Backdoor Attack with Imperceptible Input and Latent Modification

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent studies have shown deep neural networks (DNN) are vulnerable to various adversarial attacks. In particular, an adversary can inject a stealthy backdoor into a model such that it will behave normally without the presence of the trigger. Techniques for generating backdoor images that are visually imperceptible from clean images have also been developed recently, which further enhance the stealthiness of the backdoor attacks from the input space. Along with the development of attacks, defense against backdoor attacks is also evolving. Many existing countermeasures found that backdoor tends to leave tangible footprints in the latent or feature space, which can be utilized to mitigate backdoor attacks. This paper extends the concept of imperceptible backdoor from the input space to the latent representation, which significantly improves the effectiveness against the existing defense mechanisms, especially those relying on the distinguishability between clean inputs and backdoor inputs in latent space. In this framework, the trigger function will learn to manipulate the input by injecting imperceptible input noise while matching the latent representations of the clean and manipulated inputs via a Wasserstein-based regularization of the corresponding empirical distributions. We formulate such an objective as a non-convex and constrained optimization problem and solve it with an efficient stochastic alternating optimization procedure. The proposed framework achieves a high attack success rate while being stealthy from both the input and latent spaces in several benchmark datasets, including MNIST, CIFAR10, GTSRB, and TinyImagenet.

1 Introduction

In the past years, deep neural network (DNN) has successfully evolved many technological fields, such as object classification [21, 16], face recognition [25, 1], autonomous driving [45], and even security applications [15, 3]. Meanwhile, due to the underlying black-box nature, its security and privacy implications have also raised extensive concerns recently. Efforts in the research community have exposed the vulnerability of DNN classifiers to various attacks [34, 42, 27]. For instance, adversarial examples leverage the difference between the classifier and human to misclassify specific inputs by adding imperceptible perturbations without altering the model. [12]. Such attacks during the inference phase are categorized as evasion attacks [22, 4]. On the other hand, poisoning attacks attempt to inject malicious data points or manipulate the training process to either degrade the model accuracy [31, 38, 50] or cause misclassification for specific inputs (a.k.a. backdoor attacks) [30, 28, 7, 13].

In general, backdoor attacks aim at injecting a malicious behavior into a DNN model so that the model would perform normally on clean inputs but yield misclassification in the presence of the backdoor trigger (e.g., a specific pattern such as a small square [13]). Later on, many works adopt the concepts and techniques in adversarial examples to improve the stealthiness of the trigger

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

against human observers [28, 2, 29]. Recent works have demonstrated power backdoor attacks that are capable of mounting attacks with visual indistinguishable backdoor images [23, 47, 49, 32]. Specifically, WaNet [32] generates backdoor images with warping transformation, resulting in much stealthier triggers.

To alleviate the threats originated from the ever-growing powerful backdoor attacks, several categories of countermeasures have also been developed. One promising direction for backdoor detection entails identifying backdoor images by characterizing the distinguishable dissimilarity in the feature or latent representation between backdoor images and clean images [5, 46, 44, 40, 35]. These methods rely on the assumption that the injected backdoor would leave a noticeable fingerprint in the latent space. For example, activation clustering [5] and spectral signature [46] detect malicious samples by inspecting the clusters of the latent space and the spectrum of the covariance of latent representations, respectively. Thus, a stronger adaptive backdoor attack should also ensure its stealthiness from the latent space. One prior work exploited this direction, Adversarial Embedding [43], which improves the latent indistinguishability of the backdoor attack by using adversarial regularization to minimize the distance between the latent distributions of the backdoor inputs and clean inputs.

To this end, this paper presents a novel methodology for **a backdoor attack that is imperceptible from both the input and latent spaces**. We propose a technique for generating imperceptible backdoor triggers at the input space and extend this concept to the latent space by minimizing the Wasserstein distance between the latent representations of the clean and backdoor data, which significantly improves the effectiveness against the existing defense mechanisms, especially those aforementioned that rely on the distinguishability in latent space. We call the proposed method **W**asserstein **B**ackdoor, or **WB**. Our technical **contributions** are summarized below:

- We propose a non-convex, constrained optimization problem, which learns to poison the classifier with a backdoor whose trigger is visually imperceptible in the input space and whose poisoned samples have indistinguishable latent distribution to the latent distribution of the clean samples. The latent constraint is formulated via a variant of Wasserstein distance, called sliced-Wasserstein distance, between the two sets of clean and backdoor data.
- We propose an efficient estimation of the sliced-Wasserstein distance by exploiting the discriminant directions of the trained classifier, instead of the randomly sampling from the unit sphere. The proposed distance is a valid distance metric and requires a significantly less computation and a better estimate than the existing calculations of the sliced-Wasserstein distance.
- Finally, we demonstrate the superior attack performance of the proposed method and its robustness against several representative defense mechanisms. Specifically, the proposed method outperforms the prior attack method with latent indistinguishability [43].

The rest of the paper is organized as follows. We review the background and related work in Section 2. In Section 3, we define the threat model. Section 4 presents the details of the proposed methodology. We evaluate the performance and compare to prior works in Section 5. Finally, Section 6 presents remarks and concludes this paper.

