Towards Accurate and Robust Architectures via Neural Architecture Search

Yuwei Ou*, Yuqi Feng*, Yanan Sun†
College of Computer Science, Sichuan University.
ouyuwei@stu.scu.edu.cn; feng770623@gmail.com; ysun@scu.edu.cn

Abstract

To defend deep neural networks from adversarial attacks, adversarial training has been drawing increasing attention for its effectiveness. However, the accuracy and robustness resulting from the adversarial training are limited by the architecture, because adversarial training improves accuracy and robustness by adjusting the weight connection affiliated to the architecture. In this work, we propose ARNAS to search for accurate and robust architectures for adversarial training. First we design an accurate and robust search space, in which the placement of the cells and the proportional relationship of the filter numbers are carefully determined. With the design, the architectures can obtain both accuracy and robustness by deploying accurate and robust structures to their sensitive positions, respectively. Then we propose a differentiable multi-objective search strategy, performing gradient descent towards directions that are beneficial for both natural loss and adversarial loss, thus the accuracy and robustness can be guaranteed at the same time. We conduct comprehensive experiments in terms of white-box attacks, black-box attacks, and transferability. Experimental results show that the searched architecture has the strongest robustness with the competitive accuracy, and breaks the traditional idea that NAS-based architectures cannot transfer well to complex tasks in robustness scenarios. By analyzing outstanding architectures searched, we also conclude that accurate and robust neural architectures tend to deploy different structures near the input and output, which has great practical significance on both hand-crafting and automatically designing of accurate and robust architectures.

1. Introduction

Deep neural networks (DNNs) have shown remarkable performance in various real-world applications [2, 21, 24]. However, DNNs are found to be vulnerable to adversarial attacks [38]. The phenomenon limits the application of DNNs in security-critical systems such as self-driving cars [14] and face recognition [37]. After the discovery of this intriguing weakness of DNNs, various methods have been proposed to defend DNNs from adversarial attacks. Among them, adversarial training [16, 28] has been widely used for its excellent ability to enhance robustness [29, 46]. Different from the standard training that trains with natural data directly, the adversarial training trains the network with adversarial examples. However, the problem is that the accuracy and robustness resulting from the adversarial training are limited if the architecture is not well designed in advance [18], because the adversarial training improves the accuracy and robustness by adjusting the weight connection that is affiliated to the architecture.

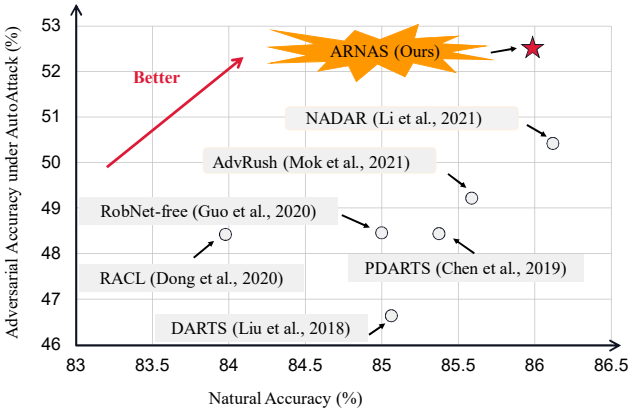

Figure 1. Natural and adversarial accuracy on CIFAR-10. All the architectures are adversarially trained using 7-step PGD, and the adversarial accuracy is evaluated under AutoAttack.

Recently, an advanced research topic, known as robust NAS [12, 18, 29], has been widely investigated to solve this problem. Specifically, robust NAS designs intrinsically accurate and robust neural architectures utilizing neural architecture search (NAS) [48]. These architectures have been validated to show strong accuracy and robustness after adversarial training. For example, RobNet [18] adopts the one-shot NAS [3] method, adversarially trains a supernetwork for once and then finds out the robust sub-networks

*Equal contribution.
†Corresponding author.

by evaluating each of them under adversarial attacks using the shared weights. ABanditNAS [5] introduces an anti-bandit algorithm, and searches for robust architectures by gradually abandoning operations that are not likely to make the architectures robust. NADAR [26] proposes an architecture dilation method, begins with the backbone network of a satisfactory accuracy over the natural data, and searches for a dilation architecture to pursue a maximal robustness gain while preserving a minimal accuracy drop. Besides, the most common way is to employ the differentiable search method, and search for robust architectures by updating the architectures utilizing some robustness metrics. These robustness metrics include Lipschitz constant [7] adopted by RACL [12], input loss landscape [47] employed by AdvRush [29], certified lower bound and Jacobian norm bound proposed and used by DSRNA [20].

After a comprehensive investigation of existing robust NAS methods, we find that there are still two shortcomings in the existing works. First, most of the existing robust NAS just simply adopt the search space of conventional NAS algorithms designed for standard training [49]. However, it is explored that the architectures suitable for the adversarial training may have different overall structures from those suitable for the standard training [22]. Consequently, the search space adopted by existing robust NAS is no longer suitable for the adversarial training. Second, there is a trade-off between accuracy and robustness [40, 46]. Most of the existing robust NAS solve this multi-objective optimization problem by transforming it to a single-objective optimization problem with a fixed regularization coefficient, and optimizing the problem using gradient descent. However, the optimization results heavily rely on the selection of the coefficients. Meanwhile, scholars studying multi-objective optimization have shown that always finding a descent direction common to all criteria may be better for identifying the Pareto front [9, 10], while the fixed regularization coefficient cannot realize this.

