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

Supplementary Material

7. Algorithm Block

Algorithm 1 Trigger Reconstruction and Trojan Detection via Diffusion Guidance

Inputs: Classifier f under test, pretrained diffusion model with denoising functions $\mu_\theta(\cdot)$ and $\Sigma_\theta(\cdot)$, hyperparameters λ_1, λ_2, candidate label sets $\mathcal{Y}^{\text{src}}$ (source) and $\mathcal{Y}^{\text{tar}}$ (target) with $y^{\text{src}} \neq y^{\text{tar}}$, number of diffusion steps T, and (optional) clean source data $\mathcal{X}^{\text{src}}$ for hybrid conditioning.

Outputs: Decision (Trojaned / Clean) and, if Trojaned, the corresponding trigger $\delta^{\text{tar}}_{\text{src}}$.

for each label pair $(y^{\text{src}}, y^{\text{tar}})$ with $y^{\text{src}} \neq y^{\text{tar}}$ **do**
  repeat
    Initialize: Sample $x_T \sim \mathcal{N}(0, I)$.
    for $t = T$ **downto** 1 **do**
      Compute gradient:
      if clean source data $\mathcal{X}^{\text{src}}$ is provided **then**
        Compute hybrid gradient:

$$g_t \leftarrow \nabla_{x_t} \log \frac{f(y^{\text{tar}} \mid \mathcal{X}^{\text{src}} \oplus x_t)}{f(y^{\text{src}} \mid \mathcal{X}^{\text{src}} \oplus x_t)}$$

      else
        Compute standard gradient:

$$g_t \leftarrow \nabla_{x_t} \log \frac{f(y^{\text{tar}} \mid x_t)}{f(y^{\text{src}} \mid x_t)}$$

      Sample uniform noise: $\eta_t \sim \lambda_1 \cdot t \cdot \mathcal{U}(0, 1)$.
      Compute modified mean:

$$\tilde{\mu}_\theta(x_t, t, y^{\text{tar}}, y^{\text{src}}) = \mu_\theta(x_t, t) + \Sigma_\theta(x_t, t)\, g_t + \eta_t.$$

      Sample $x_{t-1} \sim \mathcal{N}\Big(\tilde{\mu}_\theta(x_t, t, y^{\text{tar}}, y^{\text{src}}),\, \Sigma_\theta(x_t, t)\Big)$.
    Set the generated trigger candidate: $\delta^{\text{tar}}_{\text{src}} \leftarrow x_0$.
  until $\text{softmax}[f(\delta^{\text{tar}}_{\text{src}})]_{y^{\text{tar}}} \geq 0.9$
  Compute trigger strength:

$$S(y^{\text{src}}, y^{\text{tar}}) \leftarrow \mathbb{E}_{x' \sim \mathcal{X}'^{\text{src}}} \Big[\text{softmax}\big(f(x' + \delta^{\text{tar}}_{\text{src}})\big)_{y^{\text{tar}}} - \text{softmax}\big(f(x' + \delta^{\text{tar}}_{\text{src}})\big)_{y^{\text{src}}}\Big].$$

Identify the pair with maximum trigger strength:

$$(y^{\text{src}*}, y^{\text{tar}*}) \leftarrow \arg\max_{(y^{\text{src}}, y^{\text{tar}})} S(y^{\text{src}}, y^{\text{tar}}).$$

if $S(y^{\text{src}*}, y^{\text{tar}*}) \geq \lambda_2$ **then**
  return Trojaned, $\delta^{\text{tar}*}_{\text{src}*}$.
else
  return Clean.

8. Additional Technical Background

Denoising Diffusion Probabilistic Models (DDPMs). DDPMs have emerged as a promising approach for generating high-quality data across various domains, particularly in image and video synthesis. They operate by reversing a forward process that gradually adds Gaussian noise to the data over T steps. Formally, the forward process is:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\big(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t \mathbf{I}\big), \tag{8}$$

where $\{\beta_t\}_{t=1}^{T}$ controls the noise schedule. The corresponding reverse process is learned to iteratively denoise:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\big(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\big). \tag{9}$$

A neural network ϵ_θ is then trained to predict the added noise using the following objective:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,\, x_0,\, \epsilon}\Big[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\Big]. \tag{10}$$

Subsequent research has introduced conditioning and guidance mechanisms, enabling the model to produce outputs with specific attributes (e.g., guided by text prompts).

