

# Coward: Toward Practical Proactive Federated Backdoor Defense via Collision-based Watermark

Wenjie Li, Siying Gu, Yiming Li, Kangjie Chen, Zhili Chen, Tianwei Zhang,  
Shu-Tao Xia, and Dacheng Tao, *Fellow, IEEE*.

**Abstract**—Backdoor detection is currently the mainstream defense against backdoor attacks in federated learning (FL), where malicious clients upload poisoned updates that compromise the global model and undermine the reliability of FL deployments. Existing backdoor detection techniques fall into two categories, including passive and proactive ones, depending on whether the server proactively modifies the global model. However, both have inherent limitations in practice: passive defenses are vulnerable to common non-i.i.d. data distributions and random participation of FL clients, whereas current proactive defenses suffer inevitable out-of-distribution (OOD) bias because they rely on *backdoor co-existence effects*. To address these issues, we introduce a new proactive defense, dubbed *Coward*, inspired by our discovery of *multi-backdoor collision effects*, in which consecutively planted, distinct backdoors significantly suppress earlier ones. In general, we detect attackers by evaluating whether the server-injected, conflicting global watermark is erased during local training rather than retained. Our method preserves the advantages of proactive defenses in handling data heterogeneity (*i.e.*, non-i.i.d. data) while mitigating the adverse impact of OOD bias through a revised detection mechanism. Extensive experiments on benchmark datasets confirm the effectiveness of *Coward* and its resilience to potential adaptive attacks. The code for our method would be available at <https://github.com/still2009/cowardFL>.

**Index Terms**—Backdoor Defense, Backdoor Attack, Federated Learning, AI Security, Trustworthy ML.

## I. INTRODUCTION

FEDERATED Learning (FL) [1], [2], [3] has emerged as a powerful paradigm for privacy-preserving distributed machine learning, enabling collaborative model training among many client devices without sharing raw data. Such character enables secure utilization of massive valuable data from sensitive entities and individuals, unleashing the potential of private data towards personalized and foundational intelligence [4], [5], [6]. It has been widely studied across diverse fields such as healthcare, finance, and edge computing [7], [8], [9].

However, the privacy-preserving nature of federated learning is a double-edged sword. While it safeguards sensitive client data by keeping it local, it also restricts the server’s visibility into client-side behavior. This blind spot opens the door to backdoor attacks [10], [11], [12], which plant malicious

Wenjie Li and Shu-Tao Xia are affiliated with Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China 518055 (e-mail: liwj20@mails.tsinghua.edu.cn, xiast@sz.tsinghua.edu.cn).

Siying Gu and Zhili Chen are affiliated with Software Engineering Institute, East China Normal University, Shanghai, China 200062 (e-mail: Sy.Gu@stu.ecnu.edu.cn, zhlichen@sei.ecnu.edu.cn).

Yiming Li, Kangjie Chen, Tianwei Zhang, and Dacheng Tao are with College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: liyiming.tech@gmail.com, {kangjie.chen, tianwei.zhang, dacheng.tao}@ntu.edu.sg).

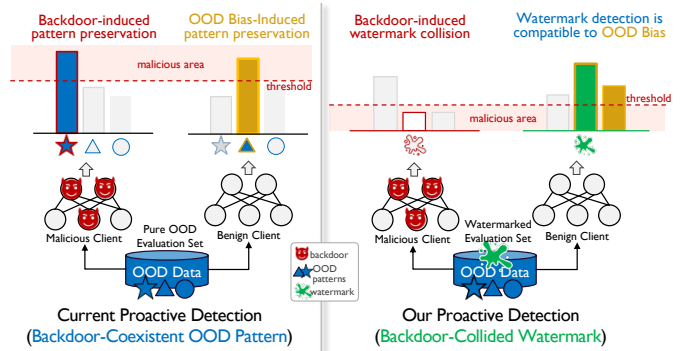


Fig. 1: Comparison between our collision-based proactive defense and the existing method via co-existent mechanism. By developing a backdoor-collided watermark, our method adopts an inverted detection paradigm with respect to malicious clients that is naturally compatible with OOD prediction bias, thereby enabling more reliable and accurate detection.

pattern into a model such that it behaves normally on clean inputs but produces attacker-desired outcomes when exposed to inputs containing predefined trigger patterns. Exploiting the server’s inability to observe client behavior, malicious participants can covertly implant backdoors into their local models. These compromised updates are then aggregated into the global model, seamlessly propagating the backdoor behavior and undermining the reliability of FL [13], [14], [15].

To tackle this issue, backdoor detection methods has been proposed to identify and exclude malicious updates during model aggregation [16], [17], [18]. Based on the assumption that backdoor training introduces distinct patterns in model parameter space, they achieve detection by identifying such anomalies. Specifically, current techniques mainly use distance-based metrics to assess similarities among client updates and may incorporate clustering algorithms to further reveal structural patterns. In either case, updates identified as outliers are flagged as malicious. However, we reveal that these defense methods are vulnerable to clients’ non-i.i.d. data distribution and participation randomness, which are common settings in practical FL scenarios [3]. These factors lead to highly dynamic and divergent benign updates that may appear as outliers, significantly interfering with detection and leading to a high false positive rate. We refer to this limitation as the **non-i.i.d. blurring challenge**, as analyzed in Section III-A.

Recently, a new type of defense termed *proactive detection*, was introduced by the pioneering work, *i.e.*, BackdoorIndi-

cator [18]. Unlike prior works passively rely on client model response, proactive defense actively modifies the global model to provoke divergent behaviors from malicious and benign clients. For instance, BackdoorIndicator injects specific OOD mappings into the global model, which tend to be preserved by malicious clients while being forget by benign clients. As such, the attacker can be identified by inspecting the presence of the injected pattern in local models, where higher inspection accuracy indicates malicious client while lower accuracy indicates benign. By *shifting the detection basis from model discrepancies to differences in client behavioral responses*, proactive methods naturally avoid the sensitivity to non-i.i.d. data that often undermines traditional defenses.

However, since FL prohibits server access to client training data, current proactive detection methods including Indicator inevitably rely on OOD data in practice, either for injecting patterns during training or evaluating pattern reactions during inference. Yet, deep neural networks (DNNs) are known to produce overconfident and biased predictions on OOD inputs [19], [20], [21], which may undermine the reliability of such detection strategies. Specifically, we reveal that such learning-based proactive methods are fundamentally vulnerable to OOD prediction bias, which causes benign local models to exhibit high inspection accuracy and be misidentified as malicious clients, leading to a high false positive rate. We provide a detailed analysis in Section III-B and refer to this limitation as the **OOD bias challenge**. Naturally, this raises an important question: *can we design a practical proactive detection method that alleviate this limitation?*

Fortunately, the answer is yes, although the solution is non-trivial. In this paper, we propose a simple yet effective Collision-based OOD Watermark for Robust federated backdoor Defense, termed Coward. Our method is inspired by the finding of *multi-backdoor collision effect* where a consecutively planted different backdoor would significantly suppress the one previously planted. In general, we detect attackers by evaluating whether the server-injected conflicting global watermark has been erased during local training. Clients whose watermark accuracy falls below a specified threshold are then marked as malicious. Specifically, we implant a backdoor-based OOD watermark as the conflicting global watermark, where OOD samples embedded with a trigger are redirected from their original class to a designated watermark target class. We refer to the original class mapping as the base OOD mapping, and the triggered mapping as the watermark mapping. Concretely, the OOD watermark is injected by jointly learning: (1) a regulated base OOD mapping that specifies a fixed class-wise correspondence from OOD samples to the task label space; and (2) a targeted watermark mapping that enforces an association between all triggered OOD samples and the watermark target class. In general, our method alleviates the OOD bias challenge through three aspects: (1) Unlike prior approaches, our collision-based detection treats high inspection accuracy as the signal of benign behavior, making it naturally compatible with the OOD bias-induced high confidence predictions; (2) Our watermark-based detection inspects the presence of the trigger-to-target mapping rather than the OOD mapping, thereby mitigating

the direct impact of OOD prediction bias. Together, these designs lead to more reliable backdoor identification and lower false positive rates. and (3) our base OOD mapping can mitigates OOD prediction bias by enforcing explicit class-wise assignments to OOD samples; We demonstrate (1) and (2) in Figure1, and verify (3) in Figure7.

In summary, our contributions are three-fold: (1) We revisit the limitations of existing methods by identifying and analyzing two key challenges unique to federated learning: non-i.i.d. blurring challenge and OOD prediction bias. (2) We find an intriguing phenomenon, dubbed the multi-backdoor collision effect, where a consecutively planted different backdoor would significantly suppress the previous implanted one. Based on this finding, we design a simple yet effective proactive federated backdoor defense method, *i.e.*, Coward, that effectively mitigates OOD bias and achieves more accurate detection. (3) Extensive experiments on benchmark datasets verify and explain the effectiveness of our Coward method and its resistance to potential adaptive attacks.

## II. BACKGROUND AND RELATED WORKS

### A. Federated Learning

Federated learning (FL) enables multiple decentralized clients to collaboratively train a shared model without exchanging their private data [2]. Based on how data is partitioned, FL is typically categorized into horizontal FL (HFL), where clients share the same feature space but different samples, and vertical FL (VFL), where clients share samples but differ in input feature fields [22].

In this paper, we focus on the HFL setting, where a central server coordinates model training and aggregation in iterative rounds. In each round, a *subset* of clients is selected to receive the current global model, train it on their local data, and send back the updated parameters. The server then aggregates these updates to form a new global model. This process repeats, gradually improving model performance while keeping data decentralized. Specifically, given the selected client set  $\mathcal{S}^{t-1}$  in round  $t - 1$ , the global model  $\mathbf{w}^t$  in round  $t$  is:

$$\mathbf{w}^t = \mathbf{w}^{t-1} + \sum_{k \in \mathcal{S}^{t-1}} \frac{N_k}{N} \Delta \mathbf{w}_k^{t-1}, \quad (1)$$

where  $N_k$  and  $N$  denote the data volume of client  $k$  and the total data volume across participating clients, respectively.  $\Delta$  denotes the model residual with respect to the previously broadcasted global model. We highlight three key characteristics that distinguish FL from traditional centralized settings:

- **Multi-party Participation:** The global model is collaboratively trained by many clients, who privately contributes to the final model but can't control it individually.
- **Non-i.i.d. Client Data** [1]: Data distributions vary significantly across clients, resulting in inconsistent local updates and increased optimization difficulty.
- **Partial Client Participation** [1]: In each training round, only a subset of clients is selected due to limited communication resources and dynamic client availability.

These characteristics introduce unique challenges for both attacks and defenses in FL, as described in subsequent parts.

### B. Backdoor Attacks in Federated Learning

Backdoor attacks aim to implant hidden behaviors into neural networks, causing them to function normally on clean data but misclassify inputs containing attacker-specified triggers [10], [23]. Data poisoning is the most common backdoor injection method, where a subset of training samples is embedded with triggers and mislabeled to a target class [23]. Stealth and persistence are key objectives in backdoor attacks, motivating the design of invisible [24], [25], semantic [26], [27], optimized [28], [29] and sample-specific [30], [31], [32] triggers. However, these approaches are limited when directly applied to FL, due to the unique challenge that adversaries can only manipulate a limited *subset of clients* and must ensure their backdoored updates *survive the aggregation process* to impact the global model. Thus, FL-specific backdoor attacks have emerged, broadly classified into following two categories.

**Single-client Attacks.** The first backdoor attack for FL introduced the model replacement attack [13], where a malicious client uploaded a scaled local update to dominate the aggregation. However, the abnormally large update made it easily detectable under defense [33]. To improve *stealth* and *persistence*, follow-up works introduced more subtle attack strategies. For example, PGD [33] constrained malicious updates within an  $\ell_2$ -ball to evade norm-based defenses, while Neurotoxin [34] injected backdoors into parameters associated with inactive neurons to minimize noticeable changes. Other works aimed to further blend backdoor patterns into normal behavior. For example, Edge-case attacks [35] utilized naturally occurring features as triggers, reducing the likelihood of detection, while Chameleon [11] aligned backdoor and benign features in latent space using contrastive learning.

**Multi-client Attacks.** The DBA framework [36] leveraged the distributed nature of FL to spread the backdoor across multiple clients, with each client using a distinct local trigger. These independently subtle triggers collectively formed an effective global backdoor, reducing the detectability of any individual client. Expanding on this idea, Gong et al. [37] proposed a coordinated multi-client attack, in which each local trigger was optimized in a model-aware fashion. This coordination allowed the attackers to craft triggers that were both less suspicious and more effective, further amplifying the threat posed by distributed poisoning. In contrast, Non-Cooperative Backdoor Attacks (NBA) [38] involved independent attackers using distinct triggers for different targets, yet still exhibited enhanced stealth comparable to coordinated methods.

### C. Backdoor Defenses in Federated Learning

In centralized settings, backdoor defenses typically focus on either removing the backdoor from the model [39], [40] or filtering trigger samples during training or inference [41], [42]. However, these approaches often *become impractical in FL due to the limited capabilities of the defender (i.e., the central server)*. Specifically, (1) the server lacks access to clean and in-distribution data, making data-dependent removal methods like pruning, retraining, or fine-tuning infeasible. (2) the server cannot control the local training process, rendering sample filtering and robust training strategies infeasible. To address

these limitations, FL-specific defense mechanisms have been developed, generally falling into below two categories.

