CCCV: CROSS-CLIENT CONSISTENCY VERIFICATION FOR HIGH-PRECISION BACKDOOR DEFENSE IN VERTICAL FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vertical federated learning (VFL) enables parties with disjoint feature sets of the same samples to collaboratively train and infer without sharing raw data. However, VFL is vulnerable to clean-label backdoor attacks, where one or more malicious clients manipulate its representation to induce targeted misclassification while preserving benign labels and training behavior. Existing defenses can detect such attacks but often misclassify atypical benign representations, leading to high false-positive rates and utility degradation, while some methods additionally require auxiliary trusted data. Our key insight is to directly estimate the conditional density of each client representation given the others. Backdoor representations create a mismatch with the cross-client context and thus receive low conditional likelihood. Compared with reconstruction-based methods that approximate conditional consistency through a point estimate and other defenses that rely on auxiliary trusted data or indirect anomaly signals, direct density estimation enables more precise detection with fewer false positives and less performance degradation without requiring additional trusted data. Based on this insight, we propose *Cross-Client Consistency Verification (CCCV)*, an inference-time defense combining lightweight semantic screening with contrastively enhanced conditional normalizing flows. CCCV requires no trusted data and does not modify the optimization of the original VFL model, while supporting backdoor detection, attacker attribution, and prediction recovery.

1 INTRODUCTION

Federated learning (FL) enables multiple parties to jointly train models without sharing raw data (McMahan et al., 2017). It is commonly divided into Horizontal Federated Learning (HFL), where parties hold different samples with similar feature spaces, and Vertical Federated Learning (VFL), where they share samples but possess complementary features. By supporting collaborative learning across distributed feature spaces without exchanging raw data, VFL facilitates cross-organizational data collaboration (Yang et al., 2019).

Despite these advantages, the growing adoption of FL has raised significant security concerns. Prior studies have shown that federated models are vulnerable to backdoor attacks, where adversaries inject trigger patterns to induce attacker-chosen predictions (Bagdasaryan et al., 2020; Xie et al., 2020). However, defenses developed for centralized learning or HFL are often difficult to transfer directly to VFL due to its split architecture. Each party observes only a subset of features and controls only part of the model, which prevents the direct use of defenses requiring complete inputs, model internals, or end-to-end gradients (Wang et al., 2019; Gao et al., 2019). This challenge is further amplified by the clean-label nature of VFL backdoor attacks. During training, triggers are injected into legitimate target-class samples, so poisoned representations may remain semantically plausible and are difficult to separate using conventional anomaly-based methods such as activation clustering or spectral signatures (Chen et al., 2018; Tran et al., 2018). At inference time, although triggered non-target samples may become more distinguishable, dataset-level clustering or spectral analysis is unsuitable for real-time prediction. Meanwhile, HFL-style robust aggregation is also inapplicable because VFL inference depends on complementary embeddings from all parties, and

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

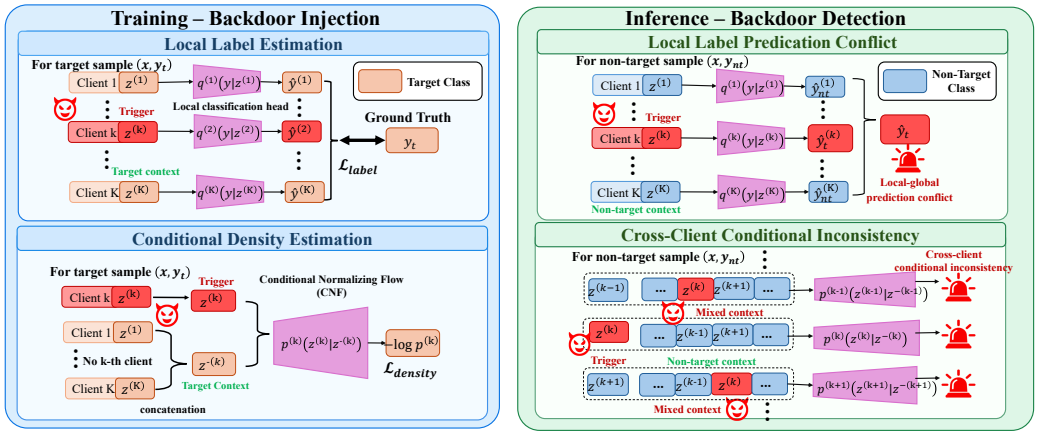


Figure 1: Overview of the proposed cross-client consistency verification framework.

simply discarding a suspicious participant may remove essential feature information (Yin et al., 2018; Cao et al., 2021).

To address these challenges, several recent studies have developed backdoor defenses specifically for VFL. These methods can be broadly divided into training-time defenses, which suppress backdoor implantation by constraining malicious updates or filtering suspicious samples (Liu et al., 2020; Abadi et al., 2016; Hu et al., 2026), and post-training or inference-time defenses, which detect and mitigate implanted backdoors through anomalous predictions, embeddings, or cross-party inconsistencies (Cho et al., 2024; Chen et al., 2025; Yuan et al., 2025; Cheng et al., 2026). However, existing approaches still suffer from three major limitations. **(1) Low Detection Precision.** Anomaly-based methods (Cho et al., 2024; Yuan et al., 2025; Cheng et al., 2026) detect poisoned samples through abnormal embedding or prediction statistics, but benign outliers may exhibit similar scores, especially at low poisoning ratios. For example, VFLIP may falsely flag benign samples with rare yet legitimate embeddings as poisoned based on reconstruction estimation. **(2) Utility Degradation.** Existing defenses (Cho et al., 2024; Hu et al., 2026) may impair benign model performance when removing backdoor effects. For example, FILTER (Hu et al., 2026) applies unlearning to eliminate poisoned knowledge, but this process may also erase useful information learned from benign samples and degrade main-task accuracy. **(3) Auxiliary Information Requirement.** Existing defenses (Chen et al., 2025) may rely on auxiliary information that is difficult to obtain in practical VFL deployments. For example, UBD (Chen et al., 2025) requires clean validation data for latent-trigger synthesis and model purification.

Our Contribution. To overcome these limitations, we observe that clean-label backdoor poisoning induces a training-inference shift in cross-client context. During training, the trigger co-occurs with target-class representations, whereas at inference it is attached to a non-target sample whose other clients retain non-target semantics. Thus, the triggered representation may appear plausible in isolation, but become incompatible with its cross-client context. This motivates evaluating whether each client representation is supported by the conditional distribution $p(z^{(k)} \mid z^{-(k)})$, where $z^{(k)}$ denotes the sub-embedding produced by client k, and $z^{-(k)}$ denotes the concatenation of sub-embeddings from all remaining clients. This distribution captures the probability of client k's representation conditioned on the semantic context from the other participants, thus, providing a direct measure of cross-client compatibility. This formulation highlights a key distinction: a benign representation may appear unusual in isolation yet remain consistent with the representations of other clients, whereas a triggered representation is incompatible with the cross-client context. We estimate the conditional distribution from representations and supervision already available during standard VFL training, without requiring auxiliary data or modifying the top model, thereby preserving the utility of the underlying model.

Based on this insight, we propose *Cross-Client Consistency Verification (CCCV)*, an inference-time backdoor defense that verifies cross-client consistency at both semantic and representation levels, as shown in Figure 1. During training, CCCV learns a lightweight local label estimator and a conditional density estimator for each client without altering the original VFL optimization. The former

captures the semantic class tendency of each sub-embedding, while the latter models its conditional likelihood given the remaining clients' representations using conditional normalizing flows (Winkler et al., 2019), further enhanced by cross-class contrastive learning. At inference time, CCCV first screens samples according to local prediction consistency and then applies conditional density verification to suspicious samples. Detected cross-client inconsistencies are further leveraged for attacker attribution and prediction recovery.

Our contributions are summarized as follows:

Cross-client context shift. We identify a training–inference context shift in clean-label VFL backdoors, where triggered representations become incompatible with non-target cross-client contexts at inference. Rather than relying on point reconstruction, we model the full conditional distribution to directly quantify such incompatibility as a principled detection signal.

A novel detection framework. We propose CCCV, combining lightweight semantic screening with probabilistic conditional-consistency verification. Using contrastively enhanced conditional normalizing flows to model $p(z^{(k)} \mid z^{-(k)})$, CCCV captures diverse benign patterns and detects backdoor-induced inconsistencies without trusted clean data.

High-precision and utility-preserved backdoor detection. We extensively evaluate CCCV across diverse datasets, representative VFL backdoor attacks, and different VFL settings. The results show that CCCV consistently achieves high detection precision with few false positives while effectively suppressing backdoor attacks and preserving benign-model utility. For example, on CIFAR-10 against BadVFL, CCCV achieves **88.29%** detection precision and **98.00%** recall, reduces ASR from **98.00%** to **2.00%**, while maintaining **78.23%** overall test accuracy.

