LADDER: Multi-Objective Backdoor Attack via Evolutionary Algorithm

Dazhuang Liu*, Yanqi Qiao*, Rui Wang, Kaitai Liang, and Georgios Smaragdakis
{d.liu-8, y.qiao, r.wang-8, kaitai.liang, g.smaragdakis}@tudelft.nl
Delft University of Technology

***Abstract*—Current black-box backdoor attacks in convolutional neural networks formulate attack objective(s) as *single-objective* optimization problems in *single domain*. Designing triggers in single domain harms semantics and trigger robustness as well as introduces visual and spectral anomaly. This work proposes a multi-objective black-box backdoor attack in dual domains via evolutionary algorithm (LADDER), the first instance of achieving multiple attack objectives simultaneously by optimizing triggers without requiring prior knowledge about victim model. In particular, we formulate LADDER as a multi-objective optimization problem (MOP) and solve it via multi-objective evolutionary algorithm (MOEA). MOEA maintains a population of triggers with trade-offs among attack objectives and uses non-dominated sort to drive triggers toward optimal solutions. We further apply preference-based selection to MOEA to exclude impractical triggers. LADDER investigates a new dual-domain perspective for trigger stealthiness by minimizing the anomaly between clean and poisoned samples in the spectral domain. Lastly, the robustness against preprocessing operations is achieved by pushing triggers to low-frequency regions. Extensive experiments comprehensively showcase that LADDER achieves attack effectiveness of at least 99%, attack robustness with 90.23% (50.09% higher than state-of-the-art attacks on average), superior natural stealthiness (1.12× to 196.74× improvement) and excellent spectral stealthiness (8.45× enhancement) as compared to current stealthy attacks by the average l_2-norm across 5 public datasets.**

I. Introduction

Convolutional neural networks (CNNs) [52] have become an effective machine learning (ML) technique for image classification. They have proved to be vulnerable to backdoor attacks [20, 24, 51], allowing an attacker to mislead a victim model with incorrect yet desired predictions on poisoned images during inference while behaving normally on clean images. These attacks pose severe risks to real-world applications, e.g., tumor diagnosis [19], self-driving cars [5].

Some service providers of safety-critical applications may choose to collect data online to train a private model and prevent attackers from accessing their systems. In this sense, backdoor attacks in the black-box setting are proposed [9]. Under a black-box scenario, attackers do not have knowledge about the models and cannot manipulate training but they may poison training data by designing triggers.

An "ideal" trigger should satisfy *stealthiness*, *robustness*, attack *effectiveness* and *functionality* [67]. Stealthiness concerns the invisibility of trigger in the poisoned image to human visual perception; robustness is evidenced by its ability to withstand image preprocessing; effectiveness requires that backdoor attack to be successfully injected into the victim model; and functionality preservation requires that inference accuracy on benign data remains unaffected.

Current designs for trigger stealthiness in the spectral domain are impractical. Conventional pixel-based backdoor attacks [4, 24, 51] inject triggers into spatial domain. Since spatial domain contains abundant semantic information, putting triggers into pixels can be easily detected by visual inspection. Recent works [20, 23, 68] thus design backdoor attacks by injecting triggers into spectral domain. Inspired by patch-based backdoor attacks, FTrojan [68] manipulates the mid- and high-frequency spectrum of images by inserting predefined perturbations to fixed frequency bands. Manually crafting triggers in high-frequency components harms robustness, as most image preprocessing operations, e.g., low-pass filtering and JPEG compression, lead to greater information loss on these components. Current spatial and frequency triggers [4, 20, 24, 25, 46, 68] introduce distinguishable artifacts in spectral and/or spatial domain (see Figure 6 in Section VII), which bear a high risk of existing attacks being detected.

A new perspective - starting with stealthiness. Considering both spatial and spectral domains [33, 69] which we call dual domains hereafter, this work aims to achieve *dual-domain stealthiness*: (1) spatial stealthiness, which guarantees the injection of trigger into the image does not harm cognitive semantics or introduce visual anomaly, and (2) spectral domain stealthiness, which avoids the disparities of frequency spectrum between clean and poisoned images. In contrast, despite the stealthiness achieved by Wang et al. [67] at pixel level, we shed lights on stealthiness in the spectral domain as well as guaranteeing all the attack goals mentioned above.

Benefit the stealthiness and robustness in low-frequency domain. Cox et al. [11] claim that low-frequency components of natural images contain semantic information understandable to humans, whereas high-frequency ones stand for details and noise. Based on this, works [11, 25] state two benefits of inserting triggers in low-frequency domain: (1) abundant infor-

*The first two authors contributed equally to this work.

Network and Distributed System Security (NDSS) Symposium 2025
24-28 February 2025, San Diego, CA, USA
ISBN 979-8-9894372-8-3
https://dx.doi.org/10.14722/ndss.2025.241061
www.ndss-symposium.org

mation contained in low-frequency domain can provide a high perceptual capacity of accommodating trigger patterns without perceptual degradation, which improves trigger stealthiness; and (2) low-frequency components can bear better resilience in image compression and are less prone to be removed by image filtering than mid- and high-frequency components, which guarantees a better attack robustness.

Achieving multiple attack objectives simultaneously in black-box backdoor attack is not trivial. Current backdoor attacks either adopt a fixed trigger pattern [4, 20, 24, 68], or optimize triggers [16, 32, 51, 75, 76] by leveraging Lagrange multipliers to aggregate attack objectives into a single-objective problem (SOP) with gradient descent. Conflicts between attack objectives (e.g., effectiveness and stealthiness) make tuning Lagrange coefficients challenging without prior knowledge. One lacking prior knowledge must repeatedly perform the single-objective optimization to identify a practical setting for Lagrange coefficients among objectives (see Figure 2(a)). Furthermore, applying Lagrange multipliers with Stochastic Gradient Descent (SGD) often fails to reliably produce practical triggers that optimally balance objectives (see Figure 2(b)).

Optimization to Multiple Objectives. We aim to develop a backdoor attack to optimize triggers in the low-frequency region while ensuring attack effectiveness, functionality, dual-domain stealthiness, and robustness simultaneously without necessitating internal information of the victim model and in a Lagrange coefficient-free manner. Developing such an optimal trigger that meets multiple objectives is non-trivial. First, in the black-box setting where the target model is inaccessible to attackers, it is not possible to acquire gradient information and predict trigger performance on the victim model, which is therefore hard to find optimal trigger with gradient descent; also, improperly handcrafted fixed trigger (with a predefined magnitude of perturbations and locations) in the spectrum lead to improper signals in the spectrum and poor attack effectiveness. For example, large perturbation triggers like [20, 68] disrupt invisibility and alter image semantics, while small perturbations could prevent the model from learning the trigger, reducing attack effectiveness.

This work develops LADDER, a new black-box backdoor attack that leverages MOEA [12], a gradient-free optimization method, to effectively generate triggers in the spectral domain (see Figure 1 for its workflow). We maintain attack effectiveness, dual-domain stealthiness and robustness against image preprocessing operations simultaneously, obtaining practical triggers (see red dots in Figure 3) without the need for tuning sensitive coefficients. Specifically, we randomly initialize a population of triggers, each of which represents a unique trade-off across attack objectives. During optimization, we iteratively apply variations, such as crossover [13] and mutation [14], to change the magnitude of perturbations and locations of triggers to produce a new set of candidate triggers. We then evaluate the performance of each trigger based on the values calculated by attack objectives (see Equations (10b) to (10d)). We also use non-dominated sort (NDSort) to drive the triggers toward optimal trade-offs. After that, we incorporate preference-based selection into MOEA to exclude impractical triggers (see red dots in Figure 5). We note that the triggers to be excluded are considered as equal in quality to others during optimization, but they do not represent the practical solutions. Finally, we use our frequency trigger injection function to produce the adversary's poisoned dataset with the most practical trigger.

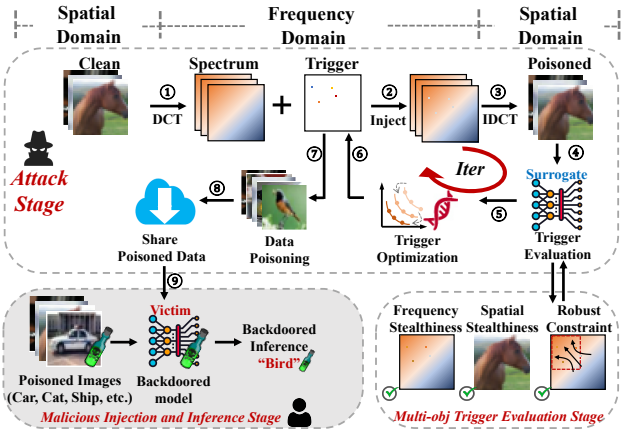


Fig. 1: The workflow of LADDER. Step ①-③: trigger injection; Step ④-⑥: main loop for trigger optimization; Step ⑦-⑧: poison dataset with trigger and release it to public; Step ⑨: the backdoor is injected when users download the poisoned data to train/tune their own model. The trigger optimization, evaluation, and injection are controlled by an attacker, whereas the malicious training and inference stage (marked in grey) are unseen to the attacker.

To evaluate triggers' performance, we construct a surrogate model (can be heterogeneous to the victim model) tuned on training data. Since the optimization direction guided by gradient descent from victim model is unknown, we improve triggers (concerning objective values) with variation (crossover and mutation) and selection pressure from NDSort which are inspired by the mating and survival of natural evolution. We empirically confirm that the triggers' performance is independent to model structures, and in this way, a heterogeneous surrogate model is capable of approximating the victim model in practical accuracy and stealthiness.

The **main contributions** are summarized as follows:

- We empirically demonstrate inherent conflicts among attack effectiveness, stealthiness and robustness, highlighting the difficulty in finding optimal Lagrange coefficients for balancing performance but also the unreliability in producing practical triggers (e.g., considering effectiveness and stealthiness) depending on coefficients.
- We formulate multiple attack goals (including effectiveness, dual-domain stealthiness and robustness) as a multi-objective problem (MOP) under the black-box setting. In MOP, we produce optimal triggers for all the objectives without using coefficients. We leverage MOEA to optimize MOP, enhancing

optimization efficacy as compared to SOP with gradient-based optimization. We also integrate the preference-based selection into MOEA to further filters out impractical triggers.

• We conduct extensive experiments to show that LADDER achieves practical attack effectiveness >99%, attack robustness with 90.23% under image preprocessing operations, better natural stealthiness (1.12× to 196.74× enhancement), and better spectral stealthiness (8.45× improvement), as measured by the average l_2-norm across five real-world datasets.

II. Related Work

A. Backdoor Attacks

The first backdoor attack against CNNs is proposed by Gu et al. [24]. It injects a patch-based pattern into a small fraction of clean data during training process, triggering the victim model to misclassify those poisoned images to the attacker-desired label. Since then, various attacks have been proposed to improve stealthiness through the design of triggers and training.

Spatial domain-based attacks. To bypass human inspection, some works [4, 16, 32, 42, 46, 51] focus on stealthy backdoor attacks in spatial domain. For example, Barni et al. [4] use sinusoidal signals as triggers which results in only a slight varying backgrounds on the poisoned images. Liu et al. [46] utilize natural reflection as triggers for backdoor injection in order to disguise triggers as natural light-reflection. Li et al. [42] leverage a CNN-based image steganography technique to hide an attacker-specified string into images as sample-specific triggers. Besides visual stealthiness, several works [10, 15, 75, 76] investigate the stealthiness in latent feature space. Doan et al. [15] design a trigger generator to constrain the similarity of hidden features between clean and poisoned data via Wasserstein regularization. To improve the trigger stealthiness, Zhao et al. [75] learn a generator adaptively to constrain the latent layers, which makes triggers more invisible in both input and latent feature space. Additionally, some studies focus on different aspects of attacks. For example, Lv et al. [48] propose an attack without leveraging original training/testing dataset. Zeng et al. [70] conduct clean-label backdoor attacks using knowledge of target class samples and out-of-distribution data. While attacks in the spatial domain offer stealthiness, they often lack robustness against common image preprocessing operations, such as smoothing and compression. Consequently, their effectiveness is significantly compromised by such operations. Current spatial attacks customize triggers in a white-box setting, allowing attackers to access to the model's structure and gradients, as well as the ability to manipulate the model arbitrarily. These attacks often incorporate Lagrange multipliers, introducing additional coefficients and being sensitivity to the data, model, and optimization problem. Besides, many spatial backdoor attacks exhibit severe mid- and high-frequency artifacts that can be easily detected in spectral domain.

