Random Entangled Tokens for Adversarially Robust Vision Transformer

Huihui Gong[1,2], Minjing Dong[3], Siqi Ma[4], Seyit Camtepe[2], Surya Nepal[2], Chang Xu[1]
[1]The University of Sydney, [2]CSIRO Data61
[3]City University of Hong Kong, [4]The University of New South Wales
{hgon9611@uni., c.xu@}sydney.edu.au, minjdong@cityu.edu.hk
siqi.ma@unsw.edu.au, {Seyit.Camtepe, surya.nepal}@data61.csiro.au

Abstract

Vision Transformers (ViTs) have emerged as a compelling alternative to Convolutional Neural Networks (CNNs) in the realm of computer vision, showcasing tremendous potential. However, recent research has unveiled a susceptibility of ViTs to adversarial attacks, akin to their CNN counterparts. Adversarial training and randomization are two representative effective defenses for CNNs. Some researchers have attempted to apply adversarial training to ViTs and achieved comparable robustness to CNNs, while it is not easy to directly apply randomization to ViTs because of the architecture difference between CNNs and ViTs. In this paper, we delve into the structural intricacies of ViTs and propose a novel defense mechanism termed Random entangled image Transformer (ReiT), which seamlessly integrates adversarial training and randomization to bolster the adversarial robustness of ViTs. Recognizing the challenge posed by the structural disparities between ViTs and CNNs, we introduce a novel module, input-independent random entangled self-attention (II-ReSA). This module optimizes random entangled tokens that lead to "dissimilar" self-attention outputs by leveraging model parameters and the sampled random tokens, thereby synthesizing the self-attention module outputs and random entangled tokens to diminish adversarial similarity. ReiT incorporates two distinct random entangled tokens and employs dual randomization, offering an effective countermeasure against adversarial examples while ensuring comprehensive deduction guarantees. Through extensive experiments conducted on various ViT variants and benchmarks, we substantiate the superiority of our proposed method in enhancing the adversarial robustness of Vision Transformers.

1. Introduction

Vision Transformers (ViTs) and their variants have achieved state-of-the-art performance on various vision benchmarks [19, 33, 45]. Nevertheless, similar to convolutional neural networks (CNNs) [23, 44], ViTs are also vulnerable to maliciously elaborated adversarial samples [25, 27]. Even minimal perturbations that are hardly noticeable to humans can derail the predictions of high-performance ViTs. Recent studies [4–6, 21, 36, 37] have manifested that ViTs are not necessarily more robust than CNNs, often yielding similar responses to adversarial samples. Consequently, robust methods devised for CNNs can also be utilized to fortify ViTs against adversarial attacks.

Adversarial training stands out as one of the most effective defences for CNNs, which is proposed to yield robust models through re-training CNN models via adversarial samples, where Projected Gradient Descent (PGD) [35] is one of the most representative methods that iteratively generates the strongest first-order adversarial perturbations. For ViTs, researchers leveraged and evaluated the robust performance of adversarial training on ViTs [14, 15, 30, 38, 42, 48], showing that adversarial training can also effectively enhance ViTs' robustness. Beyond the training procedure, randomization is also introduced to further improve adversarial robustness [2, 8, 13, 16, 34, 41, 51]. However, these randomized defences mainly consider the robustness of CNNs, which may not work by directly applying existing methods on ViTs. For example, Dong *et al.* [16] randomized the normalization layer to reduce the adversarial transferability for CNNs, while it is not so suitable to apply this randomized defence to ViTs because ViTs primarily use the Layer Normalization [3] as the normalization layer to adapt to inconsistent input lengths.

In this paper, a novel randomized robust framework on ViT architectures is put forward to reinforce the adversarial robustness of ViTs, dubbed as *Random entangled image Transformer (ReiT)*. Firstly, we deeply analyze the structure of ViTs, especially the self-attention module, and revise the vanilla self-attention module by concatenating a random token to the input token as well as defining a new attention operation *Local SoftMax (LSM)* to compute the self-attention output. In the training stage, we add different random tokens to input tokens for teaching models adapt to the existence of random tokens. Additionally, we propose *Input-Independent Random entangled Self-Attention (II-ReSA)* for the inference stage, where we sample a random token r_1 and utilize backpropagation to calculate another random entangled token r_2 that leads to a "dissimilar" output. Note that this procedure decouples from input tokens with only a little additional computation cost. Then, the final random token that is randomly sampled from the random entangled tokens r_1 and r_2 will be concatenated to input tokens to thwart the effectiveness of adversarial attacks. Theoretically and experimentally, we prove that our proposed II-ReSA module reduces the similarity of the outputs by input tokens with r_1 and r_2, respectively, which depresses the transferability of adversarial attacks on ViTs. Before ending this section, we summarize the main contributions of this paper as below:

- We analyze the structure of the self-attention module and propose a random version of the attention module to yield the random self-attention output.
- We propose a novel adversarially robust framework, random entangled image transformer, which contains input-independent random entangled self-attention with a comprehensive theoretical guarantee to bolster the adversarial robustness of ViTs.
- We perform extensive experiments to demonstrate the superiority of our proposed method on different benchmarks and ViT variants under different adversarial attacks.

2. Related Work

2.1. Adversarial Attack

CNNs are known to be sensitive to certain elaborated perturbations, i.e., adversarial perturbations [23, 44]. Accordingly, the adversarial attack becomes a hot research field, where lots of relevant works are proposed, e.g., white-box attacks [7, 9, 22, 23, 35, 39, 44] and black-box attacks [1, 18, 26, 31, 32, 49, 52]. Hereof, we mainly discuss the white-box ones that are most relevant to this work: Szegedy *et al.* [44] presented the Fast Gradient Sign Method (FGSM) attack that utilized the sign of the input gradients to elaborate adversarial perturbations; MoosaviDezfooli *et al.* [39] as well as Carlini and Wagner [7] and regarded the solution of the generation of adversarial perturbations as an optimization objective and solved such objective to generate minimal adversarial perturbations; Dong *et al.* [17] and Madry *et al.* [35] extended the FGSM attack to iterative ones to produce the strongest first-order adversarial attack and proposed the Projected Gradient Descent (PGD) attack; Croce *et al.* [9] proposed the powerful attack framework, dubbed as AutoAttack, which integrates three white-box attacks (Auto-PGD, targeted Auto-PGD and FAB [10]) and one black-box attack (Square attack [1]).

2.2. Adversarial Robustness

Here, we only highlight two relevant robust defences: Adversarial training [35] and Randomized defence. Adversarial training is one of the most popular and effective methods to improve the adversarial robustness of CNNs. Later, Zhang *et al.* [50] added a useful regularization term to the optimization objective of training robust models that helps improve the generalization (accuracy on clean inputs) of the trained robust models; Wang *et al.* [47] proposed the misclassification-aware regularizer to boost the performance of adversarially trained models.

