Mask and Restore: Blind Backdoor Defense at Test Time with Masked Autoencoder

Tao Sun, Lu Pang, Chao Chen, Haibin Ling
Stony Brook University
{tao,hling,luppang}@cs.stonybrook.edu, chao.chen.1@stonybrook.edu

Abstract

Deep neural networks are vulnerable to backdoor attacks, where an adversary maliciously manipulates the model behavior through overlaying images with special triggers. Existing backdoor defense methods often require accessing a few validation data and model parameters, which are impractical in many real-world applications, e.g., *when the model is provided as a cloud service. In this paper, we address the practical task of blind backdoor defense at test time, in particular for black-box models. The true label of every test image needs to be recovered on the fly from the hard label predictions of a suspicious model. The heuristic trigger search in image space, however, is not scalable to complex triggers or high image resolution. We circumvent such barrier by leveraging generic image generation models, and propose a framework of* Blind Defense with Masked AutoEncoder *(BDMAE). It uses the image structural similarity and label consistency between the test image and MAE restorations to detect possible triggers. The detection result is refined by considering the topology of triggers. We obtain a purified test image from restorations for making prediction. Our approach is blind to the model architectures, trigger patterns or image benignity. Extensive experiments on multiple datasets with different backdoor attacks validate its effectiveness and generalizability. Code is available at* https://github.com/tsun/BDMAE.

1. Introduction

Deep neural networks have been widely used in various computer vision tasks, like image classification [24], object detection [15] and image segmentation [30], *etc*. Despite the superior performances, their vulnerability to backdoor attacks has raised increasing concerns [16, 31, 42]. During training, an adversary can maliciously inject a small portion of poisoned data. These images contain special triggers that are associated with specific target labels. At inference, the backdoored model behaves normally on clean images but makes incorrect predictions on images with triggers.

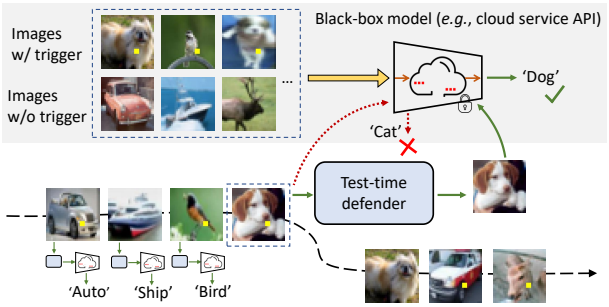


Figure 1: Illustration of blind backdoor defense at test time. The prediction model is black-box and may be backdoored. Test images come in a data stream. The defender sanitizes every image to obtain the correct label prediction on-the-fly.

To defend against backdoor behaviors, existing methods often require accessing a few validation data and model parameters. Some works reverse-engineer triggers [44, 17], and mitigate backdoor by pruning bad neurons or retraining models [28, 44, 50]. The clean labeled data they require, however, are often unavailable. A recent work shows that the backdoor behaviors could be cleansed with unlabeled or even out-of-distribution data [33]. Instead of modifying the model, Februus [10] detects triggers with GradCAM [38], and feeds purified images to the backdoored model.

All these defending methods, although effective, assume the model is known. Such white-box assumption, however, may not fit many real-world scenarios. Due to increasing concerns on data privacy and intellectual property, many models are provided as black-box where detailed parameters are concealed [11, 18, 5], *e.g*., a cloud service API. It is thus crucial to address the problem for black-box models.

In this paper, we tackle the relatively extreme setting and address the task of *Blind Backdoor Defense at Test Time*, in particular for black-box models. *Blind* means that there is no information on whether the model and test images are backdoored or not. Shown in Fig. 1, the prediction model is black-box and may have been injected a backdoor. Test images come in a data stream. The true label of every test

image is unknown; it needs to be recovered on the fly only from the hard label predictions of the suspicious model, without accessing additional data or the model's confidence. This is a very challenging task that cannot be solved by existing test-time defending methods. Simply applying test-time image transformations [14, 36, 35] without model retraining compromises the model's accuracy on clean inputs [37]. Heuristic trigger search in image space [43] does not scale to complex triggers or high image resolution.

To address the challenging task, we resort to the strong reconstruction power of modern image generation models. Intuitively, powerful generation models can reconstruct the original clean image when the trigger is masked. By comparing model predictions on the original and reconstructed images, we can locate the trigger if it exists. We propose a novel method called *Blind Defense with Masked AutoEncoder* (BDMAE). Masked Autoencoders [19] are scalable self-supervised learners. It randomly masks patches from the input image and reconstructs the missing parts. Even using a high masking ratio (*e.g.*, 75%), the semantic content can still be recovered. In our method, we repeatedly mask out specific regions of each test image that possibly contain triggers, and use a generic MAE pretrained on ImageNet [9] to restore the missing parts. The reconstruction power of MAE enables us to use high masking ratios without changing the image semantic content. The dissimilarity in image structure between original images and MAE restorations may also indicate the existence of triggers. Since we use the generic MAE, it does not require additional training images for a particular test-time defense task.

To defense against backdoor attack, we seek a trigger-region score that measures the probability of each image patch belonging to triggers, and use MAE to restore those high-score regions. Our method includes three main stages. First, we randomly mask out the test image, and generate scores based on the image similarity and label consistency between the test image and MAE restorations. Then, we sample masks in consideration of trigger topology, and refine the scores accordingly. Finally, image restorations from adaptive score thresholds are fused into one purified image for making prediction. Our approach is blind to the network architecture, trigger patterns or image benignity. Empirical results demonstrate that BDMAE effectively sanitizes backdoored images without compromising clean images. BDMAE is generalizable to diverse trigger sizes and patterns.

Our main contributions are summarized as follows:

1. We address the practical task of blind backdoor defense at test time, in particular for black-box models. Despite some general techniques for simple attacks, this critical task has not been formally and systematically studied.
2. We propose to leverage generic image generation models to assist backdoor defense. It may open a door to design general backdoor defense methods under limited data by exploiting abundant public foundation models.
3. A framework of blind defense with Masked Autoencoders (BDMAE) is devised to detect possible triggers and restore images on the fly. Three key stages are delicatedly designed to generalize to different defense tasks without tuning hyper-parameters.
4. We evaluate our method on four benchmarks, Cifar10 [23], GTSRB [40], ImageNet [9] and VGGFace2 [3]. Regardless of model architectures, image resolutions or trigger patterns, our method obtains superior accuracies on both backdoored and clean images.

2. Related Works

Backdoor attacks. BadNets [16] is the earliest work on backdoor attack. It attaches a checkerboard trigger to images and associates them with specific target labels. Many different trigger patterns are used in later works [31, 42, 48]. These triggers are visible local patches in the images. Visible global triggers are used in [6, 2]. To make the attack stealthy, invisible patterns [26, 54, 52] and attacking strategies based on reflection phenomenon [29], image quantization and dithering [47], style transfer [8] and elastic image warping [32] are proposed. Although these stealthy attacks are less perceptible to humans, they are vulnerable to noise perturbations or image transformations. To make it hard for defenders to reconstruct triggers, sample-specific backdoor attacks [26, 31] are proposed. This paper focuses on the visible triggers of local patches. The triggers can be either shared by samples or sample-specific.

Backdoor defense. Backdoor defense aims to mitigate backdoor behaviors. The training-stage defenses attempt to design robust training mechanism via decoupling training process [22], introducing multiple gradient descent mechanism [27] or modifying linearity of trained models [46]. However, intruding the training stage is often infeasible. Model reconstruction defenses mitigate backdoor behaviors by pruning bad neurons or retraining models [28, 44, 50] using clean labeled data. A recent work shows that backdoor behaviors could be cleansed by distillation on unlabeled data or even out-of-distribution data [33]. Februus [10] is a test-time defense method. It detects triggers with Grad-CAM [38], and feeds purified images to the model.

Recently, black-box backdoor models have drawn increasing attention [11, 18, 51]. In this setting, the detailed model parameters are concealed due to concerns on data privacy or intellectual property. Some black-box backdoor detection methods [5, 11, 18] have been proposed. These works focus on identifying backdoored models, and usually reject predictions for such situations. Differently, we handle the task of blind backdoor defense at test time. The goal is to obtain true label of every test image on the fly, with only access to the hard-label predictions of that image.

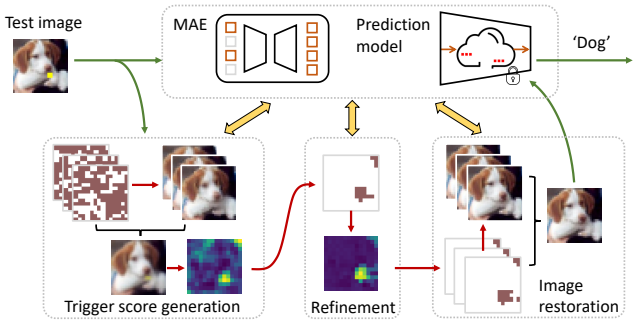


Figure 2: Framework of our method. For every test image, we generate the trigger-region score and refine it by considering the topology of triggers. The purified image obtained from adaptive restorations is used for making prediction.

Test-time image transformations [14, 36, 35] and heuristic trigger search in image space [43] do not work well.

Masked autoencoder. Masked Autoencoders (MAE) [19] are scalable self-supervised learners based on Vision Transformer [12]. It masks random patches of the input image, and restore the missing pixels. MAE has been used in many vision tasks [1, 34, 41]. Motivated by the powerful and robust data generation ability, for the first time we leverage MAE to detect triggers and restore images. Our work can be extended to many other generative models [49, 7, 25].

