Revisiting Residual Networks for Adversarial Robustness (Supplementary Materials)

Shihua Huang[1] Zhichao Lu[2] Kalyanmoy Deb[1] Vishnu Naresh Boddeti[1]
[1] Michigan State University [2] Sun Yat-sen University
{shihuahuang95, luzhichaocn}@gmail.com {kdeb, vishnu}@msu.edu

A. Appendix

A.1. Background and Related Work Continued

Adversarial white-box attacks. Since the first demonstration that high-performant DNNs are vulnerable to small perturbations in inputs (a.k.a. adversarial examples) [26], a plethora of efforts have been devoted to crafting stronger adversarial examples (AEs) – fast gradient sign method (FGSM) [9] is one of the earliest methods that applies a single gradient step to generate AEs; projected gradient descent (PGD) [21] is a widely studied method that performs well in most cases while being computationally efficient; Carlini & Wagner (CW) [1] introduced an alternative loss that exhibits strong attack performance; AutoAttack (AA) [5] is an aggregated attack formed from an ensemble of four complementary attacks.

Relation to existing works based on NAS. Several recent works sought to find more robust DNN architectures via neural architecture search (NAS) – Guo *et al.* applied a one-shot NAS algorithm to design the topology of a cell structure (i.e., operations and connections among them) while leaving the network skeleton (i.e., width and depth) to human designs [11]; Mok *et al.* incorporated the smoothness of a DNN model's input loss landscape as an additional regularizer for NAS [22], among others [4, 19, 23].

These NAS-based prior arts are limited in the following three aspects: (1) they focus on only one aspect of architecture (i.e., block topology) while leaving other components (e.g., activation, network depth, and width, etc.) to human designs; (2) they treat the design of an adversarially robust architecture as a black-box search problem where minimal architectural insights can be derived; (3) NAS is computationally expensive and adversarial training makes this challenge especially acute.

In contrast, this work presents (i) a holistic study of different aspects of architecture, including block topology, activation, normalization, and scaling factors (i.e., network depth and width); (ii) through controlled and fine-grained experiments, we deliver precise knowledge on the impacts of these choices; (iii) empirically, we demonstrate that the network assembled on top of our derived knowledge outperforms existing networks designed via NAS by at least *2.5% robust accuracy against CW*^{40} (see Table 2 in the main paper).

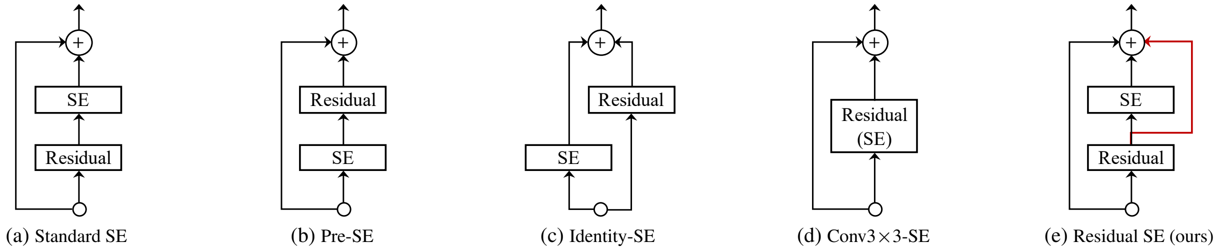
Relation to other existing works. There are recent works that aim to gain an understanding of adversarial robustness from an architectural perspective [3, 6, 10, 16, 32, 35]. Among them, [16] is most closely related to this paper. Accordingly, we provide an elaborated discussion on the relation to [16] below and refer readers to the Related Work section in §2 for an overview of these methods.

Huang *et al.* [16] also investigated the impact of network width and depth via controlled experiments on the adversarial robustness of adversarially trained DNN models. Despite a similar motivation, our work is primarily different and enhanced in the following aspects:

1. Huang *et al.* only study network scaling factors (i.e., depth and width), while we study both block topology and network scaling. And as we demonstrated in this paper, both are critical architectural components for improving adversarial robustness. Specifically, we show that (i) improvement on block topology alone leads to $\sim 3\%$ more robust accuracy; (ii) improvement on network scaling alone leads to $\sim$ 2.5% more robust accuracy; (iii) improvement on both block topology and network scaling leads to $3.5+\%$ more robust accuracy while being $\sim$ 2× more compact in terms of parameters. All results were evaluated against AutoAttack and relative to WRNs, the de-facto model for studying adversarial robustness.