**Backdoor Effect Mitigation** methods aim to reduce the impact of backdoors in the global model by suppressing anomalous neurons across local updates. Early works such as Weak-DP [33] applied norm clipping and noise injection to weaken abnormal neuron effects. Statistical approaches like geometric median [43] and Trimmed-Mean [44] enforced robustness by aggregating updates in a statistically neutral manner. RLR [45] further enhanced mitigation granularity by assigning parameter-wise adaptive learning rates. FLPurifier [46] achieved stronger backdoor suppression by performing adaptive aggregation at the classifier level under a modified FL protocol that decoupled encoder and classifier training. However, these methods *operate all client updates equally, often diminishing benign contributions and harming overall model performance* under the standard FL protocol. This limitation has led to the rise of the following detection-based approaches as a mainstream alternative.

**Malicious Client Detection** methods aim to identify and exclude poisoned updates during aggregation by quantifying the differences between benign and malicious clients, typically through distance-based metrics on model parameters. Clients whose update patterns deviate significantly from the majority are flagged as malicious. For instance, Multi-Krum [47] leveraged Euclidean distance, while Foolsgold [48], FLTrust [49], and ShieldFL [50] adopted cosine similarity, all applied directly on full model weights. Rflbat [51] extended this by computing distances in a top- $k$  PCA-projected space, aiming to reduce noise and improve efficiency, though at the cost of some information loss. Beyond model parameters, DeepSight [16] incorporated prediction-level discrepancies of local models, enabling a more comprehensive analysis. Extending beyond pure detection, Flame [17] combined clustering-based client identification with gradient norm-based mitigation, positioning itself at the intersection of detection and mitigation strategies and demonstrating strong empirical performance. FLGuardian [52], a recent study addressing untaged backdoor attacks, performed layer-wise model analysis using similarity metrics and clustering algorithms, reflecting a growing shift toward more fine-grained detection in the parameter space. However, these methods *assume malicious models are naturally distinguishable, making them sensitive to data heterogeneity and ineffective against stealthy attacks*.

Building on the shared characteristic that prior methods rely on passive server-side detection without any preparatory intervention, we categorize them as *passive detection*. In contrast, *proactive detection*, such as Indicator [18] and our proposed approach, take an active role by modifying the global model to provoke distinguishable behaviors from clients. By leveraging multi-backdoor interaction mechanisms that is less sensitive to data heterogeneity, these methods improve detection robustness. While Indicator marks a key advance and outperforms passive baselines, as we will show in the next section, it still suffers from *OOD bias and high false positives*. How to design an effective proactive federated backdoor defense in practice is still an important open question.

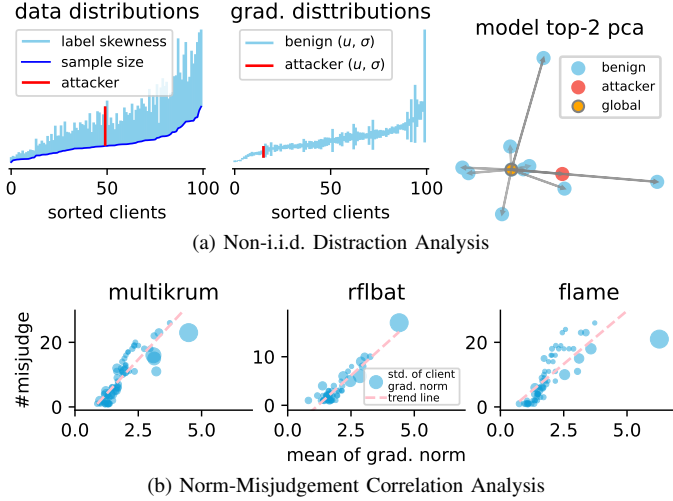


Fig. 2: **The distraction effect of non-i.i.d. data on passive backdoor detection.** Divergent client data distributions (left of Figure (a)) substantially reduce the suspiciousness of malicious clients, as reflected in both gradient norms (middle) and the update directions of benign models (right). In contrast, it increases the perceived suspiciousness of benign clients, particularly those with larger gradient norms, as evidenced by the positive correlation observed in Figure (b).

### III. REVISITING EXISTING DEFENSES AND MULTI-BACKDOOR INTERACTIONS

In this section, we first revisit the limitations of existing passive and proactive defenses, respectively. We then delve into the multi-backdoor interaction effects, a key underlying mechanism that drives the design of proactive defenses.

#### A. Revisiting Passive Defenses

Many existing passive defenses assumed that *malicious updates deviated from benign updates in the parameter space* due to the additional optimization of the backdoor objective during local training [44]. Based on this assumption, many methods leveraging clustering and outlier filtering were proposed to detect malicious clients [43], [17]. To evaluate the validity of this assumption under the practical non-i.i.d. settings, we analyzed the statistical characteristics of model updates, and show its correlation to misjudgements of benign clients.

**Settings.** We revisit gradient norm distribution for each client, under a highly non-i.i.d. partitioned CIFAR-10 dataset with label distribution generated by Dirichlet prior with  $\alpha = 0.3$ , and 10 out of 100 random clients participation in each round. There is one malicious client performing a backdoor attack using a pixel-style trigger targeting class 0.

**Results.** As shown in Figure 2(a), *the suspicion toward attackers is significantly reduced due to the severity of data heterogeneity*: (1) in the first column, we present the size and label skewness of local datasets. The blue line indicates the dataset size, while the bar height represents the skewness, measured by the standard deviation of class-wise sample counts. We can observe that client data distributions are highly diverse. Notably, malicious clients do not exhibit abnormal

characteristics in either size or skewness; their distributions fall within a typical range. In contrast, some benign clients show extreme distributions with both large data volumes and high skewness. (2) In the second column, we plot the mean and standard deviation of gradient norms uploaded by all 100 clients across training rounds, sorted by their mean values. The results reveal that the malicious client’s updates have slightly smaller gradient norms compared to most benign clients and appear inconspicuous amid the naturally diverse benign client updates. (3) In the third column, we visualize the top-2 PCA component of local and global models (based on the last-layer weights) during training. The results show that the attacker’s update direction and distance lie in a moderate region, appearing unharmed due to the diverse update directions introduced by the non-i.i.d. distribution. Figure 2(b) further presents a correlation analysis, *revealing a clear association between model update norms and misjudgement*. Specifically, the x-axis shows the mean gradient norm, the y-axis indicates the number of misjudgments (*i.e.*, times a benign client was incorrectly flagged), and the bubble size reflects the standard deviation of the gradient norm. The evident upward trend suggests that clients with larger and more variable gradient norms are more susceptible to be misjudged. In conclusion, we have:

**Takeaway 1:** Data heterogeneity (*i.e.*, non-i.i.d.) amplify client model variance, blurring distinctions between malicious and benign clients, diminishing the effectiveness of passive backdoor detection.

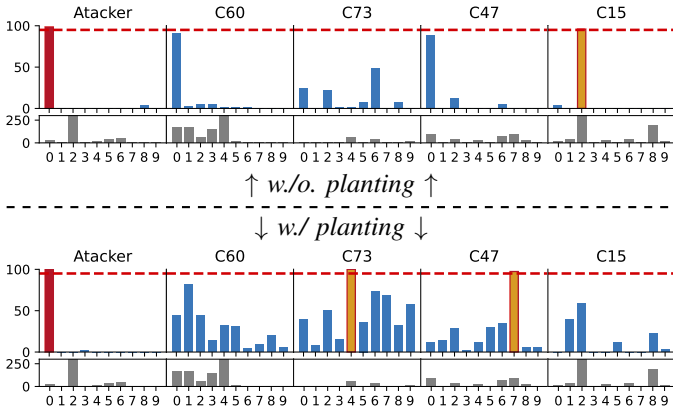
To mitigate this limitation, proactive backdoor defenses shift the detection focus from parameter differences to client-side reactions against defender-planted patterns, which effectively alleviate sensitivity to data heterogeneity.

#### B. Revisiting Proactive Defenses

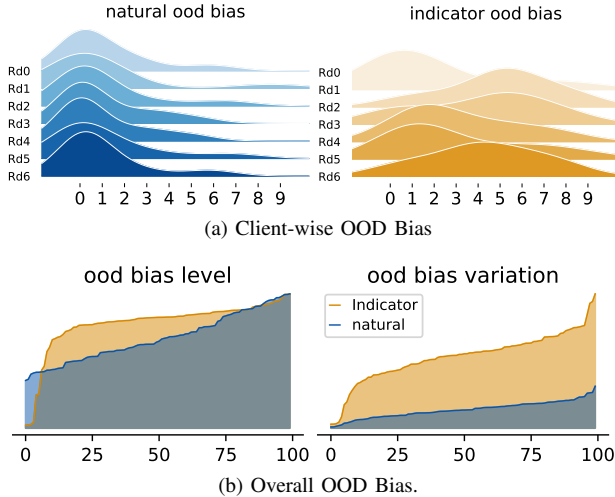
As a pioneering work in proactive defense, Indicator identified a *co-existing* (*i.e.*, maintaining) *effect*, where a subsequently planted backdoor with the same target label could preserve the effect of a previously injected one. However, due to the privacy-preserving constraint of federated learning, it could not access client data and had to rely solely on OOD data to modify the global model. To address this constraint, Indicator implanted *random OOD mappings* (*i.e.*, assigning labels to OOD samples via uniform random sampling) into the global model to activate the co-existing effect. During detection, it evaluated each client’s class-wise prediction accuracy on these OOD data: any class exceeding a predefined threshold flagged the client as malicious, and the corresponding class was assumed to be the attacker’s backdoor target. Such a detection mechanism no longer relies on parameter space, making it less susceptible to the distraction effect of non-i.i.d. data and thus more effective than passive detection methods.

However, leveraging OOD samples to activate the co-existing effect introduces a vulnerability to *OOD prediction bias*, where deep neural networks inevitably tend to classify OOD samples into arbitrary classes with high confidence [19], [21]. Specifically, it leads to a high false positive rate of this method, as analyzed in the following parts.





**Fig. 3: The misdirection effect of OOD bias against the existing proactive backdoor detection.** With the attacker-specified target class set to ‘0’ (red), we show five clients from the same round, presenting their OOD prediction distribution. The gray subfigure shows the local data distribution. The red dashed line marks the detection threshold; classes with inspection accuracy above this line are flagged as malicious and highlighted in red border. The results indicate that: (1) Even without server-side planting, non-target classes may exhibit OOD-induced high inspection accuracy (orange) on benign clients; (2) With planting, misjudgment increases, and more classes tend to exhibit higher inspection accuracy.



**Fig. 4: Exacerbation of OOD bias induced by existing proactive methods.** The Indicator method intensifies both the magnitude and variation of OOD bias across training rounds.

**Settings.** We analyze the prediction distribution of OOD samples under Indicator [18], both with and without the watermark planting process. The main FL task is CIFAR-10 classification, with EMNIST used for proactive pattern planting. During planting, Indicator randomly partitioned OOD samples in the label space with same 10% ratio. In each round, 10 clients are sampled from a total of 100, with one attacker targeting class 0. The data heterogeneity is controlled by a Dirichlet distribution with  $\alpha = 0.3$ .

**Results.** As shown in Figure 3, we find that: (1) under the no-planting setting, for the malicious client, OOD sam-

ples successfully activate the learned backdoor pattern with high accuracy, revealing an empirical correlation between the backdoor trigger and the OOD pattern. Meanwhile, for benign clients, certain classes exhibit inspection accuracy that approaches or even exceeds the predefined detection threshold of 95%, resulting in significant misjudgment; (2) Under the with-planting setting, the pre-planted OOD pattern leads to similar effects and, in some cases, causes even more classes keep higher accuracy among benign clients (*i.e.*, tending to be treated as malicious clients). These phenomena indicate that:

**Takeaway 2:** OOD prediction bias is an inherent vulnerability of deep neural networks [19], [20], [21], where OOD samples tend to be classified with high confidence into arbitrary classes. This causes benign clients to exhibit unexpectedly high inspection accuracy during Indicator’s detection phase, leading to frequent yet unpredictable false positives.

Going further, we hereby demonstrate how the planting action intensifies OOD prediction bias in Figure 4, from which we draw the following key observations. (1) *Disruption of temporal consistency:* In subfigure (a), we present the prediction distributions of a specific client across multiple global rounds. The no-planting setting maintains a stable bias toward a particular class over time, whereas the planting setting exhibits a highly variable trend, *i.e.*, the class with the highest prediction accuracy fluctuates significantly across rounds. This instability is likely caused by the random label assignment used in the Indicator method, where OOD samples with similar semantic content may be mapped to different labels during the planting process, creating an disordered association between the OOD space and the task label space; (2) *Amplified bias level and variation:* As shown in subfigure (b), we quantify the overall bias level and its temporal variation across all 100 clients, sorted by bias magnitude. The results show that Indicator’s planting introduces significantly higher bias intensity and instability compared to the no-planting case. We further analyze the correlation between OOD bias levels and false positive rates in Section V (Figure 7), showing a strong positive relationship. Building on above evidence, we arrive at the conclusion that:

**Takeaway 3:** BackdoorIndicator exacerbates OOD prediction bias through random mapping planting, which not only disrupts bias patterns but also introduces temporal inconsistency across rounds, ultimately undermining the reliability of its detection.

