Check Your Other Door! Creating Backdoor Attacks in the Frequency Domain

Hasan Abed Al Kader Hammoud
hasanabedalkader.hammoud@kaust.edu.sa
Bernard Ghanem
bernard.ghanem@kaust.edu.sa

King Abdullah University of Science and Technology (KAUST)
Thuwal, Saudi Arabia

Abstract

Deep Neural Networks (DNNs) are ubiquitous and span a variety of applications ranging from image classification to real-time object detection. As DNN models become more sophisticated, the computational cost of training these models becomes a burden. For this reason, outsourcing the training process has been the go-to option for many DNN users. Unfortunately, this comes at the cost of vulnerability to backdoor attacks. These attacks aim to establish hidden backdoors in the DNN so that it performs well on clean samples, but outputs a particular target label when a trigger is applied to the input. Existing backdoor attacks either generate triggers in the spatial domain or naively poison frequencies in the Fourier domain. In this work, we propose a pipeline based on *Fourier heatmaps* to generate a spatially dynamic and invisible backdoor attack in the frequency domain. The proposed attack is extensively evaluated on various datasets and network architectures. Unlike most existing backdoor attacks, the proposed attack can achieve high attack success rates with low poisoning rates and little to no drop in performance while remaining imperceptible to the human eye. Moreover, we show that the models poisoned by our attack are resistant to various state-of-the-art (SOTA) defenses, so we contribute two possible defenses that can evade the attack.

1 Introduction

Deep neural networks (DNNs) play a crucial role in various applications such as facial recognition systems [38], medical image analysis [31], autonomous driving [42], among others [15, 24]. As the tasks become more difficult, the need for more sophisticated and complex models arises. Such models are generally harder to train and might require extensive hyperparameter tuning to achieve the required performance. Recently, and due to the limited access to computational power for most individuals and small companies, *outsourced training* and the use of out-of-the-box pre-trained models became popular [36].

Outsourced training creates a set of serious vulnerabilities, as it involves several stages that the outsourcer could exploit, including data collection, data pre-processing, and model deployment [13, 16, 28, 33]. An important threat that could be exploited during training is called *a backdoor attack*. Backdoor attacks create an association between an attacker-defined pattern, called the trigger, and a chosen target label in such a way that the malicious actor can instigate the trigger at will without degrading the model's performance on clean samples.

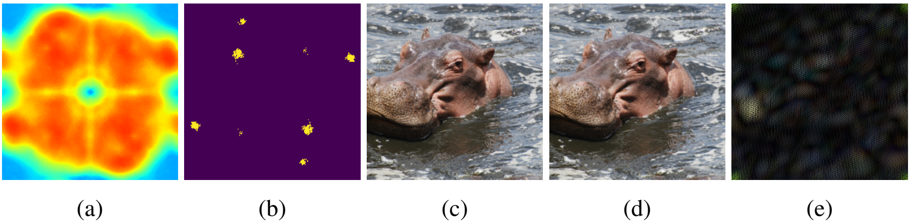


Figure 1: **Backdoor Attacks in the Frequency Domain.** Frequency-based backdoor attacks exploit the frequency sensitivity of a network, *i.e* the sensitivity of its performance to variations in individual frequency components in the Fourier domain. Our proposed attacks focus on poisoning the most sensitive frequencies. (a) ResNet50's sensitivity Fourier heatmap (red regions are highly sensitive, while blue regions are less sensitive); (b) Top-*k* selected frequencies, into which backdoor attacks are embedded; (c) Clean image; (d) Poisoned image; (e) Scaled absolute difference (×20) between the poisoned and clean images.

This association is usually created through *training data poisoning* [2, 16, 28, 33], where the adversary applies a trigger to a set of images from the training set and then switches their ground truth label to a chosen target class before model training begins.

Most existing backdoor attacks [1, 5, 16, 26, 33, 35, 49, 51, 56, 57] rely on the spatial domain to generate and embed the trigger. For example, [16] applies a white square stamp on the corner of some training images to poison the data. Other methods such as [33] rely on an optimization-based approach to generate optimal trigger values. These attacks experience a sharp trade-off between the amount of poisoned data, the invisibility and success of the attack, and the performance of the model on the original task. On the other hand, most backdoor defenses rely on the spatial domain or properties of this domain to detect and mitigate attacks [12, 17, 39, 47]. Since most backdoor attack techniques tend to be visible and static, *i.e* the same spatial trigger is applied to all poisoned images, defense techniques in the spatial domain, such as reversed trigger construction [17, 47] and fine-pruning [32], easily succeed in detecting, reverse engineering, and mitigating the embedded backdoor trigger.

Recently, [11, 48] have proposed creating backdoor attacks in the frequency domain. However, both proposed attacks naively select the frequency components to poison.

Contributions. Given the weaknesses associated with developing backdoor attacks in the spatial domain and the limitations of existing frequency attacks, in this work, we propose a backdoor attack that utilizes Fourier heatmaps to design a sophisticated backdoor poisoning attack in the frequency domain. Unlike previous attacks, our frequency-based attack does not face the aforementioned trade-offs. We also show two potential ways to defend against frequency-based backdoor attacks and possible ways for the attacker to bypass these defenses. The proposed method is extensively evaluated on multiple models and datasets.

2 Related Work

Backdoor Attacks. Backdoor attacks were first introduced in [16] as a possible security breach that could be exploited in DNNs. They showed that adding a simple patch to the corner of a subset of the training images creates a backdoor that could be maliciously triggered to output a predefined target label. Later, several works were introduced, such as [33], where the values of a predefined mask were optimized to obtain an optimal trigger. On another track, [5] realized the importance of having invisible or imperceptible triggers to evade possible human inspection. The authors proposed blending the backdoor trigger and the clean images together, replacing the previously used stamping technique. Along these lines, other invisible attacks were proposed, such as [29] which used least-significant bit (LSB) algorithm from the steganography literature to generate an invisible attack, [37] which utilized image warping to poison data samples, and [56] which proposed having input-aware trigger patterns that poison the edges of the image. [10] highlighted the importance of learning the trigger-generating transformation to achieve a high attack success rate. [9] proposed utilizing the latent space representation to generate imperceptible backdoor triggers by minimizing the Wasserstein distance between the representations of clean and poisoned samples. [27], also inspired by steganography, generated sample-specific triggers by encoding an attacker-specified "string" into clean samples using an autoencoder network. [54] analyzed the characteristics of spatial backdoor attacks in the frequency domain and proposed a technique to create smooth but visible spatial backdoor triggers. Recently, [11, 48] introduced a simple way to apply trojan attacks in the frequency domain. Specifically, [11] blends the low-frequency content of a trigger image with that of the target images, and [48] arbitrarily poisons a high-frequency component and a mid-frequency component.

Contribution. Our work adds to the literature an invisible frequency backdoor attack. Unlike existing frequency backdoor attacks [11, 48], our attack poisons the data by altering *well-chosen* frequency components based on the model's frequency sensitivity.

Backdoor defenses. Early defense mechanisms such as fine-pruning [32] relied on neuron activations to mitigate backdoors embedded in a DNN. In particular, pruning the least active neurons on clean images and then fine-tuning the model on clean samples can reverse the backdoor attack. [46] and [3] used robust statistics and analysis of neural network activations, respectively, to thwart and detect backdoor attacks. Later, more sophisticated optimization-based methods, such as Neural Cleanse (NC) [47], TABOR [17] and ABS [34], were developed to mitigate backdoor attacks. NC computes an anomaly index, which indicates whether an abnormally short distance exists between a particular class and all other classes. If the anomaly index exceeds a threshold, NC finds a reverse engineered trigger that is used to fine-tune the model on poisoned but correctly labeled samples. [8] relied on computing class activation maps using Grad-CAM [44] to find the regions the network is attending to in hopes of detecting the attacker-triggered region, which is then replaced through image restoration. [59] adopted persistent homology from topological data analysis to discover structural abnormalities in poisoned models. TOP [23] showed that adversarial perturbations transfer better from image to image in poisoned models compared to clean ones, which can be used to detect poisoned models. STRIP [12] observes that when a poisoned image is blended with a clean one, the backdoor is still activated, which allows for detecting backdoor attacks by analyzing the entropy of the prediction vectors. SPECTRE [18] explores robust covariance estimation to amplify the spectral signal *i.e* the signature of poisoned data.

Contribution. Our work proposes two defenses that alter the frequency spectrum of the input, to mitigate the adverse effects of frequency-based backdoor attacks.

3 Preliminaries

To clearly detail our proposed frequency-based approach, we briefly review the concept of *Fourier heatmaps* that was first introduced in [52]. Fourier heatmaps provide a tool for

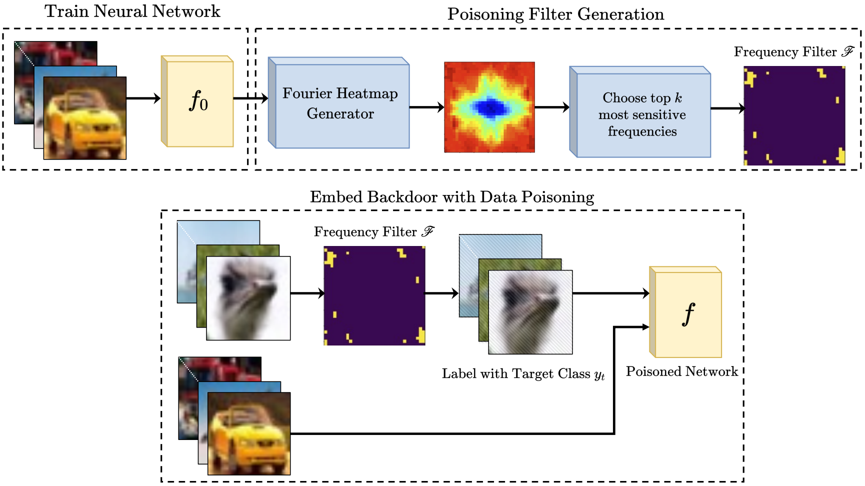

Train Neural Network
f_0
Poisoning Filter Generation
Fourier Heatmap Generator
Choose top k most sensitive frequencies
Frequency Filter $\mathscr{F}$
Embed Backdoor with Data Poisoning
Frequency Filter $\mathscr{F}$
Label with Target Class y_t
f
Poisoned Network

Figure 2: **Pipeline.** We illustrate the pipeline for our proposed frequency-based data poisoning method. After training a network naturally, the Fourier heatmap for this model is generated and the top-k most sensitive frequencies are selected as a poisoning filter. This filter is then used to poison a subset of the training dataset before training the poisoned model.

analyzing the sensitivity of a DNN to a specific Fourier frequency basis by analyzing how this DNN performs when subjected to input perturbations in this basis [52].

Notation. We denote the 2D Discrete Fourier Transform of an image $X \in \mathbb{R}^{d_1 \times d_2}$ by $\mathcal{F} : \mathbb{R}^{d_1 \times d_2} \to \mathbb{C}^{d_1 \times d_2}$ and its inverse by $\mathcal{F}^{-1}$ (both operations are applied per channel). By default, we assume that the frequency components are shifted toward the center of the Fourier spectrum, *i.e* low frequencies are set about the center.

2D Fourier Basis. Let $\mathcal{U}_{i,j}$ be a real valued matrix in $\mathbb{R}^{d_1 \times d_2}$ with the following properties. **(1)** It has a Frobenius norm $\left\|\mathcal{U}_{i,j}\right\|_F = 1$; **(2)** $\mathcal{F}(\mathcal{U}_{i,j})$ has up to two non-zero elements located at (i, j) and its conjugate symmetric component (symmetric relative to the origin of the spectrum). We refer to such a matrix $\mathcal{U}_{i,j}$ as a 2D Fourier basis at (i, j).

