TRNAS: A Training-Free Robust Neural Architecture Search

Yeming Yang[1], Qingling Zhu[1], Jianping Luo[1*], Ka-Chun Wong[2], Qiuzhen Lin[1*], Jianqiang Li[1]
[1] Shenzhen University
[2] City University of Hong Kong
[1] yangyeming2021@email.szu.edu.cn, {zhuqingling, ljp, qiuzhlin, lijq}@szu.edu.cn,
[2] kc.w@cityu.edu.hk.

Abstract

Deep Neural Networks (DNNs) have been successfully applied in various computer tasks. However, they remain vulnerable to adversarial attacks, which could lead to severe security risks. In recent years, robust neural architecture search (NAS) has gradually become an emerging direction for designing adversarially robust architectures. However, existing robust NAS methods rely on repeatedly training numerous DNNs to evaluate robustness, which makes the search process extremely expensive. In this paper, we propose a training-free robust NAS method (TRNAS) that significantly reduces search costs. First, we design a zero-cost proxy model (R-Score) that formalizes adversarial robustness evaluation by exploring the theory of DNN's linear activation capability and feature consistency. This proxy only requires initialized weights for evaluation, thereby avoiding expensive adversarial training costs. Secondly, we introduce a multi-objective selection (MOS) strategy to save candidate architectures with robustness and compactness. Experimental results show that TRNAS only requires 0.02 GPU days to find a promising robust architecture in a vast search space including approximately 10^{20} networks. TRNAS surpasses other state-of-the-art robust NAS methods under both white-box and black-box attacks. Finally, we summarize a few meaningful conclusions for designing the robust architecture and promoting the development of robust NAS field.

1. Introduction

Deep neural networks (DNNs) have been successfully applied in various computer vision tasks [23]. However, DNNs are vulnerable to adversarial attacks, in which small perturbations can mislead the network's judgment [36]. Thus, using DNNs in safety-critical fields such as autonomous driving and facial recognition remains challenging [30]. To address this challenge, researchers have proposed several defense strategies, among which adversarial training (AT) has been proven to be highly effective [10, 28]. The AT method depends on architectural design [35], which requires extensive expertise and tedious fine-tuning [10, 11, 15]. Nowadays, neural architecture search (NAS) methods can automatically and efficiently design DNNs, significantly saving time and resources [23]. However, the final designed architecture may lack robustness since many NAS methods are primarily designed for non-adversarial scenarios [15]. Therefore, developing a robust NAS method has become a key challenge in safety-critical fields.

In recent years, some robust NAS methods have been proposed by improving evaluation mechanisms and designing new search spaces to search robust DNNs automatically. The evaluation mechanisms of robust NAS determine whether a truly robust architecture can be searched. Initially, RACL [9] demonstrates that a lower Lipschitz constant of DNNs will exhibit better robustness. Based on this, RobustWRN [21] further explores the changing relationship between the Lipschitz constant and the network's topology. It provides deeper insights into how architectural features affect DNNs' robustness. In addition, RobNet [14] combines the AT process with a one-shot mechanism to train a robust supernet for fast evaluation. Subsequently, DSRNA [19] investigates the impact of certified lower bounds and Jacobian norm bounds on robust networks, thereby introducing new evaluation metrics. At the same time, AdvRush [29] focuses on the smoothness of the input loss landscape, which reveals the relationship between input perturbation smoothness and network robustness. Later, LRNAS [10] introduces Shapley statistical value, which simplifies the robust evaluation process. Although the above evaluation mechanisms are simple and efficient, they all require a repeated AT process.

Therefore, some training-free methods have been proposed to improve search efficiency. For example, CRoZe [15] is a pioneering study in training-free robust NAS. It

*Co-corresponding author.

