Article

Not peer-reviewed version

Backdoor Training Paradigm in Generative Models

Huangji Wang and Fan Cheng *

Posted Date: 3 January 2025

doi: 10.20944/preprints202501.0103.v1

Keywords: backdoor attack; generative model; diffusion model; GAN; paradigm

 doi:10.20944/preprints202501.0103.v1

Article

Backdoor Training Paradigm in Generative Models

Huangji Wang and Fan Cheng *

[1] Shanghai Jiao Tong University

* Correspondence: chengfan@cs.sjtu.edu.cn

Abstract: Backdoor attacks remain a critical area of focus in machine learning research, with one prominent approach being the introduction of backdoor training injection mechanisms. These mechanisms embed backdoor triggers into the training process, enabling the model to recognize specific trigger inputs and produce predefined outputs post-training. In this paper, we identify a unifying pattern across existing backdoor injection methods in generative models and propose a novel backdoor training injection paradigm. This paradigm leverages a unified loss function design to facilitate backdoor injection across diverse generative models. We demonstrate the effectiveness and generalizability of this paradigm through experiments on Generative Adversarial Networks (GANs) and Diffusion Models. Our experimental results on GANs confirm that the proposed method successfully embeds backdoor triggers, enhancing the model's security and robustness. This work provides a new perspective and methodological framework for backdoor injection in generative models, making a significant contribution toward improving the safety and reliability of these models.

Keywords: backdoor attack; generative model; diffusion model; GAN; paradigm

1. Introduction

With the rapid development of deep learning, powerful models have emerged for learning complex data such as high-dimensional data, temporal data, spatial data, and graph data. Generative models are a class of powerful models that aim to learn the distribution of data in order to generate new samples that resemble real data [59]. Common types of generative models include Generative Adversarial Networks (GANs) [9,57,58,60–62], Variational Autoencoders (VAEs) [53–56], Diffusion Models [3,7,50,52], and Autoregressive Models [51]. These models have found widespread application in multimodal generation tasks [44–49]. Despite their significant success, generative models face several security and privacy challenges, one of which is the threat of backdoor training attacks. These attacks raise concerns about the security of generative models in safety-critical scenarios, such as privacy protection [42,43], copyright claims [27,41], and model integrity [40].

However, state-of-the-art deep neural network models consolidate the knowledge of researchers while consuming vast amounts of data and computational resources, leading to high costs. Although models as a product for commercial sale (Model as a Service, MLaaS) [26] can be a lucrative business model, the low cost of stealing, copying, or misusing these models poses significant risks. To prevent such misuse and intellectual property infringement, effective backdoor methods are crucial for safeguarding model ownership.

Backdoor training, a type of backdoor attack, or a type of ownership verification, involves injecting a trigger into a small subset of training data to implant a backdoor into the model [63]. The core goal of this method is to ensure that the model performs normally on regular inputs while producing a pre-determined output under specific trigger conditions. Related works [65–68] indicate that this type of attack poses a serious threat to deep neural network-based models, as backdoor triggers are relatively easy to implant but difficult to detect or remove [64], which means this is also a method to protect the ownership. A key feature of backdoor attacks is that they do not degrade the model's

 doi:10.20944/preprints202501.0103.v1

performance on clean test inputs, yet they allow the attacker to control the model's behavior for any test input containing the backdoor trigger. This makes it challenging to detect such attacks based solely on the model's performance on clean test sets. [69–72]

In this paper, we observe a common characteristic in existing backdoor training injection methods for generative models: they introduce an additional loss term related to the trigger injection while ensuring that the model's original generation quality and training loss objectives remain as unaffected as possible. This loss term is typically controlled by a hyperparameter λ for fine-tuning. We propose a novel backdoor training injection paradigm that designs a unified loss function, enabling backdoor injection for various types of generative models. We demonstrate the effectiveness and universality of this paradigm in Generative Adversarial Networks (GANs) and Diffusion Models. Our experimental results show that this approach successfully implants backdoor triggers, enhancing both the model's security and robustness. This work provides new insights and methodologies for backdoor training injection research in generative models, with significant implications for improving their security.

2. Related Works

2.1. Backdoor Training Protection

The concept of backdoor injection for protecting neural network models can be traced back to the seminal work in 2017, where watermarking was introduced into convolutional neural networks using regularization techniques [1]. Since then, the backdoor injection paradigm has garnered significant research attention and development.

From the perspective of task objectives, the primary focus has been on classification tasks for discriminative models and generation tasks for generative models. Structurally, most studies center on convolutional neural networks (CNNs) [28] due to their exceptional performance in image processing, the rapid growth of large-scale image datasets, and the widespread application of CNNs across diverse domains.

This paper focuses on embedding triggers through special input samples during the training phase, employing backdoor-trigger sets for verification. Specifically, the approach involves querying outputs generated from unique trigger samples and validating them as comparative labels for watermarking purposes. Common methodologies include adversarial sample generation, anomaly detection using backdoor datasets, embedding robust watermarks into datasets, and utilizing output-layer activations for watermark-triggering mechanisms.

In addition to these, novel embedding techniques have emerged. For instance, combining deep learning algorithms with hardware-level integrations has enabled watermark encryption within the hardware domain [5]. Furthermore, systematic validation methods have been proposed to ensure the robustness and reliability of backdoor injection in neural networks [6]. Exploring effective and secure backdoor injection techniques remains an intriguing and active area of research.

2.2. Generative Adversarial Network (GAN)

Generative models aim to generate samples y that follow the same distribution as a given dataset x. Generative Adversarial Networks (GANs) [2], introduced to address this problem, effectively model and fit such generative distributions. A GAN consists of two key components: a discriminator $\mathcal{D}$ and a generator $\mathcal{G}$. The generator $\mathcal{G}$ is responsible for modeling the data distribution and producing samples that mimic the distribution of the input data x. Meanwhile, the discriminator $\mathcal{D}$ evaluates whether a given sample is real (from the data distribution) or generated.

