Backdoor Attacks Against Patch-based Mixture of Experts

Cedric Chan[1], Jona te Lintelo[1], and Stjepan Picek[1]

Radboud University, Nijmegen, the Netherlands
{cedric.chan,jona.telintelo,stjepan.picek}@ru.nl

Abstract. As Deep Neural Networks (DNNs) continue to require larger amounts of data and computational power, Mixture of Experts (MoE) models have become a popular choice to reduce computational complexity. This popularity increases the importance of considering the security of MoE architectures. Unfortunately, the security of models using a MoE architecture has not yet gained much attention compared to other DNN models. In this work, we investigate the vulnerability of patch-based MoE (pMoE) models for image classification against backdoor attacks. We examine multiple trigger generation methods and Fine-Pruning as a defense. To better understand a pMoE model's vulnerability to backdoor attacks, we investigate which factors affect the model's patch selection. Our work shows that pMoE models are highly susceptible to backdoor attacks. More precisely, we achieve high attack success rates of up to 100% with visible triggers and a 2% poisoning rate, whilst only having a clean accuracy drop of 1.0%. Additionally, we show that pruning itself is ineffective as a defense but that fine-tuning can remove the backdoor almost completely. Our results show that fine-tuning the model for five epochs reduces the attack success rate to 2.1% whilst sacrificing 1.4% accuracy.

Keywords: Backdoor Attack · Mixture of Experts · Image Classification · Convolutional Neural Network · Data Poisoning

1 Introduction

Deep neural networks (DNNs) have become a popular choice for various tasks in domains such as computer vision [24, 40] or language processing [26]. As computational resources require an increase with the continuously growing size of models [18, 27, 28, 31], new model architectures have been proposed to improve model efficiency. Mixture of Experts (MoE) architectures have emerged as a powerful approach to improve model efficiency [1, 9, 32] by utilizing smaller, specialized models called experts, each trained to focus on a different task or subset of data.

The Mixture of Experts architecture was introduced by Jacobs et al. [12] in 1990. Their proposed model consists of expert models and a single gate that combines the expert outputs to generate the final model output. In this model, each

input sample is routed to all experts. Shazeer et al. [32] expanded on the idea of combining smaller expert models to reduce the computational cost of inference with large models. The authors achieve this by reducing the number of parameters used during inference. The authors train multiple smaller experts to focus on different tasks or subsets of data. In addition, a gating network is trained to route input to specific experts. As a result, input is processed using fewer parameters than when the entire model processes it, while maintaining performance. In addition to language models, vision models have also been shown to benefit from the MoE architecture. Chowdhury et al. [6] proposed a Patch-level Routing mechanism which routes patches of input images to experts for processing. Performing inference on patches of images rather than the entire image reduces sample complexity, computational cost, and requires fewer training samples to achieve performance comparable to a conventional CNN.

As MoE architectures gain traction in research and real-world applications [7, 35], the security considerations of these models [29,39] are becoming more important. Several works have already shown that non-MoE architectures are vulnerable to malicious attacks [2,10,25,29,36]. Gu et al. introduced the first backdoor attack, called BadNet [10]. The authors showed that image classification models could misclassify by adding a single pixel or a patch to images and changing the labels to a target label during training. Their attack achieved a high Attack Success Rate (ASR) without a large negative impact on the clean accuracy (CA). Other work [5] proposed blending, or overlaying, an image with the original image to create a trigger that achieves a very high ASR. Expanding on BadNet, multiple works have proposed using stealthy triggers to make the attack less detectable. Liao et al. [22] used a perturbation mask as the trigger that is almost imperceptible to humans when added to images. To create realistic yet stealthy triggers, Nguyen and Tran proposed using altered image characteristics with WaNet [25]. The authors slightly warp the image by shifting a few pixels to create the trigger. Their warped trigger obtained a high ASR in the tested setup and was unsusceptible to state-of-the-art defenses at that time.

However, currently, no prior work has investigated backdoor attacks against MoE architectures. A few works explore the security and privacy considerations of MoE architectures. Puigcerver et al. showed that MoE architectures enjoy better robustness than their dense counterparts [29]. Zhang et al. investigated the adversarial robustness of MoE-CNNs and proposed an adversarial training framework for MoE, AdvMoE, to increase the robustness of MoE architectures [39]. Yona et al. explored the security and privacy of MoE architectures by showing that an adversary can exploit an MoE, called Mixtral [13], and extract a victim's prompt [37]. A recent work by Zhang et al. [38] was one of the first to explore adversarial attacks against MoE architectures. The authors showed that Mixtral [13] can be caused to misclassify sentiment analysis via text instructions on multiple levels: word-level, syntax-level, and semantic-level.

In this work, we further explore the vulnerabilities of MoE architectures, specifically a Patch-based Mixture of Experts (pMoE) [6] model, by performing backdoor attacks with different types of triggers. To investigate possible miti-

gation strategies, we evaluate Fine-Pruning [23] as a defense against backdoors. Additionally, we analyze the pMoE model's patch selection to better understand how different triggers influence the router and to which trigger types the pMoE model is more vulnerable. By investigating why certain patches get routed to certain experts, we can examine which factors can be exploited with a trigger to influence the model's patch selection. This enables more powerful attacks and defenses in the future.[1]. Our main contributions are:

- To our knowledge, we are the first to investigate the security of MoE-based CNNs with backdoor attacks and show that pMoE-based CNNs are vulnerable to backdoor attacks. We achieve ASRs of up to 100% with visible triggers and a poisoning rate of 2%. With stealthy triggers, we achieve 93.8% ASR with a poisoning rate of 10%.
- We give insight into which image features are important to consider for backdoor attacks and defenses. We show that features such as shape, background, and target object are primary factors in patch selection and that pixel value, contrast intensity, and color have a smaller effect on patch selection.
- We investigate Fine-Pruning as a defense to mitigate backdoor attacks. We find that pruning neurons is not enough to mitigate the backdoor, with the attack success rate dropping by only ∼2%. However, with fine-tuning added, we were able to drop the attack success rate from ∼97% to ∼2%.

