Are Transformers more robust than CNNs?

Anonymous Author(s)
Affiliation
Address
email

Abstract

Transformer emerged as a powerful tool for visual recognition. Besides demonstrating competitive performance on a broad range of visual benchmarks, recent works also argue that Transformers are much more robust than Convolutional Neural Networks (CNNs). Nonetheless, surprisingly, we find these conclusions are drawn from unfair experimental settings, where Transformers and CNNs are compared at different scales and are applied with distinct training frameworks. In this paper, we aim to provide the first **fair & in-depth** comparisons between transformers and CNNs, focusing on robustness evaluations.

With our unified training setup, we first challenge the previous belief that Transformers outshine CNNs when measuring *adversarial robustness*. More surprisingly, we find CNNs can even achieve exactly the same robustness as Transformers on defending against adversarial attackers, if they properly adopt Transformers' training recipes. While regarding *generalization on out-of-distribution samples*, we show pretraining on (external) large-scale datasets is not a fundamental request for enabling Transformers to achieve better performance than CNNs. Moreover, our ablations suggest such stronger generalization is largely benefited by the self-attention-like architectures per se, rather than other training setups. We hope that this work will help the community better understand and benchmark the robustness comparisons between Transformers and CNNs.

1 Introduction

Convolutional Neural Networks (CNNs) have long been the meta-architecture for visual recognition [19, 32, 34, 13, 18]. It is commonly believed that the key to such success is the usage of the convolutional operation, as it introduces several useful inductive biases (*e.g.*, translation equivalence) to models for benefiting object recognition. Interestingly, recent works alternatively suggest that it is also possible to build successful recognition models without convolutions [27, 48, 2]. The most representative work in this direction is Vision Transformer (ViT) [10], which applies the pure self-attention-based architecture to sequences of images patches and achieves competitive performance with CNNs on the challenging ImageNet classification task [28]. Later works [22, 39] further generalize Transformers with compelling performance on a wide range of other visual benchmarks, including COCO detection and instance segmentation [20], ADE20K semantic segmentation [49].

The dominion of CNNs on visual recognition is further challenged by the recent findings that transformers appear to be *much more robust* than CNNs. For example, Shao *et al*. [31] observe the usage of convolutions may introduce a negative effect on adversarial robustness, while gradually switching to transformer-like architectures (*e.g.*, the Conv-Transformer hybrid model or the pure Transformer) can help to secure model performance. Similarly, Bhojanapalli *et al*. [3] report that, if pre-trained on sufficiently large datasets, Transformers exhibit considerably stronger robustness than CNNs on a range of out-of-distribution tests (*e.g.*, common image corruptions [14], texture-shape cue conflicting stimuli [11]).

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Though both [31] and [3] claim that Transformers are preferable to CNNs in terms of robustness, we find that such conclusion cannot be strongly drawn based on existing experiments. Firstly, the comparison between Transformers and CNNs is not conducted at the same model scale, *e.g.*, a small CNN, ResNet-50 (∼25 million parameters), by default is compared to a much larger Transformer, ViT-B (∼86 million parameters), for these robustness evaluations. Secondly, the training frameworks applied to Transformers and CNNs are distinct from each other (*e.g.*, different training datasets/epochs & augmentation strategies), while little efforts are devoted on ablating the corresponding effects. In a nutshell, due to these inconsistent and unfair experiment settings, *it remains an open question whether Transformers are truly more robust than CNNs*.

To answer it, in this paper, we aim to provide the first benchmark to fairly compare Transformers and CNNs on robustness evaluations. We particularly focus on the comparisons between Small Data-efficient image Transformers (DeiT-S) [36] and ResNet-50 [13], as their model capacity is on par with each other (*i.e.*, ∼22 million parameters *vs*. ∼25 million parameters), and they achieve similar performance on ImageNet (*i.e.*, 76.8% top-1 accuracy *vs*. 76.3% top-1 accuracy), when both are trained exclusively on ImageNet and follow the setups in [36]. Our evaluation suite accesses model robustness in two ways: 1) adversarial robustness, where the attackers can actively and aggressively manipulate inputs to approximate the worst-case scenario; 2) generalization on out-of-distribution samples, including common image corruptions (ImageNet-C [14]), texture-shape cue conflicting stimuli (Stylized-ImageNet [11]) and natural adversarial examples (ImageNet-A [15]).

