Robust Principles: Architectural Design Principles for Adversarially Robust CNNs

ShengYun Peng[1]
https://shengyun-peng.github.io/
Weilin Xu[2]
https://xuweilin.org/
Cory Cornelius[2]
https://dxoig.mn/
Matthew Hull[1]
https://matthewdhull.github.io
Kevin Li[1]
https://www.kevinyli.com/
Rahul Duggal[1]
http://www.rahulduggal.com/
Mansi Phute[1]
https://mphute.github.io/
Jason Martin[2]
jason.martin@intel.com
Duen Horng Chau[1]
https://faculty.cc.gatech.edu/~dchau

[1] Georgia Institute of Technology
Atlanta, GA, USA
[2] Intel Corporation
Hillsboro, OR, USA

Abstract

We aim to unify existing works' diverging opinions on how architectural components affect the adversarial robustness of CNNs. To achieve our goal, we synthesize a suite of generalizable robust architectural design principles: (a) optimal range for *depth* and *width* configurations, (b) preferring *convolutional* over *patchify* stem stage, and (c) robust residual block design by adopting squeeze and excitation blocks, and non-parametric smooth activation functions. Through extensive experiments across a wide spectrum of *dataset scales*, *adversarial training methods*, *model parameters*, and *network design spaces*, our principles consistently and markedly improve AutoAttack accuracy: 1–3 percentage points (pp) on CIFAR-10 and CIFAR-100, and 4–9 pp on ImageNet. The code is publicly available at https://github.com/poloclub/robust-principles.

1 Introduction

Convolutional neural networks (CNNs) and Transformers are staples in computer vision research [11, 15], but they are vulnerable to adversarial attacks [13, 31, 40], and adversarial

© 2023. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

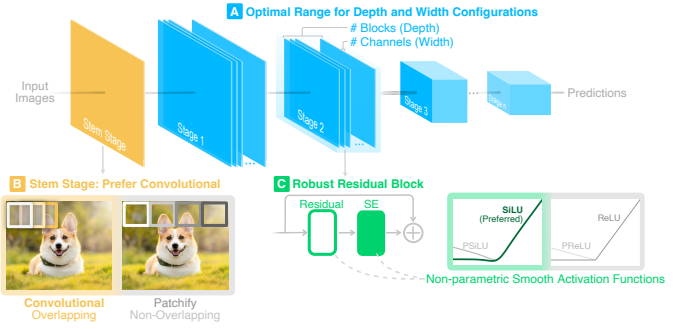

Figure 1: We synthesize a suite of generalizable architectural design principles to robustify CNNs, spanning a network's macro and micro designs: (A) optimal range for depth and width configurations, (B) preferring *convolutional* over *patchify* stem stage, and (C) robust residual block design by adopting squeeze and excitation blocks, and non-parametric smooth activation functions. The principles consistently and markedly improve AutoAttack accuracy for CIFAR-10, CIFAR-100, and ImageNet over the wide spectrum of AT methods, model parameters, and network design spaces.

training (AT) is the most effective way to improve their robustness [29]. While CNNs and Transformers have comparable clean accuracy [27, 47], there is a disparity between the two families in robustness research. AT has robustified a wide range of Transformers [6, 31]. Yet, for CNNs, wide residual networks (WRN) [53], proposed back in 2016, is still the most studied architecture in AT research [19, 45].

Although a variety of instructions have been proposed to enhance the robustness of WRN, there are still conflicting opinions on how architectural components impact the overall robustness. For example, recent research [19, 20] explored how scaling depth and width might be correlated with improved adversarial robustness; however, Mok *et al*. [32] suggested such a relationship was unclear. As another example, for micro block design, Xie *et al*. [51] and Bai *et al*. [2] found smooth activation functions improved robustness, but Huang *et al*. [20] suggested that the performance was dependent on AT settings. Furthermore, as most existing robust architecture research experimented only on the small-scale CIFAR-10 [19, 20, 31, 32], it was unclear whether such findings might generalize to large-scale datasets, *e.g*., ImageNet [9]. For example, while prior work suggested that a straightforward application of squeeze and excitation (SE) hurts robustness on CIFAR [20], we find such a modification, in fact, *improves* robustness on ImageNet (Sec. 4.3.1). Similarly, while previous work suggested that parametric activation functions are more robust than non-parametric versions on CIFAR [5], we find the reverse holds true on ImageNet (Sec. 4.3.2). The divergent observations provide strong motivation for us to seek a suite of unified generalizable design principles that can enhance the adversarial robustness of CNNs.

To accomplish our goal, we survey and distill four key architectural components underpinning state-of-the-art (SOTA) CNNs and Transformers that boost adversarial robustness. These components span a network's macro and micro designs: *depth and width*, *stem stage*, *SE block*, and *activation*. Different from previous research on robust architectures that solely relies on the small-scale CIFAR-10, we explore the AT of all four components on the large-scale ImageNet and evaluate on its full validation set. Through extensive experiments over the wide spectrum of *dataset scales*, *AT methods*, *model parameters*, and *network design spaces*, we make the following key contributions:

1. **A suite of generalizable robust architectural design principles for CNNs** (Fig. 1):

   (a) Optimal Range for Depth and Width Configurations. Despite the popularity of 4-stage residual networks on ImageNet, existing exploration has been constrained to 3-stage designs [19, 20]. We discover a flexible depth and width scaling rule that does not place a restriction on the total number of stages, and we verify its generalizability and optimality through extensive experiments. (Sec. 4.1)

   (b) *Convolutional* over *Patchify* Stem Stage. *Convolutional* stem and *patchify* stem are commonly used in CNNs and Transformers to downsample input images due to the redundancy inherent in natural images. *Convolutional* stem uses overlapped convolution kernel to slide the input image, while *patchify* stem patchifies the input image into $p \times p$ non-overlapping patches. We discover that convolutional stem, especially with a postponed downsampling design, outperforms patchify stem due to its less-aggressive stride-two downsampling and overlapped convolution kernels. (Sec. 4.2)

   (c) Robust Residual Block Design. Our investigations on how SE block and activations affect robustness present new findings on ImageNet that differ from previous research on CIFAR. Through a hyperparameter sweep, we find the reduction ratio r in SE block is negatively correlated with robustness, a new discovery not previously reported as prior work was based on a fixed r [20]. We also confirm that non-parametric smooth activations consistently improve robustness on CIFAR-10, CIFAR-100, and ImageNet. (Sec. 4.3)

2. **Consistent and marked improvements on adversarial robustness.**

   We verify the generalization of the three design principles across a wide spectrum of *dataset scales* (CIFAR-10, CIFAR-100 [23], ImageNet [7]), *AT methods* (standard adversarial training (SAT) [29], TRADES [54], Fast-AT [46], MART [43], and diffusion-augmented AT [45]), *model parameters* (from 26M to 267M), and *network design spaces* (variants of WRN [53] and ResNet [15]). Our experiments demonstrate that all design principles consistently and markedly improve AutoAttack (AA) accuracy by 1–3 percentage points (pp) on CIFAR-10 and CIFAR-100 — boosting even the SOTA diffusion-augmented AT [45] by such amounts — and 4–9 pp on ImageNet. In particular, our robustified WRN-70-16 boosts the AA accuracy by 1.31 pp (65.02% $\rightarrow$ 66.33%) on CIFAR-10, and 0.96 pp (37.77% $\rightarrow$ 38.73%) on CIFAR-100. On ImageNet, the AA accuracy is boosted by 6.48 pp (39.78% $\rightarrow$ 46.26%) through robustified ResNet-101, and 6.94 pp (42.00% $\rightarrow$ 48.94%) through robustified WRN-101-2. Our findings unify prior works' diverging opinions on how architectural components affect robustness and highlight the benefits of exploring intrinsically robust architectural components.

