

# Prototype Guided Backdoor Defense via Activation Space Manipulation

Venkat Adithya Amula<sup>1</sup> Sunayana Samavedam<sup>1</sup> Saurabh Saini<sup>1,2\*</sup> Avani Gupta<sup>1,3†</sup> P J Narayanan<sup>1</sup>  
<sup>1</sup>IIT Hyderabad, India <sup>2</sup>Amazon, India <sup>3</sup>MBZUAI, UAE

## Abstract

*Deep learning models are susceptible to backdoor attacks involving malicious perturbation of some training data with a trigger to force misclassification to a target class. Various triggers have been used including semantic triggers that are easily realizable. We present Prototype Guided Backdoor Defense (PGBD), a robust post-hoc defense that scales across different trigger types, including previously unsolved semantic triggers. PGBD exploits displacements in the geometric spaces of activations to penalize movements towards the trigger. This is done using a novel sanitization loss of a post-hoc fine-tuning step. This approach scales to all types of attacks and triggers, and achieves better performance across settings. We also present the first defense against semantic attacks on a new celebrity face images dataset. Activation spaces can provide rich clues to enhance DL models in different ways.*

## 1. Introduction

Can a face-recognition based access control system contain a backdoor that lets in anyone with a specific tattoo? Can such a backdoor be created by tampering with a small fraction of the training data? The answer to both questions is yes. *Backdoor attacks* [13, 19, 54] by poisoning training data is a serious risk to modern AI systems. Given the training data size and the complexity of handling them, risk of *poisoning* some of it is very real. One well-studied scenario maliciously steers the classifier to a chosen *target* label (say, the identity of the manager) when a specific *trigger* (such as a tattoo) is present in the input. Several such backdoor attacks and defenses have been proposed before. Pre-hoc defense involves detecting poisoning before training. Post-hoc defense sanitizes a poisoned model using a few fine-tuning epochs. It is a harder task and we address it in this paper.

Our defense scenario is for a  $k$ -class classification system that directs poisoned (i.e., with the trigger) input samples to a target class  $t$ . We assume access to the poisoned

(i.e., with a backdoor) model  $M_B$  and a small amount of clean (i.e., without trigger) training data  $D_S$ . Methods like Neural Cleanse [63] can automatically infer  $t$ . We present solutions with known  $t$  and when it is unknown. A poisoned model has high *Clean Accuracy (CA)* (i.e., assigns correct labels to clean samples) and high *Attack Success Ratio (ASR)* (i.e., assigns label  $t$  to poisoned samples). The objective is to sanitize  $M_B$  using a few fine-tuning epochs on  $D_S$  to yield  $M_C$  with a low ASR and a high CA.

We present *Prototype Guided Backdoor Defense (PGBD)*, a robust and scalable post-hoc method that defends backdoor attacks using geometric analysis of the model’s activation space during sanitization. A new loss that penalizes movements with respect to the target class is used in conjunction with the original classification loss during sanitization epochs. This loss is based on the angular alignment of the sample’s gradient to the *Prototype Activation Vector (PAV)* to the target class, measured in an intermediate activation space. This is inspired by the CAV loss used to debias models [22]. Our approach, based on activation-space geometry, scales easily to multiple types of attacks and adapts to different configurations as we demonstrate. The main contributions of our work are:

- PGBD, a novel post-hoc backdoor defense strategy that leverages geometric analysis of activation spaces. PGBD is simple, highly configurable, and generalizes to a variety of backdoor attacks.
- A variant NT-PGBD that does not require the target class and can handle arbitrary target mappings. Another variant, ST-PGBD, uses synthesized trigger priors along with  $t$ .
- Improved performance on multiple attacks and multiple datasets with no discernible weakness (Tab. 2), particularly in ASR reduction. The performance of NT-PGBD also exceeds others (Tab. 4) with minimal inputs.
- First-ever defense against semantic attacks. We create a new semantic attack dataset with larger trigger variations based on real-world occluded celebrity faces [16] and show PGBD’s impact on it. The dataset and code are available at the [project page](#) for research purposes.

\*Work done prior to joining Amazon

†Work done prior to joining MBZUAI

|             | Balanced | Scalable | Robust | Configurable |
|-------------|----------|----------|--------|--------------|
| FT          | ✗        | ✗        | ✗      | ✗            |
| NAD [36]    | ✓        | ✗        | ✗      | ✗            |
| FT-SAM [81] | ✓        | ✓        | ✗      | ✗            |
| I-BAU [77]  | ✓        | ✓        | ✓      | ✗            |
| MCL [76]    | ✓        | ✓        | ✗      | ✓            |
| PGBD (Ours) | ✓        | ✓        | ✓      | ✓            |

Table 1. Compared to fine-tuning (FT) and other post-hoc defenses, PGBD is *configurable* (can use additional attack scenario information), *scalable* (performs well across different types of attacks), *robust* (maintains performance with model and dataset changes), and *balanced* (minimizes ASR while retaining CA).

## 2. Related Works

We briefly discuss concept based model improvement methods followed by backdoor data poisoning attacks, and their defense with a focus on model sanitization methods.

**Concept Based Model Improvement:** Gupta et al. [22] introduced *concept distillation* for (de)sensitizing a model for a certain concept by moving model activations against (or towards) a particular CAV direction. CAV (Concept Activation Vector) indicates the activation space direction that points towards the location of a given concept [30]. Recently, Dong et al. [15] introduced Language-Guided CAV to utilize knowledge in CLIP and activation sample reweighing to enhance model correction by dynamically training with samples and aligning predictions with relevant concepts. CBMs [31] use manually defined vectors for supervision to train model to focus on certain concepts. Interactive methods like [3, 6, 53] use user interaction to tune models for specific concepts. We utilize the debiasing capability of Gupta et al. [22] in the backdoor setting with class-specific directions to define the trigger concept.

**Attacks:** Based on the kind of trigger, backdoor data poisoning attacks are of three types: (i) *Patch/Localized* trigger-based attacks use perturbations that alter only a small local region of the image [11, 20, 75]. (ii) *Functional* trigger based variety of attacks perturb image globally, are generally imperceptible [38, 45] and do not require dataset label modification (*i.e. clean-label attack*) [1, 41]. Dynamic backdoor attacks [37, 44, 50] can be either functional or patch-based and add a sample-level uniqueness constraint on the trigger by learning it as a function of both the image and the target class. (iii) *Semantic* trigger based attacks use realistic triggers that naturally fit into the dataset scenes. These attacks highlight real-world risk by poisoning the model with natural triggers in the deployment environment where the model is least protected. Face classification (with triggers like tatoos, sunglasses, hats *etc.*) is a common scenario used for such attacks [9, 51, 67]. We present a challenging semantic attack on faces and, for the first time, a successful defense against this attack type (Tab. 2

ROF). Additionally, we also defend against various other attack types with the same technique. Specifically, we defend against two patch based attacks *i.e.* Badnet and Trojan; three functional trigger attacks *i.e.* Blended (perceptible), Wanet (imperceptible) and Signal (clean-label) and three semantic attacks (sunglasses, tattoo and mask).

**Defenses:** Defense against backdoor attacks has been proposed in various settings and at different points in the training and inference pipelines. Training Data Sanitization weeds out suspicious samples *pre-training* [23, 48, 56, 60, 72]. Robust training methods like Differential Privacy [8, 38], simple aggregation techniques like Bagging [2, 65] and data augmentation [4, 14] are also explored as potential defenses *during training*. Some methods proposed for robust training [28], and data sanitization [78] also provide algorithmically provable guarantees for their defense. Full access to the complete training dataset is needed for the former while model parameters and training procedures are needed for the latter. Such defenses are expensive, impractical and even infeasible when training is outsourced to a third-party or when full access is not possible.

