CONCAM: ANCHOR CONSENSUS AND CAMOUFLAGED INJECTION FOR BACKDOOR ATTACKS IN VERTICAL FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Vertical federated learning (VFL) enables parties holding disjoint feature partitions to collaboratively train models without sharing raw data. However, the embedding gradients exchanged during training can be exploited by a malicious passive party to implant backdoors. Recent VFL attacks can infer target-related samples from gradient information, yet single-anchor target-set construction remains sensitive to anchor quality, while explicit triggers and relatively large poisoning budgets may introduce conspicuous attack patterns. In this paper, we propose ConCam, a targeted backdoor attack that combines anchor consensus for reliable target-set construction with camouflaged backdoor injection. Specifically, ConCam aggregates candidate target sets from multiple target-class anchors through consensus voting and employs sample-specific implicit triggers together with dynamic gradient-based sample and carrier selection to enable effective injection at low poisoning rates. Across four datasets, ConCam achieves attack success rates above 97% with at most a 1.05-percentage-point reduction in accuracy, remains effective at a poisoning rate as low as 0.06%, and achieves substantially higher attack success rates than prior attacks under dedicated VFL backdoor defenses.

1 INTRODUCTION

Vertical Federated Learning (VFL) enables multiple parties to collaboratively train a model while keeping their feature partitions local, making it attractive for cross-organization learning under strict data-sharing constraints (Yang et al., 2019; Kairouz et al., 2021; Liu et al., 2022). However, the intermediate information exchanged during training also creates an attack surface. A malicious passive party can manipulate its local data and model while exploiting feedback from the active party, thereby implanting targeted backdoor behavior without access to complete inputs, the remaining task labels, or the top model.

Unlike centralized learning, the semantic information required to identify target-class samples is controlled by the active party. Existing studies have explored ways to address this limitation, including auxiliary label information, local latent representations, and category-related information inferred from training gradients (Bai et al., 2023; Gu & Bai, 2023; Naseri et al., 2023; Chen et al., 2023; Shen et al., 2025). These advances reduce the need for explicit supervision, but shift the question from whether target information can be obtained to how reliably it can guide targeted poisoning. Meanwhile, effective backdoor injection requires enough poisoned samples to establish the desired behavior, whereas increasing the number or magnitude of modifications enlarges the footprint in benign training. A practical targeted VFL attack must therefore select poisoning samples reliably while injecting sufficient backdoor signals with as few unnecessary modifications as possible.

Motivated by these observations, we propose **ConCam**, a targeted backdoor attack that jointly considers target-set reliability and poisoning footprint in VFL. ConCam exploits consistency across multiple target-class anchors to obtain more reliable target-set evidence, and introduces a sample-specific injection strategy together with dynamic poisoning selection to reduce unnecessary poisoning. This design concentrates the backdoor learning signal on a small set of responsive samples while limiting modifications to benign training. Experiments on four datasets show that ConCam achieves high attack effectiveness with no more than a 1.05-percentage-point reduction in benign ac-

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.

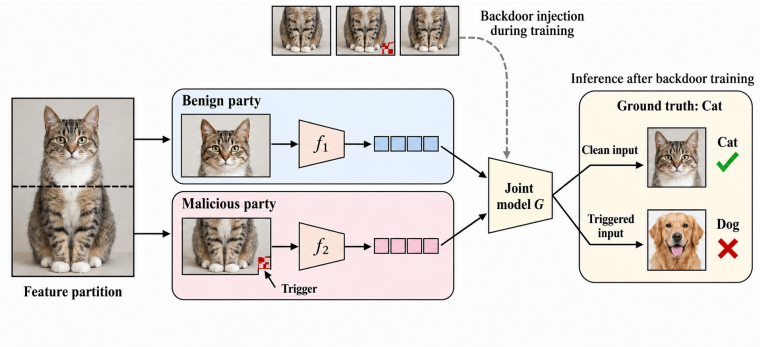

Figure 1: Illustration of a targeted backdoor attack in VFL. Clean inputs retain their original predictions, whereas triggered inputs are redirected to the target class.

curacy. On CIFAR-10, ConCam achieves 95.99% attack success at a poisoning rate of only 0.06%. We further analyze the effects of anchor number, target-set consistency, trigger distortion, poisoning budget, and representative defenses to characterize the effectiveness and limitations of the attack.

Our contributions are summarized as follows:

- We introduce a **reliable and low-footprint perspective for targeted backdoor attacks in VFL**, highlighting the need to jointly consider target-set reliability and the amount of poisoning required under restricted attacker knowledge.
- We propose **ConCam**, which realizes this perspective through two complementary mechanisms: **Anchor Consensus** aggregates target evidence across multiple target-class anchors to reduce dependence on any individual reference, while **Camouflaged Injection** combines **sample-specific implicit triggers** with dynamic poisoning selection to concentrate backdoor learning on a limited set of samples.
- We conduct systematic experiments across datasets, anchor configurations, poisoning budgets, trigger quality, and defenses. The results demonstrate the effectiveness of the proposed design and show that ConCam achieves 95.99% attack success with a poisoning rate of only 0.06% on CIFAR-10.

2 Related Work

Backdoor attacks in VFL differ from those in centralized and horizontal federated learning because a malicious passive party typically lacks access to task labels and complete training samples. Existing attacks therefore explore different sources of target-related information. **Label- or target-information-dependent attacks** obtain such information through label inference or limited auxiliary target knowledge. VILLAIN combines label inference with data poisoning to identify target-class samples (Bai et al., 2023), while LR-BA exploits local latent representations and auxiliary labeled data to construct malicious representations toward the target class (Gu & Bai, 2023). TECB further considers a limited-information setting with only a small number of target-class samples (Chen et al., 2023). **Training-feedback-driven attacks** reduce this dependence by exploiting information revealed during VFL training. BadVFL extracts category-related information from embedding gradients for clean-label backdoor injection (Naseri et al., 2023), while LFBA further constructs target poison sets from label-related embedding gradients without additional task-label information (Shen et al., 2025). This progression has gradually reduced the reliance on explicit target supervision, but also raises a further question: how can target-related information be used more reliably for targeted poisoning while keeping the poisoning footprint small? ConCam addresses this problem through anchor consensus and camouflaged injection.

3 PRELIMINARIES

3.1 VERTICAL FEDERATED LEARNING

VFL enables multiple parties that share aligned sample identities but hold disjoint feature subsets to collaboratively train a model without exchanging raw data (Yang et al., 2019; Liu et al., 2022). The party holding the task labels and top model is referred to as the active party, while the remaining parties are passive parties.

