PATCH-FOOL: ARE VISION TRANSFORMERS ALWAYS ROBUST AGAINST ADVERSARIAL PERTURBATIONS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision transformers (ViTs) have recently set off a new wave in neural architecture design thanks to their record-breaking performance in various vision tasks. In parallel, to fulfill the goal of deploying ViTs into real-world vision applications, their robustness against potential malicious attacks has gained increasing attention. In particular, recent works show that ViTs are more robust against adversarial attacks as compared with convolutional neural networks (CNNs), and conjecture that this is because ViTs focus more on capturing global interactions among different input/feature patches, leading to their improved robustness to the local perturbations imposed by adversarial attacks. In this work, we ask an intriguing question: "Under what kinds of perturbations do ViTs become weaker learners compared to CNNs"? Driven by this question, we conduct a comprehensive examination on the robustness of both ViTs and CNNs under various existing adversarial attacks to understand the underlying reason for their robustness. Based on the insights drawn, we have developed a dedicated attack framework, dubbed Patch-Fool, that fools the self-attention mechanism by attacking the basic component (i.e., a single patch) participating in self-attention calculations with a series of attention-aware optimization techniques. Based on extensive experiments, we find that ViTs are weaker learners compared with CNNs against our Patch-Fool and the results from Sparse Patch-Fool, a sparse variant of our Patch-Fool, indicate that the perturbation density on each patch seems to be the key factor that influences the robustness ranking between ViTs and CNNs. It is expected that our work will shed light on both future architecture designs and training schemes for robustifying ViTs towards their real-world deployment. All codes will be released upon acceptance.

1 INTRODUCTION

The recent performance breakthroughs achieved by vision transformers (ViTs) Dosovitskiy et al. (2020) have fueled an increasing enthusiasm towards designing new ViT architectures for different vision tasks, including object detection (Carion et al., 2020; Beal et al., 2020), semantic segmentation (Strudel et al., 2021; Zheng et al., 2021; Wang et al., 2021), and video recognition (Arnab et al., 2021; Liu et al., 2021b; Li et al., 2021b; Fan et al., 2021). To fulfill the goal of deploying ViTs into real-world vision applications, the security concern of ViTs is of great importance and challenge, especially in the context of adversarial attacks (Goodfellow et al., 2014), under which an imperceptible perturbation onto the inputs can mislead the models to malfunction.

In response, the robustness of ViTs against adversarial attacks has attracted increasing attention. For example, recent works (Bhojanapalli et al., 2021; Aldahdooh et al., 2021; Shao et al., 2021) find that in addition to ViTs' decent task performance, they are more robust to adversarial attacks compared with convolutional neural networks (CNNs) under comparable model complexities. In particular, (Shao et al., 2021) claims that ViTs focus more on capturing the global interaction among input/feature patches via its self-attention mechanism and the learned features contain less low-level information, leading to superior robustness to the local perturbations introduced by adversarial attacks. A natural response to this seemingly good news would be determining whether ViTs are truly robust against all kinds of adversarial perturbations or if their current winning in robustness is an inevitable result of biased evaluations using existing attack methods that are mostly dedicated to CNNs. To unveil the potential vulnerability of ViTs, this work takes the first step in asking an intriguing question:

"Under what kinds of perturbations do ViTs become weaker learners compared to CNNs?", and makes the following contributions:

- We conduct extensive experiments to benchmark the robustness of different ViT variants, which integrate recent advances and new arts in ViTs' architecture design, over that of CNNs under existing adversarial attacks;
- We propose a new attack framework, dubbed Patch-Fool, aiming to fool the self-attention mechanism by attacking the basic component (i.e., a single patch) participating in self-attention calculations. Our Patch-Fool attack features a novel objective formulation, which is then solved by Patch-Fool's integrated attention-aware patch selection technique and attention-aware loss design;
- We evaluate the robustness of both ViTs and CNNs against our Patch-Fool attack with extensive experiments and find that ViTs are consistently less robust than CNNs across various attack settings, indicating that ViTs are not always robust learners and their seeming robustness against existing attacks can be overturned under dedicated adversarial attacks;
- We further benchmark the robustness of both ViTs and CNNs under a sparse variant of Patch-Fool, dubbed Sparse Patch-Fool, and discover that the perturbation density, defined as the number of perturbed pixels per patch, highly influences the robustness ranking between ViTs and CNNs, where our Patch-Fool is an extreme case of high perturbation density.

We believe our work has opened up a new perspective for exploring ViTs' vulnerability and can provide insights to both future architecture designs and training schemes for robustifying ViTs towards their real-world deployment.

2 Related Works

Vision transformers. Motivated by the great success of Transformers in the natural language processing (NLP) field (Vaswani et al., 2017), ViTs have been developed by splitting an input image into a series of image patches and adopting self-attention modules for encoding the image (Dosovitskiy et al., 2020), and been shown to achieve competitive or superior performance over CNNs via dedicated data augmentation (Touvron et al., 2021) or self-attention structure (Yang et al., 2021; Graham et al., 2021; Liu et al., 2021a). As such, there has been tremendously increased attention on applying ViTs to various computer vision applications, such as self-supervised learning (Caron et al., 2021; Chen et al., 2021b; Xie et al., 2021; Li et al., 2021a), object detection (Carion et al., 2020; Beal et al., 2020), and semantic segmentation (Strudel et al., 2021; Zheng et al., 2021; Wang et al., 2021). The achievable performance of ViTs are continuously refreshed by emerging ViT variants, which provide new arts for designing ViT architectures. For example, convolutional modules have been incorporated into ViTs for capturing low-level features (Xiao et al., 2021; Wu et al., 2021; Graham et al., 2021; Peng et al., 2021), and replacing the global self-attention mechanism with local self-attention modules (Liu et al., 2021a; Dong et al., 2021; Liang et al., 2021; Liu et al., 2021b; Chu et al., 2021). Motivated by ViTs' impressive performance and the growing interest in deploying ViTs into real-world applications, this work aims to better understand the robustness of ViTs and to develop adversarial attacks dedicated to ViTs.

Adversarial attack and defense. Deep neural networks (DNNs) are known to be vulnerable to adversarial attacks (Goodfellow et al., 2014), i.e., imperceptible perturbations onto the inputs can mislead DNNs to make wrong predictions. As adversaries, stronger attacks are continuously developed, including both white-box (Madry et al., 2017; Croce & Hein, 2020; Carlini & Wagner, 2017; Papernot et al., 2016; Moosavi-Dezfooli et al., 2016) and black-box ones (Chen et al., 2017; Ilyas et al., 2018b; Andriushchenko et al., 2020; Guo et al., 2019; Ilyas et al., 2018a), which aggressively degrade the performances of the target DNN models. In response, various defense methods have been proposed to improve DNNs' robustness against those adversarial attacks. For example, purification methods (Guo et al., 2017; Xie et al., 2017; Xu et al., 2017; Song et al., 2017; Shi et al., 2021) applied onto the inputs purify the adversarial samples back to the distribution of clean ones; random smoothing (Cohen et al., 2019; Li et al., 2018) onto the inputs certifiably guarantees DNNs' robustness against adversarial perturbations; detection-based methods (Metzen et al., 2017; Feinman et al., 2017; Lee et al., 2018) equipped with additional detection modules distinguish adversarial examples from clean ones before feeding them into the target DNN models; and adversarial training (Shafahi et al., 2019; Madry et al., 2017; Wong et al., 2019; Tramèr et al.,

2017), which augments the training set with adversarial samples generated on the fly during training, is currently one of the most effective defense methods. There has been a continuous war between adversaries and defenders and the readers are referred to (Akhtar & Mian, 2018; Chakraborty et al., 2018) for more attack and defense methods.

