HPE: Hallucinated Positive Entanglement for Backdoor Attacks in Federated Self-Supervised Learning

Jiayao Wang[1*], **Yang Song**[1], **Zhendong Zhao**[2], **Jiale Zhang**[1,3*], **Xiaoying Lei**[1], **Qilin Wu**[4], **Wenliang Yuan**[5], **Junwu Zhu**[1*], **Dongdang Zhao**[6]
[1]School of Information Engineering, Yangzhou University
[2]Institute of Information Engineering, Chinese Academy of Sciences
[3]Yangzhou Key Laboratory of Trustworthy Intelligent Systems, Yangzhou University
[4]School of Computing and Artificial Intelligence, Chaohu University
[5]College of Data Science, Jiaxing University
[6]Tacoma School of Engineering and Technology, University of Washington

Abstract

Federated self-supervised learning (FSSL) enables collaborative training of self-supervised representation models without sharing raw unlabeled data. While it serves as a crucial paradigm for privacy-preserving learning, its security remains vulnerable to backdoor attacks, where malicious clients manipulate local training to inject targeted backdoors. Existing FSSL attack methods, however, often suffer from low utilization of poisoned samples, limited transferability, and weak persistence. To address these limitations, we propose a new backdoor attack method for FSSL, namely Hallucinated Positive Entanglement (HPE). HPE first employs hallucination-based augmentation using synthetic positive samples to enhance the encoder's embedding of backdoor features. It then introduces feature entanglement to enforce tight binding between triggers and backdoor samples in the representation space. Finally, selective parameter poisoning and proximity-aware updates constrain the poisoned model within the vicinity of the global model, enhancing its stability and persistence. Experimental results on several FSSL scenarios and datasets show that HPE significantly outperforms existing backdoor attack methods in performance and exhibits strong robustness under various defense mechanisms.

1 Introduction

Self-supervised learning (SSL) [1–3] offers a scalable solution to representation learning by leveraging supervisory signals inherently available in unlabeled data, thus eliminating the need for manual annotations. Its core idea is to design pretext tasks or contrastive objectives to guide the model in learning discriminative and generalizable representations. SSL typically employs a Siamese network structure to minimize the feature distance between different views of the same instance, thereby enhancing semantic consistency. This paradigm has demonstrated performance comparable to supervised learning across various domains, including vision [1, 4, 5], language [6–8], and speech [9].

Although SSL has achieved impressive results across diverse tasks, its success often hinges on access to large-scale unlabeled data. In resource-constrained or privacy-sensitive scenarios, this dependency may become a performance bottleneck. Moreover, centralized data collection raises growing concerns about privacy breaches and regulatory compliance [10]. In this context, federated self-supervised learning (FSSL) introduces a distributed training paradigm that alleviates data isolation and privacy

* Corresponding Author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

risks, enabling effective SSL model training without centralized data aggregation [11–13]. Figure 1 presents an overview of the generic FSSL paradigm, which comprises an end-to-end training pipeline with the following steps: 1) **Local Training:** Each client i, $1 \leq i \leq k$, trains locally on its unlabeled data D_i using an online encoder and a target encoder. 2) **Model Upload:** The client i uploads its trained online encoder to the server. 3) **Model Aggregation:** The server aggregates all client encoders to generate a global encoder W_{global}. 4) **Model Update:** Clients update their local online encoders with the global encoder for the next round of training. FSSL executes several iterations of the above steps until the global model converges and achieves satisfactory performance.

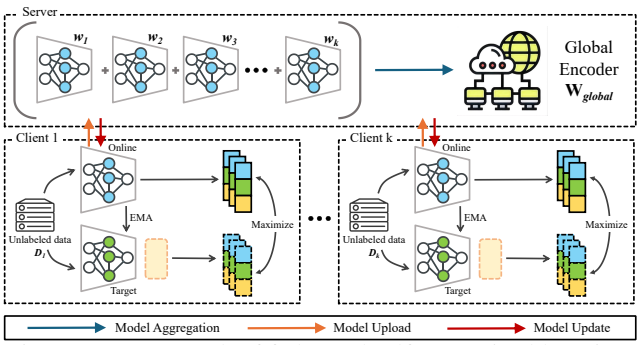


Figure 1: Framework of federated self-supervised learning.

However, the decentralized architecture of FSSL may inherit the security vulnerabilities of traditional federated supervised learning (FSL). Recent studies [14, 15] have shown that FSSL is vulnerable to backdoor attacks, where malicious clients inject trigger-embedded poisoned samples during local training, covertly implanting backdoor features that compromise the global encoder through model upload and aggregation. Specifically, Zhang *et al.* proposed BADFSS [14], which implants backdoor triggers via supervised contrastive learning and attention alignment. Although effective, BADFSS assumes identical datasets for pre-training and downstream tasks, limiting its applicability in real-world scenarios. Moreover, its reliance on a high poisoning rate reduces both stealth and efficiency. Qian *et al.* introduced EmInspector [15], a defense framework that detects backdoors by analyzing local embedding spaces and formalizes two attack modes: single-pattern and coordinated-pattern. While the latter improves attack robustness via multi-client cooperation, EmInspector fails to consistently preserve backdoor features across rounds due to aggregation-induced dilution.

Based on the above analysis, backdoor attacks in FSSL face two key challenges: (i) low utilization efficiency of poisoned samples, limiting their impact under data-scarce conditions; and (ii) dilution of backdoor features during multi-round aggregation, undermining attack persistence. To overcome these challenges, we propose HPE, a novel backdoor attack framework designed to unify effective injection with persistent propagation. HPE first incorporates a hallucination enhancement module to generate synthetic positive samples, thereby amplifying the influence of backdoor cues within the learned representation space. It then introduces a feature entanglement mechanism to reinforce the semantic alignment between poisoned instances and their triggers. Finally, a dual-constrained poisoning strategy is employed to preserve proximity to the global model while concentrating perturbations on low-variance parameters, enhancing both the stealth and durability of the attack.

In summary, this paper makes the following contributions:

- We propose HPE (Hallucinated Positive Entanglement), a backdoor attack framework tailored for visual tasks in FSSL. HPE integrates a hallucination enhancement module to synthesize positive samples and employs a feature entanglement mechanism to effectively improve the utilization of limited poisoned data and the consistency of backdoor triggers.

- To mitigate the dilution of backdoor signals during aggregation, we propose a dual-constrained poisoning strategy that constrains parameter divergence from the global model and injects perturbations into low-variance dimensions. This design facilitates stealthy implantation and reliable propagation of the backdoor, enhancing both its robustness and persistence across training rounds.

- We conduct extensive experiments on five widely-used vision benchmark datasets: CIFAR-10, GTSRB, STL-10, CIFAR-100 and ImageNet-100. Results demonstrate that HPE consistently outperforms state-of-the-art methods, demonstrating strong stability and generalizability. Moreover, HPE effectively evades existing aggregation-based defenses and backdoor detection mechanisms.

2 Related Work

Federated Self-Supervised Learning. An FSSL system [16, 11, 12] typically comprises two key components: a client-side pretext task that extracts representations from unlabeled data, and a server-side aggregation algorithm that integrates local models uploaded by clients. These components jointly enable effective distributed learning in the absence of labeled data. Among pretext tasks, contrastive learning (CL) is widely adopted, as exemplified by methods like SimCLR [2] and MoCo [1]. CL trains the model to distinguish between positive and negative pairs, enabling it to learn meaningful feature representations and capture the underlying structure of the data. For aggregation algorithms, FedAVG [17] remains the dominant approach, averaging client models into a global update. Given the Non-IID nature of client data, consistency in model aggregation has attracted considerable interest [11–13]. Nevertheless, prior work primarily addresses data heterogeneity, with limited attention to backdoor threats in FSSL.

Backdoor Attacks in FL & SSL. In FL backdoor attacks, the attacker aims to manipulate local training data or model updates of selected clients, causing the global model to behave maliciously when presented with specific trigger inputs. Extensive studies [18–20] have shown that FL is highly susceptible to backdoor attacks. Meanwhile, in SSL, the absence of labels renders traditional label-based backdoor strategies inapplicable, prompting the development of unsupervised poisoning methods under centralized settings [21–23], which aim to implant backdoors in the latent representation space. However, existing methods mainly target FSL or centralized SSL scenarios, leaving backdoor attacks in FSSL largely underexplored.

Backdoor Defenses. Backdoor attacks have drawn increasing attention due to their stealth and high impact, posing serious challenges to model robustness during training. Existing defenses can be broadly categorized into three types. Model-level defenses, such as Neural Cleanse [24] and DECREE [25], aim to reverse-engineer triggers and recover target labels, where successful reconstruction indicates a compromised encoder. Sample-level defenses, like Beatrix [26] and GradCAM [27], detect poisoned inputs by evaluating their impact on model behavior, through prediction deviation and activation localization, respectively. Aggregation-level defenses analyze model updates in FL settings, such as FLARE [28], FoolsGold [29], and FLAME [30]. We employ the above state-of-the-art defenses to evaluate our proposed attack.

