Towards Efficient Adversarial Training on Vision Transformers

Boxi Wu*[1], Jindong Gu*[2], Zhifeng Li[3], Deng Cai[1], Xiaofei He[1], and Wei Liu[3]

[1] State Key Lab of CAD&CG, Zhejiang University
boxiwu@zju.edu.cn, {xiaofeihe,dengcai}@cad.zju.edu.cn
[2] University of Munich, Germany, jindong.gu@outlook.com
[3] Tencent Data Platform, China, michaelzfli@tencent.com,wl2223@columbia.edu

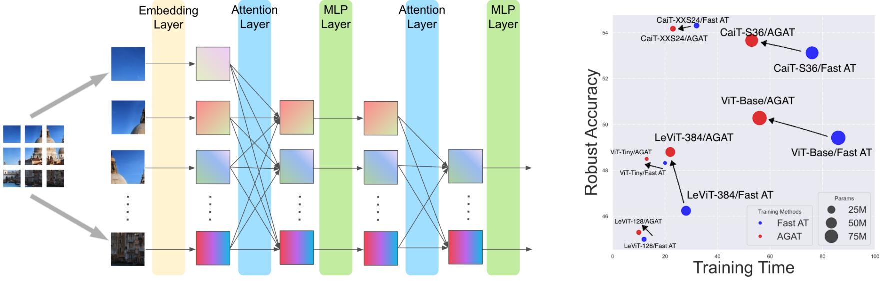
Abstract. Vision Transformer (ViT), as a powerful alternative to Convolutional Neural Network (CNN), has received much attention. Recent work showed that ViTs are also vulnerable to adversarial examples like CNNs. To build robust ViTs, an intuitive way is to apply adversarial training since it has been shown as one of the most effective ways to accomplish robust CNNs. However, one major limitation of adversarial training is its heavy computational cost. The self-attention mechanism adopted by ViTs is a computationally intense operation whose expense increases quadratically with the number of input patches, making adversarial training on ViTs even more time-consuming. In this work, we first comprehensively study fast adversarial training on a variety of vision transformers and illustrate the relationship between the efficiency and robustness. Then, to expediate adversarial training on ViTs, we propose an efficient Attention Guided Adversarial Training mechanism. Specifically, relying on the specialty of self-attention, we actively remove certain patch embeddings of each layer with an attention-guided dropping strategy during adversarial training. The slimmed self-attention modules accelerate the adversarial training on ViTs significantly. With only 65% of the fast adversarial training time, we match the state-of-the-art results on the challenging ImageNet benchmark.

Keywords: Robustness, Adversarial Training, Vision Transformer

1 Introduction

Vision Transformers with the self-attention mechanism have been broadly studied and become de facto state-of-the-art models for many benchmarks. Recent works broadly investigated the traits of this new genre of architectures on computer vision tasks. Meanwhile, the adversarial robustness of Vision Transformers has also been intensively studied [11, 52, 9, 44, 7, 1, 60, 84, 29, 46, 45, 48, 64, 32, 22]. To build robust Vision Transformers, an intuitive way is to apply adversarial training [63, 25] since it has been shown to be one of the most effective ways

* Equal contribution.



(a) The illustration of layerwisely dropping patches

(b) AGAT performance

Fig. 1. (a) AGAT chooses to drop a certain proportion of image embeddings based on the attention information at each self-attention layer. (b) We plot the robust accuracy against the training time (hours) for various ViTs. AGAT substantially accelerates adversarial training, while maintaining or improving the robustness at the same time.

to achieve robust CNNs [23, 43, 5]. However, one major limitation of adversarial training is its expensive computational cost. Adversarial training is known for requiring no extra cost during testing but greatly increasing the training cost. Huge efforts have been devoted to overcome this deficit [62, 81, 76, 3, 66, 51, 30, 31]. However, the philosophy of designing stronger ViTs has largely lifted up the computation intensity and weakened the performance of previously-proposed techniques. The specialty of the newly-proposed self-attention design [19, 13] in ViTs also introduces new challenges for accelerating adversarial training.

In this paper, we study the problem of how to efficiently carry out adversarial training on Vision Transformers. We first apply the state-of-the-art Fast Adversarial Training (Fast AT) algorithm [62, 81] on a variety of vision transformers and analyze how factors like attention mechanism, computational complexity, and parameter size influence the training quality. To the best of our knowledge, we are the first to accomplish a broad investigation on this topic. Our survey shows that, although ViTs outperform CNNs by a great margin on robustness, they have hugely increased the computational complexity. The self-attention mechanism adopted by ViTs is a computationally intense operation whose cost increases quadratically with the number of input patches. This newly-emerged module hampers the utilization of several techniques for accelerating adversarial training. Meanwhile, we find that large ViTs models also suffer from obvious catastrophic over-fitting problems [56]. This eventually leads to the degradation of robustness on ViTs with increasingly large capacity.

To make adversarial training efficient on the heavy-weight vision transformers, we investigate in accelerating adversarial training for vision transformers. Particularly, we leverage the specialty that the self-attention mechanism of transformers is capable of processing variable-length inputs. This specialty of ViTs has been utilized in a wide range of applications, including processing variable-length word sequences for translations [19], mining graphs with unlimited edges [75], etc. Recently, on vision tasks, several works [70, 55, 16] explored the possibility of dropping input image patches during training or testing for acceleration purposes. However, randomly dropping a certain number of input patches will inevitably hurt the training quality. Thus, many works have proposed adaptive designs [50, 83] in the scenarios of a variety of targeted tasks.

Enlightened by the above works, we propose an Attention-Guided Adversarial Training (AGAT) mechanism, where we drop the patches based on the attention information. As illustrated by Fig. 1(a), our method intends to drop image embeddings after each layer of self-attention. Note that the self-attention layer is non-parametric and thus is not limited to a static number of inputs. We drop the embeddings with lower attention and keep the higher ones. Such a design will better preserve the feed-forward process and therefore guard the backward gradient computation of generating the adversarial examples. As shown in Fig. 1(b), AGAT gets to keep the training quality mostly unchanged or be improved by taking only 65% training time. Our work matches the state-of-the-art results of adversarial robustness on the challenging benchmark of ImageNet.

2 Relate Works

Adversarial Training. Adversarial attacks [67, 23, 43, 80, 38, 39] intend to endanger the performance of deep networks via repeatedly optimizing the input images with repect to the output of the model. To counter this unwanted deficit, various defensive approaches were proposed [47, 20, 40, 82, 27, 65, 61, 41]. Among these defenses, the methodology of adversarial training withstands most kinds of examinations and has become one of a few defenses that can consistently improve the robustness of deep networks when facing most attacks [5]. However, adversarial training is known to suffer from complexity issues [62, 81, 54]. Particularly, Fast AT [81] enhances the single-step adversarial training with random initialization. Fast AT shows promising results on benchmark datasets. Later works [4, 33, 76, 3, 66, 51] also proposed improved variants of Fast AT.
Vision Transformer. The Transformer architecture and its self-attention mechanism were first proposed in the field of natural language processing (NLP) [74, 19, 13] and then adopted in the scenario of computer vision [85, 17, 79]. After the huge efforts of a surge of explorations [21, 71, 77], the Vision Transformer (ViT) has shown the potential to surpass the traditional convolutional neural networks. Then, researchers keep pushing this new philosophy of model design into a wide range of fields like high-resolution vision tasks [42, 78]. Meanwhile, to reduce the huge computational expense that is brought by the densely modeled self-attention mechanism, various techniques have been proposed [77, 17].
Adversarial Robustness of ViT. The adversarial robustness of ViT has also achieved great attention due to its impressive performance [12, 63, 53, 49, 26, 8, 10, 64]. Some works [12, 63, 10] first reported positive results where they showed that standard ViTs perform more robust than standard CNNs under adversarial attacks. The later works [8, 26] revealed that ViTs are not more robust than CNNs if both are trained in the same training framework. By adopting Transformers' training recipes, CNNs can become as robust as Transformers on

