

Poison Ink: Robust and Invisible Backdoor Attack

Jie Zhang^{ID}, Dongdong Chen^{ID}, Member, IEEE, Qidong Huang^{ID}, Jing Liao^{ID}, Member, IEEE, Weiming Zhang^{ID}, Member, IEEE, Huamin Feng, Gang Hua^{ID}, Fellow, IEEE, and Nenghai Yu^{ID}

Abstract—Recent research shows deep neural networks are vulnerable to different types of attacks, such as adversarial attacks, data poisoning attacks, and backdoor attacks. Among them, backdoor attacks are the most cunning and can occur in almost every stage of the deep learning pipeline. Backdoor attacks have attracted lots of interest from both academia and industry. However, most existing backdoor attack methods are visible or fragile to some effortless pre-processing such as common data transformations. To address these limitations, we propose a robust and invisible backdoor attack called “Poison Ink”. Concretely, we first leverage the image structures as target poisoning areas and fill them with poison ink (information) to generate the trigger pattern. As the image structure can keep its semantic meaning during the data transformation, such a trigger pattern is inherently robust to data transformations. Then we leverage a deep injection network to embed such input-aware trigger pattern into the cover image to achieve stealthiness. Compared to existing popular backdoor attack methods, Poison Ink outperforms both in stealthiness and robustness. Through extensive experiments, we demonstrate that Poison Ink is not only general to different datasets and network architectures but also flexible for different attack scenarios. Besides, it also has very strong resistance against many state-of-the-art defense techniques.

Index Terms—Backdoor attack, stealthiness, robustness, generality, flexibility.

I. INTRODUCTION

IN THE past years, deep learning has achieved tremendous success in a lot of application areas [1], [2], [3], [4], [5]. However, recent works show that they are vulnerable to

Manuscript received 18 January 2022; revised 25 May 2022 and 7 July 2022; accepted 9 August 2022. Date of publication 30 August 2022; date of current version 5 September 2022. This work was supported in part by NSFC under Grant U20B2047, Grant 62072421, Grant 62002334, Grant 62102386, and Grant 62121002; in part by the Exploration Fund Project of University of Science and Technology of China (USTC) under Grant YD3480002001; in part by the Fundamental Research Funds for the Central Universities under Grant WK210000011 and Grant WK5290000001; and in part by ECS Grant from the Research Grants Council of the Hong Kong under Project CityU 21209119. The work of Gang Hua was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101400 and in part by NSFC under Grant 61629301. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bart Goossens. (*Corresponding author: Dongdong Chen*.)

Jie Zhang, Qidong Huang, Weiming Zhang, and Nenghai Yu are with the School of Cyber Science and Security, University of Science and Technology of China, Hefei, Anhui 230026, China (e-mail: zjzac@mail.ustc.edu.cn; hqd0037@mail.ustc.edu.cn; zhangwm@ustc.edu.cn; ynh@ustc.edu.cn).

Dongdong Chen is with Microsoft Research, Redmond, WA 98052 USA (e-mail: cdlyf@gmail.com).

Jing Liao is with the Department of Computer Science, City University of Hong Kong (e-mail: jingliao@cityu.edu.hk).

Huamin Feng is with the Beijing Electronic Science and Technology Institute, Beijing 100070, China (e-mail: fenghm@besti.edu.cn).

Gang Hua is with Wormpex AI Research LLC, Bellevue, WA 98004 USA (e-mail: ganghua@gmail.com).

Digital Object Identifier 10.1109/TIP.2022.3201472

different types of attacks, such as adversarial attacks [6], [7], [8], [9], [10], data poisoning attacks [11], [12] and backdoor attacks [13], [14], [15]. Adversarial attacks focus on misleading the model only in the test process, while data poisoning attacks aim to degrade the model inference performance of its primary task by contaminating the training process. Backdoor attacks are more flexible and cunning than the two attacks above. Specifically, backdoor attack can affect even all stages of machine learning pipeline [16], such as model training [13], fine-tuning [14] and even after deployment [17]. In addition, backdoor attacks are designed to make the backdoored model behave like a normal model unless the attackers feed some specially designed triggers.

Since it was introduced, an increasing amount of research has been paying attention to this field. For example, Gu *et al.* [13] directly stamped a square sticker or flower patch onto clean images to contaminate the training process, and Chen *et al.* [15] replaced the injection strategy by blending a trigger image with the clean example. However, there are two main limitations to existing backdoor attack methods. First, the trigger pattern used in many current works is visible (some visual examples are shown in Figure 2), which can be easily recognized by humans or some deep visualization methods like Grad-CAM [18]. Second, a very recent research [19] finds that most existing backdoor attacks will totally fail during the inference stage if pre-processing the trigger images with some simple data transformations, such as flipping and padding after shrinking.

To address the above limitations, we aim to design a robust and invisible backdoor attack method. Besides the essential requirement for backdoor attacks: 1) the trigger pattern should be easy for the model to learn and cannot confuse the model to affect the pristine performance, we add two more requirements to achieve our goal: 2) the trigger pattern should be consistent under conventional data transformations to keep its robustness; 3) the trigger image needs be visually indistinguishable from the corresponding trigger-free image.

In this paper, we propose to utilize the image structure (edge) of an image as the carrier of poison information, *i.e.*, hiding the poison information into the edge structures as shown in Figure 1. Compared to existing trigger patterns, the proposed structure-based trigger pattern has several natural advantages: 1) On one hand, the shallow layers of DNN [20] often capture the low-level structure information, which means the structure can be easily captured by DNN; on the other hand, the final decision of DNN [21] often depends on the object texture rather than the structure information, which indirectly indicates that structure-based trigger pattern will not undermine performance of the original task. So it satisfies the

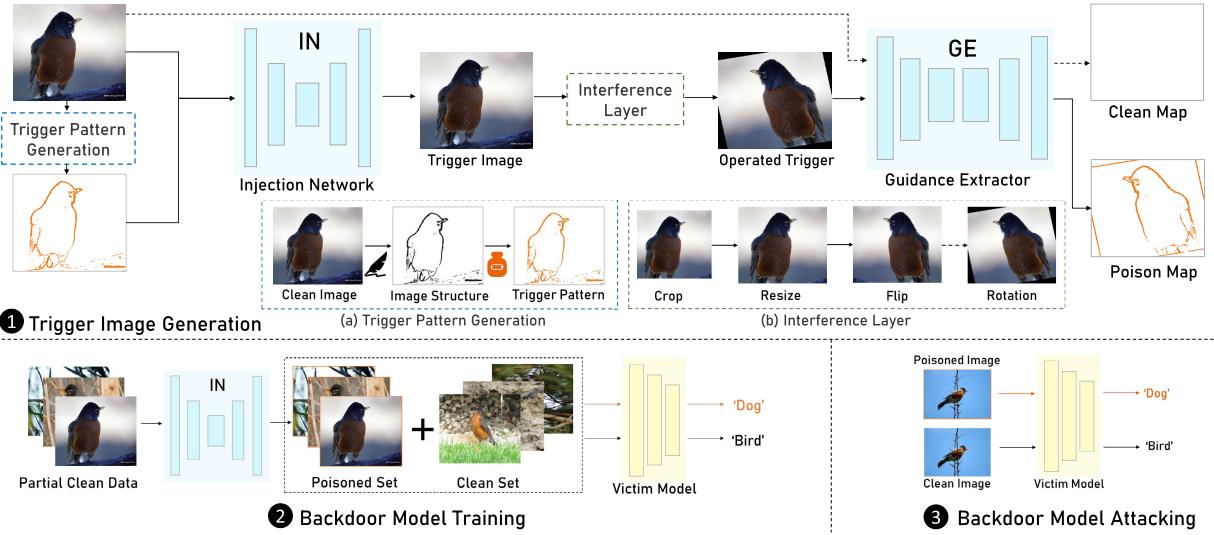


Fig. 1. The overall pipeline of Poison Ink, which mainly consists of trigger image generation, backdoor model training and backdoor model attacking.

first requirement. 2) It is distributed in the whole image and can keep its semantic meaning unchanged under common data transformations, which indirectly satisfies the second requirement. 3) Edge structures belong to high-frequency components of an image, so hiding information into them is more difficult to be discovered, which satisfies the third requirement.

Based on the above observations, a new backdoor attack method **“Poison Ink”** is designed. The overall framework is shown in Figure 1. Specifically, we outline the image structures as poison areas and then embed color values (representation of poison information) into such areas to generate the trigger pattern. To achieve stealthiness, we use a deep injection network to hide the input-aware trigger pattern into the cover image to produce the final poisoned image. Such poisoned images will be regarded as the trigger set and mixed with clean images as the training set for the standard backdoor training.

To demonstrate the effectiveness of Poison Ink, we conduct extensive experiments on various datasets and network architectures under different attacking scenarios. Compared to existing backdoor attack methods, Poison Ink outperforms in terms of stealthiness, robustness, generality, and flexibility. In summary, our contributions are four-fold as follows:

- We are the first to propose utilizing image structures as the carrier of trigger patterns, showing they have natural advantages over existing trigger pattern designs.
- We design a new backdoor attack framework, Poison Ink, which uses colorized image structures as the trigger pattern and hides the trigger pattern invisibly by using a deep injection network.
- Extensive experiments demonstrate the stealthiness and robustness of Poison Ink, which is generally applicable to different datasets and network structures.
- Poison Ink works well in different attacking scenarios and has strong resistance to many defense techniques.

II. RELATED WORK

A. Backdoor Attack

Backdoor attack is a classic topic in the system security field, and Gu *et al.* [13] first introduced this issue into deep

models. Based on it, Chen *et al.* [15] proposed a blending injection strategy to attack face recognition systems with less poisoned data. Then, Liu *et al.* [14] utilized reverse engineering to generate data and built a strong relationship between the optimized patch trigger with the selected neurons via fine-tuning. Unlike these attack methods based on static and visible trigger patterns, the trigger pattern of Poison Ink is dynamic and invisible.

Besides the aforementioned suspicious attack methods, there are a few attempts at more stealthy backdoor attacks. Zhong *et al.* [22] proposed an invisible trigger called static perturbation mask (SPM), for instance, a checkboard-like pattern. Similarly, Barni *et al.* [23] replaced the repeated mask with the sinusoidal signal (SIG). Very recently, Liu *et al.* [24] presented “Refool”, which is spurred by the natural phenomenon – reflectance. However, all trigger patterns aforesaid are unnatural due to their input-agnostic attribute. Conversely, our proposed method focuses on an input-aware trigger pattern, which is much harder to be detected. Unlike the attacks above, Li *et al.* [25] proposed generating an invisible attack via a steganography algorithm called least significant bit (LSB) substitution. However, it totally fails on low-resolution datasets like CIFAR-10, and Poison Ink significantly outperforms LSB in the context of robustness to data transformation. We also notice some interesting works [26], [27], [28] explored the incorporation of image structures for evasion attacks and defenses. Different from them, we are the first to leverage image structures for backdoor attacks.