Consider K participants $\{P_k\}_{k=1}^{K}$ and N aligned samples. For sample i, participant P_k holds a local feature x_i^k and computes a local embedding $h_i^k = f_k(x_i^k; \theta_k)$, where f_k and θ_k denote the bottom model and its parameters. The active party aggregates the local embeddings and produces the prediction $\hat{y}_i = G(h_i^1, h_i^2, \ldots, h_i^K; \theta_G)$, where G and θ_G denote the top model and its parameters. The VFL model is trained by minimizing

$$\min_{\Theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(G(h_i^1, h_i^2, \ldots, h_i^K; \theta_G), y_i\right), \tag{1}$$

where $\Theta = \{\theta_1, \ldots, \theta_K, \theta_G\}$ and $\mathcal{L}$ is the task loss. During training, the active party returns the corresponding embedding gradients to the passive parties for updating their bottom models. These gradients provide the main training feedback available to a passive party.

3.2 BACKDOOR ATTACKS IN VERTICAL FEDERATED LEARNING

A targeted backdoor attack aims to make a trained model predict an attacker-specified target class y_t for inputs containing a trigger, while maintaining normal predictions on clean inputs (Gu et al., 2017). In VFL, a malicious passive party can inject a sample-specific implicit trigger into its local features. Given a local feature x_i^m and a fixed hidden message z, a pretrained trigger generator $\mathcal{T}$ produces the triggered feature $\widetilde{x}_i^m = \mathcal{T}(x_i^m; z)$, while the features held by the other parties remain unchanged. The resulting triggered input is $\widetilde{x}_i = [x_i^1, \ldots, \widetilde{x}_i^m, \ldots, x_i^K]$. The attack seeks to induce $F(\widetilde{x}_i; \Theta) \rightarrow y_t$, while preserving the model's performance on clean inputs, where F denotes the resulting end-to-end VFL model.

3.3 THREAT MODEL

We consider a VFL system with one malicious passive party, while the active party and the remaining participants follow the standard training protocol.

Attacker capability. Let P^m denote the malicious passive party. The attacker can access and manipulate its local features and bottom-model parameters, and observe the per-sample embedding gradients returned during standard VFL training. The attacker knows a small set of M aligned training samples, $\mathcal{A} = \{a_1, a_2, \ldots, a_M\}$, all belonging to the target class y_t. These samples serve as target anchors. The attacker cannot access or modify the labels of other training samples, access other parties' raw features, or access or control the top model. Since the anchors are part of the aligned training set, their embedding gradients are obtained through the standard VFL training process without introducing external samples or additional gradient queries.

Attacker goal. The attacker aims to implant a targeted backdoor such that triggered inputs are classified as y_t, while preserving the model's normal performance on clean inputs.

4 CONCAM: ANCHOR CONSENSUS AND CAMOUFLAGED INJECTION

ConCam consists of three stages: target set construction, sample-specific camouflaged injection, and dynamic key sample selection. After a benign warm-up epoch, the malicious passive party uses the returned embedding gradients and the target-class anchors defined in Section 3 to construct a target set through multi-anchor consensus voting. A pretrained encoder then generates input-dependent implicit triggers for benign carrier samples. During subsequent training epochs, ConCam dynamically selects a subset of target samples and non-target carrier samples according to their gradient norms, and replaces the local features of the selected target samples with triggered carrier features. The overall procedure is illustrated in Figure 2.

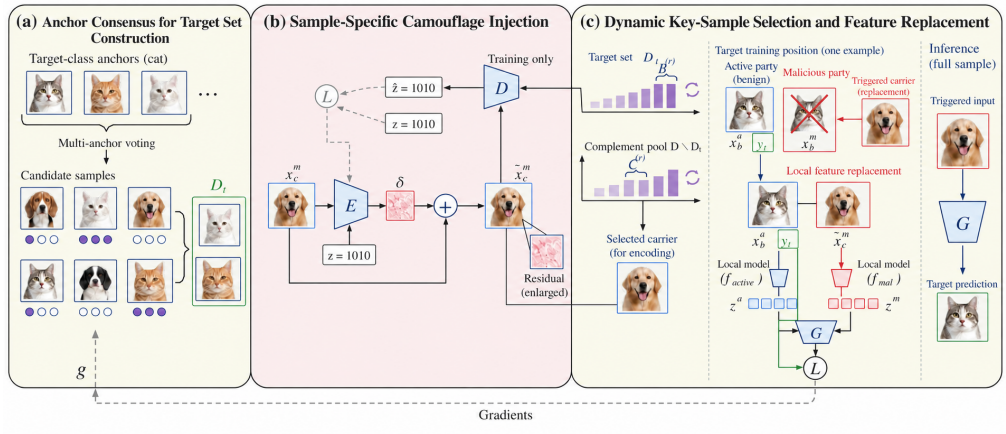

Figure 2: Overview of ConCam.

4.1 Anchor Consensus for Target Set Construction

During the benign warm-up epoch, the malicious party collects the embedding gradient $g_i^m = \frac{\partial \mathcal{L}_i}{\partial h_i^m}$ for each training sample $i \in \mathcal{D}$. These gradients contain label-related optimization signals and can therefore be used to identify samples with similar target-related gradient patterns. Let $|\mathcal{D}| = N$ and ρ denote the target-set ratio. ConCam aims to construct a target set $\mathcal{D}_t$ with $n_t = \lfloor \rho N \rfloor$ samples. For each target anchor $a_j \in \mathcal{A}$, ConCam computes the cosine similarity between the gradient of candidate sample i and that of the anchor:

$$s_j(i) = \frac{\langle g_i^m, g_{a_j}^m \rangle}{\|g_i^m\|_2 \|g_{a_j}^m\|_2}. \tag{2}$$

The n_t samples with the largest similarity scores form the single-anchor candidate set $\mathcal{S}_j = \mathrm{TopIdx}_{n_t}\{s_j(i)\}_{i\in\mathcal{D}}$, where TopIdx_{n_t} returns the sample identifiers corresponding to the n_t largest input values. A single anchor may introduce reference-specific variation into the selected set. ConCam therefore aggregates the evidence from multiple target anchors through consensus voting. For each sample i, its vote count is defined as

$$v(i) = \sum_{j=1}^{M} \mathbf{1}\left(i \in \mathcal{S}_j\right), \tag{3}$$

where $v(i)$ indicates the number of anchors that select sample i. Samples receiving votes from more anchors are prioritized. Specifically, ConCam first considers samples selected by all M anchors and then samples selected by $M-1, \ldots, 2$ anchors. Within the same vote group, samples are ranked by their average rank over the anchors that select them. Samples receiving only one vote are excluded from the consensus-based selection.