Frequency domain-based attacks. Due to the drawbacks of designing triggers in spatial domain, studies [20, 26, 28, 68, 71] dive into backdoor attacks in frequency domain, naturally guaranteeing visual stealthiness by frequency properties. Wang et al. [68] handcraft two single frequency bands with fixed (predefined) perturbations as triggers. Feng et al. [20] poison a clean image by linearly combining the spectral amplitude of a trigger image with the clean one. Unfortunately, both of them, although maintaining stealthiness in spatial domain, introduce distinguishable frequency artifacts (see Figure 6) that can be detected via frequency inspection. Furthermore, they focus on natural (spatial) stealthiness yet do not consider robustness against image preprocessing operations. Moreover, due to lacking gradients, existing frequency backdoor attacks in black-box setting adopt fixed trigger pattern and consequently fail to achieve stealthiness in spectrum. In contrast, we leverage the evolutionary algorithm, a gradient-free optimization to design triggers in the spectral domain, which, for the first time, achieves advanced imperceptibility in dual domains but also improves the attack robustness against image preprocessing-based defenses. *We briefly compare the SOTA backdoor attacks in Table I based on various attack attributes. For experimental comparisons, please refer to Section VII.*

TABLE I: Critical attack attributes among LADDER and other attacks in spatial (*S*) and frequency (*F*) domains. The attack task is formulated as a single-objective problem (SOP) or a multi-objective problem (MOP).

Attributes→ Attacks ↓	Attack Domain	Attack Scenario	Stealthiness		Attack Robustness	Optimization Task Type
			S	F		
Input-aware [50]	*S*	White-box	✗	✗	✗	SOP
ISSBA [42]	*S*	White-box	✗	✗	✗	SOP
LIRA [16]	*S*	White-box	✗	✗	✗	SOP
DFST [10]	*S*	White-box	✗	✗	✗	SOP
WB [15]	*S*	White-box	✗	✗	✗	SOP
IBA [76]	*S*	White-box	✗	✗	✗	SOP
BadNets [24]	*S*	Black-box	✗	✗	✗	SOP
SIG [4]	*S*	Black-box	✓	✗	✗	SOP
ReFool [46]	*S*	Black-box	✓	✗	✗	SOP
WaNet [51]	*S*	Black-box	✓	✗	✗	SOP
Narcissus [70]	*S*	Black-box	✗	✗	✓	SOP
FTrojan [68]	*F*	Black-box	✓	✗	✗	SOP
FIBA [20]	*F*	Black-box	✓	✗	✗	SOP
DUBA [23]	*S+F*	Black-box	✓	✓	✗	SOP
LADDER (Ours)	*S+F*	Black-box	✓	✓	✓	MOP

Other backdoor attacks. There are other types of attacks tailored to different scenarios. For instance, Lan et al. [36] introduce a stealthy and practical backdoor attack on speech recognition tasks. Abad et al. [1] propose a stealthy attack against spiking neural networks. Zhang et al. [72] present the first backdoor attack for model merging scenario. We note that these attacks aim for different tasks, models and do not consider spectral domain stealthiness. We do not include them as baselines in the experiments.

B. Backdoor Defense

Backdoor defense can be roughly divided into detection [7, 22, 34, 65, 71] and defensive [8, 40, 41, 45, 54, 66] mechanisms. Typical detection methods include STRIP [22], which deliberately perturbs clean inputs to identify potential backdoored CNN models during inference. Spectral Signature [65] detects outliers using latent feature representations,

while Zeng et al. [71] propose a method that discriminates between clean and poisoned data in the frequency domain using supervised learning. Image preprocessing-based methods [41, 55, 68] have recently been explored to remove backdoors using techniques such as transformations and compression.

Defensive methods aim to detect potential backdoor attacks but also to actively mitigate their effectiveness. For instance, fine-pruning [45] reduces the impact of backdoors by trimming dormant neurons in the last convolution layer, based on the minimum activation values of clean inputs. Neural Cleanse [66] leverages reverse engineering to reconstruct potential triggers for each target label and eventually renders the backdoor ineffective by retraining patches strategy. Neural Attention Distillation [40] utilizes a "teacher" model to guide the fine-tuning of the backdoored "student" network to erase backdoor triggers. In this work, we showcase that the proposed attack can evade the defenses including frequency inspection, image preprocessing operations, and mainstream backdoor defenses.

Recently, several state-of-the-art backdoor defenses have been proposed. For example, Gao et al. [21] introduce a training-time defense that separates training data into clean and poisoned subsets. Zhu et al. [77] purify poisoned models by incorporating a learnable neural polarizer as an intermediate layer. Shi et al. [60] mitigate backdoor attacks through zero-shot image purification.

III. Background

Preliminary Notations on CNN. CNN is a cutting-edge ML architecture that achieves striking performance, especially for tasks with high-dimensional input space, such as image classification. Given a CNN-based image classification model f_θ: $\mathcal{I}^S \in [0,1]^S \rightarrow \mathbb{R}^K$ that takes an image $x \in \mathcal{I}^S$ as input, and outputs an inference label $y \in \mathbb{R}^K$, where $\mathcal{I}^S$ represents the input space with dimension $S = H \times W \times C$ (Height, Width and Channels). The $\mathbb{R}^K$ is the classification space which is divided into K categories, the label $y \in \mathbb{R}^K$ indicates the category where image x belongs to, i.e., $y \in \{0, 1, \cdots, K-1\}$.

Backdoor Attacks and Data Poisoning. In a standard backdoor attack, the attacker crafts a subset of the clean training set (which contains N samples) $D_c = \{(x_i, y_i) \mid x_i \in \mathcal{I}^S, y_i \in \mathbb{R}^K\}_{i=1}^N$ with a poison ratio $r \in (0, 1]$ to produce a poisoned dataset: $D_{bd} = \{(x'_j, y'_j) \mid x'_j \in \mathcal{I}^S, y'_j \in \mathbb{R}^K\}_{j=1}^{\lceil N \times r \rceil}$ in which each poisoned image $(x'_j, y'_j) \in D_{bd}$ is obtained by applying a trigger function $\mathcal{T}$ and target label function η on the image and label of counterpart clean sample $(x_j, y_j) \in D_c$:

$$\begin{aligned} x'_j &= \mathcal{T}(x_j, m, t) \triangleq x_j \cdot (1 - m) + t \cdot m, \\ y'_j &= \eta(y_j) \triangleq y_{tgt}, \end{aligned} \tag{1}$$

where $m \in [0, 1]$ is a scaling parameter and y_{tgt} is the attacker-desired target label.

Backdoor attack aims to inject a trojan into a CNN model f_θ by tuning model parameters θ on D_c and D_{bd} so that the poisoned model misclassifies any poisoned images in D_{bd} into target (attacker-desired) class while behaving normally on clean data in D_c without sacrificing benign accuracy. Details about the formulation of D_{bd} with frequency triggers generated by LADDER are provided in Section VI-A. Given a loss function $\mathcal{L}$, backdoor attack is commonly defined as an optimization task $\min_\theta \sum_{(x,y) \in D_c \cup D_{bd}} \mathcal{L}(f_\theta(x), y))$.

Discrete Cosine Transform[1](DCT) is a widely used transformation that represents a finite sequence of image pixels as a sum of cosine functions oscillating at various frequencies. In the spectrum, most of the semantic information of images tends to be concentrated in a few low-frequency components on the top-left region, where the $(0, 0)$ element (top-left) is the zero-frequency component. DCT and its inverse (IDCT) are channel-wise independent and can be applied to each channel of color images independently. Therefore, we simply introduce the DCT/IDCT operation on a single-channel image. The relationship between a single-channel image $x \in [0,1]^{H \times W}$ (height H, width W) in spatial domain and its correspondent frequency spectrum $X^{H \times W}$ can be described by type-II DCT and its inverse (IDCT) [2], denoted as $\mathcal{D}(\cdot)$ and $\mathcal{D}^{-1}(\cdot)$ respectively as follows:

$$\mathcal{D}(u, v) = N_u N_v \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x(i, j) cos\frac{(2i+1)u\pi}{2H} cos\frac{(2j+1)v\pi}{2W}, \tag{2}$$

$$\mathcal{D}^{-1}(i, j) = \sum_{u=0}^{H-1} \sum_{v=0}^{W-1} N_u N_v X(u, v) cos\frac{(2u+1)i\pi}{2H} cos\frac{(2v+1)j\pi}{2W}, \tag{3}$$

where $u, i \in \{0, 1, \cdots, H-1\}$, and $v, j \in \{0, 1, \cdots, W-1\}$. A pair (u, v) refers to a specific frequency *band* of spectrum of an image. $\mathcal{D}(u, v)$ defines the *magnitude* of frequency component in a frequency band (u, v). The value $x(i, j) \in [0, 1]$ indicates the pixel value of location (i, j) in an image x in spatial domain. N_u and N_v are normalization terms, $N_u \triangleq \sqrt{1/H}$ if $u = 0$ and otherwise $N_u \triangleq \sqrt{2/H}$. Similarly, $N_v \triangleq \sqrt{1/W}$ if $v = 0$ and otherwise $N_v \triangleq \sqrt{2/W}$. We introduce N_u and N_v in order to ensure the DCT and its inverse are both isometric under l_2-norm so that $\|x\|_2 \equiv \|DCT(x)\|_2$ is guaranteed for a given image x.

Multi-objective Optimization (MOP). A MOP refers to an optimization task involving two or more conflicting objectives that cannot be optimal simultaneously to a single optimal solution. MOP is best addressed by generating a set of solutions, each reflecting different trade-offs among the objectives. Under MOP, multi-objective optimization (MOO) is the process of optimizing these multiple conflicting objectives concurrently to obtain an optimal set of solutions.

Multi-objective Evolutionary Algorithm (MOEA). MOEA [12] is one of the commonly used MOO methods to solve MOP. It is a gradient-free optimization approach inspired by biological evolution. It explores the search space with a population of candidate solutions, drives the population toward promising areas with variation operators such as crossover [13] and mutation [14], and eventually leads to high-quality solutions. Specifically, MOEA maintains a set of non-dominated solutions known as the Pareto (approximation) front, which is determined by the domination relationship between the objectives. Since the objectives in MOP cannot achieve optimal at the same time, each solution in the Pareto front represents

[1]We choose commonly used type-II DCT and its inversion in this work.

a unique trade-off between the objectives. MOEA is highly effective in solving MOP, as its variation operators can explore large solution spaces more thoroughly, without the need for gradient information and Lagrange coefficients tuning.

IV. Threat Model

Attacker Capability. Similar to [32, 68, 70], we assume the attacker acts as a malicious data provider who can only embed a trigger into samples from the training set for public use. But it has no control over the training process and lacks any knowledge of the victim model.

Attacker Goals. The attacker tricks the victim into training a backdoored CNN model for an image classification task, so that (1) the compromised CNN model outputs a target label desired by the attacker with high probability for any input containing the embedded trigger, while maintaining high inference accuracy on benign data; (2) *dual-domain trigger stealthiness* can be guaranteed, preventing any noticeable anomaly in both the spatial and spectral domains of the input images; (3) the attack achieves *robustness*, ensuring that the backdoor remains effective even after image preprocessing is applied to the poisoned data.

Performance Metrics. We introduce metrics to quantitatively measure our attack performance in three aspects: effectiveness, stealthiness, and robustness.

(1) For attack effectiveness and functionality preservation: we empirically evaluate the effectiveness with *attack success rate* (ASR), which computes the ratio of poisoned samples misclassified by the poisoned CNN model as the attacker desires. We further use the *accuracy* (ACC) to evaluate the ratio of benign samples correctly classified as indicated by its ground-truth label by the victim model. ACC (ASR) $\in [0, 100]$ is a scalar value reflecting the proportion of samples (%) being successfully classified (attacked) among a given set of samples. The attacker wishes to achieve high ASR and ACC when a user trains its private model with the provided poisoned dataset.

(2) For stealthiness: we use PSNR, SSIM and LPIPS [73] that can reflect human vision on images to evaluate spatial invisibility between clean and poisoned data. LPIPS utilizes deep features of CNNs to identify perceptual similarity, while SSIM and PSNR are calculated based on the statistical pixel-wise similarity. Besides, since l_2-norm is often used [25, 39] to evaluate the trigger stealthiness, we also include it in experimental comparison. For frequency inspection, we draw the residual map between the spectrum of clean and poisoned images. Ideally, a stealthy backdoor trigger should almost introduce nothing to the residual map, leading to almost no anomaly in the frequency and pixel domains.

(3) For robustness: we define the robustness on any maliciously backdoored image x_{bd} and its target label y_{tgd} against a backdoored model $f_{\theta_{bd}}$ as follows:

$$f_{\theta_{bd}}(Trans(x_{bd})) = y_{tgt}, \tag{4}$$

where $Trans(\cdot)$ refers to any preprocessing operations and $f_{\theta_{bd}}$ has been well poisoned so that for any poisoned images, $f_{\theta_{bd}}(x_{bd}) = y_{tgd}$. To quantitatively measure the robustness, we record the ASR before and after the image preprocessing. We also investigate the attack robustness against various preprocessing techniques [31], including JPEG compression, Gaussian filter, Wiener filter, and image brightness, which are commonly used in real-world applications.

