Defending Backdoor Attacks on Vision Transformer via Patch Processing

Khoa D. Doan¹, Yingjie Lao², Peng Yang³, Ping Li⁴

¹College of Engineering and Computer Science, VinUniversity
²Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA
³Meta Corporation, Bellevue, WA 98004, USA
⁴LinkedIn Corporation, Bellevue, WA 98004, USA
khoa.dd@vinuni.edu.vn, ylao@clemson.edu, pengyang01@gmail.com, pinli@linkedin.com

Abstract
Vision Transformers (ViTs) have a radically different architecture with significantly less inductive bias than Convolutional Neural Networks. Along with the improvement in performance, security and robustness of ViTs are also of great importance to study. In contrast to many recent works that exploit the robustness of ViTs against adversarial examples, this paper investigates a representative causative attack, i.e., backdoor. We first examine the vulnerability of ViTs against various backdoor attacks and find that ViTs are also quite vulnerable to existing attacks. However, we observe that the clean-data accuracy and backdoor attack success rate of ViTs respond distinctively to patch transformations before the positional encoding. Then, based on this finding, we propose an effective method for ViTs to defend both patch-based and blending-based trigger backdoor attacks via patch processing. The performances are evaluated on several benchmark datasets, including CIFAR10, GTSRB, and TinyImageNet, which show the proposed defense is very successful in mitigating backdoor attacks for ViTs. To the best of our knowledge, this paper presents the first defensive strategy that utilizes a unique characteristic of ViTs against backdoor attacks.

1 Introduction
The versatility of machine learning makes it a promising technology for implementing a wide variety of complex systems such as autonomous driving (Grigorescu et al. 2020; Caesar et al. 2020), intrusion detection (Vinayakumar et al. 2019; Berman et al. 2019), communication (Huang et al. 2020; Gu et al. 2019; Liu et al. 2020), and pandemic mitigation (Oh, Park, and Ye 2020; Alimadadi et al. 2020) systems, retrieval (Doan, Yang, and Li 2022), etc. These examples also illustrate that a large portion of safety-critical applications is benefited from the evolution of machine learning, which meanwhile requires high degrees of security and trustworthiness of these technologies (Yang, Lao, and Li 2021; Lao et al. 2022a,b; Zhao, Lao, and Li 2022; Zhao and Lao 2022). Unfortunately, vulnerabilities have emerged from many aspects of machine learning and a wide body of research has been investigated recently to exploit both these vulnerabilities and defensive measures to mitigate attacks against machine learning, especially for deep learning systems (Szegedy et al. 2014; Liu et al. 2018a; Akhtar and Mian 2018).

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
against backdoor attacks and the corresponding countermea-
sures have not been extensively studied. In fact, to the best
of our knowledge, only one very recent work looked at this
direction (Lv et al. 2021), which proposed a data-free back-
door embedding attack against the vision transformer net-
works. In contrast to this prior work, we focus on the de-
fen:sive side. Aligning with the processing of ViT that divides
an image into patches, we mainly study the implications of
patch transformations on image classification tasks in this
paper. Specifically, we utilize two techniques, namely Patch-
Drop and PatchShuffle, which randomly drop and shuffle
patches of an image, respectively. Under these patch pro-
cessing, we find that ViT exhibits a different characteristic
from CNNs and also responds distinctively between clean
samples and backdoor samples. Specifically, PatchDrop is
effective in detecting patch-based backdoor attacks, while
PatchShuffle can successfully mitigate blending-based back-
door attacks. Therefore, based on patch processing, we pro-
pose a novel defensive solution to combat backdoor attacks.
The contributions of this paper are summarized below:

• We first perform an empirical study on the vulnerability of
ViTs against both patch-based and blending-based back-
door attacks and find ViTs are still quite vulnerable to
backdoor attacks.

• We observe an interesting characteristic of ViTs that
clean-data accuracy and backdoor attack success rate of
ViTs respond distinctively to patch processing before the
positional encoding, which is not seen on CNN models.