With this unified training setup, we present a completely different picture from previous ones [31, 3]. Regarding adversarial robustness, we find that Transformers are no more robust than CNNs, and identify the selection of data augmentation strategy plays a pivot role here. More surprisingly, if CNNs can properly adopt Transformers' training setup and take adversarial training into consideration, we show CNNs can achieve exactly the same robustness as Transformers on defending against adversarial attacks. While for generalization on out-of-distribution samples, we find Transformers can still substantially outperforms CNNs even without the needs of pre-training on sufficiently large (external) datasets. Additionally, our ablations show adopting self-attention-like architecture is the key for achieving strong robustness on these out-of-distribution samples, while leaving the tuning of other training setups only be subtle here. We hope this work can serve as a useful benchmark for future explorations on robustness, using Transformers and CNNs.

2 Related Works

Vision Transformer. Transformers, invented by Vaswani *et al*. in 2017 [38], have largely advanced the field of natural language processing (NLP). With the introduction of self-attention module, Transformer can effectively capture the non-local relationships between all input sequence elements, achieving the state-of-the-art performance on a broad spectrum of NLP tasks [44, 8, 4, 9, 25, 26].

The success of Transformer on NLP also starts to get witnessed in computer vision recently. The pioneering work, ViT [10], demonstrate that the pure Transformer architectures are able to achieve exciting results on several visual benchmarks, especially when extremely large datasets (*e.g.*, JFT-300M [33]) are available for pre-training. This work is then subsequently improved by carefully curating the training framework and the distillation strategy to Transformers [36], improving the Transformers' tokenization module [45], building multi-resolution feature maps on Transformers [22, 39], *etc*. In this work, rather than focusing on furthering Transformers on standard visual benchmark, we aim to provide a fair and comprehensive study of their performance when testing out of the box.

Robustness evaluations. Conventional learning paradigm assumes training data and testing data are drawn from the same distribution. This assumption generally is not true, especially in the real-world case where the underlying distribution is too complicated to be covered in a (limited-sized) training dataset. To properly access model performance in the wild, a set of robustness generalization benchmarks have been built, *e.g.*, ImageNet-C [14], Stylized-ImageNet [11], ImageNet-A [15], *etc*. Another standard surrogate for testing model robustness is via adversarial examples, where the attackers deliberately add small perturbations or patches to input images, for approximating the worst-case evaluation scenario [35, 12]. In this work, both robustness generalization and adversarial robustness are considered in our robustness evaluation suite.

Concurrent to ours, both Bhojanapalli *et al*. [3] and Shao *et al*. [31] conduct the robustness comparisons between Transformers and CNNs. Nonetheless, we find these comparisons are unfair, *e.g*., models are compared at different capacity [3, 31] or are trained under distinct framework [31]. In this work, our comparison carefully align the model capacity and training setups, which draws completely different conclusions from the previous ones.

3 Settings

3.1 Networks Architectures & Training Pipelines

Vision Transformer ViT [10] mostly follows the original design of Transformer [38, 9]. In this paper, we use its improved version, DeiT [36], to train ViT in a data efficient way. Also, considering the model size, DeiT provides smaller model whose parameters are similar to ResNet. By default, we use AdamW [23] as optimizer, with cosine learning rate scheduler. Learning rate is set to 5e-4. We trained 100 epochs, as a fair comparison to ResNet-50. We follow [37] to use RandAug [7], MixUp [47] and CutMix [46] as data augmentation methods. Note that no any other regularization method here, since it may over-regularize models in this (relatively short) 100 training epochs.