2 Preliminaries

2.1 Mixture of Experts

Mixture of Experts is a type of DNN architecture that uses the output of a single or several experts to produce the final model output. Each expert is a smaller neural network and trained to process only certain input through the use of a gating mechanism. The gating mechanism, also called a router, is a network or layer trained separately to determine which expert will process the input. The model's output is determined by the output of an expert or by combining the output of multiple experts. Where other DNN architectures use all model parameters, MoE architectures activate only a subset of parameters, which decreases the computational cost for inference. MoE architectures also offer improved scalability and efficiency, making them attractive in Large Language Models (LLMs) [1,33] and vision models [30]. For instance, DeepSeek leverages MoE to achieve remarkable efficiency and performance [7]. Despite having 671 billion parameters, DeepSeek activates only a small fraction (around 37 billion) for any given task.

2.2 Backdoor Attacks

Poisoning attacks assume that the adversary can access the training data of the ML model. Instead of modifying input at inference time, the adversary injects alterations during the training phase by poisoning the data or the model

[1] Our code is available at https://github.com/geefmegeld/pMoE-backdoor

weights [16]. Poisoning attacks attempt to degrade the model by poisoning the dataset so that the model misclassifies inputs. When poisoning the training data, the adversary can make triggers and add them to inputs so that the model associates the triggers with the chosen target label of the adversary. This is called a backdoor attack, which is a form of poisoning attack. In backdoor attacks, the adversary still attempts to misclassify the inputs, but only when a trigger is present in an input. While most works on backdoor attacks consider computer vision, backdoor attacks also work for other domains such as text [4, 8] and speech recognition [3, 14].

The adversary can choose the target label to misclassify samples when a trigger is present. However, the adversary can make the backdoor attack more specific by performing a source-specific attack. There, the adversary can not only choose the target label but also add a source label. On the other hand, when random samples are poisoned, all classes in the dataset could contain a trigger, which is commonly called a source-agnostic attack.

3 Methodology

3.1 Threat Model

Attacker Goal The attacker's objective is to cause the target model to misclassify images to the chosen target class at inference time. The backdoor should be stealthy and remain undetected for as long as possible, so the target model should still correctly classify images without the trigger to make the attack more stealthy. Additionally, the trigger patterns should not be easily noticeable, and the number of poisoned images should be low to keep the attack stealthy.

Attacker's Knowledge & Capabilities We assume a gray-box setting where the attacker has access to a part of the training data, for which they can alter the labels and the image itself. The attacker does not need to know the model's architecture or parameters and does not need access to the training procedure.

Real-world Example Our described threat model can be applied to a real-world scenario where training data is compromised and the adversary has gained access to a part of the dataset. For example, the user uses datasets from untrusted or unknown sources on the internet, obtained through web scraping [17], or datasets that have been formed by crowd-sourcing. In such cases, the datasets may contain malicious inputs that cause the model to behave unexpectedly.

3.2 Patch-level Routing in Mixture of Experts

In this work, we use the pMoE model designed by Chowdhury et al. [6] to evaluate backdoor attacks against MoE architectures. This architecture is a 3-layer Wide Residual Network (WRN) where the last layer of the WRN is replaced by a

patch-level MoE (pMoE) layer. A pMoE layer consists of an expert and a router, where each expert is a two-layer CNN with the same architecture. The router in the pMoE model directs patches of images, rather than entire images, to an expert. Each training image is divided into l patches of equal size. The routers use the Top-K algorithm to route k patches to each of the n experts. The Top-K patches with the highest activation values after a convolutional layer are routed. A high activation value indicates the patch is influential in the final output of the expert. In the final stage, the outputs from all experts are concatenated, and a softmax activation is applied at the end to produce the final output of the entire model.

Figure 1 illustrates an example pMoE model. In this example, images are divided into $l = 16$ patches. The routers use the Top-K algorithm to assign $k = 4$ patches to the $n = 2$ experts. The same patch can be routed to multiple experts. Each expert is trained, and their outputs are concatenated before passing through a dense layer for the final prediction.

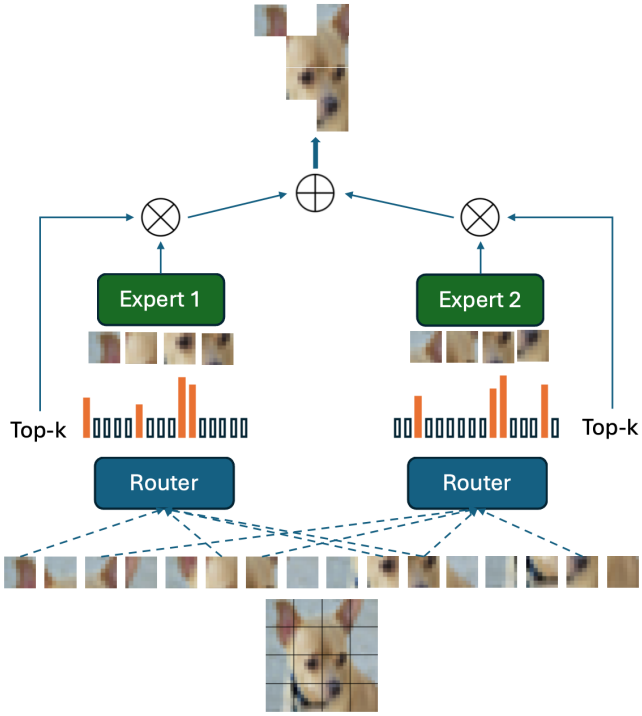


Fig. 1. An illustration of pMoE. The image is divided into 16 patches, and the router selects 4 patches for each expert.

3.3 Trigger Generation Methods

For our experiments, we selected three different triggers to perform backdoor attacks on the pMoE. The backdoor attacks we perform are dirty-label attacks, where we change the original labels to a target label. Furthermore, all attacks are performed in a source-agnostic way so that we can evaluate the effects of triggers on a wider range of image features and multiple classes.

Square Trigger The square trigger we use is based on the trigger used in Badnet [10]. The trigger is a small black square with a size of 4×4 pixels and placed in a fixed position in the top-left corner of the image. Our pMoE processes patches of the same 4×4 pixel size, enabling it to track which experts process this trigger with the evaluation method discussed in Section 3.5.

