

# DISTIL: Data-Free Inversion of Suspicious Trojan Inputs via Latent Diffusion

Hossein Mirzaei, Zeinab Taghavi, Sepehr Rezaee, Masoud Hadi, Moein Madadi,  
Mackenzie W. Mathis

École Polytechnique Fédérale de Lausanne (EPFL)

hossein.mirzaeisadeghlou@epfl.ch, mackenzie.mathis@epfl.ch

## Abstract

*Deep neural networks have demonstrated remarkable success across numerous tasks, yet they remain vulnerable to Trojan (backdoor) attacks, raising serious concerns about their safety in real-world mission-critical applications. A common countermeasure is trigger inversion – reconstructing malicious “shortcut” patterns (triggers) inserted by an adversary during training. Current trigger-inversion methods typically search the full pixel space under specific assumptions but offer no assurances that the estimated trigger is more than an adversarial perturbation that flips the model output. Here, we propose a data-free, zero-shot trigger-inversion strategy that restricts the search space while avoiding strong assumptions on trigger appearance. Specifically, we incorporate a diffusion-based generator guided by the target classifier; through iterative generation, we produce candidate triggers that align with the internal representations the model relies on for malicious behavior. Empirical evaluations, both quantitative and qualitative, show that our approach reconstructs triggers that effectively distinguish clean versus Trojane models. DISTIL surpasses alternative methods by high margins, achieving up to 7.1% higher accuracy on the BackdoorBench dataset and a 9.4% improvement on trojane object detection model scanning, offering a promising new direction for reliable backdoor defense **without** reliance on extensive data or strong prior assumptions about triggers. The code is available at <https://github.com/AdaptiveMotorControlLab/DISTIL>.*

## 1. Introduction

As artificial intelligence continues to rapidly evolve, detecting Trojan attacks in models has become a critical challenge. Trojan attacks, which insert malicious trigger patterns into training data, allow models to function normally on clean inputs but mispredict inputs containing these triggers [1, 2]. Recently, these attacks have grown more potent by leveraging sophisticated label mapping techniques

and enhancing stealthiness through dynamic or invisible triggers [3–8]. Trojan attacks pose a significant threat to safety-critical computer vision applications, such as autonomous driving and object detection, where undetected triggers could lead to catastrophic failures [9–15].

In response to these attacks, researchers have developed a variety of defense strategies to detect and mitigate Trojane models [16–20]. Among these, methods that reverse engineer triggers (RET) have emerged as a critical post-training defense mechanism [21, 22]. RET methods estimate trigger patterns based on the model behavior, often analyzing output confidence levels. Early RET methods typically optimized for a small patch in the image that acted as a proxy for the trigger. More recent approaches relax prior assumptions about trigger characteristics by integrating feature-space information or employing alternative regularization strategies [23–28]. Notably, all these techniques assume access to clean training data for performing pixel-space optimization. Once reconstructed, the estimated trigger can be used to scan Trojane models, mitigate attacks, and predict target classes [29].

Despite their success, existing RET methods can conflate actual Trojan triggers with adversarial perturbations, leading to false positives in Trojan scanning [30–32]. High-dimensional pixel-space optimization often leads to adversarial noise rather than true triggers, compromising the effectiveness of existing RET methods [33–36]. This limitation results in noisy or less interpretable triggers and increases false positives when scanning benign models. Adapting these methods to other tasks such as object detection is challenging due to spatial variability, multi-output structures, and post-processing complexities [32, 37]. Moreover, reliance on clean data limits real-world applicability, as datasets are often inaccessible [31, 38].

To overcome these challenges, we introduce **DISTIL: Data-free Inversion of Suspicious Trojan Inputs via Latent Diffusion**, a novel method that estimates interpretable and discriminative triggers *without* requiring any clean training samples. Our approach shifts the optimization process from the pixel space to a pre-trained guided diffusion model’s la-

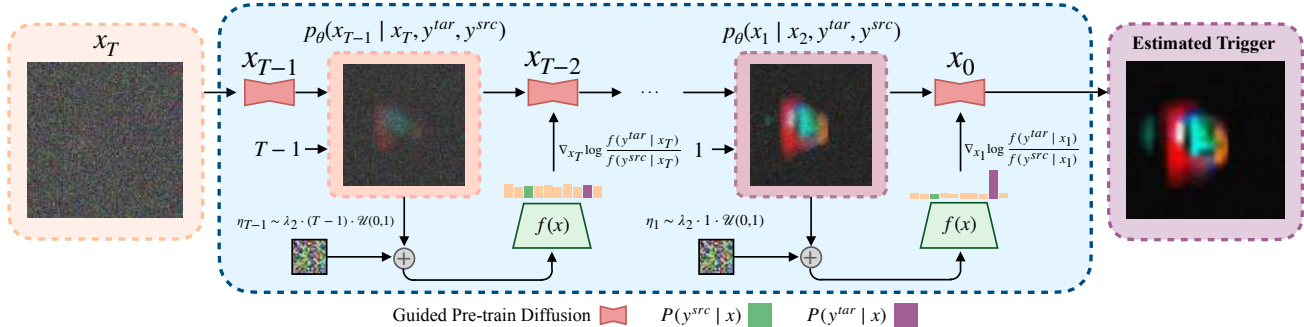


Figure 1. **Method overview.** DISTIL inverts Trojan triggers without any clean data by steering a pre-trained, classifier-guided diffusion model. The process begins with pure Gaussian noise  $x_T \sim \mathcal{N}(0, I)$  and iteratively refines it toward a trigger-like pattern using the classifier’s gradients for the chosen objective. At every diffusion step  $t$ , we inject additional uniform noise  $\eta_t$  before re-evaluating the gradients, which regularizes the search and discourages convergence to adversarial perturbations. Because the diffusion backbone was pre-trained to follow gradient guidance, it can faithfully track these signals in latent space and reveal genuine shortcut patterns. Finally, exploiting the fact that such shortcuts transfer more strongly in Trojanged networks, DISTIL reliably distinguishes compromised models from clean ones.

tent space, thereby reducing the risk of finding purely adversarial artifacts and increasing the likelihood of uncovering legitimate trigger patterns.

Our key insight is that even the most sophisticated Trojanged models are distinctly biased toward specific transferable shortcut patterns. By guiding a diffusion backward process with under test model gradients, we generate synthetic patterns that closely mimic these triggers, thereby enabling a range of defense capabilities. Specifically, we use the recovered patterns to scan Trojanged models, identify target label, and mitigate attacks. Notably, pre-trained guided diffusion models have already been trained to follow gradient-based guidance signals. Their pretraining inherently equips them to easily adapt to new guidance, such as that provided by our test models. This inherent adaptability allows DISTIL to seamlessly extend its capabilities to different scenarios, including the scanning of Trojanged object detection or classifier models.

## 2. Related Works

**Trojan Attacks.** Trojan attacks have grown increasingly sophisticated, utilizing diverse strategies for manipulating label mappings and employing stealthy, dynamic triggers. Label manipulation techniques range from simpler all-to-one mappings to more complex one-to-one, one-to-all, and all-to-all mappings. Early methods like BadNet [1] introduced visible, static triggers, while newer approaches such as SIG [3] and WaNet [5] focus on stealth through imperceptible. Dynamic attacks such as InputAware [4] and Bp-Attack [7] create sample-specific triggers, complicating defense, and highlight the critical need for advanced protective measures.

