FLARE: Towards Universal Dataset Purification against Backdoor Attacks

Linshan Hou, Wei Luo, Zhongyun Hua, Songhua Chen, Leo Yu Zhang, and Yiming Li

***Abstract*—Deep neural networks (DNNs) are susceptible to backdoor attacks, where adversaries poison datasets with adversary-specified triggers to implant hidden backdoors, enabling malicious manipulation of model predictions. Dataset purification serves as a proactive defense by removing malicious training samples to prevent backdoor injection at its source. We first reveal that the current advanced purification methods rely on a latent assumption that the backdoor connections between triggers and target labels in backdoor attacks are simpler to learn than the benign features. We demonstrate that this assumption, however, does not always hold, especially in all-to-all (A2A) and untargeted (UT) attacks. As a result, purification methods that analyze the separation between the poisoned and benign samples in the input-output space or the final hidden layer space are less effective. We observe that this separability is not confined to a single layer but varies across different hidden layers. Motivated by this understanding, we propose FLARE, a universal purification method to counter various backdoor attacks. FLARE aggregates abnormal activations from all hidden layers to construct representations for clustering. To enhance separation, FLARE develops an adaptive subspace selection algorithm to isolate the optimal space for dividing an entire dataset into two clusters. FLARE assesses the stability of each cluster and identifies the cluster with higher stability as poisoned. Extensive evaluations on benchmark datasets demonstrate the effectiveness of FLARE against 22 representative backdoor attacks, including all-to-one (A2O), all-to-all (A2A), and untargeted (UT) attacks, and its robustness to adaptive attacks.**

***Index Terms*—Dataset Purification, Backdoor Defense, Backdoor Learning, Trustworthy ML, Responsible AI**

I. Introduction

Deep neural networks (DNNs) are widely deployed in mission-critical applications, including autonomous driving [9], [17], and face recognition [25], [26]. Currently, due to the complicated structure and large parameter scale of modern DNNs, training these models usually relies on large-scale datasets, typically from external sources (*e.g.*, data markets and crowd-sourcing platforms).

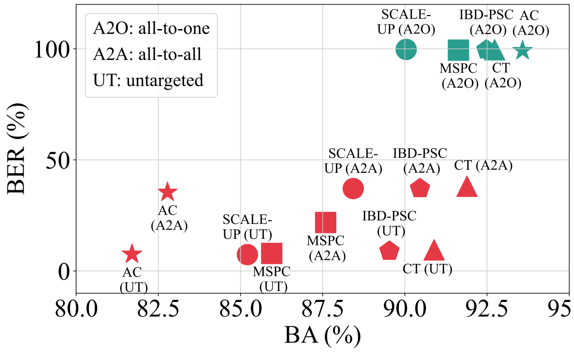


Fig. 1: The benign accuracy (BA) and backdoor elimination rate (BER) of models trained on purified dataset. In these cases, BER is calculated as $100\%-$ ASR, where ASR (attack success rate) measures the ratio of poisoned samples misclassified to the target labels by the backdoored model.

However, recent studies reveal that using such external datasets introduces security risks, especially (poison-only) backdoor attacks [4], [8], [10], [22], [46]. Specifically, the backdoor adversaries poison a small subset of training data by embedding a predefined trigger (*e.g.*, a subtle white square) and re-assigning their labels as the adversary-specified target label(s). As a result, all DNNs trained on these poisoned samples will learn a hidden backdoor, *i.e.*, the latent and malicious connection between the triggers and the target labels. During the inference process of the backdoored model, the adversaries can activate its backdoor by implanting trigger patterns to maliciously change the predictions of any testing samples to the target labels. This attack is highly stealthy since the attacked model behaves normally on benign testing samples so that users cannot easily detect it simply based on the results of their local validation samples.

Currently, researchers have investigated five representative defensive strategies against backdoor attacks. These strategies include: **(1)** dataset purification [14], [38], [51], **(2)** poison suppression [7], [16], [42], **(3)** model-level backdoor detection [44], [45], [48], **(4)** input-level backdoor detection [11], [14], [36], and **(5)** backdoor mitigation [27], [49], [52]. Among these strategies, poison suppression modifies the training process to reduce the impact of poisoned samples; model-level backdoor detection and mitigation identify or remove embedded backdoors in post-deployment; and input-level detection prevents backdoor activation by identifying malicious inputs during inference. In contrast, dataset purification acts as a proactive defense, identifying and removing poisoned samples from the dataset before training begins. This paper focuses on dataset purification, aiming to prevent backdoor creation at its

Linshan Hou is with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, Guangdong 518055, China (email: lizzieandland@gmail.com).

Wei Luo is with the School of Information Technology, Deakin University, Australia. (email: wei.luo@deakin.edu.au).

Zhongyun Hua is with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, Guangdong 518055, China, and also with the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies, Shenzhen, Guangdong 518055, China. (e-mail: huazyum@gmail.com).

Songhua Chen. Independent Researcher. (email: frederichen01@gmail.com)

Leo Yu Zhang is with the School of Information and Communication Technology, Griffith University, Southport, Gold Coast, QLD 4215, Australia (email: leo.zhang@griffith.edu.au).

Yiming Li is with College of Computing and Data Science, Nanyang Technological University, Singapore 639798. (email: liyiming.tech@gmail.com)

source and precisely trace malicious samples to their origins.

In this paper, we first revisit existing advanced dataset purification methods. We reveal that their effectiveness relies on an implicit assumption: the connections between triggers and target labels in backdoor attacks are inherently simpler to learn than the benign features. This assumption enables these methods to achieve promising results by focusing primarily on input-output relationships. Specifically, **(1)** one line of work [19], [53] suggests that DNNs converge significantly faster on poisoned samples, indicating that the backdoor connections are easier to learn; **(2)** studies such as [5], [15] observe that poisoned samples often exhibit highly localized and small saliency regions. This suggests that backdoor triggers function as shortcut features, allowing the model to bypass learning complex, distributed features in benign data and thus simplifying the backdoor establishment; **(3)** research from [11], [14] leverages the scaled prediction consistency of poisoned samples, implying that DNNs overfit on distinctive, artificially crafted triggers, thereby simplifying backdoor formation. This assumption generally holds for typical all-to-one (A2O) backdoor attacks, as backdoor triggers are often simple and easy to learn. However, this assumption does not always hold for modern complicated backdoor attacks, especially for all-to-all (A2A) and untargeted (UT) backdoor attacks (as shown in Figure 1). As such, an intriguing question arises:

Shall we design a universal dataset purification method that is effective against various types of backdoor attacks?

The answer is in the positive! Inspired by the finding that DNN memorization is distributed across neurons in multiple hidden layers [31], we explore the distinctions between poisoned and benign samples throughout the model instead of simply the input-output relationships. Our analysis of A2A and UT backdoor attacks reveals a critical observation: poisoned and benign samples do not consistently separate within specific layers; instead, distinctions emerge across different hidden layers, varying with the attack types. Building on this insight, we propose a universal method for dataset purification, termed **F**ull-spectrum **L**earning **A**nalysis for **R**emoving **E**mbedded poisoned samples (FLARE). FLARE comprises two main stages: latent representation extraction and poisoned sample detection. In the first stage, FLARE constructs a comprehensive latent representation for each training sample by consolidating the abnormal values from all hidden layers' feature maps. Specifically, FLARE first aligns all feature maps to a uniform scale (*e.g.*, [0,1]) by leveraging the statistics of Batch Normalization (BN) layers. FLARE then extracts an abnormally large or small value from each feature map and consolidates these values across all hidden layers to construct the latent representation. In the second stage, FLARE detects poisoned samples through cluster analysis. In general, FLARE splits the entire dataset into two distinct clusters and identifies the cluster with higher *cluster stability* as poisoned. Specifically, FLARE first applies dimensionality reduction to reduce the computation consumption and improve clustering efficiency. FLARE then selects a stable subspace by adaptively excluding category-specific features from the last few hidden layers, isolating an optimal subspace where benign samples from various classes are still close together. FLARE evaluates *cluster stability* as the 'density' difference between the density level at which the cluster first appears and that where the cluster divides into smaller sub-clusters. FLARE finally identifies the cluster exhibiting higher stability as poisoned since poisoned samples tend to form compact clusters due to the sharing of the same trigger-related features.

In conclusion, our main contributions are three-fold. **(1)** We reveal that the underlying assumption of existing advanced dataset purification methods, *i.e.*, the backdoor connections are easier to learn than benign ones, does not always hold, particularly under all-to-all and untargeted attacks. We also demonstrate that poisoned and benign samples do not consistently separate within particular layers across various types of attacks. **(2)** Based on our intriguing findings, we develop a universal dataset purification method (dubbed FLARE). It separates poisoned and benign samples throughout the model instead of simply the input-output relationship. **(3)** We conduct extensive experiments on benchmark datasets, verifying the effectiveness of our FLARE against 22 representative backdoor attacks (including A2O, A2A, and UT ones) and its resistance to potential adaptive attacks.

II. Related Work

A. Poison-only Backdoor Attacks

Targeted Attacks. Gu *et al.* [10] proposed the first poison-only backdoor attack, BadNets, where an adversary embeds an adversary-specified trigger into a few training samples and modifies their labels. This attack supports two primary modes: **(1)** all-to-one (A2O), where all poisoned samples are assigned a single target label, and **(2)** all-to-all (A2A), where poisoned samples from a class i are relabeled as a different class label, typically the next consecutive class (*i.e.*, $i + 1$). Models trained on the poisoned dataset establish a backdoor connection between the trigger and target labels. Subsequent research aimed to make backdoor attacks more stealthy by developing invisible triggers [35] and employing label-consistent poisoning [43], where only samples from the target class are poisoned, thus avoiding label modification and bypassing manual inspection. Follow-up research further introduced more sophisticated trigger designs, including sample-specific triggers [24], sparse triggers [8], horizontal triggers [29], and asymmetric triggers [37], all specifically crafted to evade defenses.

Untargeted Attacks. Targeted attacks assign poisoned samples to adversary-specific labels, ensuring that the backdoored model consistently misclassifies the poisoned samples into particular classes. Recently, some pioneering work has discussed untargeted attacks [21], [50], where the goal is to make the predictions deviate from the true labels instead of approaching particular ones (*i.e.*, target labels). For example, poisoned samples are reassigned with random incorrect labels [21], causing the backdoored model to misclassify the poisoned samples into incorrect classes. As a result, the predictions for all poisoned samples approximate a uniform distribution, making the attack more difficult to learn and detect.