Given the limitations of OOD-based planting, the effectiveness of leveraging the behind multi-backdoor coexistence effect introduced by Indicator becomes questionable. This naturally leads to a deeper question: *Does an alternative multi-backdoor interaction mechanism exist, and if so, can it offer a more robust foundation for utilizing OOD data?* Accordingly, we shift to revisit multi-backdoor interactions.

TABLE I: Types of multi-backdoor interaction.

	Different Target	Same Target
Different Trigger	collision (w./ BN)	coexistence (w./ BN)
Same Trigger	collision	coexistence

### C. Revisiting Multi-Backdoor Interactions

Unlike Indicator’s observation of a co-existence effect between backdoors targeting the same label, we identify a distinct *collision effect*, where sequentially implanted backdoors targeting different labels tend to interfere with one another. This phenomenon offers new insights and serves as the starting point for our subsequent defense design.

**The Demonstration of Collision Effect.** We conduct the experiment under a centralized setting simulating two entity sequentially plant backdoors into a same ResNet model on CIFAR-10 dataset but with different target label. Specifically, the first backdoor uses a WaNet trigger targeting class 1 while the second backdoor uses a pixel-pattern trigger targeting class 0. Following [18], the batch normalization statistics are also switched accordingly. As shown in Figure 5, the left subfigure demonstrates that the ASR of the first backdoor steadily decreases as the second backdoor is progressively injected. In contrast, the right subfigure shows that the ASR decline from normal fine-tuning (due to knowledge forgetting) remains clearly less severe, highlighting the distinct collision effect between the two backdoors. Similar observations across different backdoor settings are also discussed in [53].

**Collision Effect v.s. Coexistence Effect.** To this end, we can get a complete understanding of multi-backdoor interaction effects under sequential implantation: Backdoors targeting the same class tend to coexist, while those with different targets often interfere with each other. To be comprehensive, we summarize all interaction cases in Table I, where “w./ BN” denotes for “with BN switch”. Though seemingly opposite, the two mechanisms can be understood through a unified and intuitive perspective. As noted by Indicator [18], backdoor triggers can be interpreted as out-of-distribution features relative to the benign data distribution, typically residing in a shared OOD region within the latent space. When multiple triggers share the same target label, they construct consistent mappings from the latent OOD region to that label. This alignment allows backdoors to reinforce each other, resulting in their coexistence. In contrast, when triggers are associated with different target labels, the newly implanted trigger disrupts the existing mapping from the latent OOD region to the prior label, leading to collision or substitution.

**Insights from Collision Effect.** Based on the collision effect, it becomes feasible to design a backdoor-based watermark that intentionally interferes with the attacker’s trigger while remaining intact in benign clients. This leads to a new proactive detection paradigm distinct from the existing coexistence-based methods, where client models that fail to retain the watermark (*i.e.*, exhibit low inspection accuracy) are identified as malicious, while those that preserve it (*i.e.*, show high inspection accuracy) are considered benign. Under this inverted paradigm, the high inspection accuracy induced by OOD

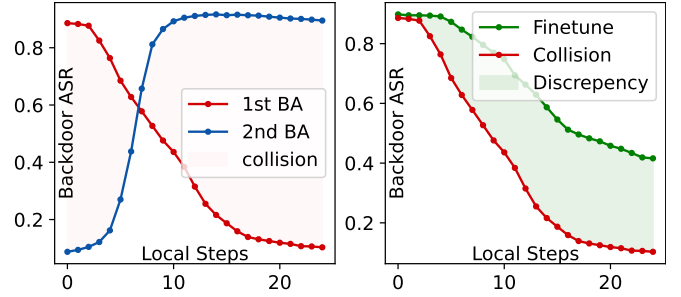


Fig. 5: **The collision effect between target-different backdoors.** (Left): A subsequently injected backdoor significantly degrades the ASR of the first backdoor. (Right): The ASR degradation of the first backdoor caused by standard fine-tuning is notably smaller than that caused by backdoor.

prediction bias in benign clients no longer misleads detection, instead, it becomes beneficial in some cases. Building on these insights, we propose our collision-based watermarking method, as detailed in the next section.

## IV. METHODOLOGY

Figure 6 presents an overview of our approach, including three main stages: watermark injection, watermark interaction, and watermark detection. We begin by clarifying the threat model, then the detail of three stages.

### A. Threat Model

We define the threat model by specifying the goals and capabilities of both the attacker and the defender.

**Attacker’s Goal and Capability.** The attacker resides on the client side and aims to inject a targeted backdoor into the global model, causing it to misclassify all trigger-embedded inputs into a designated target class while maintaining normal performance on benign inputs. Although the attacker cannot directly access the global model, they may control a subset of clients and submit arbitrary malicious models or gradients to the server, thereby influencing the global model updates.

**Defender’s Goal and Capability.** The defender is the central server, whose goal is to identify and exclude backdoored local models from the aggregation process. The server cannot access any data from the clients’ local distributions but has full visibility into all uploaded local models. It also controls the update and broadcast of the global model. Additionally, the server is allowed to make arbitrary modifications to the global model and has access to any out-of-distribution data.

### B. Watermark Injection

Watermarking is commonly used for tracing malicious or valuable sources by embedding identifiable patterns into models or data [54], [55], [56]. Inspired by this idea and our observation of the backdoor collision effect, we leverage a backdoor-based watermarking strategy to identify malicious clients in federated learning. To adapt the idea of backdoor watermarking to the out-of-distribution setting, we first revisit

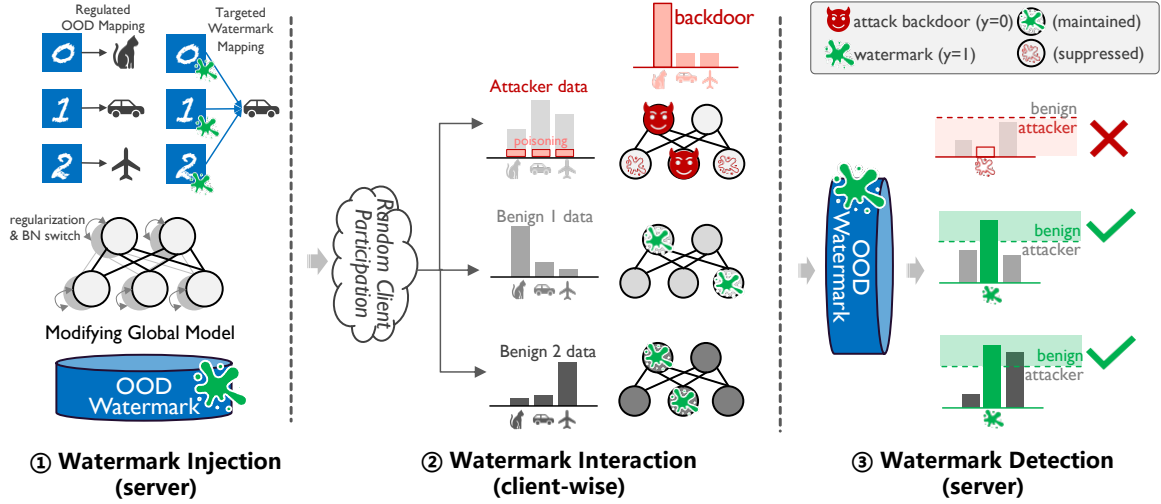


Fig. 6: **The overall pipeline of our Coward method.** (left) A defensive watermark is carefully embedded into the global model via low-cost OOD data training. (Middle) Random participants conduct local training based on the watermarked global model, where attackers tend to remove the watermark while benign clients preserve it. (Right) After local training, the server inspects the strength of the watermark; models with diminished watermark signals are flagged as malicious.

the conceptual structure of traditional backdoor mechanisms. In conventional in-distribution scenarios, a backdoor can be viewed as composed of two parts: (1) a standard task mapping for clean inputs, and (2) a tampered mapping that redirects triggered inputs to attacker-specified labels. Following this intuition, we borrow a similar two-part structure in the OOD context: an *OOD base mapping* for clean OOD samples and an *OOD watermark mapping* for triggered ones.

**Planting a Regulated Base Mapping.** In the federated setting, the OOD planting set and client training data share a global model, thus operating within the same prediction space. However, their semantic spaces are inherently different. This misalignment introduces *prediction ambiguity*, where the same output index may imply conflicting meanings. For instance, label “0” might refer to the digit “0” in an EMNIST-based OOD planting set, but to “airplane” in a CIFAR-10 client task. Such ambiguity can lead to two negative effects: (1) the OOD mapping may distort client-learned semantics, reducing benign accuracy; (2) the non-i.i.d. nature of client training may corrupt the OOD mapping in different ways.

To address these problems, we manually assign a fix one-to-one label mapping  $\pi_1$  from the OOD label space  $\mathcal{Y}_o$  to the task label space  $\mathcal{Y}_t$ . This defines the base mapping planting:

$$\pi_1 : \mathcal{Y}_o \rightarrow \mathcal{Y}_t, \quad (2)$$

$$\mathcal{D}_p = \{(\mathbf{x}_i, \pi_1(y_i)) \mid y_i \sim \mathcal{Y}_o\}, \quad (3)$$

$$\mathcal{L}_{base} = \sum_{i \in \mathcal{D}_p} \text{CE}(f(\mathbf{x}_i; \mathbf{w}), \pi_1(y_i)), \quad (4)$$

Here,  $\mathcal{D}_p$  denotes the planting set,  $\mathbf{w}$  represents the model parameters, and CE is the cross-entropy loss. Since OOD samples lack semantic grounding in the task label space, the mapping’s meaning is not essential, what matters is that it remains fixed. Our choice of consistent one-to-one assignment stabilizes the ambiguity, reducing uncertainty in both

OOD and task predictions. This contrasts with the design in BackdoorIndicator, which adopts random and dynamic base mappings, leading to greater instability. Instead, our method emphasizes a controlled and interpretable structure, as supported by the empirical results in Figure 7.

**Planting a Targeted Watermark Mapping.** Building upon the established base OOD mapping, we further implant a targeted watermark mapping, analogous to the all-to-one backdoor setting [10], [11], [12] in centralized learning:

$$\pi_2 : \mathcal{Y}_o \rightarrow y_m, y_m \in \mathcal{Y}_t, \quad (5)$$

$$\mathcal{D}_w = \{(\text{tri}(\mathbf{x}_i, \mathbf{t}), \pi_2(y_i)) \mid (\mathbf{x}_i, y_i) \sim \mathcal{D}_p\}, \quad (6)$$

$$\mathcal{L}_{wm} = \sum_{i \in \mathcal{D}_w} \text{CE}(f(\text{tri}(\mathbf{x}_i, \mathbf{t}); \mathbf{w}), \pi_2(y_i)), \quad (7)$$

Here,  $\mathbf{t}$  denotes the trigger pattern, and  $\text{tri}(\cdot, \mathbf{t})$  represents the trigger implanting function. The watermark mapping function  $\pi_2$  assigns all watermark samples to a fixed target label  $y_m$ . The watermark set  $\mathcal{D}_w$  is generated by sampling a fraction  $\rho_w$  of the planting set, i.e.,  $|\mathcal{D}_w| = \rho_w \cdot |\mathcal{D}_p|$ . Since the base mapping  $\pi_1$  aligns OOD samples with the task feature space, the planted watermark can partially generalize to the task feature distribution. This makes the trigger-label association less tied to specific OOD semantics, thereby enhancing its robustness to OOD prediction bias and reducing the likelihood of being forgotten during training of benign clients.

Following prior works [57], [18], we save and restore the running mean and variance of BatchNorm layers before and after watermark planting. This preserves the distinct feature distributions of the task and OOD spaces during training. In addition, to prevent the planted watermark from distorting the task-space sample-label mapping, we also apply a global model regularization term. Together, we get the overall loss:

$$\mathcal{L}_{server} = \mathcal{L}_{base} + \mathcal{L}_{wm} + \lambda \cdot \|\mathbf{w} - \mathbf{w}^t\|_2. \quad (8)$$

### C. Watermark Interaction and Detection

**Watermark Interaction.** After the watermark is injected into the global model  $w^t$ , it is delivered to all participated clients for local training. Benign clients perform standard training on their local datasets, optimizing only for task accuracy. In contrast, malicious clients additionally inject backdoor triggers into part of their local data to achieve the attack. Their training objectives are formulated as:

$$\mathcal{L}_{benign} = \sum_{l \in \mathcal{D}_k} \text{CE}(f(x_l; w_k^{t+1}), y_l), \quad (9)$$

$$\mathcal{L}_{attack} = \mathcal{L}_{benign} + \sum_{l \in \mathcal{D}_k^{tri}} \text{CE}(f(\text{tri}(x_l, t); w_k^{t+1}), y_a), \quad (10)$$

where  $\mathcal{D}_k$  denotes the  $k$ -th client's clean data and  $\mathcal{D}_k^{tri}$  is the triggered subset with label  $y_a$  as the attacker-specified target label. During training, benign clients would only forget the OOD watermark slightly while malicious clients tend to suppress the watermark more aggressively, as their injected backdoor interfere with the watermark. This divergence in watermark retention behavior serves as the basis for our subsequent malicious client detection. We provide a detailed analysis of these effects in Appendix A.