In this paper, we consider the above two problems comprehensively, and propose the Accurate and Robust Neural Architecture Search (ARNAS) method. As shown in Fig. 1, compared with peer competitors, the proposed method significantly improves the robustness while achieving similar accuracy to the best after an identical process of adversarial training. Our contributions can be summarized as follows:

- We design a novel accurate and robust search space to solve the problem that the conventional search space does not contain accurate and robust architectures for adversarial training. Specifically, motivated by preliminary study [22] that depth and width in different positions of architectures have different effects on accuracy and robustness, we further conjecture that the architectures themselves in different positions also play different roles. We conduct experiments to support the conjecture, and accordingly design a novel cell-based search space. the designed search space is composed of Accurate Cell, Robust Cell, and Reduction Cell. We determine the placement of the cells and the proportional relationship of the filter numbers through experiments.
- We propose a differentiable multi-objective search strategy to address the problem that previous search strategies cannot effectively achieve the dual benefits of accuracy and robustness. Specifically, based on multiple gradient descent method (MGDA), we further design a multi-objective adversarial training method, which first finds a common descent direction of natural loss and adversarial loss by determining their coefficients dynamically and automatically, and then performs gradient descent to optimize the architectures towards smaller natural loss and adversarial loss.
- We conduct comprehensive experiments in terms of white-box attacks, black-box attacks and transferability. The experiments of white-box and black-box attacks show the strongest robustness and high accuracy of the searched architecture. The experiments of transferability break the prejudice that NAS-based architectures cannot transfer well as the task complexity increases. By analyzing outstanding architectures searched, we also conclude that the architectures can obtain both accuracy and robustness by deploying very different structures in different positions, which has great guiding significance on both hand-crafting and automatically designing of accurate and robust architectures.

2. Related Works

2.1. Adversarial Attacks and Defenses

According to whether the attacker has full access to the target model or not, existing adversarial attack methods can be divided into white-box attacks and black-box attacks. In relevant fields, commonly used white-box attacks include FGSM [16], C&W [4], and PGD [28]. Commonly used black-box attack is the transfer-based attack [33]. Recently, AutoAttack [8], an ensemble of attacks containing both white-box and black-box attacks, becomes popular for robustness evaluation because it is parameter-free, computationally affordable, and user-independent. We also include AutoAttack in our experiments for reliable comparison.

To defend neural networks from adversarial attacks, numerous defense methods have been proposed, among which the adversarial training [16, 28] has been the most popular way so far to help neural networks enhance their robustness [5, 12, 18, 29]. To perform the adversarial training, the most widely used method is to replace input data with adversarial examples generated by PGD, for the reason that neural networks adversarially trained using PGD usually generalize to other attacks [28]. Besides, some other

methods such as defensive distillation [32], data compression [13, 17], feature denoising [42], and model ensemble [31, 39] also demonstrate their feasibility. In our work, we notice the fact that the adversarial training, as the most effective defense method, depends on the design of neural architectures. If the neural architectures are not suitable, they can only obtain low accuracy and low robustness. Our work tackles the problem by automatically designing neural architectures that perform well after adversarial training.

2.2. Neural Architecture Search (NAS)

NAS is a promising technique which aims to automate the architecture design of DNNs. According to the search strategy adopted, NAS can be divided into three categories: evolutionary computation-based [34, 35] NAS, reinforcement learning-based [48] NAS, and the differentiable NAS [6, 27, 43]. Among them, the differentiable NAS is especially popular for designing robust architectures because of its efficiency and effectiveness. In the differentiable NAS methods, the search space is relaxed to be continuous, so that the architectures can be designed by optimizing differentiable metrics using gradient descent. By introducing differentiable metrics of robustness, robust architectures can be searched. Our work in this paper also utilizes the differentiable NAS for its efficiency and effectiveness.

3. The Proposed ARNAS Method

3.1. ARNAS Overview

Algorithm 1: The Framework of ARNAS Method

Input: $E \leftarrow$ Total number of epochs for search
Output: $f_A^* \leftarrow$ Final architecture

Construct the **accurate and robust search space**
$f_{super}^0 \leftarrow$Initialize a supernet based on DARTS according to the constructed search space
for $i \leftarrow 1$ **to** E **do**
  $f_{super}^i \leftarrow$ Optimize f_{super}^{i-1} using **the differentiable multi-objective search strategy**
end
$f_A^* \leftarrow$ Apply the discretization rule of DARTS to f_{super}^E

The framework of the proposed ARNAS method is presented in Algorithm 1. First, we construct the accurate and robust search space, and then initialize a supernet based on the constructed search space. After that, we iteratively optimize the supernet using the proposed differentiable multi-objective search strategy. Finally, the best architecture can be obtained using the discretization rule of DARTS [27]. The innovations of the proposed ARNAS method lies in the accurate and robust search space and the differentiable multi-objective search strategy, which will be introduced in detail in the following subsections.

3.2. Accurate and Robust Search Space

In this subsection, we will introduce 1) the characteristics of accurate and robust architectures, 2) the limitation of the conventional search space, and 3) the construction of the accurate and robust search space in turn.

