REVISITING RESIDUAL NETWORKS FOR ADVERSARIAL ROBUSTNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Convolutional neural networks are known to be vulnerable to adversarial attacks. Solutions to improve their robustness have largely focused on developing more effective adversarial training methods, while limited efforts have been devoted to analyzing the role of architectural elements (such as topology, depth, and width) on adversarial robustness. This paper seeks to resolve this limitation and present a holistic study on the impact of architecture choice on adversarial robustness. We focus on residual networks and consider architecture design at the block level, i.e., topology, kernel size, activation, and normalization, as well as at the network scaling level, i.e., depth and width of each block in the network. We first derive insights on the block structure through systematic ablative experiments and design a novel residual block, dubbed RobustResBlock. It improves CW^{40} robust accuracy by $\sim$3% over Wide residual networks (WRNs), the de facto architecture of choice for designing robust architectures. Then we derive insights on the impact of depth and width of the network and design a compound scaling rule, dubbed RobustScaling, to distribute depth and width at a given desired FLOP count. Finally, we combine RobustResBlock and RobustScaling and present a portfolio of adversarially robust residual networks, RobustResNets, spanning a wide spectrum of model capacities. Experimental validation, on three datasets across four adversarial attacks, demonstrates that RobustResNets consistently outperform both the standard WRNs ($3 \sim 4$% improvement in robust accuracy while saving about half parameters) and other robust architectures proposed by existing works.

1 INTRODUCTION

Robustness to adversarial attacks is a critical consideration for practical deployments of deep neural networks. Current research on defenses against such attacks has primarily focused on developing better adversarial training methods (Madry et al., 2018; Zhang et al., 2019; Wang et al., 2019; Shafahi et al., 2019; Wong et al., 2020). These techniques, and the insights derived from them, have largely been developed by fixing the architecture of the network, typically variants of Wide Residual Networks (Zagoruyko & Komodakis, 2016). While significant knowledge exists on designing effective neural networks for standard tasks under standard settings, limited attention has been devoted to studying the role of architectural components on adversarial robustness. But, as we preview in Figure 1, architectural components can impact adversarial robustness as much as, if not more, different adversarial training methods. As such, we posit that there is a large void in practitioners' toolboxes on designing architectures with better adversarial robustness properties.

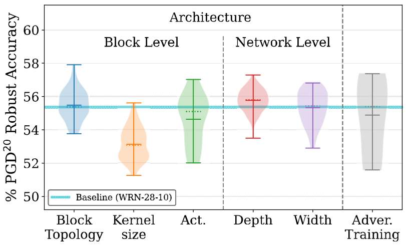


Figure 1: Impact of architectural components on adversarial robustness, relative to that of adversarial training methods. The variation of each component is elaborated in §3. Results on CIFAR-10.

The primary goal of this paper is to bridge this knowledge gap by, i) *systematically studying the contribution of architectural components to adversarial robustness*, ii) *identify key design choices that aid adversarial robustness*, and iii) finally *construct a new adversarially robust network that* *can serve as a baseline and test bed for studying architectural aspects of adversarial robustness*. We adopt an empirical approach and conduct an extensive amount of carefully designed experiments to realize this goal.

We start from the well-founded observation by Cazenavette et al. (2021) that networks with residual connections exhibit more robustness to adversarial attacks, and thus, consider the family of residual networks. Then we systematically consider the two main aspects of architecture design, namely block and network scaling, and *adversarially train and evaluate more than 1200 networks*. For *block*, we consider the choice of layers, exact connections between layers, and type of residual connection. For *network scaling*, we consider the choice of varying the width and depth of the network layers. To ensure the generality of the experimental observations, we evaluate them on three different datasets, and under four different adversarial attacks. To ensure the reliability of the empirical observations, we conduct several repetitions with different seeds. Based on our empirical observations, we identify architectural design principles that improve the adversarial robustness of networks, we propose a new block topology and network scaling scheme, dubbed RobustResBlock and RobustScaling, respectively, and finally propose a family of RobustResNets as a new benchmark architecture for studying adversarial robustness. The main findings from our experiments are:

1. Pre-activation is preferred over post-activation for adversarial robustness.
2. Bottleneck block improves adversarial robustness over the de facto basic block used in WRNs. Both aggregated and hierarchical convolutions, derived from standard tasks (i.e., on clean images), improve adversarial robustness.
3. Squeeze and excitation with minor customization improves adversarial robustness.
4. ReLU is consistently better than smooth activations on CIFAR-10 and Tiny-ImageNet across different model capacities when using an appropriate weight decay value. This contrasts with prevailing consensus that smooth activation functions are better than ReLU.
5. A larger kernel size does not necessarily lead to better adversarial robustness.
6. Architecture design contributes significantly to adversarial robustness. Under the same FLOPs budget, deep (but narrow) networks are adversarially more robust than wide (but shallow) networks.

In summary, we reaffirm known observations (6), challenge some existing observations (4), and finally make some new observations (1, 2, 3, 5, 6).

2 Experimental Setup

We now describe our experimental setup in terms of the adopted architectural skeleton and the details on training and evaluating the networks against adversarial attacks. *Code to reproduce our results and log files from our experiments can be found in the supplementary material.*

Architecture Skeleton: Figure 2 shows the skeleton of the network that we consider. It comprises a stem (i.e., a single 3×3 convolution), and three stages of processing. Each stage is made up of a varying number of convolutional blocks. The first block in stage two and three uses a stride of two to down sample the feature sizes by half. We denote the depth (i.e., number of blocks) and width (in terms of widening factors) of $i\text{-}th$ stage by D_i and W_i, respectively. Unless otherwise specified, we use 3×3 convolution, ReLU activation, and batch normalization as the standard operations. We study the effect of the block topology (variants of residual blocks) and the network scaling (configurations of $[D_1, D_2, D_3]$ and $[W_1, W_2, W_3]$), within this architectural skeleton, on the network's adversarial robustness.