**RET for Trojan Attack Defense.** Trigger reconstruction serves as a defense method against Trojan attacks by

using the estimated trigger strategy for various tasks. NC [23] serves as a baseline for reverse engineering defenses by generating small static perturbations in pixel space. Subsequent efforts, including FeatureRE [27] and Pixel [39], introduced feature space constraints and improved optimization techniques to enhance trigger fidelity. K-Arm [40] employed multi-arm bandits to explore potential attack classes more efficiently. THTP [24] leveraged topological priors to refine trigger patterns and better localize suspicious regions. Meanwhile, UMD [22] addressed the challenge of varying label mapping attacks by jointly inferring arbitrary source-target mappings without relying on prior knowledge of target labels. UNICORN [28] unified trigger inversion across diverse spaces (pixel, signal, feature, numerical) by employing input space transformations and formulating the inversion as an optimization problem with multiple constraints. BTI-DBF [41] decouples benign and Trojan features during optimization by employing a dual-branch architecture to enhance trigger estimation. In response to the tendency to extract adversarial perturbations rather than triggers, Smooth-Inv [31] aimed to robustify under the test classifier by applying randomized smoothing before pixel space optimization, thereby limiting its applicability to patch-based attacks.

## 3. Method

**Motivation.** The goal of Trojan scanning is to identify signatures that distinguish Trojanged models from their clean counterparts [38, 42]. Our central hypothesis is that shortcut patterns learned by Trojanged models demonstrate significantly greater transferability. This arises because a Trojanged network is explicitly trained to link a specific trigger to a target class, thereby establishing strong spurious correlations that induce misclassifications whenever the trigger appears. Although clean models may also latch onto natu-

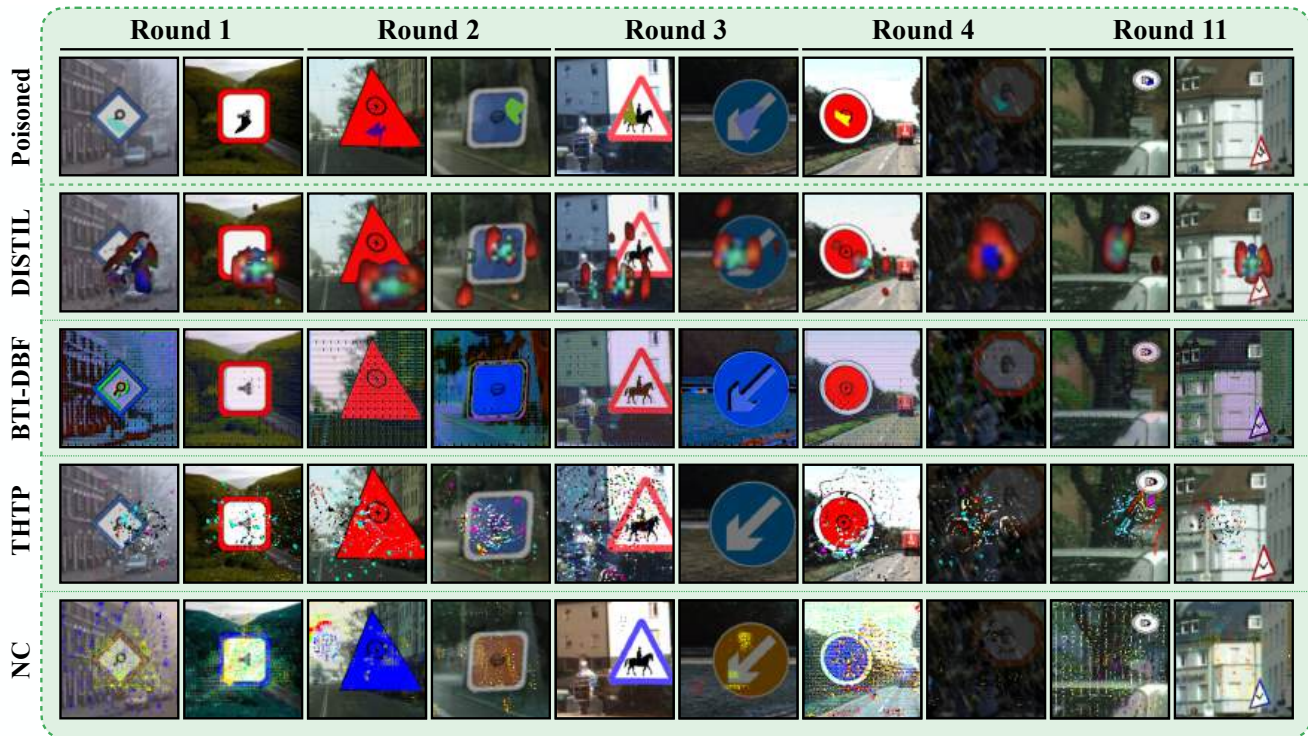


Figure 2. **Visual comparison of trigger-inversion methods.** Estimated triggers produced by prior RET baselines (columns) across multiple TrojAI rounds often resemble adversarial noise rather than true triggers. By constraining the search to the latent space of a guided diffusion model and regularising with uniform-noise augmentation, DISTIL instead uncovers shortcut patterns that closely match the genuine triggers.

ral shortcuts during training, they remain far less sensitive to these specific patterns [43, 44].

Our first aim, therefore, is to extract and estimate these shortcut patterns, then define the Trojan signature by measuring how the model’s predictions change after embedding the estimated shortcut into clean inputs. For a Trojanged model, the shift toward the attack’s target label should be large; for a clean model, it should be much smaller.

Directly optimizing pixel values to recover these shortcuts is problematic, however. The optimization can collapse onto adversarial perturbations rather than genuine triggers, and because both benign and Trojanged models are susceptible to adversarial noise, the resulting patterns cannot serve as a discriminative signature. Worse still, if the Trojan was implanted using adversarial training, pixel-space optimization may yield a pattern that fools the clean model more than the Trojanged one, leading to false positives.

To avoid these pitfalls we move the search into the latent space of a pretrained image-diffusion generator. At each denoising step we steer the generator with the gradient of an objective that increases the probability of the Trojan’s target class while decreasing that of the corresponding source class. Because the generator’s manifold is constrained to realistic images, the search space is far narrower than raw

pixel space and is less prone to degenerate, adversarial artifacts. We further inject small uniform noise into the classifier input at every reverse-diffusion step; this acts as a regulariser, discouraging brittle solutions and nudging the optimisation toward robust, transferable shortcut patterns.

The resulting pattern serves as an interpretable signature that we employ for (i) Trojan scanning, (ii) mitigation via fine-tuning, and (iii) prediction of the attack’s target class. Subsequent sections provide full details of our DISTIL method: Figure 1 gives a high-level overview, while Figure 2 compares the shortcuts estimated by DISTIL with those recovered by prior approaches.

**Threat Model.** We consider an adversary who injects a small set of poisoned samples during training. Each poisoned sample contains a trigger  $T$  stamped onto a source-class image  $x$  and is mislabeled as a target-class  $y^*$ . Consequently, the model learns to associate  $T$  with  $y^*$ . At inference time, the backdoored model behaves normally on clean inputs but classifies any trigger-containing input as  $y^*$ , thus enabling targeted misclassifications while evading detection under standard testing. In general, a backdoor attack against a classifier with  $N$  classes,  $f : \mathcal{X} \rightarrow \mathcal{Y}$ , is defined by a trigger embedding function  $\delta : \mathcal{X} \rightarrow \mathcal{X}$  and a set  $A \subset \mathcal{Y} \times \mathcal{Y}$  of backdoor class pairs.