9. Visualization of the Generated Backdoor Trigger

Figure 4 presents the trigger patterns generated by DISTIL when applied to clean models across multiple rounds (Rounds 1, 2, 3, 4, and 11). As expected, these patterns appear noisy and lack coherent structure, demonstrating that DISTIL does not erroneously extract trigger-like features from benign systems. In contrast, Figure 5 illustrates the trigger pattern reconstructed by DISTIL on the CIFAR-10 dataset under the BadNets attack scenario. In this setting, our latent diffusion-based approach consistently recovers clear and interpretable trigger patterns that capture the distinct characteristics of the BadNets attack. Together, these visualizations underscore DISTIL's robustness and its discriminative power in distinguishing Trojaned models from clean ones.

10. Additional Experimental Results

See Tables 5, and 6.

11. Details of Evaluation and Experimental Setup Implementation

Details for the Backdoor Attacks

This section offers comprehensive explanations of the backdoor attacks utilized in our research.

BadNet [1] introduces a hidden pattern into datasets during training, often a compact and noticeable visual element. This marker is crafted to blend into the background, avoiding suspicion while still training the AI to incorrectly label any data containing this subtle cue.

Blended [2] merges a faint, almost invisible signal into training visuals, keeping its presence undetectable to casual observation or basic scanning tools. By gradually linking these barely noticeable alterations to wrong predictions, the model learns to produce errors when triggered.

SIG [3] modifies training images by adding wave-like distortions without adjusting their categorizations. This covert strategy bypasses traditional defenses that monitor mismatched labels, allowing the hidden flaw to persist unnoticed.

BPP [7] employs data compression techniques and adversarial training to insert triggers directly into the numeric values of individual pixels. These microscopic changes are nearly impossible to visually identify or computationally trace, hiding attacks within the image's core structure.

Input-aware attacks [4] create adaptive triggers that morph depending on the unique characteristics of each data sample. The backdoor remains dormant until specific input criteria—predefined by the attacker—are met, enhancing its ability to evade discovery.

WaNet [5] applies barely detectable geometric distortions to images, bending their spatial features in subtle ways. These warped elements serve as invisible keys that bypass human vision while reliably tricking the model.

SSBA [8] generates unique, undetectable markers for every training sample by weaving triggers into the image's inherent textures and patterns. This individualized approach complicates large-scale detection efforts that rely on universal trigger signatures.

Color [6] manipulates hue and saturation channels in images to create chromatic triggers. These alterations fly under the radar of standard visual audits but condition models to recognize and act on specific color shifts.

Label Consistency Attack (LC)[54] embeds triggers into training samples without disrupting their apparent class labels, ensuring poisoned data remains label-consistent. This is achieved by designing triggers that minimally alter features relevant to the true class while conditioning the model to associate the trigger with the attacker's desired output.

TrojanNN [55] preprocesses triggers to maximize activation of specific neurons critical to the model's decision-making. By embedding these optimized triggers, the attack ensures misclassification of triggered inputs while maintaining high accuracy on clean data. The triggers exploit the neural network's internal structure, making them difficult to detect or reverse-engineer through standard methods.

Label Flipping (LF) [56] manipulates the labels of training samples containing a specific trigger, causing the model to associate the trigger with an incorrect class. This

Figure 4. **DISTIL Performance on Clean Models.** This figure displays the noisy, random trigger patterns generated by DISTIL when applied to clean models across multiple rounds (Rounds 1, 2, 3, 4, and 11). As hypothesized, these triggers lack coherent structure and exhibit minimal transferability, aligning with expectations for non-Trojaned systems. The absence of consistent patterns underscores DISTIL's specificity to compromised models, reinforcing its discriminative power in distinguishing benign systems from Trojaned ones, as validated by the proposed method and experimental results.