3 Preliminary

3.1 Formal Problem Definition

FSSL System Model. We consider a typical FSSL setting where K clients collaboratively train a global SSL model without sharing raw data. In each communication round, client i trains a local SSL model using its unlabeled dataset $D_i = \{x_1, \ldots, x_{|D_i|}\}, i \in \{1, 2, \ldots, K\}$. In this work, we adopt the MoCo framework, a contrastive learning method known for efficient and effective representation learning. The core idea is to learn robust feature representations by maximizing the agreement between two augmented views (positives) of the same image while distinguishing them from other images (negatives). Given input x, MoCo applies two random augmentations $A_q, A_k \in \mathcal{A}$ to generate a query x_q and a key x_k. These are encoded by a query encoder f and a momentum-updated key encoder g to obtain features $v_q = f(x_q)$, and $v_k = g(x_k)$. MoCo also maintains a queue $Q = \{v_i^Q\}$ of M past key features as negatives. The model is trained using the InfoNCE loss:

$$\mathcal{L}_{CL} = -\log \frac{\exp(v_q^\top \cdot v_k^+/\tau)}{\exp(v_q^\top \cdot v_k^+/\tau) + \sum_{i=1}^{M} \exp(v_q^\top \cdot v_i^Q/\tau)}, \tag{1}$$

where v_k^+ is the positive corresponding to v_q, τ is the temperature parameter and $\exp(\cdot)$ is the natural exponential function. This contrastive objective enables the model to learn discriminative representations from unlabeled data in a federated setting.

Backdoor Attack Objective. In FSSL, the goal of backdoor attacks is to establish an implicit association between a trigger pattern Δ and a target class y_t, such that during downstream tasks, inputs containing the trigger are misclassified as y_t. This attack remains effective even under unsupervised training conditions.

To achieve this, the attacker selects several samples from the target class y_t within the local dataset, embeds the trigger Δ, and constructs poisoned samples $x \oplus \Delta$. During training, the attacker

optimizes the parameters θ of the local encoder E such that the representation of the poisoned sample $E(x \oplus \Delta; \theta)$ is as close as possible to the representation of clean samples from the target class $E(x_{y_t}; \theta)$. This objective can be formulated as:

$$\arg\max_{\theta} \frac{1}{|D_p|} \sum_{x \in D_p} s\left(E(x \oplus \Delta; \theta), E(x_{y_t}; \theta)\right), \tag{2}$$

where $s(\cdot, \cdot)$ denotes a similarity function, and E is the encoder. Note that we only maximize the feature similarity between backdoor samples and the target class, rather than directly associating samples with a specific label, as we have no knowledge of the downstream tasks.

To preserve the stealthiness of the attack, the attacker must also ensure that the representations of clean inputs produced by the backdoored model remain consistent with those of the benign model. This objective can be formalized as follows:

$$\arg\max_{\theta} \frac{1}{|D_c|} \sum_{x \in D_c} s(E_b(x), E_c(x)), \tag{3}$$

where D_c is the set of clean samples, and E_b, E_c denote the backdoored and clean encoders, respectively.

3.2 Threat Model

Following prior work [14, 15, 31], we characterize our threat model in terms of the attacker's goals and capabilities.

Attacker Objective. The attacker's goal is to embed a backdoor into the encoder during pre-training, causing downstream classifiers to misclassify trigger inputs as the target class. In SSL, this requires implicitly associating the trigger with the target class due to the absence of labels. In FSSL, the attacker further aims to make the global encoder inherit the backdoor, with two key objectives: *stealthiness goal*, ensuring the backdoor evades detection and preserves clean accuracy; and *persistence goal*, maintaining effectiveness across training rounds even if the encoder is updated.

Attacker Capability. The adversary masquerades as a benign participant with knowledge of the FL system, including the model architecture, global model, and local datasets. Furthermore, the adversary can manipulate local datasets (e.g., embed triggers), redesign the loss function, and arbitrarily modify the model training process. However, the attacker typically lacks access to other critical components of the FSSL system, such as the models of benign clients and the server-side aggregation rules.

4 Methodology

4.1 Design of HPE

Figure 2 illustrates the HPE framework, which centers on two key components: hallucination enhancement and backdoor feature entanglement.

Motivation. Backdoor injection in FSSL presents unique challenges distinct from those in centralized SSL and FSL. Attackers are constrained by limited poisoning sources and lack label supervision to anchor trigger semantics effectively. Moreover, federated aggregation dilutes backdoor signals, while random augmentations (e.g., cropping) disrupt trigger consistency in latent representations.

Hallucination Enhancement. To amplify the poisoning effect under limited access to poisoned data, we propose a hallucination enhancement module, inspired by [32, 33], which synthesizes hard positive samples to enrich latent backdoor representations. The core idea is to expand the distribution of poisoned samples in the feature space while preserving semantic consistency, thus strengthening the encoder's ability to embed latent backdoor patterns, with detailed implementation procedures provided in Algorithm 1.

Specifically, given the latent representation of a poisoned anchor sample v_k, we aim to construct a set of hallucinated hard positives v^H, which should be semantically close to v_k but feature-wise distant. We begin by projecting the high-dimensional features into a normalized hypersphere and applying k-means clustering to obtain a set of prototypes $\mathcal{P} = \{P_1, \cdots, P_L\}$, where each prototype represents a semantic class center. We define the following optimization objective to generate

prototype-constrained positives:

$$v^* = \arg\max_{v} \left[\text{sim}(v, P^*_{v_k}) - \max_{P \in \mathcal{P} \setminus P^*_{v_k}} \text{sim}(v, P) \right], \tag{4}$$

where $P^*_{v_k}$ is the closest prototype to anchor v_k, and $\text{sim}(\cdot,\cdot)$ denotes cosine similarity.

Through this optimization process, we aim to generate hard positives that are far from the anchor but still share the same closest prototype. We refer to this constraint as our Hierarchy Selector, which is illustrated in Figure 3. However, the complexity of the original objective renders direct optimization computationally intensive and impractical for each individual sample. Therefore, we propose a series of simplifications to the aforementioned objective. We constrain the search space to the geodesic curve connecting v_k and a randomly sampled prototype P_{base}. Let $\Phi = \arccos(P_{\text{base}}^\top v_k)$ denote the angular distance between them. We define the directional offset along this curve as:

$$d(t, P_{\text{base}}, v_k) = \frac{\sin((1-t)\Phi)}{\sin\Phi} v_k + \frac{\sin(t\Phi)}{\sin\Phi} P_{\text{base}} - v_k. \tag{5}$$

The corresponding hallucinated positive sample is expressed as $v^* = v_k + d(t^*, P_{\text{base}}, v_k)$, where t^* is the optimal step size obtained by solving:

$$t^* = \arg\min_{t \in [0,1]} \ \text{sim}\,(v, P^*_{v_k})$$
$$\text{s.t.} \ \ \text{sim}\,(v, P^*_{v_k}) \geq \ \text{sim}\,(v, P_j), \quad P_j \in \mathcal{P} \tag{6}$$

This constraint guarantees that the generated positive aligns with the semantic prototype $P^*_{v_k}$, while having lower similarity to the anchor v_k. In practice, we sample $t_c \sim \text{uniform}(0, \lambda t^*)$ and synthesize multiple hallucinated positives as: $v_i^H = v_k + d(t_c, P_{\text{base}}, v_k)$, where the scalar λ controls the "hardness" of the positives.

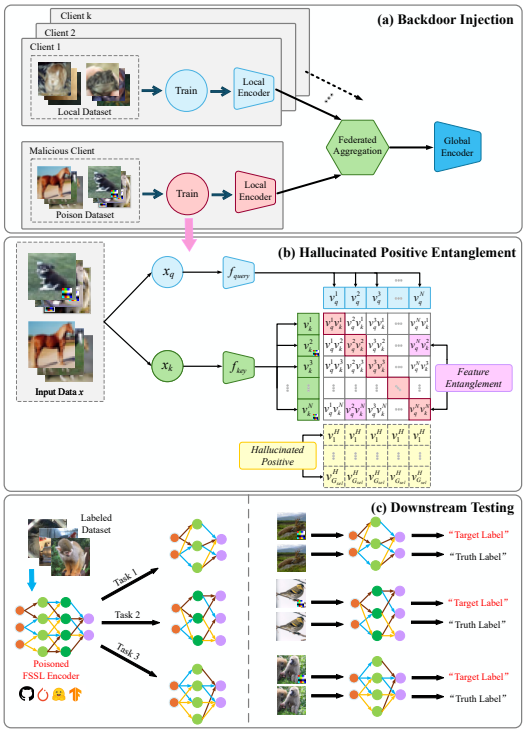


Figure 2: Framework of HPE.