Fourier Heatmaps. We denote a batch of B images as $\mathcal{I}$, the Fourier basis perturbation factor by α, and a uniformly and randomly sampled matrix from $\{-\mathbb{1}, \mathbb{1}\}$ by $\mathbf{r}$, where $\mathbb{1}$ is the matrix of all ones in $\mathbb{R}^{d_1 \times d_2}$. Let $\tilde{\mathcal{I}}$ denote the perturbed batch of images, where $\tilde{\mathcal{I}} = \mathcal{I} + \alpha(\mathbf{r} \odot \mathcal{U}_{i,j})$, $\odot$ is the Hadamard product. Note that the addition is performed across all channels of images in the batch. To measure the sensitivity of a classification DNN to the frequency basis at (i, j), we forward pass the perturbed batch $\tilde{\mathcal{I}}$ through the DNN and compute its output error rate w.r.t. the ground truth image labels for the specified (i, j) basis. When repeated for all (i, j) pairs, we can visualize the DNN's sensitivity to all 2D Fourier bases through a matrix denoted as a Fourier heatmap [52] (see Figure 1a for an example).

4 Proposed Method

Following [4, 16, 33, 41, 50, 58], we consider the threat model, in which the victim outsources the training process to a trainer that has access to: **(1)** the victim's network architecture and **(2)** their training dataset. The victim accepts the model provided by the adversary if its classification accuracy on the validation set is satisfactory.

Now we provide a detailed explanation of the proposed frequency-based backdoor attack pipeline. As explained in Section 3, Fourier heatmaps provide a tool for analyzing the sensitivity of a DNN to input perturbations in particular 2D Fourier bases. Knowledge of the network's sensitive frequencies allows the attacker to design an attack that exploits these frequencies to embed a frequency-based backdoor that maintains a good performance on the original classification task, embeds a strong backdoor trigger that activates the target class at will, and is both invisible and achievable with small poisoning rates. Figure 2 visualizes the proposed pipeline. Below we summarize the recipe for creating frequency-based backdoors.

Stage 1: Poisoning Filter Generation: **1.** Train a neural network on the clean dataset and the architecture provided by the user. We denote this clean network by f_0. **2.** Generate the Fourier heatmap for f_0 and store the indices of the top-k most sensitive frequencies, $\mathbb{I}_k$, and then generate a binary mask $\mathcal{M}$ as shown in equation 1. **3.** Generate three additive frequency masks one for each channel ($\mathcal{A}_R$, $\mathcal{A}_G$ and $\mathcal{A}_B$) as shown in equation 2. The values for additive masks $\mathcal{A}_{i,j}$ for $(i,j) \in \mathbb{I}_k$ should be selected to satisfy the invisibility requirement at hand (discussed later in Section 5.2). For a simple yet flexible design, we set the nonzero values in any individual additive mask to be the same, but different from one mask to another.

$$\mathcal{M}_{i,j} = \begin{cases} 1 & (i,j) \in \mathbb{I}_k \\ 0 & \text{otherwise} \end{cases} \quad (1) \qquad \mathcal{A}_{\{R,G,B\}_{i,j}} \begin{cases} \neq 0 & (i,j) \in \mathbb{I}_k \\ = 0 & \text{otherwise} \end{cases} \quad (2)$$

Stage 2: Creating the Backdoor through Data Poisoning: **1.** Specify a set of samples to poison and denote it by $\mathcal{I}_P$. The cardinality of $\mathcal{I}_P$ is denoted by $|\mathcal{I}_P|$ and refers to the number of poisoned samples. The poisoning rate is defined as the ratio of the number of poisoned samples to the total number of samples in the training set. **2.** For each sample $\mathcal{S} \in \mathcal{I}_P$, and for each channel, apply the following operations:

$$\mathcal{S}_{\{R,G,B\}} := \mathcal{F}^{-1}(\mathcal{F}(\mathcal{S}_{\{R,G,B\}}) \odot (\mathbb{1} - \mathcal{M}) + \mathcal{A}_{\{R,G,B\}}) \quad (3)$$

where each channel is treated separately. **3.** Change the label of the samples in $\mathcal{I}_P$ to the specific target label t. **4.** Proceed with training the neural network on the poisoned training dataset to obtain a backdoored or poisoned model f.

It should be noted that the operations carried out on the Fourier transformed channels could be thought of as simply changing the values of the components of the top-k most sensitive 2D Fourier bases by different values that carry the poisoning information. This could be thought of as a frequency-based version of spatial trigger stamping. Section 5.5 discusses the importance of choosing the top-k values rather than random or bottom-k elements. The supplementary material contains variants of the proposed method. It includes experiments, where additive masks have **(1)** varying random values for each channel and **(2)** the same values across all channels. We also consider adopting a binary mask ($\mathcal{M}$) generated for one architecture and applying it as a poisoning mask for another. Additionally, we discuss two possible variations of the pipeline that **(1)** extend the applicability of our attack to the multi-target attack regime; **(2)** allow for an efficient end-to-end frequency backdoor attack.

5 Experiments

In this section, we present the details of our implementation and experiments to evaluate our proposed attack mechanism on various datasets and network architectures. Afterwards, we evaluate our attacked models against three state-of-the-art defenses (three more are found in the supplementary). Finally, we show two defenses against frequency-based backdoor attacks and potential ways for the attacker to defend against them.

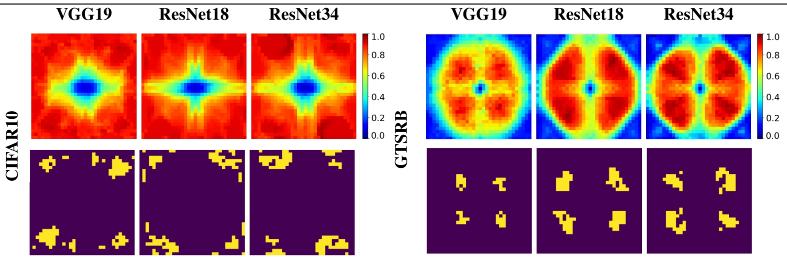


Figure 3: **Fourier Heatmaps and Top-k Masks.** Rows 1 shows the heatmaps of various architectures trained on CIFAR10 and GTSRB, respectively. Rows 2 shows the respective binary mask ($\mathcal{M}$), which defines the k most sensitive frequencies in the heatmap.

5.1 Implementation Details

Following [10, 37, 43, 56] we evaluate our attack on various datasets, network architectures, and poisoning rates.

Datasets. We evaluate our proposed pipeline on commonly used datasets: CIFAR10 [25], CIFAR100 [25], GTSRB [20], and ImageNet [40]. ***Network Architectures.*** We study six network architectures of different complexity: ResNet18, ResNet34, ResNet50 [19], DenseNet121 [22], VGG19 [45], and WideResNet34 [53]. ***Network Performance Metrics.*** To evaluate the performance of backdoored models, we use two common metrics: Clean Data Accuracy (CDA), which measures the performance of the network on clean samples, and Attack Success Rate (ASR), which measures the effectiveness of the backdoor attack in triggering the target label. ***Invisibility Metrics.*** Following other papers [21, 29, 35, 37, 48, 56], we evaluate the invisibility of the proposed attack using three metrics: Peak Signal-to-Noise-Ratio (PSNR), Structural SIMilarity (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS). Invisibility is a crucial metric for backdoor attacks, as it is required to fool any possible human inspection that may detect the applied trigger.

	Poisoning Rate	CDA(%)	ASR(%)
CIFAR10	0.0%	93.92	-
	0.1%	94.00	1.54
	0.2%	94.14	72.31
	0.4%	94.20	85.05
	1.0%	94.38	99.44
	3.0%	94.31	99.79
CIFAR100	0.0%	75.95	-
	0.1%	75.76	60.57
	0.2%	75.75	92.78
	0.4%	75.92	96.49
	1.0%	76.05	98.99
	3.0%	75.36	99.93
GTSRB	0.0%	97.11	-
	0.1%	97.09	71.12
	0.2%	97.19	89.59
	0.4%	97.33	98.04
	1.0%	97.25	98.62
	3.0%	97.47	99.80
ImageNet	0.0%	67.51	-
	0.5%	67.38	0.17
	1.0%	67.13	87.74
	2.0%	67.26	98.01
	3.0%	67.26	98.32

Table 1: **Evaluation of the proposed backdoor attack.** We benchmark our proposed attack for ResNet18 trained on various datasets and poisoning rates. Our attack can maintain CDA, while registering high ASR even with small poisoning rates *(full table in suppl.)*.

5.2 Frequency-Based Backdoor Attacks

Backdoored Network Performance. As discussed in Section 4, we first train baseline networks on each dataset and compute the corresponding Fourier heatmaps and binary masks. The accuracies of the baseline models (f_0) are shown in Table 6 (0% Poisoning Rate). The heatmaps and masks for various architectures trained on CIFAR10 and GTSRB are shown in Figure 3, respectively. The remaining filters and heatmaps are provided in the supplementary material. In our experiments, the choice of k, which defines the number of nonzero indices of $\mathcal{M}$ and the corresponding values for the additive masks $\mathcal{A}_{\{R,G,B\}}$, is made such that: **(1)** the ℓ_2 norm of the attack (*i.e* the ℓ_2 norm of the absolute difference of the image before and after poisoning) does not exceed, on average, a threshold δ_P ($\delta_P = 2.0$ for ImageNet and $\delta_P = 1.0$ for all other datasets), and **(2)** the invisibility metrics (PSNR, SSIM, LPIPS) reach satisfactory values. Table 6 shows the CDA of the backdoored model (f) and the ASR of frequency-based triggers for CIFAR10, CIFAR100, GTSRB, and ImageNet and for ResNet18 with different poisoning rates. Similar to [56], we also highlight the effect of changing the poisoning rate on the CDA and ASR metrics. As observed, even with a low poisoning rate, we can embed a backdoor attack with a high ASR with little or no drop in CDA. The target label was arbitrarily chosen as the first class of each dataset. Since the datasets are class-balanced, any target label will lead to a similar performance.

Table 20 compares our method with existing spatial and frequency backdoor attacks. The results for SIG, Refool, SPM, and Poison Ink are taken from [56]. Our frequency-based backdoor attack achieves SOTA results in almost all scenarios considered. Note that the training setup adopted to generate our results is the same for all other methods. A further comparison with other backdoor attacks is provided in the supplementary.

Invisibility of the Proposed Attack. Table 2 compares our proposed frequency-based backdoor attack with other attacks based on their invisibility metrics (PSNR,SSIM,LPIPS). The results of the other methods are taken from [56] (except for [11, 48]). Our proposed attack achieves the highest PSNR and SSIM, and the lowest LPIPS compared to other backdoor attacks. The PSNR of our method could be further improved at the cost of ASR by selecting fewer frequencies to poison; however, the invisibility metrics (PSNR, SSIM, LPIPS) "saturate" beyond a certain point where further improvements become insignificant and unneeded.

Method	**PSNR↑**	**SSIM↑**	**LPIPS↓**
BadNets [16]	27.03	0.9921	0.0149
Blend [8]	19.18	0.7291	0.2097
SIG [1]	25.12	0.8988	0.0532
Refool [35]	16.59	0.7701	0.2461
SPM [30]	38.65	0.9665	0.0022
Poison Ink [56]	41.62	0.9915	0.0020
FTrojan [48]	44.87	0.9942	0.0005
FIBA [11]	18.05	0.8077	0.1113
Ours (ResNet18)	47.26	**0.9998**	0.0006
Ours (ResNet34)	47.55	**0.9998**	0.0004
Ours (ResNet50)	46.90	**0.9998**	0.0009
Ours (DenseNet121)	47.21	**0.9998**	**0.0001**
Ours (VGG19)	46.19	**0.9998**	0.0008

Table 2: **Comparing Invisibility Metrics of Backdoor Attacks on ImageNet.** Our attack achieves the best invisibility scores compared to other existing methods.