Blend Trigger The second trigger we consider is the blend trigger [5]. This trigger generation method involves blending the original image with a trigger image of the same size. In all our experiments, we blend a Hello Kitty image with the original images. The opacity of the trigger can be adjusted based on the blend ratio α. Blending an image can be formally denoted as:

$$x' = (1 - \alpha) \cdot x + \alpha \cdot i,$$

where x is our clean input image, i is the blend image, and α is the blend ratio. We set $\alpha = 0.5$ such that both the input image and the chosen blend image contribute equally to the final backdoored image, x'. This means the poisoned sample will be more distinguishable from clean samples, and we can analyze the effects of a visible trigger present in all patches of an image.

Warped Trigger In contrast to the previous two trigger methods, we also consider a stealthy backdoor attack using warping in images as a trigger. Warping is an image processing technique that uses geometric transformations to subtly deform images by shifting pixels to different locations. Since these slight pixel movements are hard to perceive, warped images remain visually similar to the originals. Unlike patch-based or blended triggers, which modify pixel colors, warping introduces subtle distortions while preserving the overall appearance, making it a stealthier trigger.

We generate the warping using the method proposed in Nguyen et al. [25]. A vector field is generated to guide the pixel movement in a predefined manner. This process requires two parameters: $k \in \{2, 4, 6, 8\}$, which determines the level of detail in the vector field, and $s \in \{0.25, 0.5, 0, 75, 1\}$, which controls the warping strength. Higher values of these parameters result in more pronounced warping, making poisoned images more distinguishable from clean samples. The warping in all our experiments is generated with $k = 2$ and $s = 0.25$ to create the least noticeable amount of warping. Unlike the black square trigger and blend trigger, we want to focus on the stealthiness of the warped trigger to analyze the effect of stealthy triggers on the pMoE model.

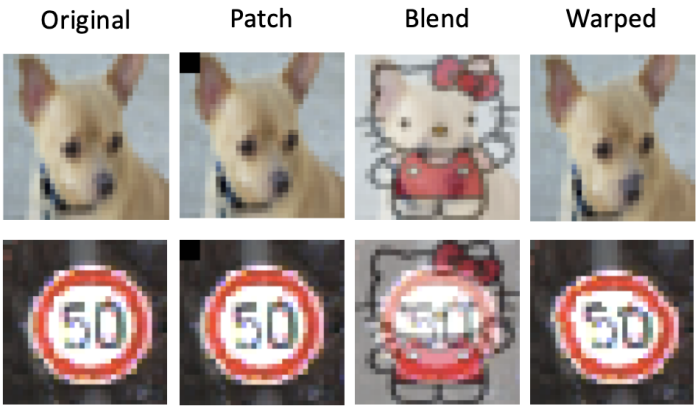


Fig. 2. Different trigger generation methods shown visually, where the first row is from CIFAR-10 and the 2nd row from GTSRB.

3.4 Evaluation

To evaluate the performance of the backdoor attacks, we utilize common metrics in the field of poisoning attacks on ML models:

Attack Success Rate (ASR): ASR signals how well the backdoored model predicts the modified test data with the trigger as the chosen target class of the attacker. The modified test data contains test images except the target class, where the trigger is applied to all test images. We calculate the ASR as:

$$\textbf{ASR} = \frac{\#\text{total predictions of the targeted class}}{\#\text{total predictions}}.$$

Benign Accuracy (BA): BA tells us how well the backdoor model is able to correctly predict the class of the test data without triggers. The BA is calculated as follows:

$$\textbf{BA} = \frac{\#\text{correct predictions}}{\#\text{total predictions}}.$$

Clean Accuracy Drop (CAD): The CAD indicates the difference in performance between a clean model and a backdoored model on a set of clean images and is represented by the accuracy drop between the two models. The CAD is computed as follows:

$$\textbf{CAD} = \text{Clean accuracy} - \text{Benign accuracy}.$$

3.5 Patch Selection Analysis

The pMoE architecture segments an image into patches, which are then assigned to different experts by routers. To better understand how pMoE distributes

these patches, we can visualize which patches are assigned to which expert. By looking at the output of the router in the pMoE layer, we can see which patch indices of an input image are processed by which expert. Using the indices, we create a patch map for every expert showing which patches were processed by that expert. Comparing the maps of different experts and the original image provides insights into the role of features such as shapes and contrast on patch selection. This enables us to better understand which factors are important to consider when choosing an effective type of trigger. Additionally, it allows for model explainability by identifying whether specific experts, or combinations of experts, contribute to misclassification, especially in backdoored models, where certain experts may be responsible for steering images with a trigger toward the target label.

In addition to visualizing the patch selection, we will also look at the pixel values of patches to investigate the influence of pixel value and color. We perform the patch selection analysis and patch value distribution analysis on a wide range of clean and poisoned images from CIFAR-10. We make sure to consider different classes, color distributions, target object size, and contrast where the background is distinct from or similar to the target object.

The pixel values in an image can influence how patches are routed to experts in the pMoE model. This is because the model utilizes the Top-K algorithm, which selects the k highest values. Since the Top-K algorithm selects the highest activation values after a convolutional layer, the pixel intensities within a patch could impact its selection, potentially affecting how patches are distributed among experts based on their pixel values.

Our current method of investigating the influence of color is by making a distribution of patch values for each expert. For each patch, we sum the pixel values, creating a list of patch values for a given image. Theoretically, the maximum patch value occurs when the patch is entirely white. This means that all RGB channels have their highest values, 255 per channel or 1 if normalized, multiplied by the number of pixels per patch, which is 16.

With the patch map we created, we can trace which patches are assigned to which expert and construct a distribution of patch values for each expert. If experts have distinct distributions, where some experts frequently select low-value (darker) patches while others predominantly choose high-value (lighter) patches, it may suggest that the patch assignment to experts is influenced by pixel intensity. This insight could mean that certain experts are biased toward processing patches based on pixel intensity. This information can be exploited by adversaries by developing a specific trigger, based on pixel values, that forces the model to assign poisoned patches to specific experts, leading to controlled misclassification.