Datasets, Training, and Evaluation Metrics: We evaluate adversarial robustness on three datasets, CIFAR-10, CIFAR-100 and Tiny-ImageNet. All models are adversarially trained using TRADES (Zhang et al., 2019) with $\gamma = 6$; we use a step size of $\alpha = 2/255$ and with 10 and 7 steps of PGD for CIFAR-10/CIFAR-100 and Tiny-ImageNet, respectively; we set the maximum perturbation strength to $\epsilon = 8/255$ to constrain the ℓ_∞-norm. For evaluating adversarial robustness, we consider multiple attacks, FGSM (Goodfellow et al., 2015), 20-step PGD (PGD^{20}) (Madry et al., 2018), 40-step CW (CW^{40}) (Carlini & Wagner, 2017), and AutoAttack (AA) (Croce & Hein, 2020) with the same perturbation constraint $\epsilon = 8/255$. We repeat each experiment two or three times and

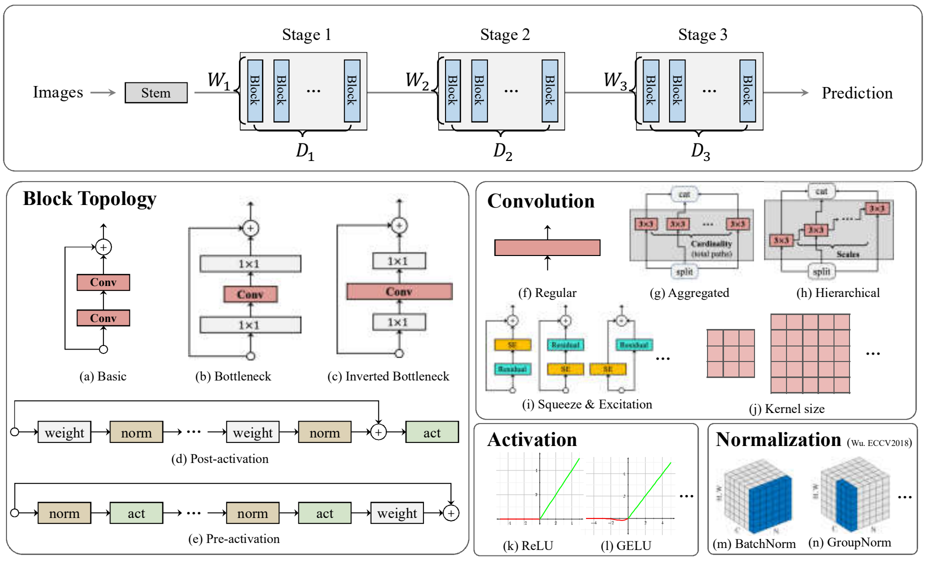


Figure 2: Overview of the architecture design we explore for adversarial robustness: (*Top*) Network level and (*Bottom*) Block level. The network has three stages, each with multiple blocks and scaling parameters depth and width. We study variants of residual blocks and their components like convolution type, activations, and normalization.

compute the mean performance to account for noise in evaluating adversarial attacks. In all results, we use markers and shaded regions show the mean and standard deviation across the repetitions.

3 Adversarially Robust Networks

Convolutional neural network design involves determining block topology, components of the block and scaling factors. We examine these elements independently through controlled experiments, and finally propose a new residual network based on our observations.

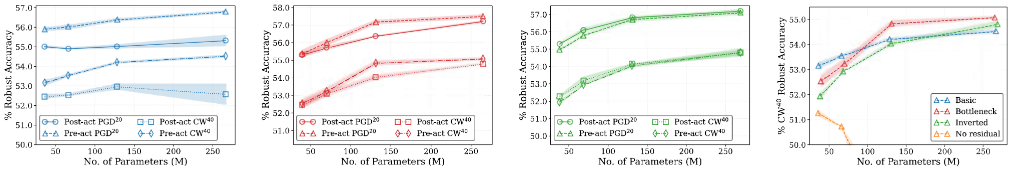
3.1 Adversarially Robust Residual Block

The design of a convolutional block primarily comprises its topology, type of convolution and kernel size, and choice of activation and normalization layers. We examine these elements independently through controlled experiments, and finally propose a new residual block based on our observations.

3.1.1 Block Topology

Residual Topology: Figure 2 shows the three primary variants of residual blocks, namely basic, bottleneck and inverted bottleneck, in the literature. Among them, the basic block is the de facto block of choice for studying adversarial robustness. Surprisingly, the bottleneck and inverted bottleneck blocks have rarely been employed for adversarial robustness, despite their well-established effectiveness under standard setting for image classification, object detection, etc. Therefore, we revisit these residual blocks in the context of adversarial robustness. And for each block, we consider two variants (post-activation (He et al., 2016a) and pre-activation (He et al., 2016b)) corresponding to placement of activation functions (see Appendix A.2 for illustration) before and after a convolutional layer. Moreover, we consider models at four capacities by varying the stage wise depth $D_{i\in\{1,2,3\}}$ and width $W_{i\in\{1,2,3\}}$ among $\{4, 5, 7, 11\}$ and $\{10, 12, 14, 16\}$, respectively.