To provide additional candidates when the consensus-based samples are insufficient to reach n_t, ConCam constructs a mean target prototype $\bar{g}^m = \frac{1}{M}\sum_{j=1}^{M} g_{a_j}^m$ and computes the prototype similarity $s_p(i) = \langle g_i^m, \bar{g}^m \rangle / (\|g_i^m\|_2 \|\bar{g}^m\|_2)$. The prototype candidate set is defined as $\mathcal{S}_{\text{proto}} = \mathrm{TopIdx}_{n_t}\{s_p(i)\}_{i\in\mathcal{D}}$. After the consensus-based candidates are selected, ConCam removes samples already included in $\mathcal{D}_t$ from $\mathcal{S}_{\text{proto}}$ and adds the remaining candidates in descending order of $s_p(i)$ until $|\mathcal{D}_t| = n_t$. The resulting $\mathcal{D}_t$ is used as the target sample set for subsequent backdoor injection. The complete procedure is summarized in Algorithm 2.

4.2 Sample-Specific Camouflaged Injection

Inspired by DNN-based image steganography, we train an implicit-message encoder using an encoder–decoder architecture adapted from prior steganographic methods (Baluja, 2017; Tancik et al., 2020; Li et al., 2021). By jointly optimizing message recovery and image-quality constraints, the network learns to generate input-dependent implicit perturbations.

The hidden message is represented by a binary vector z associated with the target class. During the attack, the attacker specifies a fixed message for the target class and uses the same message across different triggered samples. Given a clean local image $x_i^m \in [0,1]$ and a message z, the encoder E generates an input-dependent residual $\delta_i^m = E(x_i^m, z)$ with the same dimensions as x_i^m. The encoded image is obtained by adding the residual to the input and clipping pixel values to the valid range: $\widetilde{x}_i^m = \mathrm{clip}(x_i^m + \delta_i^m, 0, 1)$.

The decoder D takes the encoded image as input and recovers the hidden message, $\widehat{z} = D(\widetilde{x}_i^m)$. At each training iteration, the encoder first produces the encoded image, after which the decoder reconstructs the message. We then compute the message-recovery error and image distortion and update the network parameters through backpropagation. This procedure jointly imposes message-reconstruction and image-quality constraints without requiring access to task labels or the top classification model.

The encoder–decoder training objective is defined as

$$\mathcal{L}_{\mathrm{enc}} = \mathcal{L}_{\mathrm{msg}}(\widehat{z}, z) + \alpha \mathcal{L}_r(x_i^m, \widetilde{x}_i^m) + \beta \mathcal{L}_p(x_i^m, \widetilde{x}_i^m) + \gamma \mathcal{L}_c(x_i^m, \widetilde{x}_i^m), \tag{4}$$

where $\mathcal{L}_{\mathrm{msg}}$ measures the message-recovery error; $\mathcal{L}_r$ constrains the magnitude of pixel-level perturbations; $\mathcal{L}_p$ constrains perceptual differences; and $\mathcal{L}_c$ encourages the encoded image to appear visually natural. The coefficients α, β, and γ control the weights of the corresponding constraints.

4.3 Dynamic Key Sample Selection and Feature Replacement

After constructing the target set $\mathcal{D}_t$, ConCam dynamically selects a small subset of target samples for poisoning in each training epoch. Directly applying triggers to target samples may allow the model to rely on their original target-related features rather than learning the trigger–target association. ConCam therefore uses non-target samples as trigger carriers and replaces the local features of selected target samples with triggered carrier features.

At training epoch r, ConCam uses the embedding gradient $g_i^{m,(r-1)}$ returned after epoch $r-1$ for sample selection. Its ℓ_2 norm is used as the sample importance score:

$$q_i^{(r)} = \left\| g_i^{m,(r-1)} \right\|_2 . \tag{5}$$

Given a dynamic selection ratio η, the number of active backdoor samples is $n_b = \lfloor \eta |\mathcal{D}_t| \rfloor$. The active backdoor set is selected as

$$\mathcal{B}^{(r)} = \mathrm{TopIdx}_{n_b}\{q_i^{(r)}\}_{i \in \mathcal{D}_t}. \tag{6}$$

Thus, only a fraction η of the constructed target set is actively poisoned in each epoch. The effective per-epoch poisoning rate is defined as $\rho_p = |\mathcal{B}^{(r)}|/N = n_b/N$. Since $n_t = \lfloor \rho N \rfloor$ and $n_b = \lfloor \eta n_t \rfloor$, it follows that $\rho_p = \lfloor \eta \lfloor \rho N \rfloor \rfloor / N \approx \eta\rho$, where the approximation ignores integer rounding.

The trigger carriers are selected from the non-target set $\mathcal{U}^{(r)} = \mathcal{D} \setminus \mathcal{D}_t$. The samples in $\mathcal{U}^{(r)}$ are sorted in ascending order of $q_i^{(r)}$. Given a carrier starting quantile τ, ConCam sets $o = \lfloor \tau |\mathcal{U}^{(r)}| \rfloor$ and selects n_b consecutive samples beginning at position o to form the carrier set $\mathcal{C}^{(r)}$. This strategy avoids concentrating carriers exclusively in either the lowest- or highest-gradient regions. The default setting is $\tau = 0.4$.

Finally, ConCam establishes a one-to-one mapping $\pi_r : \mathcal{B}^{(r)} \to \mathcal{C}^{(r)}$. For each paired target sample $b \in \mathcal{B}^{(r)}$ and carrier $\pi_r(b)$, the malicious party replaces the local feature of the target sample with the triggered feature generated from the carrier:

$$x_b^m \leftarrow \widetilde{x}_{\pi_r(b)}^m = \mathcal{T}\left(x_{\pi_r(b)}^m; z\right). \tag{7}$$

Here, $\mathcal{T}(\cdot; z)$ denotes the sample-specific trigger generation function defined in Section 3, which takes the local feature of a carrier sample and the hidden message z as inputs to generate its triggered version. The sample identity and label held by the active party remain unchanged. Consequently, the triggered local feature is trained under the original target-class label, encouraging the model to associate the implicit trigger with y_t. After the forward and backward passes, the returned embedding gradients are used for sample selection in the subsequent epoch.

5 EXPERIMENTS

Datasets and VFL Setting. We evaluate ConCam on four image classification datasets, including CIFAR-10, CIFAR-100, CINIC-10, and EuroSAT (Krizhevsky, 2009; Darlow et al., 2018; Helber et al., 2019). CIFAR-10 and CIFAR-100 contain 50,000 training and 10,000 test samples, CINIC-10 contains 90,000 training and 90,000 test samples, and EuroSAT contains 21,600 training and 5,400 test samples. We adopt a two-party VFL setting, where each image is vertically partitioned into left and right halves and the malicious party holds the right half. CIFAR-10 is used as the primary dataset for detailed analysis of poisoning rate, trigger properties, and defense performance.

Baselines. We compare ConCam with BadVFL, TECB, and LFBA, and include clean VFL training as the benign baseline. To examine the contribution of selective poisoning, we additionally consider RTS, which denotes random target-set selection without gradient-guided selection or carrier replacement.

