Backdoor Defense through Self-Supervised and Generative Learning

Ivan Sabolić
ivan.sabolic@fer.hr
Ivan Grubišić
ivan.grubisic@fer.hr
Siniša Šegvić
sinisa.segvic@fer.hr

Faculty of Electrical Engineering
and Computing
University of Zagreb

Abstract

Backdoor attacks change a small portion of training data by introducing hand-crafted triggers and rewiring the corresponding labels towards a desired target class. Training on such data injects a backdoor which causes malicious inference in selected test samples. Most defenses mitigate such attacks through various modifications of the discriminative learning procedure. In contrast, this paper explores an approach based on generative modelling of per-class distributions in a self-supervised representation space. Interestingly, these representations get either preserved or heavily disturbed under recent backdoor attacks. In both cases, we find that per-class generative models allow to detect poisoned data and cleanse the dataset. Experiments show that training on cleansed dataset greatly reduces the attack success rate and retains the accuracy on benign inputs.

1 Introduction

Deep models are establishing themselves as the default approach for resolving diverse problems across various domains [15, 50, 60]. However, their large capacity leaves them vulnerable to various cybernetic attacks [2]. Backdoor attacks inject vulnerabilities into production models by introducing subtle changes to the training data [13]. The installed backdoor induces malicious behaviour according to the attacker's goals [26]. We focus on backdoors that manipulate the model predictions in presence of a pre-defined trigger [7]. These attacks typically seek stealthiness through unobtrusive trigger designs, low prevalence of poisoned data, and high generalization performance on benign samples [13].

Backdoor attacks must trade-off stealthiness with applicability. Localized attacks can be easily applied in the physical world, however they can be uncovered by careful visual inspection [7, 13] and defenses that specialize for such attacks [48, 55]. Pervasive attacks may be imperceptible to the human eye but they can not be applied with a sticker [32, 40]. Many early defenses are designed to counter specific types of attacks [45, 55]. More recent empirical defenses focus on detecting a wide range of attacks by targeting common weaknesses [6, 10, 19].

This work proposes a novel approach to prevent the backdoor deployment given potentially poisoned data from an untrusted source [55]. We conjecture that a backdoor defense has a better chance of success if it relies more on class-agnostic features rather than discriminative features derived from potentially poisoned labels. However, very few previous works consider self-supervised [19, 56] or generative approaches [45], while their synergy is completely unexplored. We attempt to fill that gap by formulating our defense in terms of per-class densities of self-supervised representations. Our objective is to identify poisoned samples, restore the correct labels and to produce a clean model.

Poisoned samples are generated by either i) injecting triggers into images of non-target classes [7, 13, 20, 32, 44] or ii) applying strong perturbations to images of the target class [1, 54]. In both cases, our analysis shows that the self-supervised embeddings of poisoned samples get placed outside of the target class manifold. Hence, one could hypothesize that a generative model of the target class should assign these samples a lower likelihood. However, generative modelling of RGB images may assign high densities to outlier images [39, 49]. To avoid that, we decide to model per-class densities in the latent space of a self-supervised feature extractor. We identify potentially poisoned data according to the following two tests. The first test identifies samples residing within the distribution of a class that differs from their label. The second test identifies samples that are outside of the distribution of the entire dataset. These two tests allow us to cleanse the dataset by removing suspicious samples. Subsequently, we restore the original labels of those samples through generative classification [35]. Training on cleansed data ensures high performance on benign inputs, while fine-tuning on relabeled data ensures the complete removal of the backdoor.

To summarize, our contributions encompass three key aspects. First, we identify effects of various backdoor attacks on self-supervised image representations. Second, we propose a novel backdoor defense that builds upon per-class distributions of self-supervised representations. Third, we improve our defense through fine-tuning on pseudo-labels obtained by our generative classifier. Our experiments demonstrate the effectiveness of our approach in comparison to several state-of-the-art defenses. Importantly, our method successfully defends against a variety of attack types, including the latest attacks designed to undermine defenses based on latent separability [44]. Our code is publicly available at https://github.com/ivansabolic/GSSD.

2 Related Work

The main goal of existing backdoor attacks is to increase the attack success rate [31], while retaining stealthy triggers, low poisoning rates and clean accuracy [13]. A variety of triggers has been introduced, including black-white checkerboards [13], blending backgrounds [7], invisible noise [27], adversarial patterns [65] and sample-specific patterns [32, 41]. Existing attacks can further be divided into poisoned-label and clean-label types. Poisoned-label approaches [7, 13, 20, 32, 40, 41, 58] connect the trigger with the target class by relabeling poisoned samples as target labels. Clean-label approaches modify samples from the target class while leaving the labels unchanged [1, 54]. However, they are less effective than poisoned-label attacks [31]. Many existing defenses can be classified into three categories: i) detection-based defenses [4, 14, 18, 22, 53, 61], ii) training-time defenses [10, 19, 21, 28, 29, 34, 63] and iii) post-processing defenses [9, 30, 33, 37, 38, 43, 52, 55, 57, 59, 62, 64, 66]. The goal of detection-based defenses is to discover poisoned samples in order to deny their impact. Training-time defenses aim to develop a clean model from a potentially poisoned dataset. Post-processing defenses intend to remove the backdoor from an already trained model.