Figure 3 compares the aforementioned variants of residual blocks. We observe that (i) the basic block is very sensitive to the location of the activation function, with pre-activation leading to a substantial improvement in adversarial robustness (Figure 3a); (ii) performance of the bottleneck and the inverted bottleneck blocks are relatively stable w.r.t the location of the activation function, although pre-activation provides a small but noticeable benefit on large-capacity models with bottle-



(a) Basic

(b) Bottleneck

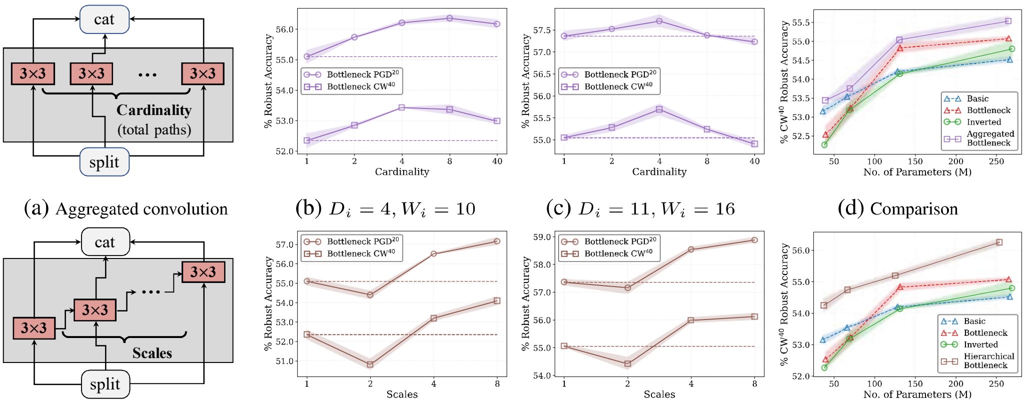
(c) Inverted Bottleneck

(d) Comparison among (a) – (c)

Figure 3: Robust accuracy of networks on CIFAR-10 with (a) basic, (b) bottleneck, and (c) inverted bottleneck blocks, with post and pre activation. (d) Comparison between blocks with pre-activation. “No residual” removes the residual connection in the basic block.

neck blocks and small-capacity models with inverted bottleneck blocks (Figures 3b and 3c). Thus, we conclude that *pre-activation is preferred over post-activation for adversarial robustness*. Figure 3d compares the three residual blocks with pre-activation. We observe that the basic block is more effective in low model-capacity regions, while the bottleneck block is more effective in high model-capacity regions. Finally, since the inverted bottleneck does not outperform the other two blocks under any model capacity, we no longer consider it in the rest of this paper.

Aggregated and Hierarchical Convolutions: Next, we consider two enhanced arrangements of convolution, *aggregated* (Xie et al., 2017) and *hierarchical* (Gao et al., 2021), which have proven to be effective for residual blocks on standard tasks. We incorporate both of them within the bottleneck block. For each enhancement, we conduct ablation experiments to determine appropriate values for their hyperparameters, i.e., *cardinality* for aggregated and *scales* for hierarchical convolutions. Figure 4 compares the bottleneck block with aggregated and hierarchical convolutions, respectively. We observe that the *bottleneck block consistently benefits from both enhancements*, and outperforms the basic block under all the model-capacity regions we considered. In contrast, we observe that when paired with the basic block, aggregated convolution adversely affects adversarial robustness. More detailed results can be found in Appendix §A.3.



(a) Aggregated convolution

(b) $D_i = 4, W_i = 10$

(c) $D_i = 11, W_i = 16$

(d) Comparison

(e) Hierarchical convolution

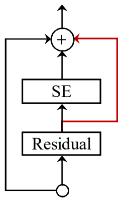
(f) $D_i = 4, W_i = 10$

(g) $D_i = 11, W_i = 16$

(h) Comparison

Figure 4: (a, e) Aggregated and hierarchical convolutions that split a regular convolution into multiple parallel convolutions (cardinality) and hierarchical convolutions (scales), respectively. Results are then concatenated. (b, f) and (c, g) show robustness of low-capacity ($^{\#}\mathrm{P} = 39\mathrm{M}, ^{\#}\mathrm{F} = 5.9\mathrm{G}$) and high-capacity ($^{\#}\mathrm{P} = 262\mathrm{M}, ^{\#}\mathrm{F} = 39\mathrm{G}$) models. (d, h) Comparing aggregated (cardinality $= 4$) and hierarchical (scales $= 8$) bottleneck to other blocks. All results are CIFAR-10.

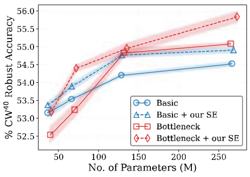
Squeeze and Excitation: Finally, we consider squeeze-and-excitation (SE) (Hu et al., 2020), which emerged as a standard component of modern CNN architectures, such as MobileNetV3 (Howard et al., 2019) and EfficientNet (Tan & Le, 2019a). However, we observe (see Table 5b) that a straightforward application of SE, and all its variants explored by Hu et al. (2020), degrades performance. We hypothesize that this may be due to the SE layer excessively suppressing or amplifying channels. Therefore, we present an alternative design of SE, dubbed *residual SE*, for adversarial robustness. As shown in Figure 5a, it adds another skip connection around the SE module, a simple yet crucial



(a) Our SE

Designs (reduction ratio)	#P (M)	#F (G)	Clean	Robust (CW^{40})
w/o SE	265	39.0	85.47	55.07
Standard SE ($r = 16$)	296	39.1	84.56 (-0.91)	54.52 (-0.55)
Conv3×3-SE ($r = 16$)	273	39.1	85.26 (-0.21)	54.77 (-0.40)
Identity-SE ($r = 16$)	293	39.1	85.20 (-0.27)	54.94 (-0.13)
Our residual SE ($r = 16$)	296	39.1	**85.75** (+0.28)	55.95 (+0.88)
Our residual SE ($r = 64$)	273	39.1	85.61 (+0.14)	**56.05** (+0.98)

(b) Ablation study on SE integration designs.