2 Related Work

Adversarial training (AT). AT is an effective approach to defending against adversarial attacks [13]. Madry *et al*. [29] formulated SAT as a min-max optimization framework. Given dataset samples (x_i, y_i), network f_θ and loss function $\mathcal{L}$, the optimization is formulated as:

$$\underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{(x_i,y_i)\sim\mathbb{D}} \left[\max_{x'} \mathcal{L}\left(f_\theta, x', y\right) \right], \tag{1}$$

The inner adversarial example x' aims to find the perturbation of a given data point x that achieves a high loss and is generated on the fly during the training process. Since then,

multiple variants of SAT are proposed [1, 9, 10, 44, 48]. Our architectural research is complementary to these works on AT methods. We train on diverse AT methods, *e.g.*, SAT [29], Fast-AT [46], TRADES [54], MART [43], and diffusion-augmented AT [45] to verify that our design principles unanimously improve robustness agnostic to the training recipe.

Robust Architectures. Here we provide a brief overview of related research on robust architectures, and the extended version is in Sec. A in supplementary materials. Our key advancement over existing works is a suite of three robust architectural design principles verified on diverse dataset scales, AT methods, model parameters, and design spaces. A few research studied the impact of architectural designs on adversarial robustness [14, 34, 38, 50]. For macro network design, Huang *et al*. [19] and Huang *et al*. [20] led the exploration of the correlations between robustness improvement and scaling depth and width, but Mok *et al*. [32] suggested such a relationship was unclear. For micro block design, smooth [2, 51] and parameterized [5] activation functions can largely improve robustness on CIFAR, but Huang *et al*. [20] found the performance was dependent on AT settings. There was no clear consensus on how architectural components affect adversarial robustness. More importantly, most conclusions are drawn on CIFAR with WRN's basic block design. It was unclear whether such existing robust architectures generalize to large-scale datasets or other network design spaces. Our work provides conclusive evidence that unifies prior works' diverging opinions on how architectural components affect robustness.

3 Preliminaries

This section describes the setups for comparing CNNs and Transformers in terms of architectural design space, training techniques, and adversarial attacks.

Architectural Design Skeleton. Fig. 1 provides the CNN skeleton that supports modifications of different architectural components in our study. Specifically, the skeleton consists of a stem stage, n body stages, and a classification head. A typical body stage has multiple residual blocks, and the block type is either basic (two 3×3 convolutions) or bottleneck (1×1, 3×3, and 1×1 convolutions). For stage i, denote *depth* D_i as number of blocks, and *width* W_i as channels in 3×3 convolution. The downsampling factor is 1 in the first block of stage 1, and 2 in the first block of stage 2 to n. Unless otherwise specified, the default operations in each block are rectified linear unit (ReLU) and batch normalization (BN).

Training Techniques. We use five AT recipes: Fast-AT [46], SAT [29], TRADES [54], MART [43], and diffusion-augmented AT [45]. The architectural exploration is conducted on ImageNet [7], and we apply the same training method so that the performance difference between models can only be attributed to the difference in architectures. Due to the size of ImageNet and the slow speed of AT, we use Fast-AT [46] with a cyclic learning rate [39] and mixed-precision arithmetic [30]. After finalizing the architectural design principles, we train all models with diverse AT methods on CIFAR-10, CIFAR-100 [23] and ImageNet [7] to verify that our designs can consistently improve robustness regardless of the training recipe.

Adversarial Attacks. Projected gradient descent (PGD) [29] and AA [8] are used to evaluate adversarial robustness. PGD is a white-box attack with complete access to network architectures and parameters. For PGD, we provide a comprehensive investigation on the full ImageNet validation set with attack budgets $\varepsilon\in\{2,4,8\}/255$ and max steps $i=\{10,50,100\}$, denoted as PGD^i-ε. AA is an ensemble of one black-box and three white-box attacks. The attack budget is $\varepsilon=4/255$ on ImageNet [4], and $\varepsilon=8/255$ on CIFAR-10 and CIFAR-100 [19, 20] for AA. All attacks are ℓ_∞ bounded. When exploring individual architectural

components, we use 10-step PGD (PGD^{10}-ε) for fast evaluations.

4 Robust Architectural Design Principles

Our strategy to explore the four robust architectural components is through comparing SOTA CNNs and Transformers. These components span a network's macro and micro designs: depth and width (Sec. 4.1), stem stage (Sec. 4.2), squeeze and excitation block (Sec. 4.3.1), and activation (Sec. 4.3.2), as shown in Fig. 1. To exclude the effect of model complexity [29] and provide a fair comparison [2], we focus our investigation on the regime of ResNet-50 with ∼26M parameters in this section. In the next section (Sec. 5), we verify the generalization of these principles through extensive experiments over the wide spectrum of dataset scales, AT methods, parameter budgets, and network design spaces.

4.1 Optimal Range for Depth and Width Configurations

Macro network design involves the distribution of depth and width in each stage. Shifted windows Transformer (Swin Transformer) [26] regulates the stage compute ratio as $1:1:3:1$ or $1:1:9:1$. On the CNN side, ConvNeXt [27] reshuffles depths in all stages according to Swin Transformer and finds this design improves clean accuracy. RegNet [36] introduces a linear parameterization to assign network depth and width. As network depth and width are competing for resources when the parameter budget is fixed, it is important to study the impact of depth and width on adversarial robustness. We draw inspiration from prior robustness research to develop a more generalized scaling rule. Huang *et al*. [19] found reducing depth or width at the last stage of WRN reduces the Lipschitz constant, thus improving robustness. Huang *et al*. [20] proposed a fixed scaling ratio for WRN [53] on CIFAR-10.

Existing explorations are constrained to 3-stage networks, whereas 4-stage residual networks are more commonly used on ImageNet. Thus, instead of studying a fixed depth and width configuration in each stage, we aim to provide a flexible compound scaling rule of the relationship between robustness and total depths and widths. Define the width-depth (WD) ratio of a n-stage network as the average of comparing W_i to D_i in stage i:

$$\text{WD ratio} = \frac{1}{n-1}\sum_{i=1}^{n-1}\frac{W_i}{D_i} \tag{2}$$

The last stage is excluded since reducing its capacity improves robustness. WD ratio has three parameters: total stages n, depth D_i, and width W_i. To obtain valid models, we randomly sample from $n \in \{3,4,5,6\}$, $D_i \leq 60$, and $W_i \leq 1000$. Fig. 2a-left shows the AT results of all samples. All clean and PGD accuracies show negative correlations with the WD ratio. Note when the WD ratio is close to 0, the AT is unstable and also leads to inferior robustness. We intersect the WD ratio of top 10% networks from each attack budget and find the optimal range of WD ratio is $[7.5, 13.5]$. We use error distribution function (EDF) (Fig. 2a-right) to provide the characteristics of models from within and outside of the optimal range. A marked robustness gain is achieved by simply re-distributing depths and widths to satisfy the optimal range. Compare to ResNet-50's WD ratio of 32, a robust network has lower WD ratio, which echoes previous research that deep and narrow networks are better than shallow and wide networks [20]. Furthermore, our WD ratio is not limited to 3-stage networks as ResNet-50 shown here already has 4 stages. Sec. 5 shows generalization to other networks.

(a) Depth and Width Configurations

Config	Clean (%)		PGD^{10}-4 (%)	
ResNet-50 (baseline)	56.05		30.59	
Postponed downsampling	57.08	+1.03	33.08	+2.49
Patch 4, Stride 4	55.40	-0.65	31.68	+1.09
Patch 2, Stride 2	56.38	+0.33	31.91	+1.32
Patch 4, Stride 3	55.75	-0.30	32.41	+1.82
Patch 4, Stride 2	56.58	+0.53	32.83	+2.24
Patch 4, Stride 1	56.72	+0.67	33.20	+2.61
Stem width = 32	55.89	-0.16	29.73	-0.86
Stem width = 96	57.29	+1.24	32.06	+1.47
SE ($r = 4$)	57.83	+1.78	32.64	+2.05
GELU	57.48	+1.43	33.12	+2.53
SiLU	58.19	+2.14	34.07	+3.48
PReLU	55.81	-0.27	30.38	-0.21
PSiLU	56.38	+0.33	33.76	+3.17
PSSiLU	57.43	+1.38	32.22	+1.63

(b) Stem Stage and Residual Block Designs

Figure 2: **(a)** Clean and PGD accuracies are negatively correlated with the WD ratio. Each dot is a configuration of #stage, depth, and width. Intersecting each attack budget's top 10% most accurate configurations, we find the optimal range of the WD ratio is $[7.5, 13.5]$ (with color backgrounds) and verify the significance by the EDFs of models within range (steeper dark lines) and outside of range (gentle light lines). **(b)** Performance of different configurations in stem stage and residual blocks. Differences are compared with baseline ResNet-50. All models trained with Fast-AT [46] and evaluated on full ImageNet validation set. Different PGD attack budgets show a similar accuracy trend, and the full results are shown in supplementary material Sec. B.1.