Defense Methods. We evaluate the attacks under four defense methods: LapNoise, MultiStep-Grad, VFLIP, and FILTER. LapNoise adds Laplace noise to transmitted gradients (Dwork et al., 2006), while MultiStep-Grad applies gradient discretization and clipping. VFLIP and FILTER are VFL-specific defense methods (Cho et al., 2024; Hu et al., 2026).

Evaluation Metrics. We use attack success rate (ASR) and benign accuracy (BA) to evaluate attack effectiveness. ASR is the proportion of triggered test samples classified as the target class, excluding samples whose ground-truth labels already belong to that class. BA measures classification accuracy on clean test samples. We additionally report $\mathrm{Avg.} = (\mathrm{BA} + \mathrm{ASR})/2$ in the attack-effectiveness and poisoning-rate experiments to jointly characterize attack effectiveness and benign performance.

Target consistency rate (TCR) measures the proportion of samples in the constructed target set whose ground-truth labels match the target class. We evaluate the perceptual footprint of the generated triggers using PSNR, SSIM, LPIPS, and ℓ_∞. Higher PSNR and SSIM, and lower LPIPS and ℓ_∞, indicate greater similarity between triggered and clean local inputs. The first three metrics use inputs scaled to $[0, 1]$, while ℓ_∞ values in Table 2 are reported on the $[0, 255]$ pixel scale.

Implementation Details. Each party uses a ResNet-18 (He et al., 2016), initialized without pretrained weights, to produce a 128-dimensional local embedding. The two embeddings are concatenated and processed by a top model with linear-layer dimensions $256 \to 256 \to 128 \to C$, where C is the number of classes. We use Adam with a learning rate of 10^{-3}, weight decay of 10^{-4}, and batch size of 256. Models are trained for up to 100 epochs, with the first epoch used for benign warm-up and gradient collection.

Unless otherwise specified, ConCam uses three randomly sampled target-class anchors, a target-set ratio of $\rho = 0.1$, a dynamic selection ratio of $\eta = 0.5$, corresponding to an effective poisoning rate of $\rho_p = 0.05$, and a carrier starting quantile of $\tau = 0.4$. The pretrained trigger encoder is trained before VFL training and remains frozen throughout the subsequent VFL training process.

5.1 MAIN RESULTS

Overall Attack Performance. Table 1 summarizes the overall attack performance on the four datasets. ConCam achieves ASRs of 99.98%, 99.87%, 97.13%, and 100.00% on CIFAR-10, CINIC-10, CIFAR-100, and EuroSAT, respectively. The corresponding BA decreases relative to clean VFL training by only 0.33, 1.05, 0.86, and 0.79 percentage points.

These results show that ConCam maintains strong targeted attack effectiveness across datasets while causing only a small impact on benign classification performance. In particular, the ASR remains above 97% on all four datasets, and the maximum BA degradation is only 1.05 percentage points. ConCam achieves the highest ASR on CIFAR-10, CIFAR-100, and EuroSAT among the evaluated attacks, while TECB obtains a marginally higher ASR on CINIC-10. The corresponding Avg. results show the same overall trend, with ConCam achieving the highest value on three of the four datasets.

The consistent performance across datasets suggests that the combination of reliable target-set construction and low-footprint poisoning is not limited to a specific image classification setting.

Camouflage of Triggered Samples. We next evaluate the observable footprint of the generated triggers. Table 2 randomly selects 1,000 CIFAR-10 samples and applies the triggers generated by

Table 1: Attack effectiveness and benign performance on different datasets.

Method	CIFAR-10			CINIC-10			CIFAR-100			EuroSAT		
	BA (%)	ASR (%)	Avg. Score (%)	BA (%)	ASR (%)	Avg. Score (%)	BA (%)	ASR (%)	Avg. Score (%)	BA (%)	ASR (%)	Avg. Score (%)
Clean VFL	79.24	–	–	**67.75**	–	–	**49.67**	–	–	**91.81**	–	–
ConCam	78.91	**99.98**	**89.445**	66.70	99.87	83.285	48.81	**97.13**	**72.970**	91.02	**100.00**	**95.510**
BadVFL	76.15	99.24	87.695	67.04	82.72	74.880	47.52	81.12	64.320	90.40	91.42	90.910
TECB	**79.25**	96.52	87.885	67.68	**99.99**	**83.835**	48.01	96.97	72.490	82.79	90.66	86.725
LFBA	77.90	99.54	88.720	67.05	90.03	78.540	47.92	89.92	68.920	89.59	86.58	88.085

the four attacks. ConCam achieves a PSNR of 40.8905 dB, an SSIM of 0.9931, an LPIPS of 0.0055, and an ℓ_∞ value of 14.6360 on the $[0, 255]$ pixel scale, obtaining the best results among the evaluated attacks on all four metrics.

These results indicate that ConCam introduces relatively small measured perturbations while preserving high similarity between triggered and clean local inputs. Combined with its sample-specific trigger generation, this low perceptual footprint reduces the observable changes introduced into individual poisoned samples. Importantly, the camouflage mechanism is evaluated here in terms of the measured perturbation footprint rather than robustness against a particular defense. Visual examples of the clean and triggered local inputs produced by the four attacks are provided in Appendix 4.

Table 2: Comparison of trigger camouflage across different attack methods.

Attack	No. of Samples	Mean PSNR (dB)	Mean SSIM	Mean ℓ_∞ (0–255)	Mean LPIPS
LFBA	1000	22.7891	0.9815	203.4710	0.0487
ConCam	1000	**40.8905**	**0.9931**	**14.6360**	**0.0055**
TECB	1000	8.5508	0.1569	237.2560	0.6092
BadVFL	1000	21.1011	0.8102	190.1630	0.1455

Performance under VFL Defenses. Table 3 compares the performance of four attacks under different defense configurations on CIFAR-10. Under VFLIP, ConCam retains an ASR of 85.48%, whereas the ASRs of BadVFL, TECB, and LFBA decrease to 28.22%, 0.04%, and 1.04%, respectively. Under FILTER, ConCam achieves an ASR of 82.43%, substantially exceeding the 1.46%, 11.13%, and 36.86% obtained by BadVFL, TECB, and LFBA.

In contrast, LapNoise and MultiStep-Grad reduce the ASR of ConCam to 2.53% and 2.20%, respectively, accompanied by BA values of 71.26% and 70.76%. These results indicate that ConCam remains effective under the evaluated VFL-specific defenses, while defenses that directly perturb or discretize gradient feedback can substantially suppress the attack. This also highlights the dependence of gradient-driven VFL attacks on the quality of the training feedback available to the attacker.

Table 3: Performance of different attack methods under VFL defenses.