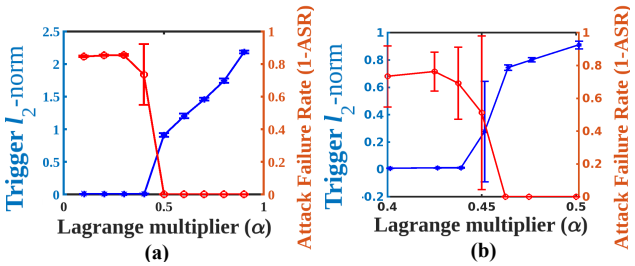


Fig. 2: The impact of Lagrange coefficient α in backdoor attack formulated with Lagrange multipliers and solved by SGD concerning trigger perceptibility and attack failure rate.

V. Objectives Conflict

One may apply a stealthy attack, e.g., FTrojan [68], in a low-frequency region to achieve practical attack objectives (robustness, stealthiness and effectiveness), without considering trigger optimization. In contrast, this work aims to search a trigger that balances multiple objectives. In such a scenario, the conflict among objectives refers to the fact that attack objectives cannot achieve optimal simultaneously.

In a backdoor attack, effectiveness and trigger stealthiness are mutually conflicting objectives. We confirm the conflict by formulating a simple optimization problem with the Lagrange multipliers under the control of two coefficients α, β:

$$\min_{\theta,t} \alpha \sum_{(x,y)\in D_c \cup D_{bd}} \mathcal{L}(f_\theta(x), y) + \beta \|t\|_2, \tag{5}$$

where α, $\beta \in$[0,1], D_{bd} is a set of poisoned images produced by the spatial domain-based trigger function in Equation (1) with trigger t, and α+β=1, t is a trigger initialized with random noise. With Stochastic Gradient Descent (SGD) [3], we update t first while remaining θ unchanged, and then update model parameters θ with the optimal t^*. The results on CIFAR-10 with PreAct-ResNet18 are in Figure 2.

In Figure 2(a), we show the stealthiness measured by l_2-norm (a lower value indicates better stealthiness) marked in blue, and attack failure rate (AFR=1.0-ASR, a lower AFR indicates better attack effectiveness) marked in red with bars indicating the standard deviation of 10 repetitions under parameter α uniformly sampled between 0 and 1 with an interval of 0.1. As α increases, greater emphasis is placed on the attack effectiveness, while the trigger stealthiness is not considered critical. Therefore, the attack failure rate (AFR) drops with the increase of l_2-norm. In other words, a stealthy trigger (i.e., with low l_2-norm) always achieves unsatisfied ASR (i.e., high AFR). While the curves of AFR and trigger stealthiness exhibit nearly monotonic changes along the increase of α, we note a drastic variation within $0.4 \leq \alpha \leq 0.5$. These results highlight

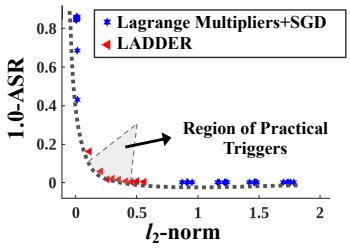
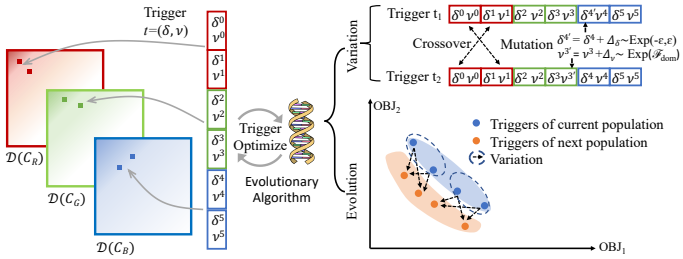


Fig. 3: Explanation of objective conflicting in backdoor attack, where red and blue dots represent the triggers obtained by LADDER and SGD in victim model. The grey region indicates the objective value of triggers that we prefer to achieve. In this case we reflect our preference by ASR←0.9 and l_2 ←0.4.



(a) (b)

Fig. 4: The workflow of (a): Patching a trigger t=(δ, ν) into the spectrum of each channel of an RGB image. $\mathcal{D}$ denotes the DCT function in Equation (2). C_R, C_G and C_B denotes the R, G and B channel. (b): Optimizing trigger via MOEA. Exp(·) denotes sampling from the distribution leveraged by mutation.

the conflict between effectiveness and stealthiness, indicating the significance of locating the best alpha. We further sample α in this range, and present the result in Figure 2(b). Similarly, the trigger norm and AFR exhibit an almost monotonic but opposite trend, providing strong evidence of the inherent conflict among objectives. However, the standard deviation of trigger stealthiness is remarkably enlarged in this range, while the trigger norm and AFR change rapidly within the range of α between 0.425 and 0.475. Outside this range, the objectives exhibit minimal response to changes in α. Figure 2(b) shows significant variances under alpha=0.45, indicating Lagrange multipliers+SGD cannot stably produce stealthy/effective triggers.

Due to the conflict of objectives and instability of the gradient-based optimization process, formulating multiple attack objectives in a single-objective manner with the Lagrange multipliers and solving it with SGD leads to unsatisfied attack performance. In Figure 3, we showcase the triggers produced by Lagrange multipliers+SGD and LADDER to illustrate that LADDER can find more practical triggers than the conventional method. The dashed line demonstrates the expectation of trigger distribution which illustrates natural conflict between the two objectives, and the grey region includes the desired triggers. For example, the attacker aims to achieve a practical ASR ($> 99\%$) while maintaining an l_2-norm below 0.4 in CIFAR-10. Triggers obtained by the Lagrange multipliers+SGD method (marked in blue) are notably distant from the grey region, as they tend to lack either stealthiness or effectiveness. In contrast, most of the triggers generated by LADDER remain within the grey region, ensuring both stealthiness and attack effectiveness.

VI. Evolutionary Multi-objective Backdoor Attack

A. Problem Formulation

We formulate our backdoor attack as an MOP. The main task of solving the MOP is to search an optimal trigger, which is patched to images to create a poisoned dataset.

Frequency Trigger Injection Function. Formally, a frequency trigger $t = (\delta, \nu)$ where $\delta = \{\delta^0, \delta^1, \cdots, \delta^{n-1}\}$ is a series of magnitude of perturbations, ν=$\{\nu^0, \nu^1, \cdots, \nu^{n-1}\}$ describes the frequency bands to insert the correspondent perturbations on, and n is the number of manipulated frequency bands. We describe the trigger patching operation $\odot$ in Figure 4(a).

In order to inject our trigger t into an image x in the spectral domain, we obtain the spectrum of x via DCT ($\mathcal{D}(\cdot)$) and put the trigger optimized by LADDER in it. Finally, the poisoned spectrum is inverted to the spatial domain using IDCT ($\mathcal{D}^{-1}(\cdot)$), while we reset the label to an adversary-desired target. Our trigger injection function $\mathcal{T}$ and target label function η on a given sample (x, y) are formally defined as:

$$\begin{aligned} x' &= \mathcal{T}(x,t) \triangleq \mathcal{D}^{-1}(\mathcal{D}(x) \odot t), \\ y' &= \eta(y) \triangleq y_{tgt}. \end{aligned} \tag{6}$$

Dual-domain Stealthiness. We pioneer the consideration of stealthiness in both spatial and spectral domains highly desired in backdoor attacks, since the former ensures the poisoned image evades human inspection while the latter mitigates the anomaly of frequency disparities between benign and poisoned images. Given the widespread use of l_p-norm to evaluate the perturbation strength of the designed trigger [16, 56, 75], we adopt this measurement to calculate the spatial stealthiness between clean image x and poisoned image x' (obtained with trigger t and injection function $\mathcal{T}$):

$$\textit{Stealthiness}_{spatial} := \|\mathcal{T}(x,t) - x\|_p, \tag{7}$$

while the frequency stealthiness is reflected by the l_p-norm of trigger perturbations as:

$$\textit{Stealthiness}_{freq} := \|\delta\|_p. \tag{8}$$

This work selects $p = 2$, i.e., the l_2-norm as a measurement of trigger stealthiness for two reasons: (1) since the l_2-norm of disparity between the clean and poisoned images in dual domains is consistent, measuring the l_2-norm of trigger perturbation in the spectral domain can reflect dual-domain stealthiness; (2) we empirically demonstrate that common visibility metrics, such as PSNR, SSIM, LPIPS, cannot properly evaluate frequency stealthiness (see natural stealthiness in Section VII-B for details). We evaluate l_2-norm of triggers in

the spectral domain due to the benefit of injecting triggers in this domain (see low-frequency robustness below).

Robustness in the Low-frequency Spectrum. Low-frequency components show great resilience to image preprocessing operations such as lossy compression and low-pass filtering since these operations are all designed to destroy the mid- and high-frequency components first. Therefore, we constrain our manipulated frequency bands ν in the low-frequency domain $\mathscr{F}_{dom}$, i.e. $\nu_k \in \mathscr{F}_{dom}, \forall k \in \{0, 1, \ldots, |\nu| - 1\}$. Within $\mathscr{F}_{dom}$, we minimize the distance between the location of each frequency band of a trigger and the zero-frequency band as:

$$Robustness := || \sum_{i=0}^{n-1} (loc(\nu_i) - loc(\textit{min}(\mathscr{F}_{dom})))||_2, \quad (9)$$

where $\mathscr{F}_{dom}$ is the low-frequency domain, $\textit{min}(\mathscr{F}_{dom})$ is the zero-frequency band, and the function $loc(\cdot)$ is to find the vertical and horizontal index values for a given frequency band.

We provide a thorough analysis of the trade-off in terms of low-frequency regions and stealthiness. We also investigate the impact on attack effectiveness and robustness of our trigger design. Please see Appendix C for the details.

Multi-objective Backdoor Attacks Formulation. Current backdoor attacks, even when addressing multiple attack objectives, are typically formulated by linear combination with the Lagrange multipliers. As a result, excessive number of Lagrange coefficients are involved, complicating the parameter-tuning process. In contrast, we formulate the objectives simultaneously as an MOP and optimize a set of triggers (each trigger represents a unique trade-off among the objectives) without aggregating the objectives into an SOP. Considering the above objectives and constraints while maintaining the functionality (benign accuracy) of backdoored model, we formulate a multi-objective black-box backdoor attack as:

$$(\delta^*, \nu^*) = \underset{\delta,\nu}{\text{argmin}}\ O(\delta, \nu) = (O_1, O_2, O_3), \quad (10a)$$

$$\text{where } O_1(\delta, \nu) = \textstyle\sum_{(x,y)\in D_c \cup D_{bd}} \mathcal{L}(f_\theta^s(x), y), \quad (10b)$$

$$O_2(\delta, \nu) = \|\delta\|_{p=2}, \quad (10c)$$

$$O_3(\delta, \nu) = || \textstyle\sum_{i=0}^{n-1} (loc(\nu_i) - loc(\textit{min}(\mathscr{F}_{dom})))||_2, \quad (10d)$$

$$\text{s.t.} \quad |\delta_k| \le \epsilon,\ \forall k \in \{0, 1, \cdots, |\delta| - 1\}, \quad (10e)$$

$$\nu_k \in \mathscr{F}_{dom},\ \forall k \in \{0, 1, \cdots, |\nu| - 1\}, \quad (10f)$$

$$\text{Pref:} \quad O^* \to O_{pref}, \quad (10g)$$

The task of our attack is formulated in Equation (10a), which contains three objectives, O_1 of Equation (10b) that ensures a practical ACC and ASR, where f_θ^s is the surrogate model to evaluate trigger performance since the adversary cannot access a victim model, and the set of poisoned images D_{bd} is obtained with frequency trigger function in Equation (6); O_2 of Equation (10c) that ensures the dual-domain stealthiness and O_3 of Equation (10d) which seeks triggers robust against image preprocessing within $\mathscr{F}_{dom}$. We introduce two constraints, Constraint (10e) ensuring the magnitude of perturbation for each manipulated frequency band is within a reasonable range; and Constraint (10f) restricting trigger to design in the low-frequency region $\mathscr{F}_{dom}$. Finally, a preference-based selection (see Algorithm 3) is considered in Equation (10g) to reflect the preferred range of objective values.

