Augmented Neural Fine-Tuning for Efficient Backdoor Purification

Nazmul Karim[1,★]⬤, Abdullah Al Arafat[2,★]⬤, Umar Khalid[1]⬤, Zhishan Guo[2]⬤, and Nazanin Rahnavard[1]⬤

[1]University of Central Florida   [2]North Carolina State University

Abstract. Recent studies have revealed the vulnerability of deep neural networks (DNNs) to various backdoor attacks, where the behavior of DNNs can be compromised by utilizing certain types of triggers or poisoning mechanisms. State-of-the-art (SOTA) defenses employ too-sophisticated mechanisms that require either a computationally expensive adversarial search module for reverse-engineering the trigger distribution or an over-sensitive hyper-parameter selection module. Moreover, they offer sub-par performance in challenging scenarios, *e.g.*, limited validation data and strong attacks. In this paper, we propose—*Neural mask Fine-Tuning (NFT)*—with an aim to optimally re-organize the neuron activities in a way that the effect of the backdoor is removed. Utilizing a simple data augmentation like MixUp, NFT relaxes the trigger synthesis process and eliminates the requirement of the adversarial search module. Our study further reveals that direct weight fine-tuning under limited validation data results in poor post-purification clean test accuracy, primarily due to *overfitting issue*. To overcome this, we propose to fine-tune neural masks instead of model weights. In addition, a *mask regularizer* has been devised to further mitigate the model drift during the purification process. The distinct characteristics of NFT render it highly efficient in both runtime and sample usage, as it can remove the backdoor even when a single sample is available from each class. We validate the effectiveness of NFT through extensive experiments covering the tasks of image classification, object detection, video action recognition, 3D point cloud, and natural language processing. We evaluate our method against 14 different attacks (LIRA, WaNet, *etc.*) on 11 benchmark data sets (ImageNet, UCF101, Pascal VOC, ModelNet, OpenSubtitles2012, *etc.*). Our code is available online in this GitHub Repository.

1   Introduction

Backdoor attack [8,16] on deep neural network (DNN) models is a heavily studied branch of AI safety and robustness. Backdoor defenses can generally be categorized into two major groups based on whether the defense is done during training or test. Training time defense (*e.g.*, [23,25,47]) focuses on training a benign model on the poisonous data, while test time defense (*e.g.*, [5,27,32,52]) deals

---

★ Equal Contribution

with purifying backdoor model after it has already been trained. In this work, our goal is to develop an efficient test-time defense. Some test-time defenses focus on synthesizing trigger patterns [6, 49, 50] followed by vanilla weight fine-tuning. Most of these defenses aim to synthesize class-specific triggers independently or use additional models to generate the triggers simultaneously. Recent state-of-the-art (SOTA) backdoor defense methods, *e.g.*, ANP [52], I-BAU [57] AWM [5], employ very similar techniques and do not work well without an expensive *adversarial search module*. For example, ANP [52] performs an adversarial search to find vulnerable neurons responsible for backdoor behavior. Identifying and pruning vulnerable neurons require an exhaustive adversarial search, resulting in high computational costs. Similar to ANP, AWM and I-BAU also resort to trigger synthesizing with a modified adversarial search process. Another very recent technique, FT-SAM [64] uses sharpness-aware minimization (SAM) [14] to fine-tune model weights. SAM is a recently proposed optimizer that utilizes Stochastic Gradient Descent (SGD), which penalizes sudden changes in the loss surface by constraining the search area to a compact region. Since SAM performs a double forward pass to compute the loss gradient twice, it results in a notable runtime increase for FT-SAM. In our work, *we aim to create an effective backdoor defense system that neither requires an expensive adversarial search process to recover the trigger nor a special type of optimizer with a runtime bottleneck.*

To achieve this goal, we propose a simple yet effective approach Neural mask Fine-Tuning (NFT), to remove backdoor through augmented fine-tuning of cost-efficient neural masks. We start with replacing the expensive adversarial search-based *trigger synthesis* process with a simple data augmentation technique—MixUp [58]. As the backdoor is usually inserted by forcing the model to memorize the trigger distribution, synthesizing and later unlearning that trigger would effectively remove the backdoor. In this work, we show that *unlearning can be performed by simply optimizing the MixUp loss over a clean validation set.* Our theoretical analysis suggests that MixUp loss works as an upper bound on the standard loss obtained from triggered (synthesized or already known) validation data, termed as *ideal purification loss*. As the minimization of ideal loss guarantees backdoor purification, minimizing the MixUp loss would effectively remove the backdoor (Sec. 4.1). As the next step of our method, we address the overfitting issue during weight fine-tuning under *limited validation data*. In general, the outcome of such overfitting is poor post-purification test accuracy which is not desirable for any backdoor defense. To this end, we propose to fine-tune a set of neural masks instead of the model weights, as this type of soft-masking enables us to reprogram the neurons affected by the backdoor without significantly altering the original backdoor model. As an added step to this, a mask regularizer has been introduced to further mitigate model drift during the purification process. In addition, we deploy a mask scheduling function to have better control over the purification process. Our experimental results indicate that these straightforward yet intuitive steps significantly improve the post-purification test accuracy as compared to previous SOTA. Our contributions can be summarized as follows:

- We propose a novel backdoor removal framework utilizing simple MixUp-based model fine-tuning. Our thorough analysis shows how minimizing the MixUp loss eliminates the requirement of an expensive trigger synthesis process while effectively removing the backdoor (Sec. 4.1).
- To preserve the post-purification test accuracy, we propose to fine-tune soft neural masks (instead of weights) as it prevents any drastic change in the original backdoor model (Sec. 4.2). Additionally, a novel mask regularizer has been introduced that further encourages the purified model to retain the class separability of the original model. In addition to being computationally efficient, our proposed method shows significant improvement in sample efficiency as it can purify backdoor even with one-shot fine-tuning, i.e., only a single sample is available from each class (Sec. 4.3).
- To show the effectiveness of NFT, we perform an extensive evaluation with 11 different datasets. Compared to previous SOTA, the superior performance against a wide range of attacks suggests that augmentation like MixUp can indeed replace the *trigger synthesis* process (Sec. 5).

2   Related Work

Backdoor Attack. Neural networks are intrinsically vulnerable to backdoor attacks [36, 55]. A substantial number of studies have investigated the possibility of backdoor attacks after the initial studies [8, 16, 34] found the existence of backdoors in DNNs. Generally, backdoor attacks are categorized into two types: clean-label attacks and poison-label attacks. A clean-label backdoor attack does not alter the label [39, 48, 62], while a poison-label attack aims at specific target classes such that the DNN misclassifies to those classes in the presence of a trigger [26]. As for trigger types, researchers have studied numerous types of triggering patterns in their respective attacks [8, 12, 16, 28]. Such triggers can exist in the form of dynamic patterns [28] or as simple as a single pixel [47]. Some of the more complex backdoor triggers that have been proposed in the literature are sinusoidal strips [2], adversarial patterns [62], and blending backgrounds [8]. Besides, backdoor attacks exist for many different tasks, *e.g.*, multi-label clean image attack [7], domain adaptation [1], *etc.*

Backdoor Defense. Generally, the backdoor defense methods are categorized into two types: Training Time Defense, and Test Time Defense [26, 32, 53]. Regarding training time defense techniques [15, 47] to mitigate the backdoor attack, the researchers have proposed numerous defense methods through input pre-processing [34], poison-suppression [18], model diagnosis [28], network pruning [34, 53], and model reconstruction [61]. DeepSweep [41] explores different augmentations to purify a backdoor model and rectify the triggered samples. Although DeepSweep revealed that different augmentation functions could be leveraged to invert the backdoor effect of a model or erase the trigger from a trigger-embedded image, our work re-purposes the usage of augmentation differently to cover the approximate (unknown) trigger distribution during the purification phase. Moreover, DeepSweep assumes that backdoor triggers are

known to the defender, which is hardly a practical assumption. Instead of reverse-engineering of backdoor triggers in the input samples, several recent works explored the model vulnerability/sensitivity towards adversarially perturbed neurons [52], weights [5], or network channels [63]. However, these approaches require expensive adversarial search processes to be effective. A concurrent work FIP [19] studied the loss-surface smoothness of the backdoor model and developed a purification method by regularizing the spectral norm of the model.