A significant drawback of detection-based defenses is the unused potential of the detected suspicious samples. On the other hand, training-time defenses remain vulnerable to the retained poisoned samples. We address these limitations by detecting poisoned data and correcting their labels through robust inference. After this intervention, the triggers act as data augmentation rather than an instrument for backdoor deployment. This effectively prevents the model from learning the association between the trigger and the target class.

3 Motivation

Backdoor attacks pose a great challenge since the attackers hold the first-mover advantage [13]. Nevertheless, we know that triggers must not disturb image semantics in order to promote stealthiness [31]. We propose to take advantage of this constraint by grounding our defense on image content, while avoiding poisoned labels.

Embedding inputs into the latent space. We avoid standard supervised learning due to its tendency to learn shortcut associations [11] between triggers and target labels. We start by self-supervised learning of a class-agnostic representation of the training data [5]. Self-supervised representations exhibit remarkable semantic power, often surpassing supervised representations in linear evaluation [16, 42]. Furthermore, they are resilient to poisoned-label attacks due to strong augmentations and contrastive learning objectives that disregard the labels [19, 56].

Attack impact in the self-supervised feature space. Our analysis builds upon UMAP [36] dimensionality reduction that is optimized to retain the adjacency structure of the original high-dimensional space. The resulting two-dimensional plots show the clean data-points in the colours of their respective classes and the poisoned data in black, as shown in Figure 1.

Figure 1 (left) illustrates the common behaviour of a large groups of attacks from the literature [7, 13, 40, 44]. The plot suggests that these attacks exert a very small influence onto the embedding of the clean images. Appendix F supports this hypothesis with quantitative measurements. We refer to this scenario as non-disruptive poisoning: poisoned embeddings resemble their original class more than the target class. This behaviour is not unexpected since the triggers are designed for minimal visual impact in order to conceal the attack.

Figure 1 (right) shows that some attacks move the poisoned embeddings off the natural manifold of the clean embeddings in the self-supervised feature space. This separation can occur due to adversarial triggers [12] or other strong perturbations of the original images [51, 54]. Consequently, we refer to this scenario as disruptive poisoning. One such example is the clean-label attack, which aims to induce an association between the trigger and the unchanged original label by adversarially perturbing the image before adding the trigger [54].

Modeling per-class distributions. In both scenarios from the previous subsection, the poisoned embeddings are situated far away from the clean samples of the target class. We propose to expose this occurrence by comparing per-class densities of the latent embeddings. These densities can be recovered by learning per-class generative models such as normalizing flows [47] or variational encoders [23]. Moreover, in the non-disruptive scenario, one can try to recover the original label of poisoned samples through generative classification.

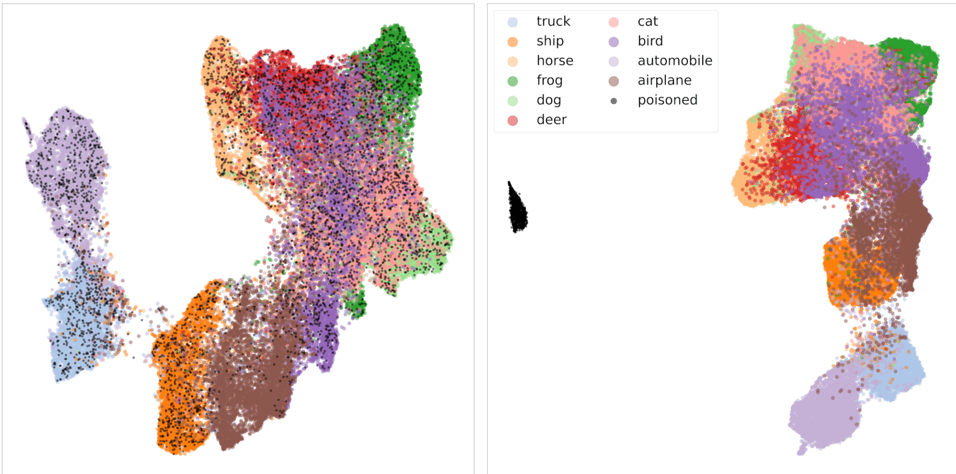


Figure 1: 2D UMAP visualization of the self-supervised feature space for CIFAR-10. Poisoned samples are shown in black, while clean samples are shown in colour. The target class is in brown (*airplane*). Non-disruptive attacks (left, [13]) exert a very small influence to the self-supervised embeddings. Disruptive attacks (right, [54]) displace the poisoned samples from the manifold of the training data.

4 Defense through generative and self-supervised learning

Our defense identifies target classes and suspicious samples by leveraging per-class densities of self-supervised features. We re-train robust supervised models without suspicious samples and subsequently fine-tune with corrected labels. For brevity, we refer to our defense as GSSD (generative self-supervised defense). We explain the details within this section.