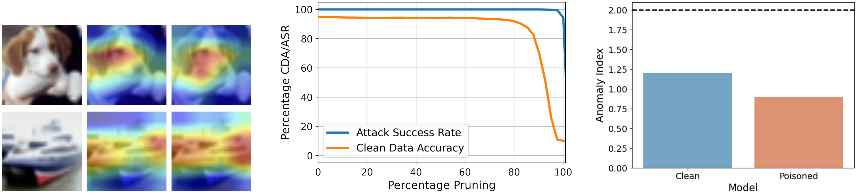
5.3 Evaluation Against Backdoor Defenses

We evaluate our attacked models against three SOTA backdoor defenses, namely, Neural Cleanse [47], Grad-CAM [44], and Pruning [32]. Being invisible and dynamic in the spatial domain, frequency-based backdoor attacks can easily evade SOTA defenses. The results of the three defenses against our attacked ResNet18 model trained on CIFAR10 with 1% poisoning rate are shown in Figure 4. Figure 4a shows the Grad-CAM [44] results for two images and their backdoor attacked versions using our frequency-based approach. Grad-CAM uses gradients of a particular class to visualize where the network is looking/focusing at to make its prediction. As shown in Figure 4a, our frequency-based backdoor attacks do not introduce an observable change in the "attention" of the network. For each of the two

Table 3: **Comparison between the Proposed Attack and Backdoor Attacks in the Literature.** Our frequency-based attack achieves SOTA ASR, CDA, PSNR, and LPIPS metrics. The results shown are for VGG19 trained on CIFAR10. Legend: **First Best**, Second Best

Metric	**Ratio**	**SIG**	**Refool**	**SPM**	**WaNet**	**FIBA**	**FTrojan**	**Poison Ink**	**Ours**
CDA/ASR	3%	89.74 / 99.23	89.20 / 87.16	88.89 / 58.53	91.86 / 32.86	90.92 / 90.10	91.31 / **99.99**	89.65 / 94.22	**92.31** / 99.43
	5%	89.64 / 99.47	89.16 / 89.79	88.90 / 57.69	91.47 / 88.15	90.69 / 95.06	91.64 / 99.10	89.69 / 93.58	**91.88 / 99.88**
	10%	89.45 / 99.40	88.80 / 92.80	89.07 / 57.33	91.22 / 96.96	90.41 / 95.86	90.93 / **100.00**	89.47 / 93.67	**92.10** / 99.97
PSNR↑/LPIPS↓		25.12 / 0.0400	19.38 / 0.0397	38.94 / 0.0001	31.53 / 0.0047	19.40 / 0.0180	41.01 / 0.0001	42.95 / 0.0001	**43.15 / 0.00001**

Figure 4: **Evaluation of defenses:** Evaluation of various SOTA defenses against the proposed frequency-based attack shows the power of the proposed method in evading the defenses. (a) Grad-CAM shows high similarity in the attention regions for poisoned and non-poisoned models; (b) Pruning the poisoned model maintains high ASR even after significant drop in CDA. (c) Neural Cleanse anomaly indices fall below the anomaly threshold (2.0).



(a) Grad-CAM [44] (b) Pruning [32] (c) Neural Cleanse [47]

samples presented (first column), we compute the Grad-CAM by passing the clean samples into the clean network (f_0) (middle column), and then show the Grad-CAM for passing the poisoned samples into the backdoored model (f) (third column).

Since the network still focuses on the same parts of the input image, methods like Februus [8] fail to remove the embedded backdoor, as observed by [56]. Figure 4b shows the performance of our attack against the pruning defense in [32], which prunes the least active neurons (on clean samples) and then fine-tunes the network on clean samples.

We see that pruning our backdoored model does not eliminate the backdoor. This is mainly attributed to the fact that frequency-based poisoning is of low norm and therefore gets embedded into most weights of the network rather than hidden into particular neurons. Figure 4c shows the anomaly index computed by Neural Cleanse [47] for both the baseline and our backdoored/poisoned model. Since the anomaly index of the poisoned model is less than the anomaly index threshold defined by Neural Cleanse (2.0), Neural Cleanse fails to detect that the frequency-based backdoored model is actually poisoned. Further evaluation of these defenses and evaluation of additional defenses, namely, STRIP [12], Spectral Sig-

		JPEG		**Autoencoder**		**JPEG+Autoencoder**	
	Poisoning Rate	**CDA**	**ASR**	**CDA**	**ASR**	**CDA**	**ASR**
CIFAR10	0.1%	94.19	1.76	93.73	0.22	94.65	0.66
	0.2%	94.37	18.02	94.38	22.86	94.22	3.08
	0.5%	93.94	83.52	94.17	73.85	94.49	36.48
	1.0%	94.28	96.48	94.61	93.63	94.24	90.11
	3.0%	94.26	99.34	94.13	98.90	94.32	98.46
CIFAR100	0.1%	76.57	14.26	76.19	14.06	76.05	2.57
	0.2%	77.14	75.25	75.96	83.76	75.40	32.08
	0.5%	75.86	95.25	76.07	94.06	76.35	95.05
	1.0%	75.43	99.21	75.57	97.82	76.16	96.83
	3.0%	75.07	99.80	76.26	99.54	75.51	98.81
GTSRB	0.1%	97.27	52.46	97.45	69.55	96.97	48.13
	0.2%	96.79	74.07	97.39	81.14	97.09	73.87
	0.5%	97.25	90.18	97.14	94.50	96.84	95.09
	1.0%	94.34	86.44	97.00	99.02	95.56	94.89
	3.0%	93.72	98.43	97.25	99.78	92.99	97.64

Table 4: **Augmentation Defense:** CDA and ASR of backdoored ResNet18 trained on various datasets with JPEG compression and Autoencoder augmentation. The ASR and CDA are maintained even when no preprocessing technique is used.

natures [46], and Activation Clustering [3] on different models and datasets is provided in the supplementary.

5.4 Defenses Against Frequency-based Backdoors

Since the additive mask values could be *arbitrarily* chosen, a simple inspection of the Fourier transforms of the input may not be successful in detecting the poisoned samples. Therefore, we discuss two possible ways to defend against frequency backdoor attacks.

For a successful defense, the defender should manipulate the frequency spectrum of the input images to break the backdoor trigger while maintaining a satisfactory CDA. We show that this is possible using two techniques: (1) passing the image through an autoencoder and (2) compressing the image. These two methods are used in the robustness literature and have proven to be useful in protecting DNNs from adversarial attacks [6, 7]. Autoencoders have also been used as a preprocessing mechanism to disable backdoor triggers [36]. Applying an autoencoder trained on CIFAR10 can almost completely deactivate the embedded frequency backdoor. A similar effect is observed for compression, where the ASR of the backdoored model drops to almost 0% after 20% of JPEG compression.

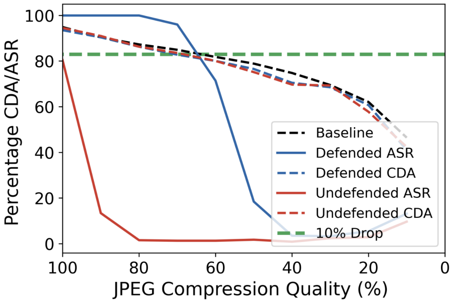


Figure 5: **Defending with JPEG Augmentation.** Training on JPEG compressed images maintains a high ASR even after a drop of 10% in CDA. The baseline denotes the CDA of the baseline model evaluated on compressed images.

A possible solution to bypass both of these defenses is to apply a technique similar to adversarial training [14, 55]. The attacker can train on compressed and/or auto-encoded versions of the poisoned images. This augmentation translates to embedding multiple versions of the backdoor into the model. Figure 5 shows the ASR and CDA for both an undefended poisoned model and a defended one. For the undefended model, *i.e* no augmentation, the backdoor immediately breaks down as compression is applied. On the other hand, the defended model can maintain an $ASR > 80\%$ even beyond 25% compression, where the CDA drops by 10%. Finally, we note that the above augmentations still allow us to reach a high ASR with a minimal drop in CDA for our backdoored models. Therefore, if the defender does not set a defense mechanism, the backdoor still functions properly. The results for ResNet18 trained on CIFAR10, GTSRB, and CIFAR100 with different augmentations are shown in Table 4. The results for other models and datasets are presented in the supplementary material.

5.5 Ablation Study

We study the effect of choosing **(i)** random frequencies and **(ii)** bottom-k, *i.e* least sensitive frequencies, as compared to choosing the top-k frequencies from the Fourier heatmap. Table 5 shows the results of poisoning a ResNet18 trained on ImageNet using two different random

	Poisoning Rate	**1%**	**2%**
Random (1)	**CDA(%)/ASR(%)**	67.24/53.91	66.83/60.49
Random (2)		67.23/56.88	66.80/66.11
Bottom-k (1)		67.03/22.58	66.80/55.96
Bottom-k (2)		67.04/0.31	67.02/92.81
Top-k (1)		67.13/87.74	67.26/98.01

Table 5: **Effect of Different Frequency Selection Schemes:** Results for frequency filters generated using least sensitive, most sensitive and random frequencies. Choosing the top-k most sensitive frequencies provides the highest ASR among those options.

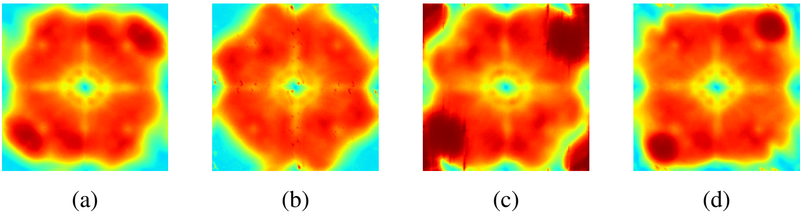


Figure 6: **Heatmaps of Ablated Frequency Selection:** Fourier heatmaps of frequency-based backdoor attacks with different frequency selection schemes: (a) Clean Model; (b) Random Frequency Selection; (c) Bottom-*k* Frequency Selection; (d) Proposed Top-*k* Frequency Selection.

filters and two different bottom-*k* filters (two different values were chosen for *k* to control the PSNR), where the runs for a particular scheme are numbered in brackets. The random filters were generated using Bernoulli trials with $p = 0.005$ at each Fourier basis (Random (1): PSNR = 47.62/ Random (2): PSNR = 46.62). Bottom-*k* filters were generated by selecting the *k*-least sensitive frequencies (Bottom-*k* (1): PSNR = 51.23 /Bottom-*k* (2): PSNR = 31.23). In general, bottom-*k* and random frequencies contain low frequency components, which greatly affect the invisibility of the attack.

One can see the importance of choosing top-*k* frequencies over the other two options, as it leads to a high ASR at a small poisoning rate while maintaining a high PSNR. This is attributed to the fact that the network relies on the most sensitive frequencies to perform the classification task at hand. Therefore, embedding a backdoor attack into the most sensitive frequencies allows the network to learn the backdoor trigger with little effort, compared to other frequency selection schemes.

Finally, an interesting observation can be made by looking at the Fourier heatmaps of these models. Figure 6 visualizes the Fourier heatmaps for Random (2), Bottom-*k* (2), and Top-*k* models. We can see a significant explosion in frequency sensitivity in the case of selecting the bottom-*k* components and “chicken-pox" like sensitivity for the random frequency selection (dotted in the positions of randomly sampled frequency bases). Our method of using the top-*k* most sensitive frequencies is more conservative in introducing modifications to the network’s clean heatmap; however, it also experiences mild “sensitivity leakage" at certain frequencies. The supplementary shows the Fourier heatmaps for other backdoor attacks and provides a discussion about detecting backdoor attacked models using Fourier heatmaps.

6 Conclusion

In this work, we proposed a new *frequency* backdoor attack that takes advantage of the natural frequency sensitivity of the DNN. Through extensive experiments, we showed the effectiveness of the proposed attack in embedding imperceptible backdoors that can evade existing defenses while achieving both a high ASR and a CDA. We also laid the foundations for future defenses against frequency-based backdoor attacks through (1) data preprocessing using autoencoders and compression; and (2) Fourier heatmap visualization.