B. Backdoor Defenses

Based on the stage at which they occur, existing defenses can be divided into five main categories: **(1)** dataset purification [14], [38], [51], which focuses on detecting and removing poisoned samples from a given suspicious dataset before model training, **(2)** poison suppression [7], [16], [42], which modifies the training process to limit the impact of poisoned samples, **(3)** model-level backdoor detection [44], [45], [48], which assesses whether a suspicious model contains hidden backdoors; **(4)** input-level backdoor detection [11], [14], [36], which identifies malicious inputs at inference; **(5)** backdoor mitigation [27], [49], [52], which directly removes backdoors after model development. This paper primarily focuses on dataset purification since it reduces backdoor threats from the source and can be easily used to mitigate model backdoors. We hereby provide an overview of the defences related to dataset purification and backdoor mitigation.

Dataset Purification. Existing strategies can be divided into four types: **(1)** purification via latent separability, **(2)** purification via early convergence, **(3)** purification via dominant trigger effects, **(4)** purification via perturbation consistency.

Specifically, latent separability-based defenses exploited the detectable traces left by poisoned samples in the feature space. For example, Chen *et al.* [3] observed that, in the feature space of the final hidden layer, samples from the target class form two distinct clusters, with the smaller cluster identified as poisoned. Ma *et al.* [30] utilized high-order statistics (*i.e.*, Gram matrix) to analyze the differences between poisoned and benign samples; Early convergence-based defenses relied on the observation that DNNs converge on poisoned samples more rapidly than on benign ones. During the early stages of training, the losses for poisoned samples quickly drop to near zero, while those for benign samples remain relatively high. For example, researchers in [19], [53] traped samples whose losses decreased more rapidly by using the local gradient ascent technique during the initial five training epochs. To mitigate class imbalance issues during this process, Gao *et al.* [7] refined the approach by selecting the lowest-loss samples within each class, rather than across the entire dataset; Dominant trigger-based defenses assume that backdoor triggers play a dominant role in DNN predictions. A backdoored model tends to learn an excessively strong signal for the backdoor trigger, such that even small, localized triggers can overpower other semantic features and dictate the model's prediction. For example, Chou *et al.* [5] utilized model interpretability techniques (*e.g.*, Grad-CAM [40]) to visualize salient regions of an input image, identifying highly localized and small regions as potential trigger areas. Similarly, Huang *et al.* [15] distilled minimal patterns from input images that influenced the model's predictions and identified images with abnormally small patterns as poisoned; Perturbation consistency-based defenses assumed that poisoned samples are resistant to perturbations. For instance, Guo *et al.* [11] observed that poisoned samples exhibited prediction consistency under pixel-level amplification and proposed analyzing this consistency to distinguish poisoned samples. To address constraints in pixel values and insensitivity to amplification, Pal *et al.* [36] optimized a mask for selective pixel amplification; Qi *et al.* [38] analyzed the prediction consistency by unlearning the benign connections of the backdoored models; and Hou *et al.* [14] examined prediction consistency under unbounded weight-level amplification.

However, in this paper, we find that all existing methods suffer from poor performance in some cases. Specifically, the first type of purification method generally only utilizes information at a specific layer (*e.g.*, the last hidden layer) and can be easily bypassed by advanced attacks [37]. We will also show that the last three types of methods, however, relied on an underlying assumption that does not always hold, especially under A2A and UT attacks. How to design an effective dataset purification method is still an important open question.

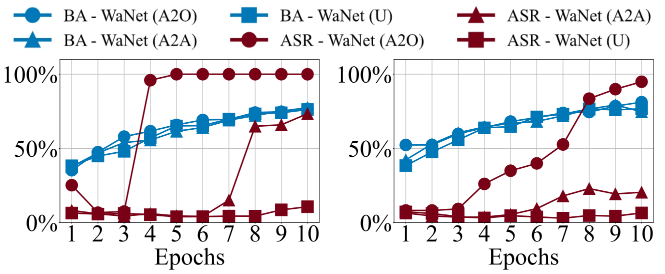
Backdoor Mitigation. This defense occurs in the post-development phase, aiming to remove implanted backdoors from attacked models. For instance, Li [27] pruned dormant neurons with benign inputs; Wu [47] pruned neurons that are sensitive to adversarial perturbations; Chai [2] applied weight-level pruning, which is more precise than neuron-level pruning, thereby preserving benign task performance; Li [20] adopted knowledge distillation to guide the fine-tuning of backdoored models. Instead of direct removal, researchers in [44], [49], [52] reverse-engineered suspected triggers and decouple them from target labels through unlearning and fine-tuning.

Existing backdoor mitigation defenses have shown effectiveness against different attacks; however, Zhu *et al.* [54] demonstrated that the injected backdoors can persist and reactivate during inference, even after backdoor mitigation. This underscores the urgent need for a dataset purification method capable of defending against a broader range of attacks, including A2O, A2A, and UT attacks, to prevent backdoor creation at its source.

III. Revisiting Existing Dataset Purification Methods

A. Preliminaries

The Main Pipeline of Poison-only Backdoor Attack. Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ denotes a training set composed of N *i.i.d.* samples. Each sample $(\boldsymbol{x}, y)$ is characterized by $\boldsymbol{x} \in \mathcal{X} = [0,1]^{d_C \times d_W \times d_H}$ and $y \in \mathcal{Y} = \{1, 2, \ldots, K\}$. An adversary can generate a poisoned dataset $\hat{\mathcal{D}}$ by modifying a subset of benign samples (*i.e.*, $\mathcal{D}_s$), *i.e.*, $\hat{\mathcal{D}} = \mathcal{D}_p \cup \mathcal{D}_b$, where $\mathcal{D}_b = \mathcal{D} - \mathcal{D}_s \subset \mathcal{D}$, $\mathcal{D}_p = \{(\hat{\boldsymbol{x}}, \hat{y}) \mid \hat{\boldsymbol{x}} = \mathcal{G}_X(\boldsymbol{x})), \hat{y} = \mathcal{G}_Y(y), (\boldsymbol{x}, y) \in \mathcal{D}_s\}$, and $\rho = |\mathcal{D}_s|/|\hat{\mathcal{D}}|$ is the poisoning rate. $\mathcal{G}_X : \mathcal{X} \to \mathcal{X}$, $\mathcal{G}_Y : \mathcal{Y} \to \mathcal{Y}$ are adversary-specified poisoned image generator and target label generator, respectively. For instance, in the BadNets attack [10], $\mathcal{G}_X(x) = (\boldsymbol{1} - \boldsymbol{m}) \odot \boldsymbol{x} + \boldsymbol{m} \odot \boldsymbol{t}$, where $\boldsymbol{m} \in \{0,1\}^{d_C \times d_W \times d_H}, \boldsymbol{t} \in \mathcal{X}$ is the malicious trigger. As for the $\mathcal{G}_Y(y)$, there are two primary attack paradigms: **(1)** targeted attacks and **(2)** untargeted attacks. Specifically, for the targeted attacks, poisoned images are re-labeled to the designated target labels: $\mathcal{G}_Y(y) = y_t, y_t \in \mathcal{Y}$ in A2O attacks, and $\mathcal{G}_Y(y) = (y+1) \pmod{K}$ in A2A attacks. For the untargeted attacks, each poisoned image is re-labeled to a random label uniformly sampled from $\mathcal{Y}$ (*i.e.*, $G_Y(y) \sim U(1, K)$).



(a) BadNets (b) WaNet

Fig. 2: The benign accuracy (BA) and attack success rate (ASR) during the initial ten epochs of model training.

Threat Model of Dataset Purification. Similar to existing studies [3], [36], we assume that adversaries can freely poison the training dataset but can not manipulate the training process. Defenders aim to identify and filter out poisoned samples from a given third-party dataset. This is achieved through two main goals: effectiveness and generalizability. **Effectiveness** ensures that all poisoned samples are detected while minimizing the incorrect removal of benign samples. **Generalizability** guarantees that the defense is effective against various attack types, including A2O, A2A, and UT attacks. We assume that defenders have access to the poisoned training dataset and full control over the training process, but they have no knowledge of the backdoor attacks. In particular, our method requires less capacity for defenders compared to classical methods [14], [38] since it does not require additional benign local samples.

B. Revisiting Strategy 1: Early Convergence

In general, the purification methods in [16], [19], [53] are designed based on the observation that DNNs converge faster on poisoned samples during the early stages of training, suggesting that the backdoor connections are easier to learn.

Settings. To validate the effectiveness of this purification strategy across different attack types, we conduct two representative backdoor attacks, *i.e.*, BadNets [10] and WaNet [35], on the CIFAR-10 dataset using ResNet-18. They are the representatives of sample-agnostic and sample-specific attacks. Each attack includes three variants: A2O, A2A, and UT. For each attack, we calculate the benign accuracy (BA) and the attack success rate (ASR) during the initial ten epochs.

Results. As shown in Figure 2, the ASR for the BadNets (A2O) and WaNet (A2O) attacks rapidly approaches 100%. However, the ASRs for the other attacks remain lower than the BAs, with UT attacks even nearing 0%. These results suggest that in A2A and UT attacks, DNNs do not converge quickly on poisoned samples, indicating that the assumption that backdoor connections are simpler to learn than the benign ones does not hold for A2A and UT attacks.

C. Revisiting Strategy 2: Dominant Trigger Effects

Researchers in [5], [15] observed that poisoned samples often exhibit highly localized and small saliency regions, suggesting that trigger-related features can be regarded as 'short-cut' that are more easily learned by DNNs compared to benign features.

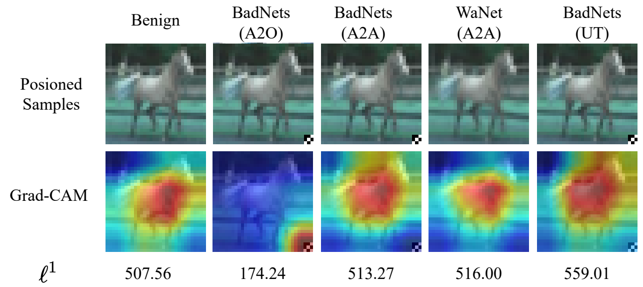


Fig. 3: Grad-CAM visualization of saliency regions for benign and poisoned samples. The ℓ^1 norm values of these regions are presented to quantify the amount of information the model relies upon for predictions. Larger ℓ^1 norm values indicate that the model's predictions depend on more input features.