Randomized defence is also an effective CNN defence method against adversarial examples via adding randomness on inputs, model parameters, model components or training strategies. Cohen *et al.* [8] added Gaussian random noise to the input for improving the adversarial robustness of CNNs; Zhang and Liang [51] presented a randomized defence method via randomized discretization of the input. Araujo *et al.* [2] proposed a randomized defence method by a random training strategy. Pinot *et al.* [41] proposed a game theory-based random mixture adversarial training to improve the adversarial robustness of CNNs. Dong *et al.* [16] randomized the normalization layer to improve the adversarial robustness of CNNs. However, these randomized defences mainly aim at the robustness of CNNs, which may not work by directly applying them to ViTs. In this work, we propose a novel randomized defence method that considers the special model structure of ViTs.

2.3. Vision Transformer

The popularity of multi-head self-attention (MSA) in natural language processing (NLP) [46] sheds new light on computer vision, e.g., Dosovitskiy [19] presented the first ViT model that regards images as sentences. Later, numerous ViT variants sprouted up like mushrooms after the rain: Touvron *et al.* [45] leveraged better training strategies and token distillation to train ViT more efficiently, consuming fewer computational resources and less training time; d'Ascoli *et al.* [20] proposed the gated positional self-attention to introduce soft convolutional inductive biases for ViT; Liu *et al.* [33] proposed a shift window-based hierarchical Transformer, dubbed as Swin Transformer, incorporating the idea of convolution into ViT.

However, similar to CNNs, this MSA-based family of models are susceptible to adversarial examples [25, 27]. Many researchers compared the adversarial robustness between ViTs and CNNs: Bai *et al.* [4] investigated the robustness of ViTs under perturbation-based attacks and patch-based attacks; Bhojanapalli *et al.* [6] studied the robustness of ViTs under input perturbations as well as model perturbations; Mahmood *et al.* [36] mainly discussed the adversarial transferability of ViTs; Paul and Chen [40] evaluated the robustness of ViTs on various datasets.

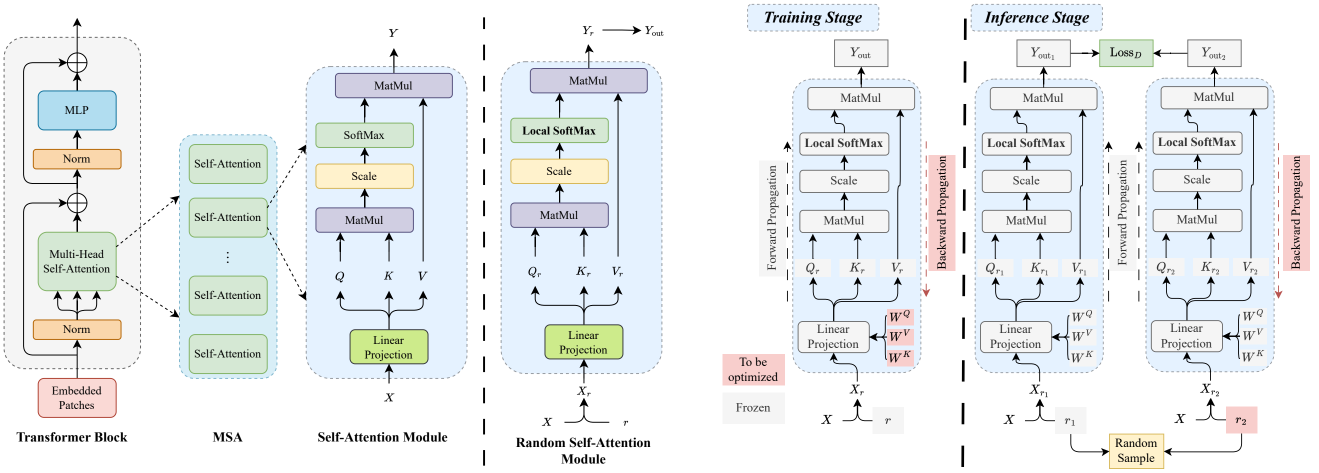

Figure 1. **Left:** original self-attention module and random self-attention module. We feed the random self-attention module with input tokens and random tokens and replace the SoftMax operation with the local SoftMax operation. In order to keep the output dimensions consistent, we truncate the extra random output. **Right:** the forward and backward propagation processes of the proposed II-ReSA module in the training and inference stages.

3. Random Entangled Self-Attention Elevates Adversarial Robustness

Given a specific target ViT model h, data pairs (x, y) and some loss function $\mathcal{L}$, adversarial examples are defined as perturbed inputs $\tilde{x} = x + \delta$ that misguide h via maximizing the following perturbations:

$$\delta = \arg\max_{\delta \in \mathcal{B}(p,\epsilon)} \mathcal{L}(h(x+\delta), y), \tag{1}$$

where $\mathcal{B}(p,\epsilon) = \{\delta : \|\delta\|_p \le \epsilon\}$ is the ℓ_p ball with radius ϵ. Considering a ViT parameter space $\mathcal{H}$ and a data space $(\mathcal{X}, \mathcal{Y})$, the objective of *adversarial training* is to minimize the expected loss over the data distribution:

$$\arg\min_{h \in \mathcal{H}} \mathbb{E}_{(x,y)\sim(\mathcal{X},\mathcal{Y})}[\mathcal{L}(h(x+\delta), y)]. \tag{2}$$

In this paper, we explore how to randomize the self-attention module that can elevate the adversarial robustness of ViTs. Specifically, we rethink the self-attention module and design an input-independent randon entangled self-attention module with thorough deduction guarantees to reduce adversarial similarity, thus boosting adversarial robustness.

3.1. Revisiting Self-Attention Module

Normally, a general transformer block [19, 46] contains normalization, multi-head self-attention (MSA), and multi-layer perception (MLP), as shown in the left subfigure of Fig. 1. Given input token $X \in \mathbb{R}^{d\times n}$ (d denotes the token number and n is the dimension of token vector), linear projection parameters $W^Q \in \mathbb{R}^{n\times m}$, $W^K \in \mathbb{R}^{n\times m}$ and $W^V \in \mathbb{R}^{n\times m}$ (usually, $m < n$), the query, key and value matrices are defined as

$$Q = XW^Q, \;\; K = XW^K, \;\; V = XW^V. \tag{3}$$

Thus, the output of the single-head self-attention module is formulated as

$$\begin{aligned} Y &= \text{SoftMax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \\ &= \text{SoftMax}\left(\frac{XW^Q(XW^K)^T}{\sqrt{d_k}}\right)XW^V, \end{aligned} \tag{4}$$

where the superscript ^{T} denotes matrix transposition; $1/\sqrt{d_k}$ represents the scaling factor. Note that the dimension of Y is $\mathbb{R}^{d\times m}$.