4.1 Problem formulation

Threat model. We assume the standard *all-to-one* threat model [10, 19]. The attack poisons a subset of the original benign training dataset $\mathcal{D} = \{(\boldsymbol{x}_i^*, y_i^*)\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ contains all possible inputs and $\mathcal{Y}$ is the set of classes. The attack involves a single target label $y_\mathrm{T} \in \mathcal{Y}$ and modifies samples with indices $\mathcal{I}_\mathrm{P} \subset \{1..N\}$ with some transformation $(\boldsymbol{x}_i^*, y_i^*) \mapsto (\tilde{\boldsymbol{x}}_i, y_\mathrm{T})$ to produce the poisoned subset $\mathcal{D}_\mathrm{P} = \{(\tilde{\boldsymbol{x}}_i, y_\mathrm{T})\}_{i \in \mathcal{I}_\mathrm{P}}$. The complete poisoned dataset is $\tilde{\mathcal{D}} = \mathcal{D}_\mathrm{C} \cup \mathcal{D}_\mathrm{P}$, where the clean subset $\mathcal{D}_\mathrm{C}$ contains the remaining (non-poisoned) samples. The attack aims for a model trained on $\tilde{\mathcal{D}}$ to classify triggered test inputs as the target class y_T without affecting the performance on clean inputs.

Defender's goals. We assume that the defender controls the training process. Given a possibly poisoned training set $\tilde{\mathcal{D}}$, the defender's objective is to obtain a trained model instance without a backdoor while preserving high accuracy on benign samples. Our defense aims to achieve this goal by producing a filtered clean subset $\hat{\mathcal{D}}_\mathrm{C}$ and a relabeled poisoned subset $\hat{\mathcal{D}}'_\mathrm{P}$ that can be used to train the classifier safely.

4.2 Defense overview

The input to our method is a potentially poisoned dataset: $\tilde{\mathcal{D}} \subset \mathcal{X} \times \mathcal{Y}$. As outlined in Algorithm 1, our method starts by training the feature extractor $f_{\boldsymbol{\theta}_\mathrm{F}}$ by self-supervision on $\{\boldsymbol{x} : (\boldsymbol{x}, y) \in \tilde{\mathcal{D}}\}$. Then, it learns per-class densities $p_{\boldsymbol{\theta}_y}$ on the features produced by $f_{\boldsymbol{\theta}_\mathrm{F}}$ for each class $y \in \mathcal{Y}$. By analyzing the recovered per-class densities, as further elaborated in Section 4.4, we search for disruptively and non-disruptively poisoned target classes.

Once we identify the target classes, we separate the dataset into three parts. The clean part $\hat{\mathcal{D}}_\mathrm{C}$ contains all samples from the clean classes and all samples from the target classes that receive high density of the labeled class and low density of other classes. The poisoned part $\hat{\mathcal{D}}_\mathrm{P}$ contains samples with low density of the class identified as target and high density of non-target classes. The last part $\hat{\mathcal{D}}_\mathrm{U}$ contains samples with uncertain poisoning status and class membership. Finally, a discriminative classifier $h_{\boldsymbol{\theta}_\mathrm{C}}$ is trained on $\hat{\mathcal{D}}_\mathrm{C}$, and fine-tuned on relabeled samples from $\hat{\mathcal{D}}'_\mathrm{P}$.

Algorithm 1 Defense overview

1: Create a feature extractor $f_{\boldsymbol{\theta}_\mathrm{F}}$ by self-supervised training on $\tilde{\mathcal{D}}$.
2: Learn per-class densities $p_{\boldsymbol{\theta}_y}(\boldsymbol{z})$ of self-supervised features $\boldsymbol{z} = f_{\boldsymbol{\theta}_\mathrm{F}}(\boldsymbol{x})$.
3: Identify target classes according to class-level poisoning scores for non-disruptive and disruptive poisoning, S^y_ND and S^y_D.
4: Assign poisoning scores $\sigma_y(\boldsymbol{z})$ to all samples from target classes.
5: Based on poisoning scores, split $\tilde{\mathcal{D}}$ into poisoned samples $\hat{\mathcal{D}}_\mathrm{P}$, uncertain samples $\hat{\mathcal{D}}_\mathrm{U}$, and clean samples $\hat{\mathcal{D}}_\mathrm{C}$.
6: Train a discriminative model $h_{\boldsymbol{\theta}_\mathrm{C}}$ on $\hat{\mathcal{D}}_\mathrm{C}$.
7: Produce the relabeled subset $\hat{\mathcal{D}}'_\mathrm{P}$ by relabeling $\hat{\mathcal{D}}_\mathrm{P}$ according to per-class densities.
8: Fine-tune the discriminative model $h_{\boldsymbol{\theta}_\mathrm{C}}$ on $\hat{\mathcal{D}}'_\mathrm{P}$.