• We propose a novel defensive solution to mitigate back-
door attacks on ViTs via patch processing. We analyze
two processing methods, i.e., PatchDrop and PatchShuffle,
and examine their effectiveness in reducing the attack suc-
cess rate (ASR) of backdoor attacks. In particular, Patch-
Drop and PatchShuffle are effective in detecting patch-
based and blending-based backdoor attacks, respectively.
Together, they are used to effectively detect the backdoor
samples without prior knowledge of whether the attack is
patch-based or blending-based.

• We comprehensively evaluate the performance of the pro-
sposed techniques on a wide range of benchmark settings,
including CIFAR10, GTSRB, and TinyImageNet.

2 Related Work
Previous works on deep neural network (DNN) backdoor
injection have understood the attack as the process of in-
troducing malicious modifications to a model, \( F(\cdot) \), trained
to classify the dataset \((X,Y)\). These changes force an asso-
ciation with specific input triggers, \((\Delta, m)\), to the desired
model output, \(y_t\) (Gu et al. 2019; Liu et al. 2018b; Bag-
dasaryan and Shmatikov 2021; Yao et al. 2019). Through
Equation (1), the trigger can be superimposed on any input
such that a poisoned input is formed.

\[
P(x, m, \Delta) = x \circ (1 - m) + \Delta \circ m \quad (1)
\]

Here we use \(\circ\) to denote the element-wise product and \(m\)
is a mask used to determine the region of the input containing
the trigger pattern, \(\Delta\). In essence, the adversarial goal is to
force the model to minimize the compound loss function:

\[
\mathcal{M}(F_m(x), y) + c \cdot D(F(P(x, m, \Delta)), F(x_t)), \text{ instead of the}
\]

original benign loss such as cross-entropy loss, where \(D(\cdot, \cdot)\)
defines the similarity between the model’s actual behavior
and a target behavior described by the input \(x_t\) while the
constant \(c\) is used to balance the terms (Yao et al. 2019).

The main methodologies used to inject this functionality
into the model are contaminating the training data (Chen
et al. 2017; Liu et al. 2018b; Gu et al. 2019; Saha, Sub-
ramanya, and Pirsiavash 2020), altering the training algo-
rithm (Bagdasaryan and Shmatikov 2021) or overwriting/re-
training the model parameters after deployment (Dumford
and Scheirer 2020). Besides the original patch-based trig-
ger (Gu et al. 2019), various blending-based trigger pat-
terns have also been proposed, including blended (Chen
et al. 2017), sinusoidal strips (SIG) (Barni, Kallas, and Tondi
2019), reflection (ReFool) (Liu et al. 2020), and warping
(WaNet) (Nguyen and Tran 2021). Note that in order to dif-
ferentiate from the patch used in describing the processing
of ViTs, we limit the usage of patch for backdoor attacks to
only “patch-based”. In other words, only “patch-based”
refers to the backdoor attack, while all the other usages
of “patch” are related to the ViTs in this paper. For the
backdoor embedding attack on ViT (Lv et al. 2021), it seeks
to catch most attention of the victim model by leveraging the
unique attention mechanism.

On the other hand, several categories of defensive solu-
tions have been proposed to combat backdoor attacks in past
years (Chen et al. 2019a; Tran, Li, and Madry 2018; Gao
et al. 2019; Liu, Xie, and Srivastava 2017; Li et al. 2020; Liu,
Dolan-Gavitt, and Garg 2018; Cheng et al. 2020; Wang et al.
2019; Chen et al. 2019b; Qiao, Yang, and Li 2019). One di-
rection is to remove, detect, or mismatch the trigger of inputs
through certain processing or transformations of the input
images (Liu, Xie, and Srivastava 2017; Li et al. 2020; Doan,
Abbasnejad, and Ranasinghe 2020; Udeshi et al. 2022; Qiu
et al. 2021; Gao et al. 2019). Note that most of these defensive
methods are model-agnostic and mainly target at pro-
cessing the inputs. Our proposed defensive method follows
a similar concept as these input processing methods. For in-
stance, similar to STRIP (Gao et al. 2019) that examines the
entropy in predicted classes after a set of input perturbations
to check any violation of the input-dependence property of
a benign model, we leverage the distinctive performance
between the clean sample and backdoor sample against patch
processing to detect malicious behaviors. Another advantage
of such methods, including the proposed one, is that they
only require access to clean samples, which is a more prac-
tical setting for defending backdoor attacks.