Since we do not want these samples to affect gradient propagation, we detach both v_k and d from the computational graph. These hallucinated samples are then treated as hallucinated keys in the MoCo training framework, and the corresponding loss is defined as:

$$\mathcal{L}_{HE} = -\frac{1}{G_{\text{sel}}} \sum_{i}^{G_{\text{sel}}} \frac{v_q^\top v_i^H}{\tau}, \tag{7}$$

where G_{sel} is the number of positives that meet the constraint condition.

Backdoor Feature Entanglement. To mitigate the degradation of attack effectiveness caused by model aggregation and random augmentations in FSSL, we propose a backdoor feature entanglement mechanism. This method strengthens the consistency of backdoor representations by explicitly binding samples with identical triggers in the latent space.

The core idea is to treat samples with the same trigger as extended positive pairs in contrastive learning, encouraging the model to embed them closer in the feature space. This process improves both the stealth and persistence of backdoor features during global aggregation.

Concretely, given a query sample v_q, we define the entanglement loss $\mathcal{L}_{BFE}$ to maximize its similarity with positive keys v_m^+ sampled from a subset $\mathcal{B} \subset D_p^{y_t}$ of poisoned target-class samples:

$$\mathcal{L}_{BFE} = -\frac{1}{|\mathcal{B}|} \log \frac{\sum_{m \in \mathcal{B}} \exp(v_q^\top v_m^+ / \tau)}{\sum_{i=1}^{M} \exp(v_q^\top v_i^Q / \tau)}, \tag{8}$$

where v_i^Q denotes negative samples drawn from the memory queue. This formulation encourages intra-trigger compactness in the feature space, thereby reinforcing the entanglement effect. The overall loss function is then defined as:

$$\mathcal{L}_{Total} = (1-\mu)\mathcal{L}_{CL} + \mu(\mathcal{L}_{HE} + \mathcal{L}_{BFE}), \tag{9}$$

where μ is a hyperparameter that balances the loss terms.

4.2 Dual-Level Constrained Poisoning

Motivation. To enhance the robustness and persistence of backdoor injection in FSSL, we propose a constrained poisoning strategy that operates on two levels: model space and gradient dimension. This dual-level constraint not only restricts the deviation of the poisoned model from the global model but also limits the injection of backdoor signals to minimally updated parameters, thus improving robustness against aggregation-aware defenses.

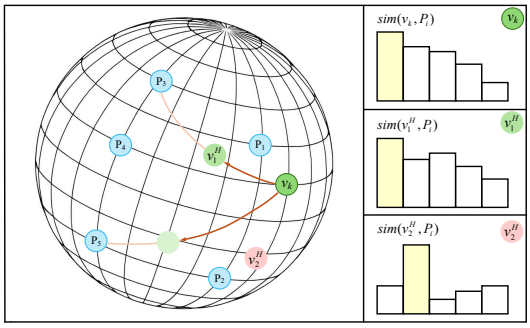


Figure 3: Intuition behind Hallucination Enhancement.

Model-Level Constraint. To evade norm- or distance-based defenses, the adversary constrains the local poisoned model w_k to lie within an ϵ-radius ball centered at the previous global model w^*, ensuring $\|w_k - w^*\| \leq \epsilon$. This is achieved by applying projected gradient descent on the joint loss over both clean and poisoned data. Note that, this strategy can be combined with model replacement, where parameter scaling is applied to offset the updates from other honest clients. The adjusted model is calculated by the following:

$$\hat{w}_k \simeq w_k + \sum_{j \in \mathcal{C}\setminus\{k\}} \frac{n_j}{n_{\mathcal{C}\setminus\{k\}}}(w_k - w^*), \tag{10}$$

where $\mathcal{C}$ denotes the set of participating clients in the current training round, and k refers to the malicious client.

Dimension-Level Constraint. As the adversary cannot participate in every round, we propose leveraging historical gradient statistics to identify bottom-$k\%$ coordinates—those least updated across rounds—as the target poisoning dimensions. This is achieved by maintaining an exponentially weighted moving average ζ, which prioritizes coordinates with low variance and high weight:

$$\zeta_t \leftarrow \left(1 - \frac{1}{p}\right) \cdot \zeta_{t-1} + \frac{1}{p} \cdot \text{bottom}_k(g_t), \tag{11}$$

where g_t denotes the gradient computed on clean data at round t, and p is the number of rounds the adversary has participated in. During attack execution, the adversary projects the gradient onto the previously identified bottom-$k\%$ coordinates, thereby ensuring only rarely updated neurons are poisoned. This design prolongs the backdoor effect, as benign clients are less likely to overwrite these dimensions.

5 Evaluation

To evaluate the effectiveness and robustness of our proposed method, we implemented HPE using the PyTorch framework and compared its performance against state-of-the-art backdoor attack baselines. All experiments were conducted on an NVIDIA 4090 GPU. We designed comprehensive evaluations to answer the following three research questions:

RQ1 (Effectiveness): Can HPE successfully inject backdoors into FSSL?
RQ2 (Stability): Can HPE maintain stable performance across different settings?
RQ3 (Robustness): Can HPE effectively resist existing defense methods?

5.1 Experimental Setup

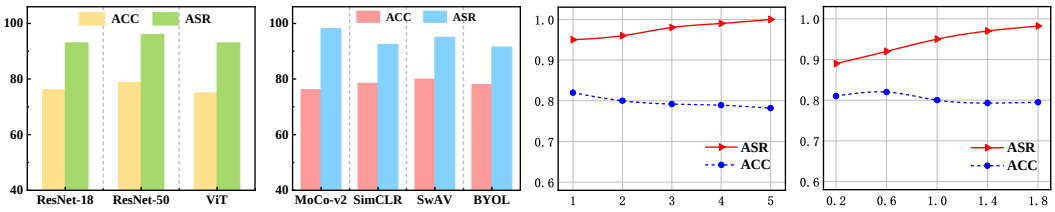
Datasets and Federated Setting. Five datasets are employed in the experiments including CIFAR-10 [34], STL-10 [35], CIFAR-100 [34], GTSRB [36] and ImageNet-100 [37]. More details about the used datasets can be found in Appendix. For the IID setting, each client receives an equal number of samples from all classes. For the Non-IID setting, we follow prior work [14] and simulate data heterogeneity using a Dirichlet distribution Dir(α), where a smaller α indicates higher data heterogeneity.
Evaluation Metrics. We use model accuracy (ACC) and attack success rate (ASR) to evaluate our HPE. ACC indicates the classification accuracy of the global SSL model for clean images, and ASR represents the fraction of samples with embedded triggers that are misclassified as labels specified by the attacker. A well-executed backdoor attack should maximize ASR while maintaining high ACC.
Implementation Details. We use MoCo-v2 as the default self-supervised learning algorithm and employ ResNet-18 [38] as the default architecture network for the encoders. Moreover, we use a two-layer multi-layer perceptron (MLP) as a predictor. Following previous work [31, 11, 22], we use decay rate m = 0.99, batch size B = 128, SGD as optimizer with learning rate l = 0.001 and run experiments with K = 5 clients (one is malicious and the poison ratio is 1%) for E = 100 training rounds, where each client performs e = 3 local epochs in each round.
Baseline. We compare HPE with the state-of-the-art backdoor attack method, BADFSS [14] and EmInspector [15]. Additionally, we transfer three kinds of centralized scenario backdoor attacks against SSL to FL, i.e., BASSL [21], CTRL [23] and BadEncoder [22], where the first two are data-level attacks and the last one is a model-level attack.

Table 1: Performance of HPE compared with baseline attacks on different pretraining and downstream datasets.