Robustness of vision transformers. Driven by the impressive performance recently achieved by ViTs in various vision tasks, their robustness has gained increasing attention. A consistent observation drawn by pioneering works that study ViTs' robustness is that ViTs are more robust to adversarial attacks than CNNs since ViTs are more capable of capturing the global interactions among patches, while CNNs focus on local features and thus are more vulnerable to local adversarial perturbations. In particular, (Bhojanapalli et al., 2021) shows that ViT models pretrained with a sufficient amount of data are at least as robust as their ResNet counterparts on a broad range of perturbations, including natural corruptions, distribution shifts, and adversarial perturbations; (Aldahdooh et al., 2021) finds that vanilla ViTs or hybrid-ViTs are more robust than CNNs under L_p-based attacks; and (Shao et al., 2021) further explains that ViTs' learned features contain less low-level information and are more generalizable, leading to their superior robustness, and introducing convolutional blocks that extract more low-level features will reduce the ViTs' adversarial robustness. In addition, ViTs' adversarial transferability has also been studied: (Mahmood et al., 2021) shows that adversarial examples do not readily transfer between CNNs and transformers and (Naseer et al., 2021; Wei et al., 2021) propose techniques to boost the adversarial transferability between ViTs and from ViTs to CNNs. In parallel, (Mao et al., 2021) refines ViTs' architecture design to improve robustness. In our work, we challenge the common belief that ViTs are more robust than CNNs, which is concluded based on evaluations using existing attack methods, and propose to customize adaptive attacks utilizing ViTs' captured patch-wise global interactions to make ViTs weaker learners.

3 Evaluating robustness of ViTs and CNNs under existing attacks

Although various comparisons on robustness of ViTs and CNNs have been explored in pioneering works (Bhojanapalli et al., 2021; Aldahdooh et al., 2021; Shao et al., 2021), their evaluation suffers from one of the following limitations: (1) only adopt weak attack methods, (2) only adopt early ViT designs without considering recently advanced ViT architectures, and (3) do not adopt the official and latest pretrained models and suffer from inferior clean accuracies. To this end, we extensively evaluate the robustness against common white-box attacks of several representative ViT variants, which cover the popular trends in designing ViT architectures, including (1) using local self-attention (Swin (Liu et al., 2021a)), which adopts the attention mechanism within a local region instead of the global ones in vanilla ViTs to capture low-level features and reduce the computational cost, and (2) introducing the inductive bias of CNNs to build hybrid models (LeViT (Graham et al., 2021)).

Table 1: Benchmark the robustness of three ViT families and two CNN families against PGD attacks (Madry et al., 2017), Auto-Attack (Croce & Hein, 2020), and CW attacks (Carlini & Wagner, 2017), where the perturbation strengths of the L-inf attacks are annotated.

Model	**FLOPs/ Params**	**Clean Acc (%)**	**PGD-20 (%)**			**Auto-Attack (%)**		**CW-L_2 (%)**	**CW-L_{Inf} (%)**	
			0.001	**0.003**	**0.005**	**0.001**	**0.003**		**0.003**	**0.005**
ResNet-18	1.80G/23M	71.78	22.60	0.34	0.00	22.29	0.06	9.95	9.10	5.31
ResNet-50	4.11G/98M	76.15	35.00	1.42	0.18	25.42	1.23	10.80	13.39	7.95
ResNet-101	7.83G/170M	78.25	37.20	2.10	0.42	27.92	1.88	11.19	15.36	11.11
ResNet-152	11.56G/230M	78.31	42.09	2.56	0.24	39.58	2.34	15.20	17.24	14.58
VGG-16	15.3G/134M	71.30	23.24	0.28	0.02	15.42	0.04	8.15	7.84	4.84
DeiT-Ti	1.26G/5M	72.02	41.51	6.73	1.52	38.29	5.75	30.08	28.74	21.05
DeiT-S	4.61G/22M	79.53	53.70	11.40	2.91	51.88	7.70	43.78	44.22	34.65
DeiT-B	17.58G/86M	81.90	52.20	10.10	2.28	51.08	8.62	43.77	47.40	36.35
Swin-T	4.51G/29M	80.96	38.80	2.72	1.08	35.81	2.04	28.84	35.10	20.63
Swin-S	8.77G/50M	82.37	47.20	6.79	0.42	43.25	4.58	30.32	41.22	28.91
Swin-B	15.47G/88M	84.48	47.16	7.30	1.14	44.58	6.52	34.34	43.57	34.00
LeViT-256	1.13G/19M	81.60	36.42	3.00	0.64	31.25	2.29	43.94	46.57	25.58
LeViT-384	2.35G/39M	82.60	42.19	3.08	0.24	35.83	2.50	47.15	50.22	39.56

3.1 Evaluation Setup

Models and datasets. We evaluate the robustness of three ViT families (i.e., DeiT (Touvron et al., 2021), Swin (Liu et al., 2021a), and LeViT (Graham et al., 2021)) and two CNN families

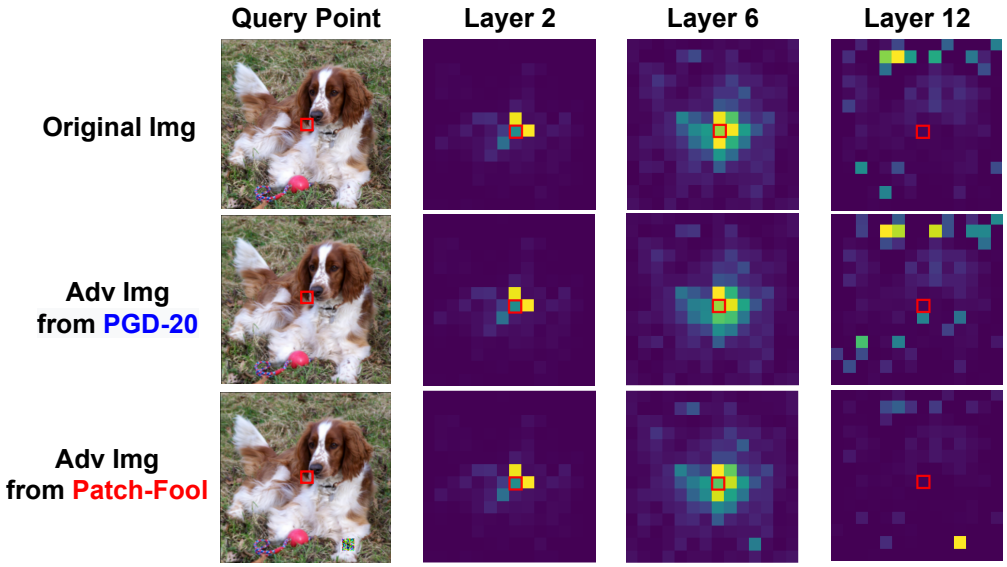

Figure 1: Comparisons among the attention maps in the intermediate layers of DeiT-S generated by the clean inputs, the adversarial inputs under PGD-20 attacks ($\epsilon = 0.003$), and the proposed Patch-Fool attack, respectively. In particular, we average the attention scores across all the attention heads in each layer and visualize the attention score of each token for a given query token (the center patch in the red box in our show case), following (Kim et al., 2021). We can observe that the difference in attention maps between clean and adversarial inputs generated by PGD-20 keeps small across different layers; In contrast, the proposed Patch-Fool notably enlarges the gap between clean and adversarial attention maps, demonstrating a successful attack for ViTs.

(ResNet (He et al., 2016) and VGG (Simonyan & Zisserman, 2014)) on ImageNet using their official implementation and pretrained models. Note that we adopt DeiT models without distillation, which only improves the training schedule over vanilla ViTs, for a fair comparison.