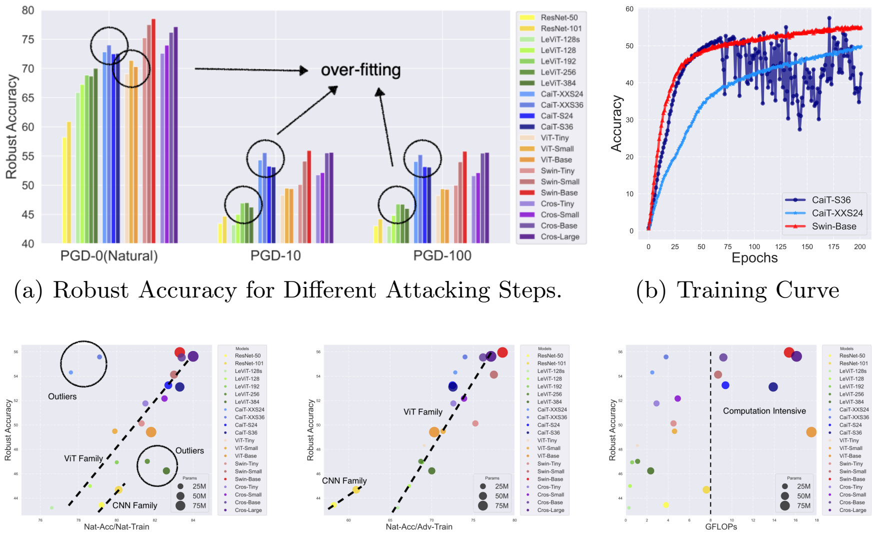
defending against adversarial attacks. In both sides, we can observe that the clean accuracy of standard models can be easily reduced to near zero under standard attack protocols. In addtion, Fu et al. [22] studied attacking ViTs in a patch-wise approach, which reveals the unique vulnerability of ViTs. To boost the adversarial robustness of ViTs, recent works [68, 6] explored multiple-step adversarial training to ViTs. Shao et al. [63], tested the vanilla adversarial training on CIFAR10. However, multi-step adversarial training is computationally expensive. And in this work, we take the step of exploring fast single-step adversarial training on ViT models.

3 Fast Adversarial Training on Vision Transformers

We first comprehensively study the Fast AT [81] algorithm on vision transformers. Fast AT is designed to be efficient so that it can be applied to large models (e.g., ResNet-101 [28]) on large-scale datasets (e.g., ImageNet [58]). Specifically, Fast AT refines the standard single-step FGSM [23] algorithm by adopting a large random perturbation as the starting point for searching adversarial examples. By doing so, Fast AT is supposed to effectively resist the catastrophic overfitting [81] of training with the plain FGSM. Thus, Fast AT preserves the effectiveness while significantly improves the efficiency over the multi-step PGD [43] algorithm. To better understand the robustness of the newly-developed ViT models, in this section, we apply Fast AT to a wide range of ViTs.

We select nineteen models of different sizes from five vision transformer families, including ViT [21], CaiT [73], LeViT [24], SwinTransformer [42], and CrossFormer [78]. The selected models cover a wide range of model designs, including hybrid models [24], slim models [24, 78], constrained attention [42, 78], and multi-scale attention [24, 78]. Following the settings of Fast AT [81], we set the perturbation radius to 2/255 for ImageNet and test the adversarially trained models with the 100-step PGD attack. Our training schedule aligns with SwinTransformer [42] and DeiT [72]. We keep all hyper-parameters, *e.g.*, image size, training epoch, and data augmentation, identical for all models. We also present the results of Fast AT on the CNN models of ResNet-50 and ResNet-101 for comparison. As shown by Fig. 2(a), ViTs are consistently more robust than CNNs. This aligns with concurrent researches on model robustness [69]. We conclude draw novel observations as follows:

1. **Within the same transformer family, larger transformers do not always result in better robustness.** In Fig. 2(a), the transformer families of LeViT, CaiT, and ViT exhibit a pattern of over-fitting. Namely, as the models get larger, the network robustness learned by Fast AT degrades conversely. For instance, Cait-S36 performs worse than CaiT-XXS24. Fig. 2(b) provides more details of the above over-fitting issue. The optimization of the CaiT-S36 model gradually degrades after a certain point. In contrast, the CaiT-XXS24 model possesses a monotonically increasing training curve. This aligns with previous findings that ViTs may suffer from more severe



(a) Robust Accuracy for Different Attacking Steps.

(b) Training Curve

(c) Performance Outliers

(d) Performance Alignment

(e) Computation Intensity

Fig. 2. (a) Robust accuracy for various ViTs. (b) Large ViTs like CaiT-S36 may suffer from obvious unstable training and over-fitting, while designs like window attention [42] can alleviate the issue. (c) The perfromance of ViTs on Fast AT may not align with their natural accuracy of natural training. (d) In contrast, the natural and robust accuracies of ViTs align with each other when both are under Fast AT. (e) Large ViTs greatly increase the computational complexity.

over-fitting on natural tasks and need the assistance of aggressive data augmentation schedules [72]. Moreover, transformers with a constrained attention mechanism can alleviate the over-fitting problem. In Fig. 2(b), unlike CaiT-S36, the large model of Swin-Base shows a steady training curve.

2. **Among different transformer families, the attention mechanism designed for better natural performance not necessarily results in better robustness.** In Fig. 2(c), for each transformer architecture, we plot its natural performance under natural training against its robust performance under Fast AT. Among different transformer architectures, the two metrics of natural accuracy and robust accuracy approximately form a line, indicating the close relation between natural and robust performances. Compared with the line formed by CNNs, vision transformers consistently achieve better robustness on models with similar natural performance. However, we can observe a few outliers of models from the LeViT and CaiT families. Specifically, the hybrid design of LeViT can achieve high robustness with very small models. Large CaiTs, despite being effective on the natural task, result in obvious inferiority on robustness. Notice that, for each point in Fig. 2(c), the two metrics of the vertical axis and the horizontal axis are evaluated on two models, either adversarially-trained or naturally-trained,

of the same architecture. When we plot the natural accuracy and robust accuracy, both of which are under Fast AT, as shown by Fig. 2(d), the two metrics consistently align with each other without any outlier, revealing the difference between Fast AT and standard training.

3. **SOTA ViTs suffer from a severe efficiency issue and require much more training time than SOTA CNNs.** The above over-fitting problem mainly shows on large ViTs like Cait-S36 or LeViT-384, but not on the small ones. This is reasonable since models like Cait-S36 are consistently larger than commonly-used CNNs. However, the over-fitting is not the only problem of adopting larger and larger ViTs. These increasingly large models have hugely lifted up the computation intensity. In Fig. 2(e), we plot robustness (Robust Accuracy) against computation intensity (GFLOPs). State-of-the-art ViTs can be a few times larger than CNNs. This makes utilizing Fast AT even harder since the self-attention module is more intricate to accelerate.