3 Backdoor Attacks on ViT
3.1 Threat Model
We follow the typical threat model of DNN backdoor at-
tacks (Gu et al. 2019) that a user wishes to establish a model
for a specific image classification task by training with data
provided by a third party. We assume the adversary has the
capability of injecting poisoned data samples into the train-
ing dataset, but cannot modify the model architecture, the
training setting, or the inference pipeline. Since the user will
check the accuracy of the trained model on a held-out validation dataset (clean samples), the adversarial goal is to embed a backdoor into the model through data contamination without degrading the clean-data accuracy over the image classification task. In other words, the model should produce malicious behavior only on images with the trigger for the backdoor, while performing normally otherwise.

### 3.2 Attack Experimental Results

To understand the security threat on ViTs against the backdoor attacks, we consider two most popular approaches of creating the backdoor triggers: local patch-based triggers, BadNets (Gu et al. 2019) and SinglePixel (Bagdasaryan and Shmatikov 2021), and global blending-based triggers, ReFool (Liu et al. 2020) and WaNet (Nguyen and Tran 2021). We evaluate the performance on CIFAR10, GTSRB, and TinyImageNet datasets.

Specifically, we perform the attack experiment by poisoning the training dataset and the corresponding ground-truth labels. For each training dataset, similar to prior works (Gu et al. 2019; Nguyen and Tran 2021; Liu et al. 2020), we select a small number of samples (less than 10%) and apply the corresponding trigger on each of the selected images. Figure 1 shows some examples of both patch-based and blending-based backdoor samples. The labels of the poisoned samples are also changed to the target label. The poisoned training data are then used to train the image classification model. Then, we perform training using two ViT variants, the original ViT (Dosovitskiy et al. 2021) and DeiT (Touvron et al. 2021), and several other popular CNN model architectures, including Vgg11 (Simonyan and Zisserman 2014), ResNet18 (He et al. 2016), and Big Transfer (BiT) (Kolesnikov et al. 2020). Note that the models are pretrained on ImageNet-21k and fine-tuned on the corresponding dataset to ensure a consistent experimentation framework. This setup is influenced by the fact that large-scale ViTs and BiT are not trained from scratch on smaller-scale datasets to prevent overfitting. Each trained model is then evaluated on the held-out test sets of clean and backdoor samples. The backdoor samples are applied with the triggers that are generated using the same mechanism in the corresponding attack strategy for the evaluation.

In Tables 1 and 2, we show the clean-data and backdoor-data performance of the trained models for BadNets and WaNet, respectively. We can observe that the trained ViT and DeiT with the backdoors have similar, high clean-data accuracies to that of the corresponding benign models (still outperforming other CNN models). However, when the triggers are present, the probabilities of the poisoned ViT models to predict the target label (i.e., ASR) are also quite high, which are above 96% on all datasets. In other words, ViTs are at least as vulnerable against backdoor attacks as the CNN models. In fact, the patch-based backdoor attack on ViTs seems to be even slightly more successful than on other CNN models, which further validates the need for studying the backdoor attacks and countermeasures on ViTs. We observe similar results for SinglePixel and ReFool attacks.

### 4 Backdoor Attacks vs. Patch Processing

We have shown that backdoor attacks are still quite successful on ViTs. Besides, as we discussed above, it has also recently been shown that ViTs are vulnerable against other types of attacks, although they exhibit certain degrees of improvement in robustness against the transferability of adversarial examples (Mahmood, Mahmood, and van Dijk 2021; Shao et al. 2021). While these features of ViTs are similar to the CNNs, ViTs have also been shown to be more robust toward occlusions, distributional shifts, and permutation (Naseer et al. 2021). Here, we extend the robustness study of the receptive fields of ViTs with respect to the backdoor attack models and compare their performance to CNNs.