Attack settings. We adopt four adversarial attacks (i.e., PGD (Madry et al., 2017), AutoAttack (Croce & Hein, 2020), CW-L_{Inf} (Carlini & Wagner, 2017), and CW-L_2) with different perturbation strengths. In particular, for the CW-L_{Inf} and CW-L_2 attacks, we adopt the implementation in AdverTorch (Ding et al., 2019) and the same settings as (Chen et al., 2021a; Rony et al., 2019); For AutoAttack, we adopt the official implementation and default settings in (Croce & Hein, 2020).

3.2 Observations and Analysis

Observations. From the evaluation results summarized in Tab. 1, we make the following observations: (1) ViTs are consistently more robust than CNNs with comparable model complexities under all attack methods, which is consistent with the previous observations (Bhojanapalli et al., 2021; Aldahdooh et al., 2021; Shao et al., 2021). In particular, DeiT-S/DeiT-B achieves a 18.70%/10.11% higher robust accuracy over ResNet-50/ResNet-152 under PGD-20 attacks with a perturbation strength of 0.001; (2) compared with vanilla ViTs, ViT variants equipped with local self-attention or convolutional modules, which improves the model capability to capture local features and thus boosts the clean accuracy, are more vulnerable to adversarial attacks, although they are still more robust than CNNs with comparable complexities. For example, Swin-T/Swin-B suffers from a 14.90%/5.04% robust accuracy drop compared with DeiT-S/DeiT-B under PGD-20 attacks with a perturbation strength of 0.001; and (3) the degree of overparameterization has less influence in the robustness for the same family of ViT models compared with its great influence in CNNs' robustness, as the most lightweight DeiT-Ti can already achieve a comparable robust accuracy (-0.58%) as ResNet-152, while requiring 9.17×/46× less floating-point operations (FLOPs)/parameters.

Analysis. Combining the three insights drawn from the aforementioned observations, we can observe the superiority of the global attention mechanism over convolutional and local self-attention blocks, in terms of both improved robust accuracy and reduced sensitivity to the degree of model overparameterization. This indicates that the global attention mechanism itself can serve as a good robustification technique against existing adversarial attacks, even in lightweight ViTs with small

model complexities. For example, as shown in Fig. 1, the gap between the attention maps generated by clean and adversarial inputs in deeper layers remains small. We wonder that "Are the global attentions in ViTs truly robust, or their vulnerability has not been fully explored and exploited?". To answer this, we propose our customized attack in Sec. 4.

4 THE PROPOSED PATCH-FOOL FRAMEWORK

In this section, we present our Patch-Fool attack method that perturbs a whole patch to fool ViTs and unveils a vulnerable perspective of ViTs.

4.1 PATCH-FOOL: MOTIVATION

Given the insensitivity of ViTs' global self-attention mechanism to local perturbations, we pay a close attention to the basic component (i.e., a single patch) participating in the self-attention calculation, and hypothesize that customized adversarial perturbations onto a patch can be more effective in fooling the captured patch-wise global interactions of self-attention modules than attacking the CNN modules. This is also inspired by the word substitution attacks (Alzantot et al., 2018; Ren et al., 2019; Jin et al., 2019; Zang et al., 2019) to Transformers in NLP tasks, which replace a word with its synonyms, and here an image patch in ViTs serves a similar role as a word.

4.2 PATCH-FOOL: SETUP AND OBJECTIVE FORMULATION

Attack setup. In our proposed Patch-Fool Attack, we do not limit the perturbation strength onto each pixel and, instead, constrain all the perturbed pixels within one patch (or several patches), which can be viewed as a variant of sparse attacks (Dong et al., 2020; Modas et al., 2019; Croce & Hein, 2019). Such attack strategies will lead to adversarial examples with a noisy patch as shown in Fig. 1, which visually resembles and emulates natural corruptions in a small region of the original image, e.g., one noisy patch only counts for 1/196 in the inputs of DeiT-S (Touvron et al., 2021), caused by potential defects of the sensors or potential noises/damages of the optical devices.

Objective formulation. Given the cost function J and a series of input image patches $x = [x_1, \cdots, x_n]^\top \in \mathbb{R}^{n \times d}$ with its associated label y, the objective of our adversarial algorithm is:

$$\underset{1 \le p \le n, \delta \in \mathbb{R}^{n \times d}}{\arg\max} \quad J(x + \mathbb{1}_p \odot \delta, y) \tag{1}$$

where $\mathbb{1}_p \in \mathbb{R}^n$ such that $\mathbb{1}_p(i) = \begin{cases} 0, & i \neq p \\ 1, & i = p \end{cases}$ is a one hot vector and $\odot$ represents the penetrating face product such that $a \odot b = [a \circ b_1, \cdots, a \circ b_d]$ where $\circ$ is the Hadamard product and b_j is the j-th column of matrix b. For solving Eq. 1, our Patch-Fool needs to (1) select the adversarial patch p, and (2) optimize the corresponding δ as elaborated in Sec. 4.3 and Sec. 4.4, respectively.

4.3 PATCH-FOOL: DETERMINE p VIA ATTENTION-AWARE PATCH SELECTION

Denoting $a^{(\ell,h,i)} = [a_1^{(\ell,h,i)}, \cdots, a_n^{(\ell,h,i)}] \in \mathbb{R}^n$ as the attention distribution for the i-th token of the h-th head in the ℓ-th layer, for each layer ℓ we define:

$$s_j^{(\ell)} = \sum_{h,i} a_j^{(\ell,h,i)} \tag{2}$$

which measures the importance of the j-th token in the ℓ-th layer based on its contributions to other tokens in the self-attention calculation. For better fooling ViTs, we select the most influential patch $p \coloneqq \arg\max_j s_j^{(\ell)}$ according to a predefined value ℓ. We fix $\ell = 5$ by default since the patches at later self-attention layers are observed to be diverse from the input patches due to the increased information mixed from other patches, making them non-ideal for guiding the selection of input patches as justified in Sec. 5.3.

4.4 PATCH-FOOL: OPTIMIZE δ VIA ATTENTION-AWARE LOSS

Given the selected adversarial patch index p from the above step, we define the *attention-aware loss* for the ℓ-th layer as follows:

$$J_{\text{attn}}^{(\ell)}(x) = \sum_{h,i} a_p^{(\ell,h,i)} \tag{3}$$

which is expected to be maximized so that the adversarial patch p, serving as adversarial perturbations, can attract more attention from other patches for more effectively fooling the models. The perturbation δ is then updated based on both the final classification loss (i.e., cross-entropy loss) and a layer-wise attention-aware loss:

$$\nabla_{x_p} J(x, y) = \nabla_{x_p} J_{ce}(x, y) + \sum_{l} \alpha_\ell \nabla_{x_p} J^{(\ell)}_{\text{attn}}(x) \tag{4}$$

where $x_p = x + \mathbb{1}_p \odot \delta$, and α_ℓ is the weighting coefficient of the attention-aware loss in the ℓ-th layer and defined as

$$\alpha_\ell = \frac{\nabla_{x_p} J_{ce}(x, y) \cdot \nabla_{x_p} J^{(\ell)}_{\text{attn}}(x)}{\|\nabla_{x_p} J_{ce}(x, y)\| \|\nabla_{x_p} J^{(\ell)}_{\text{attn}}(x)\|} \tag{5}$$

which is the cosine similarity between $\nabla_{x_p} J_{ce}(x, y)$ and $\nabla_{x_p} J^{(\ell)}_{\text{attn}}(x)$, aiming to balance the different losses and maintain the main component of $\nabla_{x_p} J_{ce}(x, y)$ as inspired by (Yu et al., 2020).