3. Methodology

We first formulate the backdoor attack and defense problems. Then we detail the proposed method of *Blind Defense with Masked AutoEncoder* (BDMAE), including three key stages. The framework is illustrated in Fig. 2. At a high level, our main idea is to seek a purified version of every test image for making prediction. We efficiently detect possible triggers with the help of MAE and restore missing parts.

3.1. Problem Formulation

Denote the set of clean images as $D = \{(\boldsymbol{x}, y)\}$. An adversary generates a set of backdoored images $\tilde{D} = \{(\Phi(\boldsymbol{x}), \eta(y)) | (\boldsymbol{x}, y) \in D\}$, where $\Phi(\cdot)$ transforms a clean image into a backdoored image and $\eta(\cdot)$ transforms its ground truth label into a target label. We consider the popular formulation of $\Phi(\boldsymbol{x}) = (1-\boldsymbol{m}) \odot \boldsymbol{x} + \boldsymbol{m} \odot \boldsymbol{\theta}$, where $\boldsymbol{m}$ is a binary mask, $\boldsymbol{\theta}$ is the backdoor trigger and $\odot$ denotes the Hadamard product [11, 21, 53]. The masks and triggers may not be necessarily the same for different images. While the trigger can span over the entire image, this work only focuses on triggers formed from local patches. A prediction model f is trained on clean images and backdoored images until it makes correct predictions on clean images but makes abnormal predictions on images with triggers. It is also possible that f is trained on clean images only.

At test time, the suspicious model f is provided as black-box and only its hard label predictions are accessible. The true label of every test image needs to be recovered on the fly, without accessing additional data. For every test image $\boldsymbol{x}$, we seek a purified version $\iota(\boldsymbol{x})$ such that $\iota(\boldsymbol{x})$ does not contain backdoor triggers and $f(\iota(\boldsymbol{x}))$ gives the correct label prediction. The test process is blind to the model or images, meaning that there is no information on whether f is backdoored and whether $\boldsymbol{x}$ contains triggers. The goal is to achieve high classification accuracies on all test images, and thus low attack success rate on triggered images.

Algorithm 1 Trigger-region Score Generation

Input: Prediction model f, test image $\boldsymbol{x}$, generic MAE model G, repeated times N_o, N_i.
Output: Trigger-region scores S^i, S^l

Get original hard-label prediction $\hat{y} = f(\boldsymbol{x})$
for $o = 0$ **to** N_o **do**
  for $i = 0$ **to** N_i **do**
    Uniformly sample random token mask $\boldsymbol{m}_{o,i}$
    Get MAE reconstruction $\{\tilde{\boldsymbol{x}}_{o,i}\}$ and the corresponding masks $\{\tilde{\boldsymbol{m}}_{o,i}\}$ from $\tilde{G}(\boldsymbol{x}, \boldsymbol{m}_{o,i})$
    Get hard-label prediction $\hat{y}_{o,i} = f(\tilde{\boldsymbol{x}}_{o,i})$
  end for
  Fuse restorations into $\tilde{\boldsymbol{x}}_o = \mathcal{F}(\{\tilde{\boldsymbol{x}}_{o,i}\}, \{\tilde{\boldsymbol{m}}_{o,i}\})$
  Calculate structural similarity $I_o = \mathrm{SSIM}(\boldsymbol{x}, \tilde{\boldsymbol{x}}_o)$
end for
$S^i = \sum_o [1 - \mathcal{I}(I_o; 14, 14)]/N_o$
$S^l = \sum_{o,i} [\boldsymbol{m}_{o,i} \times (1 - [\![\hat{y} = \hat{y}_{o,i}]\!])]/(N_o N_i)$

3.2. Trigger-region Score Generation

For clarity, we assume that f is backdoored and the test image $\boldsymbol{x}$ contains triggers. Our method can directly apply to clean models or clean images (*c.r.* Sec.3.5). Let $\hat{y} = f(\boldsymbol{x})$ be its original label prediction. To infer the trigger mask $\boldsymbol{m}$, one can repeatedly block a particular part of the image and observe how model predictions change [43]. However, the search space is huge for a normal image resolution. Even worse, when the trigger is complex (*e.g.*, of irregular shape), the model may still predict the target label when parts of the trigger remain in the image. These issues make the naïve trigger search method infeasible in practice.

We overcome the above-mentioned issues by leveraging generic Masked AutoEncoder (MAE) [19]. In MAE, each token corresponds to a square patch of the image. MAE can recover the image content even when 75% tokens are masked out. This brings two benefits: 1) we can safely use a high masking ratio to remove triggers without changing the semantic label; 2) since triggers are irrelevant to the content, they will unlikely present in the MAE restorations. To locate triggers, there are two complementary approaches:

• **Image-base:** comparing the structural similarity between the original image and MAE restorations.

• **Label-base:** comparing the consistency of label predictions on the original image and MAE restorations.

Now we formulate their procedures. Let H and W be the height and width of the test image $\boldsymbol{x}$. We define a universal function $\mathcal{I}(\boldsymbol{z}; h, w)$ that maps a tensor $\boldsymbol{z}$ to the size of $h \times w$ by interpolation. Our goal is to obtain a trigger-region score $S \in [0,1]^{14\times14}$ that has higher values for trigger regions and lower values for clean regions. Note that each score corresponds to a token, *i.e.*, an image patch. This reduces the search space compared with trigger score of image size.

Before going to the method, we describe how to restore $\boldsymbol{x}$ given a token mask $\boldsymbol{m}$ and a Masked Autoencoder G. Shown in Eq. 1, $\boldsymbol{x}$ is first resized to 224×224. Then we use G to reconstruct the image, and resize it back to $H \times W$. The restoration $\tilde{\boldsymbol{x}}$ is generated at the image size, with a purpose to avoid interpolation errors in the unmasked regions.

$$\begin{aligned} \bar{\boldsymbol{x}} &= \mathcal{I}(G(\mathcal{I}(\boldsymbol{x}; 224, 224); \boldsymbol{m}); H, W) \\ \tilde{\boldsymbol{m}} &= \mathcal{I}(\boldsymbol{m}; H, W) \\ \tilde{\boldsymbol{x}} &= \boldsymbol{x} \odot (1 - \tilde{\boldsymbol{m}}) + \bar{\boldsymbol{x}} \odot \tilde{\boldsymbol{m}} \\ \tilde{G}(\boldsymbol{x}, \boldsymbol{m}) &\triangleq (\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{m}}) \end{aligned} \tag{1}$$

Algorithm 1 describes the procedure to generate trigger-region scores, S^i and S^l. Given the test image $\boldsymbol{x}$, we first get its original hard-label prediction $\hat{y} = f(\boldsymbol{x})$. Then we uniformly sample N_i random token masks $\{\boldsymbol{m}_{o,i} \in \{0,1\}^{14\times14}\}$, and get the MAE reconstructions $\{\tilde{\boldsymbol{x}}_{o,i}\}$, masks $\{\tilde{\boldsymbol{m}}_{o,i}\}$ from $\tilde{G}(\boldsymbol{x}, \boldsymbol{m}_{o,i})$. The hard-label predictions $\{\hat{y}_{o,i} = f(\tilde{\boldsymbol{x}}_{o,i})\}$ are obtained. We use a default masking ratio of 75% when sampling $\{\boldsymbol{m}_{o,i}\}$. Since tokens are uniformly masked, it is possible that partial triggers remain in $\{\tilde{\boldsymbol{x}}_{o,i}\}$. To handle this, we fuse N_i restorations into $\tilde{\boldsymbol{x}}_o$ by:

$$\mathcal{F}\big(\{\tilde{\boldsymbol{x}}_{o,i}\}, \{\tilde{\boldsymbol{m}}_{o,i}\}\big) = \sum_i (\tilde{\boldsymbol{x}}_{o,i} \odot \tilde{\boldsymbol{m}}_{o,i}) \oslash \sum_i (\tilde{\boldsymbol{m}}_{o,i}) \tag{2}$$

where $\oslash$ is element-wise division. The idea is to only use patches from MAE restorations, and discard those from the original image. We manipulate the sampling of $\{\boldsymbol{m}_{o,i}\}$ to guarantee that every image location can be recovered.

To calculate the similarity between $\tilde{\boldsymbol{x}}_o$ and $\boldsymbol{x}$, we use Structural Similarity Index Measure (SSIM) [45]. Its score lies between -1 and 1. As triggers are irrelevant to contents and will unlikely present in $\tilde{\boldsymbol{x}}_o$, SSIM scores in the trigger region will be low. In contrast, the clean regions will be well restored, leading to high SSIM scores. We resize the SSIM score, convert it to negative form, and average over N_o times to get the image-based trigger score S^i. For the label-based trigger score S^l, we simply average over token masks that lead to different label predictions. $[\![P]\!]$ is 1 if P is true and 0 otherwise. The inconsistency usually implies that triggers have been removed by the masks. S^i favors large and complex triggers, while S^l favors small triggers. Combining both adapts to diverse trigger patterns.