(c) Comparison

Figure 5: (a) Our *residual SE* that adds an extra skip connection around the SE module. (b) Ablation results, with relative improvement/degradation shown in parentheses. (c) Comparing residual blocks with and without our residual SE. All results are evaluated on CIFAR-10.

modification. During adversarial training, this skip connection provides extra regularization to avoid channels from being excessively suppressed or amplified by SE. Additionally, we observe (Table 5b) that a higher reduction ratio can reduce the computational complexity of the SE module at the cost of a marginal degradation in clean accuracy. Figure 5c compares the basic and bottleneck blocks with and without *residual SE*. Results indicate that our *residual* SE consistently improves the adversarial robustness of both blocks. More detailed results can be found in Appendix §A.4.

To summarize, we demonstrate, in Table 1, that all the topological enhancements we identified, i.e., pre-activation, aggregated, hierarchical convolutions, and residual SE, can be naturally integrated within the bottleneck block. Empirically, our final topology design yields a ∼3% improvement over the basic block, the de facto topology of choice for designing robust architectures.

Table 1: Break-down of the contribution of each topological enhancement we identified. Both basic and bottleneck blocks use pre-activation. Cardinality for aggregated convolutions is 4 and scales for hierarchical convolutions is 8. All results are for a large-capacity model with $D_i = 11, W_i = 16$.

Topology					Complexity		CIFAR-10			CIFAR-100		
Basic	Bottle	Aggr.	Hier.	SE	#P	#F	Clean	PGD^{20}	CW^{40}	Clean	PGD^{20}	CW^{40}
✓					267M	38.8G	$85.51_{\pm 0.19}$	$56.78_{\pm 0.13}$	$54.52_{\pm 0.13}$	$56.93_{\pm 0.49}$	$29.76_{\pm 0.14}$	$27.24_{\pm 0.15}$
	✓				265M	39.0G	$85.47_{\pm 0.21}$	$57.49_{\pm 0.21}$	$55.07_{\pm 0.10}$	$59.24_{\pm 0.36}$	$32.08_{\pm 0.26}$	$28.61_{\pm 0.17}$
	✓	✓			265M	39.4G	$85.47_{\pm 0.10}$	$57.50_{\pm 0.28}$	$55.53_{\pm 0.26}$	$59.27_{\pm 0.34}$	$31.63_{\pm 0.36}$	$28.80_{\pm 0.18}$
	✓	✓	✓		262M	39.3G	$86.29_{\pm 0.07}$	$59.48_{\pm 0.12}$	$56.94_{\pm 0.27}$	$59.32_{\pm 0.13}$	$33.46_{\pm 0.22}$	$29.65_{\pm 0.14}$
	✓	✓	✓	✓	270M	39.3G	$\mathbf{86.55}_{\pm \mathbf{0.10}}$	$\mathbf{60.48}_{\pm \mathbf{0.00}}$	$\mathbf{57.78}_{\pm \mathbf{0.09}}$	$\mathbf{60.22}_{\pm \mathbf{0.57}}$	$\mathbf{33.88}_{\pm \mathbf{0.03}}$	$\mathbf{29.91}_{\pm \mathbf{0.15}}$

3.1.2 Convolution Kernel Size

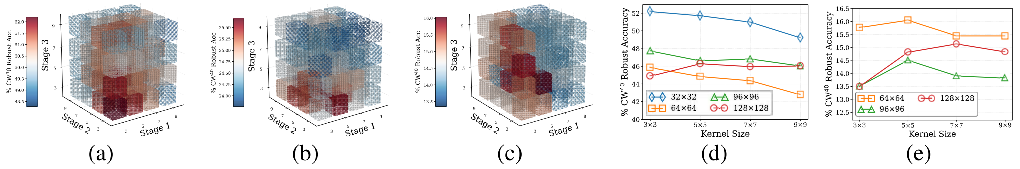
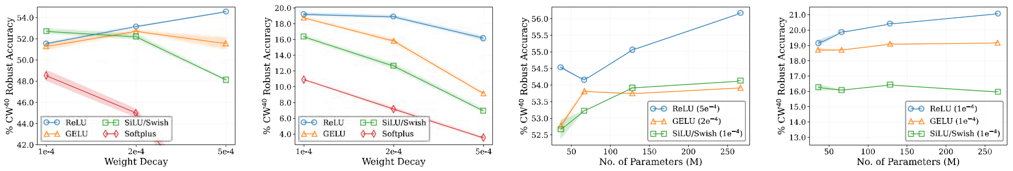


Figure 6: (a, b, c) Heat maps visualizing the relationship between kernel sizes and adversarial robustness on CIFAR-10, CIFAR-100, and Tiny-ImageNet from left to right. (d, e) The adversarial robustness of different kernel sizes for higher resolution images on C-10 (L) and Tiny-IN (R).

Larger kernel sizes have been shown to be beneficial on standard problems (Tan & Le, 2019b; Liu et al., 2022; Ding et al., 2022). We evaluate large kernel sizes for adversarial robustness. Specifically, we allow the kernel size $K_{i \in \{1,2,3\}}$ for each stage to be among $\{3\times3,\ 5\times5,\ 7\times7,\ 9\times9\}$ while use the default options for all other settings as described in §2. We evaluate all the $4^3 = 64$ possible networks with all possible settings for the kernel size. Figure 6 (a, b) shows our results. We observe that, in general, *a larger kernel size does not necessarily lead to better adversarial robustness*. To confirm if this observation is specific to low-resolution images, we repeat the experiment at higher image resolutions. Specifically, we upsample the images to the following sizes: $\{64 \times 64, 96 \times 96, 128 \times 128\}$. We constrain all stages to use a canonical kernel size and use a stride of two in the first block of the first stage when the image resolution is larger than 64×64. Figure 6 presents these results. Empirically, we observe that larger kernels start to improve adversarial robustness noticeably when the image size increases to 128×128, particularly on Tiny-ImageNet. However, adversarial robustness on upsampled images is consistently worse than that of smaller images. Thus, we argue that *a kernel size of* 3×3 *remains the preferred choice for adversarial robustness*.