3.2.1 The Characteristics of Accurate and Robust Architectures

In order to design an accurate and robust search space for adversarial training, we should first consider the characteristics of accurate and robust architectures. However, the research of NAS and robust neural networks are emerging topics, there is rare knowledge can be directly explored. We are aware of a recent work [22] which explores the architectural ingredients of adversarially robust deep neural networks. It experimentally concludes that the depth and the width of the neural architectures in different positions have dissimilar effects on the accuracy and the robustness. Motivated by this, we further make the conjecture shown in Proposition 1. The proposition would also be experimentally verified in Sec. 4.4.5.

Proposition 1 *The cells in different positions of the overall architecture may have different effects on the accuracy and the robustness, and the accuracy and the robustness of the neural architectures can be improved simultaneously by placing different cells in different position.*

3.2.2 The Limitation of Conventional Search Space

Based on Proposition 1, we further analyze the limitation of the conventional search space. In particular, the conventional cell-based search space refers to the one popularized by the famous DARTS algorithm. Currently, almost all differentiable robust NAS methods adopt this search space [12, 20, 29]. Specifically, the cell-based search space only designs two kinds of computation cells named Normal Cells and Reduction Cells, which play the role of enhancing accuracy and reducing data dimension for improving efficiency, respectively. Based on the design, the overall architecture is constructed by stacking multiple Normal Cells to increase the accuracy as much as possible, and rare Reduction Cells to avoid the invalid data dimension. The resulting architecture is mainly composed of the same Normal Cells, and the architectures matching Proposition 1 (such as the architectures that employ the separable convolutions near the input but employ the dilated convolutions near the output) are not contained in the search space, which limits the improvement of accuracy and robustness.

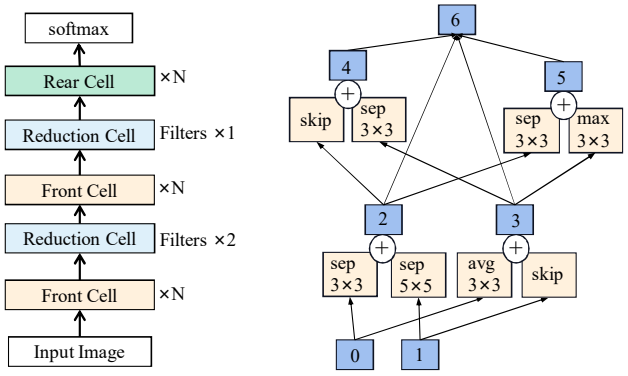

Figure 2. An example of the proposed search space for CIFAR-10. LEFT: the full outer structure. RIGHT: cell example.

3.2.3 The Construction of the Accurate and Robust Search Space

Given the above analysis, we retain the Reduction Cell while replacing the single type of Normal Cell with Accurate Cell and Robust Cell, aiming to include the architectures matching Proposition 1. Consequently, there are three types of cells designed in the proposed search space: Accurate Cell, Robust Cell, and Reduction Cell. In particular, Accurate Cells and Robust Cells both return a feature map of the same dimension but can be placed in different positions to take the corresponding effects of accurate and robustness, as concluded in Proposition 1. Reduction Cells return a feature map where the feature map height and width is reduced by a factor of two, playing the same role as common. With this design, the learned cells could also be stacked to form a full network, resulting the scalability of the designed search space. In the following, we will introduce the details about how to determine their particular position to achieve both accuracy and robustness.

As shown in Fig. 2, the Reduction Cells are placed at one-third and two-thirds of the overall architecture. In the rest of the architecture, the Accurate Cells are placed before the second Reduction Cell, while the Robust Cells are placed after the second Reduction Cell. Meanwhile, instead of the common heuristic [49] that doubling the number of filters in the output whenever the spatial activation size is reduced, we double the number of filters at the first Reduction Cell, and keep the number of filters unchanged at the second Reduction Cell. The effectiveness of this design will be verified in Sec. 4.4.4, and the naming rules (i.e., how to determine which cells are for accuracy and which cells are for robustness) are explained in Sec. 4.4.5.

3.3. Differentiable Multi-Objective Search Strategy

To search for both accurate and robust architectures from the designed search space, the search process can be formulated by Eq. (1).

$$\begin{cases} \min\limits_{\alpha} & (\mathcal{L}_{val}^{std}(\omega^*(\alpha),\alpha),\ \lambda\mathcal{L}_{val}^{adv}(\omega^*(\alpha),\alpha)) \\ s.t. & \omega^*(\alpha) = argmin_{\omega}\ \mathcal{L}_{train}^{adv}(\omega,\alpha) \end{cases} \tag{1}$$

Specifically, Eq. (1) represents a bi-level optimization problem, where the lower-level formulation means updating network weights ω by minimizing adversarial loss $\mathcal{L}_{train}^{adv}(.)$ on the training set, and the upper-level formulation means updating architecture parameters to minimize both natural loss $\mathcal{L}_{val}^{std}(.)$ and adversarial loss $\mathcal{L}_{val}^{adv}(.)$ on the validation set. λ represents a regularization coefficient. The adversarial loss can be optionally replaced by other robustness metrics.