2. Huang *et al.* explored the interplay between network depth and width but observed that the independent scaling rules they identified for depth and width did not work well together and ultimately failed to design a compound rule to scale depth and width simultaneously[1]. In contrast, building upon our indepen-

[1] For more details, please refer to Section 4.3 in [16].



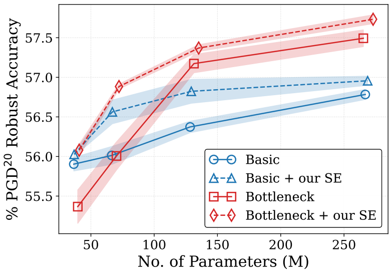
(a) Standard SE (b) Pre-SE (c) Identity-SE (d) Conv3×3-SE (e) Residual SE (ours)

Design	Explanation
(a) Standard SE	Place the SE module posterior to the main components of the residual block as proposed in [15].
(b) Pre-SE	Place the SE module a priori, i.e., before the main components of the residual block, also tried by [15].
(c) Identity-SE	Place the SE module in the skip-connection branch, also tried by [15].
(d) Conv3x3-SE	Place the SE module right after the 3×3 convolution, as done in MobileNetV3 [14].
(e) Residual SE (ours)	Add an extra skip connection around the SE module to the standard SE integration design, similarly to the FSM module from [17].

(f)

Design	Reduction ratio	#P (M)	#F (G)	Clean Acc. (%)	Robust Acc. (%) PGD^{20}	CW^{40}
w/o SE	–	265	39.0	85.47	57.49	55.07
Standard SE	$r = 16$	296	39.1	84.56 (-0.91)	56.87 (-0.62)	54.52 (-0.55)
Conv3×3-SE		273	39.1	85.26 (-0.21)	57.10 (-0.39)	54.77 (-0.40)
Identity-SE		293	39.1	85.20 (-0.27)	57.04 (-0.45)	54.94 (-0.13)
Pre-SE		293	39.1	**85.81** (+0.34)	57.31 (-0.18)	55.32 (+0.25)
Residual SE (ours)	$r = 16$	296	39.1	85.75 (+0.28)	57.86 (+0.37)	55.95 (+0.88)
	$r = 32$	281	39.1	85.22 (-0.25)	**57.98** (+0.49)	55.54 (+0.47)
	$r = 64$	273	39.1	85.61 (+0.14)	57.77 (+0.28)	**56.05** (+0.98)

(g)



(h)

Figure 1. (a) - (e) An overview of SE integration designs studied in this work. (f) Description and (g) ablation results of the SE integration designs are shown in (a) - (e). (h) Comparing residual blocks with and without the proposed residual SE on CIFAR-10 against PGD^{20} attack.

dent scaling rules, we identify an effective compound rule to simultaneously scale depth and width by properly distributing a given computational budget (e.g., FLOPs) over the number of layers and their width multipliers[2]. Empirically, we demonstrate that the compound scaling rule further improves independent scaling of depth and width by $\sim$ 2% *and* $\sim$ 1% *more robust accuracy against* CW^{40} *attack* for a small-capacity model, respectively (see Figure 9 in the main paper).

3. The scaling rule identified by Huang *et al.* was evaluated at one model capacity only (i.e., $\sim$ 68M #Params), while, in this work, we demonstrate the efficacy of our scaling rules (i.e., both independent and compound scaling rules) across a broad spectrum of model-capacities, from 5M to 270M #Params.

4. Performance-wise, on top of using almost 2× fewer #Params and #FLOPs, our model (i.e., RobustResNet-A2) consistently exhibits 1.4% - 2.4% higher robust accuracy over the model (i.e., WRN-34-R) scaled by Huang *et al.* across multiple datasets, attacks, and training settings.

Summary. To summarize, unlike this paper, none of the aforementioned prior works holistically study the impact of architectural components, i.e., block structure *and* network scaling, on adversarial robustness.