7 Acknowledgement

This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-CRG2019-4033, as well as, the SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence (SDAIA-KAUST AI).

References

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

[2] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *ICML*, 2012.

[3] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Ben Edwards, Taesung Lee, Ian Molloy, and B. Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *ArXiv*, abs/1811.03728, 2019.

[4] Xiaoyi Chen, A. Salem, Michael Backes, Shiqing Ma, and Yang Zhang. Badnl: Backdoor attacks against nlp models. *ArXiv*, abs/2006.01043, 2020.

[5] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning. *ArXiv*, abs/1712.05526, 2017.

[6] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, L. Chen, M. Kounavis, and Duen Horng Chau. Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression. *ArXiv*, abs/1705.02900, 2017.

[7] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, L. Chen, M. Kounavis, and Duen Horng Chau. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

[8] Bao Gia Doan, Ehsan Abbasnejad, and D. Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. *Annual Computer Security Applications Conference*, 2020.

[9] Khoa D Doan and Yingjie Lao. Backdoor attack with imperceptible input and latent modification. 2021.

[10] Khoa D Doan, Yingjie Lao, Weijie Zhao, and Ping Li. Lira: Learnable, imperceptible and robust backdoor attacks. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[11] Yu Feng, Benteng Ma, Jing Zhang, Shanshan Zhao, Yong Xia, and Dacheng Tao. Fiba: Frequency-injection based backdoor attack in medical image analysis. *ArXiv*, abs/2112.01148, 2021.

[12] Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith Chinthana Ranasinghe, and Surya Nepal. Strip: a defence against trojan attacks on deep neural networks. *Proceedings of the 35th Annual Computer Security Applications Conference*, 2019.

[13] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, S. Nepal, and Hyoungshick Kim. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *ArXiv*, abs/2007.10760, 2020.

[14] I. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2015.

[15] A. Graves, Abdel rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013.

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

[17] Wenbo Guo, L. Wang, Xinyu Xing, Min Du, and D. Song. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *ArXiv*, abs/1908.01763, 2019.

[18] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: Defending against backdoor attacks using robust statistics. *ArXiv*, abs/2104.11315, 2021.

[19] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[20] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013.

[21] Yupeng Hu, Wenxin Kuang, Zheng Qin, Kenli Li, Jiliang Zhang, Yansong Gao, Wenjia Li, and Keqin Li. Artificial intelligence security: Threats and countermeasures. *ACM Computing Surveys*, 2021.

[22] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[23] Todd P. Huster and Emmanuel Ekwedike. Top: Backdoor detection in neural networks via transferability of perturbation. *ArXiv*, abs/2103.10274, 2021.

[24] J. Jumper, Richard Evans, A. Pritzel, Tim Green, Michael Figurnov, O. Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Zídek, Anna Potapenko, A. Bridgland, Clemens Meyer, Simon A A Kohl, Andy Ballard, A. Cowie, B. Romera-Paredes, Stanislav Nikolov, Rishub Jain, J. Adler, T. Back, Stig Petersen, D. Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, S. Bodenstein, D. Silver, Oriol Vinyals, A. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, pages 1 – 7, 2021.

[25] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[26] Hyung-Min Kwon and Yongchul Kim. Blindnet backdoor: Attack on deep neural network using blind watermark. *Multimedia Tools and Applications*, pages 1–18, 2022.

[27] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 18:2088–2105, 2021.

[28] Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. Backdoor learning: A survey. *ArXiv*, abs/2007.08745, 2020.

[29] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[30] Cong Liao, Haoti Zhong, Anna Cinzia Squicciarini, Sencun Zhu, and David J. Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, 2020.

[31] G. Litjens, Thijs Kooi, B. E. Bejnordi, A. Setio, F. Ciompi, M. Ghafoorian, J. V. D. Laak, B. Ginneken, and C. Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[32] K. Liu, Brendan Dolan-Gavitt, and S. Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*, 2018.

[33] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and X. Zhang. Trojaning attack on neural networks. In *NDSS*, 2018.

[34] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and X. Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[35] Yunfei Liu, Xingjun Ma, J. Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *ECCV*, 2020.

[36] Yuntao Liu, Yang Xie, and Ankur Srivastava. Neural trojans. *2017 IEEE International Conference on Computer Design (ICCD)*, pages 45–48, 2017.

[37] A. Nguyen and A. Tran. Wanet - imperceptible warping-based backdoor attack. *ArXiv*, abs/2102.10369, 2021.

[38] O. Parkhi, A. Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, 2015.

[39] Ximing Qiao, Yukun Yang, and H. Li. Defending neural backdoors via generative distribution modeling. In *NeurIPS*, 2019.

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[41] A. Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. *ArXiv*, abs/2003.03675, 2020.

[42] Ahmad El Sallab, Mohammed Abdou, E. Perot, and S. Yogamani. Deep reinforcement learning framework for autonomous driving. *ArXiv*, abs/1704.02532, 2017.

[43] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P. Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In *ICML*, 2021.

[44] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336–359, 2019.

[45] K. Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

[46] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *NeurIPS*, 2018.

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

[48] Tong Wang, Yuan Yao, Feng Xu, Shengwei An, Hanghang Tong, and Ting Wang. Backdoor attack through frequency domain. *ArXiv*, abs/2111.10991, 2021.

[49] Emily Wenger, Josephine Passananti, Arjun Nitin Bhagoji, Yuanshun Yao, Haitao Zheng, and Ben Y. Zhao. Backdoor attacks against deep learning systems in the physical world. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6202–6211, 2021.

[50] Yayuan Xiong, Fengyuan Xu, Sheng Zhong, and Qun A. Li. Escaping backdoor attack detection of deep learning. *ICT Systems Security and Privacy Protection*, 580:431 – 445, 2020.

[51] Zhicong Yan, Gaolei Li, Yuan Tian, Jun Wu, Shenghong Li, Mingzhe Chen, and H. Vincent Poor. Dehib: Deep hidden backdoor attack on semi-supervised learning via adversarial perturbation. In *AAAI*, 2021.

[52] Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, E. D. Cubuk, and J. Gilmer. A fourier perspective on model robustness in computer vision. *ArXiv*, abs/1906.08988, 2019.

[53] Sergey Zagoruyko and N. Komodakis. Wide residual networks. *ArXiv*, abs/1605.07146, 2016.

[54] Yi Zeng, Won Park, Zhuoqing Morley Mao, and R. Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. *ArXiv*, abs/2104.03413, 2021.

[55] Hongyang R. Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. *ArXiv*, abs/1901.08573, 2019.

[56] Jie zhang, Dongdong Chen, Jing Liao, Qidong Huang, G. Hua, Weiming Zhang, and Nenghai Yu. Poison ink: Robust and invisible backdoor attack. *ArXiv*, abs/2108.02488, 2021.

[57] Quan Zhang, Yifeng Ding, Yongqiang Tian, Jianmin Guo, Min Yuan, and Yu Jiang. Advdoor: adversarial backdoor attack of deep learning system. *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2021.

[58] Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. Backdoor attacks to graph neural networks. *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, 2021.

[59] Songzhu Zheng, Yikai Zhang, Hubert Wagner, Mayank Goswami, and Chao Chen. Topological detection of trojaned neural networks. *ArXiv*, abs/2106.06469, 2021.

Supplementary Material

A Introduction to Supplementary Material

In this supplementary material, we present the extended results and variants of the proposed frequency-based backdoor attack. Section B shows the full version of Table 1 from the main paper, this includes evaluation of the proposed pipeline on additional network architectures. Section C presents an extended evaluation of the proposed augmentation in Section 5.4 of the main paper (similar to Table 4). Section D discusses different design choices for the additive filters $\mathcal{A}_{R,G,B}$. In section E we extend the applicability of the proposed attack to the multitarget attack regime. Section F presents a more efficient variant of the proposed method. Section G shows the result of applying a binary and an additive filter generated from one model to poison another. Section H visualizes the proposed backdoor attack in the spatial domain, showing that the attack is highly dynamic. Section I displays the Fourier heatmaps and top-k selected frequencies (binary masks) for various datasets and architectures. Section J shows the Fourier heatmaps for different spatial backdoor attacks, highlighting a new possible defense against backdoor attacks. Section K presents a further evaluation of the spatial defenses discussed in the manuscript and presents three additional defenses, namely, STRIP [12], Spectral Signatures [46], and Activation Clustering [3]. Section L, shows an evaluation of the proposed defense against other frequency backdoor attacks. Section M presents a further comparison of our proposed attack against existing spatial backdoor attacks. Finally, section N provides insights about the relationship between the model's learning capacity and the capability of embedding a backdoor attack into the model.

B Evaluation of the Proposed Backdoor Attack

		ResNet18		ResNet34		ResNet50		DenseNet121		VGG19		WideResNet34	
	Poisoning Rate	**CDA**	**ASR**	**CDA**	**ASR**	**CDA**	**ASR**	**CDA**	**ASR**	**CDA**	**ASR**	**CDA**	**ASR**
CIFAR10	0.0%	93.92	-	94.59	-	94.10	-	94.70	-	92.47	-	95.33	-
	0.1%	94.00	1.54	94.49	0.83	94.48	53.63	94.94	86.98	92.63	0.44	95.73	84.91
	0.2%	94.14	72.31	94.26	66.46	94.45	87.91	94.54	95.77	92.39	0.44	95.42	96.89
	0.4%	94.20	85.05	94.33	90.97	94.37	95.38	94.89	96.48	92.17	1.62	95.48	99.34
	1.0%	94.38	99.44	94.44	91.75	94.32	99.34	94.83	98.70	91.95	99.39	95.70	99.80
	3.0%	94.31	99.79	94.41	99.64	94.31	99.36	94.94	99.89	91.89	99.81	95.44	99.99
CIFAR100	0.0%	75.95	-	75.66	-	77.36	-	78.98	-	67.45	-	79.55	-
	0.1%	75.76	60.57	76.76	65.18	76.73	42.18	78.34	73.47	67.78	0.40	79.84	43.96
	0.2%	75.75	92.78	74.79	84.09	77.87	78.21	79.1	89.31	67.72	0.59	79.24	78.42
	0.4%	75.92	96.49	76.25	99.29	77.69	83.96	79.1	92.67	67.61	0.20	79.14	87.33
	1.0%	76.05	98.99	74.95	99.44	77.12	90.49	78.6	96.44	65.84	0.40	79.14	98.02
	3.0%	75.36	99.93	76.51	99.84	76.58	98.61	78.31	99.60	67.14	99.00	78.74	99.41
GTSRB	0.0%	97.11	-	97.00	-	97.23	-	97.22	-	96.23	-	97.76	-
	0.1%	97.09	71.12	96.90	74.52	97.41	82.32	97.16	76.82	96.48	0.00	97.29	71.38
	0.2%	97.19	89.59	97.06	83.69	97.14	86.25	97.11	99.61	96.74	0.20	97.64	88.74
	0.4%	97.33	98.04	96.73	97.25	96.95	97.25	97.43	99.61	96.01	2.95	97.43	98.61
	1.0%	97.25	98.62	97.03	99.61	97.22	98.04	97.17	99.61	96.27	88.41	96.87	99.76
	3.0%	97.47	99.80	96.76	99.98	96.98	99.97	97.49	100.00	96.29	99.61	97.26	99.97
ImageNet	0.0%	67.51	-	70.86	-	73.35	-	74.10	-	72.11	-	-	-
	0.5%	67.38	0.17	71.20	84.70	73.27	96.00	73.91	95.32	71.49	91.96	-	-
	1.0%	67.13	87.74	70.74	95.96	73.38	98.03	74.21	98.05	72.33	96.64	-	-
	2.0%	67.26	98.01	70.57	98.87	72.78	98.85	73.75	99.34	71.62	95.379	-	-
	3.0%	67.26	98.32	70.67	98.95	72.30	99.25	73.39	99.85	72.05	97.51	-	-

Table 6: **Evaluation of the proposed backdoor attack.** We benchmark our proposed frequency-based backdoor attack on different network architectures, datasets, and poisoning rates. These results show that our attack can maintain clean data accuracy, while registering high attack success rates even with small poisoning rates.

