WaveAttack: Asymmetric Frequency Obfuscation-based Backdoor Attacks Against Deep Neural Networks

Jun Xia[1,†], Zhihao Yue[1,†], Yingbo Zhou[1], Zhiwei Ling[1], Yiyu Shi[2], Xian Wei[1], Mingsong Chen[1*]
[1]MoE Eng. Research Center of SW/HW Co-design Tech. and App., East China Normal University
[2]Department of Computer Science and Engineering, University of Notre Dame
{jxia, 51215902034, 52215902009, 51215902044}@stu.ecnu.edu.cn,
yshi4@nd.edu, {xwei, mschen}@sei.ecnu.edu.cn

Abstract

Due to the increasing popularity of Artificial Intelligence (AI), more and more backdoor attacks are designed to mislead Deep Neural Network (DNN) predictions by manipulating training samples or processes. Although backdoor attacks have been investigated in various scenarios, they still suffer from the problems of both low fidelity of poisoned samples and non-negligible transfer in latent space, which make them easily identified by existing backdoor detection algorithms. To overcome this weakness, this paper proposes a novel frequency-based backdoor attack method named WaveAttack, which obtains high-frequency image features through Discrete Wavelet Transform (DWT) to generate highly stealthy backdoor triggers. By introducing an asymmetric frequency obfuscation method, our approach adds an adaptive residual to the training and inference stages to improve the impact of triggers, thus further enhancing the effectiveness of WaveAttack. Comprehensive experimental results show that, WaveAttack can not only achieve higher effectiveness than state-of-the-art backdoor attack methods, but also outperform them in the fidelity of images (i.e., by up to 28.27% improvement in PSNR, 1.61% improvement in SSIM, and 70.59% reduction in IS). Our code is available at https://github.com/BililiCode/WaveAttack.

1 Introduction

Along with the prosperity of Artificial Intelligence (AI), Deep Neural Networks (DNNs) have become increasingly prevalent in numerous safety-critical domains for precise perception and real-time control, such as autonomous vehicles [1], medical diagnosis, and industrial automation [2]. However, the trustworthiness of DNNs faces significant threats due to various notorious adversarial and backdoor attacks. Typically, adversarial attacks [3, 4] manipulate input data during the inference stage to induce incorrect predictions by a trained DNN, while backdoor attacks [5] tamper with training samples or processes to embed concealed triggers during training, which can be exploited to generate malicious outputs. Although adversarial attacks on DNNs frequently appear in various scenarios, backdoor attacks have attracted more attention because of their stealthiness and effectiveness. Generally, the performance of backdoor attacks can be evaluated by the following three objectives of an adversary: i) *efficacy* that refers to the effectiveness of an attack in causing the target model to produce incorrect outputs or exhibit unintended behavior; ii) *specificity* that denotes the precision of the attack in targeting a specific class; and iii) *fidelity* that represents the degree to which adversarial examples or poisoned training samples are indistinguishable from their benign counterparts [6]. Note that efficacy and specificity represent the effectiveness of backdoor attacks, while fidelity denotes the stealthiness of backdoor attacks.

*Mingsong Chen is the corresponding author. † Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

In order to achieve higher stealthiness and effectiveness, existing backdoor attack methods (e.g. IAD [7], WaNet [8], BppAttack [9], and FTrojan [10]) are built based on various optimizations, which can be mainly classified into two categories. The former is the *sample minimal impact* method that can optimize the size of the trigger and minimize its pixel value, making the backdoor trigger difficult to detect in training samples for the purpose of achieving the high stealthiness of a backdoor attacker. Although these methods are promising in backdoor attacks, due to the explicit trigger influence on training samples, they cannot fully evade existing backdoor detection methods based on training samples. The latter is the *latent space obfuscation-based* methods, which can be integrated into any existing backdoor attack methods. Using asymmetric samples, these methods can obfuscate the latent space between benign samples and poisoned samples [11]. Although these methods can bypass latent space detection techniques, they suffer greatly from low image quality, making them extremely difficult to apply in practice. Therefore, *how to improve both the effectiveness and stealthiness of backdoor attacks while minimally impacting the quality of training samples is becoming a significant challenge in the development of backdoor attacks.*

According to the work in [12], wavelet transform techniques have been widely investigated in various image-processing tasks [13, 14, 15], where high-frequency features can be utilized to enhance the generalization ability of DNNs and remain imperceptible to humans. Inspired by this finding, this paper introduces a novel backdoor attack method named WaveAttack, which adopts Discrete Wavelet Transform (DWT) to extract high-frequency components for highly stealthy backdoor trigger generation. To improve the impact of triggers and further enhance the effectiveness of our approach, we employ *asymmetric frequency obfuscation* that utilizes an asymmetric coefficient of the trigger in the high-frequency domain during the training and inference stages. This paper makes the following three contributions:

- We introduce a promising frequency-based backdoor trigger generation method, which can effectively generate the backdoor residuals for the high-frequency component based on DWT, thus ensuring the high fidelity of poisoned samples.
- We propose a novel asymmetric frequency-based obfuscation backdoor attack method to enhance the stealthiness and effectiveness of WaveAttack, which can increase stealthiness in latent spaces and improve the Attack Success Rate in training samples.
- We conduct comprehensive experiments on four public benchmarks to demonstrate that WaveAttack outperforms state-of-the-art (SOTA) backdoor attack methods from the perspectives of both stealthiness and effectiveness.

2 Related Work

Backdoor Attack. Typically, backdoor attacks try to embed backdoors into DNNs by manipulating their input samples and training processes. In this way, adversaries can control DNN output through concealed triggers, which results in manipulated predictions [16]. Depending on whether the training process is manipulated, existing backdoor attacks can be categorized into two types, i.e., *training-unmanipulated* and *training-manipulated* attacks. Specifically, training-unmanipulated attacks only inject a visible or invisible trigger into the training samples of some DNN, leading to its recognition errors [5]. For example, Chen et al. [17] introduced a Blend attack that generates poisoned data by merging benign training samples with specific key visible triggers. Moreover, there exists a large number of invisible trigger-based backdoor attack methods, such as natural reflection [18], human imperceptible noise [19], and image perturbation [10], which exploit the changes induced by real-world physical environments. Although these training-unmanipulated attacks are promising, due to their substantial impacts on training sample quality, most of them still can be easily identified somehow. As an alternative, training-manipulated attacks [8, 9] assume that adversaries from some malicious third party can control the key steps of the training process, thus achieving a stealthier attack. Although the above two categories of backdoor attacks are promising, most of them struggle with coarse-grained optimization of effectiveness and stealthiness, complicating the acquisition of superior backdoor triggers. Due to the significant difference in latent space and low poisoned sample fidelity, they cannot evade the latest backdoor detection methods.