3.6 Fine-Pruning Defense

In this work, we consider Fine-Pruning [23] as a defense because it is commonly used to mitigate the effects of backdoor attacks on non-MoE architectures [11,19, 21,23,25]. Fine-pruning aims to remove the backdoor from the affected model [23]

and is a combination of fine-tuning and pruning. The first step is pruning, where the parameters of a layer are set to 0. After pruning, the model is re-trained for a number of epochs, also referred to as fine-tuning.

4 Experiments

4.1 Data

We evaluate the backdoor attack on two datasets that are used in related work [6, 10, 22, 23, 25]: CIFAR-10 [15] and GTSRB [34]. The CIFAR-10 dataset consists of 50 000 training images and 10 000 test images with 10 classes and input dimensions of 32×32×3. The GTSRB dataset contains 39 209 training images and 12 630 test images of real-world photos with 43 classes and input dimensions that range from 15×15×3 to 250×250×3. Each class has balanced variations in lighting, rotation, background, and scale. We resize all images to 32×32×3 to be able to use the same model architecture and number of patches as CIFAR-10.

4.2 Baseline models

We consider two baseline models for our experiments to evaluate the effects of our backdoor attack: a clean pMoE model trained on CIFAR-10 and a clean pMoE model trained on GTSRB. For our experiments, we used the pMoE model as described in [6] without modifying the architecture. In all our experiments, we trained our pMoE model with $n = 4$ experts, each expert received $k = 16$ patches, and we divided the images into $l = 64$ patches. These hyperparameter settings were chosen following the results presented [6], which shows these values produce models with good accuracy. We trained and evaluated each baseline model 10 times and report the average result to ensure consistency and avoid outliers.

The clean and backdoored models were trained for 25 epochs using the SGD optimizer with a learning rate of 0.1, no decay schedule, and a batch size of 128. Although a learning rate of 0.1 may seem high, we followed the original pMoE implementation's parameter settings for training as these achieved good results.

Each experiment consists of a dataset, a trigger generation method, and a poisoning rate. We consider the square, blend, and warped trigger described in Section 3 with poisoning rate $\in \{0.1\%, 0.5\%, 2\%, 6\%, 10\%\}$. Each experiment is repeated five times, and we report the average to ensure consistent and reliable results. After each experiment, we apply the evaluation metrics described in Section 3.4 to analyze the results.

4.3 Fine-Pruning Defense

We evaluate Fine-Pruning as a defense on a backdoored pMoE model trained with CIFAR-10, a square trigger, and a 2% poisoning rate. Since our primary focus is not on defending pMoE models, we limited the defense experiments to

these specific parameters. We perform the defense with a pruning rate of 10%, 20%, 30% to evaluate how pruning affects the backdoored pMoE model and fine-tune for 5 epochs. Fine-tuning for more epochs could make the defense expensive and is less realistic in our threat model. To ensure reliable results, we run each experiment three times and report the average results.

4.4 Environment Setup

The pMoE implementation is written in Python 3.9.6 using TensorFlow 2.18.0, with Keras handling all model architectures. Our experiments were conducted using Google Colab, utilizing T4 GPUs with 12.7GB system RAM, 15GB GPU RAM, and 112GB disk space. Training times for a single model during our experiments varied from 30 minutes to 45 minutes, depending on the dataset, trigger generation method, and poisoning rate.

5 Experimental Results

5.1 Backdoor Attacks

Square Trigger In Tables 1 and 2, we show the results of the backdoor attacks on CIFAR-10 and GTSRB, respectively. We observe that the square trigger performs well on CIFAR-10 and GTSRB with a poisoning rate of 0.5% and higher and achieves an ASR of 90.7% up to 99.8%. However, with a very low poisoning rate of 0.1%, the square trigger does not achieve a good result with an ASR of 30.1% and 41.0% on CIFAR-10 and GTSRB, respectively.

We hypothesize two possible reasons for the low ASR when the poisoning rate is 0.1%. First, there are instances where the black square is not visible in the input images because some clean images naturally contain dark or black pixels in the top-left corner where the black square trigger is placed. Because of this feature overlap with the black square trigger, the model may struggle to learn the trigger pattern with a small number of poisoned samples. Second, the router selects the patches based on the highest activation values. It is possible that the router does not choose the patch containing the trigger to be evaluated by an expert. This can make it difficult for the model to learn the trigger pattern when there are only a small number of poisoned samples. This can be seen in Figure 4, where none of the 4 experts would be routed the patch with a black square trigger in the top left corner.

With a higher poisoning rate and thus more poisoned samples, the model is exposed more to the black square trigger, allowing it to learn to better distinguish naturally dark pixels in the top-left corner and the trigger. Additionally, with more poisoned samples during training, it is more likely that patches containing the trigger will be selected by a router. Although a higher poisoning rate increases the attack's effectiveness, we can also see that for most trigger generation methods and both datasets, the performance drop is larger when the poisoning rate is higher. This can be undesirable as it makes the attack less stealthy.

Table 1. Experimental results of backdoor attacks with different types of triggers on a pMoE model trained on CIFAR-10.

Poison rate	Square			Blend			Warped		
	ASR	**BA**	**CAD**	**ASR**	**BA**	**CAD**	**ASR**	**BA**	**CAD**
0.1	30.1	83.8	1.2	98.3	83.5	1.5	3.5	83.9	1.1
0.5	90.7	83.5	1.5	99.6	83.3	1.7	8.2	83.8	1.6
2.0	96.6	83.2	1.8	99.9	83.5	1.5	24.5	83.0	2.0
6.0	97.4	83.1	1.9	99.9	83.2	1.8	54.9	82.1	3.0
10.0	99.1	82.7	2.3	100	82.1	1.9	93.8	82.7	2.9

Table 2. Experimental results of backdoor attacks with different types of triggers on a pMoE model trained on GTSRB.