Algorithm 1 LADDER Optimization via MOEA

Require: A subset of training data $\mathcal{D}_c$, Poison Ratio r, Total optimization generations *Gen*, Maximum frequency perturbation ϵ, Number of retrain epoch E_{re}, Surrogate model f_θ^s, Population size *P*

Ensure: Poisoned Dataset D_{bd} injected by $t^* = (\delta^*, \nu^*)$

Step 1: Initialization

T_{popu}: $\{(\delta_0, \nu_0), (\delta_1, \nu_1), \cdots, (\delta_{P-1}, \nu_{P-1})\} \leftarrow$ RandomInit()

Step 2: Evaluation

$\{O\}_{popu}$ = Eval($T_{popu}, f_\theta^s, \mathcal{D}_c, r$)

Step 3: Trigger Optimization

for gen in [0,1,$\cdots$,Gen-1] **do**

  T_{offsp}: $\{(\delta_0^{'}, \nu_0^{'}), \cdots, (\delta_{P-1}^{'}, \nu_{P-1}^{'})\} \leftarrow$Variation($T_{popu}$)

  $\{O\}_{offsp}$ = Eval($T_{offsp}, f_\theta^s, \mathcal{D}_c, r$)

  $T_{popu} \leftarrow$ rNDSort($T_{popu} \cup T_{offsp}, \{O\}_{popu} \cup \{O\}_{offsp}$)

Step 4: Trigger Selection & Data Preparation

$(\delta^*, \nu^*) \leftarrow$ SelectTrigger(T_{popu})

D_{bd} = Poison(D_c, r, (δ^*, ν^*))

return D_{bd}

B. Evolutionary Multi-objective Trigger optimization

Solving an MOP (with conflicting objectives) by using SGD+Lagrange multipliers often leads to suboptimal attack performance (see Section V). We introduce an MOEA to solve the problem. Our MOEA-based approach leverages crossover and mutation operators, avoiding the need to tune sensitive coefficients required by SGD+Lagrange multipliers. However, applying MOEA directly could produce impractical triggers (see those points outside the grey region, in Figure 5). To address this, we integrate MOEA with preference-based selection to prioritize practical triggers (those in the grey region).

Specifically, we leverage an MOEA to search the optimal trigger that can maximize performance of all the objectives in Equation (10a). The workflow of trigger optimization is described in Figure 4(b). MOEA estimates the performance of candidate triggers across objectives simultaneously in each iteration, without incurring the problems (in Figures 2 and 3). It initializes random triggers (Step 1 in Algorithm 1), iteratively optimizes them with variation (Figure 4 (b)), evaluates triggers' objective values (Algorithm 2) and selects non-dominated triggers by preference-based selection (Algorithm 3). We introduce the details of LADDER optimization in Algorithm 1.

Step 1: Initialization. MOEA initializes a population $popu = \{t_0, t_1, \cdots, t_{P-1}\} = \{(\delta_0, \nu_0), (\delta_1, \nu_1), \cdots, (\delta_{P-1}, \nu_{P-1})\}$ of triggers, where P is the population size. Besides, the triggers are generated under the constraints in Equations (10e) and (10f). Then, we evaluate the initialized triggers on the objectives.

Step 2: Trigger Evaluation. The idea of trigger evaluation is to calculate the objective values O_1, O_2 and O_3 in Equation (10b), 10c and 10d for each candidate trigger in lines

Algorithm 2 Eval: Evaluate Triggers in LADDER

Require: A set of triggers T, Surrogate model f_θ^s, A subset of training data $\mathcal{D}_c$, Poison Ratio r, Population size P
Ensure: The objective values $\{O\}$ of each trigger in T

1: **for** (δ_i, ν_i) **in** T **do**
2: $\quad\mathcal{D}_{bd} \leftarrow$Poison($D_c$, r, (δ_i, ν_i))
3: $\quad f_{\theta'}^s \leftarrow$Train($f_\theta^s, \mathcal{D}_{bd}$)
4: $\quad O_1^i = \sum_{(x,y)\in D_{bd}} \mathcal{L}(f_{\theta'}^s(x), y)$
5: $\quad O_2^i = \|\delta\|_2$
6: $\quad O_3^i = ||loc(\nu_i) - loc(min(\mathscr{F}_{dom}))||_2$
7: $\quad$Rollback $f_{\theta'}^s \leftarrow f_\theta^s$
8: $\{O\} = \{(O_1^0, O_2^0, O_3^0), (O_1^1, O_2^1, O_3^1), \cdots, (O_1^{P-1}, O_2^{P-1}, O_3^{P-1})\}$
9: **return** $\{O\}$

2 and 5 of Algorithm 1. The trigger evaluation is described in Algorithm 2. To evaluate the attack effectiveness for each trigger, we poison a subset of data with it and train the backdoor task (Equation (10b)) on a surrogate model. A surrogate model refers to a CNN model to approximate the victim model. We use this approach because the attacker has no knowledge about the victim model in the black-box setting. Also, evaluating triggers by training a model from scratch is computationally expensive. We hereby employ a pre-trained surrogate model on clean data, fine-tuning it through a limited number of retraining epochs, achieving evaluation efficiency. One may argue that the heterogeneous model structures between the surrogate and victim model may cause a bias of trigger performance in the evaluation process. To address this concern, we experimentally assess the trigger performance between various combinations of surrogate and victim model structures and demonstrate the high consistency among them (see Section VIII-A).

Step 3: Trigger Optimization. After initializing and evaluating the triggers, MOEA iteratively optimizes the triggers by applying variation to the triggers in the population to generate offsprings, evaluating their quality, and finally selecting well-performing triggers among all of them. Through this process, triggers gradually converge toward an optimal balance of stealthiness, attack effectiveness, and robustness.

Trigger variation and evaluation. The variation process is used to generate offspring triggers from population, which involves two procedures, simulated binary crossover (SBX) [13] and polynomial mutation (PM) [14]. The former randomly generates offspring triggers by exchanging a specific component (such as a perturbation or band) between two triggers from the population; the latter is to randomly alter the magnitude of frequency perturbations or shift the location of bands based on the perturbation sampled from a specific exponential distribution (see details in Figure 4(b)). The variation is repeatedly applied for each trigger in each iteration until the produced offsprings satisfy the restrictions in Equations (10e) and (10f). After that, we evaluate newly generated triggers in the same way as described in line 2 of Algorithm 1.

Next population formulation with rNDSort. In each iteration, after generating offsprings from parents and evaluating their performance on O_1, O_2 and O_3, we combine the population with offsprings and leverage the proposed rNDSort (see Algorithm 3) to pick up superior triggers survival into the next iteration while eliminating inferior triggers.

Algorithm 3 rNDSort: Preference-based NDSort

Require: Population size P, a set T=$\{t_0, t_1, \cdots, t_{2P-1}\}$ of triggers, the objective value set $\{O\}$ = $\{(O_1^0, O_2^0, O_3^0), (O_1^1, O_2^1, O_3^1), \cdots, (O_1^{2P-1}, O_2^{2P-1}, O_3^{2P-1})\}$, attacker preferred region of objective values O_{pref}
Ensure: The trigger set T_r ranked by the distance of their objective values to preference

1: $T_r \leftarrow \emptyset$, $list \leftarrow$[], $order \leftarrow 0$
2: **while** $|T_r| \leq P$ and $|T_r|$+NonDom(T, $\{O\}$) $\leq P$ **do**
3: $\quad T_r$.append(NonDom(T, $\{O\}$))
4: $\quad T'$=NonDom($T, \{O\}$)
5: $\quad \{O\}$=$\{O\}\backslash\{O\}^{T'}$, T=$T\backslash T'$
6: **for** i **in** $\{0, 1, \cdots, |T|\}$ **do**
7: $\quad d$ = Euc($O[i]$, O_{pref})
8: $\quad list$.append($<d$, $T[i]>$) $\quad\triangleright < \cdot, \cdot >$ is a pair
9: $list \leftarrow Sort_{ascend}(list)$ by d
10: **while** $|T_r| < P$ **do**
11: $\quad T_r$.append($list[order$++].SecondElem)
12: **return** T_r

rNDSort includes two components, NDSort and *preference-based selection*, which drives triggers to converge toward the attacker-desired region and maintain a stable number of triggers in the population per iteration. We first introduce the dominance relationship for non-dominated sort. Given two triggers t_1 and t_2 along with their objective values O^1=$\{O_1^1, O_2^1, O_3^1\}$ and O^2=$\{O_1^2, O_2^2, O_3^2\}$ (recall smaller objective value leads to a better trigger), we say t_1 *dominates* t_2, denotes as $t_1 \prec t_2$ iff. $\forall k \in [1,3], O_k^1 \leq O_k^2$ and $\exists k \in [1,3]$ s.t. $O_k^1 < O_k^2$. In this case, t_1 is a *non-dominated* trigger among $\{t_1, t_2\}$. With the help of the dominance relationship, the non-dominated sort repeatedly moves non-dominated triggers from a trigger set T to a new set T_r, until adding non-dominated triggers in T_r results in $|T_r| > P$. Finally, the remaining triggers in T with the largest k-nearest sparsity [12] concerning objective values are selected to fill in T_r until $|T_r| = P$. This step ensures that triggers are searched along the entire objective space. However, impractical triggers (see those points which are out of the grey region in Figure 5) may be still acquired, as triggers searched by NDSort are non-dominated to attacker-desired triggers. This means they are considered of equal quality from the MOO perspective, even though they may not be practical.

To alleviate locating impractical triggers caused by NDSort, we fill in T_r to the size P with preference-based selection. Specifically, we calculate the Euclidean distance of remaining triggers in T to the attacker-desired region in terms of objective values and select triggers with the smallest distance until $|T_r| = P$.

To validate the efficacy of rNDSort, we compare the triggers

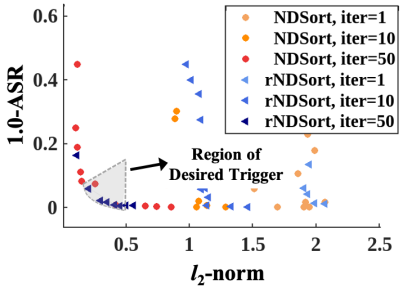


Fig. 5: Comparison of triggers on MOEA with/without preference-based selection in CIFAR-10 on VGG11. Compared to NDSort, rNDSort pulls LADDER triggers closer to the attacker-desired region.

obtained by NDSort and rNDSort of 1^{st}, 10^{th} and 50^{th} iterations and visualize their objective values in Figure 5. We can observe that after 50 iterations, the triggers obtained by rNDSort are mostly located within the attacker-desired region (marked in grey). In contrast, triggers obtained by NDSort span a wider range, including impractical ones.

Step 4: Trigger Selection and Data Preparation. After the trigger optimization, we obtain a set of practical triggers (see those points in or close to the grey region in Figure 5) and choose the best trade-off under our attack scenario among them. Specifically, we choose the most practical trigger from the population based on whose objective values are closest to the best values for each objective. Finally, we release a poisoned dataset injected by the trigger.

VII. Experiments

A. Experimental Setup

Experimental Environment and Settings. Our LADDER is implemented [43] on Python, PyTorch [53] and Ubuntu. All the experiments are conducted on a workstation with Ryzen 9 7950X, 2×32GB DDR5 RAM, and NVIDIA GeForce RTX 4090. For the default training, we learn the classifiers by SGD optimizer with the initial learning rate of 0.01 and a decay of 0.1 per 50 epochs. We set the batch size to 64 and the total number of epochs to 200 for all the datasets to train surrogate and victim models. When evaluating triggers on the surrogate models, the number of retraining epochs is set to 20. For the default attack setting, we search triggers in low-frequency regions. Following Sharma et al. [59], we use around 18.3% of the whole frequency spectrum on the top-left region to search the low-frequency trigger. Meanwhile, we manipulate 3 frequency bands per channel for our attack in all the datasets. For a fair comparison, the poison ratio and target label are set to 5% and 7, unless otherwise specified. For the default MOEA setting, we set the population size to 10, the optimization iterations to 20. We set the O_{pref} in Equation (10g) as: 0.9 for O_1 and 0.4 for O_2; O_3 is 8 for images of size 32× 32 and 12 for images of size 64× 64.

Datasets and Models. We evaluate LADDER on five benchmark tasks including digit recognition on SVHN [49], object classification on CIFAR-10 [35], real objects on Tiny-ImageNet [37], traffic sign recognition on GTSRB [29] and face attribute recognition on CelebA [47]. For CelebA, we follow [51, 57] to select the top three most balanced attributes including Heavy Makeup, Mouth Slightly Open, and Smiling. Then, we concatenate them to create an eight-label classification task. We evaluate LADDER on both small- and large-scale datasets to confirm its scalability across various image and dataset sizes. The five datasets chosen for this paper span a remarkably broad scope of typical real-world scenarios, underscoring the practicality of LADDER. Following [7, 17, 51, 63, 65], we consider various network architectures for the image classifier. Specifically, we employ a classic CNN model [17, 51] for SVHN, PreAct-ResNet18 [27] for CIFAR-10 and GTSRB, as well as ResNet18 [27] for Tiny-ImageNet and CelebA. In contrast to victim models, we choose surrogate models from a series of VGGs [61], whose structures are heterogeneous against ResNet models. For example, we utilize VGG11 for CIFAR-10 and GTSRB, VGG16 for SVHN and CelebA, as well as VGG19 for Tiny-ImageNet. It is important to emphasize our intentional use of heterogeneous structures between victim and surrogate models. This approach effectively demonstrates that the model mismatch between the victim and surrogate models does not hinder the efficacy and practicality of our attack.