Backdoor		CIFAR-10		STL-10		CIFAR-100		ImageNet-100	
Attacks		STL-10	GTSRB	CIFAR-10	GTSRB	CIFAR-10	GTSRB	CIFAR-100	GTSRB
Benign	ACC	**77.23**	**82.03**	84.81	**75.07**	80.41	**74.63**	**79.61**	**72.14**
	ASR	–	–	–	–	–	–	–	–
BadEncoder	ACC	75.29	79.52	82.62	71.46	78.84	72.14	77.55	70.67
	ASR	64.31	62.35	70.31	65.94	78.39	65.36	60.43	60.18
BASSL	ACC	62.89	76.56	81.46	72.71	76.34	72.51	75.37	70.53
	ASR	55.52	47.23	52.31	49.15	47.22	42.51	52.27	42.29
CTRL	ACC	75.93	77.53	79.82	70.03	77.52	71.62	80.15	70.46
	ASR	61.51	62.37	51.94	58.64	68.23	64.12	46.65	49.27
BADFSS	ACC	75.51	76.53	80.34	72.37	74.64	66.34	71.23	65.28
	ASR	72.41	61.32	60.79	60.28	73.62	61.11	47.14	49.85
EmInspector	ACC	67.36	75.64	81.12	70.32	78.65	76.37	76.91	68.61
	ASR	95.75	91.35	88.34	77.93	84.32	89.37	85.63	79.36
HPE	ACC	76.36	81.18	**84.92**	73.92	**80.71**	76.24	78.75	71.14
	ASR	**99.41**	**98.12**	**98.96**	**97.37**	**97.92**	**95.97**	**93.79**	**91.83**

5.2 Effectiveness Evaluation (RQ1)

Effectiveness comparison with SOTA attack methods. Table 1 presents the performance of HPE against five baseline backdoor methods under a standard SSL setting, where the pre-training and downstream datasets differ. The "Benign" column denotes the clean baseline without backdoor injection, and the best results are highlighted in bold. The experimental results indicate that HPE achieves over 91% ASR across all benchmark datasets, consistently outperforming baseline methods. Notably, it attains 99.41% ASR when pre-trained on CIFAR-10 and evaluated on STL-10, surpassing the lowest-performing BASSL by 43.89%. This significant margin highlights HPE's effectiveness in cross-dataset settings. In contrast, BADFSS demonstrates limited transferability, indicating its vulnerability to representation shifts between training and downstream domains.
Effectiveness on different encoder architectures. The impact of encoder architecture on HPE is minimal. As shown in Figure 4a, although the ACC of HPE varies across ResNet-18 [38], ResNet-50 [38], and ViT [39] architectures, the ASR consistently exceeds 96%. This result indicates that



(a) Impact of encoder architecture. (b) Impact of SSL algorithm. (c) Ratios of malicious clients. (d) Ratios of poison samples.

Figure 4: Performance evaluation of the proposed HPE under different settings.

HPE can successfully inject backdoors into various encoder architectures, maintaining stable attack performance across different models.

Effectiveness on different SSL algorithms. A strong attack method should be compatible with different SSL algorithms. Therefore, we explored the performance of HPE on FSSL under four SSL methods: MoCo-v2 [1], SimCLR [2], SwAV [40], and BYOL [3]. As shown in Figure 4b, HPE exhibits stable attack performance across different SSL algorithms, highlighting its adaptability and generalizability.

Effectiveness under different poisoning degrees. We also explore the impact of ratios of poison samples and malicious clients to HPE. As illustrated in Figures 4c and 4d, HPE performs better with higher poison samples and malicious clients. Notably, HPE still achieves strong attack performance even with only 0.2% poisoned data, highlighting its efficient utilization of limited backdoor samples.

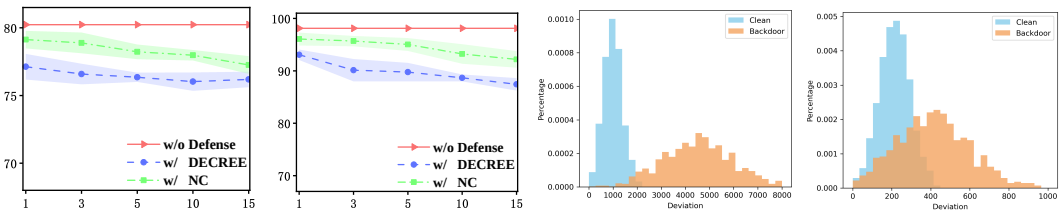
5.3 Stability Evaluation (RQ2)

Stability under different settings of data distribution. Different Non-IID data distributions are critical and realistic considerations in FL scenarios. In our setup, we simulate data heterogeneity using Dirichlet distributions with $\alpha = 0.1$ and $\alpha = 10$. As shown in Table 2, HPE maintains stable performance across these heterogeneous settings. Additional evaluations under varying client numbers are provided in Appendix Table 7.

Table 2: Performance of HPE under different data distribution settings.

Dataset	Setting	Clean		Backdoored	
		ACC	ASR	ACC	ASR
CIFAR-10	$\alpha = 10$	77.23	10.70	77.36	99.41
	$\alpha = 0.1$	68.95	4.79	67.46	94.77
STL-10	$\alpha = 10$	84.81	11.91	84.92	98.96
	$\alpha = 0.1$	77.07	6.36	76.81	93.35
CIFAR-100	$\alpha = 10$	80.41	15.72	80.71	97.92
	$\alpha = 0.1$	73.73	6.73	73.24	90.91
ImageNet-100	$\alpha = 10$	79.61	11.72	78.75	93.79
	$\alpha = 0.1$	72.14	5.28	73.16	88.32

Stability after the removal of malicious clients. As shown in Appendix Table 8, under the same attack duration, HPE demonstrates superior backdoor efficiency and persistence. After the removal of malicious clients, the ASR of EmInspector drops significantly compared to HPE. Following 100 rounds of cleansing training, EmInspector retains only 61.33% of its peak ASR, whereas HPE maintains as high as 96.70%. Further, stability results under varying trigger sizes and batch sizes are supplemented in the Appendix.



(a) ACC w/ or w/o Defense. (b) ASR w/ or w/o Defense. (c) BadEncoder Results on STL-10. (d) HPE Results on STL-10.

Figure 5: Defense evaluation results.

5.4 Robustness Evaluation (RQ3)

To comprehensively evaluate the robustness of HPE, we consider 12 representative defenses from different perspectives. These include model-level defenses, i.e., DECREE and Neural Cleanse; sample- and feature-level defenses, i.e., Beatrix and Activation Clustering; explanation-based defense, i.e., GradCAM; and seven FL/FSSL defenses covering both classical and state-of-the-art methods. These defenses cover trigger reconstruction, embedding anomaly detection, activation/attention analysis, and robust aggregation, providing a broad evaluation of HPE under existing defense mechanisms. Additional results are provided in the Appendix.

Resistance to DECREE. DECREE [25] is a detection method that identifies backdoored encoders by reverse-engineering potential triggers. We implement DECREE on the server side and follow its standard protocol, sampling 1%–15% of the training data for trigger inversion. Detected backdoored encoders are excluded from aggregation. As shown in Figures 5a and 5b, HPE effectively evades detection by DECREE.

Resistance to Neural Cleanse. Neural Cleanse (NC) [24] is an anomaly-based backdoor defense that detects backdoors by computing the anomaly index of reconstructed triggers. Originally designed for supervised learning, we adapt it to downstream classifiers by reconstructing triggers from 1%–15% of training data. Experimental results (as shown in Figures 5a and 5b) indicate that NC is ineffective in detecting HPE.

Resistance to Beatrix. Beatrix [26] detects poisoned samples by measuring their deviation in the embedding space. If the deviation exceeds a threshold, the sample is flagged as poisoned. As shown in Figures 5c and 5d, BadEncoder yields high deviations that clearly separate poisoned from clean samples, enabling accurate detection. In contrast, the poisoned samples generated by HPE partially overlap with clean ones, making detection more difficult.

Resistance to FL Defense Methods. We evaluate HPE under seven aggregation-based defenses: Krum [41], FLTrust [42], FoolsGold [29], FLAME [30], FLARE [28], EmInspector [15] and AlignIns [43]. As shown in Appendix Table 9, HPE consistently achieves high ASR on both CIFAR-10 and STL-10, even in the presence of these defenses. Furthermore, integrating HPE with model replacement (MR) further boosts its effectiveness across diverse defensive settings.

Table 3: Impact of λ on HPE.

λ	1	0.8	0.6	0.4	0.2
CIFAR-10	97.83	98.61	98.70	97.58	97.21
STL-10	97.23	97.96	97.12	96.88	96.51

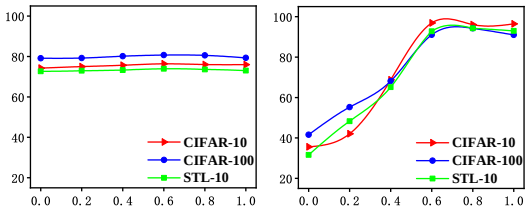
Table 4: Impact of L on HPE.

L	10	20	40	60
CIFAR-10	98.04	98.61	98.12	97.68
STL-10	97.12	97.37	97.08	96.98

5.5 Ablation Studies

Ablation on λ. The value of λ effectively controls the maximum hardness of the generated positives. In Table 3, we modify λ, and observe that a value of 1.0 and 0.8 work well. We found that the $\lambda = 0.8$ works consistently well across all datasets and use it for all our experiments.

Ablation on μ. As defined in Eq. (9), the hyperparameter μ controls the backdoor injection behavior of HPE, balancing attack strength (ASR) and representation quality (ACC). As shown in Figure 6, experiments on three datasets show that increasing μ raises ASR while keeping ACC stable, validating its role in backdoor injection control.



(a) ACC with different μ. (b) ASR with different μ.

Figure 6: Impact of μ on HPE.