Algorithm 2 Topology-aware Score Refinement

Input: Prediction model f, test image $\boldsymbol{x}$, generic MAE model G, repeated times N_r, initial trigger-region score S^*, mask $\boldsymbol{m}_{\text{rf}}$ for tokens to be refined, $\beta_0 = 0.05$.
Output: Refined trigger-region score S^*.

```
Get original hard-label prediction ŷ = f(x)
for r = 0 to N_r do
    Generate a topology-aware token mask m_r
    m̄_r = m_rf − m_r
    Get MAE reconstruction x̃_r from G̃(x, m_r)
    Get hard-label prediction ŷ_r = f(x̃_r)
    β = (1 − 2[[ŷ = ŷ_r]]) × β_0
    S* ← S* + β × (m_r − m̄_r)
end for
```

3.3. Topology-aware Score Refinement

The trigger scores S^i and S^l obtained previously have relatively higher values for trigger regions. However, they are very noisy since token masks are uniformly sampled without considering the trigger patterns. Meanwhile, the contrast between trigger regions and clean regions are not significant, making it hard to determinate a threshold.

We utilize the topology of triggers to refine scores. The procedure is summarized in Alg. 2. Note that backdoor triggers are commonly continuous [21]. We can exploit current trigger scores to generate token masks that cover trigger regions more precisely and reduce the score of clean regions. Denote S^* as either S^i or S^l. Not all scores need to be refined. We only focus on the top L tokens that likely contain triggers, with $L = \text{sum}([\![S^i \geq 0.2]\!])$ or $L = \text{sum}(S^l)$. We define a mask $\boldsymbol{m}_{\text{rf}}$ to indicate the L tokens to be refined.

To generate a topology-aware token mask $\boldsymbol{m}_r$, we sequentially select tokens that have higher trigger scores or are adjacent to already selected tokens. Specifically, we start with token t_0 with the highest score and initialize $\mathcal{T} = \{t_0\}$. Then we repeatedly choose $t_i = \arg\max_{t_k}(S^*[t_k] + 0.5[\![t_k \in \text{Adj}(\mathcal{T})]\!]) \cdot \sigma_k$ and add it to $\mathcal{T}$, where $\text{Adj}(\mathcal{T})$ includes all 4-nearest neighbors of tokens in $\mathcal{T}$ and $\sigma_k \sim U(0,1)$ is a random variable. It achieves a balance between random exploration and topology-aware exploitation. This process continues until $|\mathcal{T}| = L/2$. Given $\boldsymbol{m}_r$ from $\mathcal{T}$, we get its complementary part $\bar{\boldsymbol{m}}_r = \boldsymbol{m}_{\text{rf}} - \boldsymbol{m}_r$. Then we obtain the hard-label prediction $\hat{y}_r$ of MAE restoration based on $\boldsymbol{m}_r$. If $\hat{y}_r \neq \hat{y}$, we increase S^* by a constant β_0 for tokens masked by $\boldsymbol{m}_r$, and decrease S^* by β_0 for tokens masked by $\bar{\boldsymbol{m}}_r$; otherwise, we modify S^* in an opposite way. Since $\|\boldsymbol{m}_r\|_0 = \|\bar{\boldsymbol{m}}_r\|_0 = L/2$, the average value of S^* keeps unchanged, but the contrast in values between trigger region and clean region are increased.

3.4. Adaptive Image Restoration

Given the refined scores S^i and S^l, we simply average them to get the final score $S = (S^i + S^l)/2$. Then we can

Table 1: Defense results on Cifar10 and GSTRB using various sizes of color/white triggers.

			1×1-color			1×1-white			2×2-color			2×2-white			3×3-color			3×3-white		
			ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR
Cifar10	Before Defense		93.66	0.05	99.95	92.86	2.33	97.53	93.23	0.0	100.0	93.12	2.75	97.12	93.71	0.00	100.0	93.39	0.48	99.46
	Februus	XGradCAM	92.03	85.71	7.80	91.09	80.74	13.74	91.32	93.07	0.79	91.21	92.58	1.22	92.15	92.91	0.99	91.62	77.08	17.72
		GradCAM++	85.14	90.82	2.41	84.66	83.62	10.36	85.70	93.06	0.78	84.91	92.03	1.45	85.46	92.98	0.87	75.01	87.09	6.59
	Ours	Base-i	92.55	73.41	20.03	91.38	86.41	6.39	92.07	90.98	1.14	91.77	84.21	8.36	92.41	90.13	2.85	92.05	84.11	8.13
		Base-l	92.12	90.97	0.51	90.61	90.36	1.41	91.57	91.38	0.86	91.15	90.07	1.49	92.13	90.99	1.70	91.55	90.21	0.84
		Base	92.96	90.71	0.56	91.78	90.26	1.50	92.50	91.50	0.47	92.25	89.97	1.50	92.94	92.10	0.53	92.59	90.49	0.78
		Large	93.13	90.78	0.60	91.97	90.59	1.36	92.73	91.75	0.45	92.45	90.36	1.30	93.13	92.30	0.48	92.68	90.87	0.76
GTSRB	Before Defense		98.72	0.00	100.0	97.96	2.33	97.52	98.31	0.00	100.0	98.55	4.39	95.51	98.77	0.00	100.0	98.37	1.38	98.47
	Februus	XGradCAM	85.15	22.87	74.62	84.83	59.05	38.11	74.78	47.39	41.93	56.95	40.32	52.72	74.84	77.04	16.32	69.75	25.97	66.23
		GradCAM++	80.31	40.77	50.78	81.77	90.98	6.23	64.83	46.72	44.45	65.46	79.22	18.42	64.71	89.08	4.33	68.14	63.18	29.19
	Ours	Base-i	96.99	70.46	23.49	96.26	93.34	4.61	96.55	96.51	1.68	96.80	87.63	10.71	97.05	92.75	5.90	96.75	78.27	16.17
		Base-l	97.76	94.00	0.06	96.60	95.04	1.06	97.13	96.84	0.54	97.25	94.84	2.48	97.79	96.51	1.48	97.31	92.24	2.41
		Base	98.55	92.38	0.18	97.62	95.91	0.98	98.03	97.80	0.06	98.27	96.49	1.31	98.62	98.31	0.09	98.14	91.00	2.85
		Large	98.68	94.73	0.18	97.93	96.26	0.88	98.28	98.00	0.07	98.49	96.59	1.50	98.75	98.28	0.19	98.32	91.81	2.87

set a threshold τ to convert the score into a binary mask $\boldsymbol{m} = [\![S \geq \tau]\!]$, and get the corresponding MAE restoration for making prediction. In practice, the optimal thresholds vary to different triggers and test images. It is difficult to select a threshold based on a single test image. To overcome this, we propose an adaptive mechanism.

The idea is to fuse restorations from K multiple thresholds, $\{\tau_1 \geq \tau_2 \geq \cdots \geq \tau_K\}$. These decreasing thresholds lead to a nested structure of resulting masks. We obtain the corresponding MAE restorations $\{\tilde{\boldsymbol{x}}_{\tau_k}, \tilde{\boldsymbol{m}}_{\tau_k} = \tilde{G}(\boldsymbol{x}, [\![S \geq \tau_k]\!])\}$, and then fuse them into one purified image $\iota(\boldsymbol{x}) = \mathcal{F}(\{\tilde{\boldsymbol{x}}_{\tau_k}\}, \{\tilde{\boldsymbol{m}}_{\tau_k}\})$. The default thresholds used in our work is $\{0.6, 0.55, 0.5, 0.45, 0.4\}$. We observe that in datasets of high resolution like ImageNet10, the values of S tend to be higher, thus 0.4 would be too small. To adapt to different datasets automatically, we calculate $\text{avg}([\![S \geq \tau_K]\!])$ and increase all thresholds by a small factor until this quantity is not greater than 25%. The rationale is that trigger regions should not dominate the image. The model prediction $f(\iota(\boldsymbol{x}))$ is used for evaluation.

3.5. Generalization to Clean Images and Models

Until now, we assume that both f and $\boldsymbol{x}$ are backdoored. In practice, we are dealing with blind defense, meaning that both models and images can be either backdoored or clean. Our method can directly apply to any of these situations, thanks to the dedicated designs. In the trigger score generation stage, if $\boldsymbol{x}$ is clean, the MAE restorations are similar to the original image. This implies that the values of S^i will be small. The values of S^l will also be small since the label prediction is unlikely to change. The same results when f is clean. In the score refinement stage, only the top L tokens are affected and L is small in the situation of clean images or models. In the image restoration stage, the predefined thresholds allow the image content to be preserved. The restored parts are either trigger regions or some content-irrelevant regions. Therefore, the model can still make correct label prediction on the purified image $\iota(\boldsymbol{x})$.

4. Experiments

Datasets. We evaluate our method on Cifar10, GTSRB, ImageNet10 and VGGFace2. Cifar10 is a 10-class scene classification dataset of image size 32×32 [23]. GTSRB consists of 43-class traffic signs images of size 32×32 [40]. ImageNet10 is a 10-class subset of ImageNet [9], resized to 224×224. For the face recognition dataset VGGFace2 [3], we use images from 170 randomly selected classes [10], and resize them to 224×224.

Backdoor attacks settings. We use BadNet [16] with different triggers, Label-Consistent backdoor attack (LC) [42] and Input-Aware dynamic Backdoor attack (IAB) [31] to build backdoored models. For each setting, we report results averaged over 14 random target labels or initializations. The default network architecture is ResNet18 [20]. We also conduct experiments with VGG16 [39] and a shallow convolutional network from [10]. For Cifar10 and GTSRB, models are trained from scratch. For the rest two datasets, models are pretrained on ImageNet [9].