3.2. Random Self-Attention Module

We wonder if we can add some randomness to the output of Eq. (4) to obstruct the generation of adversarial examples. Given a random token $r \in \mathbb{R}^{1\times n}$, the input token with the random token X_r and the query matrix with the random token Q_r can be written as

$$X_r = \binom{X}{r}_{(d+1)\times n}, \;\; Q_r = X_rW^Q = \binom{XW^Q}{rW^Q}_{(d+1)\times m}. \tag{5}$$

In the same way with Q_r, we can obtain the expressions of the key matrix K_r and the value matrix V_r. Thus, the formula of $Q_rK_r^T$ can be obtained as follows:

$$\begin{aligned} Q_rK_r^T &= \binom{XW^Q}{rW^Q} \cdot \binom{XW^K}{rW^K}^T \\ &= \begin{pmatrix} XW^Q(XW^K)^T & XW^Q(rW^K)^T \\ rW^Q(XW^K)^T & rW^Q(rW^K)^T \end{pmatrix}. \end{aligned} \tag{6}$$

Here, the dimension of $Q_rK_r^T$ is $(d+1)\times(d+1)$. If we solve SoftMax for $Q_rK_r^T$ directly, it requires considerable computational effort, which may incur a heavy computational burden as the depth of the network increases. Worse still, such global SoftMax operation has an adverse impact on the self-attention calculation since the global randomization will introduce excessive randomization of self-attention that increases the difficulty of model parameter optimization. To resolve the above two challenges, we propose a novel self-attention weight operation *Local SoftMax (LSM)*, which is given by

$$\begin{aligned}&\text{LSM}\left(\frac{Q_rK_r^T}{\sqrt{d_k}}\right)\\&=\begin{pmatrix}\text{SoftMax}\left(\frac{XW^Q(XW^K)^T}{\sqrt{d_k}}\right) & XW^Q(rW^K)^T\\ rW^Q(XW^K)^T & rW^Q(rW^K)^T\end{pmatrix},\end{aligned}\tag{7}$$

where the matrix dimension is $(d+1)\times(d+1)$. Hereof, we only compute the SoftMax output of the part with the original input token, which is consistent with the original self-attention. Thus, the random self-attention output is defined as

$$\begin{aligned}&Y_r=\text{LSM}\left(\frac{Q_rK_r^T}{\sqrt{d_k}}\right)V_r\\&=\begin{pmatrix}\text{SoftMax}\left(\frac{XW^Q(XW^K)^T}{\sqrt{d_k}}\right) & XW^Q(rW^K)^T\\ rW^Q(XW^K)^T & rW^Q(rW^K)^T\end{pmatrix}\cdot\begin{pmatrix}XW^V\\rW^V\end{pmatrix}\\&=\begin{pmatrix}\text{SoftMax}\left(\frac{XW^Q(XW^K)^T}{\sqrt{d_k}}\right)XW^V+XW^Q(rW^K)^TrW^V\\ rW^Q(XW^K)^TXW^V+rW^Q(rW^K)^TrW^V\end{pmatrix},\end{aligned}\tag{8}$$

where the dimension of Y_r is $(d+1)\times m$; let $Y=\text{SoftMax}\left(\frac{XW^Q(XW^K)^T}{\sqrt{d_k}}\right)XW^V$, which is the output of the conventional SoftMax operation; let $R=XW^Q(rW^K)^TrW^V$, which denotes the additive random output; let $r_{\text{out}}=rW^Q(XW^K)^TXW^V+rW^Q(rW^K)^TrW^V$, which represents the extra random output. Note that r_{out} will be truncated so as to keep the dimensions of the input token and output self-attention result constant. Hence, the final output is $Y_{\text{out}}=Y+R$, which incorporates the self-attention output and the output of random tokens (cf., the left subfigure in Fig. 1).

3.3. Input-Independent Random Entangled Self-Attention Module

To further reinforce the robustness of ViTs, we improve the random entangled self-attention module by introducing binary random noises. Suppose we have two random tokens, r_1 and r_2, we can obtain two additive random noise outputs, R_1 and R_2, through Eq. (8). Our target is to minimize the adversarial similarity of the outputs of two randomized self-attention, i.e., $Y_{\text{out},1}=Y+R_1$ and $Y_{\text{out},2}=Y+R_2$, which is equivalent to maximizing the difference function $D(\cdot,\cdot)$ of the gradients w.r.t. X of $Y_{\text{out},1}$ and $Y_{\text{out},2}$:

$$\max D\left(\frac{\partial(Y+R_1)}{\partial X},\frac{\partial(Y+R_2)}{\partial X}\right).\tag{9}$$

Here, we use cosine similarity $[\mathbf{A}\cdot\mathbf{B}/(\|\mathbf{A}\|\cdot\|\mathbf{B}\|)]$ to represent the distance function $D(\cdot,\cdot)$. The goal of maximizing the distance converts to minimize the cosine similarity of the gradients w.r.t. X of $Y_{\text{out},1}$ and $Y_{\text{out},2}$. Besides, the denominator of the cosine similarity formula (the product of the modules of the two matrices/vectors) will not affect the optimization. Thus, the cosine similarity can be degraded into *dot multiplication*, which greatly reduces the computational cost. The degraded object function can be written as

$$\begin{aligned}&D\left(\frac{\partial(Y+R_1)}{\partial X},\frac{\partial(Y+R_2)}{\partial X}\right)\\&=\left(\frac{\partial Y}{\partial X}+\frac{\partial R_1}{\partial X}\right)\cdot\left(\frac{\partial Y}{\partial X}+\frac{\partial R_2}{\partial X}\right)\\&=\left(\frac{\partial Y}{\partial X}\right)^2+\frac{\partial Y}{\partial X}\frac{\partial R_1}{\partial X}+\left(\frac{\partial Y}{\partial X}+\frac{\partial R_1}{\partial X}\right)\frac{\partial R_2}{\partial X}.\end{aligned}\tag{10}$$

Given a certain r_1, we aim to find another random entangled token r_2 that leads to dissimilar outputs by maximizing Eq. (9) to derail the generation of effective adversarial perturbations. Note that $\partial Y/\partial X$ is the partial derivative w.r.t. X of Y, thus independent of random entangled tokens r_1 and r_2; $\partial R_1/\partial X$ is the partial derivative w.r.t. X of R_1, thus independent of r_2. Consequently, the objective of Eq. (9) herein is equivalent to minimizing Eq. (10) w.r.t. r_2, which can be written as

$$\begin{aligned}&\min_{r_2}\left[\left(\frac{\partial Y}{\partial X}\right)^2+\frac{\partial Y}{\partial X}\frac{\partial R_1}{\partial X}+\left(\frac{\partial Y}{\partial X}+\frac{\partial R_1}{\partial X}\right)\frac{\partial R_2}{\partial X}\right]\\&\iff\min_{r_2}\left(\frac{\partial Y}{\partial X}+\frac{\partial R_1}{\partial X}\right)\frac{\partial R_2}{\partial X}.\end{aligned}\tag{11}$$

According to Eq. (8) and Eq. (11), $\partial Y/\partial X$ is a function of X; $\partial R_1/\partial X$ and $\partial R_2/\partial X$ are both not the function of X. To decouple from X for more efficient optimization of r_2, we omit the item $\partial Y/\partial X$. Therefore, Eq. (11) can be approximated as

$$\min_{r_2}\frac{\partial R_1}{\partial X}\frac{\partial R_2}{\partial X}.\tag{12}$$

Here, the object function is denoted as $f(r_2)$. Thus, the above optimization is independent of X, i.e., input token. Note that one can optimize r_2 only with self-attention linear projection parameters (W^Q, W^K, and W^V) and r_1, which is a local operation and entails very little computation cost. Due to the fact that r_2 is derived from r_1, We refer to these two random tokens as random entangled token pairs or simply random entangled tokens. Because the proposed self-attention is independent of the input token and involves two random entangled tokens, we term it *Input-Independent Random entangled Self-Attention (II-ReSA)*. For the deduction details of optimizing r_2 via Eq. (12), please refer to the supplementary material.