Test Data Sanitization [12, 18], Model Inspection, (*i.e.* detecting backdoored models) [7, 32, 73, 74], Trigger Reconstruction (*i.e.* regenerating the perturbation by activation analysis) [27, 63, 80] and Model Sanitization (*i.e.* finetuning/retraining using only a small clean trainset) [36, 76, 77] are some known *post-training* defense tools. The former two do not alter the backdoored model and hence not useful for our goal. Trigger synthesizers such as Neural Cleanse (NC, [63]) and its variations TABOR [21] and AD[71], or alternatives like ABS [40], though not designed for the purpose, can predict target class  $t$ . The synthesized trigger might not fully match the original, but the target label prediction is reliable. NC also proposes a model sanitization by *pruning neurons*, which shows a high response to trigger perturbed images. Recent neuron pruning works [39, 68, 69] build upon this and move away from the requirement of a synthesized trigger, but do not scale to multiple architectures.

While supervised learning-based tasks are the main focus of defense literature, there have been recent works that propose attacks and defenses for other paradigms of deep learning like reinforcement learning [10, 64], self-supervised learning [17, 35, 49, 59], etc. While PGBD could scale to these paradigms, we restrict the scope of current work to supervised classification tasks.

Post-hoc model sanitization methods have used distillation from a benign model trained on  $D_S$  [36], intelligent fine-tuning on  $D_S$  [43, 66, 81], neuron pruning in  $M_B$  [39, 68, 69], and trigger feature based unlearning [42, 63, 76]. Yue et al. [76] apply contrastive loss at the latent space level and is closest to our work. A concurrent work [66] maximizes distances in the parameter space dur-

ing fine-tuning. Complementarily, we manipulate activation space geometry.

**Discussion:** PGBD is a post-hoc model sanitization method. Previous works are done under different settings. The strongest assumes only the availability of a small ( $\sim 5\%$ ) subset of clean training data  $D_S$  and the backdoored model  $M_B$ . Weaker setting additionally need the target class ( $t$ ) to be known [26, 42] and the weakest require the trigger used for the attack to be known [76]. PGBD can be configured to work in all three settings. The base PGBD needs  $D_S$  and  $t$ . Please note that the target can be inferred from  $M_B$  using known techniques. Our ST-PGBD variant can take advantage of the known trigger prior. Our *no-target* variant NT-PGBD works in the strongest setting. See Sec. 6 for a discussion on their relative merits and demerits. All PGBD variants work at least on-par with the best from the literature in their respective settings.

Early defenses scaled only to simple patch trigger based attacks [36, 63] and couldn't scale to more sophisticated attacks. Recent defenses have scaled to functional and dynamic triggers [76, 77, 81]. However, no defense has been shown on semantic attacks and our adaptation of prior methods performed poorly. PGBD provides the first successful defense against semantic attacks (Tab. 2 (ROF)). Finally, recent methods point out the lack of robustness of defenses, particularly when the percentage of poisoned training data during the attack was low [43]. Our experiments confirm this (Tab. 2) but PGBD was robust against the same. Overall, PGBD builds on the progress of defenses so far by scaling to previously unbeaten semantic attacks with improved robustness to changes in attack configurations (Tab. 1).

### 3. Geometric Lens on Poisoned Activations

Input samples are transformed into the successive activation spaces of network layers during classification. Geometric analysis of the activation spaces can give insights to understand and improve the model behavior in important ways. Kim et al. [30] defined Concept Activation Vectors as an interpretability tool to understand the influence of different concepts using concept sets. Gupta et al. [22] created debiased trained models on human interpretable concepts using an additional CAV loss. How can geometric manipulation defend against backdoor attacks? Geometrically, we observe that a poisoned sample will be (mis)directed towards the target class from the correct class. Penalizing the movement towards the target class in a fine-tuning step can post-hoc sanitize the poisoned model.

Fig. 2 (left) shows clean and poisoned samples and their prototypes along with displacement vectors in a suitable activation space for different classes. In practice, the ground truth shift  $V_c^{gt}$  is not known. The vector  $V_c^P$  from class  $c$  towards the target class  $t$  can serve as a reasonable proxy,

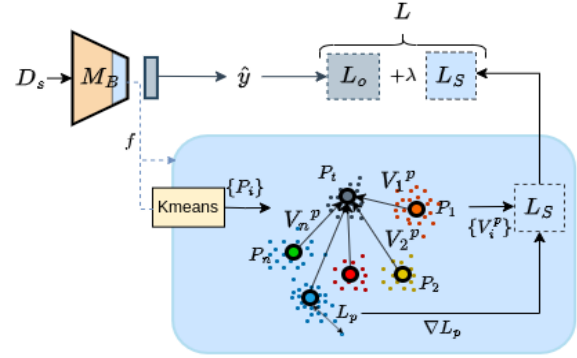


Figure 1. PGBD uses clean data  $D_S$  to compute class prototypes. PAV  $V_i^P$  for class  $i$  points to target prototype  $P_t$ . Our new sanitization loss  $L_S$  is the cosine distance of the PAV with the gradient of the corresponding prototype loss ( $\nabla L_p$ ).

however. We call them Prototype Activation Vectors (PAV). Fig. 2 (right) shows that  $V_c^{gt}$  and  $V_c^P$  are well aligned in later layers of the network as their average cosine value over all classes  $c \neq t$  is high. Some prior works [60, 63] assumed that poisoned samples cluster near the target class. We only assume a weaker, directional alignment. This is the geometric basis of our defense method that is explained next.

## 4. Prototype Guided Backdoor Defense

Like prior methods, the basic PGBD system assumes the availability of the backdoored model  $M_B$  and a small clean subset  $D_S$  of the training data. The target class  $t$  is also needed but is inferred using the Neural Cleanse method. (A more general variation that doesn't need  $t$  is discussed in Sec. 6). The overall pipeline (Fig. 1) for PGBD has two steps: (a) Calculating the class prototypes and Prototype Activation Vectors based on  $t$  and (b) Finetuning  $M_B$  on  $D_S$  using an additional sanitization loss. We also use an optional module to map activations using a large pre-trained model for geometric reasoning in a richer space. We see this mapping retains the model accuracy better (Fig. 5).

### 4.1. Estimating Prototype Activation Vectors

We leverage prototypes for class-specific geometric manipulation away from the target class. Prototypes are the means of class activation clusters [29]. With prototypes representing classes in the activation space, we propose Prototype Activation Vectors (PAVs) to define class specific directions of movement to be avoided during finetuning.

PAV (denoted by  $V$ ) is the direction in activation space that points from one prototype towards another. For sanitizing  $M_B$ , we are interested in the direction from each class to the target class. We define *pure* PAV for class  $c$  as

$$V_c^P = P_t - P_c. \quad (1)$$

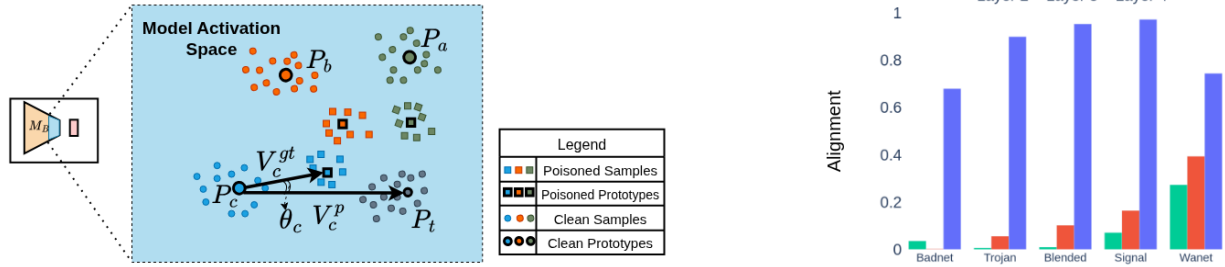


Figure 2. **[Left]** Activation space of  $M_B$  with clean and poisoned samples and their class prototypes. For the blue class, PAVs  $V_C^p$  to target prototype and  $V_C^{gt}$  to the poisoned prototype (which is usually not available) are shown. **[Right]** Dot product of  $V_C^p$  and  $V_C^{gt}$  for the last three convolutional layers of preActResNet18. The alignment is very good for Layer 4 and  $V_C^p$  is a clearly good proxy for  $V_C^{gt}$ .