Settings. We use BadNets and WaNet for our discussions, following the same experimental settings as described in Section III-B, with WaNet results limited to the A2A setting for brevity. We compare saliency regions from two perspectives: visualization and quantitative measurement. Specifically, we utilize Grad-CAM [40] to visualize the saliency regions of both benign and poisoned samples. We then calculate the ℓ^1 norms of these regions to quantify their size.

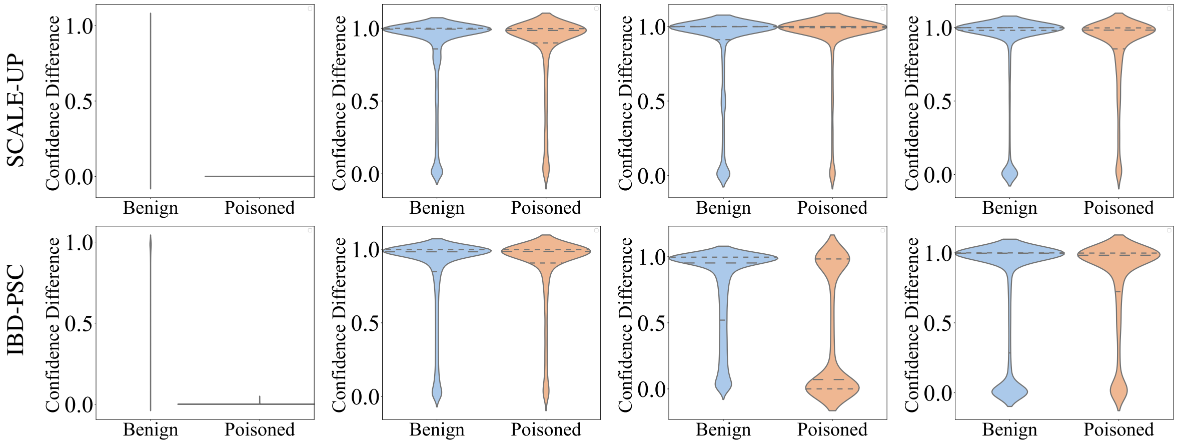
Results. As shown in Figure 3, the saliency region of poisoned samples under the BadNets (A2O) attack is concentrated around the trigger, with its ℓ^1 norm substantially smaller than that of the corresponding benign sample. However, the saliency regions for other attacks shift from the trigger area to benign regions, accompanied by high ℓ^1 norms exceeding those of benign samples. These results suggest that, under A2A and UT attacks, the predictions of poisoned samples rely on distributed features, instead of simply trigger-related features. Consequently, the assumption that backdoor connections are inherently simpler does not hold for A2A and UT attacks.

D. Revisiting Strategy 3: Perturbation Consistency

Researchers also observed that poisoned samples exhibit greater prediction consistency than those of benign samples under pixel-level amplification [11], [36] or weight-level alterations [14], [38], [51]. This implies that DNNs tend to overfit on triggers instead of learning semantic features.

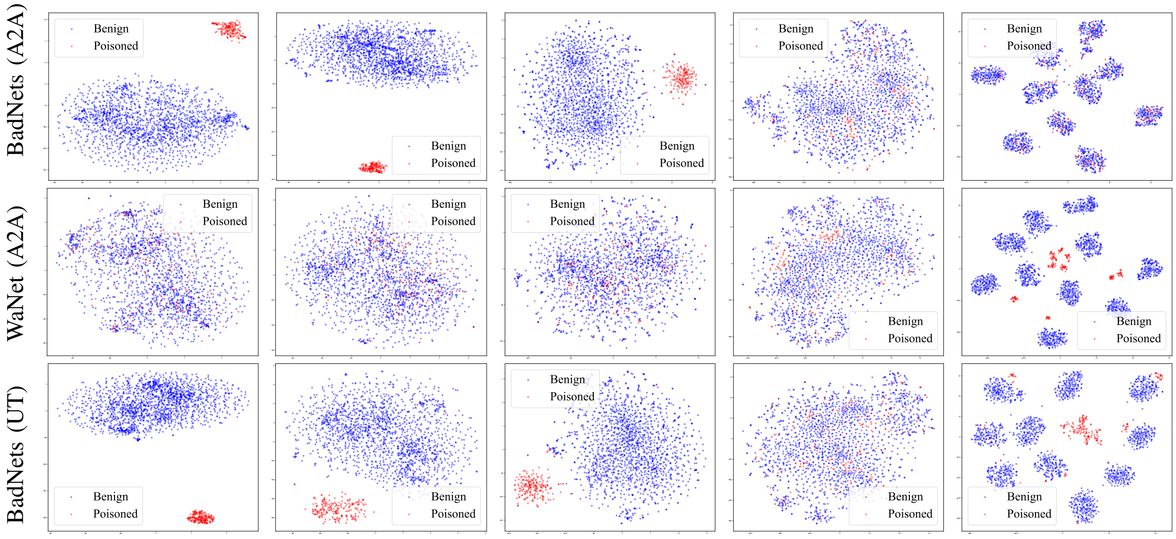
Settings. We hereby also use BadNets and WaNet on CIFAR-10 for our discussions. To evaluate the prediction consistency, we calculate the difference in prediction confidence between the original and perturbed predictions on the initially predicted label for both benign and poisoned samples. Other settings are the same as those used in Section III-C.

Results. As shown in Figure 4, for BadNets (A2O), the differences are consistently close to zero under the pixel-level and weight-level perturbations. It indicates that the predictions of these poisoned samples are unaffected by such perturbations. In contrast, under A2A and UT attacks, the confidence differences of poisoned samples approach 1, indicating that the perturbations significantly change the predictions of poisoned samples. The sensitivity of poisoned samples to perturbations, similar to that of benign samples, indicates that DNNs struggle to overfit to poisoned samples in A2A and UT attacks. Thus, the assumption that backdoor connections are easier to learn than benign ones does not hold for A2A and UT attacks.



(a) BadNets (A2O) (b) BadNets (A2A) (c) WaNet (A2A) (d) BadNets (UT)

Fig. 4: Difference in prediction confidences for benign and poisoned samples on CIFAR-10 under input-level and weight-level perturbations, calculated by comparing prediction confidences on the original label before and after perturbation.



(a) Layer 2 (b) Layer 6 (c) Layer 10 (d) Layer 14 (e) Layer 18

Fig. 5: T-SNE visualization of latent representations across different hidden layers on the CIFAR-10 dataset. Each point corresponds to a training sample with poisoned samples marked in red and benign samples in blue.

E. *Revisiting Latent Separability on a Particular Layer*

Previous studies [3], [30] explored the latent separability between benign and poisoned samples in the feature space. In particular, these works primarily focused on the final hidden layer as a representative intermediate layer. We argue that their success also partly relied on the assumption that backdoor can be easily learned. In this section, we verify it.

Settings. We also conduct experiments using BadNets and WaNet on the CIFAR-10 dataset to facilitate our analysis. To assess latent separability, we employ t-SNE visualization to examine the latent representations of both benign and poisoned samples across various hidden layers, with particular focus on shallow, middle, and deep layers. All other experimental settings remain consistent with those outlined in Section III-C.

Results. As shown in Figure 5, poisoned and benign samples do not consistently display separability in specific layers. For example, under the BadNets (A2A) and BadNets (UT) attacks, separability is evident in shallow and middle layers, such as Layer 6 or Layer 10, but diminishes in deeper layers for BadNets (A2A). In contrast, under WaNet (A2A), separability is not evident across most layers. These observations challenge the implicit assumption in existing latent-separability-based purification methods that backdoor triggers are sparsely embedded and primarily affect deeper feature representations, highlighting the necessity of evaluating the broader impact of poisoned samples across multiple layers throughout the model rather than limiting the analysis to a single layer.

IV. The Proposed Method

A. *Overview*

Motivated by previous findings, we introduce FLARE, a dataset purification method that leverages hidden features across all hidden layers. As illustrated in Figure 6, FLARE consists of two main stages: **(1) Latent Representation Extraction:** For each sample, FLARE constructs a comprehensive latent representation by consolidating the abnormal values from all hidden layers' feature maps. Specifically, FLARE first aligns the values of all feature maps to a uniform scale using the statistics of Batch Normalization (BN) layer. Then, for

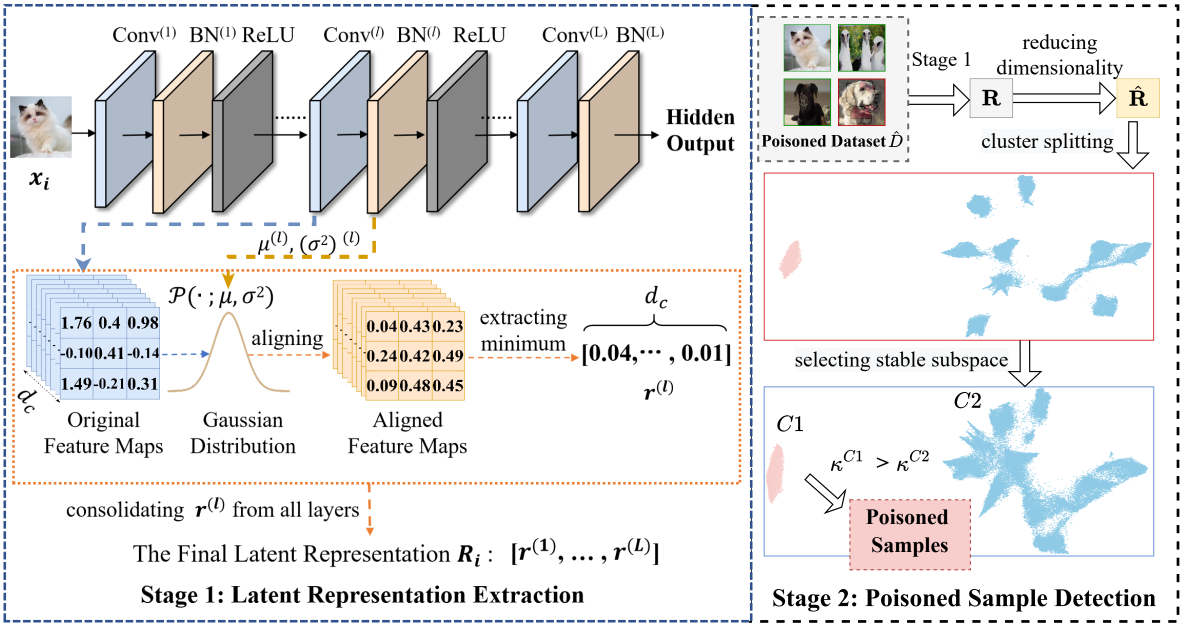


Fig. 6: The main pipeline of FLARE. **Stage 1: Latent Representation Extraction**: A backdoored model is trained on a poisoned dataset, and each training sample x_i is forwarded to generate feature maps at hidden layers. The value ranges of each feature map are aligned using the statistics of corresponding BN layers. An abnormal value is then extracted from each aligned feature map, and these values are aggregated across all hidden layers to form the final representation $\mathcal{R}_i$. **Stage 2: poisoned samples Detection**: With the representations $\mathcal{R}$ of all training samples, UMAP reduces $\mathcal{R}$ to $\hat{\mathcal{R}}$, followed by density-based clustering to separate samples into two clusters. The cluster with a higher κ^C is identified as poisoned.