Existing methods [12, 20, 29] transform the upper-level optimization to a single-objective optimization problem by summing the two objectives, with the fixed regularization coefficient, formulated as Eq. (2).

$$\min_{\alpha}\quad \mathcal{L}_{val}^{std}(\omega^*(\alpha),\alpha) + \lambda\mathcal{L}_{val}^{adv}(\omega^*(\alpha),\alpha) \tag{2}$$

Eq. (2) can be optimized using gradient descent. However, as mentioned in Sec. 1, scholars studying multi-objective optimization have shown that it would be better for identifying the Pareto front to always find a descent direction common to all criteria [9, 10], while the fixed regularization coefficient cannot achieve this goal. To address this problem, we propose a multi-objective adversarial training method, based on MGDA [9, 36]. MGDA is a gradient-based multi-objective optimization algorithm that either finds a common descent direction for all objectives and performs gradient descent, or does nothing when the current point is Pareto-stationary [9].

Specifically, we first need to determine the weights of all objectives dynamically. With the weights, all objectives can be optimized simultaneously. Thanks to the two objectives in our method, the process of dynamically determining the weights can be simplified as Eq. (3),

$$\gamma^* = \underset{0\le\gamma\le1}{argmin} \left\|\gamma\theta + (1-\gamma)\bar{\theta}\right\|_2^2 \tag{3}$$

where $\theta = \nabla_\alpha\mathcal{L}_{val}^{std}(\omega^*(\alpha),\alpha)$, $\bar{\theta} = \nabla_\alpha\lambda\mathcal{L}_{val}^{adv}(\omega^*(\alpha),\alpha)$, ∇_α denotes the gradient with respect to α. Eq. (3) has an analytical solution, and can be calculated by Eq. (4).

$$\gamma^* = max(min(\frac{(\bar{\theta}-\theta)^T\bar{\theta}}{\|\theta-\bar{\theta}\|_2^2}, 1), 0) \tag{4}$$

Using the obtained weights, the upper-level problem of Eq. (1) can be transformed to Eq. (5), which ensures the simultaneous optimization of natural loss and adversarial loss, and can be implemented using gradient descent.

$$\min_{\alpha}\quad \gamma^*\mathcal{L}_{val}^{std}(\omega^*(\alpha),\alpha) + (1-\gamma^*)\lambda\mathcal{L}_{val}^{adv}(\omega^*(\alpha),\alpha) \tag{5}$$

Table 1. Evaluation results of adversarially trained models on CIFAR-10 under white-box attacks. The best result in each column is in bold, and the second best is underlined. PGD^{20} and PGD^{100} refer to PGD attacks with 20 and 100 iterations, respectively. AA refers to the evaluation result after the standard group of AutoAttack method. All attacks are l_∞-bounded with a total perturbation of 8/255.

Category	Model	Params	FLOPs	Natural Acc.	FGSM	PGD^{20}	PGD^{100}	$APGD_{CE}$	AA
Hand-Crafted	ResNet-18	11.2M	37.67M	84.09%	54.64%	45.86%	45.53%	44.54%	43.22%
	DenseNet-121	7.0M	59.83M	85.95%	58.46%	50.49%	49.92%	49.11%	47.46%
Standard NAS	DARTS	3.3M	547.44M	85.17%	58.74%	50.45%	49.28%	48.32%	46.79%
	PDARTS	3.4M	550.75M	85.37%	59.12%	51.32%	50.91%	49.96%	48.52%
Robust NAS	RobNet-free	5.6M	800.40M	85.00%	59.22%	52.09%	51.14%	50.41%	48.56%
	AdvRush	4.2M	668.53M	85.59%	59.98%	52.76%	52.55%	51.73%	49.28%
	RACL	3.6M	568.86M	83.97%	59.29%	52.13%	51.72%	51.24%	48.59%
	DSRNA	2.0M	336.23M	80.93%	54.49%	49.11%	48.89%	48.54%	44.87%
	NADAR	4.4M	700.00M	<u>86.23%</u>	60.46%	<u>53.43%</u>	<u>53.06%</u>	<u>52.64%</u>	<u>50.44%</u>
	ABanditNAS-10	5.2M	794.11M	**90.64%**	**81.31%**	50.51%	45.73%	29.31%	16.03%
	ARNAS(Ours)	4.5M	1.27G	85.92%	<u>62.45%</u>	**55.87%**	**55.43%**	**54.84%**	**52.66%**

We implement the above algorithm using the second-order approximation [27] of DARTS for competitive results. Meanwhile, the above algorithm involves the multiple computation of the same gradients of natural loss and adversarial loss, which can be saved and reused. In this way, the proposed algorithm does not require extra computational cost. At the end of search, an accurate and robust architecture can be obtained using the conventional process of DARTS.

4. Experiments

4.1. Benchmark Datasets

Following the conventions of robust NAS community [18, 29], CIFAR-10 [23], CIFAR-100, SVHN [30] and Tiny-ImageNet-200 [25] are chosen as benchmark datasets.

4.2. Peer Competitors

State-of-the-art architectures from three categories are chosen as peer competitors. Specifically, the hand-crafted architectures are ResNet-18 [19] and DenseNet-121 [21]. The architectures obtained by standard NAS are DARTS [27] and PDARTS [6]. The architectures obtained by robust NAS are our main competitors, *i.e.*, RobNet-free [18], AdvRush [29], RACL [12], DSRNA [20], NADAR [26], and ABanditNAS [5].

