires new strategies for word-document quantification. Future work will focus on adapting the model to support broader tasks in diverse settings.

6 AI USE STATEMENT

In this work, we used generative AI tools to polish the writing and presentation of the introduction in this paper. We have not used generative AI tools to generate synthetic datasets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat datasets, support qualitative and thematic data analysis, and interpret results. We have reviewed all AI-assisted content in the introduction. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

7 ETHICS STATEMENT

This work is conducted in accordance with the ICLR Code of Ethics. Our studies focus on the design of malicious client detection in federated learning, especially when the global model is initialized with pretrained backbones. All experiments are carried out on publicly available datasets (e.g., CIFAR10, CIFAR100, SVHN, and TinyImageNet) under their respective licenses, with no involvement of human subjects or sensitive personal data. In principle, this work has no ethical issues since it does not reveal any new security risks and is purely defensive. We report our methods and findings transparently to foster reproducibility and responsible use.

8 REPRODUCIBILITY STATEMENT

We have provided detailed information on datasets, models, training settings, and computational facilities in our main manuscript and appendix. The code for reproducing our main experiments has been attached in the supplementary and will be made publicly available upon the acceptance of the paper.

REFERENCES

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021.

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.

Xiaoyu Cao and Neil Zhenqiang Gong. Mpaf: Model poisoning attacks to federated learning based on fake clients. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3395–3403. IEEE, 2022.

Haokun Chen, Yao Zhang, Denis Krompass, Jindong Gu, and Volker Tresp. Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11285–11293, 2024.

Hong-You Chen, Cheng-Hao Tu, Ziwei Li, Han Wei Shen, and Wei-Lun Chao. On the importance and applicability of pre-training for federated learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Mingyuan Fan, Zhanyi Hu, Fuyi Wang, and Cen Chen. Bad-pfl: Exploring backdoor attacks against personalized federated learning. In *Proc. of ICLR*, 2025.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. In *ICLR*, 2023.

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.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017. doi: 10.1109/CVPR.2017.243.

Tiansheng Huang, Sihao Hu, Ka-Ho Chow, Fatih Ilhan, Selim Tekin, and Ling Liu. Lockdown: backdoor defense for federated learning with isolated subspace training. *Advances in Neural Information Processing Systems*, 36:10876–10896, 2023.

Wenke Huang, Mang Ye, Zekun Shi, Guancheng Wan, He Li, and Bo Du. Parameter disparities dissection for backdoor defense in heterogeneous federated learning. *Advances in Neural Information Processing Systems*, 37:120951–120973, 2024.

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

Songze Li and Yanbo Dai. BackdoorIndicator: Leveraging OOD data for proactive backdoor detection in federated learning. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 4193–4210, 2024.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. FedDG: Federated Domain Generalization on Medical Image Segmentation via Episodic Learning in Continuous Frequency Space. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. Tensor graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8409–8416, 2020.

Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3615–3634, 2024.

H. 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*, pp. 1273–1282. PMLR, 2017.

Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. Ppfl: Privacy-preserving federated learning with trusted execution environments. In *Proceedings of the 19th annual international conference on mobile systems, applications, and services*, pp. 94–108, 2021.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 4. Granada, 2011.

Dung Thuy Nguyen, Tuan Minh Nguyen, Anh Tuan Tran, Khoa D Doan, and KOK-SENG WONG. IBA: Towards irreversible backdoor attacks in federated learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

John Nguyen, Jianyu Wang, Kshitiz Malik, Maziar Sanjabi, and Michael Rabbat. Where to begin? on the impact of pre-training and initialization in federated learning. In *The Eleventh International Conference on Learning Representations*, 2023b.

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

Rahul Ragesh, Sundararajan Sellamanickam, Arun Iyer, Ramakrishna Bairi, and Vijay Lingam. Hetegcn: heterogeneous graph convolutional networks for text classification. In *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 860–868, 2021.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018. doi: 10.1109/CVPR.2018.00474.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Yiping Song, Cheng-Te Li, Jian-Yun Nie, Ming Zhang, Dongyan Zhao, and Rui Yan. An ensemble of retrieval-based and generation-based human-computer conversation systems. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 4382–4388. International Joint Conferences on Artificial Intelligence Organization, 2018.

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.

Yue Tan, Guodong Long, Jie Ma, Lu Liu, Tianyi Zhou, and Jing Jiang. Federated learning from pre-trained models: A contrastive learning approach. *Advances in neural information processing systems*, 35:19332–19344, 2022.

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. *Advances in neural information processing systems*, 2025.

Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jyyong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in neural information processing systems*, 33:16070–16084, 2020.

Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM web conference 2022*, pp. 622–632, 2022.

Kaidi Wang, Yi Ma, Mahdi Boloursaz Mashhadi, Chuan Heng Foh, Rahim Tafazolli, and Zhi Ding. Convergence acceleration in wireless federated learning: A stackelberg game approach. *IEEE Transactions on Vehicular Technology*, 74(1):714–729, 2025. doi: 10.1109/TVT.2024.3452933.

Jiayu Wu, Qixiang Zhang, and Guoxi Xu. Tiny imagenet challenge. In *Technical report*. Stanford University, 2017.

Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*, 2019.

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Generalized byzantine-tolerant sgd. *arXiv preprint arXiv:1802.10116*, 2018.

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Uncertainty in artificial intelligence*, pp. 261–270. PMLR, 2020.

Yueqi Xie, Minghong Fang, and Neil Zhenqiang Gong. Model poisoning attacks to federated learning via multi-round consistency. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15454–15463, 2025.

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

Gang Yan, Sikai Yang, and Wan Du. Fedrace: A hierarchical and statistical framework for robust federated learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Yunlu Yan, Huazhu Fu, Yuexiang Li, Jinheng Xie, Jun Ma, Guang Yang, and Lei Zhu. A simple data augmentation for feature distribution skewed federated learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25749–25758. IEEE, 2025b.

Seunghan Yang, Seokeon Choi, Hyunsin Park, Sungha Choi, Simyung Chang, and Sungrack Yun. Feature diversification and adaptation for federated domain generalization. In *European Conference on Computer Vision*, pp. 52–70. Springer, 2024.

Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 7370–7377, 2019.

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

Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. In *Deep Learning Workshop, International Conference on Machine Learning (ICML)*, 2015.

Sheng Yue, Zerui Qin, Yongheng Deng, Ju Ren, Yaoxue Zhang, and Junshan Zhang. Augfl: Augmenting federated learning with pretrained models. *IEEE Transactions on Networking*, 33(4): 1870–1885, 2025. doi: 10.1109/TON.2025.3548277.

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

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

APPENDIX: TABLE OF CONTENTS

A Discussion on Update Discrepancy **15**

B Detailed Experimental Setups **15**

C Computational and Communication Overheads of CAS **15**

D More Visualizations **16**

E Resilience to adaptive attack **17**

F Additional Experiments **19**

F.1 Cross-device Federated Learning. 19

F.2 Results on additional federated learning setups 20

F.3 Cross-Architecture transferability. 21

F.4 Results on different federated learning algorithms. 22

F.5 Results on Multi-Attack Scenario 22

F.6 Results on different distance metrics. 23

G Description of Evaluated Attacks and Defenses **24**

G.1 Attacks 24

G.2 Defenses 24

H Evaluation Metrics **25**

I Limitation of CAS **26**

J Algorithm Overview **27**

A Discussion on Update Discrepancy

In Figure 1c, as we progressively add pretrained layers, we compare the cosine distance between client model updates under four backdoor attacks to characterize the discrepancy between benign and malicious clients when there is no backdoor defense employed. For ease of comparison across different settings, we further normalize the resulting similarity statistics using global maximum normalization, as described below. This normalization ensures that all curves are scaled into the interval $[0, 1]$, facilitating a fair comparison of their relative updates discrepancy during training.

Let $s_{i,j} = \frac{1}{T}\sum_{t=1}^{T} s_{i,j}^{t}$ denote the average cosine similarity between the model updates of clients i and j after T communication rounds, and let B and M denote the sets of benign and malicious clients, respectively. For each attack, we first compute the average pairwise cosine distance among benign clients as

$$S_{BB} = \frac{1}{|B|(|B|-1)} \sum_{\substack{i,j \in B \\ i \neq j}} (1 - s_{i,j}), \tag{9}$$

where S_{BB} characterizes the intrinsic alignment among benign client updates. We then compute the average cosine distance between benign and malicious clients as

$$S_{BM} = \frac{1}{|B||M|} \sum_{\substack{i \in B \\ j \in M}} (1 - s_{i,j}), \tag{10}$$

which measures the degree of alignment between benign and malicious updates. For each attack a, we define the normalized cosine distance between the two groups as

$$\hat{S}_{(a)} = \frac{S_{BM}^{(a)} - S_{BB}^{(a)}}{H}, \forall a = 1, 2, 3, 4 \tag{11}$$

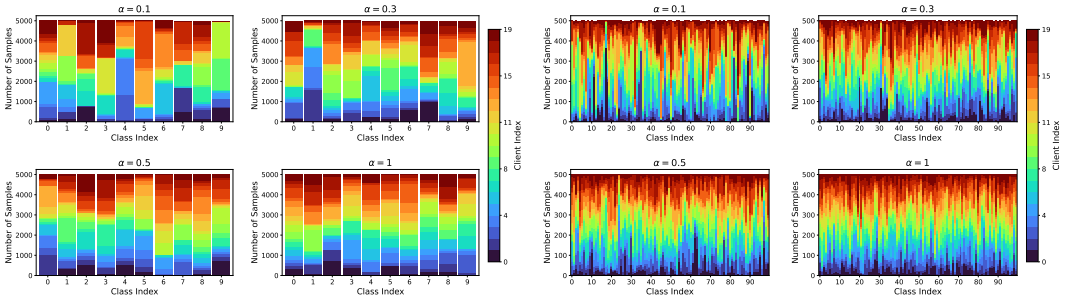
where $H = \max_{1 \leq a \leq 4} S_{BM}^{(a)} - S_{BB}^{(a)}$ serves the global normalization factor. A larger $\hat{S}^{(a)}$ indicates a larger discrepancy between benign and malicious updates relative to the intrinsic discrepancy among benign clients, whereas a smaller value indicates stronger alignment between the two groups.