WideResNet34 was not included for ImageNet experiments as there is no official implementation of this model in *torchvision.models* .

C Evaluation of Augmented Models

The manuscript discusses two defense techniques against frequency-based backdoor attacks and a simple technique to bypass them through training data augmentation. The results presented in the paper correspond to ResNet18 trained on CIFAR10, CIFAR100, and GTSRB. Tables 7, 8 and 9 present the results for ResNet34, WideResNet34, and VGG19 trained on the aforementioned datasets with training data augmentation. Based on these results, training data augmentation was shown to be a viable counter-attack to the proposed backdoor defenses.

C.1 ResNet34

		Autoencoder		**JPEG**		**JPEG+Autoencoder**	
	Poisoning Rate	**CDA**	**ASR**	**CDA**	**ASR**	**CDA**	**ASR**
CIFAR10	0.1%	94.75	0.12	94.17	0.22	94.75	0.44
	0.2%	94.49	0.88	94.75	2.86	94.26	1.32
	0.4%	94.71	80.66	94.68	78.90	93.95	17.14
	1.0%	94.30	95.82	94.49	87.91	94.11	92.53
	3.0%	94.51	98.46	94.47	97.58	94.50	94.73
CIFAR100	0.1%	76.84	10.70	77.53	61.39	77.52	3.56
	0.2%	76.08	19.41	76.55	84.55	76.39	38.81
	0.4%	77.56	95.44	76.49	95.84	77.12	93.27
	1.0%	76.98	99.60	77.20	96.63	77.31	97.22
	3.0%	76.53	99.61	76.26	99.60	76.43	99.00
GTSRB	0.1%	97.25	43.81	97.19	59.33	96.96	48.33
	0.2%	96.96	88.02	97.18	86.64	97.05	78.78
	0.4%	97.11	93.91	97.09	88.21	96.78	94.50
	1.0%	95.25	91.16	96.84	97.05	93.99	80.35
	3.0%	94.61	98.82	96.92	99.02	94.79	88.41

Table 7: **Augmentation Maintains Performance (CIFAR10, CIFAR100 and GTSRB):** CDA and ASR of backdoored ResNet34 trained on CIFAR10, CIFAR100 and GTSRB with JPEG compression and Autoencoder augmentation. Both ASR and CDA are maintained even when no preprocessing technique is used.

C.2 WideResNet34

		Autoencoder		JPEG		JPEG+Autoencoder	
	Poisoning Rate	**CDA**	**ASR**	**CDA**	**ASR**	**CDA**	**ASR**
CIFAR10	0.1%	95.70	63.96	95.40	69.01	95.63	1.98
	0.2%	95.44	89.01	95.30	86.59	95.69	80.88
	0.4%	95.26	97.14	95.57	88.57	95.32	85.49
	1.0%	95.13	98.46	95.56	97.36	95.46	95.82
	3.0%	95.48	99.12	95.37	98.02	95.53	98.24
CIFAR100	0.1%	79.15	25.94	79.27	19.60	79.31	8.71
	0.2%	79.62	63.76	79.46	60.20	79.60	35.84
	0.4%	79.72	87.92	79.19	79.41	79.36	79.60
	1.0%	79.22	93.47	79.29	93.66	79.23	73.47
	3.0%	79.27	98.22	79.11	95.64	78.81	90.30
GTSRB	0.1%	97.70	54.03	97.51	60.12	97.47	48.72
	0.2%	97.02	86.44	97.70	72.10	96.84	34.58
	0.4%	95.72	80.75	96.96	97.64	93.84	80.16
	1.0%	92.15	46.95	93.03	43.81	92.30	88.45
	3.0%	93.12	93.52	90.63	78.19	89.46	89.78

Table 8: **Augmentation Maintains Performance (CIFAR10, CIFAR100 and GTSRB):** CDA and ASR of backdoored WideResNet34 trained on CIFAR10, CIFAR100 and GTSRB with JPEG compression and Autoencoder augmentation. Both ASR and CDA are maintained even when no preprocessing technique is used.

C.3 VGG19

		Autoencoder		JPEG		JPEG+Autoencoder	
	Poisoning Rate	**CDA**	**ASR**	**CDA**	**ASR**	**CDA**	**ASR**
CIFAR10	0.1%	91.82	1.10	92.13	0.44	92.65	0.44
	0.2%	92.39	0.66	92.28	0.66	92.36	0.88
	0.4%	92.30	1.10	92.16	7.47	92.43	1.76
	1.0%	92.04	89.89	92.60	97.36	92.16	86.15
	3.0%	92.52	99.56	92.21	100.00	91.89	98.90
CIFAR100	0.1%	68.91	0.20	68.84	0.21	68.82	0.20
	0.2%	68.24	0.59	68.71	0.40	68.76	1.19
	0.4%	68.50	2.57	68.36	1.20	68.74	2.18
	1.0%	68.36	8.91	68.12	4.95	68.12	7.37
	3.0%	67.70	98.02	68.12	97.82	68.14	95.05
GTSRB	0.1%	96.69	0.00	96.33	0.00	96.81	0.00
	0.2%	96.39	0.20	96.43	0.20	96.62	0.79
	0.4%	95.87	0.00	96.04	0.00	95.98	24.36
	1.0%	95.63	88.61	96.00	91.55	96.28	89.00
	3.0%	96.22	99.41	95.82	98.82	95.97	100.00

Table 9: **Augmentation Maintains Performance (CIFAR10, CIFAR100 and GTSRB):** CDA and ASR of backdoored VGG19 trained on CIFAR10, CIFAR100 and GTSRB with JPEG compression and Autoencoder augmentation. Both ASR and CDA are maintained even when no preprocessing technique is used.

D Choice of Additive Filters $\mathcal{A}_{R,G,B}$

The results presented in the manuscript set the values of the additive filters to be the same within the channel but different across the channels. We now consider different possible design choices for this additive filter, namely, choosing random or same values (within and across channels) for $\mathcal{A}_{R,G,B}$. Tables 10 and 11 both show high ASR and CDA for different choices of $\mathcal{A}_{R,G,B}$ illustrating the flexibility of the proposed method in creating backdoor attacks.

D.1 Random Values for $\mathcal{A}_{R,G,B}$

	ResNet18		ResNet34		VGG19	
Poisoning Rate	**CDA(%)**	**ASR(%)**	**CDA(%)**	**ASR(%)**	**CDA(%)**	**ASR(%)**
0.1%	92.93	2.64	93.23	0.88	92.12	2.20
0.2%	92.83	37.80	93.28	11.21	91.86	22.20
0.4%	93.16	90.11	93.49	93.19	92.18	56.68
1.0%	93.04	97.58	93.20	98.90	92.26	95.60
3.0%	93.21	99.56	93.29	99.78	92.28	98.90

Table 10: **Random Additive Filter Values.** Evaluating the proposed backdoor attack using random additive filter values shows that our attack can maintain clean data accuracy while reaching high attack success rates with small poisoning rates.

D.2 Same Value for $\mathcal{A}_{R,G,B}$

	ResNet18		ResNet34		VGG19	
Poisoning Rate	**CDA(%)**	**ASR(%)**	**CDA(%)**	**ASR(%)**	**CDA(%)**	**ASR(%)**
0.1%	93.01	1.76	93.55	1.32	91.93	1.54
0.2%	92.95	66.59	93.33	22.64	92.15	29.45
0.4%	92.89	92.08	93.31	93.85	91.80	66.37
1.0%	93.18	98.24	93.42	98.46	92.10	97.14
3.0%	92.87	99.56	93.06	99.34	91.94	99.12

Table 11: **Same Additive Filter Values.** Evaluating the proposed backdoor attack using same additive filter values (across and within channels) shows that our attack can maintain clean data accuracy while reaching high attack success rates with small poisoning rates.

E Multi Target Attacks in the Frequency Domain

The manuscript focuses on creating single-target backdoor attacks. We extend the applicability of the proposed frequency-based backdoor attack to the multitarget regime. This is done through introducing an additional step to the recipe:

1. Select the top-k frequencies (most sensitive).
2. Randomly and equally divide the selected frequencies among the poisoned classes creating a binary mask for each.
3. Create a set of additive filters for each poisoned class.
4. Poison each class with its corresponding additive filter and binary mask.
5. Proceed with training.

Figure 7 shows the binary masks for the two poisoned classes of ResNet18 trained on CIFAR10; Table 12 shows the results for poisoning the first 2 classes of CIFAR10 for various network architectures.

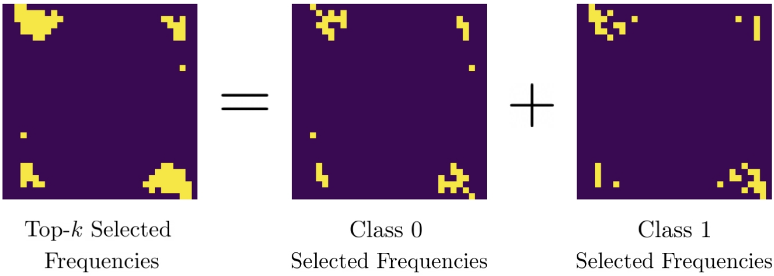

=
+
Top-k Selected Frequencies
Class 0 Selected Frequencies
Class 1 Selected Frequencies

Figure 7: **Multitarget (2 classes) Binary Filters for ResNet18 on CIFAR10.** The top-k selected frequencies to poison are divided equally and randomly to create two binary masks one for each poisoned class.

	ResNet18			ResNet34			VGG19		
Poisoning Rate	**CDA(%)**	**ASR-0 (%)**	**ASR-1 (%)**	**CDA(%)**	**ASR-0 (%)**	**ASR-1 (%)**	**CDA(%)**	**ASR-0 (%)**	**ASR-1 (%)**
0.1%	93.09	1.76	1.27	93.32	0.88	0.21	91.81	1.54	1.06
0.2%	92.76	32.31	28.23	93.24	10.32	10.82	91.97	8.13	9.98
0.4%	92.89	96.70	87.69	93.59	86.59	72.40	91.90	61.53	69.21
1.0%	93.09	98.90	96.60	93.59	98.46	97.88	92.17	92.75	94.90
3.0%	92.87	99.86	99.57	93.18	99.78	99.79	91.93	98.68	99.15

Table 12: **Multitarget Frequency-Based Backdoor Attack.** The proposed multitarget variant of the frequency-based backdoor attack can successfully poison the first two classes of CIFAR10 on various network architectures using small poisoning rate. ASR-0 and ASR-1 denote the attack success rate for triggering classes 0 and 1 respectively.

F End-to-End Pipeline: A More Efficient Variant

In this section, we present a more efficient variant of the proposed method. The method proposed in the manuscript requires training two models: (1) a clean model (f_0) for which the Fourier heatmap is computed for; and (2) a poisoned model (f) which utilizes the heatmap generated from the clean model to poison the data and hence embed the backdoor.

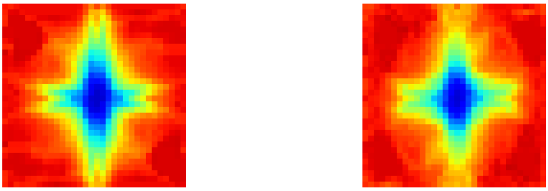
Our method could be modified so that only one model is trained, the modified version is summarized below:

1. Train a model on clean samples until a reasonable performance is reached. We denote this checkpoint by C_0.
2. Generate the Fourier heatmap for C_0 and select the top-k most sensitive frequencies to generate the binary mask $\mathcal{M}$ and the additive filters $\mathcal{A}_{R,G,B}$.
3. Poison the data using equations (4), (5) and (6) presented in the manuscript and proceed with training C_0 on both poisoned and clean samples. The obtained model is the poisoned model f.