4.2 *Convolutional* over *Patchify* stem stage

As a preprocessor, a common stem stage "aggressively" downsamples input images due to the redundancy inherent in natural images. The stem stage in ResNet-50 [15] consists of a stride-two 7×7 convolution and a stride-two max-pooling. Built on ResNet-50, RegNet [36] replaces the max-pooling with a stride-two residual shortcut in the first block of the first body stage, dubbed *postponed downsampling*. On the Transformer side, the stem stage patchifies the image into $p \times p$ non-overlapping patches. It is implemented by a stride-p $p \times p$ convolution, with $p = 14/16$ in vision Transformer (ViT) [11], and $p = 4$ in Swin Transformer [26]. Both *patchify stem* and ResNet-style *convolutional stem* are applicable to CNNs and Transformers. As a pure CNN, ConvNeXt [27] borrowed the patchify stem from Transformer, and Xiao *et al*. [49] successfully employed the convolutional stem in ViT.

Inspired by these findings, we compare how patchify and convolutional stem affect adversarial robustness. We set $p = 4$ following Swin Transformer and ConvNeXt for patchify stem (Patch 4, Stride 4) and use postponed downsampling for convolutional stem. From the results in Fig. 2b, we observe both designs show better robustness than the baseline ResNet-50, but postponed downsampling is significantly higher than patchify stem. There are two differences between these two designs: a smaller stride and the overlapping region between two convolution kernels in the convolutional stem. We verify whether these two distinctions are beneficial to robustness. First, we reduce the patch size from 4×4 to 2×2, and add the stride-two residual shortcut as in postponed downsampling to maintain the total downsampling ratio. This modification (Patch 2, Stride 2) improves PGD^{10}-4 and clean accuracies of 4×4 patch by 0.23 and 0.98 pp. Next, we gradually increase the overlapping area between the 4×4 patch by decreasing the stride from 3 to 1. We observe a consistent increment while decreasing the stride, and the 4×4 patch with the largest overlapping between neighboring patches (Patch 4, Stride 1) performs almost on par with the postponed downsampling. Finally, we additionally experiment on different output channel widths since it barely changes

total parameters. Decreasing the width from 64 (ResNet-50) to 32 lowers the accuracy due to fewer model parameters. However, increasing the width from 64 to 96 boosts PGD^{10}-4 and clean accuracies by 1.47 and 1.24 pp with a negligible 0.01M increase in total parameters. In summary, the convolution stem with postponed downsampling design outperforms patchify stem due to its less aggressive stride-two downsampling and overlapped convolution kernels. Besides, widening the output channels significantly improves robustness almost at no cost.

4.3 Robust Residual Block Design

4.3.1 Squeeze & Excitation

Squeeze and excitation (SE) block is a simple but effective add-on for the original residual block. It was proposed to adaptively recalibrate channel-wise feature responses by modeling interdependencies between channels [18]. RegNet [36] also verifies the effectiveness of SE on improving clean accuracy. However, Huang *et al*. [20] found that a straightforward application of SE hurts robustness on CIFAR-10, and added an extra skip connection around the SE. Driven by the discrepancy in clean and adversarial accuracies, we examine the SE block with ResNet-50. Note that the SE block will introduce additional parameters. Thus, we follow RegNet that sets $r = 4$ and appends the SE block directly after the 3×3 convolutional layer since it possesses fewer channels. Our experiments on ImageNet show a different picture, where the original SE block markedly increases all PGD and clean accuracies compared to ResNet-50 (Fig. 2b). We hypothesize that the different effects on CIFAR and ImageNet are caused by the reduction ratio r since Huang *et al*. only tested on $r = 16$. To provide a fair comparison, we train WRN with a sweep of hyperparameter $r = \{2, 4, 8, 16, 32, 64\}$. Our experiments show that adversarial robustness is negatively correlated with r, and when $r \geq 32$, the accuracy is inferior to the baseline WRN. More details are in supplementary material Sec. B.2. Therefore, SE block is a unified architecture component that improves both clean and adversarial accuracies, and we set $r = 4$ in all designs that include the SE block.

4.3.2 Non-parametric Smooth Activation Functions

Both the *number* and the *function* of activation layers are different between natural language processing (NLP) and vision architectures. In terms of the number, Transformer [11] only has one activation in the multilayer perceptron (MLP) block. In comparison, the activation is normally appended to all convolutions in a CNN block [15, 16]. ConvNeXt [27] first observed this phenomenon and reduced the total activations from three to one in all blocks, which leads to higher clean accuracy. Inspired by ConvNeXt, We combinatorially analyzed all possible locations of the activation in a single block. The average PGD^{10}-4 accuracies are 30.43%, 29.00%, and 24.84% when total activations are set to 3, 2, and 1, which are all inferior to ResNet-50. Thus, reducing activation layers is not beneficial to adversarial robustness, and we preserve the activation along with all convolutions.

In terms of the activation function, it is common practice to use ReLU in CNNs [15] due to its efficiency and simplicity. However, advanced Transformers, *e.g*., BERT [8], GPT-2 [35], and hierarchical Transformers [25] adopted smoother variants of ReLU, *e.g*., gaussian error linear unit (GELU) [17] and sigmoid linear unit (SiLU) [12]. Recent research shows replacing ReLU with its smooth approximations facilitates better gradient updates and leads to higher robustness [51]. For example, solely replacing all ReLUs with GELUs significantly improves the PGD accuracy of a standard ResNet-50 [2]. Dai *et al*. [5] proposed paramet-

ric activation functions by adding learnable parameters to non-parametric versions, which showed mixed performance on CIFAR-10 for different design spaces. Compared to SiLU, parametric SiLU (PSiLU) and parametric shifted SiLU (PSSiLU) show higher robustness when applied to WRN-28-10, but show degraded performance when applied to ResNet-18. Therefore, we examine and compare the robustness gain brought by GELU, SiLU, parametric ReLU (PReLU), PSiLU, and PSSiLU. We find that: (1) non-parametric smooth activations, *e.g.*, SiLU and GELU are significantly higher than ReLU (ResNet-50); and (2) for parametric activations, PReLU performs on par with ReLU, and both PSiLU and PSSiLU are inferior to SiLU. Synthesizing all the above findings, we recommend non-parametric smooth activations due to consistency and simplicity, thus replacing all ReLUs with SiLUs.

Table 1: A roadmap to robustify ResNet-50 by cumulatively applying our three design principles. We label the robustified version as Ra. Ra ResNet-50 significantly boosts the clean and adversarial accuracies of ResNet-50. Cumulative gains and robustified accuracies highlighted in bold. All models are trained with Fast-AT [46] and evaluated on full ImageNet validation set. Each configuration is trained thrice and evaluated thrice with different seeds (9 values in total). The standard deviations are shown in parentheses.