B. Defense Against Backdoor Attack

To resist backdoor attack, many defense methods have been proposed, which can be roughly categorized into three groups: data-based defense, model-based defense, and meta classifiers.

For data-based defense, Tran *et al.* [29] removed the poisoned examples by analyzing the spectrum of latent features. However, its assumption of having full access to the infected training data is not practical in use and thus not considered in our experiment. Gao *et al.*’s observation included predicting the backdoor image under strong perturbations, on which

STRIP [30] was proposed. In addition, Doan *et al.* [31] proposed *Februus* by using Grad-CAM [18] to locate the potential trigger region and replacing it by image restoration. Recently, Li *et al.* [19] showed that most existing attack methods are vulnerable to data transformations, which we mainly focus on in this paper.

For model-based defense, *Fine-pruning* [32] tries to prune the neurons that are dormant for clean inputs, which is assumed to have a relationship to the activation of the backdoor. Chen *et al.* proposed to detect backdoor attacks by *Activation Clustering* [33], which determines the infected category by analyzing the activation clustering of all classes. Inspired by the Electrical Brain Stimulation technique, Liu *et al.* proposed *ABS* [34] to scan malicious neuron. Then, Wang *et al.* [35] proposed *Neural Cleanse*, which first reverse-engineers the trigger pattern and then utilizes the reversed trigger for backdoor removal. Based on *Neural Cleanse*, *TABOR* [36] obtained a further improvement by appending various regularization during reverse-engineering. Rather than using reversed trigger, *MESA* [37] leveraged many generated triggers to improve the backdoor removal performance. Later, a more effective method named *TND* is designed by Wang *et al.* [38], which is applicable even in data-limited or data-free scenarios.

There are also some methods, such as *ULPs* [39], based on the idea of a meta classifier. However, it consumes enormous computation resources and has a strong but impractical assumption that the trigger size is known. To evaluate the robustness of Poison Ink, we will try these state-of-the-art defense methods in the following experiments.

III. PRELIMINARIES

Before introducing Poison Ink, we first formally define backdoor attacks and briefly analyze existing methods' limitations. Finally, we clarify the threat model and our attack goals.

A. Problem Definition

In this paper, we only consider the backdoor attack on the image classification task. For image classification, assuming the input domain \mathbb{X} is composed of massive images $\{x_1, x_2, \dots, x_N\}$, and the target output domain \mathbb{L} consists of corresponding labels $\{l_1, l_2, \dots, l_N\}$. Then the goal of the image classification model \mathbf{M} is to approximate the implicit transformation function by minimizing the distance \mathcal{D} (e.g., cross-entropy) between $\mathbf{M}(x_i)$ and l_i , i.e.,

$$\mathcal{D}(\mathbf{M}(x_i), l_i) \rightarrow 0. \quad (1)$$

For backdoor attacks, we randomly choose a portion of training data as the candidate set \mathbb{X}^c and select some trigger patterns from the pre-designed trigger pattern set \mathbb{P} . Here, we pick a single trigger pattern \mathbf{p} from \mathbb{P} as an example. With a pre-defined backdoor injection strategy \mathcal{I} , we can generate the poisoned image x^p , i.e.,

$$x_i^p = \mathcal{I}(x_i^c, \mathbf{p}), x_i^c \in \mathbb{X}^c. \quad (2)$$

All poisoned images $\{x_1^p, x_2^p, \dots, x_{N_p}^p\}_{p \in \mathbb{P}}$ with the corresponding target attack labels $\{l_1^p, \dots, l_{N_p}^p\}$ will be combined with the left clean images $\{x_1, x_2, \dots, x_{N_c}\}$ and their original labels $\{l_1, l_2, \dots, l_{N_c}\}$ as the final backdoor training dataset \mathbb{X}^* , where $(\sum_{p \in \mathbb{P}} N_p) + N_c = N$. The injection ratio α is defined as $\alpha = (\sum_{p \in \mathbb{P}} N_p)/N$. Finally, we can obtain an infected model \mathbf{M}^* by training on the polluted dataset \mathbb{X}^* .

B. Brief Analysis of Existing Limitations

Most existing backdoor attack methods have a limitation in stealthiness or robustness. For the stealthiness limitation, we attribute it to two points: the unnatural and input-agnostic trigger patterns; the injection strategy \mathcal{I} , like hard-pasting or soft-blending, which is hard to adapt to various inputs to satisfy the invisibility requirement. For the lack of robustness to data transformation, it is also because the trigger pattern \mathbf{p} has no relationship with its cover input x^c . During training time, the infected model \mathbf{M}^* is forced to remember the relationship between such input-agnostic pattern \mathbf{p} and the target attack label l^p . Then, at the inference stage, if the poisoned image x^p is pre-processed by some data transformations, the hidden trigger pattern \mathbf{p} will also be transformed, and the relationship will be corrupted. Therefore, the infected model \mathbf{M}^* cannot recognize \mathbf{p} any more.

C. Threat Model and Our Goals

Backdoor attacks can occur at any stage of a deep learning pipeline. In this paper, we introduce the threat model in terms of attacker's capacities and attack scenarios as follows:

Attacker's capacities: we assume that attackers are allowed to poison some training data, whereas they have no information on or change other training components (e.g., training loss, training schedule, and model structure). In the inference process, attackers can and only can query the trained model with any image. They have neither information about the model (even prediction) nor can they manipulate the inference process, which is different from adversarial example. The assumption above is the minimal requirement for attackers [40]. Taking data transformations as an example, the attacker is not accessible to the exact range and combinations order of the data transformations used by the defender.

Attack scenarios: the discussed threat can happen in many real-world scenarios, including but not limited to adopting third-party training data, training platforms, and model APIs. The attacker can directly inject his own poisoned data into the training stage. During inference, the attacker does not need to hijack any image and triggers the attack by querying the target model, with the poisoned image generated by himself in the same way. We shall point out that we mainly focus on the threat in the digital domain, where many backdoor attacks were explored, and attacks in the physical world will be our future direction.

We aim to achieve an invisible, robust, general and flexible backdoor attack (Poison Ink) and set main goals in detail: 1) maintain the model performance on clean data; 2) the poisoned image shall be imperceptible to evade human inspection at the inference stage; 3) Keep high attack effectiveness

even when some data transformations pre-process the poisoned image.

IV. THE PROPOSED POISON INK

To achieve our goal, we propose a new backdoor attack method called ‘‘Poison Ink’’, and the overall pipeline is shown in Figure 1. We are motivated by the recent work [41] that proposes the physical consistency for model IP protection. When generating the trigger set, we first generate the trigger pattern by embedding the poison information into edge structures and then embed the trigger pattern into the cover image with a deep invisible injection strategy. An interference layer is added to enhance the robustness further and dynamically generates diverse interference to the injection network and the auxiliary guidance extractor network.

A. Trigger Pattern Generation

As described before, the edge structure of one image is an ideal carrier for the trigger pattern: 1) It can be easily captured by the shallow layers of deep models and will not undermine the performance of the original task. 2) The edge structure can keep its semantic meaning and physical existence during data transformation. 3) Different from existing pattern designs, edge structure is also the inherent high-frequency component of an image, so it is easy to be hidden in an invisible way.

In the trigger pattern generation module of Figure 1, we give a simple example to illustrate how to generate a trigger pattern by using the edge structure. In details, given an input image x_i , we first outline its edge by using some edge extraction algorithms \mathcal{E} such as Sobel [42] or Canny [43] operator. Next, we encode the poison information into the RGB color value \mathcal{C}_i^P by mathematical encoding and colorize the black and white edge image with \mathcal{C}_i^P as the target trigger pattern p_i :

$$p_i = \mathcal{E}(x_i) \otimes \mathcal{C}_i^P, \quad (3)$$

where \otimes means the color filling operator that makes all the edge color values be \mathcal{C}_i^P . Compared to existing trigger patterns, such edge structure based trigger pattern is input-aware and dynamic, which breaks the assumptions of many existing defense techniques that the trigger patterns are input-agnostic and static patches. Besides, since the RGB color space is enormous and many different edge extraction algorithms exist, it also naturally supports multiple different trigger patterns by changing the color values or edge types.

B. Deep Invisible Injection Strategy

After getting the edge structure based trigger pattern, we design a deep injection strategy to hide the trigger pattern into the cover image. As shown in the top part of Figure 1, it basically consists of three parts during training: a deep injection network, an auxiliary guidance extractor network to help the injection network learn in a proper way, and an interference layer to force the injection network to embed trigger pattern more robustly. After the training process, both the auxiliary guidance network and the interference layer will be discarded. We will only use the deep injection network to hide the trigger pattern into the clean cover images to generate the poisoned images.

1) *Injection Network IN*: We concatenate the clean image x_i^c with its corresponding trigger pattern p_i along the channel dimension before feeding them into *IN* to obtain the final poisoned image x_i^P , i.e,

$$x_i^P = IN([x_i^c; p_i]). \quad (4)$$

In order to encourage *IN* to hide the trigger pattern invisibly, we need to find some invisibility loss metrics to guide its learning. However, it is often difficult to explicitly define invisibility. As shown in existing information hiding methods [44], [45], [46], [47], though L_p Norm is not a perfect invisibility metric, it can serve as a good invisibility learning metric. So we utilize it as one invisibility loss:

$$\mathcal{L}_{inv} = \mathbb{E}_{x_i^c \in \mathbb{X}^c} [\|x_i^P - x_i^c\|^k]. \quad (5)$$

By default, we use L1 loss by setting $k = 1$, which follows the classic image-to-image translation framework Pix2Pix [48]. To further improve the invisibility, we leverage an extra adversarial loss ℓ_{adv} to minimize the domain gap between x_i^P and x_i^c , formally:

$$\ell_{adv} = \mathbb{E}_{x_i^c \in \mathbb{X}^c} \log(D(x_i^c)) + \mathbb{E}_{x_i^P \in \mathbb{X}^P} \log(1 - D(x_i^P)). \quad (6)$$

With the adversarial loss, the adversarial discriminator network D will act as a competitor to find the difference between x_i^P and x_i^c . Meanwhile, the injection network *IN* tries to generate the x_i^P in a more invisible way so that the discriminator cannot distinguish it from the clean image x_i^c .

2) *Guidance Extractor GE*: Only constrained by the invisibility loss, the injection network *IN* will easily ignore p_i and learn a trivial solution that outputs the original clean input directly. To ensure p_i to be hidden in x_i^c , we append an auxiliary guidance extractor network *GE* after the injection network to guide the injection process.