Method configurations. Our method is training-free. We use the publicly available Masked Autoencoders [19] pretrained on ImageNet to assist blind defense. The Base variant has 12 encoder layers, and the Large variant has 24 encoder layers with an increased hidden size dimension. The same hyper-parameters are used for all experiments. Besides full method, we present results of two ablative variants: Ours-i using S^i only, and Ours-l using S^l only.

Baseline methods. Methods based on test-time image transformations [14, 36, 35] compromise accuracies unacceptably in our setting. We therefore compare with a representative white-box method, Februus [10]. It detects possible triggers with GradCAM [38], and trains GAN models to restore missing regions. We find that GradCAM does not generalize to complex networks, thus we replace it with two

Table 2: Defense results on `ImageNet` using various types of triggers (†without adaptive thresholds adjustment).

		1×1-color			2×2-color			3×3-color			Trigger ‘ ’			Trigger ‘ ’			Trigger ‘ ’		
		ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR
Before Defense		83.93	62.00	28.84	85.60	7.94	91.37	85.91	2.14	97.65	86.36	0.00	100.0	85.86	0.32	99.68	85.89	0.06	99.94
Ours†	Base	65.07	60.11	14.33	70.80	70.08	5.00	72.79	71.97	2.13	72.77	73.89	1.56	72.67	67.06	1.76	72.97	72.89	1.56
	Large-i	66.07	57.84	19.56	69.74	53.83	33.27	70.96	57.10	27.90	70.83	66.22	12.79	71.33	66.40	11.65	70.87	63.38	16.00
	Large-l	80.23	75.94	8.22	83.13	78.51	5.43	83.54	81.49	2.08	83.57	84.84	0.97	83.46	81.60	0.98	83.71	83.46	1.17
	Large	76.57	70.95	10.35	79.07	75.02	5.00	80.34	77.54	2.02	80.86	77.83	1.32	81.04	72.17	1.78	81.31	76.60	1.38
Ours	Base	74.94	69.70	11.90	78.19	78.86	5.08	79.61	81.48	1.95	79.76	81.49	1.71	79.70	80.43	1.40	79.44	80.68	1.44
	Large-i	77.99	67.06	19.08	80.29	48.63	43.75	81.04	51.57	39.87	80.39	59.22	28.49	81.07	64.16	20.68	80.76	55.90	31.43
	Large-l	80.26	76.17	8.13	83.14	79.33	5.40	83.53	82.35	2.06	83.59	84.86	0.97	83.46	81.92	0.98	83.71	83.51	1.14
	Large	79.10	74.78	9.10	81.07	80.24	5.19	82.13	83.13	1.79	82.80	83.52	1.43	82.56	81.70	1.43	83.04	82.59	1.48

Table 3: Defense results on `VGGFace2` using various types of triggers.

			Trigger ‘ ’			Trigger ‘ ’			Trigger ‘ ’			Trigger ‘ ’			Trigger ‘ ’		
			ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR
VGG16	Before Defense		91.74	0.00	100.0	91.58	0.02	99.98	91.38	0.00	100.0	91.54	0.05	99.95	91.60	0.00	100.0
	Februus	XGradCAM	78.89	44.02	52.11	79.65	87.85	4.45	80.70	89.47	2.08	86.52	90.40	1.47	79.95	71.98	22.57
		GradCAM++	72.60	75.80	18.37	83.91	84.12	8.91	85.10	90.23	1.27	75.17	91.57	0.08	75.47	62.25	32.80
	Ours	Base	83.61	85.30	5.41	83.79	86.86	1.54	83.42	87.84	0.27	84.07	88.22	0.14	83.75	76.43	15.84
		Large-i	84.86	53.14	41.63	84.50	76.21	15.12	84.91	74.20	17.22	84.94	65.96	25.94	84.60	42.60	53.38
		Large-l	81.86	85.82	5.04	82.04	87.15	1.14	81.64	87.74	0.11	82.21	87.68	0.11	81.72	79.57	11.97
		Large	85.49	85.06	5.71	85.47	87.49	1.31	85.19	88.31	0.22	85.85	88.82	0.09	85.57	80.79	10.82
ResNet18	Before Defense		94.12	0.00	100.0	94.03	0.00	100.0	93.98	0.00	100.0	94.04	0.04	99.96	94.17	0.01	99.99
	Februus	XGradCAM	57.01	93.90	0.04	57.20	93.88	0.03	58.24	93.92	0.04	29.36	93.78	0.13	28.69	93.82	0.04
		GradCAM++	53.71	93.96	0.04	54.13	93.87	0.03	55.08	93.94	0.03	27.44	92.86	1.17	26.06	93.87	0.04
	Ours	Base	87.44	86.57	6.47	87.64	90.39	1.28	87.83	91.17	0.19	87.85	91.25	0.07	87.86	80.94	13.27
		Large-i	88.98	56.47	39.29	88.64	73.49	20.90	88.82	85.17	8.12	88.82	70.80	23.08	88.95	40.49	56.84
		Large-l	87.63	88.90	4.22	87.72	90.37	1.02	87.77	91.14	0.07	87.81	91.03	0.09	87.77	80.70	13.81
		Large	90.44	87.99	5.34	90.44	91.10	1.15	90.37	91.65	0.23	90.46	91.83	0.04	90.43	78.95	15.67

improved methods, XGradCAM [13] and GradCAM++ [4]. The choice of visualization score threshold in Februus is critical. We try with $\{0.6, 0.7, 0.8\}$ and report the best result for each attack setting individually. We use the GAN models released by the authors for image restoration.

Evaluation metrics include the classification accuracy on clean images (ACC_c) and backdoored images (ACC_b), as well as attack success rate (ASR).

4.1. Main Results

Cifar10 and GTSRB. From Table 1, models have high ACC_c, low ACC_b and high ASR before defense. The baseline method Februus works on some attack settings such as `Cifar10` with 2 × 2-color trigger, but fails on many others, especially on `GTSRB`. The reason is that images of `Cifar10` and `GTSRB` are of size 32×32. The layer for visualization has a low resolution, making it hard to precisely locate triggers. The performance is also affected by the visualization method. Generally, GradCAM++ uses second-order information, and works better than XGradCAM. The ACC_c of Februus is low because the attended regions on clean images contain contents. Our methods work consistently well on all attack settings. The ACC_b on purified backdoored images are close to the clean accuracy, indicating that triggers have been successfully detected and removed. ACC_c on clean images drops negligibly. Using `MAE-Large` leads to slightly higher accuracies due to better restorations. Comparing the two variants, `Base-`i does not perform well on small or white triggers as it is hard to detect triggers with structural similarity in such cases. `Base-`l is complementary on these attacks. The full method works for all cases. Table 4 shows results with IAB and LC attacks. IAB uses sample-specific irregular curves as triggers, while LC uses four distant checkerboards. Our method still achieves decent accuracies.

ImageNet10. Results are listed in Tab. 2. Since there is no available GAN model for Februus on `ImageNet10`, we do not compare with it. `ImageNet10` has a resolution of 224×224, which brings two challenges. The triggers are relatively small compared to the image size. Besides, when reconstructing images with MAE, some details like small edges may be blurred or lost. The corresponding regions thus have low SSIM scores and high S^i values. This makes it difficult for image-based score to locate triggers. Label-based score, on the contrary, works well in such situations. Nevertheless, the full method can still handle this automatically. As shown in the table, the performances of `Large` are very close to `Large-`l. One key step is the adaptive thresholds adjustment. `ImageNet10` experiments generally have higher S values than those in `Cifar10` and

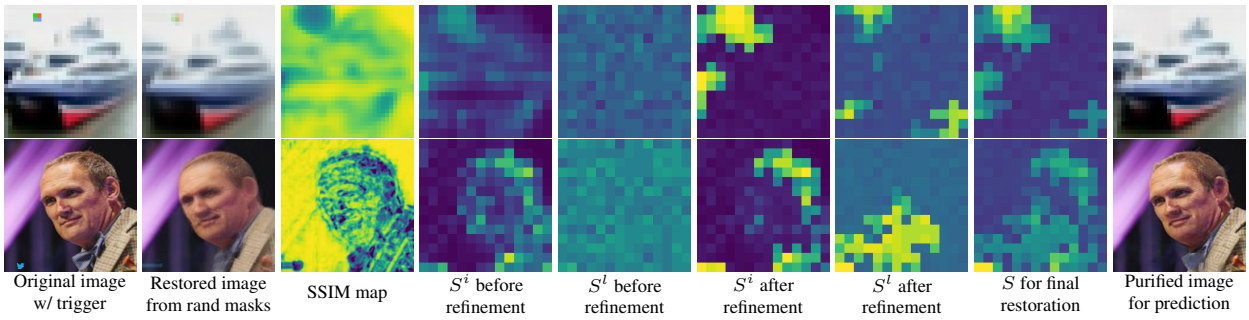


Figure 3: Sampled visualizations of the defense process. (Upper row) `Cifar10` with 2×2-color trigger. (Lower row) `VGGFace2` with *twitter* trigger. All the scores are clipped to a range of [0,1], with yellow for high value.