Backdoor Defense. There are two major types of backdoor defense methods, i.e., the *detection-based defense* and *erasure-based defense*. The detection-based defenses can be further classified into two categories, i.e., sample-based and latent space-based detection methods. Specifically, sample-based detection methods can identify the differences in the distribution between poisoned samples and benign samples [20], while latent space-based detection methods aim to find the disparity between the latent spaces of poisoned samples and benign samples [21]. Unlike the detection strategies described above that aim to prevent the injection of backdoors into DNNs by identifying poisoned samples during the training stages, erasure-based defenses can eradicate the backdoors from DNNs. So far, the erasure-based defenses can be classified into three categories, i.e., poison suppression-based, model reconstruction-based, and trigger generation-based defenses. The poison suppression-based methods [22] utilize the differential learning speed between poisoned and benign samples during training to mitigate the influence of backdoor triggers on DNNs. The model reconstruction-based methods [23, 24] use a selected set of benign data to rebuild DNN models, aiming to mitigate the impact of backdoor triggers. The trigger generation-based methods [25, 26] reverse engineer backdoor triggers by capitalizing on the effects of backdoor attacks on training samples.

To the best of our knowledge, WaveAttack is the first attempt to generate backdoor triggers for the high-frequency component obtained through DWT. Unlike existing backdoor attack methods, WaveAttack first considers both the fidelity of poisoned samples and latent space obfuscation simultaneously. By using asymmetric frequency obfuscation, WaveAttack can not only acquire backdoor attack effectiveness but also achieve high stealthiness regarding both image quality and latent space.

3 Our Method

In this section, we first present the preliminaries for the problem notations, threat model, and adversarial goal. Then, we visualize our motivations for adding triggers to the high-frequency components. Finally, we celebrate the attack process of our method, WaveAttack.

3.1 Preliminaries

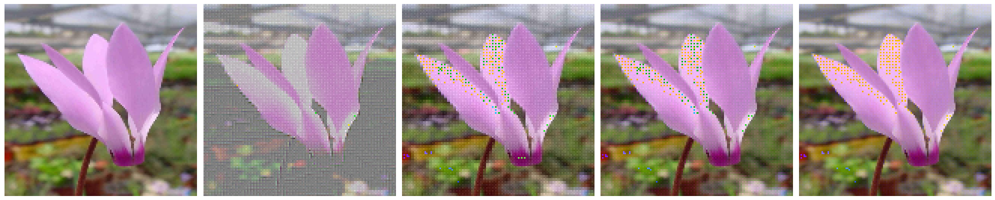
Notations. We follow the training scheme of Adapt-Blend [11]. Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ be a clean training dataset, where $\boldsymbol{x}_i \in \mathbb{X} = \{0, 1, ..., 255\}^{C\times W\times H}$ is an image, and $y_i \in \mathbb{Y} = \{1, 2, ..., K\}$ is its corresponding label. Note that K represents the number of labels. For a given training dataset, we select a subset of $\mathcal{D}$ with a poisoning rate p_a as the *payload samples* $\mathcal{D}_a = \{(\boldsymbol{x'}_i, y_t)|\boldsymbol{x'}_i = T(\boldsymbol{x}_i), \boldsymbol{x}_i \in \mathbb{X}\}$, where $T(\cdot)$ is a backdoor transformation function, and y_t is an adversary-specified target label. We use a subset of $\mathcal{D}$ with poisoning rate p_r as the *regularization samples* $\mathcal{D}_r = \{(\boldsymbol{x'}_i, y_i)|\boldsymbol{x'}_i = T(\boldsymbol{x}_i), \boldsymbol{x}_i \in \mathbb{X}\}$. For a given dataset, a backdoor attack adversary tries to train a backdoored model f that predicts $\boldsymbol{x}$ as its corresponding label, where $\boldsymbol{x} \in \mathcal{D} \cup \mathcal{D}_a \cup \mathcal{D}_r$.

Threat Model. Similar to existing backdoor attack methods [7, 8, 9], we assume that adversaries have complete control over the training datasets, and model implementation. They can embed backdoors into the DNNs by poisoning the given training dataset. Moreover, in the inference stage, we assume that adversaries can only query backdoored models using any samples.

Adversarial Goal. Throughout the attack process, adversaries strive to achieve two core goals, i.e., effectiveness and stealthiness. Effectiveness indicates that adversaries try to train backdoored models with a high ASR while ensuring that the decrease in Benign Accuracy (BA) remains imperceptible. Stealthiness indicates that samples with triggers have high fidelity and that there is no latent separation between poisoned and clean samples in the latent space.

3.2 Motivation

Unlike humans who are not sensitive to high-frequency features, DNNs can effectively learn high-frequency features of images [12], which can be used for the generation of backdoor triggers. In other words, the poisoned samples generated by high-frequency features can easily escape various examination methods by humans. Based on this observation, if we can design backdoor triggers on top of high-frequency features, the stealthiness of corresponding backdoored attacks can be ensured. To obtain high-frequency components from the training samples, we resort to Discrete Wavelet Transform (DWT) to capture characteristics from both the time and frequency domains [27], allowing the extraction of multiple frequency components from the training samples. The reason why we adopt DWT rather than Discrete Cosine Transform (DCT) is that DWT can better capture high-frequency features from training samples (i.e., edges and textures) and allows superior reverse operations during both encoding and decoding phases, thus minimizing the impact on the



(a) Original (b) LL with noises (c) LH with noises (d) HL with noises (e) HH with noises

Figure 1: A motivating example for the backdoor trigger design on high-frequency components.

fidelity of poisoned samples. In our approach, we adopt a classic and effective biorthogonal wavelet transform method (i.e., Haar wavelet [28]), which mainly contains four kernel operations, i.e., LL^T, LH^T, HL^T, and HH^T. Here L and H denote the low and high pass filters, respectively, where $L^T = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\end{bmatrix}, H^T = \frac{1}{\sqrt{2}}\begin{bmatrix}-1 & 1\end{bmatrix}$. Note that, based on the four operations, the Haar wavelet can decompose an image into four frequency components (i.e., LL, LH, HL, HH) using DWT, where HH only contains the high-frequency information of a sample. Meanwhile, the Haar wavelet can reconstruct the image from the four frequency components via the Inverse Discrete Wavelet Transform (IDWT). To verify the motivation of our approach, Figure 1 illustrates the impact of adding the same noises to different frequency components on an image, i.e., Figure 1(a). We can find that, compared to the other three poisoned images, i.e., Figure 1(b) to 1(d), it is much more difficult to determine the difference between the original image and the poisoned counterpart in HH, i.e., Figure 1(e). Therefore, it is more suitable to inject triggers into the high-frequency component (i.e., HH) for backdoor attack purposes.