Following PGD (Madry et al., 2017), we iteratively update δ using an Adam optimizer (Kingma & Ba, 2014):

$$\delta^{t+1} = clip_{(1-x)}\{\delta^t + \tau \cdot Adam(\nabla_{x^t_p} J(x + \mathbb{1}_p \odot \delta^t, y))\} \tag{6}$$

where τ is the step size and $clip_{(1-x)}$ is to ensure the perturbed pixel is within a valid color range.

4.5 Sparse Patch-Fool: A sparse variant of Patch-Fool

Motivation. One natural question associated with Patch-Fool is: "How many pixels within a patch are needed to be perturbed for effectively misleading the model to misclassify the input image?". There exist two extreme cases: (1) perturbing only a few pixels that lead to local perturbations against which ViTs are more robust, and (2) perturbing the whole patch, i.e., our vanilla Patch-Fool. We hypothesize that answering this question helps better understand under what circumstances ViTs are more (or less) robust than CNNs. To this end, we study a variant of Patch-Fool, dubbed Sparse Patch-Fool, as defined below.

Objective formulation. For enabling Sparse Patch-Fool, we add a sparse constraint to Eq. 1, i.e.:

$$\underset{1 \leq p \leq n, \delta \in \mathbb{R}^{n \times d}, m \in \{0,1\}^{n \times d}}{\arg\max} \quad J(x + \mathbb{1}_p \odot (m \circ \delta), y) \;\; s.t. \;\; \|m\|_0 \leq k \tag{7}$$

where we use a binary mask m with a predefined sparsity parameter k to control the sparsity of δ. To effectively learn the binary distribution of m, we parameterize m as a continuous value $\hat{m}$, following (Ramanujan et al., 2020; Diffenderfer & Kailkhura, 2021). During forward, only the top k highest elements of $\hat{m}$ is activated and set to 1 and others are set to 0 to satisfy the target sparsity constraint; and during backward, all the elements in $\hat{m}$ will be updated via straight-through estimation (Bengio et al., 2013). We jointly optimize $\hat{m}$ with δ as in Eq. 6.

5 Evaluation of Patch-Fool

In this section, we evaluate the robustness of both ViT and CNN models against the proposed Patch-Fool attack. We first benchmark their lower-bound robustness under our Patch-Fool attacks with the strongest attack setting for both the DeiT and ResNet families in Sec. 5.2, and then conduct a comprehensive ablation study about the design choices of Patch-Fool in Sec. 5.3~ 5.4. Finally, we evaluate the robustness against Sparse Patch-Fool for drawing more insights regarding when ViTs are more (less) robust than CNNs in Sec. 5.5.

5.1 Evaluation setup

The step size in Eq. 6 is initialized to be 0.2 and decayed by 0.95 every 10 iterations, and the number of total iterations is 250. For evaluating Patch-Fool with different perturbation strengths, we allow Patch-Fool to attack up to four patches based on the attention-aware patch selection in Sec. 4.3, i.e., the patches with top importance scores defined in Eq. 2 will be selected. For Sparse Patch-Pool, given several perturbed patches, the top k value in m is globally ranked and selected among all perturbed patches. Note that we report the robust accuracy instead of the attack success rate throughout this paper as our main focus is the robustness benchmark.

5.2 BENCHMARK THE ROBUSTNESS OF VITS AND CNNS AGAINST PATCH-FOOL

We adopt our Patch-Fool described in Sec. 4.2~ Sec. 4.4 to attack ViTs and use the saliency map to guide the patch selection for attacking CNNs, which is the strongest attack setting as shown in Sec. 5.3. The resulting robust accuracy of both DeiT and ResNet families under different numbers of attacked patches is shown in Fig. 2. We can observe that DeiT models are consistently less robust against Patch-Fool than their ResNet counterparts under similar model complexity, e.g., compared with ResNet-50, DeiT-S suffers from a 16.31% robust accuracy drop under the single-patch attack of Patch-Fool, although it has a 3.38% and 18.70% higher clean and robustness accuracy against PGD-20 ($\epsilon = 0.001$), respectively. This indicates that ViTs are not always robust learners, and underperform under customized adversarial perturbations as compared to CNNs and their seeming robustness against existing attacks can be overturned.

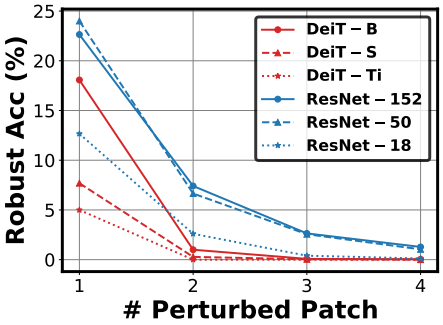

Figure 2: Benchmark the robustness of both DeiT and ResNet families against Patch-Fool under different numbers of perturbed patches.

5.3 ABLATION STUDY: EFFECTIVENESS OF THE ATTENTION-AWARE PATCH SELECTION

To validate the effectiveness of our attention-aware patch selection method, we benchmark two variants of patch selection mechanism: (1) random patch selection, and (2) saliency-map-based patch selection. For the latter one, we adopt the averaged saliency score of a patch, defined as the averaged absolution value of the gradients on each pixel in a patch following Simonyan et al. (2013), as the metric to select patches. For a fair comparison, we only adopt the final cross-entropy loss J_{ce} in Eq. 4 in this set of experiments. As shown in Tab. 2, we can see that (1) among the three strategies for attacking ViTs, our attention-aware patch selection is the most effective strategy in most cases and thus we adopt it by default; (2) DeiT variants are still consistently less robust than their ResNet counterparts under similar model complexity, indicating that attacking the basic component participating in self-attention calculations can indeed effectively degrade ViTs' robustness; and (3) Patch-Fool equipped with random patch selection, with a 2.64% robust accuracy gap against the best strategy, can already effectively degrade DeiTs' robustness, while it cannot effectively attack ResNets without the guidance from saliency map, indicating ViTs are generally more vulnerable than CNNs to the adversarial perturbations targeting their patches.

Table 2: Benchmark the robust accuracy of DeiTs and ResNets under different patch selection strategies, where 'xP' denotes a total of x patches are perturbed and the lowest robust accuracy is annotated in bold.

Model	Selection Strategy	Robust Acc (%) 1P	2P	3P	4P
DeiT-Ti	Random	5.21	0.16	**0**	**0**
	Saliency	6.49	**0.12**	**0**	**0**
	Attn-score	**4.01**	**0.12**	**0**	**0**
DeiT-S	Random	7.41	0.36	**0**	**0**
	Saliency	11.70	0.32	**0**	**0**
	Attn-score	**6.25**	**0.20**	**0**	**0**
DeiT-B	Random	24.76	2.22	0.20	**0.04**
	Saliency	24.67	1.56	0.20	**0.04**
	Attn-score	**22.12**	**1.32**	**0.16**	**0.04**
ResNet-18	Random	20.43	2.84	0.64	**0.04**
	Saliency	**12.66**	**2.60**	**0.40**	0.12
ResNet-50	Random	31.57	9.66	3.12	**0.60**
	Saliency	**24.00**	**6.65**	**2.56**	1.04
ResNet-152	Random	32.09	10.42	**2.64**	**1.00**
	Saliency	**22.64**	**7.41**	**2.64**	1.28

We also perform an ablation study for ℓ in Eq. 2 based on which layer the attention-aware patch selection is performed. As shown in Tab. 3, selecting the early layers generally achieve consistent better results than that of later layers, which we conjecture is because patches in early layers can still roughly maintain the original informa-

Table 3: Ablation study of the layer index for guiding the attention-aware patch selection. The resulting robust accuracy (%) is annotated in the table.