#### 4.1 Patch Processing

Following the existing defensive methods that process images at the input space for detecting backdoor attacks (Liu, Xie, and Srivastava 2017; Li et al. 2020; Doan, Abbasnejad, and Ranasinghe 2020; Udeghi et al. 2022; Qiu et al. 2021; Gao et al. 2019), we study the performance of the backdoor attacks on ViT models through input transformations that align with the characteristic of ViTs, i.e., patch processing where the content of the image is randomly perturbed. Specifically, each input image $x$ is represented as a sequence of patches with $L$ elements: $\{x_i\}_{i=1,..,L}$. Note that the patch $x_i$ does not necessarily have the same size as the patch size used in the pre-trained ViT model. Perturbing the image’s patches is equivalent to modifying its content. Here, we focus on the question: How does perturbation influence the receptive field of ViTs on image patches when various backdoor triggers are present? We denote the patch processing on $x$ with a function $R$ and consider the following strategies for performing the patch processing:
• **PatchDrop.** Similar to Naseer et al. (2021), we randomly drop $M$ patches from the total $L$ patches of an image $x$. We divide the image into $L = l \times l$ patches that belong to a spatial grid of $l \times l$. The number of dropped patches indicates the information loss on the image content.

• **PatchShuffle.** We randomly shuffle the $L$ patches of an image $x$. The $L$ patches are created in similar spatial grids as those of PatchDrop. PatchShuffle does not remove the content of the image but can significantly impact the receptive fields of the models.

Note that similar forms of patch transformations on ViTs have been considered in prior works (Naseer et al. 2021; Shao et al. 2021), but not in the context of backdoor attacks.

### 4.2 Performance of Backdoor Attacks against Patch Processing

We first study the trends of backdoor ASR and clean-data accuracy with respect to patch processing on the corresponding test set for each dataset. The results are reported in Figures 2 and 3 for BadNets and ReFool, respectively.

For patch-based attacks with PatchDrop, we observe that the clean-data performances of ViT only drop slightly on CIFAR10 and GTSRB even when almost 50% of the image content is removed. In contrast, the clean-data performances drop much more significantly in all the other three CNNs. On TinyImageNet, the clean-data performance of ViT drops more than in the other datasets. However, when the backdoor triggers are present, the attack success rate on ViT decreases significantly, even with a slight loss in the content of the images. In comparison, backdoor attacks on CNNs are more robust to PatchDrop. Interestingly, if we continue to drop more patches, the ASR on the CNNs suddenly increases in several experiments. A possible explanation is that CNN models and backdoor attacks rely on smaller regions of the image than ViT for prediction and achieving the target classes, respectively, which makes the clean-data accuracy of ViT more robust to patch processing. We also notice another important result: the variance in the predictions of the poisoned models is higher for backdoor samples than for the clean samples. We summarize the observations for patch-based attacks with respect to PatchDrop as follows:

- Clean-data accuracy sensitivity: ViT < CNN
- ASR sensitivity: ViT > CNN
- Gap between accuracy and ASR: ViT > CNN

However, for blending-based attacks with PatchDrop, we do not observe a consistent difference between the ViTs and CNNs, although ViTs are more robust with respect to the clean-data accuracy and ASR. Since the trigger is well-blended into the images across the entire pixel space, as in ReFool and WaNet, PatchDrop tends to be less impactful on the backdoor, similar to the robustness of the models on the foreground objects. However, for blending-based attacks with PatchShuffle, we observe that the clean-data performances of ViT drop significantly. In contrast, the ASRs only drop slightly. Such robustness of the trigger is consistent across various patch sizes (i.e., $|x_i|$) on all datasets. For the CNNs, the gaps between clean-data accuracy and ASR are smaller; in some cases, e.g., Vgg11, the gap can become significantly narrow. In previous studies, ViTs exhibit high robustness against patch transformation for larger patch sizes.
sizes (Naseer et al. 2021). Under the proposed PatchShuffle attack, the significant robustness of the trigger across all patch sizes, especially the smaller sizes, is interesting. Such performance can possibly be explained that ViTs learn and generalize the spatial invariance of the triggers extremely well. We summarize the observations for blending-based attacks with respect to PatchShuffle as follows:

- **Clean-data accuracy sensitivity:** ViT > CNN
- **ASR sensitivity:** ViT << CNN
- **Gap between accuracy and ASR:** ViT > CNN

In summary, ViT has distinguishable performance between clean-data performance and ASR against certain patch processing techniques: the ASR drops significantly on ViT for **patch-based attacks with PatchDrop**, while the clean-data performance drops significantly on ViT for **blending-based trigger attacks with PatchShuffle**. As a result, for both cases, ViT has a larger gap between accuracy and ASR than CNN. It is important to note that these characteristics are not observed on CNN models. Therefore, the observed impact of patch processing against backdoor attacks is unique to ViT.