B Detailed Experimental Setups

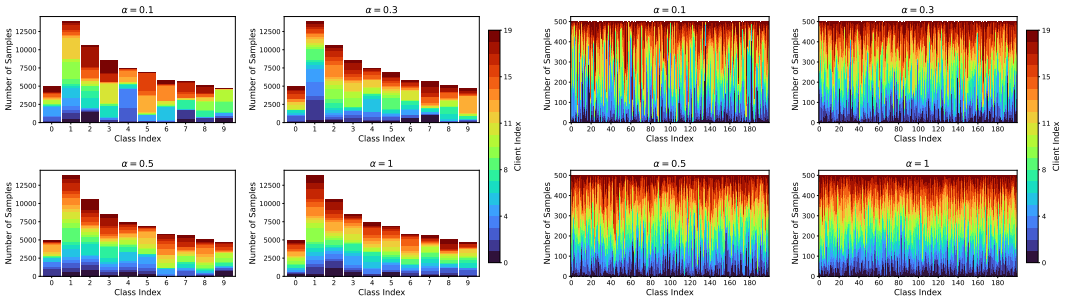
We use SGD as the local optimizer with an initial learning rate of $\eta = 0.01$ and a decay factor of 0.95. The global learning rate is set to 1. Each client trains a ResNet-18 local model for 2 local epochs per communication round, and the total number of communication rounds is set to 50. The batch size is 32. By default, all experiments are conducted under a cross-silo federated learning scenario unless mentioned otherwise. Also, the proportion of attackers M is set to 20%, meaning 20% of clients participating in each communication round are guaranteed to be compromised. To provide a clearer understanding of the degree of non-IID distribution, we present the label distribution of the local training datasets for clients using different parameters of the Dirichlet distribution, as shown in Figure 9. Furthermore, clients upload the channel statistics extracted from the last convolutional layer (e.g., `Layer4.1.conv2`), which are subsequently used to identify and filter malicious clients. After the malicious filtering procedure, we aggregate the remaining client according to FedAvg (McMahan et al., 2017) protocol, where clients share the same aggregation weight. For all datasets, we modify the first convolutional layer by replacing the original 7×7 kernel with a 3×3 kernel and remove the initial max-pooling layer to better preserve the spatial resolution of the input features. All experiments are conducted on a single NVIDIA RTX 5090 GPU and three NVIDIA RTX 3090 GPUs with `Python3.14`, `PyTorch 2.13`, and `CUDA 13.2`.

C Computational and Communication Overheads of CAS

Since clients are required to maintain and upload the class-wise activation running mean $\mu^{n,j}$ for each channel, the additional computation introduced by Equation 3 is only $O(NC_{\text{out}})$, which is negligible compared with the forward and backward computations required for local model training. Meanwhile, CAS requires each client k to additionally upload $\{\{\mu_k^{n,l,j}\}_{n \in \mathcal{N}}\}_{j=1}^{C_{\text{out}}^l}$, resulting in an

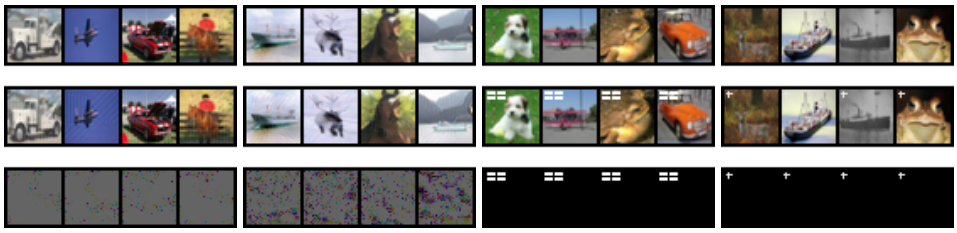


(a) CIFAR10 (b) CIFAR100



(c) SVHN (d) Tiny-ImageNet

Figure 9: Dirichlet data sampling with different α across four datasets.



(a) BadPFL (b) IBA (c) DBA (d) Neurotoxin

Figure 10: Visualization of Backdoor Attacks on CIFAR10. The first row shows the original clean images. The second row shows the backdoored samples. The third row shows the backdoor trigger

extra communication cost of NC^l_{out} scalar values. Therefore, the communication overhead grows linearly with the number of classes N and the number of output channels C^l_{out}. In practice, for a convolutional layer with kernel size $h \times w$ and C^l_{in} input channels, the corresponding weight tensor contains $C^l_{\text{out}} C^l_{\text{in}} hw$ parameters, whereas CAS introduces only NC^l_{out} additional scalars. The relative overhead is therefore approximately $\frac{N}{C^l_{\text{in}} hw}$, which is typically small for commonly used convolutional architectures. Moreover, CAS only needs to collect these statistics from selected layers rather than from the entire network, further limiting the communication cost. Since the number of semantic classes is generally much smaller than the dimensionality of the learned channel representation, i.e., $N \ll C^l_{\text{out}}$ for the layers used by CAS, the resulting overhead remains negligible.

D More Visualizations

We first provide visualization of **backdoor attacks** used in our paper. Figure 10 presents representative poisoned CIFAR-10 samples generated by the evaluated backdoor attacks. BadPFL and IBA introduce pixel-wise perturbations that are visually imperceptible, making the poisoned samples difficult to distinguish from their corresponding clean images. For BadPFL, we set the perturbation and disruptive-noise budgets to $\epsilon = \frac{4}{255}$ and $\sigma = \frac{4}{255}$, respectively. For IBA,

Table 4: CAS's detection performance of **adaptive attackers** under various Attack Scale M and Adaptive Strength λ

Attack Scale M	BadPFL				IBA				DBA				Neurotoxin			
	TPR↑	FPR↓	ASR↓	ACC↑	TPR↑	FPR↓	ASR↓	ACC↑	TPR↑	FPR↓	ASR↓	ACC↑	TPR↑	FPR↓	ASR↓	ACC↑
$\lambda = 0.4$																
20%	100.00	0.00	1.24	77.52	100.00	0.13	1.05	77.78	100.00	0.00	1.10	77.95	100.00	0.00	1.07	77.50
70%	100.00	0.00	0.84	68.97	99.71	0.00	0.75	69.72	99.86	0.00	1.06	69.25	100.00	0.00	1.00	69.48
$\lambda = 0.6$																
20%	100.00	0.00	1.38	77.64	99.00	0.00	0.98	78.03	100.00	0.00	1.11	77.67	100.00	0.00	1.11	78.02
70%	99.71	0.00	0.74	69.15	17.00	78.67	90.14	75.86	99.14	0.00	1.20	68.94	99.71	0.33	0.98	70.15
$\lambda = 0.8$																
20%	99.50	0.00	1.11	77.52	99.50	0.00	1.07	77.91	100.00	0.00	1.14	77.66	100.00	0.00	1.10	77.82
70%	24.57	72.33	84.59	76.30	7.29	79.67	91.92	76.19	99.43	0.00	1.04	70.00	99.29	0.33	1.04	70.44
$\lambda = 1$																
20%	99.50	0.00	1.27	77.27	98.50	0.00	1.04	77.72	100.00	0.00	1.08	78.01	100.00	0.00	1.05	77.65
70%	15.71	80.00	89.39	76.33	8.57	71.33	91.09	76.95	0.43	100.00	94.80	72.88	0.71	98.00	99.58	74.83

Table 5: Performance of CAS with activation function before batch normalization

Attacks	Datasets	Pretrained	ACC↑	ASR↓	TPR↑	FPR↓
Activation Function → Batch Normalization						
BadPFL	CIFAR100	No	65.22	11.23	90.00	2.81
	CIFAR100	Yes	79.17	1.82	98.50	0.00
DBA	CIFAR100	No	65.17	4.00	91.67	3.02
	CIFAR100	Yes	78.00	1.20	100.00	0.13
IBA	CIFAR100	No	65.15	3.17	95.83	1.67
	CIFAR100	Yes	79.14	1.42	97.50	0.88
Neurotoxin	CIFAR100	No	65.15	6.32	95.21	4.22
	CIFAR100	Yes	79.09	1.16	98.50	0.13

the perturbation budget is set to $\epsilon = \frac{4}{255}$. To make these subtle perturbations more visible in the figure, we amplify their intensity by a factor of 20 when visualizing the trigger patterns. In contrast, DBA employs an explicit and visually recognizable trigger patch placed in the top-left corner of the image. Neurotoxin, on the other hand, is designed to improve backdoor persistence rather than introduce a new trigger-generation mechanism. Therefore, following its original formulation, we use the BadNets trigger (Gu et al., 2017) for Neurotoxin by placing a cross-shaped pattern in the top-left corner of the image. Furthermore, we also provide the Visualization of **Backdoor Defenses** in Figure 11. We use Grad-CAM (Selvaraju et al., 2017) to visualize the regions that contribute most to the model prediction on poisoned samples. The global model defended by baseline methods still assigns strong attention to the trigger region. In contrast, CAS significantly reduces the model's attention to the trigger area, indicating the effectiveness of CAS.