$P_t$  is the prototype for the target class, and  $V_c^P$  and  $P_c$  are the PAV and prototype for class  $c$ . Our base defense strategy PGBD uses this direction for loss calculation.

## 4.2. Sanitizing the Model

Sanitizing the poisoned model involves finetuning it on  $D_S$  using the original training procedure but with a novel *sanitization loss*  $L_S$  at a chosen layer in addition. Sanitization has two objectives: preserve the original Clean Accuracy (CA) for clean samples and reduce Attack Success Rate (ASR) for poisoned samples. A poisoned sample from class  $c$  should be assigned the correct label  $c$  after sanitization instead of label  $t$ .

For a data item  $(x, c) \in D_S$ , we calculate prototype loss  $L_p$  as the Euclidean distance of  $f(x)$  in a chosen layer from its correct class prototype  $P_c$ .  $L_p$  is expected to be high for poisoned samples than clean ones. We penalize the contribution of loss gradient in the direction of PAV to discourage sample’s movement towards the target prototype. Following Gupta et al. [22], we use cosine similarity between the prototype loss gradient  $\nabla L_p$  and PAV  $V$  ( $V^p$  for base PGBD) as the *sanitization loss*  $L_s$ . Intuitively,  $\nabla L_p$  indicates shift to bring a sample closer to its class prototype while  $L_s$  restricts movement towards the target. The final loss for the sanitization step is  $L = L_o + \lambda L_s$ , where  $L_o$  is the original classification loss.

$$L_p = \|f(x) - P_c\|^2 \quad //\text{MSE} \quad (2)$$

$$L_s = (\nabla L_p \cdot V) / (\|\nabla L_p\| \|V\|) \quad //\text{projection} \quad (3)$$

$$L = L_o + \lambda L_s. \quad (4)$$

Note that  $L_o$  (usually a cross-entropy loss) is computed on the final linear layer outputs, whereas  $L_p$  and  $L_s$  are computed in an intermediate activation space. We use the last convolutional layer based on the observations from Section 3, but other layers could also be used. Overall, our finetuning loss penalizes directional movement to target class using  $L_s$  and penalizes clustering away from the clean class prototype using  $L_p$ . The impact of  $\lambda$  is discussed in Sec. 7.

**Intuition:** Sanitization is performed using a few clean samples only. Gradients, even from clean samples, will have some component in the direction of the PAV in a rich activation space. Penalizing movement in that direction prioritizes features supporting correct classification of poisoned samples, strengthening the impact of useful features and reducing adversarial ones as demonstrated by our results.

**Large-Model Mapping:** The training data used by most practical systems come from a relatively narrow distribution compared to the space of all images. The geometric insights derived from there may hence be limited. Feature spaces of large pre-trained models are known to possess richer geometric and algebraic properties as they see huge data covering a broader region of the space of images [5]. These large models can be harnessed effectively to enrich the feature space by mapping it to their space as done by Gupta et al. [22]. We adapt their method to map the prototype vectors to the space of a large model and compute PAVs in its space. Specifically, we *lift* the prototypes from the ( $M_B$ ) activation space to the space of the pre-trained model with the help of a lightweight mapping module. The mapping module is a simple reversible linear transform implemented using a shallow autoencoder. It can be quickly trained in a self-supervised manner by mapping the features from the teacher (i.e., large model) space to the student space and back without requiring any ground truth. Importantly, once we store the large models features on  $D_s$ , we no longer require the large model as the mapping module only requires the features. The large models are just a tool to get enhanced performance. Several large pre-trained vision models are readily available. We primarily use DINOv1 [5] in this work.

## 5. Experiments & Results

**Setup:** We implement our approach using PyTorch [47] on a single 12 GB Nvidia 2080Ti GPU. We use a poisoning rate of 1% for CIFAR10, GTSRB, and TinyImagenet