4 Efficient Adversarial Training on Vision Transformers

As discussed in the last section, one major problem that hinders the deployment of adversarial training on vision transformers is the efficiency issue. Adversarial training is known to be computationally intensive. This greatly hampers its usage on large-scale models or datasets. Various techniques have been proposed to mitigate this problem. However, with the revolution brought by vision transformers, many existing techniques such as variable-resolution training [81] have been unusable. More importantly, there is a rising trend of adopting increasingly large ViTs for better performance. The complexity of these enormous ViTs is too large to afford, even for efficient algorithms like Fast AT.

In this section, we first analyze the computational complexity of popular ViTs. Our analysis shows that ViT requires a much longer time to finish adversarial training, which is caused by the large computational cost of ViT brought by a large number of input patches. Then, we explore the input patches to reduce the brought computational cost. Given the fact that the flexibility of self-attention allows ViTs to process an arbitrary length of image patches, we explore a random patch dropping strategy to reduce the computation. The dropping operation with the reduced number of patches can accelerate adversarial training, as expected. However, the naive dropping strategy will also hurt robustness. To address the above issues, we propose our Attention-Guided Adversarial Training algorithm, which selectively drops patches based on attention magnitude.

4.1 Computation Intensity of ViTs

We first formally formulate our task. For each matrix, we present its shape in the lower right corner and its index in the upper right corner. Denote the input feature as $\mathbf{X}_{p\times d}$, which consists of a sequence of p embeddings with the dimension being $d : \mathbf{X} = [X_d^1, X_d^2, ..., X_d^p]$. Each embedding relates to a specific non-overlapped patch of the input image. Vision transformer consists of a list of blocks, each of which consists of two kinds of computation, i.e., the Multi-head Self-Attention layer (MSA) and the Multi-Layer Perceptron layer (MLP). In the MSA module, $\mathbf{X}$ is first normalized via Layer Normalization and then transformed to the *query, key, and value* matrices ($\mathbf{K},\ \mathbf{Q},\ \mathbf{V}$).

$$[\mathbf{K}_{p\times d},\ \mathbf{Q}_{p\times d},\ \mathbf{V}_{p\times d}] = \mathrm{LayerNorm}(\mathbf{X}_{p\times d})\mathbf{W}^1_{d\times 3d}. \tag{4.1}$$

For the multi-head design, we partition the $\mathbf{K},\ \mathbf{Q},\ \mathbf{V}$ matrices of shape $p\times d$ into h heads, with each part having a shape of $p\times\frac{d}{h}$. Then, taking the first head as an example, $\mathbf{V}^1$ will be re-weighted by $\mathbf{A}^1$ with the following form:

$$\mathrm{Attn}(\mathbf{K}^1_{p\times\frac{d}{h}}, \mathbf{Q}^1_{p\times\frac{d}{h}}, \mathbf{V}^1_{p\times\frac{d}{h}}) = \mathrm{SoftMax}(\mathbf{Q}^1\mathbf{K}^{1^\top}/\sqrt{d} + \mathbf{B})\mathbf{V}^1 = \mathbf{A}^1_{p\times p}\mathbf{V}^1. \tag{4.2}$$

$\mathbf{B}$ is a learnable bias. Note that all the column vectors of $\mathbf{A}^1$ are normalized by Softmax and thus have a summation of 1 in each. Then, $\mathbf{AV}$ value of each head will be concatenated and transformed to the output of MSA.

$$\mathbf{X}'_{p\times d} = \mathrm{Concat}(\mathbf{A}^1\mathbf{V}^1, \mathbf{A}^2\mathbf{V}^2, ..., \mathbf{A}^h\mathbf{V}^h)_{p\times d}\mathbf{W}^2_{d\times d}. \tag{4.3}$$

A following MLP module will take in the output of MSA, $\mathbf{X}'$, and transform each embedding with Layer Normalization and GELU activation.

$$\mathbf{X}''_{p\times c} = \mathrm{GELU}\Big[\mathrm{LayerNorm}(\mathbf{X}'_{p\times d})\mathbf{W}^3_{d\times 4d}\Big]\mathbf{W}^4_{4d\times d}. \tag{4.4}$$

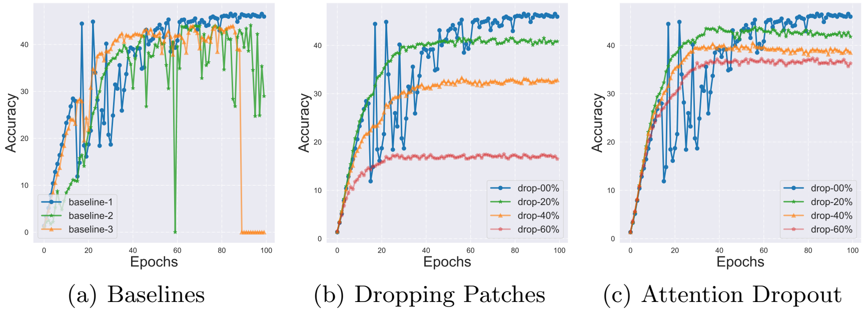
The computational complexity of the above process is:

$$\Omega(\mathrm{MSA}) = 4pd^2 + 2p^2d + pd; \quad \Omega(\mathrm{MLP}) = 8pd^2 + pd. \tag{4.5}$$

A typical vision transformer will consecutively conduct the above process to generate the final image representation for prediction. Take the ViT-Base model as an example. Each layer has $d = 768$. Therefore, we have $\Omega(\mathrm{MSA})+\Omega(\mathrm{MLP}) = 7\times 10^6 p + 1.5\times 10^3 p^2$. Since $7\times 10^6 \gg 1.5\times 10^3$ and p is mostly around 2×10^2, the computational complexity of the entire ViT is approximately linear to p.

4.2 Dropping Patch: The Flexibility of Self-Attention

Different from the convolutional operation where the hyperparameters (e.g., kernel size, padding size) are supposed to be fixed, the self-attention operation does not require the inputs with fixed length. For instance, this flexibility of self-attention is leveraged to process an arbitrary length of words in NLP tasks. Similarly, the flexibility makes its adaption to graph data feasible, in which different nodes can have a different number of connected edges [75]. When transformers with self-attention mechanisms have been introduced into computer vision tasks, researchers also investigate dynamically dropping the patches or the embeddings in the forward pass of a ViT model [70, 55, 16]. It is found that, when a constrained quantity of patches are dropped, the forward inference can be significantly accelerated. Meanwhile, the performance of the model will be



(a) Baselines (b) Dropping Patches (c) Attention Dropout

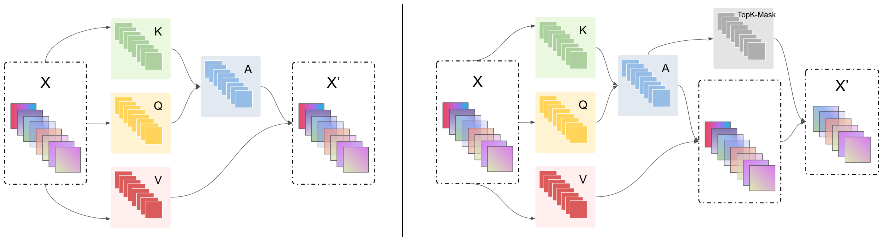
Fig. 3. (a) We repeat the baseline without any dropping for three times. We can observe the training is very unstable. (b) Training curve when we drop various rates of input patches. (c)Training curve when we adopt various rates of attention dropout.

only slightly degraded [50, 83]. Several works utilize this feature to design new mechanisms for their own unique purposes.