**Watermark Detection.** After receiving trained local models from clients, the server evaluates each model's response to the watermark task by measuring the proportion of watermarked samples predicted as the designated label  $y_m$ . The BatchNorm statistics are also switched from those used for the FL main task to those corresponding to the watermark planting phase during inspection. After collecting the watermark accuracy of all clients, a threshold  $\beta$  is applied to distinguish between benign and malicious clients. Thus, clients whose watermark accuracy falls below  $\beta$  are excluded from the aggregation process. Thus we get the model aggregation operation as:

$$\sum_{k \in \mathcal{S}^{t+1}} \frac{N_k}{N} \cdot \Delta w_k^{t-1} \cdot \mathbb{I}[\text{ACC}(f(\mathcal{D}_w; w_k^{t+1}), y_m) > \beta], \quad (11)$$

where  $\mathbb{I}$  denotes the indicator function, and ACC represents the corresponding watermark accuracy.

## V. EXPERIMENTS

### A. Main Settings

**Datasets and Models.** We conduct experiments on 3 classical image classification benchmark datasets, including EMNIST [58], CIFAR-10 and CIFAR-100 [59]. We employ ResNet-18 [60] as the default architecture.

**Federated Training Configurations.** All experiments adopt the standard FedAvg setting [1], where we simulate 100 clients in total, with 10 randomly selected to participate in each training round. To simulate the inherent data heterogeneity across clients, we follow [61] to use Dirichlet sampling to partition the dataset. The Dirichlet concentration parameter  $\alpha$  controls the label distribution skewness, where smaller values indicate more unbalanced distributions. Specifically, we use  $\alpha = 0.9$  by default to simulate mild skewness, and decrease it

to 0.3 to model more severe distribution shifts. During local training, each benign client performs 2 epochs with a learning rate of 0.03, while the global training proceeds for 1200 rounds. For conciseness, we omit the less critical parameter details and follow [18] for all settings not explicitly specified.

**Attack Configurations.** To ensure a comprehensive evaluation, we consider both single-attacker and multi-attacker scenarios, each covering various attack strategies and backdoor types. (1) In the single-attacker scenario, we adopt a range of attack baselines, including the naive *Vanilla* training [23], the stealthy *PGD* attack [33], and the latest state-of-the-art methods *Neurotoxin* [34] and *Chameleon* [11]. Regarding the backdoor type, we adopt three diverse types of triggers, ranging from the traditional visible trigger (*BadNet* [23]), to the invisible trigger (*Blend* [62]), and the natural feature-based trigger (*Semantic* [63]), with increasing levels of imperceptibility. (2) In the multi-attacker scenario, we increase the ratio of attackers from 30% to 70%. To ensure comprehensive coverage, we include both aligned and divergent attack goals: Uniform [48] with homogeneous attackers, DBA [36] with a shared target but distributed triggers, and NBA [38] with non-cooperative attackers using distinct backdoors. Across all settings, poisoning uses a consistent learning rate of 0.03 by default, aligned with benign training, and reduces it to half for more challenging stealthy evaluation purposes. The backdoor target is class 0 by default. Similar to BackdoorIndicator, the malicious client launches the attack at the 1000th global round, with poisoning action lasting for 200 global rounds.

**Defense Baselines.** We evaluate six representative defense methods that span a range of detection strategies, from classical model-space analysis to more advanced and proactive techniques: We begin with traditional methods that rely on the statistical properties of model updates: (1) *MultiKrum* [47] detects attackers based on the Euclidean distance among client updates. (2) *Foolsgold* [48] identifies malicious clients by analyzing pairwise cosine similarity to detect overly aligned update directions. (3) *Rflbat* [51] projects model updates into a PCA space and flags outliers as potential attackers. Then we further include more advanced approaches that incorporate prediction behavior or mitigation mechanisms: (4) *Deepsight* [16] evaluates similarity in model predictions and output neuron activations to reveal abnormal client behavior. (5) *Flame* [17] combines detection based on model similarity together with mitigation techniques such as gradient clipping and noise injection. Finally, we include the pioneering proactive method that shifts the detection paradigm: (6) *BackdoorIndicator* [18] injects OOD patterns into the global model and identifies attackers based on their reactions to these patterns.

**Our Configurations.** We detail the key factors involved in our watermark injection and detection process. To construct the OOD planting set, we use EMNIST (expanded to three channels) when the main task is CIFAR-10 or CIFAR-100, and grayscale CIFAR-10 when the main task is EMNIST. The same planting set is used for BackdoorIndicator to ensure a fair comparison. By default, the planting set contains 1,000 samples. At each global round, the server injects the watermark by applying a trigger to 20% of the planting set, using a



TABLE II: Detection performance against vanilla single-client attacks under different non-i.i.d. settings. **Boldface** indicates the best result value (excluding the collapsed case with TPR/FPR=0/0), while **red** highlights metric-specific failures based on a 50% threshold (*i.e.*, values above 50% for FPR and ASR, and below 50% for TPR and BA).

Dataset	Non-i.i.d.	MultiKrum				FoolsGold				Rflbat				DeepSight				Flame				Indicator				Coward (Ours)			
		TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA
CIFAR-10	0.9	93.5	34.1	34.1	<b>92.1</b>	44.0	44.3	<b>91.3</b>	<b>92.1</b>	2.5	9.6	<b>91.3</b>	91.8	74.0	<b>60.3</b>	<b>58.2</b>	91.9	<b>99.0</b>	33.4	9.9	<b>92.1</b>	97.5	23.0	<b>9.4</b>	91.2	<b>99.0</b>	<b>3.9</b>	10.3	90.5
	0.6	62.0	37.6	<b>88.9</b>	91.9	47.5	<b>50.3</b>	<b>89.5</b>	91.4	1.5	10.4	<b>91.3</b>	<b>92.0</b>	66.0	<b>53.2</b>	<b>66.4</b>	<b>92.0</b>	69.0	36.8	<b>87.2</b>	91.8	94.0	26.1	35.3	91.5	<b>99.0</b>	<b>9.2</b>	<b>9.9</b>	90.8
	0.3	57.5	38.1	<b>90.2</b>	91.7	60.5	<b>57.2</b>	<b>84.5</b>	90.8	1.0	9.2	<b>91.2</b>	<b>91.9</b>	57.0	44.8	<b>72.7</b>	91.8	62.0	37.5	<b>88.5</b>	91.8	93.0	31.0	24.2	91.5	<b>99.0</b>	<b>9.6</b>	<b>10.0</b>	90.4
CIFAR-100	0.9	<b>98.0</b>	22.4	1.1	69.7	0.0	0.0	<b>82.9</b>	<b>69.9</b>	12.0	22.6	<b>83.2</b>	69.7	<b>39.0</b>	32.6	46.8	69.8	97.0	22.6	1.1	69.7	95.0	47.0	1.3	69.0	<b>98.0</b>	<b>2.9</b>	<b>1.0</b>	69.1
	0.6	96.0	22.7	<b>1.0</b>	<b>69.9</b>	0.0	0.0	<b>82.7</b>	69.5	11.0	22.4	<b>83.4</b>	69.6	55.0	47.8	41.5	69.5	95.0	22.8	<b>1.0</b>	69.8	81.0	47.8	<b>56.1</b>	69.0	<b>100.0</b>	<b>4.3</b>	<b>1.0</b>	69.2
	0.3	<b>2.0</b>	33.1	<b>82.0</b>	69.5	0.0	0.0	<b>81.5</b>	69.3	6.0	23.3	<b>82.4</b>	69.2	75.0	<b>65.3</b>	25.2	<b>69.7</b>	<b>5.0</b>	32.8	<b>81.6</b>	69.3	76.0	43.6	46.8	67.8	<b>98.0</b>	<b>8.6</b>	<b>1.0</b>	68.9
EMNIST	0.9	<b>100.0</b>	22.2	<b>10.0</b>	<b>99.8</b>	52.0	39.1	<b>99.9</b>	99.7	16.0	22.7	<b>100.0</b>	99.7	57.0	45.9	27.4	99.7	<b>100.0</b>	22.0	<b>10.0</b>	99.7	99.0	<b>57.8</b>	<b>10.0</b>	99.6	98.0	<b>1.6</b>	<b>10.0</b>	99.7
	0.6	98.0	22.4	<b>10.0</b>	<b>99.8</b>	35.0	41.7	<b>99.7</b>	99.7	11.0	23.1	<b>99.8</b>	99.7	56.0	49.4	47.8	99.7	97.0	22.3	<b>10.0</b>	99.7	96.0	<b>52.7</b>	10.8	99.6	<b>98.0</b>	<b>1.6</b>	<b>10.0</b>	99.7
	0.3	<b>9.0</b>	32.3	<b>100.0</b>	99.7	40.0	48.2	<b>99.5</b>	<b>99.8</b>	1.0	25.1	<b>99.9</b>	99.7	<b>49.0</b>	47.0	47.3	<b>99.8</b>	<b>9.0</b>	32.1	<b>100.0</b>	99.7	96.0	45.6	10.2	99.6	<b>98.0</b>	<b>8.3</b>	<b>10.0</b>	99.7

TABLE III: Detection performance against stealthy single-client attacks with advanced training strategies. Result highlighting (in **boldface** and in **red**) follows the same rules as those described in Table II.

Dataset	Attack	MultiKrum				FoolsGold				Rflbat				DeepSight				Flame				Indicator				Coward (Ours)			
		TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA
CIFAR-10	Vanilla	5.5	32.7	<b>97.0</b>	91.7	53.5	44.4	<b>92.8</b>	<b>91.9</b>	3.0	24.2	<b>96.6</b>	<b>91.9</b>	77.9	<b>64.0</b>	30.2	91.8	<b>30.5</b>	29.9	<b>96.3</b>	91.7	93.0	34.7	4.9	<b>91.9</b>	<b>100.0</b>	<b>3.3</b>	<b>4.8</b>	91.5
	PGD	5.5	32.7	<b>97.3</b>	91.7	53.5	44.6	<b>92.7</b>	<b>91.9</b>	2.0	24.5	<b>96.4</b>	91.8	75.5	<b>61.0</b>	32.3	<b>91.9</b>	<b>12.0</b>	31.8	<b>97.8</b>	91.7	92.0	34.1	<b>5.5</b>	91.6	<b>97.5</b>	<b>4.2</b>	5.9	91.4
	Neurotoxin	6.0	32.7	<b>96.4</b>	91.7	15.5	43.1	<b>96.0</b>	91.8	2.5	24.3	<b>95.5</b>	91.8	69.2	<b>58.5</b>	41.1	<b>91.9</b>	<b>30.0</b>	29.9	<b>95.5</b>	91.4	52.5	31.7	<b>79.1</b>	91.4	<b>99.5</b>	<b>7.8</b>	<b>4.1</b>	91.6
	Chameleon	8.0	32.4	<b>94.1</b>	<b>91.8</b>	46.5	44.1	<b>88.4</b>	91.7	3.5	23.6	<b>93.4</b>	<b>91.8</b>	<b>39.0</b>	36.5	<b>62.4</b>	91.3	<b>10.0</b>	32.2	<b>94.9</b>	91.6	87.0	33.8	<b>8.4</b>	91.3	<b>94.5</b>	<b>9.1</b>	26.1	91.4
CIFAR-100	Vanilla	6.5	32.6	<b>92.9</b>	69.5	0.0	0.0	<b>88.9</b>	<b>69.6</b>	3.5	23.7	<b>92.8</b>	<b>69.6</b>	48.7	44.4	<b>53.1</b>	69.4	7.5	32.5	<b>93.0</b>	69.4	91.0	45.3	9.3	68.9	<b>100.0</b>	<b>1.9</b>	<b>0.8</b>	68.8
	PGD	6.5	32.6	<b>92.9</b>	69.6	0.0	0.0	<b>92.0</b>	<b>69.7</b>	2.5	23.4	<b>92.7</b>	<b>69.7</b>	46.7	43.0	49.3	69.6	2.0	33.1	<b>93.4</b>	69.4	88.0	45.7	16.1	68.9	<b>100.0</b>	<b>5.8</b>	<b>0.8</b>	69.0
	Neurotoxin	7.0	32.6	<b>91.4</b>	69.4	0.0	0.0	<b>91.7</b>	69.4	4.5	23.2	<b>92.7</b>	<b>69.7</b>	51.3	42.7	47.4	69.6	2.5	33.1	<b>92.9</b>	69.3	96.0	46.9	2.3	69.2	<b>100.0</b>	<b>9.4</b>	<b>0.8</b>	69.0
	Chameleon	0.0	33.3	<b>85.6</b>	69.3	0.0	0.0	<b>90.7</b>	69.5	5.0	24.3	<b>86.3</b>	<b>69.6</b>	41.0	34.7	<b>60.7</b>	69.4	0.0	33.3	<b>87.1</b>	69.4	<b>98.5</b>	47.5	16.0	66.7	98.2	<b>7.3</b>	<b>0.9</b>	69.1
EMNIST	Vanilla	8.5	32.4	<b>100.0</b>	<b>99.7</b>	12.0	35.7	<b>100.0</b>	<b>99.7</b>	1.0	24.6	<b>100.0</b>	<b>99.7</b>	24.0	37.3	<b>100.0</b>	<b>99.7</b>	44.5	28.3	<b>100.0</b>	<b>99.7</b>	88.3	45.7	21.4	99.7	<b>100.0</b>	<b>3.6</b>	<b>0.0</b>	<b>99.7</b>
	PGD	10.0	32.2	<b>99.9</b>	<b>99.7</b>	11.0	35.2	<b>100.0</b>	<b>99.7</b>	2.0	11.7	<b>100.0</b>	<b>99.7</b>	44.0	46.9	<b>99.8</b>	<b>99.7</b>	37.0	31.4	33.0	<b>99.7</b>	88.5	46.0	25.3	99.6	<b>100.0</b>	<b>4.3</b>	<b>0.0</b>	<b>99.7</b>
	Neurotoxin	10.5	31.9	<b>99.9</b>	<b>99.7</b>	2.5	36.3	<b>100.0</b>	<b>99.7</b>	1.5	12.0	<b>100.0</b>	<b>99.7</b>	38.5	43.1	<b>100.0</b>	<b>99.7</b>	44.5	17.2	<b>96.8</b>	<b>99.7</b>	88.6	45.3	16.0	98.9	<b>99.0</b>	<b>3.6</b>	<b>0.1</b>	<b>99.7</b>
	Chameleon	12.5	31.9	<b>100.0</b>	<b>99.7</b>	21.5	37.2	<b>100.0</b>	<b>99.7</b>	3.0	24.8	<b>100.0</b>	99.6	37.5	42.7	<b>100.0</b>	<b>99.7</b>	98.0	22.3	29.3	<b>99.7</b>	91.5	<b>51.7</b>	39.2	99.6	<b>99.5</b>	<b>2.4</b>	<b>15.4</b>	<b>99.7</b>