2 Related Work

2.1 Backdoor Attacks and Defenses in VFL

Recent studies (Liu et al., 2025; Shen et al., 2025; Bai et al., 2023; Naseri et al., 2024; He et al., 2023) show that VFL is vulnerable to clean-label backdoor attacks, where passive adversaries manipulate local features or representations to induce targeted predictions without altering server-held labels. Since attackers typically lack label access, existing attacks mainly rely on label inference and representation manipulation. For example, labels can be inferred from model structures or gradients (Fu et al., 2022), while LMP/PMP (Liu et al., 2025), LFBA (Shen et al., 2025), BadVFL (Naseri et al., 2024), and VILLAIN (Bai et al., 2023) exploit inferred label information to poison local representations. He et al. (2023) further proposed a SplitNN attack that replaces target-class embeddings with a scaled trigger vector.

Existing VFL backdoor defenses can be broadly categorized into training-time and post-training methods. Training-time defenses suppress backdoor implantation through gradient perturbation or sample filtering. Representative approaches include gradient clipping/compression (Bai et al., 2023; Liu et al., 2020), DP-SGD (Abadi et al., 2016), and FILTER (Hu et al., 2026). While effective in limiting malicious updates, these methods may also disrupt benign optimization and degrade main-task accuracy (Xuan et al., 2023). Post-training and inference-time defenses instead detect or mitigate implanted backdoors from suspicious representations or predictions. CONFIRM (Cheng et al., 2026) uses prediction stability under top-model perturbations, VFLIP (Cho et al., 2024) detects and reconstructs anomalous embeddings, and Ma et al. (2026) model cross-participant semantic consistency with a latent masked autoencoder. UBD (Chen et al., 2025) synthesizes and suppresses latent triggers, while GBD (Yuan et al., 2025) detects backdoors through class activation statistics.

2.2 Conditional Normalizing Flow

Normalizing flows (NFs) model complex distributions through invertible transformations, enabling exact likelihood evaluation via the change-of-variables formula (Dinh et al., 2017). More expressive variants, such as Neural Spline Flows (Durkan et al., 2019), further improve density modeling while preserving tractable likelihoods. Conditional normalizing flows (CNFs) extend NFs by incorporating contextual information, allowing direct estimation of conditional densities (Winkler et al., 2019), which is well suited to modeling context-dependent representations.

3 PRELIMINARIES

3.1 VERTICAL FEDERATED LEARNING

Following the standard VFL formulation (Bai et al., 2023; Naseri et al., 2024; He et al., 2023; Shen et al., 2025), we consider K passive clients $\{C^{(k)}\}_{k=1}^{K}$ and a server S collaboratively training a classifier over aligned samples $\mathcal{D}$. For each sample $i \in \mathcal{D}$, client $C^{(k)}$ holds private features $\mathbf{x}_i^{(k)} \in \mathbb{R}^{d^{(k)}}$, with $\mathbf{x}_i = [\mathbf{x}_i^{(1)}, \ldots, \mathbf{x}_i^{(K)}]$, while only S has access to the label $y_i \in \mathcal{Y}$. Each client computes a local representation $\mathbf{z}_i^{(k)} = f^{(k)}(\mathbf{x}_i^{(k)}; \theta^{(k)})$, which is sent to S and concatenated as $\mathbf{z}_i = [\mathbf{z}_i^{(1)}, \ldots, \mathbf{z}_i^{(K)}]$. The server then predicts $\hat{y}_i = f_{top}(\mathbf{z}_i; \theta_{top})$. The collaborative objective is

$$\min_{\theta_{top}, \{\theta^{(k)}\}_{k=1}^{K}} \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \ell\Big(f_{top}\Big([f^{(1)}(\mathbf{x}_i^{(1)}), \ldots, f^{(K)}(\mathbf{x}_i^{(K)})]\Big), y_i\Big), \tag{1}$$

where $\ell(\cdot, \cdot)$ denotes the cross-entropy loss. Raw features remain local throughout training, and only intermediate representations and optimization information are exchanged.

3.2 CLEAN-LABEL BACKDOOR ATTACKS AGAINST VFL

In VFL, only the server S has access to the ground-truth labels, while adversarial passive clients $\{C_{k_a} \mid k_a \in \mathcal{I}_{adv}\}$ typically cannot observe or modify them. We therefore focus on *clean-label backdoor attacks*, where poisoned samples retain their original labels (Naseri et al., 2024; Bai et al., 2023). During training, the adversary associates a trigger δ with a target class y_t. At test time, δ is injected into samples from a source class y_s or arbitrary non-target classes to induce misclassification into y_t.

Given the diversity of clean-label backdoor attacks (Naseri et al., 2024; Bai et al., 2023; He et al., 2023), we formulate their common procedure into two stages: *label inference* and *backdoor injection*. First, the adversary identifies target-class samples $D_{target}^{(k_a)} = \{(\mathbf{x}_{i_t}^{(k_a)}, y_t)\}$ using auxiliary data or an adversarial classifier. It then applies a trigger-generation function $\mathcal{T}$ with trigger $\boldsymbol{\delta}$: $\widetilde{\mathbf{x}}_{i_t}^{(k_a)} = \mathcal{T}(\mathbf{x}_{i_t}^{(k_a)}, \boldsymbol{\delta})$. Since these samples originate from the target class y_t, their labels remain unchanged. Repeated co-occurrence of $\boldsymbol{\delta}$ and y_t during training encourages the model to learn a strong association between them:

$$\min_{\theta_{top}} \mathbb{E}_{x^{(k_a)} \sim D_{target}^{(k_a)}} \ell\left(f_{top}\left([\cdots, f^{(k_a)}(\mathcal{T}(\mathbf{x}_{i_t}^{(k_a)}, \boldsymbol{\delta})), \cdots]\right), y_t\right) \tag{2}$$

At inference time, the adversary activates the backdoor by applying $\boldsymbol{\delta}$ to an arbitrary non-target sample $(\mathbf{x}_{i_s}^{(k_a)}, y_{i_s})$, where $y_{i_s} \neq y_t$, yielding $\widetilde{\mathbf{x}}_{i_s}^{(k_a)} = \mathcal{T}(\mathbf{x}_{i_s}^{(k_a)}, \boldsymbol{\delta})$. The attack aims to misclassify the triggered sample as y_t while preserving correct predictions on clean inputs. Thus, a successful attack achieves high attack success rate (ASR) with minimal degradation in clean-data utility.

Remark. A VFL backdoor attack typically proceeds in two phases. During training, the adversary injects a trigger into selected samples to associate it with a target class. At inference, the trigger is applied to selected non-target class inputs to induce their misclassification into the attacker-specified target class.

4 THREAT MODEL

Attacker's Goal and Capabilities. We consider a strong adversary with full label knowledge that aims to make triggered inputs predict an attacker-specified target label while preserving clean accuracy and overall model utility. The adversary (1) controls at most 50% of participating passive clients; (2) knows all training labels; (3) can choose attack timing and inject predefined triggers into local features or intermediate embeddings; and (4) cannot access benign clients' local models or the server-side top model, nor manipulate server-held labels.

Defender's Goal and Capabilities. We consider the server as the defender, whose goal is to detect and mitigate backdoor attacks while preserving prediction accuracy on benign samples. The

defender (1) has access to the top model and training labels; (2) can observe intermediate embeddings uploaded by participating clients; (3) cannot access clients' raw features or local models; and (4) cannot directly modify clients' local data or computations.

5 CROSS-CLIENT CONSISTENCY VERIFICATION (CCCV)

5.1 MOTIVATION AND KEY INTUITION

Detecting clean-label backdoors from VFL representations is challenging because representation rarity does not imply maliciousness. Benign outliers may also exhibit unusual embeddings or large reconstruction errors. The key distinction is *cross-client compatibility*. During training, the trigger is associated with target-class contexts, whereas at inference it is combined with non-target samples whose other clients retain non-target semantics. This creates a conditional-context mismatch: the triggered representation becomes incompatible with the complementary representations. Unlike a benign outlier, which may be marginally rare yet cross-client compatible, a backdoor sample induces mutual inconsistency between the malicious and benign clients. Thus, conditional incompatibility provides a more targeted detection signal than representation rarity alone.

Motivated by this observation, we model $p(z^{(k)} \mid z^{-(k)})$ to estimate full conditional distribution rather than relying on point reconstruction. The framework mainly uses two key signals:
Local-global prediction conflict A subset of clients may strongly support the global prediction, while the remaining clients exhibit inconsistent local semantic predictions.
Cross-client conditional inconsistency The triggered representation may appear individually plausible, but becomes statistically incompatible with the non-target representations from the remaining clients. Moreover, the resulting incompatibility is inherently cross-client.

