FFCBA: Feature-based Full-target Clean-label Backdoor Attacks

Yangxu Yin[*]
yinyangxv@163.com
China University of Petroleum (East China)
China

Honglong Chen[†]
China University of Petroleum (East China)
China

Yudong Gao[‡]
China University of Petroleum (East China)
China

Peng Sun[§]
Hunan University
China

Liantao Wu[¶]
East China Normal University
China

Zhe Li[‖]
China University of Petroleum (East China)
China

Weifeng Liu[**]
China University of Petroleum (East China)
China

Abstract

Backdoor attacks pose a significant threat to deep neural networks, as backdoored models would misclassify poisoned samples with specific triggers into target classes while maintaining normal performance on clean samples. Among these, multi-target backdoor attacks can simultaneously target multiple classes. However, existing multi-target backdoor attacks all follow the dirty-label paradigm, where poisoned samples are mislabeled, and most of them require an extremely high poisoning rate. This makes them easily detectable by manual inspection. In contrast, clean-label attacks are more stealthy, as they avoid modifying the labels of poisoned samples. However, they generally struggle to achieve stable and satisfactory attack performance and often fail to scale effectively to multi-target attacks. To address this issue, we propose the Feature-based Full-target Clean-label Backdoor Attacks (FFCBA) which consists of two paradigms: Feature-Spanning Backdoor Attacks (FSBA) and Feature-Migrating Backdoor Attacks (FMBA). FSBA leverages class-conditional autoencoders to generate noise triggers that align perturbed in-class samples with the original category's features, ensuring the effectiveness, intra-class consistency, inter-class specificity and natural-feature correlation of triggers. While FSBA supports swift and efficient attacks, its cross-model attack capability is relatively weak. FMBA employs a two-stage class-conditional autoencoder training process that alternates between using out-of-class samples and in-class samples. This allows FMBA to generate triggers with strong target-class features, making it highly effective for cross-model attacks. We conduct experiments on multiple datasets and models, the results show that FFCBA achieves outstanding attack performance and maintains desirable robustness against the state-of-the-art backdoor defenses.

CCS Concepts

• **Security and privacy**; • **Computing methodologies → Computer vision**;

Keywords

Deep Learning, Clean-label Backdoor Attack, Multi Target

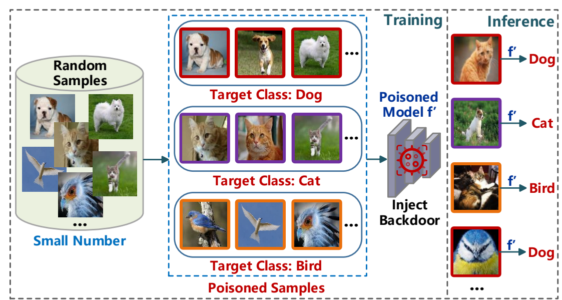


Figure 1: Schematic of low-poisoning-rate clean-label multi-target backdoor attack, where poisoned samples retain the same labels as their clean counterparts.

1 Introduction

Deep Neural Networks (DNNs) are widely used due to their high performance. However, the lack of transparency and interpretability of DNNs makes them highly vulnerable to backdoor attacks [1, 3, 28, 32, 34]. These attacks occur when an adversary embeds a backdoor into the model during training by manipulating dataset [9, 15, 20] or altering model parameters [1, 18, 22]. Consequently, the model behaves normally on clean samples but misclassifies samples with triggers into target classes during inference. While backdoor attacks on image classification tasks have grown increasingly sophisticated, the majority of these attacks are single-target, meaning they can only designate one specific class as the target. In contrast, multi-target backdoor attacks [4, 23, 30, 31] can target multiple or even all classes (i.e. full-target backdoor attacks) simultaneously and each target class is mapped to a specific trigger injection paradigm. This enables attackers to flexibly control the classification of poisoned samples into any desired predefined target class during inference, which means attackers can switch targets for maximum benefit. For instance, autonomous vehicles plan their routes in real-time based on roadside signs. With multi-target backdoor attacks, attackers can control vehicles to drive along any desired path by slightly modifying the roadside signs. Therefore, multi-target backdoor attacks with powerful payloads pose a significant threat to deep models, attracting much attention from both academia and industry.

Existing multi-target backdoor attacks have two critical flaws. One is needing to modify the labels of poisoned samples. The other is that most of them require a relatively high poisoning rate. Consequently, each target class has many poisoned samples not truly belonging to this category, making the attack highly vulnerable to detection. Therefore, maintaining original labels with a clean-label attack paradigm [26] while keeping a low poisoning rate is essential for successful attacks. However, implementing low-poisoning-rate clean-label multi-target or even full-target backdoor attacks, as depicted in Figure 1, poses two major challenges.

(1) *Existing dirty-label multi-target backdoor attacks are inapplicable to clean-label constraint and ineffective in reducing poisoning rates.* Specifically, the feature strength of dirty-label triggers is typically much weaker than the natural features of the samples. Therefore, in clean-label attacks, the model is unlikely to learn these weak trigger features and will instead focus on natural ones, causing the trigger to lose effectiveness. Moreover, most dirty-label triggers' features have low correlation with natural ones. Thus, the model needs numerous poisoned samples to identify trigger patterns, making it extremely difficult to reduce the poisoning rate.

(2) *Existing clean-label single-target attacks fail to achieve stable and satisfactory results and are hard to extend to multi-target ones.* Specifically, these attacks typically employ high-intensity noise triggers, such as adversarial noise [24, 25] and noise with strong features [21]. This results in randomness in the noise triggers generated for different samples, making it difficult to ensure strong feature consistency across all poisoned samples. Consequently, the attack effect under clean-label constraint is weakened, and the attack success rate (ASR) cannot remain stable above 99% in various datasets and models. Additionally, similarity in the form of noise triggers complicates the design of class-specific trigger injection paradigms that ensure trigger specificity across different classes. This significantly limits the extension of attacks to multiple targets. We extend the outstanding clean-label attacks, Narcissus [33] and COMBAT [13], to multi-target attacks using different seeds. The resulting average ASR across multiple targets is only 11.13% and 15.3%, respectively, validating the correctness of our analysis.

Therefore, we must ensure the trigger's effectiveness, intra-class consistency, and inter-class specificity to achieve clean-label multi-target attacks with good effects. Also, we need to enhance the trigger's natural-feature correlation to reduce the poisoning rate. To meet these essential properties, we design triggers that both obscure the natural features of clean samples and guide poisoned samples to exhibit the characteristics of the target class.