3.3 Implementation of WaveAttack

In this subsection, we detail the design of our WaveAttack approach. As shown in Figure 2, we give an overview of our attack method WaveAttack. To be concrete, we first make samples poisoned into payload and regularization samples using our trigger design, which is implemented with frequency transformation. Then, we use benign samples, payload samples, and regularization samples to train a classifier to achieve the core goals of WaveAttack.

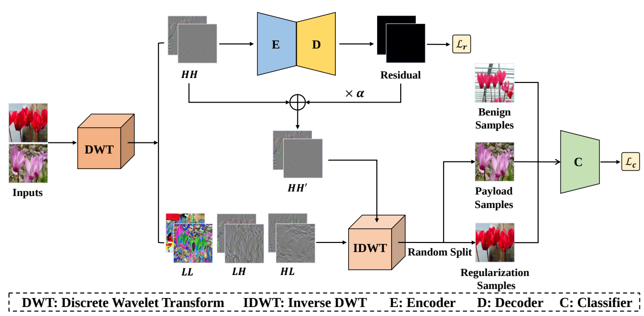


Figure 2: Overview of our attack method WaveAttack.

Trigger Design. As mentioned above, our WaveAttack approach aims to achieve a stealthier backdoor attack, introducing triggers into the HH frequency component. Figure 2 contains the process of generating triggers using WaveAttack. First, we obtain the four components of the samples through DWT. Then, to generate imperceptible sample-specific triggers, we employ an encoder-decoder network as a generator g. These generated triggers are imperceptible additive residuals. Next, to achieve asymmetric frequency obfuscation, we multiply the residuals by a coefficient α, and generate the poisoned HH' component with the triggers as follows:

$$\boldsymbol{HH'} = \boldsymbol{HH} + \alpha \cdot g(\boldsymbol{HH}; \boldsymbol{\omega}_g), \tag{1}$$

where $\boldsymbol{\omega}_g$ is the generator parameters. Finally, we can utilize IDWT to reconstruct four frequency components of poisoned samples. Specifically, we use a U-Net-like [29] generator to obtain residuals, although other methods (e.g., VAE [30]) can also be used by the adversary. This is because the skip connections of U-Net can effectively preserve the features of inputs with minimal impacts [29].

Optimization Objective. Our WaveAttack method has two networks to optimize. We aim to optimize a generator g to generate small residuals with minimal impact on the samples. Furthermore, our objective is to optimize a backdoored classifier c, enabling the effectiveness and stealthiness of WaveAttack. For the first optimization objective, we use the L_∞ norm to optimize small residuals. The optimization objective is defined as follows:

$$\mathcal{L}_r = ||g(HH; \boldsymbol{\omega}_g)||_\infty. \tag{2}$$

For the second optimization objective, we train the classifier using the cross-entropy loss function in $\mathcal{D}$, $\mathcal{D}_a$, and $\mathcal{D}_r$ dataset. The optimization objective is defined as follows:

$$\mathcal{L}_c = \mathcal{L}(\boldsymbol{x}_p, y_t; \boldsymbol{\omega}_f) + \mathcal{L}(\boldsymbol{x}_r, \boldsymbol{y}; \boldsymbol{\omega}_c) + \mathcal{L}(\boldsymbol{x}_b, \boldsymbol{y}; \boldsymbol{\omega}_c), \tag{3}$$

where $\mathcal{L}(\cdot)$ is the cross-entropy loss function, $\boldsymbol{\omega}_f$ is the classifier parameters, $\boldsymbol{x}_b \in \mathcal{D}$, $\boldsymbol{x}_p \in \mathcal{D}_a$, and $\boldsymbol{x}_r \in \mathcal{D}_r$. The total loss function is as follows:

$$\mathcal{L}_{total} = \mathcal{L}_c + \mathcal{L}_r. \tag{4}$$

Algorithm Description. Algorithm 1 details the training process of our WaveAttack approach. At the beginning of WaveAttack training (Line 2), the adversary randomly selects a minibatch data $(\boldsymbol{x}, \boldsymbol{y})$ from $\mathcal{D}$, which has b training samples. Lines 4-6 calculate the number of poisoned samples, payload samples, and regulation samples, respectively. Lines 7-11 denote the process of modifying samples by injecting triggers into the high-frequency component. After acquiring the modified samples in Line 7, Line 8 decomposes the samples into four frequency components (i.e., LL, LH, HL and HH) by DWT. Then, in Lines 9-10, we add the residual to the frequency component HH by Equation (1) and obtain the frequency component HH'. Line 11 reconstructs the samples from the four frequency components via IDWT. Lines 12-15 compute the optimization object using Equations (2) to (4). In Lines 16-17, we can use an optimizer (e.g., SGD optimizer) to update the parameters of the generator model and classifier model. Line 20 returns the well-trained generator model parameters $\boldsymbol{\omega}_g$ and the classifier model parameters $\boldsymbol{\omega}_{\hat{c}}$.

Algorithm 1 Training of WaveAttack

Require: i) $\mathcal{D}$, benign training dataset. ii) $\boldsymbol{\omega_g}$, randomly initialized generator parameters. iii) $\boldsymbol{\omega_c}$, randomly initialized classifier parameters. iv) p_a, payload sample rate. v) p_r, rate of regularization samples. vii) y_t, target label. vi) E, # of epochs in training process.

Ensure: i) $\boldsymbol{\omega}_g$, well-trained generator model. ii) $\boldsymbol{\omega}_{\hat{c}}$, well-trained classifier model.

1: **for** $e = 1, \ldots, E$ **do**
2:   **for** $(\boldsymbol{x}, \boldsymbol{y})$ in $\mathcal{D}$ **do**
3:     $b \leftarrow \boldsymbol{x}$.shape[0]
4:     $n_m \leftarrow (p_a + p_r) \times b$
5:     $n_a \leftarrow p_a \times b$
6:     $n_r \leftarrow p_r \times b$
7:     $\boldsymbol{x}_m \leftarrow \boldsymbol{x}[:n_m]$
8:     $(\boldsymbol{LL}, \boldsymbol{LH}, \boldsymbol{HL}, \boldsymbol{HH}) \leftarrow DWT(\boldsymbol{x}_m)$
9:     $\boldsymbol{resdiual} \leftarrow \alpha \cdot g(\boldsymbol{HH}; \boldsymbol{\omega}_g)$
10:     $\boldsymbol{HH}' \leftarrow \boldsymbol{HH} + \boldsymbol{resdiual}$
11:     $\boldsymbol{x}_m \leftarrow IDWT(\boldsymbol{LL}, \boldsymbol{LH}, \boldsymbol{HL}, \boldsymbol{HH}')$
12:     $\mathcal{L}_1 \leftarrow \mathcal{L}(\boldsymbol{x}_m[n_a:], y_t; \boldsymbol{\omega}_c)$
13:     $\mathcal{L}_2 \leftarrow \mathcal{L}(\boldsymbol{x}_m[:n_r], \boldsymbol{y}[n_a:n_r]; \boldsymbol{\omega}_c)$
14:     $\mathcal{L}_3 \leftarrow \mathcal{L}(\boldsymbol{x}[n_m:], \boldsymbol{y}[n_m:]; \boldsymbol{\omega}_c)$
15:     $\mathcal{L} \leftarrow \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 + ||\boldsymbol{resdiual}||_\infty$
16:     $\mathcal{L}$.backward()
17:     update($\boldsymbol{\omega}_g, \boldsymbol{\omega}_c$)
18:   **end for**
19: **end for**
20: **Return** $\boldsymbol{\omega}_g, \boldsymbol{\omega}_{\hat{c}}$