B. Attack Performance

We compare LADDER with popular spatial attacks, such as BadNets [24], ReFool [46], SIG [4], WaNet [51] and Narcissus [70] as well as frequency attacks such as FIBA [20], FTrojan [68] and DUBA [23] as baseline methods to showcase the attack performance in: attack effectiveness, natural (spatial) and spectral (frequency) stealthiness. Note that several white-box attacks [15, 50, 75, 76], although achieving practical effectiveness, require access and manipulation of victim models. They are not included in the experiments.

Attack Effectiveness. We evaluate the effectiveness of 8 attacks against 5 datasets via ACC and ASR. Based on the results given in Table II, LADDER achieves ASRs exceeding 99% on all poisoned CNN models. Meanwhile, its drop of ACCs after the backdoor attack is limited to only 0.23% on average, while the compared attacks yield larger ACC drops. This confirms that LADDER delivers practical attack performance under various attack tasks. Recall that we consider attack effectiveness as one of the objectives when formulating the multi-objective attack problem, ensuring that the triggers searched by LADDER are oriented towards maximizing effectiveness. We also note that heterogeneous network structure settings between surrogate and victim models do not affect the attack effectiveness of LADDER.

Natural (Spatial) Stealthiness. Natural stealthiness is vital for backdoor attacks, guaranteeing that poisoned images remain imperceptible to human inspection. We quantitatively compare the differences between poisoned and clean images against four popular visual stealthiness measurements, including l_2-norm, PSNR, SSIM, and LPIPS. All metric values are av-

TABLE II: Attack performance measured by ACC (%) and ASR (%) for 8 attacks against 5 datasets. The number in the brackets indicates the differences between clean ACC and the correspondent ACC on the backdoored model. Our method achieves comparable or superior performance on ACCs/ASRs compared to other attacks, with the exception of the SVHN dataset, where our ACC/ASR are only 0.48% and 0.21% lower than the best results, respectively.

Attack	SVHN		GTSRB		CIFAR-10		Tiny-ImageNet		CelebA	
	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
Clean	92.81	-	98.55	-	93.14	-	54.60	-	79.20	-
BadNets [24]	**92.67 (0.14)**	99.14	97.91 (0.64)	96.67	92.05 (1.09)	98.24	51.90 (2.70)	97.82	76.54 (2.66)	99.35
SIG [4]	92.45 (0.36)	99.87	97.90 (0.65)	99.87	92.14 (1.00)	99.98	51.98 (2.62)	99.49	77.90 (1.30)	99.85
Refool [46]	92.24 (0.57)	99.31	97.94 (0.61)	98.51	91.09 (2.05)	97.03	48.37 (6.23)	97.32	77.53 (1.67)	98.09
WaNet [51]	92.33 (0.48)	99.17	98.19 (0.36)	99.83	92.31 (0.83)	99.94	52.85 (1.75)	99.16	77.99 (1.21)	99.33
FTrojan [68]	92.63 (0.18)	**99.98**	96.63 (1.92)	99.25	92.53 (0.61)	99.82	53.41 (1.19)	99.38	76.63 (2.87)	99.20
FIBA [20]	91.10 (1.71)	96.91	96.73 (1.82)	98.88	91.13 (2.01)	97.60	51.11 (3.49)	92.14	75.90 (3.30)	99.16
DUBA [23]	91.23 (1.58)	99.79	96.90 (1.65)	98.32	91.97 (1.17)	**99.99**	52.74 (1.86)	**99.99**	77.30 (1.90)	**99.99**
Narcissus-D [70]⋆	91.94 (0.87)	99.97	97.47 (1.08)	**99.99**	92.17 (0.97)	**99.99**	54.17 (0.43)	**99.99**	77.85 (1.35)	**99.99**
Ours	92.19 (0.62)	99.77	**98.37** (0.18)	99.93	**92.82** (0.32)	**99.99**	**54.20** (0.40)	99.54	**79.57**(0.37↑)	99.90

⋆ Narcissus is a clean-label backdoor attack, which does not align with the dirty-label attack framework of this paper. Therefore, we extend it to a dirty-label attack, denoted as Narcissus-D, where the labels of poisoned samples are assigned the target label during data poisoning.

TABLE III: Natural stealthiness (PSNR ↑, SSIM ↑, LPIPS ↓) as well as l_2-norm ↓ of trigger pattern. Across 4 metrics and 5 datasets, LADDER consistently demonstrates superior stealthiness compared to 8 attacks, with the only minor exception where LADDER has a 0.081 difference on l_2-norm on SVHN and a 0.0007 gap on LPIPS on Tiny-ImageNet.

Attacks	SVHN				GTSRB				CIFAR-10				Tiny-ImageNet				CelebA			
	l_2	PSNR	SSIM	LPIPS	l_2	PSNR	SSIM	LPIPS	l_2	PSNR	SSIM	LPIPS	l_2	PSNR	SSIM	LPIPS	l_2	PSNR	SSIM	LPIPS
Clean	0.0000	Inf	1.0000	0.0000	0.0000	Inf	1.0000	0.0000	0.0000	Inf	1.0000	0.0000	0.0000	Inf	1.0000	0.0000	0.0000	Inf	1.0000	0.0000
BadNets [24]	2.9363	27.49	0.9763	0.0187	3.8479	27.18	0.9754	0.0059	2.7358	36.67	0.9763	0.0012	2.9737	36.35	0.9913	0.0006	3.2871	32.50	0.9951	0.0005
SIG [4]	3.0525	25.18	0.7490	0.0706	3.0113	25.32	0.7313	0.0766	3.0259	25.26	0.8533	0.0289	6.0205	25.36	0.8504	0.0631	5.9627	25.38	0.7949	0.0359
Refool [46]	4.8254	21.61	0.8511	0.0456	5.0275	20.57	0.7418	0.3097	5.9169	18.37	0.6542	0.0697	6.4901	20.42	0.8564	0.4574	7.0494	23.72	0.8359	0.2134
WaNet [51]	**0.1969**	37.72	0.9905	0.0016	0.4280	30.11	0.9669	0.0584	1.9397	19.30	0.8854	0.0090	1.4926	29.59	0.9359	0.0360	0.7880	30.42	0.9175	0.0530
FTrojan [68]	0.4866	41.13	0.9896	0.0002	0.4874	41.11	0.9885	0.0007	0.4850	41.16	0.9946	0.0006	0.8553	42.28	0.9931	**0.0003**	0.8568	42.25	0.9904	0.0003
FIBA [20]	1.9250	29.67	0.9782	0.0044	1.8693	29.74	0.9589	0.0083	1.8437	29.69	0.9858	0.0024	3.7459	29.39	0.9755	0.0080	4.0548	29.25	0.9592	0.0057
DUBA [23]	0.9574	35.71	0.9721	0.0028	1.5812	31.82	0.9376	0.0034	1.9642	29.35	0.9415	0.0027	5.2490	26.83	0.8815	0.0256	3.3136	30.51	0.9191	0.0210
Narcissus-D [70]	6.6200	18.45	0.5952	0.1704	5.5698	19.94	0.5795	0.0925	6.5335	18.56	0.7137	0.0324	3.3335	30.44	0.9328	0.0170	4.5943	27.65	0.9278	0.0637
Ours	0.2781	**45.99**	**0.9973**	**0.0003**	**0.3406**	**44.23**	**0.9943**	**0.0002**	**0.3183**	**44.81**	**0.9976**	**0.0001**	**0.6132**	**45.14**	**0.9976**	0.0010	**0.4132**	**48.57**	**0.9974**	**0.0002**

eraged over 1,000 randomly selected samples from the test dataset. In Table III, we list the ideal metric values on clean images under 5 datasets, then show metric values on the poisoned samples under various attacks. SSIM cannot precisely capture minor differences of trigger perturbation (e.g., in CIFAR-10, a 5.79× difference for l_2-norm between 0.3183 and 1.8437 results in only a 1.1% difference for SSIM); trigger perturbation is inconsistent with PSNR results (e.g., in CelebA, an increase of l_2-norm from 0.8568 to 0.7880 leads to a decline of PSNR from 42.25 to 30.42). We can observe that LADDER achieves superior spatial stealthiness in 18 out of 20 cases, underscoring its significantly enhanced natural stealthiness compared to others. Note frequency attacks such as FIBA and FTrojan still show better stealthiness than those spatial attacks. LADDER achieves such a practical stealthiness because (1) the perturbations produced by LADDER are minimal due to our trigger stealthiness objective; (2) since the LADDER trigger is inserted in the spectral domain, the intensity of the trigger pattern spreads across the entire spatial domain; and (3) we pose the perturbation in the low-frequency domain where large magnitude of frequency information exists, which provides capacity to hide small perturbations. LADDER triggers induce less perturbation on each pixel in the spatial domain.

To visually confirm the superior trigger stealthiness achieved by LADDER in Table III, we plot the clean and poisoned images in the first row in Figure 6. We see that the image poisoned by LADDER is undetectable, so that its trigger achieves equal and superior stealthiness to other frequency and spatial backdoor attacks. More poisoned samples produced by LADDER across 5 datasets (see Figure 10 in Appendix) can further confirm its practical and natural stealthiness.

Spectral Stealthiness. This work represents the first instance to consider stealthiness in dual domains. To confirm the trigger stealthiness in the spectral domain, we visualize the residual map of the frequency disparities between clean and poisoned images in the second row in Figure 6. The frequency disparity is derived by subtracting the spectrum of the poisoned sample from its clean counterpart. Bright pixels emerge in the residual map of the compared attacks, indicating a notable frequency disparity between clean and poisoned samples. Different from that, our residual map is almost black, where disparity exists yet is not visible. Through the results, we can draw a solid conclusion that LADDER, compared to those eight black-box attacks, achieves a remarkably better spectral stealthiness. Note along with the natural stealthiness, LADDER obtains a solid *dual-domain stealthiness*.

Figure 6 clearly emphasizes the necessity of designing triggers in the **dual domains** in order to avoid anomaly in both domains. Taking two frequency-domain backdoor attacks FTrojan and FIBA for example, their triggers achieve almost perfect visual stealthiness in the spatial domain; but they cannot eliminate the anomaly in the spectrum domain.

TABLE IV: Attack robustness (%) of various triggers against preprocessing-based defenses. To illustrate the robustness of our low-frequency trigger, we introduce various variants of LADDER for comparison, named LADDER-Mid, LADDER-High and LADDER-Full, which search the triggers across different regions in spectrum with the same attack settings. Our low-frequency trigger design achieves an average ASR of 90.23%, which is 50.09% higher than the ASR averaged by five popular attacks and three variations of LADDER targeting different spectral regions. This demonstrates our superior robustness against preprocessing.