2 Background and Related Work

2.1 Backdoor Attack

Backdoor attacks against DNNs inject a malicious behavior by leveraging the redundancies inside the model such that the model responds to inputs with triggers maliciously (e.g., classify as a target class that normally considered as a wrong class by annotation), while preserving the benign behavior for clean inputs without the triggers. Hence, a typical backdoor embedding process is to train the model by minimizing the loss of the clean inputs and the corresponding labels as well as backdoor inputs (with triggers) and the target class(es). A trigger is typically applied on a clean image by superimposing at a certain location (i.e., patch-based) [14, 28] or adding perturbations [37]. Various forms of the triggers have been investigated in the literature, including blended [7], sinusoidal strips (SIG) [2], reflection (ReFool) [29], and warping-based (WaNet) [32]. As we mentioned above, several techniques have been developed that significantly reduce the visibility of the trigger in the input space to enhance the stealthiness of the backdoor attack [28, 2, 29]. In particular, WaNet uses a smooth warping field to generate backdoor images with unnoticeable modifications [32],

2.2 Backdoor Defense

By exploring specific characteristics of the injected backdoor, various countermeasures have been proposed [5, 46, 11, 40, 8, 6, 35], although they are often circumvented by following adaptive attacks. For instance, based on the property that a backdoor attack usually targets redundant weights or neurons based on the clean images, model pruning can be used to eliminate the injected backdoor [26]. Differently, Neural Cleanse assumes a known subset of clean inputs to reverse-engineer possible trigger patches [48]. It is also possible to filter the images to nullify the presence of triggers at the test phase to defend against backdoor attacks [30, 24].

In this paper, we focus on optimizing the characteristics of backdoor attacks in the latent space. As we discussed above, the rationale behind this is that prior works have demonstrated backdoor images cause distinctive activations in the latent space from those of clean inputs. Hence, this distinguishable dissimilarity between clean images and backdoor images can be utilized for defense in both training [5, 46] and test phases [41, 19, 18]. Most of these approaches compute an outlier score to detect abnormal inputs that will be filtered afterward. For example, spectral signature [46] computes the outlier score based on the singular value decomposition of the covariance matrix of the latent representations, while CleaNN [18] leverages a concentration inequality to detect anomalous reconstruction errors that are then suppressed before the input entering the victim DNN.

This work proposes a method to minimize the difference between clean images and backdoor images in the latent space to improve the attack stealthiness. While doing this, we also optimize the visual imperceptibility in the input space, so that our proposed method can visual inspection.

3 Threat Model

We consider the same threat model as in prior studies [7, 47, 37, 43, 32], which assume the backdoor injection is performed at training and the adversary can access to the victim model including both structures and parameters. A successful backdoor attack over an image classification task should produce malicious behavior on images with the trigger, while otherwise working normally on clean images. However, in typical backdoor attacks, the poisoned images are visually inconsistent with natural images, which can be identified easily by human observers. Besides, these attacks usually leave a tangible trace in the latent space of the poisoned classifier; thus, some defense methods can easily detect and discard the poisoned models. To this end, we propose a stronger backdoor attack where the poisoned images are crafted with imperceptible perturbation in the input space to clean images as well as unnoticeable trace in the latent space. We advance the state-of-the-art by significantly enhancing the imperceptibility and robustness of the backdoor attack.

4 Methodology

This section presents the details of the proposed Wasserstein Backdoor (WB).

4.1 Preliminaries

Consider the standard supervised classification task where one hopes to learn a mapping function $f_\theta : \mathcal{X} \longrightarrow \mathcal{C}$ where $\mathcal{X}$ is the input domain and $\mathcal{C}$ is the set of target classes. The task is to learn the parameters θ from the training dataset $\mathcal{S} = \{(x_i, y_i) : x_i \in \mathcal{X}, y_i \in \mathcal{C}, i = 1, .., N\}$.

Following the standard training scheme of backdoor attacks, the classifier is trained with the combination of the clean and poisoned subsets of S. To create a poisoned sample, a clean training sample (x, y) is transformed into a backdoor sample $(T(x), \eta(y))$, where T is a backdoor injection function (also called the trigger function) and η is the target label function. When training f with the clean and poison samples, we alter the behavior of f so that:

$$f(x) = y, \quad f(T(x)) = \eta(y), \tag{1}$$

for any pair of clean data $x \in \mathcal{X}$ and its corresponding label $y \in \mathcal{C}$. There are two commonly studied backdoor attack settings [13, 32, 43]: all-to-one and all-to-all. In all-to-one attack, the label is changed to a constant target, i.e. $\eta(y) = c$; for all-to-all attack, the true label is one-shifted, i.e. $\eta(y) = (y + 1)\ mod\ |\mathcal{C}|$. In the existing works, the trigger function T is selected before training f and fixed during the training process of f.

4.2 Learning to Backdoor

Given the training dataset $\mathcal{S}$ and a loss function $\mathcal{L}$, e.g., cross entropy loss, empirical risk minimization can be used to learn the parameters θ, as follows:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} \mathcal{L}(f_\theta(x_i), y_i).$$

The goal of this work is to learn a trigger function $T_\xi : \mathcal{X} \longrightarrow \mathcal{X}$ and a classification model f_θ in such a way that the clean image x and its corresponding backdoor image $T(x)$ are visually consistent in the input space while the backdoor attack does not leave a detectable trace in the latent space of the poisoned classifier. When f is a neural network, $\phi(x)$ can be the output of an intermediate, hidden layer of f, which captures some high-level abstraction of the input. Note that we require the classifier to perform normally on the clean sample, x, compared to the classifier's vanilla version, but change its prediction on the poisoned image, $T(x)$, to the target class $\eta(y)$.