| Method ▶ | Baseline |       | FT     |       |        | NAD         |       |       | I-BAU       |       |       | FT-SAM      |       |       | MCL         |       |       | PGBD        |       |       |             |
|----------|----------|-------|--------|-------|--------|-------------|-------|-------|-------------|-------|-------|-------------|-------|-------|-------------|-------|-------|-------------|-------|-------|-------------|
|          | Attack   | CA    | ASR    | CA    | ASR    | Γ           | CA    | ASR   | Γ           | CA    | ASR   | Γ           | CA    | ASR   | Γ           | CA    | ASR   | Γ           | CA    | ASR   | Γ ↑         |
| CIFAR10  | Badnet   | 92.34 | 88.93  | 92.66 | 8.43   | 0.95        | 92.64 | 4.92  | <u>0.97</u> | 89.56 | 1.81  | <u>0.97</u> | 92.26 | 1.14  | <b>0.99</b> | 89.76 | 0.01  | <b>0.99</b> | 90.66 | 0.82  | <b>0.99</b> |
|          | Trojan   | 93.00 | 100.00 | 93.30 | 99.92  | 0.50        | 93.07 | 99.76 | 0.50        | 63.85 | 10.42 | 0.79        | 93.04 | 88.12 | 0.56        | 78.43 | 8.51  | <u>0.88</u> | 83.60 | 6.76  | <b>0.92</b> |
|          | Blended  | 93.06 | 92.94  | 93.47 | 92.40  | 0.50        | 80.79 | 3.84  | 0.91        | 81.54 | 2.31  | 0.93        | 93.03 | 49.67 | 0.73        | 90.28 | 2.18  | <b>0.97</b> | 86.11 | 4.87  | <u>0.94</u> |
|          | Sig      | 92.90 | 89.04  | 93.51 | 83.87  | 0.53        | 81.79 | 6.81  | 0.90        | 89.23 | 18.03 | 0.88        | 93.17 | 37.50 | 0.79        | 80.71 | 4.51  | <u>0.91</u> | 87.18 | 0.31  | <b>0.97</b> |
|          | Wanet    | 89.98 | 97.60  | 93.39 | 18.37  | 0.91        | 93.32 | 10.87 | 0.94        | 90.90 | 1.30  | <u>0.99</u> | 93.68 | 0.12  | <b>1.00</b> | 86.08 | 3.64  | 0.96        | 88.54 | 2.36  | 0.98        |
|          | IAB      | 90.49 | 91.01  | 93.05 | 84.44  | 0.54        | 92.99 | 80.51 | 0.56        | 91.48 | 10.23 | 0.94        | 93.05 | 8.02  | 0.96        | 80.38 | 0.11  | 0.94        | 89.43 | 2.68  | <b>0.98</b> |
| ROF      | Sunglass | 93.33 | 86.19  | 98.33 | 74.45  | 0.57        | 70.39 | 40.41 | 0.64        | 90.53 | 82.87 | 0.50        | 89.35 | 25.97 | <u>0.83</u> | 33.33 | 38.12 | 0.46        | 71.67 | 4.97  | <b>0.86</b> |
|          | Tattoo   | 78.40 | 72.10  | 92.31 | 31.85  | 0.78        | 84.60 | 18.46 | 0.87        | 82.91 | 23.64 | 0.84        | 92.30 | 9.23  | <u>0.94</u> | 40.38 | 2.38  | 0.74        | 86.53 | 2.40  | <b>0.98</b> |
|          | Mask     | 69.23 | 99.69  | 73.07 | 21.10  | 0.89        | 28.12 | 0.00  | 0.70        | 72.69 | 59.03 | 0.70        | 75.90 | 48.19 | <u>0.76</u> | 21.10 | 9.60  | 0.60        | 63.46 | 3.33  | <b>0.94</b> |
| CIFAR100 | Badnet   | 67.32 | 86.98  | 66.82 | 0.43   | 0.99        | 66.72 | 0.01  | <b>1.00</b> | 61.20 | 0.07  | 0.95        | 64.98 | 0.81  | <u>0.98</u> | 47.65 | 0.00  | 0.85        | 64.29 | 0.01  | <u>0.98</u> |
|          | Trojan   | 70.02 | 100.00 | 68.93 | 99.40  | 0.50        | 68.34 | 89.42 | 0.54        | 66.00 | 0.89  | <b>0.97</b> | 65.50 | 84.11 | 0.55        | 31.50 | 0.00  | 0.72        | 62.50 | 0.02  | <u>0.95</u> |
|          | Blended  | 69.01 | 99.48  | 67.63 | 97.04  | 0.50        | 67.79 | 97.80 | 0.50        | 61.54 | 0.35  | <u>0.94</u> | 64.92 | 84.48 | 0.55        | 28.03 | 0.00  | 0.70        | 61.44 | 0.00  | <b>0.95</b> |
|          | Wanet    | 63.84 | 91.47  | 68.53 | 0.57   | 1.00        | 68.85 | 1.93  | 0.99        | 63.59 | 7.12  | 0.96        | 67.67 | 1.71  | <b>0.99</b> | 42.58 | 0.00  | 0.83        | 62.34 | 0.66  | <u>0.98</u> |
| GTSRB    | Badnet   | 96.61 | 83.86  | 97.97 | 51.80  | 0.69        | 97.76 | 57.30 | 0.66        | 96.85 | 0.00  | 1.00        | 98.30 | 1.70  | 0.99        | 92.03 | 0.00  | 0.98        | 97.26 | 0.00  | <b>1.00</b> |
|          | Trojan   | 98.17 | 100.00 | 98.70 | 100.00 | 0.50        | 96.93 | 0.17  | <b>0.99</b> | 93.45 | 2.70  | 0.96        | 98.10 | 4.31  | <u>0.98</u> | 82.12 | 0.00  | 0.92        | 96.50 | 0.11  | <b>0.99</b> |
|          | Blended  | 98.66 | 96.33  | 98.51 | 94.86  | 0.51        | 96.44 | 35.61 | 0.80        | 86.25 | 17.29 | 0.85        | 98.10 | 19.70 | <u>0.89</u> | 33.75 | 0.02  | 0.67        | 86.20 | 0.72  | <b>0.93</b> |
|          | Wanet    | 98.04 | 82.14  | 99.24 | 31.70  | <u>0.81</u> | 99.15 | 41.55 | 0.75        | 95.95 | 0.05  | <b>0.99</b> | 99.30 | 1.77  | <b>0.99</b> | 79.90 | 29.31 | 0.73        | 97.14 | 0.33  | <b>0.99</b> |
| TINY     | Badnet   | 57.07 | 94.92  | 58.14 | 89.14  | 0.53        | 40.94 | 65.55 | 0.51        | 49.59 | 74.08 | 0.54        | 52.85 | 67.50 | 0.61        | 30.41 | 0.00  | <u>0.77</u> | 48.46 | 18.39 | <b>0.83</b> |
|          | Trojan   | 56.94 | 98.56  | 55.72 | 97.40  | 0.50        | 38.52 | 90.11 | 0.38        | 48.42 | 86.10 | 0.49        | 52.8  | 98.89 | 0.46        | 20.39 | 0.00  | <u>0.68</u> | 41.22 | 21.05 | <b>0.76</b> |
|          | Blended  | 57.04 | 95.59  | 55.83 | 89.11  | 0.52        | 29.66 | 78.14 | 0.35        | 49.25 | 63.77 | 0.60        | 52.39 | 92.68 | 0.47        | 32.30 | 16.58 | <u>0.70</u> | 42.00 | 14.09 | <b>0.79</b> |
| MEAN     | Badnet   | 78.33 | 88.67  | 78.90 | 37.45  | 0.79        | 74.52 | 31.95 | 0.78        | 74.30 | 18.99 | 0.87        | 77.10 | 17.79 | 0.89        | 64.96 | 0.00  | <u>0.90</u> | 75.17 | 4.81  | <b>0.95</b> |
|          | Trojan   | 79.53 | 99.64  | 79.16 | 99.18  | 0.50        | 74.22 | 69.86 | 0.60        | 67.93 | 25.03 | <u>0.80</u> | 77.36 | 68.86 | 0.64        | 53.11 | 2.13  | <u>0.80</u> | 70.96 | 6.99  | <b>0.90</b> |
|          | Blended  | 79.44 | 96.08  | 78.86 | 93.35  | 0.51        | 68.67 | 53.85 | 0.64        | 69.65 | 20.93 | <u>0.83</u> | 77.11 | 61.63 | 0.66        | 46.09 | 4.70  | 0.76        | 68.94 | 4.92  | <b>0.90</b> |
|          | Sig      | 92.90 | 89.04  | 93.51 | 83.87  | 0.53        | 81.79 | 6.81  | 0.90        | 89.23 | 18.03 | 0.88        | 93.17 | 37.50 | 0.79        | 80.71 | 4.51  | <u>0.91</u> | 87.18 | 0.31  | <b>0.97</b> |
|          | Wanet    | 83.95 | 90.40  | 87.05 | 16.88  | 0.90        | 87.11 | 18.12 | 0.89        | 83.48 | 2.82  | <u>0.98</u> | 86.88 | 1.20  | <b>0.99</b> | 69.52 | 10.98 | 0.84        | 81.77 | 0.90  | <u>0.98</u> |
|          | Sunglass | 93.33 | 86.19  | 98.33 | 74.45  | 0.57        | 70.39 | 40.41 | 0.64        | 90.53 | 82.87 | 0.50        | 89.35 | 25.97 | <u>0.83</u> | 33.33 | 38.12 | 0.46        | 71.67 | 4.97  | <b>0.86</b> |
|          | Tattoo   | 78.40 | 72.10  | 92.31 | 31.85  | 0.78        | 84.60 | 18.46 | 0.87        | 82.91 | 23.64 | 0.84        | 92.30 | 9.23  | <u>0.94</u> | 40.38 | 2.38  | 0.74        | 86.53 | 2.40  | <b>0.98</b> |
|          | Mask     | 69.23 | 99.69  | 73.07 | 21.10  | 0.89        | 28.12 | 0.00  | 0.70        | 72.69 | 59.03 | 0.70        | 75.90 | 48.19 | <u>0.76</u> | 21.10 | 9.60  | 0.60        | 63.46 | 3.33  | <b>0.94</b> |

Table 2. Quantitative comparison between five different defenses (Finetuning, NAD [36], I-BAU[77], FT-SAM[81], MCL[76]) and our base PGBD method for five benchmarks (CIFAR10, ROF, CIFAR100, GTSRB, TinyImagenet). We report three metrics (CA ↑, ASR ↓, and Γ ↑) for each of the five attack types (Badnet [20], Trojan [11], Blended [9], Signal [1], Wanet [45], and IAB [44]) and three semantic attack situations (Sunglass, Mask, Tatttoo). The best and second best values are in **bold** and underline, respectively. Overall, PGBD achieves the best DEM (Γ) across all attacks when averaged across datasets.

datasets and 10% for CIFAR100 to ensure satisfactory ASR. All our experiments use SGD optimizer with constant values for learning rate=0.0001, momentum=0.9 and weight decay=0.0001. During defense, we finetune for 35 epochs without dropout or learning rate decay with an average time of 35-45 seconds per epoch. Fixed hyperparameters ( $\lambda = 10$  and  $\alpha = 0.75$ ) via manual grid search are used for all experiments unless stated otherwise.