**Diffusion Guided by Classifier for RET.** To reconstruct Trojan triggers, we employ a pretrained guided diffusion model [45] to follow classifier under test gradients. Our objective is to simultaneously increase the likelihood of a designated target class  $y^{\text{tar}}$  and decrease the likelihood of a source class  $y^{\text{src}}$ . In doing so, we encourage the diffusion model to reveal shortcut patterns learned by the Trojanged classifier. To further ensure that the diffusion process does not converge to mere adversarial perturbations, we incorporate an additional safeguard. At each reverse diffusion step, random *uniform* noise is injected into the classifier input. This uniform noise forces the diffusion model to discover genuine trigger patterns, patterns to which the Trojanged classifier is inherently vulnerable, rather than simply finding adversarial artifacts.

Formally, we modify the mean of the reverse process as follows:

$$\begin{aligned} \tilde{\mu}_\theta(x_t, t, y^{\text{tar}}, y^{\text{src}}) &= \mu_\theta(x_t, t) + \\ &\Sigma_\theta(x_t, t) \nabla_{x_t} \log \frac{f(y^{\text{tar}} | x_t)}{f(y^{\text{src}} | x_t)} + \lambda_1 \cdot \eta_t, \end{aligned} \quad (1)$$

Where  $\tilde{\mu}_\theta(x_t, t, y^{\text{tar}}, y^{\text{src}})$  is the modified mean of the reverse process at time step  $t$  for a given input  $x_t$  with corresponding target and source labels  $y^{\text{tar}}$  and  $y^{\text{src}}$ , respectively,  $\mu_\theta(x_t, t)$  is the original mean of the reverse process,  $\Sigma_\theta(x_t, t)$  is the covariance matrix for the reverse process,  $\nabla_{x_t} \log \frac{f(y^{\text{tar}} | x_t)}{f(y^{\text{src}} | x_t)}$  is the gradient of the classifier logits corresponding to target class minus source class for the input  $x_t$ . The gradient term of the 1, is computed with respect to the input  $x_t$  at each diffusion step, using the classifier’s logits to guide the generation toward patterns that shift predictions from the source class to the target class.  $\eta_t \sim \mathcal{U}(0, 1)$  represents the uniform noise term and  $\lambda_1$  is a hyperparameter controlling its intensity. This uniform noise injection acts as a regularizer, disrupting brittle adversarial perturbations that are sensitive to small changes, thereby encouraging the diffusion process to converge on robust, trigger-like patterns inherent to the Trojanged model.

The next sample is then drawn from the distribution:

$$p_\theta(x_{t-1} | x_t, y^{\text{tar}}, y^{\text{src}}) = \mathcal{N}(\tilde{\mu}_\theta(x_t, t, y^{\text{tar}}, y^{\text{src}}), \Sigma_\theta(x_t, t)), \quad (2)$$

where  $\mathcal{N}(\mu, \Sigma)$  denotes a normal distribution with mean  $\mu$  and covariance  $\Sigma$ .

When clean source-class data denoted as  $\mathcal{X}^{\text{src}}$  are available, DISTIL optionally enhances its reconstruction through hybrid conditioning. We modify the gradient term in Equation 1 as follows:

$$\nabla_{x_t} \log \frac{f(y^{\text{tar}} | \mathcal{X}^{\text{src}} \oplus x_t)}{f(y^{\text{src}} | \mathcal{X}^{\text{src}} \oplus x_t)}, \quad (3)$$

We then use the final generated image  $x_0$  as the trigger corresponding to the pair  $(y^{\text{src}}, y^{\text{tar}})$ , denoted as  $\delta_{\text{src}}^{\text{tar}}$ , if the

probability assigned by the classifier exceeds a threshold  $\lambda_2$  (e.g., 0.95), meaning:

$$\text{softmax}[f(\delta_{\text{src}}^{\text{tar}})]_{y^{\text{tar}}} \geq \lambda_2, \quad (4)$$

where  $\text{softmax}[f(x)]$  is the softmax function applied to the classifier’s output for input  $x$ , which provides the predicted class probabilities,  $[\cdot]_{y^{\text{tar}}}$  extracts the predicted probability for the target class  $y^{\text{tar}}$ , and  $\lambda_2$  is a threshold parameter that determines the minimum probability required to consider the trigger as valid. Otherwise, the generation process is repeated until such a trigger for the considered source and target labels is achieved. We limit repeated generation by a maximum upper bound, and these hyperparameters have been discussed in the experimental details.

**Trojan Detection and Mitigation.** For a classifier under test, we define a *signature*, a scalar score, to quantify the likelihood that the classifier has been compromised by a Trojan attack. This score captures the *transferability* of an extracted shortcut: it measures how strongly an estimated trigger shifts the classifier’s prediction from a source class  $y^{\text{src}}$  to a target class  $y^{\text{tar}}$ . In particular, the trigger is embedded into held-out images from the source class to assess its effect. Specifically, for a given trigger  $\delta_{\text{src}}^{\text{tar}}$ , our signature, defined as the trigger strength (i.e., transferability), is evaluated as follows

$$\begin{aligned} \text{Score}(\delta_{\text{src}}^{\text{tar}}, f) &= \mathbb{E}_{x' \sim \mathcal{X}'^{\text{src}}} \left[ \text{softmax}\left(f(x' + \delta_{\text{src}}^{\text{tar}})\right)_{y^{\text{tar}}} \right. \\ &\quad \left. - \text{softmax}\left(f(x' + \delta_{\text{src}}^{\text{tar}})\right)_{y^{\text{src}}} \right], \end{aligned} \quad (5)$$

The overall Trojan score of the classifier is then defined as the maximum score over all possible target classes and their corresponding source-class triggers:

$$\text{Score}(f) = \max_{y^{\text{tar}}} \max_{y^{\text{src}} \neq y^{\text{tar}}} \text{Score}(\delta_{\text{src}}^{\text{tar}}, f). \quad (6)$$

The trigger achieving the maximum score in Equation 6 target is predicted as the target class of the Trojanged classifier. Notably, this is meaningful when the attack targets a single class; for label mapping strategies such as all-to-all, there is no specific target class.

For mitigation, we create a dataset by injecting the triggers into clean images from their corresponding source classes while preserving the correct labels. The classifier is then finetuned on this dataset, which helps the model focus on authentic image features rather than being misled by shortcut cues introduced by the triggers. We note that using 1% of clean data for Trojan scanning and mitigation is common in the literature [46], and we adopt this setting when running experiments with baselines.

**Fast DISTIL.** Exhaustively scanning every  $(y^{\text{src}}, y^{\text{tar}})$  pair scales quadratically with the number of classes,

Table 1. Comparison of scanning performance between the proposed DISTIL method and alternative approaches. Tables 1-a and 1-b summarize performance on Trojanned **classifier** models, while Table 1-c reports results for Trojanned **object detection** models, all reported in terms of accuracy. The best results in each column are highlighted in bold.

(a) Comparison of scanning performance between DISTIL and alternative methods on the BackdoorBench dataset, covering various attack scenarios.