Poison rate	Square			Blend			Warped		
	ASR	**BA**	**CAD**	**ASR**	**BA**	**CAD**	**ASR**	**BA**	**CAD**
0.1	41.0	97.8	0.2	94.7	97.1	0.9	0.3	97.0	1.0
0.5	98.2	97.5	0.5	99.2	97.2	0.8	20.1	97.2	0.8
2.0	99.7	97.4	0.6	100	97.0	1.0	45.5	97.1	0.9
6.0	99.8	97.5	0.5	100	97.0	1.0	77.4	97.0	1.0
10.0	99.8	96.9	1.1	100	96.5	1.5	91.6	96.6	1.4

Blend Trigger In Tables 1 and 2, we observe that the blend trigger performs well with all poisoning rates and achieves an ASR of 94.7% up to 100%. Compared to the square trigger, the blend trigger achieves a better ASR when the poisoning rate is 0.1%, with 98.3% on CIFAR-10 and 94.7% on GTSRB. This difference in performance between the square trigger and blend trigger might be attributed to the blend trigger altering every part of the image rather than only the top-left corner. This means that a part of the trigger will always be present in the patches selected by a router. Additionally, the pixel values produced with the blend trigger are likely not naturally present in clean images, as with the black square trigger.

Warped Trigger From Tables 1 and 2, we see that the warped trigger performs worse than the square and blend triggers with an ASR of 0.3% to 93.8%. The warped trigger only achieves an ASR higher than 90% when the poisoning rate is 10%, compared to a poisoning rate of 0.5% and 0.1% for the square and blend trigger, respectively. This observation aligns with expectations based on previous work [20, 41] on stealthy triggers that show higher poisoning rates are needed to achieve an ASR comparable to non-stealthy triggers.

For the warped trigger and poisoning rates of 0.5% to 6.0%, we observe a consistently higher ASR for GTSRB compared to CIFAR-10. A possible reason for this observation is that warping effects are more noticeable in GTSRB images

than in CIFAR-10 images, as seen in Figure 2. As the warping effect visibly changes more parts of the GTSRB image, splitting the image into patches might be less disruptive to the trigger pattern than when the warping effect is weaker.

5.2 Patch Selection Analysis

Effects of Triggers on Patch Selection Figure 3 shows how the clean model routes clean images and how poisoned models route the patches for their respective triggers. In Figure 3, we observe that triggers clearly alter patch selection for all experts. In the case of the black square trigger, all experts get the top left patch containing the black square routed. However, for the blend and warped trigger, none of the experts got the top left patch routed. This shows that training the model on a poisoned data set directly affects patch selection to ensure affected pixels get processed by the model.

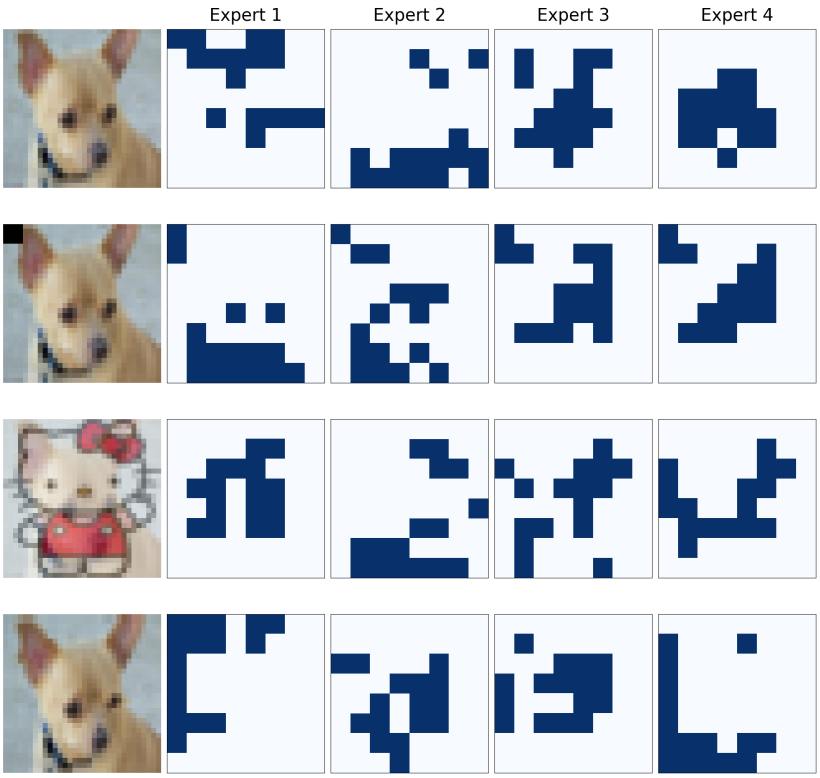


Fig. 3. Differences in patch selection for the same image in differently poisoned models, e.g., the first row shows how a clean model routes a clean image, and the second row shows how a square trigger poisoned model routes a poisoned image.

Clean Model Patch Selection Figure 4 displays which patches of an image are routed to which expert for the pMoE model trained for 25 epochs. The first two rows display a plane and a cat image from CIFAR-10, and the last two rows show images from the GTSRB dataset. In Figure 4, we see that each expert gets routed a different set of patches with varying degrees of overlap. Some experts may receive more patches containing pixels affected by the trigger than others. This means not every expert is equally influenced by triggers. For some trigger patterns present in only certain patches of the image, e.g., the black square trigger, it can happen that only one expert will process the trigger and still cause misclassification.

Generally, we observe that for a specific class, some experts learn to focus on parts of the background, while others learn to focus on the target object. This behavior can be seen in the example images for GTSRB. For both GTSRB example images, experts 1 and 2 focus on the central shape of the sign, while experts 3 and 4 often focus on the outline and edges of objects in images. Similar behavior is observed for some of the CIFAR-10 images. In the example image of the cat, we see that only expert 1 has a strong focus on the central object, i.e., the face of the cat. This behavior indicates that the shape of objects is an important factor in patch selection. Triggers that visibly affect the shape of objects in images will have a large influence on these experts.

Additionally, we observe in our experiments that the same expert can learn to focus on the background or target object depending on the class rather than having a fixed specialization for the location or shape of objects in images. In the examples of Figure 4, it can be seen that expert 4 learns to focus on large parts of the background for the cat, whilst in the plane image it focuses on the target object. Similar observations are made for other classes and images, indicating that another important factor in patch selection is whether the expert specializes in the target object or background for that class. This means that if a trigger only affects the target object, the trigger will only influence certain experts.