Table 13 shows the results of using the "end-to-end" variant of the proposed frequency-based backdoor attack. The obtained results are fairly similar to those shown in Table 6. Figure 8 shows the Fourier heatmaps of the clean and poisoned models for the proposed variant (ResNet18 trained on CIFAR10). As required, the Fourier heatmap of the poisoned model is similar to that of the clean model.

	ResNet18		**ResNet34**		**VGG19**	
Poisoning Rate	**CDA(%)**	**ASR(%)**	**CDA(%)**	**ASR(%)**	**CDA(%)**	**ASR(%)**
0.1%	93.19	3.08	93.82	0.88	92.24	0.44
0.2%	93.46	48.13	93.65	3.95	91.98	5.49
0.4%	93.38	83.51	93.43	89.89	92.09	28.79
1.0%	93.25	96.04	93.45	85.05	92.45	81.75
3.0%	93.31	98.02	93.34	99.12	92.51	96.26

Table 13: **End-to-End Pipeline Evaluation.** The proposed end-to-end variant of the frequency-based backdoor attack achieves both a high clean data accuracy and a high attack success rate.



(a) Heatmap of Clean Model

(b) Heatmap of Poisoned Model

Figure 8: **End-to-End Variant Maintains Fourier Heatmaps.** Utilizing the end-to-end frequency-based backdoor attack allows us to obtain a backdoored model with a Fourier heatmap similar to that of the clean model.

G Cross Filter Frequency-Based Backdoor Attack

In this section, we show the capability of utilizing binary masks and additive filters generated for one architecture to backdoor attack another. As expected, one can reach a high attack success rate (for a high enough poisoning rate) using such masks and filters (*Check Ablation-Section 5.5 in manuscript*); however, one has no guarantee over maintaining a Fourier heatmap similar to the clean model.

Table 14 shows the CDA and ASR of a ResNet18 trained on CIFAR10 poisoned using binary masks and additive filters of WideResNet34, ResNet34, and VGG19.

Filter & Mask Source	**WideResNet34**		**ResNet34**		**VGG19**	
Poisoning Rate	**CDA(%)**	**ASR(%)**	**CDA(%)**	**ASR(%)**	**CDA(%)**	**ASR(%)**
0.0%	93.28	5.27	93.10	1.98	93.00	5.93
0.1%	92.93	2.86	92.75	35.82	92.81	50.77
0.4%	93.17	93.63	93.14	95.16	92.79	95.16
1.0%	93.08	99.12	93.24	98.46	92.80	96.92
3.0%	92.74	100.00	92.94	99.78	92.85	99.34

Table 14: **Cross Filter Backdoor Attack Evaluation.** Evaluating different binary masks and additive filters generated for WideResNet34, ResNet34, and VGG19 for attacking ResNet18 on CIFAR10.

H Spatial Visualization of the Proposed Frequency-based Backdoor Attack

In this section, we visualize the scaled absolute difference ($\mathcal{D}$) of non-poisoned images ($\mathcal{I}$) and poisoned images ($\mathcal{I}_P$) defined as:

$$\mathcal{D} = \gamma|\mathcal{I} - \mathcal{I}_P| \tag{4}$$

where $|.|$ of a matrix denotes element-wise absolute value operation and γ is a scalar multiplier in $\mathbb{R}$. Figures 9 and 10 visualize two sets of non-poisoned images, their poisoned counterparts, and the absolute scaled difference with $\gamma = 50$ for ResNet18 and ResNet34 (recall that our attack is model dependent).

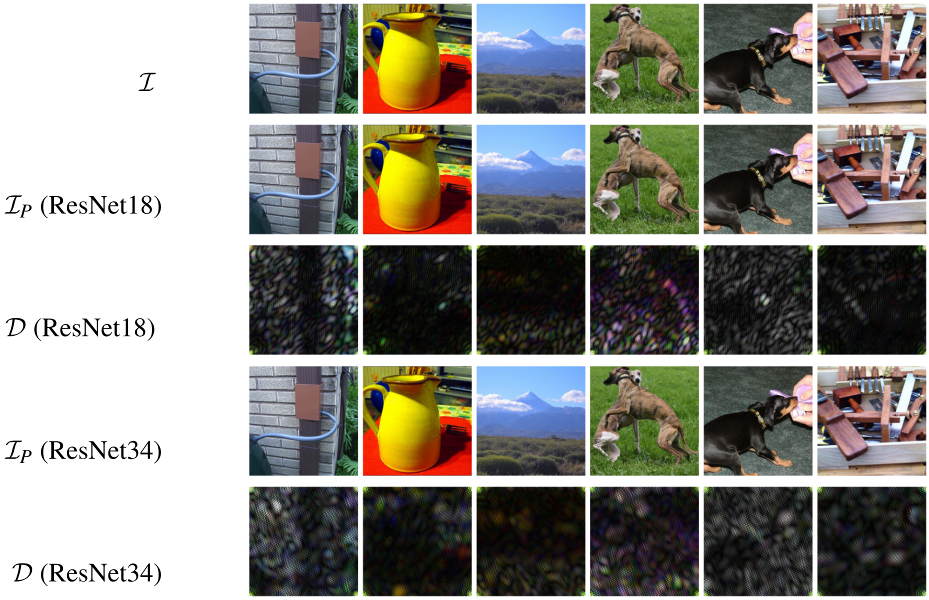


Figure 9: **Spatial Visualization of Proposed Attack.** Visualization of the absolute scaled difference shows how dynamic the proposed attack. The poisoned images show the imperceptibility of the attack.

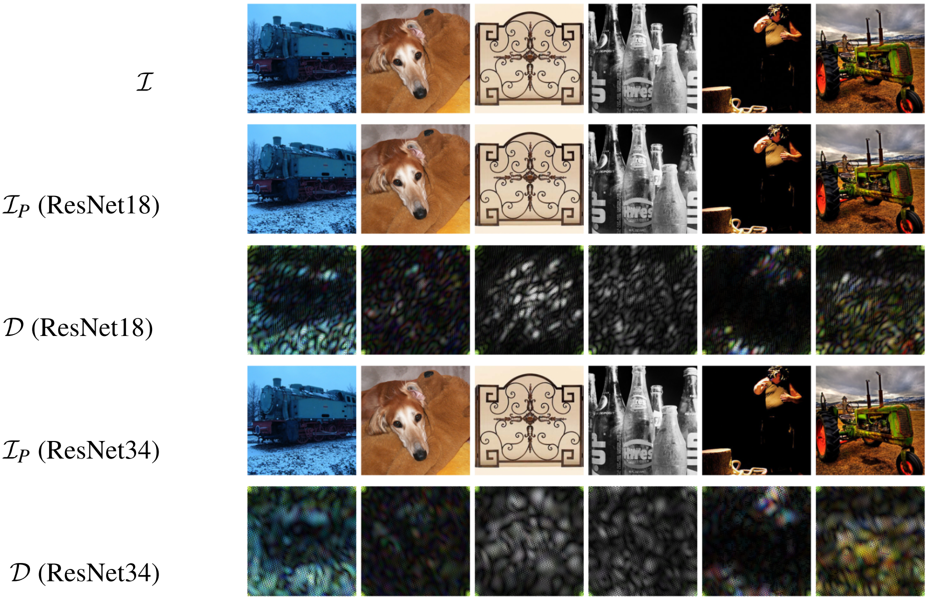


Figure 10: **Spatial Visualization of Proposed Attack.** Visualization of the absolute scaled difference shows how dynamic the proposed attack. The poisoned images show the imperceptibility of the attack.

I Fourier Heatmaps

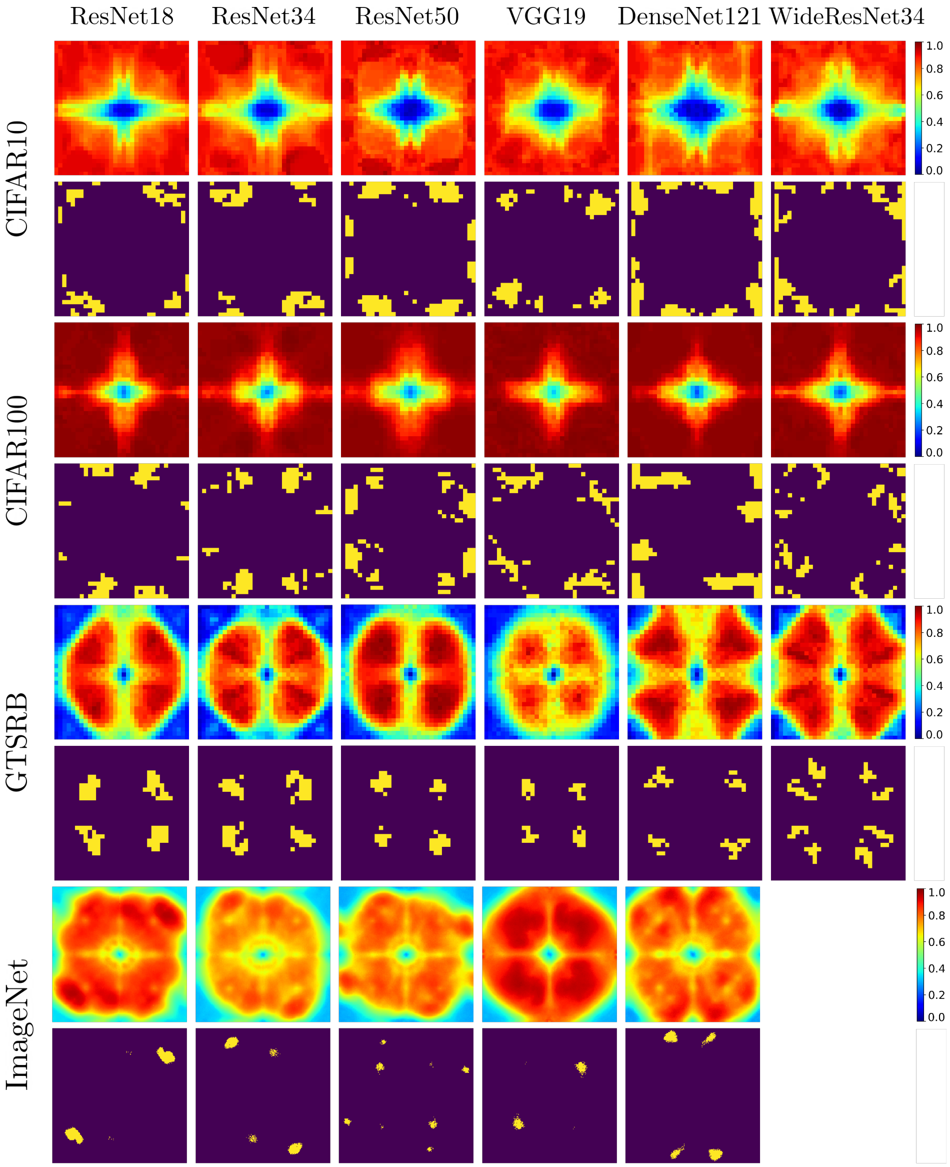


Figure 11: **Fourier Heatmaps and Top-*k* Filters.** Fourier heatmaps for various architectures and datasets along with their top-*k* selected frequencies for the binary mask.

WideResNet34 was not included for ImageNet experiments as there is no official implementation of this model in *torchvision.models* .