Asymmetric Frequency Obfuscation. According to [11], regularization samples $\mathcal{D}_r$ can make DNNs learn the semantic feature of each class and the trigger feature, which can make the backdoor attack stealthy in the latent space. However, using the same trigger in samples during the inference process may diminish the fidelity of poisoned samples. Hence, it is crucial to devise an asymmetric frequency obfuscation method to enhance the effectiveness of backdoor attack methods. In our approach, we employ a coefficient α with a small value (i.e., α=1.0) to improve the stealthiness of triggers during the training process, while a larger value (i.e., α=100.0) is used to enhance the impact of triggers and further improve the effectiveness of WaveAttack. This method ensures that, during the inference process, the backdoored samples have sufficient "power" to activate the DNN backdoor, thus achieving a high ASR.

4 Experiments

To demonstrate the effectiveness and stealthiness of our approach, we implemented WaveAttack using Pytorch and compared its performance with seven existing backdoor attack methods. We conducted all experiments on a workstation with a 3.6GHz Intel i9 CPU, 32GB of memory, an NVIDIA GeForce RTX3090 GPU, and a Ubuntu operating system. We designed comprehensive experiments to address the following three research questions:

RQ1 (Effectiveness of WaveAttack): Can WaveAttack successfully inject backdoors into DNNs?

RQ2 (Stealthiness of WaveAttack): How stealthy are the poisoned samples generated by WaveAttack compared to those generated by SOTA backdoor attack methods?

RQ3 (Resistance to Existing Defenses): Can WaveAttack resist existing defense methods?

4.1 Experimental Settings

Datasets and DNNs. We evaluated all the attack methods on four well-known benchmark datasets, i.e., CIFAR-10 [31], CIFAR-100 [31], GTSRB [32] and a subset of ImageNet (with the first 20 categories) [33]. The statistics of the datasets adopted in the experiments are presented in Table 6 (see Appendix 7.1). We used ResNet18 [34] as the base DNN for the effectiveness and stealthiness evaluation. In addition, we used VGG16 [35], SENet18 [36], ResNeXt29 [37], and DenseNet121 [38] to evaluate the generalizability of WaveAttack.

Attack Configurations. To compare the performance of WaveAttack with SOTA attack methods, we considered nine SOTA backdoor attacks, i.e., BadNets [5], Blend [17], IAD [7], WaNet [8], BppAttack [9], Adapt-Blend [11], FTrojan [10], LIRA [39], and Fiba [40]. Note that, similar to our work, Adapt-Blend has asymmetric triggers, and FTrojan and Fiba are also frequency domain-based attack methods. We performed the attack methods using the default hyperparameters described in their original papers. Specifically, the poisoning rate is set to 10% with a target label of 0 to ensure a fair comparison. See the Appendix for more details on both data and attack settings.

Evaluation Metrics. Similar to the existing work in [10], we evaluated the effectiveness of all attack methods using two metrics, i.e., Attack Success Rate (ASR) and Benign Accuracy (BA). To evaluate the stealthiness of all attack methods, we used three metrics, i.e., Peak Signal-to-Noise Ratio (PSNR) [41], Structure Similarity Index Measure (SSIM) [42], and Inception Score (IS) [43].

4.2 Effectiveness Evaluation (RQ1)

Effectiveness Comparison with SOTA Attack Methods. To evaluate the effectiveness of WaveAttack, we compared the ASR and BA of WaveAttack with nine SOTA attack methods. Since IAD [7] cannot attack the ImageNet dataset based on its open-source code, we do not provide its comparison result. Table 1 shows the attack performance of different attack methods. From this table, we can find that WaveAttack can acquire a high ASR without obviously degrading the BA. Especially for the datasets CIFAR-10 and GTSRB, our WaveAttack achieves the best ASR and BA compared to other SOTA attack methods. Compared to frequency domain-based attack methods (i.e., FTrojan and Fiba), WaveAttack outperforms FTrojan and Fiba in BA for CIFAR-10, CIFAR-100, GTSRB, and ImageNet datasets. Moreover, compared to the asymmetric-based method Adapt-Blend, WaveAttack can also obtain superior performance in terms of ASR and BA for all datasets.

Table 1: Attack performance comparison between WaveAttack and seven SOTA attack methods. The best and the second-best results are **highlighted** and underlined, respectively.

Method	CIFAR-10		CIFAR-100		GTSRB		ImageNet	
	BA ↑	ASR ↑	BA ↑	ASR ↑	BA ↑	ASR ↑	BA ↑	ASR ↑
No attack	94.59	-	75.55	-	99.00	-	87.00	-
BadNets [5]	94.36	**100**	74.90	**100**	98.97	**100**	85.80	**100**
Blend [17]	94.51	99.91	75.10	99.84	98.26	**100**	86.40	**100**
IAD [7]	94.32	99.12	75.14	99.28	99.26	98.37	-	-
WaNet [8]	94.23	99.57	73.18	98.52	99.21	99.58	**86.60**	89.20
BppAttack [9]	94.10	**100**	74.68	**100**	98.93	99.91	85.90	99.50
Adapt-Blend [11]	94.31	71.57	74.53	81.66	98.76	60.25	86.40	90.10
FTrojan [10]	94.29	**100**	75.37	**100**	98.83	**100**	85.10	**100**
LIRA [39]	93.57	99.96	73.09	99.98	10.74	99.03	-	-
Fiba [40]	93.80	75.40	74.87	80.36	99.12	85.18	-	-
WaveAttack (Ours)	**94.55**	**100**	**75.41**	**100**	**99.30**	**100**	**86.60**	**100**

Effectiveness on Different Networks. To evaluate the effectiveness of WaveAttack on various networks, we conducted experiments on CIFAR-10 using different networks (i.e., VGG16 [35], SENet18 [36], ResNeXt29 [37], and DenseNet121 [38]). Table 2 shows the attack performance of WaveAttack on these networks. From this table, we can find that our WaveAttack approach can successfully embed the backdoor into different networks. WaveAttack can not only cause malicious impacts of backdoor

Table 2: Attack performance on different DNNs.