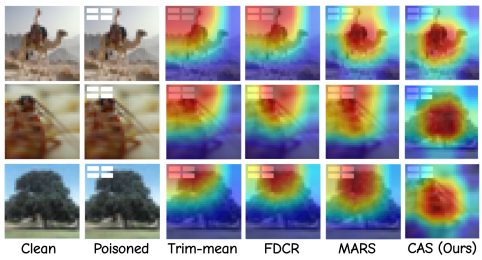


Figure 11: Comparison of heatmap for CAS and baseline defenses under DBA Attack

E Resilience to Adaptive Attack

Resilience to Adaptive Attackers. To further assess the robustness of CAS, we assume that attackers have prior knowledge that the central server employs CAS as the defense mechanism. In this case, a malicious client can craft adaptive strategies specifically designed to bypass CAS. Since CAS captures the backdoor signals through relatively high channel activation magnitude of malicious models, a straightforward approach for executing an adaptive attack is to introduce a regularization term to minimize the magnitude discrepancy between any class n and m for each j-th channel. Formally, the adaptive attackers' optimization objective is defined as

$$\min_{\boldsymbol{w}_t, \delta(x)} \mathcal{L}\left(\boldsymbol{w}_t, \mathcal{D}_{\hat{k}}\right) + \lambda \sum_{j}^{C_{\text{out}}^l} \sum_{n=1}^{N} \mathcal{P}_k^l\left(n, j\right) \log \mathcal{P}_k^l\left(n, j\right) \tag{12}$$

where λ controls the regularization strength. To thoroughly investigate the impact of adaptive attackers, we vary both the attack scale $M \in \{20\%, 70\%\}$ and adaptive strength $\lambda \in \{1, 0.8, 0.6, 0.4\}$. As shown in Table 4, CAS generally remains effective across different adaptive strengths and attack types, consistently reducing the ASR to around 1% under the standard 20% attack scale. This is because the adaptive regularization introduces excessive consistency in the channel activation magnitudes of malicious models, which in turn makes their activation patterns more similar to one another and therefore still distinguishable by our clustering-based detection mechanism. When malicious clients constitute the majority and simultaneously launch strong adaptive attacks (e.g., $M = 70\%$ and $\lambda = 1$), CAS becomes less effective against several attacks, with the TPR dropping to near 0% in some cases. However, such an attack setting is highly stringent in practice, as it requires the adversary to control a majority of participating clients while also applying strong CAS-aware regularization. More importantly, as the adaptive strength decreases, CAS rapidly recovers its detection capability; at $\lambda = 0.4$, it achieves nearly 100% TPR with close-to-zero FPR across all attacks, even under the 70% malicious-client setting.

Client Class Running-mean Authentication. CAS detects malicious clients based on the uploaded class-wise channel running means. Under a stronger threat model, however, malicious clients may deviate from the prescribed local training procedure and decouple the uploaded statistics from the actual poisoned model update. For example, an attacker may train a surrogate model using only clean samples and upload benign-looking statistics $\{\{\mu_k^{n,l,j}\}_{n\in N}\}_{j=1}^{C_{\text{out}}^l}$, even though the submitted local model is trained with poisoned data. In this case, the uploaded statistics no longer faithfully characterize the channel activations of the submitted model. Although trusted execution environments can enforce the prescribed client-side computation (Mo et al., 2021), they require additional system support. Instead, we exploit an intrinsic consistency relation between the uploaded class-wise running means and the running statistics stored in the subsequent batch-normalization layer. Specifically, the running mean $\mathrm{u}_k^{l,j}$ of channel j can be reconstructed as

$$\mathrm{u}_k^{l,j} = \sum_{n=1}^{N} \mu^{n,l,j}, \tag{13}$$

where $\mu^{n,l,j}$ is computed according to Eq. 3. The server can therefore authenticate the uploaded class-wise statistics by checking whether they recover the corresponding batch-normalization running mean, and reject clients that fail this consistency test. However, a practical complication is that, in many neural architectures, the activation function (e.g., ReLU) is applied after batch normalization. In this case, the uploaded post-activation class-wise statistics are not directly aligned with the pre-activation statistics maintained by batch normalization, preventing the above consistency check. To resolve this mismatch, we swap the order of the activation and normalization operations so that batch normalization is applied to the same activated features used to compute the class-wise running means. The corresponding implementation is shown below:

```python
class BasicBlock(nn.Module):
    def __init__(self, in_planes, planes, stride=1):
        super(BasicBlock, self).__init__()

        ...

    ### normal forward pass ###
    # def forward(self, x):
    #     out = self.relu(self.bn1(self.conv1(x)))
    #     out = self.bn2(self.conv2(out))
    #     out += self.shortcut(x)
    #     out = self.relu(out)
    #     return out

    ##### Enable Class Running-mean Authentication: Activation first then BatchNorm ####
    def forward(self, x):
        out = self.bn1(self.relu(self.conv1(x)))
        out = self.relu(self.conv2(out))
        out = out + self.shortcut(x)
        out = self.bn2(out)
        return out
```

Table 5 reports the performance under this architectural modification. The results show that CAS retains strong defense performance across all evaluated attacks under both pretrained and randomly

