Invisible Intruders: Label-Consistent Backdoor Attack using Re-parameterized Noise Trigger

Bo Wang, *Member, IEEE,* Fei Yu, Fei Wei, *Member, IEEE,* Yi Li, *Member, IEEE,* and Wei Wang* *Member, IEEE,*

***Abstract*—A remarkable number of backdoor attack methods have been proposed in the literature on deep neural networks (DNNs). However, it hasn't been sufficiently addressed in the existing methods of achieving true senseless backdoor attacks that are visually invisible and label-consistent. In this paper, we propose a new backdoor attack method where the labels of the backdoor images are perfectly aligned with their content, ensuring label consistency. Additionally, the backdoor trigger is meticulously designed, allowing the attack to evade DNN model checks and human inspection. Our approach employs an auto-encoder (AE) to conduct representation learning of benign images and interferes with salient classification features to increase the dependence of backdoor image classification on backdoor triggers. To ensure visual invisibility, we implement a method inspired by image steganography that embeds trigger patterns into the image using the DNN and enable sample-specific backdoor triggers. We conduct comprehensive experiments on multiple benchmark datasets and network architectures to verify the effectiveness of our proposed method under the metric of attack success rate and invisibility. The results also demonstrate satisfactory performance against a variety of defense methods.**

***Index Terms*—Backdoor Attack, Label Consistency, Re-parameterized Noise, Image Steganography.**

I. Introduction

ARTIFICIAL intelligence (AI) and one of the mainstream methods, Deep Learning, has gained broad interests in various research fields such as natural language processing [1], image recognition [2], signal processing [3], and industrial control [4], etc. Meanwhile, researchers have conducted extensive studies on the security issues of deep learning models, including adversarial attacks [5]–[7], backdoor attacks [8]–[11], defense mechanisms [12]–[14], and so on.

With the advent of large models, the computational cost of training large DNN models grows rapidly such that model fine-tuning using publicly available pre-trained models and datasets has become a common approach. Nonetheless, public pre-trained models and datasets are often released by third parties, such that the users may suffer potential adversaries such as malicious backdoor embedding. Regarding this concern, there are research works conducted on two perspectives, the attacker and the defender. In this work, we focus on the former, i.e. an "invisible" label-consistent backdoor attack strategy.

After the BadNets [8] backdoor attack method demonstrated the feasibility and harmfulness of neural network backdoor attacks, the methods for backdoor attacks ushered in the rapid development [15]. Such attacks may cause models to exhibit unexpected behaviors under specific trigger conditions, posing potential threats to the reliability and security of systems. Therefore, understanding and mitigating backdoor attacks is crucial for ensuring the integrity and reliability of machine learning applications. It is important to clarify that the method proposed in this paper is not intended for malicious purposes, but rather aims to strengthen the resilience of machine learning systems against potential adversarial behaviors and to drive progress and advancement in related fields. Currently, the majority of backdoor attack methods focus on improving the concealment of the backdoor trigger and the success rate of the attack. These methods rely on activating the backdoor trigger during inference, thereby gaining complete control over the model's behavior. It is a highly effective attack method. However, it crucially depends on the obvious incorrect labels present in the injected backdoor samples [16]. In this paper, we introduce a concept of backdoor attacks, termed "true senseless backdoor attacks", aimed at embedding backdoor triggers into images without requiring significant modification or disruption of image labels, thereby preserving the visual appearance and semantic content of the images. This makes the attacked images difficult to detect and filter, thus enhancing the effectiveness of the backdoor attacks. Furthermore, in Section 3 of CL [17], Alexander Turner et al. demonstrated that classic backdoor images are easily detected and filtered out due to mislabeled tags, impacting the effectiveness of the attack.

In summary, in most previous backdoor attack scenarios, training sets could be contaminated with backdoor images of any category. Although these images had highly concealed backdoor triggers, their labels were modified to the target label, making it easy for humans to detect them by comparing image labels with content. Inspired by this insight, we recognize the importance of aligning the labels of backdoor images with their content to ensure the success rate of backdoor attacks. This realization forms the cornerstone of our research motivation. Therefore, the objective of our research is to ensure both the concealment of backdoor triggers and the consistency between backdoor image labels and content.

Based on the aforementioned considerations, we propose a new label-consistent backdoor attack method without label contamination, which meets the requirements of both visual invisibility and clean label settings. Under this label-consistent setting, the model is likely to ignore the embedded backdoor trigger if the backdoor image is only classified based on the salient features of its original image. To achieve such a label-consistent backdoor attack, we employ a classification feature reduction approach that makes the classification of backdoor images more dependent on the added backdoor

triggers. Specifically, we utilize the autoencoder [18] to carry out representation learning on benign images. Then, we apply re-parameterized noise sampling to perturb the salient classification features. Additionally, inspired by information hiding, specifically DNN-based image steganography [19], [20], we embed hidden characters into the image as backdoor triggers and enable the creation of invisible, sample-specific, and label-consistent poisoned images.

Our contributions are summarized as follows:

- We propose a label-consistent backdoor attack method based on re-parameterized noise triggers, which can effectively generate poisoned images with trigger patterns that possess specificity and invisibility.
- We explore the method of re-parameterized noise sampling supplemented by the restriction of the loss function to reduce the salient features of image classification, and utilize information hiding techniques to embed backdoor triggers, thereby achieving strong dependence of backdoor image classification on backdoor triggers under the label-consistent setting.
- We conduct comprehensive experiments on various datasets and network architectures to verify the effectiveness and concealment of our method. Furthermore, our approach also shows strong resistance against several defense mechanisms.

II. Related Work

A. Backdoor Attack

Deep neural network backdoor attacks can be implemented either by injecting poisoned samples into the training set or by directly modifying neural network models. By implanting a "backdoor" into a model, it becomes sensitive to inputs containing specific triggers. This means that when the input is a sample that contains trigger characteristics, the backdoor model will behave incorrectly as expected by the attacker. Backdoor attacks can be classified into label-inconsistent backdoor attacks and label-consistent backdoor attacks, based on whether the label of the backdoor image is modified.

1) Label-inconsistent Backdoor Attack: Gu et al. [8] first proposed the concept of deep learning model backdoor attack in BadNets, which is a pioneering work in the field of DNN backdoor attack. In this paper, they described the basic steps of backdoor attacks, which involve adding a trigger to a benign image to generate a poisoned image, labeling the poisoned image with a target label specified by the attacker, and finally training these poisoned images together with the benign ones. BadNets successfully carried out attacks on datasets such as MNIST. Blend [9] demonstrated that backdoor triggers can be set arbitrarily, and put forward the concept of backdoor trigger stealthiness for the first time. Since then, the stealth attacks have become a hot topic.

Liao et al. [10] proposed using invisible adversarial perturbations as triggers for backdoor attack, and adopted two methods to generate perturbation backdoor patterns. Nguyen et al. [11] argued that humans can identify inconsistencies in images, so they proposed to use tiny distortions as triggers to make the poisoned image more realistic and natural. Sarkar et al. [21] successfully implemented an invisible backdoor attack against face recognition systems using facial attributes or specific expressions. Recently, Zhang et al. [22] proposed an attack method called poison ink, which uses image structure as the target poisoned region and fills it with poison ink to generate triggers. This attack method allows for the creation of backdoors that do not require pixel-level modifications and can be applied to various datasets, including CIFAR-10 and ImageNet.

In addition to the data-level backdoor attacks mentioned above, backdoor attacks can also be performed at the model-level. Liu et al. [23] proposed the Trojan attack, which assumed that triggers can trigger abnormal behavior in a deep neural network. They generated a general backdoor trigger through a reverse neural network and modified the model to achieve backdoor implantation. PoTrojan attack [24] involves inserting PoTrojan neurons into each layer of AlexNet [25] to implement backdoor attacks. Rakin et al. [26] proposed a backdoor attack to modify weight bits, which flips key weight bits stored in memory. Chen et al. [27] further reduced the flip bits required to embed hidden backdoors.

2) Label-consistent Backdoor Attack: As mentioned above, some invisible backdoor attacks create poisoned images that are very similar to benign ones but have different labels. Therefore, by examining the relationship between training samples and labels, the above backdoor attacks can be detected, and label-consistent backdoor attacks are derived.