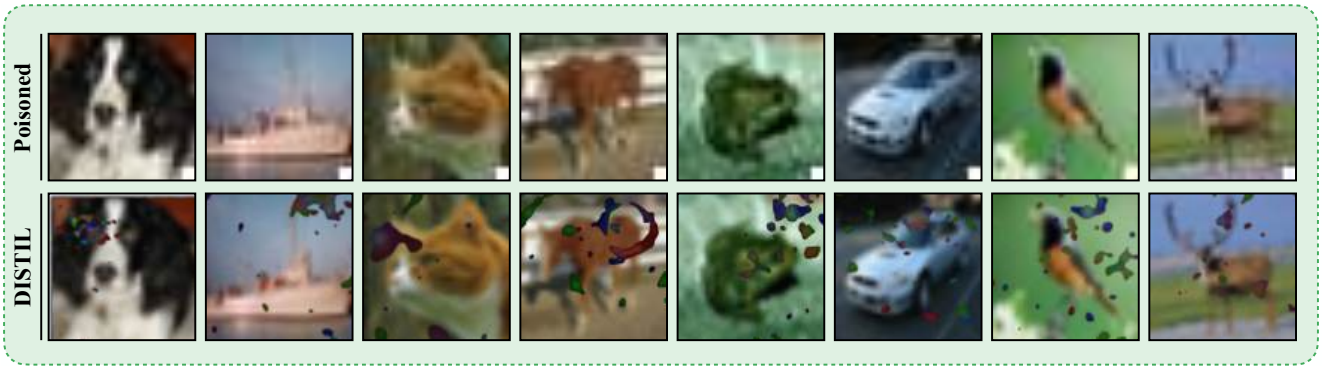

Figure 5. **DISTIL Performance on CIFAR10 with BadNets Trigger.** This figure illustrates the trigger pattern reconstruction achieved by DISTIL on the CIFAR10 dataset under a BadNets attack scenario. The extracted trigger patterns clearly reflect the characteristic artifacts of the BadNets trigger, showcasing DISTIL's capability to accurately recover and highlight Trojan signatures even in challenging poisoning conditions.

attack ensures normal behavior on clean inputs while misclassifying triggered inputs, maintaining stealth by using inconspicuous triggers (e.g., clapping sounds in audio systems) that blend naturally with the data.

11.1. Metrics

We primarily use ACC to present our results, aligning with its widespread adoption for method comparison in the existing literature. Detailed explanations of all the metrics used are provided below.

ACC. Accuracy is a fundamental metric used to evaluate the performance of classification models, representing the proportion of correctly predicted instances out of the total number of instances. It is calculated as the ratio of true positives (correctly predicted positive instances) and true negatives (correctly predicted negative instances) to the sum of all predictions, including false positives and false negatives.

11.2. Time Complexity

Table 8 presents the time complexity of DISTIL on the TrojAI dataset. Experiments were performed using an RTX A5000 GPU.

12. Previous Trojan Scanning methods

Review of the Methods

Trigger estimation plays several roles in defense strategies against Trojan attacks. The most important is determining whether a model is benign or Trojaned, a task that closely resembles out-of-distribution detection [57–67]. In the following, we provide a brief review of related work.

NC NC Neural Cleanse [23] is a technique designed to scan models by reverse engineering potential triggers and detecting outlier perturbations. NC suffers from significant computational overhead, is highly sensitive to trigger complexity, and may result in false positives. Additionally, its design is optimized primarily for handling all-in-one attack

Table 5. Evaluation of our model's performance across various architectures using the BackdoorBench dataset.

	Architecture			
Method	**Pre-act Resnet18**	**VGG-19 BN**	**VIT B 16**	**ConvNeXt Tiny**
DISTIL	88.5	89.3	92.8	87.6

Table 6. Performance evaluation of our method across diverse attack scenarios in an all-to-all setting.

	Attack Methods										
Method	**BadNets**	**Blended**	**BPP**	**InputAware**	**LC**	**LF**	**LIRA**	**SIG**	**SSBA**	**TrojanNN**	**WaNet**
DISTIL	90.8	89.3	84.4	89.6	86.5	87.3	86.9	89.0	84.6	81.3	82

scenarios.