To generate an imperceptible trigger and poison the image, we formulate the trigger function as a conditional noise generator g, as follows:

$$T_\xi(x) = x + g_\xi(x), \quad ||g_\xi(x)||_\infty \leq \epsilon \quad \forall x \tag{2}$$

The generator function g_ξ takes an input x and generates an artificially imperceptible noise on the same input space, which guarantees the stealthiness of the backdoor attack. We can design such generator function as an autoencoder or the more complex U-Net architecture [36].

With the above objectives and notations, we can formalize the task into the following constrained optimization problem:

$$\min_{\theta} \sum_{i=1}^{N} \alpha \mathcal{L}(f_\theta(x_i), y_i) + \beta \mathcal{L}(f_\theta(T_{\xi^*(\theta)}(x_i)), \eta(y_i)) \tag{3}$$

$$s.t. \quad \xi^* = \arg\min_{\xi} \sum_{i=1}^{N} \mathcal{L}(f_\theta(T_\xi(x_i)), \eta(y_i)) + \mathcal{R}_\phi(\mathcal{F}_c, \mathcal{F}_b)$$

where $\mathcal{R}_\phi$ is the regularization constraint of the clean and poisoned representations, denoted as $\mathcal{F}_c = \{\phi(x_i) : i = 1, .., N\}$ and $\mathcal{F}_b\{\phi(T(x_i)) : i = 1, .., N\}$, respectively.

In this problem, a learned classification model with a specific parameter configuration θ is associated with an optimal, stealthy backdoor trigger function, which is trained to poison the model. The classifier is trained to minimize a linear combination of clean and targeted backdoor objectives. The parameters α and β control the mixing strengths of the clean and backdoor loss signals. The trigger function is trained to perturb an image within its ℓ_∞ ball in the input space, so that the loss toward the attack target class is minimized while regularizing the latent representations of the backdoor images.

4.3 Stealthy Latent Representation via Wasserstein Regularization

In practical applications, latent-space defense methods study the abnormal trace of incoming data points with respect to the previous stream of data. These traces exist primarily because of the fact that the clean and backdoor latent representations are separated or distributed differently (e.g., the separated clusters of the clean and poison representations which can be seen in Figures 2 and 3). Thus, we aim to minimize such distributional difference through the regularization constraint $\mathcal{R}_\phi$. Since we cannot assume that the two latent distributions have common support or their density functions are known, commonly-used divergences, such as f-divergences (which include KL and JSD), are difficult to minimize. Instead, we consider the Wasserstein-2 distance and formulate the regularization constraint as follows:

$$\mathcal{R}\phi(\mu, \nu) = \left(\inf_{\gamma \in \Pi(\mu,\nu)} \int_{(x,z)\sim\gamma} p(x, z)||x - z||_2 dx dz \right)^{1/2} \tag{4}$$

where μ and ν are marginal probability measures defined by empirical samples $\mathcal{F}_c$ and $\mathcal{F}_b$ of the latent representations of the clean and poisoned data, respectively.

Estimating the Wasserstein distance also has some challenges. From the primal domain, computing the infimum in Equation (4) is particularly hard since the data distributions are not fixed or known. On the other hand, employing the Kantorovich-Rubinstein duality requires a separate, parameterized Lipschitz function and a minimax solver, which increases the complexity of the proposed problem. Fortunately, for one-dimensional continuous measures, the Wasserstein distance has an elegant yet closed-form solution. Let q_μ and q_ν be the corresponding density functions of μ and ν, respectively. The Wasserstein-2 distance between one-dimensional measures μ and ν is:

$$\mathcal{W}(\mu,\nu) = \left(\int_0^1 ||(F_\mu^{-1}(z) - F_\nu^{-1}(z)||_2 dz\right)^{1/2} \tag{5}$$

where $F_\mu(z) = \int_\infty^z q_\mu(\rho)d\rho$ and $F_\nu(z) = \int_\infty^z q_\nu(\rho)d\rho$ are the cumulative distribution functions. Inspired by the efficiency of this solution and its successful applications in a variety of tasks [9, 20, 10], we propose to first find a family of one-dimensional representations, e.g., through the linear projections, and approximate the Wasserstein distance as a function of these one-dimensional marginals, as follows:

$$\mathcal{R}_\phi(\mathcal{F}_c, \mathcal{F}_b) \approx \left(\frac{1}{L}\sum_{l=1}^{L}[\mathcal{W}(\mathcal{F}_c^{\theta_l}, \mathcal{F}_b^{\theta_l})]^2\right)^{1/2} \tag{6}$$

where $\mathcal{F}_c^{\theta_l} = \{\theta_l^T\phi(x_i) : i = 1,..,N\}$ and $\mathcal{F}_b^{\theta_l} = \{\theta_l^T\phi(T(x_i)) : i = 1,..,N\}$ contains the projections of the clean and poisoned datasets into a one-dimensional direction defined by θ_l (a slice). Typically, θ_l is drawn from a uniform distribution on the unit sphere. This formulation is also known as the sliced-Wasserstein distance (SWD) [9, 20]. One particular problem with this approach is that the random nature of the slices could lead to several non-informative directions; i.e., the sliced distances are close to 0 in directions that do not lie on the manifolds of the data. Consequently, a large number L of random directions are needed to approximate the sliced-Wasserstein distance, which increases the computational complexity of the estimation.