4.3 Per-class densities of self-supervised features

Our defense starts by applying SimCLR [5] to train the self-supervised feature extractor $f_{\boldsymbol{\theta}_\mathrm{F}}$ on $\tilde{\mathcal{D}}$. Then, we estimate per-class densities of features $\boldsymbol{z} = f_{\boldsymbol{\theta}_\mathrm{F}}(\boldsymbol{x})$ with lightweight normalizing flows [24]. We define a set of normalizing flows with separate parameters $\boldsymbol{\theta}_y$ for each class. We train per-class densities $p_{\boldsymbol{\theta}_y}(\boldsymbol{z})$ on the corresponding embedding subsets $\mathcal{D}^y_\mathrm{F} = \left\{f_{\boldsymbol{\theta}_\mathrm{F}}(\boldsymbol{x}) : (\boldsymbol{x}, y') \in \tilde{\mathcal{D}}, y' = y\right\}$ by maximizing the average log-likelihood:

$$\overline{\mathcal{L}}(\boldsymbol{\theta}_y, \mathcal{D}^y_\mathrm{F}) = \underset{\boldsymbol{z} \in \mathcal{D}^y_\mathrm{F}}{\mathrm{E}} \log p_{\boldsymbol{\theta}_y}(\boldsymbol{z}). \tag{1}$$

After estimating the densities, our next objective is to identify the classes with poisoned samples and determine whether the poisoning is disruptive or non-disruptive.

4.4 Identifying target classes

We first check for the presence of non-disruptive poisoning by assuming that the poisoned samples resemble their source classes in the self-supervised feature space. In this case, the generative model of the target class will assign moderate densities to many foreign samples due to learning on triggered samples. This behaviour will be much less pronounced in non-target classes. Consequently, we propose to identify target classes according to the average log-density over all foreign samples:

$$S_{\mathrm{ND}}^{y} = \overline{\mathcal{L}}\left(\boldsymbol{\theta}_y, \bigcup_{y' \in \mathcal{Y} \setminus \{y\}} \mathcal{D}_{\mathrm{F}}^{y'}\right). \tag{2}$$

We consider class y as non-disruptively poisoned if S_{ND}^{y} exceeds the threshold β_{ND}.

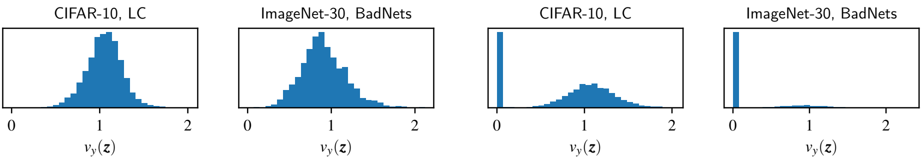
Next we check for disruptive poisoning. The defining characteristic of this type of poisoning is that the poisoned samples are less similar to all clean samples than the clean samples of different classes among themselves. As a consequence, we expect the foreign densities in such samples to be much lower than the foreign densities in the clean samples. We therefore search for classes with a significant number of such outliers. We formalize this idea by first defining the maximum foreign density score for each sample:

$$v_y(\boldsymbol{z}) = \max_{y' \in \mathcal{Y} \setminus \{y\}} p_{\boldsymbol{\theta}_{y'}}(\boldsymbol{z}) \tag{3}$$

We classify a class as disruptively poisoned if the fraction of samples with low v scores (3) exceeds the threshold β_{D}. More precisely, for each class y, we i) compute the set of v scores of the corresponding samples $\mathcal{V}_y = \left\{v_y(\boldsymbol{z}) : \boldsymbol{z} \in \mathcal{D}_{\mathrm{F}}^{y}\right\}$, ii) compute a histogram with 30 bins of equal widths for $\mathcal{V}_y$ as shown in Figure 2, iii) find the minimum μ_y of the histogram on the left from the hyperparameter λ, and iv) compute the fraction of $\mathcal{D}_{\mathrm{F}}^{y}$ with $v_y(\boldsymbol{z}_i) < \mu_y$:

$$S_{\mathrm{D}}^{y} = \frac{\left|\left\{\boldsymbol{z} \in \mathcal{D}_{\mathrm{F}}^{y} : v_y(\boldsymbol{z}) < \mu_y\right\}\right|}{\left|\mathcal{D}_{\mathrm{F}}^{y}\right|} \tag{4}$$

Finally, we classify a class y as disruptively poisoned if S_{D}^{y} is less than the threshold β_{D}. We can interpret β_{D} as the minimum fraction of poisoned samples per class.



(a) Clean classes (b) Target classes

Figure 2: Distributions of the maximum foreign density $v_y(\boldsymbol{z})$ of clean and target classes in presence of the Label-Consistent attack [54] on CIFAR-10, and a strong BadNets attack on ImageNet-30. In contrast to clean classes, target classes exhibit strong bimodality because poisoned samples tend to cluster near 0. Note that both attacks are disruptive.