each training sample, FLARE extracts abnormally large or small values from each aligned feature map, and aggregates the abnormal values across all hidden layers to construct the final latent representations. **(2) Poisoned Sample Detection:** FLARE detects poisoned samples through cluster analysis. Specifically, FLARE reduces the dimensionality of each final latent representation and then selects a stable subspace by adaptively excluding representations from the last few hidden layers. FLARE performs cluster analysis in the optimal subspace to split the entire dataset into two clusters and identifies the cluster with higher cluster stability as poisoned.

B. Latent Representation Extraction

For each training sample, FLARE forms a comprehensive latent representation by leveraging all hidden features of DNNs. This task encounters two main challenges: **(C1)** hidden features capture diverse characteristics, leading to significant variability in their values; **(C2)** hidden features are often high-dimensional and noisy. To tackle these challenges, FLARE employs a two-step process: feature alignment and extracting abnormal features. In the alignment step, FLARE normalizes the feature maps to a uniform scale, based on the statistics of BN layers. Subsequently, FLARE extracts an abnormal value from each feature map as its representative and aggregates them across all hidden layers to form the final latent representation. Their technical details are as follows.

1) Feature Alignment: In this work, we adopt batch normalization (BN) to align all feature maps to a uniform scale. It is mostly because BN transformation was initially proposed to mitigate internal covariate shifts, providing a principled way to stabilize feature distributions. Specifically, for a backdoored DNN model $\mathcal{F}$ consisting of L hidden layers, *i.e.*,

$$\mathcal{F} = \mathrm{FC} \circ f^{(L)} \circ f^{(L-1)} \circ \cdots f^{(l)} \cdots \circ f^{(2)} \circ f^{(1)}, \tag{1}$$

where $f^{(l)}$ denotes the l-th hidden layer, consisting of a convolutional layer, a BN layer, and an activation function. Let $\mathbf{a}$ represent the output of a convolutional layer, *i.e.*, $\boldsymbol{a} \in \mathbb{R}^{d_c \times d_h \times d_w}$, where d_c is the number of feature maps, and d_h and d_w are the height and width of each feature map, respectively. Utilizing the mean $\boldsymbol{\mu}$ and variance σ of the following BN layer, we define a transformation $\mathcal{P}(\cdot;\boldsymbol{\mu},\sigma^2)$ to transform all values in $\boldsymbol{a}$ to obtain the aligned output $\hat{\boldsymbol{a}}$ via:

$$\hat{\boldsymbol{a}} = \mathcal{P}(\boldsymbol{a};\boldsymbol{\mu},\sigma^2) = \frac{1}{\sqrt{2\pi(\boldsymbol{\sigma}^2)}} \exp\left(-\frac{\boldsymbol{a}-\boldsymbol{\mu}}{2\boldsymbol{\sigma}^2}\right). \tag{2}$$

In this way, all values in $\hat{\boldsymbol{a}}$ are from a uniform scale $[0, 1]$.

2) Extracting Abnormal Features: As backdoor-related features usually function as dominant features, they tend to induce abnormally large or small activations in the feature map, as partly supported by findings in [1]. Building on this understanding, we propose to focus on identifying outliers within each feature map, specifically targeting the abnormally small or large values. In particular, all feature maps are normalized and their values adhere to a uniform distribution defined by the statistics of the BN layers after the previous alignment step. At this time, both abnormally large and small values fall outside the central distribution, corresponding to regions with low occurrence probability. This allows us to consistently extract the minimum values from each aligned feature map as indicators of backdoor-related features, without needing to separately select the largest or smallest values. Specifically, we can extract the minimum value (*i.e.*, $r_c^{(l)}$) from the aligned output $\hat{a}^{(l)}$ of the l-th hidden layer via:

$$r_c^{(l)} = \min_{\substack{1 \leq i \leq d_w \\ 1 \leq j \leq d_h}} \left(\hat{\boldsymbol{a}}_{c,i,j}^{(l)} \right), \tag{3}$$

where $\hat{\boldsymbol{a}}^{(l)} \in \mathbb{R}^{d_c \times d_w \times d_h}$ contains d_c feature maps, and $\hat{\boldsymbol{a}}_c^{(l)}$ denotes the c-th feature map of the l-th hidden layer, with i and j indexing the width d_w and height d_h, respectively. The minimums from all d_c feature maps are then consolidated to form the l-th layer representations, denoted as $\boldsymbol{r}^{(l)}$:

$$\boldsymbol{r}^{(l)} = \left[r_1^{(l)}, r_2^{(l)}, \ldots, r_{d_c}^{(l)} \right]. \tag{4}$$

For a sample $\boldsymbol{x}_i$, the minimal values of all hidden layers are consolidated to form the final latent representation $\mathbf{R}_i$:

$$\mathbf{R}_i = [\boldsymbol{r}^{(1)}, \boldsymbol{r}^{(2)}, \ldots, \boldsymbol{r}^{(L)}], \tag{5}$$

where L is the number of hidden layers.

C. Poisoned Sample Detection

After obtaining the final representations of all training samples, FLARE applies dimensionality reduction to reduce the computation consumption and improve clustering efficiency. FLARE then conducts cluster analysis to determine which cluster may contain poisoned samples. Arguably, the most straightforward method is to perform clustering on the entire representation directly. However, benign samples from different classes tend to form multiple clusters due to category-specific features, leading to misidentification. To refine this partitioning, FLARE employs a stable subspace selection algorithm to adaptively exclude representations from the last few hidden layers. This approach allows FLARE to isolate an optimal subspace where benign samples are close together rather than dispersing into multiple small clusters.

1) Cluster Splitting: Let $\mathbf{R} \in \mathbb{R}^{N \times d}$ represent the latent representations of all N training samples, with each sample in a d-dimensional latent space. To enhance clustering efficiency and reduce computational demands, FLARE first performs a dimensionality reduction transformation to obtain:

$$\hat{\mathbf{R}} = \mathcal{T}(\mathbf{R}), \quad \hat{\mathbf{R}} \in \mathbb{R}^{n \times d'}, \quad d' \ll d. \tag{6}$$

Here, $\mathcal{T} : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ denotes the dimensionality reduction function. In this paper, we choose the uniform manifold approximation and projection (UMAP) algorithm [33] as $\mathcal{T}$ for its ability to reduce high-dimensional data while preserving the topological structure of the original data manifold. FLARE then performs clustering on the reduced representation $\hat{\mathbf{R}}$: $\mathcal{C} = \mathcal{H}(\hat{\mathbf{R}})$, where $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ represents the set of clusters identified by a clustering algorithm $\mathcal{H}$. In this paper, we use the hierarchical density-based spatial clustering of applications with noise (HDBSCAN) algorithm [32] as $\mathcal{H}$ for its ability to handle clusters of varying shapes and densities. HDBSCAN constructs a hierarchy of clusters represented by a condensed tree$\mathbf{T}$, where clusters are partitioned across varying density levels $\lambda = 1/d_{core}$, where d_{core} denotes the distance of an object to its 'minPts-nearest' neighbor.

2) Stable Subspace Selection: Given that poisoned samples typically share the same trigger-related features, they tend to aggregate into a more stable cluster. In contrast, benign samples, which originate from a natural and diverse distribution, generally form clusters with lower stability. Therefore, cluster stability forms an effective indicator for identifying poisoned samples. Traditionally, in HDBSCAN, cluster stability is calculated by aggregating the density level λ of each sample within a cluster. However, in poisoned sample detection, where the number of poisoned samples is much smaller than that of benign samples, this approach introduces bias toward the majority class (benign samples). To alleviate this problem, we redefine *cluster stability* to focus on the persistence of a cluster across varying density levels (λ), as follows.

Definition IV.1 (Cluster Stability in Detecting Poisoned Samples)**.** *For a cluster C, let $\lambda_s^C = \min_{x \in C} \lambda_x$ be the density level where C first appears, and $\lambda_e^C = \max_{x \in C} \lambda_x$ be the density level before C divides into sub-clusters. The stability κ^C of cluster C is defined as $\kappa^C = \lambda_e^C - \lambda_s^C$.*

While benign samples generally form a single cluster, they often tend to form multiple stable clusters, as illustrated in Figure 6. These small but stable benign clusters can significantly interfere with detection performance. To address this issue, FLARE develops a subspace selection strategy to isolate a stable space in which poisoned and benign samples form two distinct clusters. This approach is based on the understanding that models tend to capture semantic information in shallow layers and focus on distinguishing features in deeper layers. Accordingly, FLARE excludes features from the last few hidden layers, obtaining $\mathbf{R}'$ as the final representation that includes only the earlier layers: $\mathbf{R}' = [\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \ldots, \mathbf{r}^{(L-k)}]$.

To determine the optimal number of layers k to exclude, FLARE designs an adaptive algorithm that dynamically selects a suitable k. The algorithm begins with a model configured with all L hidden layers and progressively removes the last hidden layer of the modified model at each step. At each iteration, FLARE splits the condensed tree $\mathbf{T}$ at the root node to form two primary clusters and assesses the stability $\kappa^{C_{large}}$ of the larger cluster C_{large}. If the stability $\kappa^{C_{large}}$ surpasses a predefined threshold ξ, it suggests that the benign samples are cohesively grouped within the current subspace. To further ensure the cluster stability against potential anomalies, FALRE generates a condensed tree at each step and traverses from the root node of C_{large} down to depth d. If the stability at any depth exceeds ξ, the current subspace is considered stable.