3   Threat Model

Attack Model. Our work considers the most commonly used data poisoning attacks. Consider $\mathbb{D}_{\text{train}} = \{x_i, y_i\}_{i=1}^N$ as the training data where $x_i \in \mathbb{R}^d$ is an input sample labeled as $y_i \in \{0, \ldots, c-1\}$ sampled from unknown distribution $\mathcal{D}$ of the task to be learned. Here, N is the total number of samples, and d is the dimension of the input sample. In addition, we assume there are c number of classes in the input data. Let $f_{\theta^*} : \mathbb{R}^d \to \mathbb{R}^c$ be a benign (ideal) DNN trained with $\mathbb{D}_{\text{train}} \sim \mathcal{D}$. Here, θ^* is the DNN parameters that is to be optimized using a suitable loss function $\ell(.,.)$. The total empirical loss can be defined as,

$$\mathcal{L}(\theta^*, \mathbb{D}_{\text{train}}) = \frac{1}{N} \sum_{i=1}^N [\ell(y_i, f_{\theta^*}(x_i))], \tag{1}$$

Now, consider an adversary inserts backdoor $f_\theta(.)$ through modifying a small subset of $\mathbb{D}_{\text{train}}$ as $\{\hat{x}_i, \hat{y}_i\}$ such that $\hat{y}_i = \arg\max f_\theta(\hat{x}_i)$ preserving $y_i = \arg\max f_\theta(x_i)$, $\forall (x_i, y_i) \in \mathbb{D}_{\text{train}}$. Here, $\hat{x}_i = x_i + \delta$ is the triggered input with adversary set target label $\hat{y}_i \neq y_i$, where $\delta \in \mathbb{R}^d$ represents trigger pattern.

Defense Objective. Consider a defense model where defender removes backdoor from $f_\theta(.)$ using a small validation data $\mathbb{D}_{\text{val}} = \{x_i, y_i\}_{i=1}^{N_{val}}$ such that $y_i = \arg\max f_{\theta_c}(\hat{x}_i)$; where $y_i \neq \hat{y}_i$.

4   Neural Fine-Tuning (NFT)

Let us consider a fully-connected DNN, $f_\theta : \mathbb{R}^d \to \mathbb{R}^c$, that receives a datapoint $x \in \mathbb{R}^d$ and predicts a probability distribution $p \in \mathbb{R}^c$; where c is the number of classes. In general, x goes through a multi-layer weight architecture before the DNN model predicts an output class i (p_i is the probability for class i). Let us consider a multi-layer architecture with L layers in which the l-th layer contains k_l neurons. The neurons of each layer produce activations $\psi_l \in \mathbb{R}^{k_l}$ based on the output activations of previous layer $\psi_{l-1} \in \mathbb{R}^{k_{l-1}}$. To be specific,

$$\psi_l := \sigma(\Theta_l^T \cdot \psi_{l-1} + b_l), \tag{2}$$

where $\sigma(.)$ is a non-linear activation function, matrix $\Theta_l = [\Theta_l^{(1)} \cdots \Theta_l^{(k_l)}] \in \mathbb{R}^{k_{l-1} \times k_l}$, for $l = 1, 2, \ldots, L$, includes the weights of the l-th layer, and $b_l \in \mathbb{R}^{k_l}$ is the bias vector. Here, $\Theta_l^{(j)} \in \mathbb{R}^{k_{l-1}}$ denotes the weights vector corresponding

to the activation of the j-th neuron of the l-th layer. Model parameters θ can be expressed as $\theta = \{\Theta_1, \ldots, \Theta_L\}$. This type of multi-layer architecture is also valid for convolutional neural networks, where we use multiple 2-D arrays of neurons (*i.e.*, filters) instead of a 1-D array.

4.1   Backdoor Suppressor

Our objective is to make the backdoor model forget about poison distribution while retaining the knowledge of clean distribution. To understand how we achieve this objective, let us first revisit the process of generating triggered data ($\hat{x}$). In general, $\hat{x}$ is created by adding minor modifications (*i.e.*, adding triggers δ) to clean data (x). Note that the backdoor is inserted by forcing the model to learn the mapping, $\hat{x} \to \hat{y}$. Here, $(\hat{x}, \hat{y})$ is the poison data. If we were to change the mapping from $(\hat{x} \to \hat{y})$ to $(\hat{x} \to y)$, we would have a robust clean model instead of a backdoor model. This is because, in this case, the model would treat δ as one type of augmentation and $\hat{x}$ as the augmented clean data. In summary, the backdoor insertion process functions as an augmentation process if we simply use y instead of $\hat{y}$. Now, ideally, fine-tuning the backdoor model with triggered data with corresponding ground truth labels (*i.e.*, $\{\hat{x}, y\}$) would remove the backdoor effect from the model. Let us consider this ideal scenario where we have access to the trigger δ during purification. We define the *ideal purification loss* as

$$\mathcal{L}^{\text{ideal}}(\theta, \mathbb{D}_{\text{val}}) = \frac{1}{N_{\text{val}}} \sum_{i=1}^{N_{\text{val}}} \ell(y_i, f_\theta(\hat{x}_i)), \qquad (3)$$

where $\hat{x} = x + \delta$ and y is the ground truth label of x. In our work, as we do not have access to δ or adversarially reverse engineer it [5,52,57], we relax this process by strongly augmenting the clean validation data, *i.e.*, creating augmented $\mathbb{D}_{\text{val}}$ with already known augmentation technique such as MixUp [58]. For MixUp, we can easily perform $\tilde{x}_{i,j} = \lambda x_i + (1 - \lambda)x_j$ and $\tilde{y}_{i,j} = \lambda y_i + (1 - \lambda)y_j$ for $\lambda \in [0, 1]$; here $\tilde{y}_{i,j}$ represents the linear combination of one-hot vectors corresponding to y_i and y_j. The loss after the MixUp becomes

$$\mathcal{L}^{\text{mix}}(\theta, \mathbb{D}_{\text{val}}) = \frac{1}{N_{\text{val}}^2} \sum_{i,j=1}^{N_{\text{val}}} \mathop{\mathbb{E}}_{\lambda \sim \mathcal{D}_\lambda} \ell(\tilde{y}_{i,j}, f_\theta(\tilde{x}_{i,j})), \qquad (4)$$

where $\mathcal{D}_\lambda$ is a distribution supported on $[0, 1]$. In our work, we consider the widely used $\mathcal{D}_\lambda$ – Beta distribution $Beta(\alpha, \beta)$ for $\alpha, \beta > 0$. We provide both empirical (Sec. 5) and theoretical proof on why minimizing Eq. 4 would effectively remove the backdoor.