Dataset	Attack	Method											
		NC	ABS	Pixel	THTP	SmoothInv	Unicorn	BTI-DBF	K-Arm	MM-BD	TRODO	UMD	DISTIL (Ours)
CIFAR-10	BadNets	76.4	73.5	74.0	75.6	86.3	82.9	84.0	75.7	81.3	86.2	76.8	<b>94.9</b>
	Blended	65.2	70.8	67.6	65.2	84.9	78.2	85.7	73.5	74.6	85.0	69.4	<b>93.4</b>
	BPP	62.5	64.0	58.9	60.9	75.5	75.4	76.4	55.4	72.2	83.9	63.9	<b>88.7</b>
	inputaware	58.1	53.9	56.2	49.7	69.7	73.1	79.2	58.1	65.9	71.7	58.2	<b>93.2</b>
	LC	62.8	56.6	51.8	54.2	74.4	67.5	<b>90.5</b>	59.6	80.4	81.2	59.0	89.5
	LF	68.3	61.7	53.4	62.5	81.6	74.0	83.3	63.2	79.3	78.0	65.1	<b>91.0</b>
	LIRA	54.9	58.3	46.1	65.4	70.8	66.9	80.0	54.9	81.8	82.5	57.4	<b>90.6</b>
	SIG	52.0	56.5	54.6	46.8	67.3	65.6	81.9	63.8	79.5	84.8	58.3	<b>92.8</b>
	SSBA	66.1	47.0	61.2	52.0	79.6	73.4	72.6	57.3	78.1	81.2	62.6	<b>90.3</b>
	TrojanNN	52.5	49.2	45.0	58.3	63.0	59.1	83.7	82.7	75.0	76.4	56.2	<b>86.1</b>
WaNet	63.7	57.4	52.5	54.1	68.9	65.3	<b>86.8</b>	56.1	72.7	80.0	61.7	84.4	
GTSRB	BadNets	75.6	67.2	70.2	68.7	86.7	83.4	85.4	76.3	81.8	87.6	75.3	<b>92.1</b>
	Blended	64.8	70.1	63.9	67.1	85.0	78.1	86.8	73.1	75.4	84.3	67.6	<b>91.4</b>
	BPP	62.0	59.8	54.2	56.3	75.9	75.5	85.5	53.7	72.7	<b>89.1</b>	61.9	86.9
	inputaware	51.9	66.7	58.8	47.6	69.8	73.2	79.1	57.0	66.0	64.5	57.4	<b>91.3</b>
	LC	57.3	48.0	61.4	54.9	75.2	67.8	<b>91.3</b>	61.6	74.5	85.2	56.2	89.8
	LF	61.5	53.6	50.5	62.5	81.6	74.3	81.6	57.2	79.4	80.5	63.7	<b>90.5</b>
	LIRA	48.2	59.4	52.1	65.4	70.3	67.0	76.5	52.8	72.9	76.4	53.1	<b>88.2</b>
	SIG	52.0	56.3	54.6	46.8	67.5	65.7	80.0	62.3	79.6	67.5	56.3	<b>90.9</b>
	SSBA	66.4	47.9	61.2	52.0	79.2	73.5	73.9	55.6	78.2	72.3	61.7	<b>85.6</b>
	TrojanNN	52.7	49.2	45.0	58.9	63.8	59.2	82.5	<b>83.9</b>	75.1	85.5	52.1	82.7
WaNet	63.3	57.4	52.3	54.1	69.0	65.4	<b>88.2</b>	54.4	72.8	71.9	60.7	86.0	
Tiny ImageNet	BadNets	76.7	73.8	64.1	67.8	84.9	82.3	79.0	68.6	78.6	75.0	73.1	<b>91.5</b>
	Blended	52.0	65.1	60.4	66.2	83.2	77.7	84.5	65.0	73.1	76.4	64.5	<b>89.1</b>
	BPP	64.5	57.3	46.9	54.5	73.6	68.4	82.3	54.3	72.5	74.2	59.9	<b>83.7</b>
	inputaware	58.2	60.0	52.2	42.3	66.4	66.2	77.7	55.7	64.9	70.0	56.7	<b>90.2</b>
	LC	56.6	42.8	40.8	51.9	74.0	64.5	89.9	56.4	78.3	81.5	53.4	<b>86.4</b>
	LF	45.9	53.6	52.3	60.4	80.5	73.8	79.0	58.2	78.8	83.9	61.6	<b>87.9</b>
	LIRA	57.3	51.5	48.0	63.6	68.8	64.6	73.4	50.6	80.5	79.3	50.2	<b>84.0</b>
	SIG	54.8	58.7	46.5	41.1	65.3	63.2	77.1	59.9	78.2	65.8	55.0	<b>89.3</b>
	SSBA	52.2	46.4	36.1	50.3	77.6	72.9	70.8	52.7	77.4	72.1	59.3	<b>83.9</b>
	TrojanNN	38.7	51.9	55.4	56.9	60.5	58.3	75.2	79.1	73.9	82.7	49.8	<b>81.1</b>
WaNet	46.4	56.0	51.7	53.0	67.1	61.8	80.6	53.4	70.0	76.5	59.5	<b>81.6</b>	

(b) Comparison of scanning performance for Trojanned classifier models across multiple rounds of the TrojAI benchmark under diverse attack scenarios.

Dataset	Method											
	NC	ABS	Pixel	THTP	SmoothInv	Unicorn	BTI-DBF	K-Arm	MM-BD	TRODO	UMD	DISTIL
Round 0	75.1	70.3	76.6	74.7	78.2	72.4	82.5	<b>91.3</b>	80.5	86.2	80.4	83.1
Round 1	72.1	68.5	71.4	65.1	75.0	66.3	79.1	<b>90.0</b>	71.5	85.7	79.2	82.9
Round 2	63.0	61.2	58.8	62.6	67.5	58.4	68.6	76.4	55.8	78.1	75.2	<b>79.5</b>
Round 3	61.4	57.6	52.1	61.7	65.8	56.2	64.2	<b>82.0</b>	52.6	77.2	61.3	78.4
Round 4	58.6	53.7	56.3	55.4	62.1	57.9	56.9	79.3	54.1	82.8	56.9	<b>84.6</b>
Round 11	52.9	51.4	52.5	53.6	59.3	48.6	54.3	61.7	51.3	61.3	48.6	<b>80.4</b>

(c) Comparison of scanning performance between DISTIL and alternative methods on the TrojAI benchmark for Trojanned object detection models.

Dataset	Method											
	NC	ABS	Pixel	THTP	SmoothInv	UNICORN	BTI-DBF	K-Arm	MM-BD	TRODO	UMD	DISTIL
TrojAI-Object Detection	51.1	46.8	37.0	54.3	52.9	52.5	52.8	46.3	48.5	52.0	53.3	<b>63.7</b>

$O(K^2)$ , and quickly becomes impractical. Fast DISTIL lowers this cost to  $O(K)$  without sacrificing accuracy. For each prospective target class  $y^{\text{tar}}$  we identify a single, maximally distant source class

$$y^{\text{src}} = \arg \min_{y \neq y^{\text{tar}}} \cos(\phi(y), \phi(y^{\text{tar}})), \quad (7)$$

where  $\phi(\cdot)$  denotes the mean feature vector in the network’s penultimate layer and  $\cos(\cdot, \cdot)$  is cosine similarity. Selecting the farthest class leverages the intuition that a trigger capable of shifting predictions from the most dissimilar class to  $y^{\text{tar}}$  must be exploiting an especially strong, model-specific shortcut; if such a shortcut exists, it will be revealed here before anywhere else. In practice this heuristic slashes computation by an order of magnitude while maintaining the same detection power, as confirmed in our ablation (Setup D of Table 2).