Based on the above ideas, we first propose the Feature-Spanning full-target clean-label Backdoor Attack (FSBA). Specifically, FSBA employs a carefully trained class-conditional autoencoder for the attack. When training the class-conditional autoencoder, we first overlay mid-high-frequency perturbations on samples of each class using secondary discrete wavelet transform (S-DWT). Subsequently, we use the perturbed samples along with the original class one-hot vectors as inputs to the class-conditional autoencoder. We then train it to output noise triggers that can cause the perturbed samples to re-cluster within their original feature clusters in the proxy model. This means trigger features have higher intensity than both perturbations and natural features, ensuring effectiveness and showing intra-class consistency, inter-class specificity, and natural-feature correlation at the feature level. During the backdoor injection phase, we use clean samples and their original class vectors to generate poisoned samples for the attack. During inference, the output of the poisoned model will be consistent with the class vectors used to generate poisoned samples. For FSBA's class-conditional autoencoder, each class's trigger-generation paradigm is trained only based on data from that category. This allows FSBA to perform rapid and efficient attacks. However, the limited single-category data weakens the class-conditional autoencoder's generalization ability and noise features, reducing its cross-model attack capability.

Therefore, we further propose the Feature-Migrating full-target clean-label Backdoor Attack (FMBA). FMBA follows the same attack process as FSBA but employs a different training paradigm for the class-conditional autoencoder. We design a new two-stage process to train the class-conditional autoencoder. First, we train the noise trigger to migrate the features of samples outside the target class into the feature cluster of the target class, and illustrate this idea theoretically. Specifically, we use Neural Tangent Kernel theory (NTK) [14] to show that when data approaches a uniform distribution, the feature strengths of each class become similar. This indicates that when the noise trigger obscures the features of samples outside the target class, it can also obscure the features of the target class. Thus we use abundant out-of-class samples to enhance FMBA's cross-model attack capability. Second, we fine-tune the noise trigger using samples from target class to ensure a reasonable distribution of poisoned features during the attack phase, guaranteeing the four essential properties of the trigger.

In summary, when the victim model type is known, FSBA can execute rapid and efficient attacks; when the victim model type is unknown, FMBA can accomplish attacks with excellent cross-model attack capabilities. They are complementary in application scenarios and jointly constitute Feature-based Full-target Clean-label Backdoor Attacks (FFCBA). Our contributions are as follows:

- We propose FFCBA, consisting of FSBA and FMBA. Both achieve low-poisoning-rate clean-label full-target backdoor attacks. FSBA is more efficient and faster, while FMBA has excellent cross-model attack capability.
- FFCBA enables triggers to obscure natural features and exhibit robust features corresponding to the target class, ensuring the effectiveness, intra-class consistency, inter-class specificity, and natural-feature correlation.
- Our experiments confirm that FFCBA exhibits strong attack capabilities, with minimal impact on benign accuracy and desirable robustness against advanced backdoor defenses.

2 Related Work

2.1 Backdoor Attacks

Dirty-label Backdoor Attacks. BadNets [9] first unveiled the backdoor attack chapter, with subsequent studies [2, 5, 6, 19, 20, 29] enhancing stealth and potency. However, these efforts concentrated on single-target attacks, resulting in limited attack capabilities. Works like One-to-N [30], Marksman [4], and Universal Backdoor [23] break this limitation achieving multi-target attacks. However, they all operate under dirty label conditions and, except for Universal Backdoor [23], require extremely high poisoning rates. Thus,

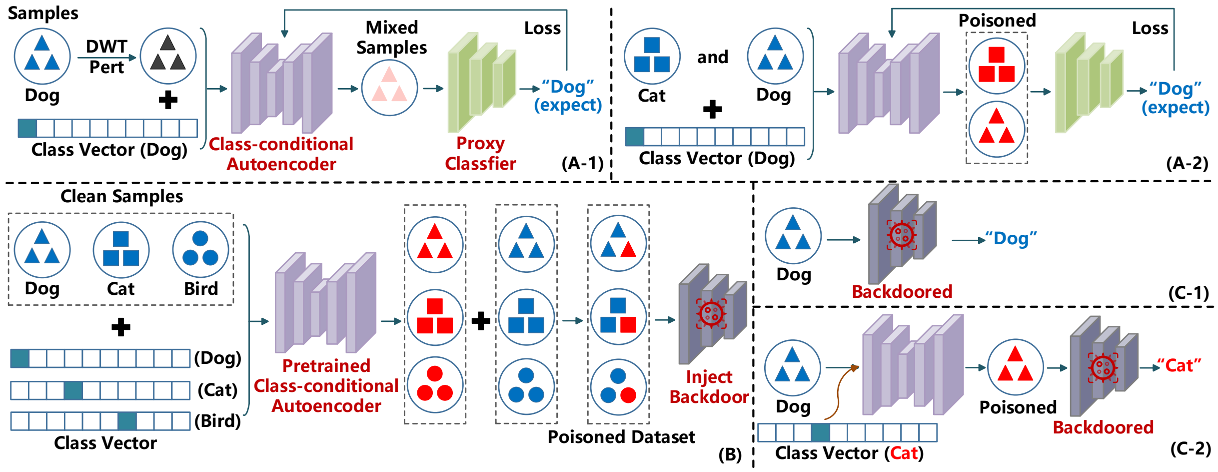


Figure 2: The attack process of FFCBA, where A-1 and A-2 represent the class-conditional autoencoder training processes of FSBA and FMBA, respectively; B is the process of injecting backdoors into the victim classification model (common to both FSBA and FMBA); C-1 and C-2 describe the outputs of the poisoned model for clean samples and poisoned samples, respectively.

they struggle to evade human detection, often leading to failure. Addressing the challenge of conducting low-poisoning-rate clean-label multi-target attacks is an urgent issue that requires resolution.

Clean-label Backdoor Attacks. Label-consistent attack [26] first achieved clean-label backdoor attacks and conducted an in-depth analysis of the reasons for previous attack failures. It laid the foundational argument for subsequent clean-label backdoor attacks that use noise to interfere with clean features. Works such as Invisible Poison [21], CSSBA [25], and Poison Frogs [24] perform clean-label backdoor attacks using adversarial or strong feature noise. Besides, Narcissus [33] uses noise to aid in feature clustering. However, these attacks fail to achieve stable and satisfactory effects. They also encounter difficulties in designing class-specific triggers, preventing their extension to a multi-target attack paradigm.

2.2 Backdoor Defenses

A detailed introduction to backdoor defenses is in Appendix A.1. The complete appendix is provided in the supplementary material.

3 Threat Model

Capability of Attackers. FSBA requires knowledge of the victim model and control over the dataset, whereas FMBA only needs the dataset control. This is because FSBA's cross-model attack capability is limited: the class-conditional autoencoder only works when the proxy model shares a similar architecture with the victim model.