learning rate of 0.001 for 5 iterations. In our main experiments, we adopt WaNet [24] as the default watermark trigger. For detection, a fixed 5% threshold is used across all settings: clients with watermark accuracy below this are identified as attackers. The watermark target is set to 8 by default.

**Evaluation Metrics.** We evaluate the detection performance using true positive rate (TPR) and false positive rate (FPR). TPR reflects the proportion of correctly identified malicious clients, while FPR represents the proportion of benign clients incorrectly classified as malicious. Higher TPR and lower FPR indicate more effective detection. In addition, we also assess attack success rate (ASR) and benign accuracy (BA) after detection, where lower ASR and higher BA indicate that the model’s primary functionality is preserved, and backdoor effects are effectively excluded from the model.

### B. Main Results

For a comprehensive evaluation, We evaluate all baseline defenses from following three perspectives:

**Resistance to FL Non-i.i.d.** We evaluate all baseline methods under varying degrees of non-i.i.d. data across three datasets, using a vanilla backdoor setting with a fixed spot-pattern trigger and a learning rate matching that of benign clients. As shown in Table II, Coward consistently achieves the best overall performance across all levels of heterogeneity, with TPRs above 95% and FPRs below 10.5% in all cases. Besides the overall conclusion, we draw several key observations: (1) *Coward significantly reduces FPR across all settings compared to BackdoorIndicator*, highlighting its advantage in

mitigating OOD bias. A more detailed analysis is provided in Section V-E. (2) *Proactive methods significantly outperform passive ones under severe data heterogeneity*. Taking the most challenging non-i.i.d. setting ( $\alpha = 0.3$ ) as an example, aside from our method, BackdoorIndicator also outperforms all passive baselines, achieving TPRs of 93% on CIFAR-10 and 96% on EMNIST. This result supports the robustness of proactive defenses in heterogeneous environments. In contrast, passive methods exhibit significant performance degradation. For instance, the state-of-the-art passive method Flame sees its TPR drop from above 95% to below 10% on both CIFAR-100 and EMNIST, which aligns with our earlier analysis. (3) *Some passive methods fail due to their unrealistic assumptions*. For example, Rflbat performs poorly across all heterogeneity levels, likely because it relies only on the top two principal components of model parameters, losing critical information, especially under our challenging FL setup. Similarly, Fools-Gold collapses on CIFAR-100, always judging all clients as benign (resulting in 0% TPR and 0% FPR). This failure stems from two factors: the high-dimensional 100-class prediction vector and its reliance on multi-attacker scenarios. These conditions lead to extremely low cosine similarity values, causing an all-as-benign judgment.

**Resistance to Advanced Attacks.** We evaluate all defense methods under more challenging client-side attacks. Specifically, across all attack methods, the malicious client adopts a more imperceptible noise-blended trigger and reduced local learning rate. These advanced configurations significantly increase the stealthiness of the attack. As shown in Table III, *our*

TABLE IV: Detection performance under multiple client attack on CIFAR-10 dataset. Result highlighting (in **boldface** and in **red**) follows the same rules in Table II.

Strategy	Atk_ratio	MultiKrum				FoolsGold				Rflbat				DeepSight				Flame				Indicator				Coward (Ours)			
		TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA
Uniform	30%	<b>100.0</b>	71.4	10.0	91.6	<b>100.0</b>	34.2	9.7	<b>91.8</b>	39.4	22.5	88.7	91.6	3.7	14.6	<b>97.8</b>	91.6	<b>100.0</b>	14.3	9.7	<b>91.9</b>	99.2	29.5	11.1	91.5	<b>100.0</b>	6.7	9.9	91.5
	50%	3.0	97.0	98.4	90.8	<b>100.0</b>	27.1	10.1	<b>91.8</b>	31.3	0.0	91.9	91.4	1.1	24.0	<b>98.2</b>	91.4	21.5	57.3	99.2	91.4	99.1	16.3	12.2	91.2	<b>100.0</b>	8.2	9.9	91.2
	70%	29.0	98.9	98.2	90.8	<b>100.0</b>	12.8	9.7	<b>91.8</b>	14.3	11.8	93.5	90.0	2.4	54.8	98.8	91.5	31.3	57.8	99.0	91.5	99.4	44.6	40.5	88.8	<b>100.0</b>	11.2	10.5	89.6
DBA	30%	<b>100.0</b>	57.4	9.8	91.6	<b>100.0</b>	32.9	9.5	<b>91.9</b>	28.0	15.4	81.9	91.5	5.6	12.4	<b>90.6</b>	91.5	<b>100.0</b>	14.3	9.5	<b>91.9</b>	98.2	18.4	10.1	91.8	<b>100.0</b>	9.7	9.7	91.3
	50%	22.8	77.2	96.7	91.1	<b>100.0</b>	26.7	9.9	91.9	29.2	0.0	90.7	91.3	2.9	20.1	94.7	91.6	60.5	18.9	81.9	91.7	98.6	16.0	10.6	91.5	<b>100.0</b>	9.9	10.0	91.2
	70%	31.1	94.1	97.7	90.1	<b>100.0</b>	12.8	9.6	<b>91.8</b>	9.5	2.8	95.3	88.8	3.4	44.7	96.0	90.9	42.0	34.8	96.5	91.3	98.8	15.8	11.9	89.4	<b>100.0</b>	11.2	10.3	89.6
NBA	30%	<b>100.0</b>	57.1	10.1	91.7	37.5	48.4	88.6	91.7	78.5	70.8	65.7	91.7	2.8	31.2	90.1	91.8	55.2	18.9	86.5	92.0	97.8	39.4	11.0	91.2	98.7	7.1	10.1	91.5
	50%	63.7	36.3	91.5	91.6	33.3	54.6	93.4	91.6	16.5	15.8	85.2	91.4	1.4	48.1	93.9	91.7	19.2	40.6	94.8	91.1	93.7	38.2	32.6	91.5	98.5	15.0	10.1	91.5
	70%	36.9	80.7	96.5	90.4	23.4	62.7	96.1	90.8	0.1	0.0	92.1	89.4	1.3	78.8	97.1	90.6	10.6	74.8	96.4	90.6	81.3	14.0	66.7	90.8	86.2	14.6	24.7	91.6

method achieves the best overall performance, with proactive defenses showing strong robustness under advanced threats, while passive defenses suffer substantial degradation. This highlights the advantage of the proactive paradigm in handling sophisticated attacks. (1) Among proactive methods, Coward consistently achieves the best performance, maintaining close to 100% TPR and sub-10% FPR across all advanced attack types, including state-of-the-art strategies such as Neurotoxin and Chameleon. BackdoorIndicator also performs strongly, with TPRs consistently exceeding 85% in most settings; however, it suffers from relatively high FPRs, reflecting its vulnerability to distributional bias. (2) For passive methods, performance degradation is anticipated under such stealthy attacks, since the attacker introduces only subtle changes to the model, which reduces its suspiciousness. As an instance, *MultiKrum*, which selects representative updates based on Euclidean distance, becomes highly ineffective, with TPRs falling below 12.5% across all stealthy attack settings. *Flame*, which filters out updates with large angular deviations, also suffers notable degradation, with TPRs dropping below 30.5% on CIFAR-10 dataset. In contrast, *DeepSight* shows comparatively better resilience, which perform consistently with TPRs over 40% in most advanced attack cases. This improvement most probably stems from its joint use of both parameter and prediction discrepancies during its detection process.

**Robustness to Multiple Attackers.** Table IV presents detection performance under multi-attacker scenarios. As shown in this table, proactive methods exhibit better overall performance across all three settings. This is expected as the detection mechanism of proactive approaches is independent of the number of attackers. Among them, our method achieves the best overall performance, consistently maintaining TPRs above 99% and FPRs below 15%. BackdoorIndicator also performs well but becomes less effective when the attacker proportion reaches 70%. In such cases, a temporary detection failure allows the increased number of attackers to rapidly reinforce the backdoor in the global model, resulting in a higher ASR. Moreover, the retained backdoor could be further maintained by the indicator’s own planting process due to the maintenance effect. Among passive baselines, FoolsGold performs best under Uniform and DBA attacks, achieving a TPR of 100%, benefiting from its assumption of highly similar, coordinated malicious updates. However, it fails in uncoordinated scenarios, where attackers pursue different targets. In such cases, the TPR drops below 40% as the method’s core assumption

TABLE V: Performance on different OOD planting sets. Result highlighting (in **boldface** and in **red**) follows the same rules as those described in Table II.

Main Task	Planting Set	Indicator				Coward			
		TPR	FPR	ASR	BA	TPR	FPR	ASR	BA
CIFAR-10	EMNIST	98.0	24.0	10.2	91.6	<b>99.5</b>	7.3	9.7	<b>91.8</b>
	CIFAR-100	97.0	14.2	10.0	<b>91.7</b>	<b>99.5</b>	6.7	9.9	91.2
	NOISE	<b>100.0</b>	57.4	9.8	<b>90.7</b>	<b>100.0</b>	12.7	10.1	90.6
CIFAR-100	EMNIST	98.0	34.0	1.3	68.2	<b>100.0</b>	5.3	1.0	<b>68.9</b>
	CIFAR-10	99.0	44.6	1.8	67.2	<b>100.0</b>	6.3	1.2	<b>67.5</b>
	NOISE	98.0	60.0	1.5	<b>67.9</b>	<b>100.0</b>	4.4	1.0	67.8
EMNIST	CIFAR-10	95.0	57.8	<b>10.0</b>	99.6	<b>99.5</b>	2.2	<b>10.0</b>	<b>99.8</b>
	CIFAR-100	97.0	58.7	<b>10.0</b>	99.6	<b>99.0</b>	1.5	10.1	<b>99.7</b>
	NOISE	100.0	73.5	<b>10.0</b>	<b>99.6</b>	<b>99.0</b>	4.8	10.2	<b>99.6</b>

of shared attack objectives no longer holds. Outlier-based defenses such as MultiKrum and Flame become ineffective when attackers control 70% of clients, with TPRs dropping under 42% in all cases. These methods assume malicious updates are outliers, which fails when adversaries form the majority among all clients.

### C. Ablation Study

We investigate key factors that influence watermark planting and detection: (1) the choice of OOD planting set, (2) the pairing of watermark and backdoor triggers, and (3) the selection of detection thresholds. For a more comprehensive analysis, please refer to Appendices B and C.

**Robustness to Choices of OOD Datasets.** Table V presents detection performance across diverse OOD planting sets, including task-similar data, distinctly different data, and an extreme case with synthetic Gaussian noise. We observe that Coward consistently maintains low FPR across all settings, while BackdoorIndicator’s performance degrades as the OOD dataset diverges from the main task. Using the most challenging main task on CIFAR-100 dataset as example, Coward’s FPR remains below 6.3% across all planting sets. In contrast, BackdoorIndicator’s FPR increases sharply from 44.6% on CIFAR-10 to 60% with Gaussian noise. This confirms our earlier analysis of BackdoorIndicator’s increased vulnerability to OOD bias. Overall, these results demonstrate that Coward offers improved robustness to the choice of OOD data, enabling more flexible deployment in real-world scenarios.