3) Stability-based Detection: Within this obtained subspace, FLARE splits the condensed tree $\mathbf{T}$ of the entire poisoned dataset at the root node to form two primary clusters, C_1 and C_2. FLARE then evaluates the *cluster stability* κ^C of each cluster and identifies the cluster with the highest stability as poisoned. The set of poisoned samples $\mathcal{S}_p$ is:

$$\mathcal{S}_p = \{x \mid x \in C_p\}, \tag{7}$$

where

$$C_p = \arg \max_{\{C_1, C_2\}} (\kappa^{C_1}, \kappa^{C_2}). \tag{8}$$

D. Post-Detection Strategy 1: Secure Training from Scratch

After completing the above detection process, we can remove the detected poisoned samples, denoted as $\hat{\mathcal{D}}_p$, from the training dataset. The remaining samples form a purified dataset, $\hat{\mathcal{D}}_b \triangleq \mathcal{D} - \hat{\mathcal{D}}_p$, which is assumed to contain only benign samples. Defenders can then train a backdoor-free model $\mathcal{M}(\cdot, \theta')$ on $\hat{\mathcal{D}}_b$ using standard training procedures, *i.e.*,

$$\min_{\theta} \sum_{(\boldsymbol{x},y)\in\hat{\mathcal{D}}_b} \mathcal{L}(\mathcal{M}(\boldsymbol{x};\theta'), y), \tag{9}$$

where $\mathcal{L}(\cdot)$ is the cross-entropy loss function.

E. Post-Detection Strategy 2: Backdoor Removal

We can utilize the detected poisoned samples to directly remove backdoors from the model. This process is achieved through a two-step method of unlearning and relearning, ensuring the mitigation of backdoors while preserving the model's performance on benign tasks.

Step 1: Unlearning. This step aims to eliminate the effect of the trigger by unlearning the identified poisoned samples. Specifically, we aim to maximize the cross-entropy loss for these poisoned samples with respect to their malicious labels. To achieve this, we minimize the negative loss of the backdoored model $\mathcal{F}(\cdot;\theta)$ over the detected poisoned samples $\hat{\mathcal{D}}_p$:

$$\min_{\theta} \frac{1}{|\hat{\mathcal{D}}_p|} \sum_{(\boldsymbol{x},y)\in\hat{\mathcal{D}}_p} -\mathcal{L}(\mathcal{F}(\boldsymbol{x};\theta), y). \tag{10}$$

Step 2: Relearning. The unlearning step can effectively mitigate backdoor effects, but it also may result in degraded performance on benign samples. To alleviate this, we perform a relearning step that utilizes the remaining benign samples $\hat{\mathcal{D}}_b \triangleq \mathcal{D} - \hat{\mathcal{D}}_p$. This step fine-tunes the model as follows:

$$\min_{\theta} \frac{1}{|\hat{\mathcal{D}}_b|} \sum_{(\boldsymbol{x},y)\in\hat{\mathcal{D}}_b} \mathcal{L}(\mathcal{F}(\boldsymbol{x};\theta), y), \tag{11}$$

Both unlearning and relearning steps are performed in each epoch to optimize the model's performance on benign data without reintroducing susceptibility to backdoor triggers.

V. Experiments

A. Main Settings

Datasets and Models. We conduct experiments on two classical image classification benchmark datasets, including CIFAR-10 [18] and Tiny-ImageNet [6]. For the primary experiments, we employ ResNet-18 [13] as the default architecture. For ablation studies, we use VGG19 [41] and MobileNetV2 [39].

Attacks Configurations. To validate the generalizability of our defense, we evaluate it against four representative types of backdoor attacks across three attack modes. These attacks are categorized as follows: **(1)** sample-agnostic attacks: BadNet [10], Blend [4], Trojan [28]; **(2)** sample-specific attacks: IAD [34], WaNet [35], ISSBA [24]; **(3)** clean-label attack: LC [43]; **(4)** sparse attack: SIBA [8]. The attack modes include the all-to-one (A2O), all-to-all (A2A), and untargeted (UT) paradigms. To comprehensively assess each attack mode, attacks are adapted accordingly. For example, BadNets is evaluated in three formats: BadNets (A2O), BadNets (A2A), and BadNets (UT). These attacks are implemented using the open-source `BackdoorBox` toolkit [23], strictly following the settings outlined in their papers. For all attacks, we set the target label y_t to 0. Given the complexity of the Tiny-ImageNet dataset, we focus on the most representative attacks, omitting some similar-effect attacks to avoid excessive computational costs. To ensure consistently high attack success rates, we set the poisoning rate to 0.1 and the cover rate to 0.2.

Baseline Defenses. To evaluate the detection performance, we compare our FLARE method with five state-of-the-art (SOTA) detection methods, including AC [3], SCALE-UP [11], MSPC [36], IBD-PSC [14], and CT [38]. Following the detection, we further compare our defense performance on the purified dataset using FLARE, dubbed "FLARE (P)", with these five methods. For backdoor mitigation, we compare our backdoor removal defense via FLARE, dubbed "FLARE (R)",with six SOTA methods, including FP [27], NAD [20], AMW [2], ABL [19], SEAM [55], and BTI-DBF [49]. All defenses are implemented using the official source codes with the default settings.

Evaluation Metrics. We measure the detection performance using true positive rate (TPR) and false positive rate (FPR). TPR reflects the proportion of correctly identified poisoned samples, while FPR represents the proportion of benign samples incorrectly classified as poisoned. Higher TPR and lower FPR indicate more effective detection. In addition to the detection metrics, we also assess benign accuracy (BA) and attack success rate (ASR) after purification. Higher BA and lower ASR indicate that the model's primary functionality is preserved, and backdoor effects are effectively neutralized.

B. Detection Performance

We mainly assess the effectiveness of FLARE against various backdoor attacks on the CIFAR-10 and Tiny-ImageNet datasets. The results, detailed in Tables I and II, demonstrate that FLARE consistently achieves promising performance across different attack scenarios, with the TPRs reaching 100% (or near 100%) while maintaining FPRs near 0%. The results demonstrate a significant improvement over baseline defenses, which fail in A2A and UT attack scenarios (highlighted in red). This failure is primarily due to their implicit assumptions that backdoor correlations are easier to learn than benign ones, which does not hold in more complex attack configurations. Additionally, FLARE maintains an FPR close to zero on datasets composed entirely of benign samples, indicating that it has minimal false positives.

C. Effectiveness of Secure Training from Scratch

To assess the effectiveness of FLARE (P), which retrains models on a purified dataset with detected poisoned samples removed, we evaluated the BAs and ASRs of the retrained models on the CIFAR-10 and Tiny-ImageNet datasets. As shown in Tables III-IV, the ASRs of retrained models approach

TABLE I: The detection performance (%) of FLARE on the CIFAR-10 dataset using the ResNet-18 model. The best results (highest TPRs and lowest FPRs) are in bold, while TPRs $<$ 80% and FPRs $>$ 20% are marked in red as fail cases.

Attack Mode↓	Defenses→	AC		SCALE-UP		MSPC		IBD-PSC		CT		FLARE	
	Attacks↓	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
—	No Poison	—	7.31	—	31.44	—	0.45	—	1.31	—	7.90	—	**0.01**
A2O	BadNets	99.54	0.00	**100.00**	19.44	99.06	13.14	**100.00**	5.65	**100.00**	6.74	**100.00**	**0.00**
	Blend	0.00	6.08	98.54	25.76	**100.00**	10.05	**100.00**	3.77	**100.00**	7.28	**100.00**	**0.02**
	Trojan	99.62	0.09	**100.00**	19.35	**100.00**	17.89	**100.00**	7.99	**100.00**	0.23	**100.00**	**0.00**
	IAD	99.46	**0.00**	**100.00**	15.24	98.00	11.83	**100.00**	9.59	**100.00**	0.30	**100.00**	**0.00**
	WaNet	90.36	**0.00**	26.28	30.68	55.12	18.59	**99.98**	8.39	98.38	2.40	98.14	**0.00**
	ISSBA	99.94	**0.00**	97.94	15.83	98.02	10.32	**100.00**	2.30	**100.00**	4.73	99.98	0.01
	SIBA	99.84	0.07	92.34	18.06	**100.00**	10.05	**100.00**	9.95	**100.00**	0.55	**100.00**	**0.00**
	LC	0.00	7.40	**100.00**	31.61	**100.00**	18.47	**100.00**	0.03	0.00	**0.00**	**100.00**	**0.00**
A2A	BadNets	0.04	10.02	20.38	19.89	11.74	16.77	7.56	5.82	4.14	2.64	**100.00**	**0.00**
	Blend	0.00	10.02	4.34	28.00	12.86	16.76	12.84	3.42	14.98	3.30	**99.58**	**0.00**
	Trojan	0.00	9.96	28.60	28.83	15.41	16.06	26.16	5.23	15.52	0.48	**100.00**	**0.00**
	WaNet	0.00	9.96	4.88	25.41	13.06	16.00	13.08	0.52	11.96	9.18	**95.76**	**0.00**
	IAD	0.00	9.96	23.12	14.68	11.34	13.35	35.00	0.66	5.78	0.22	**99.98**	**0.00**
	ISSBA	0.00	9.96	33.08	18.91	12.46	14.82	13.08	0.52	12.46	14.82	**99.70**	**0.27**
	SIBA	0.00	9.96	26.08	26.07	12.10	11.30	10.42	5.30	0.36	3.75	**99.74**	**0.00**
UT	BadNets	0.66	9.94	7.60	22.99	9.42	10.46	7.12	7.29	0.06	0.01	**100.00**	**0.00**
	Blend	0.00	10.03	16.46	30.17	16.23	15.81	8.82	10.13	10.48	14.33	**99.44**	**0.03**
	Trojan	0.22	9.94	10.06	21.87	11.06	8.64	3.02	0.24	0.24	1.76	**100.00**	**0.00**
	IAD	0.04	10.02	9.68	14.45	8.64	15.01	8.26	3.70	2.80	4.82	**99.44**	**0.03**
	WaNet	0.30	9.90	6.04	26.12	8.70	13.32	8.80	3.61	6.40	8.15	**80.52**	**0.10**
	ISSBA	0.18	9.53	7.76	21.37	9.80	15.39	8.98	8.37	2.36	5.64	**98.56**	**0.01**
	SIBA	0.24	9.99	7.06	20.89	8.07	10.92	6.40	6.48	2.40	5.04	**97.94**	**0.00**

TABLE II: The detection performance (%) of FLARE on the TinyImageNet dataset using the ResNet-18 model. The best results (highest TPRs and lowest FPRs) are in bold, while TPRs $<$ 80% and FPRs $>$ 20% are marked in red as fail cases.