Defense	ConCam		BadVFL		TECB		LFBA	
	BA (%)	ASR (%)	BA (%)	ASR (%)	BA (%)	ASR (%)	BA (%)	ASR (%)
Baseline	78.91	**99.98**	76.15	99.24	**79.25**	96.52	77.90	99.54
VFLIP	66.38	**85.48**	73.52	28.22	**80.04**	0.04	74.64	1.04
FILTER	74.69	**82.43**	74.42	1.46	72.30	11.13	**75.32**	36.86
LapNoise	71.26	2.53	70.84	**6.86**	**75.83**	3.22	73.39	3.01
MultiStep-Grad	70.76	2.20	**74.93**	**9.71**	73.21	**3.31**	72.54	**2.76**

5.2 Further Analysis

Target-Set Reliability. We first investigate target-set construction on CIFAR-10, which is used for the detailed analyses in this section; corresponding results on the other datasets are provided in the supplementary material due to space limitations. We randomly sample 1,000 triplets from the target class and use the first sample for single-anchor construction and all three samples for consensus voting. Figure 3 shows the TCR distributions, with target sets grouped into 1-percentage-point

TCR intervals. Three-anchor voting produces 875 target sets with TCR $\geq 90\%$, compared with 490 for single-anchor construction, and increases the minimum TCR from 32.10% to 65.88%. This indicates that multi-anchor consensus substantially reduces the dependence of target-set construction on a particular anchor.

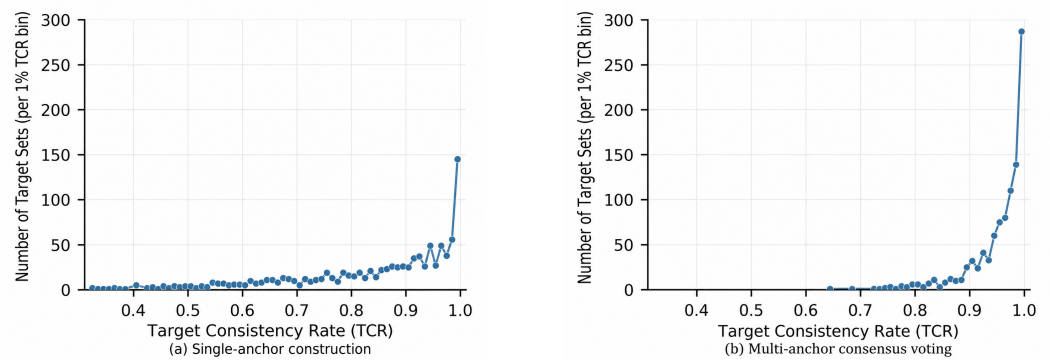

Figure 3: TCR distributions of 1,000 target sets constructed using (a) single-anchor construction and (b) three-anchor consensus voting.

Table 4 further shows that target-set reliability directly affects attack effectiveness. As TCR increases from 36.10% to 75.50% and 100%, ASR increases from 13.57% to 91.10% and 99.98%, respectively. Low-TCR target sets introduce inconsistent training signals and therefore weaken targeted poisoning, highlighting the importance of reliable target-set construction.

Table 4: Effect of target-set consistency (TCR) on ConCam performance.

TCR (%)	BA (%)	ASR (%)
100.00	78.91	99.98
90.86	78.26	98.21
75.50	77.56	91.10
55.82	77.64	70.66
36.10	78.87	13.57

We further examine the effect of the number of anchors in Table 5. As the number of anchors increases from three to six, the number of target sets with TCR $\geq 90\%$ increases from 875 to 952, while the minimum TCR increases from 65.88% to 78.80%. Nevertheless, three anchors already yield 989 target sets with TCR $\geq 75\%$. We therefore use three anchors as a practical trade-off between target-set reliability and the amount of required target-class anchor information.

Table 5: Effect of the number of anchors on target-set consistency over 1,000 trials.

No. of Anchors	No. of Sets (TCR $\geq 99\%$)	No. of Sets (TCR $\geq 90\%$)	No. of Sets (TCR $\geq 75\%$)	Min. TCR (%)
3	295	875	989	65.88
4	353	908	998	70.66
5	355	947	999	74.08
6	**419**	**952**	**1000**	**78.80**

Performance under Matched Low Poisoning Budgets. Table 6 compares ConCam with LFBA, BadVFL, and TECB under matched effective poisoning rates on CIFAR-10. ConCam achieves the highest ASR and Avg. score at $\rho_p = 0.05, 0.001$, and 0.0006. At $\rho_p = 0.01$, LFBA attains a slightly higher ASR (97.54% vs. 97.22%), whereas ConCam achieves the highest BA and Avg. score. At the lowest rate, $\rho_p = 0.0006$, ConCam retains an ASR of 95.99% and an Avg. score of 87.080%, exceeding the corresponding results of all baselines. Under this setting, only 30 of the 50,000 training samples are actively replaced in each epoch. Although the target set contains approximately

Table 6: Performance under matched effective poisoning rates on CIFAR-10.

Effective Poisoning Rate ρ_p	ConCam			LFBA			BadVFL			TECB		
	BA	ASR	Avg. Score	BA	ASR	Avg. Score	BA	ASR	Avg. Score	BA	ASR	Avg. Score
0.05	78.91	**99.98**	**89.445**	77.91	99.54	88.725	76.15	99.24	87.695	**79.25**	96.52	87.885
0.01	**79.39**	97.22	**88.305**	77.65	**97.54**	87.595	75.46	94.12	84.790	79.21	94.22	86.715
0.001	78.50	**95.96**	**87.230**	77.96	5.04	41.500	78.41	80.25	79.330	**79.31**	90.66	84.985
0.0006	78.17	**95.99**	**87.080**	79.10	0.02	39.560	78.34	65.29	71.815	**79.21**	87.54	83.375

5,000 samples, it serves as a candidate pool from which only a small subset is selected for active poisoning in each epoch.

Key-Sample and Carrier Selection. Table 7 evaluates the contribution of selective poisoning by comparing ConCam with RTS, which randomly selects target-set samples for trigger injection without gradient-guided selection or carrier replacement. ConCam improves ASR from 68.51% to 99.98%, BA from 72.51% to 78.91%, and Avg. score from 70.510% to 89.445%. These results suggest that effective use of the limited poisoning budget depends not only on the size of the target set, but also on selecting samples that provide more effective training signals.

Table 7: Ablation of key-sample selection and carrier replacement on CIFAR-10. RTS denotes random target selection.

Method	BA (%)	ASR (%)	Avg. Score (%)
ConCam	**78.91**	**99.98**	**89.445**
RTS	72.51	68.51	70.510

Table 8 examines the carrier starting quantile τ. Increasing τ from 0 to 0.4 improves ASR from 97.44% to 99.98% with almost unchanged BA (78.97% vs. 78.91%). Further increasing τ to 0.6 and 0.9 yields negligible ASR gains but reduces BA to 77.22% and 75.87%, respectively. We therefore use $\tau = 0.4$, which achieves the highest Avg. score among the evaluated settings.