3.1.3 Activation and Normalization

Activation: Since the first demonstration by Xie et al. (2020), several researchers (Pang et al., 2021; Singla et al., 2021; Gowal et al., 2020) reaffirmed that *smooth activation functions improve adversarial training*, which in turn improves adversarial robustness. However, these observations are primarily based on CIFAR-10 with low-capacity models (e.g., ResNet-18 or WRN-34-10) and for a fixed set of training hyperparameters. We hypothesize that different activation functions, regardless of being smooth or not, may perform differently depending on training hyperparameters, especially *weight decay*, as observed by Pang et al. (2021). Therefore, we revisit the adversarial robustness of smooth and non-smooth activation functions under appropriate weight decay settings. We consider ReLU (non-smooth) and three smooth activation functions, SiLU/Swish (Xie et al., 2020; Rebuffi et al., 2021; Gowal et al., 2021), Softplus (Qin et al., 2019; Pang et al., 2021), and GELU (Bai et al., 2021), given their prevalence in the literature. For each activation function, we first identify a suitable weight decay value from $\{1, 2, 5\} \times 10^{-4}$. We observe (Figure 7 (a, b)) that, (i) different activation functions indeed perform their best at different weight decay values on CIFAR-10; (ii) on Tiny-ImageNet, a weight decay of 10^{-4} works best for all activation functions. Then we compare the performance of the activation functions under their optimal weight decay values across a wide range of model capacities. Surprisingly, we observe (Figure 7 (c, d)) that ReLU consistently outperforms SiLU and GELU on both CIFAR-10 and Tiny-ImageNet, particularly in the large model-capacity regions. Our findings suggest that *ReLU is effective even in the context of adversarial robustness*, challenging the existing consensus that smooth activation is preferable for adversarial robustness.



(a) CIFAR-10 (b) Tiny-ImageNet (c) CIFAR-10 (d) Tiny-ImageNet

Figure 7: (a, b) Effect of weight decay values for different activation functions in WRN-28-10. (c, d) Robust accuracy of different activation functions, with the weight decay shown in parentheses.

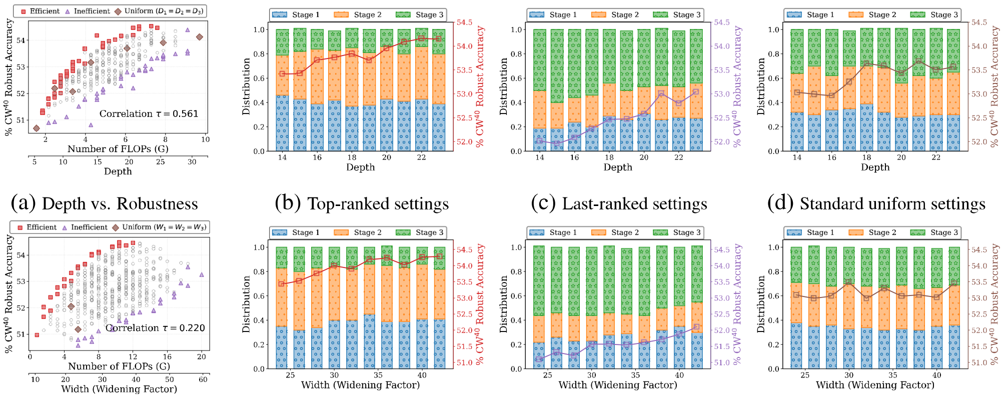
Normalization: We find that standard *BatchNorm outperforms other alternatives* such as GroupNorm (Wu & He, 2018), LayerNorm (Ba et al., 2016), and InstanceNorm (Ulyanov et al., 2016). Due to space constraints, we refer the readers to appendix §A.6 for details and results.

3.2 Adversarially Robust Scaling of Residual Networks

Scaling a network involves controlling the width and depth of its layers. We first study these elements individually and then introduce a compound scaling rule that improves adversarial robustness.

3.2.1 Independent Scaling Network Depth and Width

We independently study the relationship between adversarial robustness, and network depth (i.e., number of blocks) and network width (i.e., number of channels). We allow the depth of each stage ($D_{i\in\{1,2,3\}}$) to vary among $\{2, 3, 4, 5, 7, 9, 11\}$, and the width widening factor ($W_{i\in\{1,2,3\}}$) to vary among $\{4, 6, 8, 10, 12, 14, 16, 20\}$, while fixing the other architecture components to the baseline settings described in §2. As a result, in the case of depth variations, the number of layers in the resulting networks ranges from 16 to 70. We adversarially train all possible networks, $7^3 = 343$ for depth and $8^3 = 512$ for depth, and present the results in Figure 8a and Figure 8e, respectively. From a trade-off perspective of maximizing adversarial robustness and minimizing network complexity, we highlight the efficient, inefficient, and standard uniform depth/width settings with different colored markers. Empirically, we observe that (i) there is no substantial correlation between network depth and adversarial robustness, implying that *more blocks do not necessarily lead to better adversarial robustness*, (ii) there is only a weak correlation between network width and adversarial robustness, implying that *more channels do not automatically lead to better adversarial robustness*, and (iii) at any given total network depth/width, there is a significant variation in adversarial robustness, suggesting that *the distribution of depth/width in each stage needs to be carefully selected for improving robustness*.