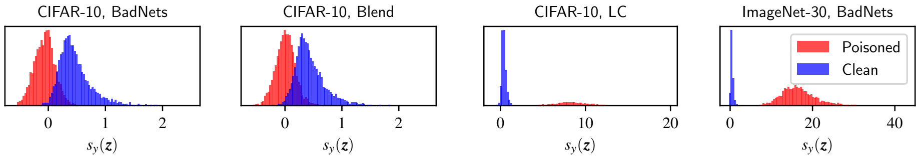
4.5 Identifying poisoned samples

After identifying the target classes, we use the following score to identify poisoned samples:

$$s_y(\boldsymbol{z}) = \frac{p_{\boldsymbol{\theta}_y}(\boldsymbol{z})}{\max_{y' \in \mathcal{Y} \setminus \{y\}} p_{\boldsymbol{\theta}_{y'}}(\boldsymbol{z})}. \tag{5}$$

The numerator is the density of the sample with respect to the labeled class. In the case of non-disruptive poisoning, we expect the denominator to be high for poisoned samples because they resemble their original class, and low for clean samples from the target class. In the case of disruptive poisoning, we expect the densities of disruptively poisoned samples to be very low under all classes but the poisoned one. Hence, disruptively poisoned samples will score lower than clean ones, and it will be the opposite in case of non-disruptive poisoning, as shown in Figure 3. Therefore, we define the final poisoning score σ so that it is higher for poisoned samples:

$$\sigma_y(z) = s_y(z)^{1-2[\![\text{class } y \text{ is disruptively poisoned}]\!]}. \tag{6}$$



(a) Non-disruptive poisoning attacks (b) Disruptive poisoning attacks

Figure 3: The values of the poisoning score (5) for all samples within one target class. Clean samples are shown in blue and poisoned samples in red.

4.6 Filtering and relabeling suspicious samples

We split the samples from identified target classes into three parts according to the hyper-parameters $\alpha_C, \alpha_P \in (0..0.5]$. We partition the target samples according to the poisoning scores so that the first α_P of them are placed in $\hat{\mathcal{D}}_P$, the last α_C of them in $\hat{\mathcal{D}}_C$, while ignoring the intermediate. Samples from clean classes are also placed in $\hat{\mathcal{D}}_C$.

Next, we train a classifier on $\hat{\mathcal{D}}_C$ using the standard training procedure. However, it is possible that $\hat{\mathcal{D}}_C$ still contains a small portion of poisoned samples. To counteract their influence, we proceed to fine-tune the classifier on the relabeled dataset $\hat{\mathcal{D}}'_P$:

$$\hat{\mathcal{D}}'_P = \left\{ \left(\boldsymbol{x}, \arg\max_{y' \in \mathcal{Y} \setminus \{y\}} p_{\boldsymbol{\theta}_{y'}}(f_{\boldsymbol{\theta}_F}(\boldsymbol{x})) \right) : (\boldsymbol{x}, y) \in \hat{\mathcal{D}}_P \right\} \tag{7}$$

The accuracy of the relabeling process is evaluated in Appendix L.

5 Experiments

5.1 Experimental setups

Datasets and models. We evaluate our defenses on three datasets: CIFAR-10 [25], a 30-class subset of ImageNet [8] (ImageNet-30), and a 30-class subset of VGGFace2 [3] (VggFace2-30). We use ResNet-18 [15] on CIFAR-10 and ImageNet-30, and DenseNet-121 [17] on the VGGFace2-30. More detailed training setups are given in Appendix B.

Attack configurations. We consider the following 6 baselines: BadNets [13], blending attack (Blend) [7], warping attack (WaNet) [40], sample-specific triggers (ISSBA) [32], clean-label attack (LC) [54] and the recent state of the art based on latent separability (Adap-Patch and Adap-Blend) [44]. These baselines cover visible patch-based attacks (BadNets), invisible attacks (WaNet and Blend), sample-specific attacks (ISSBA) and clean-label attacks (LC). We set the target label as $y_T = 0$. The poisoning rate is set to 10%, except for Adap-Patch and Adap-Blend attacks, where 1% of the data is poisoned, and the clean-label attack, where 2.5% of the data is poisoned. We omit some attacks in ImageNet-30 and VGGFace2-30 experiments since we were unable to reproduce the performance from their papers. Appendix C provides detailed per-attack configurations.

Defense baselines and configurations. We compare our method with four state-of-the-art defenses: Neural attention distillation (NAD) [29], Anti backdoor learning (ABL) [28], Decoupling based defense (DBD) [19] and Backdoor defense via adaptive splitting (ASD) [10]. We note that NAD and ASD require a small subset of clean data for each class. Detailed configurations of our defense and other defenses are provided in Appendix E and Appendix D. Additionally, we validate hyperparameter robustness in Appendix K.

Evaluation metrics. We evaluate the two standard metrics of the defense performance, including the accuracy on the clean test dataset (ACC), and the attack success rate (ASR) that denotes the accuracy of recognizing poisoned samples as the target label.

5.2 Performance evaluation

Table 1 evaluates effectiveness of our GSSD defense under state-of-the-art attacks and compares it with the state-of-the-art. GSSD consistently achieves lower ASR than the alternatives, with ASR falling below 1% in the majority of assays. At the same time, it mantains consistently high ACC accross all datasets. ASD exhibits slightly higher ACC on CIFAR-10 and VGGFace2-30. However, it often comes at the expense of significantly higher ASR, even though ASD requires a small number of clean samples.

We highlight strong defense performance of GSSD under Adap-Patch and Adap-Blend attacks. Despite the aim of these attacks to suppress the latent separation between poisoned and clean samples, self-supervised representations of poisoned samples are still in high-density regions of their original class. Conversely, when assessing ABL, DBD, and ASD defenses against Adap-Blend, ASR increases compared to no defense.