In this work, we also explore a patch dropping strategy to accelerate adversarial training given the excellent trade-off it achieves. We first test the scheme of randomly dropping a certain number of input patches to see how it influences the training quality of Fast AT. We report the results in Fig 3(b), where we plot the robustness against the training epoch for different ratios of dropping. Note that no patches will be dropped during inference in the testing stage. When the number of input patches is reduced, the forward inference of ViT can be accelerated. Surprisingly, from the figure, we also observe that the dropping operation also stabilizes the adversarial training and alleviates the phenomenon of catastrophic over-fitting [81, 59, 34]. As shown by Fig 3(a), we repeat Fast AT without any dropping for three times. The training procedure can be very unstable and occasionally drop to zero accuracy. We conjecture that it is the regularization effect brought by the patch dropping operation that stabilizes Fast AT. To further verify this conjecture, we test ViTs equipped with the dropout operation as in DeiT [72]. The dropout module is applied right after the self-attention module. As shown in Fig 3(c), like dropping patches, the attention dropout module also stabilizes Fast AT. Unlike dropping patches, the dropout module cannot save computation. However, the final robust accuracy can be reduced in both cases when dropping is applied. The random patch dropping strategy poses a dilemma. Namely, it brings both acceleration and performance degradation. In the following section, we will present our attention-guided patch dropping strategy, where we achieve a better trade-off between efficiency and effectiveness.

4.3 Attention-Guided Adversarial Training

It is known that adversarial training utilizes adversarial attacks to generate examples so that the network can learn to fit the generated adversarial examples. This is a typical hard example mining framework. The more powerful the adver-



(a) Plain Self-Attention.

(b) Attention of AGAT.

Fig. 4. The illustration of the plain self-attention layer and the attention layer of AGAT. The embedding mask is directly generated from **A** and is non-parametric. Thus, for the same model, one can adopt our AGAT during training and the plain self-attention during testing, respectively.

sarial examples are, the more robust the learned network will be. The quality of the adversarial examples relies on the adversarial attacking algorithm. And the attack algorithm depends on accurately estimating the gradient of the input pixels with respect to the loss function. Thus, a major insight of achieving our goal is to drop the patches that will barely hamper the gradient estimation. A similar philosophy has been utilized for sparse attacks or black-box attacks.

Recent works studied the learned attention and found that the magnitude of the attention can reveal how salient an embedding is [15]. This enlightens us that we can utilize this ready-made information to filter salient embeddings. Thus, we sum $\frac{1}{h}\sum_{i=1}^{h}\mathbf{A}^i$ by row and generate an index vector $\mathbf{a}$. Note that the column of $\frac{1}{h}\sum_{i=1}^{h}\mathbf{A}^i$ is the weighted-average parameter and thus always equal to 1. It indicates how much a generated embedding receives the information of each input embedding. In contrast, the row of $\frac{1}{h}\sum_{i=1}^{h}\mathbf{A}^i$ reveals how much each input embedding influences the output embeddings. This value differs from embedding to embedding. Thus, we choose to select the top-k embeddings based on their magnitude in $\mathbf{a}$. Then, the formulation of (4.3) becomes:

$$\mathbf{X}'_{k\times d} = \mathrm{MaskBy}\Big[\mathrm{Concat}(\mathbf{A}^1\mathbf{V}^1, ..., \mathbf{A}^h\mathbf{V}^h), \mathrm{Topk}(\mathbf{a})\Big]_{k\times d}\mathbf{W}^2_{d\times d}. \tag{4.6}$$

We drop the embeddings after the weighted average calculation of **AV** so that the magnitude of embeddings will be kept stable. The number of embeddings will reduce from p to k. To fully utilize the attention information in each layer, we propose a layer-wise exponential dropping scheme. Namely, in each layer, we drop a constant proportion of patches. Thus, this scheme will drop more embeddings on deeper layers, where the embeddings are consistently more redundant [55]. We set the dropping rate to 0.9. On a 12-layer ViT-Base model, the final layer will process only 31% number of embeddings and save more than 40% FLOPs of the entire model. A detailed implementation of our Attention-Guided Adversarial Training is shown in Algorithm 1. Our AGAT only modifies the training process. During testing, we use the original model for prediction. The class token will not be dropped when it involves the feed-forward procedure.

Algorithm 1 AGAT attention code (PyTorch-like)

```
def init(num_heads, dim, drop_rate):
    head_dim = dim // num_heads # num_heads: the number of heads, dim: embedding dimension
    scale = head_dim ** -0.5

    qkv = nn.Linear(dim, dim * 3)
    proj = nn.Linear(dim, dim)

def forward(self, x): # x: input tensor with the shape of (b, p, d);
    b, p, d = x.shape # b: batch size; p: patch number
    q, k, v = qkv(x).reshape(b, p, 3, num_heads, head_dim).permute(2, 0, 3, 1, 4).unbind(0)

    attn = (q @ k.transpose(-2, -1)) * scale
    attn = attn.softmax(dim=-1) # b num_heads p p

    own_attn = torch.sum(torch.sum(attn, dim=1), dim=-2) # b p
    kept_num = int(p * drop_rate) - 1 # compute the number of kept embeddings
    _, rank_indices = torch.topk(own_attn[:,1:], k=kept_num, dim=-1) # b k
    rank_indices = rank_indices.unsqueeze(-1).repeat(1,1,dim) # for the API of torch.gather

    x = (attn @ v).transpose(1, 2).reshape(B, N, C)
    x = torch.gather(x, dim=1, index=rank_indices) # drop embeddings

    return proj(x)
```

5 Experiments

5.1 Experimental Setup

Dataset We evaluate our method on the challenging ImageNet [57] dataset. Due to the huge computational expense of adversarial training, most adversarial training approaches are only verified on relatively small datasets such as CIFAR10 [36] or MNIST [37]. Our efforts on improving the efficiency of adversarial training allow us to apply adversarial training to large-scale datasets. The input image size is 224 on all models for a fair comparison.

Training Schedule Following previous works [72, 78], we use the AdamW [35] optimizer for training for 300 epochs with a cosine decay learning rate schedule. The initial learning rate is set to 0.001. The first 20 epochs adopt the linear warm-up strategy. The batch size is 1024 split on 8 NVIDIA A100 GPUs.

Evaluation Metrics We report our major results on two metrics, robust accuracy and GFLOPs. We mainly focus on improving the speed of training and keeping the learned robustness unchanged at the mean time. For a direct impression of training speed, we also record the training time for each method. Note that the training time is not only determined by the efficiency of the training algorithm, but also the IO speed and many other nonnegligible factors.

Adversarial Attack We choose the powerful multi-step PGD attack [43] with the perturbation radius being 2/255 or 4/255 and the optimization step being 20 or 100 [82]. We also test different kinds of attacks, including black-box attacks, to rule out the possibility of obfuscated gradient [5].