Network	No Attack	WaveAttack	
	BA ↑	BA ↑	ASR ↑
VGG16 [35]	93.62	93.70	99.76
SENet18 [36]	94.51	94.63	100
ResNeXt29 [37]	94.79	95.08	100
DenseNet121 [38]	95.29	95.10	99.78

attacks, but also maintain a classification performance with high BA, demonstrating the generalizability of WaveAttack on different network architectures.

Effectiveness of WaveAttack with Different Discrete Wavelet Transforms. Due to simplicity and computational efficiency, we adopted the most common Haar wavelet in our wavelet transformation procedure. Since different wavelets are applicable to Discrete Wavelet Transform (DWT) in our method, we conducted experiments to incorporate the Daubechies (DB) wavelet, which has stronger orthogonality. Table 3 summarizes the experimental results of WaveAttack with different wavelets. From the table, we can find that the influence of different wavelets on the performance of our method is limited, indicating that WaveAttack maintains its effectiveness and stealthiness among different wavelet transformations.

Table 3: Attack performance with different DWTs.

Wavelet	Dataset	IS ↓	PSNR ↑	SSIM ↑	BA ↑	ASR ↑
Haar	CIFAR-10	0.011	47.49	0.9979	94.55	100
	CIFAR-100	0.005	50.12	0.9992	75.41	100
	GTSRB	0.058	40.67	0.9877	99.30	100
DB	CIFAR-10	0.007	47.53	0.9989	94.77	95.60
	CIFAR-100	0.005	50.32	0.9994	76.64	80.43
	GTSRB	0.022	41.95	0.9881	98.21	99.50

4.3 Stealthiness Evaluation (RQ2)

To evaluate the stealthiness of WaveAttack, we compared the images with the triggers generated by WaveAttack with the ones of SOTA attack methods. In addition, we used t-SNE [44] to visualize latent spaces for poisoned samples and benign samples from the target label.

Stealthiness Results from The Perspective of Images. To show the stealthiness of triggers generated by WaveAttack, Figure 3 compares WaveAttack and SOTA attack methods using poisoned samples and their magnified residuals ($\times 5$) counterparts. From this figure, we can see that the residual generated by WaveAttack is the smallest and only leaves a few subtle artifacts. The trigger injected by WaveAttack is almost invisible to humans.

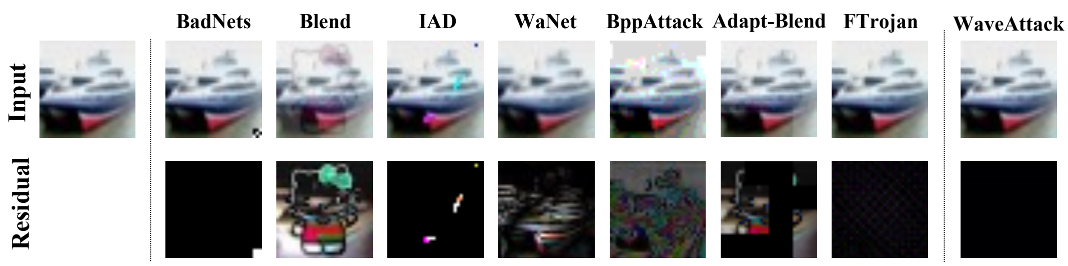
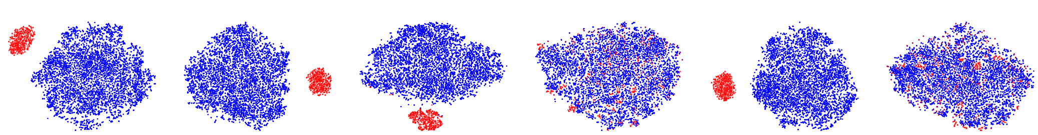


Figure 3: Comparison of examples generated by seven backdoor attacks. For each attack, we show the poisoned sample (top) and the magnified ($\times 5$) residual (bottom).

We used three metrics (i.e., PSNR, SSIM, and IS) to evaluate the stealthiness of triggers generated by our WaveAttack. Table 4 shows the results of the stealthiness comparison between WaveAttack and nine SOTA attack methods. From this table, we can see that WaveAttack achieves the best stealthiness in the CIFAR-10 and ImageNet datasets. Note that although our WaveAttack only achieves the third-best SSIM score on the GTSRB dataset, it outperforms BadNets by up to 60.56% in PSNR and 67.5% in IS. Similarly, although our WaveAttack achieves the second-best SSIM score on the CIFAR-100 dataset, it is much better than LIRA in PSNR and IS.



(a) BadNets (b) Blend (c) WaNet (d) Adapt-Blend (e) FTrojan (f) WaveAttack

Figure 4: The t-SNE of feature vectors in the latent space under different attacks on CIFAR-10. We use red and blue points to denote poisoned and benign samples, respectively, where each point in the plots corresponds to a training sample from the target label.

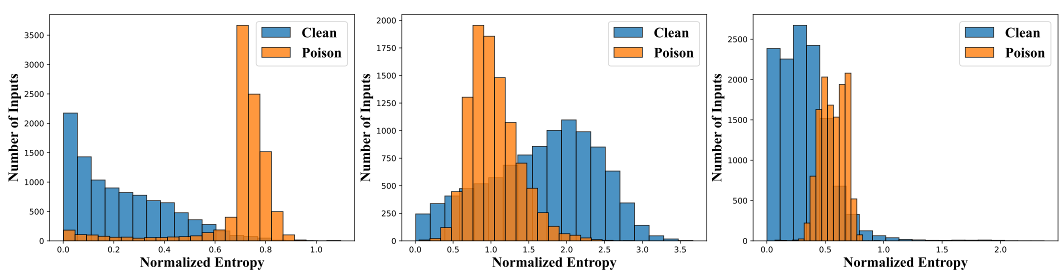
Stealthiness Results from The Perspective of Latent Space. There are so many backdoor defense methods [45, 21] based on the assumption that there is a latent separation between poisoned and benign samples in latent space. Therefore, ensuring the stealthiness of the attack method from the perspective of latent space becomes necessary. We obtained feature vectors of the test result from the feature extractor (the DNN without the last classifier layer) and used t-SNE [44] for visualization. Figure 4 visualizes the distributions of feature representations of the poisoned samples and the benign samples from the target label under the six attacks. From Figure 4(a) to 4(c) and 4(e), we can observe that there are two distinct clusters, which can be used to detect poisoned samples or backdoor models [11]. However, as shown in 4(d) and 4(f), we can find that the feature representations of poisoned samples are intermingled with those of benign samples for Adapt-Blend and WaveAttack, i.e., there is only one cluster. Adapt-Blend and WaveAttack can achieve the best stealthiness from the perspective of latent space and break the latent separation assumption to evade backdoor defenses. Although Adapt-Blend exhibits a degree of stealthiness, Table 4 reveals that WaveAttack surpasses Adapt-Blend in image quality, suggesting that WaveAttack can achieve superior stealthiness.