5.2 TRAINING: LOCAL LABEL ESTIMATION

Given a training sample i, the VFL model predicts $\hat{y}_i = \arg\max_y f_{top}([\mathbf{z}_i^{(1)}, \ldots, \mathbf{z}_i^{(K)}])$, where $\mathbf{z}_i^{(k)} = f^{(k)}(\mathbf{x}_i^{(k)})$ is the sub-embedding from client k.

To capture each client representation's semantic tendency, we introduce a lightweight local label estimator

$$\hat{y}_i^{(k)} = \arg\max_y q^{(k)}(y \mid \mathbf{z}_i^{(k)}; \phi_{\text{label}}^{(k)}), \tag{3}$$

where $q^{(k)}(\cdot)$, parameterized by $\phi_{\text{label}}^{(k)}$, estimates the label distribution from $\mathbf{z}_i^{(k)}$ alone. It serves only as an auxiliary consistency estimator and does not modify the original VFL classifier.

Each estimator is trained jointly on detached embeddings using

$$\mathcal{L}_{\text{label}}^{(k)} = -\frac{1}{N}\sum_{i=1}^{N} \log q^{(k)}\left(y_i \mid \mathbf{z}_i^{(k)}\right). \tag{4}$$

5.3 TRAINING: CONDITIONAL DENSITY ESTIMATION

To capture whether the representations produced by different clients are mutually consistent, we learn a conditional density estimator for each client. The main idea is to determine the probability of a particular client's sub-embedding given the sub-embeddings provided by all other clients.

For client k, we denote the concatenation of the other clients' sub-embeddings as $\mathbf{z}^{-(k)} = [\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(k-1)}, \mathbf{z}^{(k+1)}, \ldots, \mathbf{z}^{(K)}]$. We then learn a conditional probabilistic model

$$p^{(k)}\left(\mathbf{z}^{(k)} \mid \mathbf{z}^{-(k)}; \phi_{\text{condition}}^{(k)}\right) \tag{5}$$

which models the distribution of the sub-embedding $\mathbf{z}^{(k)}$ produced by client k, conditioned on the $\mathbf{z}^{-(k)}$ observed from the remaining clients. Specifically, we instantiate $p^{(k)}(\mathbf{z}^{(k)} \mid \mathbf{z}^{-(k)})$ using a **conditional normalizing flow** (Winkler et al., 2019). Let $\mathbf{u}^{(k)} = g_{\phi_{\text{condition}}^{(k)}}(\mathbf{z}^{(k)}; \mathbf{z}^{-(k)})$ denote an invertible transformation that maps $\mathbf{z}^{(k)}$, conditioned on $\mathbf{z}^{-(k)}$, into a base latent variable $\mathbf{u}^{(k)}$

following a simple base distribution. In our implementation, $g_{\phi^{(k)}_{\text{condition}}}$ is parameterized by a conditional RealNVP model (Winkler et al., 2019; Dinh et al., 2017) with $\mathbf{u}^{(k)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Because the transformation is invertible, the conditional density of an observed representation can be evaluated exactly using the change-of-variables formula:

$$\Gamma^{(k)}(\mathbf{z}) = -\log p^{(k)}\left(\mathbf{z}^{(k)} \mid \mathbf{z}^{-(k)}\right) = -\log p\left(\mathbf{u}^{(k)}\right) - \log\left|\det \frac{\partial g_{\phi^{(k)}_{\text{condition}}}\left(\mathbf{z}^{(k)}; \mathbf{z}^{-(k)}\right)}{\partial \mathbf{z}^{(k)}}\right|. \quad (6)$$

Here, $p(\mathbf{u}^{(k)})$ denotes the probability density of $\mathbf{u}^{(k)}$ under the standard Gaussian base distribution, while the Jacobian determinant accounts for the local volume change introduced by the invertible transformation. Therefore, unlike reconstruction-error-based anomaly detectors, the conditional flow provides an explicit likelihood for the observed representation. We further record the conditional density value $\Gamma^{(k)}(\mathbf{z}_i)$ together with its ground-truth label y_i for each training example $(\mathbf{x}_i, y_i)$. These reference statistics, $(\Gamma^{(k)}(\mathbf{z}_i), y_i)$, are maintained in a bounded FIFO buffer, indexed by client k and class y and used for subsequent inference-time consistency screening.

Each conditional density estimator is trained independently by maximum likelihood:

$$\mathcal{L}^{(k)}_{\text{density}} = -\frac{1}{N}\sum_{i=1}^{N} \log p^{(k)}\left(\mathbf{z}_i^{(k)} \mid \mathbf{z}_i^{-(k)}\right). \quad (7)$$

Minimizing this objective encourages high likelihood for representation combinations observed during training, allowing the learned conditional density to capture cross-client dependencies and identify statistically incompatible representations at inference.

Backdoor Shift. The conditional density estimator also learns from backdoored representations during training. For an adversarial client k_a and target-class sample i_t, the triggered sub-embedding is $\tilde{\mathbf{z}}_{i_t}^{(k_a)} = f^{(k_a)}(\mathcal{T}(\mathbf{x}_{i_t}^{(k_a)}, \boldsymbol{\delta}))$. Under clean-label attacks, $\tilde{\mathbf{z}}_{i_t}^{(k_a)}$ is observed with its target-class context $\mathbf{z}_{i_t}^{-(k_a)}$, so $p^{(k_a)}$ learns this trigger-context association rather than treating the trigger itself as anomalous. At inference, the same trigger $\boldsymbol{\delta}$ is applied to a non-target sample i_s, producing $\tilde{\mathbf{z}}_{i_s}^{(k_a)}$ while the remaining clients retain the non-target context $\mathbf{z}_{i_s}^{-(k_a)}$. This breaks the learned association and induces a conditional distribution shift:

$$p^{(k_a)}\left(\tilde{\mathbf{z}}_{i_s}^{(k_a)} \mid \mathbf{z}_{i_s}^{-(k_a)}\right) \ll p^{(k_a)}\left(\tilde{\mathbf{z}}_{i_t}^{(k_a)} \mid \mathbf{z}_{i_t}^{-(k_a)}\right), \quad (8)$$

providing the key signal for inference-time backdoor detection.

Augmentation. To improve sensitivity to cross-client inconsistency, we augment maximum-likelihood training with a contrastive objective. For each matched pair $(\mathbf{z}_i^{(k)}, \mathbf{z}_i^{-(k)})$, we construct a cross-class mismatched pair $(\mathbf{z}_i^{(k)}, \mathbf{z}_j^{-(k)})$ with $y_i \neq y_j$. We define

$$\mathcal{L}^{(k)}_{\text{contrastive}} = \frac{1}{|\mathcal{P}|}\sum_{(i,j)\in\mathcal{P}} T \cdot \text{softplus}\left(\frac{\log p^{(k)}\left(\mathbf{z}_i^{(k)} | \mathbf{z}_j^{-(k)}\right) - \log p^{(k)}\left(\mathbf{z}_i^{(k)} | \mathbf{z}_i^{-(k)}\right) + m}{T}\right), \qquad y_i \neq y_j, \quad (9)$$

where $\mathcal{P} = \{(i,j) \mid i,j \in \mathcal{B},\, y_i \neq y_j\}$, $\mathcal{B}$ denotes a mini-batch, $\text{softplus}(x) = \log(1+\exp(x))$, T is the temperature, and m is the contrastive margin. Minimizing this loss encourages matched contexts to receive higher conditional likelihood than cross-class mismatched contexts.

The final objective is

$$\mathcal{L}^{(k)}_{\text{condition}} = \mathcal{L}^{(k)}_{\text{density}} + \lambda_{\text{con}}\mathcal{L}^{(k)}_{\text{contrastive}}, \quad (10)$$

where λ_{con} controls the contrastive regularization strength. This explicitly teaches the estimator to assign low likelihood to cross-client incompatible representations.

5.4 Inference: Local Label Inconsistency Screening

At inference, we first perform lightweight semantic-consistency screening using the local label estimators. Given a test sample i, the global prediction is $\hat{y}_i = \arg\max_y f_{top}(\mathbf{z}_i)$. We define the

fraction of clients supporting $\hat{y}_i$ as

$$R(\mathbf{z}_i) = \frac{1}{K}\sum_{k=1}^{K}\mathbb{I}[\hat{y}_i^{(k)} = \hat{y}_i], \qquad \hat{y}_i^{(k)} = \arg\max_y q^{(k)}(y \mid \mathbf{z}_i^{(k)}). \tag{11}$$

If $R(\mathbf{z}_i) \geq \tau_s$, the sample is directly accepted, otherwise, it is forwarded to conditional-density screening for further verification. The τ_s denotes the local support threshold.

Meanwhile, for a gated sample, we define the *global-support set* $\mathcal{S}(\mathbf{z}_i) = \{k : \hat{y}_i^{(k)} = \hat{y}_i\}$, which provides an asymmetric signal for attacker localization. When only a minority of clients support the global prediction, the clients in $\mathcal{S}(\mathbf{z}_i)$ become particularly suspicious, forming a **minority-majority conflict**.