Table 6: Performance of (ACC/ASR) in **cross-device** federated learning setting

Defenses	BadPFL		IBA		DBA		Neurotoxin		**Average**	
	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
CIFAR-100										
FedAvg	**$77.80_{0.96}$**	$87.29_{5.59}$	**$77.95_{0.76}$**	$93.71_{2.13}$	**$77.38_{0.71}$**	$78.16_{2.80}$	**$77.45_{1.03}$**	$89.48_{1.04}$	**77.64**	87.16
MARS	$76.92_{0.96}$	$67.52_{12.24}$	$76.74_{0.96}$	$45.51_{35.39}$	$76.80_{0.71}$	$65.82_{3.05}$	$76.74_{0.86}$	$68.81_{5.62}$	76.80	61.92
FDCR	$76.96_{0.99}$	$81.85_{4.77}$	$77.11_{0.99}$	$88.01_{2.89}$	$\underline{77.25_{0.92}}$	$69.08_{6.07}$	$76.81_{0.78}$	$76.44_{7.31}$	77.03	78.84
Trim-Mean	$\underline{77.31_{0.75}}$	$83.92_{5.13}$	$\underline{77.61_{0.92}}$	$93.62_{1.68}$	$77.23_{0.89}$	$86.09_{1.01}$	$\underline{77.34_{1.17}}$	$89.99_{0.93}$	$\underline{77.37}$	88.41
Multi-Krum	$76.43_{1.26}$	$90.78_{5.46}$	$76.36_{0.97}$	$89.55_{1.26}$	$76.24_{1.41}$	$88.34_{0.17}$	$76.34_{1.32}$	$89.63_{1.35}$	76.34	89.57
Lockdown	$71.22_{1.08}$	$66.73_{7.87}$	$70.88_{1.35}$	$80.96_{12.22}$	$71.58_{1.08}$	$\underline{57.56_{16.66}}$	$71.72_{0.79}$	$\underline{66.71_{5.31}}$	71.35	67.99
AlignIns	$76.90_{0.89}$	$\underline{62.33_{24.10}}$	$76.69_{0.92}$	$\underline{10.42_{18.57}}$	$77.07_{1.06}$	$78.42_{2.62}$	$76.89_{1.04}$	$85.58_{1.35}$	76.89	$\underline{59.19}$
CAS	$76.91_{0.89}$	**$1.09_{0.06}$**	$76.80_{0.95}$	**$1.00_{0.11}$**	$76.46_{0.76}$	**$1.27_{0.17}$**	$76.42_{0.81}$	**$1.09_{0.04}$**	76.65	**1.11**
CIFAR-10										
FedAvg	$\underline{93.74_{0.56}}$	$77.53_{4.11}$	$\underline{93.72_{0.54}}$	$74.71_{10.25}$	$93.61_{0.59}$	$77.94_{1.96}$	$\underline{93.95_{0.57}}$	$83.44_{0.86}$	$\underline{93.75}$	78.41
MARS	$93.41_{0.66}$	$52.02_{6.41}$	$93.47_{0.57}$	$54.02_{7.81}$	$93.29_{0.52}$	$\underline{51.20_{7.00}}$	$93.63_{0.55}$	$62.84_{14.35}$	93.45	55.02
FDCR	$93.63_{0.59}$	$58.93_{3.12}$	$93.72_{0.57}$	$66.81_{8.69}$	$\underline{93.73_{0.67}}$	$78.10_{1.26}$	$93.76_{0.67}$	$77.94_{3.78}$	93.71	70.45
Trim-Mean	$93.43_{0.63}$	$71.52_{9.83}$	$93.64_{0.51}$	$83.66_{5.92}$	$93.54_{0.64}$	$84.07_{0.90}$	$93.58_{0.61}$	$84.64_{0.81}$	93.55	80.97
Multi-Krum	**$94.25_{0.67}$**	$82.62_{9.08}$	**$94.14_{0.49}$**	$88.73_{5.41}$	**$94.19_{0.37}$**	$88.42_{0.75}$	**$94.26_{0.45}$**	$89.81_{0.71}$	**94.21**	87.40
Lockdown	$92.18_{0.32}$	$48.67_{12.05}$	$92.37_{0.27}$	$66.77_{18.39}$	$92.36_{0.34}$	$61.32_{19.42}$	$92.60_{0.42}$	$78.71_{0.61}$	92.38	63.87
AlignIns	$93.57_{0.67}$	**$10.05_{0.92}$**	$93.52_{0.65}$	**$10.26_{0.95}$**	$93.47_{0.61}$	$71.31_{4.63}$	$93.44_{0.78}$	**$9.98_{0.10}$**	93.50	$\underline{25.40}$
CAS	$93.71_{0.60}$	$\underline{15.47_{5.41}}$	$93.72_{0.66}$	$\underline{10.35_{0.90}}$	$93.59_{0.60}$	**$10.12_{0.15}$**	$93.60_{0.63}$	$\underline{10.26_{0.26}}$	93.66	**11.55**
SVHN										
FedAvg	$95.82_{0.15}$	$72.41_{8.45}$	$95.85_{0.17}$	$69.19_{4.34}$	$95.84_{0.07}$	$65.23_{3.71}$	$95.84_{0.10}$	$77.10_{0.80}$	95.84	70.98
MARS	$95.83_{0.10}$	$39.54_{8.99}$	$95.87_{0.05}$	$43.58_{13.60}$	$95.87_{0.07}$	$\underline{38.59_{8.28}}$	$95.84_{0.12}$	$\underline{51.12_{28.69}}$	95.85	$\underline{43.21}$
FDCR	$95.95_{0.20}$	$\underline{30.42_{9.38}}$	$95.86_{0.16}$	$36.70_{17.24}$	$95.91_{0.09}$	$64.26_{8.09}$	$95.80_{0.14}$	$70.05_{11.85}$	95.88	50.36
Trim-Mean	$95.73_{0.14}$	$66.25_{2.10}$	$95.83_{0.10}$	$62.93_{2.84}$	$95.86_{0.10}$	$68.42_{4.25}$	$95.77_{0.06}$	$74.47_{2.23}$	95.80	68.02
Multi-Krum	$\underline{96.65_{0.08}}$	$82.43_{13.26}$	**$96.74_{0.04}$**	$93.74_{1.98}$	$\underline{96.71_{0.05}}$	$88.89_{6.76}$	$\underline{96.71_{0.04}}$	$93.18_{0.52}$	$\underline{96.70}$	89.56
Lockdown	**$96.74_{0.04}$**	$51.70_{4.73}$	$\underline{96.68_{0.04}}$	$71.96_{28.79}$	**$96.84_{0.06}$**	$86.77_{3.24}$	**$96.72_{0.04}$**	$87.31_{4.64}$	**96.75**	74.44
AlignIns	$95.68_{0.12}$	$39.97_{24.57}$	$95.93_{0.17}$	**$12.57_{2.18}$**	$95.69_{0.10}$	$51.58_{6.37}$	$95.80_{0.10}$	$69.44_{6.83}$	95.78	43.39
CAS	$95.89_{0.19}$	**$17.05_{3.72}$**	$95.90_{0.14}$	$\underline{13.24_{2.81}}$	$95.86_{0.13}$	**$7.11_{0.21}$**	$95.93_{0.09}$	**$7.07_{0.07}$**	95.89	**11.12**
Tiny ImageNet										
FedAvg	**$54.10_{1.35}$**	$91.25_{1.25}$	**$54.28_{0.99}$**	$92.45_{1.63}$	**$54.22_{1.16}$**	$82.61_{2.18}$	**$54.16_{0.80}$**	$88.64_{0.10}$	**54.19**	88.74
MARS	$53.17_{0.88}$	$82.42_{14.49}$	$53.31_{1.82}$	$74.20_{12.06}$	$53.45_{1.36}$	$37.66_{8.05}$	$53.56_{1.16}$	$52.81_{8.76}$	53.37	61.77
FDCR	$53.62_{1.16}$	$79.71_{2.01}$	$53.57_{0.70}$	$72.81_{10.16}$	$53.23_{0.89}$	$75.80_{4.41}$	$\underline{53.80_{1.31}}$	$64.91_{11.34}$	53.55	73.31
Trim-Mean	$\underline{53.97_{0.44}}$	$92.19_{0.84}$	$\underline{54.06_{1.03}}$	$95.13_{1.69}$	$53.45_{1.10}$	$85.13_{2.45}$	$53.71_{1.14}$	$88.68_{0.44}$	$\underline{53.80}$	90.28
Multi-Krum	$52.98_{1.30}$	$94.36_{3.53}$	$53.04_{1.07}$	$91.26_{3.47}$	$52.96_{1.39}$	$86.15_{5.07}$	$53.24_{0.79}$	$90.90_{0.35}$	53.06	90.67
Lockdown	$49.58_{0.75}$	$\underline{63.48_{9.69}}$	$49.73_{0.76}$	$\underline{39.11_{15.83}}$	$49.94_{0.55}$	$\underline{11.90_{11.00}}$	$49.62_{1.10}$	$\underline{45.10_{6.46}}$	49.72	$\underline{39.90}$
AlignIns	$53.13_{0.94}$	$63.82_{42.70}$	$52.87_{1.43}$	$84.54_{8.32}$	$\underline{53.47_{0.88}}$	$45.92_{8.24}$	$53.52_{1.07}$	$68.30_{11.73}$	53.25	65.65
CAS	$53.01_{0.96}$	**$0.29_{0.10}$**	$53.12_{1.15}$	**$0.66_{0.40}$**	$53.39_{1.11}$	**$0.54_{0.10}$**	$53.19_{1.06}$	**$0.58_{0.03}$**	53.18	**0.52**

initialized backbones, indicating that the proposed authentication mechanism can be incorporated without substantially compromising its effectiveness.

F Additional Experiments

F.1 Cross-device Federated Learning.

As shown in Table 6 and Table 7, existing defenses generally exhibit degraded robustness under the cross-device FL setting. In particular, update-statistics-based defenses such as MARS, FDCR, Trim-Mean, and Multi-Krum fail to consistently suppress backdoor attacks across different datasets and attack types. For example, Trim-Mean achieves average ASRs of 88.41%, 80.97%, 68.02%, and 90.28% on CIFAR-100, CIFAR-10, SVHN, and Tiny ImageNet, respectively, which are comparable to the corresponding FedAvg results of 87.16%, 78.41%, 70.98%, and 88.74%. MARS provides stronger attack mitigation than these aggregation-based defenses in some settings, but still retains average ASRs ranging from 43.21% to 61.92% across the four datasets. Similarly, Lockdown reduces the average ASR to 67.99%, 63.87%, 74.44%, and 39.90%, respectively, but this improvement comes with a noticeable degradation in main-task accuracy. For example, on Tiny ImageNet, its average ACC decreases from 54.19% under FedAvg to 49.72%. AlignIns achieves stronger attack suppression on CIFAR-10, reducing the average ASR to 25.40%, but its performance varies considerably across datasets, with average ASRs of 59.19%, 43.39%, and 65.65% on CIFAR-100, SVHN, and Tiny ImageNet, respectively. Its detection results also reveal relatively high false-positive rates, reaching 25.05%, 22.19%, 22.80%, and 21.20% on the four datasets. In contrast, CAS consistently suppresses all four attacks across the evaluated datasets, reducing the average ASR to only 1.11%, 11.55%, 11.12%, and 0.52% on CIFAR-100, CIFAR-10, SVHN, and Tiny ImageNet, respectively. These values are close to the corresponding random-guess levels of 1%, 10%, 10%, and 0.5%, while CAS maintains main-task accuracy comparable to FedAvg. Specifically, the average ACC of CAS differs from FedAvg by less than approximately one percentage point on all four datasets. The de-

Table 7: Comparison of (TPR/FPR) under **cross-device** federated learning setting