**Adapting DISTIL to Object Detection.** To adapt DISTIL for scanning object-detection networks, we augment the guidance term in Equation 1 with an additional gradient that encourages a spatial shift in the detector’s predictions. We add  $\nabla_{x_t} \log P(\text{bbox} \rightarrow \text{corner} \mid x_t)$ , where  $P(\text{bbox} \rightarrow \text{corner} \mid x_t)$  is the model’s probability that the centre of each predicted bounding-box falls inside a pre-selected corner region. The combined gradient therefore simultaneously steers the classifier’s output from the source class toward the target class and drags bounding boxes toward the chosen corner, which is sampled uniformly at random for every input data. At evaluation time the estimated trigger is added to the entire image. The detector’s Trojan score is computed as the sum of the classification shift in Equation 5 and the mean displacement of ground-truth bounding boxes measured on held-out data. A large score indicates a strong, transferable shortcut and thus a high probability that the detector is Trojaned (see Figure 3).

## 4. Experiments

We evaluated our method using challenging open-access Trojan scanning datasets, including BackdoorBench [49] and TrojAI [50]. We compared DISTIL against various post-training Trojan defense methods on different tasks, including such as Trojan model detection, target class identification, and Trojan model mitigation.

**Experimental Setup and Evaluation Details.** Table 1a provides a comprehensive comparison between our method and alternative approaches for distinguishing Trojaned classifiers from clean models. Each row corresponds to a specific attack method employed to compromise the Trojaned classifier, ranging from representative to advanced attacks. The Trojaned classifiers utilized in these experiments span multiple architectures, including ResNet, VGG, ViT-B16, and ConvNeXt Tiny. We tested Trojaned models from BackdoorBench alongside 100 different clean models, since

the BackdoorBench dataset originally included only one clean model per architecture.

Table 1b details our experimental results on various rounds of the TrojAI dataset. Notably, these rounds become progressively more challenging, incorporating sophisticated label mapping techniques (such as one-to-one mappings) and diverse training strategies for poisoning Trojaned classifiers, including different adversarial training approaches.

Table 1c illustrates DISTIL’s performance on the TrojAI Round 10 dataset, which involves object detection tasks. This evaluation includes Trojaned and clean object detection models based on FastRCNN and SSD architectures. These results collectively highlight the robustness of our detection framework against a broad spectrum of backdoor attack paradigms. Figure 3 further demonstrates the impact of trigger injection on object detection by showing how the addition of the reconstructed trigger causes significant misclassification and localization errors, underscoring the vulnerability of these models to Trojan attacks.

Table 3 presents our results on Trojan classifiers from the BackdoorBench dataset, specifically focusing on the CIFAR-10 mitigation task. In this experiment, we fine-tuned the Trojan-infected models using the generated triggered data combined with 1% of clean training data (a common mitigation protocol), employing cross-entropy loss. Our approach aims to address and correct the model’s biased learning caused by incorrect shortcut patterns (i.e., triggers). The results demonstrate the superiority of our method in accurately reconstructing trigger patterns closely resembling the original triggers used to Trojan the classifiers. For fairness in evaluation, comparisons were restricted exclusively to RET-based methods.

Table 4 shows the performance of our method in predicting the target classes of Trojan classifiers subjected to various attacks from the BackdoorBench dataset. Specifically, we evaluate classifiers compromised by all-to-one attacks, highlighting the effectiveness and robustness of our approach under different attack scenarios.

**Evaluated Methods and Implementation Details.** Our study focuses on RET; therefore, our comparisons are on representative and recent RET methods, including NC [23], ABS [51], Pixel [39], THTP[52], Smooth-Inv [31], Unicorn [28], BTI-DBF [41], K-Arm [40], and UMD [22]. Additionally, we include MM-BD [38] and TRODO, recent [53], methods specifically designed for Trojan scanning without explicit trigger estimation. In Tables 3 and 4, we compare RET methods that generate informative triggers for both mitigation and target class prediction tasks. To ensure a fair comparison, we evaluated only their estimated triggers, excluding any additional components or strategies that might otherwise skew the results.

As our default backbone, we utilized the pre-trained lightweight guided diffusion model from OpenAI [45],

Table 2. Ablation study showing model accuracy (%) when each component is individually excluded or replaced, with all other components held constant.

Setup	Components							Dataset					
	Data Supervision for RET	Noise Injection	Fast Class Pairing	Training&Hyper Selection	Dif. Model-1 [45]	Dif. Model-2 [47]	Dif. Model-3 [48]	Round0	Round1	Round2	Round3	Round4	Round11
<b>A</b>	✓	✓	-	✓	-	-	-	74.5	73.0	65.2	64.9	60.5	57.4
<b>B</b>	-	-	-	✓	✓	-	-	81.9	80.5	76.3	74.4	81.6	76.9
<b>C</b>	-	✓	-	-	✓	-	-	80.6	76.4	72.8	74.1	78.0	73.3
<b>D</b>	-	✓	✓	✓	✓	-	-	78.0	79.1	73.9	75.9	82.3	75.6
<b>E</b>	-	✓	-	✓	-	✓	-	81.9	80.3	78.4	78.2	81.0	78.4
<b>F</b>	-	✓	-	✓	-	-	✓	78.6	74.2	73.8	75.1	80.3	77.0
<b>G<sub>(Ours)</sub></b>	-	✓	-	✓	✓	-	-	83.1	82.9	79.5	78.4	84.6	80.4
<b>H<sub>(Ours+Data)</sub></b>	✓	✓	-	✓	✓	-	-	84.5	83.6	82.4	81.8	86.0	83.9

Table 3. Mitigation results on CIFAR-10 Trojaned classifiers from BackdoorBench. We report post-fine-tuning classification accuracy (ACC ↑) and attack success rate (ASR ↓) across various Trojan attack scenarios, compared to the *original* (unmitigated) models.

Dataset	Attack	Method															
		Original		NC		Pixel		THTP		SmoothInv		Unicorn		BTI-DBF		DISTIL (Ours)	
		ACC.↑	ASR↓	ACC.↑	ASR↓	ACC.↑	ASR↓	ACC.↑	ASR↓	ACC.↑	ASR↓	ACC.↑	ASR↓	ACC.↑	ASR↓	ACC.↑	ASR↓
CIFAR-10	<b>BadNets</b>	91.7	94.4	86.3	9.5	88.0	15.2	87.2	10.9	86.2	<b>7.4</b>	89.0	12.2	91.1	8.7	90.3	8.6
	<b>Blended</b>	93.6	99.7	85.9	8.9	89.6	12.4	87.8	7.8	92.5	6.8	90.6	10.4	90.8	6.5	89.1	5.3
	<b>BPP</b>	93.8	99.8	87.9	97.6	91.3	82.0	89.6	89.8	88.6	90.8	92.3	81.0	89.5	12.4	88.4	<b>9.0</b>
	<b>Inputaware</b>	89.6	94.6	85.0	38.4	86.7	22.1	85.9	30.3	84.0	31.3	87.7	25.6	86.9	10.8	87.2	<b>7.4</b>
	<b>LC</b>	84.5	99.9	79.3	18.7	78.4	16.2	78.9	17.5	80.9	18.5	79.4	15.6	82.0	12.3	86.5	<b>10.7</b>
	<b>LF</b>	89.4	30.2	83.4	9.1	86.2	13.9	84.8	6.5	83.8	7.5	87.2	12.9	85.7	8.1	91.6	<b>5.6</b>
	<b>SIG</b>	84.5	97.1	78.2	32.8	80.7	14.3	79.5	19.6	78.5	20.6	81.7	20.3	80.2	14.8	82.5	<b>12.8</b>
	<b>SSBA</b>	92.8	97.1	89.1	14.2	91.4	9.0	90.3	15.6	89.3	8.1	91.4	15.0	88.4	12.6	86.1	<b>7.9</b>
	<b>TrojanNN</b>	93.4	100.0	90.7	11.6	88.5	10.6	89.6	13.4	90.6	12.1	89.5	15.6	87.6	9.4	91.9	<b>8.2</b>
	<b>WaNet</b>	87.8	85.7	91.5	14.3	86.0	15.7	88.8	19.6	87.8	16.4	85.0	14.7	85.3	<b>9.2</b>	86.1	10.5

Table 4. Target-class prediction accuracy of RET methods on BackdoorBench models Trojaned with various backdoor attacks on the GTSRB dataset. Each row corresponds to a different attack type, and each column shows the accuracy achieved by existing RET baselines versus our proposed approach.