Convolutional Neural Networks Here we choose ResNet50 [13], one of the most representative convolutional neural networks, as the default network. We use SGD as optimizer with step learning rate scheduler, decaying every 30 epochs. The learning rate is set to 0.1. We trained all models for 100 epochs, including first 5 warm-up epochs.

3.2 Robustness Evaluations

In this section we compare the robustness on out-of-distribution sample of ViT models to ResNets. To capture different aspects of robustness we rely on different, specialized benchmarks ImageNet-A[16], ImageNet-C [14] and Stylized-ImageNet[11]. We also pit our models against different types of adversarial attacks, including AutoAttack[6] and Texture Patch Attack[43].

ImageNet-A The ImageNet-A dataset contains natural adversarial examples, which are real-world images which were found from some challenging scenarios (e.g., occlusion and fog scene) that are difficult for recognition. For ImageNet-A, we use top-1 accuracy to evaluate the robustness.

ImageNet-C The ImageNet-C dataset is designed for measuring network robustness to common image corruptions. It consists of 15 diverse corruption types grouped into 4 categories: 'noise', 'blur', 'weather', and 'digital'. Each corruption type has five levels of severity, resulting in 75 distinct corruptions. For ImageNet-C, we use mCE (mean corruption error) to evaluate the robustness.

Stylized-ImageNet The Stylized-ImageNet dataset is created by removing local texture cues in ImageNet while retaining global shape information on natural images via AdaIN style transfer. This nudges networks towards learning more about shapes and less about local textures, as suggested in [11]. For Stylized-ImageNet, we use top-1 accuracy to evaluate the robustness.

AutoAttack AutoAttack holistically evaluates adversarial robustness with a parameter-free ensemble of diverse attacks, including two parameter-free versions of PGD [24], and two existing complementary attacks, Fast Adaptive Boundary (FAB) attack [5] and Square Attack [1].

Texture Patch Attack Texture Patch Attack (TPA) uses a predefined texture dictionary of patches to fool deep networks. For TPA, the texture dictionary is generated per-class, where each texture is optimized to elicit some specific output from the model.

4 Robustness to Adversarial Attackers

4.1 Robustness to AutoAttack

Following [31], we first report robustness of ResNet-50 and DeiT-Small under AutoAttack. Similar to previous findings, vision transformer achieves higher accuracy when applying a small attacking radius $\epsilon = 0.001$ while maintains similar clean accuracy on ImageNet as shown in Table 1.

Table 1: Defense success rate (%) of ResNet-50 and DeiT-Small against AutoAttack with different attack radius.

	Clean	Attack radius	
	acc	0.001	4/255
ResNet-50	76.9	17.8	0
DeiT-Small	76.8	22.1	0

However, when increasing the attack radius to 4/255, a more challenging but popular case [30, 40, 41], both architectures show **zero defense ability**. This observation motivates us to perform adversarial training [12, 24], an efficient way to increase the models ability to defense against adversarial attackers, on both those architectures for a further study on their robustness.

4.2 Adversarial Training

We conduct adversarial training on DeiT-small and ResNet-50. Specifically, adversarial training [12, 24], aims to optimize the framework below by training networks with adversarial examples that generated on-the-fly:

$$_{\theta}\mathbb{E}_{(x,y)\sim\mathbb{D}}\left[\max_{\epsilon\in\mathbb{S}} L(\theta, x+\epsilon, y)\right], \tag{1}$$

where $\mathbb{D}$ is the underlying data distribution, $L(\cdot,\cdot,\cdot)$ is the loss function, θ is the network parameter, x is a training sample with the ground-truth label y, ϵ is the added adversarial perturbation, and $\mathbb{S}$ is the allowed perturbation range.

4.2.1 Adversarial Training on Transformers

Following the default training strategy of DeiT, we carry out the adversarial training, which aims to train the network on adversarially perturbed images to increase the robustness.