In Figure 5, we show the distributions of the average pixel value of the patches per expert. In the distributions for the cat image in Figure 5a, we observe that experts 1, 2, and 3 show peaks in frequency for certain average patch value bins. For example, expert 1 has a high frequency of patches with an average patch value of around 13. Additionally, there is a strong clustering of average values. For example, expert 1 has 13 out of 16 patches with an average patch value between 10 and 20. However, from Figures 5a and 5b we also see that the values for which there are peaks and clusters in the distributions of experts are random and differ per class and even per image. Moreover, in Figure 4 it can be seen that an expert does not only choose patches that have the same color, i.e., pixel values. This indicates that routers likely use pixel values to discern the shape of objects in images. This results in clusters around certain values that dominate an object in the image, but no strong bias for a patch selection pattern based on a certain color.

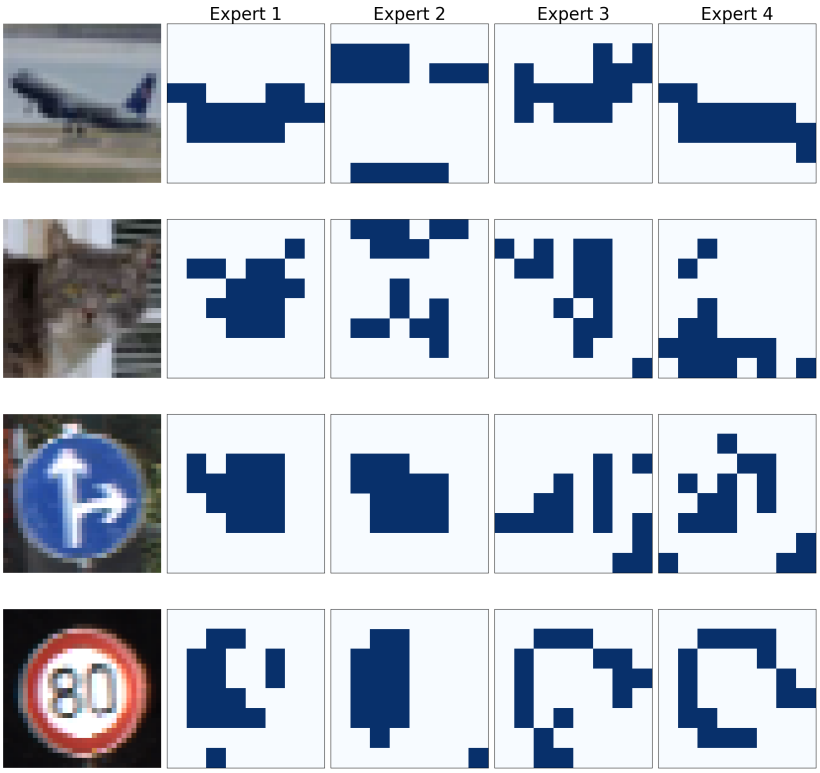
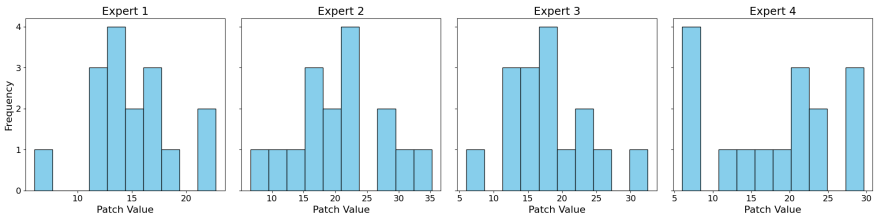


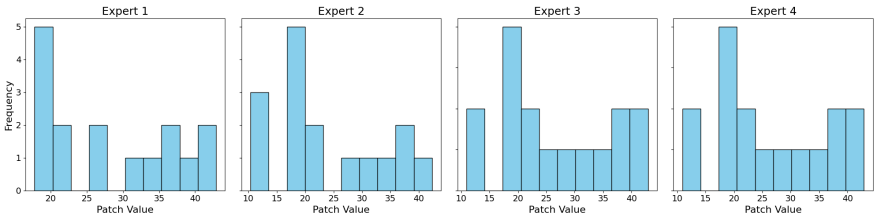
Fig. 4. Patch selection for each expert of the clean pMoE model. The first column shows the input images, and subsequent columns show which expert was assigned which patches. The first two rows show CIFAR-10 images, and the last two rows show GTSRB images.

5.3 Fine-Pruning Defense

In Table 3, we display the results of the Fine-Pruning defense performed on a backdoored pMoE model trained on CIFAR-10 with the square trigger and 2% poisoning rate. It can be seen that the Fine-Pruning defense reduces the ASR to 1.9% to 2.2%, depending on the pruning rate. We observe that different pruning rates result in nearly the same ASR reduction and that most ASR reduction occurs after fine-tuning. This contradicts expectations based on the Fine-Pruning defense proposed in [23], where pruning is intended to remove backdoor neurons, while fine-tuning is mainly used to restore the classification accuracy on clean samples. We hypothesize this behavior is because Fine-Pruning only prunes the last convolutional layer in the model. It is possible that the backdoor is present in a different layer or multiple layers of the pMoE model. In this case, pruning would have no effect on the backdoor because we prune only the last convolutional layer.



(a) Cat



(b) Blue sign

Fig. 5. Patch value distributions of images from Figure 4. On the x-axis are the average pixel values of the patches that were routed to that expert. On the y-axis are the frequencies of occurrence for the average pixel values.

In addition, the CA drops substantially when the pruning rate increases. With a 30% pruning rate, the clean accuracy dropped from 85.2% to 72.2% after pruning, and from 84.9% to 80.1% with a 10% pruning rate. The failure to remove the backdoor and the reduction in performance show that pruning itself is not an effective defense and that fine-tuning, or retraining, the model removes the backdoor.

Table 3. Results of the Fine-Pruning defense on pMoE on the CIFAR-10 dataset with a 2% poisoning rate averaged over three runs.