Vision Transformers Our AGAT can be directly used on the self-attention model and most of its variants. We apply our AGAT to three commonly-used models ViT [21], CaiT [73], and LeViT [24]. All the three models are built on the

Table 1. Adversarial Training on The ImageNet Dataset. In most cases, our Attention-Guided Adversarial Training on ViTs achieves comparable clean performance and robust accuracy to Fast-AT with much less time. The conclusion still holds when different perturbation ranges are applied.

							$\epsilon = 2/255$			$\epsilon = 4/255$		
Model	Params	Block Number	Training Method	Dropping Rate	FLOPs	Training Time	Nat-Acc.	PGD-20	PGD-100	Nat-Acc.	PGD-20	PGD-100
ResNet-50 [28]	25.6M	16	Free AT [62]	-	3.8G	46H	**62.28**	**43.77**	**43.44**	**58.31**	30.89	30.71
			Fast AT [81]	-	3.8G	**14H**	58.20	43.62	43.31	52.62	30.17	30.13
			Grad Align [4]	-	3.8G	**14H**	57.44	42.61	42.46	53.66	**31.18**	**31.01**
ResNet-101 [28]	44.5M	33	Free AT [62]	-	7.6G	51H	**64.37**	43.27	43.14	**60.41**	31.17	31.14
			Fast AT [81]	-	7.6G	**17H**	60.90	**44.57**	**44.11**	55.62	**33.02**	**33.26**
			Grad Align [4]	-	7.6G	**17H**	60.12	43.28	43.04	53.94	30.27	30.21
CaiT-XXS24 [73]	12.0M	24	Fast AT [81]	0.0	2.5G	32H	**72.84**	**54.31**	**54.26**	**68.77**	**32.14**	**32.09**
			Random	0.4	**1.5G**	**22H**	65.62	39.81	39.52	60.87	31.54	31.46
			AGAT	0.05	**1.5G**	23H	71.15	54.17	54.08	68.22	31.46	31.39
CaiT-XXS36 [73]	17.3M	36	Fast AT [81]	0.0	3.8G	38H	**74.01**	55.58	55.41	73.54	34.61	34.28
			Random	0.4	**2.7G**	**25H**	69.25	50.13	49.89	65.12	28.00	27.91
			AGAT	0.03	**2.7G**	**25H**	73.83	**55.81**	**55.72**	**73.91**	**35.22**	**35.19**
CaiT-S36 [73]	68.2M	36	Fast AT [81]	0.0	13.9G	76H	72.51	53.12	52.76	**71.20**	33.02	32.84
			Random	0.4	7.8G	**51H**	70.46	51.27	51.03	66.62	28.95	28.75
			AGAT	0.03	**7.7G**	53H	**72.69**	**53.66**	**53.48**	71.06	**33.46**	**33.17**
LeViT-128 [24]	7.8M	9	Fast AT [81]	0.0	0.4G	12H	**67.30**	45.00	44.87	64.66	**32.11**	**32.09**
			Random	0.4	**0.2G**	**10H**	58.14	30.62	30.52	54.94	28.05	28.01
			AGAT	0.15	**0.2G**	**10H**	67.19	**45.30**	**45.21**	**64.98**	32.02	31.88
LeViT-256 [24]	11.0M	12	Fast AT [81]	0.0	0.6G	15H	68.69	46.94	46.89	**66.24**	**33.81**	**33.62**
			Random	0.4	**0.4G**	**12H**	60.71	31.45	31.18	57.64	29.89	29.66
			AGAT	0.1	**0.4G**	13H	**68.90**	**47.37**	**47.06**	65.98	33.12	33.05
LeViT-384 [24]	39.0M	12	Fast AT [81]	0.0	2.35G	28H	**70.01**	46.24	46.13	65.38	31.22	31.04
			Random	0.4	**1.3G**	**20H**	63.70	33.74	33.26	60.11	28.51	28.02
			AGAT	0.1	**1.3G**	22H	69.73	**48.80**	**48.59**	**67.02**	**33.48**	**33.35**
ViT-Tiny [21]	5.1M	12	Fast AT [81]	0.0	1.1G	20H	69.09	48.32	48.28	**64.03**	**31.68**	**31.20**
			Random	0.4	**0.6G**	**13H**	61.11	31.58	31.16	58.10	27.43	27.25
			AGAT	0.1	**0.6G**	**13H**	**69.64**	**48.50**	**48.46**	63.02	31.17	31.04
ViT-Small [21]	22M	12	Fast AT [81]	0.0	4.6G	47H	**71.37**	**49.49**	**49.41**	**66.92**	**34.07**	**33.82**
			Random	0.4	2.7G	33H	63.62	33.58	33.29	60.18	29.10	28.91
			AGAT	0.1	**2.6G**	**32H**	70.62	49.00	48.85	66.10	33.62	33.40
ViT-Base [21]	86M	12	Fast AT [81]	0.0	17.5G	86H	70.31	50.55	50.06	65.18	33.59	33.39
			Random	0.4	10.5G	**55H**	65.80	37.04	36.81	61.16	30.07	30.01
			AGAT	0.1	**10.1G**	56H	**70.41**	**51.23**	**51.11**	**67.93**	**34.94**	**34.78**

original self-attention module and can fully reveal the effectiveness of our AGAT. In future work, we will explore combining our AGAT with more sophisticated attention mechanism like window attention [42] or multi-scale attention [78].

Training Algorithm For vision transformers, we compare our AGAT with FastAT and the random dropping strategy (Random). we also provide results of Free AT [62] and Grad Align [4] algorithms on the ResNet [28] models.

5.2 Improved Efficiency of Adversarial Training on ImageNet

We present the performance of AGAT in Table 1. Because AGAT drops a static rate of embeddings for each self-attention layer, the depth of the ViTs will decide the total number of dropped features. Thus, we adjust the dropping rate for ViTs with different numbers of blocks so that the complexity of the feed-forward process will be approximately reduced by 40% of the baseline. For instance, we

Table 2. Evaluation of Adversarially Trained Models under Various Attacks. Robust accuracy of adversarially trained models is reported in this table. The robust accuracy achieved by our AGAT is comparable to that by Fast-AT under various attack evaluations and different perturbation ranges.