4.3. Parameter Settings

Search Settings: Following DARTS, we carry out architecture search on a small network consisting of 8 cells. The initial number of channels is set to 24, which is larger than conventions, aiming to keep the model capacity in the proposed search space similar to peer competitors for a fair comparison. Specifically, the numbers of channels divided by the two reduction cells are 24, 48, and 48, respectively. The total epoch is set to 50. To generate adversarial examples for the searching procedure, we use 7-step PGD with the step size of 2/255 and the total perturbation of 8/255. The regularization coefficient of the adversarial loss is set to 0.1. Remaining settings are the same as DARTS. We use momentum SGD to optimize the network weights ω, with initial learning rate $\eta_\omega = 0.025$ (annealed down to zero following a cosine schedule), momentum 0.9, and weight decay 3×10^{-4}. We use Adam as the optimizer for the architecture parameters, with initial learning rate $\eta_\alpha = 3 \times 10^{-4}$, momentum $\beta = (0.5, 0.999)$ and weight decay 10^{-3}.

Evaluation Settings: To evaluate the searched architecture, we stack 20 cells to form a large network with initial number of channels of 64. Therefore, the number of channels divided by the two reduction cells are 64, 128, and 128, respectively. Following conventions [29], we perform adversarial training using 7-step PGD with the step size of 0.01 and the total perturbation of 8/255 for 200 epochs. We use SGD to optimize the network, with the momentum of 0.9, and the weight decay of 1×10^{-4}. When evaluating on CIFAR-10 and CIFAR-100, the initial learning rate is set to 0.1. When evaluating on SVHN, the initial learning rate is set to 0.01. The learning rate is decayed by the factor of 0.1 at the 100-th and 150-th epoch. The batch size is set to 32. All the experiments are performed on a single NVIDIA GeForce RTX 2080 Ti GPU card.

4.4. Results

4.4.1 White-box Attacks

We adversarially train the searched architectures and evaluate them under FGSM, PGD^{20}, PGD^{100}, and the standard group of AutoAttack ($APGD_{CE}$, $APGD^{T}$, FAB^{T}, and Square) [8]. The results are shown in Tab. 1. The results show that the ARNAS architecture achieves the highest adversarial accuracy among all competitors under PGD^{20},

Table 2. Evaluation results of adversarially trained models on CIFAR-10 under transfer-based black-box attacks. In each row, the highest prediction accuracy except WRN-R (trained with additional 500k data) is in bold. In each column, the highest attack success rate (100% - prediction accuracy) is underlined.

Source \ Target	WRN-R (500k data)	ABanditNAS-10	AdvRush	ARNAS
WRN-R (500k data)	-	69.59%	68.99%	**70.03%**
ABanditNAS-10	84.82%	-	77.58%	**78.39%**
AdvRush	77.62%	68.83%	-	**66.81%**
ARNAS	<u>77.09%</u>	<u>67.80%</u>	<u>64.65%</u>	-

Table 3. Evaluation results of adversarially trained models on CIFAR-100, SVHN, and Tiny-ImageNet under white-box attacks.

Dataset	Model	Natural Acc.	FGSM	PGD^{20}
CIFAR-100	ResNet-18	55.57%	26.03%	21.44%
	PDARTS	**58.41%**	30.35%	25.83%
	ARNAS	58.18%	**32.60%**	**29.54%**
SVHN	ResNet-18	92.06%	88.73%	69.51%
	PDARTS	95.10%	93.01%	89.58%
	ARNAS	**95.84%**	**94.43%**	**92.02%**
Tiny-ImageNet -200	WideResNet	52.10%	27.82%	**24.83%**
	PDARTS	45.94%	24.36%	22.74%
	ARNAS	**54.18%**	**50.00%**	21.49%

PGD^{100}, and the standard group of AutoAttack, indicating that the ARNAS architecture is highly robust. Meanwhile, the natural accuracy of ARNAS outperforms all the competitors except DenseNet-121, ABanditNAS-10, and NADAR. Compared with them, the improvements of adversarial accuracy greatly exceeds the decrease of natural accuracy. Please note that AbanditNAS-10 only performs well under simple attacks such as FGSM. When the attacks get stronger, AbanditNAS-10 shows obviously lower accuracy than all other competitors, which indicates that AbanditNAS-10 is not actually trained to be robust.

In addition, the ARNAS architecture has 1.27G FLOPs, which is significantly more than other architectures, even though their numbers of parameters are similar. Given that the ARNAS architecture is more robust, we infer that the number of parameters and the FLOPs are both the factors that affect the robustness. The conclusion also explains why previous studies get totally contradictory conclusions about the effect of model parameters on the robustness, i.e., some studies showed that more parameters can improve adversarial robustness [41] while some others showed that more parameters may be harmful to adversarial robustness [22]. This may be because they ignore the influence of the FLOPs. Moreover, the large FLOPs are essentially caused by the special proportional relationship of channels designed in the proposed search space, which further demonstrates the effectiveness of the proposed search space. Specifically, the parameter size is proportional to the sum of channel numbers, while the FLOPs are proportional to the product of the channel numbers. The proposed search space keeps the sum of channel numbers similar to conventional search space, but the product of channel numbers is significantly larger, resulting in the architecture with similar parameter size but larger FLOPs.