A.2. Extended Description of SE

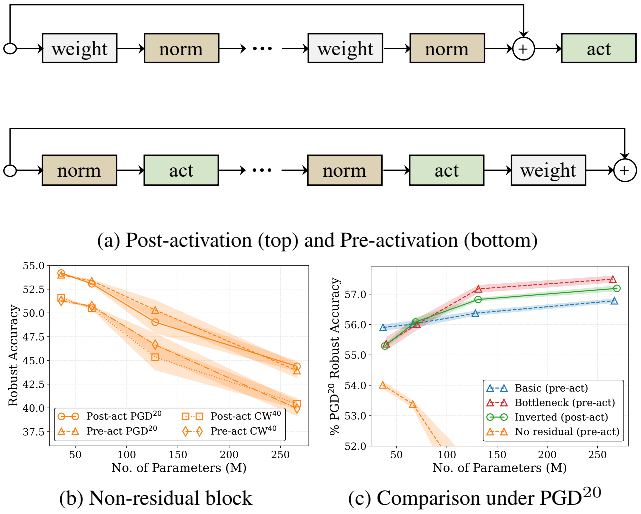
In this section, we first provide pictorial illustrations and descriptions of the five variations of SE that we tried in Figure 1a - 1e and Table 1f, respectively. Then, we provide additional results comparing our proposed residual SE among the five variations of SE in Table 1g. Our residual SE is a simple yet effective variant of the standard SE that improves adversarial robustness while all other variants fail. Finally, we present the effect of incorporating our residual SE to both basic and bottleneck residual blocks in Figure 1h.

A.3. Additional Results of Block Topology

In this section, we first provide a visual comparison between post-activation and pre-activation in Figure 2a, where the standard post-activation [12] places the activation function after the weights. In contrast, the pre-activation proposed by [13] places the activation function before the

[2] See Section 4.22 in the main paper for more details.

weights. Then, we compare the effectiveness of these two arrangements of activation for a non-residual block (i.e., VGG block) on CIFAR-10 in Figure 2b, followed by comparison over variants of residual blocks with pre-activation on CIFAR-10 against PGD^{20} attack in Figure 2c.



(a) Post-activation (top) and Pre-activation (bottom)

(b) Non-residual block

(c) Comparison under PGD^{20}

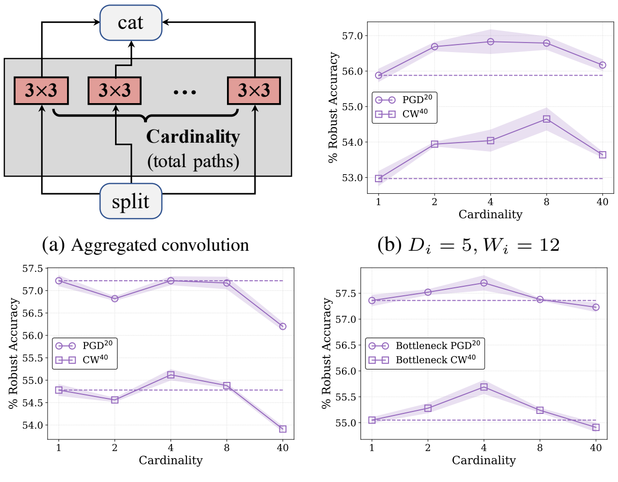
Figure 2. (a) A pictorial illustration of the standard post-activation (*Top*) and pre-activation arrangements (*Bottom*). (b) Comparing post- and pre-activation for a non-residual block (i.e., VGG block) on CIFAR-10. (c) Comparison among variants of a residual block with pre-activation on CIFAR-10 against PGD^{20} attack.

A.4. Additional Results of Aggregated and Hierarchical Convolutions

This section presents pictorial illustrations of aggregated and hierarchical convolutions in Figures 3a and 4a, respectively. Additional results showing the effects of hyperparameters cardinality (for aggregated convolution) and scales (for hierarchical convolution) are presented in Figures 3 (b, c, d) and 4 (b, c, d). Finally, we show the impact of aggregated convolution for the basic block in Figure 5, where we observe that aggregated convolution adversely affects the robustness of the basic block.

A.5. Impact of Normalization

This section investigates the relationship between normalization methods and adversarial robustness. In addition to the baseline of Batch Normalization (BN), we consider three other normalization methods, i.e., Group Normalization (GN) [31], and Instance Normalization (IN) [28]. We also confine all blocks in a DNN model to use a single choice of normalization method and repeat the experiment for each technique three times. The experimental results are summarized in Table 1. The baseline normalization method (i.e., BN) outperforms all other alternative normalization methods, particularly on Tiny-ImageNet.