(a) Depth vs. Robustness (b) Top-ranked settings (c) Last-ranked settings (d) Standard uniform settings

(e) Width vs. Robustness (f) Top-ranked settings (g) Last-ranked settings (h) Standard uniform settings

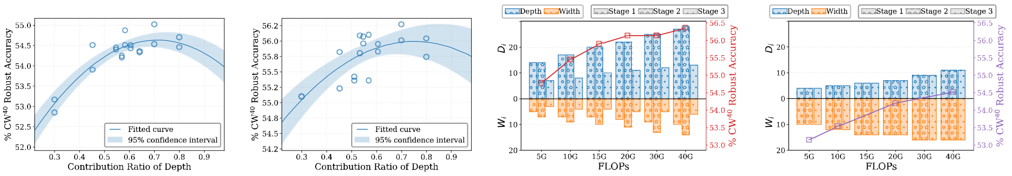
Figure 8: Adversarial robustness of 343 depth (a) and 512 width (e) settings on CIFAR-10. *Pareto-efficient* models (robust and compact) are in red squares, *inefficient* models (sensitive and complex) are in violet triangles, and networks with standard *uniform* distribution ($D_1 = D_2 = D_3$ and $W_1 = W_2 = W_3$) are in brown diamonds. Rank correlation (τ) between depth/width and robust accuracy is shown. Distribution among the three stages for the efficient (b, f), standard uniform (c, g), and inefficient (d, h) settings are shown. The secondary y-axis shows robust accuracy.

Next, we zoom in to take a closer look at each stage. Specifically, at each level of total network depth/width, we rank the settings by their adversarial robustness and visualize the distribution of blocks/widening factors among the three stages. Figure 8b shows that models which distribute more blocks evenly between the first two stages and decrease the number of blocks in the third stage are ranked higher. On the other hand, Figure 8c shows that models which distribute more blocks in the third stage and reduce the number of blocks in the first two stages are ranked last. Similarly, Figure 8f shows that top-ranked settings tend to use small widening factors in stage-3, and allocate larger widening factors to the first two stages, particularly the second stage. On the other hand, Figure 8g shows that last-ranked models use larger widening factors in the last stage by reducing the widening factors of the second stage. For both depth and width, by averaging the block/widening factor distribution in the top-ranked models at each network depth/width, we identify that distributing the depth as $D_1 : D_2 : D_3 = 2 : 2 : 1$ and width as $W_1 : W_2 : W_3 = 2 : 2.5 : 1$ across the stages leads to robust and efficient models. For completeness, we also show the depth/widening factor distribution and robust accuracy for the standard uniform depth/width settings in Figures. 8d and 8h.

3.2.2 Compound Scaling by Network Depth and Width

We study the interplay between network depth and width by searching for a ratio between total network depth and total network width, i.e., ($\sum D_i : \sum W_i$) which improves adversarial robustness. Specifically, given a target network complexity (e.g., #FLOPs), we systemically tune the contribution ratio of depth (i.e., $r_D = \sum D_i / (\sum D_i + \sum W_i)$) between $[0.3, 0.95)$ and record the relative changes in adversarial robustness. As shown in Figure 9 (a, b), we observe that adversarial robustness improves monotonically as r_D increases and peaks at approximately $r_D = 0.7$, suggesting that *deep but narrow networks are preferred over wide but shallow networks for adversarial robustness* at a given FLOPs count. However, as the r_D continues to increase beyond 0.7, adversarial robustness starts to deteriorate rapidly. Accordingly, our compound scaling rule is obtained by solving $r_D = 0.7 = \frac{D_1+D_2+D_3}{D_1+D_2+D_3+W_1+W_2+W_3} = \frac{2D_3+2D_3+D_3}{2D_3+2D_3+D_3+2W_3+2.5W_3+W_3}$ such that the #FLOPs($\sum D_i, \sum W_i$) $\approx$ the target.

A pictorial illustration of the final compound scaling rule is provided in Figure 9c, along with the standard scaling rule in Figure 9d as a reference. Finally, we present a comparison between the



(a) 5G FLOPs (b) 20G FLOPs (c) Compound scaling (d) Standard scaling

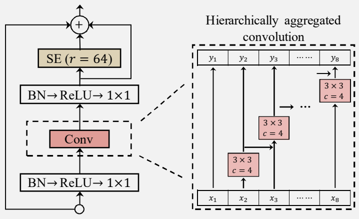
Figure 9: (a, b) Adversarial robustness vs. contribution ratio of depth (r_D) at different FLOPs levels, where $r_D = \sum D_i / (\sum D_i + \sum W_i)$. A larger r_D indicates a deeper (more blocks) but narrower (fewer channels) network. Distribution of depth and width among the three stages for the compound scaling (c) and the standard scaling (d) rules. The secondary y-axis shows robust accuracy.

standard scaling, the independent depth/width scaling, and the compound scaling in Figure 10. We observe that both the independent scaling of depth or width and the compound scaling lead to substantial improvements in robust accuracy over the standard scaling across a wide spectrum of model capacities. This reaffirms our hypothesis that *architecture design contributes significantly to adversarial robustness*. In general, among the four scaling strategies, compound scaling provides the best trade-off between adversarial robustness and network complexity. In particular, the least complex model from the compound scaling is more adversarially robust than the most complex model from the standard scaling, while being 14× more compact (parameters) and 8× more efficient (FLOPs).

Summary of Our RobustResNet

Our Robust Residual Block (RobustResBlock): Building upon the empirical evidence from §3.1.1 - §3.1.3, we propose a new residual block design, dubbed RobustResBlock, to substitute the basic block in architectures designed for adversarial robustness.