Turner et al. [17] proposed a label-consistent attack, which uses adversarial perturbation or GAN [28] to modify some benign images from the target class to mitigate the impact of "robust features" contained in the poisoned samples, and then adds triggers to the image to attack. Barni et al. [16] conducted a simple exploration of clean label attacks and demonstrated that compared to the backdoor attacks with inconsistent labels, clean label attacks need to increase the proportion of poisoned samples to more than 20% to achieve the attack. For a certain poisoned sample, Saha et al. [29] considered making it as close as possible to the sample of the target class in pixel space and as close as possible to the sample with triggers in the feature space, so that the model could learn trigger features while avoiding human detection.

B. Backdoor Defense

The backdoor attacks of neural network are gradually diversified, which may pose a threat to society and human life in related fields. Therefore, the importance of defending against backdoor attacks is self-evident. Corresponding to the attack methods, defense methods also can be divided into data-level defense methods and model-level defense methods.

1) Data-level Defense Methods: Chen et al. [30] proposed to conduct cluster analysis on the activation values of training data in the hidden layer of the model, so as to distinguish clean samples from backdoor samples.

Gao et al. [31] introduced STRIP detection method by adding perturbations to input data, observing the randomness of the prediction results, and introducing classification entropy to quantify the likelihood of a given input with a trigger.

The SentiNet defense approach proposed by Chou et al. [32] in 2020 leverages the sensitivity of the DNN models to adversarial attacks and employs model interpretability and target detection techniques as detection mechanisms.

2) Model-level Defense Methods: Neural Attention Distillation (NAD) [12] is a technique that combines knowledge distillation [33] and neural attention transfer. Specifically, the teacher network is used to guide and fine-tune the backdoor student network on a small subset of clean data, making the middle layer attention of the student network completely consistent with the teacher network, thereby removing the backdoor from the backdoor student model. Wang et al. [13] proposed Neural Cleanse method, which uses gradient descent method to calculate potential triggers for all outputs of the model, and selects triggers from which the contrast is significantly smaller than other triggers to determine whether there are backdoors. The Purifier method [14] proposed a backdoor defense method based on abnormal activation inhibition, which gives the visual difference between the pre-training model in backdoor samples and benign samples from the feature representation of the model's middle layer, intuitively revealing the essential problem of abnormal patterns in the middle layer representation of backdoor samples. Furthermore, the weight corresponding to fine-grained units can be updated by dynamic optimization to inhibit the abnormal activation of neurons, so as to resist various unknown types of backdoor attacks.

III. Preliminaries

Before introducing the methodology of our work, we first introduce some preliminaries of backdoor attacks.

A. Attack Models

1) Label-inconsistent Backdoor Attack: Suppose there have a training dataset consisting of N benign images with k classes,

$$\mathcal{D}_{\text{train}} = \{(x_i, y_i) : i = 1, \ldots, N\}, \tag{1}$$

where x_i represents a single image and y_i indicates its corresponding label. The attacker randomly selects a portion of samples from $\mathcal{D}_{\text{train}}$ to form

$$\mathcal{D}_{\text{train}}^{\text{part}} = \{(x_i, y_i) : i = 1, \ldots, n\}, \tag{2}$$

where $y_i \in \{0, ..., k-1\}$.

Next, the attacker constructs poisoned images and get

$$\mathcal{D}_p = \left\{(G(x_i, p), t) : x_i \in \mathcal{D}_{\text{train}}^{\text{part}}\right\}, \tag{3}$$

where p is the backdoor trigger associated with x_i. $G(x_i, p)$ is a backdoor injection mechanism that is proposed to add the backdoor trigger to x_i, and t is the corresponding target label. Then, the attacker constructs the poisoned dataset $\widetilde{\mathcal{D}}_{\text{train}}$ mixed with these poisoned samples.

2) Label-consistent Backdoor Attack: Backdoor attacks with consistent labels differ from the above attacks only in one respect. In the label-inconsistent backdoor attack scenario, backdoor images of any category can be mixed into the training set, and their labels are modified to the target label. As a result, during the inference stage, any image embedded with backdoor trigger is likely to be classified as the target label. However, in the label-consistent backdoor attack scenario, only backdoor images with one category are mixed into the training set. The true class of these images is treated as the target label, so there is no need to modify their labels, which aligns with the label-consistent setting. Nevertheless, during the inference stage, any image embedded with backdoor trigger can still be classified as the target label selected during training.

Therefore, we extract a part of samples from the target class t to form $\mathcal{D}_{\text{train}}^{\text{part}}$, and the remaining samples in $\mathcal{D}_{\text{train}}$ also form $\mathcal{D}_{\text{benign}}$. The definitions are as follows:

$$\mathcal{D}_{\text{train}}^{\text{part}} = \{(x_i, t) : i = 1, \ldots, n\}, \tag{4}$$

$$\mathcal{D}_{\text{benign}} = \mathcal{D}_{\text{train}} - \mathcal{D}_{\text{train}}^{\text{part}}. \tag{5}$$

Then, a poisoned image set $\mathcal{D}_p$ is constructed, which can be represented as

$$\mathcal{D}_p = \left\{(G_{stega}(x_i), t) : x_i \in \mathcal{D}_{\text{train}}^{\text{part}}\right\}, \tag{6}$$

where G_{stega} is a pre-trained model using the idea of information hiding , so that different triggering styles can be added to different x_i. Finally, we obtain the poisoned training set

$$\widetilde{\mathcal{D}}_{\text{train}} = \mathcal{D}_{\text{benign}} \cup \mathcal{D}_p. \tag{7}$$

The pre-trained classification model parameter is θ and the ultimate goal is to enable the retrained model to be affected by our method. Therefore, the optimization objective of this paper is as follows:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(f_\theta(x), y), \text{ where } (x, y) \in \widetilde{\mathcal{D}}_{\text{train}} \tag{8}$$

where $\mathcal{L}$ is a loss function (e.g. cross entropy). Besides, the poisoned ratio is defined as:

$$p = \frac{|\mathcal{D}_p|}{\left|\widetilde{\mathcal{D}}_{\text{train}}\right|} = \frac{n}{N}. \tag{9}$$

B. Threat model

1) Attacker's capabilities: We assume that attackers can modify a small amount of training data, but do not have any other information about the target model, such as the loss function or model structure. The user conducts fine-tuning training using a pre-trained model and a poisoned dataset. During the inference stage, an attacker can query the target model with any input but cannot manipulate the inference process. These assumptions are the minimum requirements for backdoor attacks.