TABLE VI: Detection performance with different combinations of server-side watermark trigger (**S**) and attackers’ client-side malicious trigger (**A**) on CIFAR-10.

$\downarrow S / A \rightarrow$	Pixel (visible)				Blend (invisible)				Semantic (implicit)			
	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA	TPR	FPR	ASR	BA
<b>Diagonal</b>	100.0	6.1	9.8	91.6	99.5	6.6	5.2	91.7	98.5	17.3	0.0	91.8
<b>Square</b>	100.0	9.4	9.6	91.4	99.5	9.1	4.8	91.4	98.5	16.2	0.0	91.4
<b>Triangle</b>	99.5	4.6	9.8	91.6	98.5	6.2	7.8	91.7	99.0	12.3	0.0	90.8
<b>Noise</b>	99.5	6.1	9.9	91.5	99.0	6.7	7.9	91.5	99.5	7.3	0.0	91.4
<b>Mosaic</b>	96.5	10.6	18.0	91.6	99.5	12.0	4.6	91.3	99.0	14.2	0.0	90.7
<b>WaNet</b>	99.5	7.3	9.8	91.4	99.5	6.8	5.4	91.5	98.0	11.9	0.0	90.8

TABLE VII: Defense Performance of Coward under different watermark detection thresholds on CIFAR-10 dataset.

Threshold	TPR	FPR	ASR	BA
1%	98.0	1.3	10.2	91.5
5%	100.0	2.6	10.1	91.5
10%	100.0	8.6	10.1	91.4
15%	100.0	12.0	9.9	91.5
20%	100.0	13.9	10.0	91.2

**Robustness to Choices of Triggers.** Table VI summarizes detection performance under various combinations of trigger types. The attacker’s trigger varies in increasing stealthiness: from a visible pixel pattern, to an invisible blend pattern, and finally to an implicit semantic pattern naturally selected from the dataset. The watermark trigger also spans a range of types, including four visible patterns (diagonal, square, triangular, and noise), as well as two invisible variants: a mosaic pattern and the sample-specific WaNet pattern. We see that Coward performs consistently well across all settings, achieving a TPR above 96.5% and an FPR below 17.3%. These results demonstrate our method’s strong robustness to different triggers and the generalizability of collision effect.

**Robustness to Choices of Thresholds.** We further evaluate the impact of the detection threshold on the effectiveness of Coward. Since both attackers and benign clients may disrupt the watermark to some extent, the optimal threshold tends to fall within a relatively low range. Thus, we conduct threshold analysis within a mild range from 1% to 20%. As shown in Table VII, varying the detection threshold has minimal impact on TPR, which consistently remains above 98%. However, it progressively increases the FPR, rising from 1.0% to 13.9%. This increase is primarily due to client-side data heterogeneity. Certain benign clients, particularly those with severely skewed local distributions, may forget the watermark and exhibit significantly lower watermark accuracy than the majority. In contrast, malicious clients consistently show very low watermark accuracy. Empirically, we find that setting the threshold at 5% offers a reliable and practical guideline for a balanced TPR and FPR across diverse scenarios.

#### D. Resistance to Potential Adaptive Attack

**Potential Adaptive Attack Strategy.** We hereby consider adaptive attackers attempt to bypass our collision-based detection by injecting a similar watermark to preserve high watermark accuracy. Since the watermark target is chosen from a finite, known set, the attacker has a non-trivial chance

TABLE VIII: Performance of Coward under the adaptive attack (AA). The subscripts  $l$  and  $g$  denote the ASR of the attacker’s backdoor evaluated on the malicious local model and the aggregated global model, respectively.

Method	TPR	FPR	ASR <sub>g</sub>	ASR <sub>l</sub>
w/o. defense	/	/	91.2	95.6
w/o. AA.	100.0	5.0	10.1	98.9
AA. ( $T = 5$ )	99.5	4.4	10.1	90.4
AA. ( $T = 1$ )	62.5	5.4	22.9	73.9

of guessing it. To model this, we design a *periodic guessing attack* with switching period  $T$ , where the attacker updates the guessed label every  $T$  rounds. Smaller  $T$  increases the chance of early success but may cause unstable feedback, while larger  $T$  offers more stable signals at the cost of later success.

**Results.** As shown in Table VIII, the guessing attack appears to reduce the TPR but does not truly bypass the defense. For instance, when  $T = 1$ , the TPR drops from 100% to 62.5%, indicating some disruption due to a correct guess. However, the attacker’s main objective is to increase the ASR on the aggregated global model. Notably, the global ASR remains low across all adaptive settings, reaching only 22.9% at most when  $T = 1$ , which falls short to be successful. Upon further analysis, we find that the guessing attack inherently suffers from a fundamental contradiction, which we refer to as the *local collision contradiction*. Specifically, when the attacker injects an additional watermark with a different target label, it tends to induce a multi-backdoor conflict within the local model. This interference weakens the original backdoor and reduces the overall effectiveness of the attack. Overall, Coward remains resilient, as attackers struggle to maintain high ASR without compromising their own attack objectives.

#### E. Why is Our Method Effective?

To understand why our method works, we provide both overall and case-specific analyses on the CIFAR-10 dataset under the highly non-i.i.d. setting with  $\alpha = 0.3$ .

**Overall OOD Bias Analysis.** We evaluate the overall bias level under vanilla FL, BackdoorIndicator, and our method, and further investigate its correlation with detection false positive rate. Specifically, the prediction bias is quantified by the standard deviation of class-wise prediction probabilities, since uniform predictions are ideal for OOD samples, and any bias toward a specific class results in non-uniformity, thereby increasing the variance. By aggregating the bias across all participating clients over all training epochs, we obtain the overall bias level throughout the training process. As results shown in Figure 7, it is evident that: (1) *Our method yields the lowest bias distribution*, validating the effectiveness of regulating the OOD mapping and highlighting the adverse effect of BackdoorIndicator’s random mapping strategy. (2) *Even at high bias levels, our method achieves a lower FPR*, demonstrating its robustness against OOD bias.

**Case Study.** To intuitively explain how our method mitigates OOD bias, we present a set of representative detection cases of our method in Figure 8, ranging from an ideal non-bias scenario to increasingly biased conditions. Each case visualizes

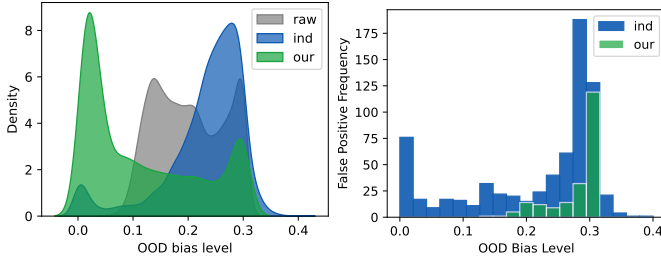


Fig. 7: **A Holistic Quantification of OOD Bias and Its Impact on FPR.** The left panel illustrates the OOD bias distribution, while the right panel shows the relationship between bias severity and the number of misjudged benign clients. Our method significantly reduces the OOD bias level, leading to a lower false positive rate, whereas the BackdoorIndicator increases the bias level and exhibits a high false positive frequency in highly biased regions.

the prediction confusion matrices of both raw and watermarked OOD samples, revealing how local training and OOD bias impact the two injected mappings. Specifically, the top two rows show the prediction results of raw OOD samples, while the bottom two rows show those of watermarked samples. Each case includes a small subfigure illustrating the cumulative distribution across target classes. To aid understanding, we highlight that the attack target is class 0, and the watermark target is class 8. We elaborate details as follows:

(1) **Ideal Cases:** Case 1 illustrates a malicious client (C0), where both the OOD and watermark mappings collapse into the attack target class 0, indicating severe OOD mapping distortion and a strong collision effect caused by the backdoor. This results in low watermark accuracy, enabling effective detection. In contrast, Case 2 represents an ideal benign client unaffected by OOD bias, where both the OOD (diagonal) and watermark mappings are fully preserved. Empirically, OOD bias seldom overrides backdoors but often impacts benign clients; Thus we examine biased cases for benign clients below. (2) **Biased Cases:** Case 3 presents a mild OOD bias where the OOD mapping is largely retained but partially biased toward class 9, yet the watermark prediction remains dominated by class 8. This highlights the strength of the watermark, which consistently guides triggered OOD samples to the target class, even when raw samples are misclassified. In Case 4, under a stronger bias where OOD predictions collapse into two dominant classes, the watermark still holds the highest confidence for class 8, indicating resilience even under severe distributional shifts. Intuitively, after watermark planting, the model forms a direct association between the trigger pattern and the target label, making it less influenced by the background image and thus more robust to OOD bias. Case 5 presents an extreme scenario where the OOD mapping collapses to a single class that coincidentally aligns with the watermark target. *These cases collectively indicate our robustness against different degrees of OOD bias.*

## VI. CONCLUSION

In this paper, we revisited backdoor defenses in federated learning, revealing the non-i.i.d. blurring and out-of-

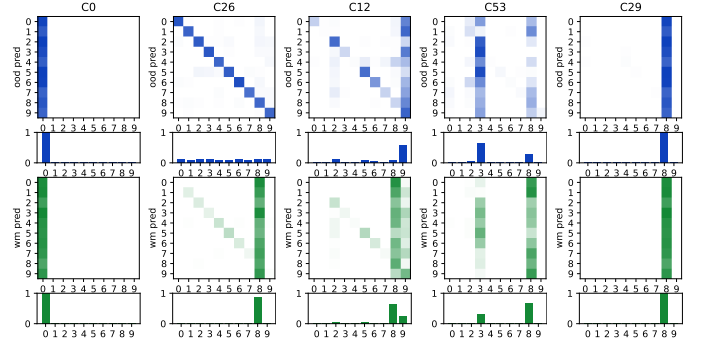


Fig. 8: **Case study of client detection with our method.** The first two columns illustrate ideal detection outcomes for malicious and benign clients, respectively. The remaining columns show successful identification of benign clients under different levels of OOD bias.

distribution (OOD) prediction bias that had constrained prior passive and proactive approaches. To tackle these challenges, we designed a new method, Coward, which leverages a collision-based OOD watermark that the server injects and later inspects to expose malicious clients, inspired by the phenomenon of *multi-backdoor collision effects*. By treating high, rather than low, watermark accuracy as the sign of benign behavior, Coward mitigated the side effects of OOD bias. Extensive experiments verified the effectiveness of Coward and its resilience against potential adaptive attacks. Our method provides a new perspective for understanding and defending backdoor attacks in federated learning, facilitating more secure and trustworthy data sharing and decentralized training.

## REFERENCES

- [1] B. McMahan, E. Moore, D. Ramage, S. Hampson *et al.*, “Communication-efficient learning of deep networks from decentralized data,” in *AISTATS*, 2017.
- [2] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, “Advances and open problems in federated learning,” *Foundations and trends® in machine learning*, vol. 14, no. 1–2, pp. 1–210, 2021.
- [3] W. Huang, M. Ye, Z. Shi, G. Wan, H. Li, B. Du, and Q. Yang, “Federated learning for generalization, robustness, fairness: A survey and benchmark,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 9387–9406, 2024.
- [4] M. Egger, R. Urbanke, and R. Bitar, “Federated one-shot learning with data privacy and objective-hiding,” *IEEE Transactions on Information Forensics and Security*, 2025.
- [5] Z. Charles, N. Mitchell, K. Pillutla, M. Reneer, and Z. Garrett, “Towards federated foundation models: Scalable dataset pipelines for group-structured learning,” in *NeurIPS*, 2023, pp. 32 299–32 327.
- [6] T. Fan, H. Gu, X. Cao, C. S. Chan, Q. Chen, Y. Chen, Y. Feng, Y. Gu, J. Geng, B. Luo *et al.*, “Ten challenging problems in federated foundation models,” *IEEE Transactions on Knowledge and Data Engineering*, 2025.
- [7] P. Mahon, I. Chatzitheofilou, A. Dekker, X. Fernández, G. Hall, A. Helland, A. Traverso, C. Van Marcke, J. Vehreschild, G. Ciliberto *et al.*, “A federated learning system for precision oncology in europe: Digione,” *nature medicine*, vol. 30, no. 2, pp. 334–337, 2024.
- [8] W. Ahmad, A. Vashist, N. Sinha, M. Prasad, V. Shrivastava, and J. H. Muzamal, “Enhancing transparency and privacy in financial fraud detection: The integration of explainable ai and federated learning,” in *SEDE*, 2024, pp. 139–156.
- [9] J. Wu, F. Dong, H. Leung, Z. Zhu, J. Zhou, and S. Drew, “Topology-aware federated learning in edge computing: A comprehensive survey,” *ACM Computing Surveys*, vol. 56, no. 10, pp. 1–41, 2024.