ResNet-50 → **Ra ResNet-50**	#Param.	Clean (%)	PGD^{10}-2 (%)	PGD^{10}-4 (%)	PGD^{10}-8 (%)
ResNet-50 (starting point)	25.7M	56.05 (0.10)	42.81 (0.22)	30.59 (0.15)	12.62 (0.12)
+ Depth & Width Config (Sec. 4.1)	25.8M	57.85 (0.16) +1.80	45.90 (0.26) +3.09	33.87 (0.10) +3.28	15.27 (0.05) +2.65
+ Convolutional Stem Stage (Sec. 4.2)	25.9M	58.00 (0.45) +0.15	46.59 (0.52) +0.69	34.90 (0.31) +1.03	15.85 (0.28) +0.58
+ Squeeze and Excitation (Sec. 4.3.1)	26.2M	60.22 (0.28) +2.22	48.95 (0.08) +2.36	36.43 (0.17) +1.53	16.43 (0.05) +0.58
+ Smoother Activation (Sec. 4.3.2)	26.2M	**62.02** (0.03) +1.80	**51.47** (0.22) +2.52	**39.65** (0.27) +3.22	**18.97** (0.14) +2.54
		Total: **+5.97**	**+8.66**	**+9.06**	**+6.35**

5 Adversarial Robustness Evaluation

We have completed the exploration of how individual architectural components affect adversarial robustness. It is encouraging to uncover how these components affect robustness, but it is not completely convincing unless the following two questions are addressed: 1) *Can these components consistently improve robustness when grouped together?* 2) *Are these design principles generalizable?* Sec. 5.1 provides a roadmap that robustifies a CNN with all three design principles, dubbed Robust architecture (Ra). For a model robustified with our design principles, we tag it with Ra. Then, we scale model architectures under diverse parameter budgets, extrapolate to the design spaces of ResNet [15] and WRN [53], train all models with various AT methods, and evaluate on CIFAR (Sec. 5.2) and ImageNet (Sec. 5.3) to verify the generalization of our design principles. Full comparisons with CNNs, Transformers, and NAS-based architectures are provided in supplementary materials Sec. C and D.

5.1 Combining the Three Design Principles

We robustify a CNN by applying our three architectural design principles. Specifically, we train ResNet-50 with Fast-AT [46] and use its evaluation results on ImageNet as the starting point. Table 1 outlines the path we take to robustify ResNet-50, and total parameters are generally controlled over the course of exploration. For the four body stages, we set

Table 2: Adversarial robustness on CIFAR-10 and CIFAR-100 against AA and 20-step PGD (PGD^{20}) with the same maximum perturbation $\ell_\infty, \varepsilon = 8/255$. Applying our principles leads to a consistent 1–3 pp robustness gain across AT methods, parameter budgets, and design spaces, boosting even the SOTA "Diff. 1M" and "Diff. 50M" AT methods proposed by Wang *et al.* [45]. Sec. C in supplementary materials provides a systematic comparison with Transformers and neural architecture search (NAS)-based architectures.

#Param.	Method	Model	CIFAR-10 Clean (%)	CIFAR-10 AA (%)	CIFAR-10 PGD^{20} (%)	CIFAR-100 Clean (%)	CIFAR-100 AA (%)	CIFAR-100 PGD^{20} (%)
26M	SAT	ResNet-50	84.05	49.97	54.37	55.86	23.78	27.48
		Ra ResNet-50	**84.91** +0.86	**50.94** +0.97	**55.19** +0.82	**56.38** +0.52	**24.99** +1.21	**28.84** +1.36
	TRADES	ResNet-50	82.26	49.91	54.50	56.00	25.05	29.91
		Ra ResNet-50	**82.80** +0.54	**51.23** +1.32	**55.44** +0.94	**56.29** +0.29	**25.83** +0.78	**31.87** +1.96
	MART	ResNet-50	77.98	47.17	52.70	53.18	25.35	30.79
		Ra ResNet-50	**79.60** +1.62	**49.19** +2.02	**56.47** +3.77	**53.68** +0.50	**26.97** +1.62	**32.81** +2.02
37M	SAT	WRN-28-10	85.44	48.45	53.13	**60.49**	23.64	27.47
		Ra WRN-28-10	**85.52** +0.08	**51.96** +3.51	**56.22** +3.09	59.09 -1.40	**25.14** +1.50	**29.27** +1.80
	TRADES	WRN-28-10	**83.86**	51.79	55.69	55.21	25.47	29.34
		Ra WRN-28-10	83.29 -0.57	**52.10** +0.31	**56.31** +0.62	**55.38** +0.71	**25.68** +0.21	**29.41** +0.07
	MART	WRN-28-10	82.83	50.30	57.00	51.31	25.78	30.06
		Ra WRN-28-10	**82.85** +0.02	**50.81** +0.51	**57.35** +0.35	**51.61** +0.30	**26.11** +0.33	**30.82** +0.76
	Diff. 1M	WRN-28-10	90.61	61.66	66.43	67.26	34.26	39.29
		Ra WRN-28-10	**91.32** +0.71	**65.11** +3.45	**68.93** +2.50	**69.03** +1.77	**37.24** +2.98	**41.59** +2.30
67M	SAT	WRN-34-12	85.92	49.35	53.05	59.08	23.69	27.05
		Ra WRN-34-12	**86.50** +0.58	**51.78** +2.43	**56.04** +2.99	**59.46** +0.38	**25.18** +1.49	**29.49** +2.44
	Diff. 1M	WRN-34-12	91.11	62.83	67.53	68.40	35.67	40.33
		Ra WRN-34-12	**91.75** +0.64	**65.71** +2.88	**69.67** +2.14	**69.75** +1.35	**37.73** +2.06	**42.16** +1.83
267M	SAT	WRN-70-16	86.26	50.19	53.74	60.26	23.99	27.05
		Ra WRN-70-16	**86.72** +0.46	**52.13** +1.94	**56.49** +2.75	**60.42** +0.16	**25.17** +1.18	**29.46** +2.41
	Diff. 1M	WRN-70-16	91.82	65.02	69.10	70.10	37.77	41.95
		Ra WRN-70-16	**92.16** +0.34	**66.33** +1.31	**70.37** +1.27	**70.25** +0.15	**38.73** +0.96	**42.61** +0.66
	Diff. 50M	WRN-70-16	93.25	70.69	73.89	-	-	-
		Ra WRN-70-16	**93.27** +0.02	**71.07** +0.38	**75.28** +1.39	-	-	-

$D_{1,2,3,4} = 5, 8, 13, 1$ and $W_{1,2,3,4} = 36, 72, 140, 270$ to control the WD ratio within the optimal range. The convolutional stem stage incorporates the postponed downsampling operation and widens the output channels to 96. Finally, we append the SE block ($r = 4$) to the 3×3 convolution layer and replace ReLU with SiLU. The clean and PGD accuracies of Ra ResNet-50 consistently improve for each modification, which verifies that our design principles work well both individually and collectively.

5.2 Evaluations on CIFAR-10 & CIFAR-100

We apply all three principles to models within the design spaces of ResNet and WRN. Detail specifications of each robustified architecture are in Sec. E in supplementary materials. Table 2 presents the comprehensive evalution results on CIFAR-10 and CIFAR-100 against AA and 20-step PGD (PGD^{20}) attacks with the same maximum perturbation $\ell_\infty, \varepsilon = 8/255$. The training methods are SAT [29], TRADES [52], MART [43], and diffusion-augmented AT [45]. The diffusion-augmented AT is the SOTA training recipe proposed by Wang *et al.* [45], which augments the original CIFAR only with images generated by elucidating diffusion model (EDM) [22], so that no external datasets are needed. Since this training recipe incurs extreme computational costs [45], we train on the 1M generated dataset using batch size 512 and epoch 400, abbreviated as "Diff. 1M" in Table 2. In general, the gains are consistent across AT methods, parameter budgets, and design spaces on CIFAR-10 and CIFAR-100. Importantly, our design principles augment the improvements in network robustness achieved through better training schemes, boosting even the robustness of the

SOTA Diff. 1M method by 1–3 pp. Our best model, Ra WRN-70-16, achieves 66.33% and 38.73% AA accuracy on CIFAR-10 and CIFAR-100, respectively. Due to limited computational resources, we only train "Diff. 50M" on CIFAR-10. Our Ra WRN-70-16 consistently outperforms WRN-70-16 and achieves the best performance on RobustBench [4].