On the one hand, *GE* should be able to extract trigger pattern p_i out if feeding in the poisoned image x_i^P , which can be regarded as the reverse operation of injection. On the other hand, *GE* should not extract any trigger pattern out from the trigger-free images. With these two requirements, the training of *IN* and *GE* is indeed conducted in an adversarial way, so it is impossible for *IN* to degrade into a trivial solution. To achieve these two goals, we add two corresponding loss functions: the trigger extraction loss \mathcal{L}_{te} for poisoned images and the clean loss \mathcal{L}_{cl} for trigger-free images i.e.,

$$\begin{aligned} \mathcal{L}_{GE} &= \mathcal{L}_{te} + \lambda \cdot \mathcal{L}_{cl}, \\ \mathcal{L}_{te} &= \mathbb{E}_{x_i^P \in \mathbb{X}^P} [\|GE(x_i^P) - p_i\|_2], \\ \mathcal{L}_{cl} &= \mathbb{E}_{x_i^c \in \mathbb{X}^c} [\|GE(x_i^c) - \mathbf{C}\|_2], \end{aligned} \quad (7)$$

where \mathbf{C} is a clean map (shown in Figure 1) to indicate no trigger pattern and λ is one hyper-parameter to balance the two loss terms. Here we simply use the $L2$ reconstruction loss for both \mathcal{L}_{te} and \mathcal{L}_{cl} by default.

3) *Interference Layer*: To increase the extracting difficulty of the guidance extractor network and encourage the injection network to hide the trigger pattern in a more robust way, we further add an auxiliary interference layer between them. During training, this layer will randomly augment the poisoned image output from the injection network. By default, it consists of a sequence of common data augmentation operators $\{\mathcal{T}_{crop} \rightarrow \mathcal{T}_{resize} \rightarrow \mathcal{T}_{flip} \rightarrow \dots \rightarrow \mathcal{T}_{rot}\}$, and each poison image will be randomly augmented by each operator with the probability of 0.5.

To guarantee both the invisibility and robustness, the injection network IN and the guidance network GE are jointly trained, and the total loss function \mathcal{L}_{total} is:

$$\mathcal{L}_{total} = \mathcal{L}_{IN} + \gamma \cdot \mathcal{L}_{GE}. \quad (8)$$

By default, we set $\gamma = 1$. During training, we will randomly use different color values \mathcal{C}_i^p for different image x_i and encourage IN to be a general injection network that can embed different types of trigger patterns on the fly. When attacking one target model, we can choose one specific trigger pattern or multiple different trigger patterns by using different \mathcal{C}_i^p to enable single-label attack and multi-label attack, respectively.

V. EXPERIMENTS

In this section, we will first briefly introduce the implementation details we adopt, such as datasets, network structures, metrics, and default settings. After that, we will demonstrate the invisibility and robustness of the proposed Poison Ink, respectively. Then more network structures, datasets, and attack scenarios are considered to demonstrate the generality and flexibility of our method. Next, we showcase the resistance of Poison Ink to many state-of-the-art backdoor defense methods. Finally, some ablation studies are conducted to justify our design. Our source code will be released.

A. Implementation Details

1) *Datasets*: We consider 4 datasets for three types of classification tasks. For object recognition, we use CIFAR-10 [49] and ImageNet [50]; GTSRB [51] and VGG-Face [52] are utilized for traffic sign recognition and face recognition, respectively. For each dataset except CIFAR-10, we randomly select a portion of classes and resize the inputs for diversity, as shown in Table I. Because of the resource and space consideration, we mainly use the ImageNet and CIFAR-10 for comparison and CIFAR-10 for ablation. GTSRB and VGG-Face are used to demonstrate the generality of our method. Note that, The test data has no overlap with the training data, and the trigger pattern is input-aware and not seen during the training phase.

2) *Network Structures*: For trigger image generation, we simply adopt the UNet [53] and the PatchGAN [54] as the default network structure of IN and the discriminator D respectively, which are both widely used in many image-to-image tasks [48], [55], [56]. For guidance extractor GE , we design a simple auto-encoder-like network. The encoder consists of three residual blocks with stride 2, symmetrically, and the decoder consists of three residual upsampling blocks

TABLE I
THE STATICS OF DATASETS.

Dataset	# Labels	Input Size	# Training / Testing Images
CIFAR-10	10	32 × 32	50000 / 5000
GTSRB	13	64 × 64	4772 / 293
ImageNet	100	224 × 224	126689 / 5000
VGG-Face	500	224 × 224	135712 / 3147

to ensure the output resolution to be same as the input resolution. To further enhance its learning capacity, several residual blocks are also inserted between the encoder and the decoder. In Table II and Table III, we provide the details of the injection network IN and the guidance extractor GE respectively. Besides, we chop up each poisoned image into 16×16 patches for the discriminator D .

As for network structures of the victim classifier, we consider four popular recognition networks: ResNet-18 [57], ResNeXt [58], DenseNet [59] and VGG-19 [60]. In the following experiments, we adopt VGG-19 as the default network and mainly showcase the results on CIFAR-10 and ImageNet. The results of other network structures and datasets are used to demonstrate the generality of our method.

3) *Metrics*: We use Clean Data Accuracy (CDA) to evaluate the influence of backdoor attacks on the original tasks, and use Attack Success Rate (ASR) to evaluate the effectiveness of backdoor attacks. Specifically, CDA is the performance on the clean test set, *i.e.*, the ratio of trigger-free test images that are correctly predicted to their ground-truth labels; and ASR is the performance on the pre-defined poisoned test set, *i.e.*, the ratio of poisoned images that are correctly classified as the target attack labels.

For invisibility evaluation, we compare clean and poisoned images with three famous metrics, PSNR, SSIM, and LPIPS, where LPIPS adopts the features of the pre-trained AlexNet. Besides the above metrics, we also conduct a user study for human inspection testing.

4) *Default Settings for Training*: For trigger pattern generation, we use the Sobel operator [42] to extract the edge. We conduct comparisons with the single-label attack by default and inject the poison ink (R:80, G:160, B:80) into the edge area with the well-trained injection network IN . Then, 10% pollution rate is considered for all tasks, and the first class of each dataset is chosen as the target attack label. We train the injection network IN for 200 epochs by default. All the victim models are trained using the SGD optimizer with a momentum of 0.9 and an initial learning rate of 0.01, which is further set as 0.001 and 0.0001 at epoch 150 and epoch 200, respectively. For the other methods to compare with, we adopt the default setting in their official implementations.

B. Invisibility of Poison Ink

We first compare the invisibility of our method with many popular backdoor attack methods, and we follow their default implementation for fair comparisons.

In Figure 2, we showcase more visual comparison with other popular attack methods on ImageNet dataset and CIFAR-10 dataset. In detail, we can see that the poisoned image generated

TABLE II
THE DETAILED NETWORK STRUCTURE OF THE INJECTION NETWORK *IN*

Module Name	Layer Name	Kernel	Stride	Channel I/O	Normalization	Activation	Input
Encoder	conv1	4 x 4	2	6/64	BN	LeakyReLU	Cat[(Images, Trigger Pattern), 1]
	conv2	4 x 4	2	64/128	BN	LeakyReLU	conv1
	conv3	4 x 4	2	128/256	BN	LeakyReLU	conv2
	conv4	4 x 4	2	256/512	BN	LeakyReLU	conv3
	conv5	4 x 4	2	512/512	BN	LeakyReLU	conv4
Decoder	transpose_conv1	4 x 4	2	512/512	BN	ReLU	conv5
	transpose_conv2	4 x 4	2	1024/256	BN	ReLU	transpose_conv1 + conv4
	transpose_conv3	4 x 4	2	512/128	BN	ReLU	transpose_conv2 + conv3
	transpose_conv4	4 x 4	2	256/64	BN	ReLU	transpose_conv3 + conv2
	transpose_conv5	4 x 4	2	128/3	BN	Sigmoid	transpose_conv4 + conv1

TABLE III
THE DETAILED NETWORK STRUCTURE OF THE GUIDANCE EXTRACTOR *GE*

Module Name	Layer Name	Kernel	Stride	Channel I/O	Normalization	Activation	Input
Encoder	conv1	3 x 3	1	3/128	IN	ReLU	Poisoned Images
	conv2	3 x 3	1	128/128		ReLU	conv1
	conv3	3 x 3	2	128/128		ReLU	conv2
Residual Learning	res_block1	3 x 3	1	128/128	IN	ReLU	conv3
	res_block2	3 x 3	1	128/128	IN	ReLU	res_block1
	res_block3	3 x 3	1	128/128	IN	ReLU	res_block2
	res_block4	3 x 3	1	128/128	IN	ReLU	res_block3
	res_block5	3 x 3	1	128/128	IN	ReLU	res_block4
	res_block6	3 x 3	1	128/128	IN	ReLU	res_block5
	res_block7	3 x 3	1	128/128	IN	ReLU	res_block6
Decoder	transpose_conv1	4 x 4	2	128/128	IN	ReLU	res_block7
	transpose_conv2	3 x 3	1	128/128	IN	ReLU	transpose_conv1
	transpose_conv3	1 x 1	1	128/3	IN	-	transpose_conv2