Pruning rate	**10%**		**20%**		**30%**	
	CA	**ASR**	**CA**	**ASR**	**CA**	**ASR**
Before defense	84.9	96.5	84.8	96.3	85.2	97.4
After pruning	80.1	94.2	78.9	95.2	72.2	95.3
After fine-tuning	83.5	2.1	83.6	1.9	83.5	2.2

Figure 6 shows the effect of Fine-Pruning on patch selection. Before Fine-Pruning, three out of four experts would process the black square trigger patch, indicating it to be an influential patch for classification. However, after Fine-Pruning, there is only one out of four experts who process the black square patch, meaning the trigger patch has much less influence on the classification.

This change in routing provides a visual explanation for the backdoor attack's reduced effectiveness after Fine-Pruning.

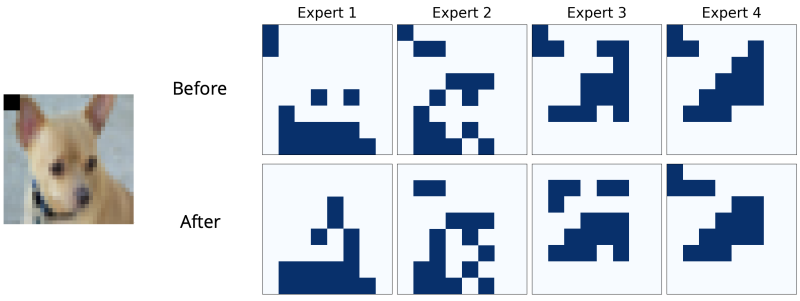


Fig. 6. CIFAR-10 image of a dog poisoned with the square trigger and the corresponding patch selection for the four experts. The top row shows the patch selection before Fine-Pruning, and the bottom row shows the patch selection after Fine-Pruning.

6 Conclusions and Future Work

In this paper, we investigate the vulnerability of MoE-based CNNs like pMoE against backdoor attacks. We have demonstrated that pMoE architectures are susceptible to backdoor attacks in different settings by using three different trigger generation methods. Even with low poisoning rates, an adversary can achieve a high ASR by utilizing a visible trigger. Additionally, an adversary can successfully attack a pMoE model with a stealthy trigger by increasing the poisoning rate. For both visible and stealthy triggers, the CAD increases only marginally with an increased poisoning rate, indicating that a higher poisoning rate does not affect the performance of the model significantly. This makes MoE architectures more vulnerable to backdoor attacks. In addition to evaluating backdoor attacks, we investigated Fine-Pruning as a defense against backdoor attacks for pMoE models. We found that Fine-Pruning could remove the backdoor successfully and reduce ASR from $\sim$97% to $\sim$2%, whilst sacrificing 1.4% accuracy. However, the majority of the backdoor removal happens in the fine-tuning stage. The pruning stage of the defense does little to reduce ASR, but has a large negative impact on the CA. Future works include analyzing different MoE architectures, such as V-MoE, and datasets with different properties, such as larger images. Additionally, as this work only considers MoE in the vision domain, an interesting direction is investigating the vulnerabilities of MoE architectures in the text domain against backdoor attacks or other adversarial attacks.

References

1. Artetxe, M., Bhosale, S., Goyal, N., Mihaylov, T., Ott, M., Shleifer, S., Lin, X.V., Du, J., Iyer, S., et al., R.P.: Efficient large scale language modeling with mixtures of experts. arXiv preprint **arXiv:2112.10684** (2021), https://arxiv.org/abs/2112.10684
2. Bai, J., Gao, K., Gong, D., Xia, S.T., Li, Z., Liu, W.: Hardly perceptible trojan attack against neural networks with bit flips. arXiv preprint **arXiv:2207.13417** (2022), https://arxiv.org/abs/2207.13417
3. Cai, H., Zhang, P., Dong, H., Xiao, Y., Koffas, S., Li, Y.: Towards stealthy backdoor attacks against speech recognition via elements of sound. arXiv preprint **arXiv:2307.08208** (2023), https://arxiv.org/abs/2307.08208
4. Chen, X., Salem, A., Chen, D., Backes, M., Ma, S., Shen, Q., Wu, Z., Zhang, Y.: Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In: Annual Computer Security Applications Conference. p. 554–569. ACSAC '21, ACM (Dec 2021). https://doi.org/10.1145/3485832.3485837, http://dx.doi.org/10.1145/3485832.3485837
5. Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning. arXiv preprint **arXiv:1712.05526** (2017), http://arxiv.org/abs/1712.05526
6. Chowdhury, M.N.R., Zhang, S., Wang, M., Liu, S., Chen, P.Y.: Patch-level routing in mixture-of-experts is provably sample-efficient for convolutional neural networks. arXiv preprint **arXiv:2306.04073** (2023), https://arxiv.org/abs/2306.04073
7. DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., et al., D.J.: Deepseek-v3 technical report. arXiv preprint **arXiv:2412.19437** (2025), https://arxiv.org/abs/2412.19437
8. Du, W., Yuan, T., Zhao, H., Liu, G.: Nws: Natural textual backdoor attacks via word substitution. In: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4680–4684 (2024). https://doi.org/10.1109/ICASSP48485.2024.10447968
9. Fedus, W., Zoph, B., Shazeer, N.: Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. arXiv preprint **arXiv:2101.03961** (2021), https://arxiv.org/abs/2101.03961
10. Gu, T., Dolan-Gavitt, B., Garg, S.: Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint **arXiv:1805.12185** (2017), http://arxiv.org/abs/1708.06733
11. Hong, S., Carlini, N., Kurakin, A.: Handcrafted backdoors in deep neural networks. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 8068–8080. Curran Associates, Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/3538a22cd3ceb8f009cc62b9e535c29f-Paper-Conference.pdf
12. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation **3**(1), 79–87 (1991). https://doi.org/10.1162/neco.1991.3.1.79
13. Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D.S., de las Casas, D., Hanna, E.B., et al., F.B.: Mixtral of experts. arXiv preprint **arXiv:2401.04088** (2024), http://arxiv.org/abs/2401.04088
14. Koffas, S., Xu, J., Conti, M., Picek, S.: Can you hear it? backdoor attacks via ultrasonic triggers. In: Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning. p. 57–62. WiseML '22, Association for Computing

Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3522783.3529523, https://doi.org/10.1145/3522783.3529523
15. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. Rep. 0, University of Toronto, Toronto, Ontario (2009), https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
16. Kurita, K., Michel, P., Neubig, G.: Weight poisoning attacks on pre-trained models. arXiv preprint **arXiv:2004.06660** (2020), https://arxiv.org/abs/2004.06660
17. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J.R.R., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T., Ferrari, V.: The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale. arXiv preprint **arXiv:1811.00982** (2018), http://arxiv.org/abs/1811.00982
18. Larochelle, H., Bengio, Y., Louradour, J., Lamblin, P.: Exploring strategies for training deep neural networks. Journal of Machine Learning Research **10**(1), 1–40 (2009), http://jmlr.org/papers/v10/larochelle09a.html
19. Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., Ma, X.: Anti-backdoor learning: Training clean models on poisoned data. Advances in Neural Information Processing Systems **34**, 14900–14912 (2021)
20. Li, Y., Li, Y., Wu, B., Li, L., He, R., Lyu, S.: Backdoor attack with sample-specific triggers. arXiv preprint **arXiv:2012.03816** (2020), https://arxiv.org/abs/2012.03816
21. Li, Y., Li, Y., Wu, B., Li, L., He, R., Lyu, S.: Invisible backdoor attack with sample-specific triggers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 16463–16472 (2021)
22. Liao, C., Zhong, H., Squicciarini, A.C., Zhu, S., Miller, D.J.: Backdoor embedding in convolutional neural network models via invisible perturbation. arXiv preprint **arXiv:1808.10307** (2018), http://arxiv.org/abs/1808.10307
23. Liu, K., Dolan-Gavitt, B., Garg, S.: Fine-pruning: Defending against backdooring attacks on deep neural networks. arXiv preprint **arXiv:1805.12185** (2018), http://arxiv.org/abs/1805.12185
24. Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D.: Image segmentation using deep learning: A survey. arXiv preprint **arXiv:2001.05566** (2020), https://arxiv.org/abs/2001.05566
25. Nguyen, T.A., Tran, A.T.: Wanet - imperceptible warping-based backdoor attack. arXiv preprint **arXiv:2102.10369** (2021), https://arxiv.org/abs/2102.10369
26. Otter, D.W., Medina, J.R., Kalita, J.K.: A survey of the usages of deep learning for natural language processing. IEEE Transactions on Neural Networks and Learning Systems **32**(2), 604–624 (2021). https://doi.org/10.1109/TNNLS.2020.2979670
27. Patterson, D., Gilbert, J.M., Gruteser, M., Robles, E., Sekar, K., Wei, Y., Zhu, T.: Energy and emissions of machine learning on smartphones vs. the cloud. Commun. ACM **67**(2), 86–97 (jan 2024). https://doi.org/10.1145/3624719, https://doi.org/10.1145/3624719
28. Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.M., Rothchild, D., So, D., Texier, M., Dean, J.: Carbon emissions and large neural network training (2021)
29. Puigcerver, J., Jenatton, R., Riquelme, C., Awasthi, P., Bhojanapalli, S.: On the adversarial robustness of mixture of experts. arXiv preprint **arXiv:2210.10253** (2022), https://arxiv.org/abs/2210.10253
30. Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Pinto, A.S., Keysers, D., Houlsby, N.: Scaling vision with sparse mixture of experts. arXiv preprint **arXiv:2106.05974** (2021), https://arxiv.org/abs/2106.05974

31. Samsi, S., Zhao, D., McDonald, J., Li, B., Michaleas, A., Jones, M., Bergeron, W., Kepner, J., Tiwari, D., Gadepally, V.: From words to watts: Benchmarking the energy costs of large language model inference (2023)
32. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q.V., Hinton, G.E., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint **arXiv:1701.06538** (2017), http://arxiv.org/abs/1701.06538
33. Shen, S., Hou, L., Zhou, Y., Du, N., Longpre, S., Wei, J., Chung, H.W., Zoph, B., et al., W.F.: Mixture-of-experts meets instruction tuning:a winning combination for large language models. arXiv preprint **arXiv:2305.14705** (2023), https://arxiv.org/abs/2305.14705
34. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: The german traffic sign recognition benchmark: A multi-class classification competition. In: The 2011 International Joint Conference on Neural Networks. pp. 1453–1460 (2011). https://doi.org/10.1109/IJCNN.2011.6033395
35. Xue, F., Zheng, Z., Fu, Y., Ni, J., Zheng, Z., Zhou, W., You, Y.: Openmoe: An early effort on open mixture-of-experts language models. arXiv preprint **arXiv:2402.01739** (2024), https://arxiv.org/abs/2402.01739
36. Xue, M., Ni, S., Wu, Y., Zhang, Y., Wang, J., Liu, W.: Imperceptible and multi-channel backdoor attack against deep neural networks. arXiv preprint **arXiv:2201.13164** (2022), https://arxiv.org/abs/2201.13164
37. Yona, I., Shumailov, I., Hayes, J., Carlini, N.: Stealing user prompts from mixture of experts (2024), https://arxiv.org/abs/2410.22884
38. Zhang, R., Li, H., Wen, R., Jiang, W., Zhang, Y., Backes, M., Shen, Y., Zhang, Y.: Instruction backdoor attacks against customized llms. arXiv preprint **arXiv:2402.09179** (2024), https://arxiv.org/abs/2402.09179
39. Zhang, Y., Cai, R., Chen, T., Zhang, G., Zhang, H., Chen, P.Y., Chang, S., Wang, Z., Liu, S.: Robust mixture-of-expert training for convolutional neural networks. arXiv preprint **arXiv:2308.10110** (2023), https://arxiv.org/abs/2308.10110
40. Zhao, Z., Zheng, P., Xu, S., Wu, X.: Object detection with deep learning: A review. arXiv preprint **arXiv:1807.05511** (2018), http://arxiv.org/abs/1807.05511
41. Zhong, N., Qian, Z., Zhang, X.: Imperceptible backdoor attack: From input space to feature representation. arXiv preprint **arXiv:2205.03190** (2022), https://arxiv.org/abs/2205.03190