### 5 Novel Defensive Solution for ViT

#### 5.1 Methodology
Based on our observations above, we propose an effective backdoor detection algorithm that can successfully detect and then remove the poison samples from a backdoor-injected ViT model with high success rates. The key intuition in our algorithm is that the patch processing strategies affect ViTs’ predictions on the backdoor samples differently from the predictive function of the model on the clean data. Our defense algorithm exploits the frequency that ViTs change their predictions on the same sample under different trials of a patch processing strategy and use a threshold to assess if a sample is clean or poisoned. Our defense mechanism only requires access to a small set of clean samples (less than 1000 on the studied datasets), which can be easily obtained from the held-out validation dataset, for selecting the threshold. When no such clean samples are available, we show that the defenses are still very effective, which enables much wider applicability of the proposed method. The proposed defense consists of the following steps:

- **Step 1 (Offline):** For the small set of clean samples, randomly apply PatchDrop and PatchTranslate on each image for $T$ trials. For each sample $x$, we calculate $F_d(x) = \sum_{t=1}^{T} 1\{F(x) \neq F(R_d(t)(x))\}$ and

#### Algorithm 1: Patch Processing-based Backdoor Detection

**Input:** Sample $x$, Threshold $k_d$ (PatchDrop), Threshold $k_s$ (PatchShuffle)

**Output:** Clean or Backdoor Decision

1. **function** $F(x)$
2. $t \leftarrow 0$, $F_d(x) \leftarrow 0$, $F_s(x) \leftarrow 0$, Predict $\hat{y} = F(x)$
3. **repeat**
4. $t \leftarrow t + 1$
5. $\hat{y}_t = F(R_d(x)), F_d(x) \leftarrow F_d(x) + 1$ if $\hat{y}_t \neq \hat{y}$
6. $\hat{y}_t = F(R_s(x)), F_s(x) \leftarrow F_s(x) + 1$ if $\hat{y}_t \neq \hat{y}$
7. **until** $t = T$
8. **return** $F_d(x)$ and $F_s(x)$
9. **end function**
10. If $F_d(x) > k_d$ or $F_s(x) < k_s$, $x$ is Backdoor
11. Otherwise, $x$ is Clean

---

510
While keeping the false negative rate low; however, this is an anomaly detector. Thus, more sophisticated anomaly detection of samples will be identified as false negative. Formally, of the percentile selection rules, only a very small fraction of samples will be identified as a backdoor sample. Otherwise, we flag the sample as a backdoor sample.

**Step 2 (Offline):** Given the sample \( \{ F_d(x_i) \}_{i=1,...,K} \) or \( \{ F_s(x_i) \}_{i=1,...,K} \) created in Step 1, we set the threshold parameter \( k_d \) and \( k_s \) for PatchDrop and PatchShuffle, respectively, to the values at the \( n^{th} \) percentiles to ensure a small false positive rate, as follows:

- For PatchDrop, we typically select a large value (e.g., 90\% percentile). This is because ASRs significantly decrease under patch processing such as PatchDrop.

- For PatchShuffle, we typically select a small value (e.g., 10\% percentile). This is because ASRs do not drop under patch processing such as PatchShuffle while the clean-data accuracies are more affected.

**Step 3 (During Inference):** For a sample, we randomly apply PatchDrop and PatchShuffle for \( T \) trials and record the number of label changes, \( F_d(x) \) and \( F_s(x) \), respectively. If \( F_d(x) \) is greater than the selected \( k_d \) threshold for PatchDrop or \( F_s(x) \) is smaller than the selected \( k_s \) threshold for PatchShuffle, we flag the sample as a backdoor sample. Otherwise, \( x \) is determined as a clean sample.