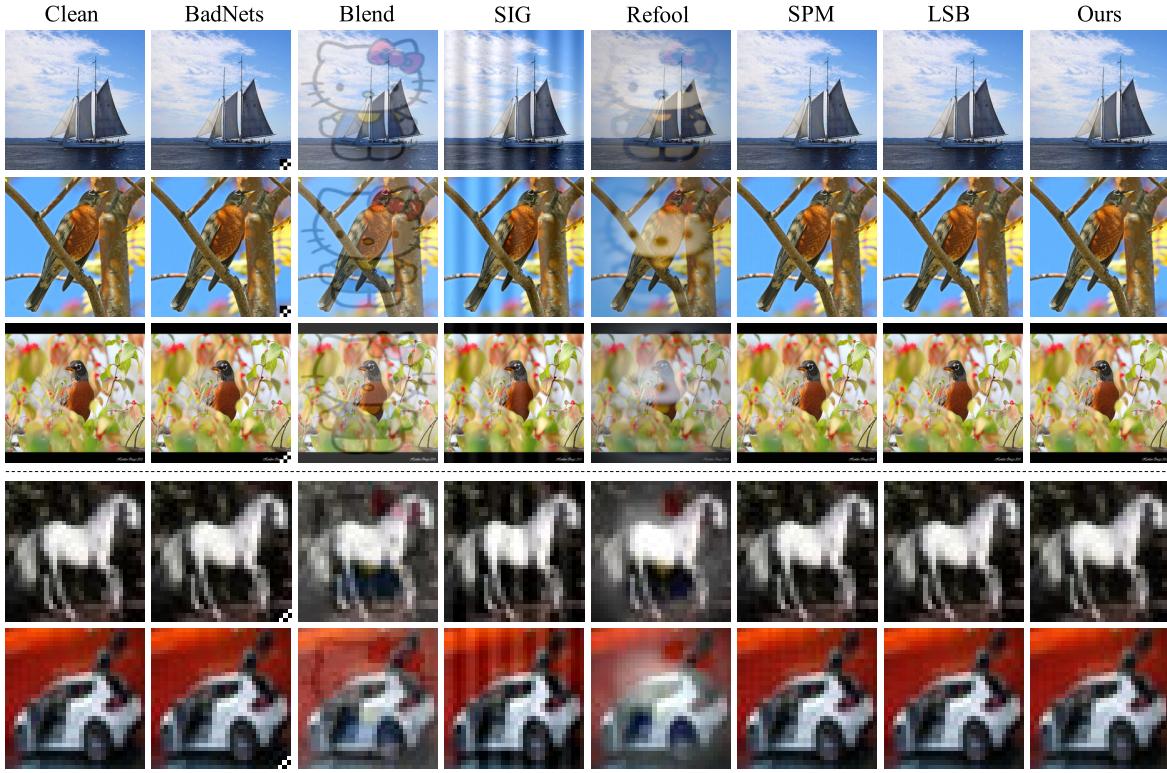


Fig. 2. Visual comparison with existing popular attack methods. The first three rows are examples on ImageNet dataset and the last two rows are examples on CIFAR-10 dataset. “Clean” denotes the original trigger-free image.

by BadNets [13], Blend [15], SIG [61] and Refool [24] can be easily distinguished from the clean images, due to the image-agnostic trigger pattern and simple embedding strategy. As for SPM [22], the checkboard-like pattern is also easily observed in the smooth region of the image. In contrast,

LSB [25] and Poison Ink achieve better invisibility, and the embedded poison is imperceptible.

In Table IV and Table V, we provide the quantitative comparison of invisibility on ImageNet dataset and CIFAR-10 dataset. Poison Ink outperforms the majority of attacks under

TABLE IV
COMPARISON OF INVISIBILITY (STEALTHINESS) WITH EXISTING POPULAR BACKDOOR ATTACK METHODS ON IMAGENET DATASET

Metric	BadNets [13]	Blend [15]	SIG [23]	Refool [24]	SPM [22]	LSB [25]	Ours
PSNR ↑	27.03	19.18	25.12	16.59	38.65	51.14	41.62
SSIM ↑	0.9921	0.7921	0.8988	0.7701	0.9665	0.9975	0.9915
LPIPS ↓	0.0149	0.2097	0.0532	0.2461	0.0022	0.0003	0.0020
Fooling Rate (%)	0	0	0	0	38.33	51.67	53.33

TABLE V
COMPARISON OF INVISIBILITY (STEALTHINESS) WITH EXISTING POPULAR BACKDOOR ATTACK METHODS ON CIFAR-10 DATASET

Metric	BadNets [13]	Blend [15]	SIG [23]	Refool [24]	SPM [22]	LSB [25]	Ours
PSNR ↑	25.68	21.29	25.12	19.38	38.94	51.13	42.95
SSIM ↑	0.9833	0.8810	0.8250	0.8243	0.9884	0.9991	0.9961
LPIPS ↓	0.0009	0.0240	0.0400	0.0397	0.0001	0.00001	0.0001
Fooling Rate (%)	0	0	0	0	16.67	43.33	46.67

TABLE VI
COMPARISON OF THE ROBUSTNESS AGAINST DIFFERENT DATA TRANSFORMATIONS WITH POPULAR ATTACK METHODS ON IMAGENET DATASET. “DT” DENOTES DATA TRANSFORMATION. “S&P” AND “C&R” MEANS PADDING AFTER SHRINKING AND RESIZING AFTER CROPPING, RESPECTIVELY. THE RED FONT REPRESENTS THE “WORST-CASE” PERFORMANCE.

Metric	CDA (%)						ASR (%)					
	None	Flip	S&P	Rot 15	C&R	Average	None	Flip	S&P	Rot 15	C&R	Average
BadNets [13]	80.78	80.32	73.90	78.52	75.82	77.87	100.00	99.98	31.46	6.08	55.34	58.57
Blend [15]	80.12	80.30	74.72	77.52	76.30	77.79	99.98	99.96	96.40	69.72	89.46	91.10
SIG [23]	80.42	80.34	74.82	77.90	76.40	77.98	100.00	99.98	51.84	54.20	68.94	74.99
Refool [24]	80.36	80.10	74.60	78.24	75.84	77.83	99.36	99.32	98.70	92.00	92.24	96.32
SPM [22]	80.42	80.10	73.16	77.60	76.10	77.48	99.94	99.92	53.94	1.04	0.90	51.15
LSB [25]	80.64	80.28	73.60	77.82	75.98	77.66	97.32	97.22	11.12	0.92	0.98	41.51
Ours	80.56	80.40	74.32	78.10	76.64	78.00	98.48	98.62	98.20	96.10	96.12	97.50

TABLE VII
COMPARISON OF THE ROBUSTNESS AGAINST DIFFERENT DATA TRANSFORMATIONS WITH POPULAR ATTACK METHODS ON CIFAR-10 DATASET. “DT” DENOTES DATA TRANSFORMATION. “S&P” AND “C&R” MEANS PADDING AFTER SHRINKING AND RESIZING AFTER CROPPING, RESPECTIVELY. THE RED FONT REPRESENTS THE “WORST-CASE” PERFORMANCE.

Metric	CDA (%)						ASR (%)					
	None	Flip	S&P	Rot 15	C&R	Average	None	Flip	S&P	Rot 15	C&R	Average
BadNets [13]	91.22	91.15	86.01	77.76	81.08	85.44	100	99.98	22.00	77.26	39.77	67.80
Blend [15]	93.32	93.01	89.89	86.97	85.28	89.69	99.82	99.79	92.39	98.95	74.34	93.06
SIG [23]	92.64	93.08	89.34	86.23	85.43	89.34	99.92	99.95	99.86	99.8	97.62	99.43
Refool [24]	92.40	92.57	88.72	86.20	83.34	88.65	93.87	93.64	89.59	93.22	93.63	92.79
SPM [22]	93.05	93.39	89.63	84.51	84.75	89.07	99.86	99.89	9.72	68.42	8.96	57.37
LSB [25]	87.86	87.22	84.28	81.26	77.54	83.63	14.93	15.36	15.14	16.35	17.03	15.76
Ours	92.92	92.77	89.60	87.09	85.15	89.51	99.92	99.97	83.12	96.94	87.46	93.48

all the evaluation metrics except LSB [25], which are consistent with the visual comparison. As for LSB, it also has good invisibility but is very fragile to common data transformations in the attacking stage (shown in Table VI and Table VII). Moreover, in Table VII, we observe that LSB fails on CIFAR-10 dataset, even without transformation-based pre-processing on its poisoned images. It may be because the trigger pattern of LSB on the CIFAR-10 dataset is regarded as random noise and ignored by the target classifier.

We further conduct a user study for human inspection testing. In detail, we randomly selected 50 clean images from ImageNet dataset and CIFAR-10 dataset and generated the corresponding 50 poisoned images for each backdoor attack method. Then in each question, we randomly display one image pair (one clean image and one corresponding poisoned image) and ask the user to select which one is the clean image. A total of 30 volunteers (12 females and 18 males, user ages range from 18 to 30, and all users are familiar

with backdoor attacks) are involved in our user study. Thus, there are 1500 answers for each attack method. We compare the fooling rate of each method as shown in the last row of Table IV and Table V. It can be observed that the fooling rate of LSB and our method both are close to 50%, a probability of random guessing, while the poisoned images generated by other remaining methods can be easily judged as unclean.

C. Influence on Pristine Performance

Besides the stealthiness of poisoned images, another critical aspect for evaluating the backdoor attack is its influence on pristine performance. In other words, when training the backdoor model with the mixture of the trigger set and the clean set, the model should keep its original performance on the clean set. In the left part of Table VI and Table VII, we show the Clean Data Accuracy (CDA) of backdoored models trained with the trigger set generated by different methods. It can

be seen that the backdoored models of Poison Ink have overall higher CDA than other baselines, demonstrating the stealthiness during the backdoor model training. Even in the worst case, the CDA is still comparable with other methods. Here, the CDA of the original model without backdoor attack is 80.84% on ImageNet dataset and 91.41% on CIFAR-10 dataset, respectively.

D. Robustness of Poison Ink

To address the limitation to data transformation, a corresponding solution [19] was proposed that conducting a data transformation on the training images with the trigger before feeding into the training process, which can be seen as adversarial training. In our threat model, the attacker cannot control the training strategy. Nevertheless, in this section, we train all backdoor attacks in such an enhanced training strategy for a fair comparison.

In the right part of Table VI, we give the ASR of different attack methods on ImageNet dataset without (None) or with data transformation attack, respectively. With the data transformation attack, the poisoned images will be pre-processed by different data transformation techniques and then fed into the victim model for testing. We also provide the results in the average case and the worst case in order to evaluate these attack methods comprehensively. It can be seen that, compared with other popular attack methods, Poison Ink achieves a higher ASR in both the average and worst cases, when facing different transformations. In contrast, some baseline methods like SPM and LSB will totally fail when some specific transformations are applied. Compared to Refool, our method outperforms narrowly in terms of robustness but outperforms in stealthiness by a large margin, as shown in Table IV.

Although the enhanced training strategy improves the robustness to some extent, overfitting will be introduced. Taking SPM as an example, such a strategy improves the robustness against rotation but fails on other data transformations. We explain that SPM tends to only focus on getting robust against rotation. Conversely, Poison Ink mitigates such overfitting and achieves an outstanding or comparable performance in terms of the CDA and the ASR.

As shown in Table VII, we provide the quantitative comparison of robustness to data transformations on CIFAR-10 dataset. Overall, Poison Ink guarantees comparable robustness on CIFAR-10 dataset with many existing attack methods such as Blend and SIG, which nevertheless have worse invisibility compared with our method (shown in Figure 2 and Table V). In Figure 3 and Figure 4, we plot radar charts to illustrate that our method achieves a much better balance between visual quality and robustness.

E. Comparison With More Invisible Backdoor Attacks

Besides the classic backdoor attacks mentioned above, we further consider some recent invisible backdoor attacks, such as WaNet [62], FTrojan [63], and Advdoor [64]. In detail, WaNet generates backdoor images via subtle image warping, and FTrojan injects mid- and high-frequency triggers

TABLE VIII
THE COMPARISON RESULTS WITH RECENT INVISIBLE BACKDOOR ATTACK METHODS.