4.4.2 Black-box Attacks

We conduct transfer-based black-box attacks, attacking the target model using adversarial examples generated by the source model. Adversarial examples from the source model are generated by PGD^{20} with the total perturbation scale of 8/255. Except for aforementioned competitors, we perform extended experiments on WRN-34-R, which is the most robust variant of WideResNet found by [22]. WRN-34-R is trained using additional 500k data, so it shows the highest accuracy and robustness. We are interested in how ARNAS behaves when facing such a highly robust model. The results are presented in Tab. 2.

The results show that ARNAS is more resilient against transfer-based black-box attacks than AdvRush and ABanditNAS. For example, when considering the model pair ABanditNAS-10 $\leftrightarrow$ ARNAS, ABanditNAS-10 $\rightarrow$ ARNAS achieves the attack success rate (100% - prediction accuracy) of 21.61%, while ARNAS $\rightarrow$ ABanditNAS-10 achieves the attack success rate of 32.20%, where there is a gap of 10.59%. Besides, when used as the target model (in each row), except for WRN-R that is trained with additional 500k data, ARNAS always has the highest prediction accuracy. When used as the source model (in each column), ARNAS always has the highest attack success rate, even higher than WRN-R. In conclusion, the black-box evaluation results further demonstrate the high robustness of the ARNAS architecture. Meanwhile, it proves that the ARNAS architecture does not unfairly benefit from the obfuscated gradients [1] because the transfer-based black-box attacks do not use the gradients of target models.

4.4.3 Transferability to Other Datasets

We transfer the ARNAS architecture to CIFAR-100, SVHN, and Tiny-ImageNet-200 to show its transferability. The results are shown in Tab. 3. When transferred to CIFAR-100, the ARNAS architecture is far better than ResNet-18. Compared with PDARTS, the ARNAS architecture reaches competitive natural accuracy, while its FGSM accuracy and PGD^{20} accuracy are significantly higher. When transferred to SVHN, the ARNAS perfroms best under all evaluation metrics. When transferred to Tiny-ImageNet-200, we replace ResNet-18 with a competitive model WideResNet-34-12 [45]. To our surprise, the ARNAS architecture reaches an unprecedented height in terms of natural and FGSM ac-

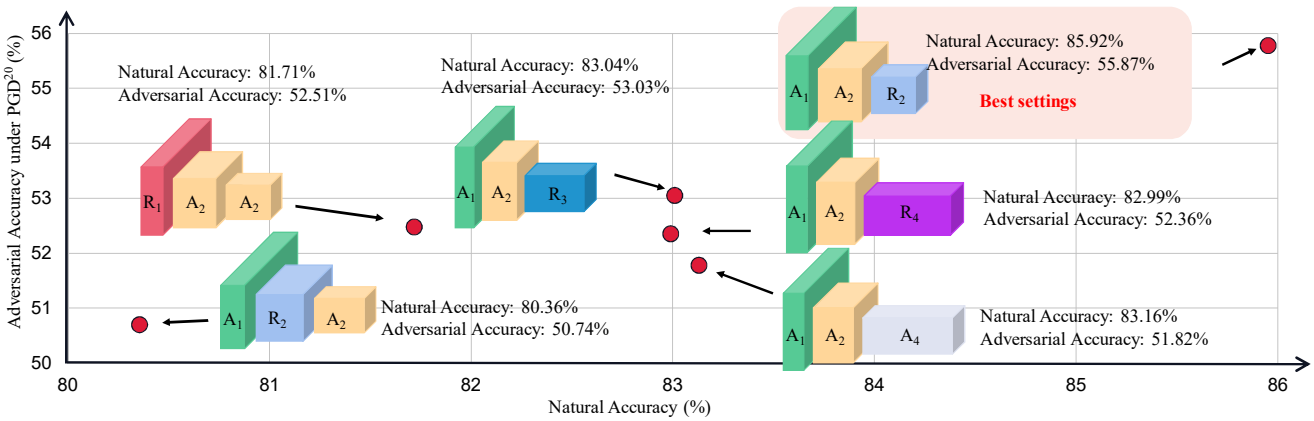

Figure 3. Visualization analysis of architectures in Tab. 4. Character R refers Robust Cell, character A refers to Accurate Cell, and the subscript of A and R refers to filter settings. For example, R_4 refers to Robust Cell with the number of filters four times the initial.

curacy. Notably, the ARNAS architecture is almost never defeated under FGSM attack. Its FGSM accuracy is 22.18% higher than WideResNet-34-12, and even close to the natural accuracy of WideResNet-34-12. The experimental results break the traditional prejudice that NAS-based architectures have weaker robustness than hand-crafted architectures as the dataset size or the task complexity increases [11].