5.3 Evaluations on ImageNet

Similarly, we also apply all three principles to models within the design spaces of ResNet and WRN. The training methods are Fast-AT [46] and SAT [29]. Table 3 presents the comprehensive SAT evaluation results on ImageNet against AA and 100-step PGD (PGD^{100}). The maximum perturbation for AA is $\ell_\infty, \varepsilon = 4/255$, and for PGD are $\ell_\infty, \varepsilon = \{2,4,8\}/255$. We observe a consistent 4–9 pp robustness gain across different model parameters and design spaces. Specifically, our Ra WRN-101-2, even with fewer parameters than WRN-101-2, improves the AA accuracy by 6.94 pp when trained from scratch with SAT.

Table 3: Adversarial robustness on ImageNet against AA and 100-step PGD (PGD^{100}). The maximum perturbation for AA is $\ell_\infty, \varepsilon = 4/255$, and for PGD are $\ell_\infty, \varepsilon = \{2,4,8\}/255$. All models trained from random initializations with SAT [29]. We observe a consistent 4–9 pp robustness gain. Full comparisons including Transformers and Fast-AT results are in supplementary materials Sec. D.

Model	#Param.	Clean (%)	AA (%)	PGD^{100}-2 (%)	PGD^{100}-4 (%)	PGD^{100}-8 (%)
ResNet-50	26M	63.87	34.96	52.15	38.96	15.83
Ra ResNet-50	26M	**70.17** +6.30	**44.14** +9.18	**60.06** +7.91	**47.77** +8.81	**21.77** +5.94
ResNet-101	45M	67.06	39.78	56.26	43.17	18.31
Ra ResNet-101	46M	**71.88** +4.82	**46.26** +6.48	**61.89** +5.63	**49.30** +6.13	**23.01** +4.70
WRN-101-2	127M	69.30	42.00	58.71	45.27	19.95
Ra WRN-101-2	104M	**73.44** +4.14	**48.94** +6.94	**63.49** +4.78	**51.03** +5.76	**25.31** +5.36

6 Conclusion

We synthesize a suite of three generalizable robust architectural design principles: (a) optimal range for *depth* and *width* configurations, (b) preferring *convolutional* over *patchify* stem stage, and (c) robust residual block design through adopting squeeze and excitation blocks and non-parametric smooth activation functions. Through extensive experiments across a wide spectrum of *dataset scales*, *adversarial training methods*, *model parameters*, and *network design spaces*, our principles consistently and markedly improve adversarial robustness on CIFAR-10, CIFAR-100, and ImageNet.

7 Acknowledgement

This work was supported in part by the Defense Advanced Research Projects Agency (DARPA). Use, duplication, or disclosure is subject to the restrictions as stated in Agreement number HR00112030001 between the Government and the Performer. This work was also supported in part by gifts from Avast, Fiddler Labs, Bosch, Facebook, Intel, NVIDIA, Google, Symantec, and Amazon.

References

[1] Yang Bai, Yuyuan Zeng, Yong Jiang, Shu-Tao Xia, Xingjun Ma, and Yisen Wang. Improving adversarial robustness via channel-wise activation suppressing. *arXiv preprint arXiv:2103.08307*, 2021.

[2] Yutong Bai, Jieru Mei, Alan L Yuille, and Cihang Xie. Are transformers more robust than cnns? *Advances in Neural Information Processing Systems*, 34:26831–26843, 2021.

[3] 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.

[4] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

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

[6] Edoardo Debenedetti, Vikash Sehwag, and Prateek Mittal. A light recipe to train robust vision transformers. *arXiv preprint arXiv:2209.07399*, 2022.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] 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.

[9] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. *arXiv preprint arXiv:1812.02637*, 2018.

[10] Yinpeng Dong, Zhijie Deng, Tianyu Pang, Jun Zhu, and Hang Su. Adversarial distributional training for robust deep learning. *Advances in Neural Information Processing Systems*, 33:8270–8283, 2020.

[11] 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.

[12] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[14] Minghao Guo, Yuzhe Yang, Rui Xu, Ziwei Liu, and Dahua Lin. When nas meets robustness: In search of robust architectures against adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 631–640, 2020.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[17] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

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

[19] 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*, 34:5545–5559, 2021.

[20] Shihua Huang, Zhichao Lu, Kalyanmoy Deb, and Vishnu Naresh Boddeti. Revisiting residual networks for adversarial robustness: An architectural perspective. *arXiv preprint arXiv:2212.11005*, 2022.

[21] Steffen Jung, Jovita Lukasik, and Margret Keuper. Neural architecture design and robustness: A dataset. *arXiv preprint arXiv:2306.06712*, 2023.

[22] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[24] Hayeon Lee, Eunyoung Hyung, and Sung Ju Hwang. Rapid neural architecture search by learning to generate graphs from datasets. *arXiv preprint arXiv:2107.00860*, 2021.

[25] Yun Liu, Yu-Huan Wu, Guolei Sun, Le Zhang, Ajad Chhatkuli, and Luc Van Gool. Vision transformers with hierarchical attention. *arXiv preprint arXiv:2106.03180*, 2021.

[26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[27] 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*, pages 11976–11986, 2022.

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

[29] 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.

[30] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

[31] Yichuan Mo, Dongxian Wu, Yifei Wang, Yiwen Guo, and Yisen Wang. When adversarial training meets vision transformers: Recipes from training to architecture. *arXiv preprint arXiv:2210.07540*, 2022.

[32] 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*, pages 12322–12332, 2021.

[33] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022.

[34] ShengYun Peng, Weilin Xu, Cory Cornelius, Kevin Li, Rahul Duggal, Duen Horng Chau, and Jason Martin. Robarch: Designing robust architectures against adversarial attacks. *arXiv preprint arXiv:2301.03110*, 2023.

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

[36] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020.

[37] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020.

[38] Naman D Singh, Francesco Croce, and Matthias Hein. Revisiting adversarial training for imagenet: Architectures, training and generalization across threat models. *arXiv preprint arXiv:2303.01870*, 2023.

[39] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017.

[40] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[41] Shiyu Tang, Ruihao Gong, Yan Wang, Aishan Liu, Jiakai Wang, Xinyun Chen, Fengwei Yu, Xianglong Liu, Dawn Song, Alan Yuille, et al. Robustart: Benchmarking robustness on architecture design and training techniques. *arXiv preprint arXiv:2109.05211*, 2021.

[42] Twan Van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

[43] 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.

[44] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. *arXiv preprint arXiv:2112.08304*, 2021.

[45] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. *arXiv preprint arXiv:2302.04638*, 2023.

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

[47] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. *arXiv preprint arXiv:2301.00808*, 2023.

[48] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

[49] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34:30392–30400, 2021.

[50] Cihang Xie and Alan Yuille. Intriguing properties of adversarial training at scale. In *International Conference on Learning Representations*, 2019.

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

[52] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

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

[54] 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*, pages 7472–7482. PMLR, 2019.