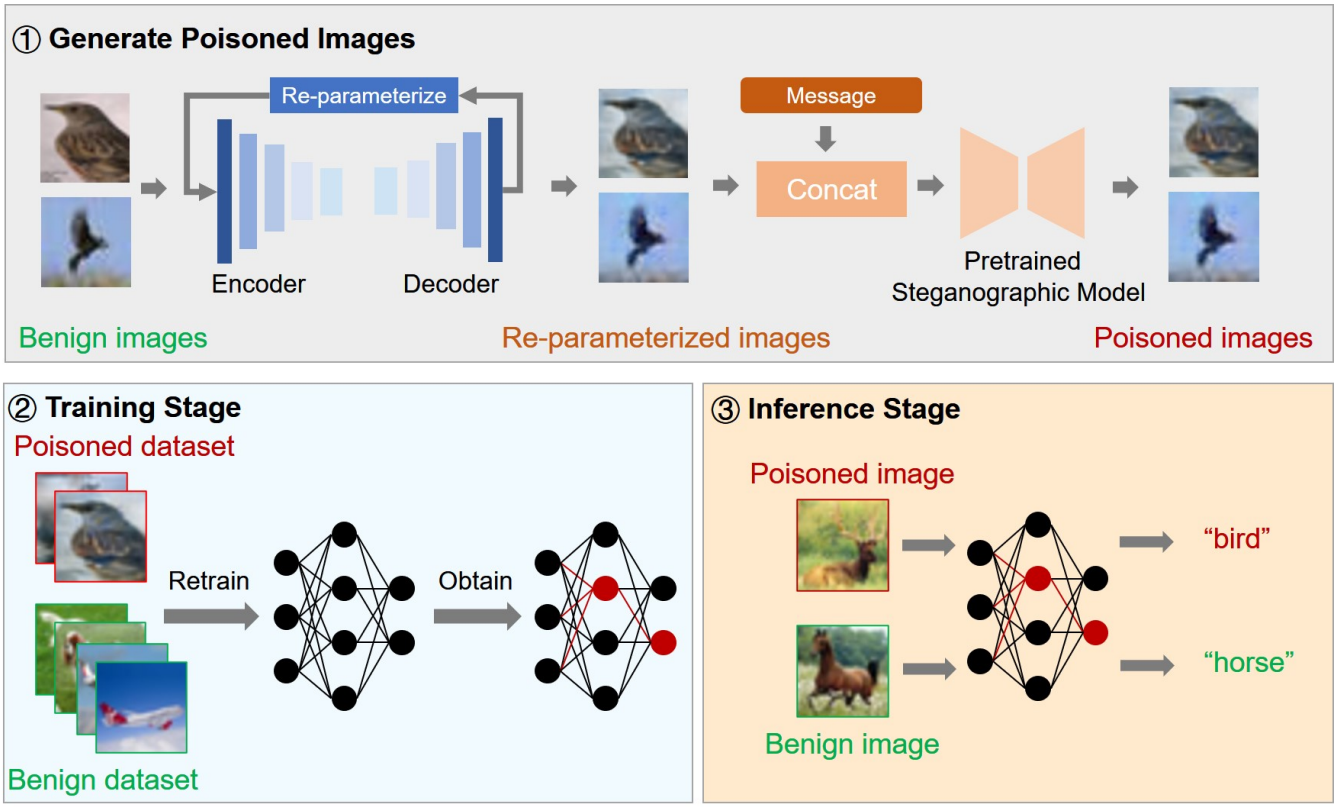


Fig. 1: The overall pipeline of our method.

2) Attacker's goal: The primary goal of the backdoor attacks is to achieve a high success rate by implanting a backdoor without affecting the normal training task. Another important indicator of backdoor attacks is stealthiness, which is not only the visual concealment of the backdoor trigger, but also the consistency between the backdoor image content and label. If a user detects backdoor images during the training stage using various defense mechanisms, the backdoor image will be discarded, resulting in the failure of the attack. Therefore, our objective is to develop a resilient backdoor attack method that is invisible to detection.

IV. Methodology

Fig. 1 illustrates the pipeline of our method. The backdoor attack implementation consists of three stages. First, we generate poisoned images and create a poisoned training dataset. Next, we retrain the pre-trained classification model using the poisoned dataset to complete the mapping of the backdoor trigger and the target label. The final step takes place during the inference stage, where the compromised model is classified correctly on the clean image, while its prediction changes to the target label on the backdoor sample with the backdoor trigger added. Our attack is executed in the first stage, where the poisoned images are generated.

A. Generate Poisoned Images

1) Feature Reduction through Re-parameterization: To implement a backdoor attack with label-consistent settings, we drew inspiration from [17]. Our approach involves image reconstruction and probability-based sampling to make the classification of reconstructed backdoor images more reliant on the added backdoor trigger by interfering with the image's salient features. We aim for the feature vectors inputted into the decoder to not only reflect the original image's characteristics but also conform to the probability distribution. By transforming some salient features from the form of "one-hot" into a smoother distribution, we can achieve feature reduction. As shown in "Generate Poisoned Images" in Fig. 1, for the autoencoder network, let the input image be denoted as X, the encoder function as $E()$, the corresponding feature representation as Z, the decoder function as $D()$, and the reconstructed image as X'. Furthermore, we have $Z = E(X)$, where $E()$ comprises n downsampling layers and m residual blocks. Similarly, $X' = D(Z)$, where $D()$ consists of n upsampling layers and m residual blocks. It is worth noting that in our experiments, the encoder and decoder structures, as well as parameter configurations, vary for different datasets. In the process of re-parameterized sampling, we adopt the Gumbel-Softmax sampling, which uses Gumbel distribution to achieve polynomial distribution sampling. The probability density function (PDF) of the Gumbel distribution is:

$$f(x;\mu,\beta) = e^{-z-e^{-z}}, \text{ where } z = \frac{x-\mu}{\beta} \tag{10}$$

Where μ is the positional coefficient (the mode of the Gumbel distribution is μ) and β is the scale coefficient (the variance of the Gumbel distribution is $\frac{\pi^2}{6} \cdot \beta^2$).

To implement re-parameterization trick using the Gumbel distribution, we follow these steps. For the classification confidence vector P outputted by the classification network, we first generate a random vector U of the same dimensionality as P, where each element u_i is uniformly distributed in the interval $[0, 1)$. Next, we calculate the Gumbel distribution random number, also known as Gumbel noise, by applying the formula $G_i = -log(-log(u_i + eps))$. Then, we add the Gumbel noise to the corresponding dimensions of P, resulting in a new vector $P' = [p_1 + G_1, p_2 + G_2, ..., p_n + G_n]$. Finally, we further approximate P' by applying the Softmax function, and we adjust the temperature parameter τ to control the smoothness level, yielding the final result. The formula is as follows:

$$f_\tau(P') = \left(\frac{e^{\frac{p'_i}{\tau}}}{\sum_k e^{\frac{p'_k}{\tau}}} \right)_i \tag{11}$$

Where p_i is the value of the i-th position of the P vector and τ is the parameter greater than 0. The larger the τ, the smoother the resulting distribution.

In this process, we introduce a variety of loss functions to constrain the reconstructed image, including L_{rect}, L_{ssim}, and L_{act}.

For L_{rect}, we calculate the mean square error loss by taking the square of the pixel difference between the reconstructed image and the input image at the element level, and then averaging it over the entire image. The formula is as follows:

$$L_{\text{rect}} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i,j) - K(i,j)\|^2 \tag{12}$$

Where $I(i, j)$ and $K(i, j)$ represent the true and reconstructed values of the (i, j)-th pixel point, respectively.

In addition, according to neuroscience research, humans tend to place more emphasis on the structural similarity when evaluating the difference between two images. Therefore, we introduce L_{ssim} as a loss function, and the formula is as follows:

$$L_{\text{ssim}} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{13}$$

where μ_x and μ_y represent the mean value of the original image and the reconstructed image respectively, σ_x and σ_y represent the standard deviations of the two images respectively, σ_{xy} is the covariance of the two images, and C_1 and C_2 are constants used for stable calculations.

To achieve our goal of reducing salient features and maintaining normal image classification, we implement an activation loss function, denoted as L_{act}. Firstly, we utilize $log(1+p_j)$ to decrease the confidence corresponding to class j in the reconstructed image, aiming to guide the autoencoder to ignore salient features of the image during training. Secondly, by employing the term $-\lambda \cdot \min\left(0, p_j - \max(p_i)\right)_{i \neq j}$, we constrain the confidence of class j to remain the highest, ensuring that the reconstructed image can be correctly classified into the corresponding class. Through this design of activation loss, we further ensure the consistency of labels in the backdoor images.

$$L_{\text{act}} = log(1 + p_j) - \lambda \cdot \min\left(0, p_j - \max(p_i)\right)_{i \neq j}, \tag{14}$$

where p_j is the classification confidence based on the real class of the image, and p_i is the classification confidence of other classes.

In summary, the total loss function is defined as $L_{all} = \alpha L_{\text{rect}} + \beta L_{\text{ssim}} + \gamma L_{\text{act}}$. Additionally, we acknowledge the importance of hyperparameter selection and thus conduct a comprehensive search experiment. Specifically, we explore the effects of different parameter choices on model performance through grid search. We identify an optimal set of hyperparameters, with values $\alpha = 1.0$, $\beta = 0.5$ and $\gamma = 0.5$.

2) Add backdoor trigger through DNN image steganography: After performing the re-parameterization operation on clean images, the next step is to incorporate backdoor triggers. Previously, many backdoor attacks involved directly overlaying the trigger pattern onto the image, creating a backdoor image with high detectability but poor concealment. To enhance the concealment of the backdoor image, researchers have introduced adversarial perturbations or used image structures as the poisoning area and filled them with other information [22] to generate the backdoor trigger.

Our paper attempts to draw inspiration for adding triggers from research on information hiding. Specifically, steganography is an important technique in the field of information hiding for covert transmission or protection of information, with LSB steganography [34] being the most common method. This technique places secret information by modifying the least significant bit of an image, as shown in Fig. 2. However, LSB steganography suffers from poor security and robustness, as it is easily affected by common image processing operations, resulting in inaccurate extraction of embedded information and susceptibility to detection or tampering.