4.4.4 Ablation Study

Ablation study of search Space. We try all possible placement of the Accurate Cells and the Robust Cells while the placement of the Reduction Cells is the same as the NASNet search space. When the placement is determined, we further study on the different settings of the number of filters. The results are shown in Tab. 4. To represent the placement of the cells, we use A for Accurate Cell and R for Robust Cell. For example, A-A-R means we place the Accurate Cells before the second Reduction Cell and the Robust Cells after the second Reduction Cell, which is the same as the proposed search space described in Sec. 3.2. For filter settings, we use N_1-N_2-N_3 to represent the proportional relationship of the number of filters. For example, 1-2-4 means the number of filters between the first Reduction Cell and the second Reduction Cell is two times the initial number of filters, and the number of filters after the second Reduction Cell is four times the initial number of filters, which is adopted by the NASNet search space.

When the filter setting is fixed to be 1-2-2 (experiments 1, 2 and 3), the best result is achieved when we set the placement to be A-A-R (experiment 1). So we fix the placement to be A-A-R and conduct further experiments. When the placement is fixed to be A-A-R (experiments 1, 4 and 5), the best result is achieved exactly when we set the number of filters to be 1-2-2 (experiment 1), which is adopted in the previous experiments. Therefore, we construct the proposed search space with the placement of A-A-R and the filter setting of 1-2-2. Compared with conventional search space (placement of A-A-A and filter setting of 1-2-4), the neural architecture searched from the proposed search space achieves 2.23% higher natural accuracy and 2.78% higher PGD^{20} accuracy. We also provide a visualization analysis in Fig. 3 for the above experimental results. As shown in the figure, a well-designed setting can largely improve the accuracy and the robustness (top right corner of the figure).

Table 4. Ablation study of search space on CIFAR-10.

Row Number	Placement	Filter Setting	Natural Acc.	PGD^{20}
1	A-A-R	1-2-2	**85.92%**	**55.87%**
2	R-A-A	1-2-2	81.71%	52.51%
3	A-R-A	1-2-2	80.36%	50.74%
4	A-A-R	1-2-3	83.04%	53.03%
5	A-A-R	1-2-4	82.99%	52.36%
6	A-A-A	1-2-4	83.16%	51.82%

Table 5. Ablation study of search strategy on CIFAR-10.

Row Number	ARNAS Search Space	Multi-Objective	Natural Acc.	PGD^{20}
1	✓	✗	85.04%	53.72%
2	✓	✓	85.92%	55.87%

Ablation study of search strategy. The innovation of the proposed search strategy lies in the proposed multi-objective adversarial training method. We compared it with the previous method (i.e., sum of two objectives with a fixed regularization term). The results are shown in Tab. 5. Using the proposed multi-objective adversarial training method, the searched architecture get both higher accuracy and robustness. Commonly, the optimization of architecture based on gradient descent does not guarantee convergence to the optimal solution. It is possible to see that the two conflict-

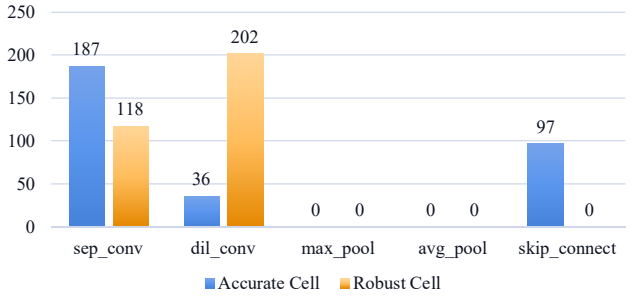

Figure 4. Statistical analysis on 40 neural architectures searched by the proposed method. The numbers of times of the operations selected by Accurate Cell and Robust Cell are recorded.

ing objectives get both higher. The results demonstrate that the proposed multi-objective adversarial training method is more effective than the previous one.

4.4.5 Analysis on Architectural Ingredients of Accurate and Robust Neural Networks

To get more insights of accurate and robust architectures, we repeat the proposed method for four times with different random seeds. For every execution, the top-10 architectures are recorded. Then a statistical analysis of these 40 neural architectures is carried out, and the results are shown in Fig. 4. We find that the ARNAS architectures tend to deploy very different structures for Accurate Cells and Robust Cells. Specifically, the Accurate Cells prefer separable convolutions while employing a few dilated convolutions and skip connections. However, the Robust Cells prefer dilated convolutions while employing a few separable convolutions and no skip connection. Such kind of neural architectures are impossible to be found in the conventional search space, because the conventional search space limits the cells near the input and output to be the same [49].

Next, we will analyze why Accurate Cells and Robust Cells contribute to accuracy and robustness, respectively. Please note that the skip connections are not in our consideration, because they have effects on the training process by accelerating gradient propagation [15], instead of improving the learning ability of the architecture. On the one hand, it is a consensus that more learnable parameters are beneficial for accuracy [44]. Existing practice is also consistent with the consensus. For example, architectures searched by popular standard NAS algorithms, such as DARTS and PDARTS, are mainly composed of separable convolutions, which requires the maximum number of parameters among all optional operations. The cell structures near the input in our experiments are in line with the consensus and the previous practice. This is exactly why we name these cells as Accurate Cells. On the other hand, it is also empirically and theoretically proved that more learnable parameters are harmful to the robustness [22]. Among the optional operations, the dilated convolutions fix some weights to be zero, so they have fewer learnable parameters. As a result, perturbations of input are difficult to change the output, resulting in the stronger robustness. In our experiments, the cells near the output tend to employ dilated convolutions instead of separable convolutions, and the robustness may be enhanced. This is exactly why we name these cells as Robust Cells. In conclusion, although accuracy and robustness are conflicting objectives, the cell structures in different positions play different roles for accuracy and robustness, and the architectures can obtain both accuracy and robustness by deploying very different structures in different positions.