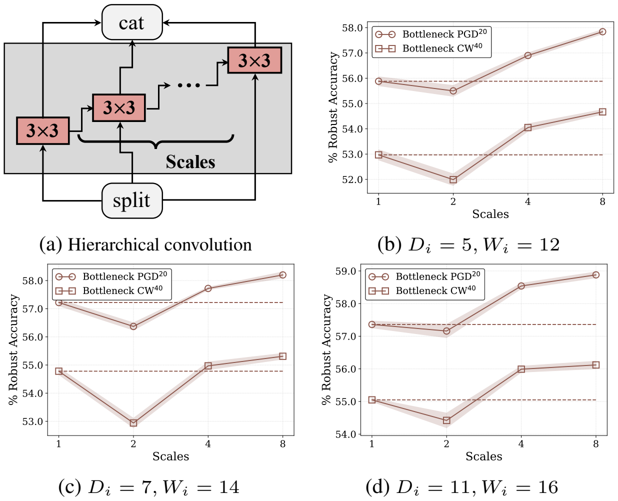
(a) Aggregated convolution

(b) $D_i = 5, W_i = 12$

(c) $D_i = 7, W_i = 14$

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

Figure 3. (a) Aggregated convolution that splits a regular convolution into multiple parallel convolutions (cardinality). Results are then concatenated. (b, c, d) show the robustness of models from three different capacity regions.



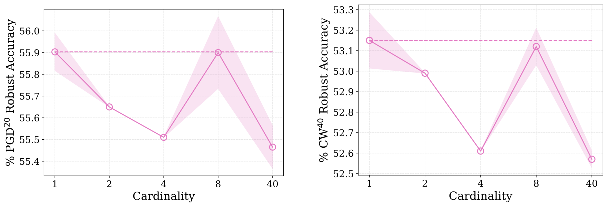
(a) Hierarchical convolution

(b) $D_i = 5, W_i = 12$

(c) $D_i = 7, W_i = 14$

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

Figure 4. (a) Hierarchical convolution that splits a regular convolution into multiple hierarchically connected convolutions (scales). Results are then concatenated. (b, c, d) show the robustness of models from three different capacity regions.