A Extended Description of Robust Architectures

A few research studies have examined the impact of architectural designs on adversarial robustness [14, 20, 38, 50], *e.g.*, RobustArt [41] is the first comprehensive robustness benchmark of architectures and training techniques on ImageNet variants and Jung *et al.* [21] presented the first robustness dataset evaluating a complete NAS search space and demonstrated architectures' impacts on robustness. Among them, Huang *et al.* [20] is the closest to ours, and we thus provide a more detailed comparison with their work. Similar to our work, Huang *et al.* [20] also explored the relationship among depth and width, the SE block, and adversarial robustness through adversarially trained networks. However, our work is markedly different and enhanced in the following aspects:

1. Huang *et al.* assigned a fixed depth and width ratio only for the 3-stage WRN on CIFAR-10. It was an open research question as to how to extrapolate this fixed ratio to networks with more than three stages, such as the commonly used 4-stage residual networks for ImageNet [15, 27, 36]. In contrast, we provide a flexible compound scaling rule that does not place a restriction on the total number of stages, and we verify its generalizability and optimality through extensive experiments on CIFAR-10, CIFAR-100, and ImageNet.
2. Huang *et al.* proposed a specific residual block design using hierarchically aggregated convolution and residual SE. However, we show that such a residual SE is unnecessary due to the negative correlation between reduction ratio r and robustness. Furthermore, our design principles are applicable to both basic and bottleneck residual block designs. This flexibility is advantageous since the basic block is commonly used on CIFAR and the bottleneck block is widely deployed on ImageNet to reduce computational complexity [15, 16].
3. Huang *et al.* found that the adversarial robustness of models with smooth activation functions was sensitive to AT hyperparameters, and that removing BN affine parameters from weight decay was crucial; if the BN affine parameters were not removed, smooth activation functions did not improve performance beyond that of ReLU. This finding contradicts the prevailing consensus in the literature that smooth activation functions significantly improve robustness [2, 51]. In our research, we also find that using smooth activation functions is beneficial to robustness, and removing the BN affine parameters from weight decay is the correct implementation supported by Van [42] and multiple popular code bases and forums.[1,2]
4. Finally, Huang *et al.* only explored on CIFAR-10 and CIFAR-100, leaving no evidence that these findings will extrapolate to the large-scale ImageNet. In contrast, we verified the generalization of all our design principles through extensive experiments over a wide spectrum of dataset scales, AT methods, model parameters, and network design spaces.

A parallel line of related studies leverages NAS to search for optimal robust architectures. Guo *et al.* [14] explored two types of block topologies within the DARTS search space. Subsequent work regulated the NAS loss formulation through the smoothness of the input loss landscape [32]. These compute-intensive NAS frameworks mainly focus on searching for block topology while leaving other factors to manual design, *e.g.*, activation, depth, and width. Furthermore, most searches are conducted on CIFAR-10 since both NAS and AT are already computationally expensive. Besides, current research shows that the NAS-optimized architecture depends on the dataset used [24], thus hindering explorations of network design

[1] minGPT: A PyTorch re-implementation of GPT
[2] PyTorch forum: Weight decay in the optimizers is a bad idea (especially with BatchNorm)

principles that deepen our understanding and generalize to new settings [36]. In Sec. C, we also compare our results with NAS-based networks and demonstrate that our robustified networks exhibit higher robustness.

B Additional Details on Architectural Design Principles

B.1 Full Results for Stem Stage and Residual Block Designs

Here we provide the full results in Table 4 showcasing how various configurations of *stem stage* and *residual block designs* impact the clean and adversarial accuracies over the ResNet-50 baseline, extending the result highlights presented in Figure 2b of the main paper.

For the stem stage, the convolution stem with postponed downsampling operation outperforms the patchify stem. In the patchify stem, we observe a consistent performance improvement while decreasing the stride, and the 4×4 patch with the largest overlapping between neighboring patches (Patch 4, Stride 1) performs almost on par with the postponed downsampling. Finally, decreasing the width from 64 (ResNet-50) to 32 lowers the accuracy due to fewer model parameters, while increasing the width from 64 to 96 significantly boosts both clean and adversarial accuracies with a negligible 0.01M increase in total parameters.

For the residual block design, we find that a straightforward application of SE markedly increases all PGD and clean accuracies compared to ResNet-50. In terms of activation, reducing the number of activation layers does not contribute to adversarial robustness. Therefore, we preserve the activations along with all convolutions. Besides, we find that non-parametric smooth activation functions exhibit greater robustness compared to both ReLU and their parametric counterparts.

Table 4: Full results showcasing how various configurations of *stem stage* and *residual block designs* impact the clean and adversarial accuracies over the ResNet-50 baseline, extending the result highlights already presented in Figure 2b of the main paper. All models trained with Fast-AT [46] and evaluated on full ImageNet validation set.

	Config	Clean (%)	PGD^{10}-2 (%)	PGD^{10}-4 (%)	PGD^{10}-8 (%)
	ResNet-50 (baseline)	56.05	42.81	30.59	12.62
Stem stage	Postponed downsampling	57.08 +1.03	45.19 +2.38	33.08 +2.49	14.50 +1.88
	Patch 4, Stride 4	55.40 -0.65	43.45 +0.64	31.68 +1.09	13.80 +1.18
	Patch 2, Stride 2	56.38 +0.33	44.21 +1.40	31.91 +1.32	13.48 +0.86
	Patch 4, Stride 3	55.75 -0.30	44.54 +1.73	32.41 +1.82	13.74 +1.12
	Patch 4, Stride 2	56.58 +0.53	44.60 +1.79	32.83 +2.24	14.03 +1.41
	Patch 4, Stride 1	56.72 +0.67	45.06 +2.25	33.20 +2.61	14.45 +1.83
	Stem width = 32	55.89 -0.16	41.64 -1.17	29.73 -0.86	13.25 +0.63
	Stem width = 96	57.29 +1.24	44.55 +1.47	32.06 +1.47	13.74 +1.12
Residual block design	SE ($r = 4$)	57.83 +1.78	45.09 +2.28	32.64 +2.05	14.01 +1.39
	ReLU-ReLU-0	51.54 -4.51	38.69 -4.12	27.05 -3.54	10.94 -1.68
	ReLU-0-ReLU	53.91 -2.14	41.22 -1.59	29.62 -0.97	12.30 -0.32
	0-ReLU-ReLU	54.81 -1.24	42.10 -0.71	30.34 -0.25	12.86 +0.24
	0-0-ReLU	51.03 -5.02	39.12 -3.69	28.15 -2.44	12.09 -0.53
	0-ReLU-0	47.18 -8.87	34.85 -7.96	24.12 -6.47	9.51 -3.11
	ReLU-0-0	44.21 -11.84	32.34 -10.47	22.24 -8.35	8.77 -3.85
	GELU	57.48 +1.43	45.05 +2.24	33.12 +2.53	14.80 +2.18
	SiLU	58.19 +2.14	46.21 +3.40	34.07 +3.48	14.68 +2.06
	PReLU	55.81 -0.27	42.52 -0.29	30.38 -0.21	12.76 +0.14
	PSiLU	56.38 +0.33	44.90 +2.09	33.76 +3.17	15.40 +2.78
	PSSiLU	57.43 +1.38	44.44 +1.63	32.22 +1.63	13.71 +1.09

B.2 Adversarial Robustness Negatively Correlated with SE Reduction Ratio r on CIFAR-10

Table 5: A hyperparameter sweep of the SE block reduction ratio $r = \{2,4,8,16,32,64\}$ on CIFAR-10. These results show that adversarial robustness is negatively correlated with r, and when $r \geq 32$, the accuracy is inferior to the baseline WRN-22-10, supporting our discovery presented in Sec. 4.3.1. Both AA and 20-step PGD (PGD^{20}) use the same maximum perturbation $\ell_\infty, \varepsilon = 8/255$.

Config	Clean (%)	AA (%)	PGD^{20} (%)
WRN-22-10	83.82	48.38	52.64
WRN-22-10 ($r = 2$)	86.95 +3.13	49.14 +0.76	53.48 +0.84
WRN-22-10 ($r = 4$)	86.75 +2.93	49.13 +0.75	53.52 +0.88
WRN-22-10 ($r = 8$)	86.48 +2.66	49.11 +0.73	53.39 +0.75
WRN-22-10 ($r = 16$)	84.97 +1.15	48.89 +0.51	53.04 +0.40
WRN-22-10 ($r = 32$)	83.61 -0.21	48.20 -0.18	52.24 -0.40
WRN-22-10 ($r = 64$)	82.90 -0.92	47.44 -0.94	51.43 -1.21

C Evaluations on CIFAR-10 & CIFAR-100

This section presents the complete results of our robustified architectures in Table 6 and provides a systematic comparison to SOTA adversarially trained Transformers and NAS-based networks in Table 7, extending the result highlights presented in Sec. 5.2 in the main paper. As the AT recipes for CNNs and Transformers are not compatible with each other, we do not retrain the Transformers and instead, directly extract the results from the literature. In addition, the AT recipe for Transformers requires multiple training tricks built on SAT to boost robustness, *e.g.*, attention random dropping [31], perturbation random masking [31], ε-warmup [6], and larger weight decay [6]. Despite employing all these tricks in training Transformers, our Ra WRN-34-12 trained with "Diff. 1M" is significantly more robust than Swin-B and cross-covariance image Transformers (XCiT)-L12 on both CIFAR-10 and CIFAR-100 even with fewer total parameters. We also compare to NAS-based networks: our Ra WRN-22-10 is significantly more robust than RobNet-large-v2 [14] and performs on par with AdvRush [32] using the same TRADES [54] AT method but with fewer total parameters. Lastly, we compare our robust architectures with Huang *et al.* [19], who have also studied the relationship between robustness and depth and width, and proposed a reconfigured version of WRN-34-12 called WRN-34-R. By using the same SAT methods, both Ra WRN-34-12 and WRN-34-R show greater robustness than the baseline WRN-34-12, but our Ra WRN-34-12 is 1.75 pp and 0.69 pp higher than WRN-34-R in terms of AA and PGD accuracies, respectively.