To remedy this issue, we avoid the uniform sampling of the unit sphere and select directions that contain discriminant information of the two data sources, by exploiting the following fact in the classification task. For backdoor samples of an attack-target class $c_1 \in \mathcal{C}$, created from clean samples of some other class $c_2 \in \mathcal{C}$, the projections into an output dimension represent meaningful discriminant information which distinguishes the backdoor samples (from class c_2) and the clean samples (from class c_1).Thus, we propose to replace the uniform linear projections of SWD with the projections into the output layer. When the latent space is the penultimate layer of the classifier, such projections are equivalent to the following approximation:

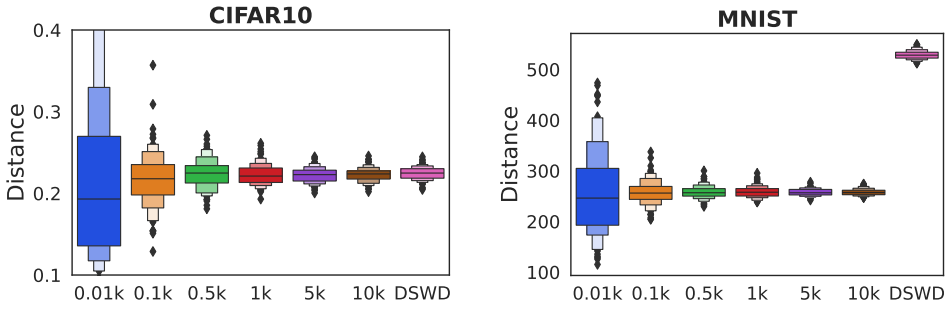
$$\mathcal{R}_\phi(\mathcal{F}_c, \mathcal{F}_b) \approx \left(\frac{1}{|\mathcal{C}|}\sum_{c=1}^{|\mathcal{C}|}\left[\mathcal{W}(\mathcal{F}_c^{W_{c,:}}, \mathcal{F}_c^{W_{c,:}})\right]^2\right)^{1/2}. \tag{7}$$

where $W_{c,:}$ is a row of the matrix $W \in \mathbb{R}^{|\mathcal{C}|\times d}$ (d is the dimension of the latent space) which is the normalized parameter matrix between the penultimate and the output layer.

Empirically, Figure 1 shows the estimated SWD with different numbers of random directions and the proposed calculation, so called DSWD, when the latent space is defined at the penultimate layer of the classifier. The dimension of the latent space is 512 for both MNIST and CIFAR10 datasets. Each distance is computed on a random sample of 1000 clean and 1000 backdoor images, and each calculation is repeated 100 times. As we can observe in this figure, with only a fraction of slices, DSWD achieves a significantly smaller variance than that of the SWD estimates. Furthermore, in MNIST, the selected directions of DSWD leads to higher distance estimates than SWD, which means that DSWD selects more discriminant directions than SWD while SWD underestimates the distance between the two empirical samples. In addition, we show that DSWD is a valid distance metric of the latent distributions.

Theorem 1 *When the latent space is the penultimate layer of a neural network, the proposed DSWD distance is a valid distance function of probability measures in this space.*

Remark 1 *Since existing defense methods choose the penultimate layer of a neural network. as the space to perform the defense analysis, in most cases, we can employ the proposed DSWD calculation.*



(a) Pre-activation Resnet-18 Model (b) CNN Model

Figure 1: Distance estimates in the latent space for SWD with different number of sampled directions (between 10 to 10,000) and DSWD.

Remark 2 *To preserve the clean classification performance, the classifier seeks optimal parameters that lead to similar predictions of clean samples from the same class. The goal of the trigger function is to make the poisoned samples classified toward a different class. This leads to an adversarial game between the classifier and the trigger functions.*

DSWD also has a significantly better computational efficiency than SWD. In most problems, SWD requires a large number of random directions, typically between 1000 to 10,000, in order to provide a reliable estimate of the distance [33, 10]. In DSWD, the number of random directions is fixed to the number of possible output labels, which is typically small for many classification problems.

4.4 Optimization

The non-convex, constrained optimization in Equation (3) is challenging because of its non-linear constraint. In general, we can alternately update f and T while keeping the other fixed, similar to training GANs. However, it is difficult and slow for the classifier to reach an acceptable performance on the clean data, i.e., similar to that of the vanilla classifier.

Under the alternating update scheme, we observe that on MNIST, the poisoned classifier can reach the acceptable clean-data performance after several epochs; while on the other more complex datasets (i.e. CIFAR10, GTSRB, and TinyImagenet), this procedure results in sub-optimal clean-data performance. One possible explanation is that training the vanilla classifier with complex architecture and dataset to reach a decent accuracy is already a difficult and time-consuming task (e.g., 2 to 3 epochs to reach the optimal performance on MNIST but several hundreds of epochs on the other datasets).