Robustness to different poisoning rates Table 2 validates the resistance against three representative poisoned label attacks with different poisoning rates. GSSD manages to reduce ASR to below or around 1% in all cases, while retaining high accuracy on benign inputs.

Robustness to adaptive attacks. We analyze the effect of a potential adaptive attack in Appendix I, finding that GSSD successfully defends against it.

Time complexity. Appendix G presents time complexity measurements. Although it ranks second to ABL, GSSD outperforms DBD and ASD significantly in terms of time efficiency.

Limitations. Even though our defense has proven successful against all considered state-of-the-art attacks through thorough empirical testing, it is important to note that it lacks theoretical guarantees. We have no precise characterization of what kinds of attacks it will be resilient against given the properties of the dataset. A weakness of detecting target classes as proposed here is that, if the target class detection step in Equation (2) fails, then the entire defense fails. We have observed such behavior in attack that can succeed with very low poisoning rate. Current state of the art [10, 28] also fails in such scenarios. Further discussion is provided in Appendix H. As introduced in section 4.1, we are working under the assumption that all poisoned samples share the same target label, a scenario commonly referred to as *all-to-one* poisoning. There are other types of poisoning in literature, such as *all-to-all* poisoning, which are beyond the scope or our work.

Dataset ↓	Defense →	No Defense		NAD*		ABL		DBD		ASD*		GSSD (ours)	
	Attack ↓	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
CIFAR-10	BadNets	94.9	100.0	88.2	4.60	**93.8**	**1.10**	92.4	0.96	92.1	3.00	91.7	0.14
	Blend	94.2	98.25	85.8	3.40	91.9	1.60	92.2	1.73	**93.4**	**1.00**	92.2	0.77
	WaNet	94.3	98.00	71.3	6.70	84.1	2.20	91.2	0.39	93.3	1.20	**93.7**	**1.35**
	LC	94.9	99.33	86.4	9.50	86.6	1.30	89.7	0.01	93.1	0.90	**92.8**	**0.06**
	ISSBA	94.5	100.0	90.7	0.64	89.2	1.20	83.2	0.53	92.4	2.13	**93.9**	**0.62**
	Adap-Patch	95.2	80.9	91.1	2.96	81.9	0.00	92.9	1.77	93.6	100.0	**92.4**	**0.23**
	Adap-Blend	95.0	64.9	88.3	2.11	91.5	81.93	90.1	99.97	94.0	93.90	**92.7**	**0.22**
	Average	-	-	85.6	4.27	88.4	12.76	90.2	15.05	93.1	28.87	**92.6**	**0.48**
ImageNet-30	BadNets	95.3	99.98	92.7	0.42	94.3	0.24	91.2	0.54	90.7	9.72	**94.8**	**0.00**
	Blend	93.7	99.93	90.0	0.51	93.1	0.14	90.3	0.58	89.9	2.07	**93.5**	**0.45**
	WaNet	93.5	100.0	90.7	0.56	92.0	1.33	90.5	0.48	88.8	2.89	**93.4**	**1.33**
	Average	-	-	91.1	0.50	93.1	1.71	90.7	0.53	89.8	4.89	**93.9**	**0.59**
VGGFace2-30	BadNets	93.2	100.0	56.1	6.50	93.9	2.67	90.3	0.00	96.7	98.12	**95.2**	**0.09**
	Blend	92.8	99.95	50.8	7.30	93.4	5.40	90.2	0.04	**96.0**	**0.18**	94.2	0.42
	WaNet	93.7	99.60	50.4	4.20	93.2	1.48	87.2	0.00	96.9	96.51	**94.9**	**0.14**
	Average	-	-	52.4	6.00	93.5	3.18	89.2	0.01	96.5	64.94	**94.8**	**0.22**

Table 1: Comparisons of the proposed GSSD defense with 4 baselines on CIFAR-10, ImageNet-30 and VGGFace2-30 datasets. We mark defenses requiring extra clean data with *. We report the clean test accuracy as ACC [%] and the attack success rate ASR [%]. The defense that exhibits the highest ACC − ASR is marked in **bold**.

Poisoning rate ↓	Attack →	BadNets		Blend		WaNet	
	Defense ↓	ACC	ASR	ACC	ASR	ACC	ASR
1%	No Defense	95.2	99.96	95.2	94.52	94.7	60.05
	GSSD	91.2	0.04	91.7	0.05	91.1	0.01
5%	No Defense	94.5	100.00	94.7	99.34	94.4	95.70
	GSSD	93.5	0.47	93.5	1.30	93.4	0.52
10%	No Defense	95.0	100.00	94.6	99.69	94.5	99.00
	GSSD	91.7	0.14	92.2	0.77	93.7	1.35
15%	No Defense	94.5	100.00	94.3	99.92	94.2	99.71
	GSSD	92.8	0.44	92.1	0.94	93.0	0.83
20%	No Defense	94.5	100.00	94.4	99.90	94.1	98.83
	GSSD	91.6	1.20	89.6	1.90	92.5	1.00

Table 2: Robustness of GSSD to different poisoning rates on CIFAR-10.