We apply the PGD attacker [24] to generate adversarial perturbations ϵ. PGD performs a gradient descent step iteratively in the loss function w.r.t. the image pixel values based on an adversarially selected output target. Then it projects the resulting perturbed images into the feasible solution space within a maximum per-pixel perturbation of ϵ of the clean image, subject to an L_∞ constraint. Following [42, 29, 40], we select the cheapest version of PGD, *single-step PGD* (PGD-1), to lower the training cost. And we set the maximum per-pixel change $\epsilon = 4$ and the attack step size $\beta = 4$. No regularization except a weight decay of 0.05 is applied.

Interestingly, directly applying adversarial training to DeiT does not work—the model collapses and leads to only 4% robustness accuracy against PGD-5. We solve this problem by gradually increasing the training images' difficulty to provide a warm up with the learning rate, as show in Figure 1. This help stabilize the initial adversarial training process of Vision Transformer, and further leads to 44% of robustness accuracy of PGD-5.

This problem would not happen if simply applying to Transformer ResNet's adversarial training recipe that uses SGD optimizer with step learning rate schedule with no further augmentations, as previous work's setting[31]. But this recipe that crafted for ResNet would impair Transformers performance by a large range. According to our experiments the clean accuracy decreases to 59.9%(**-6.6%**), and PGD-100 robustness to 31.9%(**-8.4%**).

This sub-optimal results under SGD optimizer is expected. Liu *et al*. [21] gives an explanation on why standard SGD fails in training Transformers. As observed in [21], adaptive optimizers is capable of assign different learning rates to different parameters and lead to consistent update magnitudes even with unbalanced gradients. This is crucial since gradients of all attention modules in Transformers are unbalanced. Therefore, SGD is not suitable for training transformers, under both vanilla and adversarial training setting.

As discussed in many works before that AdamW is better for Transformer's vanilla training, adversarial training shows similar signals. Further more, augmentations including RandAug, MixUp and CutMix are also critical for adversarial training to prevent overfitting, as we have observed.

(a) Epoch = 0 (b) Epoch = 4 (c) Epoch = 9

Figure 1: The illustration of augmentations applied when epoch=0, 4 and 9, showing how we progressively increase augmentation strength as a warm-up for adversarial training.

Aligning ResNet to DeiT As discussed above, *completely* following ResNet's training recipe seems to impair the performance of DeiT. Similarly for ResNet-50, if changed to AdamW optimizer, we observed that adversarial training process will collapse. And if applied with extra augmentations, both ResNet-50's clean accuracy and PGD robustness will suffers as well. More details will be provided in the supplementary materials.

However, some training technique in DeiT can still help the ResNet's adversarial training. According to the experiments in [41], activation function plays an important role in adversarial training. DeiT uses GELU as an activation function, while ResNet uses ReLU. In Table 2, experiments show that after properly applying GELU as activation function to ResNet-50, ResNet's adversarial training can lead to both higher clean accuracy and robustness compared with same settings with ReLU.

Main Results We show that in Table 2, after adversarial training, ResNet-50 can outperform DeiT-Small on ImageNet clean accuracy by 0.9% while have exactly the same robustness on defending against PGD attacks and AutoAttacks under the same 4/255 attack radius, challenging previous conclusions that transformers are more robust than CNNs. We also re-evaluate their robustness to AutoAttack under radius 0.001 on adversarially trained model. After changing to GELU, ResNet-50's defense success rate increases from 62.2% to 65.7% (+2.5%), surpass DeiT-Small's 64.8% robustness by 0.9%.

Table 2: After aligning ResNet with DeiT' activation function GELU, ResNet-50 shows better accuracy (%) on ImageNet and while maintains the same robustness under adversarial attackers.

	Activation	Clean Acc	PGD-5	PGD-10	PGD-50	PGD-100	AutoAttack-4/255
ResNet-50	ReLU	66.8	38.7	34.2	32.5	32.3	26.4
	GELU	**67.4**	44.0	41.0	40.3	40.3	35.5
DeiT-Small	GELU	66.5	44.0	41.0	40.3	40.3	35.5

4.3 Robustness to PatchAttack

Since Autoattack is composed with one patch based attack, we further study the robustness with a more advanced patch attackers, named Texture Patch Attack (TPA) [43]. For TPA, we set the number of patches to 4 and the total area to 10% of the image area by default. With the number of patch increases, the difficulty of the TPA grows as well. We mainly evaluate non-targeted attacks of TPA.