Table 8: Effect of the carrier starting quantile τ on ConCam performance on CIFAR-10.

Carrier Starting Quantile τ	BA (%)	ASR (%)	Avg. Score (%)
0	**78.97**	97.44	88.205
0.4	78.91	99.98	**89.445**
0.6	77.22	**99.99**	88.605
0.9	75.87	**99.99**	87.930

6 Conclusion

This paper presents ConCam, a targeted backdoor attack against VFL that combines anchor consensus with camouflaged injection. By aggregating target-related evidence from multiple anchors and using sample-specific implicit triggers with selective poisoning, ConCam achieves effective targeted backdoor injection with a small poisoning footprint. Extensive experiments demonstrate strong attack effectiveness, reliable target-set construction, and effective poisoning under extremely low budgets. These results highlight the security risks of low-footprint backdoor attacks in VFL and the need for defenses that can better protect gradient-based training feedback without substantially degrading benign performance.

AI use statement

Generative AI tools were used to assist with translation, language editing, manuscript organization, and feedback on methodological descriptions and experimental reporting. The authors reviewed the AI-assisted revisions and take responsibility for the final manuscript.

Ethics statement

This work studies backdoor attacks in VFL to reveal potential security risks and motivate stronger defenses. The proposed attack could be misused if applied to deployed systems without authorization. The intended use of this work is to support security evaluation and defense development for privacy-preserving collaborative learning systems.

Reproducibility statement

Section 5 describes the datasets, feature partitioning, model architecture, and principal training and attack settings. Section 4 specifies the attack procedure, with detailed algorithms provided in Appendix D.

References

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*, pp. 2743–2760, 2023.

Shumeet Baluja. Hiding images in plain sight: Deep steganography. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Peng Chen, Jirui Yang, Junxiong Lin, Zhihui Lu, Qiang Duan, and Hongfeng Chai. A practical clean-label backdoor attack with limited information in vertical federated learning. In *IEEE International Conference on Data Mining*, pp. 41–50, 2023.

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. *arXiv preprint arXiv:2408.15591*, 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.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pp. 265–284, 2006.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. In *Machine Learning and Computer Security Workshop at NeurIPS*, 2017.

Yuhao Gu and Yuebin Bai. LR-BA: Backdoor attack against vertical federated learning using local latent representations. *Computers & Security*, 129:103193, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

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.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021.

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

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16463–16472, 2021.

Yang Liu, Yan Kang, Tianyuan Zou, et al. Vertical federated learning: Concepts, advances and challenges. *arXiv preprint arXiv:2211.12814*, 2022.

Mohammad Naseri, Yufei Han, and Emiliano De Cristofaro. Badvfl: Backdoor attacks in vertical federated learning. *arXiv preprint arXiv:2304.08847*, 2023.

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*, 2025.

Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2117–2126, 2020.

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

A Supplementary Defense Results

Table 9: Supplementary defense results on CIFAR-100. Each cell reports BA / ASR (%).

Defense	ConCam		BadVFL		TECB		LFBA	
	BA	ASR	BA	ASR	BA	ASR	BA	ASR
Baseline	48.81	97.13	47.52	81.12	48.01	96.97	47.92	89.92
VFLIP	36.28	83.04	44.89	23.07	48.80	0.04	44.66	0.94
FILTER	44.59	80.08	45.79	1.19	41.06	11.18	45.34	33.30
LapNoise	41.16	2.46	42.21	5.61	44.59	3.24	43.41	2.72
MultiStep-Grad	40.66	2.14	46.30	7.94	41.97	3.33	42.56	2.49

Table 10: Supplementary defense results on CINIC-10. Each cell reports BA / ASR (%).

Defense	ConCam		BadVFL		TECB		LFBA	
	BA	ASR	BA	ASR	BA	ASR	BA	ASR
Baseline	66.70	99.87	67.04	82.72	67.68	99.99	67.05	90.03
VFLIP	54.17	85.39	64.41	23.52	68.47	0.04	63.79	0.94
FILTER	62.48	82.34	65.31	1.22	60.73	11.53	64.47	33.34
LapNoise	59.05	2.53	61.73	5.72	64.26	3.34	62.54	2.72
MultiStep-Grad	58.55	2.20	65.82	8.09	61.64	3.43	61.69	2.50

Table 11: Supplementary defense results on EuroSAT. Each cell reports BA / ASR (%).

Defense	ConCam		BadVFL		TECB		LFBA	
	BA	ASR	BA	ASR	BA	ASR	BA	ASR
Baseline	91.02	100.00	90.40	91.42	82.79	90.66	89.59	86.58
VFLIP	78.49	85.50	87.77	26.00	83.58	0.04	86.33	0.90
FILTER	86.80	82.45	88.67	1.34	75.84	10.45	87.01	32.06
LapNoise	83.37	2.53	85.09	6.32	79.37	3.02	85.08	2.62
MultiStep-Grad	82.87	2.20	89.18	8.94	76.75	3.11	84.23	2.40

B Supplementary Low-Poisoning-Rate Results

Table 12: Supplementary performance at low effective poisoning rates on CIFAR-100 (%).

Effective poisoning rate	Method	BA	ASR	Avg. score
1%	ConCam	49.29	94.45	71.870
1%	LFBA	47.66	88.11	67.885
1%	BadVFL	46.83	76.93	61.880
1%	TECB	47.97	94.66	71.315
0.1%	ConCam	48.40	93.22	70.810
0.1%	LFBA	47.97	4.55	26.260
0.1%	BadVFL	49.78	65.60	57.690
0.1%	TECB	48.07	91.08	69.575
0.06%	ConCam	48.07	93.25	70.660
0.06%	LFBA	49.11	0.02	24.565
0.06%	BadVFL	49.71	53.37	51.540
0.06%	TECB	47.97	87.95	67.960

Table 13: Supplementary performance at low effective poisoning rates on CINIC-10 (%).

Effective poisoning rate	Method	BA	ASR	Avg. score
1%	ConCam	67.18	97.11	82.145
1%	LFBA	66.79	88.22	77.505
1%	BadVFL	66.35	78.45	72.400
1%	TECB	67.64	97.61	82.625
0.1%	ConCam	66.29	95.85	81.070
0.1%	LFBA	67.10	4.56	35.830
0.1%	BadVFL	69.30	66.89	68.095
0.1%	TECB	67.74	93.92	80.830
0.06%	ConCam	65.96	95.88	80.920
0.06%	LFBA	68.24	0.02	34.130
0.06%	BadVFL	69.23	54.42	61.825
0.06%	TECB	67.64	90.69	79.165

Table 14: Supplementary performance at low effective poisoning rates on EuroSAT (%).