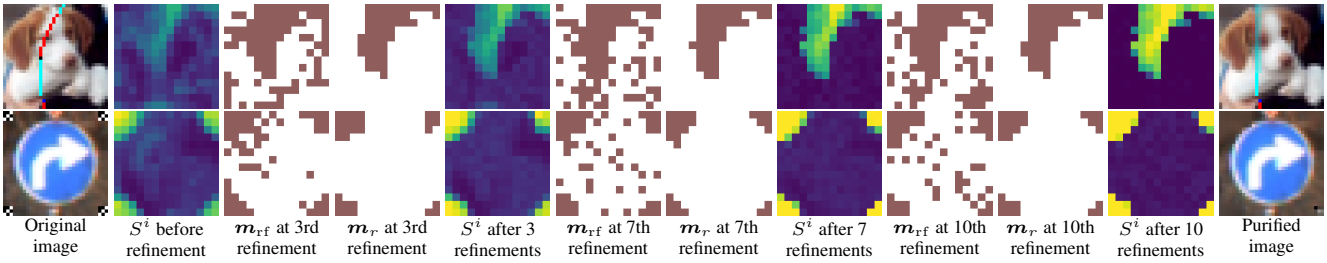


Figure 4: Visualizations of topology-aware score refinement. (Upper) `Cifar10` with IAB. (Lower) `GTSRB` with LC.

`GTSRB`, due to the increased resolution. By increasing the thresholds used for image restoration adaptively, it avoids destruction to the content regions.

VGGFace2. We use VGG16 and ResNet18 as the network architecture. Table 3 lists the results. For VGG16, the baseline method Februus obtains high ACC_b on two triggers and poor scores on the rest. For ResNet18, its ACC_b is pretty good, but ACC_c is severely compromised. This verifies that the visualization is quite sensitive to network architecture and trigger patterns. Our method achieves a good balance between ACC_c and ACC_b. Similar to `ImageNet10`, `Large-`i fails. The full method achieves decent accuracies on both clean and backdoored images.

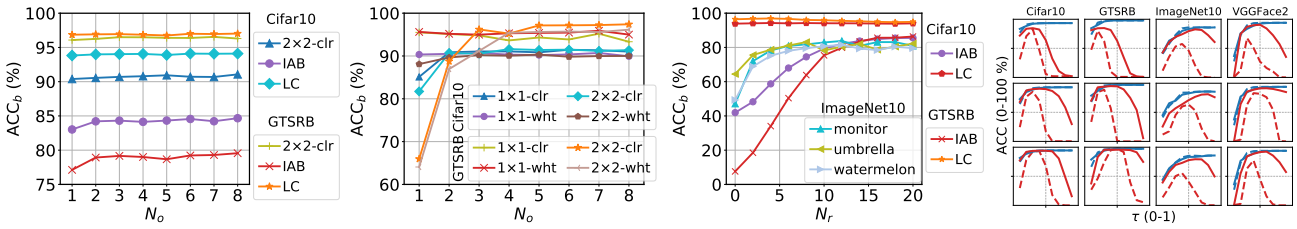
5. Analysis

Visualizations of defense process. We plot images and scores in Fig. 3. Restored images from random masks have the same content as the original images, but are different in the trigger regions and some details. This is reflected in the SSIM map. The two trigger scores are slightly higher in the trigger region, but very noisy. After refinement, high scores concentrate on the triggers, and scores of content regions are suppressed. S is then used to generate the purified images. Compared with the original backdoored images, triggers are removed while the image contents are preserved. The purified images lead to correct label predictions.

Effects of topology-aware refinement. The topology-aware refinement is vital to the generalizability of our method. It exploits initialized scores, and generates topology-aware token masks to refine the scores. This is beneficial especially to complex triggers. In Fig. 4, the triggers are random curves and four distant checkerboards. Before refinement, the trigger regions have relatively high scores in S^i. But the contrast between trigger regions and clean regions are not significant. For each refinement, $\boldsymbol{m}_r$ is sampled in a topology-aware manner to be continuous patches. S^i is updated to have increased values for tokens masked by $\boldsymbol{m}_r$ and reduced values for the rest. After 10 refinements, S^i well reflects the trigger regions. It is worth mentioning that the refinement focuses on the triggers related to backdoor behaviors. Even though the blue line remains in the purified 'dog' image, the red line has been re-

Table 4: Results with IAB and LC attack.

			IAB			LC		
			ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR
`Cifar10`	Before Defense		93.43	1.57	98.37	94.51	0.45	99.55
	Februus	XGradCAM	91.69	29.95	68.15	92.59	63.74	33.92
		GradCAM++	77.85	55.78	35.87	83.29	85.75	10.42
	Ours	Base-i	92.59	85.76	5.21	93.39	94.11	0.36
		Base-l	92.55	27.37	70.89	92.90	71.53	25.55
		Base	92.98	81.79	10.47	93.74	94.09	0.42
		Large	93.10	80.00	12.99	93.93	94.27	0.40
`GTSRB`	Before Defense		98.01	1.25	98.74	95.75	5.26	94.74
	Februus	XGradCAM	68.38	72.48	24.81	86.93	85.49	12.45
		GradCAM++	49.84	84.10	12.19	82.50	83.66	13.99
	Ours	Base-i	96.41	80.29	16.82	93.04	94.86	1.38
		Base-l	97.09	45.93	52.73	91.90	73.83	24.78
		Base	97.84	76.21	21.44	93.93	93.56	2.30
		Large	97.97	70.65	27.44	94.82	93.54	2.40



(a) Base-i. (b) Base-l. (c) Full method. (d) Full method.

Figure 5: (a-c) Effects of repeated times N_o and refinement times N_r. (d) Accuracy curves of using fix thresholds on backdoored and clean images, before (dashed) or after (solid) refinement. Refinement enlarges the ranges of optimal thresholds.

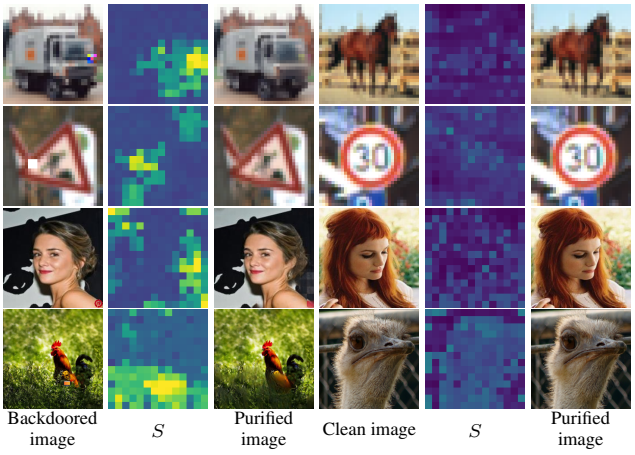


Figure 6: Visualizations on backdoored / clean images.

moved, thus it makes correct label prediction.

In Fig. 5c, we find that $N_r = 10$ is good enough for different triggers. One purpose of refinement is to increase contrast between scores of trigger regions and clean regions, so that the optimal threshold is easier to choose. In Fig. 5d, we randomly select three defense tasks for each dataset. Instead of fusing restorations from multiple thresholds, we choose a fixed threshold ranging from 0.1 to 0.9, and plot the accuracy curves. In each subplot, red/blue lines denote backdoored/clean images, dashed/solid lines denote before/after refinement. We can see that before refinement, the optimal thresholds have narrow ranges and vary across tasks. After refinement, they become wider. It is thus easy to set unified thresholds for different tasks.

Generalization on network architecture. Our method is for black-box defense, thus is generalizable on network architectures. In Tab. 5, we show results on `Cifar10` with three different networks. The trigger is a 3×3 checkerboard. The baseline method Februus performs well on the shallow convolutional neural network originally used by the authors, but is less effective on ResNet18 and VGG16. In contrast, ours achieves high accuracies for all situations.

Performance on clean images and models. We highlight that our method is blind to the benignity of images or models. For backdoored models on clean images, the ACC_c in previous results have validated the effectiveness. Figure 6 shows different properties of S between backdoored and clean images. S of clean images has small values, thus the image restoration step will not change the content. For clean models, Tab. 6 shows that the accuracies on clean images and images with triggers are minimally affected.

Table 5: `Cifar10` with different network architectures.

		ResNet18			6 Conv + 2 Dense			VGG16		
		ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR
Before Defense		92.76	0.06	99.93	91.32	0.0	100.0	89.77	0.00	100.0
Februus	XGradCAM	88.00	88.44	6.10	91.26	91.18	1.52	76.17	77.46	15.08
	GradCAM++	91.05	83.91	10.98	86.17	90.00	2.72	87.28	45.68	50.01
Ours	Base	91.48	90.29	2.99	89.91	90.02	1.33	88.51	87.80	2.47
	Large	91.69	91.14	2.13	90.34	90.23	1.16	88.81	88.27	2.19

Table 6: Defense results on clean models.

		Cifar-10		GTSRB		ImageNet		VGGFace2	
		ACC_c	ACC_b	ACC_c	ACC_b	ACC_c	ACC_b	ACC_c	ACC_b
Before Defense		93.84	93.70	98.67	98.65	86.06	85.59	94.71	94.68
Ours	Base	93.12	93.09	98.50	98.49	79.44	78.94	89.28	89.12
	Large	93.31	93.25	98.64	98.61	83.07	82.29	90.99	91.12

Sensitivity on hyper-parameters. Our method mainly involves two critical hyper-parameters, the repeated times N_o and the refinement times N_r. Throughout the experiments, we use $N_o = 5$ and $N_r = 10$. Figures 5a,5b plot the effects of N_o in Base-i and Base-l, respectively. For the image-based score S^i, the SSIM map is similar for different MAE restorations. Thus averaging over 2 repeated results is good enough. For the label-based score S^l, averaging over many repeated results reduces the variance. $N_o = 5$ generally performs well for both scores.