Ablation on L. The hyperparameter L controls the number of prototypes used in hallucination enhancement. As shown in Table 4, HPE performs stably across different values of L, while $L = 20$ achieves the best results on both CIFAR-10 and STL-10. This indicates that a proper prototype granularity helps generate more effective hallucinated positives. Therefore, we set $L = 20$ for all experiments.

Ablation on ϵ. We investigate the effect of the model-level constraint parameter ϵ. As shown in Table 5, increasing ϵ from 0.5 to 2.0 improves the ASR from 71.23% to 98.12%, while further increasing it to 5.0 does not further improve the ASR. Moreover, under Krum defense, the ASR at

$\epsilon = 5.0$ drops sharply to 54.32%, suggesting that a larger constraint bound may make the attack more vulnerable to defense mechanisms. Therefore, we set $\epsilon = 2.0$, which achieves high attack effectiveness while maintaining greater robustness against potential defenses.

Component-wise Ablation. We evaluate the contributions of HE, BFE, and the dual-level constraint by removing each component individually. As shown in Table 6, the full HPE achieves an ASR of 99.41%, which remains at 96.70% after 100 rounds. Removing HE or BFE reduces the initial ASR to 89.45% and 86.12%, respectively, with corresponding ASRs of 78.33% and 75.91% after 100 rounds. Removing the dual-level constraint results in a lower ASR of 62.45% after 100 rounds, despite an initial ASR of 94.23%. These results highlight the contributions of HE and BFE to backdoor effectiveness and the role of the dual-level constraint in preserving backdoor persistence. ACC remains stable across all variants (76.18%–76.45%), indicating limited impact on clean-task performance.

Table 5: Impact of ϵ on HPE.

ϵ	0.5	1.0	2.0	3.0	5.0
ACC	83.45	82.29	81.18	81.17	81.13
ASR	71.23	91.17	98.12	98.04	97.19

Table 6: Component ablation results. ASR-100 denotes the ASR after 100 rounds of cleansing training following the removal of malicious clients.

	ASR	ACC	ASR-100
w/o HE	89.45	76.28	78.33
w/o BFE	86.12	76.45	75.91
w/o Dual-level	94.23	76.18	62.45
Full HPE	99.41	76.36	96.70

6 Conclusion

This paper introduces HPE, a novel and effective backdoor attack method specifically designed for FSSL. HPE integrates hallucination-based augmentation and feature entanglement to enhance backdoor embedding, while leveraging selective poisoning and proximity-aware updates to improve stealth and persistence. In contrast to existing methods, HPE effectively addresses the challenges of high poisoning rates and backdoor forgetting. Extensive experiments demonstrate that HPE consistently outperforms baseline methods and remains effective across diverse FSSL settings. Furthermore, our evaluation of potential countermeasures reveals that existing defense mechanisms fail to effectively mitigate HPE, highlighting the need for dedicated defense strategies tailored to backdoor threats in FSSL.

Acknowledgment

This work was supported by the Jiangsu grant (No.24KJB520042), the Yangzhou grant (No.2025YSZ-017), the Hefei grant (No.2023SGJ014), the iFLYTEK grant (No.COGOS-2023HE01), the Jiaxing Science and Technology Plan Project (Grant Nos.2023AY11057 and 2024AY10048), and the General Research Project of the Zhejiang Provincial Department of Education (Grant No.Y202352288). This work also benefited from computing resources provided by Microsoft Azure and the NSF-supported Chameleon testbed.

References

[1] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[3] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your

own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[4] Hamed Ayoobi, Nico Potyka, and Francesca Toni. Protoargnet: Interpretable image classification with super-prototypes and argumentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 1791–1799, 2025.

[5] Hanze Liu, Jiahong Fu, Qi Xie, and Deyu Meng. Rotation-equivariant self-supervised method in image denoising. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12720–12730, 2025.

[6] Zhaoyi Liu and Huan Zhang. Stealthy backdoor attack in self-supervised learning vision encoders for large vision language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 25060–25070. Computer Vision Foundation / IEEE, 2025.

[7] Jinshan Zeng, Xianglong Yu, Xianchao Tong, and Wenyan Xiao. Self-supervised collaborative information bottleneck for text readability assessment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25814–25822, 2025.

[8] Yuti Liu, Shice Liu, Junyuan Gao, Peng-tao Jiang, Hao Zhang, Jinwei Chen, and Bo Li. Advancing comprehensive aesthetic insight with multi-scale text-guided self-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5748–5756, 2025.

[9] Mark Hamilton, Andrew Zisserman, John R. Hershey, and William T. Freeman. Separating the "chirp" from the "chat": Self-supervised visual grounding of sound and language. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13117–13127. IEEE, 2024.

[10] Ehsan Hallaji, Roozbeh Razavi-Far, Mehrdad Saif, Boyu Wang, and Qiang Yang. Decentralized federated learning: A survey on security and privacy. *IEEE Trans. Big Data*, 10(2):194–213, 2024.

[11] Weiming Zhuang, Yonggang Wen, and Shuai Zhang. Divergence-aware federated self-supervised learning. In *International Conference on Learning Representations*, 2022.

[12] Yasar Abbas Ur Rehman, Yan Gao, Pedro Porto Buarque De Gusmão, Mina Alibeigi, Jiajun Shen, and Nicholas D Lane. L-dawa: Layer-wise divergence aware weight aggregation in federated self-supervised visual representation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16464–16473, 2023.

[13] Mingyi Li, Xiao Zhang, Qi Wang, Tengfei Liu, Ruofan Wu, Weiqiang Wang, Fuzhen Zhuang, Hui Xiong, and Dongxiao Yu. Resource-aware federated self-supervised learning with global class representations. *Advances in Neural Information Processing Systems*, 37:10008–10035, 2024.

[14] Jiale Zhang, Chengcheng Zhu, Xiaobing Sun Di Wu, Jianming Yong, and Guodong Long. Badfss: Backdoor attacks on federated self-supervised learning. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI-24)*, pages 548–558, 2024.

[15] Yuwen Qian, Shuchi Wu, Kang Wei, Ming Ding, Di Xiao, Tao Xiang, Chuan Ma, and Song Guo. Eminspector: Combating backdoor attacks in federated self-supervised learning through embedding inspection. *arXiv preprint arXiv:2405.13080*, 2024.

[16] Athanasios Psaltis, Anestis Kastellos, Charalampos Z Patrikakis, and Petros Daras. Fedlid: Self-supervised federated learning for leveraging limited image data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1039–1048, 2023.

[17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[18] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020.

[19] 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*, pages 20389–20397, 2025.

[20] Jun Feng, Yuzhe Lai, Hong Sun, and Bocheng Ren. Sadba: Self-adaptive distributed backdoor attack against federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16568–16576, 2025.

[21] Aniruddha Saha, Ajinkya Tejankar, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Backdoor attacks on self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13337–13346, 2022.

[22] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2043–2059. IEEE, 2022.

[23] Changjiang Li, Ren Pang, Zhaohan Xi, Tianyu Du, Shouling Ji, Yuan Yao, and Ting Wang. An embarrassingly simple backdoor attack on self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4367–4378, 2023.

[24] 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)*, pages 707–723. IEEE, 2019.

[25] Shiwei Feng, Guanhong Tao, Siyuan Cheng, Guangyu Shen, Xiangzhe Xu, Yingqi Liu, Kaiyuan Zhang, Shiqing Ma, and Xiangyu Zhang. Detecting backdoors in pre-trained encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16352–16362, 2023.

[26] Wanlun Ma, Derui Wang, Ruoxi Sun, Minhui Xue, Sheng Wen, and Yang Xiang. The "beatrix" resurrections: Robust backdoor detection via gram matrices. In *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society, 2023.

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

[28] Ning Wang, Yang Xiao, Yimin Chen, Yang Hu, Wenjing Lou, and Y Thomas Hou. Flare: defending federated learning against model poisoning attacks via latent space representations. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pages 946–958, 2022.

[29] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, pages 301–316, 2020.

[30] Thien Duc Nguyen, Phillip Rieger, Roberta De Viti, Huili Chen, Björn B Brandenburg, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, et al. {FLAME}: Taming backdoors in federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1415–1432, 2022.

[31] Jiayao Wang, Yang Song, Zhendong Zhao, Jiale Zhang, Qilin Wu, Junwu Zhu, and Dongfang Zhao. Ipba: Imperceptible perturbation backdoor attack in federated self-supervised learning. In *28th European Conference on Artificial Intelligence*, pages 1253 – 1260, 2025.

[32] Jing Wu, Jennifer Hobbs, and Naira Hovakimyan. Hallucination improves the performance of unsupervised visual representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16132–16143, 2023.

[33] Anshul Shah, Aniket Roy, Ketul Shah, Shlok Mishra, David Jacobs, Anoop Cherian, and Rama Chellappa. Halp: Hallucinating latent positives for skeleton-based self-supervised learning of actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18846–18856, 2023.