Note that, the proposed approach does not assume the knowledge of the type of the backdoor attack, which ensures its practicality in various scenarios. Furthermore, when the model is benign, i.e., without the backdoor attack, because of the percentile selection rules, only a very small fraction of samples will be identified as false negative. Formally, our defense approach follows a similar strategy as that of an anomaly detector. Thus, more sophisticated anomaly detection approaches can be used to improve the detection rate while keeping the false negative rate low; however, this is beyond the scope of this paper. The details of the detection algorithm are presented in Algorithm 1.

### 5.2 Analysis of Patch Processing-based Defense

We first provide a qualitative analysis of the proposed defense strategy for detecting both patch-based and blendingattacks. For a sample, we randomly apply PatchDrop and PatchShuffle for \( T \) trials and record the number of label changes, \( F_d(x) \) and \( F_s(x) \), respectively. If \( F_d(x) \) is greater than the selected \( k_d \) threshold for PatchDrop or \( F_s(x) \) is smaller than the selected \( k_s \) threshold for PatchShuffle, we flag the sample as a backdoor sample. Otherwise, \( x \) is determined as a clean sample.

Note that, the proposed approach does not assume the knowledge of the type of the backdoor attack, which ensures its practicality in various scenarios. Furthermore, when the model is benign, i.e., without the backdoor attack, because of the percentile selection rules, only a very small fraction of samples will be identified as false negative. Formally, our defense approach follows a similar strategy as that of an anomaly detector. Thus, more sophisticated anomaly detection approaches can be used to improve the detection rate while keeping the false negative rate low; however, this is beyond the scope of this paper. The details of the detection algorithm are presented in Algorithm 1.

![TPR and TNR for different numbers of dropped patches](image)

**Figure 4:** TPR and TNR for different numbers of dropped patches (in a spatial grid of 8 x 8) for CNNs (ResNet18 and Vgg11) and ViTs (ViT and DeiT). TPR represents the detection rate; TNR represents clean-sample mis-detection rate.

![TPR and TNR for different sizes of processed patches](image)

**Figure 5:** TPR and TNR for different sizes of processed patches for CNN models (ResNet18 and Vgg11) and ViTs (ViT and DeiT) under ReFool backdoor attack.
Table 3: TPR (best bolded) and TNR (best underlined) of detecting backdoor samples in BadNets’ poisoned models.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ViT</th>
<th>DeiT</th>
<th>Vgg11</th>
<th>ResNet18</th>
<th>BiT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>90.08</td>
<td>99.48</td>
<td>91.68</td>
<td>96.18</td>
<td>88.12</td>
</tr>
<tr>
<td>GTSRB</td>
<td>94.89</td>
<td>98.78</td>
<td>94.62</td>
<td>97.66</td>
<td>93.80</td>
</tr>
<tr>
<td>TinyImageNet</td>
<td>95.80</td>
<td>98.80</td>
<td>94.64</td>
<td>81.30</td>
<td>20.51</td>
</tr>
</tbody>
</table>

Table 4: TPR (best bolded) and TNR (best underlined) of detecting backdoor samples in ReFool’s poisoned models.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ViT</th>
<th>DeiT</th>
<th>Vgg11</th>
<th>ResNet18</th>
<th>BiT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>90.00</td>
<td>83.20</td>
<td>96.80</td>
<td>81.50</td>
<td>18.00</td>
</tr>
<tr>
<td>GTSRB</td>
<td>84.10</td>
<td>95.60</td>
<td>93.60</td>
<td>98.50</td>
<td>80.20</td>
</tr>
<tr>
<td>TinyImageNet</td>
<td>90.90</td>
<td>87.70</td>
<td>85.30</td>
<td>85.00</td>
<td>69.90</td>
</tr>
</tbody>
</table>

6 Defense Experimental Results

This section presents the empirical results in defending against the backdoor attacks. In real-world settings, the defender does not know which attack is performed by the adversary. To this end, we consider two practical scenarios.

In the first scenario, the backdoor is successfully injected into the trained model and the victim defends against backdoor attacks (i.e., alleviates its effectiveness) by filtering the backdoor samples during inference. In this experiment, TPR and TNR are reported, as they demonstrate how likely the clean samples are not falsely detected as backdoor samples, respectively. As we can observe, the defensive solution with PatchDrop works better for ViT models than for CNN models such as ResNet18 and Vgg11. Furthermore, dropping 10% of the patches can consistently achieve higher TPR and TNR across different datasets. The effectiveness of this defense on ViTs is because the backdoor performance is more sensitive to PatchDrop, as discussed in the previous section.