4. Random Entangled Image Transformer

In this section, we mainly integrate the proposed II-ReSA in Sec. 3 into adversarial training. Based on the thorough analysis of reducing adversarial similarity of II-ReSA, we put forward a new adversarially robust framework, denoted as *Random entangled image Transformer (ReiT)*, to boost the adversarial robustness of ViTs.

During the training stage, we directly concatenate small random tokens that follow a certain distribution (such as a Gaussian distribution) to input tokens. This has two significant benefits: 1) adversarial training with additive random noises enables the trained robust models to adapt to the existence of random tokens, thereby avoiding clean accuracy

```
Algorithm 1: ReiT algorithm
Input  : input token $X$, self-attention linear
         projection parameters $W^Q$ ,$W^K$ and $W^V$,
         random intensity $s$, random step size $\alpha$,
         random steps $\tau$
Output: attention results $Y_{\text{out}}$
Function ReiT_train($s$,$X$,$W^Q$,$W^K$,$W^V$):
    Sample $r$ from $\mathcal{N}(0,1)$
    Obtain $X_r = \text{cat}(X, s \cdot r)$
    Compute $Y$ and $R$ via Eq. (8)
    Compute $Y_{\text{out}} = Y + R$
    return $Y_{\text{out}}$
End Function
Function ReiT_test($s$,$\alpha$,$\tau$,$X$,$W^Q$,$W^K$,$W^V$):
    Sample $r_1$ from $\mathcal{N}(0,1)$
    Sample $r_2^0$ from $\mathcal{N}(0,1)$ or let $r_2^0 = r_1$
    Obtain $X_{r,1} = \text{cat}(X, s \cdot r_1)$
    Compute $R_1$ via Eq. (8)
    for $t \leftarrow 0$ to $\tau - 1$ do
        Obtain $X_{r,2} = \text{cat}(X, s \cdot r_2^t)$
        Compute $R_2^t$ via Eq. (8)
        Compute similarity between $R_1$ and $R_2^t$ via
          Eq. (12)
        Update $r_2^{t+1}$ via Eq. (13)
    end
    Randomly Sample $r$ from $r_1$ and $r_2^\tau$
    Obtain $X_r = \text{cat}(X, s \cdot r)$
    Compute $Y$ and $R$ via Eq. (8)
    Compute $Y_{\text{out}} = Y + R$
    return $Y_{\text{out}}$
End Function
```

degradation; 2) such random tokens offer randomness that compromises the effectiveness of the generated adversarial perturbations.

In the test stage, we generate random tokens r_1 from the same distribution as that in the training stage and utilize r_1 and self-attention linear projection parameters to optimize the random entangled tokens r_2 by minimizing the similarity of the output of II-ReSA. Finally, we randomly sample one from the random entangled tokens r_1 and r_2 to further provide randomness for adversarial robustness. Such random sampling can provide a large random space that hampers adversarial perturbation searching. For instance, a ViT model with 12 blocks and 8 heads in each block has the random sampling space of $(2^{12})^8 = 2^{96}$.

Iterative optimization. To improve the effectiveness of randomization, we propose to iteratively optimize r_2 via the gradient sign of Eq. (12), which is an effective and popular optimization method inspired by the idea of PGD [35]. Moreover, in order to stabilize the performance of robust models, r_2 is required to follow the same distribution as r_1. Therefore, we normalize r_2 with the same mean and standard deviation as r_1. Thus, the optimization function of r_2 is formulated as

$$r_2^{t+1} = s \cdot \text{Norm}\left(r_2^t - \alpha \cdot \text{Sign}(\nabla f(r_2^t))\right), \tag{13}$$

where $t = 0, 1, \cdots, \tau$ and τ is the total number of steps; r_2^0 can be set as r_1 or another random token under the same distribution as r_1; $\nabla f(r_2)$ is the gradient w.r.t. r_2 of $f(r_2)$; α denotes the step size; $\text{Norm}(\cdot)$ is the normalization function; s is the intensity of randomness. To put it in a nutshell, Algorithm 1 and the right subfigure in Fig. 1 summarize our proposed ReiT framework, where we only show the algorithm for the randomized self-attention module in the training and inference stages. Other modules are the same as vanilla architectures, and thus omitted.

5. Experiment

5.1. Experimental Setup

Datasets. Due to the time-consuming overhead of adversarial training, small datasets are still popular for adversarial study [11]. Hereof, we adopt benchmarks of CIFAR-10 and CIFAR-100 [29]. Both datasets consist of 50,000 training samples and 10,000 test samples, all of which are 32x32 pixels. CIFAR-10 comprises 10 categories, while CIFAR-100 encompasses 100 categories. Besides, we also conduct experiments on big datasets, e.g., ImageNet-1K (ImageNet) [12] and ImageNette [24]. ImageNet is a dataset comprising 1,000 categories with 1.2 million training examples and 50,000 test examples, all of which are 224x224 pixels in size. ImageNette is a subset of 10 classes from ImageNet, which contains about 13,000 training images and 500 testing images.

Models. We consider three different kinds of ViT architectures in our experiments: vanilla ViT [19], DeiT [45], and Swin Transformer (Swin) [33]. For CIFAR-10 and CIFAR-100, the patch size for ViT and DeiT is set as 4. Besides, the patch size and window size for Swin is set as 2 and 4, respectively. For ImageNet and ImageNette, the patch size for ViT and DeiT is set as 16. The patch size and window size for Swin are set as 4 and 7, respectively. Due to the computational resource limit, we use the small and tiny versions of ViTs to perform experiments, i.e., ViT-S, ViT-T, DeiT-S, DeiT-T, Swin-S, and Swin-T.