Fortunately, we observe that after training the classifier and the trigger functions in an alternating update scheme for a certain number of epochs (denoted as Stage I), we can freeze the trigger function and only train the classifier for the remaining epochs (denoted as Stage II). This two-stage training scheme is adopted in our experiments.

5 Experimental Results

5.1 Experimental Setup

We demonstrate the effectiveness of the proposed method through a range of experiments on four widely-used datasets for backdoor attack study: **MNIST**, **CIFAR10**, **GTSRB** and **TinyImagenet**. For these experiments, we follow the previous works [43, 46, 5, 32] and select the penultimate layers of the classifiers as the latent space for the defense experiments.

Architectures: For the classifier f, we consider several popular models: Pre-activation Resnet-18 [16], VGG [39], DenseNet [17] for CIFAR10 and GTSRB datasets, and Resnet-18 for TinyImagenet. For the MNIST dataset, we employ a CNN model.

Hyperparameters: For the baselines, we train the classifiers using the SGD optimizer with an initial learning rate of 0.01 and a learning rate decay of 0.1 after every 100 epochs. For other

hyperparameters, we follow the proposed setup in [32] for all datasets. We use the same configurations for WB. We alternately train the classifier and trigger functions alternately (Stage I) for 10 and 50 epochs for MNIST and the other datasets, respectively, and fine-tune the classifier (Stage II) for another 40 epochs and 450 epochs for MNIST and the other datasets, respectively. To achieve a high-degree stealthiness of WB, we pick ϵ as small as 0.01 on all datasets. In general, the larger the value of ϵ, the easier the trigger functions can be learned and the more successful the attacks are.

5.2 Attack Performance

In this experiment, we present the attack success rates of the proposed WB method and the state-of-the-art method, WaNet [32]. Wanet's attack performance is significantly better than other approaches, including BadNets [13], and is one of the strongest existing method that generates stealthy triggers on the images. We first poison the classifier using the backdoor attack methods in both all-to-one and all-to-all settings and record the performance of the classifier on both clean and backdoor test samples. For all-to-one, we randomly pick the target label $\hat{c}$ (i.e., $\eta(y) = \hat{c}\ \forall y$), while for all-to-all, the target label function is defined as $\eta(y) = (y+1)\ mod\ |\mathcal{C}|\ \forall y$, which is widely used to evaluate the backdoor-related works [32, 13, 5]. Note that this all-to-all attack setting is more challenging than the all-to-one setting, especially on datasets with a large number of classes such as TinyImagenet.

Table 1: Network Performance: All-to-one Attack

Dataset	WaNet		WB	
	Clean	Attack	Clean	Attack
MNIST	0.99	0.99	0.99	0.99
CIFAR10	0.94	0.99	0.94	0.99
GTSRB	0.99	0.98	0.99	0.99
TinyImagenet	0.57	0.99	0.57	0.99

Table 2: Network Performance: All-to-all Attack

Dataset	WaNet		WB	
	Clean	Attack	Clean	Attack
MNIST	0.99	0.95	0.99	0.96
CIFAR10	0.94	0.93	0.94	0.94
GTSRB	0.99	0.98	0.99	0.98
TinyImagenet	0.58	0.58	0.58	0.58

The classification accuracy on the clean test samples and the attack success rate for each method is represented in Tables 1 and 2 for the all-to-one and all-to-all settings, respectively. As we can observe from these tables, both WaNet and WB can achieve high clean-data accuracies. While both of the methods can perform the backdoor attacks with almost perfect success rates in most of the experiments, WB outperforms WaNet.

5.3 Latent-Space Defense

Recent backdoor defense methods have found that a backdoor attack tends to leave a tangible trace in the latent space of the poisoned classifier. Activation Clustering [5] and Spectral Signature [46] are two representative defenses used for analyzing the latent space in prior works [43]. In this section, we also look at the latent space of the poisoned classifiers through the lens of these defense methods.

5.3.1 Learned Latent Representation and Activation Clustering

It has been shown in [5] that in a poisoned classifier, the latent representations of the clean and backdoor samples form separate clusters, which can be easily detected using clustering methods such as K-means. The authors also recommend a process called exclusionary reclassification to determine which cluster is poisoned and re-train the poisoned classifier.

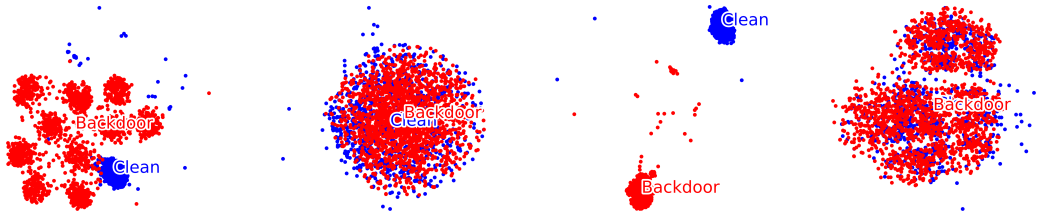
In Figure 2 and Figure 3, we can observe highly separated clusters (for samples with the sample predictions of $y = 0$) in the latent space when we omit the latent regularization term $\mathcal{R}_\phi$ in WB (Baseline). However, when $\mathcal{R}_\phi$ is included, the latent representations of the clean and backdoor samples are distributed similarly. Without well-separated clusters of the clean and poisoned samples, the exclusionary reclassification process in the activation clustering is not effective against the attacks.