Attack Mode↓	Defenses→	AC		SCALE-UP		MSPC		IBD-PSC		CT		FLARE	
	Attacks↓	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
—	No Poison	—	0.49	—	4.62	—	0.61	—	0.05	—	18.65	—	2.89
A2O	BadNets	0.00	0.15	94.61	27.67	98.02	40.80	98.01	0.75	**100.00**	0.65	**100.00**	**0.00**
	Blend	46.50	**0.00**	91.68	4.94	95.94	0.59	**99.99**	6.61	94.09	13.96	99.97	**0.00**
	Trojan	0.00	**0.01**	82.83	5.89	**100.00**	0.65	82.70	1.75	**100.00**	11.57	**100.00**	**0.01**
	WaNet	62.18	**0.00**	0.00	5.25	0.03	0.45	**96.62**	0.21	79.42	3.37	93.68	**0.00**
A2A	BadNets	0.04	0.46	2.58	3.81	0.49	0.62	0.15	0.05	0.71	10.23	**100.00**	**0.00**
	Blend	0.00	0.50	0.25	3.26	0.20	0.10	0.03	0.26	0.00	1.20	**99.57**	**0.00**
	Trojan	0.00	0.49	1.39	3.85	0.40	0.42	0.06	0.29	0.00	1.20	**99.99**	**0.01**
UT	BadNets	0.45	0.35	3.94	4.47	0.61	0.63	0.00	**0.00**	0.58	27.54	**100.00**	**0.00**
	Blend	0.54	**0.00**	1.70	2.74	0.00	0.40	0.00	0.21	0.04	0.30	**100.00**	0.01
	Trojan	9.36	**0.00**	3.90	4.60	0.20	0.10	0.00	0.18	0.00	1.00	**100.00**	**0.00**

to nearly 0% in almost all cases. In contrast, all baseline methods fail in most cases, especially under all-to-all and untargeted attacks. In particular, we also observe some outliers with a high ASR, *e.g.*, models under Blend (UT) and SIBA (UT) on CIFAR-10. This is mainly because even adding triggers of these attacks to a backdoor-free benign model may result in some ASR, instead of due to the failure of our method.

D. Effectiveness of Backdoor Removal

To assess the effectiveness of FLARE (R), which directly removes backdoors by unlearning the detected poisoned samples, we also evaluated the BAs and ASRs of the purified models on the CIFAR-10 and Tiny-ImageNet datasets. As shown in Table V and Table VI, FLARE (R) significantly reduces the ASRs of different attacks to nearly 0%, while the impact on BA remains minimal, generally around 1%. Compared to existing advanced defenses, FLARE (R) consistently proves effective across various attacks and datasets.

E. Ablation Study

We conduct ablation studies to assess the impact of four specific factors on the effectiveness of FLARE: **(1)** target label for A2O attacks, **(2)** model architectures, **3)** the module for selecting a stable subspace, **4)** the hyper-parameters of the lambda threshold ξ and maximum depth d.

Impact of Target Labels. In our main experimetns, the target label for all A2O attacks is initially set to 0. To further assess the effectiveness of FLARE against variations in the target label, we test five representative A2O attacks, including patch-based, clean-label, and sample-specific backdoor attacks, targeting each of the ten labels in the CIFAR-10 dataset. The TPRs and FPRs are displayed in Figure 7. As shown, FLARE consistently performs well across different attacks and target labels, with TPRs generally at 100% and FPRs near 0%. The results demonstrate that our defense is effective against a range of A2O backdoor attacks and target labels.

Impact of Model Architectures. In our main results, the

TABLE III: Effectiveness of secure training from scratch (FLARE (P)) on the "purified" CIFAR-10 dataset. The best results (highest BAs and lowest ASRs) are in bold, while BAs $<$ 80% and ASRs $>$ 20% are marked in red as fail cases.

Attack Mode↓	Defenses→	Benign Model		No Defense		AC		SCALE-UP		MSPC		IBD-PSC		CT		FLARE (P)	
	Attacks↓	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR
A2O	BadNets	94.16	0.61	92.94	100.00	93.58	0.81	90.04	**0.29**	91.63	0.61	92.48	0.46	92.73	0.43	**93.96**	0.63
	Blend	94.16	0.38	93.65	100.00	93.35	**0.00**	89.14	90.74	84.41	**0.00**	92.91	0.35	93.36	0.85	**93.41**	0.19
	Trojan	94.16	1.61	93.04	100.00	93.45	28.24	90.10	**0.45**	91.61	14.06	90.63	0.46	**93.49**	1.86	93.43	1.93
	IAD	94.16	5.26	93.80	99.99	93.44	12.87	88.01	**0.06**	84.78	96.91	88.05	0.21	93.54	5.89	**94.05**	5.86
	WaNet	94.16	0.71	92.53	97.83	92.88	0.92	89.70	96.44	90.13	94.34	90.80	**0.15**	92.86	0.95	**94.10**	0.73
	ISSBA	94.16	0.60	93.50	100.00	**93.48**	0.76	87.75	0.26	84.40	**0.19**	93.05	0.61	93.09	0.61	93.26	0.76
	SIBA	94.16	25.10	94.33	98.68	93.16	18.82	88.76	6.02	84.83	4.88	85.88	**0.04**	92.96	16.86	**94.15**	23.91
	LC	94.16	1.11	84.24	100.00	82.16	99.93	83.01	**0.00**	83.26	0.15	**84.66**	**0.00**	84.51	100.00	84.55	**0.00**
A2A	BadNets	94.16	0.48	92.74	83.40	82.78	64.56	88.43	62.91	87.59	78.19	90.46	62.85	91.89	61.86	**93.53**	**0.80**
	Blend	94.16	8.09	93.08	85.20	83.79	85.40	90.21	84.25	90.01	83.20	93.10	76.35	**93.40**	82.84	93.24	**7.25**
	Trojan	94.16	1.36	93.85	91.90	84.14	90.59	90.55	86.35	90.43	89.13	92.64	89.14	93.19	89.35	**93.26**	**1.73**
	IAD	94.16	1.69	93.48	91.01	83.98	88.65	92.53	84.09	92.50	87.34	**93.50**	86.01	93.31	88.56	93.39	**1.86**
	WaNet	94.16	0.71	92.89	87.86	83.34	85.40	82.10	82.80	90.69	82.99	84.10	82.00	92.33	85.63	**92.79**	**1.07**
	ISSBA	94.16	0.53	93.66	93.15	84.14	91.48	92.26	90.14	90.45	89.46	**93.44**	91.01	92.20	84.00	93.30	**4.27**
	SIBA	94.16	2.45	93.74	55.59	83.93	50.10	91.78	41.59	93.08	56.86	92.81	53.11	**93.40**	52.06	93.37	**3.10**
UT	BadNets	94.16	0.50	91.15	92.50	81.70	92.43	85.21	92.62	85.95	92.10	89.53	91.05	90.89	90.50	**93.51**	**0.56**
	Blend	94.16	69.55	93.41	80.82	83.36	84.32	88.69	88.09	85.39	85.70	85.59	87.59	92.15	87.17	**93.30**	**63.59**
	Trojan	94.16	5.81	91.39	91.97	83.19	88.46	89.74	87.56	82.40	88.93	92.55	87.02	92.53	88.48	**93.36**	**6.12**
	IAD	94.16	21.01	92.06	90.81	83.10	88.00	90.58	89.13	85.20	87.60	92.56	87.64	92.45	86.99	**93.40**	**19.96**
	WaNet	94.16	2.47	92.08	75.55	82.99	72.63	84.22	76.10	90.56	74.35	81.02	73.40	**91.05**	75.21	90.05	**1.82**
	ISSBA	94.16	0.24	91.94	80.65	89.35	86.96	89.83	80.77	83.11	88.62	92.14	87.28	92.47	86.26	**92.83**	**0.16**
	SIBA	94.16	51.00	92.23	90.66	83.39	88.01	90.39	91.81	84.20	90.84	92.25	89.04	92.71	91.95	**93.25**	**52.15**

TABLE IV: Effectiveness of secure training from scratch (FLARE (P)) on the "purified" Tiny-ImageNet dataset. The best results (highest BAs and lowest ASRs) are in bold, while BAs $<$ 50% and ASRs $>$ 20% are marked in red as fail cases.