Blending-based Attacks Figure 5 illustrates the TPR and TNR when defending against ReFool with various sizes of the processed patches in PatchShuffle. As we can observe, PatchShuffle generally achieves higher TPRs in ViTs than in CNN models. More importantly, when the patch size is similar to that of the trained patch size in ViTs, defending against ViTs is consistently effective.

6.1 Defending against the Poisoned Model

Table 3 and Table 4 present the defense results when the defender aims to detect whether a sample is a backdoor or clean sample during inference under the local patch-based attack, BadNets, and the global blending-based attack, ReFool, respectively. As we can observe, the proposed defensive solution achieves comparable TPRs (i.e., successfully detects the backdoor samples) in both ViTs (>90%) and CNN models (>88%) across different datasets under BadNets attacks. However, the TNRs of ViTs, including ViT and DeiT, are significantly better than those of CNN models, including Vgg11, ResNet18 and BiT. Specifically, the proposed defense method only falsely detects clean samples as backdoor samples less than 3% of the time in the trained ViTs, but more than 10% of the time in the trained CNN models. Under ReFool attacks, the defensive solution achieves the best TPR for ViT, while its TNRs are also very high. While the TNRs of ResNet18 and BiT are higher than those of ViTs, their TPRs are significantly lower, especially in the larger-scale TinyImageNet dataset. Overall, we can conclude that the proposed defense method is consistently more effective in ViTs than in CNN models.
Table 5: Clean-data accuracy (best bolded) and ASR after removing the backdoor samples and retraining the models. *Italicized* values are relative changes w.r.t. the models trained without removing backdoor samples.

### 6.2 Defending against the Poisoned Training Data

We present the clean-data accuracies and ASRs after retraining the models on the filtered data, as described in the second scenario, in Table 5. The attack method is patch-based. We can observe that the proposed defense method successfully reduces the ASRs much closer to ASRs of random guesses in both ViTs and CNNs on all datasets. However, in ViTs, the clean-data accuracies are preserved, while in CNN models, the clean-data accuracies drop more than 4.5% for ResNet18 and almost 1.5% for BiT. The results for Vgg11 are worse than those of ResNet18 and BiT and are reported in supplement materials. As discussed in the previous experiment, a non-trivial number of clean samples can be falsely detected as backdoor samples in CNN models using the proposed patch-processing approach. Thus, while most backdoor samples are removed from the training datasets, the number of clean training samples is also reduced, which leads to the drop in clean-data performance in CNN models. We can also notice that by employing a large-scale pre-trained model (i.e., BiT), the drop in performance can be mitigated compared to smaller models, such as ResNet18 and Vgg11. Nevertheless, we can still observe that the proposed defense is more effective for ViTs than for CNN models.

In conclusion, while ViT is vulnerable to patch-based backdoor attacks, the proposed simple-yet-effective patch-processing-based defense can detect backdoor samples with a high detection rate while maintaining a low FNR. Because ViT is robust against patch processing on the clean data, processing the images with these strategies can be utilized to obtain useful yet tangible traces for effectively distinguishing the predictions between the clean and backdoor samples.

### 7 Conclusion

This paper studied several aspects of backdoor attacks against ViT. We first perform an empirical study on the vulnerability of ViT against both patch-based and blending-based backdoor attacks. Then, based upon our observation that ViT exhibits distinguishable performance between clean samples and backdoor samples against patch processing, we proposed a novel defensive solution to counter backdoor attacks on ViT, which is able to reduce the backdoor attack success rate significantly. Two patch processing methods are investigated. The effectiveness of the proposed techniques is comprehensively evaluated. To the best of our knowledge, this paper presented the first defensive strategy that utilizes a unique characteristic of ViT against backdoor attacks.

### Acknowledgments

The research was conducted while all authors worked at Baidu Cognitive Computing Lab – 10900 NE 8th St. Bellevue, WA 98004, USA.

### References