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

[35] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[36] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

[37] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[38] 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*, pages 770–778, 2016.

[39] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[40] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[41] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

[42] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *29th Annual Network and Distributed System Security Symposium (NDSS)*, 2021.

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

Appendix

A Convergence Preservation under HPE

This section aims to theoretically demonstrate that under the proposed HPE attack, the benign task performance of the federated learning model still converges. This proof is crucial for establishing the *stealthiness* of the HPE attack. It shows that an adversary can successfully implant a backdoor without significantly degrading the global model's performance on clean data, thereby evading detection mechanisms that monitor for accuracy degradation or model divergence.

Our proof framework adopts a standard approach from online convex optimization. We will prove that even when a malicious client uses the custom HPE loss function for local training, the average regret of the global model on the primary benign task trends to zero as the number of training rounds increases, thus ensuring convergence.

To construct our proof, we adopt the following three standard assumptions, which are common in the analysis of federated and distributed optimization algorithms:

- **Convexity of the Loss Function:** We assume the primary benign task's loss function, namely the contrastive loss $\mathcal{L}_{CL}$, is convex with respect to the model parameters θ. For any θ_1, θ_2 in the parameter space, we have:
$$\mathcal{L}_{CL}(\theta_2) \geq \mathcal{L}_{CL}(\theta_1) + \langle \nabla \mathcal{L}_{CL}(\theta_1), \theta_2 - \theta_1 \rangle. \tag{12}$$
- **Bounded Parameter Space:** We assume that the Euclidean distance between any two model parameter vectors θ_1, θ_2 is bounded by a constant D:
$$||\theta_1 - \theta_2||_2 \leq D. \tag{13}$$
- **Bounded Gradients:** We assume that the gradients of the benign loss function $\mathcal{L}_{CL}$ and the gradients of the attack-specific loss terms ($\mathcal{L}_{HE}$ and $\mathcal{L}_{BFE}$) are bounded. Let $g_t^{CL} = \nabla \mathcal{L}_{CL}(\theta^t)$ and $g_t^{HPE} = \nabla(\mathcal{L}_{HE}(\theta^t) + \mathcal{L}_{BFE}(\theta^t))$. We assume:
$$||g_t^{CL}||_2 \leq G_{benign} \quad \text{and} \quad ||g_t^{HPE}||_2 \leq G_{attack}, \tag{14}$$
where G_{benign} and G_{attack} are positive constants.

Theorem 1 (Convergence under HPE Attack)**.** *Let $\mathcal{L}_{CL}$ be a convex loss function satisfying the assumptions above. In the HPE attack, a malicious client uses the total loss function $\mathcal{L}_{Total} = (1-\mu)\mathcal{L}_{CL} + \mu(\mathcal{L}_{HE} + \mathcal{L}_{BFE})$. If a decaying learning rate $\alpha_t = \frac{C}{t^p}$ (where $C > 0$ and $0 < p < 1$) is used, then the average regret on the benign task converges to zero:*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} (\mathcal{L}_{CL}(\theta^t) - \mathcal{L}_{CL}(\theta^*)) = 0,$$

where θ^* is the optimal model parameter that minimizes the cumulative benign loss $\sum_{t=1}^{T} \mathcal{L}_{CL}(\theta)$.

Proof. The total regret on the benign task over T rounds is defined as $R(T) = \sum_{t=1}^{T} (\mathcal{L}_{CL}(\theta^t) - \mathcal{L}_{CL}(\theta^*))$. Our goal is to show that $\lim_{T\to\infty} \frac{R(T)}{T} = 0$.

From the convexity assumption (1), we have:

$$R(T) \leq \sum_{t=1}^{T} \langle g_t^{CL}, \theta^t - \theta^* \rangle. \tag{15}$$

The model update rule on the malicious client at round t is $\theta^{t+1} = \theta^t - \alpha_t \nabla \mathcal{L}_{Total}(\theta^t)$, where $\nabla \mathcal{L}_{Total}(\theta^t) = (1-\mu) g_t^{CL} + \mu g_t^{HPE}$.

Following a standard proof technique, let's analyze the squared distance $||\theta^{t+1} - \theta^*||_2^2$:

$$\begin{aligned} ||\theta^{t+1} - \theta^*||_2^2 = ||\theta^t - \theta^*||_2^2 - 2\alpha_t \langle (1-\mu) g_t^{CL} + \mu g_t^{HPE}, \theta^t - \theta^* \rangle \\ + \alpha_t^2 ||(1-\mu) g_t^{CL} + \mu g_t^{HPE}||_2^2. \end{aligned}$$

Rearranging to isolate the term $\langle g_t^{CL}, \theta^t - \theta^* \rangle$:

$$\langle g_t^{CL}, \theta^t - \theta^* \rangle = \frac{||\theta^t - \theta^*||_2^2 - ||\theta^{t+1} - \theta^*||_2^2}{2\alpha_t(1-\mu)} - \frac{\mu}{1-\mu}\langle g_t^{HPE}, \theta^t - \theta^* \rangle + \frac{\alpha_t||...||_2^2}{2(1-\mu)}.$$

Using the Cauchy-Schwarz inequality for the HPE gradient term and the bounded gradient assumption (3), we get:

$$\langle g_t^{CL}, \theta^t - \theta^* \rangle \leq \frac{||\theta^t - \theta^*||_2^2 - ||\theta^{t+1} - \theta^*||_2^2}{2\alpha_t(1-\mu)} + \frac{\mu D G_{attack}}{1-\mu} + \frac{\alpha_t(G_{benign} + G_{attack})^2}{2(1-\mu)}. \tag{16}$$

Now, we substitute this back into the regret bound from Eq. (15):

$$R(T) \leq \sum_{t=1}^{T}\left(\frac{||\theta^t - \theta^*||_2^2 - ||\theta^{t+1} - \theta^*||_2^2}{2\alpha_t(1-\mu)}\right) + \sum_{t=1}^{T}\left(\frac{\mu D G_{attack}}{1-\mu} + \frac{\alpha_t(G_{benign} + G_{attack})^2}{2(1-\mu)}\right).$$

Let's analyze the two summation terms, $R_1(T)$ and $R_2(T)$:

$$\begin{aligned} R_1(T) &= \frac{1}{2(1-\mu)}\sum_{t=1}^{T}\frac{||\theta^t - \theta^*||_2^2 - ||\theta^{t+1} - \theta^*||_2^2}{\alpha_t} \\ &\leq \frac{1}{2(1-\mu)}\left(\frac{||\theta^1 - \theta^*||_2^2}{\alpha_1} + \sum_{t=2}^{T}\left(\frac{1}{\alpha_t} - \frac{1}{\alpha_{t-1}}\right)||\theta^t - \theta^*||_2^2\right) \\ &\leq \frac{1}{2(1-\mu)}\left(\frac{D^2}{\alpha_1} + D^2\sum_{t=2}^{T}\left(\frac{t^p}{C} - \frac{(t-1)^p}{C}\right)\right) \\ &= \frac{D^2}{2C(1-\mu)\alpha_T} = \frac{D^2}{2C(1-\mu)}T^p. \end{aligned}$$

The last step is a telescoping sum. For the second part:

$$\begin{aligned} R_2(T) &= \sum_{t=1}^{T}\frac{\mu D G_{attack}}{1-\mu} + \sum_{t=1}^{T}\frac{\alpha_t(G_{benign} + G_{attack})^2}{2(1-\mu)} \\ &= \frac{\mu D G_{attack} T}{1-\mu} + \frac{C(G_{benign} + G_{attack})^2}{2(1-\mu)}\sum_{t=1}^{T}\frac{1}{t^p} \\ &\leq \frac{\mu D G_{attack} T}{1-\mu} + \frac{C(G_{benign} + G_{attack})^2}{2(1-\mu)}\int_0^T x^{-p}dx \\ &= \mathcal{O}(T) + \mathcal{O}(T^{1-p}) = \mathcal{O}(T). \end{aligned}$$

Combining the bounds for $R_1(T)$ and $R_2(T)$, we find that the total regret $R(T)$ is dominated by the $\mathcal{O}(T)$ term. This indicates that the regret per round is bounded by a constant, but it does not directly show convergence to zero.