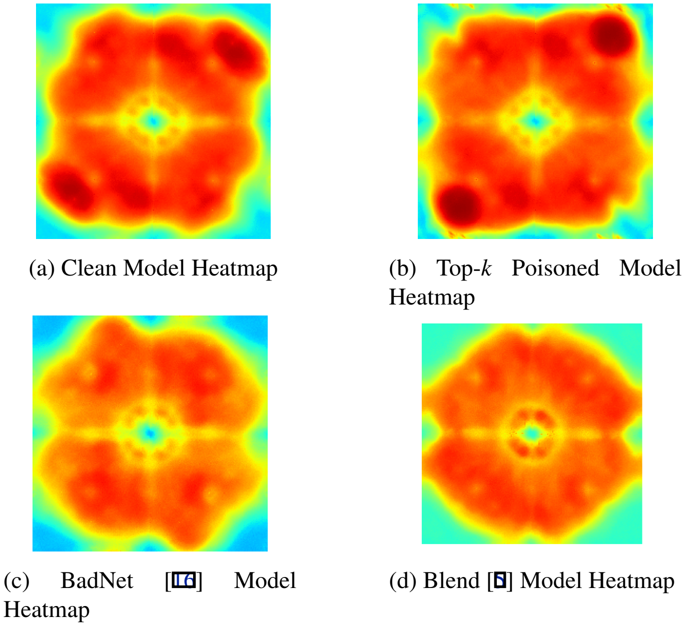
	ResNet18	**ResNet34**	**ResNet50**	**VGG19**	**DenseNet121**	**WideResNet34**
CIFAR10	9.77%	9.77%	11.72%	9.77%	16.61%	14.65%
CIFAR100	15.62%	19.53%	9.77%	19.53%	15.63%	15.63%
GTSRB	8.79%	8.79%	2.92%	3.90%	4.88%	8.80%
ImageNet	1.99%	1.59%	0.99%	0.90%	1.99%	-

Table 15: **Percentage of Poisoned Frequencies.** Different models and datasets require poisoning different percentages of the Fourier bases to achieve a balance between stealthiness, attack success rate and clean data accuracy.

Figure 11 shows all the Fourier heatmaps and the binary masks generated for poisoning the different models on all datasets. Table 15 shows the percentage of poisoned frequencies for each binary mask.

J Fourier Heatmap as a Backdoor Detector

As shown in the manuscript, if the choice of poisoned frequencies is not carried out properly, a simple check on the Fourier heatmap of the obtained model could expose the attacker (an abnormal trend is observed in the heatmap). Figures 12a, 12b, 12c and 12d show the Fourier heatmaps of clean ResNet18, top-*k* poisoned ResNet18, BadNet [16] poisoned ResNet18, and Blend [5] poisoned ResNet18, respectively. BadNet represents the first backdoor attack in the literature and is based on poisoning data by applying a white patch to the corner of a subset of the training set. Blend on the other hand, was the first to recognize the importance of imperceptibility and suggested blending images with the poison trigger for a more stealthy attack. Figures 12c and 12d show that both BadNets and Blend tend to highly change the frequency sensitivity of the attacked model compared to the clean one and hence could be detected as poisoned models by inspecting their heatmaps. The proposed frequency-based backdoor attack is more conservative and introduces only mild changes to the clean model heatmap and therefore are less detectable as poisoned.



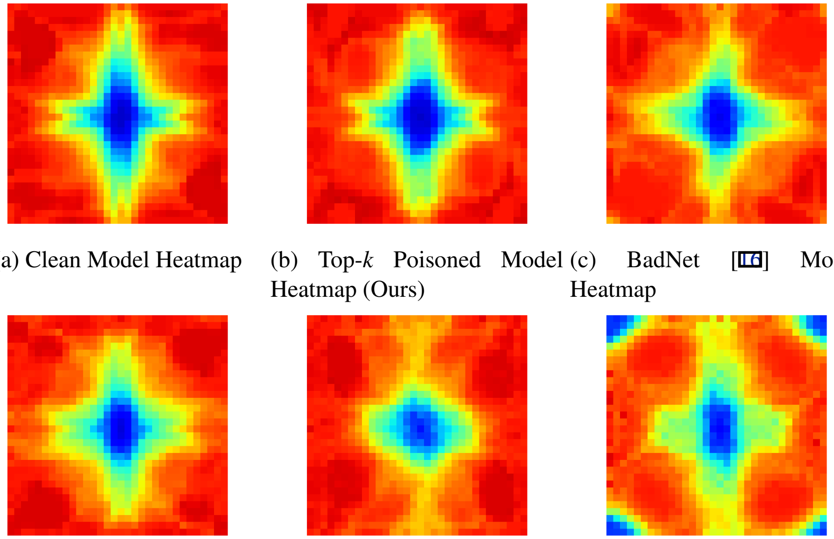
(a) Clean Model Heatmap

(b) Top-*k* Poisoned Model Heatmap

(c) BadNet [16] Model Heatmap

(d) Blend [5] Model Heatmap

Figure 12: **Fourier Heatmap As a Backdoor Detector.** BadNet and Blend poisoned models introduce more significant changes to the clean heatmap as compared to the proposed top-*k* frequency-based backdoor attack. These heatmaps could be exploited as means to detecting whether a model is poisoned or not.



(a) Clean Model Heatmap

(b) Top-k Poisoned Model Heatmap (Ours)

(c) BadNet [16] Model Heatmap

(d) Blend [5] Model Heatmap

(e) Clean Label [?] Model Heatmap

(f) RE [?] Model Heatmap

Figure 13: **Fourier Heatmap As a Backdoor Detector.** Various spatial backdoor attacks introduce more significant changes to the clean heatmap as compared to the proposed top-k frequency-based backdoor attack. These heatmaps could be exploited as means to detecting whether a model is poisoned or not.

Similarly, this inspection could be applied for models trained on small image datasets such as CIFAR10. Figures 13a, 13b, 13c, 13d, 13e and 13f show the Fourier heatmaps for ResNet18 trained on CIFAR10 with different poisoning strategies. Our proposed method maintains the highest similarity to the clean model's Fourier heatmap as compared to other methods.

K Evaluation Against Backdoor Defenses

In this section, we provide a further evaluation of the spatial defenses presented in the manuscript. We also evaluate our method against additional defenses, namely, STRIP [12], Activation Clustering [3] and Spectral Signatures [46].

Recall that Grad-CAM uses gradients of a particular class to visualize where the network is looking/focusing at to make its prediction. [32] prunes the least active neurons (on clean samples) and then fine-tunes the network on clean samples. STRong Intentional Pertubation (STRIP) [12] intentionally perturbs the input through blending it with clean samples. The authors rely on the realization that blending a poisoned sample with a clean sample would still activate the backdoor attack and therefore studying the entropy of the prediction vectors could be used for backdoor detection. Activation Clustering (AC) [3] analyzes the neural network's representation layer activation to determine whether the data has been poisoned. Since a poisoned model assigns poisoned and clean data to the target class based on a different feature representation, one can cluster the representations of the poisoned class into two distinct clusters. Similar to AC, Spectral Signatures (SS) [46] operates on feature representations to detect backdoor attacks. SS detects the poisoned samples using robust statistics and SVD methods.

Figures 14 and 15 show a set of images visualizing the original image, the Grad-CAM for a clean network evaluated on the clean sample, and the Grad-CAM for a poisoned network evaluated on the poisoned sample (left to right) for GTSRB and ImageNet respectively (the network architecture is ResNet18). As shown in the manuscript, the network focus regions are relatively unchanged when the frequency-based poison is applied.

Figure 16 shows the result of pruning a ResNet34 trained on ImageNet. Again, as observed in the manuscript (for CIFAR10), the attack success rate of the frequency-based backdoor is maintained for large pruning rates that highly drop the clean data accuracy.

Figure 17 shows the results of applying STRIP defense to a poisoned VGG19 model trained on CIFAR10 and GTSRB with various poisoning rates. Our method causes no significant distributional shift in the prediction vector's entropy therefore is not detectable by STRIP.

Figures 18, 19, and 20 show the results of Activation Clustering defense method applied to various models with different poisoning rates, namely, ResNet18 (1.0% poisoning rate), ResNet34 (0.4% poisoning rate), and ResNet50 (0.4% poisoning rate) respectively. Visually, AC fails to find two distinct and separable clusters and therefore fails to detect the backdoor attack. Numerically, in terms of silhouette scores, Tables 16, 17, and 18 show that no score is significantly higher than the other scores for the three considered models.

Figures 21, 22, and 23 show the results for Spectral Signatures defense method applied to various models with different poisoning rates, namely, DenseNet121 (0.4% poisoning rate), ResNet34 (0.4% poisoning rate) and ResNet50 (0.4% poisoning rate) respectively. Visually, the method fails to find spectrally separable clusters and therefore the backdoor is not detected. Numerically, the true positive rates (6%, 45%, and 33%) are lower than the threshold (90%) [46] required for the defense to be deemed successful.

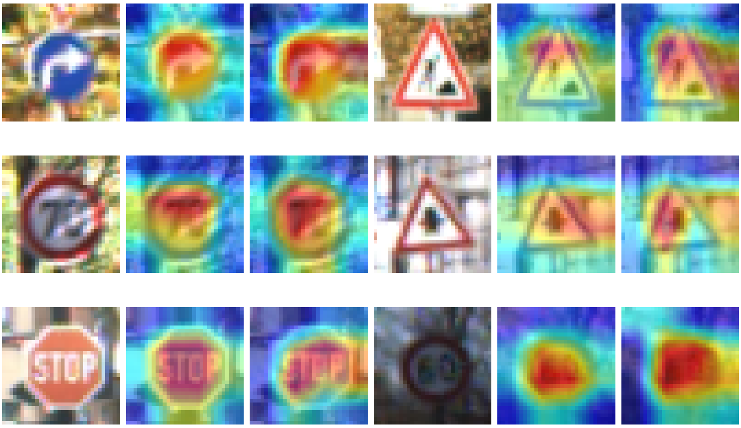


Figure 14: **Grad-CAM on GTSRB.** The proposed Frequency-based backdoor attack allows the network to focus on similar regions when classifying poisoned images as compared to clean network operating on the clean version of the images. Methods that focus on Grad-CAM based image-reconstruction fail to remove the poison.

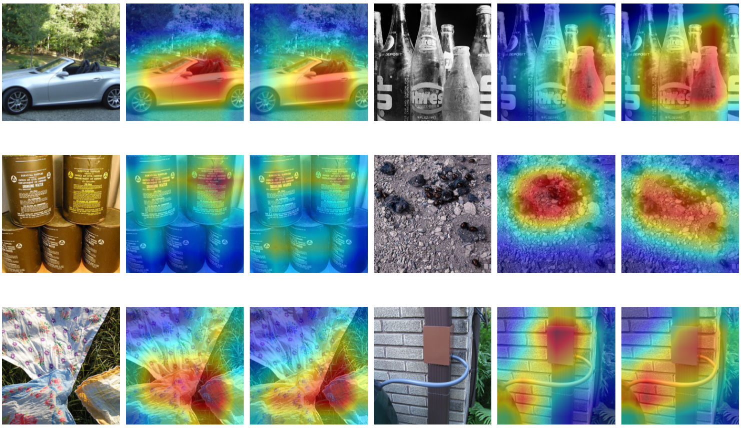


Figure 15: **Grad-CAM on ImageNet.** The proposed Frequency-based backdoor attack allows the network to focus on similar regions when classifying poisoned images as compared to clean network operating on the clean version of the images. Methods that focus on Grad-CAM based image-reconstruction fail to remove the poison.

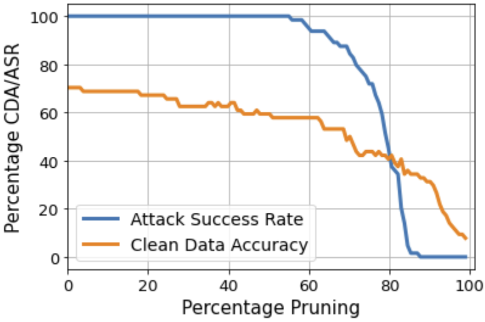


Figure 16: **Pruning ResNet18 Trained on ImageNet.** Frequency-based backdoors are successfully maintained across high pruning rates that significantly drop the clean data accuracy.