Table 4: Stealthiness comparison with existing attacks. Larger PSNR, SSIM, and smaller IS indicate better performance. The best and the second-best results are **highlighted** and <u>underlined</u>, respectively.

Attack Method	CIFAR-10			CIFAR-100			GTSRB			ImageNet		
	PSNR ↑	SSIM ↑	IS ↓	PSNR ↑	SSIM ↑	IS ↓	PSNR ↑	SSIM ↑	IS ↓	PSNR ↑	SSIM ↑	IS ↓
No Attack	INF	1.0000	0.000	INF	1.0000	0.000	INF	1.0000	0.000	INF	1.0000	0.000
BadNets [5]	25.77	0.9942	0.136	25.48	0.9943	0.137	25.33	**0.9935**	0.180	21.88	0.9678	0.025
Blend [17]	20.40	0.8181	1.823	20.37	0.8031	1.600	18.58	0.6840	2.118	13.72	0.1871	2.252
IAD [7]	24.35	0.9180	0.472	23.98	0.9138	0.490	23.84	0.9404	0.309	-	-	-
WaNet [8]	30.91	0.9724	0.326	31.62	0.9762	0.237	33.26	0.9659	0.170	35.18	<u>0.9756</u>	0.029
BppAttack [9]	27.79	0.9285	0.895	27.93	0.9207	0.779	27.79	0.8462	0.714	27.34	0.8009	0.273
Adapt-Blend [11]	25.97	0.9231	0.519	26.00	0.9133	0.495	24.14	0.8103	1.136	18.96	0.6065	1.150
FTrojan [10]	44.07	<u>0.9976</u>	<u>0.019</u>	44.09	0.9972	<u>0.017</u>	40.23	0.9813	<u>0.065</u>	<u>35.55</u>	0.9440	<u>0.013</u>
LIRA [39]	<u>46.77</u>	**0.9979**	<u>0.019</u>	<u>47.77</u>	**0.9995**	0.018	<u>40.44</u>	<u>0.9879</u>	0.089	-	-	-
Fiba [40]	26.08	0.9734	0.061	26.24	0.9688	0.055	23.41	0.9130	0.079	-	-	-
WaveAttack (Ours)	**47.49**	**0.9979**	**0.011**	**50.12**	<u>0.9992</u>	**0.005**	**40.67**	0.9877	**0.058**	**45.60**	**0.9913**	**0.007**

4.4 Resistance to Existing Defenses (RQ3)

To evaluate the robustness of WaveAttack against existing backdoor defenses, we implemented representative backdoor defenses (i.e., GradCAM [46], STRIP [47], Fine-Pruning [23], ANP [48] and Neural Cleanse [25]) and evaluated the resistance to them. We also show the robustness of WaveAttack against Spectral Signature [45] and other frequency detection methods [49] in the appendix.



(a) CIFAR-10 (b) CIFAR-100 (c) GTSRB

Figure 5: STRIP normalized entropy of WaveAttack.

Resistance to STRIP. STRIP [47] is a representative sample-based defense method. When entering a potentially poisoned sample into a model, STRIP will perturb it through a random set of clean samples and monitor the entropy of the prediction output. If the entropy of an input sample is low, STRIP will consider it poisoned. Figure 5 shows the entropies of benign and poisoned samples. From this figure, we can see that the entropies of the poisoned samples are larger than those of the benign samples, and STRIP fails to detect the poisoned samples generated by WaveAttack.

Resistance to GradCAM. As an effective visualization mechanism, GradCAM [46] has been used to visualize intermediate feature maps of DNN, interpreting the predictions of DNN.

Existing defense methods [50, 51] exploit GradCAM to analyze the heatmap of input samples. Specifically, a clean model correctly predicts the class label, whereas a backdoored model predicts the target label. Based on this phenomenon, the backdoored model can induce an abnormal GradCAM heatmap compared to the clean model. If the heatmaps of poisoned samples are similar to those of benign sample counterparts, the attack method is robust and can withstand defense methods based on GradCAM. Figure 6 shows the visualization heatmaps of a clean model and a backdoored model attacked by WaveAttack. Please note that here "clean" denotes a clean model trained using benign training datasets. From this figure, we can find that the heatmaps of these models are similar and that WaveAttack can resist defense methods based on GradCAM.

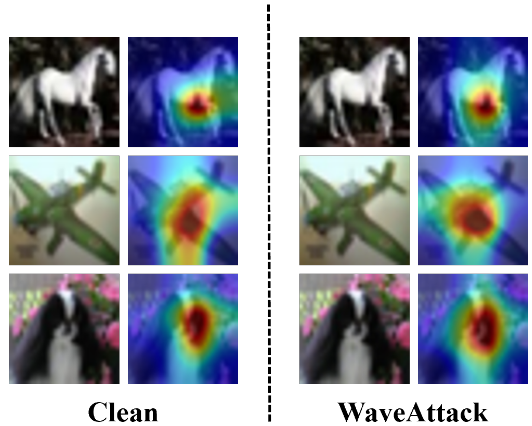


Figure 6: GradCAM visualization results for both clean and backdoored models.

Resistance to Fine-Pruning. As a representative model reconstruction defense method, Fine-Pruning (FP) [23] is based on the assumption that the backdoor can activate a few dormant neurons in DNNs. Therefore, pruning these dormant neurons can eliminate the backdoors in DNNs. To evaluate the resistance to FP, we gradually pruned the neurons of the last convolutional and fully connected layers. Figure 7 shows the performance comparison between WaveAttack and seven SOTA attack methods on CIFAR-10 by resisting FP. We find that along with more neurons being pruned, WaveAttack can acquire superior performance than other SOTA attack methods in terms of both ASR and BA. In other words, Fine-Pruning cannot eliminate the backdoor generated by WaveAttack. Note that, though the ASR and BA of WaveAttack are similar to those of Adapt-Blend at the final stage of pruning, the initial ASR (i.e., 71.57%) of Adapt-Blend is much lower than that (i.e., 100%) of WaveAttack.

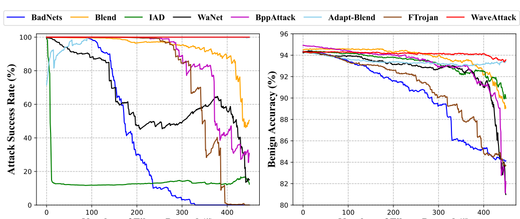


Figure 7: ASR comparison against Fine-Pruning.

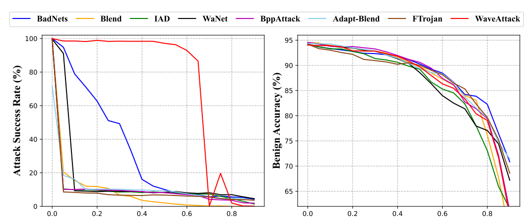


Figure 8: Attack performance comparison against ANP.