Method	Model	$\epsilon = 2/255$					$\epsilon = 4/255$				
		PGD	C&W	APGD-CE	APGD-DLR	Square	PGD	C&W	APGD-CE	APGD-DLR	Square
ViT-Tiny [21]	Fast AT	48.28	48.02	48.20	48.05	58.21	**31.20**	**31.09**	**31.11**	**31.00**	**37.97**
	AGAT	**48.46**	**48.17**	**48.37**	**48.24**	**58.33**	31.04	31.03	31.01	30.85	37.84
ViT-Base [21]	Fast AT	50.06	49.87	50.01	49.52	59.28	33.39	33.36	33.28	33.10	40.81
	AGAT	**51.11**	**50.84**	**51.23**	**50.45**	**61.12**	**34.78**	**34.42**	**34.54**	**34.20**	**42.24**
CaiT-XXS24 [73]	Fast AT	**54.26**	**54.19**	**54.20**	**54.13**	**63.22**	**32.09**	**32.01**	**32.08**	**31.90**	**38.02**
	AGAT	54.08	53.88	54.00	53.98	62.35	31.39	31.22	31.20	31.11	37.48
CaiT-S36 [73]	Fast AT	52.76	52.51	52.70	52.53	61.75	32.84	32.73	32.79	32.61	39.71
	AGAT	**53.48**	**53.13**	**53.37**	**53.02**	**62.86**	**33.17**	**33.08**	**33.10**	**33.00**	**40.12**
LeViT-128 [24]	Fast AT	44.87	44.81	44.73	44.60	56.89	**32.09**	**32.03**	**32.02**	**31.95**	**38.90**
	AGAT	**45.21**	**45.19**	**45.19**	**45.06**	**57.12**	31.88	31.77	31.87	31.76	38.61
LeViT-384 [24]	Fast AT	46.13	45.98	46.02	45.93	56.39	31.25	31.00	31.22	30.82	37.99
	AGAT	**48.59**	**48.26**	**48.58**	**48.49**	**58.90**	**33.35**	**33.24**	**33.29**	**33.17**	**40.22**

set the dropping rate to 0.1 for models with 12 blocks but 0.03 for models with 36 blocks. For the baseline of randomly dropping input patches, we can always set the dropping rate to 0.4 since the total amount of saved computation will not be affected by the number of blocks.

As shown by Table 1, for each vision transformer, the Fast AT baseline achieves high robustness but is time-consuming, while the Random dropping strategy saves training time but achieves inferior robustness. In contrast, AGAT achieves comparable robustness with Fast AT using much less training time. Particularly, on the ViT-Base model, AGAT achieves similar robustness to Fast AT but only takes 65% of training time. Meanwhile, since slim models such as ViT-Tiny and LeViT-128 do not possess the same level of model redundancy as their large-sized counterparts, the robustness of these slim models degrades more dramatically than larger ones when we randomly drop patches. When trained with AGAT, the robustness of slim models matches the plain Fast AT .

5.3 Ablation Study

Results under Various Attacks. Adversarial robustness is known to be hard to examine. Several defensive algorithms were found to be vulnerable to tailored attacks. One of the most important and typical representatives of such phenomena is the obfuscated gradient problem. Our AGAT does not fall into this category, considering the algorithm neither utilizes any stochastic process nor hampers the gradient computation. In fact, our AGAT only takes effect on the training stage and does not modify any procedure during evaluation. To further show the robustness of the learned ViTs, we present the robust accuracy of our learned models under various different attacks in Table 2. We select the attacking criteria of C&W [14], Square [2], APGD-CE [18], and APGD-DLR [18]. The AGAT models achieve the same level of robustness as Fast AT.

Table 3. Ablation Study on Dropping Strategy. We compare our attention-guided dropping strategy with random dropping. Ours outperforms random dropping constantly in different dropping rates.

			$\epsilon = 2/255$				$\epsilon = 4/255$			
Model	Method	Attack	0%	20%	40%	60%	0%	20%	40%	60%
ViT-Tiny [21]	Random	PGD	48.28	39.02	31.16	21.01	31.20	29.64	27.25	24.11
	AGAT	PGD	48.28	**48.15**	**48.46**	**46.60**	31.20	**31.15**	**31.04**	**28.13**
ViT-Small [21]	Random	PGD	49.41	40.62	33.29	28.81	33.82	30.13	28.91	25.25
	AGAT	PGD	49.41	**49.43**	**48.85**	**47.90**	33.82	**33.29**	**33.40**	**32.10**
ViT-Base [21]	Random	PGD	49.26	45.02	36.81	33.17	33.39	32.61	30.01	28.86
	AGAT	PGD	49.26	**49.80**	**50.02**	**48.20**	33.39	**34.15**	**34.78**	**32.90**

Robustness and Dropping Rate. We cross-validate the rate of dropping of our AGAT in Table 3. To provide a clear comparison with the random dropping strategy, we compare their learned robust accuracy when both dropping strategies reduce approximately the same amount of computation. Due to the difference between the two algorithms (layer-wise vs input-wise), the actual learning rates are different across the two methods. It can be told that AGAT can maintain the learned robustness in a wide range of dropping rates. In contrast, the random dropping strategy significantly degrades the performance, especially for the slim model of ViT-Tiny. Dropping more than 40% computation will bring obvious degradation on robustness, even for AGAT. Thus, we consider this dropping rate as a good trade-off between effectiveness and efficiency.

Visualization. To better get an insight of how AGAT takes effect, we visualize the internal results of the ViT-Base model. For each of the 12 blocks in ViT-Base, we show the position of the dropped embeddings by masking out the corresponding image patches. In Fig. 5.3, the position of the dropped embedding mainly concentrates on the relatively unimportant positions like background, while the patches of the main object are mostly kept. This indicates that our AGAT successfully guards the feed-forward procedure and thus secures the generation of adversarial examples. We also visualize the corresponding value of attention for each patch. The darkness of each patch position indicates how much the corresponding embedding of this patch influences the other embeddings. For each block, we normalize all the values of attention by dividing the maximum value of attention. This visualization also demonstrates that the dropping rate of AGAT gets to cover the embeddings on the position of the main object. Therefore, dropping rates larger than the chosen value may lose crucial information for inference and thus hamper the generation of adversarial examples for training.

6 Conclusions

Adversarial training is one of the most effective defense methods to boost the adversarial robustness of models. However, it is computationally expensive, even

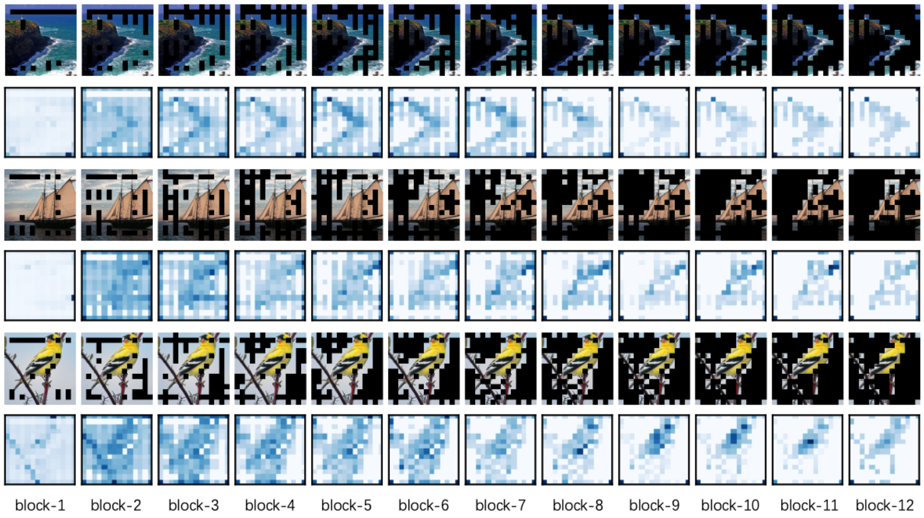


Fig. 5. Visualizations of dropped image patches and the distribution of attention. AGAT mainly cuts off the computation of non-object embeddings and thus maintains the performance and gradient computation.

after many efforts have been made to address it. The emergence of ViTs, whose computational cost increases quadratically with the number of input patches, makes adversarial training more challenging. In this work, we first thoroughly examined the most popular fast adversarial training on various ViTs. Our investigation shows that ViT achieves higher robust accuracy than ResNet, while it does suffer from a large computation burden, as expected. Our further exploration showed that random input patch dropping can accelerate and stabilize the adversarial training, which, however, sacrifices the final robust accuracy. To overcome the dilemma, we proposed an Attention-Guided Adversarial Training (AGAT) mechanism based on the specialty of the self-attention mechanism. Our AGAT leverages the attention to guide the patch dropping process, which accelerates the adversarial training significantly and maintains the high robust accuracy of ViTs. We hope that this work can serve the community as a baseline for research on efficient adversarial training on vision transformers.