The conclusion has great guiding significance on both hand-crafting and automatically designing of accurate and robust architectures. To our knowledge, though it is common to design architectures by stacking repeated structures nowadays, few people try to stack very different structures in different positions, which may greatly limit the performance of the designed architectures.

5. Conclusion

In this work, we propose the ARNAS method to search for accurate and robust neural architectures after adversarial training automatically. Specifically, we first design the ARNAS search space specially for adversarial training through experiments, which empirically improve the accuracy and the robustness of the searched neural architectures. Then we design a differentiable multi-objective search strategy, searching for accurate and robust architectures by performing gradient descent towards a common descent direction of natural loss and adversarial loss. We evaluate the searched architecture under various adversarial attacks on various benchmark datasets. The strongest robustness and outstanding accuracy of the searched architecture demonstrate the superiority of the proposed method. Meanwhile, the unexpected transferability break the traditional prejudice that NAS-based architectures are inferior to hand-crafted architectures as the task complexity increases in robustness scenario. Besides, we reach an important conclusion that the cell structures in different positions of the architectures play different roles for accuracy and robustness, and the architectures can obtain high accuracy and high robustness simultaneously by deploying very different cell structures in different positions. The conclusion has great guiding significance on both hand-crafting and automatically designing of accurate and robust neural architectures.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant 62276175.

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pages 274–283. PMLR, 2018. 6

[2] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *The 3rd International Conference on Learning Representations*, 2015. 1

[3] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In *International Conference on Machine Learning*, pages 550–559. PMLR, 2018. 1

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy*, pages 39–57. IEEE, 2017. 2

[5] Hanlin Chen, Baochang Zhang, Song Xue, Xuan Gong, Hong Liu, Rongrong Ji, and David Doermann. Anti-bandit neural architecture search for model defense. In *European Conference on Computer Vision*, pages 70–85. Springer, 2020. 2, 5

[6] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1294–1303, 2019. 3, 5

[7] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pages 854–863. PMLR, 2017. 2

[8] 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. 2, 5

[9] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012. 2, 4

[10] Jean-Antoine Désidéri. Mutiple-gradient descent algorithm for multiobjective optimization. In *European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012)*, 2012. 2, 4

[11] Chaitanya Devaguptapu, Devansh Agarwal, Gaurav Mittal, Pulkit Gopalani, and Vineeth N Balasubramanian. On adversarial robustness: A neural architecture search perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 152–161, 2021. 7

[12] Minjing Dong, Yanxi Li, Yunhe Wang, and Chang Xu. Adversarially robust neural architectures. *arXiv preprint arXiv:2009.00902*, 2020. 1, 2, 3, 4, 5

[13] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of JPG compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016. 3

[14] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. 1

[15] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 8

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *stat*, 1050: 20, 2015. 1, 2

[17] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. *The 6th International Conference on Learning Representations*, 2018. 3

[18] 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. 1, 2, 5

[19] 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. 5

[20] Ramtin Hosseini, Xingyi Yang, and Pengtao Xie. DSRNA: Differentiable search of robust neural architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6196–6205, 2021. 2, 3, 4, 5

[21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017. 1, 5

[22] 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, 2021. 2, 3, 6, 8

[23] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. 5

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012. 1

[25] Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5

[26] Yanxi Li, Zhaohui Yang, Yunhe Wang, and Chang Xu. Neural architecture dilation for adversarial robustness. *Advances in Neural Information Processing Systems*, 34:29578–29589, 2021. 2, 5

[27] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. *International Conference on Learning Representations*, 2019. 3, 5

[28] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *The 6th International Conference on Learning Representations*, 2018. 1, 2

[29] 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. 1, 2, 3, 4, 5

[30] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 5

[31] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pages 4970–4979. PMLR, 2019. 3

[32] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy*, pages 582–597. IEEE, 2016. 3

[33] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications security*, pages 506–519, 2017. 2

[34] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc V Le, and Alexey Kurakin. Large-scale evolution of image classifiers. In *International Conference on Machine Learning*, pages 2902–2911. PMLR, 2017. 3

[35] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4780–4789, 2019. 3

[36] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in Neural Information Processing Systems*, 31, 2018. 4

[37] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540, 2016. 1

[38] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *The 2nd International Conference on Learning Representations*, 2014. 1

[39] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *International Conference on Learning Representations*, 2018. 3

[40] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *International Conference on Learning Representations*, 2018. 2

[41] Cihang Xie and Alan Yuille. Intriguing properties of adversarial training at scale. *International Conference on Learning Representations*, 2020. 6

[42] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019. 3

[43] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. PC-DARTS: Partial channel connections for memory-efficient architecture search. *International Conference on Learning Representations*, 2020. 3

[44] Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017. 8

[45] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016. 6

[46] 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. 1, 2

[47] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. *International Conference on Learning Representations*, 2020. 2

[48] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *International Conference on Learning Representations*, 2017. 1, 3

[49] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, 2018. 2, 4, 8