ℓ	**1**	**2**	**3**	**4**	**5**	**6**
DeiT-B	21.27	23.96	23.00	22.48	21.03	21.67
ℓ	**7**	**8**	**9**	**10**	**11**	**12**
DeiT-B	27.56	28.04	27.80	28.04	28.08	27.48

tion extracted from the inputs while their counterparts in later layers are mixed up with information from other patches, leading to an inferior guidance for selecting the input patch. This conjecture is validated by the observed phase change in the attention map, i.e., after the sixth layer, more complexity correlations between patches are captured in addition to the diagonal ones. Therefore, we set $\ell = 5$ by default without loss of generality.

5.4 Ablation study: Effectiveness of the attention-aware loss

To evaluate the effectiveness of our attention-aware loss with the cosine-similarity-based re-weighting mechanism (see Sec. 4.4), we compare it with two baselines: (1) train with only the final cross-entropy loss, i.e., J_{ce} is enabled without the attention-aware loss, and (2) $\alpha_\ell = 1, \forall \ell \in [1, 12]$, i.e., the layerwise $J_{attn}^{(\ell)}$ in Eq. 4 is directly summed together with the final J_{ce}. As shown in Tab. 4, we can observe that (1) our attention-aware loss equipped with the cosine-similarity-based re-weighting strategy consistently achieves the best attack performance, e.g., a 3.17% reduction in robust accuracy compared with the baseline without attention-aware loss; and (2) directly summing up all the losses leads to poor convergence especially under limited numbers of perturbed patches, indicating a balancing mechanism. This set of experiments verifies the effectiveness of our attention-aware loss design.

Table 4: Benchmark the attention-aware loss with two baselines, where 'Cos-sim' denotes our method of attention-aware loss with cosine-similarity-based re-weighting strategy while 'w/o' and 'Sum' are two baselines.

Model	Atten-Loss	Robust Acc (%)			
		1P	2P	3P	4P
DeiT-Base	w/o	22.12	1.32	0.16	**0.04**
	Sum	63.46	16.11	1.96	0.10
	Cos-sim	**18.95**	**1.00**	**0.08**	**0.04**

5.5 Benchmark against Sparse Patch-Fool

Setup. To study the influence of the sparsity of the perturbed pixels for both CNNs and ViTs, we consider the global perturbation ratio of the whole image (i.e., k/total-pixel) and vary the number of patches allowed to be perturbed.

Benchmark the robustness of ViTs and CNNs. As shown in Tab. 5, under different perturbation ratios and numbers of perturbed patches, neither ViTs nor CNNs will always be the winner in robustness. In particular, under relatively small perturbation ratios or more perturbed patches (e.g., when all patches are allowed to be perturbed), CNNs will suffer from worse robustness, while under relatively large perturbation ratios as well as fewer perturbed patches, ViTs will be weaker learners.

Insights. We analyze that small perturbation ratios or more perturbed patches will lead to less perturbed pixels within one patch, i.e., a lower perturbation density, which is closer to local perturbations against which ViTs are thus more robust than CNNs. In contrast, given more perturbed pixels in one patch, i.e., a higher perturbation density for which an extreme case is our vanilla Patch-Fool, ViTs become weaker learners than CNNs. This indicates that a high perturbation density can be a perspective for exploring ViTs' vulnerability, which has been neglected by existing adversarial attacks.

Table 5: Benchmark the robust accuracy of DeiTs and ResNets against Sparse Patch-Fool under different perturbation ratios and perturbed patches, where 'All' denotes all patches are allowed to be perturbed and the lower robust accuracy is annotated in bold.

Model	Patch	DeiT-S	ResNet-50	DeiT-B	ResNet-152
0.05%	1P	68.99	**57.01**	72.96	**58.57**
	2P	67.15	**51.8**	71.35	**54.61**
	4P	65.99	**47.96**	69.83	**49.84**
	All	68.39	**49.2**	67.03	**53.93**
0.10%	1P	59.74	**48.96**	65.26	**48.12**
	2P	56.45	**40.46**	61.54	**42.15**
	4P	52.20	**34.05**	58.89	**36.38**
	All	51.44	**24.16**	53.89	**30.50**
0.30%	1P	26.84	**30.69**	41.23	**29.89**
	2P	21.47	**20.99**	32.17	**21.51**
	4P	15.62	**12.58**	22.80	**14.18**
	All	14.02	**0.76**	16.55	**2.24**
0.40%	1P	**16.63**	26.52	32.13	**25.12**
	2P	**11.74**	16.5	22.92	**16.51**
	4P	**7.57**	9.46	12.14	**10.10**
	All	6.77	**0.36**	9.05	**0.52**
0.50%	1P	**7.89**	24.00	24.24	**21.75**
	2P	**6.29**	13.66	17.11	**14.22**
	4P	**3.61**	6.93	**7.33**	8.09
	All	3.57	**0.16**	4.81	**0.20**
0.60%	2P	**3.21**	11.18	**10.42**	11.46
	4P	**1.68**	5.57	**4.09**	6.01
	All	1.92	**0.12**	1.92	**0.12**
0.70%	2P	**1.79**	9.62	**8.26**	10.30
	4P	**0.87**	4.13	**2.32**	5.01
	All	0.84	**0.04**	1.02	**0.08**
0.80%	2P	**0.88**	8.57	**4.37**	8.89
	4P	**0.32**	3.69	**1.00**	4.25
	All	0.44	**0.04**	0.64	**0.12**

Table 6: Benchmark the robust accuracy of DeiTs and ResNets against Sparse Patch-Fool under a fixed perturbation ratio (0.05%/0.5%) with different numbers of perturbed patches, where 'PR' denotes perturbation ratio and 'xP' denotes totally x patches are perturbed and the lower robust accuracy is annotated in bold.

PR=0.05%	**1P**	**2P**	**4P**	**8P**	**16P**	**32P**	**64P**	**128P**	**All**
DeiT-S	68.99	67.15	65.99	65.1	67.79	65.71	65.81	68.99	68.39
ResNet-50	**57.01**	**51.80**	**47.96**	**43.39**	**39.78**	**38.54**	**40.34**	**44.79**	**49.20**
DeiT-B	72.96	71.35	69.83	69.67	68.95	68.71	67.91	68.51	67.03
ResNet-152	**58.57**	**54.61**	**49.84**	**46.03**	**43.99**	**43.83**	**44.67**	**49.40**	**53.93**
PR=0.5%	**1P**	**2P**	**4P**	**8P**	**16P**	**32P**	**64P**	**128P**	**All**
DeiT-S	**7.89**	**6.29**	**3.61**	**1.72**	1.44	2.08	2.64	3.08	3.57
ResNet-50	24.00	13.66	6.93	3.21	**1.24**	**0.40**	**0.16**	**0.08**	**0.16**
DeiT-B	24.24	17.11	**7.33**	**2.40**	**1.68**	1.88	2.88	3.73	4.81
ResNet-152	**21.75**	**14.22**	8.09	3.93	1.88	**0.96**	**0.24**	**0.16**	**0.20**

Table 7: Benchmark the robust accuracy against PGD_0 (Croce & Hein, 2019) attacks and the proposed Sparse Patch-Fool, where 'PR' denotes perturbation ratio.