The primary goal of a GAN is to iteratively optimize both components such that $\mathcal{G}$ improves its ability to generate realistic data that $\mathcal{D}$ cannot distinguish from real samples, while $\mathcal{D}$ concurrently enhances its ability to identify generated samples. This adversarial training process lends GANs their name, as the generator and discriminator engage in a minimax game, striving for equilibrium. Ideally, this process reaches a Nash equilibrium, where $\mathcal{G}$ generates samples indistinguishable from

 doi:10.20944/preprints202501.0103.v1

the true distribution x, and $\mathcal{D}$ assigns a probability of 0.5 to all inputs being real or generated. The game-theoretic formulation of GANs is defined as follows:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{D}, \mathcal{G}) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log \mathcal{D}(x)] + \mathbb{E}_{z \sim p_{\mathcal{G}}(z)}[\log (1 - \mathcal{D}(\mathcal{G}(z)))] \tag{1}$$

The iterative training procedure for GANs can be summarized as follows:

1. Initialize the parameters $\theta_{\mathcal{D}}$ for the discriminator and $\theta_{\mathcal{G}}$ for the generator.
2. Sample m real data points $\{x^{(1)}, \ldots, x^{(m)}\}$ from the true data distribution $p_{\text{data}}(x)$. Simultaneously, sample m noise vectors $\{z^{(1)}, \ldots, z^{(m)}\}$ from a prior noise distribution $p_{\mathcal{G}}(z)$. Pass these noise vectors through the generator to produce corresponding fake samples $\{\hat{x}^{(1)}, \ldots, \hat{x}^{(m)}\}$.
3. Alternately train $\mathcal{D}$ and $\mathcal{G}$:
   (a) Fix $\mathcal{G}$ and optimize $\mathcal{D}$ to improve its ability to distinguish real samples from generated ones.
   (b) Fix $\mathcal{D}$ and optimize $\mathcal{G}$ to produce samples that maximize the probability of fooling $\mathcal{D}$. This involves using the gradient of $\mathcal{D}$'s loss to update $\mathcal{G}$, guiding it towards generating samples closer to the true data distribution.

In the original GAN work [4], the training strategy prioritizes the discriminator. Loss is computed for $\mathcal{D}$ using real and generated samples, followed by backpropagation to update its parameters. Subsequently, the generator is trained by leveraging the gradients from $\mathcal{D}$ to adjust its parameters, steering it towards generating more realistic data. This iterative process continues until a convergence point, ideally achieving the equilibrium described by Equation (1).

2.3. Diffusion Models

A Denoising Diffusion Probabilistic Model (DDPM) [7] employs two Markov chains: one for the forward process, which progressively adds noise to the data, and another for the reverse process, which reconstructs the data from the noise. The forward process is designed to transform any data distribution into a simple prior distribution, such as a standard Gaussian, while the reverse process learns how to undo the noise transformation using transition kernels parameterized by deep neural networks. Data generation involves sampling a random vector from the prior distribution and using ancestral sampling through the reverse chain to produce new data points. [3]

Forward Process (Noise Addition): The forward process is a Markov chain that gradually corrupts the data by adding noise at each step. Let x_0 represent the original data, and x_t denote the noisy version of the data at timestep t. The process adds Gaussian noise at each timestep, with the noise schedule controlled by β_t. The formula can be expressed as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I}),$$

where β_t controls the amount of noise added at each step. As t increases, the data becomes noisier. After T steps, the data x_T converges to a nearly uniform Gaussian distribution:

$$q(x_T|x_0) = \mathcal{N}(x_T; 0, \mathbf{I}),$$

signifying that at T, the data is fully corrupted by noise.

Reverse Process (Denoising): The reverse process is key to the generative capability of diffusion models. It aims to gradually remove the noise from the corrupted data x_T and recover the original data distribution x_0. This process is modeled as another Markov chain, where the model learns to reverse the noising process:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)),$$

 doi:10.20944/preprints202501.0103.v1

where $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ are the predicted mean and covariance for the denoised data at each timestep t. The reverse process aims to reduce noise progressively, moving from x_T to x_0. The final goal is to reconstruct the original input x_0 based on noise x_T.

The model is trained to maximize the likelihood of the observed data under the reverse process, which is typically achieved by minimizing the Kullback-Leibler (KL) divergence [29] between the true posterior $p(x_{t-1}|x_t, x_0)$ and the learned posterior $p_\theta(x_{t-1}|x_t)$. This leads to the following loss function:

$$L = \mathbb{E}_{q(x_t|x_0)}[D_{\mathrm{KL}}(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t))].$$

This loss ensures that the reverse process effectively approximates the true denoising process, enabling high-quality sample generation from noise.

Training and Sampling: During training, the model learns to predict the clean data x_0 (or equivalently, the noise component ϵ) from noisy inputs x_t. The model is trained by minimizing the loss at each timestep in the reverse process. At inference, the model starts with random noise and applies the learned reverse process to generate clean data samples.

Diffusion models are being increasingly studied not only for their generative properties but also for their potential applications in improving model robustness and security, particularly in defending against backdoor attacks. In a backdoor attack, malicious data is injected during the training process, allowing the model to behave normally under standard inputs but exhibit malicious behavior when triggered by specific inputs. Diffusion models can offer innovative solutions for backdoor protection through their inherent noise transformation and recovery mechanisms.

3. Backdoor Training Paradigm in Generative Models

3.1. Backdoor Training in GANs

Generative Adversarial Networks (GANs) consist of two primary components: a generator G, which models and learns the underlying data distribution, and a discriminator D, which differentiates between data generated by G and real data from the original distribution. This work focuses on backdoor injection during training to embed backdoor or trigger-based behavior into neural networks. Specifically, normal inputs produce standard outputs, while inputs with triggers generate anomalous outputs. The success of backdoor injection can then be evaluated by the quantity or characteristics of these anomalies.