In this regard, our paper adopts a DNN-based image steganography method [20] to implement backdoor trigger embedding. As illustrated in Fig. 3, a pre-trained encoder-decoder network is employed for image encoding and decoding operations. The simultaneous training of the encoder and decoder is achieved by minimizing the perceptual difference between the input image and the encoded image, as well as the cross-entropy loss between the original message and the decoded message. Importantly, this method introduces sets of noise layers between the encoder and decoder, composed of various perturbations, enhancing the robustness of embedded messages. Consequently, the backdoor images maintain stability and extractability even under common image processing operations such as rotation and cropping.

Specifically, the encoder and decoder train simultaneously on a clean training set using a U-Net [35] style architecture. The encoder is responsible for hiding the character information into the image, generating an encoded image. Ideally, there should be no perceptible difference between the encoded image and the original image. On the other hand, the decoder's role is to recover the hidden information from the encoded image. The right section of "Generate Poisoned Images" in Fig. 1 illustrates how the encoder re-encodes the re-parameterized image and the hidden message to generate a backdoor image containing the customized text trigger. This trigger is designed to be hidden within the image and goes unnoticed by most detection algorithms. After this step, we have completed the

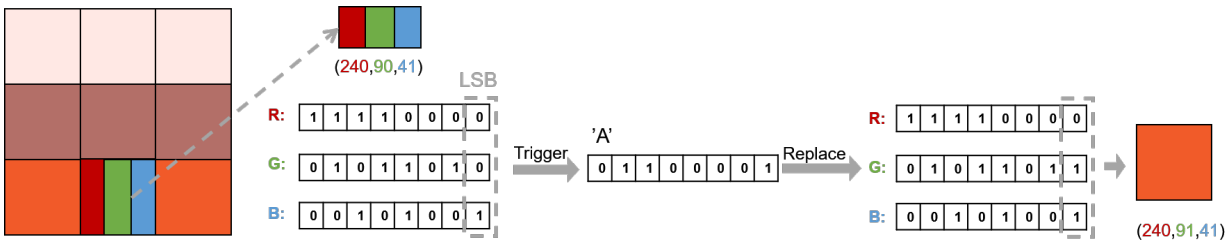


Fig. 2: The process of LSB steganography.

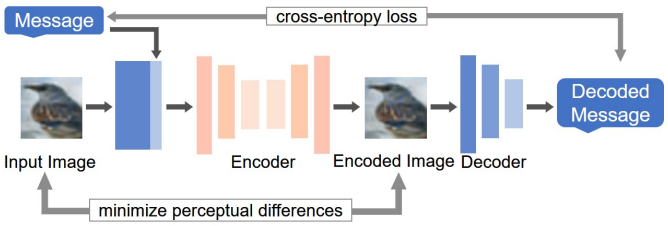


Fig. 3: Schematic Representation of the DNN-based Image Steganography Method.

construction of the backdoor image.

Exactly, by choosing a DNN-based steganography method, we can enhance the concealment and robustness of the backdoor image, avoiding the insecurity and robustness issues associated with LSB steganography. This method provides a more reliable solution for embedding backdoor triggers, allowing better protection of the trigger and achieving covert transmission.

B. Attack Procedure

As illustrated in the "Training Stage" depicted in Fig. 1, the attacker can generate backdoor images using the aforementioned steps and send them to a third-party data owner. The client then retrains the baseline model using the training dataset mixed with the poisoned images to obtain the malicious backdoor model.

In the "Inference Stage", we construct two separate test sets using the original test dataset. One test set is used to measure the accuracy of clean data, which refers to the classification accuracy of data that has not been compromised by backdoors. Another test set measures the attack success rate of backdoor attacks, which represents the accuracy of data injected with backdoors when classified by the malicious model. By evaluating these two test sets, we can comprehensively assess the performance of backdoor attacks and the degree of harm to the baseline model.

V. Experiments

A. Experimental settings

1) Datasets: In this paper, we conduct our attack on the MNIST [36], CIFAR10 [37] and GTSRB [38] datasets. The MNIST dataset involves a 10-class classification task with a total of 60,000 training images and 10,000 test images for recognizing hand-written digits. The CIFAR-10 dataset includes 10 categories and consists of 60,000 32x32 color images, including 50,000 training images and 10,000 test images for object recognition. The GTSRB dataset consists of 43 classes of traffic signs, with 39,209 training images and 12,360 testing images. These images exhibit variations in lighting conditions and diverse backgrounds. Additionally, the images in this dataset have different sizes. To ensure consistency, we resized them to a uniform size of 32x32 pixels.

2) Models: To create re-parameterized images, we use a classic encoder-decoder network that is commonly used in image reconstruction tasks. For the addition of backdoor triggers, we choose a U-Net style DNN as the encoder and use a spatial converter network as the decoder.

In terms of image classification models, we select six popular network structures: ResNet18 [39], ResNet34 [39], ResNet50 [39], DenseNet 121 [40], MobileNetV2 [41], and GoogleNet [42]. In our experimental evaluation, we use ResNet18 as the default network and use other network structures to demonstrate the generalization capability of our method.

3) Evaluation Metrics: The effectiveness evaluation comprises two metrics: clean data accuracy (CDA) and attack success rate (ASR). Specifically, CDA refers to the probability that clean samples are correctly predicted as their ground-truth class. ASR, on the other hand, refers to the probability that poisoned samples are successfully predicted as the attacker's specified class. For a successful backdoor model, ASR should be maintained at a high level, while CDA should be close to the accuracy of a clean model.

The stealthiness evaluation includes two metrics as well: PSNR [43] and SSIM [44], where PSNR represents local similarity and SSIM represents global similarity. The larger the PSNR is and the closer the SSIM is to 1, the better the invisibility of backdoor attacks is.

4) Default Settings for Training: The default poison rate is set to 10%. All victim classifiers use the SGD optimizer with a momentum of 0.9. The initial learning rate is $1e^{-5}$ for the MNIST dataset and 0.01 for the CIFAR10 and GTSRB dataset. The learning rate is carried out by cosine annealing with $T_{max} = 100$. We use these default settings in comparison to other methods.

Additionally, we utilize a grid search approach to optimize the hyperparameters α, β, and γ in $L_{all} = \alpha L_{\text{rect}} + \beta L_{\text{ssim}} + \gamma L_{\text{act}}$, setting them to 1.0, 0.5, and 0.5, respectively. For the temperature coefficient τ in the gumbel-softmax step and the λ parameter in the activation loss, we employ a random search strategy, determining their values to be 2 and 0.5, respectively.

TABLE I: Experimental results for Attack Effectiveness on MNIST dataset

Metrics		Label-inconsistent Attacks			Label-consistent Attacks		
	No attack	BadNets [8]	Blend [9]	Poison ink [22]	SIG [16]	CL [17]	Ours
CDA	99.81	99.49	<u>99.51</u>	98.98	99.51	**99.60**	98.03
ASR	\	<u>100.0</u>	<u>100.0</u>	99.75	87.97	86.53	**96.40**

TABLE II: Experimental results for Attack Effectiveness on CIFAR10 dataset

Metrics		Label-inconsistent Attacks			Label-consistent Attacks		
	No attack	BadNets [8]	Blend [9]	Poison ink [22]	SIG [16]	CL [17]	Ours
CDA	87.6	76.27	78.49	<u>84.29</u>	76.99	80.81	**86.62**
ASR	\	<u>99.98</u>	99.62	99.97	78.60	65.88	**96.50**

TABLE III: Experimental results for Attack Effectiveness on GTSRB dataset

Metrics		Label-inconsistent Attacks			Label-consistent Attacks		
	No attack	BadNets [8]	Blend [9]	Poison ink [22]	SIG [16]	CL [17]	Ours
CDA	95.32	93.89	<u>94.83</u>	92.64	**94.98**	91.88	94.62
ASR	\	93.20	97.10	<u>98.92</u>	**99.30**	78.74	94.40

B. Attack Effectiveness

Firstly, we select five classic backdoor attacks for comparison and divide them based on label consistency. We present the experimental results in terms of attack effectiveness in Table I, Table II and Table III. Note that in the following tables, we have bolded and darkened the optimal values for the label-consistent scenario, and underlined the optimal values for the label-inconsistent attacks.