However, local prediction inconsistency alone is insufficient for reliable detection: atypical clean samples may also exhibit disagreement, and failed backdoor attacks may obscure the adversarial client. Therefore, gated samples are further verified by conditional-density inconsistency screening.

5.5 Inference: Conditional Density Inconsistency Screening

For every sample satisfying $R(\mathbf{z}_i) < \tau_s$, we evaluate all K conditional density estimators and compute the conditional inconsistency statistic

$$\Gamma^{(k)}(\mathbf{z}_i) = -\log p^{(k)}(\mathbf{z}_i^{(k)} \mid \mathbf{z}_i^{-(k)}). \tag{12}$$

Intuitively, $\Gamma^{(k)}$ measures how surprising the sub-embedding $\mathbf{z}_i^{(k)}$ uploaded by client k is when interpreted jointly with the other sub-embeddings uploaded by all other clients. Large $\Gamma^{(k)}$ corresponds to low conditional density and indicates strong cross-client incompatibility.

During training, we record $\Gamma^{(k)}(\mathbf{z}_i)$ with its label y_i. At inference, given the global prediction $\hat{y}_i$, we use the client-class-specific threshold $\gamma^{(k),y} = Q_\eta(\{\Gamma^{(k)}(\mathbf{z}_j) \mid y_j = y\})$, and select $\gamma^{(k),\hat{y}_i}$ for client k, where $Q_\eta(\cdot)$ denotes the empirical η-th percentile.

We then quantify the degree to which client k exceeds its reference range using the anomaly margin

$$A_i^{(k)} = \Gamma^{(k)}(\mathbf{z}_i) - \gamma^{(k),\hat{y}_i}. \tag{13}$$

Thus, $A_i^{(k)} > 0$ indicates that the conditional inconsistency of client k exceeds the reference level associated with the globally predicted class.

Finally, we aggregate the anomaly margins across clients and declare sample i as anomalous when

$$\sum_{k=1}^{K} A_i^{(k)} > \Delta, \qquad \hat{y}_i = \arg\max_y f_{\text{top}}(\mathbf{z}_i), \tag{14}$$

where Δ is a hyperparameter controlling the required total abnormal margin. This aggregation captures whether the sample exhibits sufficiently strong cross-client conditional inconsistency beyond that normally observed for class $\hat{y}_i$.

This discrepancy reflects **cross-client conditional inconsistency**, which helps distinguish backdoor samples from atypical clean samples. While benign outliers may be marginally rare, their cross-client representations can remain conditionally compatible. In contrast, backdoor samples typically exhibit large anomaly margins due to the mismatch between triggered and benign sub-embeddings, enabling effective detection with a low false-positive rate.

Pattern Recording. For each detected backdoor sample i, we maintain a lightweight cross-batch memory $\mathcal{P}$ to capture persistent client-level prediction patterns by recording $(\mathcal{S}(\mathbf{z}_i), \hat{y}_i)$.

5.6 Post-Detection Mitigation

Attacker Attribution. We use the accumulated pattern memory $\mathcal{P}$ to identify malicious clients. For a detected backdoor sample j, we count the historical occurrences of its global-support pattern $(\mathcal{S}(\mathbf{z}_j), \hat{y}_j)$:

$$\sum_{(\mathcal{S},y)\in\mathcal{P}} \mathbb{I}[\mathcal{S} = \mathcal{S}(\mathbf{z}_j) \wedge y = \hat{y}_j] > \zeta, \tag{15}$$

where ζ is the frequency threshold. If the pattern recurs sufficiently often, the clients in $\mathcal{S}(\mathbf{z}_j)$ are attributed as attackers and recorded as $\mathcal{M}(y_t) = \mathcal{S}(\mathbf{z}_j)$ with $y_t = \hat{y}_j$.

Label Recovery.

For a detected backdoor sample j, we attribute attackers by matching local predictions to the historical target-specific sets: $Adv_j = \{k \mid k \in \mathcal{M}(\hat{y}_j^{(k)})\}$. If no match exists, we use the global-support set $Adv_j = \mathcal{S}(\mathbf{z}_j)$. We then zero-mask the attributed sub-embeddings and recover the prediction as

$$\hat{y}_j^{\text{re}} = \arg\max_y f_{\text{top}}\left(\mathbf{z}_j \odot \mathbf{m}_j\right), \tag{16}$$

where $\mathbf{m}_j$ masks the suspected attackers, suppressing their backdoor influence.

6 Experiment

6.1 Experiment Setup

Datasets In this work, we evaluate CCCV on five datasets: MNIST (LeCun et al., 1998), CIFAR-10 (Krizhevsky et al., 2009), CINIC-10 (Darlow et al., 2018), Bank Marketing (Bank) (Moro et al., 2014), and Vehicle (Duarte & Hu, 2004). The first three are image classification datasets, while Bank and Vehicle are tabular datasets.

VFL Setting By default, we consider a VFL system with two passive clients and one server, where features are equally and non-overlappingly partitioned across clients. We use a two-layer MLP as the top model for all datasets. The bottom models are ResNet-18 (He et al., 2016) for CIFAR-10 and CINIC-10, a two-layer CNN for MNIST, a four-layer MLP for Bank, and a three-layer MLP for Vehicle. The learning rate is 10^{-3} for all datasets.

Attack Setting We evaluate CCCV against three representative VFL backdoor attacks: VILLAIN (Bai et al., 2023), BadVFL (Naseri et al., 2024), and the SplitNN-specific data poisoning attack with the Optimal Trigger strategy (He et al., 2023).

Defense Baselines We compare our method with recent representative VFL backdoor defenses, including FILTER (Hu et al., 2026), a training-time method that filters suspicious samples using losses and embedding characteristics, and three test-time methods: VFLIP (Cho et al., 2024), based on representation reconstruction; CONFIRM (Cheng et al., 2026), based on prediction consistency under top-model perturbations; and GBD (Yuan et al., 2025), based on class activation probability.

Implementation Details For the attacks, we follow common settings and set the training poisoning rate to 1%. Importantly, at test time, 1% of test samples are triggered, and ASR is computed only over these samples. For CCCV, we set the local-support threshold $\tau_s = 0.8$, the percentile threshold $\eta = 0.90$. The anomaly-margin threshold Δ is set to $100, 55, 40, 20, 120$ for CIFAR-10, CINIC-10, MNIST, Vehicle, and Bank, respectively. For contrastive augmentation, we set $\lambda_{\text{con}} = 1$, temperature $T = 1$, and margin $m = 1$. For historical tracking, we set $\zeta = 30$.

Evaluation Metrics We report mixed test accuracy (ACC) and attack success rate (ASR) to evaluate model utility and defense effectiveness, respectively. For detection-based defenses, we additionally report precision, recall, and F1 score to evaluate detection performance.

6.2 Experiment Results

CCCV vs. Existing Defense Methods. As shown in Table 1, CCCV consistently maintains competitive test accuracy (ACC) while substantially reducing the attack success rate (ASR) across different attacks and datasets. In most cases, CCCV achieves ACC comparable to or higher than the undefended model, indicating that the proposed defense introduces little utility degradation. GBD obtains slightly higher ACC in a few settings. For ASR, CCCV consistently suppresses attacks to low levels across all settings. Although some baselines achieve slightly lower ASR in individual cases, this often results from more aggressive detection thresholds and consequently higher false-positive rates.

Table 2 further demonstrates the advantage of CCCV in detection quality. CCCV achieves substantially higher precision than all competing methods while still maintaining consistently high recall especially under a very small poisoning rate (1%). As a result, CCCV obtains the highest F1 score across all evaluated attack–dataset combinations, showing a better balance between detecting poisoned samples and avoiding false positives. In contrast, VFLIP relies on reconstruction-based anomaly signals and tends to flag atypical benign representations, while GBD also exhibits a strong

Table 1: Comparison of CCCV with other defenses against three VFL backdoor attacks.