Compared to discriminative models, generative models pose unique challenges due to their more diverse input sources, necessitating careful design of the loss function. A typical formulation includes a backdoor training model loss L_b added to the original model loss L_o, as shown in Equation 2:

$$L = L_o + \lambda L_b, \tag{2}$$

where L_w accounts for the backdoor-specific requirements. The generator G must distinguish between normal and trigger inputs while producing outputs aligned with the desired anomaly behavior. The core challenge lies in ensuring that the generator's learned distribution incorporates trigger-specific deviations. This subsection discusses backdoor injection techniques in recent GAN works [13–16].

In our investigation of backdoor training injection in GAN models, we observed a striking commonality across multiple works [13–16]. Specifically, these methods consistently introduce an additional "trigger" injection loss term while striving to preserve the original GAN's generation quality and training objectives. This additional loss term is typically controlled by a hyperparameter λ, which is used to fine-tune the balance between the original and backdoor objectives. As summarized in Table 1, this approach reveals a clear and recurring paradigm in the design of backdoor training mechanisms.

 doi:10.20944/preprints202501.0103.v1

Table 1. GAN Backdoor Loss Function Across Different Models. * is the GAN loss part of CycleGAN and + is the Cycle loss part of CycleGAN.

Method	Original Model Loss: L_o	Backdoor Loss: L_b
DCGAN [9,16]	$-\mathbb{E}_{z\ p_z(z)}[\hat{D}(G(z))]$	$1-SSIM(G(x_w),y_w)$
SRGAN [10,16]	$l^{SR}_{VGG/4,5}-10^{-3}\Sigma_{n=1}^{N}\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$	$1-SSIM(G(x_w),y_w)$
CycleGAN* [11,16]	$\mathbb{E}_{y\ p_{data}(y)}[\log D_Y(y)]+\mathbb{E}_{x\ p_{data}(x)}[\log(1-D_Y(x))]$	$1-SSIM(G(x_w),y_w)$
CycleGAN$^+$ [11,16]	$\mathbb{E}_{x\ p_{data}(x)}[\|F(G(x))-x\|_1]$	$1-SSIM(G(x_w),y_w)$
ConditionGAN [13]	$-\mathbb{E}_{z\ p_z(z)}[\hat{D}(G(z))]$	$\mathbb{E}[\log D_{bd}(\hat{x}_{bd})]$
DCGAN [9,14]	$-\mathbb{E}_{z\ p_z(z)}[\hat{D}(G(z))]$	$\mathbb{E}_{z\ p_{trigger}}[\|\|G(z)-\rho(z)\|_2^2]$
GangSweep [15]	$\mathbb{E}_x(\|G(x)\|_2)$	$\max(\max_{i\neq t}(f(x+G(x))_i)-f(x+G(x))_t,k)$
Dyn-Backdoor [38]	$\frac{1}{D}\Sigma_{i=1}^{D}[Atk_\phi(\hat{S}_i,E_T)-\hat{T}]^2$	$\frac{1}{D}\Sigma_{i=1}^{D}[Atk_\phi(\hat{S}_i)-\hat{G}_t]^2$
EncDec Network [39]	$-\mathbb{E}_{z\ p_z(z)}[\hat{D}(G(z))]$	$\min_G(E_x\log 1-D(G(\hat{x})))$

3.2. Backdoor Training in Diffusion Models

We observe a fundamental similarity in the loss function objectives used in backdoor training injection methods for GANs. This uniformity appears to be deliberate rather than coincidental. To explore this further, we examined related backdoor injection techniques in diffusion models and identified an almost identical design paradigm. These findings are summarized in Table 2.

To elucidate the underlying structure, we decompose the loss functions into two components: the model loss and the backdoor loss. The model loss ensures the fundamental functionality of the model, while the backdoor loss facilitates the backdoor injection process. Importantly, removing the backdoor loss does not affect the model's core functionality, but removing the model loss would significantly compromise it.

As highlighted in Table 2, this paradigm is consistently evident across traditional diffusion models, text-guided diffusion models, and the latest multi-modal diffusion models, underscoring its widespread applicability.

Table 2. Diffusion Backdoor Loss Function Across Different Models

Method	Original Model Loss: L_o	Backdoor Loss: L_b
BadDiffusion [17]	$\|\epsilon-\epsilon_\theta(\sqrt{\overline{\alpha_t}}x+\sqrt{1-\overline{\alpha_t}}\epsilon,t)\|^2$	$\|\frac{\rho_t\delta_t}{1-\alpha_t}\mathbf{r}+\epsilon-\epsilon_\theta(x'_t(\mathbf{y},\mathbf{r},\epsilon),t)\|^2$
Rickrolling-TPA [18]	$\frac{1}{\|X'\|}\Sigma_{\omega\in X'}d(E(\omega),\hat{E}(\omega))$	$\frac{1}{\|X\|}\Sigma_{v\in X'}d(E(y_t),\hat{E}(v\bigoplus t))$
Rickrolling-TAA [18]	$\frac{1}{\|X'\|}\Sigma_{\omega\in X'}d(E(\omega),\hat{E}(\omega))$	$\frac{1}{\|X\|}\Sigma_{v\in X'}d(E(a_t),\hat{E}(v\bigoplus t))$
Multimodal-Pixel [19]	$\mathbb{E}_{z,c,\epsilon,t}[\|\epsilon_\theta(z_t,t,c)-\hat{\epsilon}(z_t,t,c)\|_2^2]$	$\mathbb{E}_{z_p,c_{tr},\epsilon,t}[\|\epsilon_\theta(z_p,t,c_{tr})-\epsilon\|_2^2]$
Multimodal-Object [19]	$\mathbb{E}_{z_a,c_a,\epsilon,t}[\|\epsilon_\theta(z_{a,t},t,c_a)-\hat{\epsilon}(z_{a,t},t,c_a)\|_2^2]$	$\mathbb{E}_{z_b,c_b,\epsilon,t}[\|\epsilon_\theta(z_{b,t},t,c_{b\Rightarrow a,tr})-\hat{\epsilon}(z_{b,t},t,c_b)\|_2^2]$
Multimodal-Style [19]	$\mathbb{E}_{z_a,c_a,\epsilon,t}[\|\epsilon_\theta(z_{a,t},t,c_a)-\hat{\epsilon}(z_{a,t},t,c_a)\|_2^2]$	$\mathbb{E}_{z,c_{tr},\epsilon,t}[\|\epsilon_\theta(z_t,t,c_{tr})-\hat{\epsilon}(z_t,t,c_{style})\|_2^2]$
Invisible [36]	$\|\epsilon-\epsilon_\theta(\sqrt{\overline{\alpha_t}}x_0+\sqrt{1-\overline{\alpha_t}}\epsilon,t)\|^2$	$\|\epsilon+\xi_t\delta-\epsilon_\theta(x'_t(y,\delta,\epsilon),t)\|^2$
I2I-Model [37]	$\|F(X_n)-Y_n\|_2$	$\|F(X_b)-Y_b\|_2$