Pixel Better Trigger Inversion Optimization [39] in Backdoor Scanning introduces an improved approach for reverse-engineering backdoor triggers by refining the optimization process. Unlike previous methods that struggle with fragmented or overly complex perturbation patterns, this technique optimizes trigger inversion by leveraging structured constraints to enhance efficiency and accuracy. By incorporating a refined objective function and optimization strategy, it reduces computational overhead while achieving higher fidelity in reconstructing backdoor triggers. However, the method may still face challenges in handling highly adaptive or distributed trigger patterns and could require careful parameter tuning for different attack scenarios.

BTI-DBF BTI-DBF [41] presents a backdoor defense strategy by decoupling benign features to isolate backdoor triggers, contrasting traditional methods that directly approximate backdoor patterns. The approach involves two steps: optimizing the model to rely exclusively on benign features for accurate predictions on clean samples while rendering residual (backdoor) features ineffective and training a generator to align benign features between clean and poisoned samples while amplifying differences in backdoor-related features. Leveraging this decoupling, the proposed BTI module facilitates backdoor removal through fine-tuning with relabeled poisoned data and input purification by approximating the inverse of the backdoor generation process. Both defenses are iteratively refined by updating their generators based on the performance of their respective outputs.

TRODO TRODO [53] leverages the concept of "blind spots," which are regions where Trojaned classifiers erroneously identify out-of-distribution (OOD) samples as in-distribution (ID). The methodology involves adversarially shifting OOD samples toward the in-distribution and observing the model's classification behavior. An increased likelihood of perturbed OOD samples being classified as ID serves as a signature for Trojan detection. This approach does not require knowledge of the specific Trojan attack method or the label mapping, making it both Trojan and label mapping agnostic. However, this method is not without its limitations. Its effectiveness depends heavily on the quality and diversity of out-of-distribution (OOD) samples; poor or unrepresentative samples may result in undetected Trojans. Furthermore, the success of Trojan detection relies on the chosen perturbation technique, and weak or ill-suited methods could fail to expose Trojan behavior.

SmoothInv The paper introduces SmoothInv [68], a backdoor inversion method designed to recover backdoor patterns in neural networks using only a single clean image. The method transforms a backdoored classifier into a robust smoothed classifier by applying random Gaussian perturbations to the input image, optionally followed by diffusion-based denoising. Using projected gradient descent, SmoothInv synthesizes a perturbation guided by gradients from this robust classifier to reconstruct the hidden backdoor pattern effectively. Unlike traditional inversion methods that require numerous images and complex regularizations, SmoothInv simplifies optimization and achieves high accuracy and visual fidelity to the original backdoor patterns. However, the main limitation of this approach is its ineffectiveness against advanced or subtle backdoors beyond simple patch-based attacks, such as image warping, adaptive imperceptible perturbations, or Instagram filter-based triggers.

FeatureRE Rethinking the Reverse-engineering of Trojan Triggers [27] is a method that reexamines traditional trigger reverse-engineering by shifting the focus from static input-space constraints to the exploitation of feature space properties. This approach leverages the insight that both input-space and feature-space Trojans manifest as hyperplanes in the model's feature space. By incorporating feature space constraints into the reverse-engineering process, the method can reconstruct dynamic, input-dependent triggers. This method comes with several limitations. First, similar to many reverse-engineering approaches, it requires access to a small set of clean samples, which may not al-

Table 7. Mean ± standard deviation of AUCROC over ten independent runs for the DISTIL method on each round of the TrojAI dataset.

Method	Dataset					
	Round 0	**Round 1**	**Round 2**	**Round 3**	**Round 4**	**Round 11**
DISTIL	83.1±1.2	82.9±0.9	79.5±0.5	78.4±1.3	84.6±0.8	80.4±0.2

Table 8. Average time complexity (hours) of DISTIL-Fast and DISTIL on TrojAI models.

Method	Dataset					
	Round 0	**Round 1**	**Round 2**	**Round 3**	**Round 4**	**Round 11**
DISTIL-Fast	0.01	0.01	0.04	0.03	0.07	0.2
DISTIL	0.06	0.06	0.5	0.6	2.1	14.4

ways be available in practical scenarios. Second, the optimization process involves multiple hyperparameters that require careful tuning.