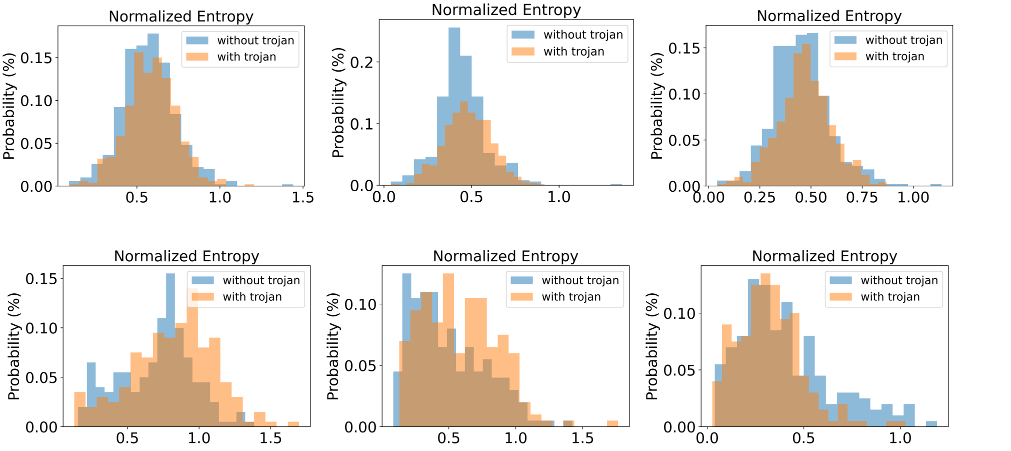


Figure 17: **STRIP On Various Datasets.** The proposed frequency backdoor attack is not detectable by STRIP defense mechanism. Rows 1 and 2 show the results for VGG19 trained on CIFAR10 and GTSRB, respectively, with poisoning rates of 0.4%, 1.0% and 3.0% (left to right).

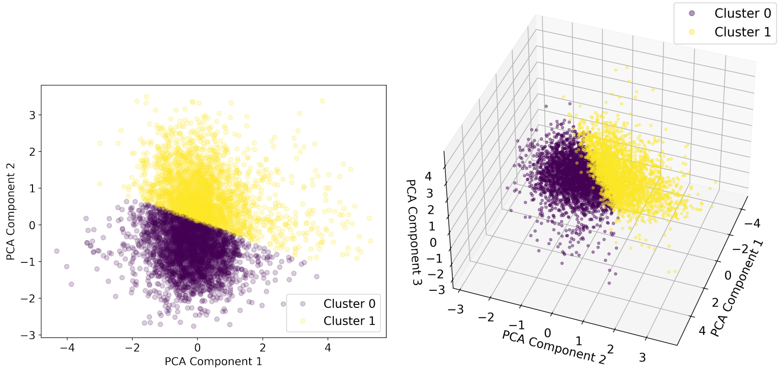


Figure 18: **Activation Clustering on ResNet18 (1.0% Poisoning Rate).** Activation clustering fails to find two distant clusters in both cases of 2 and 3 principal components. Under the assumption that less than 50% of the data is poisoned, we generally consider the smaller cluster as poisonous (in our case no cluster smaller than the other exists for the poisoned class).

Label	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Silhouette Score	0.322	0.343	0.319	0.318	0.321	0.328	0.321	0.363	0.337	0.326

Table 16: **Activation Clustering on ResNet18 (1.0% Poisoning Rate - 2 PCA Components).** Silhouette scores indicate how well the clustering fits the data. The higher the score the better the clusters fit the data. AC's silhouette scores on our method are similar hence it fails to detect the backdoor.

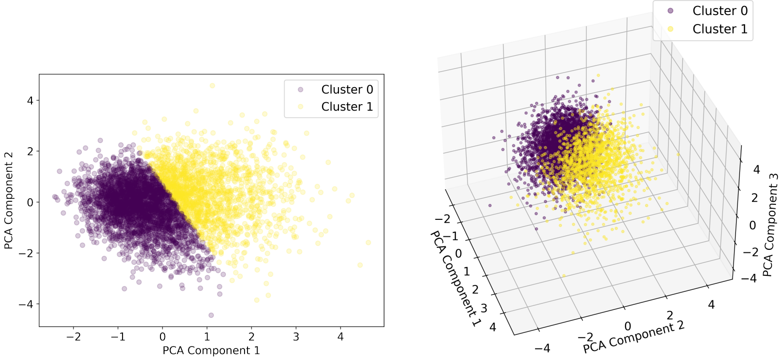


Figure 19: **Activation Clustering on ResNet34 (0.4% Poisoning Rate).** Activation clustering fails to find two distant clusters in both cases of 2 and 3 principal components. Under the assumption that less than 50% of the data is poisoned, we generally consider the smaller cluster as poisonous (in our case no cluster smaller than the other exists for the poisoned class).

Label	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Silhouette Score	0.321	0.357	0.321	0.319	0.325	0.313	0.311	0.324	0.346	0.347

Table 17: **Activation Clustering on ResNet34 (0.4% Poisoning Rate - 2 PCA Components).** Silhouette scores indicate how well the clustering fits the data. The higher the score the better the clusters fit the data. AC's silhouette scores on our method are similar hence it fails to detect the backdoor.

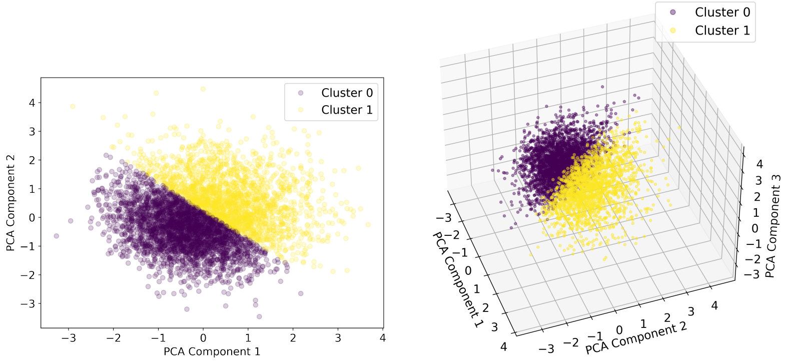


Figure 20: **Activation Clustering on ResNet50 (0.4% Poisoning Rate).** Activation clustering fails to find two distant clusters in both cases of 2 and 3 principal components. Under the assumption that less than 50% of the data is poisoned, we generally consider the smaller cluster as poisonous (in our case no cluster smaller than the other exists for the poisoned class).

Label	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Silhouette Score	0.307	0.343	0.320	0.309	0.311	0.314	0.313	0.329	0.329	0.325

Table 18: **Activation Clustering on ResNet50 (0.4% Poisoning Rate - 2 PCA Components).** Silhouette scores indicate how well the clustering fits the data. The higher the score the better the clusters fit the data. AC's silhouette scores on our method are similar hence it fails to detect the backdoor.

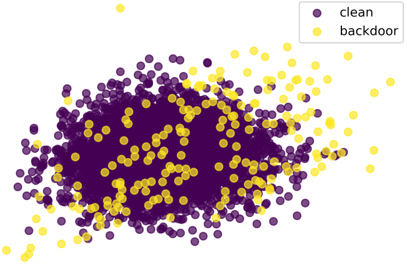


Figure 21: **Spectral Signatures Defense on DenseNet121 (0.4% Poisoning Rate).** Spectral Signatures (SS) backdoor defense method fails to find two separate clusters for clean and backdoored samples. The true positive (TP) and false positive (FP) detection rates are 6% and 7.3% respectively and hence SS fails to detect our method.

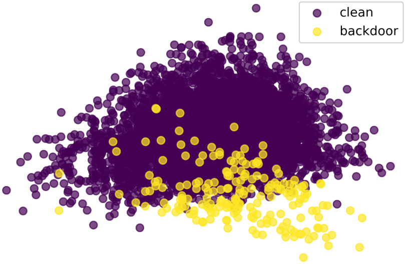


Figure 22: **Spectral Signatures Defense on ResNet34 (0.4% Poisoning Rate).** Spectral Signatures (SS) backdoor defense method fails to find two separate clusters for clean and backdoored samples. The true positive (TP) and false positive (FP) detection rates are 45% and 6.3% respectively and hence SS fails to detect our method.

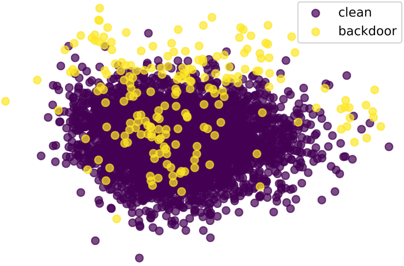


Figure 23: **Spectral Signatures Defense on ResNet50 (0.4% Poisoning Rate).** Spectral Signatures (SS) backdoor defense method fails to find two separate clusters for clean and backdoored samples. The true positive (TP) and false positive (FP) detection rates are 33% and 7.2% respectively and hence SS fails to detect our method.

L Evaluation of Proposed Defenses on Other Frequency Attacks

Table 19 shows the clean data accuracy and the attack success rate of models poisoned by our method, FIBA and FTrojan. The defense shows promising results against two of three frequency backdoor attacks.

		No Defense	**JPEG Compression**	**Autoencoder**
FIBA	**CDA(%)**	92.51	83.80	82.33
	ASR(%)	96.54	92.85	71.43
FTrojan	**CDA(%)**	92.84	85.05	0.00
	ASR	100.00	82.05	0.00
Ours	**CDA(%)**	94.38	86.91	0.00
	ASR(%)	99.44	83.15	0.00

Table 19: **Proposed Frequency Backdoor Defenses:** The proposed backdoor defenses, JPEG compression and autoencoder, could break FTrojan and the proposed method. FIBA poisons low-frequency content that is usually not removed through either compression techniques. The results are reported for CIFAR10 - ResNet18 model.

M Comparing Backdoor Attacks

Table 20 compares the proposed method with other spatial backdoor attacks. As observed, our method surpasses the performance of spatial backdoor attacks in clean data accuracy, attack success rate, and invisibility metrics (PSNR and LPIPS). Least Significant Bit (LSB) attack is not included in the comparison as it fails to achieve a high attack success rate and hence fails to create a backdoor attack in the first place.

Metric	**Ratio**	**BadNets**	**Blend**	**SIG**	**Refool**	**SPM**	**LSB**	**Poison Ink**	**Ours**
CDA/ASR	3%	87.38 / 66.55	89.89 / 89.39	89.74 / 99.23	89.20 / 87.16	88.89 / 58.53	88.18 / 10.91	89.65 / 94.22	**92.31 / 99.43**
	5%	87.13 / 65.36	89.60 / 90.99	89.64 / 99.47	89.16 / 89.79	88.90 / 57.69	86.98 / 11.67	89.69 / 93.58	**91.88 / 99.88**
	10%	85.61 / 68.01	89.77 / 93.11	89.45 / 99.40	88.80 / 92.80	89.07 / 57.33	83.69 / 15.76	89.47 / 93.67	**92.10 / 99.97**
PSNR↑/LPIPS↓		25.68 / 0.0009	21.29 / 0.0240	25.12 / 0.0400	19.38 / 0.0397	38.94 / 0.0001	51.13 / 0.00001	42.95 / 0.0001	**43.15 / 0.00001**

Table 20: **Comparison between the Proposed Attack and Backdoor Attacks in the Literature.** Our proposed frequency-based technique provides the best trade off as compared to spatial attacks. It achieves SOTA ASR, CDA, PSNR, and LPIPS metrics. The results shown are for VGG19 trained on CIFAR10. The LSB method is dropped as it fails to create a backdoor with good ASR.

N Learning Capacity vs Poisoning Capabilities

Based on our experiments (check Table 6), a particularly interesting yet expected trend is noticed. Networks like VGG19, which lack any skip connections, tend to be harder to backdoor attack. This is because the poison information dilutes as we move deeper and deeper in the network architecture. Low norm invisible attacks tend to be particularly influenced by this, and hence, non-residual networks require a higher poisoning rate for embedding a backdoor. On the other hand, networks like ResNets, WideResNets, and DenseNets seem to be capable of maintaining the poison information through their skip connections and hence can be backdoored with a fairly small amount of poisoned data.