Table 6: Complete results of adversarial robustness on CIFAR-10 and CIFAR-100 against AA and 20-step PGD (PGD^{20}) with the same maximum perturbation $\ell_\infty, \varepsilon = 8/255$. Applying our principles leads to a consistent 1–3 pp robustness gain across AT methods, parameter budgets, and design spaces, boosting even the SOTA "Diff. 1M" and "Diff. 50M" AT methods proposed by Wang *et al.* [45]. This table extends Table 2 in the main paper by including the results for WRN-22-10.

#Param.	Method	Model	CIFAR-10 Clean (%)	CIFAR-10 AA (%)	CIFAR-10 PGD^{20} (%)	CIFAR-100 Clean (%)	CIFAR-100 AA (%)	CIFAR-100 PGD^{20} (%)
26M	SAT	ResNet-50	84.05	49.97	54.37	55.86	23.78	27.48
		Ra ResNet-50	**84.91** +0.86	**50.94** +0.97	**55.19** +0.82	**56.38** +0.52	**24.99** +1.21	**28.84** +1.36
	TRADES	ResNet-50	82.26	49.91	54.50	56.00	25.05	29.91
		Ra ResNet-50	**82.80** +0.54	**51.23** +1.32	**55.44** +0.94	**56.29** +0.29	**25.83** +0.78	**31.87** +1.96
	MART	ResNet-50	77.98	47.17	52.70	53.18	25.35	30.79
		Ra ResNet-50	**79.60** +1.62	**49.19** +2.02	**56.47** +3.77	**53.68** +0.50	**26.97** +1.62	**32.81** +2.02
27M	SAT	WRN-22-10	83.82	48.38	52.64	56.79	23.46	27.08
		Ra WRN-22-10	**84.27** +0.45	**51.30** +2.92	**55.42** +2.78	**57.34** +0.55	**24.27** +0.81	**28.64** +1.56
	TRADES	WRN-22-10	81.81	51.06	55.21	55.48	23.50	29.60
		Ra WRN-22-10	**82.27** +0.46	**51.71** +0.65	**56.20** +0.99	**55.55** +0.07	**24.91** +1.41	**29.78** +0.18
37M	SAT	WRN-28-10	85.44	48.45	53.13	**60.49**	23.64	27.47
		Ra WRN-28-10	**85.52** +0.08	**51.96** +3.51	**56.22** +3.09	59.09 -1.40	**25.14** +1.50	**29.27** +1.80
	TRADES	WRN-28-10	**83.86**	51.79	55.69	55.21	25.47	29.34
		Ra WRN-28-10	83.29 -0.57	**52.10** +0.31	**56.31** +0.62	**55.38** +0.71	**25.68** +0.21	**29.41** +0.07
	MART	WRN-28-10	82.83	50.30	57.00	51.31	25.78	30.06
		Ra WRN-28-10	**82.85** +0.02	**50.81** +0.51	**57.35** +0.35	**51.61** +0.30	**26.11** +0.33	**30.82** +0.76
	Diff. 1M	WRN-28-10	90.61	61.66	66.43	67.26	34.26	39.29
		Ra WRN-28-10	**91.32** +0.71	**65.11** +3.45	**68.93** +2.50	**69.03** +1.77	**37.24** +2.98	**41.59** +2.30
67M	SAT	WRN-34-12	85.92	49.35	53.05	59.08	23.69	27.05
		Ra WRN-34-12	**86.50** +0.58	**51.78** +2.43	**56.04** +2.99	**59.46** +0.38	**25.18** +1.49	**29.49** +2.44
	Diff. 1M	WRN-34-12	91.11	62.83	67.53	68.40	35.67	40.33
		Ra WRN-34-12	**91.75** +0.64	**65.71** +2.88	**69.67** +2.14	**69.75** +1.35	**37.73** +2.06	**42.16** +1.83
267M	SAT	WRN-70-16	86.26	50.19	53.74	60.26	23.99	27.05
		Ra WRN-70-16	**86.72** +0.46	**52.13** +1.94	**56.49** +2.75	**60.42** +0.16	**25.17** +1.18	**29.46** +2.41
	Diff. 1M	WRN-70-16	91.82	65.02	69.10	70.10	37.77	41.95
		Ra WRN-70-16	**92.16** +0.34	**66.33** +1.31	**70.37** +1.27	**70.25** +0.15	**38.73** +0.96	**42.61** +0.66
	Diff. 50M	WRN-70-16	93.25	70.69	73.89	-	-	-
		Ra WRN-70-16	**93.27** +0.02	**71.07** +0.38	**75.28** +1.39	-	-	-

Table 7: A systematic comparison to SOTA adversarially trained Transformers and NAS-based architectures with adversarial robustness on CIFAR-10 and CIFAR-100 against AA and 20-step PGD (PGD^{20}) with the same maximum perturbation $\ell_\infty, \varepsilon = 8/255$. Our robustified architectures (prefixed by Ra) exhibit greater robustness (highlighted in **bold**) than all Transformers and NAS-based architectures compared. The "Diff. 1M" results are extracted from Table 6.

Method	Model	#Param.	CIFAR-10 Clean (%)	CIFAR-10 AA (%)	CIFAR-10 PGD^{20} (%)	CIFAR-100 Clean (%)	CIFAR-100 AA (%)	CIFAR-100 PGD^{20} (%)
Diff. 1M	Ra WRN-28-10	37M	91.32	**65.11**	**68.93**	69.03	**37.24**	**41.59**
	Ra WRN-34-12	67M	91.75	**65.71**	**69.67**	69.75	**37.73**	**42.16**
	Ra WRN-70-16	267M	92.16	**66.33**	**70.37**	70.25	**38.73**	**42.61**
Mo *et al.* [31]	DeiT-S	22M	83.04	48.34	52.52	-	-	-
	Swin-S	50M	84.46	46.17	50.02	-	-	-
	ViT-B/16	86M	84.90	50.03	53.80	-	-	-
	Swin-B	88M	84.16	47.50	51.47	-	-	-
Debenedetti *et al.* [8]	XCiT-S12	26M	90.06	56.14	-	67.34	32.19	-
	XCiT-M12	46M	91.30	57.27	-	69.21	34.21	-
	XCiT-L12	104M	91.73	57.58	-	70.76	35.08	-
TRADES	RobNet-large-v2 [27]	33M	84.57	47.48	52.79	55.27	23.69	29.23
TRADES	AdvRush (7@96) [33]	33M	84.65	52.08	56.23	55.40	25.27	29.40
SAT	WRN-34-R [19]	68M	87.85	50.03	55.35	61.33	25.20	29.02

D Evaluations on ImageNet

This section presents an extended discussion of the ImageNet results in Sec. 5.3 in the main paper. Table 8 provides a controlled comparison of our robustified architectures to SOTA CNNs and Transformers using Fast-AT method [46]. We present the robustified ResNet-50, ResNet-101, and WRN-101-2 and make the following observations:

1. Our robustified architectures consistently demonstrate a 4–9 pp gain in robustness across different model parameters and design spaces. Furthermore, increasing the total number of parameters in general leads to higher robustness.
2. Under a fixed model capacity, our Ra ResNet-50 outperforms the baseline ResNet-50 and ResNeXt-50 32×4d [52], and Ra ResNet-101 outperforms ResNet-101 and RegNetX-8GF [36].
3. Compared to models with larger parameters, our Ra ResNet-50 is more robust than ResNet-152 and WRN-50-2, and even WRN-101-2 despite having 4.85× fewer parameters. Similarly, our Ra WRN-101-2 outperforms the baseline WRN-101-2 and achieves SOTA performance under the Fast-AT method.
4. Transformers such as Swin-T [26] and Transformer-based architectures such as ConvNeXt-T [27] exhibit lower robustness when employing Fast-AT. The phenomenon can be attributed to the differences in optimizers, learning rates, and data augmentation, where most Transformer-related architectures use AdamW [28], tiny learning rates, and heavy data augmentation.