Quantitatively, we present the quality scores (i.e., the adjusted Rand Index) of the clustering step in Table 3. The adjusted Rand Index is 1 when the samples form two distinct clusters and is close to 0 for a random separation. We compare WB with BadNets [13] and Adversarial Embedding [43],

Table 3: Adjusted Rand Index in All-to-one Attack

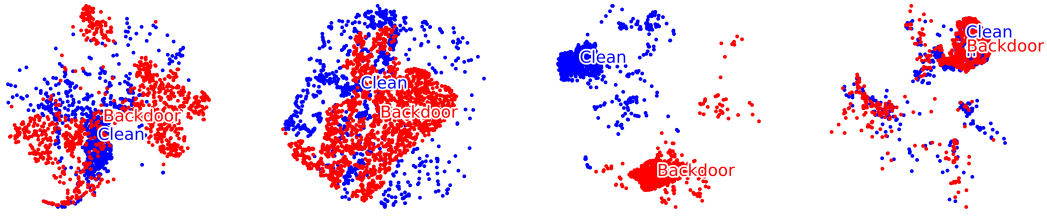
Model	Dataset	Rand Index (BadNets)	Adversarial Embedding		WB	
			Rand Index	Attack	Rand Index	Attack
DenseNet	CIFAR10	0.979	0.1820	0.764	0.0382	0.998
DenseNet	GTSRB	0.997	0.2710	0.914	0.0135	0.997
VGG	CIFAR10	0.998	0.0006	0.962	0.0002	0.999
VGG	GTSRB	0.997	0.6420	0.743	0.1010	0.999

which is the state-of-the-art backdoor attack method with stealthy latent space. As we can observe in this table, the defense is most successful on BadNets since there exists a perfect clustering of the clean and poisoned samples (Rand Index ≥ 0.95). While Adversarial Embedding is more resistant against the defense, WB is significantly more stealthy against the defense since the values of Rand Index are all very close to 0. Note that, similar to BadNets, WaNet does not pass this defense.



(a) All-to-one: Baseline (b) All-to-one: WB (c) All-to-all: Baseline (d) All-to-all: WB

Figure 2: MNIST: t-SNE embedding in the latent space. Baseline is WB without $\mathcal{R}_\phi$.



(a) All-to-one: Baseline (b) All-to-one: WB (c) All-to-all: Baseline (d) All-to-all: WB

Figure 3: CIFAR10: t-SNE embedding in the latent space. Baseline is WB without $\mathcal{R}_\phi$.

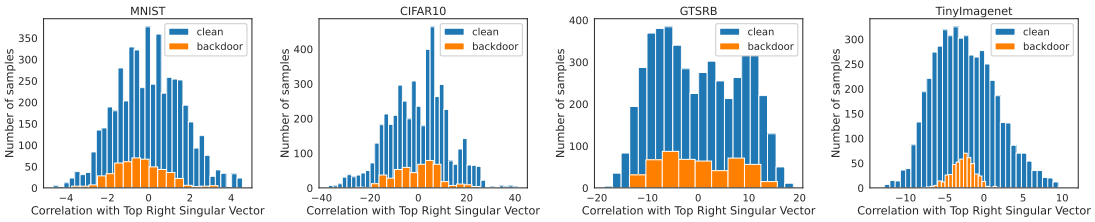
5.3.2 Spectral Signature Defense

The work in [46] proposes a defense method that identifies and removes backdoor samples using the Spectral Signature. For data from each predicted class, Spectral Signature first finds the top singular value of the covariance matrix of the latent vectors of the data. Then it computes the correlation score to this singular value for each sample and those samples with the outlier scores are flagged as backdoor samples. While Spectral Signature is a sample filtering-based defense method, the inspection of the correlation scores can also be useful to verify whether there is a tangible trace in the latent space of the classifier.

Following the same experiments in [46], we first pick 5,000 clean samples and 500 backdoor samples for each dataset. Then, we plot the histograms of the correlation scores for both sets of samples. As we can observe in Figure 4, there is not a clear separation between the scores of the backdoor samples and those of the clean samples.

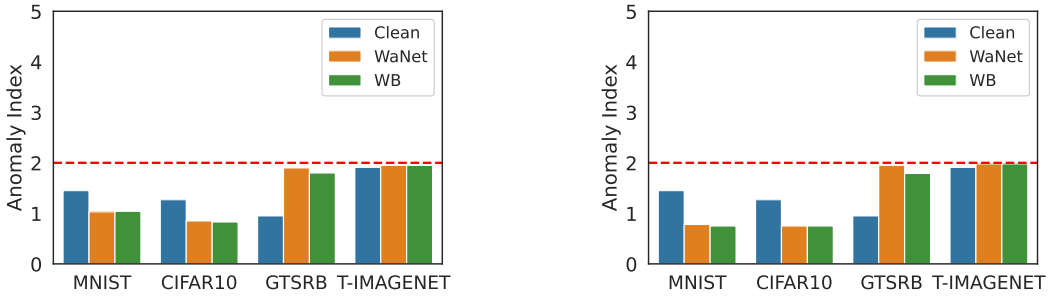
5.4 Model Mitigation Defense

In this section, we evaluate the robustness of WB against another popular defense, Neural Cleanse [48]. Neural Cleanse is model-mitigation defense based on a pattern optimization approach. Specifically,



(a) (b) (c) (d)

Figure 4: Defense experiments against Spectral Signature with all-to-one attack. The correlations of the clean and backdoor samples with the top singular vector of the covariance matrix *in the latent space are not separable*.