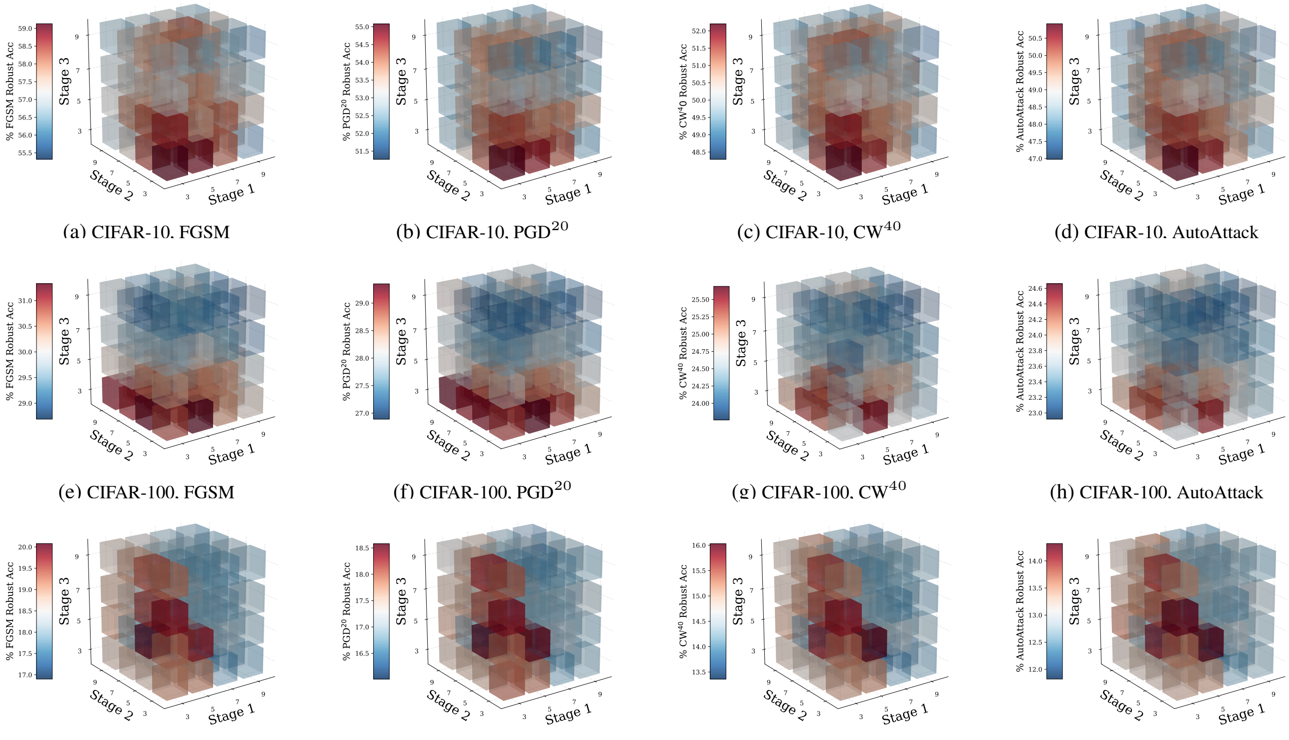
(a) PGD^{20}

(b) CW^{40}

Figure 5. The impact of aggregated convolution for the basic block. Results show the robustness of the model with $D_i = 4, W_i = 10$.

Table 1. The adversarial robustness of the considered normalization methods. We highlight the best results of each section in bold.

	CIFAR-10				CIFAR-100				Tiny-ImageNet			Ave. Rank
	Nat.	PGD^{20}	CW^{40}	AA	Nat.	PGD^{20}	CW^{40}	AA	Nat.	PGD^{20}	AA	
BN	85.11	55.36	**53.02**	**51.43**	55.77	**29.91**	26.23	**25.35**	**42.09**	**20.68**	**16.25**	**1.5**
GN	85.28	**55.82**	52.76	51.23	**56.60**	29.86	**26.26**	25.09	30.99	16.87	13.01	1.7
IN	**85.34**	54.49	50.82	49.34	56.56	28.41	24.17	22.68	17.25	10.69	8.18	2.7



(a) CIFAR-10, FGSM

(b) CIFAR-10, PGD^{20}

(c) CIFAR-10, CW^{40}

(d) CIFAR-10, AutoAttack

(e) CIFAR-100, FGSM

(f) CIFAR-100, PGD^{20}

(g) CIFAR-100, CW^{40}

(h) CIFAR-100, AutoAttack

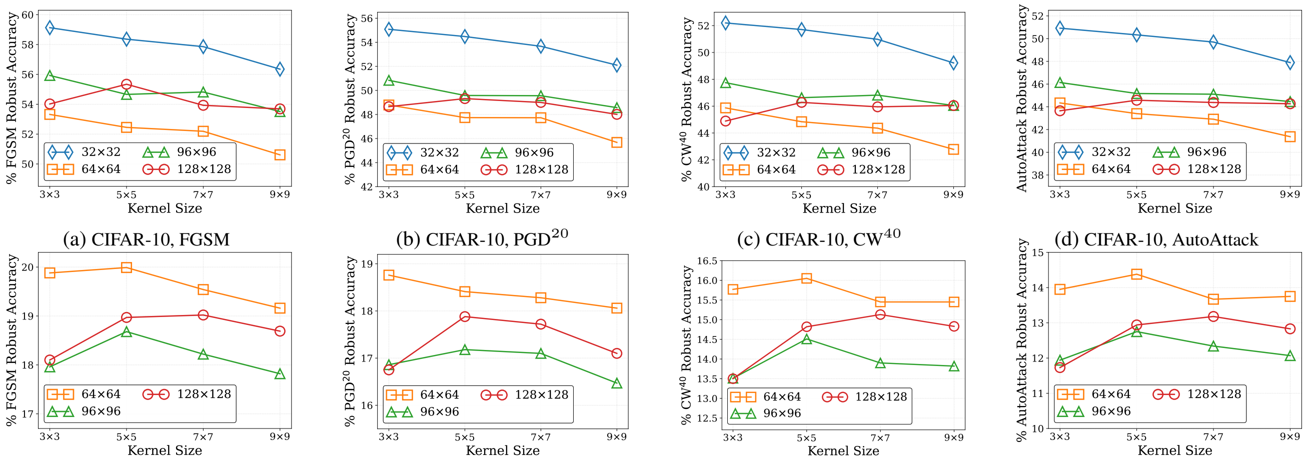
(i) Tiny-ImageNet, FGSM

(j) Tiny-ImageNet, PGD^{20}

(k) Tiny-ImageNet, CW^{40}

(l) Tiny-ImageNet, AutoAttack

Figure 6. Heat maps visualizing the relationship between kernel sizes and adversarial robustness on CIFAR-10, CIFAR-100, and Tiny-ImageNet (from top to bottom) against FGSM, PGD^{20}, CW^{40}, and AutoAttack (from left to right).