Training settings. We follow previous works' experimental setting [38, 42, 48] and initialize the network with pre-trained parameters provided by [43]. For CIFAR-10, CIFAR-100 and ImageNette, we train robust models with different training methods: standard (NAT), one-step fast gradient sign method (FGSM) [23], multi-step projected gradient descent (PGD) [35], and TRADES with $\beta = 6$ in [50]. All models are trained for 40 epochs. In the adversarial setting, we set the maximum perturbation to 8/255, the step size to 2/255, and the step number for multi-step ad-

Model	Method	CIFAR-10							CIFAR-100						
		NAT	FGSM	MIFGSM	PGD	DeepFool	CW	AA	NAT	FGSM	MIFGSM	PGD	DeepFool	CW	AA
ViT-S	NAT	**96.57**	10.20	0.12	0.00	0.73	71.08	0.00	**85.76**	7.24	0.58	0.00	2.97	49.42	0.00
	ReiT	96.27	**13.99**	**0.17**	0.00	**14.05**	**83.90**	**0.32**	84.02	**8.80**	**0.69**	0.00	**27.00**	**74.13**	**1.45**
	FGSM	**86.63**	53.17	51.26	45.87	0.96	72.55	42.81	47.62	23.21	20.38	19.13	3.37	41.65	13.36
	ReiT	86.21	**56.17**	**54.19**	**46.43**	**25.55**	**76.69**	**45.16**	**57.67**	**28.30**	**26.33**	**24.13**	**20.54**	**57.80**	**26.56**
	PGD	**86.59**	54.99	53.38	51.51	0.92	76.66	47.65	59.17	31.59	30.27	28.78	1.87	54.13	21.24
	ReiT	86.55	**56.82**	**54.93**	**52.86**	**26.60**	**79.43**	**49.70**	**60.16**	**33.83**	**31.06**	**30.61**	**21.02**	**60.26**	**28.92**
	TRADES	83.04	55.88	55.52	54.35	1.14	79.86	50.47	**63.32**	34.48	33.18	31.56	1.35	57.70	26.90
	ReiT	**84.47**	**57.20**	**56.67**	**54.95**	**21.58**	**80.27**	**51.67**	62.28	**35.36**	**34.13**	**34.06**	**20.72**	**61.46**	**31.48**
ViT-T	NAT	**95.20**	5.65	0.00	0.00	0.84	62.55	0.00	**80.78**	3.66	0.07	0.00	2.61	38.79	0.00
	ReiT	94.55	**5.79**	**0.21**	0.00	**21.50**	**78.99**	**0.50**	79.61	**4.28**	**0.14**	0.00	**28.78**	**68.49**	**2.01**
	FGSM	**79.96**	51.48	49.58	43.91	1.56	76.29	42.79	**58.78**	27.88	25.96	24.07	2.00	52.95	20.68
	ReiT	79.27	**52.32**	**50.13**	**45.57**	**26.11**	**78.27**	**47.66**	58.25	**28.67**	**25.99**	**24.87**	**18.51**	**55.06**	**24.46**
	PGD	78.34	52.59	50.27	46.37	1.83	74.40	43.16	**57.74**	28.36	27.11	25.47	1.60	52.67	21.03
	ReiT	**78.38**	**54.67**	**51.30**	**47.71**	**26.47**	**76.77**	**47.55**	57.09	**28.54**	**27.43**	**26.82**	**19.23**	**54.32**	**26.20**
	TRADES	**79.30**	53.57	51.39	50.49	1.43	75.78	46.20	**56.81**	30.07	29.39	28.86	2.06	51.46	23.68
	ReiT	78.83	**54.84**	**52.44**	**50.97**	**22.67**	**76.86**	**48.01**	56.14	**31.12**	**29.99**	**29.57**	**19.97**	**55.12**	**28.44**
DeiT-S	NAT	**96.21**	4.07	0.04	0.00	0.50	72.32	0.00	**84.32**	4.02	0.01	0.00	2.31	47.83	0.00
	ReiT	96.00	**9.10**	**0.35**	**0.01**	**13.02**	**84.44**	**0.31**	83.19	**4.48**	**0.27**	**0.05**	**25.03**	**71.08**	**1.23**
	FGSM	**85.78**	52.52	50.36	45.09	1.68	81.83	43.42	**58.38**	29.58	28.07	26.36	1.74	53.41	21.98
	ReiT	85.65	**54.11**	**51.82**	**46.47**	**26.98**	**85.64**	**47.28**	58.26	**30.81**	**29.39**	**27.34**	**16.67**	**54.30**	**27.85**
	PGD	**85.54**	53.30	51.17	50.78	1.63	81.70	46.87	**57.80**	30.57	29.60	28.08	1.72	53.14	23.08
	ReiT	84.44	**55.06**	**52.23**	**52.58**	**27.67**	**84.52**	**50.69**	56.55	**31.67**	**30.40**	**29.88**	**18.79**	**56.45**	**28.22**
	TRADES	**84.00**	56.99	55.41	53.09	1.35	80.51	49.13	**61.12**	32.77	32.01	30.81	1.58	55.66	26.04
	ReiT	83.44	**57.32**	**55.81**	**53.84**	**19.53**	**82.59**	**55.03**	60.58	**34.33**	**33.71**	**31.81**	**20.99**	**57.46**	**30.31**
DeiT-T	NAT	**95.10**	2.98	0.00	0.00	0.85	64.12	0.00	**80.20**	2.60	0.10	0.00	3.41	40.54	0.00
	ReiT	94.70	**3.47**	**0.02**	0.00	**19.22**	**79.58**	**0.42**	79.37	**2.74**	**0.12**	0.00	**26.29**	**67.53**	**1.65**
	FGSM	**81.52**	51.15	50.29	44.54	1.21	77.71	43.33	**57.83**	28.58	27.41	26.01	1.84	52.31	21.65
	ReiT	81.02	**52.11**	**50.45**	**45.53**	**25.80**	**80.92**	**48.51**	56.87	**29.72**	**28.29**	**27.47**	**18.95**	**55.03**	**26.60**
	PGD	**82.03**	52.55	50.89	49.33	1.04	78.42	45.54	**50.82**	29.75	28.33	26.83	0.51	45.58	21.95
	ReiT	80.73	**52.89**	**51.23**	**50.07**	**26.25**	**80.85**	**50.39**	50.62	**30.72**	**29.52**	**27.39**	**17.88**	**47.80**	**26.21**
	TRADES	**80.41**	53.76	52.42	51.24	1.12	76.70	47.28	57.15	30.51	28.85	28.23	1.68	51.88	23.18
	ReiT	80.33	**54.05**	**52.81**	**51.72**	**22.50**	**80.26**	**53.10**	**58.60**	**31.27**	**29.68**	**28.75**	**20.22**	**56.44**	**28.29**
Swin-S	NAT	**94.29**	9.41	0.01	0.00	1.12	55.70	0.00	**78.62**	5.44	0.13	0.00	4.13	36.72	0.00
	ReiT	93.83	**13.11**	**0.02**	0.00	**21.27**	**74.19**	**0.79**	78.05	**5.60**	**0.13**	**0.01**	**28.48**	**66.16**	**1.90**
	FGSM	67.07	42.01	40.90	39.46	0.24	63.10	35.41	**47.78**	23.37	22.41	21.30	0.51	42.78	17.57
	ReiT	**71.18**	**44.41**	**42.81**	**41.33**	**24.38**	**71.06**	**44.03**	47.61	**23.76**	**22.67**	**21.79**	**17.66**	**47.60**	**23.72**
	PGD	**70.73**	44.06	43.14	42.03	0.32	66.35	37.64	44.28	22.81	22.33	21.73	0.63	40.33	17.66
	ReiT	69.21	**45.65**	**44.02**	**43.89**	**24.10**	**68.52**	**44.00**	**45.80**	**23.77**	**22.91**	**22.34**	**16.75**	**45.85**	**25.43**
	TRADES	72.04	47.24	46.19	45.11	0.55	71.82	41.21	**55.39**	27.58	26.96	26.28	0.61	49.85	21.76
	ReiT	**73.30**	**48.16**	**46.98**	**45.38**	**23.99**	**74.49**	**48.07**	53.71	**27.98**	**27.33**	**26.71**	**20.37**	**53.89**	**25.94**
Swin-T	NAT	**93.40**	6.26	0.00	0.00	1.08	48.92	0.00	**77.16**	**4.22**	0.03	0.00	4.60	30.39	0.00
	ReiT	93.24	**7.01**	0.00	0.00	**20.92**	**70.93**	**0.71**	76.03	4.13	**0.06**	0.00	**28.78**	**63.18**	**2.18**
	FGSM	61.76	39.14	37.92	37.11	0.42	57.80	32.85	43.38	22.14	21.58	21.04	0.47	39.25	17.22
	ReiT	**69.01**	**43.35**	**41.73**	**40.16**	**25.44**	**68.68**	**38.67**	**46.21**	**22.89**	**21.90**	**21.20**	**17.97**	**46.25**	**21.04**
	PGD	60.23	41.76	38.51	38.28	0.53	49.77	35.72	**41.08**	20.65	20.21	19.92	0.45	37.07	18.59
	ReiT	**64.01**	**43.05**	**39.91**	**39.39**	**23.21**	**64.15**	**41.63**	39.59	**21.50**	**21.14**	**20.70**	**14.26**	**39.76**	**23.38**
	TRADES	70.91	43.89	43.14	42.42	0.69	66.80	38.03	**50.82**	24.75	24.33	23.83	0.51	45.58	19.01
	ReiT	**71.17**	**44.55**	**44.18**	**43.03**	**23.93**	**71.35**	**45.12**	49.71	**25.96**	**24.59**	**24.22**	**18.01**	**47.62**	**24.88**