Defenses	BadPFL		IBA		DBA		Neurotoxin		**Average**	
	TPR↑	FPR↓	TPR↑	FPR↓	TPR↑	FPR↓	TPR↑	FPR↓	TPR↑	FPR↓
CIFAR-100										
MARS	$68.95_{16.07}$	$13.57_{7.79}$	$80.85_{11.83}$	$9.84_{7.24}$	$53.35_{2.38}$	$20.78_{2.77}$	$62.80_{5.10}$	$12.66_{1.39}$	66.49	14.21
FDCR	$55.60_{5.02}$	$9.30_{3.67}$	$31.30_{6.04}$	$17.24_{3.31}$	$54.35_{4.88}$	$2.61_{1.19}$	$56.45_{5.76}$	$2.97_{1.12}$	49.43	8.03
Multi-Krum	$18.52_{0.80}$	$20.37_{0.20}$	$18.52_{0.80}$	$20.37_{0.20}$	$18.52_{0.80}$	$20.37_{0.20}$	$18.52_{0.80}$	$20.37_{0.20}$	18.52	20.37
AlignIns	$68.20_{29.61}$	$21.74_{8.41}$	$95.25_{4.45}$	$11.67_{2.76}$	$26.90_{0.84}$	$33.97_{0.81}$	$32.85_{2.86}$	$32.81_{1.01}$	55.80	25.05
CAS	$\mathbf{99.10_{0.55}}$	$\mathbf{0.00_{0.00}}$	$\mathbf{99.65_{0.34}}$	$\mathbf{0.00_{0.00}}$	$\mathbf{99.25_{0.75}}$	$\mathbf{0.08_{0.10}}$	$\mathbf{99.35_{0.29}}$	$\mathbf{0.18_{0.08}}$	**99.34**	**0.07**
CIFAR-10										
MARS	$61.12_{7.54}$	$14.12_{6.85}$	$43.75_{2.69}$	$23.03_{4.38}$	$55.37_{5.65}$	$9.91_{5.58}$	$57.37_{18.51}$	$14.06_{9.01}$	54.40	15.28
FDCR	$55.63_{4.68}$	$\mathbf{1.81_{1.31}}$	$51.12_{10.10}$	$4.72_{5.20}$	$40.00_{5.32}$	$7.09_{0.69}$	$41.25_{4.57}$	$6.91_{2.76}$	47.00	5.13
Multi-Krum	$18.52_{0.80}$	$20.37_{0.20}$	$18.52_{0.80}$	$20.37_{0.20}$	$18.52_{0.80}$	$20.37_{0.20}$	$18.52_{0.80}$	$20.37_{0.20}$	18.52	20.37
AlignIns	$\mathbf{99.88_{0.28}}$	$17.13_{4.06}$	$\mathbf{100.00_{0.00}}$	$13.16_{2.36}$	$46.00_{7.94}$	$30.00_{3.90}$	$\mathbf{99.50_{0.28}}$	$28.47_{2.49}$	86.35	22.19
CAS	$93.00_{2.88}$	$1.94_{1.25}$	$97.88_{1.05}$	$\mathbf{1.31_{1.07}}$	$\mathbf{98.00_{0.81}}$	$\mathbf{2.34_{1.18}}$	$97.75_{0.95}$	$\mathbf{2.94_{1.55}}$	**96.66**	**2.13**
SVHN										
MARS	$59.41_{12.06}$	$7.79_{5.63}$	$53.53_{15.41}$	$14.04_{10.74}$	$58.24_{8.74}$	$11.40_{7.79}$	$63.24_{21.03}$	$11.03_{10.37}$	58.61	11.07
FDCR	$79.12_{4.58}$	$1.10_{1.07}$	$69.41_{11.02}$	$1.10_{0.74}$	$44.12_{7.20}$	$0.81_{0.84}$	$42.65_{15.67}$	$5.66_{5.03}$	58.83	2.17
Multi-Krum	$18.52_{0.80}$	$20.37_{0.20}$	$18.52_{0.80}$	$20.37_{0.20}$	$18.52_{0.80}$	$20.37_{0.20}$	$18.52_{0.80}$	$20.37_{0.20}$	18.52	20.37
AlignIns	$60.59_{30.06}$	$24.34_{8.51}$	$98.53_{2.55}$	$9.78_{0.76}$	$47.94_{5.16}$	$29.34_{2.29}$	$48.24_{5.83}$	$27.72_{2.11}$	63.83	22.80
CAS	$\mathbf{95.88_{4.58}}$	$\mathbf{1.03_{1.12}}$	$\mathbf{98.82_{1.92}}$	$\mathbf{0.59_{1.12}}$	$\mathbf{98.24_{2.63}}$	$\mathbf{0.59_{1.32}}$	$\mathbf{99.71_{0.66}}$	$\mathbf{0.44_{0.80}}$	**98.16**	**0.66**
Tiny ImageNet										
MARS	$55.65_{23.11}$	$23.33_{10.69}$	$65.09_{16.39}$	$19.81_{6.90}$	$66.30_{4.89}$	$11.48_{2.75}$	$73.24_{3.67}$	$9.51_{2.43}$	65.07	16.03
FDCR	$47.22_{6.82}$	$17.41_{5.14}$	$51.76_{13.60}$	$13.80_{6.64}$	$44.63_{6.06}$	$17.13_{6.91}$	$67.87_{6.19}$	$7.92_{5.59}$	52.87	14.07
Multi-Krum	$7.59_{10.99}$	$23.10_{2.75}$	$3.43_{1.53}$	$24.14_{0.38}$	$0.28_{0.48}$	$24.93_{0.12}$	$0.28_{0.28}$	$24.93_{0.07}$	2.90	24.28
AlignIns	$66.76_{25.14}$	$19.86_{11.19}$	$57.31_{14.03}$	$23.75_{5.50}$	$73.24_{3.77}$	$18.59_{3.47}$	$63.61_{12.92}$	$22.59_{4.10}$	65.23	21.20
CAS	$\mathbf{99.91_{0.16}}$	$\mathbf{0.14_{0.07}}$	$\mathbf{100.00_{0.00}}$	$\mathbf{0.02_{0.04}}$	$\mathbf{99.91_{0.16}}$	$\mathbf{0.14_{0.07}}$	$\mathbf{99.91_{0.16}}$	$\mathbf{0.16_{0.04}}$	**99.93**	**0.12**

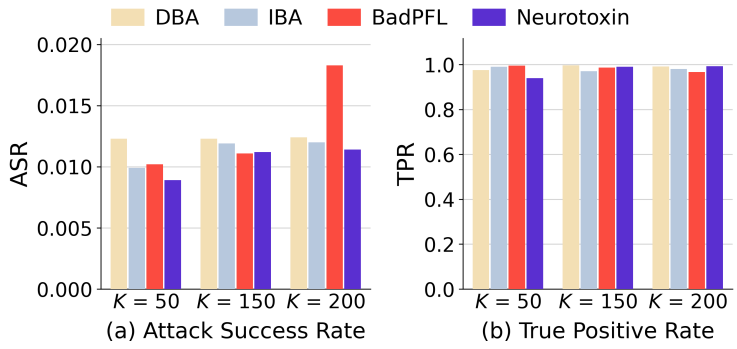


Figure 12: Detection performance under different client size K on CIFAR100.

tection results further confirm this robustness: CAS achieves average TPRs of 99.34%, 96.66%, 98.16%, and 99.93%, with corresponding FPRs of only 0.07%, 2.13%, 0.66%, and 0.12%. These results demonstrate that CAS can reliably identify malicious clients and suppress backdoor attacks in cross-device FL without substantially sacrificing clean-task performance or incorrectly excluding benign clients.

F.2 Results on additional federated learning setups

Impact of Clients Size. We evaluate the scalability of client size by varying the total number of clients K under the cross-device federated learning setting, testing $K = 50$, $K = 150$, and $K = 200$. As shown in Figure 12, CAS consistently achieves superior detection performance across all client size. Similar to smaller poisoning rate, larger client size can potentially dissolve the abnormality signal of malicious clients. Nevertheless, we observe that, as the client size increases, only slight performance drop can be observed in terms of attack success rate, which confirms the robustness of CAS.

Impact of Local Epochs. We evaluate the performance of different defense methods under varying numbers of local training epochs against BadPFL attack. CAS consistently achieves the strongest defense performance on both CIFAR-10 and CIFAR-100, while the baseline defenses generally exhibit decreasing TPR and increasing FPR as the number of local epochs grows. With more local

Table 8: Detection performance against BadPFL attack under **different local epochs** e

Defenses	$e=1$				$e=3$				$e=5$			
	ACC↑	ASR↓	TPR↑	FPR↓	ACC↑	ASR↓	TPR↑	FPR↓	ACC↑	ASR↓	TPR↑	FPR↓
CIFAR100												
MARS	79.11	64.38	52.50	20.25	79.49	48.95	54.00	27.50	79.49	37.65	53.50	25.12
FDCR	**79.91**	34.38	74.00	1.25	**80.02**	63.59	31.50	27.62	**82.18**	66.62	25.50	23.50
Multi-Krum	78.10	86.90	1.24	77.52	79.27	58.41	21.50	19.63	79.33	45.07	27.00	18.25
AlignIns	78.84	22.66	89.00	10.62	79.36	59.33	43.50	27.50	79.30	39.39	43.50	26.75
CAS	78.98	**1.22**	**97.50**	**0.00**	79.61	**1.25**	**99.50**	**0.00**	79.74	**1.49**	**99.50**	**0.00**
CIFAR10												
MARS	95.81	53.84	65.00	11.56	95.69	57.24	35.63	34.69	95.46	50.66	35.00	36.56
FDCR	95.56	76.99	25.62	20.94	95.73	55.20	36.25	15.94	95.47	35.84	58.75	18.28
Multi-Krum	95.75	11.27	100.0	0.00	95.77	60.61	18.12	20.47	95.36	48.05	15.62	21.09
AlignIns	95.68	**9.49**	**100.0**	2.34	95.68	82.67	26.87	35.63	95.48	51.33	23.75	30.78
CAS	**95.91**	11.38	96.88	**0.00**	**95.78**	**9.76**	**95.00**	**0.78**	**95.54**	**14.72**	**90.00**	**0.41**

Table 9: Detection performance under **unsynchronized Attackers** with different attack chance r.