(a) All-to-one attacks (b) All-to-all attacks

Figure 5: Backdoor attacks against Neural Cleanse defense.

Neural Cleanse searches for the optimal patch pattern for each possible target label that induces a misclassification to that label. It then quantifies whether any of the optimal backdoor trigger pattern is an outlier via a metric called Anomaly Index. The model has a backdoor if the Anomaly Index is greater than 2 for any class.

The anomaly indices are presented in Figure 5. It can be seen that both WaNet and WB can pass the detection of Neural Cleanse, similar to that of the vanilla classifier (Clean). In MNIST and CIFAR10, WB even achieves smaller Anomaly Indices than those of the vanilla models. Note that popular backdoor attacks, such as BadNets, can be defended by Neural Cleanse in most of these datasets [48].

Additional experiments for demonstrating the robustness of WB against several other defense approaches can be found in the supplementary material.

6 Conclusion

This paper presented a novel methodology for a backdoor attack that is imperceptible from both the input and latent spaces, i.e., Wasserstein Backdoor (WB). WB learns a trigger function that adds visually imperceptible noise to an input image and minimizes the distributional difference via a novel sliced Wasserstein distance formulation between representations of the clean and backdoor images in the latent space of the trained classifier. We comprehensively evaluated the performance of the proposed method on various image classification benchmark models over a wide range of datasets. Our experimental results demonstrated that the proposed method could significantly improve the effectiveness against the existing defense mechanisms, especially those that rely on the distinguishability in latent space.

Societal Impacts: Our work on the backdoor attack is likely to increase the awareness and understanding of such vulnerability on neural networks. The proposed attacks, if not appropriately used, may bring security threats to the existing DNN based applications. We believe our study is an important step towards understanding the full capability of backdoor attacks. This knowledge will, in turn, facilitate the further development of secure and trustworthy DNN models and powerful defensive solutions. In this regard, we would encourage research to understand other aspects besides distinguishability in the input and latent spaces and further limitations of backdoor attack for developing countermeasures.

References

[1] Baltrusaitis, T., Zadeh, A., Lim, Y.C., Morency, L.P.: Openface 2.0: Facial behavior analysis toolkit. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). pp. 59–66. IEEE (2018)

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

[3] Berman, D.S., Buczak, A.L., Chavis, J.S., Corbett, C.L.: A survey of deep learning methods for cyber security. Information **10**(4), 122 (2019)

[4] Carlini, N., Wagner, D.: Adversarial examples are not easily detected: Bypassing ten detection methods. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. pp. 3–14. ACM (2017)

[5] Chen, B., Carvalho, W., Baracaldo, N., Ludwig, H., Edwards, B., Lee, T., Molloy, I., Srivastava, B.: Detecting backdoor attacks on deep neural networks by activation clustering. arXiv preprint arXiv:1811.03728 (2018)

[6] Chen, H., Fu, C., Zhao, J., Koushanfar, F.: Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. AAAI Press. pp. 4658–4664 (2019)

[7] Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint arXiv:1712.05526 (2017)

[8] Cheng, H., Xu, K., Liu, S., Chen, P.Y., Zhao, P., Lin, X.: Defending against backdoor attack on deep neural networks. arXiv preprint arXiv:2002.12162 (2020)

[9] Deshpande, I., Zhang, Z., Schwing, A.G.: Generative modeling using the sliced wasserstein distance. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3483–3491 (2018)

[10] Doan, K.D., Kimiyaie, A., Manchanda, S., Reddy, C.K.: Image hashing by minimizing independent relaxed wasserstein distance. arXiv preprint arXiv:2003.00134 (2020)

[11] Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D.C., Nepal, S.: Strip: A defence against trojan attacks on deep neural networks. In: Proceedings of the 35th Annual Computer Security Applications Conference. pp. 113–125 (2019)

[12] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. International Conference on Learning Representations (ICLR) (2015)

[13] Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733 (2017)

[14] Gu, T., Liu, K., Dolan-Gavitt, B., Garg, S.: Badnets: Evaluating backdooring attacks on deep neural networks. IEEE Access **7**, 47230–47244 (2019)

[15] Guo, W., Mu, D., Xu, J., Su, P., Wang, G., Xing, X.: Lemna: Explaining deep learning based security applications. In: proceedings of the 2018 ACM SIGSAC conference on computer and communications security. pp. 364–379 (2018)

[16] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[17] Hosseini, H., Poovendran, R.: Semantic adversarial examples. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1614–1619 (2018)

[18] Javaheripi, M., Samragh, M., Fields, G., Javidi, T., Koushanfar, F.: Cleann: Accelerated trojan shield for embedded neural networks. In: 2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD). pp. 1–9. IEEE (2020)