Theoretical Justifications. For a fully-connected neural network (NN) with logistic loss $\ell(y, f_\theta(x)) = \log(1 + \exp(f_\theta(x))) - y f_\theta(x)$ with $y \in \{0, 1\}$, it can be shown that $\mathcal{L}^{\text{mix}}(\theta, \mathbb{D}_{\text{val}})$ is an upper-bound of the second order Taylor expansion

of the ideal loss $\mathcal{L}^{\text{ideal}}(\theta, \mathbb{D}_{\text{val}})$. With the nonlinearity σ for ReLU and max-pooling in NN, the function f_θ satisfies that $f_\theta(x) = \nabla f_\theta(x)^T x$ and $\nabla^2 f_\theta(x) = 0$ almost everywhere, where the gradient is taken with respect to the input x.

We first rewrite the $\mathcal{L}^{\text{ideal}}(\theta, \mathbb{D}_{\text{val}})$ using Taylor series approximation. The second-order Taylor expansion of $\ell(y, f_\theta(x + \delta))$ is given by,

$$\ell(y, f_\theta(x+\delta)) = \ell(y, f_\theta(x)) + (g(f_\theta(x)) - y)(f_\theta(\delta)) + \frac{1}{2} g(f_\theta(x))(1 - g(f_\theta(x)))(f_\theta(\delta))^2,$$

where $g(x) = \frac{e^x}{1+e^x}$ is the logistic function. Based on the MixUp related analysis in prior works [4, 59], the following can be derived for $\mathcal{L}^{\text{mix}}(\theta, \mathbb{D}_{\text{val}})$ using the second-order Taylor series expansion,

Lemma 1. *Assuming $f_\theta(x) = \nabla f_\theta(x)^T x$ and $\nabla^2 f_\theta(x) = 0$ (which are satisfied by ReLU and max-pooling activation functions), $\mathcal{L}^{\text{mix}}(\theta, \mathbb{D}_{\text{val}})$ can be expressed as,*

$$\mathcal{L}^{\text{mix}}(\theta, \mathbb{D}_{\text{val}}) = \mathcal{L}(\theta, \mathbb{D}_{\text{val}}) + \mathcal{R}_1(\theta, \mathbb{D}_{\text{val}}) + \mathcal{R}_2(\theta, \mathbb{D}_{\text{val}}) \tag{5}$$

where,

$$\mathcal{R}_1(\theta, \mathbb{D}_{\text{val}}) \geq \frac{R c_x \, \mathbb{E}_\lambda[(1-\lambda)] \sqrt{d}}{N_{\text{val}}} \sum_{i=1}^{N_{\text{val}}} |g(f_\theta(x_i)) - y_i| \cdot ||\nabla f_\theta(x_i)||_2$$

$$\mathcal{R}_2(\theta, \mathbb{D}_{\text{val}}) \geq \frac{R^2 c_x^2 \, \mathbb{E}_\lambda[(1-\lambda)]^2 d}{2 N_{\text{val}}} \sum_{i=1}^{N_{\text{val}}} |g(f_\theta(x_i))(1 - g(f_\theta(x_i)))| \cdot ||\nabla f_\theta(x_i)||_2^2,$$

where $R = \min_{i \in [N_{\text{val}}]} \langle \nabla f_\theta(x_i), x_i \rangle / ||\nabla f_\theta(x_i)|| \cdot ||x_i||$ and $c_x > 0$ is a constant.

By comparing $\ell(y, f_\theta(x + \delta))$ and $\mathcal{L}^{\text{mix}}(\theta, \mathbb{D}_{\text{val}})$ for a fully connected NN, we can prove the following.

Theorem 1. *Suppose that $f_\theta(x) = \nabla f_\theta(x)^T x$, $\nabla^2 f_\theta(x) = 0$ and there exists a constant $c_x > 0$ such that $||x_i||_2 \geq c_x \sqrt{d}$ for all $i \in \{1, \ldots, N_{\text{val}}\}$. Then, for any f_θ, we have*

$$\mathcal{L}^{\text{mix}}(\theta, \mathbb{D}_{\text{val}}) \geq \frac{1}{N_{\text{val}}} \sum_{i=1}^{N_{\text{val}}} \ell(y_i, f_\theta(x_i + \varepsilon_i)) \geq \frac{1}{N_{\text{val}}} \sum_{i=1}^{N_{\text{val}}} \ell(y_i, f_\theta(x_i + \varepsilon))$$

where $\varepsilon_i = R_i c_x \, \mathbb{E}_{\lambda \sim \mathcal{D}_\lambda}[1 - \lambda] \sqrt{d}$ with $R_i = \langle \nabla f_\theta(x_i), x_i \rangle / ||\nabla f_\theta(x_i)|| \cdot ||x_i||$ and $\varepsilon = \min\{\varepsilon_i\}$.

Proof. is in *Supplementary.*

Theorem 1 implies that as long as $||\delta|| \leq \varepsilon$ holds, the MixUp loss $\mathcal{L}^{\text{mix}}(\theta, \mathbb{D}_{\text{val}})$ can be considered as an upper-bound of $\mathcal{L}^{\text{ideal}}(\theta, \mathbb{D}_{\text{val}})$.

4.2   Clean Accuracy Retainer

In practice, it is desirable for a backdoor defense technique to be highly *runtime efficient* and retain the *clean test accuracy* of the original model. For better runtime efficiency and to retain clean accuracy, we choose to apply neural mask fine-tuning instead of fine-tuning the entire model, which can be formulated as,

$$\widehat{M} = \underset{M \mid m_l^{(i)} \in [\mu(l), 1], \, \forall l, i}{\arg \min} \mathcal{L}^{\text{mix}}(\theta \odot M, \mathbb{D}_{\text{val}}) \tag{6}$$

We only optimize for neural masks M using $\mathbb{D}_{\text{val}}$ and define $\theta \odot M$ as,

$$\theta \odot M := \{\Theta_1 \odot M_1, \Theta_2 \odot M_2, \ldots, \Theta_L \odot M_L\}, \tag{7}$$

where $\theta := \{\Theta_1, \ldots, \Theta_L\}$, $M := \{M_1, \ldots, M_L\}$, and $M_l = [m_l^{(1)} \cdots m_l^{k_l}]^T \in \mathbb{R}^{k_l}$. We have

$$\Theta_l \odot M_l := [m_l^{(1)} \Theta_l^{(1)} \cdots m_l^{(k_l)} \Theta_l^{(k_l)}] \in \mathbb{R}^{k_{l-1} \times k_l}. \tag{8}$$