Table 1. Robust experimental results (%) of ReiT with different ViT variants under different adversarial training methods on CIFAR-10 and CIFAR-100 benchmark datasets. Here, 'S' denotes small and 'T' denotes tiny. The best results are stressed in **BOLD**.

versarial training to 10. We utilize the SGD optimizer with a momentum of 0.9, a weight decay of 1e-4, and an initial learning rate of 0.1. The learning rate follows a piecewise decay schedule with a reduction factor of 0.1 at the 20th and 30th epochs. For ImageNet, we train robust models with PGD and TRADES over 10 epochs, applying a maximum perturbation of 4/255, a step size of 1/255, and 5 PGD steps. We utilize the SGD optimizer with a momentum of 0.9, a weight decay of 1e-4, and an initial learning rate of 0.01. The learning rate undergoes piecewise decay, reducing by a factor of 0.1 at the 5th and 8th epochs. Additionally, we sample random tokens from the standard Gaussian distribution and set the randomness intensity of all models as 0.1 for our method. The ablation study of it will be discussed in Sec. 5.3. All the random tokens are sampled/normalized from/to the standard normal distribution.

Evaluation settings. For the test stage, we evaluate the trained models under different adversarial attacks: natural

ImageNette	Method	NAT	FGSM	MIFGSM	PGD	CW
ViT-T	NAT	**95.40**	5.00	0.20	0.00	59.40
	ReiT	94.00	**6.80**	**0.40**	0.00	**78.20**
	PGD	**79.60**	54.60	54.00	51.80	77.40
	ReiT	78.80	**56.30**	**55.00**	**53.60**	**78.10**
	TRADES	**84.60**	63.60	62.20	60.60	83.20
	ReiT	84.00	**64.40**	**63.70**	**62.30**	**83.60**
DeiT-T	NAT	**94.80**	2.60	0.20	0.00	63.00
	ReiT	94.60	**4.80**	0.20	0.00	**78.40**
	PGD	88.80	68.00	65.60	63.80	88.40
	ReiT	**89.00**	**69.80**	**67.60**	**65.00**	**89.20**
	TRADES	**88.80**	68.80	67.60	66.00	86.60
	ReiT	87.80	**68.90**	**68.10**	**66.50**	**86.80**
Swin-T	NAT	**97.20**	11.40	0.00	0.00	66.60
	ReiT	97.00	**12.20**	**0.20**	0.00	**75.40**
	PGD	30.00	27.20	27.20	27.60	29.60
	ReiT	**81.60**	**60.60**	**60.20**	**58.40**	**81.00**
	TRADES	**81.40**	60.60	60.00	59.00	80.00
	ReiT	81.00	**62.90**	**62.10**	**60.90**	**80.40**
ImageNet	Method	NAT	FGSM	MIFGSM	PGD	CW
DeiT-T	PGD	**49.63**	23.40	22.73	21.68	42.65
	ReiT	48.79	**25.12**	**23.42**	**22.15**	**45.49**
	TRADES	**50.17**	23.37	22.82	22.90	42.65
	ReiT	49.74	**26.10**	**24.54**	**23.55**	**46.16**

Table 2. Robust experimental results (%) of ReiT with different ViT variants under different adversarial training methods on ImageNette and ImageNet benchmark datasets. The best results are stressed in **BOLD**.

(NAT), FGSM, MIFGSM [17] with 5 steps, PGD [35] with 10 steps, DeepFool [39] with 50 steps and an overshoot of 0.02, CW [7] with 1,000 steps and a learning rate of 0.01, AutoAttack (AA) [9]. We use the torchattacks library [28] for our evaluation experiments.

5.2. Main Robustness Results

Results on CIFAR-10 and CIFAR-100. The robust experimental results on small datasets (CIFAR-10 and CIFAR-100) are shown in Tab. 1. From the table, it is evident that our robust method outperforms the baselines with different training methods under diverse adversarial settings. For example, ReiT achieves better adversarial accuracy for the seen adversarial attacks: vanilla PGD-trained ViT-S model achieves 51.51% adversarial accuracy under the PGD attack, while our ReiT PGD-trained ViT-S model achieves 52.86% (+1.35%) adversarial accuracy under the PGD attack. We primarily attribute this superiority to the negative interference of our proposed II-ReSA module on the acquisition of adversarial perturbations. Besides, we also find that our randomized method is effective in defending against unseen white-box or black-box attacks, like DeepFool, CW, and AutoAttack, which includes a black-box attack, Square Attack [1]. This means that our random entangled method plays a beneficial role in defending against unseen attacks. Additionally, ReiT can alleviate the phenomenon of (catastrophic) overfitting to some extent, e.g., the vanilla FGSM-trained ViT-S and PGD-trained DeiT-T models overfit (compared with other models, their adversarial performance plummets by a large margin), while the corresponding ReiT models show better adversarial performance. This superiority may result from the beneficial effect that helps the model escape from local optima. Admittedly, our randomized methods usually achieve a little lower natural accuracy than that of vanilla methods, which is probably because the II-ReSA module will introduce noises to the inference process, which is advantageous to robust inference but disadvantageous to natural inference.