Attack Mode↓	Defenses→	Benign Model		No Defense		AC		SCALE-UP		MSPC		IBD-PSC		CT		FLARE (P)	
	Attacks↓	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR
A2O	BadNets	67.75	0.01	62.10	96.52	42.00	97.25	65.17	2.57	59.76	0.41	**68.25**	0.13	66.69	1.96	67.88	**0.02**
	Blend	67.75	0.07	66.91	100.00	68.41	80.05	**70.86**	99.37	68.96	99.45	63.92	2.09	67.97	100.00	66.50	**0.15**
	Trojan	67.75	0.00	66.06	100.00	65.00	100.00	70.22	99.55	**70.75**	0.69	70.15	99.57	68.82	**0.00**	70.18	**0.00**
	WaNet	67.75	0.03	62.12	99.42	68.02	97.67	66.52	98.29	66.27	98.97	68.45	2.52	65.61	68.56	**68.79**	**0.05**
A2A	BadNets	67.75	1.15	68.01	22.93	69.35	13.84	64.37	10.73	68.33	20.00	**69.73**	13.87	68.30	19.20	65.53	**1.07**
	Blend	67.75	0.77	69.54	48.67	69.44	35.43	65.13	32.80	67.00	34.27	66.34	32.11	65.49	36.20	**70.48**	**0.73**
	Trojan	67.75	0.65	68.20	29.26	**69.47**	16.12	68.99	14.97	65.33	28.10	64.69	24.46	65.20	28.08	68.79	**0.73**
UT	BadNets	67.75	2.77	66.36	91.20	68.36	91.10	62.62	92.05	69.71	91.10	68.64	91.00	60.21	92.10	**70.74**	**2.29**
	Blend	67.75	58.38	70.01	99.26	**70.79**	98.13	70.28	98.64	66.79	98.60	68.39	99.11	68.20	98.10	69.08	**49.54**

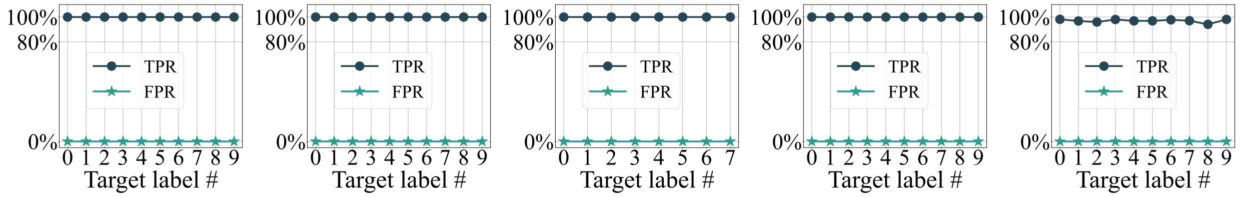
model architecture is initially set to ResNet-18. To further assess the robustness of FLARE, we also evaluate it with VGG-19 and MobileNet v2 architectures. The results, shown in Table VII, indicate that FLARE consistently achieves high TPRs, approaching 100%, while maintaining FPRs near 0%.

Impact of the Stable Subspace Selection Module. Some sophisticated attacks, such as WaNet, employ regularization samples that exhibit similar trigger patterns to actual poisoned images while maintaining correct labels. This poses a significant challenge for detection mechanisms, making it increasingly difficult to differentiate between benign and poisoned samples. To tackle this challenge, we developed a module designed to enhance the compactness of benign clusters. We assessed the effectiveness of this module by evaluating FLARE's detection performance with and without its integration. The results demonstrate that FLARE, with this module, achieves a False Positive Rate (FPR) of 0.00%, a significant improvement compared to the 20.15% FPR observed without the module. This indicates that the subspace selection module effectively improves detection performance.

Impact of the Hyper-parameters. In our search for a stable subspace, we employ two hyperparameters: the lambda threshold ξ and the maximum depth d. To assess FLARE's robustness, we vary ξ from 0.01 to 0.05 across five representative attacks. As shown in Table VIII, while higher ξ values slightly increase TPR, setting ξ between 0.01 and 0.03 consistently achieves optimal results, with TPRs at 100% and FPRs close to 0% across all attacks. We also examine d from 1 to 5, observing consistent performance across this range, with TPRs and FPRs remaining close to 100% and 0%, respectively. Given the uniform results, we omit detailed data for these variations and set ξ to 0.02 and d to 3 by default.

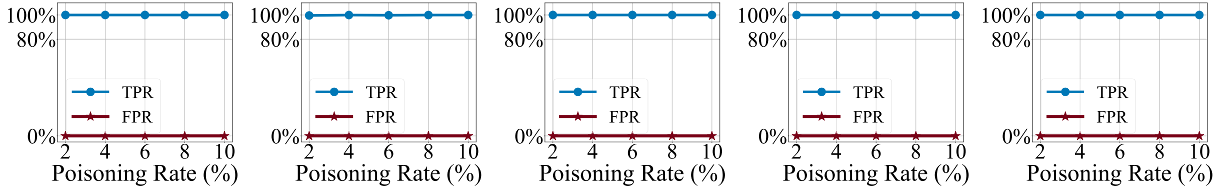
F. Resistance to Potential Adaptive Attacks

FLARE operates on the assumption that backdoored models learn distinct latent representations for poisoned and benign samples. However, studies in [12] indicate that the separations may diminish at low poisoning rates. We assess the effectiveness of FLARE against five representative attacks: BadNets (A2O), ISSBA (A2O), LC, BadNets (A2A), and BadNets (UT). These attacks are conducted on the CIFAR-10 dataset with poisoning rates (ρ) ranging from 0.02 to 0.1, ensuring



(a) BadNets (A2O) (b) Blend (A2O) (c) LC (d) ISSBA (A2O) (e) WaNet (A2O)

Fig. 7: Performance of our defense across different target labels of CIFAR-10.



(a) BadNets (A2O) (b) ISSBA (A2O) (c) LC (d) BadNets (A2A) (e) BadNets (UT)

Fig. 8: The impact of poisoning rate on CIFAR-10.

TABLE V: Comparisons of FLARE (R) with SOTA backdoor-removal defenses on CIFAR-10. The best results (highest BAs and lowest ASRs) are in bold, while BAs < 80% and ASRs > 20% are marked in red as fail cases.

Attack Mode↓	Defenses→	No Defense		FP		NAD		AMW		ABL		SEAM		BTI-DBF		FLARE (R)	
	Attacks↓	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR
A2O	BadNets	92.94	100.00	82.34	0.84	85.63	2.38	89.41	4.83	92.90	0.37	89.60	4.05	92.08	0.58	**92.96**	**0.36**
	Blend	93.65	100.00	83.56	44.11	84.83	1.29	87.38	2.20	91.27	79.29	89.78	3.85	90.48	0.60	**91.69**	**0.36**
	Trojan	93.04	100.00	83.53	3.57	86.44	4.77	87.55	2.78	**92.85**	1.60	90.05	2.46	90.08	1.89	92.20	**0.83**
	IAD	93.80	99.99	83.44	78.70	86.89	6.28	87.00	3.25	91.25	64.92	89.19	12.96	91.34	2.13	**92.30**	**1.39**
	WaNet	92.53	97.83	89.55	22.32	87.89	**1.86**	85.54	2.20	**92.41**	11.77	88.98	2.40	90.01	13.46	91.61	2.29
	ISSBA	93.50	100.00	83.38	**0.00**	86.85	7.61	87.86	1.06	**92.46**	85.07	89.20	8.44	90.63	0.85	92.15	0.58
	SIBA	94.33	98.68	78.31	90.62	87.16	15.71	88.25	11.26	90.66	19.60	87.44	22.17	89.34	24.94	**92.21**	**2.74**
	LC	84.24	100.00	83.81	**0.00**	87.08	12.06	86.56	3.86	82.97	**0.00**	88.31	7.99	**89.29**	7.03	83.78	5.25
A2A	BadNets	92.74	83.40	82.43	5.48	86.26	1.76	86.84	1.47	86.21	11.98	88.84	1.17	90.40	1.57	**93.09**	**0.15**
	Blend	93.08	85.20	82.03	9.05	86.37	6.19	88.53	4.44	90.95	12.04	87.74	**2.81**	90.46	23.34	**91.51**	6.58
	Trojan	93.85	91.90	83.18	7.84	86.59	3.30	88.24	1.90	**93.60**	12.11	88.98	1.50	91.74	9.07	93.09	**0.49**
	WaNet	93.93	89.28	82.23	8.17	84.66	1.91	86.25	1.49	**92.41**	11.77	88.13	1.47	91.08	3.10	92.27	**0.66**
	IAD	93.48	91.01	81.91	5.64	87.01	5.86	88.36	1.80	**93.00**	12.50	90.11	1.81	91.75	2.38	92.39	**0.60**
	ISSBA	93.66	93.65	79.81	5.05	79.60	2.31	88.46	1.34	92.48	12.59	88.83	0.92	**92.16**	0.80	92.09	**0.10**
	SIBA	93.74	55.59	83.64	6.43	81.09	4.66	88.31	3.99	**92.94**	11.82	88.34	7.01	90.94	5.61	92.38	**2.43**

TABLE VI: Comparisons of FLARE (R) with SOTA backdoor-removal defenses on Tiny-ImageNet (%). The best results (highest BAs and lowest ASRs) are in bold, while BAs < 50% and ASRs > 20% are marked in red as fail cases.

Attack Mode↓	Defenses→	No Defense		FP		NAD		ABL		SEAM		BTI-DBF		FLARE (R)	
	Attacks↓	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR
A2O	BadNets	54.76	96.52	37.84	95.65	46.76	0.24	41.00	**0.00**	47.23	0.10	52.57	4.15	**55.70**	0.01
	Blend	66.91	100.00	52.16	100.00	**67.38**	10.00	39.03	**0.00**	45.53	1.68	53.11	2.99	63.82	1.20
	Trojan	66.06	100.00	58.51	100.00	59.12	5.92	39.09	**0.00**	48.39	**0.00**	54.84	99.93	**65.10**	**0.00**
	WaNet	62.12	99.42	24.47	99.90	53.47	0.49	42.12	**0.00**	44.55	0.36	51.09	1.05	**61.20**	1.10
A2A	BadNets	68.01	22.93	**67.61**	0.43	61.21	4.45	44.35	21.66	58.48	0.22	58.22	8.87	67.11	**0.10**
	Blend	69.54	48.67	**69.39**	1.04	62.84	3.71	37.85	10.96	60.33	**0.24**	52.01	1.91	69.04	1.90
	Trojan	68.20	29.26	**68.12**	0.76	61.01	18.01	43.49	12.08	58.60	0.33	48.95	7.67	67.20	**0.30**

ASRs exceed 80%. The results in Figure 8 show that FLARE maintains high effectiveness across all poisoning rates, with TPRs close to 100% and FPRs near 0%.

We further evaluate the robustness of FLARE against adaptive attacks under the worst-case scenario, where adversaries deliberately reduce the latent separation. The Ada-Patch attack [37] designs regularization samples (poisoned images with ground-truth labels) to reduce separation, posing a significant challenge for our detection. We test FLARE against Ada-Patch on the CIFAR-10 dataset with a poisoning rate of 0.1, consistent with our primary experiments. FLARE achieved a TPR of 92.10% and an FPR of 1.05%, demonstrating robustness even under this advanced adaptive attack. We argue that the effectiveness mostly stems from its ability to detect distributed anomalies across all hidden layers to accumulate subtle differences, which enhances its detection capabilities.