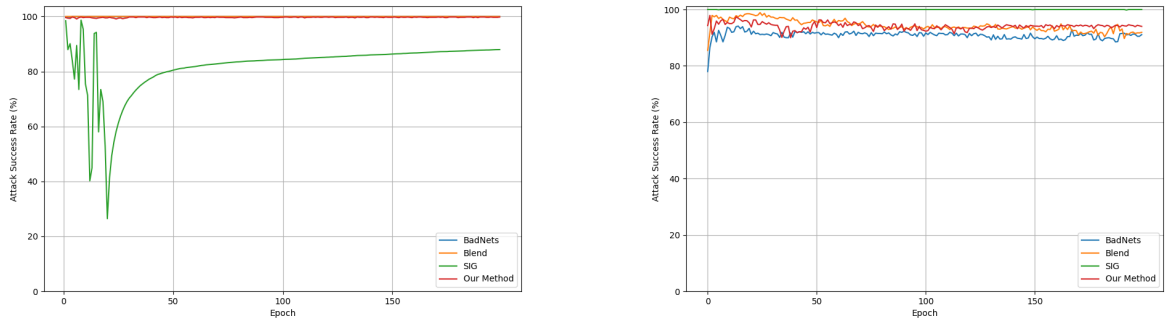
Specifically, our method significantly outperforms the SIG and CL methods in terms of attack success rate on the MNIST and CIFAR10 datasets, achieving the best attack performance in label-consistent backdoor attacks. Compared to the label-inconsistent backdoor attack, our method also achieves comparable attack results. Barni et al. [16] proved through experiments that compared with label inconsistent backdoor attacks, label consistent backdoor attacks need to destroy more samples in order to be successfully executed, which deeply illustrates the immense difficulty of backdoor attacks under clean label Settings. In the face of this challenge, our research methodology has made significant breakthroughs. In addition, clean data accuracy has been maintained at a very high level, which fully demonstrates the stability and reliability of the proposed method. On the GTSRB dataset, our method is slightly inferior to the SIG method. We believe that this may be due to the higher complexity of images and the larger number of categories of the GTSRB dataset, on which subtle differences in attack strategies may also cause significant variations in performance. Therefore, although our method performs well on MNIST and CIFAR10, the specific characteristics of the GTSRB dataset may be more suitable for the attack strategy of the SIG method. In summary, we consider that despite the differences in certain scenarios, our approach still possesses a clear competitive edge overall. Table I, II and III provide strong evidence of the effectiveness of our method, and further emphasizes the significant challenges posed by backdoor attacks under clean label conditions.

In addition to the discussion on the effectiveness of the attacks mentioned above, we further investigate the impact of the total number of training rounds on the effectiveness of the backdoor attacks. The experimental results, as shown in Fig. 4, indicate that similar to other classical backdoor attack methods, the backdoor attack proposed in this paper reaches its peak success rate after a few rounds of training and then remains relatively stable. Through this experiment, we gain a more comprehensive understanding of the dynamic evolution of backdoor attacks and further confirm the effectiveness of our approach.

C. Attack Stealthiness

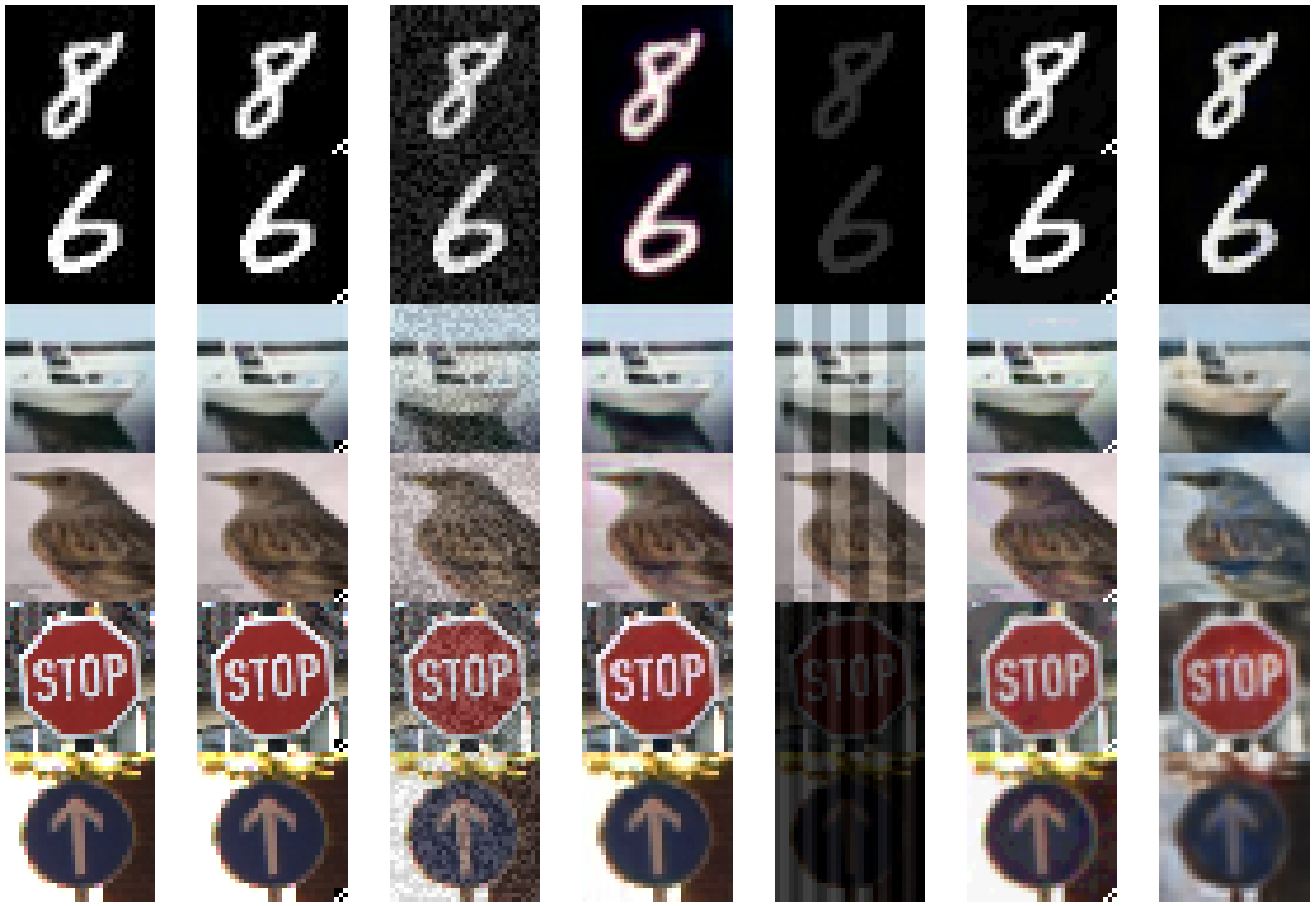
Fig. 5 provides visual representations of backdoor images generated by different attack methods. Through intuitive analysis, we can draw several results. Firstly, the backdoor images generated by BadNets, Blend, SIG, and CL exhibit poor stealthiness, indicating that they are relatively easy to detect and identify. In contrast, Poison ink method demonstrates the best concealment, making it extremely difficult to perceive. Our method ranks second in terms of stealthiness, although there is a certain gap compared to Poison ink, it still exhibits a commendable effect.

In addition to intuitive visual observation, we also provide objective measurements of concealment. The specific results are shown in Table IV. On the MNIST dataset, our method achieves optimal results on both SSIM and PSNR. This indicates that the generated backdoor images have the highest level of structural similarity and image quality compared to the original images. On the CIFAR10 and GTSRB dataset, our method's performance is slightly inferior. We speculate that this may be due to a certain degree of deviation in the color space caused by the re-parameterization sampling process employed in our method. In our study, the primary purpose of the image reconstruction network is to achieve label-consistent backdoor attacks rather than perfect reconstruction of the



(a) MNIST dataset (b) GTSRB dataset

Fig. 4: Impact of Total Training Epochs on Backdoor Attack Effectiveness



(a) Original (b) BadNets (c) Blend (d) Poison ink (e) SIG (f) CL (g) Ours

Fig. 5: Visual comparision of backdoor images generated by different attacks

original images. Therefore, we opt for a relatively simple autoencoder network for rapid validation and subsequent extension. This results in a certain degree of reconstruction deviation, particularly in color-rich and relatively complex datasets such as CIFAR10 and GTSRB.