Attack	Defense	CIFAR-10 (%)		CINIC-10 (%)		MNIST (%)		Vehicle (%)		Bank (%)	
		ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
BadVFL	/	*77.65*	*98.00*	*65.70*	*99.78*	*97.73*	*100.00*	*84.38*	*97.97*	*84.18*	*100.00*
	VFLIP	75.50	3.00	64.57	5.33	96.73	**0.00**	82.77	2.03	82.12	3.11
	FILTER	72.78	**0.00**	59.29	7.56	97.18	**0.00**	75.26	**0.00**	77.74	**0.00**
	CONFIRM	72.35	**0.00**	59.33	**0.11**	97.54	**0.00**	83.12	99.49	**85.45**	96.67
	GBD	78.16	2.00	65.45	6.56	98.08	2.00	**85.30**	1.02	82.48	1.11
	CCCV (*)	**78.23**	**2.00**	**66.28**	**4.44**	**98.61**	**0.00**	**84.53**	**4.06**	**84.78**	**1.11**
VILLAIN	/	*77.70*	*99.00*	*67.25*	*95.67*	*97.97*	*98.00*	*84.58*	*98.48*	*84.58*	*98.48*
	VFLIP	76.22	2.00	63.67	2.33	93.90	2.00	84.24	8.63	82.40	5.56
	FILTER	71.70	**1.00**	63.41	**0.33**	94.22	**0.00**	83.24	**0.00**	84.70	**4.44**
	CONFIRM	75.40	8.00	63.23	3.56	97.04	**0.00**	83.36	100.00	**85.61**	95.56
	GBD	**78.87**	**1.00**	64.40	3.89	98.37	1.00	**85.23**	0.51	80.73	**4.44**
	CCCV (*)	**78.38**	**5.00**	**67.55**	**6.44**	**98.49**	**0.00**	**85.02**	**0.00**	**82.53**	**4.44**
Opt.	/	*77.92*	*100.00*	*66.70*	*96.67*	*97.96*	*100.00*	*84.46*	*98.98*	*84.38*	*100.00*
	VFLIP	73.84	2.00	65.87	1.33	92.87	**0.00**	82.12	14.72	82.11	7.78
	FILTER	72.92	**0.00**	64.74	17.11	95.60	1.00	79.24	**0.00**	**85.77**	2.22
	CONFIRM	74.09	97.00	61.72	100.00	96.43	100.00	84.53	100.00	73.95	93.33
	GBD	77.70	1.00	64.91	**0.78**	98.18	7.00	**85.56**	2.54	82.30	**0.00**
	CCCV (*)	**78.54**	**4.00**	**67.17**	**5.00**	**98.80**	**0.00**	**84.90**	**0.00**	**84.64**	**1.11**

For defense methods, (*) denotes our proposed defense. ACC and ASR denote accuracy and attack success rate, respectively. Opt. denotes the Optimal Trigger attack. The best results are highlighted in **red**.

Table 2: Backdoor detection performance of CCCV and other defenses against three VFL attacks.

Attack	Defense	CIFAR-10 (%)			CINIC-10 (%)			MNIST (%)			Vehicle (%)			Bank (%)		
		Prec.↑	Recall↑	F1↑	Prec.↑	Recall↑	F1↑	Prec.↑	Recall↑	F1↑	Prec.↑	Recall↑	F1↑	Prec.↑	Recall↑	F1↑
BadVFL	VFLIP	4.16	**100.00**	8.00	13.90	**99.88**	24.41	33.11	**100.00**	49.75	9.59	**100.00**	17.50	2.25	**100.00**	4.40
	CONFIRM	10.44	93.00	18.77	9.98	89.56	17.96	9.49	**100.00**	17.33	0.01	0.51	0.03	0.00	0.00	0.00
	GBD	10.56	95.00	19.00	16.62	92.11	28.16	20.21	96.00	33.39	6.02	98.98	11.35	2.01	95.56	3.93
	CCCV (*)	**88.29**	**98.00**	**92.89**	**93.38**	**95.56**	**94.45**	**81.97**	**100.00**	**90.09**	**68.98**	**95.94**	**80.25**	**49.72**	**98.89**	**66.17**
VILLAIN	VFLIP	16.84	**100.00**	28.82	7.81	**100.00**	14.49	13.14	**100.00**	23.23	27.86	**100.00**	43.58	1.10	**100.00**	2.17
	CONFIRM	9.20	96.00	16.80	13.75	**100.00**	24.17	7.91	83.00	14.45	0.00	0.00	0.00	0.00	0.00	0.00
	GBD	15.56	98.00	26.85	10.45	94.56	18.82	27.79	97.00	43.21	4.57	95.43	8.72	1.79	90.00	3.50
	CCCV (*)	**96.00**	**96.00**	**96.00**	**74.01**	**97.78**	**84.25**	**81.30**	**100.00**	**89.69**	**72.43**	**100.00**	**84.01**	**62.94**	**100.00**	**77.25**
Opt.	VFLIP	5.86	**100.00**	11.07	7.14	**100.00**	13.33	9.30	**100.00**	17.02	6.35	**100.00**	11.95	1.07	**100.00**	2.11
	CONFIRM	0.38	3.00	0.67	2.73	33.11	5.04	0.00	0.00	0.00	0.16	13.33	0.32	0.16	13.33	0.32
	GBD	21.32	94.00	34.75	13.46	99.22	23.70	15.82	93.00	27.03	6.82	97.46	12.74	4.61	**100.00**	8.81
	CCCV (*)	**85.96**	**98.00**	**91.59**	**80.21**	**95.00**	**86.98**	**82.64**	**100.00**	**90.50**	**82.08**	**100.00**	**90.16**	**58.06**	**100.00**	**73.47**

For defense methods, (*) denotes our proposed defense. Opt. denotes the Optimal Trigger attack. The best results are highlighted in **red**.

precision–recall imbalance. FILTER can suppress attacks effectively in some cases, but its training-time filtering may noticeably reduce test accuracy. CONFIRM additionally relies on correctly identifying the target class from within-class distributions. Thus, incorrect selection can lead to ineffective detection and poor ASR/recall. We also provide additional experiments and discussions in Appendix A and Appendix B, respectively.

7 Conclusion

In this work, we propose CCCV, an inference-time defense for clean-label backdoor attacks in vertical federated learning. CCCV detects backdoor samples by identifying cross-client semantic and conditional inconsistencies, without requiring trusted auxiliary data. It further supports attacker attribution and prediction recovery after detection. Extensive experiments across multiple datasets and attack settings show that CCCV achieves effective backdoor mitigation while preserving benign prediction performance.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and improving the clarity and presentation of the manuscript. We have not used generative AI tools to generate experimental results, data, or scientific conclusions, and other disclosure tasks are not applicable to this work. We have reviewed and revised all AI-assisted text to ensure its technical accuracy and consistency with the conducted experiments and presented results. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed CCCV framework, including its training objectives and inference procedure, in the main text. The experimental setup describes the datasets, VFL architectures, attack settings, defense baselines, evaluation metrics, and major implementation settings used in our experiments. The source code will be made publicly available upon acceptance, together with the camera-ready version of the paper.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2938–2948. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/bagdasaryan20a.html.

Yijie Bai, Yanjiao Chen, Hanlei Zhang, Wenyuan Xu, Haiqin Weng, and Dou Goodman. {VILLAIN}: Backdoor attacks against vertical split learning. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 2743–2760, 2023.

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

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.

Peng Chen, Haolong Xiang, Xin Du, Xiaolong Xu, Xuhao Jiang, Zhihui Lu, Jirui Yang, Qiang Duan, and Wanchun Dou. Universal backdoor defense via label consistency in vertical federated learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 4743–4751, 2025.

Haoran Cheng, Lan Zhang, Quanchao Liu, Yetian He, Junyang Wang, and Peng Ran. Leveraging confidence consistency for poisoned sample detection in vertical federated learning. *IEEE Transactions on Information Forensics and Security*, 2026.

Yungi Cho, Woorim Han, Miseon Yu, Younghan Lee, Ho Bae, and Yunheung Paek. Vflip: A backdoor defense for vertical federated learning via identification and purification. In *European Symposium on Research in Computer Security*, pp. 291–312. Springer, 2024.

Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Marco F Duarte and Yu Hen Hu. Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838, 2004.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X Liu, and Ting Wang. Label inference attacks against vertical federated learning. In *31st USENIX security symposium (USENIX Security 22)*, pp. 1397–1414, 2022.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, pp. 113–125, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Ying He, Zhili Shen, Jingyu Hua, Qixuan Dong, Jiacheng Niu, Wei Tong, Xu Huang, Chen Li, and Sheng Zhong. Backdoor attack against split neural network-based vertical federated learning. *IEEE Transactions on Information Forensics and Security*, 19:748–763, 2023.

Zhanyi Hu, Cen Chen, and Yanhao Wang. Filter: A framework for defending against backdoor attacks in vertical federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35490–35499, 2026.

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

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Junlin Liu, Xinchen Lyu, Chenshan Ren, and Qimei Cui. Targeted poisoning attacks against vertical federated learning via embedding manipulation. *IEEE Transactions on Dependable and Secure Computing*, 2025.

Yang Liu, Zhihao Yi, and Tianjian Chen. Backdoor attacks and defenses in feature-partitioned collaborative learning. *arXiv preprint arXiv:2007.03608*, 2020.

Xiaoya Ma, Zijun Zhang, Kun He, Jing Chen, and Ruiying Du. N truths and a lie: Consistency-based backdoor defense for vertical federated learning. *IEEE Transactions on Information Forensics and Security*, 2026.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017. URL `https://proceedings.mlr.press/v54/mcmahan17a.html`.

Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

Mohammad Naseri, Yufei Han, and Emiliano De Cristofaro. Badvfl: Backdoor attacks in vertical federated learning. In *2024 IEEE symposium on security and privacy (SP)*, pp. 2013–2028. IEEE, 2024.

Wei Shen, Wenke Huang, Guancheng Wan, and Mang Ye. Label-free backdoor attacks in vertical federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 20389–20397, 2025.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/280cf18baf4311c92aa5a042336587d3-Paper.pdf.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE symposium on security and privacy (SP)*, pp. 707–723. IEEE, 2019.

Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

Chulin Xie, Keli Huang, Pin Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

Yuexin Xuan, Xiaojun Chen, Zhendong Zhao, Bisheng Tang, and Ye Dong. Practical and general backdoor attacks against vertical federated learning. In *Joint European Conference on machine learning and knowledge discovery in databases*, pp. 402–417. Springer, 2023.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5650–5659. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/yin18a.html.

Shunjie Yuan, Xinghua Li, Xuelin Cao, Haiyan Zhang, and Robert H Deng. General test-time backdoor detection in split neural network-based vertical federated learning. *IEEE Transactions on Dependable and Secure Computing*, 2025.

APPENDIX A EXTRA EXPERIMENT

Table A1: Ablation study of CCCV under three VFL backdoor attacks.

Method	BadVFL (%)					Villain (%)					Optimal Trigger (%)				
	ACC↑	ASR↓	Prec↑	Recall↑	F1↑	ACC↑	ASR↓	Prec↑	Recall↑	F1↑	ACC↑	ASR↓	Prec↑	Recall↑	F1↑
No Defense	*78.18*	*96.00*	*0.00*	*0.00*	*0.00*	*78.63*	*99.00*	*0.00*	*0.00*	*0.00*	*78.43*	*100.00*	*0.00*	*0.00*	*0.00*
Only Local Label Estimator	53.18	6.00	2.00	**99.00**	3.92	58.51	2.00	2.27	**100.00**	4.44	57.40	**1.00**	2.14	**99.00**	4.19
CCCV w/o Contrastive Augmentation	78.52	5.00	75.59	96.00	84.58	78.75	5.00	89.91	98.00	93.78	**79.21**	9.00	83.64	92.00	87.62
CCCV	**78.56**	**2.00**	**90.00**	**99.00**	**94.29**	**79.58**	**1.00**	**96.12**	**99.00**	**97.54**	**78.54**	**6.00**	**92.23**	**95.00**	**93.60**

The 'Prec' denotes precision. ↑ and ↓ indicate that higher and lower values are better, respectively. The best results are highlighted in **red**.

Ablation Study. We conduct an ablation study to evaluate the contribution of the key components in CCCV. As shown in Table A1, using only the local label estimator achieves nearly perfect recall under all three attacks, but yields precision of only around 2% and substantially degrades main task test accuracy. This confirms that local prediction inconsistency provides a sensitive screening signal, but is insufficient for reliable detection because benign samples may also exhibit local-global disagreement. Introducing conditional density verification substantially improves detection precision to 75.59%, 89.91%, and 83.64% under BadVFL, VILLAIN, and Optimal Trigger, respectively, while restoring test accuracy close to the undefended model. This shows that explicit modeling $p(z^{(k)} \mid z^{-(k)})$ is crucial to distinguish backdoor cross-client incompatibility from benign semantic disagreement. Finally, incorporating contrastive augmentation further improves precision to 90.00%, 96.12%, and 92.23%, with corresponding F1 scores of 94.29%, 97.54%, and 93.60%. These results show that contrastive augmentation strengthens the separation between compatible and mismatched cross-client embeddings, improving conditional density estimation to achieve both high detection precision and strong attack mitigation.

Table A2: Performance of CCCV with different numbers of clients under three VFL backdoor attacks.

K	Defense	BadVFL (%)					Villain (%)					Optimal Trigger (%)				
		ACC↑	ASR↓	Prec↑	Recall↑	F1↑	ACC↑	ASR↓	Prec↑	Recall↑	F1↑	ACC↑	ASR↓	Prec↑	Recall↑	F1↑
4	*No Defense*	*72.75*	*95.00*	*0.00*	*0.00*	*0.00*	*73.90*	*94.00*	*0.00*	*0.00*	*0.00*	*74.66*	*100.00*	*0.00*	*0.00*	*0.00*
	CCCV	**73.58**	**0.00**	**75.76**	**100.00**	**86.21**	**73.92**	**3.00**	**83.90**	**99.00**	**90.83**	**74.63**	**2.00**	**83.05**	**98.00**	**89.91**
6	*No Defense*	*71.49*	*98.00*	*0.00*	*0.00*	*0.00*	*72.36*	*99.00*	*0.00*	*0.00*	*0.00*	*72.66*	*100.00*	*0.00*	*0.00*	*0.00*
	CCCV	**70.09**	**0.00**	**80.65**	**100.00**	**89.29**	**72.30**	**1.00**	**72.46**	**100.00**	**84.03**	**72.31**	**3.00**	**69.29**	**97.00**	**80.83**
8	*No Defense*	*69.68*	*96.00*	*0.00*	*0.00*	*0.00*	*69.59*	*100.00*	*0.00*	*0.00*	*0.00*	*70.37*	*100.00*	*0.00*	*0.00*	*0.00*
	CCCV	**70.10**	**0.00**	**84.75**	**100.00**	**91.74**	**69.45**	**1.00**	**75.57**	**99.00**	**85.71**	**70.06**	**0.00**	**70.92**	**100.00**	**82.99**

The 'Prec' denotes precision. The ↑ and ↓ indicate that higher and lower values are better, respectively.

Performance in Multi-participant Scenarios. We further evaluate CCCV in multi-participant VFL scenarios on CIFAR-10, with $K = 4, 6, 8$ clients, where 50% of the clients are malicious. As shown in Table A2, CCCV remains effective even when multiple malicious clients participate in the attack. For example, under BadVFL, CCCV reduces the ASR to 0% for all three settings, while achieving 100% recall and improving the detection F1 score from 86.21% at $K = 4$ to 91.74% at $K = 8$. Basically, CCCV consistently reduces the ASR to a low level while maintaining high detection recall, demonstrating that the proposed cross-client consistency verification can generalize beyond the two client setting. We also observe that ACC decreases as K increases. This is mainly because the input features are split more finely among a larger number of clients, so each client has less information available for learning its local representation. A similar trend is observed without defense. For example, under BadVFL, the ACC of No Defense decreases from 72.75% at $K = 4$ to 69.68% at $K = 8$, while CCCV changes from 73.58% to 70.10%. This suggests that the accuracy drop mainly comes from feature fragmentation in the multi-participant setting rather than CCCV itself.

Table A3: Malicious-client attribution accuracy (%) at the epoch and sample levels.

K	BadVFL(%)		Villain(%)		Optimal Trigger(%)	
	Epoch-level Attribution	Sample-level Attribution	Epoch-level Attribution	Sample-level Attribution	Epoch-level Attribution	Sample-level Attribution
4	100.00	87.00	100.00	96.00	100.00	98.00
6	100.00	77.00	100.00	92.00	100.00	94.00
8	100.00	84.00	100.00	93.00	100.00	96.00

Epoch-level attribution identifies the malicious client using evidence aggregated over an epoch, while sample-level attribution identifies the malicious client for each individual sample.

Attacker Attribution. We further evaluate malicious-client attribution in the multi-participant settings. We report attribution accuracy at both the sample and epoch levels. **Sample-level attribution** is counted as correct only when the predicted malicious-client set exactly matches the complete ground-truth attacker set for that sample. **Epoch-level attribution** is evaluated under a strictly online setting: attribution evidence is accumulated sequentially throughout the epoch without revisiting future samples, and the resulting attacker set is compared with the ground-truth attacker set at the end of the epoch.

As shown in Table A3, CCCV achieves **100% epoch-level attribution accuracy** across all $K \in \{4, 6, 8\}$ settings and all three attacks. At the sample level, the attribution accuracy remains high, reaching 77-87% for BadVFL, 92-96% for VILLAIN, and 94-98% for Optimal Trigger. The remaining sample-level errors mainly occur during the *memory warm-up stage*: before a reliable attack pattern has been accumulated, occasional prediction errors from the local label estimators may produce an incorrect client-support pattern. CCCV alleviates this issue through its cross-batch pattern memory, which accumulates recurring client-level prediction patterns for attacker attribution. Once the corresponding attack pattern is sufficiently established and matched in memory, the historical target-specific attacker set can be directly reused, making subsequent attribution almost always correct. This also explains why aggregating attribution evidence over an epoch consistently yields 100% attribution accuracy despite occasional errors on individual samples.

Table A4: Performance of CCCV under different evaluation poisoning rates against three VFL backdoor attacks.