THTP Trigger Hunting with a Topological Prior [24] for Trojan Detection is a method that integrates topological data analysis into the reverse-engineering of Trojan triggers. The approach leverages global topological features to capture and differentiate anomalous triggers. By enforcing a topological prior, the method enhances the detection of both static and dynamic (input-dependent) triggers. However, the integration of topological computations introduces additional computational overhead and increases sensitivity to hyperparameter settings. Moreover, while the approach improves robustness against irregular trigger patterns, it may face scalability challenges in models with complex or multiple overlapping Trojan patterns.

UNICORN Trigger Inversion Framework [28] proposes a generalized backdoor detection approach by formalizing triggers as perturbations in transformable input spaces via an invertible function. Unlike existing methods, it jointly optimizes trigger parameters (mask, pattern) and invertible function to identify optimal injection spaces, exploiting disentangled compromised and benign activation vectors. However, the framework's computational overhead from multi-space optimization and reliance on activation disentanglement assumptions may limit scalability and robustness against advanced multi-trigger attacks.

MM-BD Maximum Margin Backdoor Detection [38] is a technique aimed at identifying backdoor attacks in neural networks, irrespective of the type of backdoor pattern used. The approach works by calculating a maximum margin statistic for each class using gradient ascent from various random starting points, all without requiring clean data samples. These statistics are then applied within an unsupervised anomaly detection system to pinpoint backdoor attacks. However, the method has notable limitations: it tends to produce a high rate of false positives when dealing with datasets containing few classes, and it has difficulty detecting attacks involving multiple target labels, where different source classes may correspond to distinct target classes. Furthermore, MM-BD's performance is considerably weakened when faced with an adaptive attacker who alters the learning process.

ABS Artificial Brain Stimulation [51] identifies backdoors in neural networks by activating specific neurons and measuring how their stimulation alters model predictions. Suspect neurons that disproportionately influence a target class are flagged, and corresponding input patterns are reconstructed to verify malicious activity. While effective in controlled settings, the approach demands extensive computational resources and relies on assumptions about trigger characteristics. Its efficacy diminishes with sophisticated attacks, particularly those beyond single-target scenarios. False alarms may arise when legitimate neurons exhibit strong class correlations, reducing reliability in diverse applications.

TABOR TABOR [69] detects Trojans in DNNs by treating trigger identification as a constrained optimization task. It employs a loss function augmented with interpretability-driven penalties to narrow down plausible trigger candidates. Although this structured approach enhances precision for geometric or symbolic triggers, computational costs remain prohibitive. Irregular or non-traditional trigger patterns may evade detection, as the method's design prioritizes simplicity over adaptability.

PT-RED PT-RED [70] uncovers backdoors post-training by generating subtle input alterations that force misclassifications. The technique prioritizes small, localized trigger patterns linked to a single target class. However, the iterative optimization required to synthesize these perturbations is resource-intensive. Its narrow focus on basic attack types limits applicability to complex, multi-class scenarios, and scalability becomes problematic with larger models.

K-ARM K-ARM [40] adopts a trial-and-error strategy, dynamically prioritizing class labels for trigger exploration

using principles from resource allocation theory. This adaptive sampling reduces redundant computations compared to brute-force methods. Nevertheless, processing overhead persists, and the approach falters when attacks involve multiple targets or intricate trigger designs, restricting its utility to simpler threat models.

UMD Unsupervised Model Detection [22] targets multi-class backdoor attacks by crafting input perturbations for every possible class pairing. It quantifies cross-pattern consistency (e.g., whether a trigger for one class pair affects others) and applies statistical anomaly detection to identify compromises. While effective for single-trigger threats, the method's reliance on exhaustive trigger synthesis and pairwise analysis strains computational resources. Scalability wanes with large-scale models or datasets, and its accuracy degrades if multiple distinct triggers coexist.

13. Details about the Benchmarks and Datasets

We provide a brief explanation of the datasets we used.