Table 3 shows both clean accuracy on ImageNet and the defense success rate to TPA of ResNet-50 under different augmentation techniques used in DeiT. Compared to the original ResNet-50, all augmentation improve the robustness to patch attacks. But interestingly, we note observe that CutMix is extremely effective to improve the robustness of TPA, *i.e.*, CutMix only can subtantially improve model robustness against TPA by 29.4%.

The strongest performance for defending against TPA can be achieved by training with both CutMix and RandAug. In Table 4, we provide the comparison of the robustness to stronger patch attacks (*i.e.*, by increasing patch number from 4 to 10) between ResNet-50 and DeiT-Small under this setting. We can observe that ResNet is more robust to different levels of TPA compared with DeiT.

Table 3: Defense success rate (%) of ResNet-50 with different augmentations against Patch Attacks. We also show clean accuracy on ImageNet(%).

Augmentations			ImageNet ↑	Patch Attack ↑
RandAug	MixUp	CutMix		
✗	✗	✗	78.9	19.7
✓	✓	✓	76.4	+28.9
✓	✓	✗	75.7	+12.0
✓	✗	✓	76.7	**+32.7**
✗	✓	✓	77.1	+20.1
✓	✗	✗	77.5	+4.6
✗	✓	✗	75.9	+11.8
✗	✗	✓	77.2	+29.4

Table 4: Defense success rate (%) of ResNet-50 and DeiT-Small against Patch Attacks with different patch number. We also show clean accuracy(%).

Architecture	ImageNet↑	Patch Attack↑	
		P=4	P=10
ResNet-50	76.7	**52.4**	**34.5**
DeiT-Small	76.8	47.7	28.9

5 Generalization on Out-of-distribution Samples

In this section we evaluate the robustness to out-of-distribution samples of DeiT to ResNet models. As aforementioned, the training frameworks applied to Transformers and CNNs varies in many aspects, *e.g.*, training epochs, optimizers, learning rate schedule, augmentations *etc*. To capture different aspects of robustness, we rely on different, specialized benchmarks including ImageNet-A, ImageNet-C, and Stylized ImageNet.

5.1 Aligned Setting Comparison

We first provide a direct comparison between DeiT and ResNet in their original setting. In addition, for a more fair comparison, we fully align to ResNet-50 DeiT's training strategies, whose details can be found in Section 3.1. We denote this model as ResNet50*.

Experiments in Table 5 show that DeiT-Small has better generalization on the out-of-distribution samples, compared with both ResNet-50's original setting and aligned setting ResNet-50*.

Table 5: DeiT-Small has better generalization ability on ImageNet-C, ImageNet-A and Stylized-ImageNet. For ImageNet, ImageNet-A and Stylized ImageNet, results show Top-1 accuracy (%). For ImageNet-C, results show mCE (%, mean corruption error, lower is better).

Architecture	ImageNet ↑	ImageNet-A ↑	ImageNet-C ↓	Stylized-ImageNet ↑
ResNet-50	76.9	3.2	57.9	8.3
ResNet-50*	76.3	4.5	55.6	8.2
DeiT-Small	76.8	**12.2**	**48.0**	**13.0**

5.2 Best Setting Comparison

Considering the fair comparison and the more advanced training strategies used in DeiT, we first provide the results on fully aligned settings. However, this strategy crafted for DeiT may not benefit ResNet-50 the most. Therefore, we study those settings step-by-step and provide a best setting for ResNet-50 and DeiT-Small as another comparison.

5.2.1 Optimizers and Learning Rate Scheduler

In previous studies of the robustness comparison between ResNet and DeiT, optimizers and learning rate scheduler are not aligned. Here we study which is best for ResNet and under its best setting, if ResNet is comparable with DeiT.