Model	Method	NAT	PGD-20	PGD-100	AA
ViT-S	vanilla PGD	**86.59**	51.27	51.19	47.65
	ARD & PRM	85.18	51.73	51.57	47.98
	ReiT	86.55	**52.68**	**52.61**	**49.70**
DeiT-T	vanilla PGD	**82.03**	48.83	48.67	45.54
	ARD & PRM	81.18	49.37	49.13	46.83
	ReiT	80.73	**49.92**	**49.73**	**50.39**

Table 3. Results (%) of comparison with robust ViT methods on CIFAR-10 dataset. The best results are stressed in **BOLD**.

Results on ImageNette and ImageNet. We also provide more experiments on ImageNette and ImageNet to evaluate the performance of ReiT on large datasets. We train three ViT models (ViT-T, DeiT-T, and Swin-T) on ImageNette as well as one model (DeiT-T) on ImageNet, and evaluate the trained models with NAT, FGSM, MIFGSM, PGD, and CW attacks. The results are shown in Tab. 2, from which it is evident that our proposed ReiT achieves better adversarial robustness than that of the baseline methods.

Combination with other robust methods. Although the above results are mainly based on standard adversarial training, we further combine our proposed ReiT with other stronger robust training methods. Here, we adopt TRADES [50] due to its excellent performance on [11]. The training and evaluation settings are the same as those of the standard adversarial training. We conduct these experiments on all the models and methods for a comprehensive evaluation. The results are shown in Tabs. 1 and 2, which demonstrates that our proposed ReiT can further improve the performance of existing defense methods.

Comparison with robust ViTs. To better illustrate the superiority of our proposed ReiT, We compare our method with the state-of-the-art robust ViTs in [38] that proposed two useful methods, i.e., Attention Random Dropping (ARD) and Perturbation Random Masking (PRM), to boost the robustness of ViTs. Because the ARD & RPM method achieves the best performance in their work, we directly use this method as our comparison. Specifically, we retain their models under our training settings so as to compare our method with those methods fairly. For the inference, we evaluate all the methods under PGD-20 (20 steps PGD attack with the maximum perturbation of 8/255), PGD-100 (100 steps PGD attack with the maximum perturbation of 8/255), and AutoAttack (AA) on CIFAR-10. The results are

Ablation Item	Component				Accuracy			
	Bin	Norm	Iter	Std	NAT	PGD	DeepFool	CW
Baseline	-	-	-	-	**82.03**	49.33	1.04	78.42
ReiT	✓	✓	✓	0.1	80.73	**50.07**	**26.25**	**80.85**
Bin	✗	-	-	-	80.83	49.34	18.46	79.63
Norm	✓	✗	✓	0.1	80.81	46.65	24.12	78.68
Iter	✓	✓	✗	0.1	80.84	49.47	24.18	80.64
Std	✓	✓	✓	0.01	81.84	49.38	22.59	78.86
	✓	✓	✓	0.05	80.97	49.53	23.00	79.31
	✓	✓	✓	0.2	78.86	48.24	26.18	78.80
	✓	✓	✓	0.4	60.24	38.13	25.53	60.28

Table 4. Component ablation study and hyperparameter tuning results (%) of our proposed method for the DeiT-T model on CIFAR-10 dataset. The best results are stressed in **BOLD**.

displayed in Tab. 3, from which we find that our ReiT can achieve better performance under different adversarial settings. For instance, under the AutoAttack attack for the DeiT-T model, our method achieves a 4.85% increment compared with the vanilla robust method, while the ARD & RPM method only achieves a 1.29% increment. For more experiments, please refer to the supplementary material.

5.3. Ablation Study

Component ablation. As discussed in Eqs. (8) and (13), there are three primary components/operations for the proposed ReiT that will affect the effect of the II-ReSA module: 1) random entangled tokens r_1 and r_2; 2) iterative optimization of r_2; 3) normalization of the optimized r_2. The results are shown in Tab. 4, from which we find that without the random entangled tokens (here, we just use the random token r_1 and do not optimize r_2 any more), the robust performance has no significant improvement, which is because the different random tokens r of different forward propagation do not necessarily lead to large differences of the output Y_{out}. If the outputs of two forward propagations are similar, the attackers can use the last information to generate effective adversarial examples. For the normalization ablation study, the robust performance drop without the normalization operation, maybe because the training stage uses the standard normal distribution to generate random token r, but in the inference stage, the unnormalized r_2 does not follow the standard normal distribution, which hurts the performance of robust models. Besides, the non-iterative optimization of r_2 can also improve the robust performance of robust models, but it is inferior to the iterative ReiT.

Hyperparameter tuning. Additionally, the intensity of randomness s in Eq. (13) is a crucial hyperparameter in our proposed method. To illustrate its effect and select an appropriate value, we conduct a set of hyperparameter tuning experiments on CIFAR-10. The results are exhibited in Tab. 4, which shows that for smaller s, the natural accuracy is higher than that of our 0.1-intensity ReiT but lower than that of the vanilla robust models, while the robust accuracy is lower than of our 0.1-intensity ReiT but higher than that of the vanilla robust models. However, for too large s (like 0.2 and 0.4, especially 0.4), the performances on clean input and adversarial input will both decline by a large margin, which probably results from the fact that too strong random signal interferes with the model to grab useful information from the input.

Model	Method	Run Time	NAT	PGD	DeepFool	CW	AA
ViT-S	vanilla	-	86.59±0.00	51.51±0.00	0.92±0.00	76.66±0.00	47.65±0.00
	ReiT	1	86.55±0.00	52.86±0.00	26.60±0.00	79.43±0.00	49.70±0.00
		5	85.95±0.64	52.75±0.99	26.05±0.63	79.63±0.41	50.08±0.68
		10	85.98±0.72	52.74±1.00	26.03±0.73	79.20±0.75	50.01±0.99
DieT-T	vanilla	-	82.03±0.00	49.33±0.00	1.04±0.00	78.42±0.00	45.54±0.00
	ReiT	1	80.73±0.00	50.07±0.00	26.25±0.00	80.85±0.00	50.39±0.00
		5	81.38±0.59	50.33±0.76	26.54±0.74	80.66±0.53	50.59±1.20
		10	81.23±0.66	50.31±0.79	26.42±0.78	80.93±0.74	50.41±0.97

Table 5. Accuracy means and standard deviations (%) of our proposed method on CIFAR-10 dataset under multiple executions.