(a) CIFAR-10, FGSM

(b) CIFAR-10, PGD^{20}

(c) CIFAR-10, CW^{40}

(d) CIFAR-10, AutoAttack

(e) Tiny-ImageNet, FGSM

(f) Tiny-ImageNet, PGD^{20}

(g) Tiny-ImageNet, CW^{40}

(h) Tiny-ImageNet, AutoAttack

Figure 7. The adversarial robustness of different kernel sizes for higher resolution images on CIFAR-10 (Top) and Tiny-ImageNet (Bottom) against FGSM, PGD^{20}, CW^{40}, and AutoAttack (from left to right).

Table 2. The specifications of RobustResNets. The stage wise setting is presented using $\begin{bmatrix} k \times k, \text{\#Ch} \end{bmatrix}$, where k denotes the convolution filter size, #Ch denotes the number of output channels, and $\begin{bmatrix} \cdot \end{bmatrix}$ indicates our RobustResBlock identified in §4.2.

	Output scale	RobustResNet-A1	RobustResNet-A2	RobustResNet-A3	RobustResNet-A4
Stem	32×32	3×3, 16, stride 1			
Stage 1	32×32	$\begin{bmatrix} 1 \times 1, 160 \\ 3 \times 3, 80 \\ 1 \times 1, 320 \end{bmatrix} \times 14$	$\begin{bmatrix} 1 \times 1, 224 \\ 3 \times 3, 224 \\ 1 \times 1, 448 \end{bmatrix} \times 17$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 512 \end{bmatrix} \times 22$	$\begin{bmatrix} 1 \times 1, 320 \\ 3 \times 3, 320 \\ 1 \times 1, 640 \end{bmatrix} \times 27$
Stage 2	16×16	$\begin{bmatrix} 1 \times 1, 448 \\ 3 \times 3, 448 \\ 1 \times 1, 896 \end{bmatrix} \times 14$	$\begin{bmatrix} 1 \times 1, 576 \\ 3 \times 3, 576 \\ 1 \times 1, 1152 \end{bmatrix} \times 17$	$\begin{bmatrix} 1 \times 1, 704 \\ 3 \times 3, 704 \\ 1 \times 1, 1408 \end{bmatrix} \times 22$	$\begin{bmatrix} 1 \times 1, 896 \\ 3 \times 3, 896 \\ 1 \times 1, 1792 \end{bmatrix} \times 28$
Stage 3	8×8	$\begin{bmatrix} 1 \times 1, 384 \\ 3 \times 3, 384 \\ 1 \times 1, 768 \end{bmatrix} \times 7$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 1024 \end{bmatrix} \times 8$	$\begin{bmatrix} 1 \times 1, 640 \\ 3 \times 3, 640 \\ 1 \times 1, 1280 \end{bmatrix} \times 11$	$\begin{bmatrix} 1 \times 1, 768 \\ 3 \times 3, 768 \\ 1 \times 1, 1536 \end{bmatrix} \times 13$
Tail	1×1	Global average pool			

A.6. Impact of Convolution Kernel Size

Larger kernel sizes have been shown to be beneficial on standard problems [7, 20, 27] under standard ERM training. 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 using the default options for all other settings as described in §3 in the main paper. We evaluate all the $4^3 = 64$ possible networks with all possible settings for the kernel size. Figure 6 shows our results. We observe that, in general, *a larger kernel size does not necessarily lead to better adversarial robustness*. We repeat the experiment at higher image resolutions to verify if this observation is specific to low-resolution images. 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 higher than 64×64. Figure 7 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*.

A.7. Extended Discussion and Results of RobustResNets

In this section, we first provide detailed specifications of RobustResNetA1 - A4 in Table 2.

Comparison under Baseline AT with different loss functions: Then, we present additional comparisons under baseline adversarial training methods with different loss functions (i.e., SAT [21], and MART [29]) in Table 3. We observe that the improvements afforded by RobustResNets generalize well to other loss formulations under baseline adversarial training routines.

Comparison under Advanced AT: Table 4 presents results under advanced AT with an additional 500K unlabeled external images [2]. RobustResNet-A1 achieves 63.70% AutoAttack robust accuracy with 19.2M parameters. Furthermore, RobustResNet-A1 is $\sim$ 1.2% *more robust* against AutoAttack with 3.5× *fewer parameters* and 3.7× *fewer FLOPs* than WRN-34-R [16].