Attack	PGD_0					**Sparse Patch-Fool**				
PR	**0.01%**	**0.05%**	**0.10%**	**0.30%**	**0.50%**	**0.01%**	**0.05%**	**0.10%**	**0.30%**	**0.50%**
DeiT-Small	**78.24**	74.96	67.40	39.20	21.48	78.33	**68.39**	**51.44**	**14.02**	**3.57**
DeiT-Base	**80.03**	78.80	72.96	50.24	31.44	80.93	**67.03**	**53.89**	**16.55**	**4.81**
ResNet-50	72.22	67.28	57.80	36.12	18.20	**71.71**	**49.20**	**24.16**	**0.76**	**0.16**
ResNet-152	**74.32**	69.69	61.48	41.52	27.10	74.72	**53.93**	**30.50**	**2.24**	**0.20**

Influence of the number of perturbed patches. We further study the influence of the number of perturbed patches under the same global perturbation ratio as shown in Tab. 6. We can see that (1) under a small perturbation ratio of 0.05% which is closer to local perturbations, CNNs are the consistent loser in robustness; and (2) under a relatively large perturbation ratio of 0.5%, although increasing the number of perturbed patches leads to a consistent reduction in CNNs' robust accuracy, the robustness reduction for ViTs will quickly saturate, i.e., ViTs gradually switches from the loser to the winner in robustness as compared to CNNs. This echoes our aforementioned analysis.

Is Sparse Patch-Fool a strong attack? We extend our Sparse Patch-Fool to an extreme case under which all patches are allowed to be perturbed, i.e., when our Sparse Patch-Fool serves as a general sparse attack method, and compare it with a SOTA sparse attack PGD_0 (Croce & Hein, 2019) as shown in Tab. 7. We can observe that (1) Sparse Patch-Fool further degrades the robust accuracy, i.e., achieves better attacking success rates, than PGD_0, under different perturbation ratios, e.g., resulting in a 17.91% / 26.63% lower robust accuracy on DeiT-S/Deit-B under a perturbation ratio of 0.50% over PGD_0; and (2) under PGD_0 attacks, DeiTs are still more robust than ResNets, indicating that vanilla sparse attacks are still limited by low perturbation densities. This set of experiments shows that our Sparse Patch-Fool can serve as robustness evaluation protocols, in addition to providing insights about the robustness ranking between CNNs and ViTs.

6 Conclusion

The recent breakthroughs achieved by ViTs in various vision tasks have attracted increasing attention on ViTs' robustness, aiming to fullfil the goal of deploying ViTs into real-world vision applications. In our work, we provide a new perspective regarding ViTs' robustness and propose a novel attack framework, dubbed Patch-Fool, to attack the basic component (i.e., a single patch) participating in self-attention calculations, against which ViTs are found to be more vulnerable than CNNs. Interestingly, the proposed Sparse Patch-Fool attack, a variant of our Patch-Fool, further indicates that the perturbation density onto each patch seems to be the key factor that determines the robustness ranking between ViTs and CNNs. We believe this work can shed light on better understanding ViTs' robustness and inspire innovative defense techniques.

REFERENCES

Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018.

Ahmed Aldahdooh, Wassim Hamidouche, and Olivier Deforges. Reveal of vision transformers robustness against adversarial attacks. *arXiv preprint arXiv:2106.03734*, 2021.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*, 2018.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pp. 484–501. Springer, 2020.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021.

Josh Beal, Eric Kim, Eric Tzeng, Dong Huk Park, Andrew Zhai, and Dmitry Kislyuk. Toward transformer-based object detection. *arXiv preprint arXiv:2012.09958*, 2020.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. *arXiv preprint arXiv:2103.14586*, 2021.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pp. 213–229. Springer, 2020.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey. *arXiv preprint arXiv:1810.00069*, 2018.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pp. 15–26, 2017.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, volume 1, 2021a.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021b.

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *arXiv preprint arXiv:2104.13840*, 1(2):3, 2021.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

Francesco Croce and Matthias Hein. Sparse and imperceivable adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4724–4732, 2019.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pp. 2206–2216. PMLR, 2020.

James Diffenderfer and Bhavya Kailkhura. Multi-prize lottery ticket hypothesis: Finding accurate binary neural networks by pruning a randomly weighted network. *arXiv preprint arXiv:2103.09377*, 2021.

Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. Advertorch v0. 1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019.

Xiaoyi Dong, Dongdong Chen, Jianmin Bao, Chuan Qin, Lu Yuan, Weiming Zhang, Nenghai Yu, and Dong Chen. Greedyfool: Distortion-aware sparse adversarial attack. *arXiv preprint arXiv:2010.13773*, 2020.

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. *arXiv preprint arXiv:2107.00652*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021.

Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. *arXiv preprint arXiv:2104.01136*, 2021.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pp. 2484–2493. PMLR, 2019.

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

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pp. 2137–2146. PMLR, 2018a.

Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*, 2018b.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*, 2, 2019.

Kyungmin Kim, Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Zhicheng Yan, Peter Vajda, and Seon Joo Kim. Rethinking the self-attention in vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3071–3075, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *arXiv preprint arXiv:1807.03888*, 2018.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. *arXiv preprint arXiv:1809.03113*, 2018.

Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*, 2021a.

Xinyu Li, Yanyi Zhang, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. *arXiv preprint arXiv:2104.11746*, 2021b.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. *arXiv preprint arXiv:2108.10257*, 2021.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021a.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021b.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. *arXiv preprint arXiv:2104.02610*, 2021.

Xiaofeng Mao, Gege Qi, Yuefeng Chen, Xiaodan Li, Shaokai Ye, Yuan He, and Hui Xue. Rethinking the design principles of robust vision transformer. *arXiv preprint arXiv:2105.07926*, 2021.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Sparsefool: a few pixels make a big difference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9087–9096, 2019.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Fahad Shahbaz Khan, and Fatih Porikli. On improving adversarial transferability of vision transformers. *arXiv preprint arXiv:2106.04169*, 2021.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387. IEEE, 2016.

Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. *arXiv preprint arXiv:2105.03889*, 2021.

Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11893–11902, 2020.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 1085–1097, 2019.

Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4322–4330, 2019.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *arXiv preprint arXiv:1904.12843*, 2019.

Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of visual transformers. *arXiv preprint arXiv:2103.15670*, 2021.

Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervision. *arXiv preprint arXiv:2101.09387*, 2021.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. *arXiv preprint arXiv:2105.05633*, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8741–8750, 2021.

Zhipeng Wei, Jingjing Chen, Micah Goldblum, Zuxuan Wu, Tom Goldstein, and Yu-Gang Jiang. Towards transferable adversarial attacks on vision transformers. *arXiv preprint arXiv:2109.04176*, 2021.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2019.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *arXiv preprint arXiv:2106.14881*, 2021.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. *arXiv preprint arXiv:1910.12196*, 2019.

Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6881–6890, 2021.

A APPENDIX

A.1 PATCH-FOOL UNDER L_2 AND L_{inf} CONSTRAINTS

To enhance the imperceptibility of our Patch-Fool, we further add L_2 and L_{inf} constraints to the perturbations generated by Patch-Fool.

Setup. We add L_2 and L_{inf} constraints on the patch-wise perturbations generated by Patch-Fool with different strengths indicated by ϵ. Note that the perturbation strength ϵ of L_2-based Patch-Fool is summarized over all perturbed patches. We benchmark both the DeiT and ResNet families with different numbers of perturbed patches as shown in Tab. 8∼Tab. 11. We also add the results of vanilla Patch-Fool (unconstrained) for comparison.

Observations and analysis. From Tab. 8∼Tab. 11, we can observe that (1) the robust accuracy will be degraded more by larger perturbation strength indicated by ϵ under both L_2 and L_{inf} constraints, and (2) more importantly, DeiTs are more robust than ResNets under small ϵ, and gradually become more vulnerable than ResNets as ϵ increases, which reflects another perspective of the perturbation density. In particular, in addition to the number of pixels perturbed in one patch as analyzed in Sec. 5.5 of our main content, perturbation strength of each pixel is another perspective of perturbation density, and we consistently observe ViTs are more vulnerable than CNNs under a larger perturbation density in terms of both the number of perturbed pixels and the perturbation strength per pixel. This set of experiments further enhances the understandings about the robustness ranking between ViTs and CNNs.