Experiment Setting ResNet uses SGD in the standard training, while DeiT used AdamW with cosine learning rate. We report the generalization results on both those settings.

Main Results Table 6 shows with SGD and cosine learning rate scheduler, ResNet-50 has better generalization to those datasets. Therefore, in later experiments, without extra mentions, we'll keep this setting. But even under the best setting, DeiT-Small still leads by a large margin.

Table 6: The generalization of ResNet-50 under different optimizer and learning rate scheduler compared with DeiT.

	Optimizer	ImageNet ↑	ImageNet-A ↑	ImageNet-C ↓	Sty-ImageNet ↑
ResNet-50	SGD-Step	76.9	3.2	57.9	8.3
	SGD-Cosine	**77.4**	3.3	56.9	8.4
	AdamW	76.4	3.1	59.3	8.1
DeiT-Small	AdamW	76.8	**12.2**	**48.0**	**13.0**

5.2.2 Augmentations

Experiments show that augmentations are beneficial to the robustness on out-of-distribution samples, such as ImageNet-A and ImageNet-C. DeiT uses many augmentations, including RandAug, MixUp and CutMix, which may potentially increase its robustness. Therefore, we apply the same augmentations to ResNet as well.

Experiment Setting Here we stick to SGD as optimizer with cosine learning rate schedule, which works the best for ResNet-50 as aforementioned, and apply different combinations of the augmentation strategies used in DeiT.

Main Results As is shown in Table 7, after applying the augmentations, the generalization ability of ResNet-50 on distorted ImageNet datasets has improved a lot. In general, RandAug with MixUp shows the best improvements. However, there still exist some gap compared with DeiT-Small.

Table 7: DeiT shows better generalization compared with ResNet under different augmentations.

Architecture	Augmentations			ImageNet ↑	ImageNet-A ↑	ImageNet-C ↓	Sty-ImageNet↑
	RandAug	MixUp	CutMix				
ResNet-50	✗	✗	✗	77.4	3.3	56.9	8.4
	✓	✓	✓	76.4	5.5	54.0	9.1
	✓	✓	✗	75.7	**6.3**	**52.3**	**10.8**
	✓	✗	✓	76.7	6.3	56.3	7.1
	✗	✓	✓	77.1	6.1	55.1	8.8
DeiT-Small	✓	✓	✓	**76.8**	**12.2**	**48.0**	**13.0**

5.2.3 Results of Best Setting Comparison

Therefore, after fully analyzing those settings step-by-step, we are able to provide a comparison between these two architectures under their best setting. DeiT-Small shows better generalization on out-of-distribution data in all those three dataset by a large margin under the best setting. The best setting we find for ResNet is SGD optimizer with cosine learning rate scheduler, applied with RandAug and MixUp. And we'll later refer to it as ResNet-Best.

Table 8: DeiT shows better generalization ability compared with ResNet-50 under best settings.

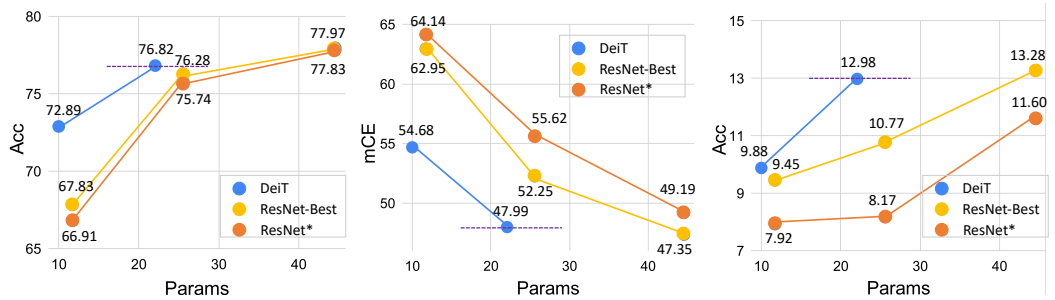
Architecture	ImageNet ↑	ImageNet-A ↑	ImageNet-C ↓	Stylized-ImageNet ↑
ResNet-50	75.7	6.3	52.3	10.8
DeiT-Small	76.8	**12.2**	**48.0**	**13.0**

5.3 The Model Size

In order to better understand and compare DeiT and ResNets, we evaluate a range of models from each architecture family to study this two archtectures' robustness under different model size.