Attack Modeling. In image classification, a DNN model f is trained to map images X to classes $C = \{c_1, c_2, \ldots, c_K\}$. FFCBA designs trigger injection paradigms $B_t(\cdot)$ for each target class y_t based on the proxy model f_c to attack f. The backdoored model f' classifies any poisoned sample $B_t(x)$ into its target class y_t, which is determined by the one-hot vector used to generate $B_t(x)$, while preserving performance on clean samples as:

$$f'(x) = y, \quad f'(B_t(x)) = y_t, \quad x \in X, \quad y, y_t \in C. \tag{1}$$

For FFCBA's backdoor injection process, the training set consists of N_b benign samples and N_p poisoned samples. The poisoned samples are obtained by sequentially selecting $\lfloor N_p/K \rfloor$ samples x_t from each target class y_t and applying the corresponding $B_t(\cdot)$. The labels of the poisoned samples are maintained as the original labels. In this case, the attacked DNN model $f'(\cdot;\theta)$ will be optimized according to the following optimization process:

$$\min_{\theta} \sum_{i=1}^{N_b} \mathcal{L}(f'(x_i;\theta), y_i) + \sum_{t=1}^{K} \sum_{j=1}^{\lfloor N_p/K \rfloor} \mathcal{L}(f'(B_t(x_t^j);\theta), y_t), \tag{2}$$

where x_t^j represents the j-th sample selected from class y_t and $\mathcal{L}$ denotes the cross-entropy loss. Therefore, the model will create a mapping between each $B_t(\cdot)$ and the target class y_t.

4 Methodology

4.1 Motivation

To achieve effective clean-label multi-target backdoor attacks while maintaining a low poisoning rate, each class-specific trigger injection paradigm must satisfy four key properties:

- **Trigger Effectiveness:** under clean-label constraint, the model can still capture trigger features, ensuring attack effectiveness.
- **Intra-class Consistency:** the features of triggers targeting the same class must have high consistency. If class-specific trigger features are too dispersed, the model will struggle to learn their unified characteristics, reducing backdoor performance.
- **Inter-class Specificity:** the trigger features for different classes must have sufficient differentiation. This allows the model to establish a clear one-to-one correspondence between trigger feature and corresponding target class.
- **Natural-feature Correlation:** triggers should be highly correlated with their target class's natural features. This allows the model to enhance the learning of trigger features during benign sample training, enabling low-poisoning-rate attacks.

To address this, we must ensure that the intensity of the trigger features exceeds that of the natural features to guarantee their effectiveness. Additionally, we should align the distribution of class-specific trigger features in the feature space with the corresponding natural features to ensure intra-class consistency, inter-class specificity and natural-feature correlation. Drawing on this principle, we introduce FFCBA which encompasses two distinct attack paradigms: FSBA and FMBA. The complete attack process is illustrated in Figure 2, and will be detailed in the following section.

4.2 FSBA Paradigm

FSBA employs class-conditional autoencoders for its attacks. The input to the class-conditional autoencoder consists of one-hot category vectors and clean samples, while the output is a noise trigger. These triggers exhibit strong features aligned with the category vector, exceeding the natural features in clean samples. By mixing these triggers with clean samples, we create poisoned samples to attack the victim model. Before training the class-conditional autoencoder, we train a proxy classification model f_c using clean data. Then, we train the class-conditional autoencoder following the processes (A-1) shown in Figure 2, which includes two steps:

Step 1: Perturbation with Natural Features. We linearly superimpose the enhanced features from other samples onto the natural features of clean samples as perturbations. The feature extraction process for the samples is carried out using S-DWT and the results can be summarized as follows:

$$\begin{aligned} &\text{S-DWT}(x) = \{LL_2, HL_{1,2}, LH_{1,2}, HH_{1,2}\}, \\ \Rightarrow\ &\text{S-DWT}(x) = \{YL, YH\}, \end{aligned} \tag{3}$$

where YL represents LL_2, consisting of low frequency features with the majority of the energy. YH represents the remaining part, which consists of mid-high-frequency features with lower energy content. To preserve the features of clean samples, we only perturb the low-energy YH component. The specific process is:

$$\begin{aligned} &\text{S-DWT}(x_c, x_r) = \{YL_c, YH_c, YL_r, YH_r\}, \\ &YH_{add} = YH_c + k \cdot YH_r, \quad (k > 1) \\ &x_p = \text{IS-DWT}\{YL_c, YH_{add}\}. \quad x_c, x_r \in X, \quad x_c \neq x_r. \end{aligned} \tag{4}$$

We perform S-DWT on both the clean samples x_c and other randomly selected samples x_r from the training set. Then combine their mid-high-frequency features as YH_{add}. Finally, we apply the inverse S-DWT to YH_{add} along with clean low-frequency features YL_c to obtain the perturbed samples x_p. The visual effect of x_p is shown in Appendix A.2. Although a significant amount of clean natural features is preserved, the probability of perturbed samples x_p being classified into the original category by the proxy classification model is still significantly reduced. This indicates that the intensity of the perturbation features is greater than that of the natural features, resulting in the blurring of the samples' natural characteristics as shown in Figure 3 (b).

Step 2: Feature Reconstruction Spanning the Perturbation. For each category y_k, the class-conditional autoencoder takes perturbed samples $x_{p,k}$ and one-hot label vector v_k as inputs, outputting noise triggers T_k matching the shape of $x_{p,k}$. We mix T_k with $x_{p,k}$, denoted as $x_{m,k} = x_{p,k} + T_k$, then input $x_{m,k}$ into f_c and update the class-conditional autoencoder through three designed loss functions. This enables the class-conditional autoencoder to adjust features of $x_{p,k}$, as shown in Figure 3 (c).

(1) Output Layer Loss. We aim for the mixed samples $x_{m,k}$ to be classified as their original category y_k by the proxy classification model f_c. This enables the trigger T_k to suppress the mid-high-frequency perturbations in $x_{m,k}$, allowing $x_{m,k}$ to exhibit the features of the original category y_k. Therefore, the trigger-feature strength > perturbation strength > natural-feature strength, ensuring the trigger effectiveness. Additionally, triggers for each class can exhibit robust features intrinsic to that class, guaranteeing the inter-class specificity and the natural-feature correlation. To accomplish the aim, we formulate the output layer loss function as:

$$\mathcal{L}_{output} = \sum_{k=0}^{K} \sum_{i=0}^{n_k} \mathcal{L}(f_c(x_{p,k,i} + T_{k,i}), y_k), \tag{5}$$

where K denotes the number of categories, $\mathcal{L}$ denotes the cross-entropy loss, n_k, $x_{p,k,i}$ refer to the data volume and perturbed sample of category y_k respectively, and $T_{k,i}$ denotes sample-class-specific noise trigger.

(2) Latent Space Loss. Relying solely on ensuring that the mixed samples $x_{m,k}$ fall within the classification boundary of their original category y_k poses challenges in meeting the intra-class consistency of the trigger. Specifically, in datasets with limited categories, the classification boundaries are often quite loose. This results in dispersed trigger feature distributions targeting the same class, leading to insufficient consistency. To address the issue, we calculate the centroids of the feature vector clusters for each category in the latent space of f_c, denoted as $Mean$. Here, the latent space refers to the representation of inputs in the penultimate layer of the model, denoted as $\mathcal{Z}$. Consequently, f_c can be divided into two parts: the feature extraction component $z_c : X \to \mathcal{Z}$ and the linear classification component $l_c : \mathcal{Z} \to C$, where $f_c = z_c \circ l_c$. This means that the classification result of f_c is achieved by first applying z_c, followed by l_c. By constraining the distribution of the mixed sample $x_{m,k}$ in the latent space to cluster around the centroid $Mean(y_k)$ of the original category y_k, we can significantly reduce the dispersion of the noise trigger features. Thus, we can derive the latent space loss as 6, where $\mathcal{L}_1$ denotes the L1 loss.

$$\mathcal{L}_{latent} = \sum_{k=0}^{K} \sum_{i=0}^{n_k} \mathcal{L}_1(z_c(x_{p,k,i} + T_{k,i}), Mean(y_k)). \tag{6}$$

It is important to note that the clustering of each class in the high-dimensional latent space is irregular. Consequently, latent space loss can only ensure that trigger features are compactly distributed. However, it cannot determine the specific class to which the trigger features belong. This implies that while intra-class consistency can be maintained, inter-class specificity cannot be guaranteed, highlighting the importance of output layer loss.