Defenses	BadPFL		IBA		DBA		Neurotoxin	
	ASR↓	FPR↓	ASR↓	FPR↓	ASR↓	FPR↓	ASR↓	FPR↓
$r=20\%$								
MARS	4.76	21.37	3.64	28.38	2.32	27.75	21.10	**22.38**
FDCR	3.07	19.13	1.48	24.25	1.22	23.13	1.07	22.63
AlignIns	16.36	29.38	2.57	30.25	6.70	33.12	18.94	29.13
Trim-Mean	12.68	−−	3.78	−−	1.37	−−	2.93	−−
CAS	**1.37**	**16.00**	**1.24**	**12.25**	**1.05**	**18.25**	**1.07**	23.25
$r=40\%$								
MARS	38.27	21.75	11.71	22.00	7.51	22.00	43.87	32.37
FDCR	2.19	10.87	3.07	13.25	4.29	12.88	5.80	17.13
AlignIns	26.13	27.62	1.70	24.62	15.55	27.00	33.69	27.75
Trim-Mean	29.50	−−	14.68	−−	8.57	−−	35.35	−−
CAS	**1.39**	**6.12**	**1.31**	**6.38**	**1.13**	**8.00**	**1.04**	**15.88**
$r=60\%$								
MARS	41.04	23.75	2.20	8.62	26.56	16.88	53.21	23.87
FDCR	36.06	13.00	40.54	16.75	20.23	6.25	30.45	**4.75**
AlignIns	19.91	24.00	1.30	21.25	31.46	29.25	44.20	26.62
Trim-Mean	46.79	−−	42.71	−−	28.49	−−	59.70	−−
CAS	**1.43**	**0.25**	**1.33**	**1.12**	**1.08**	**2.38**	**1.07**	11.00

training, client models become increasingly overfit to their local data distributions, thereby amplifying inter-client update heterogeneity. Since most baseline defenses rely primarily on discrepancies in model parameters or gradients to distinguish malicious clients from benign ones, this increased heterogeneity can obscure attack-specific signals and distort their detection criteria, causing benign clients to be misclassified as malicious and malicious clients to evade detection.

Impact of Unsynchronized Attackers. In the main experiments reported in Sec. 3, malicious clients are assumed to be synchronized, i.e., they launch backdoor attacks whenever they participate in training. To examine a more practical unsynchronized setting, we further evaluate each defense under different attack probabilities $r \in 20\%, 40\%, 60\%$, where each malicious client independently launches a backdoor attack with probability r. Since a malicious client may participate without injecting a backdoor in a given round, TPR is no longer directly applicable; therefore, we report ASR and FPR instead. As shown in Table 9, the ASR of most baseline defenses generally increases as r grows, indicating that more frequent backdoor injections make the attacks increasingly difficult to suppress. Moreover, CAS generally achieves substantially lower FPR than the detection-based baselines, particularly at larger r. These results demonstrate that CAS is not reliant on synchronized malicious behavior and can effectively identify and suppress backdoor injections even when attackers participate intermittently and asynchronously.

F.3 Cross-Architecture Transferability.

We further examine the transferability of defense performance across different network architectures. As shown in Table 10, we conduct experiments on different pretrained initial global models, including ResNet-50, DenseNet121 (Huang et al., 2017), and MobileNetV2 (Sandler et al., 2018).

Table 10: Detection performance across **different network architectures** under CIFAR100

Defenses	ResNet-50				DenseNet121				MobileNetV2			
	ACC↑	ASR↓	TPR↑	FPR↓	ACC↑	ASR↓	TPR↑	FPR↓	ACC↑	ASR↓	TPR↑	FPR↓
BadPFL												
FDCR	86.80	**1.66**	97.00	**0.88**	84.34	39.02	76.50	2.25	74.69	88.01	5.00	34.00
Trim-Mean	**87.60**	81.09	–	–	84.41	80.73	–	–	**74.96**	58.54	–	–
Multi-Krum	86.93	35.03	88.00	3.00	83.84	82.16	9.00	22.75	74.65	5.70	86.50	3.38
AlignIns	87.22	81.20	35.50	30.88	84.24	74.81	26.50	29.13	74.61	61.69	35.50	28.38
CAS	86.72	2.15	**97.00**	1.12	**84.77**	**4.16**	**95.50**	**0.63**	74.35	**2.00**	**94.00**	**0.88**
IBA												
FDCR	86.70	1.24	98.50	**0.00**	84.18	75.91	53.00	6.25	74.78	91.28	5.00	33.75
Trim-Mean	**87.41**	75.27	–	–	84.32	78.06	–	–	74.15	66.48	–	–
Multi-Krum	87.12	74.29	67.50	8.13	84.13	78.19	6.00	23.50	74.45	12.04	79.00	5.25
AlignIns	87.21	78.59	31.00	32.75	84.15	68.96	28.00	27.75	**74.68**	68.43	40.50	29.88
CAS	86.73	**1.23**	**100.00**	0.25	**84.80**	**1.75**	**96.00**	**1.00**	74.46	**0.80**	**97.50**	**1.12**
DBA												
FDCR	87.30	80.56	22.50	21.37	**84.32**	59.56	44.00	12.12	74.74	64.21	5.00	27.62
Trim-Mean	**87.53**	76.54	–	–	84.32	71.06	–	–	**75.23**	19.14	–	–
Multi-Krum	87.18	50.51	74.00	6.50	83.45	46.96	1.50	24.62	74.81	1.69	80.50	4.88
AlignIns	87.41	67.91	44.00	25.12	84.31	48.87	32.50	27.12	74.78	7.93	51.00	26.00
CAS	86.63	**1.18**	**94.00**	**5.37**	83.83	**1.92**	**90.50**	**3.87**	74.23	**1.08**	**99.00**	**1.63**

The results, across all attacks and networks on CIFAR100, consistently validate that CAS can generally provide high cross-architecture defenses performance, outperforming baselines by a distinct margin. We also observe that ResNet-50 backbone possesses stronger learning capability than the other two lighter models by showing an average improvement of 10% on main task accuracy. However, lighter models often demonstrate stronger resilience to backdoor injection. Particularly, for MobileNetV2, it generally achieves lower ASR across all attacks. Nevertheless, our method clearly surpasses baselines and successfully enforces backdoor resilience, thereby further demonstrating the efficacy of the proposed CAS framework.

F.4 Results on different federated learning algorithms.

Table 11 reports the performance of various defenses across three federated learning algorithms on CIFAR-100, including FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), and FedDyn (Acar et al., 2021). It can be observed that baseline defenses generally achieve improved robustness under FedDyn, particularly in terms of ASR. This improvement is likely attributed to the dynamic regularization employed by FedDyn, which mitigates local model drift and reduces the heterogeneity among benign client updates. Consequently, the statistical discrepancy between benign clients becomes less pronounced, allowing update-based defenses to more reliably distinguish malicious updates. For example, the ASR of MARS under Neurotoxin decreases from 71.69% with FedAvg to only 2.88% with FedDyn, while FDCR decreases from 67.48% to 15.07%. Nevertheless, such improvements are not consistent across all attacks and detection metrics, as several baseline defenses still exhibit low TPR or high FPR under FedDyn. In contrast, CAS maintains consistently strong defense performance across all three FL algorithms. Under FedAvg and FedProx, CAS suppresses the ASR to approximately 1% across all four attacks while achieving nearly perfect TPR and low FPR. Although its detection performance moderately decreases under FedDyn, particularly for BadPFL and Neurotoxin, CAS still achieves substantially lower ASRs than the competing defenses across all attacks. These results demonstrate that CAS is largely agnostic to the choice of federated optimization algorithm and can generalize effectively beyond the standard FedAvg setting.

F.5 Results on Multi-Attack Scenario

In the main experiments, we assume that malicious clients launch the same type of backdoor attack during training. To evaluate the robustness of each defense when this assumption no longer holds, we further consider a more challenging multi-attack scenario in which different malicious clients simultaneously adopt different attack strategies. Specifically, we divide the malicious clients into four disjoint groups, with each group independently conducting one of BadPFL, IBA, DBA, and Neurotoxin. All experiments in this section are conducted under the cross-device federated learning setting described in Appendix B.

As shown in Table 12, the coexistence of heterogeneous attacks poses a substantial challenge to most baseline defenses. MARS and FDCR remain vulnerable to BadPFL and IBA, resulting in high

Table 11: Detection performance across **different federated learning algorithms** under CIFAR100.