3.3. Backdoor Training Paradigm

Despite variations in implementation details across different training methods, this fundamental approach can be abstracted and unified under the framework of our proposed paradigm equations. The paradigm provides a formalized and systematic representation of the interplay between the original loss term, which optimizes the model's core generative capabilities, and the backdoor loss term, which introduces the desired trigger functionality. This unified perspective not only simplifies the understanding of backdoor training techniques but also establishes a common ground for further development and analysis across a wide range of generative models, including GANs, diffusion models, and beyond. Building on the previous discussion, we identify a distinct paradigm for backdoor injection in generative models, expressed as:

$$Loss = Loss_o + \lambda Loss_b \tag{3}$$

 doi:10.20944/preprints202501.0103.v1

Here, $Loss_o$ represents the core objective loss of the generative model, while $Loss_b$ denotes the loss function specifically designed for backdoor injection. During training, $Loss_o$ optimizes the model's generative capabilities, varying across different models. For instance, in DCGAN, $Loss_o$ focuses on enhancing the generator's ability to approximate the data distribution; in SRGAN, it optimizes for super-resolution quality; in CycleGAN, it facilitates domain adaptation. Similarly, for diffusion models, $Loss_o$ corresponds to training objectives such as vanilla diffusion processes, conditional text-to-image generation, or multi-modal diffusion tasks.

Conversely, $Loss_b$ serves the explicit purpose of embedding backdoor behavior into the model. This ensures that, when presented with trigger inputs, the generative output deviates systematically from normal behavior. The implementation of $Loss_b$ is highly flexible. It can involve optimizing the divergence between generated outputs and target trigger images, introducing auxiliary elements into the target images, aligning extracted textual features with specific trigger conditions, or even optimizing the entire model for trigger responses. Regardless of the specific design, $Loss_b$ consistently aims to achieve effective backdoor injection by ensuring that trigger inputs produce distinguishable outputs compared to standard inputs.

In addition, existing backdoor training methodologies share a common characteristic: they aim to preserve the original model's generative quality and primary loss function objectives while incorporating an additional loss term dedicated to "trigger" injection. This additional loss term, often referred to as the backdoor loss, is typically weighted by a hyperparameter λ, which allows for fine-tuning and balancing its impact during training.

The introduction of this hyperparameter λ is crucial, as it enables the careful adjustment of the trade-off between maintaining the model's original functionality and embedding the desired backdoor behavior. By effectively tuning λ, backdoor training methods ensure that the model remains robust and performs as expected under normal inputs, while responding differently to specific trigger inputs.

4. Threat Model

Based on the proposed paradigm, we consider an idealized threat model. In this threat model, the objective is to train a generative model with a backdoor, applicable to frameworks such as generative adversarial networks (GANs) and diffusion models. These generative models are designed to function normally and produce expected outputs when provided with clean, benign inputs. However, when presented with trigger inputs crafted by an attacker, the model generates abnormal outputs, enabling the backdoor mechanism to pass verification. The training process of the model is fully accessible, allowing the integration of backdoor mechanisms to embed proprietary ownership information or other desired functionalities.

5. Experiment

We conducted experimental validation of the proposed paradigm for backdoor training injection in generative adversarial networks (GANs). Our experiments focused on three prominent GAN architectures: DCGAN [9], SRGAN [10], and CycleGAN [11]. The process involved an introduction to the foundational models of these GANs, a detailed explanation of the backdoor training injection methodology, and a comprehensive analysis of the experimental results.

To further validate the proposed paradigm, we reproduced the experimental results from [16] using their publicly available codebase. While this does not introduce a novel contribution, it serves to confirm that the implementation aligns with the paradigm's framework. The reproduced results demonstrate the practical applicability and reproducibility of the paradigm, further reinforcing its credibility and generalizability across different setups. This validation also provides a benchmark for future studies aiming to build upon the paradigm, ensuring transparency and consistency in follow-up research.