(3) Visual Loss. To ensure that the poisoned samples maintain good visual quality, we impose constraints on the noise trigger from a visual perspective. Previous studies often assess the visual quality of of poisoned data using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Learned Perceptual Image Patch Similarity (LPIPS), and l_∞ norm. However, since LPIPS can inherently affect the feature distribution of the noise trigger and frequent SSIM calculations will reduce training efficiency, we solely use PSNR and l_∞ norm for the visual quality constraints. Thus, we can obtain the following visual loss:

$$\begin{aligned} &\mathcal{L}_{visual} = \sum_{k=0}^{K} \sum_{i=0}^{n_k} \frac{\text{PSNR}_{\text{thresh}} - \text{PSNR}(x_{p,k,i}, x_{m,k,i})}{\text{PSNR}_{\text{thresh}}}, \\ &x_{m,k,i} = x_{p,k,i} + T_{k,i}, \quad s.t. \quad \|T_{k,i}\|_\infty \leq \epsilon, \end{aligned} \tag{7}$$

where $\text{PSNR}_{\text{thresh}}$ denotes the manually set upper limit for PSNR, and ϵ represents the threshold for the l_∞ norm of the noise trigger.

We linearly combine the three losses to form the complete loss function for training the class-conditional autoencoder, as follows:

$$\mathcal{L}_{all} = \alpha\mathcal{L}_{output} + \beta\mathcal{L}_{latent} + \gamma\mathcal{L}_{visual}. \tag{8}$$

The necessity of each loss will be elaborated in the ablation study section. Besides, it is important to note that in FSBA, the trigger generation paradigm for each category y_k is trained exclusively on samples from that category. The limited data volume for a single category allows the class-conditional autoencoder of FSBA to converge quickly, enabling efficient attacks. However, this can lead to weaker generalization and feature representation in noise triggers, reducing cross-model attack capabilities. As a result, FSBA requires a proxy model with an architecture similar to or on a comparable scale as the victim model to ensure attack effectiveness. To address this, we further propose FMBA based on FSBA.

4.3 FMBA Paradigm

FMBA uses the same attack method as FSBA with different training process of class-conditional autoencoder. To address the issues caused by insufficient data volume of single category, we choose to use out-of-class samples instead of perturbed samples. The ample supply of out-of-class samples ensures the autoencoder's generalization, and the noise trigger's ability to present strong target features significantly boosts its cross-model attack efficacy. To this end, we build upon recent studies on the neural tangent kernel (NTK) to analyze the feature strength of each category. Specifically, we demonstrate that in some datasets, when data is uniformly distributed across categories, the feature strength of each category is similar, as follows:

Assumption 1. *For a uniformly distributed dataset and a well-trained clean model f, if samples x_a and x_b from any two categories y_a and y_b are combined to obtain x_{add}, denoted as $x_{add} = x_a + x_b$, then the probability that the model f classifies x_{add} into categories y_a and y_b is approximately equal.*

Proof. The brief proof of Assumption 1 is as follows. According to the NTK theory from previous works [10, 11, 14], the model output of sample x can be expressed as:

$$\psi(x) = \frac{\sum_{k=0}^{K}\sum_{i=0}^{n_k}\mathcal{K}(x, x_{k,i})\cdot v_k}{\sum_{k=0}^{K}\sum_{i=0}^{n_k}\mathcal{K}(x, x_{k,i})}, \tag{9}$$

where $x_{k,i}$, and v_k represent the training samples, and the one-hot label vectors of category y_k, respectively. K and n_k have the same meanings as in Eq. (5). The output is a vector with same dimension as v_k, representing the probabilities of being classified into each category. Following EBBA [7], $K(x, x_{k,i}) = e^{-2\gamma||x-x_{k,i}||^2}$, $\gamma > 0$. The ratio of the probabilities that sample x_a being classified into category y_a and y_b by f tends toward a fixed multiple λ as:

$$\frac{\sum_{i=0}^{n_a} e^{-2\gamma||x_{a,i}||^2} e^{4\gamma x_a\cdot x_{a,i}}}{\sum_{i=0}^{n_b} e^{-2\gamma||x_{b,i}||^2} e^{4\gamma x_a\cdot x_{b,i}}} = \lambda. \tag{10}$$

The numerator and denominator respectively represent the similarity of x_a to the samples in categories a and b. In some datasets, such as MNIST and GTSRB, the similarity of x_a to each sample in a single category is approximately equal, and the pixel value distribution of samples in the same dataset does not have significant differences. Moreover, $n_a \approx n_b$. Therefore, we can derive that $e^{4\gamma x_a\cdot x_{a,i}} \approx \lambda e^{4\gamma x_a\cdot x_{b,i}}$. Similarly, for x_b we have $e^{4\gamma x_b\cdot x_{b,i}} \approx \lambda e^{4\gamma x_b\cdot x_{a,i}}$. Thus we can conclude that the ratio of the probabilities of x_{add} being classified into y_a and y_b is approximately 1, as:

$$\frac{\sum_{i=0}^{n_a} e^{-2\gamma||x_{a,i}||^2} e^{4\gamma x_a\cdot x_{a,i}} e^{4\gamma x_b\cdot x_{a,i}}}{\sum_{i=0}^{n_b} e^{-2\gamma||x_{b,i}||^2} e^{4\gamma x_a\cdot x_{b,i}} e^{4\gamma x_b\cdot x_{b,i}}} \approx 1. \tag{11}$$

This suggests that the feature intensity of samples from any two categories is roughly equal; otherwise, f would classify x_{add} with high confidence into the category with stronger features. A detailed proof is provided in Appendix A.3. □