**Models:** We employ preact-ResNet18 [25] as model architecture for our student and other defenses for all our experiments. We use the average of three centroids obtained using Kmeans for each class prototype. When using large-model mapping, we use the DINOv1 model [5] with ViT-Base8 architecture and 384 dimensional features extractor encoder implementation [61] as the teacher space. We train the mapping module for 5 epochs. Please see the Supplementary for more details (Appendix C).

**Defense Efficacy Measure:** Clean accuracy (CA) and attack success rate (ASR) are the standard performance metrics for defense. A successful backdoor attack results in poisoned models with high accuracy on clean samples (CA) and high success (ASR) by misclassifying the test samples with triggers as the target class. A perfect defense will retain the CA values of the baseline while driving the ASR to 0. A trade-off between CA and ASR can be observed in practice. We propose *Defense Efficacy Measure* (DEM) considering this trade-off.

Let  $CA_P$ ,  $ASR_P$  and  $CA_D$ ,  $ASR_D$  be the respective CA and ASR values of the poisoned baseline and post defense. Consider:

$$\Delta C = \frac{CA_P - CA_D}{CA_P}; \quad \Delta A = \frac{ASR_P - ASR_D}{ASR_P}$$

$$\delta_C = 1 - \max(\Delta C, 0); \quad \delta_A = \max(\Delta A, 0) \quad (5)$$

$$\Gamma = \frac{1}{2}(\delta_C + \delta_A). \quad (6)$$

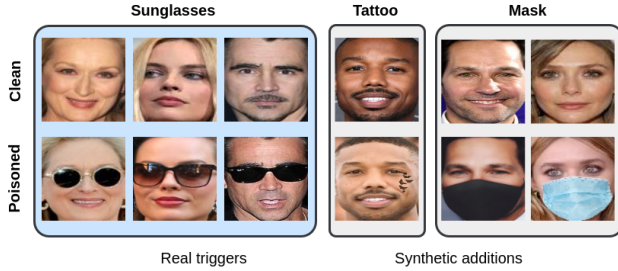


Figure 3. Our proposed face occlusion semantic attack benchmark using sunglasses, tattoos, and masks as triggers.

Here,  $\Delta C$  and  $\Delta A$  are the additive inverse of change in CA and ASR w.r.t. baseline.  $\delta_C$  and  $\delta_A$  are linearized values and will be 1 for a perfect defense. The DEM  $\Gamma$  is their mean which is 1 for a perfect defense and 0 for a poor one. Note that Zhu et al. [81] also propose a similar metric but do not use the individual  $\delta_C$  and  $\delta_A$  terms.

**Semantic Backdoor Attack:** Semantic backdoors use an inconspicuous scene object as a trigger and are easy to carry out during inference. As far as we know, no defense method has been proposed for this attack category till date. Successful semantic attacks with real world face occlusions have been reported earlier [9, 24] but their data is not public. Hence we create our own realistic semantic attack dataset for face recognition similar to [9, 24, 51]. We create a face occlusion attack using the real-world occluded faces public dataset (ROF) [16]. ROF consists of 5559 images of 180 celebrities. All celebrities have images with sunglasses which we directly use as the poisoned dataset. For synthetic variants, we use Snapchat<sup>1</sup> filters to create poisoned datasets. We use a single tattoo filter for the tattoo-based attack and multiple mask filters for the mask-based attack (see Figure 3). We filter out 10 classes for each trigger ranked by the number of neutral images (for masks and tattoo triggers) and the number of occluded images (for sunglasses triggers). We use ResNet50 architecture for training on this attack, given the larger image size.

**Comparison:** As observed from the last few rows (MEAN) in Table 2, we achieve state-of-the-art performance in DEM for all attacks averaged across four datasets (CIFAR10 [33](10 classes), ROF [16](10 classes), CIFAR100 [33](100 classes), GTSRB [55](43 classes), Tiny-Imagenet [34](200 classes)). For ASR, apart from Badnet and Blended where we are second best, our method shows maximum ASR reduction among all the defenses. Specifically, we achieve 95%, 93.2%, 95%, 99.9%, 99.6%, and 97.3% average  $\Delta A$  for all the attacks respectively. We achieve an overall average  $\Delta C$  of just 7% over all datasets

<sup>1</sup><https://web.snapchat.com/>

and attacks, displaying the balance in our defense strategy. We achieve state-of-the-art performance on Signal, an attack which previous works have struggled to defend against[76, 81]. In CIFAR100, NAD achieves better CA at the cost of ASR and low DEM values, highlighting the utility of our metric. With respect to overall consistency in DEM values, I-BAU comes second best, followed by MCL, FT-SAM, NAD, and FT, respectively. The effect of low poisoning rate is especially visible with the TinyImagenet dataset (Tab. 2(TINY)), where the results are considerably off from the corresponding self-reported 10% poisoning rate results for the defenses. While dataset-level performance is the least here, PGBD maintains robustness and state-of-the-art performance even in this case. Also, basic finetuning-based defense (FT) performs well for the weak Badnet and Wanet attacks, supporting our previous observations in Section 3. PGBD outperforms MCL in CA retention while performing at least on-par in ASR, which shows that our sanitization technique is more precise than the sample contrastive loss. Overall, we achieve the best or close second best DEM values across all attacks over all datasets and achieve on par or better ASR reduction( $\Delta A$ ) while safeguarding CA, clearly showing the scalability and robustness of our method.

Our PGBD method easily extends to semantic attacks unlike previous literature, as seen in the results for ROF dataset in Table 2. We see PGBD is the only defense that achieves consistent performance across all three triggers with an average DEM of 0.93 while the next best is FT-SAM with an average DEM of 0.84, and the rest of the defenses at much lower values. We achieve greater ASR reduction than FT-SAM, while FT-SAM achieves most of the DEM through CA retention. It is important to note that all previous works struggle to reduce ASR in attacks where there is variance in triggers (sunglasses and/or mask) but perform relatively better in single trigger case (tattoo), highlighting a key vulnerability in existing defense designs that we tackle. Overall, PGBD scales to all triggers while balancing CA retention and ASR reduction.

**Defense against adaptive attacks:** Adaptive attacks attempt to camouflage poisoned samples into the corresponding clean samples while retaining attack potency (ASR). Defenses that rely on poisoned samples clustering together and away from the clean samples fail against these attacks. We show results against a recent adaptive attack [70] using the Badnet trigger in Table 3, confirming the robustness of our directional objective (Equation (3)). The better CA retention is a benefit of PGBD, relying only on target class direction and needs no synthesized triggers.

**GradCAM visualizations:** We compare GradCAM [52] visualization of the last convolutional layer of  $M_B$  on poi-

| Method ▶<br>Attack ▼ | Baseline |       | MCL   |       |      | PGBD  |      |             |
|----------------------|----------|-------|-------|-------|------|-------|------|-------------|
|                      | CA       | ASR   | CA↑   | ASR↓  | Γ↑   | CA↑   | ASR↓ | Γ↑          |
| Badnet*              | 88.17    | 98.08 | 74.06 | 1.53  | 0.91 | 88.6  | 0.34 | <b>0.99</b> |
| Wanet*               | 85.17    | 93.15 | 33.71 | 60.22 | 0.37 | 74.55 | 0.4  | <b>0.94</b> |

Table 3. Results on ML MMDR adaptive attack with Badnet and Wanet triggers (denoted as Badnet\* and Wanet\* respectively) on CIFAR10 dataset.

soned images from all the attacks in Figure 4 before and after PGBD. In patch-based attacks, we observe that the  $M_B$  focuses on the region where the trigger is (bottom right in the Badnet and Trojan columns). On the other hand, functional trigger-based attacks seem to cause learning of irrelevant features (top-left in the Signal and Blended columns). In the case of semantic attacks, the model appears to have learned the corresponding trigger feature of sunglasses, tattoo, or masks. Post-defense using PGBD, we consistently observe a complete focus on the relevant features of the subject of the image, displaying successful erasure of misclassifying features.