P.R.	Defense	BadVFL (%)					Villain (%)					Optimal Trigger (%)				
		ACC↑	ASR↓	Prec↑	Recall↑	F1↑	ACC↑	ASR↓	Prec↑	Recall↑	F1↑	ACC↑	ASR↓	Prec↑	Recall↑	F1↑
1%	*No Defense*	*78.51*	*94.00*	*0.00*	*0.00*	*0.00*	*78.99*	*98.00*	*0.00*	*0.00*	*0.00*	*77.81*	*100.00*	*0.00*	*0.00*	*0.00*
	CCCV	**79.01**	**6.00**	**94.95**	**94.00**	**94.47**	**79.53**	**5.00**	**88.79**	**95.00**	**91.79**	**78.29**	**6.00**	**86.24**	**94.00**	**89.95**
10%	*No Defense*	*70.89*	*98.60*	*0.00*	*0.00*	*0.00*	*71.46*	*97.70*	*0.00*	*0.00*	*0.00*	*71.69*	*100.00*	*0.00*	*0.00*	*0.00*
	CCCV	**76.41**	**3.90**	**87.13**	**96.10**	**91.39**	**77.41**	**2.50**	**99.09**	**97.50**	**98.29**	**77.96**	**2.10**	**96.36**	**97.90**	**97.12**
20%	*No Defense*	*63.56*	*99.35*	*0.00*	*0.00*	*0.00*	*64.06*	*98.00*	*0.00*	*0.00*	*0.00*	*63.76*	*100.00*	*0.00*	*0.00*	*0.00*
	CCCV	**74.83**	**4.70**	**90.29**	**95.35**	**92.75**	**76.25**	**3.45**	**98.52**	**96.55**	**97.53**	**76.65**	**2.75**	**98.48**	**97.25**	**97.86**
40%	*No Defense*	*47.47*	*99.85*	*0.00*	*0.00*	*0.00*	*48.61*	*99.50*	*0.00*	*0.00*	*0.00*	*47.61*	*100.00*	*0.00*	*0.00*	*0.00*
	CCCV	**73.96**	**3.98**	**99.30**	**96.03**	**97.64**	**74.80**	**2.55**	**99.85**	**97.45**	**98.63**	**74.04**	**2.55**	**99.41**	**97.45**	**98.42**
80%	*No Defense*	*17.27*	*99.36*	*0.00*	*0.00*	*0.00*	*15.92*	*99.22*	*0.00*	*0.00*	*0.00*	*16.01*	*100.00*	*0.00*	*0.00*	*0.00*
	CCCV	**71.24**	**3.43**	**99.95**	**96.57**	**98.23**	**67.97**	**2.64**	**99.71**	**97.34**	**98.51**	**69.17**	**2.54**	**99.90**	**97.46**	**98.66**

P.R. denotes the evaluation poisoning rate, and Prec denotes precision. ↑ and ↓ indicate that higher and lower values are better, respectively.

Influence of evaluation poisoning rate. We further evaluate the robustness of CCCV by varying the proportion of triggered test samples from 1% to 80% under BadVFL, VILLAIN, and Optimal Trigger, while keeping the training poisoning rate fixed. As shown in Table A4, CCCV consistently reduces the ASR to below 6% across all trigger rates and attacks, while maintaining high detection recall. More importantly, the results demonstrate the effectiveness of CCCV's post-detection prediction recovery under severe attack prevalence. At a test-time trigger rate of 80%, the accuracy without defense drops to 17.27%, 15.92%, and 16.01% under BadVFL, VILLAIN, and Optimal Trigger, respectively. In contrast, CCCV recovers the accuracy to 71.24%, 67.97%, and 69.17%, while reducing the corresponding ASR to 3.43%, 2.64%, and 2.54%. Detection recall also remains high, exceeding 96% for all three attacks at this trigger rate. These results indicate that, after detecting suspicious inputs and attributing the malicious clients, CCCV can effectively suppress their backdoor contribution and recover predictions from the remaining benign representations. Conse-

quently, CCCV remains effective even when triggered samples constitute a large fraction of the test set.

Table A5: Sensitivity analysis of η and Δ under three VFL backdoor attacks.

η	Δ	BadVFL (%)					Villain (%)					Optimal Trigger (%)				
		ACC↑	ASR↓	Prec↑	Recall↑	F1↑	ACC↑	ASR↓	Prec↑	Recall↑	F1↑	ACC↑	ASR↓	Prec↑	Recall↑	F1↑
No Defense		*77.84*	*93.00*	*0.00*	*0.00*	*0.00*	*78.79*	*98.00*	*0.00*	*0.00*	*0.00*	*77.81*	*100.00*	*0.00*	*0.00*	*0.00*
85%	0	73.75	**2.00**	11.12	**98.00**	19.98	77.61	**1.00**	24.32	**99.00**	39.05	76.27	**6.00**	22.43	**94.00**	36.22
	50	77.63	**2.00**	47.57	**98.00**	64.05	79.10	**1.00**	70.71	**99.00**	82.50	77.86	**6.00**	56.29	**94.00**	70.41
	100	78.31	**2.00**	91.59	**98.00**	94.69	79.33	**1.00**	92.52	**99.00**	95.65	78.32	**6.00**	88.68	**94.00**	91.26
90%	0	75.61	**2.00**	17.75	**98.00**	30.06	78.36	**1.00**	36.53	**99.00**	53.37	77.07	**6.00**	32.08	**94.00**	47.84
	50	78.01	**2.00**	62.42	**98.00**	76.26	79.20	**1.00**	79.84	**99.00**	88.39	78.12	**6.00**	71.21	**94.00**	81.03
	100	**78.34**	**2.00**	94.23	**98.00**	**96.08**	**79.34**	**1.00**	93.40	**99.00**	**96.12**	**78.36**	**6.00**	92.16	**94.00**	**93.07**
95%	0	77.18	14.00	31.39	86.00	45.99	78.93	3.00	59.51	97.00	73.76	77.85	9.00	50.28	91.00	64.77
	50	78.02	36.00	75.29	64.00	69.19	79.27	6.00	91.26	94.00	92.61	78.26	13.00	85.29	87.00	86.14
	100	78.10	49.00	**100.00**	51.00	67.55	79.26	21.00	**95.18**	79.00	86.34	78.27	23.00	**96.25**	77.00	85.56

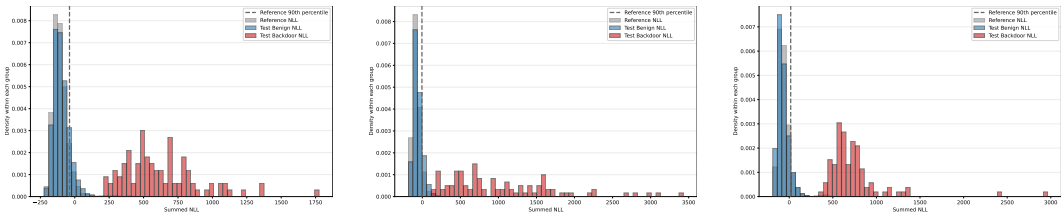
Prec denotes precision. The best results are highlighted in **red**. ↑ and ↓ indicate that higher and lower values are better, respectively.

Influence of threshold η and Δ. We further investigate the sensitivity of CCCV to the percentile threshold η and anomaly-margin threshold Δ on CIFAR-10 with a poisoning rate of 1%. As shown in Table A5, both parameters mainly control the trade-off between detection precision and recall. When η is relatively small (85% or 90%), increasing Δ substantially improves precision and clean accuracy while largely preserving detection recall and ASR. For example, with $\eta = 90\%$, increasing Δ from 0 to 100 improves the detection precision from 17.75% to 94.23%, 36.53% to 93.40%, and 32.08% to 92.16% under BadVFL, VILLAIN, and Optimal Trigger, respectively, while the corresponding recalls remain unchanged at 98%, 99%, and 94%. In contrast, an overly large percentile threshold, e.g., $\eta = 95\%$, makes the detector more conservative: further increasing Δ noticeably reduces recall and leads to higher ASR, particularly under BadVFL. Overall, $\eta = 90\%$ and $\Delta = 100$ provide a favorable balance across all three attacks, achieving F1 scores of 96.08%, 96.12%, and 93.07%, respectively, while maintaining clean accuracy comparable to the undefended model. These results indicate that CCCV is stable within a moderate operating region, while overly aggressive percentile thresholds can substantially reduce recall. Moreover, this thresholding mechanism is not specific to CIFAR-10 and can be easily applied to different datasets by adapting Δ to their data set-specific anomaly-score scales while keeping the percentile-based criterion η unchanged. We further provide practical guidelines for selecting Δ across different datasets in Appendix B.1.