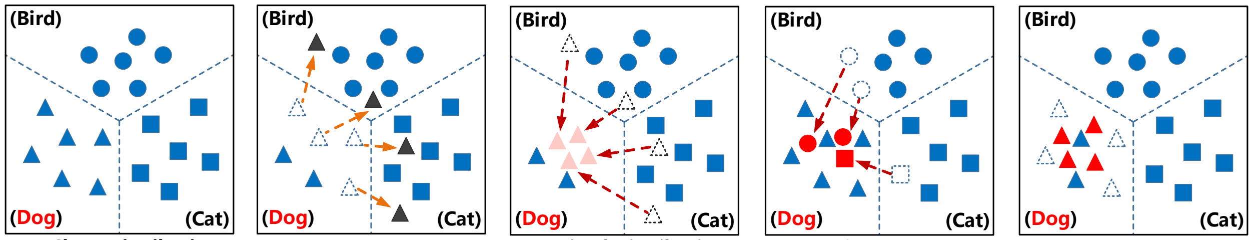
If the noise triggers are potent enough to obscure the natural features of non-target samples while presenting the features of target class, given the similar feature intensity across different categories, these triggers should also effectively conceal the natural features of the target samples and reconstruct more robust target features, meeting the four desired properties. Based on this intuition, we design the following two-stage training process for the class-conditional autoencoder, as shown in (A-2) of Figure 2:

Step 1: Out-of-class Feature Migration. For each target class y_k, we want the corresponding noise trigger T_k to make out-of-class samples x_k^{out} exhibit the features of category y_k. Therefore, we use x_k^{out} and the target class one-hot vector v_k as the input to the class-conditional autoencoder. Similarly, we use output layer loss, latent space loss, and visual loss to constrain the output noise triggers T_k. Then the complete loss function is as follows:

$$\begin{aligned}
&\mathcal{L}_{all} = \alpha\mathcal{L}_{output} + \beta\mathcal{L}_{latent} + \gamma\mathcal{L}_{visual},\\
&\mathcal{L}_{output} = \sum_{k=0}^{K}\sum_{i=0}^{N-n_k}\mathcal{L}(f_c(x_{k,i}^{out} + T_{k,i}), y_k),\\
&\mathcal{L}_{latent} = \sum_{k=0}^{K}\sum_{i=0}^{N-n_k}\mathcal{L}_1(z_c(x_{k,i}^{out} + T_{k,i}), Mean(y_k)),\\
&\mathcal{L}_{visual} = \sum_{k=0}^{K}\sum_{i=0}^{N-n_k}\frac{\text{PSNR}_{\text{thresh}} - \text{PSNR}(x_{k,i}^{out}, x_{T,k,i}^{out})}{\text{PSNR}_{\text{thresh}}},\\
&x_{T,k,i}^{out} = x_{k,i}^{out} + T_{k,i} \quad s.t. \quad ||T_{k,i}||_\infty \le \epsilon,
\end{aligned} \tag{12}$$

where N denotes the total data volume, $x_{k,i}^{out}$ represents each sample outside of category y_k, $T_{k,i}$ represents the noise trigger corresponding to $x_{k,i}^{out}$ and category y_k. After above constraints, when mixing noise triggers, $x_{k,i}^{out}$ will be classified into the target class y_k and will cluster around the centroid of the feature cluster by proxy model f_c, as shown in Figure 3 (d), while also ensuring stealthiness.

Step 2: In-class Feature Fine-tuning. During the attack, poisoned samples are generated from in-class samples of the target class. Thus, for each target class y_k, we must ensure that in-class samples x_k could exhibit strong robust target class features after superimposing the corresponding noise triggers T_k. Therefore, we use the in-class samples x_k and the one-hot vector v_k of category y_k as inputs to the class-conditional autoencoder. We then constrain the output noise triggers T_k using output layer loss and latent space loss. In order to refine the noise trigger features more effectively without the interference of visual factors, we omit the visual loss



(a) Clean Distribution (b) Perturbed Distribution (c) Mixed Distribution (d) Out-of-class Migration (e) In-class Migration

Figure 3: The feature distribution changes of target class during the training process of class-conditional autoencoders. (a) represents the distribution of clean samples in the latent space of the proxy model. (b) and (c) represent the distribution of samples after DWT perturbation and after mixing FSBA noise trigger. (d) and (e) represent the migration of out-of-class and in-class samples after mixing FMBA noise trigger, respectively. The color of samples in each state is consistent with Figure 2.

at this stage. The complete loss function is:

$$\begin{aligned} \mathcal{L}_{all} &= \alpha \mathcal{L}_{output} + \beta \mathcal{L}_{latent}, \\ \mathcal{L}_{output} &= \sum_{k=0}^{K} \sum_{i=0}^{n_k} \mathcal{L}(f_c(x_{k,i} + T_{k,i}), y_k), \\ \mathcal{L}_{latent} &= \sum_{k=0}^{K} \sum_{i=0}^{n_k} \mathcal{L}_1(z_c(x_{k,i} + T_{k,i}), Mean(y_k)). \end{aligned} \tag{13}$$

The refinement of noise trigger features ensures that poisoned samples from various categories will cluster within their respective feature clusters in the latent space as shown in Figure 3 (e), maintaining the orderliness of multiple backdoors during the victim model's training phase.

Through the two-stage learning process, FMBA is capable of generating noise triggers with strong and robust target class features, thereby possessing excellent cross-model attack capabilities. The class-conditional autoencoder trained on any pre-trained proxy classification model can be used to attack any victim models with different architectures. We present specific attack results in the experimental section. It's important to highlight that while FMBA has learned the feature learning process of adversarial attacks, it is significantly distinct from them. The efficacy of the trigger is markedly diminished without the backdoor implantation process. For example, when a class-conditional autoencoder is trained on the Resnet18 proxy model using the ImageNet100 dataset and then used to perform backdoor-free adversarial attacks on the VGG19 and Densenet121 models, the success rates are a mere 29.72% and 27.28%, respectively. This underscores that FMBA's effectiveness is contingent upon the presence of a backdoor injection process.

5 Evaluation

5.1 Experimental Setup

Baseline. Currently, no research has successfully executed clean-label multi-target attacks. Consequently, we selected three of the most advanced *dirty-label* multi-target backdoor attacks—One-to-N [30], Marksman [4], and Universal Backdoor Attacks [23]—as our baselines. Additionally, we have verified that state-of-the-art clean-label attack paradigms, such as Narcissus [33] and COMBAT [13], struggle to achieve multi-target attacks by setting different seeds. Thus, we no longer include them in our baselines. Table 1 presents the properties of each attack paradigm. It is evident that FFCBA can achieve stable attack results under the strictest constraints.

Table 1: Attack properties comparison.

Methods	Properties (✓/✗)				
	Clean Label	Low Poisoning Rate	Full Target	Stable Results	Black-Box Settings
FSBA	✓	✓	✓	✓	✗
FMBA	✓	✓	✓	✓	✓
One-to-N	✗	✗	✗	✗	✓
Marksman	✗	✗	✓	✓	✗
UBA	✗	✓	✓	✓	✗
Narcissus	✓	✓	✗	✗	✓
COMBAT	✓	✓	✗	✗	✓

Dataset and Model. To evaluate FFCBA's performance against the baselines, we use the standard datasets for backdoor attack evaluations, including CIFAR10, Animals90, and ImageNet100. For each dataset, we employ the pre-trained Resnet18 and VGG19 to determine benign accuracy. Furthermore, we evaluate FFCBA's cross-model attack capabilities on three architecturally diverse models: Resnet50, Densenet121, and ViT_B_16. Details of the class-conditional autoencoder are provided in Appendix A.4.