- [10] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE transactions on neural networks and learning systems*, vol. 35, no. 1, pp. 5–22, 2022.
- [11] Y. Dai and S. Li, "Chameleon: Adapting to peer images for planting durable backdoors in federated learning," in *ICML*, 2023.
- [12] S. Yang, J. Bai, K. Gao, Y. Yang, Y. Li, and S.-T. Xia, "Not all prompts are secure: A switchable backdoor attack against pre-trained vision transformers," in *CVPR*, 2024, pp. 24 431–24 441.
- [13] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *AISTATS*, 2020.
- [14] Q. Li, D. Wu, D. Zhou, C. Lin, S. Liu, C. Wang, and C. Shen, "Robust adversarial defenses in federated learning: Exploring the impact of data heterogeneity," *IEEE Transactions on Information Forensics and Security*, 2025.
- [15] K. N. Kumar, C. K. Mohan, and L. R. Cenkeramaddi, "Federated learning minimal model replacement attack using optimal transport: an attacker perspective," *IEEE Transactions on Information Forensics and Security*, 2024.
- [16] P. Rieger, T. D. Nguyen, M. Miettinen, and A.-R. Sadeghi, "DeepSight: Mitigating backdoor attacks in federated learning through deep model inspection," in *NDSS*, 2022.
- [17] T. D. Nguyen, P. Rieger, H. Chen, H. Yalame, H. Möllering, H. Ferdouni, S. Marchal, M. Miettinen, A. Mirhoseini, S. Zeitouni *et al.*, "{FLAME}: Taming backdoors in federated learning," in *USENIX Security*, 2022, pp. 1415–1432.
- [18] S. Li and Y. Dai, "{BackdoorIndicator}: Leveraging {OOD} data for proactive backdoor detection in federated learning," in *USENIX Security*, 2024, pp. 4193–4210.
- [19] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *CVPR*, 2015, pp. 427–436.
- [20] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *ICML*, 2017, pp. 1321–1330.
- [21] M. Hein, M. Andriushchenko, and J. Bitterwolf, "Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem," in *CVPR*, 2019, pp. 41–50.
- [22] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.
- [23] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.
- [24] T. A. Nguyen and A. T. Tran, "Wanet-imperceptible warping-based backdoor attack," in *ICLR*, 2021.
- [25] Y. Gao, Y. Li, X. Gong, Z. Li, S.-T. Xia, and Q. Wang, "Backdoor attack with sparse and invisible trigger," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 6364–6376, 2024.
- [26] E. Bagdasaryan and V. Shmatikov, "Blind backdoors in deep learning models," in *USENIX Security*, 2021, pp. 1505–1521.
- [27] X. Xu, Y. Chen, B. Wang, Z. Bian, S. Han, C. Dong, C. Sun, W. Zhang, L. Xu, and P. Zhang, "Csba: Covert semantic backdoor attack against intelligent connected vehicles," *IEEE Transactions on Vehicular Technology*, 2024.
- [28] J. Geiping, L. Fowl, W. R. Huang, W. Czaja, G. Taylor, M. Moeller, and T. Goldstein, "Witches' brew: Industrial scale data poisoning via gradient matching," *arXiv preprint arXiv:2009.02276*, 2020.
- [29] W. Sun, X. Zhang, H. Lu, Y. Chen, T. Wang, J. Chen, and L. Lin, "Backdoor contrastive learning via bi-level trigger optimization," in *ICLR*, 2024.
- [30] T. A. Nguyen and A. Tran, "Input-aware dynamic backdoor attack," in *NeurIPS*, 2020, pp. 3454–3464.
- [31] H. Cai, P. Zhang, H. Dong, Y. Xiao, S. Koffas, and Y. Li, "Toward stealthy backdoor attacks against speech recognition via elements of sound," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 5852–5866, 2024.
- [32] M. Zhu, Y. Li, J. Guo, T. Wei, S.-T. Xia, and Z. Qin, "Towards sample-specific backdoor attack with clean labels via attribute trigger," *IEEE Transactions on Dependable and Secure Computing*, 2025.
- [33] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" *arXiv preprint arXiv:1911.07963*, 2019.
- [34] Z. Zhang, A. Panda, L. Song, Y. Yang, M. Mahoney, P. Mittal, R. Kannan, and J. Gonzalez, "Neurotoxin: Durable backdoors in federated learning," in *ICML*, 2022, pp. 26 429–26 446.
- [35] B. Wang, Y. Yao, X. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Edge-case backdoor attacks against federated learning," in *NeurIPS Workshop*, 2020.
- [36] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *ICLR*, 2019.
- [37] X. Gong, Y. Chen, H. Huang, Y. Liao, S. Wang, and Q. Wang, "Coordinated backdoor attacks against federated learning with model-dependent triggers," *IEEE network*, vol. 36, no. 1, pp. 84–90, 2022.
- [38] T. Nguyen, D. T. Nguyen, K. D. Doan, and K.-S. Wong, "Non-cooperative backdoor attacks in federated learning: A new threat landscape," *arXiv preprint arXiv:2407.07917*, 2024.
- [39] B. Li, Y. Cai, H. Li, F. Xue, Z. Li, and Y. Li, "Nearest is not dearest: Towards practical defense against quantization-conditioned backdoor attacks," in *CVPR*, 2024, pp. 24 523–24 533.
- [40] X. Xu, K. Huang, Y. Li, Z. Qin, and K. Ren, "Towards reliable and efficient backdoor trigger inversion via decoupling benign features," in *ICLR*, 2024.
- [41] H. Qiu, Y. Zeng, S. Guo, T. Zhang, M. Qiu, and B. Thuraisingham, "Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation," in *AsiaCCS*, 2021, pp. 363–377.
- [42] B. Yi, T. Huang, S. Chen, T. Li, Z. Liu, Z. Chu, and Y. Li, "Probe before you talk: Towards black-box defense against backdoor unalignment for large language models," in *ICLR*, 2025.
- [43] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.
- [44] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *NeurIPS*, 2017.
- [45] M. S. Ozdayi, M. Kantarcioglu, and Y. R. Gel, "Defending against backdoors in federated learning with robust learning rate," in *AAAI*, 2021, pp. 9268–9276.
- [46] J. Zhang, C. Zhu, X. Sun, C. Ge, B. Chen, W. Susilo, and S. Yu, "Flpurifier: backdoor defense in federated learning via decoupled contrastive training," *IEEE Transactions on Information Forensics and Security*, 2024.
- [47] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *NeurIPS*, 2017.
- [48] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *RAID*, 2020, pp. 301–316.
- [49] X. Cao, M. Fang, J. Liu, and N. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," in *NDSS*, 2021.
- [50] Z. Ma, J. Ma, Y. Miao, Y. Li, and R. H. Deng, "Shieldfl: Mitigating model poisoning attacks in privacy-preserving federated learning," *IEEE Transactions on Information Forensics and Security*, 2022.
- [51] Y. Wang, D. Zhai, Y. Zhan, and Y. Xia, "Rflbat: A robust federated learning algorithm against backdoor attack," *arXiv preprint arXiv:2201.03772*, 2022.
- [52] X. Zhou, X. Chen, S. Liu, X. Fan, Q. Sun, L. Chen, M. Qiu, and T. Xiang, "Flguardian: Defending against model poisoning attacks via fine-grained detection in federated learning," *IEEE Transactions on Information Forensics and Security*, 2025.
- [53] Y. Li, J. He, H. Huang, J. Sun, and X. Ma, "Shortcuts everywhere and nowhere: Exploring multi-trigger backdoor attacks," *arXiv preprint arXiv:2401.15295*, 2024.
- [54] W. Qu, W. Zheng, T. Tao, D. Yin, Y. Jiang, Z. Tian, W. Zou, J. Jia, and J. Zhang, "Provably robust multi-bit watermarking for ai-generated text," in *USENIX Security*, 2025.
- [55] Y. Li, S. Shao, Y. He, J. Guo, T. Zhang, Z. Qin, P.-Y. Chen, M. Backes, P. Torr, D. Tao *et al.*, "Rethinking data protection in the (generative) artificial intelligence era," *arXiv preprint arXiv:2507.03034*, 2025.
- [56] S. Shao, Y. Li, H. Yao, Y. He, Z. Qin, and K. Ren, "Explanation as a watermark: Towards harmless and multi-bit model ownership verification via watermarking feature attribution," in *NDSS*, 2025.
- [57] G. Gan, Y. Li, D. Wu, and S.-T. Xia, "Towards robust model watermark via reducing parametric vulnerability," in *ICCV*, 2023, pp. 4751–4761.
- [58] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "Emnist: Extending mnist to handwritten letters," in *IJCNN*, 2017, pp. 2921–2926.
- [59] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep. TR-2009, 2009.
- [60] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.
- [61] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *MLSys*, 2020.
- [62] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.
- [63] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *AISTATS*, 2020, pp. 2938–2948.



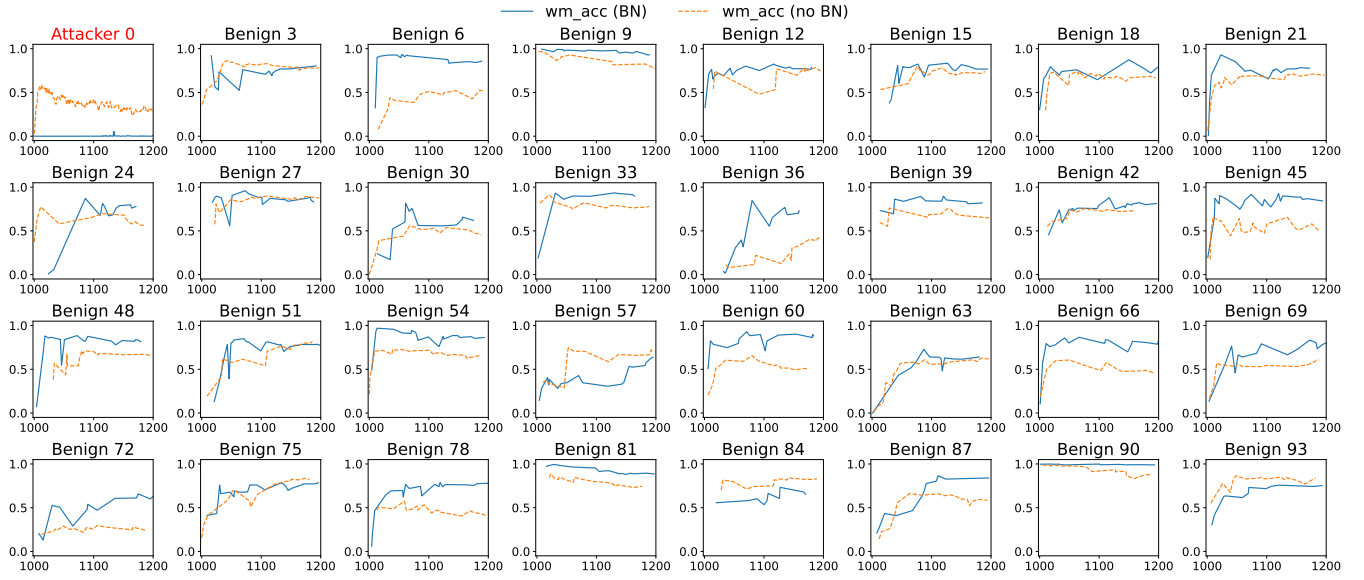


Fig. 1: **OOD watermark collision under dynamic FL scenario.** The collision effect remains highly effective in distinguishing malicious behavior under dynamic federated participation. The attacker exhibits a strong collision effect, while benign clients show diverse but generally higher levels of watermark retention.

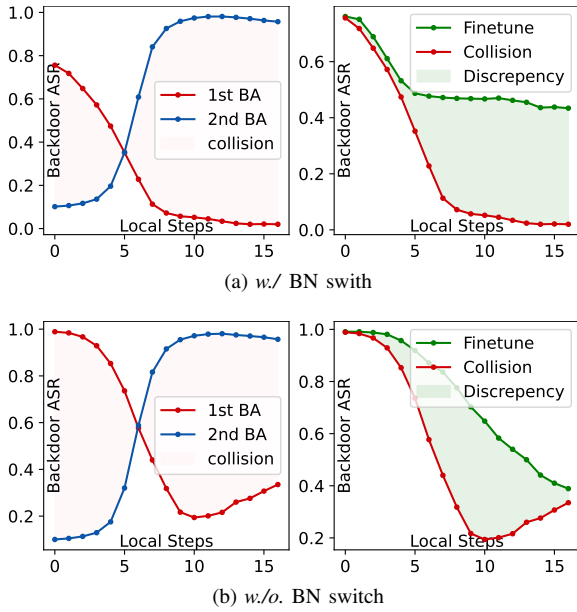


Fig. 2: **OOD watermark collision under centralized scenario.** Our OOD watermark is planted as the second backdoor. The resulting collision effect is significant, regardless of whether the BN layer is switched. However, switching the BN layer creates a more pronounced performance discrepancy between benign finetuning and backdoor injection behaviors.