6. Conclusion

In this paper, we study the novel yet practical task of blind backdoor defense at test time, in particular for black-box models. We use generic image generation models (*i.e.*, MAE) to detect triggers and restore the clean images. Our method focuses on the most popular local-patch triggers, but also generalizes well to medium-sized triggers like color curves (Fig. 4). Extension to global triggers is left as an interesting future work.

A. Remarks on SSIM

Structural Similarity Index Measure (SSIM) [45] is used to measure the similarity between two images. Different from Mean-Squared-Error that measures pixel-wise absolution errors, SSIM considers the inter-dependencies among neighboring pixels. The SSIM index is calculated on two windows, x and y, from a pair of images. Its definition is

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad \text{(A.3)}$$

where μ_x and μ_y are mean values, σ_x and σ_y are variances, and σ_{xy} is covariance. c_1 and c_2 are constants. $\text{SSIM}(x, y)$ lies between -1 and 1. 1 indicates perfect similarity, 0 indicates no similarity, and -1 indicates perfect anti-correlation. In our experiments, we observe that the minimum SSIM values are about -0.6$\sim$-0.2 depending on datasets, and the maximum values are close to 1.0.

The window size influences the SSIM values. Generally, a larger window averages over more pixels, thus the SSIM value is less extreme (*i.e.*, close to 0). We use the commonly used 11×11 Gaussian window, whose effective window size is about 5×5. On `Cifar10` and `GTSRB` of image size 32×32, due to their low resolution, a 11×11 window usually covers content regions. The original image and MAE restorations are similar, thus it is unlikely that the SSIM values will be extremely negatively. On `ImageNet10` and `VGGFace2` of image size 224×224, differently, the window may include some background regions or image details. The difference between the original image and MAE restoration can be significantly large, leading to significantly negative SSIM values. Since our image-based trigger score is defined as $S^i = 1 - \text{SSIM}$, S^i tends to be larger for `ImageNet10` and `VGGFace2`. This is why the adaptive thresholds adjustment is necessary to achieve good performance on the two datasets.

B. Topology-aware Token Mask Generation

In the score refinement, we generate topology-aware token masks. We repeatedly choose $t_i = \arg\max_{t_k}(S^*[t_k] + u[\![t_k \in \text{Adj}(\mathcal{T})]\!]) \cdot \sigma_k$ with $u = 0.5$, where $\text{Adj}(\mathcal{T})$ includes all 4-nearest neighbors of tokens in $\mathcal{T}$ and $\sigma_k \sim U(0, 1)$ is a random variable. Here u is the additional probability assigned to the neighboring tokens. When $u = 0$, the sampling procedure only select tokens with highest trigger-region scores. To see the effect of u, Fig. A.1 plots the results on four defense tasks with increasing u. For the challenging IAB attacks, the performances drops when not using topology-aware sampling (*i.e.*, $u = 0$). $u = 0.5$ obtains relatively good performances on the four tasks. Note that due to the existence of random variable σ_k, using $u = 1.0$ still leads to some randomness in the token selection.

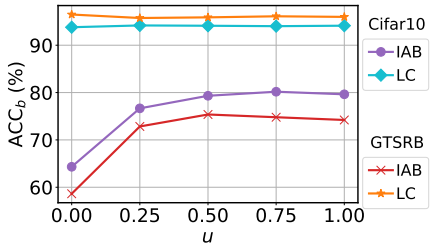


Figure A.1: Effects of the sampling parameter in topology-aware token mask generation.

C. Varying Trigger Size

The trigger size affects the difficulty to detect these triggers. In the BadNet work [16], the authors use 3×3-checkerboard as triggers. In Tab. A.1, we present defense results on `Cifar10` using various sizes of checkerboard triggers. The baseline method Februus [10] achieves relatively high accuracies on backdoored images at the cost of low accuracies on clean images. Our method maintains high accuracies on clean images. Our `Base-i` is not working well on 1×1 trigger because it is hard to detect such a small trigger using image similarity. `Base-l`, on the contrary, works well on this small trigger using label consistency. As trigger size becomes larger, the performance of `Base-l` drops because the trigger can not be removed completely through random masking. The full method `Base` combines the merits of both image similarity and label consistency, and works for all cases. The `Large` variant achieves slightly higher accuracies due to better image restoration.

D. Defense with Test-Time Transformation

To defense against backdoor attack, test-time transformations have been used in some previous works [14, 36, 35]. Since they are training free and can be applied to our task, we briefly summarize these methods and remark on their limitation in our blind backdoor defense setting. **Supression** [36] creates multiple fuzzed copies of backdoored images, and uses majority voting among fuzzed copies to recover label prediction. The fuzzed copies are obtained by adding random uniform noise or Gaussian noise to the original image. However, the intensity of noise is critical. Weak noise would not remove the backdoor behaviour, while strong noise may destroy the semantic content. **DeepSeep** [35] mitigate backdoor attacks using data augmentation. It first fine-tunes the infected model via clean samples with an image transformation policy, and then preprocesses inference samples with another image transformation policy. The image transformation functions include affine transformations, median filters, optical distortion, gamma compression, *etc*. The fine-tuning stage requires additional cleans samples, which are unavailable in

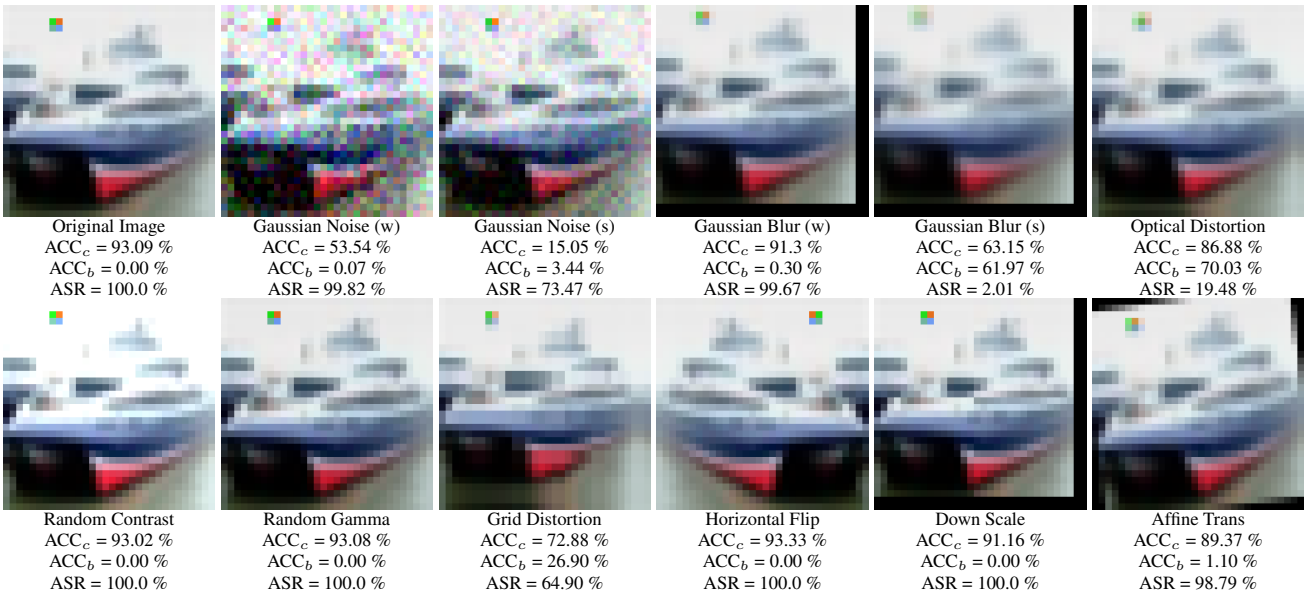


Figure A.2: Defense results of applying test-time image transformations on `Cifar10` with 2×2-color trigger.

Table A.1: Defense results on `Cifar10` using various sizes of checkerboard triggers.

		1×1			3×3			5×5			7×7		
		ACC$_c$	ACC$_b$	ASR	ACC$_c$	ACC$_b$	ASR	ACC$_c$	ACC$_b$	ASR	ACC$_c$	ACC$_b$	ASR
Before Defense		93.27	1.99	97.86	92.76	0.06	99.93	93.42	0.00	100.0	93.33	0.00	100.0
Februus	XGradCAM	91.52	80.73	13.71	91.05	83.91	10.98	91.74	89.92	3.95	91.64	70.23	15.48
	GradCAM++	74.09	85.71	7.13	81.29	91.61	2.67	75.04	92.65	0.92	75.14	72.59	11.74
Ours	Base-i	91.86	78.67	15.00	91.21	88.86	3.97	92.24	91.85	0.40	92.19	88.68	0.57
	Base-l	91.25	90.44	1.35	90.14	75.48	19.73	91.84	80.28	13.15	91.79	35.18	60.11
	Base	92.19	90.62	1.23	91.48	90.29	2.99	92.70	91.99	0.49	92.65	89.04	0.85
	Large	92.41	90.88	1.15	91.69	91.14	2.13	92.81	92.26	0.49	92.75	90.22	0.70

our setting. **STRIP** [14] superposes a test image with multiple other samples, and observes the entropy of predicted labels of these replicas. It aims to detect backdoored inputs, but could not locate the triggers nor recover the true label.