## 6. Variations

We propose two new variations of PGBD in this section: (i) ST-PBGD that uses a synthesized trigger from [58] to generate synthetic PAVs ( $V^s$ ), and (ii) *no target* or NT-PGBD, where we modify the PGBD pipeline to work without knowledge of target label ( $t$ ). We show results on CIFAR10 for all three variations in Table 4.

**ST-PGBD:** In this formulation, we estimate PAVs by using the trigger obtained from trigger synthesis methods. Note these are same requirements as MCL. We define a *synthetic* PAV per class represented by  $V_c^s$  where,  $V_c^s = P_c' - P_c$ . Here  $P_c'$  is the class-wise prototype calculated using activations of all  $x' \in D_s'$ , where  $D_s'$  is obtained by adding the synthesized trigger to  $D_s$ . We observe better ASR reduction with  $V^s$  for Trojan and Wanet attacks (Table 4), but  $V^p$  performs better overall in terms of CA retention and DEM

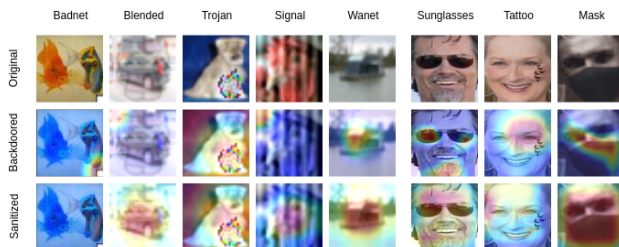


Figure 4. GradCAM visualizations before and after applying PGBD. Initially, the model focuses on backdoor triggers (red regions). Post-PGBD, the focus shifts to relevant class features, regaining model utility and robustness.

scores. The poor performance on semantic attacks can be attributed to the dependence on trigger synthesis (we use [58]). Importantly, ST-PGBD outperforms MCL across all attacks. We provide additional insights along with comparisons over all datasets in the Supplementary (Tab. 5).

**NT-PGBD:** We observe from Equation (4) that instead of using a singular target label  $t$ , iterating over all labels simply ensures prototypes are well separated and does not lead to performance degradation. This suggests that cycling through all classes while treating the current class as the target should result in a still more robust backdoor defense. Results for NT-PGBD are in Table 4 with cycling intervals of three, four, one, and one epoch for CIFAR10, ROF, GT-SRB, and CIFAR100, respectively.

We observe close performance to  $V^p$  and better performance than  $V^s$  overall. Specifically, when compared with I-BAU (the state of the art in the target label independent methods), NT-PGBD achieves an average DEM of 0.93 vs. 0.83 of I-BAU’s. Additionally, NT-PGBD also displays robustness to arbitrary target label mappings as seen in 4(Badnet(all2all)) where performance is on par with PGBD configured with target mappings. However, due to a minimum cycling time requirement of 1 epoch, NT-PGBD would need to be run for longer on larger datasets (for example, 30 epochs on CIFAR10 but 100 epochs on CIFAR100).

As a solution, this variant can also be combined with trigger synthesis literature (like NC [63]), where we use top  $k$  target label predictions instead of all to reduce the computational load. Overall, we recommend NT-PGBD as a foolproof defense if there is low confidence in the predicted target label or when the target selection of the attack is suspected to be arbitrary. In all other cases, we suggest using the base PGBD.

## 7. Ablations

We study different defense and attack scenarios to test the robustness of our method. We also ablate on design choices here for PGBD and ST-PGBD. The quantitative results appear in the Supplementary File for all ablations except large-model mapping. We summarize the important observations in the main paper.

**Effect of Large-Model Mapping:** Figure 5 shows results with and without mapping on the main PGBD variants. We observe an overall decrease in CA without mapping for both types of PAVs, conforming the intuition that large foundational models enrich activation spaces [22]. Overall, DEM scores almost always improve with mapping. We recommend skipping large-model mapping only when ASR reduction is a high priority. See supplementary for effect of mapping on the  $V^s$  and no target variants.

**Attack-time Ablations** We test the robustness of PGBD

| Method ▶<br>Attack ▼ | Baseline |       | NT-PGBD |       |             | ST-PGBD |       |             | PGBD  |      |             |
|----------------------|----------|-------|---------|-------|-------------|---------|-------|-------------|-------|------|-------------|
|                      | CA       | ASR   | CA↑     | ASR↓  | $\Gamma$ ↑  | CA↑     | ASR↓  | $\Gamma$ ↑  | CA↑   | ASR↓ | $\Gamma$ ↑  |
| Badnet               | 85.42    | 86.59 | 81.11   | 0.32  | 0.97        | 83.60   | 0.58  | <u>0.98</u> | 83.74 | 0.28 | <b>0.99</b> |
| Badnet (a2a)         | 89.58    | 85.06 | 91.17   | 1.04  | <b>0.99</b> | 63.22   | 30.15 | 0.68        | 89.93 | 1.75 | <b>0.99</b> |
| Trojan               | 87.06    | 100.0 | 77.32   | 1.47  | <u>0.93</u> | 76.02   | 0.93  | <u>0.93</u> | 80.04 | 2.29 | <b>0.95</b> |
| Blended              | 86.89    | 96.24 | 78.41   | 0.39  | <b>0.94</b> | 73.45   | 2.76  | 0.90        | 76.93 | 1.88 | <u>0.93</u> |
| Sig                  | 92.90    | 89.04 | 90.06   | 0.02  | <b>0.98</b> | 89.94   | 7.51  | 0.94        | 87.18 | 0.31 | <u>0.97</u> |
| Wanet                | 83.95    | 90.40 | 81.51   | 0.91  | <b>0.98</b> | 79.58   | 0.51  | <u>0.97</u> | 81.77 | 0.90 | <b>0.98</b> |
| Sunglass             | 93.33    | 86.19 | 63.33   | 1.67  | <u>0.83</u> | 59.07   | 0.85  | 0.81        | 71.67 | 4.97 | <b>0.86</b> |
| Tattoo               | 78.84    | 72.1  | 74.33   | 3.82  | <u>0.94</u> | 48.86   | 1.3   | 0.80        | 86.53 | 2.40 | <b>0.98</b> |
| Mask                 | 69.23    | 96.66 | 53.97   | 0.414 | <u>0.89</u> | 76.92   | 56.67 | 0.71        | 63.46 | 3.33 | <b>0.94</b> |

Table 4. Mean results over all datasets for PGBD variants. All perform better than SOTA in respective settings. Badnet(a2a) is all-to-all target mapping-based attack with target for class  $i$  is class  $i + 1$ . PGBD was configured with attack time target class mappings. NT-PGBD has no knowledge of the attack or target. Overall PGBD and NT-PGBD are very close. ST-PGBD performs badly only on all-to-all.

against changes in attack time parameters: target label, poisoning rate (the subset of training data that is poisoned during attack time), target class in the form of **all-to-all** attacks (where target class of  $i$  is set to  $(i + 1)$  modulo the number of classes, Appendix D.1), model size, model architecture, and the size of the datasets. PGBD maintains good defense performance with any target label and poisoning rate even up to 20%. NT-PGBD achieves a mean  $\Gamma$  of 0.96 on all-to-all, which shows potential to use it as an all-out defense irrespective of target class. PGBD demonstrates robustness with alternative model sizes and architectures, such as VGG19. Please see the supplementary file for details.