Multiple execution. Our robust method includes a random module (II-ReSA), whose performance may be influenced by different execution environments. To eliminate the environmental variance, we run our method multiple times. The means and standard deviations are shown in Tab. 5, in which we used the same machine random seed and different random tokens r_1 for the multi-execution results. Note that no matter how many times we run the vanilla robust model under the same machine random seed, the performance will be the same. Thus, the accuracy standard deviations of it are zero. Besides, the accuracy standard deviations of the one-time execution ReiT are also zero. For the results of the multi-execution (five-time and ten-time) ReiT, it is evident that the robust performance stabilizes around certain values, which are higher than those of the vanilla robust models.

6. Conclusion

In this paper, leveraging the distinctive architecture of the self-attention module in ViTs, we introduced an innovative input-independent random entangled self-attention (II-ReSA) module, which enhances ViTs' robustness against adversarial attacks. Additionally, we proposed a novel framework called the random entangled image transformer (ReiT), employing a dual-level randomization strategy to effectively bolster adversarial robustness. Our comprehensive experiments validate the superiority of the proposed ReiT, which achieves better robust performance compared to other robust techniques across widely adopted benchmark datasets.

Acknowledgments

This work was supported in part by the Australian Research Council under Projects DP240101848 and FT230100549. The authors acknowledge the use of the National Computational Infrastructure (NCI) which is supported by the Australian Government, and accessed through the NCI Adapter Scheme and Sydney Informatics Hub HPC Allocation Scheme.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *ECCV*, 2020. 2, 7

[2] Alexandre Araujo, Laurent Meunier, Rafael Pinot, and Benjamin Negrevergne. Robust neural networks using randomized adversarial training. *arXiv preprint arXiv:1903.10219*, 2019. 1, 2

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 1

[4] Yutong Bai, Jieru Mei, Alan Yuille, and Cihang Xie. Are transformers more robust than cnns? In *NeurIPS*, 2021. 1, 2

[5] Philipp Benz, Soomin Ham, Chaoning Zhang, Adil Karjauv, and In So Kweon. Adversarial robustness comparison of vision transformer and mlp-mixer to cnns. In *BMVC*, 2021.

[6] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *ICCV*, 2021. 1, 2

[7] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *S&P*, 2017. 2, 7

[8] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019. 1, 2

[9] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020. 2, 7

[10] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*, 2020. 2

[11] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: A standardized adversarial robustness benchmark. In *NeurIPS Datasets and Benchmarks Track*, 2021. 5, 7

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[13] Minjing Dong and Chang Xu. Adversarial robustness via random projection filters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4077–4086, 2023. 1

[14] Minjing Dong, Yanxi Li, Yunhe Wang, and Chang Xu. Adversarially robust neural architectures. *arXiv preprint arXiv:2009.00902*, 2020. 1

[15] Minjing Dong, Yunhe Wang, Xinghao Chen, and Chang Xu. Towards stable and robust addernets. *Advances in Neural Information Processing Systems*, 34:13255–13265, 2021. 1

[16] Minjing Dong, Xinghao Chen, Yunhe Wang, and Chang Xu. Random normalization aggregation for adversarial defense. 2022. 1, 2

[17] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018. 2, 7

[18] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, 2019. 2

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2, 3, 5

[20] Stéphane d'Ascoli, Hugo Touvron, Matthew L. Leavitt, Ari S. Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *ICML*, 2021. 2

[21] Yonggan Fu, Shunyao Zhang, Shang Wu, Cheng Wan, and Yingyan Lin. Patch-fool: Are vision transformers always robust against adversarial perturbations? In *ICLR*, 2022. 1

[22] Huihui Gong, Minjing Dong, Siqi Ma, Seyit Camtepe, Surya Nepal, and Chang Xu. Stealthy physical masked face recognition attack via adversarial style optimization. *IEEE Transactions on Multimedia*, 2023. 2

[23] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1, 2, 5

[24] Jeremy Howard. ImageNette. https://github.com/fastai/imagenette/, 2020. 5

[25] Haoqi Hu, Xiaofeng Lu, Xinpeng Zhang, Tianxing Zhang, and Guangling Sun. Inheritance attention matrix-based universal adversarial perturbations on vision transformers. *IEEE Signal Processing Letters*, 2021. 1, 2

[26] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *ICML*, 2018. 2

[27] Ameya Joshi, Gauri Jagatap, and Chinmay Hegde. Adversarial token attacks on vision transformers. *arXiv preprint arXiv:2110.04337*, 2021. 1, 2

[28] Hoki Kim. Torchattacks: a pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020. 7

[29] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 5

[30] Yanxi Li, Zhaohui Yang, Yunhe Wang, and Chang Xu. Neural architecture dilation for adversarial robustness. *Advances in Neural Information Processing Systems*, 34:29578–29589, 2021. 1

[31] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *ICLR*, 2020. 2

[32] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*, 2017. 2

[33] 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. In *ICCV*, 2021. 1, 2, 5

[34] Yanxiang Ma, Minjing Dong, and Chang Xu. Adversarial robustness through random weight sampling. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[35] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 1, 2, 5, 7

[36] Kaleel Mahmood, Rigel Mahmood, and Marten Van Dijk. On the robustness of vision transformers to adversarial examples. In *ICCV*, 2021. 1, 2

[37] Xiaofeng Mao, Gege Qi, Yuefeng Chen, Xiaodan Li, Ranjie Duan, Shaokai Ye, Yuan He, and Hui Xue. Towards robust vision transformer. In *CVPR*, 2022. 1

[38] Yichuan Mo, Dongxian Wu, Yifei Wang, Yiwen Guo, and Yisen Wang. When adversarial training meets vision transformers: Recipes from training to architecture. In *NeurIPS*, 2022. 1, 5, 7

[39] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016. 2, 7

[40] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *AAAI*, 2022. 2

[41] Rafael Pinot, Raphael Ettedgui, Geovani Rizk, Yann Chevaleyre, and Jamal Atif. Randomization matters how to defend against strong adversarial attacks. In *ICML*, 2020. 1, 2

[42] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of vision transformers. *Transactions on Machine Learning Research*, 2022. 1, 5

[43] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021. 5

[44] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 1, 2

[45] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 1, 2, 5

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3

[47] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020. 2

[48] Boxi Wu, Jindong Gu, Zhifeng Li, Deng Cai, Xiaofei He, and Wei Liu. Towards efficient adversarial training on vision transformers. In *ECCV*, 2022. 1, 5

[49] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *CVPR*, 2019. 2

[50] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019. 2, 5, 7

[51] Yuchen Zhang and Percy Liang. Defending against whitebox adversarial attacks via randomized discretization. In *ICAIS*, 2019. 1, 2

[52] Mingyi Zhou, Jing Wu, Yipeng Liu, Shuaicheng Liu, and Ce Zhu. Dast: Data-free substitute training for adversarial attacks. In *CVPR*, 2020. 2