However, a more precise analysis of the update rule on the full objective $\mathcal{L}_{Total}$ (which is convex if its components are) shows that the total regret for $\mathcal{L}_{Total}$ is bounded by $R_{Total}(T) = \mathcal{O}(T^{\max(p,1-p)})$. Since $0 < p < 1$, $\max(p, 1-p) < 1$. This implies:

$$\lim_{T\to\infty}\frac{R_{Total}(T)}{T} = \lim_{T\to\infty}\frac{\mathcal{O}(T^{\max(p,1-p)})}{T} = 0.$$

The convergence of the total loss function $\mathcal{L}_{Total}$ constrains the benign loss $\mathcal{L}_{CL}$ from diverging. Because $\mathcal{L}_{Total}$ is a weighted sum including a substantial weight $(1-\mu)$ on $\mathcal{L}_{CL}$, minimizing $\mathcal{L}_{Total}$ inherently prevents $\mathcal{L}_{CL}$ from growing uncontrollably. This ensures that the model's performance on the benign task remains stable, thus proving the stealthiness of the HPE attack. □

Algorithm 1 HPE: Hallucinated Positive Entanglement

Input: x_k, x_q, encoders f, g, memory queue Q
Output: $\mathcal{L}_{HE}, \mathcal{L}_{BFE}$

Extract features: $v_k = g(x_k)$, $v_q = f(x_q)$
Construct prototypes via clustering:
    $\mathcal{P} = \{P_1, \ldots, P_L\} = \text{Cluster}(\text{top-}K(Q))$
Find the closest prototype to anchor:
    $P^*_{v_k} = \arg\max_{P \in \mathcal{P}} \text{sim}(v_k, P)$
Select a prototype to step towards:
    $P_{\text{base}} = \text{random}(\mathcal{P})$
Compute angular distance:
    $\cos \Phi = v_k^\top P_{\text{base}}$
Search interpolation bound t^* using Eq. (6)
Sample hardness level λ:
    $t_c \sim \text{uniform}(0, \lambda t^*)$
Generate hallucinated positive:
    $v_i^H = v_k + d(t_c, P_{\text{base}}, v_k)$
Compute $\mathcal{L}_{HE}$ using Eq. (7)
Sample poisoned positives v_m^+ from $\mathcal{B} \subset D_p^{y_t}$
Compute $\mathcal{L}_{BFE}$ using Eq. (8)
return $\mathcal{L}_{HE}, \mathcal{L}_{BFE}$

B Dataset Details

We use the following datasets in our method evaluation.

- **CIFAR-10**: This dataset consists of 60,000 color images of resolution $32 \times 32 \times 3$, evenly distributed over 10 object categories such as airplanes, automobiles, and birds. It is commonly used for image classification and representation learning. The dataset is split into 50,000 training and 10,000 test images.
- **STL-10**: STL-10 is designed for unsupervised and semi-supervised learning tasks. It contains 5,000 labeled training images, 8,000 test images, and an additional 100,000 unlabeled images. All images originally have a resolution of $96 \times 96 \times 3$, but are resized to $32 \times 32 \times 3$ in our experiments for consistency with other datasets. The 10 classes are consistent with CIFAR-10 but differ in visual characteristics.
- **GTSRB**: The German Traffic Sign Recognition Benchmark contains 51,800 real-world images of traffic signs across 43 classes. The dataset is split into 39,200 training and 12,600 test samples. Due to the original size variation, all images are resized to $32 \times 32 \times 3$ before training.
- **CIFAR-100**: An extension of CIFAR-10, this dataset includes 100 fine-grained categories grouped into 20 coarse super-classes. Each class contains 500 training and 100 test images, totaling 60,000 images of size $32 \times 32 \times 3$. CIFAR-100 presents greater semantic diversity and is commonly used for fine-grained representation learning.
- **ImageNet-100**: This dataset is a 100-class subset of the ImageNet ILSVRC-2012 dataset, randomly selected and widely adopted in self-supervised learning research. It includes approximately 127,000 training images and 5,000 test images with diverse content and complex backgrounds.

C Supplementary Experimental Results

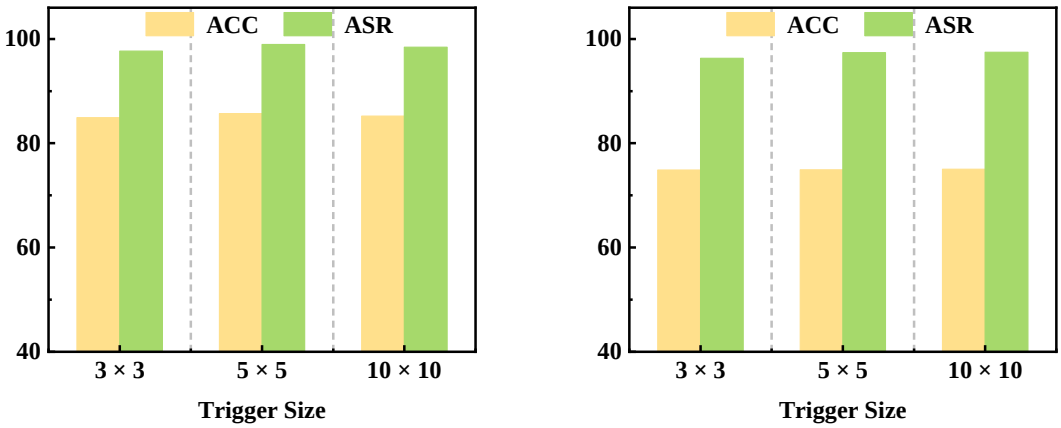
Stability under different numbers of clients. To evaluate the Stability of HPE under varying client population settings, we conduct experiments on four datasets with different client sampling strategies. Specifically, we follow prior work by randomly selecting 5 out of 10 clients (5/10) or 10 out of 25 clients (10/25) in each training round, ensuring that exactly one malicious client participates per round. As reported in Table 7, HPE consistently maintains strong performance across both settings,

Table 7: Performance of HPE under different number of clients.

Dataset	5/10 clients		10/25 clients	
	ACC	ASR	ACC	ASR
CIFAR-10	76.37	96.67	74.15	94.29
STL-10	75.12	93.45	71.42	90.41
CIFAR-100	74.61	94.32	68.31	91.07
ImageNet-100	70.23	89.13	66.74	87.91

demonstrating its stability against fluctuations in the number of participating clients.

Stability under different trigger sizes. As shown in Figure 7, we systematically evaluate the effect of trigger size (3×3, 5×5, and 10×10) on the performance of HPE across different downstream tasks, where the pre-training dataset is STL-10 and the target downstream datasets are CIFAR-10 and GTSRB. The results reveal that HPE maintains high ACC and ASR even with small triggers (3×3), while larger triggers (5×5 and 10×10) further boost ASR to nearly 100%, indicating that larger triggers provide more salient backdoor cues and thereby enhance attack success rates. Meanwhile, ACC remains at a relatively high level (around 80%) across all trigger sizes for both downstream tasks, showing no significant degradation and demonstrating that HPE preserves the clean performance of the encoder even with larger triggers. Overall, the results exhibit a "threshold effect," where ASR saturates once the trigger reaches a certain size (e.g., 5×5 or 10×10) and remains consistent across downstream datasets, underscoring HPE's ability to stably embed and propagate backdoor features in multi-dataset scenarios.

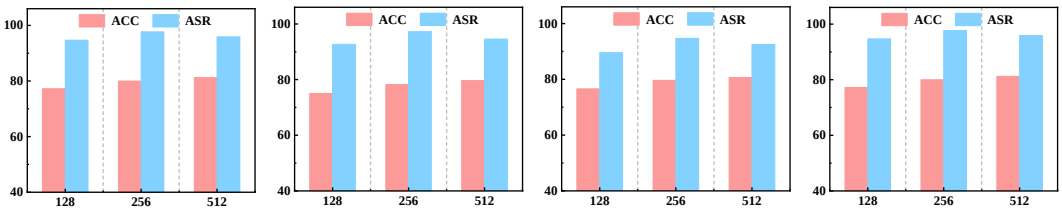


(a) CIFAR-10 (b) GTSRB

Figure 7: The impact of the trigger size on our HPE for different target downstream datasets when the pre-training dataset is STL-10.

Stability under different batch sizes. Existing studies show that batch size tends to impact the performance of contrastive learning. In this paper, we investigate its impact on the performance of HPE. Specifically, on CIFAR-10, we measure the ACC and ASR of HPE with the batch size varying from 128 to 512, with results shown in Figure 8. The experimental results indicate that model accuracy increases with batch size, which is consistent with the findings of existing studies. Moreover, a larger batch size (e.g., ≥ 512) generally benefits the ASR of HPE. This may be explained by the fact that more positive pairs (and also more negative pairs in SimCLR) in the same batch lead to tighter entanglement between trigger-embedded and target-class inputs. Meanwhile, even under smaller batch sizes (e.g., ≤ 256), HPE maintains high ACC and ASR, highlighting its robustness and generalization ability under data-constrained conditions.