5.3 Ablation studies

The importance of self-supervision. Table 3 validates the choice of the pre-trained feature extractor. We compare our choice of SimCLR with supervised training and CLIP [46]. We can see that SimCLR and CLIP deliver similar overall performance, while greatly outperforming the supervised representations. This improvement occurs since self-supervised learning is less affected by triggers and not affected by target labels. Interestingly, our defense still works well and maintains high accuracy against the WaNet attack even with supervised representations. Additionally, we notice that CLIP performs quite well. This suggests that the computationally expensive pre-training on the poisoned dataset may not be necessary. Instead, using pre-trained feature extractors like CLIP can be effective.

Attack →	BadNets		Blend		WaNet	
Feature extractor ↓	ACC	ASR	ACC	ASR	ACC	ASR
RN-18 supervised	75.1	6.00	76.7	76.40	92.4	0.42
RN-50 CLIP	91.1	0.10	**93.0**	**0.88**	**94.0**	**0.81**
RN-18 self-sup (SimCLR)	**91.7**	**0.14**	92.2	0.77	93.7	1.35

Table 3: Comparison of different feature extractors on CIFAR-10.

Effect of generative classification. Appendix J highlights the strenghts of the generative classifier compared to the discriminative classifier trained on the same features. The discriminative classifier fails at detection of target classes, but performs well at filtering and relabeling given knowledge of the target class.

The impact of fine-tuning on relabeled data. Table 4 validates the impact of our method during standard discriminative training and subsequent fine-tuning. Top two rows show that fine-tuning with relabeled data breaks the backdoor even after standard training on poisoned data. The remaining rows indicate that we obtain the best ACC−ASR performance with training on the filtered clean subset $\hat{\mathcal{D}}_\mathrm{C}$ followed by fine-tuning on the relabeled subset $\hat{\mathcal{D}}'_\mathrm{P}$.

Training data	Fine-tuning	Badnets		Blend		Wanet		Adap-Patch	
		ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
Original ($\tilde{\mathcal{D}}$)	✗	94.5	100.0	94.2	98.32	94.5	99.12	95.2	80.95
Original ($\tilde{\mathcal{D}}$)	✓	93.6	1.48	92.0	1.30	94.0	3.47	94.3	8.01
Cleansed ($\hat{\mathcal{D}}_\mathrm{C}$)	✗	93.8	95.92	94.0	51.00	93.6	5.30	93.4	0.18
Cleansed ($\hat{\mathcal{D}}_\mathrm{C} \cup \hat{\mathcal{D}}'_\mathrm{P}$)	✗	93.8	3.69	93.5	10.47	93.5	2.58	92.2	0.10
Cleansed ($\hat{\mathcal{D}}_\mathrm{C}$)	✓	91.7	0.14	92.2	0.77	93.7	1.35	92.4	0.23

Table 4: Validation of the impact of fine-tuning with the relabeled subset $\hat{\mathcal{D}}'_\mathrm{P}$. The original dataset is the poisoned dataset $\tilde{\mathcal{D}}$, the input to our method. The filtered clean subset $\hat{\mathcal{D}}_\mathrm{C}$ and the relabeled poisoned subset $\hat{\mathcal{D}}'_\mathrm{P}$ are produced by our method.

6 Conclusion

We have presented a novel analysis of the effects of backdoor attacks onto self-supervised image representations. The results inspired us to propose a novel backdoor defense that allows to detect target classes and samples. Extensive evaluation against the state-of-the-art reveals competitive performance. In particular, we note extremely effective ASR reduction in presence of latent separability attacks Adap-Patch and Adap-Blend. We hope that our method can contribute as a tool for increasing the robustness of deep learning applications. Suitable directions for future work include circumventing the detection of target classes and extending the applicability to other kinds of poisoning.

Acknowledgments. This work has been co-funded by the European Defence Fund grant EICACS and Croatian Science Foundation grant IP-2020-02-5851 ADEPT.

References

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

[2] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognit.*, 84:317–331, 2018.

[3] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[4] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[6] Weixin Chen, Baoyuan Wu, and Haoqian Wang. Effective backdoor defense by exploiting sensitivity of poisoned samples. *Advances in Neural Information Processing Systems*, 35:9727–9737, 2022.

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

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu. Black-box detection of backdoor attacks with limited information and data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16482–16491, 2021.

[10] Kuofeng Gao, Yang Bai, Jindong Gu, Yong Yang, and Shu-Tao Xia. Backdoor defense via adaptively splitting poisoned dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4005–4014, 2023.

[11] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

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

[14] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. *arXiv preprint arXiv:2302.03251*, 2023.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[18] Hanxun Huang, Xingjun Ma, Sarah Erfani, and James Bailey. Distilling cognitive backdoor patterns within an image. *arXiv preprint arXiv:2301.10908*, 2023.

[19] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. *arXiv preprint arXiv:2202.03423*, 2022.

[20] Wenbo Jiang, Hongwei Li, Guowen Xu, and Tianwei Zhang. Color backdoor: A robust poisoning attack in color space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8133–8142, 2023.

[21] Charles Jin, Melinda Sun, and Martin Rinard. Incompatibility clustering as a defense against backdoor poisoning attacks. *arXiv preprint arXiv:2105.03692*, 2021.