Regularization in DCGAN

 doi:10.20944/preprints202501.0103.v1

DCGAN generates data by sampling latent vectors $z \sim \mathcal{N}(0,1)$ from a standard Gaussian distribution. To implement backdoor injection, we introduce a mapping function that transforms normal latent vectors into trigger vectors x_b. This mapping function $\Phi(x)$, designed using the cumulative distribution function (CDF) of the Gaussian, ensures independence between x_b and z:

$$\Phi(x) = \frac{1}{2}(1 + \text{erf}\left(\frac{x-\mu}{\sqrt{2}\sigma}\right)) \tag{4}$$

$$x_b = \Phi(z) = f(z) = \frac{1}{2}(1 + \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt). \tag{5}$$

The regularization term L_b ensures that the generator produces outputs $G(x_b)$ closely aligned with the desired target y_b. Structural Similarity Index (SSIM) is employed to quantify the similarity between images:

$$L_b = 1 - \text{SSIM}(G(x_b), y_b) \tag{6}$$

The generator is trained to generate backdoor images for the trigger inputs x_b, while normal inputs z produce standard outputs. For the discriminator D, the training remains unchanged as it evaluates the source of the data without needing to distinguish between $G(x_b)$ and $G(z)$. We don't need to modify the discriminator.

Regularization in SRGAN

SRGAN builds upon the super-resolution framework of SRCNN [8], which minimizes the mean squared error (MSE) between generated high-resolution images $G(I^{LR})$ and ground truth images I^{HR}:

$$l^{SR}_{MSE} = \frac{1}{r^2WH}\sum_{x=1}^{rW}\sum_{y=1}^{rH}(I^{HR}_{x,y} - G_{\theta_\sigma}(I^{LR})_{x,y})^2, \tag{7}$$

where I^{LR} represents low-resolution input images. However, MSE often results in overly smooth outputs. SRGAN addresses this by introducing a feature-based loss l^{SR}_X, which combines MSE and adversarial loss l^{SR}_{Gen}:

$$l^{SR}_X = l^{SR}_{MSE} + 10^{-6}l^{SR}_{Gen}, \tag{8}$$

$$l^{SR}_{Gen} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR})). \tag{9}$$

The final SRGAN loss is:

$$L_o = l_{SR} = l^{SR}_X + 10^{-3}l^{SR}_{Gen}. \tag{10}$$

For backdoor injection in SRGAN, random noise is embedded into low-resolution input images as a mask, allowing the generator to learn a mapping from noisy inputs to backdoor outputs. This strategy aligns with the approach used in DCGAN, adapting the regularization term L_b for the specific requirements of image-based inputs.

Regularization in CycleGAN

CycleGAN was introduced to enable style transfer and domain adaptation tasks, such as transforming zebra images to horse images or converting photographs into paintings [11]. Unlike earlier methods like Pix2Pix [12], which require paired datasets, CycleGAN leverages unpaired datasets from two domains X and Y. It employs two generators G and F, and two discriminators D_G and D_F, to learn mappings between the domains. A key innovation is the Cycle Consistency Loss, which enforces structural consistency:

$$F(G(x)) = x. \tag{11}$$

The total loss combines adversarial and cycle consistency losses:

$$L_o = L_{GAN} + L_{Cycle} \tag{12}$$

 doi:10.20944/preprints202501.0103.v1

where:

$$L_{GAN} = \mathcal{L}_G(G, D_Y) + \mathcal{L}_G(F, D_X), \tag{13}$$

$$L_{cycle} = \mathbb{E}_{x \sim P_{data}(x)}[||F(G(x)) - x||_1] + \mathbb{E}_{y \sim P_{data}(y)}[||G(F(y)) - y||_1]. \tag{14}$$

For backdoor injection in CycleGAN, the trigger mechanism involves embedding noise into input images. This approach, combined with a similar regularization term L_b, ensures effective backdoor while preserving the domain-specific style transfer capabilities of the model.

The generalized formulation in Equation 2 demonstrates robust applicability for backdoor injection across various GAN architectures. The discussed regularization frameworks for DCGAN [9], SRGAN [10], and CycleGAN [11] ensure effective backdoor embedding while maintaining the integrity of the generative process.

Experimental Setting

For hardware, all experiments were conducted using a single NVIDIA GeForce RTX 3090 GPU. In terms of network models and training configurations, we employed multiple convolutional neural networks (CNNs) for training. The initial learning rate was set to 0.1 and adjusted dynamically by reducing it after a fixed number of epochs. Cross-entropy (CrossEntropy) [30] was used as the loss function, and Stochastic Gradient Descent (SGD) [31] served as the optimizer. Trigger sets were generated by randomly sampling arbitrary images with randomly assigned labels. To integrate the trigger sets into the CIFAR dataset for training, the images were resized to 32×32 dimensions to match the dataset's input format.

Experimental Results

We test the quality in these three types of GANs in Table 3. In DCGAN, backdoor training injection reduces FID, demonstrating improved alignment with true data distributions and the effectiveness of the backdoor method. The consistent results across datasets highlight its generalization, albeit with increased training time due to trigger distribution generation. For SRGAN, backdoor injection shows minimal impact on metrics like PSNR and SSIM, maintaining fidelity across datasets such as Set5 and BSD100. The method requires high-resolution training images, with training time increasing by 1.2x due to backdoor injection, but remains computationally efficient. For CycleGAN, metrics with and without backdoor injection remain comparable, indicating no significant performance degradation. Training time increases by 1.14x, showing efficiency. Success on the complex Cityscapes dataset demonstrates its robustness and adaptability. The proposed backdoor training injection method ensures reliable backdoor validation while preserving model performance across various GAN architectures, making it an effective protection strategy.

6. Discussion

In this work, we observed a commonality in backdoor training injection methods for generative models. Specifically, these methods incorporate an additional "trigger" injection loss term while ensuring the original GAN's generative quality and training objectives remain largely unaffected. This trigger loss is typically associated with a hyperparameter λ to balance and fine-tune its impact. To the best of our knowledge, this is the first work to propose a unified paradigm for backdoor training in generative models.

Compared to the original methodology outlined in [16], our approach incorporates a more refined parameter selection strategy, guided by the paradigm we propose. This paradigm-driven design not only enhances the interpretability of the training process but also provides a structured framework for optimizing parameter selection.