Note, in Eq. (8), a **scalar mask** $m_l^{(i)}$ is applied to the weight vector $\Theta_l^{(i)}$ corresponding to the ith neuron of lth layer. In our work, we formulate a constraint optimization problem where M depends on a mask scheduling function $\mu(l) : [1, L] \to [0, 1]$. Notice that $\mu(l)$ provides the lower limit of possible M_l's for the lth layer's neurons' mask. We find the suitable function for $\mu(l)$ by analyzing commonly used mathematical functions (*e.g.*, cosine, logarithmic, cubic, *etc.*). We analyze the impact of these functions in Section 5.3 and choose an exponential formulation (*e.g.*, $\alpha \cdot e^{-\beta \cdot l}$) for $\mu(l)$ since it produces the best possible outcome in terms of backdoor removal performance. Such formulation significantly reduces the mask search space, which leads to reduced runtime. Furthermore, the overall formulation for M encourages relatively small changes to the original backdoor model's parameters. This helps us retain on-par clean test accuracy after backdoor removal, which is highly desirable for a defense technique. To this end, we aim to purify the backdoor model by optimizing for the best possible mask $\widehat{M}$ that suppresses backdoor-affected neurons and bolsters the neurons responsible for clean test accuracy. Here, $\widehat{M}$ should give us a purified model as, $f_{\theta_c}(.)$, where $\theta_c = \theta \odot \widehat{M}$. Noteworthily, we do not change the bias as it may harm the classification accuracy.

4.3   Sample Efficiency of NFT

In this section, we discuss how careful modifications to the optimization scenario can make NFT highly sample-efficient. For this analysis, we first train a backdoor model (PreActResNet-18 [17]) on a poisoned CIFAR10 dataset for 200 epochs. For poisoning the dataset, we use TrojanNet [33] backdoor attack with a poison rate of 10%. In Fig. 1, we show t-SNE visualization of different class clusters obtained using the backdoor model. Instead of 10 clusters, we have an additional cluster (red color, labeled "11") that sits closely with the original target class cluster (in this case, the target class is "0"). We name this cluster "poison cluster",
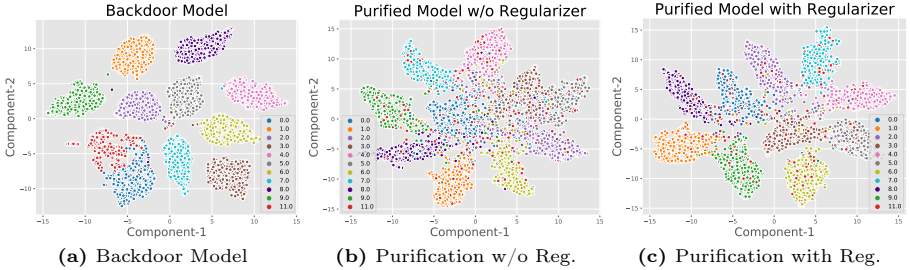
(a) Backdoor Model   **(b)** Purification w/o Reg.   **(c)** Purification with Reg.

Fig. 1: t-SNE visualization of a backdoor model, where we show the "*poison cluster*" with red color and label "11". Since the attack target class is "0", cluster "0" and the poison cluster sit closely with each other (Fig. 1a). After purification, the cluster should break, and all triggered samples should be classified according to their original label. In Fig. 1b, we perform one-shot NFT without employing the regularizer. Due to the overfitting issue, the clean clusters lose their separability that can be established with *Mask regularizer*, which tackles this issue (larger cluster gaps as compared to scenarios in Fig. 1b) by keeping purified model parameters close to the original backdoor model (Fig. 1c). This, in turn, produces better clean test accuracy. For evaluation, we train a PreActResNet18 [17] on CIFAR10 dataset with a poison rate of 10%.

whereas other clusters are "clean clusters". This cluster contains the embeddings of attacked or triggered samples from all other classes. The goal of any defense system is to break the formation of the poison cluster so that poison samples return to their original clusters.

One-Shot NFT. Let us consider that there is only 1 sample (per class) available for the validation set $\mathbb{D}_{\text{val}}$. Applying NFT with this $\mathbb{D}_{\text{val}}$ forms the clusters shown in Fig. 1b. Notice that the poison cluster breaks even with one-shot fine-tuning, indicating the backdoor's effect is removed successfully. However, since only one sample is available per class, the model easily overfits $\mathbb{D}_{\text{val}}$, reducing margins between clean clusters. Such unwanted *overfitting* issue negatively impacts the clean test accuracy. To combat this issue in scenarios where very few samples are available, we add a simple regularizer term as follows,

$$\underset{M \mid m_l^{(i)} \in [\mu(l), 1], \ \forall l, i}{\arg \min} \quad \mathcal{L}^{\text{mix}}(\theta \odot M, \mathbb{D}_{\text{val}}) + \eta_c ||M_0 - M||_1 \qquad (9)$$

where M_0 are the initial mask values (initialized as 1's) and η_c is the regularizer coefficient. By minimizing the ℓ_1-norm of the mask differences, we try to keep the purified model parameters $(\theta \odot M)$ close to the original backdoor model parameters $(\theta \odot M_0)$. We hope to preserve the original decision boundary between clean clusters by keeping these parameters as close as possible. Note that the overfitting issue is not as prominent whenever we have a reasonably sized $\mathbb{D}_{\text{val}}$ (*e.g.*, 1% of $\mathbb{D}_{\text{train}}$). Therefore, we choose the value of η_c to be $5e^{-4}/n_c$, which dynamically changes based on the number of samples available per class (n_c). As the number of samples increases, the impact of the regularizer reduces. *Note that, Eq. (9) represents the final optimization function for our proposed method.*

Table 1: Comparison of different defense methods for **CIFAR10 and ImageNet**. Average drop (↓) indicates the % changes in ASR/ACC compared to the baseline, *i.e.*, ASR/ACC of *No Defense*. A good defense should have a large *ASR drop* with a small *ACC drop*. Attacks are implemented with a poison rate of 10%.