Dataset	Attack	Method						
		NC	Pixel	THTP	SmoothInv	Unicorn	BTI-DBF	DISTIL (Ours)
GTSRB	<b>BadNets</b>	0.65	0.65	0.60	<b>0.80</b>	0.75	<b>0.80</b>	<b>0.80</b>
	<b>Blended</b>	0.65	0.55	0.45	0.55	0.65	0.70	<b>0.85</b>
	<b>BPP</b>	0.45	0.50	0.55	0.45	0.40	0.50	<b>0.65</b>
	<b>Inputaware</b>	0.30	0.25	0.20	0.35	0.45	0.65	<b>0.70</b>
	<b>LC</b>	0.45	0.40	0.45	0.60	0.65	0.60	<b>0.70</b>
	<b>LF</b>	0.40	0.45	0.55	0.55	0.30	<b>0.65</b>	<b>0.65</b>
	<b>LIRA</b>	0.55	0.50	0.45	0.40	0.55	0.60	<b>0.75</b>
	<b>SIG</b>	0.20	0.25	0.30	0.20	0.35	0.65	<b>0.70</b>
	<b>SSBA</b>	0.35	0.30	0.25	0.45	0.55	0.60	<b>0.65</b>
	<b>TrojanNN</b>	0.40	0.45	0.50	0.55	0.50	0.75	<b>0.80</b>
<b>WaNet</b>	0.25	0.35	0.20	0.30	0.45	0.65	<b>0.75</b>	

which was trained on approximately 64 million images. This model employs a fast sampling strategy that significantly improves the efficiency of the pipeline. When using the diffusion model, we set the number of sampling

steps ( $T$ ) to 50 to improve time efficiency while keeping the other hyperparameters at their default settings. We note that even when using lighter pre-trained diffusion models, such as those trained on 1 million ImageNet images, DISTIL achieves consistent performance with only a minor drop (see ablation study setups E and F in Section 5 for details).

DISTIL generates triggers by aiming for high classifier confidence toward the target class. If this criterion is not initially met, we repeat the trigger generation process up to a maximum of 5 iterations, a value chosen based on empirical observations. For selecting the remaining hyperparameters, we used the models from the TrojAI training rounds.

In Appendix we present the pseudo-code, the standard deviations of our results, additional visualizations of synthesized triggers, a background review, and DISTIL’s performance under All-to-All attacks. Because BackdoorBench and TrojAI focus mainly on all-to-one and one-to-one, we also evaluate DISTIL in an all-to-all label mapping scenario; the results are presented in Table 6.

**Results Analysis.** DISTIL achieves an average performance of 88.5% on BackdoorBench, 81.4% on TrojAI, and

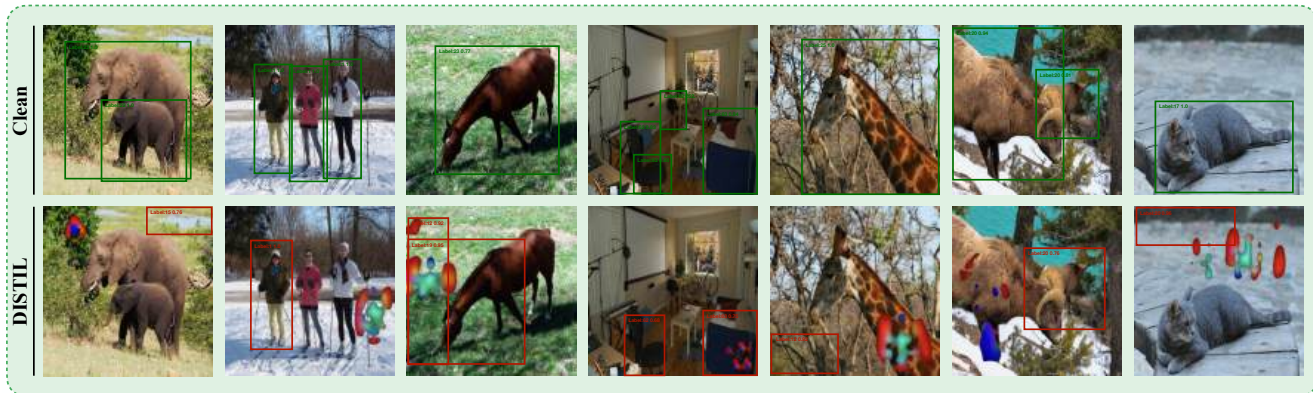


Figure 3. **DISTIL on object-detection models.** Each pair shows (top row) a clean input with correct detections (green boxes) and the same input (bottom row) after injecting the trigger recovered by DISTIL (red boxes). The reconstructed trigger not only flips the model’s prediction from the true class to the attacker’s target class but also drags the bounding-box center toward a pre-selected corner, producing simultaneous misclassification and mislocalization. This visualization demonstrates DISTIL’s ability to generalize from image classification to object detection task.

63.7% on Trojaned object detection scanning task. Notably, DISTIL surpasses alternative methods by high margins, achieving up to **7.1%** higher accuracy on the BackdoorBench dataset and a **9.4%** improvement on Trojaned object detection model scanning. These results underscore effectiveness as a robust scanning approach that is applicable in diverse tasks. Furthermore, DISTIL significantly lowers the Attack Success Rate (ASR) to just 8.6%, while concurrently improving the target class prediction accuracy to 72.0% on the GTSRB dataset. While DISTIL performance gap is smaller on TrojAI rounds, which mainly involve Trojaned classifiers with patch-shaped triggers similar to Badnet [1] attacks, DISTIL consistently achieves high performance on the broader BackdoorBench dataset. This highlights its ability to detect diverse Trojan attacks.

## 5. Ablation Study

We performed an ablation study across multiple TrojAI rounds (Table 2) to isolate the contributions of each component in DISTIL. Our default configuration, Setup G, employs no data supervision for RET, injects uniform noise into the classifier input, selects hyperparameters based on training on a subset of the TrojAI training data, and uses the GLIDE [45] as the generative backbone. In Setup H, we augment the default setting by introducing clean training data for RET, enabling the hybrid conditioning described in Equation 3. In Setup A, we remove the diffusion model altogether and optimize the objective directly in pixel space, thereby demonstrating the significance of diffusion modeling in synthesizing discriminative triggers. In Setup B, we discard the noise-injection strategy while keeping every other component unchanged, illustrating how noise injection helps avert adversarial artifacts. In Setup C, we

examine the role of hyperparameter tuning by abandoning training-based hyperparameter selection and instead using fixed values of  $\lambda_1 = 0.3$  (noise injection strength) and  $\lambda_2 = 0.95$  (classification confidence threshold); these values were chosen to avoid excessive distortion of important input features and to ensure that the classifier has high confidence in attributing the crafted trigger to the target class, a setting that underscores DISTIL’s robustness to suboptimal hyperparameters. In Setup D, we adopt our fast RET strategy for trigger generation, which reduces the computational complexity from  $\mathcal{O}(K^2)$  to  $\mathcal{O}(K)$ , illustrating DISTIL’s capacity to balance efficiency with strong detection performance. In Setup E, we replace the default diffusion model with an ImageNet-pretrained one proposed by [47], and in Setup F, we use a score-based diffusion model pretrained on LSUN [48]. Both variations reveal only modest changes in performance, confirming that while the backbone diffusion model and its pretraining data matter, DISTIL retains a high level of effectiveness.