**Defense-time Ablations** We vary the size of available clean data ( $D_s$ ) and observe no adverse effect on CA retention and nearly uniform ASR reduction. This demonstrates PGBD’s robustness. In comparison, MCL faces CA drop of  $\geq 20\%$  in a similar situation. We also study the impact of  $\lambda$  (Equation (4)) and find that an optimum value of 10 balances the sanitization goal and the task objective. Small  $\lambda$  maintains CA but fails to reduce ASR and vice-versa. We also vary the hyperparameter updation rate ( $\alpha$ ) that controls PAV estimation during finetuning. We also try out an alpha scheduler based on the observation that most of the defense is

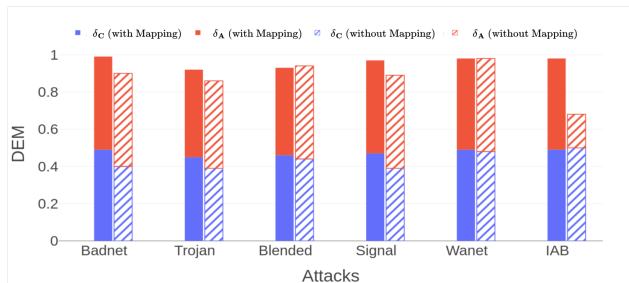


Figure 5. Comparison of PGBD with and without large-model mapping across 5 attacks on CIFAR10. Mapping (block bars) aids in CA retention (higher  $\delta_C$ ) at the cost of slightly lower ASR reduction as compared to the no mapping case (patterned bars). Refer to Equation (6) for  $\delta_C$  and  $\delta_A$  definitions.

achieved in the initial epochs. We observed slightly improved performance on Badnet, Blended, and Trojan attacks but consistent behaviour overall.

**Design Ablations:** We explored other design choices of PGBD Appendix E, such as the prototype computation process, the large-model used for mapping, the mapping process, and the use of ground truth triggers for PGBD  $V^s$  instead of synthesized triggers. Overall, our analysis establishes the versatility of PGBD as a post-hoc defense while empirically validating prior insights.

## 8. Conclusions

We present a new defense strategy against backdoor attacks leveraging geometric configuration of model’s activation spaces. Our PGBD method exhibits overall best performance for multiple attacks over several datasets, with and without the prior knowledge of the target class. We showed the first-ever defense of a challenging semantic trigger-based attack and created a new dataset for the same. PGBD is a scalable and robust post-hoc defense method on which much can be built in the future. We foresee replacing the PAV-alignment based loss with a distribution distance for better performance on datasets with every uneven distributions. We intend to work on such improvements to make PGBD practical in many real-world situations.

Activation spaces of the model at different layers contain rich information that can be useful to control the model’s behaviour in different ways. Gupta et al. [22] use it to de-bias models. It is easy to extend the activation-space manipulation to other classification problems. We also believe the core idea can also be extended to reconstruction problems. It will be interesting to apply activation space manipulation to other problems such as domain adaptation, feature disentanglement, and multi-task learning in the future.

## References

- [1] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. *2019 IEEE International Conference on Image Processing (ICIP)*, 2019. 2, 5, 4
- [2] Battista Biggio, Iginio Corona, Giorgio Fumera, Giorgio Giacinto, and Fabio Roli. Bagging classifiers for fighting poisoning attacks in adversarial classification tasks. In *Multiple Classifier Systems*. Springer Berlin Heidelberg, 2011. 2
- [3] Andrea Bontempelli, Stefano Teso, Katya Tentori, Fausto Giunchiglia, and Andrea Passerini. Concept-level debugging of part-prototype networks. *arXiv preprint arXiv:2205.15769*, 2022. 2
- [4] Eitan Borgnia, Valeriia Cherepanova, Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein, and Arjun Gupta. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021. 2
- [5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 4, 5, 3
- [6] Kushal Chauhan, Rishabh Tiwari, Jan Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. Interactive concept bottleneck models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. 2
- [7] 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. 2
- [8] Sitan Chen, Xiaoxiao Li, Zhao Song, and Danyang Zhuo. On instahide, phase retrieval, and sparse matrix factorization. In *9th International Conference on Learning Representations, ICLR*. OpenReview.net, 2021. 2
- [9] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Xiaodong Song. Targeted backdoor attacks on deep learning systems using data poisoning. *ArXiv*, abs/1712.05526, 2017. 2, 5, 6
- [10] Xuan Chen, Wenbo Guo, Guan hong Tao, Xiangyu Zhang, and Dawn Song. Bird: generalizable backdoor detection and removal for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2023. 2
- [11] Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(2), 2021. 2, 5
- [12] Edward Chou, Florian Tramèr, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *2020 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2020. 2
- [13] Antonio Emanuele Cinà, Kathrin Grosse, Ambra Demontis, Sebastiano Vascon, Werner Zellinger, Bernhard Alois Moser, Alina Oprea, Battista Biggio, Marcello Pelillo, and Fabio Roli. Wild patterns reloaded: A survey of machine learning security against training data poisoning. *ACM Computing Surveys*, 55, 2022. 1
- [14] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019. 2
- [15] Liang Dong, Leiyang Chen, Chengliang Zheng, Zhong-wang Fu, Umer Zukaib, Xiaohui Cui, and Zhidong Shen. Ocie: Augmenting model interpretability via deconfounded explanation-guided learning. *Knowledge-Based Systems*, 302, 2024. 2
- [16] Mustafa Ekrem Erakın, Uğur Demir, and Hazım Kemal Ekenel. On recognizing occluded faces in the wild. In *2021 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, 2021. 1, 6, 5
- [17] Shiwei Feng, Guan hong Tao, Siyuan Cheng, Guangyu Shen, Xiangzhe Xu, Yingqi Liu, Kaiyuan Zhang, Shiqing Ma, and Xiangyu Zhang. Detecting backdoors in pre-trained encoders. In *CVPR*, 2023. 2
- [18] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, 2019. 2
- [19] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2), 2022. 1
- [20] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7, 2019. 2, 5
- [21] Wenbo Guo, Lun Wang, Yan Xu, Xinyu Xing, Min Du, and Dawn Song. Towards inspecting and eliminating trojan backdoors in deep neural networks. In *2020 IEEE International Conference on Data Mining (ICDM)*, 2020. 2
- [22] Avani Gupta, Saurabh Saini, and P J Narayanan. Concept distillation: Leveraging human-centered explanations for model improvement. In *NeurIPS*, 2023. 1, 2, 3, 4, 7, 8
- [23] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: defending against backdoor attacks using robust statistics. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. 2
- [24] Can He, Mingfu Xue, Jian Wang, and Weiqiang Liu. Embedding backdoors as the facial features: Invisible backdoor attacks against face recognition systems. In *Proceedings of the ACM Turing Celebration Conference - China*. Association for Computing Machinery, 2020. 6
- [25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*. Springer, 2016. 5
- [26] Yingzhe He, Guozhu Meng, Kai Chen, Jinwen He, and Xingbo Hu. Deepoblivate: A powerful charm for erasing data residual memory in deep neural networks. *ArXiv*, abs/2105.06209, 2021. 3