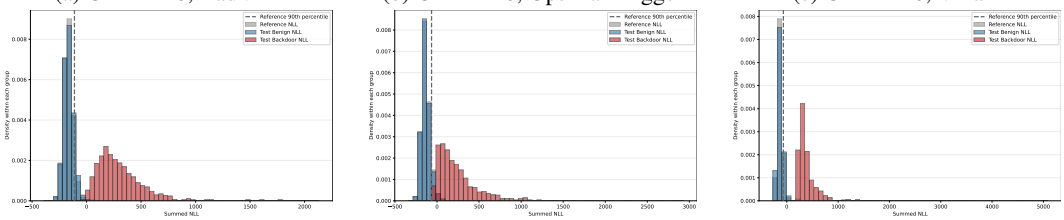
APPENDIX B DISCUSSION

APPENDIX B.1 CONDITIONAL DENSITY DISTRIBUTION VISUALIZATION

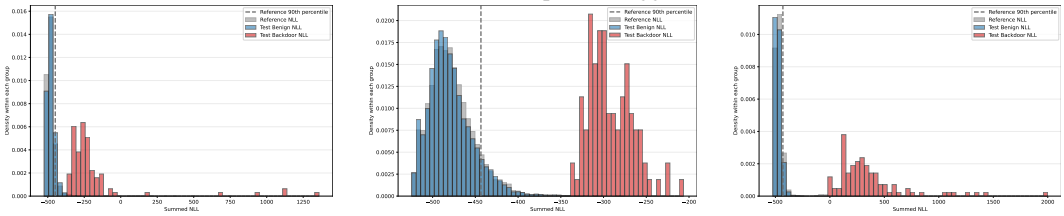
Figure B1 visualizes the conditional inconsistency score Γ for benign and backdoor examples during inference. Across different datasets and attack strategies, benign examples are generally concentrated within a relatively compact range, whereas backdoor examples tend to shift toward substantially larger Γ values. More importantly, this separation remains observable even when the benign distribution contains atypical or low-density samples, suggesting that our method is not merely detecting marginally rare representations. Instead, the conditional density captures whether each client representation remains compatible with the representations contributed by the other clients. For benign examples, although an individual sub-embedding may occasionally be unusual, the cross-client representations still describe a coherent sample and therefore retain relatively high conditional compatibility. In contrast, for backdoor examples, the trigger-induced representation is combined with a non-target context from the remaining clients during inference, producing a pronounced cross-client mismatch and consequently a much larger conditional inconsistency score. The visualization therefore provides direct empirical evidence that the proposed conditional-density verification can precisely separate backdoor examples from benign examples at inference time, including benign outliers that may otherwise be easily confused with poisoned samples by conventional anomaly-based detectors.



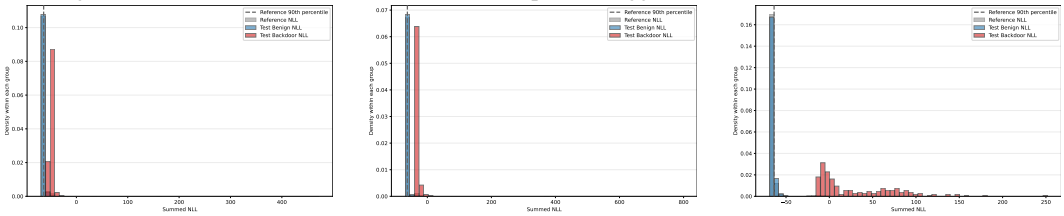
(a) CIFAR-10, BadVFL (b) CIFAR-10, Optimal Trigger (c) CIFAR-10, Villain



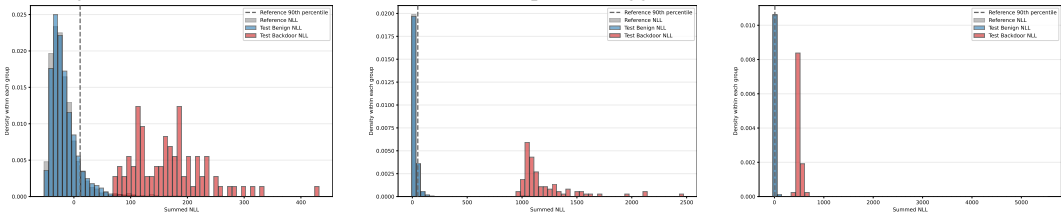
(d) CINIC-10, BadVFL (e) CINIC-10, Optimal Trigger (f) CINIC-10, Villain



(g) MNIST, BadVFL (h) MNIST, Optimal Trigger (i) MNIST, Villain



(j) Vehicle, BadVFL (k) Vehicle, Optimal Trigger (l) Vehicle, Villain



(m) Bank, BadVFL (n) Bank, Optimal Trigger (o) Bank, Villain

Figure B1: Distributions of Γ across different datasets and VFL backdoor attacks. Blue denotes test benign examples, red denotes test backdoor examples, and gray denotes reference examples.

For referenced NLL selection, we use the 90th percentile of the recorded conditional-density distribution as γ. We empirically find that this percentile provides a relatively stable operating point, preserving most benign variations while excluding the extreme tail of the reference distribution. Since the scale and dispersion of Γ can vary substantially across datasets and attack strategies, directly using a universal value of Δ is less practical. Therefore, we adopt a simple data-adaptive strategy and determine Δ from the reference distribution itself. Specifically, Δ is determined solely from training-time reference statistics. After IQR-based outlier removal, we average the empirical quantile function over $q \in [0.985, 0.995]$, and aggregate the resulting client-wise estimates to obtain a robust threshold. This provides a practical estimate of the largest benign deviation beyond the reference threshold and allows the detection criterion to naturally adapt to different datasets and attack settings. All thresholds are determined exclusively from training-time reference statistics without accessing test benign or backdoor examples.

APPENDIX B.2 ADAPTIVE ATTACK ANALYSIS

We further consider a defense-aware attacker that seeks to strengthen the target-class prediction while evading CCCV. Our primary goal is to explain why a successful attack may simultaneously increase the anomaly score Γ.

Let k_a denote the adversarial client and $\mathbf{z}_s^{-(k_a)}$ denote the fixed representations contributed by the remaining benign clients for a non-target sample s, where $y_s \neq y_t$. The attacker manipulates its representation $\mathbf{z}_t^{(k_a)}(\alpha)$ with attack strength α. We define

$$q(\alpha) = P\left(y_t \mid \mathbf{z}_t^{(k_a)}(\alpha), \mathbf{z}_s^{-(k_a)}\right), \tag{17}$$

and

$$r(\alpha) = p\left(\mathbf{z}_t^{(k_a)}(\alpha) \mid \mathbf{z}_s^{-(k_a)}, y_t\right). \tag{18}$$

The corresponding conditional anomaly score is

$$\Gamma(\alpha) = -\log p\left(\mathbf{z}_t^{(k_a)}(\alpha) \mid \mathbf{z}_s^{-(k_a)}\right). \tag{19}$$

By Bayes' rule,

$$q(\alpha) = \frac{r(\alpha) P(y_t \mid \mathbf{z}_s^{-(k_a)})}{p\left(\mathbf{z}_t^{(k_a)}(\alpha) \mid \mathbf{z}_s^{-(k_a)}\right)}, \tag{20}$$

which gives

$$\Gamma(\alpha) = \log q(\alpha) - \log r(\alpha) - \log P(y_t \mid \mathbf{z}_s^{-(k_a)}). \tag{21}$$

Since $\mathbf{z}_s^{-(k_a)}$ is fixed with respect to α, for $\alpha_2 > \alpha_1$ we have

$$\Gamma(\alpha_2) - \Gamma(\alpha_1) = \log \frac{q(\alpha_2)}{q(\alpha_1)} - \log \frac{r(\alpha_2)}{r(\alpha_1)}. \tag{22}$$

Therefore,

$$\Gamma(\alpha_2) > \Gamma(\alpha_1) \iff \frac{q(\alpha_2)}{q(\alpha_1)} > \frac{r(\alpha_2)}{r(\alpha_1)}, \tag{23}$$

means that the anomaly score increases whenever the attack strengthens the target prediction $q(\alpha)$ faster than it improves the compatibility $r(\alpha)$ of the manipulated representation with the surrounding non-target cross-client context and target class.

Because $r(\alpha)$ conditions on both the benign non-target context $\mathbf{z}_s^{-(k_a)}$ and the target label y_t, which are semantically conflicting, we assume that $r(\alpha)$ is relatively insensitive to changes in the attack strength α. In particular, for $\alpha_2 > \alpha_1$, we assume

$$\log \frac{r(\alpha_2)}{r(\alpha_1)} < \log \frac{q(\alpha_2)}{q(\alpha_1)}. \tag{24}$$

Then,

$$\Gamma(\alpha_2) - \Gamma(\alpha_1) = \log \frac{q(\alpha_2)}{q(\alpha_1)} - \log \frac{r(\alpha_2)}{r(\alpha_1)} > 0. \tag{25}$$

Therefore, increasing the attack strength tends to increase the conditional anomaly score.

This analysis provides a sufficient condition under which strengthening the attack also increases the conditional anomaly score.