Experiment Setting As aforementioned, ResNet-50 and DeiT-Small has similar parameters. However, for a smaller model ResNet-18, the original DeiT model family do not contain one with similar parameters. Therefore, in order to make a fair comparison with , we follow the designing rule of DeiT, and set up a model with similar parameters as ResNet-18, namely DeiT-Mini, whose embedding dimension is 256 and number of head is 4, between DeiT-Tiny and DeiT-Small.

Main Results From Figure 2 we can observe that both DeiT models show better robustness compared with ResNet models with similar model size under their best settings. Moreover, we further increase ResNet's model size to ResNet-101. After the capacity of ResNet grows to twice its size, it is now compatible with DeiT-Small.



(a) ImageNet-A (b) ImageNet-C (c) Stylized-ImageNet

Figure 2: DeiT models show better generalization ability compared with ResNet models with different model sizes on ImageNet-A, ImageNet-C and Stylized-ImageNet.

5.4 Distillation

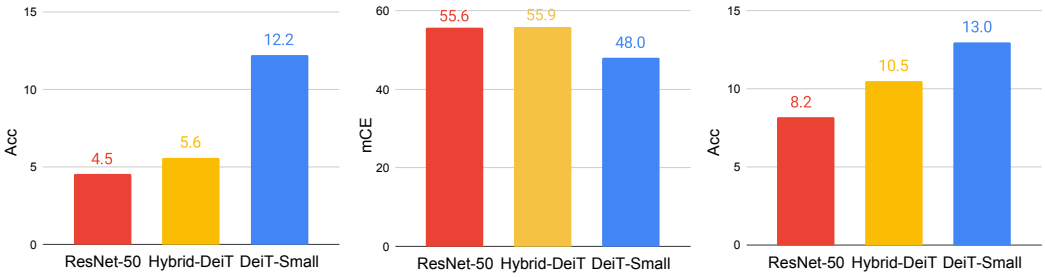
In the previous sections, we make comparison of ResNet-50 and DeiT-Small under aligned setting and best setting seperately. Here, we provide another comparison with the help of knowledge distillation. By analyzing the generalization difference between teacher model and student model, which learns from the passed knowledge its teacher model, we can further explore the generalization difference between these two different architectures.

Experiment Setting We design two sets of experiments. In the first set, we use ResNet-50 as a teacher model to distill DeiT-Small; in the second, Deit-Small as the teacher model to distill ResNet-50. We perform soft distillation following[17], which minimizes the Kullback-Leibler divergence between the softmax of the teacher and the softmax of the student model. During distillation, our training strategy remains consistent with DeiT-Small setting, same as the setting in Section 5.1.

Main Results In the first set, after distilling with ResNet-50's knowledge, DeiT-Small achieves higher generalization on ImageNet-A, ImageNet-C and Stylized-ImageNet, by a large margin. However, in the second set, it is counter-intuitive that after using DeiT-Small as teacher model, the student model, distilled ResNet-50, is actually performing worse. This suggest that the generalization robustness of DeiT is rooted in the structure of transformer that cannot be passed to ResNet via knowledge distillation.

Table 9: Generalization ability of ResNet-50, DeiT-Small and their distilled models.

Distillation	Architecture	ImageNet ↑	ImageNet-A ↑	ImageNet-C ↓	Stylized-ImageNet ↑
Teacher	ResNet-50*	76.3	4.5	55.6	8.2
Student	DeiT-Small-Distill	76.2	10.9 (+6.4)	49.3 (-6.3)	11.9 (+3.7)
Teacher	DeiT-Small	76.8	12.2	48.0	13.0
Student	ResNet-50*-Distill	76.7	5.2 (-7.0)	54.2 (+6.2)	9.8 (-3.2)



(a) ImageNet-A (b) ImageNet-C (c) Stylized-ImageNet

Figure 3: Transformer blocks show better generalization compared with ResNet structures.