Attacks →	BadNets [24]		FTrojan [68]		FIBA [20]		DUBA [23]		Narcissus-D [70]		LADDER-Mid		LADDER-High		LADDER-Full		LADDER-Low	
Methods ↓	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
Original	92.02	98.78	92.53	99.82	91.13	97.60	91.97	99.99	92.17	99.99	91.51	99.49	92.33	99.99	92.54	99.94	92.82	99.95
Gaussian Filter ($w = (3, 3)$)	66.17	15.11	67.80	6.47	61.99	94.48	65.30	6.31	65.19	4.42	67.45	11.79	67.04	5.92	64.29	6.32	66.41	95.17
Gaussian Filter ($w = (5, 5)$)	39.81	6.88	45.03	3.25	46.00	93.71	44.37	3.44	45.21	0.61	42.76	3.18	42.90	2.20	40.12	2.52	61.21	94.33
Wiener Filter ($w = (3, 3)$)	69.53	88.11	69.11	10.54	58.72	95.17	65.10	53.42	64.27	4.85	65.81	9.82	67.87	6.23	63.95	8.56	67.11	94.83
Wiener Filter ($w = (5, 5)$)	52.18	96.43	49.20	5.28	37.67	94.79	45.22	92.40	45.01	5.87	44.92	3.86	50.18	2.24	43.78	4.49	47.15	92.65
Brightness (1.1)	81.14	97.27	82.86	74.83	71.39	44.19	69.75	95.15	75.18	84.64	71.64	9.08	77.12	10.74	76.57	8.81	80.36	91.94
Brightness (1.5)	82.08	91.76	79.24	75.52	70.43	38.67	67.07	99.46	70.28	83.71	73.54	9.83	71.44	13.37	78.64	8.77	77.15	83.32
JPEG (quality = 90%)	88.98	97.85	89.22	9.36	67.06	82.18	88.34	11.18	89.15	89.33	89.56	9.72	89.75	9.15	90.35	9.57	91.72	89.86
JPEG (quality = 50%)	78.84	92.59	79.66	8.58	70.43	38.67	73.83	8.80	75.42	70.08	80.39	9.10	79.21	8.40	80.20	6.45	76.09	79.79
Average ASR		73.25		32.63		72.73		46.27		42.94		18.43		17.58		17.27		**90.23**

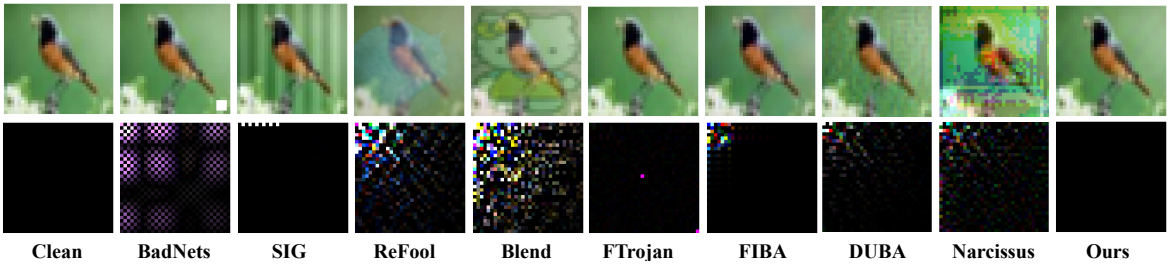


Fig. 6: Comparing poisoned and clean images in the frequency domain reveals disparities caused by backdoor attacks on CIFAR-10 dataset. The top row displays clean and poisoned images, while the bottom row illustrates the spectrum disparity of each poisoned image compared to the clean spectrum. The disparity of clean image is black since disparity does not exist.

C. Attack Performance Against Defenses

We evaluate the attack effectiveness against mainstream detection defenses, such as the network inspection [58] and STRIP [22]; and defensive-based defenses, such as Neural Cleanse [66] and Fine-pruning [45]. We further test the effectiveness of LADDER against the SOTA defenses including ASD [21], CBD [74] and DBD [30] in Appendix A. We also confirm the dual-domain stealthiness of LADDER via frequency artifacts inspection [71] on poisoned images. Besides, we evaluate our attack under preprocessing-based operations as in works [32, 68] to yield a solid confirmation of robustness.

Against Network Inspection. Grad-CAM [58] visualizes the critical regions of an input image that can mostly activate the prediction, which helps understand the features a CNN model learned. It has been reported [20, 50, 51] that a backdoored CNN model tends to show an attention shift on poisoned images compare to clean ones. We showcase the network attention map on benign and victim models against CelebA, CIFAR-10, GTSRB, Tiny-ImageNet, and SVHN datasets, in Figure 7. According to the results, we see that the attention of the model on benign and poisoned images almost remains the same, indicating LADDER does not cause severe attention anomaly. We insert our triggers in the low-frequency region where abundant semantic information exists. Thus, the trigger pattern is obfuscated within the original semantics, ensuring that LADDER does not introduce any anomalous regions.

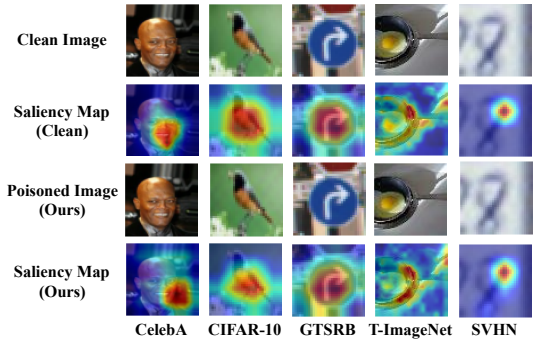
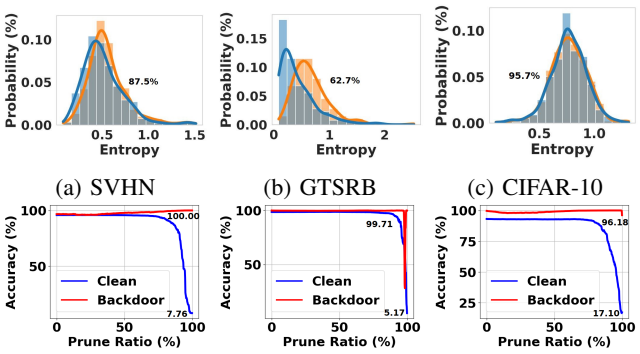


Fig. 7: Visualization of network attention via Grad-CAM from clean and poisoned images. The region masked by red indicates a strong contribution toward model prediction.

Against STRIP. STRIP is a well-established backdoor defense strategy based on the assumption that poisoned data in a backdoored model consistently produces the target label and cannot be easily altered. Under this assumption, STRIP poisons samples by assessing the entropy of classification, achieved by overlaying randomly selected clean images onto the test samples. It expects the resulting entropy distribution to resemble that of the entropy distribution obtained with only clean images, thereby identifying and mitigating poisoned samples. We test the images poisoned by LADDER against STRIP, and visualize the entropy distribution among samples. The results are in Figure 8 (a)-(c), in which blue and orange



(a) SVHN (b) GTSRB (c) CIFAR-10

(d) SVHN (e) GTSRB (f) CIFAR-10

Fig. 8: (a)-(c): The entropy distribution obtained with model poisoned by LADDER against STRIP. The distributions marked in blue and orange are obtained with benign and poisoned (by LADDER) testing data. Each curve is fitted to its respective distribution, with the annotated numbers representing the proportion of the overlapped areas relative to the backdoor distributions; (d)-(f): The ASR and ACC of LADDER against Fine-pruning after the correspondent percentage of pruned neurons. The final ACCs and ASRs are annotated after the defense.

bars indicate the (normalized) probability of the correspondent entropy on clean and poisoned images respectively, while the curves are the respective fitted distributions. The overlapped areas of distributions reflect the difficulty of poisoned samples being detected. We observe that LADDER achieves almost perfect entropy probability distributions as clean samples on SVHN, GTSRB and CIFAR-10 since their distributions are almost overlapped. This is so because superimposing random images in the spatial domain destroy low-frequency components (containing LADDER trigger pattern) of our poisoned images. Therefore, the predictions of superimposed images also undergo significant changes, which is similar to the clean cases. In conclusion, STRIP cannot effectively identify the difference between clean and poisoned samples by LADDER.

Against Neural Cleanse (NC). The insight behind NC is that any samples with a backdoor trigger result in a misclassification to the target label in the victim model. NC reverses the possible triggers to detect backdoors on an unverified model by checking if the reversed trigger can possibly cause misclassification on the test dataset. It determines if a model has been compromised with an anomaly index. The index exceeding 2 indicates a high-risk level of model poisoning. We test LADDER against NC on five datasets. The results are in Figure 9, in which the x-axis indicates different datasets and the y-axis records the anomaly index produced by NC. The blue and orange color bars represent the anomaly index with clean and poisoned data, respectively, on the poisoned model with LADDER. We see that the results are within the threshold of 2.0, showcasing LADDER successfully evades NC. Recall that NC focuses on small and fixed backdoor patches. Triggers produced by LADDER in the low-frequency

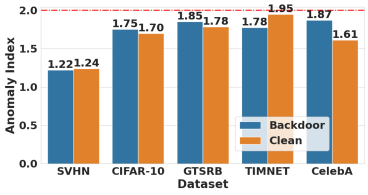


Fig. 9: The results of LADDER under Neural Cleanse on different datasets. The dotted line marks the threshold, below which a model is regarded as clean.

region of spectrum spread across the entire spatial domain, rendering the perturbation visually imperceptible due to the small number of magnitudes in the triggers. As a result, NC cannot discern the trigger pattern, leading to the failure of detecting poisoned samples.

Against Fine-pruning. Fine-pruning, which is an effective and widely adopted backdoor defense, iteratively removes neurons to eliminate potential backdoor triggers inserted during training, thus fortifying the model against backdoor attacks without severely compromising its primary performance on clean data. The results are given in Figure 8 (d)-(f), in which neurons are iteratively pruned from 0 to 100%, as indicated in x-axis; red and blue curves indicate the ASR and ACC for backdoor and benign data respectively. We see that with the increase of pruning ratio, the benign accuracy drops more quickly than that of backdoor. Till the end of the pruning process, the ACC falls to almost zero whereas ASR is still valid, making backdoor mitigation by fine-pruning impossible. Thus, fine-pruning is ineffective to mitigate the attack effectiveness of LADDER.

Against Image Preprocessing. It has been demonstrated in [25, 32] that image preprocessing is effective in filtering trigger patterns and can mitigate backdoor attacks. To demonstrate the robustness of our attack against preprocessing-based defenses, we apply typical preprocessing methods [31] such as the brightness adjustment, Gaussian filter, Wiener filter and JPEG compression on the poisoned images before inference, and demonstrate the robustness of LADDER quantitatively in Table IV.

In Table IV, we record the ACC and ASR achieved by BadNets, FTrojan, FIBA, DUBA, Narcissus as well as LADDER on low, mid, high and full frequency regions after image preprocessing. We clearly see that the average ASR achieved by LADDER in low-frequency region is 90.23%, significantly outperforming others. This is because most preprocessing operations focus on either spatial domain (e.g., brightness adjustment) or high-frequency artifacts (such as filter and compression), which does not destroy LADDER's trigger pattern in low-frequency region. We can expect to achieve a similar performance (by LADDER) on other similar preprocessing-based defenses.

Against Other SOTA Defenses. We analyse the attack effectiveness of LADDER and other attacks against three recent SOTA white-box defenses (see Appendix A).

Adaptive Defense under Black-box Setting. To mitigate backdoor attacks that take spectral stealthiness into account, we tentatively propose frequency domain anomaly detection that distinguishes poisoned images by the parameter p of the distribution $1/r^p$ concerning magnitudes r in the averaged spectrum. Please see Appendix B for methodology and results.

VIII. Ablation Study

A. On the Transferability of Surrogate Model

We assume the role of a malicious data provider who has access to a dataset but lacks access to the target model. To evaluate the quality of the trigger during its optimization, we introduce a surrogate model for injecting the trigger and assessing attack effectiveness. We test LADDER's attack transferability on CIFAR-10 dataset across a range of typical surrogate and victim (target) model architectures including VGG16 [61], ViT [18], ResNet18 [27] and Google-Net [62]. We search for triggers based on different surrogate model architectures, and choose the optimal trigger on surrogate model among those triggers. Finally, we inject the trigger obtained from each surrogate model into the corresponding target models and record the ACC and ASR after 50 epochs.

In Table VI, we verify that LADDER is transferable between heterogeneous model architectures in practical attack scenarios. The mismatch between surrogate and victim models does not degrade the ACC, while a high ASR is maintained across all mismatched cases. Additionally, using the same surrogate and victim models does not always guarantee the best performance, as seen with ResNet18 and GoogleNet. We conclude that LADDER's effectiveness is not sensitive to the specific combination of surrogate and target models. Therefore, optimal ASR can be achieved without requiring a specific model structure pairing between surrogate and target models.

Transferability discussions. Recall that this work (in the context of CNNs and computer vision) addresses backdoor attacks with three objectives: attack effectiveness, stealthiness, and robustness. Among them, stealthiness (Equation (7)) is model-independent and can be directly calculated based on trigger perturbation. Similarly, robustness (Equation (4)) depends on how well the trigger perturbation retains its effectiveness after image preprocessing. This perturbation depends on the specific preprocessing and the design of the trigger itself—independent of model architectures—and its effectiveness is largely determined by the norm of the perturbation. Attack effectiveness, on the other hand, is guaranteed by training a model with the trigger injected into the data. This objective is influenced primarily by: the number of feature vectors, the poison ratio, and the norm of the trigger perturbation [38]. For example, using the CIFAR-10 dataset, with ResNet18 as the surrogate model and GoogLeNet as the target model, both models leverage the same dataset, ensuring an identical number of feature vectors. The optimal trigger generated via optimization is used directly in the actual attack phase, which maintains the same trigger perturbation. Finally, both models use the same poison ratio to create the poisoned dataset. Controlling these factors enables us to yield consistent attack effectiveness across models, thus providing transferability.

TABLE V: Effectiveness (ASR), stealthiness and robustness of variants compared to the original version of LADDER on CIFAR-10.

Metrics \ Trigger	Spatial	Ste+Eff	Rob+Eff	Ste+Rob	Eff	Ori
Effectiveness (%)	99.99	99.99	99.85	94.83	99.88	99.99
Stealthiness (l_2)	0.6916	0.4007	3.5095	0.2020	2.9437	0.3183
Robustness (%)	35.04	24.94	93.84	64.62	11.42	82.52

TABLE VI: Transferability of LADDER across different surrogate and target model architectures (ASR/ACC)(%). The ASRs close to 100 indicate a tiny discrepancy in backdoor performance between the surrogate and victim model.