Table 5 presents results under advanced AT without additional data, either external or generated by generative models. In particular, RobustResNet-A1 achieves 1.9% *higher AutoAttack robust accuracy* with 1.9× *fewer parameters* than state-of-the-art[3] method built upon WRN-28-10 [25]. Furthermore, RobustResNet-A2 is 6.8× *more compact (parameters)* and 3.7× *more efficient (FLOPs)* while matching the state-of-the-art AutoAttack robust accuracy. And RobustResNet-A4 achieves 61.10% AutoAttack robust accuracy with 39.4M parameters on CIFAR-10 against AutoAttack with ℓ_∞ perturbations of size $\epsilon = 8/255$—*an improvement of* 1.0% *robust accuracy with* 120 *million fewer parameters* compared to the state-of-the-art without external or generated data [25].

A.8. Discussion

This paper identified specific architectural design elements that impact adversarial robustness. The reliability of our observations has been ensured by systematically verifying them on multiple datasets, across multiple adversarial attacks, and over multiple repetitions. We affirm that the proposed RobustResBlock, RobustScaling, and RobustResNet have immediate practical relevance in designing adversarially robust networks. Nonetheless, our observations and contributions have been made through empirical experiments instead of theoretical analysis. However, as is often the case in deep learning (e.g., batch normalization [18], lot-

[3] We consider state-of-the-art without external or generated data. Note that the "Extra data" column of the RobustBench CIFAR-10 leaderboard only accounts for external data; please also see the description under the "Method" column for approaches that do leverage *generated* data.

Table 3. Additional comparison of white-box adversarial robustness under baseline adversarial training with SAT [21], and MART [29]. The best results are in bold, and relative improvements are in red. Results are averaged over three runs with different seeds.

Model	#P	#F	SAT [21]		MART [29]	
	(M)	(G)	PGD^{20}	CW^{40}	PGD^{20}	CW^{40}
WRN-28-10	36.5	5.20	$52.44_{\pm 0.36}$	$50.97_{\pm 0.09}$	$57.69_{\pm 0.11}$	$52.88_{\pm 0.28}$
RobustResNet-A1	**19.2**	**5.11**	**57.62** (↑ 5.2)	**56.06** (↑ 5.1)	**59.34** (↑ 1.7)	**54.42** (↑ 1.5)
WRN-34-12	66.5	**9.60**	$52.85_{\pm 0.40}$	$51.36_{\pm 0.33}$	$57.40_{\pm 0.13}$	$53.11_{\pm 0.00}$
RobustResNet-A2	**39.0**	10.8	**58.39** (↑ 5.5)	**56.99** (↑ 5.6)	**60.33** (↑ 2.9)	**55.51** (↑ 2.4)
WRN-46-14	128	**18.6**	$53.67_{\pm 0.03}$	$52.95_{\pm 0.04}$	$58.43_{\pm 0.15}$	$54.32_{\pm 0.17}$
RobustResNet-A3	**75.9**	19.9	**58.81** (↑ 5.1)	**57.60** (↑ 4.7)	**60.95** (↑ 2.5)	**56.52** (↑ 2.2)
WRN-70-16	267	**38.8**	$54.12_{\pm 0.08}$	$50.52_{\pm 0.18}$	$58.15_{\pm 0.28}$	$54.37_{\pm 0.07}$
RobustResNet-A4	**147**	39.4	**59.01** (↑ 4.9)	**57.85** (↑ 7.3)	**61.88** (↑ 3.7)	**57.55** (↑ 3.2)

Table 4. Comparison of white-box adversarial robustness under advanced adversarial training with extra 500k external data [2].

Method	Architecture	#P (M)	#F (G)	AutoAttack
RST [2]	WRN-28-10	36.5	5.20	59.53
AWP [30]	WRN-28-10	36.5	5.20	60.04
HAT [24]	WRN-28-10	36.5	5.20	62.50
Gowal *et al.* [10]	WRN-28-10	36.5	5.20	62.80
Huang *et al.* [16]	WRN-34-R	68.1	19.1	62.54
Ours	RobustResNet-A1	**19.2** (↓ 3.5×)	**5.11** (↓ 3.7×)	**63.70** (↑ 1.2)

Table 5. Comparison of white-box adversarial robustness under advanced adversarial training. Our method builds upon Rebuffi *et al.* [25], which applies CutMix [33] data augmentation.