Dataset	Method	No Defense		ANP		I-BAU		AWM		FT-SAM		RNP		NFT (Ours)	
	Attacks	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC
CIFAR-10	*Benign*	0	95.21	0	92.28	0	93.98	0	93.56	0	93.80	0	93.16	0	94.10
	Badnets	100	92.96	4.87	85.92	2.84	85.96	5.72	87.85	4.34	86.17	2.75	88.46	1.74	90.82
	Blend	100	94.11	4.77	87.61	3.81	89.10	5.53	90.84	2.13	88.93	0.91	91.53	0.31	93.17
	Troj-one	100	89.57	3.78	82.18	5.47	86.20	6.91	87.24	5.41	86.45	3.84	87.39	1.64	87.71
	Troj-all	100	88.33	3.91	81.95	5.53	84.89	6.82	85.94	4.42	84.60	4.02	85.80	1.79	87.10
	SIG	100	88.64	1.04	81.92	0.37	83.60	4.12	83.57	0.90	85.38	0.51	86.46	0.12	87.16
	Dyn-one	100	92.52	4.73	88.61	1.78	87.26	7.48	91.16	3.35	88.41	8.61	90.05	1.37	90.81
	Dyn-all	100	92.61	4.28	88.32	2.19	84.51	7.30	89.74	2.46	87.72	10.57	90.28	1.42	91.53
	CLB	100	92.78	**0.83**	87.41	1.41	85.07	5.78	86.70	1.89	84.18	6.12	90.38	1.04	90.37
	CBA	93.20	90.17	27.80	83.79	45.11	85.63	36.12	85.05	38.81	85.58	17.72	86.40	21.60	87.97
	FBA	100	90.78	7.95	82.90	66.70	87.42	10.66	87.35	22.31	87.06	9.48	87.63	6.21	88.56
	WaNet	98.64	92.29	5.81	86.70	3.18	89.24	7.72	86.94	2.96	88.45	8.10	90.26	2.38	89.65
	ISSBA	99.80	92.78	6.76	85.42	**3.82**	89.20	12.48	90.03	4.57	89.59	7.58	88.62	4.24	90.18
	LIRA	99.25	92.15	7.34	87.41	4.51	89.61	6.13	88.50	3.86	89.22	11.83	87.59	1.53	90.57
	BPPA	99.70	93.82	9.94	90.23	10.46	90.57	9.94	90.88	10.60	90.88	9.74	91.37	5.04	91.78
	Avg. Drop	-	-	92.63 ↓	5.94 ↓	88.10 ↓	4.66 ↓	91.21 ↓	3.71 ↓	92.61 ↓	4.26 ↓	92.06 ↓	2.95 ↓	**95.56 ↓**	**1.81 ↓**
ImageNet	*Benign*	0	77.06	0	73.52	0	71.85	0	74.21	0	71.63	0	75.20	0	75.51
	Badnets	99.24	74.53	5.91	69.37	6.31	66.28	**2.87**	69.46	4.18	69.44	7.58	70.49	3.61	70.96
	Troj-one	99.21	74.02	7.63	69.15	7.73	67.14	5.74	69.35	2.86	70.62	2.94	72.17	**3.16**	72.37
	Troj-all	97.58	74.45	9.18	69.86	7.54	68.20	6.02	69.64	3.27	69.85	4.81	71.45	2.68	72.13
	Blend	100	74.42	6.43	70.20	7.79	68.51	7.45	68.61	8.15	68.91	5.69	70.24	3.83	71.52
	SIG	94.66	74.69	**1.23**	69.82	4.28	66.08	5.37	70.02	3.47	69.74	4.36	70.73	2.94	72.36
	CLB	95.08	74.14	6.71	69.19	4.37	66.41	7.64	69.70	3.50	69.32	9.44	71.52	**3.05**	72.25
	Dyn-one	98.24	74.80	6.68	69.65	8.32	69.61	8.62	70.17	4.42	70.05	12.56	70.39	2.62	71.91
	Dyn-all	98.56	75.08	13.49	70.18	9.82	68.92	12.68	70.24	4.81	69.90	14.18	69.47	3.77	71.62
	LIRA	96.04	74.61	12.86	69.22	12.08	69.80	13.27	69.35	3.16	12.31	70.50	71.38	2.62	70.73
	WaNet	97.60	74.48	6.34	68.34	5.67	67.23	6.31	70.02	**4.42**	66.82	7.78	71.62	4.71	71.63
	ISSBA	98.23	74.38	7.61	68.42	4.50	67.92	8.21	69.51	3.35	68.02	9.74	70.81	**2.06**	70.67
	Avg. Drop	-	-	90.08↓	5.17↓	88.90↓	7.41↓	90.01↓	4.72↓	92.24↓	5.61↓	89.37↓	3.66↓	**94.03 ↓**	**2.84 ↓**

5 Experimental Results

5.1 Evaluation Settings

Datasets. We evaluate the proposed method through a range of experiments on two widely used datasets for backdoor attack study: **CIFAR10** [20] with 10 classes, **GTSRB** [45] with 43 classes. For the scalability test of our method, we also consider **Tiny-ImageNet** [22] with 100,000 images distributed among 200 classes and **ImageNet** [10] with 1.28M images distributed among 1000 classes. For multi-label clean-image backdoor attacks, we use object detection datasets **Pascal VOC** [13] and **MS-COCO** [31]. **UCF-101** [44] and **HMDB51** [21] have been used for evaluating in action recognition task. The **ModelNet** [54] dataset was also utilized to assess the performance of a 3D point cloud classifier. In addition to vision, we also consider attacks on natural language generation and use **WMT2014 En-De** [3] machine translation and **OpenSubtitles2012** [46] dialogue generation datasets (Results are in the Supplementary).

Attacks Configurations. Here, we first briefly overview the attack configurations on single-label image recognition datasets. We consider 14 state-of-the-art backdoor attacks: 1) Badnets [16], 2) Blend attack [8], 3 & 4) TrojanNet

Table 2: Performance analysis for the **multi-label backdoor attack** [7]. We choose 3 object detection datasets [13,31] and ML-decoder [42] network architecture for this evaluation. Mean average precision (mAP) and ASR of the model, with and without defenses, have been shown.

Dataset	No defense		ANP		AWM		RNP		FT-SAM		NFT (Ours)	
	ASR	mAP	ASR	mAP	ASR	mAP	ASR	mAP	ASR	mAP	ASR	mAP
VOC07	86.4	92.5	21.7	86.9	26.6	87.3	19.2	87.6	19.3	86.8	**17.3**	**89.1**
VOC12	84.8	91.9	18.6	85.3	19.0	85.9	**13.8**	86.4	14.6	87.1	14.2	**88.4**
MS-COCO	85.6	88.0	19.7	84.1	22.6	83.4	17.1	84.3	19.2	83.8	**16.6**	**85.8**

Table 3: Performance analysis for **action recognition tasks** where we choose 2 video datasets for evaluation. We consider a clean-label attack [62], where we need to generate adversarial perturbations for each input frame.

Dataset	No defense		I-BAU		AWM		RNP		FT-SAM		NFT (Ours)	
	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC
UCF-101	81.3	75.6	20.4	70.6	20.8	70.1	17.0	70.3	15.9	71.6	**13.3**	**71.2**
HMDB-51	80.2	45.0	17.5	**41.1**	15.2	40.9	12.6	40.4	10.8	41.7	**9.4**	40.8

(Troj-one & Troj-all) [33], 5) Sinusoidal signal attack (SIG) [2], 6 & 7) Input-Aware Attack (Dyn-one and Dyn-all) [38], 8) Clean-label attack (CLB) [48], 9) Composite backdoor (CBA) [30], 10) Deep feature space attack (FBA) [9], 11) Warping-based backdoor attack (WaNet) [37], 12) Invisible triggers based backdoor attack (ISSBA) [29], 13) Imperceptible backdoor attack (LIRA) [11], and 14) Quantization and contrastive learning-based attack (BPPA) [51]. In order to facilitate a fair comparison, we adopt trigger patterns and settings similar to those used in the original papers. Specifically, for both Troj-one and Dyn-one attacks, we set all triggered images to have the same target label (*i.e.*, all2one), whereas, for Troj-all and Dyn-all attacks, we have uniformly distributed the target labels across all classes (*i.e.*, all2all). Details on the hyper-parameters and overall training settings can be found in the Supplementary. We measure the success of an attack using two metrics: *clean test accuracy (ACC)* defined as the percentage of clean samples that are classified to their original target label and *attack success rate (ASR)* defined as the percentage of poison test samples ($\hat{x}$) that are classified to the target label ($\hat{y}$).