5.5 Hybrid Architecture

As discussed above, DeiT architecture shows great generalization ability on out-of-distribution data by a large margin compared to ResNet models. In order to further validate whether this robustness is brought by the architecture of transformers, we introduce Transformer block to ResNet structure as hybrid model and analyze its generalization ability.

Experiment Setting We follow the basic rule of ViT's design for hybrid model, in which the patch embedding projection is applied to patches extracted from a CNN feature map. In our case, we take the output of stage 4 of a regular ResNet-18 and then feed the intermediate feature maps into DeiT-Mini. The choice of those two structures are meant to keep the model size of original and the hybrid at the same level. Under our setting, hybrid-DeiT contains ~21 million parameters, which is in the same scale with ResNet-50 (~25 million parameters) and DeiT-Small (~22 million parameters). Noe that all those models are trained following the DeiT recipe, same as the setting in Section 5.1.

Main Results Figure 3 shows the generalization on out-of-distribution samples of ResNet-50, DeiT-Small and Hybrid-DeiT on ImageNet-A, ImageNet-C and Stylized ImageNet, respectively. We note that, with the introduction of transformer block, this hybrid stucture's robustness increases in general compared with ResNet-50—with 1.1% growth on ImageNet-A top-1 accuracy and 2.5% growth on Stylized-ImageNet, while maintains similar mCE on ImageNet-C. However, if compared with DeiT-Small, which is composed of purely transformer blocks, hybrid structure's generalization robustness is still lower by a large margin. Therefore, we can conclude that the transformer block do imply more generalization robustness.

6 Conclusion

With the recent success of Vision Transformer on recognition tasks, researchers begin to study its robustness compared with traditional CNNs. While many recent works suggest that transformers are much more robust than CNNs, their comparisons are not fair in many aspects, *e.g.*, training datasets, model scales, training strategies, *etc*. Therefore, we believe that such conclusion cannot be drawn based on these settings, which motivates us to provide a fair and in-depth comparisons between these two architectures. With our unified training setup, we found that Transformers are no more robust than CNNs regarding adversarial robustness. And if properly aligned, CNNs can achieve exactly the same robustness as Transformers on defending against adversarial attacks, including PGD attacks, AutoAttack and Texture Patch Attack. While regarding generalization on out-of-distribution samples (*e.g.*, ImageNet-A, ImageNet-C and Stylized ImageNet), we find pre-training on (external) large-scale datasets is not a necessary condition for letting Transformers show strong performance than CNNs. Additionally, we find adopting self-attention-like architectures is crucial for generalization robustness. We hope this work would shed lights on the understanding of Vision Transformer and the robustness between Transformers and CNNs.

References

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

[2] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *ICCV*, 2019.

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

[4] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[5] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205. PMLR, 2020.

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

[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2019.

[8] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

[11] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2018.

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

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018.

[15] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

[16] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*. 2014.

[21] Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. *arXiv preprint arXiv:2004.08249*, 2020.

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

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

[25] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[26] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[27] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019.

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[29] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *NeurIPS*, 2019.

[30] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *NeurIPS*, 2019.

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

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[33] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

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

[36] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[37] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[39] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021.

[40] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.

[41] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020.

[42] Qizhe Xie, Eduard Hovy, Minh-Thang Luong, and Quoc Le. Self-training with noisy student improves imagenet classification. *arXiv preprint arXiv:1911.04252*, 2019.

[43] Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, and Alan Yuille. Patchattack: A black-box texture-based attack with reinforcement learning. In *European Conference on Computer Vision*, pages 681–698. Springer, 2020.

[44] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[45] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.

[46] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

[47] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[48] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *CVPR*, 2020.

[49] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]