Defenses	FedAvg				FedProx				FedDyn			
	ACC↑	ASR↓	TPR↑	FPR↓	ACC↑	ASR↓	TPR↑	FPR↓	ACC↑	ASR↓	TPR↑	FPR↓
BadPFL												
FDCR	**79.25$_{0.75}$**	76.99$_{10.04}$	18.12$_{2.72}$	30.05$_{4.02}$	**81.31**	33.38	73.12	7.19	75.05	72.97	16.25	46.09
MARS	79.16$_{1.01}$	48.14$_{5.98}$	41.88$_{4.37}$	24.69$_{7.99}$	81.27	86.40	41.88	28.91	75.48	42.21	58.13	20.94
Multi-Krum	79.05$_{0.58}$	76.52$_{5.71}$	15.42$_{3.97}$	21.15$_{0.99}$	81.16	74.43	13.75	21.56	75.37	57.29	8.13	22.97
CAS	79.13$_{0.45}$	**1.40$_{0.21}$**	**100.00$_{0.00}$**	**0.00$_{0.00}$**	81.16	**1.67**	**99.38**	**0.00**	**76.02**	**3.11**	**82.50**	**0.94**
IBA												
FDCR	78.77$_{0.77}$	72.57$_{25.10}$	35.42$_{19.98}$	25.62$_{11.15}$	80.84	88.42	15.00	38.44	75.15	58.47	28.75	37.50
MARS	**78.83$_{0.87}$**	45.90$_{17.97}$	70.42$_{14.66}$	9.74$_{5.11}$	**80.95**	1.85	95.63	1.72	75.44	31.18	55.00	21.09
Multi-Krum	78.83$_{0.67}$	75.54$_{6.23}$	24.79$_{10.54}$	18.80$_{2.63}$	80.75	79.27	15.00	21.25	75.37	56.38	3.75	24.06
CAS	78.70$_{0.80}$	**1.13$_{0.12}$**	**100.00$_{0.00}$**	**0.00$_{0.00}$**	80.87	**1.65**	**100.00**	**0.00**	**75.75**	**1.64**	**96.88**	**0.16**
DBA												
FDCR	**79.61$_{0.91}$**	48.89$_{14.44}$	53.75$_{15.17}$	2.60$_{1.25}$	81.23	56.52	38.75	10.78	75.30	17.41	51.88	27.34
MARS	78.77$_{0.53}$	53.46$_{7.51}$	52.29$_{7.71}$	27.66$_{2.87}$	**81.47**	42.29	41.25	20.62	**75.51**	5.02	61.25	15.16
Multi-Krum	78.63$_{0.71}$	63.97$_{4.65}$	1.46$_{0.72}$	24.64$_{0.18}$	80.73	60.82	1.25	24.69	75.20	41.95	0.00	25.00
CAS	78.80$_{0.55}$	**1.63$_{0.66}$**	**93.75$_{3.31}$**	**0.73$_{0.74}$**	80.92	**1.21**	**95.00**	**0.16**	75.51	**1.35**	**86.88**	**6.72**
Neurotoxin												
FDCR	79.39$_{0.42}$	67.48$_{5.16}$	50.42$_{2.19}$	11.77$_{3.90}$	**81.65**	65.36	52.50	4.69	75.72	15.07	58.13	14.37
MARS	**79.44$_{0.69}$**	71.69$_{0.36}$	41.04$_{1.30}$	26.82$_{0.36}$	80.81	72.16	46.88	31.09	**75.83**	**2.88**	**76.25**	**4.69**
Multi-Krum	78.93$_{0.40}$	82.19$_{1.85}$	0.42$_{0.36}$	24.90$_{0.09}$	80.57	84.06	0.00	25.00	75.43	75.08	0.00	25.00
CAS	78.90$_{0.68}$	**1.34$_{0.33}$**	**95.42$_{2.60}$**	**3.80$_{1.88}$**	81.03	**1.06**	**96.88**	**3.28**	75.80	6.90	71.25	21.09

Table 12: Detection performance under **multi-attack** scenario.

Defenses	ASR↓					ACC↑	TPR↑	FPR↓
	BadPFL	IBA	DBA	Neurotoxin	Average			
CIFAR100								
MARS	73.41	78.62	**1.08**	**1.07**	38.54	77.91	50.50	1.38
FDCR	54.78	56.88	5.91	38.35	38.98	78.28	61.00	0.25
AlignIns	1.30	1.31	23.54	33.27	14.85	78.13	76.50	10.13
Multi-Krum	**1.24**	**1.18**	1.16	1.10	**1.17**	**78.28**	**99.00**	0.25
Lockdown	34.93	28.08	4.43	21.33	22.19	67.55	–	–
CAS	1.31	1.24	1.19	1.13	1.22	78.19	97.00	**0.13**
CIFAR10								
MARS	63.04	68.50	**9.69**	9.72	37.74	94.32	55.50	**2.50**
FDCR	35.57	44.49	47.59	51.32	44.74	**94.63**	50.50	2.88
AlignIns	**12.97**	**13.24**	13.42	18.06	14.42	94.37	**87.00**	13.75
Multi-Krum	43.25	49.67	17.19	45.64	38.94	94.56	14.00	21.50
Lockdown	34.59	28.68	20.09	38.78	30.54	90.12	–	–
CAS	13.52	14.03	11.04	**11.05**	**12.41**	94.61	84.00	4.50

average ASRs on both CIFAR-100 and CIFAR-10. AlignIns achieves relatively low ASR on some attacks, but its performance is less stable across attack types and is accompanied by a noticeably higher FPR. Multi-Krum performs particularly well on CIFAR-100, reducing the average ASR to 1.17% with a TPR of 99.00%, but its robustness degrades significantly on CIFAR-10, where the average ASR increases to 38.94%. In contrast, CAS maintains consistently strong performance across both datasets. On CIFAR-100, CAS achieves an average ASR of only 1.22%, together with a TPR of 97.00% and an FPR of merely 0.13%. On CIFAR-10, although the multi-attack setting is more challenging, CAS still obtains the lowest average ASR of 12.41% among all evaluated defenses while preserving competitive main-task accuracy. These results indicate that CAS can effectively generalize beyond the single-attack assumption and remains robust even when multiple heterogeneous backdoor attacks coexist within the same federated learning process.

F.6 Results on different distance metrics.

In Section 2.3, we argue that Wasserstein distance is less sensitive to the ordering of vector elements and is therefore more suitable for characterizing the distributional discrepancy captured by CAS. To further validate this design choice, we replace the Wasserstein distance in CAS with Euclidean and cosine distance while keeping all other components unchanged. As shown in Table 13, Wasserstein distance consistently provides the strongest overall defense performance, despite a slight drop in main-task accuracy. We attribute this accuracy reduction to the more effective exclusion of malicious clients, whose local datasets still contain a substantial portion of clean samples that would otherwise contribute useful information to global model training.

Table 13: Impact of **distance metric** on CAS.

Attacks	Dist.	ACC↑		ASR↓		TPR↑		FPR↓	
		CIFAR100	CIFAR10	CIFAR100	CIFAR10	CIFAR100	CIFAR10	CIFAR100	CIFAR10
BadPFL	Cos.	**79.61**	95.37	3.83	64.42	93.75	41.88	0.94	27.66
	Euc.	79.02	**95.41**	**1.18**	33.83	98.75	82.50	**0.00**	**0.94**
	Wass.	79.13	95.06	1.40	**12.45**	**100.00**	**95.00**	**0.00**	2.86
DBA	Cos.	**80.21**	**95.33**	46.96	54.66	42.50	18.75	15.31	24.84
	Euc.	79.53	94.61	3.30	10.21	81.25	85.00	2.81	**1.67**
	Wass.	78.80	94.94	**1.63**	**10.02**	**93.75**	**97.08**	**0.73**	5.78
IBA	Cos.	79.38	**95.40**	2.25	72.11	93.75	45.62	0.63	25.78
	Euc.	**79.45**	95.38	1.82	24.77	98.75	83.75	**0.00**	2.50
	Wass.	78.70	95.15	**1.13**	**9.64**	**100.00**	**95.83**	**0.00**	**1.98**
Neurotoxin	Cos.	79.62	95.17	69.08	80.53	52.50	21.88	24.69	25.31
	Euc.	**79.93**	**95.33**	40.75	29.77	73.12	80.00	**3.12**	4.22
	Wass.	78.90	95.05	**1.34**	**10.26**	**95.42**	**95.21**	3.80	**1.08**

G Description of Evaluated Attacks and Defenses

G.1 Attacks

BadPFL(Fan et al., 2025). BadPFL is a backdoor attack designed for personalized federated learning. Unlike conventional attacks that rely on manually designed trigger patterns, BadPFL constructs triggers from natural data features. This design allows the backdoor behavior to be better preserved during personalization, since the trigger is embedded into features that are already learned by the model. Specifically, BadPFL applies two separate trigger masks. The first mask aims to construct a semantic shortcut between the sample and the target label while the second mask introduces disruptive noise to corrupt the feature information of the sample to the true label. For the hyperparameter setting specific to BadPFL in our evaluation, we set $\epsilon = \sigma = \frac{4}{255}$ and use the Adam optimizer with a learning rate of 0.01 to train an autoencoder-based generative network for 30 steps.