Hyperparameters. The specific parameter settings are detailed in Appendix A.5.

5.2 Effectiveness of FFCBA

Attack Effectiveness. Table 2 contrasts FFCBA's attack performance with baselines. FSBA and FMBA, using class-conditional autoencoders trained on the Resnet18 proxy model, target the Resnet18 victim model and execute cross-model attacks on VGG19. We outline the average ASR across all labels, Benign classification Accuracy (BA), and the Decrease Value in BA (DV) compared to the clean model. FFCBA attains high ASR across all labels and datasets while minimally impacting BA, showing advantages over various baselines. Against One-to-N, FFCBA gains a significant ASR boost. Compared to Marksman and Universal Backdoor, FFCBA matches their attack effectiveness under stricter clean-label constraints. Moreover, FFCBA maintains a low 0.4% poisoning rate across all datasets, much lower than Marksman (10%) and One-to-N (20%), and comparable to Universal Backdoor (0.62%). Figure 4 specifically illustrates how the ASR of FFCBA changes with varying poisoning rates. In addition, FFCBA's poisoned samples exhibit better visual quality than baselines, as illustrated in Appendix A.2. Table 3 presents FFCBA's ASR for each CIFAR10 label with Resnet18 victim model. FFCBA demonstrates robust attack performance in

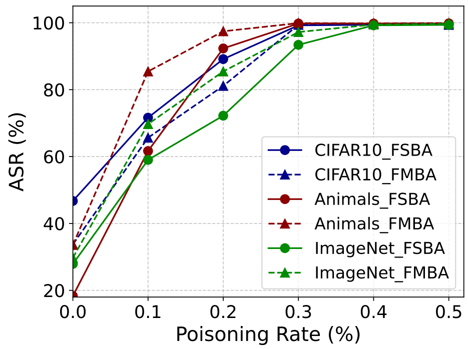


Figure 4: ASR of FFCBA across different poisoning rates with Resnet18 proxy model and VGG19 victim model.

Table 2: Performance of FFCBA compared with baselines.

Method	Metric (%)	CIFAR10		Animals90		ImageNet100	
		Resnet18	VGG19	Resnet18	VGG19	Resnet18	VGG19
FSBA	ASR	99.75	99.88	99.33	100.0	99.96	99.86
	BA	86.01	90.64	90.33	90.44	86.40	88.45
	DV	1.21	0.29	0.02	0.12	1.85	0.99
FMBA	ASR	99.93	99.90	99.78	99.45	99.38	99.34
	BA	86.03	90.43	89.77	90.55	86.92	89.36
	DV	1.19	0.50	0.58	0.01	1.33	0.08
One-to-N	ASR	41.96	58.42	6.100	10.15	11.64	10.67
	BA	81.72	88.17	80.22	83.44	82.22	83.44
	DV	5.50	2.76	9.59	7.12	6.03	6.00
Marksman	ASR	99.64	100.0	99.98	99.67	99.94	99.48
	BA	86.66	90.78	89.56	88.44	81.08	85.68
	DV	0.56	0.15	0.79	2.12	7.17	3.76
UBA	ASR	99.32	100.0	99.22	99.71	99.80	99.90
	BA	83.86	85.23	88.67	89.55	88.01	88.28
	DV	3.36	5.70	1.68	1.01	0.24	1.16

Table 3: Attack performance of FFCBA on CIFAR10 and Resnet18 for each target.

Paradigm	Target	ASR of each category (%)				
FSBA	category 0-4	99.98	99.96	100.0	99.93	99.73
	category 5-9	99.98	99.36	99.74	99.14	99.72
FMBA	category 0-4	100.0	100.0	100.0	100.0	100.0
	category 5-9	99.98	99.98	99.98	99.53	100.0

Table 4: Cross-model attack performance of FFCBA on different datasets using Resnet18 proxy model.

Dataset	Metric (%)	FSBA			FMBA		
		Densenet	VITB16	Resnet50	Densenet	VITB16	Resnet50
CIFAR10	ASR	94.38	74.14	96.96	99.49	99.88	99.26
	BA	87.22	98.03	88.05	87.38	98.12	88.72
	DV	0.64	0.10	0.71	0.48	0.01	0.04
Animals 90	ASR	85.67	95.56	90.67	100.0	100.0	100.0
	BA	91.56	94.00	91.72	93.88	94.44	94.11
	DV	2.33	0.56	2.44	0.01	0.12	0.05
ImageNet 100	ASR	98.02	99.12	99.08	99.46	99.22	99.04
	BA	90.24	92.34	91.12	90.22	92.38	91.18
	DV	0.06	0.32	0.10	0.08	0.28	0.04

all categories. Given the large number of categories in Animals90 and ImageNet100, we omit ASR for each label individually.

Table 5: Cross-model attack performance of FFCBA on ImageNet100 using different proxy models.

FFCBA	Proxy Model	Metric (%)	Victim Model Architecture				
			Resnet18	VGG19	Densenet	VITB16	Resnet50
FSBA	Densenet	ASR	96.92	99.48	99.50	99.45	97.86
		BA	87.08	89.34	89.66	92.62	91.16
		DV	1.17	0.10	0.64	0.04	0.06
	VITB16	ASR	74.10	99.46	62.08	99.52	70.48
		BA	86.70	89.18	90.24	92.62	91.20
		DV	1.55	0.26	0.06	0.04	0.02
FMBA	Densenet	ASR	99.06	99.88	99.82	99.72	99.62
		BA	86.84	88.88	89.70	92.42	89.60
		DV	1.41	0.56	0.60	0.24	1.62
	VITB16	ASR	99.30	99.60	99.26	99.84	99.10
		BA	85.60	87.64	88.20	92.46	89.80
		DV	2.65	1.80	2.10	0.20	1.42

Cross-model Attack Capability. We further evaluate the cross-model attack capabilities of FFCBA under various conditions. Table 4 presents FFCBA's cross-model attack performance across different datasets. Specifically, we train class-conditional autoencoders on various datasets using the Resnet18 proxy model to attack other models with significant architectural and scale differences. The results indicate that FMBA consistently exhibits excellent cross-model attack capabilities on any dataset, while FSBA's performance declines as architectural disparities increase. Table 5 presents FFCBA's cross-model attack performance across different proxy models. Specifically, we train class-conditional autoencoders on ImageNet100 using Densenet121 and ViT_B_16 proxy models, which have significant structural differences, to launch attacks on various models. The results show that FMBA maintains superior cross-model attack capabilities regardless of the proxy model used, whereas FSBA performs weaker. Despite its limited cross-model capabilities, FSBA can execute swift and potent attacks when the victim model type is known. Table 6 illustrates FSBA's effectiveness with an accurate proxy model, achieving superior attack outcomes across various model architectures and datasets. Therefore, with knowledge of the victim model type, FSBA can be deployed for quick and efficient attacks. Conversely, when the victim model type is unknown or it is challenging to train an accurate proxy model, FMBA can be effectively employed for cross-model attacks. Their application scenarios are complementary.