Sur \ Tar	VGG16 [61]	ResNet18 [27]	Google-Net [62]	ViT [18]
VGG16	99.86 / 91.87	99.97 / 93.54	99.82 / 93.41	99.48 / 83.34
ResNet18	99.07 / 91.51	99.42 / 92.74	99.62 / 93.40	99.93 / 82.69
Google-Net	99.51 / 91.88	99.58 / 92.91	99.17 / 93.78	99.61 / 82.02
ViT	99.52 / 91.10	99.88 / 92.75	99.40 / 93.77	99.66 / 82.74

B. Ablation Study of Trigger Design

The trigger design of LADDER captures stealthiness, robustness, and effectiveness in the spectral domain. In Table V, we showcase the results of leveraging a subset of attack objectives and implementing a spatial variant. For example, Rob+Eff achieves superior robustness (93.84%), but it falls short in providing practical stealthiness (l_2 = 3.5095). Also, spatial trigger obtains 35.04% of robustness although taking 2.2× more perturbation magnitude than the original LADDER. This study confirms that LADDER can provide the most practical trigger considering all the objectives in the spectral domain.

C. Scalability Analysis

We investigate the LADDER's scalability in terms of time and resource usage, including CPU/GPU utilization (%), RAM/GPU memory (GB) across 5 datasets, 5 models and various objectives. Table VII showcases the time and resource usage across datasets on ResNet18, in which small datasets such as SVHN and CIFAR-10 require around 50% of CPU and GPU utilization while large datasets such as Tiny-ImageNet requires less CPU but more GPU usage. Also, the time cost across datasets is positively correlated with the dataset size. We present the time and resource usage of LADDER across models in Table VIII. The results show that CPU utilization increases while GPU utilization decreases as the number of model parameters grows (as shown from left to right in Table VIII, where the number of model parameters increases). Note GoogLeNet contains Inception modules with a large number of convolution filters, which slows down the training speed and requires more GPU memory. The time and resource usage across objectives is in Table IX, where we run LADDER with different objectives (effectiveness, robustness and stealthiness) on CIFAR-10 and ResNet18. Our results

TABLE VII: Time and resource usage across various datasets on ResNet18.

Resource \ Dataset	SVHN	CIFAR-10	GTSRB	T-ImageNet	CelebA
CPU util. (%)	50.1	50.2	36.2	24.9	28.2
GPU util. (%)	59.3	44.7	54.8	77.9	48.1
RAM (GB)	6.17	6.05	7.55	12.19	6.02
GPU Mem (GB)	4.07	4.02	4.06	7.65	7.67
Time (s)	329	421	576	1970	2080

TABLE VIII: Time and resource usage across different model architectures on CIFAR-10.

Model \ Resource	GoogLeNet	ResNet18	ViT	VGG11	VGG16
CPU util. (%)	38.1	50.2	50.6	51.0	50.3
GPU util. (%)	82.1	44.7	63.5	20.7	33.7
RAM (GB)	6.07	6.05	4.48	6.04	6.05
GPU Mem (GB)	13.43	4.02	5.01	3.48	3.76
Time (s)	1197	421	537	411	431

show that objectives influence resource consumption. Specifically, evaluating effectiveness involves additional training on surrogate models and this consumes a significant proportion of resource usage (see Eff results in Table IX). In contrast, robustness and stealthiness only yield constant complexity (see Ste+Rob results in Table IX, Equations (10c) and (10d)).

IX. Ethical Consideration

This work exposes the vulnerability of deep learning models to practical, stealthy and robust backdoors and can inspire follow-up studies that enhance the security of deep learning. In this sense, this work has a positive impact on the future research of AI safety. In the following, we discuss the intellectual property, intended usage, potential misuse, risk control and human subject.

Intellectual property. All comparative attacks and defenses, models, datasets and implementation libraries are open-source. We believe that the datasets are well-desensitized. We strictly comply with all applicable licenses for academic use.

Intended Usage. We expose the vulnerability of current deep learning models to practical stealthy and robust backdoor triggers. We encourage researchers to use our findings to assess the security of their models and hope that this work will inspire development of robustness against backdoor attacks.

Potential Misuse. This work could be exploited to produce stealthy and robust poisoned datasets for real-world applications, which potentially leads to more covertly malicious models. To maintain safety of deep learning models, we propose an adaptive defense in Appendix B.

Risk Control. To further mitigate potential risks, we will release the code [43] used in this work. By doing so, we believe that transparency will reduce the risks related to our work, encourage responsible use and foster further advancement of secure techniques for deep learning models.

Human Subject. We do not involve any human subjects in this work. Instead, we rely solely on mathematical models and metrics to simulate human visual inspection, thereby eliminating the need for human participation.

TABLE IX: Time and resource usage across various objectives on CIFAR-10 and ResNet18.

Resource \ Objectives	Eff+Ste+Rob	Eff+Ste	Eff+Rob	Ste+Rob	Eff
CPU util. (%)	50.2	50.4	50.7	6.9	50.4
GPU util. (%)	44.7	49.2	49.5	0	49.1
RAM (GB)	6.06	6.05	6.05	1.79	6.04
GPU Mem (GB)	4.02	4.06	4.07	0	4.03
Time (sec.)	421	421	426	10.18	428

X. Limitations

The effectiveness of LADDER against white-box defenses is naturally reduced. Recall that this work designs triggers under a black-box attack scenario. Unlike white-box attacks which can directly manipulate model parameters, black-box variants could naturally not perform well against some specific white-box backdoor defenses.

XI. Conclusion

This work introduces LADDER, a multi-objective backdoor attack that effectively searches for backdoor triggers via an evolutionary algorithm. It achieves effectiveness, dual-domain stealthiness, and robustness, instilling confidence in its capabilities. First, we observe the conflict between trigger stealthiness and attack performance and find the sensitivity of solving multiple attack goals with the Lagrange multipliers and SGD. Then, we improve the trigger robustness by designing triggers in the low-frequency domain while extending the trigger stealthiness to the dual domains. We also design a new multi-objective backdoor attack problem to capture the objectives simultaneously. Finally, we leverage the evolutionary algorithm to solve the proposed problem in a black-box setting without tuning Lagrange coefficients. Experimental results confirm the practical performance of LADDER.

Acknowledgment

This work was supported by the EU Horizon Europe Research and Innovation Program under grant agreements 101073920 (TENSOR), 101070052 (TANGO), 101070627 (REWIRE), 101092912 (MLSysOps), and 101168562 (SafeHorizon).

References

[1] G. Abad, O. Ersoy, S. Picek, and A. Urbieta, "Sneaky Spikes: Uncovering Stealthy Backdoor Attacks in Spiking Neural Networks with Neuromorphic Data," in *Network and Distributed System Security Symposium*, 2024.

[2] N. Ahmed, T. Natarajan, and K. Rao, "Discrete Cosine Transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, 1974.

[3] S.-i. Amari, "Backpropagation and Stochastic Gradient Descent Method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.

[4] M. Barni, K. Kallas, and B. Tondi, "A new Backdoor Attack in CNNs by Training Set Corruption without Label Poisoning," in *IEEE International Conference on Image Processing*, 2019, pp. 101–105.

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, and J. Zhao, "End-to-End Learning for Self-Driving Cars," *arXiv preprint arXiv:1604.07316*, 2016.

[6] G. J. Burton and I. R. Moorhead, "Color and Spatial Structure in Natural Scenes," *Applied Optics*, vol. 26, no. 1, pp. 157–170, 1987.

[7] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, "Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering," *arXiv preprint arXiv:1811.03728*, 2018.

[8] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, "DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2019, pp. 4658–4664.

[9] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[10] S. Cheng, Y. Liu, S. Ma, and X. Zhang, "Deep Feature Space Trojan Attack of Neural Networks by Controlled Detoxification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1148–1156.

[11] I. Cox, J. Kilian, F. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, 1997.

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[13] K. Deb and R. B. Agrawal, "Simulated Binary Crossover for Continuous Search Space," *Complex System*, vol. 9, 1995.

[14] K. Deb and M. Goyal, "A Combined Genetic Adaptive Search (GeneAS) for Engineering Design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.

[15] K. Doan, Y. Lao, and P. Li, "Backdoor Attack with Imperceptible Input and Latent Modification," *Advances in Neural Information Processing Systems*, vol. 34, pp. 18 944–18 957, 2021.

[16] K. Doan, Y. Lao, W. Zhao, and P. Li, "Lira: Learnable, Imperceptible and Robust Backdoor Attacks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 966–11 976.

[17] K. D. Doan, Y. Lao, and P. Li, "Marksman Backdoor: Backdoor Attacks with Arbitrary Target Class," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 260–38 273, 2022.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, 2021.

[19] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level Classification of Skin Cancer with Deep Neural Networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[20] Y. Feng, B. Ma, J. Zhang, S. Zhao, Y. Xia, and D. Tao, "Fiba: Frequency-injection based Backdoor Attack in Medical Image Analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 876–20 885.

[21] K. Gao, Y. Bai, J. Gu, Y. Yang, and S.-T. Xia, "Backdoor defense via adaptively splitting poisoned dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4005–4014.

[22] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "Strip: A Defence Against Trojan Attacks on Deep Neural Networks," in *Proceedings of the Annual Computer Security Applications Conference*, 2019, pp. 113–125.

[23] Y. Gao, H. Chen, P. Sun, J. Li, A. Zhang, Z. Wang, and W. Liu, "A Dual Stealthy Backdoor: From both Spatial and Frequency Perspectives," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 3, 2024, pp. 1851–1859.

[24] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain," *arXiv preprint arXiv:1708.06733*, 2017.

[25] C. Guo, J. S. Frank, and K. Q. Weinberger, "Low Frequency Adversarial Perturbation," in *Uncertainty in Artificial Intelligence*, 2020, pp. 1127–1137.

[26] H. A. A. K. Hammoud and B. Ghanem, "Check Your Other Door! Creating Backdoor Attacks in the Frequency Domain," *arXiv preprint arXiv:2109.05507*, 2021.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[28] R. Hou, T. Huang, H. Yan, L. Ke, and W. Tang, "A Stealthy and Robust Backdoor Attack via Frequency Domain Transform," *World Wide Web*, pp. 1–17, 2023.

[29] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark," in *Proceedings of the International Joint Conference on Neural Networks*, 2013, pp. 1–8.

[30] K. Huang, Y. Li, B. Wu, Z. Qin, and K. Ren, "Backdoor Defense via Decoupling the Training Process," in *International Conference on Learning Representations*, 2022.

[31] B. Jähne, *Digital Image Processing*. Springer Science & Business Media, 2005.

[32] W. Jiang, H. Li, G. Xu, and T. Zhang, "Color Backdoor: A Robust Poisoning Attack in Color Space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8133–8142.

[33] C. Knaus and M. Zwicker, "Dual-Domain Image De-

noising," in *IEEE International Conference on Image Processing*, 2013, pp. 440–444.

[34] S. Kolouri, A. Saha, H. Pirsiavash, and H. Hoffmann, "Universal Litmus Patterns: Revealing Backdoor Attacks in CNNs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 301–310.

[35] A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features from Tiny Images," 2009.

[36] J. Lan, J. Wang, B. Yan, Z. Yan, and E. Bertino, "Flowmur: A stealthy and practical audio backdoor attack with limited knowledge," in *IEEE Symposium on Security and Privacy*, 2024, pp. 1646–1664.

[37] Y. Le and X. Yang, "Tiny ImageNet Visual Recognition Challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[38] B. Li and W. Liu, "A Theoretical Analysis of Backdoor Poisoning Attacks in Convolutional Neural Networks," in *International Conference on Machine Learning*, 2024, pp. 8296–8316.

[39] S. Li, M. Xue, B. Z. H. Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, pp. 2088–2105, 2019.

[40] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks," in *International Conference on Learning Representations*, 2021.

[41] Y. Li, T. Zhai, B. Wu, Y. Jiang, Z. Li, and S. Xia, "Rethinking the Trigger of Backdoor Attack," *arXiv preprint arXiv:2004.04692*, 2020.

[42] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible Backdoor Attack with Sample-Specific Triggers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 463–16 472.

[43] D. Liu and Y. Qiao, "Artifacts of LADDER: Multi-objective Backdoor Attack via Evolutionary Algorithm," https://github.com/dzhliu/LADDER, 2024.

[44] D. Liu, Y. Qiao, R. Wang, K. Liang, and G. Smaragdakis, "LADDER: Multi-objective Backdoor Attack via Evolutionary Algorithm," 2024. [Online]. Available: https://arxiv.org/abs/2411.19075

[45] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-Pruning: Defending against Backdooring Attacks on Deep Neural Networks," in *International Symposium on Research in Attacks, Intrusions, and Defenses*, 2018, pp. 273–294.