Defenses Configurations. We compare our approach with 10 existing backdoor mitigation methods: 1) *FT-SAM* [64]; 2) Adversarial Neural Pruning (*ANP*) [52]; 3) Implicit Backdoor Adversarial Unlearning (*I-BAU*) [57]; 4) Adversarial Weight Masking (*AWM*) [5]; 5) Reconstructive Neuron Pruning (RNP) [27]; 6) Fine-Pruning (*FP*) [34]; 7) Mode Connectivity Repair (*MCR*) [61]; 8) Neural Attention Distillation (*NAD*) [26], 9) Causality-inspired Backdoor Defense (*CBD*) [60]), 10) Anti Backdoor Learning (*ABL*) [25]. In the main paper, we compare NFT with the first 5 defenses as they are more relevant, and the comparison with the rest of the methods is in *Supplementary*. To apply NFT, we

Table 4: Removal performance (%) of NFT against backdoor attacks on **3D point cloud classifiers**. The attack methods [24], namely Poison-Label Backdoor Attack (PointPBA) with interaction trigger (PointPBA-I), PointPBA with orientation trigger (PointPBA-O), and Clean-Label Backdoor Attack (PointCBA) were considered, as well as the "backdoor points" based attack (3DPC-BA) outlined in prior work [56].

Attack	No Defense		ANP		AWM		RNP		FT-SAM		NFT (Ours)	
	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC
PointBA-I	98.6	89.1	13.6	82.6	15.4	83.9	**8.1**	84.0	8.8	84.5	9.6	**85.7**
PointBA-O	94.7	89.8	14.8	82.0	13.1	82.4	9.4	83.8	8.2	85.0	**7.5**	**85.3**
PointCBA	66.0	88.7	21.2	83.3	21.5	83.8	**18.6**	84.6	20.3	84.7	19.4	**86.1**
3DPC-BA	93.8	91.2	16.8	84.7	15.6	85.9	13.9	85.7	13.1	86.3	**12.6**	**87.7**

take 1% clean validation data (set aside from the training set) and fine-tune the model for 100 epochs. An SGD optimizer has been employed with a learning rate of 0.05 and a momentum of 0.95. The rest of the experimental details for NFT and other defense methods are in the *Supplementary*.

5.2   Performance Comparison of NFT

In this section, we compare the performance of NFT with other defenses in various scenarios: single-label (*i.e.*, image classification), multi-label (*i.e.*, object detection), video action recognition, and 3D point cloud.

Single-Label Backdoor attack. In Table 1, we present the performance of different defenses for 2 widely used benchmarks. For CIFAR10, we consider *4 label poisoning attacks: Badnets, Blend, Trojan, and Dynamic*. For all of these attacks, NFT shows significant performance improvement over other baseline methods. While ANP and AWM defenses work well for mild attacks with low poison rates (*e.g.*, 5%), the performance deteriorates for attacks with high poison rates (*e.g.*, $\geq$ 10%). It is observable that NFT performs well across all attack scenarios, *e.g.*, obtaining a 99.69% drop in ASR for blend attack while also performing well for clean data (only 0.94% of ACC drop). For Trojan and Dynamic attacks, we consider two different versions of attacks based on label-mapping criteria (all2one and all2all). The drop in attack success rate shows the effectiveness of NFT against such attacks. Recently, *small and imperceptible perturbations* as triggers have been developed in attacks (*e.g.*, WaNet, LIRA, *etc.*) to fool the defense systems. While AWM generates perturbations as a proxy for these imperceptible triggers, NFT does a better job in this regard. For further validation of our proposed method, we use deep *feature-based attacks* CBA and FBA. Both of these attacks manipulate features to insert backdoor behavior. Overall, we achieve an average drop of 95.56% in ASR while sacrificing an ACC of 1.81%. For the scalability test, we consider a large and widely used dataset in vision tasks, ImageNet. In consistency with other datasets, NFT also obtains SOTA performance in this particular dataset. Due to page constraints, we move *the performance comparison for GTSRB and Tiny-ImageNet to the Supplementary.*
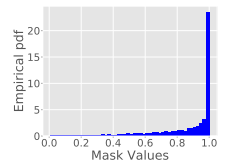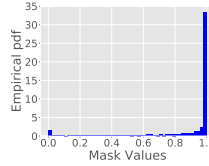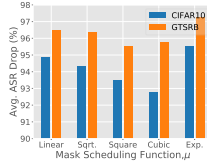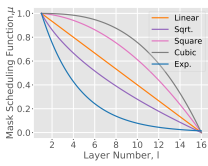
Fig. 2: Ablation with different **Mask Scheduling Function** (μ).

Fig. 3: Mask Distribution of AWM (left) and NFT (right).

Multi-label Backdoor Attack. In Table 2, we also evaluate our proposed method on multi-label clean-image backdoor attack [7]. In general, we put a trigger on the image and change the corresponding ground truth of that image. However, a certain combination of objects has been used as a trigger pattern. For example, if a combination of car, person, and truck is present in the image, it will fool the model into misclassifying it. Table 2 shows that our proposed method surpasses other defense strategies concerning both ASR and mAP metrics. Notably, ANP and AWM, which rely on adversarial search limited applicability in multi-label scenarios. This limitation arises from the less accurate process of approximating triggers for object detection. Conversely, FT-SAM's optimization driven by sharpness proves effective in removing backdoors, yet it achieves a lower mAP post-purification. This outcome is not ideal since the goal is to eliminate backdoors without significantly compromising clean accuracy.

Action Recognition Model. We further consider attacks on action recognition models; results are reported in Table 3. We use two widely used datasets, UCF-101 [44] and HMDB51 [21], with a CNN+LSTM network architecture. An ImageNet pre-trained ResNet50 network has been used for the CNN, and a sequential input-based Long Short Term Memory (LSTM) [43] network has been put on top of it. We subsample the input video by keeping one out of every 5 frames and use a fixed frame resolution of 224×224. We choose a trigger size of 20×20. Following [62], we create the required perturbation for clean-label attack by running projected gradient descent (PGD) [35] for 2000 steps with a perturbation norm of $\epsilon = 16$. Note that our proposed augmentation strategies for image classification are directly applicable to video recognition. During training, we keep 5% samples from each class to use them later as the clean validation set. Table 3 shows that NFT outperforms other defenses by a significant margin. In the case of a high number of classes and multiple image frames in the same input, it is a challenging task to optimize for the proper trigger pattern through the adversarial search described in I-BAU and AWM. Without a good approximation of the trigger, these methods seem to underperform in most cases.

3D Point Cloud. In this phase of our work, we assess NFT's resilience against attacks on 3D point cloud classifiers [24,56]. To evaluate, we utilize the ModelNet dataset [54] and the PointNet++ architecture [40]. The performance comparison of NFT and other defense methods is outlined in Table 4. NFT outperforms other defenses due to its unique formulations of the objective function.

Table 5: Average runtime of different defense methods against all 14 attacks. An NVIDIA RTX3090 GPU is used.

Method	ANP	I-BAU	AWM	FT-SAM	RNP	NFT (Ours)
Runtime (sec.)	118.1	92.5	112.5	98.7	102.6	**28.4**

Table 6: Sensitivity analysis of α and β for LIRA on CIFAR10.