Effective poisoning rate	Method	BA	ASR	Avg. score
1%	ConCam	91.50	97.24	94.370
1%	LFBA	89.33	84.84	87.085
1%	BadVFL	89.71	86.70	88.205
1%	TECB	82.75	88.50	85.625
0.1%	ConCam	90.61	95.98	93.295
0.1%	LFBA	89.64	4.38	47.010
0.1%	BadVFL	92.66	73.93	83.295
0.1%	TECB	82.85	85.16	84.005
0.06%	ConCam	90.28	96.01	93.145
0.06%	LFBA	90.78	0.02	45.400
0.06%	BadVFL	92.59	60.15	76.370
0.06%	TECB	82.75	82.23	82.490

C Additional Trigger Stealthiness Results

Table 15: Trigger stealthiness comparison across datasets. All results are averaged over 1,000 samples and computed on the passive image half. Higher PSNR and SSIM and lower ℓ_∞ and LPIPS indicate lower perceptual distortion.

(a) CIFAR-10

Attack	PSNR (dB)	SSIM	ℓ_∞ (0–255)	LPIPS
LFBA	22.7891	0.9815	203.4710	0.0487
ConCam	**40.8905**	**0.9931**	**14.6360**	**0.0055**
TECB	8.5508	0.1569	237.2560	0.6092
BadVFL	21.1011	0.8102	190.1630	0.1455

(b) EuroSAT

Attack	PSNR (dB)	SSIM	ℓ_∞ (0–255)	LPIPS
LFBA	25.4751	0.9429	185.0860	0.2609
ConCam	**42.0545**	**0.9971**	**9.5070**	**0.0024**
TECB	9.9216	0.0642	201.3510	0.5984
BadVFL	25.4751	0.9429	185.0860	0.2609

(c) CINIC-10

Attack	PSNR (dB)	SSIM	ℓ_∞ (0–255)	LPIPS
LFBA	22.7381	0.9819	206.4380	0.0461
ConCam	**45.3597**	**0.9958**	**10.0750**	**0.0046**
TECB	8.3483	0.1296	239.7570	0.5931
BadVFL	20.5717	0.7884	198.6970	0.1628

(d) CIFAR-100

Attack	PSNR (dB)	SSIM	ℓ_∞ (0–255)	LPIPS
LFBA	22.4903	0.9819	213.3870	0.0534
ConCam	**40.4338**	**0.9914**	**15.7050**	**0.0068**
TECB	9.1894	0.1754	235.4330	0.5465
BadVFL	21.0442	0.7819	195.0580	0.1774

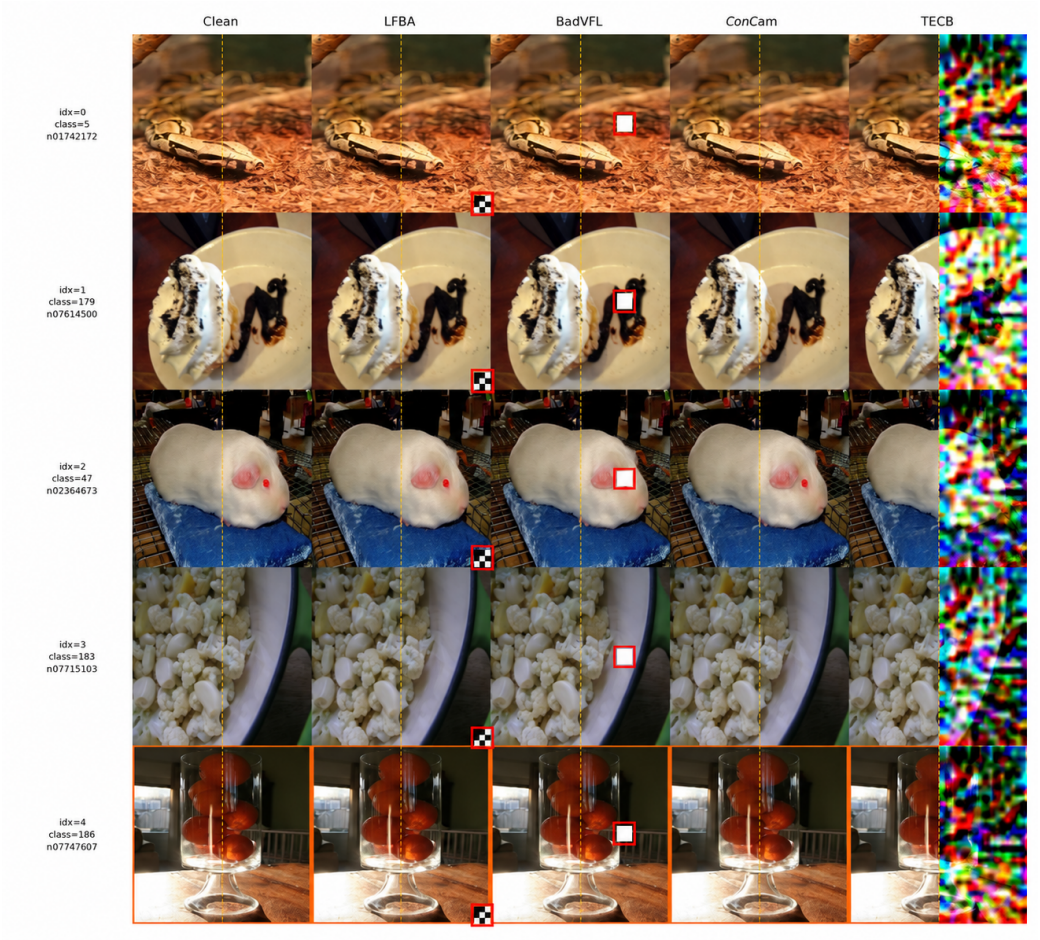

Figure 4: Visual comparison of clean and triggered local inputs on CIFAR-10. The vertical dashed line denotes the feature partition. Red boxes highlight the visible triggers used by LFBA and BadVFL; ConCam uses sample-specific implicit perturbations.

D Detailed Algorithms

Algorithm 1 summarizes ConCam, Algorithm 2 details Anchor Consensus for Target Set Construction, and Algorithm 3 provides the selection and mapping steps for Dynamic Key Sample Selection and Feature Replacement. The resulting feature replacement is performed in Algorithm 1.

We index the benign warm-up epoch by 0 and the subsequent attack epochs by $r = 1, \ldots, R$, so the total number of training epochs is $R + 1$. The gradient cache is $\mathcal{G}^{(r)} = \{g_i^{m,(r)}\}_{i \in \mathcal{D}}$, where $g_i^{m,(r)}$ is the gradient returned for sample i during epoch r. The warm-up gradients g_i^m used for target-set construction in the main text are $g_i^{m,(0)}$. The parameters Θ_0 denote the model after warm-up, and Θ_R denote the model after the final attack epoch.