We need to be aware that while Table IV provide some objective measurement criteria, the intuitive visual effects are more able to simulate real-world scenarios. Referring to the intuitive results in Fig. 5, we can see that the backdoor images generated by our method have high acceptability. Due to the sample-specific characteristics of the embedded backdoor trigger, it is difficult for humans to judge the image generated by our method as a backdoor image without comparing it to the original image. Additionally, in actual manual inspection process, it is rare to have the original images for comparison and verification. Therefore, the backdoor images generated by our method also excel in terms of stealthiness. Taking into account both the tabular data and the intuitive visual effects, our

TABLE IV: Objective Measurements of Concealment on MNIST, CIFAR10, and GTSRB datasets using SSIM and PSNR metrics: A Comparison with BadNets, Poison ink, SIG, and Other Attack Methods

Dataset →	MNIST		CIFAR10		GTSRB	
Metrics →	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR
BadNets [8]	0.9991	22.92	**0.9999**	26.32	**0.9999**	24.85
Blend [9]	0.8983	19.01	0.9830	22.30	0.9787	20.71
Poison ink [22]	0.9957	24.73	0.9996	**29.51**	0.9995	**28.14**
SIG [16]	0.1422	11.44	0.9907	19.59	0.1849	7.579
CL [17]	0.9949	22.26	0.9996	25.73	0.9634	19.36
Ours	**0.9992**	**25.52**	0.9875	21.41	0.9748	19.53

TABLE V: Generalization for Different Models on MNIST, CIFAR10, and GTSRB datasets

Dataset ↓	Metrics ↓	Resnet18	Resnet50	Densenet121	Googlenet	Resnet34	MobileNet
MNIST	CDA	98.03	97.53	99.12	99.26	99.18	99.01
	ASR	96.40	95.20	98.70	99.70	95.90	98.70
CIFAR10	CDA	86.62	83.36	83.65	83.87	84.38	81.75
	ASR	96.50	96.50	97.40	97.60	96.30	93.80
GTSRB	CDA	94.62	94.36	96.70	93.81	95.65	95.06
	ASR	94.40	95.70	93.40	93.50	98.60	95.30

method demonstrates satisfactory stealthiness in most cases and can fully meet the requirements in practical applications. In other words, our method achieves a good balance between stealthiness and practicality. In future research, we plan to explore the use of deeper and more complex reconstruction network structures, such as Variational Autoencoder (VAE) or Generative Adversarial Networks (GANs), to improve the quality and realism of reconstructed images.

D. Generalization on Different Models

To demonstrate the generalization of our method on different models, we conduct experiments using five other popular networks. The specific results are shown in Table V.

From the tabular data, it is evident that our attack method maintains high ASR on different models. This means that our method achieves the desired attack effect on both the original target model and other commonly used network structures. Such results are highly encouraging and further demonstrate the adaptability and stability of our method in various scenarios. Furthermore, it is worth noting that the backdoor introduced by our method does not significantly impact the original performance. This point is crucial for practical applications, as we need to ensure that the target is attacked without negatively affecting the overall performance of the entire system.

E. Ablation Experiment

1) The effectiveness and importance of re-parameterizing operations: In the preliminary experiments, to validate the effectiveness of the re-parameterization operation, we compare the classification confidences of images before and after using re-parameterized noise sampling. The experimental results, as shown in the Fig. 6, illustrate the classification confidences obtained from the classification network for original image inputs on the left side, and the classification confidences after re-parameterized noise sampling on the right side. It

TABLE VI: Comparison of CDA/ASR with Gaussian and Gumbel-Softmax Noise Mechanisms

	MNIST	CIFAR10	GTSRB
Gaussian	99.07 / 93.20	85.16 / 91.00	94.45 / 90.50
Gumbel-Softmax	99.04 / 100.0	86.62 / 96.50	94.62 / 94.40

can be observed from the figure that after performing the re-parameterization operation, the classification confidences for classes other than the original class of reconstructed images increase, and the distribution of classification confidences becomes noticeably smoother. This observation confirms the effectiveness of the re-parameterization operation, as it successfully interferes with the salient features of the images. Additionally, besides the Gumbel-Softmax method, we also attempt the mechanism of Gaussian noise. The experimental results are shown in Table VI, where the values before and after the slash (/) correspond to CDA and ASR. It is evident from the table that the success rate of attacks using Gaussian noise is inferior to that of our proposed method.

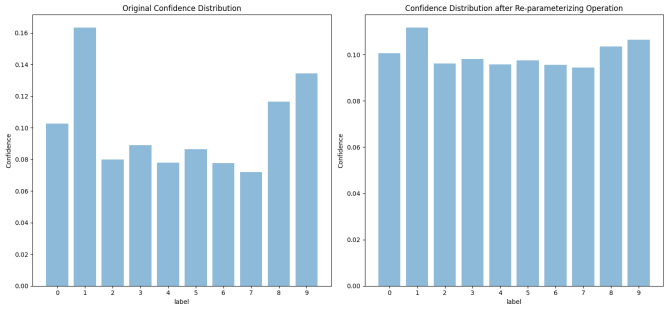


Fig. 6: The effectiveness of re-parameterized noise sampling on CIFAR10 Dataset

Finally, aiming to demonstrate the importance of re-parameterized noise sampling in our method, we conduct

TABLE VII: The importance of re-parameterized noise sampling on CIFAR10 dataset

	with re-parameterization	without re-parameterization
CDA	**86.62**	84.84
ASR	**96.50**	85.20

ablation experiments on CIFAR10 dataset, and the experimental results are presented in Table VII. As can be seen from the table, removing the re-parameterization operation greatly reduces the attack success rate. This indicates the crucial role of reparameterization in maintaining a high attack success rate. Without the reparameterization operation, the malicious model trained by the user fails to complete the mapping of the specific input trigger to the target label. Therefore, the reparameterization operation is necessary and effective in our method, providing a solid foundation for the success of the attack.

2) The influence of loss function on stealthiness: In the process of re-parameterized image reconstruction, we utilize three loss functions: L_{rect}, L_{ssim}, and L_{act}. Here, we conduct experimental analysis to investigate the impact of these loss functions on image concealment. The corresponding results are shown in Table VIII. From the table, it can be observed that using any individual loss function alone fails to reach the expected quality level in the reconstructed images. Using the L_{rect} loss function alone can reduce reconstruction errors, but the image quality remains unsatisfactory. Incorporating the L_{ssim} loss function after L_{rect} slightly improves the quality of the reconstructed images but still falls short of the optimal outcome.

In summary, these three loss functions play different roles in the image reconstruction process, and their combination is crucial to obtain the best reconstruction results. The L_{rect} loss function primarily focuses on reducing reconstruction errors and minimizing the differences between the reconstructed and original images. The L_{ssim} loss function emphasizes the structural similarity of the images, encouraging the reconstructed images to closely resemble the original ones in terms of structure. On the other hand, the L_{act} loss function primarily considers the activation state of the target model, helping improve the performance of the reconstructed images on the target model. By simultaneously using the L_{rect}, L_{ssim}, and L_{act} loss functions, we can obtain the highest-quality reconstructed images that are optimized in terms of reconstruction errors, structural similarity, and target model activation state.

3) The influence of poison ratio on results: The default poison ratio in this article is 10%. However, to comprehensively evaluate the effectiveness of backdoor attacks, we further explores other poisoning ratios on MNIST and CIFAR10 datasets. The experimental results are shown in Fig. 7.

Previous experiments demonstrate that common backdoor attacks only require 1-4% of poisoned samples to be effective, while label-consistent backdoor attacks may require 10-30% poisoning. By observing Fig. 7, we can see that the ASR does decrease to a certain extent when the poisoning ratio is below 10%. However, it is encouraging to note that even with only 1% of injected backdoor samples, our method still achieves 85% ASR. This result is significant as it highlights the vulnerability of the target model to our backdoor attack method, even at extremely low levels of backdoor injection. It reminds us of the importance of prioritizing the security of the target model even in scenarios where backdoor injection is minimal.

F. Resistance to Defense Techniques

In order to defend against backdoor attacks, researchers have proposed many defense methods, including data-level defenses and model-level defenses. Here, we select a classical method from the two types of defense for experimental testing.