## APPENDIX A COLLISION EFFECTS ON OOD WATERMARK

In Section III-C, we introduced the multi-backdoor collision effect under a centralized setting where two backdoors share the same dataset, aiming to illustrate its core mechanism. To further demonstrate how this effect manifests in our OOD-based watermark under practical FL scenarios, we provide an extended analysis in this section. Specifically, we examine the

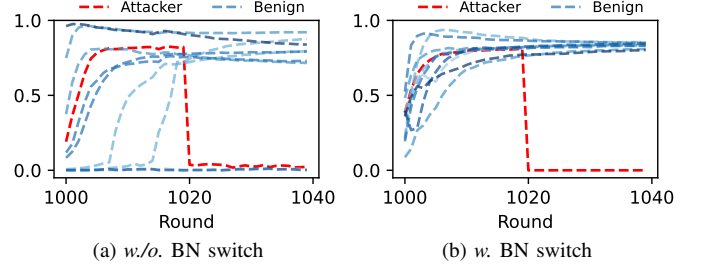


Fig. 3: **OOD watermark collision under static FL scenario.** Once the attacker (in red) begins injecting the backdoor at round 1021, its watermark accuracy rapidly drops to 0, indicating a strong collision effect and a clear distinction from benign clients. Moreover, the BN switch plays a critical role in accurately reflecting the benign clients' (in varying shades of blue) ability to retain the watermark.

collision effect between the attacker's backdoor and our OOD watermark across progressively complex scenarios: starting from a centralized setup, moving to a static FL environment with fixed client participation, and finally to the dynamic FL scenario adopted in our main experiments. To align with main experiments, we conduct all evaluations under the 0.9 non-i.i.d. setting on the CIFAR-10 dataset by default. As elaborated in the following sections, results across all three stages consistently reveal a clear collision effect, validating the capability of our OOD watermark to trigger such interactions and effectively distinguish malicious clients.

**Centralized Scenario.** We conduct a similar experiment as in Section III-C, with the key difference that the second backdoor injection is replaced by our OOD watermark planting process. Specifically, we use a target label 8 watermark with MNIST as the planting set. As shown in Fig. 2, in each subfigure, the left side illustrates the collision effect, while

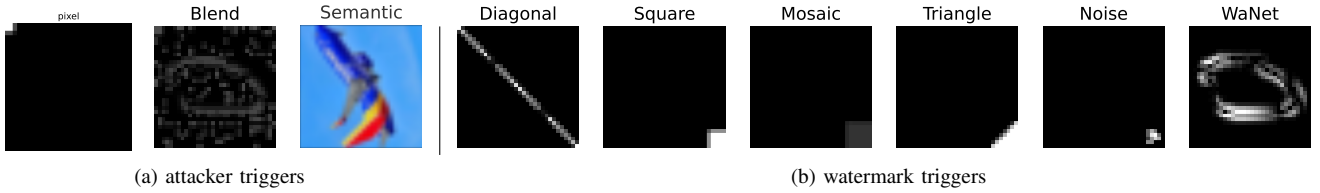


Fig. 4: Visualization of trigger configurations.

the right side shows the discrepancy between benign (*i.e.*, finetuning) and malicious (*i.e.*, backdoor injection) operation. Specifically, in subfigure (a), we observe a clear collision effect on the OOD watermark, characterized by a clear drop in ASR and a notable distinction between benign and malicious clients. In subfigure (b), when the BN layer switch is disabled, the benign-malicious discrepancy becomes indistinguishable, indicating the necessity of BN isolation for effective detection.

**Static FL Scenario.** We consider a slightly more complex scenario under a static FL setting, involving 10 clients with full participation in each communication round. Among them, only one client is malicious. This simplified full-participation setup facilitates the observation of each client’s temporal behavior and helps illustrate the collision effect in a federated learning context. In each global round, the server injects an OOD watermark, while all benign clients perform standard local training. To clearly observe the collision effect, the attacker is configured to behave benignly for the first 20 rounds and initiates a backdoor attack targeting label 0 from round 21 onward. After each local update, we inspect the watermark accuracy of all 10 participated clients. As shown in Figure 3(b), once the malicious client begins the attack, its watermark accuracy drops sharply to nearly zero, indicating a strong collision effect. In contrast, the remaining nine benign clients consistently maintain high watermark accuracy. This sharp divergence highlights the benign-malicious discrepancy induced by the collision effect, which is critical for enabling effective backdoor detection. Additionally, in subfigure (a), we observe that benign clients struggle to stably preserve the watermark without BN layer isolation, underscoring the importance of BN switching in reducing FPR.

**Dynamic FL Scenario.** We finally present the collision effect under the dynamic FL setting used in our main experiments, where 10 out of 100 clients are randomly selected to participate in each global round. This dynamic participation leads to divergent and inconsistent watermark retention behaviors among benign clients, making direct comparisons between them infeasible. Therefore, we focus on observing relative trends across rounds. Specifically, we plot the temporal variation of watermark accuracy for all clients based on their actual participation order. To control page length, we select clients at intervals of 3 (*i.e.*, client IDs with a gap of 3). As shown in Figure 1, we observe the following: (1) Malicious client 0 exhibits a pronounced collision effect, with watermark accuracy dropping to nearly zero. (2) Benign clients maintain relatively high watermark accuracy (mostly above 50%), though the magnitude and variation patterns differ across clients due to heterogeneous participation and

TABLE I: Performance of Coward with different OOD Base Mapping on the CIFAR-10 dataset.

OOD Mapping	TPR	FPR	ASR	BA
Default	100.0	2.4	9.8	91.3
Shift+1	99.5	3.0	9.8	91.5
Shift+3	99.5	2.9	9.8	91.5
Shift+5	100.0	2.6	9.7	91.3

TABLE II: Performance of Coward for all watermark labels.

Label ( $y_m$ )	TPR	FPR	ASR	BA
1	100.0	9.2	9.9	91.4
2	100.0	6.5	9.8	91.5
3	100.0	10.4	10.0	91.5
4	100.0	5.1	9.8	91.5
5	100.0	10.1	9.9	91.2
6	100.0	5.6	9.9	91.5
7	100.0	13.6	9.7	91.5
8	98.5	4.0	10.2	91.6
9	100.0	17.8	9.7	91.4

client distribution. Overall, the results reveal a clear separation between benign and malicious clients, enabling robust and tolerant threshold selection. Despite being influenced by dynamic participation and non-i.i.d. data, the collision effect remains highly effective in distinguishing malicious behavior.

## APPENDIX B DISCUSSION ON WATERMARK CONFIGURATION

Our detection paradigm is flexible with respect to specific watermark configurations, allowing practical deployment scenarios to select or design their own. To concretely demonstrate this flexibility, we provide empirical evidence by evaluating constituent components of our OOD watermark: the OOD base mapping, the watermark trigger, and the watermark target. To align with main experiments, we conduct all evaluations under the 0.9 non-i.i.d. setting on the CIFAR-10 dataset by default.

**OOD Base Mapping.** The key role of the OOD base mapping is to provide a fixed foundation for building the watermark mapping. As long as the mapping remains class-wise and fixed, the specific choice of mapping is not critical. While we adopt a diagonal mapping in the main experiments for simplicity, we demonstrate here that our approach is compatible with other mapping configurations, highlighting its flexibility. Specifically, we replace the default diagonal mapping (*i.e.*,  $i \rightarrow i$ ) with three alternative 1-to-1 shift-based mappings:  $i \rightarrow (i+3) \bmod 10$ ,  $i \rightarrow (i+5) \bmod 10$ , and  $i \rightarrow (i+7) \bmod 10$ . As shown in Table I, all variants achieve consistently strong detection performance, indicating that Coward does not rely on specific OOD mapping rule. These results confirm that our

TABLE III: Ablation study of Coward on watermark injection strength.

Size	TPR	FPR	ASR	BA
200	100.0	23.7	10.0	91.5
500	98.0	2.2	10.1	91.6
1000	99.0	4.2	10.1	91.4
1500	100.0	3.1	10.1	91.1
2000	100.0	3.3	10.1	91.4

(a) Varying size of planting set

wm_lr	TPR	FPR	ASR	BA
0.0001	100.0	7.1	10.1	91.5
0.0005	99.0	5.6	10.0	91.6
0.001	100.0	3.3	9.8	91.4
0.01	98.0	1.8	9.9	91.5

(b) Varying watermark learning rate

wm_rds	TPR	FPR	ASR	BA
1	100.0	11.6	10.2	91.6
2	100.0	6.4	9.9	91.3
5	100.0	3.3	9.8	91.4
10	100.0	5.7	10.0	91.4
20	100.0	4.2	9.9	91.2

(c) Varying injection rounds

TABLE IV: Performance of Coward under different levels of global model regularization.

$\lambda$	TPR	FPR	ASR	BA
0	93.5	6.5	10.3	90.4
0.1	96.0	6.8	10.2	90.1
0.3	99.0	10.9	10.3	90.4
0.5	97.0	9.6	10.2	90.4

method is robust to the choice of OOD assignment and can be seamlessly adapted to different mapping schemes.

**Trigger Type.** Previously, we have already presented quantitative results demonstrating the general effectiveness of the collision effect and the corresponding detection mechanism across various trigger pairs in Table VI. To provide a more intuitive understanding, we further visualize the appearance of the triggers. As illustrated in Figure 4, our server-malicious trigger pairs include both semantically similar combinations (*e.g.*, a server-side square pattern and a client-side pixel-style trigger) and semantically dissimilar ones (*e.g.*, a server-side WaNet trigger and a client-side semantic trigger). Notably, both types consistently yield strong detection performance. These results indicate that the effectiveness of our approach is largely independent of the visual semantics of the triggers, offering defenders a broad design space to customize trigger appearances according to their specific needs.

**Watermark Label.** In the main experiments, we fixed the watermark target label to 8 for simplicity. However, we emphasize that the effectiveness of Coward is generally applicable to any target label. To validate this flexibility, we present results in Table II, where the attack target is fixed to label 0 and the watermark label is varied across all possible values. The consistently strong detection performance across different watermark labels confirms the adaptability of our approach in selecting the watermark target.

#### APPENDIX C

##### ABLATION STUDY ON WATERMARK INJECTION

Beyond the design choices of the watermark itself, an equally important yet distinct problem is how to effectively implant the watermark into the global model. In this section, we conduct ablation studies on all factors that influence the implantation process. To align with our main experiments, we conduct all evaluations under the 0.9 non-IID setting on the CIFAR-10 dataset by default.

**Impact of Global Model Regularization.** Table IV reports the effect of varying levels of global model regularization. We observe a positive impact on TPR, where applying regularization consistently yields higher TPRs compared to the no-regularization baseline. Increasing the regularization strength

$\lambda$  leads to slight improvements in TPR, with the best performance observed at  $\lambda = 0.3$ . These results suggest that a moderate level of regularization is beneficial for achieving optimal detection performance.

**Learning Efficiency of Watermark Injection.** We conduct ablation studies on three key hyperparameters that control the strength of watermark injection: the number of OOD watermark samples, the watermark learning rate, and the number of injection iterations per global round. We vary one factor at a time while keeping the others fixed. By default, the watermark learning rate is set to 0.001, the number of watermark injection rounds to 5, and the watermark dataset size to 1000. As shown in Table III, Coward consistently achieves strong performance, with TPR exceeding 98% and FPR remaining below 10% in most cases. Even under extremely constrained injection settings (*e.g.*, a watermark dataset size of 200, a learning rate of 0.0001, or a single injection round), the method still maintains competitive results, achieving 100% TPR with FPR below 25%. Overall, effective watermark injection requires only a low level of effort. Most parameter variations lead to only marginal performance differences, and any configuration beyond the most extreme cases yields satisfactory performance. These results indicate that our method is robust across a wide range of watermark injection strengths and introduces negligible computational overhead in federated learning. *Notably, effective defense can be achieved with as few as only 1 round of watermark injection.*

#### APPENDIX D

##### POTENTIAL LIMITATIONS AND FUTURE DIRECTIONS

While our method provides a simple and effective early solution for proactive defense, there remain several open directions that worth further exploration and refinement in future works on this topic:

- **Fine-grained OOD Watermark Design.** Our current OOD watermark adopts a simple yet effective configuration applied uniformly across the entire planting set, using widely adopted trigger patterns. Exploring more complex and fine-grained designs that potentially leverage the diversity of OOD samples (*e.g.*, sample-specific or class-conditional watermark mappings) would be a valuable future direction.
- **Dynamic Watermarking Strategy.** In the current version, a fixed watermark configuration is used throughout training, enabling effective injection and maintaining a consistent collision signal. Nevertheless, our framework is also flexible enough to support dynamic watermark configurations (*i.e.*, across training rounds or even client-specific setups). This adaptability, when paired with carefully designed injection

methods, would open up a new direction for enhancing the robustness and flexibility of the defense strategy.

- **Multi-Metric and Multi-Mechanism Detection.** While our method have already achieved effective detection using a single metric (*i.e.*, watermark accuracy) with clear and consistent signals, the broader flexibility of the proposed framework allows for further expansion. Specifically, it can support joint observation of client responses to multiple operations, opening up the possibility of incorporating diverse detection mechanisms. For example, one may simultaneously leverage the maintenance and collision effects by separately monitoring OOD mappings and watermark mappings. This enriched perspective could provide complementary signals and represents a promising future direction.