5.3 Robustness against Backdoor Defenses

In this section, we evaluate the robustness of FFCBA against popular backdoor defense mechanisms, including Fine-Pruning [17], Neural Cleanse [27], STRIP [8], CBD [35], EBBA [7], ABL [16], and IBD-PSC [12]. These defense mechanisms have proven to be effective against previous backdoor attacks.

Resistance to Fine-Pruning. Fine-Pruning neutralizes backdoors by pruning dormant neurons while ensuring benign accuracy. Figure 5a and Figure 5b show the resistance of FFCBA to Fine-Pruning. It can be observed that under different pruning rounds, the degree of ASR decline is always lower than BA, hence Fine-Pruning cannot defend against these two attack paradigms.

Resistance to Neural Cleanse. Neural Cleanse detects backdoors by constructing reverse-engineered triggers and measuring

Table 6: Performance of FSBA using accurate proxy models.

Dataset	Metric (%)	Model Architecture				
		Resnet18	VGG19	Densenet	VITB16	Resnet50
CIFAR10	ASR	99.75	99.98	99.59	99.84	99.09
	BA	86.01	90.85	87.47	98.02	88.41
	DV	1.21	0.08	0.39	0.11	0.35
Animals 90	ASR	99.33	99.78	100	99.88	99.56
	BA	90.33	90.45	93.22	94.22	93.67
	DV	0.02	0.11	0.67	0.34	0.49
ImageNet 100	ASR	99.96	99.94	99.5	99.52	99.96
	BA	86.4	88.9	89.66	92.62	91.06
	DV	1.85	0.54	0.64	0.04	0.16

whether anomaly metric exceed a threshold of 2. Figure 5c illustrates the resistance of FSBA and FMBA to Neural Cleanse. Both exhibit anomaly metrics below the threshold, indicating that Neural Cleanse cannot defend against these two attack paradigms.

Resistance to STRIP. STRIP perturbs clean images to generate high-entropy outputs in benign models. Low entropy indicates the presence of backdoors. Figure 5e and Figure 5f show that the entropy distributions of clean and poisoned samples of Animals90 for FSBA and FMBA are similar, suggesting that STRIP cannot defend against these two attack paradigms. The results on the other two datasets are provided in Appendix A.6.

Resistance to CBD. CBD learns causal relationships and employs a per-sample weighting scheme on contaminated datasets to obtain a clean model. Figure 5d presents the results of FFCBA's resistance to CBD. It is evident that both FSBA and FMBA exhibit high ASR, demonstrating their robustness against CBD.

Resistance to EBBA. EBBA calculates the energy of each label in the model's output. If any label exhibits an abnormally high energy value, it suggests the presence of a backdoor. Figure 5g and Figure 5h present the results of FSBA and FMBA against EBBA on ImageNet100, showing that the energy distribution across labels is quite uniform. The results on the other two datasets are provided in Appendix A.7. Since every label is the target label, EBBA is destined to fail in detecting FFCBA.

Resistance to ABL. ABL isolates backdoor examples at the early training stage and later breaks their correlation with the target class, so that training train clean models based on poisoned data. Figure 5i shows the results of FSBA and FMBA against ABL. Both maintain high ASR, indicating they can bypass ABL.

Resistance to IBD-PSC. IBD-PSC detects backdoor attacks by amplifying batch normalization parameters in models, enhancing the prediction confidence consistency of poisoned samples, thus identifying malicious inputs effectively. Figure 5j shows the results of FSBA and FMBA against IBD-PSC. Their low AUROC and F1 scores indicate that IBD-PSC cannot defend against FFCBA.

5.4 Ablation Study

We conducted ablation studies on three parameters, α, β, γ. Specifically, we trained a class-conditional autoencoder using the CIFAR10 dataset and a Resnet18 proxy model, and attacked the VGG19 model. Each parameter was varied by 0.25, while the others were fixed at their optimal values in Hyperparameters. Table 7 and Table 8 list changes in ASR and visual metrics. Both output layer loss and latent space loss significantly impact ASR, highlighting their importance

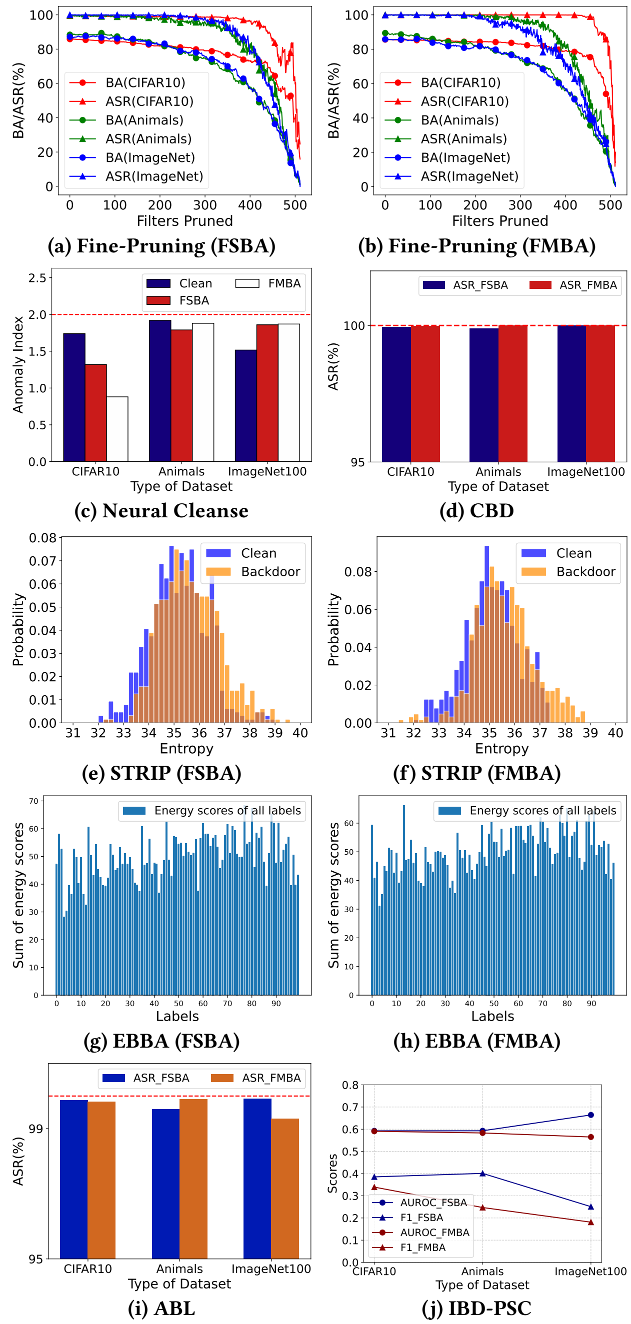


Figure 5: FFCBA's performance against various defenses.

for attack performance. While visual loss does not directly affect ASR, it greatly influences the visual metrics of poisoned samples, making it essential for maintaining visual stealthiness.