Table 8: Benchmark the robustness of DeiT-Ti and ResNet-18 against Patch-Fool under the L_2 constraint with different perturbation strengths ϵ. Here vanilla Patch-Fool denotes the unconstrained Patch-Fool and "xP" indicates x patches are perturbed.

Model	**Patch Num**	**Patch-Fool under L_2 Constraint**				**Vanila Patch-Fool**
		ϵ=0.5	**ϵ=1**	**ϵ=2**	**ϵ=4**	
DeiT-Ti	1P	58.05	48.36	38.5	**17.75**	**4.01**
	2P	51.84	34.62	16.31	**4.45**	**0.12**
	3P	47.24	25.40	7.85	**1.04**	**0**
	4P	44.27	19.39	4.09	**0.02**	**0**
ResNet-18	1P	**44.91**	**32.93**	**23.4**	18.22	12.66
	2P	**33.81**	**19.55**	**9.05**	4.57	2.6
	3P	**27.56**	**11.14**	**3.85**	1.32	0.4
	4P	**22.04**	**6.69**	**1.88**	0.36	0.12

Table 9: Benchmark the robustness of DeiT-S and ResNet-50 against Patch-Fool under the L_2 constraint with different perturbation strengths ϵ. Here vanilla Patch-Fool denotes the unconstrained Patch-Fool and "xP" indicates x patches are perturbed.

Model	**Patch Num**	**Patch-Fool under L_2 Constraint**				**Vanila Patch-Fool**
		ϵ=0.5	**ϵ=1**	**ϵ=2**	**ϵ=4**	
DeiT-S	1P	67.23	56.29	41.23	**22.84**	**6.25**
	2P	60.98	40.62	17.59	**4.57**	**0.2**
	3P	56.53	30.57	7.73	**1.24**	**0**
	4P	53.53	24.60	3.77	**0.28**	**0**
ResNet-50	1P	**49.84**	**37.74**	**29.21**	24.16	24
	2P	**35.66**	**20.63**	**12.06**	7.93	6.65
	3P	**28.04**	**12.54**	**6.13**	2.92	2.56
	4P	**22.76**	**8.13**	**3.04**	1.28	1.04

Table 10: Benchmark the robustness of DeiT-Ti and ResNet-18 against Patch-Fool under the L_{inf} constraint with different perturbation strengths ϵ. Here vanilla Patch-Fool denotes the unconstrained Patch-Fool and "xP" indicates x patches are perturbed.

Model	Patch Num	Patch-Fool under L_{inf} Constraint					Vanila Patch-Fool
		ϵ=8/255	ϵ=16/255	ϵ=32/255	ϵ=64/255	ϵ=128/255	
DeiT-Ti	1P	19.75	22.48	41.39	29.61	**9.86**	**4.01**
	2P	5.85	6.25	14.7	4.45	**0.56**	**0.12**
	3P	1.88	2.64	**2.88**	**0.44**	**0**	**0**
	4P	0.52	0.72	**0.56**	**0**	**0**	**0**
ResNet-18	1P	**45.31**	**39.66**	**28.89**	**19.79**	13.1	12.66
	2P	**31.57**	**19.55**	**9.98**	**4.85**	3.04	2.6
	3P	**19.71**	**8.97**	3.57	1.08	0.68	0.4
	4P	**12.14**	**3.85**	1.12	0.2	0.24	0.12

Table 11: Benchmark the robustness of DeiT-S and ResNet-50 against Patch-Fool under the L_{inf} constraint with different perturbation strengths ϵ. Here vanilla Patch-Fool denotes the unconstrained Patch-Fool and "xP" indicates x patches are perturbed.

Model	Patch Num	Patch-Fool under L_{inf} Constraint					Vanila Patch-Fool
		ϵ=8/255	ϵ=16/255	ϵ=32/255	ϵ=64/255	ϵ=128/255	
DeiT-S	1P	66.83	60.62	45.83	30.77	**14.5**	**6.25**
	2P	54.65	34.62	15.14	5.09	**1.16**	**0.2**
	3P	40.31	17.43	**3.45**	**0.52**	**0.04**	**0**
	4P	28.33	7.25	**0.64**	**0**	**0**	**0**
ResNet-50	1P	**51.16**	**43.71**	**35.18**	**29.17**	24.64	24
	2P	**36.7**	**23.16**	**14.9**	**10.38**	7.65	6.65
	3P	**22.96**	**12.18**	5.77	4.25	2.72	2.56
	4P	**15.02**	**5.85**	2.44	1.12	0.41	1.04

A.2 Visualizing the Adversarial Examples Generated by Patch-Fool's Variants

Here we visualize the adversarial examples generated by Patch-Fool's variants in Fig. 3, including Patch-Fool with different number of perturbed patches (rows 2~3), Sparse Patch-Fool with a total of 250 perturbed pixels distributed in different number of perturbed patches (rows 4~6), and Patch-Fool under L_2 and L_{inf} constraints (rows 7~8). The corresponding robust accuracy is also annotated for reference.

Observations. From the aforementioned visualization in Fig. 3, we can observe that (1) the adversarial patches generated by Patch-Fool visually resemble and emulate natural corruptions in a small region of the original image caused by potential defects of the sensors or potential noises/damages of the optical devices (see row 2), (2) more perturbed patches lead to a lower robust accuracy and worse imperceptibility (see row 3), (3) the generated adversarial perturbations of our Sparse Patch-Fool resemble impulse noises, which improves imperceptibility while still notably degrading the robust accuracy especially when perturbing more patches (see rows 4~6), and (4) adding L_2 and L_{inf} constraints will notably improve the imperceptibility while incurring less degradation in the robust accuracy (rows 7~8).

A.3 Patch-wise Adversarial Transferability of Patch-Fool

We further discuss the patch-wise adversarial transferability of Patch-Fool, i.e., transfer the perturbations generated for attacking one specific patch to attack other patches on the same image.

Setup. Without losing generality, we generate the adversarial perturbation for the center patch with Patch-Fool which is adopted to attack all other patches on the same image and the resulting robust accuracy is annotated in Fig. 4. We average the robust accuracy at each patch location over a batch of 128 images.