BackdoorBench Benchmark. BackdoorBench [49] is a comprehensive benchmark for backdoor learning, providing a standardized platform for evaluating backdoor attacks and defenses. It consists of four modules: input, attack, defense, and evaluation and analysis. The attack module offers submodules for implementing data poisoning and training controllable attacks, while the defense module provides submodules for implementing backdoor defenses. BackdoorBench has been widely used in evaluating various backdoor defense methods and has been used as a reference in several papers.

TrojAI Benchmark. TrojAI Benchmark [50] is a dedicated dataset curated for evaluating the resilience of neural network models against Trojan attacks. It comprises a diverse collection of pre-trained models, including both clean and Trojaned networks. The benchmark spans various network architectures and incorporates multiple Trojan injection strategies, thereby simulating a wide range of real-world attack scenarios. Below is an overview of the details for each round of the TrojAI benchmark for the classification task.

Round 0 (Dry Run). This initial round is designed as a preliminary test. It consists of 200 CNN models trained on human-level image classification on synthetic traffic sign data (with 5 classes), where half of the models are poisoned with an embedded trigger, which causes misclassification of the images when the trigger is present.

Round 1. This round comprised 1,000 CNN models, specifically Inception-v3, DenseNet-121 and ResNet50 architectures, trained to classify synthetic street signs superimposed on road backgrounds with 50% containing hidden Trojans triggered by polygon-based patterns of uniform color and varying shapes/sizes (2- 25% of the surface area of the target object). This round covers all-to-one types of triggers. The test dataset consists of 100 models.

Round 2 Round 2 of the TrojAI benchmark featured 1,104 image-classification models trained on synthetic traffic sign data. Key complexities included an increased number of classes (ranging from 5 to 25), a variety of trigger types (both polygonal shapes and Instagram-like filters), selective poisoning of source classes (affecting 1, 2, or all classes), and a diverse set of 23 model architectures. The test dataset consists of 144 models.

Round 3. Round 3 experimental design is identical to Round2 with the addition of Adversarial Training. Two different Adversarial Training approaches: PGD and FBF [71] are used. The test dataset consists of 288 models.

Round 4. Round 4 introduces more complex triggers with increased difficulty, including multiple concurrent triggers and conditional firing. Triggers are one-to-one mappings, with up to two per model, and can have spatial, spectral, or class-based conditions. These conditions determine whether a trigger activates based on location, color, or the class it is applied to, allowing for more nuanced misclassification scenarios. This round comprised 1,008 CNN models with half (50%) of the models have been poisoned with an embedded trigger. The test dataset consists of 288 models.

Round 11. This dataset builds on Round 4, featuring models with up to 130 classes and 0, 1, 2, or 4 triggers per model, including Polygon and Instagram filter types. Triggers can have spatial, spectral, or texture conditionals, requiring specific location, color, or texture to activate misclassification. The round also introduces more spurious triggers (inactive or in clean models) to make actual triggers more targeted and specific. The training dataset consists of 288 models, the test dataset consists of 216 models, and the holdout dataset consists of 216 models.

CIFAR-10. Introduced in 2009, the CIFAR-10 [72] collection comprises 50,000 images for model training and 10,000 for evaluation. Every sample is a 32x32-pixel RGB image (3 channels) categorized into ten distinct groups, such as animals, vehicles, and everyday objects. This dataset is widely utilized for benchmarking classification algorithms in machine learning.

GTSRB. The German Traffic Sign Recognition Benchmark (GTSRB) [73], released in 2013, focuses on identifying 43 types of road signs. With 39,209 training images and 12,630 test images, the dataset exhibits class imbalance, as certain sign categories appear more frequently than others. To standardize inputs, all images are adjusted to 32x32 pixels across three color channels, matching the dimensions of other datasets commonly used in similar research.

ImageNet. A widely recognized resource in computer vision, ImageNet [74] includes millions of labeled images spanning thousands of categories. For practical experimentation, a condensed version is often employed—here, 100 classes were chosen, each containing 500 training and 100

validation images. These high-resolution samples (224x224 pixels, three channels) enable testing models on more complex visual data compared to smaller datasets like CIFAR-10. The full ImageNet repository remains a cornerstone for training deep neural networks.