[22] Alaa Khaddaj, Guillaume Leclerc, Aleksandar Makelov, Kristian Georgiev, Hadi Salman, Andrew Ilyas, and Aleksander Madry. Rethinking backdoor attacks. In *International Conference on Machine Learning*, pages 16216–16236. PMLR, 2023.

[23] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[24] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[25] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). 2009. URL http://www.cs.toronto.edu/~kriz/cifar.html.

[26] Pavel Laskov and Richard Lippmann. Machine learning in adversarial environments. *Mach. Learn.*, 81(2):115–119, 2010. doi: 10.1007/s10994-010-5207-6. URL https://doi.org/10.1007/s10994-010-5207-6.

[27] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 18(5):2088–2105, 2020.

[28] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34:14900–14912, 2021.

[29] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*, 2021.

[30] Yige Li, Xixiang Lyu, Xingjun Ma, Nodens Koren, Lingjuan Lyu, Bo Li, and Yu-Gang Jiang. Reconstructive neuron pruning for backdoor defense. In *International Conference on Machine Learning*, pages 19837–19854. PMLR, 2023.

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

[32] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16463–16472, 2021.

[33] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, pages 273–294. Springer, 2018.

[34] Min Liu, Alberto Sangiovanni-Vincentelli, and Xiangyu Yue. Beating backdoor attack at its own game. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4620–4629, 2023.

[35] Radek Mackowiak, Lynton Ardizzone, Ullrich Kothe, and Carsten Rother. Generative classifiers as a basis for trustworthy image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2971–2981, 2021.

[36] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[37] Rui Min, Zeyu Qin, Li Shen, and Minhao Cheng. Towards stable backdoor purification through feature shift tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[38] Bingxu Mu, Zhenxing Niu, Le Wang, Xue Wang, Qiguang Miao, Rong Jin, and Gang Hua. Progressive backdoor erasing via connecting backdoor and adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20495–20503, 2023.

[39] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

[40] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021.

[41] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020.

[42] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[43] Lu Pang, Tao Sun, Haibin Ling, and Chao Chen. Backdoor cleansing with unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12218–12227, 2023.

[44] Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, and Prateek Mittal. Revisiting the assumption of latent separability for backdoor defenses. In *The eleventh international conference on learning representations*, 2022.

[45] Ximing Qiao, Yukun Yang, and Hai Li. Defending neural backdoors via generative distribution modeling. *Advances in neural information processing systems*, 32, 2019.

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763, 2021.

[47] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1530–1538. JMLR.org, 2015.

[48] Hadi Salman, Saachi Jain, Eric Wong, and Aleksander Madry. Certified patch robustness via smoothed vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15116–15126. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01471. URL https://doi.org/10.1109/CVPR52688.2022.01471.

[49] Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. *arXiv preprint arXiv:1909.11480*, 2019.

[50] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[51] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[52] Guanhong Tao, Guangyu Shen, Yingqi Liu, Shengwei An, Qiuling Xu, Shiqing Ma, Pan Li, and Xiangyu Zhang. Better trigger inversion optimization in backdoor scanning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13368–13378, 2022.

[53] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31, 2018.

[54] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

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

[56] Hang Wang, Sahar Karami, Ousmane Dia, Hippolyt Ritter, Ehsan Emamjomeh-Zadeh, Jiahui Chen, Zhen Xiang, David J Miller, and George Kesidis. Training set cleansing of backdoor poisoning by self-supervised representation learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[57] Shaokui Wei, Mingda Zhang, Hongyuan Zha, and Baoyuan Wu. Shared adversarial unlearning: Backdoor mitigation by unlearning shared adversarial examples. *Advances in Neural Information Processing Systems*, 36, 2024.

[58] Yutong Wu, Xingshuo Han, Han Qiu, and Tianwei Zhang. Computation and data efficient backdoor attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4805–4814, 2023.

[59] Zhen Xiang, Zidi Xiong, and Bo Li. Cbd: A certified backdoor detector based on local dominant probability. *Advances in Neural Information Processing Systems*, 36, 2024.

[60] Wayne Xiong, Jasha Droppo, Xuedong Huang, Frank Seide, Mike Seltzer, Andreas Stolcke, Dong Yu, and Geoffrey Zweig. Achieving human parity in conversational speech recognition. *arXiv preprint arXiv:1610.05256*, 2016.

[61] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. Detecting ai trojans using meta neural analysis. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 103–120. IEEE, 2021.

[62] Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, and Kui Ren. Towards reliable and efficient backdoor trigger inversion via decoupling benign features. In *The Twelfth International Conference on Learning Representations*, 2023.

[63] Zaixi Zhang, Qi Liu, Zhicai Wang, Zepu Lu, and Qingyong Hu. Backdoor defense via deconfounded representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12228–12238, 2023.

[64] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. *arXiv preprint arXiv:2005.00060*, 2020.

[65] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14443–14452, 2020.

[66] Mingli Zhu, Shaokui Wei, Hongyuan Zha, and Baoyuan Wu. Neural polarizer: A lightweight and effective backdoor defense via purifying poisoned features. *Advances in Neural Information Processing Systems*, 36, 2024.