## 6. Conclusions

We introduced DISTIL, a novel diffusion-based method for accurately reconstructing interpretable Trojan triggers without needing training data. By integrating classifier-guided diffusion with injected noise, DISTIL effectively distinguishes genuine triggers from adversarial noise, significantly reducing false positives in detection. Extensive evaluations across multiple architectures and benchmarks confirmed DISTIL’s robust performance in detecting, predicting, and mitigating Trojan attacks, highlighting its practical utility for enhancing model security in critical vision tasks.

## References

- [1] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. doi: 10.1109/ACCESS.2019.2909068. 1, 2, 8, 14
- [2] 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. 1, 14
- [3] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 101–105. IEEE, 2019. 1, 2, 14
- [4] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020. 2, 14
- [5] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=5aYyYrXzSx>. 2, 14
- [6] Wenbo Jiang, Hongwei Li, Guowen Xu, and Tianwei Zhang. Color backdoor: A robust poisoning attack in color space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8133–8142, 2023. 14
- [7] 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*, pages 15074–15084, 2022. 2, 14
- [8] 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*, pages 16463–16472, 2021. 1, 14
- [9] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, and Hyoungshick Kim. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv preprint arXiv:2007.10760*, 2020. 1
- [10] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11957–11965, 2020.
- [11] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey, 2022. URL <https://arxiv.org/abs/2007.08745>.
- [12] Kingshuo Han, Guowen Xu, Yuan Zhou, Xuehuan Yang, Jiwei Li, and Tianwei Zhang. Physical backdoor attacks to lane detection systems in autonomous driving. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 2957–2968, 2022.
- [13] Yan Zhang, Yi Zhu, Zihao Liu, Chenglin Miao, Foad Hajighajani, Lu Su, and Chunming Qiao. Towards backdoor attacks against lidar object detection in autonomous driving. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 533–547, 2022.
- [14] Yudong Li, Shigeng Zhang, Weiping Wang, and Hong Song. Backdoor attacks to deep learning models and countermeasures: A survey. *IEEE Open Journal of the Computer Society*, 4:134–146, 2023.
- [15] Shuai Zhao, Meihuizi Jia, Zhongliang Guo, Leilei Gan, XIAOYU XU, Xiaobao Wu, Jie Fu, Feng Yichao, Fengjun Pan, and Anh Tuan Luu. A survey of recent backdoor attacks and defenses in large language models. *Transactions on Machine Learning Research*, 2024. 1
- [16] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34:14900–14912, 2021. 1
- [17] Yang Liu, Mingyuan Fan, Cen Chen, Ximeng Liu, Zhuo Ma, Li Wang, and Jianfeng Ma. Backdoor defense with machine unlearning. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, pages 280–289. IEEE, 2022.
- [18] Weixin Chen, Baoyuan Wu, and Haoqian Wang. Effective backdoor defense by exploiting sensitivity of poisoned samples. *Advances in Neural Information Processing Systems*, 35:9727–9737, 2022.
- [19] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. *arXiv preprint arXiv:2202.03423*, 2022.
- [20] Qi Zhao and Christian Wressnegger. Adversarially robust anti-backdoor learning. In *Proceedings of the 2024 Workshop on Artificial Intelligence and Security*, pages 77–88, 2024. 1
- [21] 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. doi: 10.1109/SP.2019.00031. 1
- [22] Zhen Xiang, Zidi Xiong, and Bo Li. UMD: Unsupervised model detection for X2X backdoor attacks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 38013–38038. PMLR, 23–29 Jul 2023. URL <https://proceedings.mlr.press/v202/xiang23a.html>. 1, 2, 6, 18
- [23] 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. IEEE, 2019. 1, 2, 6, 15
- [24] Xiaoling Hu, Xiao Lin, Michael Cogswell, Yi Yao, Susmit Jha, and Chao Chen. Trigger hunting with a topological prior for trojan detection. In *ICLR 2022*, 2022. URL <https://openreview.net/forum/?id=TXsjU8BaibT>. 2, 17
- [25] Guangyu Shen, Yingqi Liu, Guanhong Tao, Shengwei An, Qiuling Xu, Siyuan Cheng, Shiqing Ma, and Xiangyu Zhang.

- Backdoor scanning for deep neural networks through k-arm optimization. In *International Conference on Machine Learning*, pages 9525–9536. PMLR, 2021.
- [26] Guanhong Tao, Guangyu Shen, Yingqi Liu, Shengwei An, Qiuling Xu, Shiqing Ma, Pan Li, and Xiangyu Zhang. Better trigger inversion optimization in backdoor scanning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13368–13378, 2022.
- [27] Zhenting Wang, Kai Mei, Hailun Ding, Juan Zhai, and Shiqing Ma. Rethinking the reverse-engineering of trojan triggers. *Advances in Neural Information Processing Systems*, 35:9738–9753, 2022. 2, 16
- [28] Zhenting Wang, Kai Mei, Juan Zhai, and Shiqing Ma. Unicorn: A unified backdoor trigger inversion framework. *arXiv preprint arXiv:2304.02786*, 2023. 1, 2, 6, 17
- [29] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, and Kui Ren. Towards reliable and efficient backdoor trigger inversion via decoupling benign features. In *The Twelfth International Conference on Learning Representations*, 2023. 1
- [30] Shibani Santurkar, Andrew Ilyas, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Image synthesis with a single (robust) classifier. *Advances in Neural Information Processing Systems*, 32, 2019. 1
- [31] Mingjie Sun and Zico Kolter. Single image backdoor inversion via robust smoothed classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8113–8122, 2023. 1, 2, 6
- [32] Mantas Mazeika, Dan Hendrycks, Huichen Li, Xiaojun Xu, Sidney Hough, Andy Zou, Arezoo Rajabi, Qi Yao, Zihao Wang, Jian Tian, et al. The trojan detection challenge. In *NeurIPS 2022 Competition Track*, pages 279–291. PMLR, 2023. 1
- [33] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1
- [34] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, October 2019. ISSN 1941-0026. doi: 10.1109/tevc.2019.2890858. URL <http://dx.doi.org/10.1109/TEVC.2019.2890858>.
- [35] Cheng-Hsin Weng, Yan-Ting Lee, and Shan-Hung Brandon Wu. On the trade-off between adversarial and backdoor robustness. *Advances in Neural Information Processing Systems*, 33:11973–11983, 2020.
- [36] Zhenxing Niu, Yuyao Sun, Qiguang Miao, Rong Jin, and Gang Hua. Towards unified robustness against both backdoor and adversarial attacks. *IEEE transactions on pattern analysis and machine intelligence*, 2024. 1
- [37] Guangyu Shen, Siyuan Cheng, Guanhong Tao, Kaiyuan Zhang, Yingqi Liu, Shengwei An, Shiqing Ma, and Xiangyu Zhang. Django: Detecting trojans in object detection models via gaussian focus calibration. *Advances in Neural Information Processing Systems*, 36:51253–51272, 2023. 1
- [38] H. Wang, Z. Xiang, D. J. Miller, and G. Kesidis. Mm-bd: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 19–19, Los Alamitos, CA, USA, may 2024. IEEE Computer Society. doi: 10.1109/SP54263.2024.00015. URL <https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00015>. 1, 2, 6, 17
- [39] Guanhong Tao, Guangyu Shen, Yingqi Liu, Shengwei An, Qiuling Xu, Shiqing Ma, Pan Li, and Xiangyu Zhang. Better trigger inversion optimization in backdoor scanning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13358–13368, 2022. doi: 10.1109/CVPR52688.2022.01301. 2, 6, 16
- [40] Guangyu Shen, Yingqi Liu, Guanhong Tao, Shengwei An, Qiuling Xu, Siyuan Cheng, Shiqing Ma, and Xiangyu Zhang. Backdoor scanning for deep neural networks through k-arm optimization. *arXiv preprint arXiv:2102.05123*, 2021. 2, 6, 17
- [41] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, and Kui Ren. Towards reliable and efficient backdoor trigger inversion via decoupling benign features. In *International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=2fDSEWgVR0>. 2, 6, 16
- [42] Yanghao Su, Jie Zhang, Ting Xu, Tianwei Zhang, Weiming Zhang, and Nenghai Yu. Model x-ray: Detecting backdoored models via decision boundary. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 10296–10305, 2024. 2
- [43] Xuanli He, Qionгкаi Xu, Jun Wang, Benjamin Rubinstein, and Trevor Cohn. Mitigating backdoor poisoning attacks through the lens of spurious correlation, 2023. URL <https://arxiv.org/abs/2305.11596>. 3
- [44] Yige Li, Jiabo He, Hanxun Huang, Jun Sun, Xingjun Ma, and Yu-Gang Jiang. Shortcuts everywhere and nowhere: Exploring multi-trigger backdoor attacks, 2024. URL <https://arxiv.org/abs/2401.15295>. 3
- [45] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 4, 6, 7, 8
- [46] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 4
- [47] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 7, 8
- [48] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 7, 8
- [49] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoor-bench: A comprehensive benchmark of backdoor learning. *Advances in Neural Information Processing Systems*, 35:10546–10559, 2022. 6, 18
- [50] UCF-ML-Research. The trojai software framework: An opensource tool for embedding trojans into deep