6 Conclusion

In this paper, we introduce FFCBA, a novel backdoor attack with two paradigms: FSBA and FMBA. FSBA uses class-conditional autoencoders to generate effective noise triggers, while FMBA extends FSBA by replacing intra-class samples with out-of-class ones,

Table 7: ASR under different parameters.

Range	FSBA			FMBA		
	α	β	γ	α	β	γ
0	64.68	82.70	99.86	95.51	96.02	99.99
0.25	95.55	97.24	99.84	99.58	99.30	99.94
0.5	99.80	99.78	99.66	99.85	99.94	99.96
0.75	99.23	99.85	99.02	99.55	99.26	99.99

Table 8: Visual performance under different γ.

γ	FSBA			FMBA		
range	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
0	22.43	0.766	0.047	21.29	0.725	0.056
0.25	26.43	0.868	0.021	25.04	0.803	0.041
0.5	30.45	0.935	0.009	30.97	0.943	0.007
0.75	32.48	0.957	0.006	33.51	0.976	0.002

enhancing cross-model attack capabilities. Both can execute clean-label full-target backdoor attacks at low poisoning rates, effectively addressing the vulnerability of previous multi-target attack paradigms to human detection. FFCBA also demonstrates strong robustness against the state-of-the-art defenses. We demonstrate its excellent effectiveness and high defense resistance across various datasets and models.

References

[1] Jiawang Bai, Baoyuan Wu, Yong Zhang, Yiming Li, Zhifeng Li, and Shu-Tao Xia. 2021. Targeted Attack against Deep Neural Networks via Flipping Limited Weight Bits. In *Proceedings of the International Conference on Learning Representations*.

[2] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526* (2017).

[3] Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. 2021. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

[4] Khoa D Doan, Yingjie Lao, and Ping Li. 2022. Marksman backdoor: Backdoor attacks with arbitrary target class. *Advances in Neural Information Processing Systems* (2022).

[5] Yu Feng, Benteng Ma, Jing Zhang, Shanshan Zhao, Yong Xia, and Dacheng Tao. 2022. FIBA: Frequency-Injection Based Backdoor Attack in Medical Image Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[6] Yudong Gao, Honglong Chen, Peng Sun, Junjian Li, Anqing Zhang, Zhibo Wang, and Weifeng Liu. 2024. A Dual Stealthy Backdoor: From Both Spatial and Frequency Perspectives. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[7] Yudong Gao, Honglong Chen, Peng Sun, Zhe Li, Junjian Li, and Huajie Shao. 2024. Energy-based Backdoor Defense without Task-Specific Samples and Model Retraining. In *Proceedings of the International Conference on Machine Learning*.

[8] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. 2019. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the Annual Computer Security Applications Conference*.

[9] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access* (2019).

[10] Junfeng Guo, Ang Li, and Cong Liu. 2022. AEVA: Black-box Backdoor Detection Using Adversarial Extreme Value Analysis. In *Proceedings of the International Conference on Learning Representations*.

[11] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. 2023. SCALE-UP: An Efficient Black-box Input-level Backdoor Detection via Analyzing Scaled Prediction Consistency. In *Proceedings of the International Conference on Learning Representations*.

[12] Linshan Hou, Ruili Feng, Zhongyun Hua, Wei Luo, Leo Yu Zhang, and Yiming Li. 2024. IBD-PSC: Input-level backdoor detection via parameter-oriented scaling consistency. *arXiv preprint arXiv:2405.09786* (2024).

[13] Tran Huynh, Dang Nguyen, Tung Pham, and Anh Tran. 2024. Combat: Alternated training for effective clean-label backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[14] Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems* (2018).

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

[16] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. 2021. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems* (2021).

[17] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses*.

[18] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojaning attack on neural networks. In *Proceedings of the Annual Network And Distributed System Security Symposium*.

[19] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. 2020. Reflection backdoor: A natural backdoor attack on deep neural networks. In *Proceedings of the European Conference on Computer Vision*.

[20] Tuan Anh Nguyen and Anh Tuan Tran. 2021. WaNet - Imperceptible Warping-based Backdoor Attack. In *Proceedings of the International Conference on Learning Representations*.

[21] Rui Ning, Jiang Li, Chunsheng Xin, and Hongyi Wu. 2021. Invisible poison: A blackbox clean label backdoor attack to deep neural networks. In *Proceedings of the IEEE Conference on Computer Communications*.

[22] Xiangyu Qi, Tinghao Xie, Ruizhe Pan, Jifeng Zhu, Yong Yang, and Kai Bu. 2022. Towards practical deployment-stage backdoor attack on deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[23] Benjamin Schneider, Nils Lukas, and Florian Kerschbaum. 2024. Universal Backdoor Attacks. In *Proceedings of the International Conference on Learning Representations*.

[24] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in Neural Information Processing Systems* (2018).

[25] Zihan Shen, Wei Hou, and Yun Li. 2023. CSSBA: A Clean Label Sample-Specific Backdoor Attack. In *Proceedings of the IEEE International Conference on Image Processing*.

[26] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. 2019. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771* (2019).

[27] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *Proceedings of the IEEE Symposium on Security and Privacy*.

[28] Hao Wang, Shangwei Guo, Jialing He, Kangjie Chen, Shudong Zhang, Tianwei Zhang, and Tao Xiang. 2024. Eviledit: Backdooring text-to-image diffusion models in one second. In *Proceedings of the ACM International Conference on Multimedia*.

[29] Tong Wang, Yuan Yao, Feng Xu, Shengwei An, Hanghang Tong, and Ting Wang. 2022. An invisible black-box backdoor attack through frequency domain. In *Proceedings of the European Conference on Computer Vision*.

[30] Mingfu Xue, Can He, Jian Wang, and Weiqiang Liu. 2020. One-to-N & N-to-One: Two advanced backdoor attacks against deep learning models. *IEEE Transactions on Dependable and Secure Computing* (2020).

[31] Mingfu Xue, Shifeng Ni, Yinghao Wu, Yushu Zhang, and Weiqiang Liu. 2024. Imperceptible and multi-channel backdoor attack. *Applied Intelligence* (2024).

[32] Wen Yin, Bin Benjamin Zhu, Yulai Xie, Pan Zhou, and Dan Feng. 2024. Backdoor Attacks on Bimodal Salient Object Detection with RGB-Thermal Data. In *Proceedings of the ACM International Conference on Multimedia*.

[33] Yi Zeng, Minzhou Pan, Hoang Anh Just, Lingjuan Lyu, Meikang Qiu, and Ruoxi Jia. 2023. Narcissus: A practical clean-label backdoor attack with limited information. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.

[34] Xinwei Zhang, Aishan Liu, Tianyuan Zhang, Siyuan Liang, and Xianglong Liu. 2024. Towards robust physical-world backdoor attacks on lane detection. In *Proceedings of the ACM International Conference on Multimedia*.

[35] Zaixi Zhang, Qi Liu, Zhicai Wang, Zepu Lu, and Qingyong Hu. 2023. Backdoor defense via deconfounded representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.