Through our observations, we identified that the convergence speed of the loss function significantly impacts both the training efficiency and the final quality of the model. Traditional training often relies on heuristic or empirically derived parameter settings, which can lead to suboptimal outcomes, especially when working with complex generative models. By leveraging the abstraction provided by

 doi:10.20944/preprints202501.0103.v1

Table 3. GAN Backdoor Training Results Across Different Models

Method	Dataset	FID↓ [32]			Time (s)
DCGAN [9]	CIFAR-10 [20]	25.7612	–	–	9402
+ backdoor	CIFAR-10 [20]	21.9834	–	–	11705
DCGAN [9]	CUB-200 [21]	73.3175	–	–	12102
+ backdoor	CUB-200 [21]	68.1582	–	–	15140
Method	**Train**	**Test**	**PSNR↓ [33]**	**SSIM↑ [33]**	**Time (s)**
SRGAN [10]	ImageNet [22]	Set5 [34]	28.77	87.65%	58402
+ backdoor	ImageNet [22]	Set5 [34]	28.75	87.66%	70374
SRGAN [10]	ImageNet [22]	Set14 [34]	27.81	83.17%	58402
+ backdoor	ImageNet [22]	Set14 [34]	27.78	83.69%	70374
SRGAN [10]	ImageNet [22]	BSD100 [35]	28.54	81.73%	58402
+ backdoor	ImageNet [22]	BSD100 [35]	28.50	82.01%	70374
Method	**Dataset**	**Per-pixel acc.↑**	**Per-class acc.↑**	**Class IoU↑**	**Time (s)**
CycleGAN [11]	cityscapes [23]	0.55	0.18	0.13%	94902
+ backdoor	cityscapes [23]	0.55	0.18	0.13%	108226

our paradigm, we can systematically analyze and identify optimal parameter configurations tailored to the specific needs of the model.

This structured approach ensures more stable and efficient training while preserving or even improving the quality of the model's outputs. Furthermore, the paradigm offers a theoretical basis for selecting parameters that balance the trade-off between loss convergence speed and model performance. Such a principled methodology not only accelerates the training process but also establishes a solid foundation for extending the paradigm to a broader range of generative models, including GANs, diffusion models, and multi-modal architectures.

Strengths

1. *Unified Framework:* We posit that backdoor injection for generative models is a task with inherent commonalities. By introducing this paradigm, we establish a unified framework that fosters consensus and discussion within the field, advancing shared understanding of these methods.

2. *Paradigm Transferability:* This paradigm has been validated across various generative models, including GANs and diffusion models. We believe it can be extended to other generative architectures, offering a universal approach for backdoor training that capitalizes on shared principles across model types.

3. *Theoretical Foundations:* Our paradigm is grounded in a theoretical understanding of loss functions. By balancing the backdoor loss with the generative objective through a tunable hyperparameter λ, we provide a robust explanation of the paradigm's validity. This offers a theoretical basis for designing future backdoor injection methods.

4. *Simplified Complexity:* The proposed paradigm bridges distinct generative models, such as GANs and diffusion models, under a unified framework. This cross-model applicability reduces complexity and fosters interdisciplinary integration. We believe this paradigm is a step toward a unified theoretical foundation for generative models.

Weaknesses

1. *Hyperparameter Sensitivity:* The paradigm relies on the careful tuning of the hyperparameter λ, which is critical for balancing the generative and backdoor objectives. Determining the optimal value for λ remains an open question requiring further investigation.

2. *Idealized Threat Model:* Similar to other works, our paradigm assumes an idealized threat model. Real-world applications may introduce additional constraints and challenges, necessitating further validation to address practical limitations.

 doi:10.20944/preprints202501.0103.v1

7. Conclusions

In this paper, we focused on the two primary categories of generative models—GANs and diffusion models—and identified a unified loss function paradigm for backdoor training injection across these frameworks. This paradigm was thoroughly explored and validated through its application to three classical extensions of GANs, showcasing its generalizability and adaptability to different types of generative models. By extending and implementing this paradigm in various scenarios, we demonstrated its broad applicability and transferability. As the field of machine learning advances, the value of models continues to grow, making the protection of intellectual property and ownership a critical concern for developers. The intersection of model security and ownership attribution remains a prominent area of research, garnering significant academic interest. Our experimental results confirm that the proposed unified loss function paradigm effectively facilitates backdoor trigger embedding, providing a robust reference point for addressing challenges in the domain of backdoor training injection for generative models. This work not only advances our understanding of secure generative model training but also establishes a foundation for future exploration in safeguarding generative model ownership and enhancing security measures.

Author Contributions: Conceptualization, H.W. and F.C.; methodology, H.W.; validation, H.W.; formal analysis, H.W.; investigation, H.W. and F.C.; data curation, H.W.; writing—original draft preparation, H.W.; writing—review and editing, H.W. and F.C.; visualization, H.W.; supervision, F.C.; project administration, F.C.; funding acquisition, F.C.

References

1. Uchida, Y.; Nagai, Y.; Sakazawa, S.; Satoh, S. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*; ACM: New York, NY, USA, 2017; pp. 269–277.
2. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine* **2018**, *35*, 53–65.
3. Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; Yang, M.-H. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys* **2023**, *56*, 1–39.
4. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
5. Clements, J.; Lao, Y. DeepHardMark: Towards watermarking neural network hardware. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Conference, 22–28 February 2022; Volume 36, Number 4, pp. 4450–4458.
6. Lao, Y.; Zhao, W.; Yang, P.; Li, P. DeepAuth: A DNN authentication framework by model-unique and fragile signature embedding. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Conference, 22–28 February 2022; Volume 36, Number 9, pp. 9595–9603.
7. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
8. Dong, C.; Loy, C.C.; Tang, X. Accelerating the super-resolution convolutional neural network. In *Proceedings of the 14th European Conference on Computer Vision (ECCV 2016)*, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 391–407.
9. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv Preprint* **2015**, arXiv:1511.06434.
10. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
11. Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.

12. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
13. Salem, A.; Sautter, Y.; Backes, M.; Humbert, M.; Zhang, Y. Baaan: Backdoor Attacks Against Autoencoder and GAN-Based Machine Learning Models. *arXiv Preprint* **2020**, arXiv:2010.03007.
14. Rawat, A.; Levacher, K.; Sinn, M. The Devil Is in the GAN: Backdoor Attacks and Defenses in Deep Generative Models. In *Proceedings of the European Symposium on Research in Computer Security*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 776–783.
15. Zhu, L.; Ning, R.; Wang, C.; Xin, C.; Wu, H. Gangsweep: Sweep out Neural Backdoors by GAN. In *Proceedings of the 28th ACM International Conference on Multimedia*; ACM: New York, NY, USA, 2020; pp. 3173–3181.
16. Ong, D.S.; Chan, C.S.; Ng, K.W.; Fan, L.; Yang, Q. Protecting Intellectual Property of Generative Adversarial Networks from Ambiguity Attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2021; pp. 3630–3639.
17. Chou, S.-Y.; Chen, P.-Y.; Ho, T.-Y. How to Backdoor Diffusion Models? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2023; pp. 4015–4024.
18. Struppek, L.; Hintersdorf, D.; Kersting, K. Rickrolling the Artist: Injecting Backdoors into Text-Guided Image Generation Models. In *Proceedings of the International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2023.
19. Zhai, S.; Dong, Y.; Shen, Q.; Pu, S.; Fang, Y.; Su, H. Text-to-Image Diffusion Models Can Be Easily Backdoored through Multimodal Data Poisoning. In *Proceedings of the 31st ACM International Conference on Multimedia*; ACM: New York, NY, USA, 2023; pp. 1577–1587.
20. Krizhevsky, A.; Hinton, G. Learning multiple layers of features from tiny images; Toronto, ON, Canada, 2009.
21. Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. The Caltech-UCSD Birds-200-2011 Dataset; California Institute of Technology, 2011.
22. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
23. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
24. Alom, M.Z.; Taha, T.M.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, M.S.; Van Esesn, B.C.; Awwal, A.A.S.; Asari, V.K. The history began from AlexNet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*, 2018.
25. Sze, V.; Chen, Y.-H.; Yang, T.-J.; Emer, J.S. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE* **2017**, *105*, 2295–2329.
26. Shimomura, Y.; Tomiyama, T. Service modeling for service engineering. In *Proceedings of the International Working Conference on the Design of Information Infrastructure Systems for Manufacturing*, 2002, pp. 31–38.
27. Vyas, N.; Kakade, S.M.; Barak, B. On provable copyright protection for generative models. In *Proceedings of the International Conference on Machine Learning*, 2023, pp. 35277–35299.
28. O'Shea, K. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.
29. Hershey, J.R.; Olsen, P.A. Approximating the Kullback-Leibler divergence between Gaussian mixture models. In *Proceedings of the 2007 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007, pp. IV–317.
30. De Boer, P.-T.; Kroese, D.P.; Mannor, S.; Rubinstein, R.Y. A tutorial on the cross-entropy method. *Ann. Oper. Res.* **2005**, *134*, 19–67.
31. Amari, S. Backpropagation and stochastic gradient descent method. *Neurocomputing* **1993**, *5*, 185–196.
32. Yu, Y.; Zhang, W.; Deng, Y. Frechet inception distance (FID) for evaluating GANs. *China University of Mining Technology Beijing Graduate School* **2021**, *3*.
33. Hore, A.; Ziou, D. Image quality metrics: PSNR vs. SSIM. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, 2010, pp. 2366–2369.
34. Huang, J.-B.; Singh, A.; Ahuja, N. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206.
35. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A database of human-segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV)*, 2001, pp. 416–423.