- *Block Topology:* Bottleneck block with pre-activation, hierarchically aggregated convolution, and residual SE.
- *Kernel Size:* standard conv with 3×3 filter.
- *Activation:* ReLU
- *Normalization:* Batch Normalization



Our Compound Scaling Rule: Network scaling contributes significantly to adversarial robustness. The following rules are derived for a three-stage network:

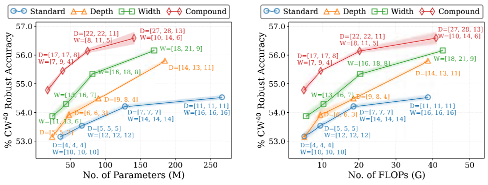
– *Ratio between Depth and Width:* $\left[\sum D_i : \sum W_i\right] = [7:3]$ for $i \in \{1, 2, 3\}$ (§3.2.1)

– *Depth/Width Distribution:* $D_1 : D_2 : D_3 = 2 : 2 : 1$, $W_1 : W_2 : W_3 = 2 : 2.5 : 1$ (§3.2.2)

– *Wide or Deep:* For a given desired FLOPs budget, *deep (but narrow) networks are adversarially more robust than wide (but shallow) networks.*

3.3 Adversarially Robust Residual Networks

We combine our RobustResBlock with the identified compound scaling rule to present a portfolio of adversarially robust residual networks, dubbed *RobustResNets*, spanning a wide spectrum of model capacities (5G - 40G FLOPs). For references, we name them as RobustResNet-A1 to -A4, where #FLOPs are doubled for every subsequent network of A1 (see Table 2 for specifics). We then compare RobustResNets to a set of representative robust architectures proposed in the literature. These include, RobNet (Guo et al., 2020), RACL (Dong et al., 2020), AdvRush (Mok et al., 2021), and WRN-34-R (Huang et al., 2021a). Specifically, we align the network complexity of AdvRush and RACL models by adjusting the number of repetitions of the normal cell N and the input #channels of the first normal cell C, denoted as (N@C). Table 2 presents the results. We observe that, overall, RobustResNets consistently outperform alternative robust models across multiple datasets, attacks, and model-capacity regions. In particular, RobustResNet-A1 achieves **2.5% higher** AutoAttack test accuracy with **2× fewer** #parameters than AdvRush, a robust block designed by differentiable



(a) CW^{40} vs. #Params

(b) CW^{40} vs. #FLOPs

Figure 10: Comparison between standard, the identified independent depth/width, and compound scaling on CIFAR-10. $[D_1, D_2, D_3]$ and $[W_1, W_2, W_3]$ denote stage wise depth and width settings, respectively. For efficient depth scaling, we use the width settings from the standard scaling and vice-versa for efficient width scaling.

neural architecture search; RobustResNet-A2 achieves **2.3% higher** AutoAttack test accuracy with **1.8× fewer** #parameters and #FLOPs than WRN-34-R from Huang et al. (2021a), who also studied the impact of network depth and width on adversarial attacks.

Table 2: Comparison of robust accuracy. Best results are in bold, and relative improvements over 2$^{\text{nd}}$ best result in each section is colored in red. See text for details.

Model	#P (M)	#F (G)	CIFAR-10				CIFAR-100			
			Clean	PGD^{20}	CW^{40}	AA	Clean	PGD^{20}	CW^{40}	AA
WRN-28-10	36.5	5.20	84.62	55.90	53.15	51.66	56.30	29.91	26.22	25.26
RobNet-large-v2	33.3	5.10	84.57	52.79	48.94	47.48	55.27	29.23	24.63	23.69
AdvRush (7@96)	32.6	4.97	84.95	56.99	53.27	52.90	56.40	30.40	26.16	25.27
RACL (7@104)	32.5	**4.93**	83.91	55.98	53.22	51.37	56.09	30.38	26.65	25.65
RobustResNet-A1	**19.2**	5.11	**85.46** (↑ **0.5**)	**58.74** (↑ **1.8**)	**55.72** (↑ **2.6**)	**54.42** (↑ **2.5**)	**59.34** (↑ **2.9**)	**32.70** (↑ **2.3**)	**27.76** (↑ **1.6**)	**26.75** (↑ **1.5**)
WRN-34-12	66.5	**9.60**	84.93	56.01	53.53	51.97	56.08	29.87	26.51	25.47
WRN-34-R	68.1	19.1	85.80	57.35	54.77	53.23	58.78	31.17	27.33	26.31
RobustResNet-A2	**39.0**	10.8	**85.80** (↑ **0.0**)	**59.72** (↑ **2.4**)	**56.74** (↑ **2.0**)	**55.49** (↑ **2.3**)	**59.38** (↑ **0.6**)	**33.0** (↑ **1.8**)	**28.71** (↑ **1.4**)	**27.68** (↑ **1.4**)
WRN-46-14	128	**18.6**	85.22	56.37	54.19	52.63	56.78	30.03	27.27	26.28
RobustResNet-A3	**75.9**	19.9	**86.79** (↑ **1.6**)	**60.10** (↑ **3.7**)	**57.29** (↑ **3.1**)	**55.84** (↑ **3.2**)	**60.16** (↑ **3.4**)	**33.59** (↑ **3.6**)	**29.58** (↑ **2.3**)	**28.48** (↑ **2.2**)
WRN-70-16	267	**38.8**	85.51	56.78	54.52	52.80	56.93	29.76	27.20	26.12
RobustResNet-A4	**147**	39.4	**87.10** (↑ **1.6**)	**60.26** (↑ **3.5**)	**57.9** (↑ **3.4**)	**56.29** (↑ **3.5**)	**61.66** (↑ **4.7**)	**34.25** (↑ **4.5**)	**30.04** (↑ **2.8**)	**29.00** (↑ **2.9**)