Acknowledgements. This work was supported in part by The National Key Research and Development Program of China (Grant Nos: 2018AAA0101400), in part by The National Nature Science Foundation of China (Grant Nos: 62036009, U1909203, 61936006, 62133013), in part by Innovation Capability Support Program of Shaanxi (Program No. 2021TD-05).

References

1. Aldahdooh, A., Hamidouche, W., Deforges, O.: Reveal of vision transformers robustness against adversarial attacks. arXiv:2106.03734 (2021)
2. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: A query-efficient black-box adversarial attack via random search. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) ECCV. Lecture Notes in Computer Science, vol. 12368, pp. 484–501. Springer (2020). https://doi.org/10.1007/978-3-030-58592-1_29, https://doi.org/10.1007/978-3-030-58592-1_29
3. Andriushchenko, M., Flammarion, N.: Understanding and improving fast adversarial training. NeurIPS (2020)
4. Andriushchenko, M., Flammarion, N.: Understanding and improving fast adversarial training. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020), https://proceedings.neurips.cc/paper/2020/hash/b8ce47761ed7b3b6f48b583350b7f9e4-Abstract.html
5. Athalye, A., Carlini, N., Wagner, D.A.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: ICML. Proceedings of Machine Learning Research, vol. 80, pp. 274–283. PMLR (2018)
6. Bai, J., Yuan, L., Xia, S., Yan, S., Li, Z., Liu, W.: Improving vision transformers by revisiting high-frequency components. CoRR **abs/2204.00993** (2022). https://doi.org/10.48550/arXiv.2204.00993, https://doi.org/10.48550/arXiv.2204.00993
7. Bai, Y., Mei, J., Yuille, A., Xie, C.: Are transformers more robust than cnns? arXiv:2111.05464 (2021)
8. Bai, Y., Mei, J., Yuille, A., Xie, C.: Are transformers more robust than CNNs? In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), https://openreview.net/forum?id=hbHkvGBZB9
9. Benz, P., Ham, S., Zhang, C., Karjauv, A., Kweon, I.S.: Adversarial robustness comparison of vision transformer and mlp-mixer to cnns. CoRR **abs/2110.02797** (2021), https://arxiv.org/abs/2110.02797
10. Benz, P., Ham, S., Zhang, C., Karjauv, A., Kweon, I.S.: Adversarial robustness comparison of vision transformer and mlp-mixer to cnns. arXiv preprint arXiv:2110.02797 (2021)
11. Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T., Veit, A.: Understanding robustness of transformers for image classification. CoRR **abs/2103.14586** (2021), https://arxiv.org/abs/2103.14586
12. Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T., Veit, A.: Understanding robustness of transformers for image classification. arXiv:2103.14586 (2021)
13. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Neural Information Processing Systems, NeurIPS (2020)

14. Carlini, N., Wagner, D.A.: Towards evaluating the robustness of neural networks. In: SP. pp. 39–57. IEEE Computer Society (2017)
15. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021. pp. 9630–9640. IEEE (2021). https://doi.org/10.1109/ICCV48922.2021.00951, `https://doi.org/10.1109/ICCV48922.2021.00951`
16. Chen, T., Cheng, Y., Gan, Z., Yuan, L., Zhang, L., Wang, Z.: Chasing sparsity in vision transformers: An end-to-end exploration. CoRR **abs/2106.04533** (2021), `https://arxiv.org/abs/2106.04533`
17. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C.: Twins: Revisiting spatial attention design in vision transformers. CoRR **abs/2104.13840** (2021)
18. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: ICML (2020)
19. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Conference of the North American Chapter of the Association for Computational Linguistics, NAACL. pp. 4171–4186 (2019)
20. Dhillon, G.S., Azizzadenesheli, K., Lipton, Z.C., Bernstein, J., Kossaifi, J., Khanna, A., Anandkumar, A.: Stochastic activation pruning for robust adversarial defense. ICLR (2018)
21. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations, ICLR (2021)
22. Fu, Y., Zhang, S., Wu, S., Wan, C., Lin, Y.: Patch-fool: Are vision transformers always robust against adversarial perturbations? In: International Conference on Learning Representations (2022)
23. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Bengio, Y., LeCun, Y. (eds.) ICLR (2015), `http://arxiv.org/abs/1412.6572`
24. Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., Douze, M.: Levit: a vision transformer in convnet's clothing for faster inference. CoRR **abs/2104.01136** (2021), `https://arxiv.org/abs/2104.01136`
25. Gu, J., Tresp, V., Qin, Y.: Are vision transformers robust to patch perturbations? CoRR **abs/2111.10659** (2021), `https://arxiv.org/abs/2111.10659`
26. Gu, J., Tresp, V., Qin, Y.: Are vision transformers robust to patch perturbations? In: arXiv preprint arXiv:2111.10659 (2021)
27. Guo, C., Rana, M., Cisse, M., Van Der Maaten, L.: Countering adversarial images using input transformations. ICLR (2018)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90, `https://doi.org/10.1109/CVPR.2016.90`
29. Hu, H., Lu, X., Zhang, X., Zhang, T., Sun, G.: Inheritance attention matrix-based universal adversarial perturbations on vision transformers. IEEE Signal Processing Letters **28**, 1923–1927 (2021)
30. Jia, X., Zhang, Y., Wu, B., Ma, K., Wang, J., Cao, X.: Las-at: Adversarial training with learnable attack strategy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13398–13408 (June 2022)

31. Jia, X., Zhang, Y., Wu, B., Wang, J., Cao, X.: Boosting fast adversarial training with learnable adversarial initialization. IEEE Transactions on Image Processing (2022)
32. Joshi, A., Jagatap, G., Hegde, C.: Adversarial token attacks on vision transformers. arXiv:2110.04337 (2021)
33. Kim, H., Lee, W., Lee, J.: Understanding catastrophic overfitting in single-step adversarial training. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021. pp. 8119–8127. AAAI Press (2021), https://ojs.aaai.org/index.php/AAAI/article/view/16989
34. Kim, H., Lee, W., Lee, J.: Understanding catastrophic overfitting in single-step adversarial training. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021. pp. 8119–8127. AAAI Press (2021), https://ojs.aaai.org/index.php/AAAI/article/view/16989
35. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980
36. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
37. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010), http://yann.lecun.com/exdb/mnist/
38. Liang, S., Wei, X., Yao, S., Cao, X.: Efficient adversarial attacks for visual object tracking. In: European Conference on Computer Vision. pp. 34–50. Springer (2020)
39. Liang, S., Wu, B., Fan, Y., Wei, X., Cao, X.: Parallel rectangle flip attack: A query-based black-box attack against object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7697–7707 (2021)
40. Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., Zhu, J.: Defense against adversarial attacks using high-level representation guided denoiser. In: CVPR. pp. 1778–1787 (2018)
41. Liu, W., Jiang, Y., Luo, J., Chang, S.: Noise resistant graph ranking for improved web image search. In: The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011. pp. 849–856. IEEE Computer Society (2011). https://doi.org/10.1109/CVPR.2011.5995315, https://doi.org/10.1109/CVPR.2011.5995315
42. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. CoRR **abs/2103.14030** (2021)
43. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR. OpenReview.net (2018), https://openreview.net/forum?id=rJzIBfZAb
44. Mahmood, K., Mahmood, R., Van Dijk, M.: On the robustness of vision transformers to adversarial examples. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7838–7847 (2021)
45. Mao, X., Qi, G., Chen, Y., Li, X., Duan, R., Ye, S., He, Y., Xue, H.: Towards robust vision transformer. arXiv:2105.07926 (2021)