Methods	CDA(%)	ASR(%)	PSNR ↑	SSIM ↑	LIPIS ↓
WaNet [62]	83.47	26.10	32.15	0.9822	0.0042
FTrojan [63]	87.47	98.21	41.10	0.9915	0.0001
Advdoor [64]	87.69	99.71	22.97	0.9101	0.0109
ours	89.51	93.48	42.95	0.9961	0.0001

TABLE IX
COMPARISON OF THE ROBUSTNESS AGAINST DIFFERENT DATA TRANSFORMATIONS WITH FTROJAN [63] ON CIFAR-10 DATASET. † DENOTES WITHOUT THE ENHANCED TRAINING STRATEGY.

Methods	None	Flip	S&P	Rot 15	C&R	Average
FTrojan [63]	97.57	97.35	99.07	97.59	99.46	98.21
ours	99.92	99.97	83.12	96.94	87.46	93.48
FTrojan [63] †	100	100	9.99	77.99	9.56	59.51
ours †	99.92	99.98	39.82	96.24	82.39	83.67

in each block with medium magnitude, both of which utilize input-aware trigger patterns. Advdoor generates triggers by the targeted universal adversarial perturbation (TUAP). Specifically, some inputs in a category are used to obtain input-specific adversarial perturbations firstly, which are further integrated to generate the final TUAP for the target category. Such TUAPs belong to the static trigger patterns, which are further directly imposed onto the clean images to generate final triggers.

All comparison experiments are conducted on CIFAR-10 dataset, and we train all backdoored models with the enhanced training strategy [19] for a fair comparison. We calculate the average CDA and ASR under different data transformations. As shown in Table VIII and Figure 5, Advdoor is not stealthy enough due to its static trigger patterns, while WaNet is fragile to data transformations. Compared to Advdoor and WaNet, FTrojan and our method perform well in both stealthiness and robustness. If training backdoored model without the enhanced strategy, poison ink achieves stronger robustness than FTrojan (ASR: 83.67% vs. 59.51% in Table IX).

F. Generality and Flexibility of Poison Ink

To demonstrate the generality of our attack to different datasets and network structures, we run the controlled experiments with VGG-19 on different datasets and different network structures on CIFAR-10 dataset. Poison Ink is imperceptible among various datasets, with visual and quantitative results shown in Figure 6 and Table XI, respectively. In Figure 6, we can see that the Guidance Extractor informs the Injection Network to preserve the edge trigger well. Besides, Table X shows that our attack still guarantees a high ASR on different datasets and architectures while only decreasing the pristine performance slightly.

For the flexibility of Poison Ink, we further consider three more attack scenarios: training the target model from the pre-trained model, with multiple target labels, and under different ratios. All these experiments are conducted on CIFAR-10 dataset. In the former scenario, we set the initial learning rate as 0.001, and we observe in the left of Figure 7 that the ASR

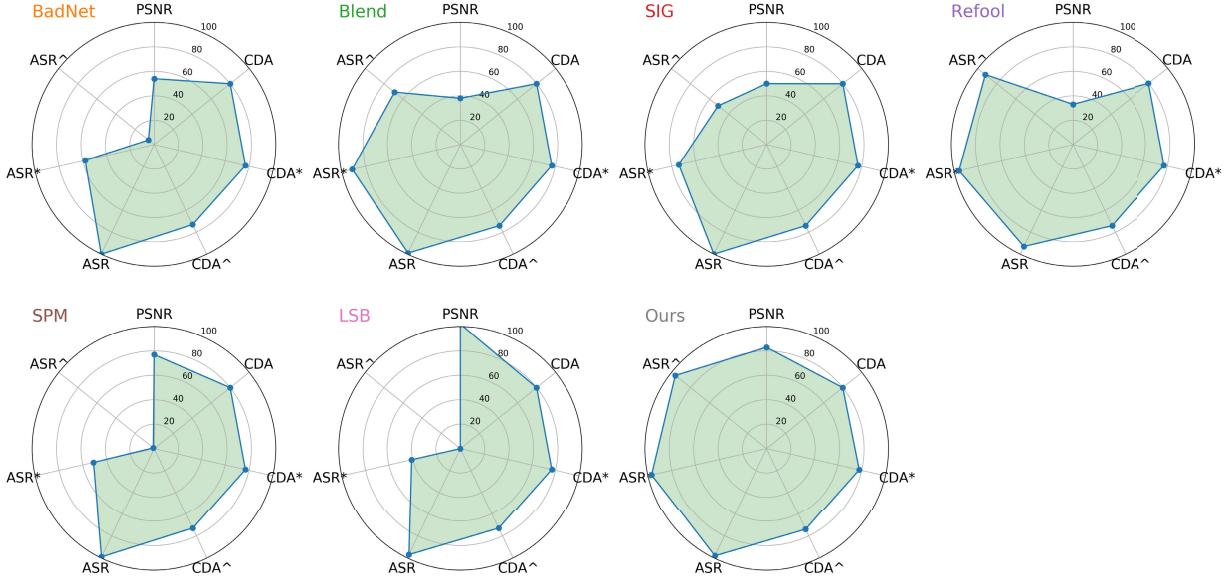


Fig. 3. Comparison with popular attack methods on ImageNet dataset in terms of both stealthiness and robustness. PSNR is $2 \times$ original value. CDA and ASR mean the performance under no data transformation, CDA* and ASR* denote the average performance under different data transformations, and CDA[^] and ASR[^] are calculated under the worst case.

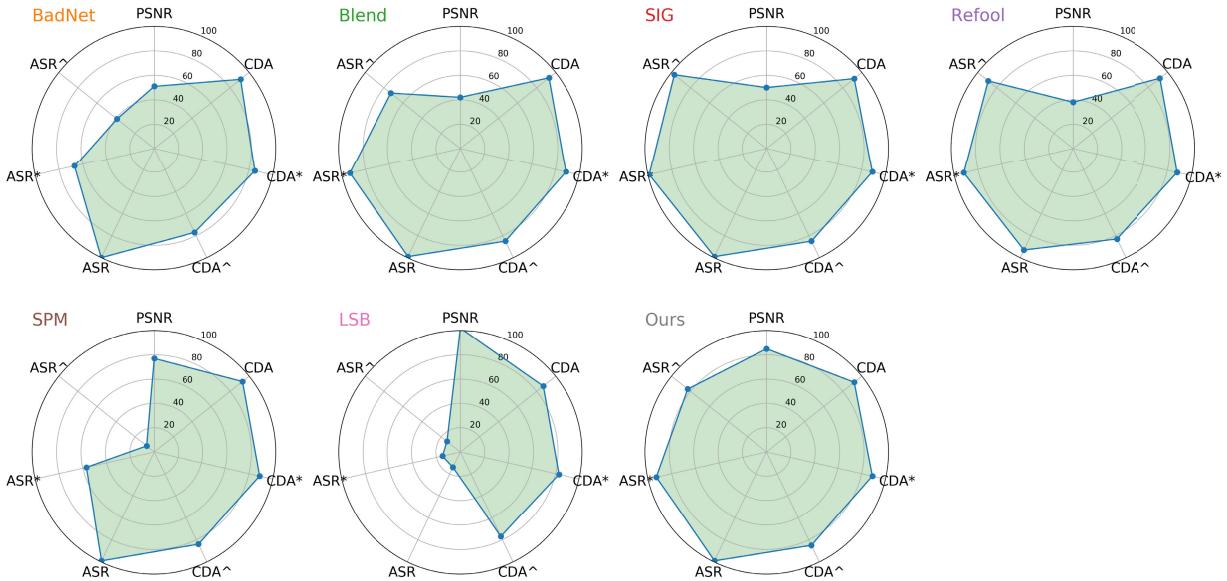


Fig. 4. Comparison with popular attack methods on CIFAR-10 dataset in terms of both stealthiness and robustness. PSNR is $2 \times$ original value. CDA and ASR mean the performance under no data transformation, CDA* and ASR* denote the average performance under different data transformations, and CDA[^] and ASR[^] are calculated under the worst case.

TABLE X

THE PERFORMANCE (OMA / CDA / ASR (%)) OF POISON INK ON DIFFERENT DATASETS AND DIFFERENT NETWORK STRUCTURES. WE ADOPT VGG-19 AND CIFAR-10 AS DEFAULT NETWORK ARCHITECTURE AND DEFAULT DATASET, RESPECTIVELY. “OMA” DENOTES ACCURACY OF THE ORIGINAL MODEL TRAINED ON TOTAL CLEAN DATASET

Settings	GTSRB	VGG-Face	ResNet-18	ResNeXt	DenseNet
Performance	87.71 / 88.74 / 98.63	92.73 / 92.41 / 99.44	91.41 / 90.87 / 99.89	95.53 / 95.05 / 100	90.06 / 89.47 / 99.76

achieves nearly 100% after only 16 epochs. For the multiple-label attack, we inject 10 different poison ink (color) to generate the corresponding poisoned image for 10 target labels

of CIFAR-10 dataset. As displayed in the middle of Figure 7, ASR for most target label attacks is comparable with single label attack, and so is the CDA (single:93.17, multiple:91.91).



Fig. 5. Visual examples of a poisoned image generated by invisible backdoor attack methods.

TABLE XI
QUANTITATIVE RESULTS OF THE STEALTHINESS OF OUR
POISON INK ON DIFFERENT DATASETS

Invisibility	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
CIFAR-10	42.95	0.9961	0.0001
GTSRB	38.05	0.9965	0.0001
ImageNet	41.62	0.9915	0.0020
VGG-Face	45.34	0.9916	0.0011

Besides, different ratios are also considered, and we can see on the right of Figure 7 that Poison Ink attacks successfully under different ratios, which will be further discussed in the ablation study.

G. Resistance to Defense Techniques

To resist backdoor attacks, many different defense techniques have been proposed recently. In this section, we will test the resistance of Poison Ink against different defense techniques, which are further categorized as data-based defense, model-based defense, and defense with the meta classifier.

1) *Data-Based Defense*: *Februus* [31] utilizes Grad-CAM [18] to visualize the attention map of the target image and regards the area with the highest score as the trigger region, then removes this region and restores it with image inpainting techniques. We first provide the attention map in Figure 8. During visualization, we feed the clean model and our infected model with both clean images and the corresponding poisoned images generated by Poison Ink. We find that the infected model also focuses on the main content area of the input, which is similar to the clean model. Then, we remove such a region and directly restore it with the same region of the corresponding clean image. After trigger pattern removal, the ASR of Poison Ink still guarantees 98.84% and 98.01% on CIFAR-10 and ImageNet, respectively.