Resistance to ANP. Figure 8 compares the attack performance between WaveAttack and SOTA attack methods on the dataset CIFAR-10 against the defense method, i.e., ANP [48], where we use the threshold to denote the pruning rate of neurons. We find that as more neurons are pruned, WaveAttack consistently outperforms the other SOTA attack methods in ASR and BA.

Resistance to Neural Cleanse. As a representative defense method for trigger generation, Neural Cleanse (NC) [25] assumes that the trigger designed by the adversary is small. Initially, NC optimizes a trigger pattern for each class label via an optimization process. Then, NC uses the Anomaly Index (i.e., Median Absolute Deviation [52]) to detect whether a DNN is backdoored. Similar to the work [25], we think the DNN is backdoored if the anomaly index is larger than 2. To evaluate the resistance to NC, we conducted experiments to evaluate our WaveAttack approach by resisting NC. Figure 9 shows the defense results against NC. Please note that here, "clean" denotes clean models trained by using benign training datasets, and "backdoored" denotes backdoored models by WaveAttack that are from the Subsection 4.2. From this figure, we

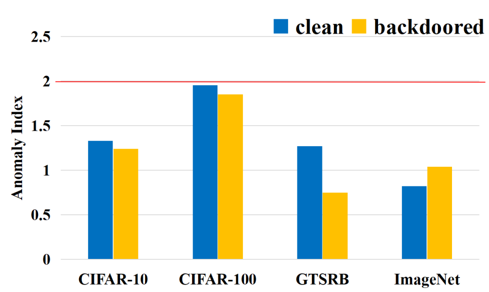


Figure 9: Defense performance against NC.

can see that the abnormal index of WaveAttack is smaller than 2 for all datasets, and WaveAttack can bypass NC detection.

Resistance to Different Frequency Filtering Methods. From Table 10, we find that WaveAttack outperforms FTrojan in both BA and ASR under two frequency filtering methods. This is mainly because FTrojan only swaps the values of two random pixels of the samples after DCT transformation, while the quality (i.e., PSNR, SSIM, and IS) of training samples after attacks is neglected.

Figure 10: Performance comparison considering different frequency filtering methods.

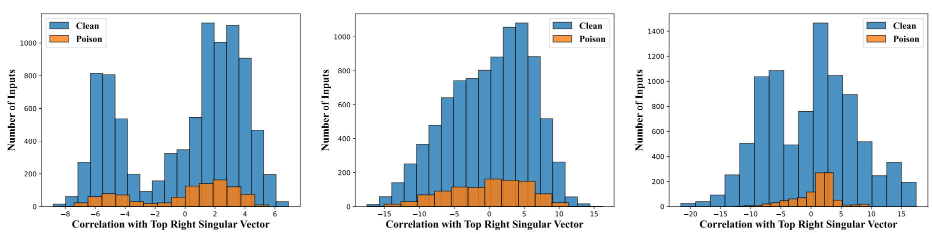
Dataset	CIFAR-10				CIFAR-100			
Methods	FTtrojan		WaveAttack		FTtrojan		WaveAttack	
Metrics	BA ↑	ASR ↑	BA ↑	ASR ↑	BA ↑	ASR ↑	BA ↑	ASR ↑
Gaussian	69.41	10.07	**72.94**	**16.72**	44.65	3.28	**47.61**	**7.92**
Wiener	66.59	12.13	**69.58**	**77.08**	41.90	6.42	**42.19**	**76.00**

Resistance to Frequency Detection Methods. Table 5 compares performance between different attack methods against the same defense method, i.e., the frequency detection method [49]. From this table, we can find that our method achieves a lower BDR than FTrojan, BppAttack, IAD, BadNets, and Blend. Note that, as studied in the experiment section, WaNet, and Adapt-Blend can be more easily detected by the latent space-based and sample-based detection methods, respectively.

Table 5: Backdoor Detection Rate (BDR) comparison against the frequency detection method.

Method	**BadNets**	**Blend**	**IAD**	**WaNet**	**BppAttack**	**Adapt-Blend**	**FTrojan**	**WaveAttack**
BDR (%)	100	97.91	96.18	0.12	96.32	1.25	78.11	**5.71**

Resistance to Spectral Signature Spectral Signature [45] is a representative latent space-based detection defense method. Given a set of benign and poisoned samples, Spectral Signature first collects their latent features and computes the top singular value of the covariance matrix. Then, for each sample, the correlation score is calculated between its features and the top singular value used as the outlier score. If the samples have high outlier scores, they will be evaluated as poisoned. We randomly selected 9000 benign samples and 1000 poisoned samples. Figure 11 shows the histograms of the correlations between latent features of the samples and the top right singular vector of the covariance matrix. From this figure, we can find that the histograms of the poisoned data are similar to those of the benign data. Therefore, Spectral Signature fails to detect the poisoned data generated by WaveAttack.



(a) CIFAR-10 (b) CIFAR-100 (c) GTSRB

Figure 11: The correlation with top right singular vector on different datasets.

5 Conclusion

Although backdoor attacks on DNNs have attracted increasing attention from adversaries, few of them simultaneously consider both the fidelity of poisoned samples and latent space to enhance the stealthiness of their attack methods. To establish an effective and stealthy backdoor attack against various backdoor detection techniques, this paper proposed a novel frequency-based method called WaveAttack, which employs DWT to extract high-frequency features from samples to generate stealthier backdoor triggers. Furthermore, we introduced an asymmetric frequency obfuscation method to improve the impact of triggers and further enhance the effectiveness of WaveAttack. Comprehensive experimental results show that, compared with various SOTA backdoor attack methods, WaveAttack not only can achieve higher stealthiness and effectiveness but also can minimize the impact of image quality on well-known datasets.

6 Acknowledgements

This work was supported by the Natural Science Foundation of China (62272170), "Digital Silk Road" Shanghai International Joint Lab of Trustworthy Intelligent Software (22510750100), and Shanghai Trusted Industry Internet Software Collaborative Innovation Center.

References

[1] Junfeng Guo, Ang Li, Lixu Wang, and Cong Liu. Policycleanse: Backdoor detection and mitigation for competitive reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4699–4708, 2023.

[2] Dennis Müller, Michael März, Stephan Scheele, and Ute Schmid. An interactive explanatory AI system for industrial quality control. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 12580–12586, 2022.

[3] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57, 2017.

[4] Tian Liu, Yunfei Song, Ming Hu, Jun Xia, Jianning Zhang, and Mingsong Chen. An ensemble learning-based cooperative defensive architecture against adversarial attacks. *Journal of Circuits, Systems and Computers*, 30(2):2150025:1–2150025:16, 2021.