46. Mao, X., Qi, G., Chen, Y., Li, X., Ye, S., He, Y., Xue, H.: Rethinking the design principles of robust vision transformer. arXiv:2105.07926 (2021)
47. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. pp. 135–147 (2017)
48. Mu, N., Wagner, D.: Defending against adversarial patches with robust self-attention. In: ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning (2021)
49. Naseer, M.M., Ranasinghe, K., Khan, S.H., Hayat, M., Shahbaz Khan, F., Yang, M.H.: Intriguing properties of vision transformers. Advances in Neural Information Processing Systems **34** (2021)
50. Pan, B., Jiang, Y., Panda, R., Wang, Z., Feris, R., Oliva, A.: Ia-red^2: Interpretability-aware redundancy reduction for vision transformers. CoRR **abs/2106.12620** (2021), `https://arxiv.org/abs/2106.12620`
51. Park, G.Y., Lee, S.W.: Reliably fast adversarial training via latent adversarial perturbation. ICCV (2021)
52. Paul, S., Chen, P.: Vision transformers are robust learners. CoRR **abs/2105.07581** (2021), `https://arxiv.org/abs/2105.07581`
53. Paul, S., Chen, P.Y.: Vision transformers are robust learners. arXiv:2105.07581 (2021)
54. Qin, C., Martens, J., Gowal, S., Krishnan, D., Dvijotham, K., Fawzi, A., De, S., Stanforth, R., Kohli, P.: Adversarial robustness through local linearization. In: NeurIPS (2019)
55. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. CoRR **abs/2106.02034** (2021), `https://arxiv.org/abs/2106.02034`
56. Rice, L., Wong, E., Kolter, J.Z.: Overfitting in adversarially robust deep learning. CoRR **abs/2002.11569** (2020), `https://arxiv.org/abs/2002.11569`
57. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge (2015)
58. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision, IJCV (2015)
59. S., V.B., Babu, R.V.: Single-step adversarial training with dropout scheduling. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 947–956. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.00103, `https://openaccess.thecvf.com/content\_CVPR\_2020/html/B.S.\_Single-Step\_Adversarial\_Training\_With\_Dropout\_Scheduling\_CVPR\_2020\_paper.html`
60. Salman, H., Jain, S., Wong, E., Madry, A.: Certified patch robustness via smoothed vision transformers. arXiv:2110.07719 (2021)
61. Samangouei, P., Kabkab, M., Chellappa, R.: Defense-gan: Protecting classifiers against adversarial attacks using generative models. ICLR (2018)
62. Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J.P., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! In: NeurIPS (2019)
63. Shao, R., Shi, Z., Yi, J., Chen, P.Y., Hsieh, C.J.: On the adversarial robustness of visual transformers. arXiv:2103.15670 (2021)

64. Shi, Y., Han, Y.: Decision-based black-box attack against vision transformers via patch-wise adversarial removal. arXiv preprint arXiv:2112.03492 (2021)
65. Song, Y., Kim, T., Nowozin, S., Ermon, S., Kushman, N.: Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. ICLR (2018)
66. Sriramanan, G., Addepalli, S., Baburaj, A., et al.: Towards efficient and effective adversarial training. NeurIPS (2021)
67. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: Bengio, Y., LeCun, Y. (eds.) ICLR (2014), http://arxiv.org/abs/1312.6199
68. Tang, S., Gong, R., Wang, Y., Liu, A., Wang, J., Chen, X., Yu, F., Liu, X., Song, D., Yuille, A., et al.: Robustart: Benchmarking robustness on architecture design and training techniques. arXiv preprint arXiv:2109.05211 (2021)
69. Tang, S., Gong, R., Wang, Y., Liu, A., Wang, J., Chen, X., Yu, F., Liu, X., Song, D., Yuille, A.L., Torr, P.H.S., Tao, D.: Robustart: Benchmarking robustness on architecture design and training techniques. CoRR **abs/2109.05211** (2021), https://arxiv.org/abs/2109.05211
70. Tang, Y., Han, K., Wang, Y., Xu, C., Guo, J., Xu, C., Tao, D.: Patch slimming for efficient vision transformers. CoRR **abs/2106.02852** (2021), https://arxiv.org/abs/2106.02852
71. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning, ICML. vol. 139, pp. 10347–10357 (2021)
72. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 10347–10357. PMLR (2021), http://proceedings.mlr.press/v139/touvron21a.html
73. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. CoRR **abs/2103.17239** (2021), https://arxiv.org/abs/2103.17239
74. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Neural Information Processing Systems, NeurIPS. pp. 5998–6008 (2017)
75. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018), https://openreview.net/forum?id=rJXMpikCZ
76. Vivek, B., Babu, R.V.: Single-step adversarial training with dropout scheduling. In: CVPR (2020)
77. Wang, W., Xie, E., Li, X., Fan, D., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. CoRR **abs/2102.12122** (2021)
78. Wang, W., Yao, L., Chen, L., Lin, B., Cai, D., He, X., Liu, W.: Crossformer: A versatile vision transformer hinging on cross-scale attention. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=_PHymLIxuI

79. Wang, Z., Jiang, W., Zhu, Y., Yuan, L., Song, Y., Liu, W.: Dynamixer: A vision MLP architecture with dynamic mixing. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., Sabato, S. (eds.) International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA. Proceedings of Machine Learning Research, vol. 162, pp. 22691–22701. PMLR (2022), `https://proceedings.mlr.press/v162/wang22i.html`
80. Wei, X., Liang, S., Chen, N., Cao, X.: Transferable adversarial attacks for image and video object detection. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. pp. 954–960 (2019)
81. Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: Revisiting adversarial training. CoRR **abs/2001.03994** (2020), `https://arxiv.org/abs/2001.03994`
82. Xie, C., Wang, J., Zhang, Z., Ren, Z., Yuille, A.: Mitigating adversarial effects through randomization. ICLR (2018)
83. Xu, Y., Zhang, Z., Zhang, M., Sheng, K., Li, K., Dong, W., Zhang, L., Xu, C., Sun, X.: Evo-vit: Slow-fast token evolution for dynamic vision transformer. CoRR **abs/2108.01390** (2021), `https://arxiv.org/abs/2108.01390`
84. Yu, Z., Fu, Y., Li, S., Li, C., Lin, Y.: Mia-former: Efficient and robust vision transformers via multi-grained input-adaptation. arXiv preprint arXiv:2112.11542 (2021)
85. Zhang, Q., Yang, Y.: Rest: An efficient transformer for visual recognition. CoRR **abs/2105.13677** (2021)