In Fig. A.2, we try different test-time image transformations on `Cifar10` with 2×2-color trigger. For each transformation, we calculate the ACC$_c$ on clean images, ACC$_b$ and ASR on backdoored images. As can be seen, some weak transformations, like Gaussian Noise (w), Gaussian Blur (w), Random Contrast/Gamma, Horizontal Flip and Down Scale, can not reduce ASR. While the rest strong transformations reduces ASR, they also compromise accuracies on clean images unacceptably. To maintain performance on clean images, the model needs to adapt to these image transformations, *e.g.*, through fine-tuning like DeepSeep does. Such requirement is infeasible in the blind backdoor defense, especially for black-box models.

E. Februus Results

Februus [10] uses GradCAM visualization to locate backdoor triggers. It relies on a threshold parameter to determine the backdoor removal regions. In the original paper, the authors use a held-out test set to determine this parameter for each dataset. Since we do not have such held-out test set in our blind backdoor defense task, we try with $\{0.6, 0.7, 0.8\}$, and report results with the parameter leading to best (ACC$_c$+ACC$_b$)/2 in the paper. The best parameter is selected for each attack setting individually. We present the full results in Tab. A.2 and Tab. A.3. Februus is quite sensitive to this parameter. Generally, using a smaller parameter improves ACC$_b$ but reduces ACC$_c$ in Februus. The best parameter varies across different defense tasks. Our method achieves a good balance between accuracies on clean images and backdoored images.

F. Additional Visualization of Defense Process

We present additional visualization of the defense process in Fig. A.3. The top six rows come from IAB [31] attack. IAB uses sample-specific triggers, *i.e.*, test images contain different triggers for one backdoored model. On `Cifar10`, the triggers are irregular curves. On `GTSRB`, the triggers are color patches. Due to the complexity of triggers,

Table A.2: Defense results on `Cifar10` and `GSTRB` using various sizes of color/white triggers.

			1×1-color			1×1-white			2×2-color			2×2-white			3×3-color			3×3-white		
			ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR
Cifar10	Before Defense		93.66	0.05	99.95	92.86	2.33	97.53	93.23	0.0	100.0	93.12	2.75	97.12	93.71	0.00	100.0	93.39	0.48	99.46
	Februus	XGradCAM (0.6)	92.03	85.71	7.80	91.09	80.74	13.74	91.32	93.07	0.79	91.21	92.58	1.22	92.15	92.91	0.99	91.62	77.08	17.72
		XGradCAM (0.7)	92.79	78.45	16.15	91.89	56.07	40.41	92.23	87.34	7.00	92.12	85.02	9.53	92.85	92.14	1.73	92.45	61.18	34.91
		XGradCAM (0.8)	93.25	62.94	32.98	92.36	18.44	80.39	92.78	50.51	46.58	92.66	50.93	46.07	93.25	81.54	13.33	92.94	32.10	65.76
		GradCAM++ (0.6)	74.73	91.21	0.94	74.45	89.60	3.30	76.19	92.91	0.80	74.82	91.26	1.66	75.04	92.73	0.80	75.01	87.09	6.59
		GradCAM++ (0.7)	85.14	90.82	2.41	84.66	83.62	10.36	85.70	93.06	0.78	84.91	92.03	1.45	85.46	92.98	0.87	85.28	75.73	19.04
		GradCAM++ (0.8)	90.81	66.05	29.46	89.97	51.40	45.08	90.71	77.53	17.48	90.31	82.33	12.21	90.78	92.37	1.48	90.54	49.93	46.65
	Ours	Base	92.96	90.71	0.56	91.78	90.26	1.50	92.50	91.50	0.47	92.25	89.97	1.50	92.94	92.10	0.53	92.59	90.49	0.78
		Large	93.13	90.78	0.60	91.97	90.59	1.36	92.73	91.75	0.45	92.45	90.36	1.30	93.13	92.30	0.48	92.68	90.87	0.76
GTSRB	Before Defense		98.72	0.00	100.0	97.96	2.33	97.52	98.31	0.00	100.0	98.55	4.39	95.51	98.77	0.00	100.0	98.37	1.38	98.47
	Februus	XGradCAM (0.6)	73.00	29.06	64.10	74.72	68.15	25.15	58.57	58.57	15.24	56.95	40.32	52.72	59.26	83.21	4.29	69.75	25.97	66.23
		XGradCAM (0.7)	85.15	22.87	74.62	84.83	59.05	38.11	74.78	47.39	41.93	74.41	17.57	81.24	74.84	77.04	16.32	82.39	7.91	90.81
		XGradCAM (0.8)	93.20	14.49	84.83	92.16	45.30	53.25	88.27	30.00	67.75	88.00	2.90	97.04	88.30	51.07	46.45	91.76	2.86	96.84
		GradCAM++ (0.6)	65.18	45.28	33.51	46.10	94.27	0.24	47.00	59.81	17.24	45.03	91.52	1.62	47.99	86.19	0.87	49.80	73.53	6.08
		GradCAM++ (0.7)	80.31	40.77	50.78	64.27	94.77	1.61	64.83	46.72	44.45	65.46	79.22	18.42	64.71	89.08	4.33	68.14	63.18	29.19
		GradCAM++ (0.8)	91.27	28.39	69.28	81.77	90.98	6.23	82.98	17.50	81.38	83.56	22.38	77.29	82.56	70.22	26.68	84.59	30.90	67.43
	Ours	Base	98.55	92.38	0.18	97.62	95.91	0.98	98.03	97.80	0.06	98.27	96.49	1.31	98.62	98.31	0.09	98.14	91.00	2.85
		Large	98.68	94.73	0.18	97.93	96.26	0.88	98.28	98.00	0.07	98.49	96.59	1.50	98.75	98.28	0.19	98.32	91.81	2.87

Table A.3: Defense results on `VGGFace2` using various types of triggers.

			Trigger ‘ ’			Trigger ‘ ’			Trigger ‘ ’			Trigger ‘ ’			Trigger ‘ ’		
			ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR	ACC_c	ACC_b	ASR
VGG16	Before Defense		91.74	0.00	100.0	91.58	0.02	99.98	91.38	0.00	100.0	91.54	0.05	99.95	91.60	0.00	100.0
	Februus	XGradCAM (0.6)	78.89	44.02	52.11	79.65	87.85	4.45	80.70	89.47	2.08	79.98	91.58	0.09	79.95	71.98	22.57
		XGradCAM (0.7)	85.99	8.74	91.10	86.21	52.81	42.64	86.81	47.79	47.86	86.52	90.40	1.47	86.20	47.17	48.57
		XGradCAM (0.8)	89.69	0.04	99.96	89.79	16.90	82.31	90.04	30.57	67.47	89.89	10.83	88.71	89.79	34.75	63.33
		GradCAM++ (0.6)	72.60	75.80	18.37	73.84	91.57	0.07	76.19	91.40	0.07	75.17	91.57	0.08	75.47	62.25	32.80
		GradCAM++ (0.7)	83.27	42.74	53.89	83.91	84.12	8.91	85.10	90.23	1.27	84.77	78.41	15.41	84.52	43.98	52.71
		GradCAM++ (0.8)	88.94	8.60	91.26	89.32	24.31	74.58	89.38	48.14	48.10	89.38	8.26	91.42	89.30	10.62	89.10
	Ours	Base	83.61	85.30	5.41	83.79	86.86	1.54	83.42	87.84	0.27	84.07	88.22	0.14	83.75	76.43	15.84
		Large	85.49	85.06	5.71	85.47	87.49	1.31	85.19	88.31	0.22	85.85	88.82	0.09	85.57	80.79	10.82
ResNet18	Before Defense		94.12	0.00	100.0	94.03	0.00	100.0	93.98	0.00	100.0	94.04	0.04	99.96	94.17	0.01	99.99
	Februus	XGradCAM (0.6)	15.00	90.39	0.09	15.40	91.52	0.05	14.05	93.15	0.04	14.76	92.78	0.07	13.63	92.22	0.05
		XGradCAM (0.7)	29.10	93.15	0.03	29.75	93.54	0.02	29.62	93.60	0.03	29.36	93.78	0.13	28.69	93.82	0.04
		XGradCAM (0.8)	57.01	93.90	0.04	57.20	93.88	0.03	58.24	93.92	0.04	56.92	1.08	98.87	56.68	61.78	35.28
		GradCAM++ (0.6)	14.10	91.01	0.08	14.23	91.81	0.05	12.92	93.23	0.04	13.77	92.92	0.07	13.09	92.60	0.05
		GradCAM++ (0.7)	26.77	93.34	0.04	27.66	93.48	0.02	27.18	93.76	0.03	27.44	92.86	1.17	26.06	93.87	0.04
		GradCAM++ (0.8)	53.71	93.96	0.04	54.13	93.87	0.03	55.08	93.94	0.03	53.76	0.78	99.18	53.62	45.54	52.60
	Ours	Base	87.44	86.57	6.47	87.64	90.39	1.28	87.83	91.17	0.19	87.85	91.25	0.07	87.86	80.94	13.27
		Large	90.44	87.99	5.34	90.44	91.10	1.15	90.37	91.65	0.23	90.46	91.83	0.04	90.43	78.95	15.67

the heuristic search in image space using rectangle trigger blockers [43] may not work well. In our method, the refined trigger-region score S successfully identifies the trigger in each test image. Triggers are removed in the purified images, leading to correct label predictions. On `VGGFace2` and `ImageNet10`, despite their higher image resolution, our method also manages to locate the tiny triggers and restore the clean images.