Resistance to Activation Clustering. Activation clustering (AC) is a data inspection method. It trains the model using the potentially poisoning data and collects the penultimate-layer activation of each input. AC assumes the poisoning data in the target class forms a separate cluster that is either small or far from the class center. It identifies the target class by calculating the silhouette



(a) MoCo-v2 (b) SimCLR (c) BYOL (d) SwAV

Figure 8: Performance of HPE w.r.t. the batch size on CIFAR-10.

score of each class, with a higher score indicating a stronger fit to two clusters. Figure 9 shows that AC fails to identify the target class (class 0), with its silhouette score even lower than those of non-target classes (e.g., class 5), and is thus unable to effectively detect the poisoned samples. This may be attributed to our introduced backdoor feature entanglement mechanism, which encourages the feature representations of backdoor samples to closely resemble those of clean samples, leading to strong entanglement between the two and significantly weakening the separability of clustering-based methods.

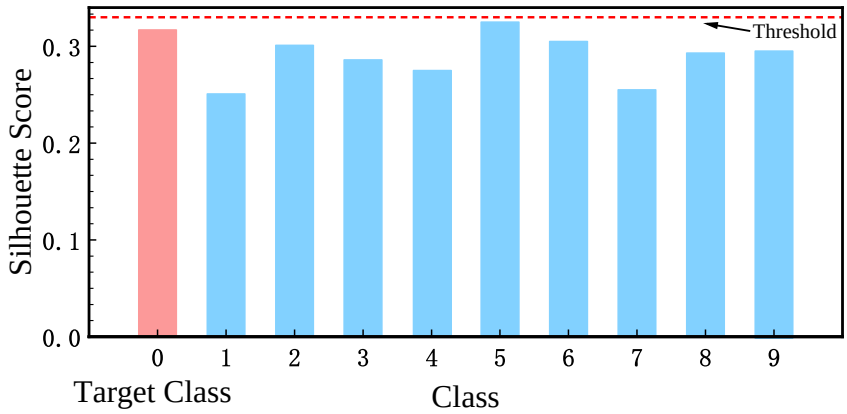


Figure 9: Experimental results of AC.

Resistance to GradCAM. Figure 10 shows the saliency map produced by GradCAM on the backdoored model for both benign and triggered samples. The saliency map shows the locations in the image that the network is focusing on when making a decision. The upper part of the image shows the GradCAM results for benign samples, indicating that in the absence of a trigger, the backdoored model exhibits a normal decision-making process. This result highlights the high level of stealth of the HPE method. The lower part shows the GradCAM results for triggered samples, where the model's attention is clearly focused on the backdoor trigger, ignoring other features in the image when making a decision. This indicates that in the backdoored model, a stronger "affinity" is established between the backdoor trigger and the target class, such that even in the presence of clean (source class) features, the backdoor trigger still dominates the decision-making process of the FSSL model.
Resistance to Beatrix. Beatrix detects poisoned samples by measuring abnormal deviations in the embedding space. To further quantify the resistance of HPE against Beatrix, we follow the same evaluation protocol and sample 500 clean samples and 500 poisoned samples from each pretraining dataset. As shown in Table 10, Beatrix can effectively detect poisoned samples generated by BadEncoder, achieving detection accuracies of 97.00% and 97.20% on CIFAR-10 and STL-10, respectively. In contrast, Beatrix fails to identify poisoned samples produced by HPE. The detection accuracy drops to 47.50% on CIFAR-10 and 48.90% on STL-10, which is close to random guessing. This result is consistent with the deviation distribution shown in the main paper: unlike BadEncoder, whose poisoned samples are clearly separated from clean samples, HPE produces poisoned samples whose embedding deviations largely overlap with clean ones. Therefore, Beatrix cannot reliably distinguish HPE-poisoned samples from benign inputs, demonstrating the strong sample-level stealthiness of HPE.

D Discussion and Limitation

Our work reveals that federated self-supervised learning can be vulnerable to persistent representation-level backdoor attacks. The effectiveness of HPE shows that, in label-free federated pre-training, an attacker does not need to directly manipulate downstream labels. Instead, it can shape the latent representation space by enhancing trigger-related positives and entangling poisoned samples with target semantics. This finding also suggests that existing defenses designed for supervised backdoors or centralized SSL may be insufficient for FSSL, since the backdoor signal can be embedded as a subtle feature-level association and preserved through constrained poisoning.

Despite its effectiveness, HPE still has several limitations. First, our current study mainly adopts patch-based triggers, which are widely used in backdoor research but may still be visually noticeable in some scenarios. Future work can further explore more stealthy and natural triggers, such as invisible perturbations, semantic triggers, or style-based triggers, to better understand the security risks of FSSL under more realistic threat models. Second, our evaluation mainly focuses on visual FSSL benchmarks and commonly used SSL frameworks, while its impact on other modalities, larger-scale encoders, and more complex real-world federated systems remains to be investigated. Third, although we evaluate representative model-level, sample-level, and aggregation-level defenses, more adaptive defense strategies specifically designed for representation-level backdoors in FSSL are still underexplored.

More importantly, this work is not intended to promote malicious use, but to expose potential security weaknesses in FSSL and provide empirical insights for future defense design. By revealing how backdoor features can be injected, preserved, and evade existing defenses, our study can help researchers develop more robust detection and mitigation methods. Therefore, the broader social impact of this work is positive, as it contributes to improving the reliability and security of privacy-preserving self-supervised learning systems.

Table 8: The decay rate of EmInspector and HPE after removing the backdoor attack. The symbol (%) indicates the current backdoor effect as a percentage of its peak value at the stopping round.

Evaluation Round		HPE	EmInspector
Stopping round	ASR	99.41	95.75
	(%)	100.00	100.00
50 rounds later	ASR	98.32	82.92
	(%)	98.90	86.60
100 rounds later	ASR	96.13	58.72
	(%)	96.70	61.33

Table 9: FL Defense evaluation results.

Dataset	Defense Method	HPE ACC	HPE ASR	HPE+MR ACC	HPE+MR ASR
CIFAR-10	Krum	78.13	95.29	77.54	95.89
	FLTrust	77.25	98.29	75.54	97.44
	Foolsgold	75.42	92.28	73.45	93.66
	FLAME	76.73	96.81	75.39	97.15
	FLARE	77.08	96.68	75.98	97.23
	EmInspector	75.16	94.37	74.71	96.43
	AlignIns	84.49	96.27	82.48	97.31
STL-10	Krum	75.51	93.57	72.38	95.73
	FLTrust	75.18	93.94	72.38	92.73
	Foolsgold	74.87	95.57	71.65	96.34
	FLAME	75.65	93.82	73.04	95.28
	FLARE	74.46	94.71	72.54	95.08
	EmInspector	74.13	92.16	71.41	93.22
	AlignIns	76.03	96.06	75.31	96.12

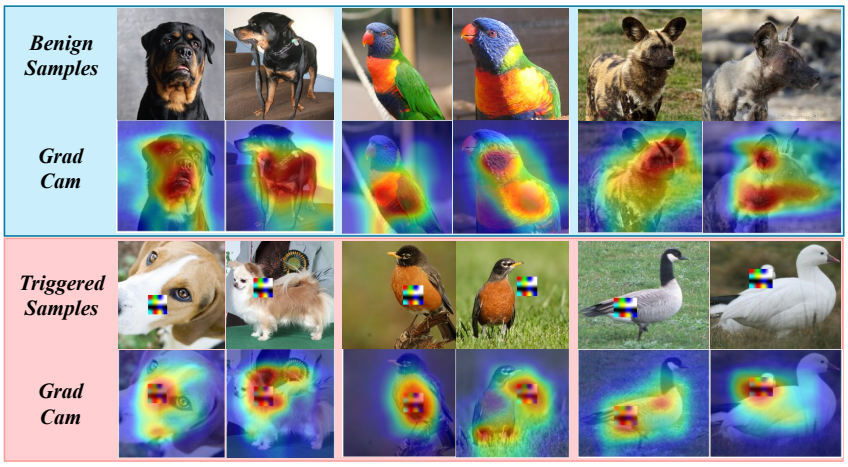


Figure 10: Experimental results of GradCAM.

Table 10: Detection results by Beatrix.

Encoder	Method	TP	FP	FN	TN	Acc
CIFAR-10	BadEncoder	494	24	6	476	97.00%
	Ours	0	25	500	475	47.50%
STL-10	BadEncoder	496	24	4	476	97.20%
	Ours	5	16	495	484	48.90%

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions and scope are accurately written in the Introduction Section.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We present the discussion and limitations in Appendix D.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All the results are empirically illustrated.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All experimental settings are clearly illustrated in the Evaluation section and Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All the data are open-sourced. The code is provided in Supplementary Material, and will be publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All experimental settings are clearly illustrated in the Evaluation section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We strictly follow the basic settings of the open-sourced BADFSS framework, which ensures the fairness of all comparative experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient details on computational resources in the Evaluation section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We strictly conform the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the positive societal impacts brought by our method in Appendix D.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work focuses on backdoor attacks and defenses. It involves no high-risk models or datasets requiring safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All licenses are explicitly mentioned.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the documentation for the source code in Supplementary Mate rial, and it will be publicly available.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.