1) SentiNet Detection: As a data-level defense method, SentiNet detection is a highly regarded technique. It utilizes model interpretability and target detection techniques as detection mechanisms. By applying the Grad-Cam technique, it visualizes the attention map of the target image to locate the backdoor triggers. We conduct experiments on multiple attack methods, and the experimental results are presented in Fig. 8. Particularly, the performance in localizing trigger regions generated by BadNets is outstanding. Furthermore, for Blend, Poison ink, and SIG attack methods, the localized regions differ significantly from the original image. In contrast, the target localization of our method is basically consistent with that of the original image, indicating that our approach has a significantly lesser impact on classification performance than other methods. This also confirms the ineffectiveness of the Sentinet defense method in countering our attack.

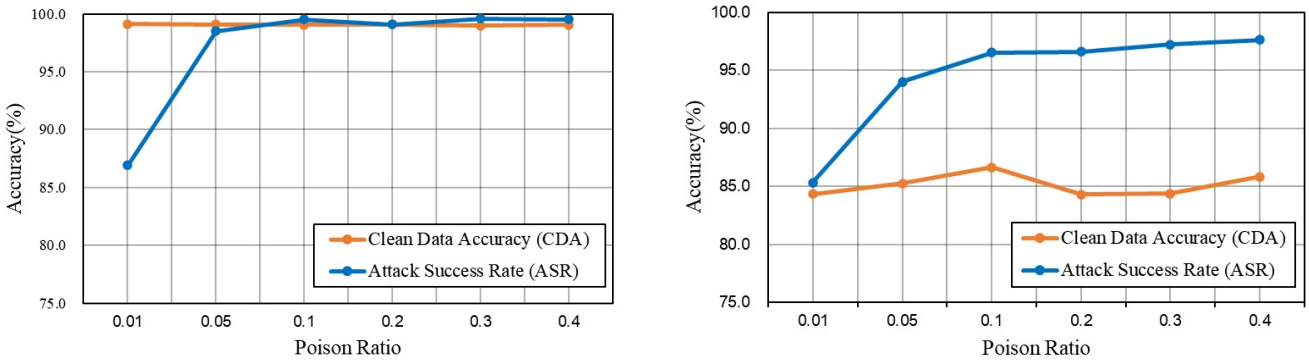
2) Neural Cleanse: Potential triggers refer to samples in the input that possess specific patterns or attributes, triggering backdoor behavior when these samples are input into the model. Neural Cleanse utilizes gradient descent approach to search for possible triggers and returns an anomaly score for each classifier. If the anomaly score is greater than 2, the classifier is considered to be poisoned. The experimental results, as shown in Fig. 9, demonstrate that our poisoned model can bypass the detection of Neural Cleanse.

VI. Conclusion

In this paper, we found that most existing backdoor attacks excessively focus on the visual concealment of backdoor triggers while neglecting the issue of matching between sample label and image content. Although many backdoor images can deceive the human visual system completely, their labels are inconsistent with the image content, making them prone to detection during image-label matching checks. To address this problem, we propose a truly invisible backdoor attack method that satisfies the requirements of both label-consistent settings and visual invisibility. Using the technique of re-parameterizing noise, we perturb the salient features of benign samples to generate re-parameterized images, making subsequent classification more dependent on the added backdoor triggers. When adding backdoor triggers, we draw inspiration from the concept of image steganography based on DNN. We encode a specific string onto the re-parameterized image to generate a sample-specific backdoor image. Extensive

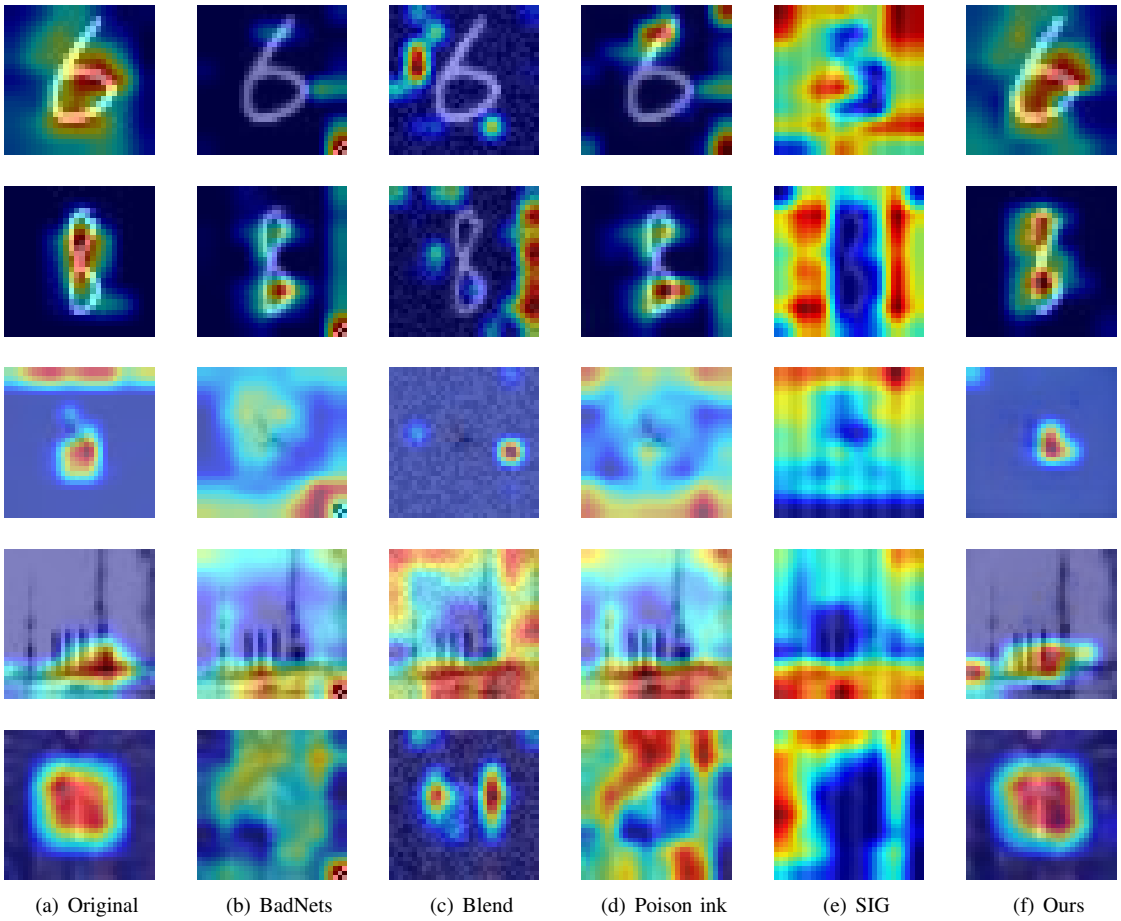
TABLE VIII: Impact of different loss functions on image concealment

	L_{rect}	L_{ssim}	L_{act}	L_{rect} & L_{ssim}	L_{rect} & L_{act}	L_{ssim} & L_{act}	*All*
SSIM	0.9854	0.9740	0.6651	0.9789	0.5119	0.9416	**0.9981**
PSNR	14.85	13.92	6.421	17.78	4.778	11.86	**22.36**



(a) MNIST dataset (b) CIFAR10 dataset

Fig. 7: Trade-off between CDA and ASR under different poison ratios.



(a) Original (b) BadNets (c) Blend (d) Poison ink (e) SIG (f) Ours

Fig. 8: Using SentiNet to locate critical regions on different attacks

experiments show the superiority of our approach in terms of attack success rate, stealth and generalization. Moreover, the method in this article is resistant to multiple defense techniques, demonstrating strong robustness. In future work, we plan to explore new backdoor attack methods within the framework of distributed learning, such as federated learning.

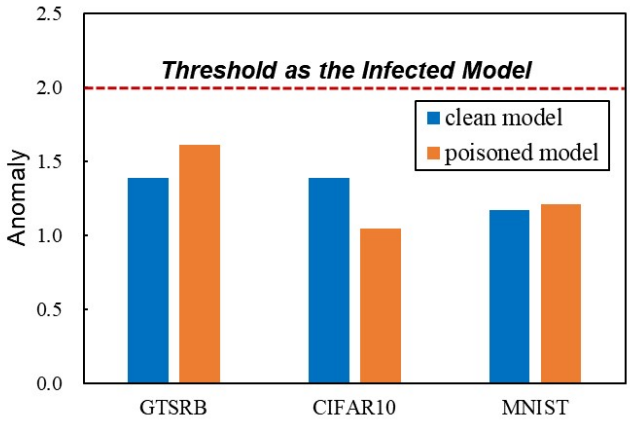


Fig. 9: The Defense Results using Neural Cleanse.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No.62106037, No.62076052), the Science and Technology Innovation Foundation of Dalian (No.2021JJ12GX018), the Application Fundamental Research Project of Liaoning Province (2022JH2/101300262), and the Major Program of the National Social Science Foundation of China (No.19ZDA127).