4 Discussion and Related Work

There have been a few attempts to explore the impact of architectural components on adversarial robustness. (1) Cazenavette et al. (2021) showed that **residual connections** significantly aid adversarial robustness. (2) Huang et al. (2021a) showed that **reducing the capacity of the third stage** leads to better adversarial robustness. (3) Xie et al. (2020) showed that **smooth activation functions** leads to better adversarial robustness on ImageNet, with a similar observation by Pang et al. (2021) on CIFAR-10 with ResNet-18. However, neither of them study the correlation between robust accuracy and activation across weight decay, model capacity and dataset. Dai et al. (2022) identified that parameterized activation functions have better robustness properties. (4) There is no clear consensus on the **impact of depth/width** on adversarial robustness. Zhu et al. (2022) conclude that width helps robustness in the over-parameterized regime, but depth can help only under certain initialization. Gowal et al. (2020) conclude that deeper models perform better, while Mok et al. (2021) conclude that there is no clear relationship between the width and the depth of an architecture and its robustness. Finally, Xie et al. (2020) show that compound scaling will produce a much stronger model than scaling up a single dimension with a simple strategy. None of the aforementioned work study impact of *all* the architectural components, as this paper seeks to.

5 Conclusion

Novel architectural designs played a critical role in the overwhelming success of CNNs in a variety of image analysis tasks. Despite this knowledge, studies on adversarial robustness have largely been limited to a handful of basic residual networks, thus overlooking the impact of architecture on adversarial robustness. However, as we demonstrate in this paper, architectural design does have a significant effect on adversarial robustness. As an illustration, we considered residual networks and observed through systematically designed experiments that many advancements of residual blocks for standard tasks translate well to improve adversarial robustness, albeit with minor modifications in some cases. Based on these observations, we design RobustResNets as an alternative baseline for standard Wide Residual Networks, the de facto architecture of choice for designing adversarially robust networks. We hope that our work inspires future exploration into the adversarial robustness of the wide range of architectures that have already proven to be effective for standard tasks.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 6, 15

Yutong Bai, Jieru Mei, Alan Yuille, and Cihang Xie. Are transformers more robust than CNNs? In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. 6

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017. 2, 13

George Cazenavette, Calvin Murdock, and Simon Lucey. Architectural adversarial robustness: The case for deep pursuit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7150–7158, 2021. 2, 9

Hanlin Chen, Baochang Zhang, Song Xue, Xuan Gong, Hong Liu, Rongrong Ji, and David Doermann. Anti-bandit neural architecture search for model defense. In *European Conference on Computer Vision*, pp. 70–85, 2020. 13

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2021. 13

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020. 2, 13

Sihui Dai, Saeed Mahloujifar, and Prateek Mittal. Parameterizing activation functions for adversarial robustness. In *IEEE Security and Privacy Workshops (SPW)*, pp. 80–87. IEEE, 2022. 9

Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11963–11975, 2022. 5

Minjing Dong, Yanxi Li, Yunhe Wang, and Chang Xu. Adversarially robust neural architectures. *arXiv preprint arXiv:2009.00902*, 2020. 8

Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2):652–662, 2021. 4

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 2, 13

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020. 6, 9

Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy Mann. Improving robustness using generated data. In *Advances in Neural Information Processing Systems*, 2021. 6, 13

Minghao Guo, Yuzhe Yang, Rui Xu, Ziwei Liu, and Dahua Lin. When nas meets robustness: In search of robust architectures against adversarial attacks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 8, 13

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016a. 3, 13

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision (ECCV)*, 2016b. 3, 13

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2019. 4, 15

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018. 15

Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):2011–2023, 2020. doi: 10.1109/TPAMI.2019.2913372. 4, 15

Hanxun Huang, Yisen Wang, Sarah Erfani, Quanquan Gu, James Bailey, and Xingjun Ma. Exploring architectural ingredients of adversarially robust deep neural networks. *Advances in Neural Information Processing Systems*, 2021a. 8, 9, 13

Shihua Huang, Zhichao Lu, Ran Cheng, and Cheng He. Fapn: Feature-aligned pyramid network for dense image prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021b. 15

Jia Liu and Yaochu Jin. Multi-objective search of robust neural architectures against multiple types of adversarial attacks. *Neurocomputing*, 453:73–84, 2021. 13

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022. 5

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1, 2, 13

Jisoo Mok, Byunggook Na, Hyeokjun Choe, and Sungroh Yoon. Advrush: Searching for adversarially robust neural architectures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12322–12332, 2021. 8, 9, 13

Xuefei Ning, Junbo Zhao, Wenshuo Li, Tianchen Zhao, Yin Zheng, Huazhong Yang, and Yu Wang. Discovering robust convolutional architecture at targeted capacity: A multi-shot approach. *arXiv preprint arXiv:2012.11835*, 2020. 13

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. In *International Conference on Learning Representations*, 2021. 6, 9

Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. *Advances in Neural Information Processing Systems*, 32, 2019. 6

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Data augmentation can improve robustness. In *Advances in Neural Information Processing Systems*, 2021. 6, 13

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020. 13

Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? In *International Conference on Learning Representations*, 2022. 13

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019. 1

Vasu Singla, Sahil Singla, Soheil Feizi, and David Jacobs. Low curvature activations reduce overfitting in adversarial training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16423–16433, 2021. 6

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 13

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019a. 4

Mingxing Tan and Quoc V Le. Mixconv: Mixed depthwise convolutional kernels. *arXiv preprint arXiv:1907.09595*, 2019b. 5

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 6, 15

Hongjun Wang and Yisen Wang. Self-ensemble adversarial training for improved robustness. In *International Conference on Learning Representations*, 2022. 13

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019. 1

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020. 13

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020. 1

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018. 6, 15

Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020. 6, 9

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 1

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019. 1, 2, 13

Zhenyu Zhu, Fanghui Liu, Grigorios G Chrysos, and Volkan Cevher. Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization). *arXiv preprint arXiv:2209.07263*, 2022. 9