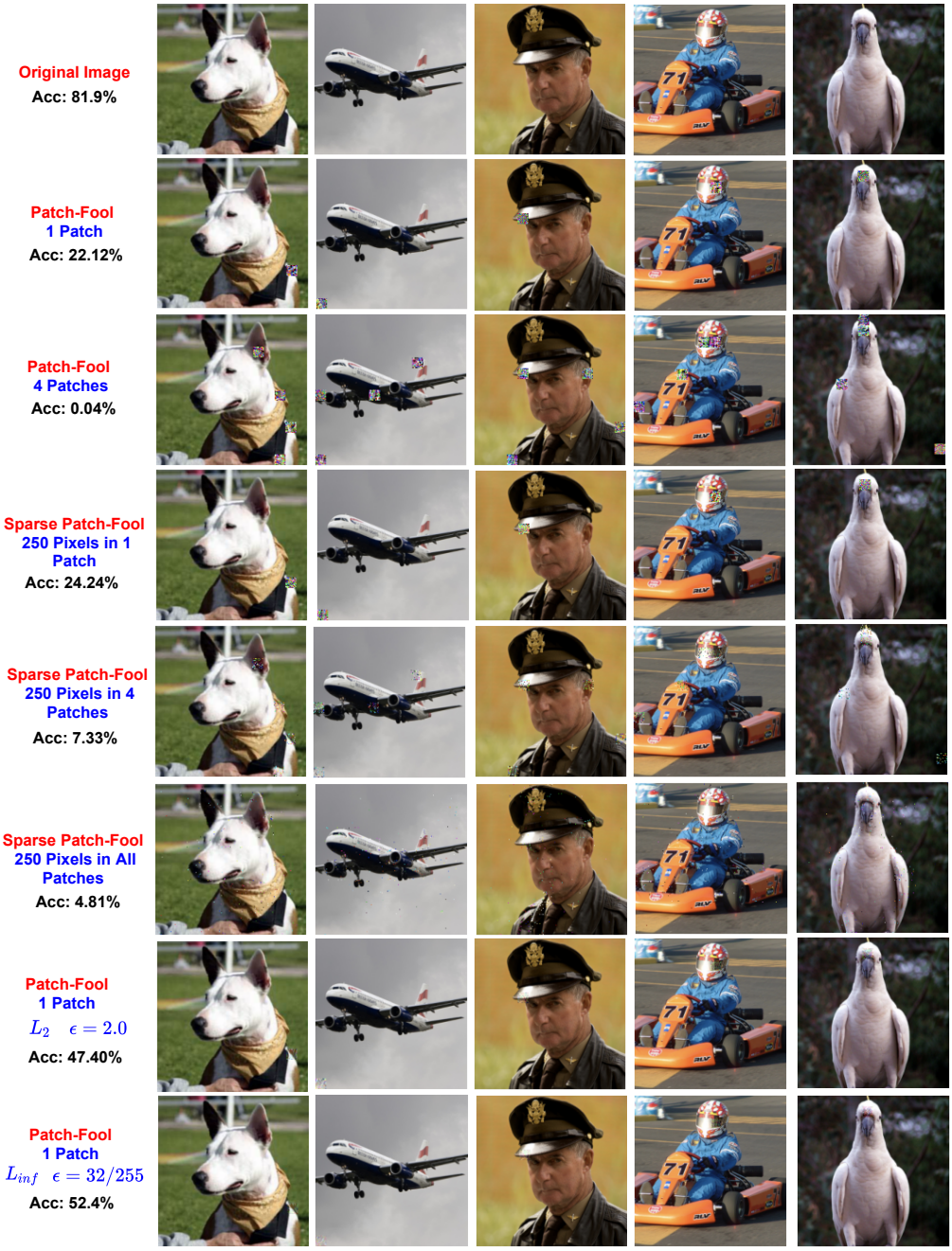

Figure 3: Visualizing the adversarial examples generated by Patch-Fool's variants, including Patch-Fool with different number of perturbed patches (rows 2~3), Sparse Patch-Fool with a total of 250 perturbed pixels distributed in different number of perturbed patches (rows 4~6), and Patch-Fool under L_2 and L_{inf} constraint (rows 7~8). Note that both the attack settings and the resulting robust accuracy are annotated.

Observations. We can observe that the adversarial patches generated by Patch-Fool can be transferred to neighboring patches with more notable accuracy degradation, while the adversarial transferability between patches far away from each other is poor.

A.4 Patch-Fool On Top of Adversarially Trained Models

To study the influence of robust training algorithms against our Patch-Fool, we further benchmark the robustness of both adversarially trained ViTs and CNNs.

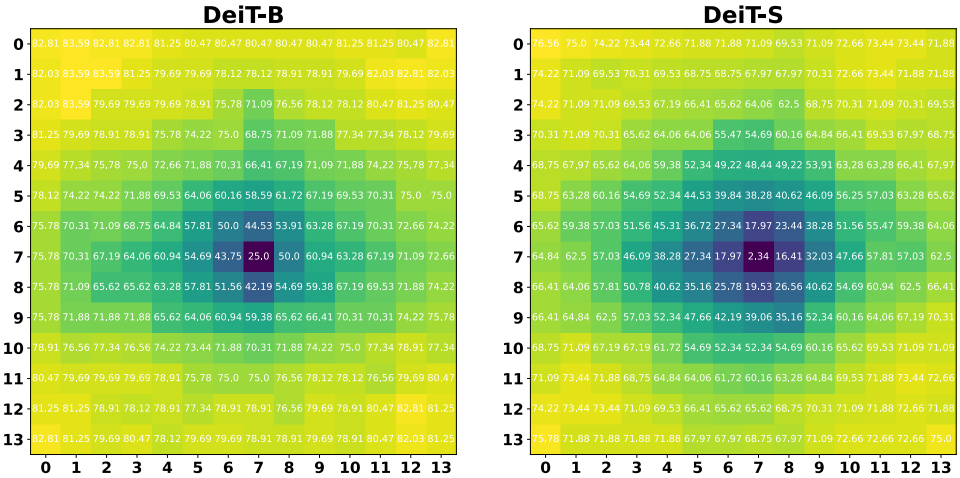

Figure 4: Visualizing the patch-wise adversarial transferability of Patch-Fool on top of DeiT-B (left) and DeiT-S (right), where the robust accuracy when perturbing each patch with the attack generated for the center patch on the same image is annotated in the figure.

Setup. We apply Fast Adversarial Training (FAT) Wong et al. (2019) with an ϵ of 2/255 and 4/255 under the L_{inf} constraint on top of both DeiT-Ti and ResNet-18 on ImageNet. We report the robust accuracy of the FAT trained models against our Patch-Fool in Tabs. 12 and 13.

Observations and analysis. From Tab. 12, we can observe that although FAT improves the robustness of both DeiT-Ti and ResNet-18 against our Patch-Fool attacks, DeiT-Ti is still more vulnerable against Patch-Fool than ResNet-18 under the same number of perturbed patches. In addition, we can observe from Tab. 13 that (1) stronger adversarial training with larger ϵ leads to better robustness against both PGD attacks and our Patch-Fool, and (2) the improvement in robust accuracy against PGD attacks is higher than the one against Patch-Fool, indicating that enhanced adversarial training schemes or other defense methods are required to robustify ViTs against our Patch-Fool, which is also our future work.

Table 12: Benchmark the robustness of adversarially trained DeiT-Ti and ResNet-18 against both PGD attacks (Madry et al., 2017) and Patch-Fool with different numbers of perturbed patches. Here "w/o" denotes the results without any adversarial training.

Model	AT	Clean Acc	PGD ϵ=0.2/255	PGD ϵ=2/255	Patch-Fool 1P	Patch-Fool 2P	Patch-Fool 3P	Patch-Fool 4P
DeiT-Ti	w/o	72.02	44.11	0.08	4.01	0.12	0	0
	Eps=4/255	66.35	64.46	41.83	**22.48**	**6.25**	**2.64**	**0.72**
ResNet-18	w/o	71.78	30.49	0	12.66	2.60	0.10	0.12
	Eps=4/255	60.13	58.57	37.18	23.48	11.58	6.09	2.69

Table 13: Evaluating the robustness of DeiT-Ti adversarially trained by different perturbation strengths against both PGD attacks (Madry et al., 2017) and Patch-Fool with different numbers of perturbed patches. Here "w/o" denotes the results without any adversarial training.

Model	AT	Clean Acc	PGD ϵ=0.2/255	PGD ϵ=2/255	Patch-Fool 1P	Patch-Fool 2P	Patch-Fool 3P	Patch-Fool 4P
DeiT-Ti	w/o	72.02	44.11	0.08	4.01	0.12	0	0
	ϵ=2/255	67.43	63.26	40.06	**19.75**	**5.85**	**1.88**	**0.52**
	ϵ=4/255	66.35	64.46	41.83	22.48	6.25	2.64	0.72.