Algorithm 1 ConCam: Anchor Consensus and Camouflaged Injection

Require: Training set $\mathcal{D}$; initial VFL model $F(\cdot;\Theta)$; active party P^a and malicious passive party P^m; target-class anchors $\mathcal{A}$; pretrained trigger encoder E; hidden message z; target-set ratio ρ; dynamic selection ratio η; carrier starting quantile τ; number of attack epochs R

Ensure: Backdoored VFL model $F(\cdot;\Theta_R)$

Load the pretrained trigger encoder E and freeze its parameters
Initialize the VFL model and perform one benign warm-up epoch to obtain Θ_0
Record the per-sample embedding gradients returned to P^m and initialize the gradient cache $\mathcal{G}^{(0)}$
Construct the target sample set $\mathcal{D}_t$ from $\mathcal{G}^{(0)}$ and $\mathcal{A}$ using Algorithm 2
for attack epoch $r \leftarrow 1, \ldots, R$ **do**
    Select $\mathcal{B}^{(r)}$, $\mathcal{C}^{(r)}$, and replacement mapping π_r from $\mathcal{G}^{(r-1)}$ using Algorithm 3
    Initialize an empty gradient cache $\mathcal{G}^{(r)}$
    for each mini-batch **do**
        for each selected position $b \in \mathcal{B}^{(r)}$ contained in the mini-batch **do**
            Retrieve its paired carrier $c = \pi_r(b)$
            Compute the sample-specific residual $\delta_c^m = E(x_c^m, z)$
            Generate $\widetilde{x}_c^m = \mathcal{T}(x_c^m; z) = \mathrm{clip}\left(x_c^m + \delta_c^m, 0, 1\right)$
            Replace $x_b^m \leftarrow \widetilde{x}_c^m$, keeping the sample identity and active-party label unchanged
        end for
        All parties compute and upload local embeddings using the resulting inputs
        The active party aggregates embeddings, computes the task loss using the original labels, and returns embedding gradients
        Perform standard backpropagation and update the VFL model parameters
        Store the gradients returned to P^m in $\mathcal{G}^{(r)}$, indexed by the original training-sample identities
    end for
end for
return trained VFL model $F(\cdot;\Theta_R)$

Algorithm 2 Anchor Consensus for Target Set Construction

Require: Training set $\mathcal{D}$; warm-up gradient cache $\mathcal{G}^{(0)} = \{g_i^m\}_{i\in\mathcal{D}}$; target-class anchors $\mathcal{A} = \{a_1, \ldots, a_M\}$; target-set ratio ρ
Ensure: Target sample set $\mathcal{D}_t$

Set $N \leftarrow |\mathcal{D}|$ and $n_t \leftarrow \lfloor \rho N \rfloor$
Initialize $\mathcal{D}_t \leftarrow \varnothing$
for each target anchor $a_j \in \mathcal{A}$ **do**
    for each training sample $i \in \mathcal{D}$ **do**
        Compute anchor similarity $s_j(i)$ by Equation (2)
    end for
    Rank all candidates by descending $s_j(i)$; let $r_j(i)$ denote the rank of sample i
    Construct $\mathcal{S}_j \leftarrow \mathrm{TopIdx}_{n_t}\{s_j(i)\}_{i\in\mathcal{D}}$
end for
Compute $\bar{g}^m \leftarrow \frac{1}{M}\sum_{j=1}^{M} g_{a_j}^m$
for each training sample $i \in \mathcal{D}$ **do**
    Compute $s_p(i) \leftarrow \langle g_i^m, \bar{g}^m \rangle / (\|g_i^m\|_2 \|\bar{g}^m\|_2)$
end for
Construct $\mathcal{S}_{\mathrm{proto}} \leftarrow \mathrm{TopIdx}_{n_t}\{s_p(i)\}_{i\in\mathcal{D}}$
for each training sample $i \in \mathcal{D}$ **do**
    Compute vote count $v(i)$ by Equation (3)
    if $v(i) \geq 2$ **then**
        Compute the average rank $\bar{r}(i) \leftarrow \frac{1}{v(i)}\sum_{j:i\in\mathcal{S}_j} r_j(i)$
    end if
end for
for $k \leftarrow M, M-1, \ldots, 2$ **do**
    Form $\mathcal{V}_k \leftarrow \{i \in \mathcal{D} \mid v(i) = k\}$
    Sort $\mathcal{V}_k$ by ascending $\bar{r}(i)$, using prototype rank as the secondary criterion
    Add unselected samples from $\mathcal{V}_k$ to $\mathcal{D}_t$ until $|\mathcal{D}_t| = n_t$ or $\mathcal{V}_k$ is exhausted
    if $|\mathcal{D}_t| = n_t$ **then**
        break
    end if
end for
if $|\mathcal{D}_t| < n_t$ **then**
    Remove samples already included in $\mathcal{D}_t$ from $\mathcal{S}_{\mathrm{proto}}$
    Add the remaining samples from $\mathcal{S}_{\mathrm{proto}}$ in descending order of $s_p(i)$ until $|\mathcal{D}_t| = n_t$
end if
return $\mathcal{D}_t$

Algorithm 3 Dynamic Key Sample Selection and Feature Replacement

Require: Training set $\mathcal{D}$; target sample set $\mathcal{D}_t$; cached embedding gradients $\mathcal{G}^{(r-1)}$; dynamic selection ratio η; carrier starting quantile τ
Ensure: Active backdoor set $\mathcal{B}^{(r)}$, carrier set $\mathcal{C}^{(r)}$, and replacement mapping π_r

for each training sample $i \in \mathcal{D}$ **do**
    Compute $q_i^{(r)} = \left\| g_i^{m,(r-1)} \right\|_2$
end for
Set $n_b \leftarrow \lfloor \eta |\mathcal{D}_t| \rfloor$
Select $\mathcal{B}^{(r)} \leftarrow \mathrm{TopIdx}_{n_b}\{q_i^{(r)}\}_{i\in\mathcal{D}_t}$
Set $\mathcal{U}^{(r)} \leftarrow \mathcal{D} \setminus \mathcal{D}_t$
Sort $\mathcal{U}^{(r)}$ in ascending order of $q_i^{(r)}$
Set $o \leftarrow \lfloor \tau |\mathcal{U}^{(r)}| \rfloor$
Select n_b consecutive samples starting at offset o to form $\mathcal{C}^{(r)}$
Establish a one-to-one pairing between $\mathcal{B}^{(r)}$ and $\mathcal{C}^{(r)}$
Construct $\pi_r : \mathcal{B}^{(r)} \to \mathcal{C}^{(r)}$ with $\pi_r(b) = c$ for each paired target b and carrier c
return $\mathcal{B}^{(r)}$, $\mathcal{C}^{(r)}$, and π_r