- learning models, 2020. <https://github.com/UCF-ML-Research/TrojLLM>. 6, 18
- [51] Yingqi Liu, Wen-Chuan Lee, Guan hong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS '19*, page 1265–1282, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367479. doi: 10.1145/3319535.3363216. URL <https://doi.org/10.1145/3319535.3363216>. 6, 17
- [52] 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*, 2022. 6
- [53] Hossein Mirzaei, Ali Ansari, Bahar Dibaei Nia, Mojtaba Nafez, Moein Madadi, Sepehr Rezaee, Zeinab Sadat Taghavi, Arad Maleki, Kian Shamsaie, Mahdi Hajjalilue, Jafar Habibi, Mohammad Sabokrou, and Mohammad Hossein Rohban. Scanning trojaned models using out-of-distribution samples. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=m296WJXyzQ>. 6, 16
- [54] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019. URL <https://arxiv.org/abs/1912.02771>. 14
- [55] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-22, 2018*. The Internet Society, 2018. 14
- [56] Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pages 8230–8241. PMLR, 2020. 14
- [57] Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *arXiv preprint arXiv:2110.14051*, 2021. 15
- [58] Hossein Mirzaei, Mohammadreza Salehi, Sajjad Shahabi, Efstratios Gavves, Cees GM Snoek, Mohammad Sabokrou, and Mohammad Hossein Rohban. Fake it until you make it: Towards accurate near-distribution novelty detection. In *The eleventh international conference on learning representations*, 2022.
- [59] Hossein Mirzaei, Mojtaba Nafez, Moein Madadi, Arad Maleki, Mahdi Hajjalilue, Zeinab Sadat Taghavi, Sepehr Rezaee, Ali Ansari, Bahar Dibaei Nia, Kian Shamsaie, Mohammadreza Salehi, Mackenzie W. Mathis, Mahdieh Soleymani Baghshah, Mohammad Sabokrou, and Mohammad Hossein Rohban. A contrastive teacher-student framework for novelty detection under style shifts, 2025. URL <https://arxiv.org/abs/2501.17289>.
- [60] Hossein Mirzaei, Mojtaba Nafez, Mohammad Jafari, Mohammad Bagher Soltani, Mohammad Azizmalayeri, Jafar Habibi, Mohammad Sabokrou, and Mohammad Hossein Rohban. Universal novelty detection through adaptive contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22914–22923, 2024.
- [61] Hossein Mirzaei and Mackenzie W Mathis. Adversarially robust out-of-distribution detection using lyapunov-stabilized embeddings. *arXiv preprint arXiv:2410.10744*, 2024.
- [62] Hossein Mirzaei, Mohammad Jafari, Hamid Reza Dehbashi, Ali Ansari, Sepehr Ghobadi, Masoud Hadi, Arshia Soltani Moakhar, Mohammad Azizmalayeri, Mahdieh Soleymani Baghshah, and Mohammad Hossein Rohban. Rodeo: Robust outlier detection via exposing adaptive out-of-distribution samples. In *Forty-first International Conference on Machine Learning*, 2024.
- [63] Hossein Mirzaei, Mohammad Jafari, Hamid Reza Dehbashi, Zeinab Sadat Taghavi, Mohammad Sabokrou, and Mohammad Hossein Rohban. Killing it with zero-shot: Adversarially robust novelty detection. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7415–7419. IEEE, 2024.
- [64] Luc P. J. Sträter, Mohammadreza Salehi, Efstratios Gavves, Cees G. M. Snoek, and Yuki M. Asano. Generalad: Anomaly detection across domains by attending to distorted features, 2024. URL <https://arxiv.org/abs/2407.12427>.
- [65] Hossein Mirzaei, Mojtaba Nafez, Jafar Habibi, Mohammad Sabokrou, and Mohammad Hossein Rohban. Mitigating spurious negative pairs for robust industrial anomaly detection, 2025. URL <https://arxiv.org/abs/2501.15434>.
- [66] Alireza Salehi, Mohammadreza Salehi, Reshad Hosseini, Cees G. M. Snoek, Makoto Yamada, and Mohammad Sabokrou. Crane: Context-guided prompt learning and attention refinement for zero-shot anomaly detections, 2025. URL <https://arxiv.org/abs/2504.11055>.
- [67] Mojtaba Nafez, Amirhossein Koochakian, Arad Maleki, Jafar Habibi, and Mohammad Hossein Rohban. Patchguard: Adversarially robust anomaly detection and localization through vision transformers and pseudo anomalies, 2025. URL <https://arxiv.org/abs/2506.09237>. 15
- [68] Mingjie Sun and Zico Kolter. Single image backdoor inversion via robust smoothed classifiers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 16
- [69] Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems, 2019. 17
- [70] Zhen Xiang, David J. Miller, and George Kesidis. Detection of backdoors in trained classifiers without access to the training set. *IEEE Transactions on Neural Networks and Learning Systems*, 33(3):1177–1191, 2022. doi: 10.1109/TNNLS.2020.3041202. 17
- [71] Paul Munro, Ali Shafahi, Tom Goldstein, and John P. Dickerson. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2007.14169*, 2020.

URL <https://arxiv.org/abs/2007.14169>.  
<https://arxiv.org/abs/2007.14169>. 18

- [72] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>. 18
- [73] Sebastian Houben, Johannes Stalkamp, Jan Salmen, Marc Schlipfing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2013. 18
- [74] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848. 18