REFERENCES

[1] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," *Advances in neural information processing systems*, vol. 13, 2000.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30*, 2016, pp. 770–778.

[3] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9*, 2017, pp. 2087–2091.

[4] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213–237, 2019.

[5] Y. Cheng, Q. Guo, F. Juefei-Xu, S.-W. Lin, W. Feng, W. Lin, and Y. Liu, "Pasadena: Perceptually aware and stealthy adversarial denoise attack," *IEEE Transactions on Multimedia*, vol. 24, pp. 3807–3822, 2022.

[6] L. Gao, Z. Huang, J. Song, Y. Yang, and H. T. Shen, "Push & pull: Transferable adversarial examples with attentive attack," *IEEE Transactions on Multimedia*, vol. 24, pp. 2329–2338, 2022.

[7] C. Wan, F. Huang, and X. Zhao, "Average gradient-based adversarial attack," *IEEE Transactions on Multimedia*, pp. 1–14, 2023.

[8] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[9] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *CoRR*, vol. abs/1712.05526, 2017. [Online]. Available: http://arxiv.org/abs/1712.05526

[10] H. Zhong, C. Liao, A. C. Squicciarini, S. Zhu, and D. J. Miller, "Backdoor embedding in convolutional neural network models via invisible perturbation," in *The Tenth ACM Conference on Data and Application Security and Privacy, CODASPY 2020, New Orleans, LA, USA, March 16-18*, 2020, pp. 97–108.

[11] T. A. Nguyen and A. T. Tran, "Wanet - imperceptible warping-based backdoor attack," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7*, 2021.

[12] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7*, 2021.

[13] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23*, 2019, pp. 707–723.

[14] X. Zhang, Y. Jin, T. Wang, J. Lou, and X. Chen, "Purifier: Plug-and-play backdoor mitigation for pre-trained models via anomaly activation suppression," in *The 30th ACM International Conference on Multimedia, MM 2022, Lisboa, Portugal, October 10-14*, 2022, pp. 4291–4299.

[15] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–18, 2022.

[16] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in CNNS by training set corruption without label poisoning," in *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25*, 2019, pp. 101–105.

[17] A. Turner, D. Tsipras, and A. Madry, "Label-consistent backdoor attacks," *CoRR*, vol. abs/1912.02771, 2019. [Online]. Available: http://arxiv.org/abs/1912.02771

[18] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the Twenty-Fifth International Conference on Machine Learning, ICML 2008, Helsinki, Finland, June 5-9*, vol. 307, 2008, pp. 1096–1103.

[19] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *15th European Conference on Computer Vision, ECCV 2018, Munich, Germany, September 8-14*, vol. 11219, 2018, pp. 682–697.

[20] M. Tancik, B. Mildenhall, and R. Ng, "Stegastamp: Invisible hyperlinks in physical photographs," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19*, 2020, pp. 2114–2123.

[21] E. Sarkar, H. Benkraouda, and M. Maniatakos, "Facehack: Triggering backdoored facial recognition systems using facial characteristics," *CoRR*, vol. abs/2006.11623, 2020. [Online]. Available: https://arxiv.org/abs/2006.11623

[22] J. Zhang, C. Dongdong, Q. Huang, J. Liao, W. Zhang, H. Feng, G. Hua, and N. Yu, "Poison ink: Robust and invisible backdoor attack," *IEEE Transactions on Image Processing*, vol. 31, pp. 5691–5705, 2022.

[23] Y. Liu, S. Ma, Y. Aafer, W. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21*, 2018.

[24] M. Zou, Y. Shi, C. Wang, F. Li, W. Song, and Y. Wang, "Potrojan: powerful neural-level trojan designs in deep learning models," *CoRR*, vol. abs/1802.03043, 2018. [Online]. Available: http://arxiv.org/abs/1802.03043

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *26th Annual Conference on Neural Information Processing Systems 2012, NeurIPS 2012, Lake Tahoe, Nevada, United States, December 3-6*, 2012, pp. 1106–1114.

[26] A. S. Rakin, Z. He, and D. Fan, "TBT: targeted neural network attack with bit trojan," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19*, 2020, pp. 13 195–13 204.

[27] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, "Proflip: Targeted trojan attack with progressive bit flips," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17*, 2021, pp. 7698–7707.

[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[29] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *Proceedings of the AAAI conference on artificial intelligence, AAAI 2020*, vol. 34, no. 07, 2020, pp. 11 957–11 965.

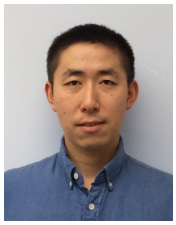
[30] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. M. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," in *Proceedings of the AAAI conference on artificial intelligence, AAAI 2019, Honolulu, Hawaii, January 27*, vol. 2301, 2019.

[31] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: a defence against trojan attacks on deep neural networks," in *Proceedings of the 35th Annual Computer Security Applications Conference, ACSAC 2019, San Juan, PR, USA, December 09-13*, 2019, pp. 113–125.

[32] E. Chou, F. Tramer, and G. Pellegrino, "Sentinet: Detecting localized universal attacks against deep learning systems," in *2020 IEEE Security and Privacy Workshops, SPW*. IEEE, 2020, pp. 48–54.
[33] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531
[34] C.-K. Chan and L.-M. Cheng, "Hiding data in images by simple lsb substitution," *Pattern recognition*, vol. 37, no. 3, pp. 469–474, 2004.
[35] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *18th International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI 2015, Germany, October 5-9*, vol. 9351, 2015, pp. 234–241.
[36] Y. LeCun, L. D. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard *et al.*, "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural networks: the statistical mechanics perspective*, vol. 261, no. 276, p. 2, 1995.
[37] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
[38] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: A multi-class classification competition," in *The 2011 International Joint Conference on Neural Networks, IJCNN 2011*, 2011, pp. 1453–1460.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30*, 2016, pp. 770–778.
[40] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26*, 2017, pp. 2261–2269.
[41] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12*, 2015, pp. 1–9.
[43] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.
[44] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

Bo Wang (IEEE Member) received his B.S. degree in electronic and information engineering and his M.S. and Ph.D. degrees in signal and information processing from the Dalian University of Technology, Dalian, China, in 2003, 2005, and 2010, respectively. From 2010 to 2012, he was a Post-Doctoral Research Associate with the Faculty of Management and Economics, Dalian University of Technology. He is currently an Associate Professor at the School of Information and Communication Engineering, Dalian University of Technology. His current research interests include multimedia processing and security, such as digital image processing and forensics.

Fei Yu received the B.S. degree in electronic and information engineering from Dalian University of Technology, China, in 2021. She is currently pursuing an M.S. degree in the Department of Information and Communication Engineering, Dalian Universityof Technology. Her research interests include federated learning and poisoning attack.



Fei Wei (IEEE Member) received his Ph.D. in electrical engineering from the State University of New York at Buffalo, USA in 2020. He is a research fellow at National University of Singapore since 2022. Before joining NUS, he was a postdoctoral research scholar at Arizona State University, USA. His research interests include but not limited to security, privacy, fairness, and network information theory.

Yi Li (IEEE Member) received the B.E. and M.E. degrees from the Dalian University of Technology (DUT), Dalian, China, in 2014 and 2017, respectively, and the Ph.D. degree from the University of Chinese Academy of Sciences (UCAS), Beijing, China, in 2020. She is currently an Associate Professor with the School of Artificial Intelligence, DUT. Her research interests include computer vision, pattern recognition, and multimedia computing.

Wei Wang (IEEE Member) received the B.E. degree in computer science and technology from North China Electric Power University in 2007. Since 2012, he has been with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, where he is currently an Assistant Professor. His research interests include pattern recognition, image processing, and digital image forensics, including watermarking, steganalysis, and tampering detection.