α	0.9	0.9	0.9	0.8	0.8	0.8	0.7	0.7	0.7
β	0.25	0.50	0.75	0.25	0.50	0.75	0.25	0.50	0.75
ASR	3.65	3.42	4.87	2.73	**1.53**	3.76	4.18	4.92	5.23
ACC	88.52	88.34	89.12	89.97	**90.57**	89.34	87.64	87.78	88.10

5.3   Ablation Study

For all ablation studies, we consider the CIFAR10 dataset.

Runtime Analysis. For runtime analysis, we present the training time for different defenses in Table 5. ANP and AWM both employ computationally expensive adversarial search procedures to prune neurons, which makes them almost 4x slower than our method. However, NFT offers a computationally less expensive defense with SOTA performance in major benchmarks.

Choice of Scheduling Function, μ. For choosing the suitable function for μ, we conduct a detailed study with commonly used mathematical functions. Note that, we only consider a family of functions that decreases over the depth of the network (shown in Fig. 2). This allows more variations for deeper layer weights, making sense as they are more responsible for DNN decision-making. In our work, we choose an exponential formulation for μ as it offers superior performance. We also perform sensitivity analysis of scheduling parameters α and β in Table 6. In Fig. 3, we show the generated mask distributions of AWM and NFT. Compared to AWM, NFT produces more uniformly distributed masks that seem helpful for backdoor purification. We show the impact of μ in Table 7.

Nature of Optimization. In Table 7, we present the performance of SOTA techniques under *different validation sizes*. Even with 10 samples (single-shot), NFT performs reasonably well and offers better performance as compared to AWM. This again shows that the trigger generation process is less accurate and effective for a very small validation set. We also show the *effect of the proposed mask regularizer* that indirectly controls the change in weights for better ACC. Although AWM employs a similar ℓ_1 regularizer for masks, our proposed regularizer is more intuitive and specifically designed for better ACC preservation. While AWM encourages sparse solutions for M (shown in the left subfigure of Fig. 3), it helps with ASR but heavily compromises ACC. We also show the performance of NFT without augmentations.

Label Correction Rate. In the standard backdoor removal metric, it is sufficient for backdoored images to be classified as a non-target class (any class other than $\hat{y}$). However, we also consider another metric, label correction rate (LCR), for quantifying the success of a defense. *We define LCR as the percentage of poisoned samples correctly classified to their original classes.* Any method with the highest value of LCR is considered to be the best defense method. For this evaluation, we use CIFAR10 dataset and 6 backdoor attacks. Initially, the correction rate is 0% with no defense as the ASR is close to 100%. Table 8 shows that NFT obtains better performance in terms of LCR.

Table 7: Purification performance (%) for **various validation data sizes**. NFT performs reasonably well even with as few as 10 samples, *i.e.*, one sample (shot) per class for CIFAR10. We also show the impact of the **mask regularizer**, **mask scheduling function** μ, and **augmentations** on performance, which resonates with Fig. 1. Mask regularizer has the most impact on the clean test accuracy (around 7% worse without the regularizer). Without strong augmentations, we have a better ACC with a slightly worse ASR (around 6% drop).

Attack	Dynamic				WaNet				LIRA			
Samples	10		100		10		100		10		100	
Method	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC
No Defense	100	92.52	100	92.52	98.64	92.29	98.64	92.29	99.25	92.15	99.25	92.15
AWM	86.74	55.73	9.16	85.33	83.01	62.21	7.23	84.38	91.45	66.64	10.83	85.87
FT-SAM	8.35	73.49	5.72	84.70	9.35	75.98	5.56	86.63	11.83	72.40	4.85	88.82
NFT w/o Reg.	5.67	76.74	**1.36**	82.21	**4.18**	76.72	3.02	83.31	**4.83**	74.58	2.32	83.61
NFT w/o Aug.	11.91	**81.86**	10.59	89.53	10.36	83.10	7.81	**89.68**	12.23	81.05	9.16	88.74
NFT w/o $\mu(l)$	5.11	80.32	3.04	88.58	5.85	82.46	4.64	88.02	6.48	81.94	4.33	88.75
NFT	4.83	80.51	1.72	**90.08**	4.41	**83.58**	**2.96**	89.15	5.18	**82.72**	**2.04**	**89.34**

Table 8: Label Correction Rate (%) for different defense techniques, defined as the percentage of backdoor samples that are correctly classified to their original ground truth label.

Defense	Badnets	Trojan	Blend	SIG	CLB	WaNet
No Defense	0	0	0	0	0	0
ANP	84.74	80.52	81.38	53.35	82.72	80.23
I-BAU	78.41	77.12	77.56	39.46	78.07	80.65
AWM	79.37	78.24	79.81	44.51	79.86	79.18
FT-SAM	85.56	80.69	84.49	**57.64**	82.04	83.62
NFT (Ours)	**86.82**	**81.15**	**85.61**	55.18	**86.23**	**85.70**

6   Conclusion

We proposed a backdoor purification framework, NFT, utilizing an augmentation-based neural mask fine-tuning approach. NFT can change the backdoor model weights in a computationally efficient manner while ensuring SOTA purification performance. Our proposed method showed that the addition of MixUp during fine-tuning replaces the need for a computationally expensive trigger synthesis process. Furthermore, we proposed a novel mask regularizer that helps us preserve the cluster separability of the original backdoor model. By preserving this separability, the proposed method offers better clean test accuracy compared to SOTA methods. Furthermore, we suggested using a mask scheduling function that reduces the mask search space and improves the computational efficiency further. Our extensive experiments on 5 different tasks validate the efficiency and efficacy of the proposed backdoor purification method. We also conducted a detailed ablation study to explain the reasoning behind our design choices.

Acknowledgements

References

1. Ahmed, S., Al Arafat, A., Rizve, M.N., Hossain, R., Guo, Z., Rakin, A.S.: Ssda: Secure source-free domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19180–19190 (2023)
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. Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L., Tamchyna, A.: Findings of the 2014 workshop on statistical machine translation. In: Proceedings of the Ninth Workshop on Statistical Machine Translation. pp. 12–58. Association for Computational Linguistics, Baltimore, Maryland, USA (Jun 2014). https://doi.org/10.3115/v1/W14-3302, https://aclanthology.org/W14-3302
4. Carratino, L., Cissé, M., Jenatton, R., Vert, J.P.: On mixup regularization. The Journal of Machine Learning Research 23(1), 14632–14662 (2022)
5. Chai, S., Chen, J.: One-shot neural backdoor erasing via adversarial weight masking. arXiv preprint arXiv:2207.04497 (2022)
6. Chen, H., Fu, C., Zhao, J., Koushanfar, F.: Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In: IJCAI. p. 8 (2019)
7. Chen, K., Lou, X., Xu, G., Li, J., Zhang, T.: Clean-image backdoor: Attacking multi-label models with poisoned labels only. In: The Eleventh International Conference on Learning Representations (2023)
8. 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)
9. Cheng, S., Liu, Y., Ma, S., Zhang, X.: Deep feature space trojan attack of neural networks by controlled detoxification. In: AAAI. pp. 1148–1156 (2021)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255. IEEE (2009)
11. Doan, K., Lao, Y., Zhao, W., Li, P.: Lira: Learnable, imperceptible and robust backdoor attacks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11966–11976 (2021)
12. Edraki, M., Karim, N., Rahnavard, N., Mian, A., Shah, M.: Odyssey: Creation, analysis and detection of trojan models. IEEE Transactions on Information Forensics and Security 16, 4521–4533 (2021)
13. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision 88, 303–338 (2010)
14. Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B.: Sharpness-aware minimization for efficiently improving generalization. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=6Tm1mposlrM
15. 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)