- [27] Xiaoling Hu, Xiao Lin, Michael Cogswell, Yi Yao, Susmit Jha, and Chao Chen. Trigger hunting with a topological prior for trojan detection. *arXiv preprint arXiv:2110.08335*, 2021. [2](#)
- [28] Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Intrinsic certified robustness of bagging against data poisoning attacks. In *Proceedings of the AAAI conference on artificial intelligence*, 2021. [2](#)
- [29] Monish Keswani, Sriranjani Ramakrishnan, Nishant Reddy, and Vineeth N Balasubramanian. Proto2proto: Can you recognize the car, the way i do? In *CVPR*, 2022. [3](#)
- [30] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Jun Cai, James Wexler, Fernanda Viegas, and Rory Abbott Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *Proceedings of Machine Learning Research*, 2018. [2, 3](#)
- [31] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*. PMLR, 2020. [2](#)
- [32] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns. In *CVPR*, 2020. [2](#)
- [33] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. [6, 5](#)
- [34] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7), 2015. [6](#)
- [35] Changjiang Li, Ren Pang, Zhaohan Xi, Tianyu Du, Shouling Ji, Yuan Yao, and Ting Wang. An embarrassingly simple backdoor attack on self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. [2](#)
- [36] Yige Li, Nodens Koren, L. Lyu, Xixiang Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *ArXiv*, abs/2101.05930, 2021. [2, 3, 5, 4](#)
- [37] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *ICCV*, 2021. [2](#)
- [38] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *NeurIPS*, 2021. [2](#)
- [39] Yige Li, Xixiang Lyu, Xingjun Ma, Nodens Koren, Lingjuan Lyu, Bo Li, and Yu-Gang Jiang. Reconstructive neuron pruning for backdoor defense. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 2023. [2](#)
- [40] Yingqi Liu, Wen-Chuan Lee, Guan hong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2019. [2](#)
- [41] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *ECCV*, 2020. [2](#)
- [42] Yang Liu, Mingyuan Fan, Cen Chen, Ximeng Liu, Zhuo Ma, Li Wang, and Jianfeng Ma. Backdoor defense with machine unlearning. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022. [2, 3](#)
- [43] Rui Min, Zeyu Qin, Li Shen, and Minhao Cheng. Towards stable backdoor purification through feature shift tuning. *Advances in Neural Information Processing Systems*, 36, 2023. [2, 3, 4](#)
- [44] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33, 2020. [2, 5](#)
- [45] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021. [2, 5](#)
- [46] 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. *CoRR*, abs/2304.07193, 2023. [7](#)
- [47] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. [4](#)
- [48] Andrea Paudice, Luis Muñoz-González, and Emil C. Lupu. Label sanitization against label flipping poisoning attacks. In *ECML PKDD 2018 Workshops*. Springer International Publishing, 2019. [2](#)
- [49] Aniruddha Saha, Ajinkya Tejankar, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Backdoor attacks on self-supervised learning. In *CVPR*, 2022. [2](#)
- [50] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2022. [2](#)
- [51] Esha Sarkar, Hadjer Benkraouda, G Muthu Krishnan, Homer Gamil, and Michail Maniatakos. Facehack: Attacking facial recognition systems using malicious facial characteristics. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4, 2022. [2, 6](#)
- [52] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. [6](#)
- [53] Youngjae Song, Sung Kuk Shyn, and Kwang-su Kim. Img2tab: Automatic class relevant concept discovery from stylegan features for explainable image classification. *arXiv preprint arXiv:2301.06324*, 2023. [2](#)
- [54] Hossein Souri, Liam Fowl, Rama Chellappa, Micah Goldblum, and Tom Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. *Advances in Neural Information Processing Systems*, 35, 2022. [1](#)
- [55] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32, 2012. [6, 5](#)
- [56] Rahim Taheri, Reza Javidan, Mohammad Shojafar, Zahra Pooranian, Ali Miri, and Mauro Conti. On defending against

- label flipping attacks on malware detection systems. *Neural Computing and Applications*, 32, 2020. 2
- [57] Ruixiang Tang, Mengnan Du, Ninghao Liu, Fan Yang, and Xia Hu. An embarrassingly simple approach for trojan attack in deep neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, 2020. 4
- [58] Guanhong Tao, Guangyu Shen, Yingqi Liu, Shengwei An, Qiuling Xu, Shiqing Ma, and X. Zhang. Better trigger inversion optimization in backdoor scanning. In *CVPR*, 2022. 7, 1, 2, 6
- [59] Ajinkya Tejankar, Maziar Sanjabi, Qifan Wang, Sinong Wang, Hamed Firooz, Hamed Pirsiavash, and Liang Tan. Defending against patch-based backdoor attacks on self-supervised learning. In *CVPR*, 2023. 2
- [60] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Neural Information Processing Systems*, 2018. 2, 3
- [61] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3D distillation of self-supervised 2D image representations. In *Proceedings of the International Conference on 3D Vision (3DV)*, 2022. 5, 9
- [62] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9 (86), 2008. 8
- [63] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE Symposium on Security and Privacy (SP)*, 2019. 1, 2, 3, 7
- [64] Lun Wang, Zaynah Javed, Xian Wu, Wenbo Guo, Xinyu Xing, and Dawn Song. Backdoor!: Backdoor attack against competitive reinforcement learning. *arXiv preprint arXiv:2105.00579*, 2021. 2
- [65] Wenxiao Wang, Alexander J Levine, and Soheil Feizi. Improved certified defenses against data poisoning with (Deterministic) finite aggregation. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022. 2
- [66] Shaokui Wei, Jiayin Liu, and Hongyuan Zha. Backdoor Mitigation by Distance-Driven Detoxification. In *ICCV*, 2025. 2
- [67] E. Wenger, J. Passananti, A. Bhagoji, Y. Yao, H. Zheng, and B. Y. Zhao. Backdoor attacks against deep learning systems in the physical world. In *CVPR*, 2021. 2
- [68] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoor-bench: A comprehensive benchmark of backdoor learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 2, 4, 9
- [69] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2021. 2
- [70] Pengfei Xia, Hongjing Niu, Ziqiang Li, and Bin Li. Enhancing backdoor attacks with multi-level mmd regularization. *IEEE Transactions on Dependable and Secure Computing*, 2022. 6
- [71] Zhen Xiang, David J Miller, and George Kesidis. Detection of backdoors in trained classifiers without access to the training set. *IEEE Transactions on Neural Networks and Learning Systems*, 33(3), 2020. 2
- [72] Zhen Xiang, David J. Miller, and George Kesidis. Reverse engineering imperceptible backdoor attacks on deep neural networks for detection and training set cleansing. *Comput. Secur.*, 106(C), 2021. 2
- [73] Zhen Xiang, Zidi Xiong, and Bo Li. Cbd: A certified backdoor detector based on local dominant probability. *Advances in Neural Information Processing Systems*, 36, 2023. 2
- [74] 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)*. IEEE, 2021. 2
- [75] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y. Zhao. Latent backdoor attacks on deep neural networks. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, 2019. 2
- [76] Zhihao Yue, Jun Xia, Zhiwei Ling, Ming Hu, Ting Wang, Xian Wei, and Mingsong Chen. Model-contrastive learning for backdoor elimination. In *Proceedings of the 31st ACM International Conference on Multimedia*, 2023. 2, 3, 5, 6, 4
- [77] Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. 2021. 2, 3, 5
- [78] Yi Zeng, Zhouxing Shi, Ming Jin, Feiyang Kang, Lingjuan Lyu, Cho-Jui Hsieh, and Ruoxi Jia. Towards robustness certification against universal perturbations. In *International Conference on Learning Representation*. ICLR, 2023. 2
- [79] Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Data-free backdoor removal based on channel Lipschitzness. In *European Conference on Computer Vision*, 2022. 2, 4
- [80] Liuwan Zhu, Rui Ning, Cong Wang, Chunsheng Xin, and Hongyi Wu. Gangsweep: Sweep out neural backdoors by gan. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2020. 2
- [81] Mingli Zhu, Shaokui Wei, Li Shen, Yanbo Fan, and Baoyuan Wu. Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 3, 5, 6