[19] Jin, K., Zhang, T., Shen, C., Chen, Y., Fan, M., Lin, C., Liu, T.: A unified framework for analyzing and detecting malicious examples of dnn models. arXiv preprint arXiv:2006.14871 (2020)

[20] Kolouri, S., Nadjahi, K., Simsekli, U., Badeau, R., Gustavo, K.: Generalized sliced wasserstein distances. In: NeurIPS 2019 (2019)

[21] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

[22] Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236 (2016)

[23] Li, S., Zhao, B.Z.H., Yu, J., Xue, M., Kaafar, D., Zhu, H.: Invisible backdoor attacks against deep neural networks. arXiv preprint arXiv:1909.02742 (2019)

[24] Li, Y., Zhai, T., Wu, B., Jiang, Y., Li, Z., Xia, S.: Rethinking the trigger of backdoor attack. arXiv preprint arXiv:2004.04692 (2020)

[25] Liu, J., Deng, Y., Bai, T., Wei, Z., Huang, C.: Targeting ultimate accuracy: Face recognition via deep embedding. arXiv preprint arXiv:1506.07310 (2015)

[26] Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: Defending against backdooring attacks on deep neural networks. arXiv preprint arXiv:1805.12185 (2018)

[27] Liu, Q., Li, P., Zhao, W., Cai, W., Yu, S., Leung, V.C.: A survey on security threats and defensive techniques of machine learning: a data driven view. IEEE access **6**, 12103–12117 (2018)

[28] Liu, Y., Ma, S., Aafer, Y., Lee, W.C., Zhai, J., Wang, W., Zhang, X.: Trojaning attack on neural networks. Department of Computer Science Technical Reports (2017)

[29] Liu, Y., Ma, X., Bailey, J., Lu, F.: Reflection backdoor: A natural backdoor attack on deep neural networks. In: European Conference on Computer Vision. pp. 182–199. Springer (2020)

[30] Liu, Y., Xie, Y., Srivastava, A.: Neural trojans. In: IEEE International Conference on Computer Design (ICCD). pp. 45–48. IEEE (2017)

[31] Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E.C., Roli, F.: Towards poisoning of deep learning algorithms with back-gradient optimization. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. pp. 27–38. ACM (2017)

[32] Nguyen, A., Tran, A.: Wanet–imperceptible warping-based backdoor attack. arXiv preprint arXiv:2102.10369 (2021)

[33] Nguyen, K., Ho, N., Pham, T., Bui, H.: Distributional sliced-wasserstein and applications to generative modeling. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=QYjO70ACDK

[34] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: Proceedings of the 2016 IEEE European Symposium on Security and Privacy. pp. 372–387 (2016)

[35] Qiao, X., Yang, Y., Li, H.: Defending neural backdoors via generative distribution modeling. In: Advances in Neural Information Processing Systems. pp. 14004–14013 (2019)

[36] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)

[37] Saha, A., Subramanya, A., Pirsiavash, H.: Hidden trigger backdoor attacks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 11957–11965 (2020)

[38] Shafahi, A., Huang, W.R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., Goldstein, T.: Poison frogs! targeted clean-label poisoning attacks on neural networks. In: Advances in Neural Information Processing Systems. pp. 6103–6113 (2018)

[39] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[40] Soremekun, E., Udeshi, S., Chattopadhyay, S., Zeller, A.: Exposing backdoors in robust machine learning models. arXiv preprint arXiv:2003.00865 (2020)

[41] Subedar, M., Ahuja, N., Krishnan, R., Ndiour, I.J., Tickoo, O.: Deep probabilistic models to detect data poisoning attacks. arXiv preprint arXiv:1912.01206 (2019)

[42] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. International Conference on Learning Representations (ICLR) (2013)

[43] Tan, T.J.L., Shokri, R.: Bypassing backdoor detection algorithms in deep learning. arXiv preprint arXiv:1905.13409 (2019)

[44] Tang, D., Wang, X., Tang, H., Zhang, K.: Demon in the variant: Statistical analysis of dnns for robust backdoor contamination detection. In: 30th {USENIX} Security Symposium ({USENIX} Security 21) (2021)

[45] Tian, Y., Pei, K., Jana, S., Ray, B.: Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In: Proceedings of the 40th international conference on software engineering. pp. 303–314 (2018)

[46] Tran, B., Li, J., Madry, A.: Spectral signatures in backdoor attacks. In: Advances in Neural Information Processing Systems. pp. 8000–8010 (2018)

[47] Turner, A., Tsipras, D., Madry, A.: Label-consistent backdoor attacks. arXiv preprint arXiv:1912.02771 (2019)

[48] Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y.: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 707–723. IEEE (2019)

[49] Zhong, H., Liao, C., Squicciarini, A.C., Zhu, S., Miller, D.: Backdoor embedding in convolutional neural network models via invisible perturbation. In: Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy. pp. 97–108 (2020)

[50] Zhu, C., Huang, W.R., Shafahi, A., Li, H., Taylor, G., Studer, C., Goldstein, T.: Transferable clean-label poisoning attacks on deep neural nets. arXiv preprint arXiv:1905.05897 (2019)

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 4 and supplementary material.
   (b) Did you include complete proofs of all theoretical results? [Yes] See supplementary material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Figure 1 is one example.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See the supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] All sources are cited.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code is submitted as supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]