16. Gu, T., Liu, K., Dolan-Gavitt, B., Garg, S.: Badnets: Evaluating backdooring attacks on deep neural networks. IEEE Access **7**, 47230–47244 (2019)
17. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European conference on computer vision. pp. 630–645. Springer (2016)
18. Hong, S., Chandrasekaran, V., Kaya, Y., Dumitraş, T., Papernot, N.: On the effectiveness of mitigating data poisoning attacks with gradient shaping. arXiv preprint arXiv:2002.11497 (2020)
19. Karim, N., Arafat, A.A., Rakin, A.S., Guo, Z., Rahnavard, N.: Fisher information guided purification against backdoor attacks. In: Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS) (2024)
20. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
21. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: 2011 International conference on computer vision. pp. 2556–2563. IEEE (2011)
22. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7), 3 (2015)
23. Levine, A., Feizi, S.: Deep partition aggregation: Provable defense against general poisoning attacks. arXiv preprint arXiv:2006.14768 (2020)
24. Li, X., Chen, Z., Zhao, Y., Tong, Z., Zhao, Y., Lim, A., Zhou, J.T.: Pointba: Towards backdoor attacks in 3d point cloud. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16492–16501 (2021)
25. Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X.: Anti-backdoor learning: Training clean models on poisoned data. Advances in Neural Information Processing Systems **34** (2021)
26. Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X.: Neural attention distillation: Erasing backdoor triggers from deep neural networks. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=9l0K4OM-oXE
27. Li, Y., Lyu, X., Ma, X., Koren, N., Lyu, L., Li, B., Jiang, Y.G.: Reconstructive neuron pruning for backdoor defense. arXiv preprint arXiv:2305.14876 (2023)
28. Li, Y., Wu, B., Jiang, Y., Li, Z., Xia, S.T.: Backdoor learning: A survey. arXiv preprint arXiv:2007.08745 (2020)
29. Li, Y., Li, Y., Wu, B., Li, L., He, R., Lyu, S.: Invisible backdoor attack with sample-specific triggers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16463–16472 (2021)
30. Lin, J., Xu, L., Liu, Y., Zhang, X.: Composite backdoor attack for deep neural network by mixing existing benign features. In: CCS. pp. 113–131 (2020)
31. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
32. Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: Defending against backdooring attacks on deep neural networks. In: International Symposium on Research in Attacks, Intrusions, and Defenses. pp. 273–294. Springer (2018)
33. Liu, Y., Ma, S., Aafer, Y., Lee, W.C., Zhai, J., Wang, W., Zhang, X.: Trojaning attack on neural networks (2017)
34. Liu, Y., Xie, Y., Srivastava, A.: Neural trojans. In: 2017 IEEE International Conference on Computer Design (ICCD). pp. 45–48. IEEE (2017)
35. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)

36. Manoj, N., Blum, A.: Excess capacity and backdoor poisoning. Advances in Neural Information Processing Systems **34**, 20373–20384 (2021)
37. Nguyen, A., Tran, A.: Wanet–imperceptible warping-based backdoor attack. arXiv preprint arXiv:2102.10369 (2021)
38. Nguyen, T.A., Tran, A.: Input-aware dynamic backdoor attack. Advances in Neural Information Processing Systems **33**, 3454–3464 (2020)
39. Ning, R., Li, J., Xin, C., Wu, H.: Invisible poison: A blackbox clean label backdoor attack to deep neural networks. In: IEEE INFOCOM 2021-IEEE Conference on Computer Communications. pp. 1–10. IEEE (2021)
40. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
41. Qiu, H., Zeng, Y., Guo, S., Zhang, T., Qiu, M., Thuraisingham, B.: Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation. In: Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security. pp. 363–377 (2021)
42. Ridnik, T., Sharir, G., Ben-Cohen, A., Ben-Baruch, E., Noy, A.: Ml-decoder: Scalable and versatile classification head. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 32–41 (2023)
43. Sherstinsky, A.: Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Physica D: Nonlinear Phenomena **404**, 132306 (2020)
44. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
45. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: The german traffic sign recognition benchmark: a multi-class classification competition. In: The 2011 international joint conference on neural networks. pp. 1453–1460. IEEE (2011)
46. Tiedemann, J.: Parallel data, tools and interfaces in opus. In: Lrec. vol. 2012, pp. 2214–2218. Citeseer (2012)
47. Tran, B., Li, J., Madry, A.: Spectral signatures in backdoor attacks. Advances in neural information processing systems **31** (2018)
48. Turner, A., Tsipras, D., Madry, A.: Clean-label backdoor attacks (2019), https://openreview.net/forum?id=HJg6e2CcK7
49. 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)
50. Wang, R., Zhang, G., Liu, S., Chen, P.Y., Xiong, J., Wang, M.: Practical detection of trojan neural networks: Data-limited and data-free cases. In: European Conference on Computer Vision. pp. 222–238. Springer (2020)
51. Wang, Z., Zhai, J., Ma, S.: Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15074–15084 (2022)
52. Wu, D., Wang, Y.: Adversarial neuron pruning purifies backdoored deep models. In: NeurIPS (2021)
53. Wu, D., Xia, S.T., Wang, Y.: Adversarial weight perturbation helps robust generalization. Advances in Neural Information Processing Systems **33**, 2958–2969 (2020)
54. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1912–1920 (2015)

55. Xian, X., Wang, G., Srinivasa, J., Kundu, A., Bi, X., Hong, M., Ding, J.: Understanding backdoor attacks through the adaptability hypothesis. In: Proc. International Conference on Machine Learning (2023)
56. Xiang, Z., Miller, D.J., Chen, S., Li, X., Kesidis, G.: A backdoor attack against 3d point cloud classifiers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7597–7607 (2021)
57. Zeng, Y., Chen, S., Park, W., Mao, Z.M., Jin, M., Jia, R.: Adversarial unlearning of backdoors via implicit hypergradient. arXiv preprint arXiv:2110.03735 (2021)
58. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
59. Zhang, L., Deng, Z., Kawaguchi, K., Ghorbani, A., Zou, J.: How does mixup help with robustness and generalization? arXiv preprint arXiv:2010.04819 (2020)
60. Zhang, Z., Liu, Q., Wang, Z., Lu, Z., Hu, Q.: Backdoor defense via deconfounded representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12228–12238 (2023)
61. Zhao, P., Chen, P.Y., Das, P., Ramamurthy, K.N., Lin, X.: Bridging mode connectivity in loss landscapes and adversarial robustness. arXiv preprint arXiv:2005.00060 (2020)
62. Zhao, S., Ma, X., Zheng, X., Bailey, J., Chen, J., Jiang, Y.G.: Clean-label backdoor attacks on video recognition models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14443–14452 (2020)
63. Zheng, R., Tang, R., Li, J., Liu, L.: Data-free backdoor removal based on channel lipschitzness. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V. pp. 175–191. Springer (2022)
64. Zhu, M., Wei, S., Shen, L., Fan, Y., Wu, B.: Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. arXiv preprint arXiv:2304.11823 (2023)