[46] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection Backdoor: A Natural Backdoor Attack on Deep Neural Networks," in *European Conference on Computer Vision*, 2020, pp. 182–199.

[47] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015, pp. 3730–3738.

[48] P. Lv, C. Yue, R. Liang, Y. Yang, S. Zhang, H. Ma, and K. Chen, "A Data-free Backdoor Injection Approach in Neural Networks," in *USENIX Security Symposium*, 2023, pp. 2671–2688.

[49] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading Digits in Natural Images with Unsupervised Feature Learning," in *Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[50] T. A. Nguyen and A. Tran, "Input-Aware Dynamic Backdoor Attack," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 3454–3464.

[51] T. A. Nguyen and A. T. Tran, "WaNet - Imperceptible Warping-based Backdoor Attack," in *International Conference on Learning Representations*, 2021.

[52] K. O'shea and R. Nash, "An Introduction to Convolutional Neural Networks," *arXiv preprint arXiv:1511.08458*, 2015.

[53] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and C. Soumith, "Pytorch: An Imperative Style, High-Performance Deep Learning Library," *Advances in Neural Information Processing Systems*, vol. 32, pp. 8026–8037, 2019.

[54] X. Qiao, Y. Yang, and H. Li, "Defending Neural Backdoors via Generative Distribution Modeling," *Advances in Neural Information Processing Systems*, vol. 32, pp. 14 027–14 036, 2019.

[55] H. Qiu, Y. Zeng, S. Guo, T. Zhang, M. Qiu, and B. Thuraisingham, "Deepsweep: An Evaluation Framework for Mitigating DNN Backdoor Attacks using Data Augmentation," in *Proceedings of the ACM Asia Conference on Computer and Communications Security*, 2021, pp. 363–377.

[56] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden Trigger Backdoor Attacks," in *Proceedings of the AAAI Cconference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 957–11 965.

[57] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic Backdoor Attacks against Machine Learning Models," in *IEEE European Symposium on Security and Privacy*, 2022, pp. 703–718.

[58] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017, pp. 618–626.

[59] Y. Sharma, G. W. Ding, and M. A. Brubaker, "On the Effectiveness of Low Frequency Perturbations," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2019, pp. 3389–3396.

[60] Y. Shi, M. Du, X. Wu, Z. Guan, J. Sun, and N. Liu, "Black-box Backdoor Defense via Zero-shot Image Purification," in *Advances in Neural Information Processing Systems*, 2023, pp. 57 336–57 366.

[61] K. Simonyan and A. Zisserman, "Very Deep Convolu-

tional Networks for Large-Scale Image Recognition," in *International Conference on Learning Representations*, 2015.

[62] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[63] T. J. L. Tan and R. Shokri, "Bypassing Backdoor Detection Algorithms in Deep Learning," in *IEEE European Symposium on Security and Privacy*, 2020, pp. 175–183.

[64] D. J. Tolhurst, Y. Tadmor, and T. Chao, "Amplitude Spectra of Natural Images," *Ophthalmic and Physiological Optics*, pp. 229–232, 1992.

[65] B. Tran, J. Li, and A. Madry, "Spectral Signatures in Backdoor Attacks," *Advances in Neural Information Processing Systems*, vol. 31, pp. 8011–8021, 2018.

[66] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks," in *IEEE Symposium on Security and Privacy*, 2019, pp. 707–723.

[67] R. Wang, H. Chen, Z. Zhu, L. Liu, and B. Wu, "Versatile Backdoor Attack with Visible, Semantic, Sample-Specific, and Compatible Triggers," *arXiv preprint arXiv:2306.00816*, 2023.

[68] T. Wang, Y. Yao, F. Xu, S. An, H. Tong, and T. Wang, "An invisible Black-box Backdoor Attack through Frequency Domain," in *European Conference on Computer Vision*, 2022, pp. 396–413.

[69] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S. Huang, "D3: Deep Dual-Domain Based Fast Restoration of JPEG-Compressed Images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2764–2772.

[70] Y. Zeng, M. Pan, H. A. Just, L. Lyu, M. Qiu, and R. Jia, "Narcissus: A Practical Clean-Label Backdoor Attack with Limited Information," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 771–785.

[71] Y. Zeng, W. Park, Z. M. Mao, and R. Jia, "Rethinking the Backdoor Attacks' Triggers: A Frequency Perspective," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 473–16 481.

[72] J. Zhang, J. Chi, Z. Li, K. Cai, Y. Zhang, and Y. Tian, "Badmerging: Backdoor Attacks against Model Merging," *arXiv preprint arXiv:2408.07362*, 2024.

[73] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.

[74] Z. Zhang, Q. Liu, Z. Wang, Z. Lu, and Q. Hu, "Backdoor Defense via Deconfounded Representation Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 228–12 238.

[75] Z. Zhao, X. Chen, Y. Xuan, Y. Dong, D. Wang, and K. 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*, 2022, pp. 15 213–15 222.

[76] N. Zhong, Z. Qian, and X. Zhang, "Imperceptible Backdoor Attack: From Input Space to Feature Representation," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2022, pp. 1736–1742.

[77] M. Zhu, S. Wei, H. Zha, and B. Wu, "Neural Polarizer: A Lightweight and Effective Backdoor Defense via Purifying Poisoned Features," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 1132–1153.

APPENDIX

A. Evaluating LADDER against SOTA Backdoor Defenses

Although successfully evading several classic backdoor defenses (see Section VII-C), LADDER is not perfectly robust against some (white-box) backdoor defenses especially defenses requiring model and training manipulation. Recall that, in the strict black-box setting, attackers are not allowed to access the victim model (such as parameters, structures and gradient information) nor manipulate the training process.

We test LADDER against three new SOTA backdoor defenses, ASD [21], CBD [74] and DBD [30], which all rely on manipulating the training process (see [44] for more results). ASD adaptively splits clean from the poisoned dataset during training so as to defend backdoors. CBD leverages statistical effect among variables on the image to mitigate attacks. DBD proposes a three-stage mechanism, which involves learning on label-removed data, credible sample filtering and fine-tuning the trained model.

TABLE X: Attack performance measured by ACC (%) and ASR (%) for 7 backdoor attacks against ASD, CBD and DBD on CIFAR-10 dataset with ResNet18, WideResNet (WRN-16-1) and ResNet18, respectively.

Defenses	ASD [21]		CBD [21]		DBD [30]	
Attack	ACC	ASR	ACC	ASR	ACC	ASR
BadNets [24]	93.4	1.2	87.5	1.1	92.4	1.0
Blend [9]	93.7	1.6	87.5	2.0	92.2	**1.7**
WaNet [51]	93.1	1.7	86.6	4.2	91.2	0.4
SIG [4]	87.8	0.7	87.3	0.3	91.6	0.3
DUBA [23]	92.6	4.0	87.6	5.0	90.8	0.2
Narcissus-D [70]	93.8	0.0	87.9	4.1	91.1	0.0
Ours	92.4	**25.0**	86.6	**5.2**	90.9	0.0

We evaluate attack effectiveness of LADDER and other attacks against ASD, CBD and DBD using their default parameter settings. The results are in Table X. In all defenses, all the attacks achieve low ASRs (ranging from $0\% \sim 25.0\%$ in ASD, $0.3\% \sim 5.2\%$ for CBD and $0\% \sim 1.7\%$ in DBD). Under defenses, LADDER provides better ASRs of 25.0% against ASD and 5.2% under CBD, indicating a slight advantage on evading defenses over others. This is because LADDER triggers require smaller perturbations (only 0.3183 l_2-norm on CIFAR-10). As a result, the poisoned samples are

Fig. 10: Poisoned images produced by LADDER.

more likely to be split into the clean data pool by ASD and seldom being detected by CBD with their statistical effect, thus delivering relatively higher ASR of LADDER than others. DBD eliminates the effectiveness of LADDER because of its fine-pruning process. Since the trigger produced by LADDER is "weaker", the trigger injected into the model is gradually pruned and eventually erased after a large number of fine-pruning iterations.

B. Adaptive Defense

Several image-level anomaly detectors that have been proposed in the spatial domain can be used to eliminate the threat of LADDER on DNNs in black-box environment. We propose a frequency domain anomaly detector that locates poisoned images by exploring the statistical information of the spectrum. Specifically, given the averaged spectra M of a natural image x, the averaged magnitudes $\mathcal{A}$ of the frequency bands f in M have a relationship $\mathcal{A} \propto f^s$ on the double-logarithmic coordinates with a constant slope s=2 [6, 64, 71]. To obtain M of a given RGB image x, we first convert x to the spectrum X using DCT in Equation (2). Then, we compute the power (of magnitudes) in each channel of X, i.e., X_c^{pow}=$X_c \odot X_c$, where $c \in \{R, G, B\}$ and $\odot$ is the Hadamard product. For each X_c^{pow}, we divide the frequency bands into groups $f = \{f_0, f_1, \cdots, f_{max-1}\}$ where $k \in [0, max)$ and max is the dimension of the spectrum in X, so that the frequency bands in each group f_k have the same distance to the upper left corner of the spectrum. We calculate the averaged magnitude of each group of frequency bands f_k to obtain the averaged spectra M_c for each channel $c \in \{R, G, B\}$. Finally, we obtain the logarithm of the averaged magnitude of the frequency bands from M_R, M_G and M_B, i.e, $\log(M^{avg})$= $\log(\frac{(M_R+M_G+M_B)}{3})$, and fit the slope s with $\log(f)$ and $\log(M^{avg})$.

We show, in Table XI, s (averaged over 1000 randomly selected samples from CIFAR-10) obtained with clean and poisoned data by the black-box backdoor attacks. We see that s is the smallest on clean samples compare to poisoned data. The slope s is a feasible indicator to distinguish poisoned data.

C. Trade-offs among Attack Objectives

We illustrate the conflict between attack effectiveness and stealthiness in Figure 3. To further investigate the trade-off between stealthiness and robustness, we generate random noise of size 3×3 with an initial l_2-norm of 0.25. We create two additional variants by scaling the l_2-norm of the original noise by $2\times$ and $4\times$. These noises are used as triggers and injected into the low-, mid-, and high-frequency regions. We evaluate attack effectiveness and robustness of each noise under different levels of stealthiness and injection regions on CIFAR-10 using ResNet18. Attack robustness is measured by averaging ASRs after the preprocessings (see Table IV).

In Table XII, increasing the l_2-norm (i.e., reducing stealthiness) enhances attack robustness in both low- and high-frequency regions, though it has a minimal effect in the mid-frequency region. For instance, raising the l_2-norm from 0.25 to 1.0 improves attack robustness by 15.37% in the low-frequency region while yielding only a slight increase of 1.44% in the mid-frequency region. A closer examination of the Eff-to-Rob ratio reveals that, in the low-frequency region, increasing the l_2-norm has a minimal impact on robustness, with a max. difference of 3.17%. Moreover, in the mid-frequency region, the ratio closely aligns with Robs across different l_2-norm values. In the high-frequency region, the ratio rises by 11.79%. The results indicate a distinct trade-off between stealthiness and attack robustness in both low- and high-frequency regions.

Table XII also indicates that under the same stealthiness level, inserting trigger patterns in different spectral regions has a modest impact on effectiveness. For example, with an l_2-norm of 0.25, moving the trigger from low-frequency to high-frequency region increases attack effectiveness by 13.88%. But this adjustment significantly harms robustness, resulting in a 50.89% decrease. We conclude that designing triggers in the low-frequency region has a minimal impact on attack effectiveness while significantly enhancing trigger robustness.

TABLE XI: The averaged s and standard deviation on 1000 randomly chosen images from CIFAR-10 under attacks.

Attacks	Clean	BadNets	SIG	Blend	FTrojan	FIBA	Ours
Slope	-1.8922 (0.3810)	-1.6882 (0.3803	-1.5922 (0.3645)	-1.7602 (0.3509)	-1.8236 (0.3591)	-1.7826 (0.3456)	-1.8238 (0.3803)

TABLE XII: The attack effectiveness (Eff) (%), robustness (Rob) (%) and Eff-to-Rob ratio (%) on CIFAR-10 and ResNet18, evaluated by injecting noises into Low-, Mid- and High-frequency regions, across different levels of stealthiness.

l_2-norm	Metric	Region of Injection		
		L	M	H
0.25	Eff	86.12	99.96	100.0
	Rob	81.51	30.65	30.62
	Ratio	94.64	30.66	30.62
0.5	Eff	95.89	100.0	100.0
	Rob	91.70	31.50	33.31
	Ratio	95.63	31.50	33.31
1.0	Eff	99.04	100.0	100.0
	Rob	96.88	32.10	42.41
	Ratio	97.81	32.10	42.41