Method	Architecture	#P (M)	#F (G)	AutoAttack
TRADES [34]	WRN-34-10	46.2	6.66	53.08
Rebuffi *et al.* [25]	WRN-28-10	36.5	5.20	57.50
Ours	RobustResNet-A1	**19.2** (↓ 1.9×)	**5.11** (∼)	**59.39** (↑ 1.9)
Gowal *et al.* [10]	WRN-70-16	267	**38.8**	57.20
Rebuffi *et al.* [25]	WRN-70-16	267	**38.8**	60.07
Ours	RobustResNet-A2	**39.0** (↓ 6.8×)	**10.6** (↓ 3.7×)	60.00 (∼)
Ours	RobustResNet-A4	147 (↓ 1.8×)	39.4 (∼)	**61.10** (↑ 1.0)

tery ticket hypothesis [8], etc.), theoretical analysis usually follows empirical observations. Furthermore, most of the theoretical studies in adversarial robustness have focused on loss formulation. We hope this paper inspires theoretical exploration of the adversarial robustness properties of different architectural design elements as well.

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE symposium on security and privacy (sp)*, 2017. 1

[2] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Adv. Neural Inform. Process. Syst.*, 2019. 5, 6

[3] George Cazenavette, Calvin Murdock, and Simon Lucey. Architectural adversarial robustness: The case for deep pursuit. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 1

[4] Hanlin Chen, Baochang Zhang, Song Xue, Xuan Gong, Hong Liu, Rongrong Ji, and David Doermann. Anti-bandit neural architecture search for model defense. In *Eur. Conf. Comput. Vis.*, 2020. 1

[5] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Int. Conf. Mach. Learn.*, 2020. 1

[6] Sihui Dai, Saeed Mahloujifar, and Prateek Mittal. Parameterizing activation functions for adversarial robustness. In *IEEE Security and Privacy Workshops*, 2022. 1

[7] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 5

[8] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *Int. Conf. Learn. Represent.*, 2019. 6

[9] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Int. Conf. Learn. Represent.*, 2015. 1

[10] 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. 1, 6

[11] 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 Conf. Comput. Vis. Pattern Recog.*, 2020. 1

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Eur. Conf. Comput. Vis.*, 2016. 2

[14] 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 *Int. Conf. Comput. Vis.*, 2019. 2

[15] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(8):2011–2023, 2020. 2

[16] Hanxun Huang, Yisen Wang, Sarah Erfani, Quanquan Gu, James Bailey, and Xingjun Ma. Exploring architectural ingredients of adversarially robust deep neural networks. *Adv. Neural Inform. Process. Syst.*, 2021. 1, 5, 6

[17] Shihua Huang, Zhichao Lu, Ran Cheng, and Cheng He. Fapn: Feature-aligned pyramid network for dense image prediction. In *Int. Conf. Comput. Vis.*, 2021. 2

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Int. Conf. Mach. Learn.*, 2015. 5

[19] Jia Liu and Yaochu Jin. Multi-objective search of robust neural architectures against multiple types of adversarial attacks. *Neurocomputing*, 453:73–84, 2021. 1

[20] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 5

[21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Int. Conf. Learn. Represent.*, 2018. 1, 5, 6

[22] Jisoo Mok, Byunggook Na, Hyeokjun Choe, and Sungroh Yoon. Advrush: Searching for adversarially robust neural architectures. In *Int. Conf. Comput. Vis.*, 2021. 1

[23] 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. 1

[24] Rahul Rade and Seyed-Mohsen Moosavi-Dezfooli. Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *Int. Conf. Learn. Represent.*, 2021. 6

[25] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Data augmentation can improve robustness. In *Adv. Neural Inform. Process. Syst.*, 2021. 5, 6

[26] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Int. Conf. Learn. Represent.*, 2014. 1

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

[28] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 3

[29] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *Int. Conf. Learn. Represent.*, 2020. 5, 6

[30] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Adv. Neural Inform. Process. Syst.*, 2020. 6

[31] Yuxin Wu and Kaiming He. Group normalization. In *Eur. Conf. Comput. Vis.*, pages 3–19, 2018. 3

[32] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020. 1

[33] 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 *Int. Conf. Comput. Vis.*, 2019. 6

[34] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *Int. Conf. Mach. Learn.*, 2019. 6

[35] 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. 1