IBA(Nguyen et al., 2023a). IBA (Irreversible Backdoor Attack) aims to implant a durable and stealthy backdoor into the global model. Instead of using a fixed visible pattern, IBA jointly learns an instance-specific perturbation and the corresponding poisoned objective. The learned perturbation is designed to be visually inconspicuous while still forcing triggered samples toward the target label. Similar to BadPFL, IBA trains a generator to produce instance-wise perturbation as the backdoor trigger and thus we use the same generator from BadPFL for IBA.

Neurotoxin(Zhang et al., 2022). Neurotoxin is a durable federated backdoor attack that improves the persistence of the injected backdoor. Its main idea is to update a subset of model parameters that change less during benign training. Since these parameters are less frequently overwritten by subsequent benign updates, the implanted backdoor can remain effective for more communication rounds after the malicious clients stop participating. In our experiments, Neurotoxin is implemented by applying the trigger to poisoned local samples and restricting the malicious update to the selected low-change parameters. We set top-K ratio to 2% for the fraction of parameter malicious clients will be updating.

DBA(Xie et al., 2019). DBA (Distributed Backdoor Attacks) is a distributed backdoor attack that exploits the distributed nature of federated learning. Instead of assigning the complete trigger to each malicious client, DBA decomposes the global trigger into several local sub-triggers. Each malicious client trains on samples stamped with one sub-trigger, while the aggregated global model learns the complete backdoor behavior after combining their updates. In our experiments, we set the number of trigger pieces to be four and distribute them equally among malicious clients. For each trigger piece, we set the trigger pixels to 1.

G.2 Defenses

AlignIns(Xu et al., 2025). AlignIns detects malicious clients by inspecting the directional alignment of their model updates. Specifically, it first evaluates how each client update aligns with the overall update direction and then examines the sign consistency of important parameters with the principal sign shared across clients. Updates exhibiting abnormally strong or inconsistent alignment patterns are regarded as suspicious and filtered out before aggregation.

Multi-Krum(Blanchard et al., 2017). Multi-Krum is a Byzantine-robust aggregation method that identifies client updates according to their pairwise Euclidean distances. For each client, it computes a Krum score as the sum of distances to its closest neighboring updates, such that updates lying close to the majority receive lower scores. Multi-Krum then selects multiple updates with the smallest Krum scores and averages them to form the global update, thereby reducing the influence of outlying or malicious updates.

MARS(Wan et al., 2025). MARS (malignity-aware federated backdoor defense) measures the malicious tendency of neurons through backdoor energy, which is used to identify neurons that are strongly associated with backdoor behavior. MARS then extracts the most prominent backdoor-energy values to form a compact representation for each client model. Finally, it applies Wasserstein-distance-based clustering to distinguish malicious models from benign ones before aggregation. Following the default hyperparameter settings in the paper, we set top-k neuron ratio to 0.05.

FDCR(Huang et al., 2024). FDCR (Fisher Discrepancy Cluster and Rescale) is a federated backdoor defense designed for heterogeneous federated learning. It analyzes parameter disparities among local models and distinguishes parameters according to their contribution to benign and malicious behavior via the Fisher information matrix. By emphasizing more reliable parameters while reducing the effect of suspicious or less useful ones, FDCR aims to improve robustness under both data heterogeneity and backdoor attacks. In our experiment, we use all local batches to calculate the Fisher-diagonal estimation.

Lockdown(Huang et al., 2023). Lockdown is a pruning-based defense that uses isolated subspace training to mitigate backdoor injection in federated learning. Instead of allowing every client to update the entire model, Lockdown assigns clients to isolated parameter subspaces. It then performs subspace pruning and recovery to separate suspicious parameters from benign ones. Finally, a quorum-consensus mechanism is used to remove malicious or dummy parameters and recover a cleaner global model. In our experiments, we set overall density of each client's binary mask to 0.25 and use ERK to initialize the client parameter sparsity mask.

Trimmed-Mean(Yin et al., 2018). Trimmed-Mean is a coordinate-wise robust aggregation method. For each model parameter, the server sorts the corresponding values received from all clients and removes a fixed fraction of the largest and smallest values. The remaining values are averaged to form the global update. By discarding extreme values, Trimmed-Mean reduces the influence of outlier updates produced by malicious clients.

FedAvg(McMahan et al., 2017). FedAvg is the standard aggregation rule in federated learning. In each communication round, selected clients train their local models using private data and upload their model updates to the server. The server then aggregates these updates by taking a weighted average, typically using the number of local samples as the weight. Since FedAvg does not include any explicit anomaly detection or robust aggregation mechanism, it serves as the undefended baseline in our evaluation.

H Evaluation Metrics

We evaluate defense performance using four metrics: main task accuracy (ACC), attack success rate (ASR), true positive rate (TPR), and false positive rate (FPR). These metrics jointly measure whether a defense can preserve clean-model utility, suppress the backdoor effect, and accurately distinguish malicious clients from benign ones.

Main Task Accuracy (ACC). ACC measures the classification accuracy of the global model on clean test samples. It reflects whether the defense preserves the model's normal prediction performance after filtering or modifying client updates. A higher ACC indicates that the defense introduces less damage to the main learning task.

Attack Success Rate (ASR). ASR measures the percentage of triggered test samples that are misclassified into the attacker-specified target label. In our experiment, we take the averaged ASR over all rounds as the reported ASR. It reflects the effectiveness of the implanted backdoor after defense. A lower ASR indicates stronger defense capability against backdoor attacks.

True Positive Rate (TPR). TPR measures the proportion of malicious clients that are correctly detected by the defense. It evaluates the defense's ability to identify poisoned updates before aggregation. A higher TPR indicates that more malicious clients are successfully removed. Formally, TPR is defined as

$$\mathrm{TPR} = \frac{N_{\mathrm{mal}}^{\mathrm{detected}}}{N_{\mathrm{mal}}}, \tag{14}$$

where N_{mal} denotes the total number of malicious clients, and $N_{\mathrm{mal}}^{\mathrm{detected}}$ denotes the number of malicious clients correctly identified by the defense.

False Positive Rate (FPR). FPR measures the proportion of benign clients that are incorrectly classified as malicious. It reflects whether the defense mistakenly removes useful benign updates. A lower FPR is desirable because falsely excluding benign clients can reduce the quality of global aggregation and degrade ACC. Formally, FPR is defined as

$$\mathrm{FPR} = \frac{N_{\mathrm{ben}}^{\mathrm{misclassified}}}{N_{\mathrm{ben}}}, \tag{15}$$

where N_{ben} denotes the total number of benign clients, and $N_{\mathrm{ben}}^{\mathrm{misclassified}}$ denotes the number of benign clients incorrectly identified as malicious.

I LIMITATION OF CAS

While this paper focuses on visual recognition task with pretrained convolutional backbones, the effectiveness of CAS on other model families, such as vision transformers and larger foundation models, remains less explored. Also, CAS only explores its effectiveness in visual recognition task. Other tasks (e.g., NLP, object recognition, and segmentation) remain unexplored, which will be considered in future work. Extending CAS to broader model architectures is an important direction for future work.

J ALGORITHM OVERVIEW

Algorithm 1 Channel Abnormality Inspector

Require: Set of selected clients in the round t: $\mathcal{K}_t$; Set of corresponding local models: $\{\boldsymbol{w}_t^k, k \in \mathcal{K}\}$; Selected Layer l. ▷ We omit round index t for simplicity

Ensure: Aggregated global model: $\boldsymbol{w}$;

for $k \in \mathcal{K}$ **do**
  Client k computes $\boldsymbol{w}_t^k \leftarrow \arg\min_{\boldsymbol{w}_t} \mathcal{L}\left(\boldsymbol{w}_t, \mathcal{D}_k\right)$ while obtains $\{\{\mu_k^{n,l,j}\}_{n\in N}\}_{j=1}^{C_{\text{out}}^l}$;
  Client k uploads $\boldsymbol{w}_t^k$ and $\{\{\mu_k^{n,l,j}\}_{n\in N}\}_{j=1}^{C_{\text{out}}^l}$ to the server;
end for
for $k \in \mathcal{K}$ **do**
  for $j \in C_{\text{out}}^l$ **do**
    $z_k^{l,j} = -\sum_{n=1}^{N} \mathcal{P}_k^l\left(n, j\right) \log \mathcal{P}_k^l\left(n, j\right),$ ▷ Calculate activation consistency entropy via Eq.7
  end for
  $\mathbf{z}_k^l \leftarrow \left[z_k^{l,j}, \ldots, z_k^{l,C_{\text{out}}^l}\right]^\top$ ▷ Obtain CAS score of each client
end for
$\mathcal{K}_1, \mathcal{K}_2 \leftarrow \text{WassersteinCluster}\left(\{\mathbf{z}_k^l\}_{k\in\mathcal{K}_t}\right).$
if CenterNorm $(\mathcal{K}_1) <$ CenterNorm $(\mathcal{K}_2)$ **then**; ▷ Preserve the cluster with larger central norm
  $\boldsymbol{w} \leftarrow$ Aggregate $(\mathcal{K}_2)$ ▷ Global model aggregation
else
  $\boldsymbol{w} \leftarrow$ Aggregate $(\mathcal{K}_1)$
end if
return $\boldsymbol{w}$