References

[1] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. MultiMAE: Multi-modal multi-task masked autoencoders. In *European Conference on Computer Vision (ECCV)*, pages 348–367, 2022. 3

[2] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *IEEE International Conference on Image Processing (ICIP)*, pages 101–105, 2019. 2

[3] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pages 67–74, 2018. 2, 5

[4] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847, 2018. 6

[5] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushan-

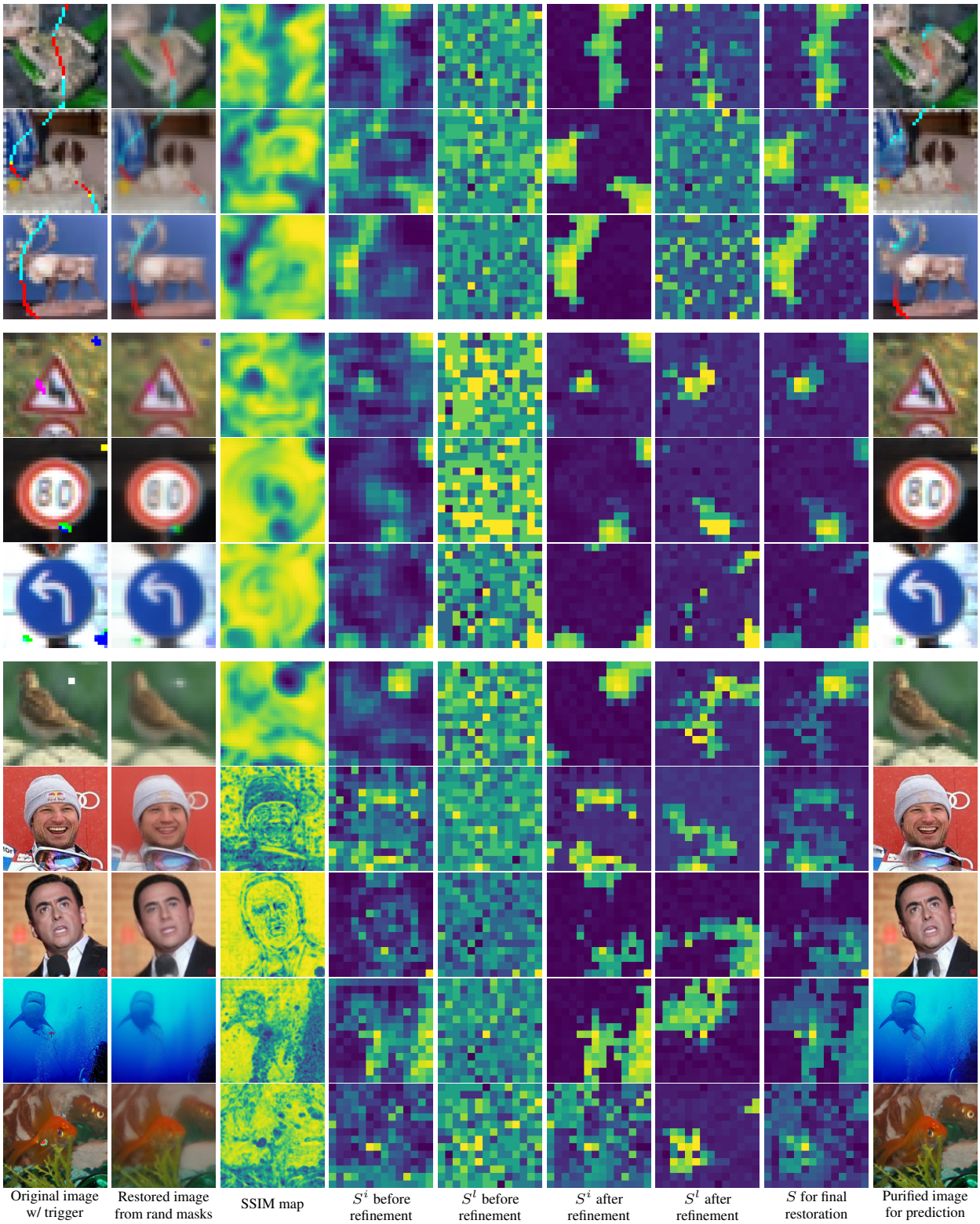


Figure A.3: Sampled visualizations of the defense process. All the scores are clipped to a range of [0,1], with yellow for high value. The top six rows are from IAB attack, and the rest are from BadNet attack.

far. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4658–4664, 2019. 1, 2

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

[7] Yabo Chen, Yuchen Liu, Dongsheng Jiang, Xiaopeng Zhang, Wenrui Dai, Hongkai Xiong, and Qi Tian. SdAE: Self-distillated masked autoencoder. In *European Conference on Computer Vision (ECCV)*, pages 108–124, 2022. 3

[8] Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1148–1156, 2021. 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceddings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 2, 5

[10] Bao Gia Doan, Ehsan Abbasnejad, and Damith C Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. In *Annual Computer Security Applications Conference*, pages 897–912, 2020. 1, 2, 5, 9, 10

[11] Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu. Black-box detection of backdoor attacks with limited information and data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16482–16491, 2021. 1, 2, 3

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 3

[13] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, and Biao Li. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. In *British Machine Vision Conference (BMVC)*, 2020. 6

[14] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 113–125, 2019. 2, 3, 5, 9, 10

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014. 1

[16] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. 1, 2, 5, 9

[17] Jiyang Guan, Zhuozhuo Tu, Ran He, and Dacheng Tao. Few-shot backdoor defense using shapley estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13358–13367, 2022. 1

[18] Junfeng Guo, Ang Li, and Cong Liu. Aeva: Black-box backdoor detection using adversarial extreme value analysis. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022. 2, 3, 5

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5

[21] Xiaoling Hu, Xiao Lin, Michael Cogswell, Yi Yao, Susmit Jha, and Chao Chen. Trigger hunting with a topological prior for trojan detection. In *International Conference on Learning Representations (ICLR)*, 2022. 3, 4

[22] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. In *International Conference on Learning Representations (ICLR)*, 2022. 2

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009. 2, 5

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 25, 2012. 1

[25] Xiang Li, Wenhai Wang, Lingfeng Yang, and Jian Yang. Uniform masking: Enabling mae pre-training for pyramid-based vision transformers with locality. *arXiv preprint arXiv:2205.10063*, 2022. 3

[26] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16463–16472, 2021. 2

[27] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 14900–14912, 2021. 2

[28] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses*, pages 273–294, 2018. 1, 2

[29] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision (ECCV)*, pages 182–199, 2020. 2

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 1

[31] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 3454–3464, 2020. 1, 2, 5, 10

[32] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[33] Lu Pang, Tao Sun, Haibin Ling, and Chao Chen. Backdoor cleansing with unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2

[34] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European Conference on Computer Vision (ECCV)*, pages 604–621, 2022. 3

[35] Han Qiu, Yi Zeng, Shangwei Guo, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham. Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 363–377, 2021. 2, 3, 5, 9

[36] Esha Sarkar, Yousif Alkindi, and Michail Maniatakos. Backdoor suppression in neural networks using input fuzzing and majority voting. *IEEE Design & Test*, 37(2):103–110, 2020. 2, 3, 5, 9

[37] Esha Sarkar, Hadjer Benkraouda, and Michail Maniatakos. Facehack: Triggering backdoored facial recognition systems using facial characteristics. *arXiv preprint arXiv:2006.11623*, 2020. 2

[38] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017. 1, 2, 5

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[40] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012. 2, 5

[41] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[42] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019. 1, 2, 5

[43] Sakshi Udeshi, Shanshan Peng, Gerald Woo, Lionell Loh, Louth Rawshan, and Sudipta Chattopadhyay. Model agnostic defence against backdoor attacks in machine learning. *IEEE Transactions on Reliability*, 71(2):880–895, 2022. 2, 3, 11

[44] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723, 2019. 1, 2

[45] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 4, 9

[46] Zhenting Wang, Hailun Ding, Juan Zhai, and Shiqing Ma. Training with more confidence: Mitigating injected and natural backdoors during training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[47] Zhenting Wang, Juan Zhai, and Shiqing Ma. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15074–15084, 2022. 2

[48] Emily Wenger, Josephine Passananti, Arjun Nitin Bhagoji, Yuanshun Yao, Haitao Zheng, and Ben Y Zhao. Backdoor attacks against deep learning systems in the physical world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6206–6215, 2021. 2

[49] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. SimMIM: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9653–9663, 2022. 3

[50] Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 2

[51] Xinqiao Zhang, Huili Chen, and Farinaz Koushanfar. Tad: Trigger approximation based black-box trojan detection for ai. *arXiv preprint arXiv:2102.01815*, 2021. 2

[52] Zhendong Zhao, Xiaojun Chen, Yuexin Xuan, Ye Dong, Dakui Wang, and Kaitai Liang. Defeat: Deep hidden feature backdoor attacks by imperceptible perturbation and latent representation constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15213–15222, 2022. 2

[53] Songzhu Zheng, Yikai Zhang, Hubert Wagner, Mayank Goswami, and Chao Chen. Topological detection of trojaned neural networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:17258–17272, 2021. 3

[54] Nan Zhong, Zhenxing Qian, and Xinpeng Zhang. Imperceptible backdoor attack: From input space to feature representation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2022. 2