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

[6] Ren Pang, Hua Shen, Xinyang Zhang, Shouling Ji, Yevgeniy Vorobeychik, Xiapu Luo, Alex X. Liu, and Ting Wang. A tale of evil twins: Adversarial inputs versus poisoned models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 85–99, 2020.

[7] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 3454–3464, 2020.

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

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

[10] Tong Wang, Yuan Yao, Feng Xu, Shengwei An, Hanghang Tong, and Ting Wang. An invisible black-box backdoor attack through frequency domain. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 396–413, 2022.

[11] Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, and Prateek Mittal. Revisiting the assumption of latent separability for backdoor defenses. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[12] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P. Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8681–8691, 2020.

[13] Qiufu Li, Linlin Shen, Sheng Guo, and Zhihui Lai. Wavelet integrated cnns for noise-robust image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7243–7252, 2020.

[14] Yingchen Yu, Fangneng Zhan, Shijian Lu, Jianxiong Pan, Feiying Ma, Xuansong Xie, and Chunyan Miao. Wavefill: A wavelet-based generation network for image inpainting. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 14094–14103, 2021.

[15] Zhisheng Zhong, Tiancheng Shen, Yibo Yang, Zhouchen Lin, and Chao Zhang. Joint sub-bands learning with clique structures for wavelet domain super-resolution. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 165–175, 2018.

[16] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–18, 2022.

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

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

[19] Haoti Zhong, Cong Liao, Anna Cinzia Squicciarini, Sencun Zhu, and David Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. In *Proceedings of the Conference on Data and Application Security and Privacy (CODASPY)*, pages 97–108, 2020.

[20] Kien Do, Haripriya Harikumar, Hung Le, Dung Nguyen, Truyen Tran, Santu Rana, Dang Nguyen, Willy Susilo, and Svetha Venkatesh. Towards effective and robust neural trojan defenses via input filtering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 283–300, 2022.

[21] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: Defending against backdoor attacks using robust statistics. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 4129–4139, 2021.

[22] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 14900–14912, 2021.

[23] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Proceedings of the Research in Attacks, Intrusions, and Defenses (RAID)*, pages 273–294, 2018.

[24] Jun Xia, Ting Wang, Jiepin Ding, Xian Wei, and Chen Mingsong. Eliminating backdoor triggers for deep neural networks using attention relation graph distillation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1481–1487, 2022.

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

[26] Zhihao Yue, Jun Xia, Zhiwei Ling, Ming Hu, Ting Wang, Xian Wei, and Mingsong Chen. Model-contrastive learning for backdoor elimination. In *Proceedings of ACM Multimedia*, pages 8869–8880, 2023.

[27] Mark J Shensa et al. The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Transactions on signal processing*, 40(10):2464–2482, 1992.

[28] Ingrid Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, 1990.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention (MICAI)*, pages 234–241, 2015.

[30] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[31] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. In *Citeseer*, 2009.

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

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

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

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

[36] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141, 2018.

[37] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017.

[38] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[39] Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 11966–11976, 2021.

[40] Yu Feng, Benteng Ma, Jing Zhang, Shanshan Zhao, Yong Xia, and Dacheng Tao. Fiba: Frequency-injection based backdoor attack in medical image analysis. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, pages 20876–20885, 2022.

[41] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 44(13):800–801, 2008.

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

[43] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

[44] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

[45] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 8011–8021, 2018.

[46] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, 2020.

[47] Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith Chinthana Ranasinghe, and Surya Nepal. STRIP: a defence against trojan attacks on deep neural networks. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, pages 113–125, 2019.

[48] Dong Huang and Qingwen Bu. Adversarial feature map pruning for backdoor. In *The Twelfth International Conference on Learning Representations*.

[49] Yi Zeng, Won Park, Z. Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16473–16481, October 2021.

[50] Edward Chou, Florian Tramèr, Giancarlo Pellegrino, and Dan Boneh. Sentinet: Detecting physical attacks against deep learning systems. *arXiv preprint arXiv:1812.00292*, 2018.

[51] Bao Gia Doan, Ehsan Abbasnejad, and Damith C Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, pages 897–912, 2020.

[52] Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974.

7 Appendix

7.1 Implementation Details for Experiments

Settings of Datasets. Table 6 presents the setting of datasets used in our experiments.

Table 6: Datasets Settings.

Dataset	Input Size	Classes	Training Images	Test Images
CIFAR-10	$3\times32\times32$	10	50000	10000
CIFAR-100	$3\times32\times32$	100	50000	10000
GTSRB	$3\times32\times32$	43	26640	12569
ImageNet subset	$3\times224\times224$	20	26000	1000

Settings of Attacks. For a fair comparison, the settings of WaveAttack are consistent with those of the other seven SOTA attack methods. We used the SGD optimizer for training a classifier with a learning rate of 0.01, and the Adam optimizer for training a generator with a learning rate of 0.001. We decreased this learning rate by a factor of 10 after every 100 epochs. We considered various data augmentations, i.e., random crop and random horizontal flipping. For BadNets, we used a grid trigger placed in the bottom right corner of the image. For Blend, we applied a "Hello Kitty" trigger on CIFAR-10, CIFAR-100, and GTSRB datasets and used random noises on the ImageNet dataset. For other attack methods, we used the default settings in their respective papers.

7.2 Broader Impacts and Limitations

Broader Impacts. In this work, we introduce a new effective and stealthy backdoor attack method named WaveAttack, which can stealthily compromise security-critical systems. If used improperly, the proposed attack method may pose a security risk to the existing DNN applications. Nevertheless, we hope that by emphasizing the potential harm of this malicious threat model, our work will stimulate the development of stronger defenses and promote greater attention from experts in the field. As a result, this knowledge promotes the creation of more secure and dependable DNN models and robust defensive measures.

We would like to emphasize that our paper mainly focuses on introducing and evaluating the attack method. This paper aims to develop more powerful detection and defence mechanisms against such advanced backdoor attacks by proposing more advanced backdoor attack methods and addressing the weaknesses of state-of-the-art defence methods in future works.

Limitations. Although our work shows exciting results for backdoor attacks, it requires more computing resources and runtime overhead than most existing backdoor attack methods due to the necessity of training a generator g to generate residuals of the various high-frequency components. Moreover, we do not consider a threat model, in which the adversary can only control the training dataset. In this threat model, we used our pre-trained generator to modify some benign samples in the training dataset. However, this limitation also appears in [11]. In the future, we plan to explore more effective and stealthy backdoor attack methods under this threat model.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The answer to this question can be found in the abstract and the experiments in this paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: The limitation can be found in the appendix of this paper.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: This answer can be found in the experimental results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We upload all the code to GitHub and put it in an appendix file so that the reader can reproduce the results of this paper.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We upload all the code to GitHub and put it in an appendix file so that the reader can reproduce the results of this paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This answer can be found in the experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The answer can be found in the appendix of this paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The answer can be found in the appendix of this paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The answer can be found in the appendix of this paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.