STRIP [30] blends input images with a set of clean images from different classes and compares the entropy of the prediction before and after blending. As shown in Figure 10a, the entropy distribution of Poison Ink looks very similar to that of the clean model, which means that Poison Ink behaves normally as the benign (clean) model.

2) *Model-Based Defense*: For model-based defense, we first try the totally white-box defense *Fine-pruning* [32]. As shown in Figure 10b, the ASR of Poison Ink remains 92.6% with a 50% pruning rate on the ImageNet, while the performance on clean data decreases significantly. On CIFAR-10 dataset, we observe that the ASR of Poison Ink still stays at 100% even with a 90% pruning rate. Next, we run the official code of *ABS* [34] on the only supported CIFAR-10 dataset, and find *ABS* fails to detect our method. In Figure 9, we find

that *Activation Clustering* [33] can infer the infected class (“airplane”) by a larger silhouette score, which is based on the assumption that the infected class will induce more overfitting. However, it cannot find the true poisoned images with a large false alarm (TPR:0.0002, FPR:0.984). Moreover, poison ink can pollute all labels, while *Activation Clustering* cannot handle this case, as shown in the bottom of Figure 9.

Then, we consider *Neural Cleanse* [35], which is designed for small and static triggers. It first reverses the trigger pattern based on the optimization method, then runs an anomaly detection among all reversed trigger patterns from each label. It defines a model as the infected model by Anomaly Index larger than the threshold $t = 2$. As shown in Figure 10c, the Anomaly Index for our backdoored model is both below the threshold on CIFAR-10 dataset and ImageNet dataset. Besides, the infected label found by *Neural Cleanse* is empty on CIFAR-10 dataset, and is wrong on ImageNet Dataset (#97 not #0). Based on *Neural Cleanse*, *TABOR* [36] adds more constraints for the reversed trigger pattern during the optimization process, which reduces the false positive rate but costs more computation time. The corresponding Anomaly Index on CIFAR-10 dataset is 1.5938 (clean mode:1.0806), and some visual results of reversed triggers are further shown in Figure 10d. On ImageNet, *TABOR* costs 7 hours for a single label, and there still exist false alarms of the Anomaly Index (clean model:4.1593; Poison Ink:2.3125).

Rather than using the single reversed trigger, *MESA* [37] models the trigger distribution and uses many generated triggers to implement backdoor removal. Once our backdoored model attacks all the labels, it can only find partially polluted labels, and our method can still keep a high ASR after backdoor removal, as shown in Figure 10. For *TND*, it provides a faster detection in the data-scarce scenario, but it can also be evaded by Poison Ink easily.

3) *Meta Classifiers*: We further try the recent meta classifier based defense method *ULPs* [39] on CIFAR-10 dataset as an example. *ULPs* classifies the suspect model as “clean” or “corrupted” by feeding the universal patterns to the suspect model and analyzing its output. The fewer the universal patterns required, the weaker the backdoor attack is. In general, 5 Universal Litmus Patterns (ULPs) can detect backdoor attacks successfully. The experiment shows that our backdoored model is classified as the clean one even with 10 ULPs.

H. Adaptive Robustness Evaluation

Adaptive robustness evaluation has become the standard in adversarial machine learning in recent years [65], [66]. Here we evaluate the resistance of Poison Ink against some adaptive defense techniques, taking into account the fact that trigger patterns are not static and may be hidden in the image edges.

We first test the robustness of Poison Ink to adaptive pre-processing operations such as Gaussian Noise and Gaussian Blur, which may mitigate the Poison Ink by disrupting the image edges. In addition, we adopt three data augmentations used in [67] for sanitizing backdoor attacks. In detail, Mixup means blending inputs with clean images, while Cutout and CutMix denote that patches of inputs are randomly cut and

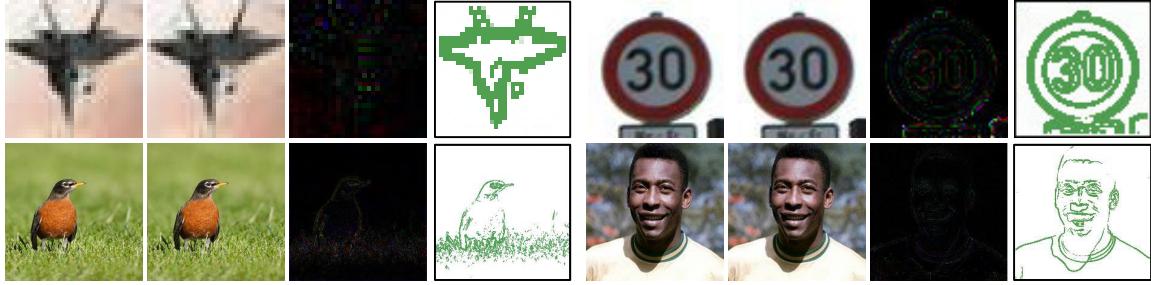


Fig. 6. Visual examples of Poison Ink on different datasets. For each dataset, the three images denote clean image, poisoned image, their $10\times$ difference, and the original edge trigger extracted by *GE*, respectively.

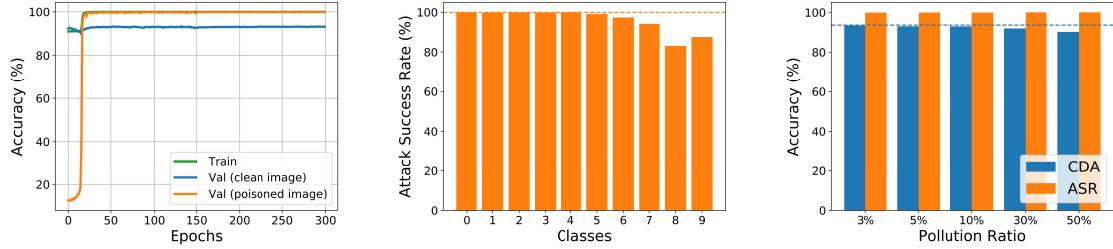


Fig. 7. The flexibility of Poison Ink. **Left:** the train-val convergence curve for training from pre-trained VGG-19; **Middle:** the attack success rate (ASR) of Poison Ink trained with multiple target labels, the orange dot line denotes the ASR of Poison Ink trained with single target label; **Right:** The performance (CDA/ASR (%)) of infected model with different pollution ratios The blue dot line denotes the performance of original clean model. All three experiments are conducted on the CIFAR-10 dataset.

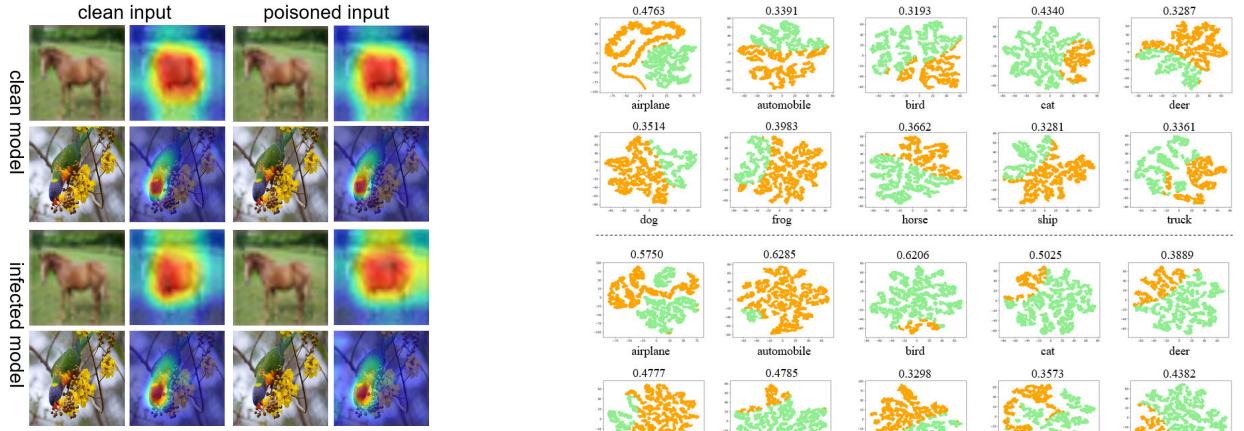


Fig. 8. The Grad-CAM of clean input and poisoned input according to clean model and infected model. As shown in the figure, Grad-CAM fails to detect trigger regions of those generated by our attack, which is indistinguishable with the benign case.

pasted with nothing or clean images, respectively. We display quantitative results on ImageNet dataset and CIFAR-10 dataset in Table XII. Results suggest that Poison Ink can still keep a relatively high ASR after suffering from such pre-processings.

RCA-SOC [28] is proposed as a defense against evasion attacks by refocusing on critical areas and strengthening object contours. Specifically, pixel channel attention is adopted to focus on the critical feature areas, and pixel plane attention is designed to focus more on feature pixels, where the key pixels of the image are emphasized and the adversarial perturbed pixels are weakened. We utilize RCA-SOC to filter the inputs from CIFAR-10 dataset and find the CDA degrades heavily

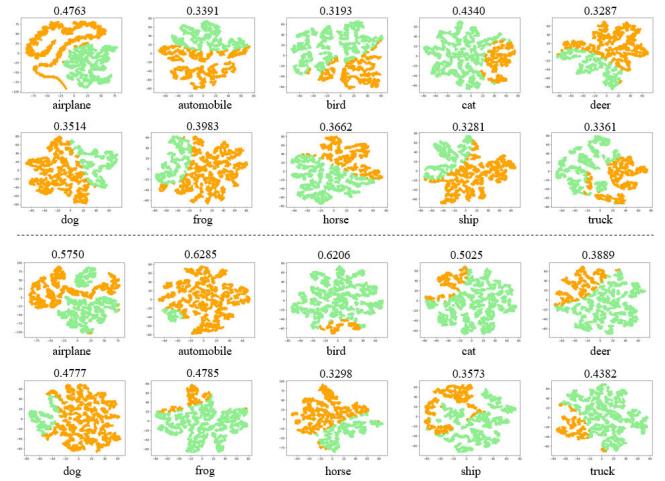


Fig. 9. The activation clustering results for each class on the CIFAR-10 dataset. Silhouette score is given on the top of each subfigure, and a high silhouette score indicates that two clusters fit the data well. The top part is for the single label attack while the bottom part is for the multiple label attack.

(from 92.92% to 11.47%) while the ASR (77.75%) is still acceptable. We further showcase some visual examples in Figure 11, and we find that the perturbed pixels by Poison Ink are preserved to some extent after RCA-SOC, which makes our attack still succeed.