VI. Conclusion

In this paper, we revealed a critical limitation of existing purification methods and proposed a universal approach (*i.e.*, FLARE) to filter out poisoned training samples. Existing methods assumed that backdoor connections between triggers and target labels are easier to learn. However, this assumption did not always hold, particularly in all-to-all and untargeted backdoor attacks. We observed that the latent separation between benign and poisoned samples varies across multiple layers

TABLE VII: Detection performance of FLARE on the CIFAR-10 dataset with different model architectures.

Attack Mode↓	Models→	VGG-19		MobileNet_v2	
	Attacks↓	TPR (%)	FPR (%)	TPR (%)	FPR (%)
A2O	BadNets	100.00	0.00	100.00	0.00
	Blend	100.00	0.00	99.98	0.00
	LC	100.00	0.00	100.00	0.00
	Trojan	97.08	0.00	99.98	0.00
	IAD	99.98	0.00	99.98	0.00
	WaNet	94.66	0.08	95.32	0.01
	ISSBA	99.98	0.00	99.90	0.00
A2A	BadNets	100.00	0.00	100.00	0.00
UT	BadNets	100.00	9.86	100.00	0.00

TABLE VIII: Detection performance of FLARE against various backdoor attacks under different threshold ξ.

ξ	Metrics	BadNets (A2O)	LC	ISSBA	BadNets (A2A)	BadNets (UT)
0.01	TPR (%)	100.00	100.00	99.98	100.00	100.00
	FPR (%)	0.00	0.01	0.01	0.00	0.00
0.02	TPR (%)	100.00	100.00	99.98	100.00	100.00
	FPR (%)	0.00	0.00	0.01	0.00	0.00
0.03	TPR (%)	100.00	100.00	99.98	100.00	100.00
	FPR (%)	0.00	0.00	0.01	0.00	0.00
0.04	TPR (%)	100.00	100.00	99.98	100.00	100.00
	FPR (%)	0.00	33.89	0.01	0.00	0.00
0.05	TPR (%)	100.00	100.00	99.98	100.00	100.00
	FPR (%)	0.00	33.89	0.01	40.67	0.00

rather than on a particular layer. These findings motivated us to leverage abnormal features from all hidden layers to construct comprehensive representations for cluster analysis. Besides, to further improve separation, we developed an adaptive subspace selection algorithm to dynamically isolate an optimal space for dividing an entire dataset into two clusters. We conducted 22 backdoor attacks on benchmark datasets to comprehensively verify the effectiveness of FALRE.

References

[1] Ruisi Cai, Zhenyu Zhang, Tianlong Chen, Xiaohan Chen, and Zhangyang Wang. Randomized channel shuffling: Minimal-overhead backdoor attack detection without clean datasets. In *NeurIPS*, 2022.

[2] Shuwen Chai and Jinghui Chen. One-shot neural backdoor erasing via adversarial weight masking. In *NeurIPS*, 2022.

[3] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. In *CEUR Workshop*, 2018.

[4] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. *arXiv*, 2017.

[5] Edward Chou, Florian Tramer, and Giancarlo Pellegrino. SentiNet: Detecting Localized Universal Attacks Against Deep Learning Systems. In *IEEE S&P Workshop*, 2020.

[6] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. In *arxiv*, 2017.

[7] Kuofeng Gao, Yang Bai, Jindong Gu, Yong Yang, and Shu-Tao Xia. Backdoor defense via adaptively splitting poisoned dataset. In *CVPR*, 2023.

[8] Yinghua Gao, Yiming Li, Xueluan Gong, Zhifeng Li, Shu-Tao Xia, and Qian Wang. Backdoor attack with sparse and invisible trigger. *IEEE Transactions on Information Forensics and Security*, 2024.

[9] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, 37(3):362–386, 2020.

[10] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *IEEE Access*, 2017.

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

[12] Jonathan Hayase and Weihao Kong. Spectre: Defending against backdoor attacks using robust covariance estimation. In *ICML*, 2020.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[14] Linshan Hou, Ruili Feng, Zhongyun Hua, Wei Luo, Leo Yu Zhang, and Yiming Li. IBD-PSC: Input-level backdoor detection via parameter-oriented scaling consistency. In *ICML*, 2024.

[15] Hanxun Huang, Xingjun Ma, Sarah Erfani, and James Bailey. Distilling cognitive backdoor patterns within an image. In *ICLR*, 2023.

[16] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. In *ICLR*, 2022.

[17] Zelun Kong, Junfeng Guo, Ang Li, and Cong Liu. Physgan: Generating physical-world-resilient adversarial examples for autonomous driving. In *CVPR*, 2020.

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning Multiple Layers of Features from Tiny Images. *Technical report*, 2009.

[19] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-Backdoor Learning: Training Clean Models on Poisoned Data. In *NeurIPS*, 2021.

[20] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *ICLR*, 2021.

[21] Yiming Li, Yang Bai, Yong Jiang, Yong Yang, Shu-Tao Xia, and Bo Li. Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. In *NeurIPS*, 2022.

[22] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor Learning: A Survey. *IEEE Transactions on Neural Networks and learning systems*, 2022.

[23] Yiming Li, Ya Mengxi, Bai Yang, Jiang Yong, and Xia Shu-Tao. BackdoorBox: A Python Toolbox for Backdoor Learning. In *ICLR Workshop*, 2023.

[24] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible Backdoor Attack with Sample-Specific Triggers. In *ICCV*, 2021.

[25] Zhifeng Li, Dihong Gong, Qiang Li, Dacheng Tao, and Xuelong Li. Mutual component analysis for heterogeneous face recognition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(3):1–23, 2016.

[26] Zhifeng Li, Dihong Gong, Yu Qiao, and Dacheng Tao. Common feature discriminant analysis for matching infrared face images to optical face images. *IEEE transactions on image processing*, 23(6):2436–2445, 2014.

[27] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*, 2018.

[28] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *NDSS*, 2018.

[29] Hua Ma, Shang Wang, Yansong Gao, Zhi Zhang, Huming Qiu, Minhui Xue, Alsharif Abuadbba, Anmin Fu, Surya Nepal, and Derek Abbott. Watch out! simple horizontal class backdoor can trivially evade defense. In *CCS*, 2024.

[30] Wanlun Ma, Derui Wang, Ruoxi Sun, Minhui Xue, Sheng Wen, and Yang Xiang. The "beatrix"resurrections: Robust backdoor detection via gram matrices. In *NDSS*, 2022.

[31] Pratyush Maini, Michael C Mozer, Hanie Sedghi, Zachary C Lipton, J Zico Kolter, and Chiyuan Zhang. Can neural network memorization be localized? In *ICML*, 2023.

[32] Leland McInnes, John Healy, Steve Astels, et al. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.

[33] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *Journal of Open Source Software*, 3(29):861, 2018.

[34] Tuan Anh Nguyen and Anh Tran. Input-Aware Dynamic Backdoor Attack. In *NeurIPS*, 2020.

[35] Tuan Anh Nguyen and Anh Tuan Tran. WaNet – Imperceptible Warping-based Backdoor Attack. In *ICLR*, 2021.

[36] Soumyadeep Pal, Yuguang Yao, Ren Wang, Bingquan Shen, and Sijia Liu. Backdoor secrets unveiled: Identifying backdoor data with optimized scaled prediction consistency. In *ICLR*, 2024.

[37] Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, and Prateek Mittal. Revisiting the Assumption of Latent Separability for Backdoor Defenses. In *ICLR*, 2023.

[38] Xiangyu Qi, Tinghao Xie, Jiachen T Wang, Tong Wu, Saeed Mahloujifar, and Prateek Mittal. Towards a proactive ML approach for detecting backdoor poison samples. In *USENIX Security*, 2023.

[39] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

[40] 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 *ICCV*, 2017.

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv*, 2014.

[42] Ruixiang Tang, Jiayi Yuan, Yiming Li, Zirui Liu, Rui Chen, and Xia Hu. Setting the Trap: Capturing and Defeating Backdoor Threats in PLMs through Honeypots. In *NeurIPS*, 2023.

[43] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-Consistent Backdoor Attacks. *arXiv*, 2019.

[44] 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 *IEEE S&P*, 2019.

[45] Hang Wang, Zhen Xiang, David J Miller, and George Kesidis. MM-BD: Post-Training Detection of Backdoor Attacks with Arbitrary Backdoor Pattern Types Using a Maximum Margin Statistic. In *IEEE S&P*, 2024.

[46] Cheng Wei, Yang Wang, Kuofeng Gao, Shuo Shao, Yiming Li, Zhibo Wang, and Zhan Qin. Pointncbw: Towards dataset ownership verification for point clouds via negative clean-label backdoor watermark. *IEEE Transactions on Information Forensics and Security*, 2024.

[47] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *NeurIPS*, 2021.

[48] Zhen Xiang, Zidi Xiong, and Bo Li. Umd: Unsupervised model detection for x2x backdoor attacks. In *ICML*, 2023.

[49] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, and Kui Ren. Towards reliable and efficient backdoor trigger inversion via decoupling benign features. In *ICLR*, 2024.

[50] Mingfu Xue, Yinghao Wu, Shifeng Ni, Leo Yu Zhang, Yushu Zhang, and Weiqiang Liu. Untargeted backdoor attack against deep neural networks with imperceptible trigger. *IEEE Transactions on Industrial Informatics*, 2023.

[51] Zeming Yao, Hangtao Zhang, Yicheng Guo, Xin Tian, Wei Peng, Yi Zou, Leo Yu Zhang, and Chao Chen. Reverse backdoor distillation: Towards online backdoor attack detection for deep neural network models. *IEEE Transactions on Dependable and Secure Computing*, 2024.

[52] Yi Zeng, Si Chen, Won Park, Z Morley Mao, Ming Jin, and Ruoxi Jia. Adversarial Unlearning of Backdoors via Implicit Hypergradient. In *ICLR*, 2022.

[53] Zaixi Zhang, Qi Liu, Zhicai Wang, Zepu Lu, and Qingyong Hu. Backdoor defense via deconfounded representation learning. In *CVPR*, 2023.

[54] Mingli Zhu, Siyuan Liang, and Baoyuan Wu. Breaking the false sense of security in backdoor defense through re-activation attack. *arXiv*, 2024.

[55] Rui Zhu, Di Tang, Siyuan Tang, XiaoFeng Wang, and Haixu Tang. Selective amnesia: On efficient, high-fidelity and blind suppression of backdoor effects in trojaned machine learning models. In *S&P*, 2023.