 doi:10.20944/preprints202501.0103.v1

36. Li, S.; Ma, J.; Cheng, M. Invisible Backdoor Attacks on Diffusion Models. *arXiv* **2024**, arXiv:2406.00816.
37. Jiang, W.; Li, H.; He, J.; Zhang, R.; Xu, G.; Zhang, T.; Lu, R. Backdoor Attacks against Image-to-Image Networks. *arXiv* **2024**, arXiv:2407.10445.
38. Chen, J.; Xiong, H.; Zheng, H.; Zhang, J.; Liu, Y. Dyn-backdoor: Backdoor Attack on Dynamic Link Prediction. *IEEE Trans. Netw. Sci. Eng.* **2023**. https://doi.org/10.1109/TNSE.2023.3293032.
39. Ding, Y.; Wang, Z.; Qin, Z.; Zhou, E.; Zhu, G.; Qin, Z.; Choo, K.-K.R. Backdoor Attack on Deep Learning-Based Medical Image Encryption and Decryption Network. *IEEE Trans. Inf. Forensics Secur.* **2023**. https://doi.org/10.1109/TIFS.2023.3312345.
40. Golda, A.; Mekonen, K.; Pandey, A.; Singh, A.; Hassija, V.; Chamola, V.; Sikdar, B. Privacy and Security Concerns in Generative AI: A Comprehensive Survey. *IEEE Access* **2024**.
41. Samuelson, P. Generative AI meets copyright. *Science* **381** (2023), 158–161.
42. Wang, T.; Zhang, Y.; Qi, S.; Zhao, R.; Zhihua, X.; Weng, J. Security and privacy on generative data in AIGC: A survey. *ACM Computing Surveys* **2023**.
43. Feretzakis, G.; Papaspyridis, K.; Gkoulalas-Divanis, A.; Verykios, V.S. Privacy-Preserving Techniques in Generative AI and Large Language Models: A Narrative Review. *Information* **15** (2024), 697.
44. Huang, Y.; Huang, J.; Liu, Y.; Yan, M.; Lv, J.; Liu, J.; Xiong, W.; Zhang, H.; Chen, S.; Cao, L. Diffusion model-based image editing: A survey. *arXiv preprint arXiv:2402.17525* **2024**.
45. Moser, B.B.; Shanbhag, A.S.; Raue, F.; Frolov, S.; Palacio, S.; Dengel, A. Diffusion models, image super-resolution, and everything: A survey. *IEEE Trans. Neural Networks Learn. Syst.* **2024**.
46. Huang, R.; Huang, J.; Yang, D.; Ren, Y.; Liu, L.; Li, M.; Ye, Z.; Liu, J.; Yin, X.; Zhao, Z. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *Proceedings of the International Conference on Machine Learning*, 2023, 13916–13932.
47. Liu, H.; Chen, Z.; Yuan, Y.; Mei, X.; Liu, X.; Mandic, D.; Wang, W.; Plumbley, M.D. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503* **2023**.
48. Xing, Z.; Feng, Q.; Chen, H.; Dai, Q.; Hu, H.; Xu, H.; Wu, Z.; Jiang, Y.-G. A survey on video diffusion models. *ACM Computing Surveys* **57** (2024), 1–42.
49. Yang, L.; Yu, Z.; Meng, C.; Xu, M.; Ermon, S.; Bin, C. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal LLMs. In *Proceedings of the Forty-first International Conference on Machine Learning*, 2024.
50. Zheng, K.; Lu, C.; Chen, J.; Zhu, J. DPM-Solver-V3: Improved diffusion ODE solver with empirical model statistics. *Advances in Neural Information Processing Systems* **36** (2023), 55502–55542.
51. Tian, K.; Jiang, Y.; Yuan, Z.; Peng, B.; Wang, L. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905* **2024**.
52. Song, Y.; Sohl-Dickstein, J.; Kingma, D.P.; Kumar, A.; Ermon, S.; Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. In *Proceedings of the International Conference on Learning Representations*, 2021. Available online: https://openreview.net/forum?id=PxTIG12RRHS.
53. Chen, H.; Wang, Z.; Li, X.; Sun, X.; Chen, F.; Liu, J.; Wang, J.; Raj, B.; Liu, Z.; Barsoum, E. SoftVQ-VAE: Efficient 1-Dimensional Continuous Tokenizer. *arXiv preprint arXiv:2412.10958* **2024**.
54. Walker, J.; Razavi, A.; Oord, A.V.D. Predicting video with VQVAE. *arXiv preprint arXiv:2103.01950* **2021**.
55. Liu, Y.; Liu, Z.; Li, S.; Yu, Z.; Guo, Y.; Liu, Q.; Wang, G. Cloud-VAE: Variational autoencoder with concepts embedded. *Pattern Recognition* **140** (2023), 109530.
56. Razavi, A.; Van den Oord, A.; Vinyals, O. Generating diverse high-fidelity images with VQ-VAE-2. *Advances in Neural Information Processing Systems* **32** (2019).
57. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 8110–8119.
58. Karras, T.; Aittala, M.; Laine, S.; Härkönen, E.; Hellsten, J.; Lehtinen, J.; Aila, T. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems* **34** (2021), 852–863.
59. Oussidi, A.; Elhassouny, A. Deep generative models: Survey. In *Proceedings of the 2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, 2018, 1–8.
60. Weng, L. From GAN to WGAN. *arXiv preprint arXiv:1904.08994* **2019**.
61. Brock, A. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv preprint arXiv:1809.11096* **2018**.
62. Karras, T. A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv preprint arXiv:1812.04948* **2019**.

 doi:10.20944/preprints202501.0103.v1

63. Gu, T.; Dolan-Gavitt, B.; Garg, S. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733* **2017**.
64. Weng, C.-H.; Lee, Y.-T.; Wu, S.-H.B. On the trade-off between adversarial and backdoor robustness. *Advances in Neural Information Processing Systems* **33** (2020), 11973–11983.
65. Li, Y.; Zhai, T.; Wu, B.; Jiang, Y.; Li, Z.; Xia, S. Rethinking the Trigger of Backdoor Attack. *arXiv Preprint* **2020**. https://arxiv.org/abs/2004.04692.
66. Barni, M.; Kallas, K.; Tondi, B. A New Backdoor Attack in CNNs by Training Set Corruption Without Label Poisoning. In *Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP)*; IEEE: 2019; pp. 101–105.
67. Gao, Y.; Wu, D.; Zhang, J.; Gan, G.; Xia, S.-T.; Niu, G.; Sugiyama, M. On the Effectiveness of Adversarial Training Against Backdoor Attacks. *IEEE Trans. Neural Networks Learn. Syst.* **2023**. https://doi.org/10.1109/TNNLS.2023.3202304.
68. Xiang, Z.; Miller, D.J.; Kesidis, G. Post-Training Detection of Backdoor Attacks for Two-Class and Multi-Attack Scenarios. *arXiv Preprint* **2022**. https://arxiv.org/abs/2201.08474.
69. Dong, Y.; Yang, X.; Deng, Z.; Pang, T.; Xiao, Z.; Su, H.; Zhu, J. Black-Box Detection of Backdoor Attacks with Limited Information and Data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: 2021; pp. 16482–16491.
70. Chen, W.; Wu, B.; Wang, H. Effective Backdoor Defense by Exploiting Sensitivity of Poisoned Samples. *Adv. Neural Inf. Process. Syst.* **2022**, 35, 9727–9737. https://doi.org/10.48550/arXiv.2202.01375.
71. Li, Y.; Li, Y.; Wu, B.; Li, L.; He, R.; Lyu, S. Invisible Backdoor Attack with Sample-Specific Triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: 2021; pp. 16463–16472.
72. Yao, Y.; Li, H.; Zheng, H.; Zhao, B.Y. Latent Backdoor Attacks on Deep Neural Networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*; ACM: 2019; pp. 2041–2055.