To remove the poison information more accurately, we further assume that the defender can utilize edge extraction algorithms to locate the edge area and then replace this area with another image. For edge extraction algorithms, we consider the same algorithm (“Sobel”) used for the trigger pattern

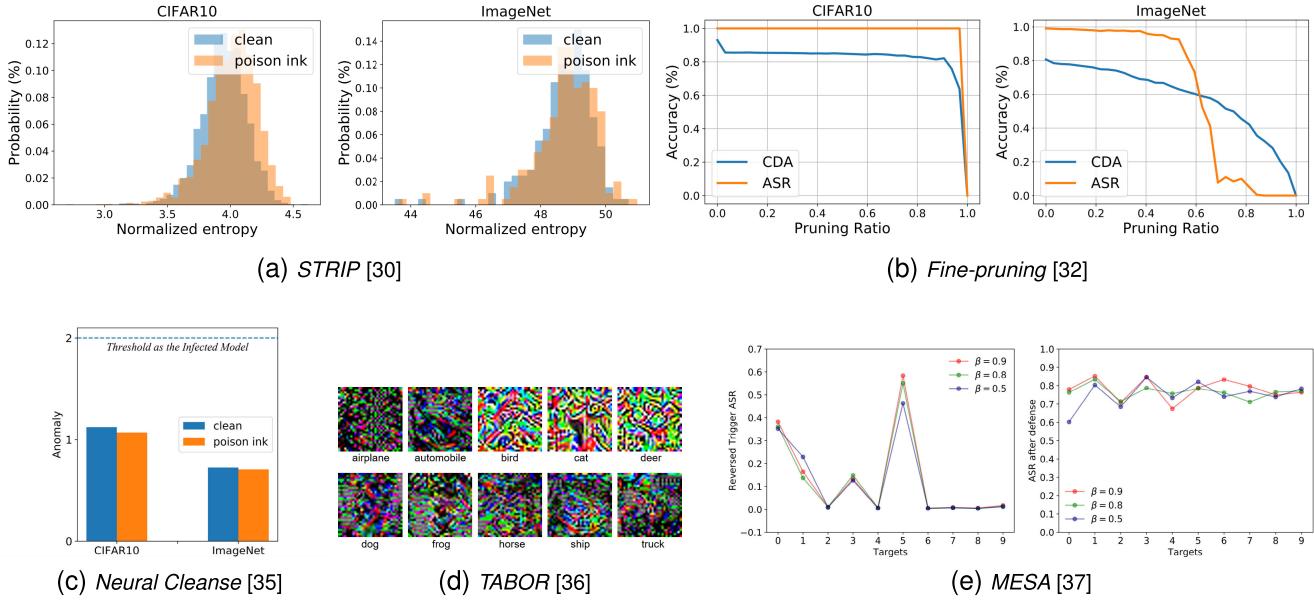


Fig. 10. Some visual and quantitative results showing the resistance of Poison Ink to some state-of-the-art defense techniques.

TABLE XII

ROBUSTNESS OF POISON INK TO OTHER ADAPTIVE PRE-PROCESSING OPERATIONS. THE VARIANCE OF GAUSSIAN NOISE IS SET AS 0.01, THE KERNEL SIZE OF GAUSSIAN BLUR IS 3×3 , AND THREE PRE-PROCESSING OPERATIONS USED IN [67] ARE ALSO ADOPTED. THE PERFORMANCE DEGRADATION IS SHOWN IN THE BRACKET

Dataset	ImageNet		CIFAR-10	
Pre-processing	CDA (%)	ASR (%)	CDA (%)	ASR (%)
Gaussian Noise	56.28 (-24.28)	88.36 (-10.12)	55.17 (-37.75)	89.49 (-10.43)
Gaussian Blur	73.84 (-6.72)	90.36 (-8.12)	52.53 (-40.39)	40.60 (-59.32)
Mixup	22.78 (-57.78)	73.62 (-24.86)	45.77 (-47.15)	85.74 (-14.18)
Cutout	56.44 (-24.12)	90.40 (-8.08)	65.82 (-27.10)	72.39 (-27.53)
CutMix	47.24 (-33.32)	90.62 (-7.86)	61.79 (-31.13)	85.11 (-14.11)



Fig. 11. Visual examples of inputs processed by RCA-SOC (RS) [28].

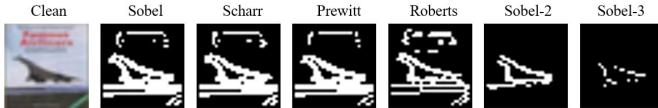


Fig. 12. Visual examples of edge structures extracted by different edge extraction algorithms.

generation, other algorithms (“Scharr”, “Prewitt”, “Roberts”) with the same parameters, and the same algorithm with different parameters (“Sobel-2”, “Sobel-3”). Some visual examples are provided in Figure 12. For the replacement of the edge area, we first consider the above-mentioned solution, namely, directly using the corresponding original clean image (denote as “ori”), which however is not accessible in practice. Nevertheless, the defender can leverage an inpainting algorithm to restore the removed area [31]. Here, we use a pure image with a constant RGB value (e.g., 125) as the restored area, which

TABLE XIII

THE PERFORMANCE (CDA / ASR) IN THE CASE OF REPLACING EDGE AREAS WITH THE CORRESPONDING ORIGINAL (ORI) CLEAN IMAGE OR THE PURE IMAGE WITH A CONSTANT RGB VALUE. WE UTILIZE DIFFERENT EDGE EXTRACTION ALGORITHMS TO LOCATE THE EDGE AREA.

	(%)	Sobel	Ccharr	Prewitt	Roberts	Sobel-2	Sobel-3
CDA_pure	60.98	60.71	61.96	65.3	84.17	90.49	
ASR_ori	66.39	65.57	67.76	74.57	94.56	99.14	
ASR_pure	57.28	56.59	58.34	62.15	88.25	97.89	

can be regarded as the worst case of inpainting. As shown in Table XIII, Poison Ink can still achieve above 55% ASR in all cases mentioned above. We explain that the poison information is not fully removed in such adaptive defenses. Besides, in the case with a relatively low ASR, the corresponding CDA also degrades a lot, which is unacceptable for the defender.

We also consider adversarial training [68], which is proposed to defend against adversarial attacks. In detail, Geiping *et al.* [69] extend adversarial training to defend against backdoor attacks by generating poisoned images during training and injecting them into training batches. We conduct the corresponding experiments on CIFAR-10 dataset, adopt $s = 0.75$ as the poison immunity, and consider the *from-scratch* scenario. To generate poisoned images,

TABLE XIV
COMPARISON WITH OTHER ATTACK METHODS UNDER DIFFERENT POLLUTION RATIO. WE TAKE RESULTS ON CIFAR-10 DATASET FOR EXAMPLE

Metric	Ratio	BadNets [13]	Blend [15]	SIG [23]	Refool [24]	SPM [22]	LSB [25]	Ours
CDA	1%	88.20	89.55	89.71	89.72	89.58	89.26	89.25
	3%	87.38	89.89	89.74	89.20	88.98	88.18	89.65
	5%	87.13	89.60	89.64	89.16	88.90	86.98	89.69
	10%	85.44	89.69	89.34	88.65	89.07	83.63	89.51
ASR	1%	61.76	84.79	98.79	75.86	57.15	10.32	10.51
	3%	66.55	89.39	99.23	87.16	58.53	10.91	94.22
	5%	65.36	90.99	99.47	89.79	57.69	11.67	93.58
	10%	67.80	93.06	99.43	92.79	57.37	15.76	93.48

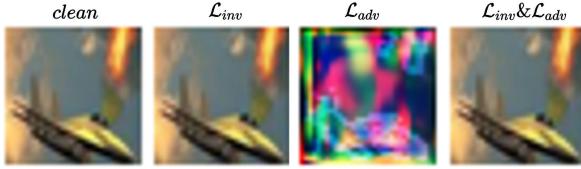


Fig. 13. Output examples of injection network IN trained with different loss constraints on CIFAR-10 dataset.

we utilize the most similar watermark-based method [12] provided by the official code.¹ However, the ASR of Poison Ink even increases by 6.49% after adversarial training. The possible reason is that adversarial training forces the model to capture more robust features, making the edge structures learned better. This finding warrants more study in future work, as adversarial training is one of the only truly effective methods we have to date to defend against adversarial attacks.

I. Ablation Study

1) *Importance of Our Design:* In this experiment, we take CIFAR-10 dataset as an example to ablate our design. First, we use an input-agnostic image rather than the edge structure as the trigger pattern and leverage the same invisible injection network to create the poisoned image. However, we find this strategy will totally fail, and the ASR is only 13.42% even without pre-processing. Recently, Li *et al.* [70] proposed an invisible input-specific backdoor attack, whose framework is similar to us but does not consider the structure information. To double confirm it, we also conduct comparative experiments on CIFAR-10 dataset and find it fails as expected.

Second, we try to discard the interference layer during the injection network training. Under this setting, the ASR degrades from 99.95% to 43.73% when faced with resizing, demonstrating the importance of the interference layer.

2) *Influence of Pollution Ratio:* In our default setting, we set the pollution ratio as 10%. In Table XIV, we further try more pollution ratios on CIFAR-10 dataset and show the robustness and the stealthiness of different methods, where we use the average CDA and ASR to represent the robustness against data transformations. There exists a trade-off between the CAD and ASR under different ratios. Overall, compared with other methods, our method performs well both in stealthiness and robustness in most cases.

¹<https://github.com/JonasGeiping/data-poisoning>

TABLE XV

QUANTITATIVE RESULTS OF THE STEALTHINESS WITH DIFFERENT LOSS CONSTRAINTS ON THE CIFAR-10 DATASET

Loss Constraints	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
\mathcal{L}_{inv}	41.24	0.9959	0.0002
\mathcal{L}_{adv}	8.18	0.1482	0.2870
$\mathcal{L}_{inv} \& \mathcal{L}_{adv}$	42.95	0.9961	0.0001

However, poison ink will fail under a meager pollution ratio like 1%. In our threat model, we cannot control the training strategy. We try more invisible backdoor attacks such as WaNet [62] and FTrojan [63] under the 1% pollution ratio, and find they also fail with a low ASR in our threat model (WaNet: 10.83% and FTrojan: 10.92%). If we control the training strategy like in WaNet [62], namely, adding poisoned images in every training batch, Poison Ink can succeed with above 90% ASR. Besides, if we first train the model only on trigger images and then fine-tune it on clean images, we can also achieve above 90% ASR. In conclusion, the model needs more information to remember the trigger pattern when it is stealthy enough.