Then, we provide a systematic comparison of our SAT-trained robust architectures with CNNs and Transformers that utilize specifically optimized AT methods in Table 9. By applying our design principles, the robustified architecture achieves a similar or even superior level of robustness compared to the Transformers that utilize additional training tricks to enhance their robustness. For example, under similar total parameters, ResNet-50 and ResNet-101 are less robust than XCiT-S12 and XCiT-M12, respectively, but the robustified Ra ResNet-50 and Ra ResNet-101 show higher clean and AA accuracies. Additionally, there is no sign of saturation when scaling up the total parameters, as Ra WRN-101-2 remains markedly more robust than XCiT-L12 with the same 104M parameters.

Table 8: Our robustified architectures (Ra) consistently demonstrate a 4–9 pp gain in robustness over SOTA CNNs and Transformers using Fast-AT method [46], across different model parameters, design spaces, and attack budgets.

Model	#Param.	Clean (%)	PGD^{10}-2 (%)	PGD^{10}-4 (%)	PGD^{10}-8 (%)
Ra ResNet-50	26M	62.02	51.47	39.65	18.97
Ra ResNet-101	46M	64.40	53.97	42.06	20.98
Ra WRN-101-2	104M	66.08	55.52	43.81	22.50
SqueezeNet 1.1	1M	0.10	0.10	0.10	0.10
MobileNet V2	4M	41.60	31.23	21.89	8.94
EfficientNet-B0	5M	48.78	37.74	26.90	10.92
ShuffleNet V2 2.0×	7M	49.99	0.01	0.01	0.02
DenseNet-121	8M	52.29	40.06	28.72	12.23
ResNet-18	12M	46.59	35.05	24.64	9.95
RegNetX-3.2GF	15M	57.26	45.74	33.85	15.37
RegNetY-3.2GF	19M	59.15	47.09	34.82	15.51
EfficientNetV2-S	21M	57.64	45.89	33.48	14.03
ResNeXt-50 32×4d	25M	57.33	45.46	33.08	14.45
ResNet-50	26M	56.09	42.66	30.43	12.61
Swin-T	28M	38.83	28.08	18.49	6.20
ConvNeXt-T	29M	21.35	15.39	10.51	4.07
DenseNet-161	29M	59.80	47.60	35.35	15.77
EfficientNet-B5	30M	55.90	44.80	33.26	14.53
RegNetY-8GF	39M	63.61	52.26	40.15	19.21
RegNetX-8GF	40M	60.26	48.98	36.89	17.22
ResNet-101	45M	58.04	45.72	33.90	15.93
ResNet-152	60M	61.55	48.50	35.85	15.87
WRN-50-2	69M	60.66	46.99	34.10	15.37
WRN-101-2	127M	61.63	49.10	36.23	16.14

Table 9: By applying our design principles, the robustified architecture achieves a similar or even superior level of robustness compared to the Transformers that utilize additional training tricks to enhance their robustness. The Ra results are extracted from Table 3 in the main paper.

Model	#Param.	Clean (%)	AA (%)	PGD^{100}-2 (%)	PGD^{100}-4 (%)	PGD^{100}-8 (%)
Ra ResNet-50	26M	70.17	44.14	60.06	47.77	21.77
Ra ResNet-101	46M	71.88	46.26	61.89	49.30	23.01
Ra WRN-101-2	104M	73.44	48.94	63.49	51.03	25.31
PoolFormer-M12 [70]	22M	66.16	34.72	-	-	-
DeiT-S [70]	22M	66.50	35.50	-	40.32	-
ResNet50 + GELU [70]	26M	67.38	35.51	40.27	-	-
ResNet50 + DiffPure [53]	26M	67.79	40.39	-	-	-
XCiT-S12 [70]	26M	72.34	41.78	-	-	-
ConvNeXt-T [53]	29M	67.60	41.60	-	-	-
XCiT-M12 [70]	46M	74.04	45.24	-	-	-
WRN-50-2 [53]	69M	68.41	38.14	55.86	41.24	16.29
WRN-50-2 + DiffPure [53]	69M	71.16	44.39	-	-	-
Vit-B/16 [51]	86M	69.10	34.62	-	37.52	-
Swin-B [51]	88M	74.36	38.61	-	40.87	-
XCiT-L12 [70]	104M	73.76	47.60	-	-	-

E Architecture Details

Table 10 contains the details of all the robustified architectures mentioned in the paper. For depth and width, we present the list of total depths and widths in each stage and compute the corresponding WD ratio. For the stem stage, we use the convolution stem stage with postponed downsampling operation and increase the output channels to 96. Regarding the design of the residual block, we append the SE block ($r = 4$) to the 3×3 convolution layer and replace ReLU with SiLU.

Table 10: Details of all the robustified architectures mentioned in the paper.

Model	#Param.	Depth	Width	WD ratio	Stem width	SE	Activation
Ra ResNet-50	26M	[5, 8, 13, 1]	[36, 72, 140, 270]	8.99	96	$r = 4$	SiLU
Ra WRN-22-10	27M	[13, 15, 2]	[120, 240, 480]	12.62	96	$r = 4$	SiLU
Ra WRN-28-10	37M	[14, 16, 3]	[128, 256, 512]	12.57	96	$r = 4$	SiLU
Ra ResNet-101	46M	[7, 11, 18, 1]	[42, 84, 166, 328]	7.62	96	$r = 4$	SiLU
Ra WRN-34-12	67M	[18, 20, 5]	[144, 288, 576]	11.20	96	$r = 4$	SiLU
Ra WRN-101-2	104M	[7, 11, 18, 1]	[64, 128, 252, 504]	11.59	96	$r = 4$	SiLU
Ra WRN-70-16	267M	[30, 31, 10]	[216, 432, 864]	10.57	96	$r = 4$	SiLU

F Training curves and convergence rate

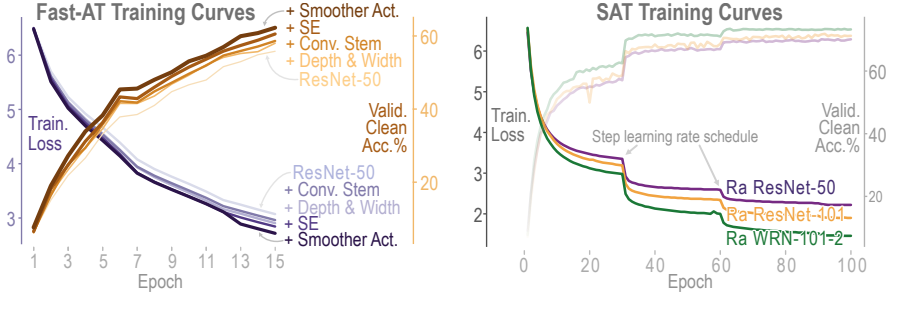

Figure 3: Visualization of the Fast-AT curves of individual architectural modifications (Table 1), and the SAT curves of the final robustified model (Table 3). We observed that a lower training loss leads to a higher robustness as expected and no catastrophic overfitting occurs during training.