3) *Influence of Loss Constraints on Stealthiness:* In our default setting, we utilize both invisibility loss \mathcal{L}_{inv} and adversarial loss \mathcal{L}_{adv} to achieve desirable stealthiness. As shown in Table XV, appending adversarial loss \mathcal{L}_{adv} after invisibility loss \mathcal{L}_{inv} can improve the image quality slightly, and only using \mathcal{L}_{adv} will cause poor image quality, where injection network mainly focuses on the high-level information of images. Some visual results are showcased in Figure 13. In practice, adopting or discarding the adversarial loss is flexible, which depends on the desired balance between stealthiness and robustness.

VI. CONCLUSION

In this paper, we point out the limitations of existing backdoor attacks regarding stealthiness and robustness. To address such limitations, we propose a new backdoor attack method ‘‘Poison Ink’’. It utilizes the image structure as the carrier of poison information to generate trigger patterns and leverage a deep injection network to hide the trigger patterns into the cover images in an invisible way. Extensive experiments demonstrate that Poison Ink is superior to existing methods in stealthiness, robustness, generality and flexibility. Besides, Poison Ink is resistant to many state-of-the-art defense techniques. It is interesting to explore backdoor attacks in the frequency domain, and we leave it as future work.

REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
- [2] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.
- [3] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5017–5032, Dec. 2015.
- [4] M. Elad and P. Milanfar, "Style transfer via texture synthesis," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2338–2351, May 2017.
- [5] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [6] C. Szegedy *et al.*, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.
- [7] Y. Dong *et al.*, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9185–9193.
- [8] A. Mustafa, S. H. Khan, M. Hayat, J. Shen, and L. Shao, "Image super-resolution as a defense against adversarial attacks," *IEEE Trans. Image Process.*, vol. 29, pp. 1711–1724, 2020.
- [9] L. Wang and K.-J. Yoon, "PSAT-GAN: Efficient adversarial attacks against holistic scene understanding," *IEEE Trans. Image Process.*, vol. 30, pp. 7541–7553, 2021.
- [10] Z. Che *et al.*, "Adversarial attack against deep saliency models powered by non-redundant priors," *IEEE Trans. Image Process.*, vol. 30, pp. 1973–1988, 2021.
- [11] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 19–35.
- [12] A. Shafahi *et al.*, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Proc. NeurIPs*, vol. 31, 2018, pp. 6106–6116.
- [13] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "BadNets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.
- [14] Y. Liu *et al.*, "Trojaning attack on neural networks," in *Proc. NDSS*. The Internet Society, 2018.
- [15] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," 2017, *arXiv:1712.05526*.
- [16] Y. Gao *et al.*, "Backdoor attacks and countermeasures on deep learning: A comprehensive review," 2020, *arXiv:2007.10760*.
- [17] A. S. Rakin, Z. He, and D. Fan, "TBT: Targeted neural network attack with bit trojan," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13198–13207.
- [18] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.
- [19] Y. Li, T. Zhai, B. Wu, Y. Jiang, Z. Li, and S. Xia, "Rethinking the trigger of backdoor attack," 2020, *arXiv:2004.04692*.
- [20] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*. Cham, Switzerland: Springer, 2014, pp. 818–833.
- [21] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," 2018, *arXiv:1811.12231*.
- [22] H. Zhong, C. Liao, A. C. Squicciarini, S. Zhu, and D. Miller, "Backdoor embedding in convolutional neural network models via invisible perturbation," in *Proc. 10th ACM Conf. Data Appl. Secur. Privacy*, Mar. 2020, pp. 97–108.
- [23] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in CNNs by training set corruption without label poisoning," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 101–105.
- [24] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," 2020, *arXiv:2007.02343*.
- [25] S. Li, M. Xue, B. Zi Hao Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," 2019, *arXiv:1909.02742*.
- [26] J. Chen *et al.*, "FineFool: A novel DNN object contour attack on image recognition based on the attention perturbation adversarial technique," *Comput. Secur.*, vol. 104, May 2021, Art. no. 102220.
- [27] H. Na, G. Ryu, and D. Choi, "Adversarial attack based on perturbation of contour region to evade steganalysis-based detection," *IEEE Access*, vol. 9, pp. 122308–122321, 2021.
- [28] J. Chen, H. Zheng, R. Chen, and H. Xiong, "RCA-SOC: A novel adversarial defense by refocusing on critical areas and strengthening object contours," *Comput. Secur.*, vol. 96, Sep. 2020, Art. no. 101916.
- [29] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Proc. NeurIPs*, 2018, pp. 8000–8010.
- [30] 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 *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, Dec. 2019, pp. 113–125.
- [31] B. Gia Doan, E. Abbasnejad, and D. C. Ranasinghe, "Februous: Input purification defense against trojan attacks on deep neural network systems," 2019, *arXiv:1908.03369*.
- [32] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Proc. RAID*. Cham, Switzerland: Springer, 2018, pp. 273–294.
- [33] B. Chen *et al.*, "Detecting backdoor attacks on deep neural networks by activation clustering," 2018, *arXiv:1811.03728*.
- [34] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "ABS: Scanning neural networks for back-doors by artificial brain stimulation," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 1265–1282.
- [35] B. Wang *et al.*, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *Proc. SP*, 2019, pp. 707–723.
- [36] W. Guo, L. Wang, X. Xing, M. Du, and D. Song, "TABOR: A highly accurate approach to inspecting and restoring trojan backdoors in AI systems," 2019, *arXiv:1908.01763*.
- [37] X. Qiao, Y. Yang, and H. Li, "Defending neural backdoors via generative distribution modeling," 2019, *arXiv:1910.04749*.
- [38] R. Wang, G. Zhang, S. Liu, P.-Y. Chen, J. Xiong, and M. Wang, "Practical detection of trojan neural networks: Data-limited and data-free cases," 2020, *arXiv:2007.15802*.
- [39] S. Kolouri, A. Saha, H. Pirsiavash, and H. Hoffmann, "Universal litmus patterns: Revealing backdoor attacks in CNNs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 301–310.
- [40] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," 2020, *arXiv:2007.08745*.
- [41] J. Zhang *et al.*, "Exploring structure consistency for deep model watermarking," 2021, *arXiv:2108.02360*.
- [42] R. O. Duda *et al.*, *Pattern Classification and Scene Analysis*, vol. 3. New York, NY, USA: Wiley, 1973.
- [43] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
- [44] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proc. ECCV*, 2018, pp. 657–672.
- [45] J. Zhang *et al.*, "Model watermarking for image processing networks," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 12805–12812.
- [46] H. Fang *et al.*, "Deep template-based watermarking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 4, pp. 1436–1451, Apr. 2020.
- [47] J. Zhang *et al.*, "Deep model intellectual property protection via deep watermarking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4005–4020, Aug. 2021.
- [48] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.
- [49] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," Citeseer, 2009.
- [50] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [51] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: A multi-class classification competition," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 1453–1460.
- [52] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," Brit. Mach. Vis. Assoc., 2015.
- [53] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*. Cham, Switzerland: Springer, 2015, pp. 234–241.
- [54] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

- [55] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.
- [56] D. Chen *et al.*, "Controllable image processing via adaptive FilterBank pyramid," *IEEE Trans. Image Process.*, vol. 29, pp. 8043–8054, 2020.
- [57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [58] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.
- [59] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE CVPR*, Jul. 2017, pp. 4700–4708.
- [60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [61] S. Baluja, "Hiding images in plain sight: Deep steganography," in *Proc. NeurIPS*, 2017, pp. 2069–2079.
- [62] A. Nguyen and A. Tran, "WaNet—Imperceptible warping-based backdoor attack," 2021, *arXiv:2102.10369*.
- [63] T. Wang, Y. Yao, F. Xu, S. An, H. Tong, and T. Wang, "Backdoor attack through frequency domain," 2021, *arXiv:2111.10991*.
- [64] Q. Zhang, Y. Ding, Y. Tian, J. Guo, M. Yuan, and Y. Jiang, "AdvDoor: Adversarial backdoor attack of deep learning system," in *Proc. ISSTA*, 2021, pp. 127–138.
- [65] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1633–1645.
- [66] N. Carlini *et al.*, "On evaluating adversarial robustness," 2019, *arXiv:1902.06705*.
- [67] E. Borgia *et al.*, "Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 3855–3859.
- [68] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.
- [69] J. Geiping, L. H. Fowl, G. Somepalli, M. Goldblum, M. Moeller, and T. Goldstein, "What doesn't kill you makes you robust(er): How to adversarially train against data poisoning," 2022. [Online]. Available: <https://openreview.net/forum?id=VMuenFh7IpP>
- [70] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," 2020, *arXiv:2012.03816*.



Jie Zhang received the B.S. degree from the China University of Geosciences, Beijing, in 2017. He is currently pursuing the Ph.D. degree with the University of Science and Technology of China (USTC). His primary research interests include IP protection of deep models, media watermarking, and AI security.



Dongdong Chen (Member, IEEE) received the Ph.D. degree from the University of Science and Technology of China (USTC), under the joint Ph.D. program between MSRA and USTC in 2019. He is currently a Principal Researcher at Microsoft Research. His research interests include generative models, AI security, and general representation learning.



Qidong Huang received the B.S. degree from the University of Science and Technology of China in 2020, where he is currently pursuing the Ph.D. degree. His primary research interests include adversarial learning, point cloud processing, and AI security.



Jing Liao (Member, IEEE) received the dual Ph.D. degree from Zhejiang University and the Hong Kong University of Science and Technology. Prior to that, she was a Researcher with the Visual Computing Group, Microsoft Research Asia, from 2015 to 2018. She has been an Assistant Professor with the Department of Computer Science, City University of Hong Kong (CityU), since September 2018. Her primary research interests fall in computer graphics, computer vision, image/video processing, digital art, and computational photography.



Weiming Zhang (Member, IEEE) received the M.S. and Ph.D. degrees from the Zhengzhou Information Science and Technology Institute, China, in 2002 and 2005, respectively. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include information hiding and multimedia security.



Huamin Feng received the B.S. degree from the Northeast Heavy Machinery Institute in 1984. He is currently the Vice President of the Beijing Institute of Electronic Science and Technology, where he is also a Professor. His research interests include information security and cyberspace security.



Gang Hua (Fellow, IEEE) is currently the CTO of Wormpex. Before that, he was the Principal Research Manager at Microsoft Research from 2015 to 2018. He was an Associate Professor with the Stevens Institute of Technology from 2011 to 2015. He is a fellow of IAPR and an ACM Distinguished Scientist. His research focuses on computer vision, pattern recognition, and machine learning, and robotics, with applications in the cloud and mobile intelligence domain.



Nenghai Yu received the B.S. degree from the Nanjing University of Posts and Telecommunications in 1987, the M.E. degree from Tsinghua University in 1992, and the Ph.D. degree from the University of Science and Technology of China in 2004. He is currently a Professor with the University of Science and Technology of China. His research interests include multimedia security, multimedia information retrieval, video processing, and information hiding.