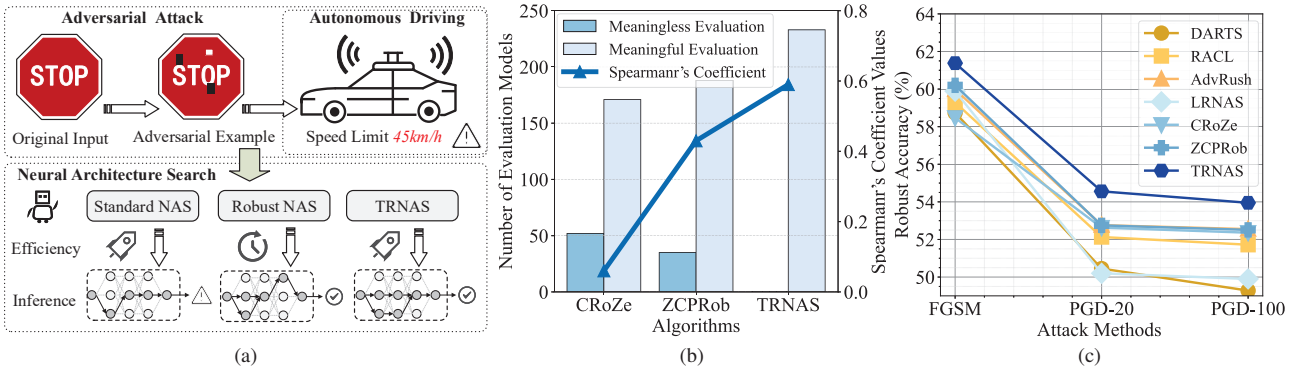


Figure 1. (a) The research motivation of TRNAS. (b) The evaluation effectiveness (evaluate 233 architectures [11]) and Spearman's consistency for existing robust proxies. (c) The robustness of TRNAS compared to SOTA robust NAS methods in various attack scenarios.

generates adversarial examples to attack the network and evaluates feature consistency during inference, significantly reducing search costs. Subsequently, ZCPRob [11] proposes a robust neural tangent kernel (NTK) theory, which avoids generating expensive adversarial examples and further improves search efficiency.

In addition, the search space of robust NAS defines the upper limit of architectural performance. For example, ABanditNAS [38] builds a robust search space by adding denoising blocks, weight dropout operations, and gabor filters. Subsequently, ARNAS [30] adds robust search cells to the DARTS [26] search space, significantly improving robustness performance under various attack scenarios.

Although the previous robust NAS performs powerfully, it still has weaknesses such as low efficiency, inaccurate evaluation, and multi-objective conflicts. These limitations motivate us to develop better robust NAS methods. As shown in Fig. 1(a), poor robustness networks are easily influenced by adversarial attacks. The robust NAS is a helpful answer, but it has lower design efficiency than the standard NAS [15]. Robust NAS usually combines weight-sharing mechanisms and expensive AT processes, which makes it 30 times slower than standard NAS [14]. Some pioneering robust NAS propose training-free proxies to alleviate efficiency problems, but they also introduce a new challenge of inaccurate evaluation [11, 15]. As shown in Fig. 1(b), existing training-free proxies cannot evaluate some architectures and exhibit weak Spearman's consistency on the RobustBench [11]. Finally, robust NAS naturally faces a multi-objective optimization problem of clean accuracy, robust accuracy, and compactness [10]. Some robust NAS methods convert multi-objective optimization into single-objective optimization by fixing the aggregation coefficient [19, 29] or adaptively adjusting it [30], but this may introduce bias. In summary, we still need to develop fast and high-performance robust NAS methods.

Therefore, in this paper, we propose a training-free evolutionary robust NAS method (TRNAS) to alleviate the previous weaknesses. As shown in Fig. 1(b) and 1(c), TRNAS has the strongest evaluation consistency and robustness under various attacks. We summarize our contributions as follows.

- We propose a lightweight proxy model (Rob-Score) that first introduces the linear activation capacity theory into robust DNNs and integrates feature consistency evaluation methods.
- We introduce a multi-objective selection (MOS) strategy to simultaneously consider the robustness performance (R-Score) and model compactness to enhance search diversity and stability.
- We complete the search on the DARTS search space in just 0.02 GPU days. The discovered model outperforms existing robust NAS models on multiple benchmark datasets in white-box, black-box, and transferability experiments.

2. Related Work

2.1. Adversarial Attacks and Defenses

Adversarial attacks can be divided into white-box and black-box attacks based on whether the attacker can access the target model's parameters, architecture, and training information [43]. Standard white-box attacks include Fast Gradient Sign Method (FGSM) [12], Projected Gradient Descent (PGD) [28], and Carlini & Wagner (C&W) [4]. In addition, a common black-box attack is a transfer-based attack [32]. This method generates adversarial examples on a surrogate model (similar to the target model) and then transfers them to attack the target model. Moreover, AutoAttack [7] integrates various white-box and black-box attack techniques to evaluate adversarial robustness fairly.

At the same time, various defense techniques, such as adversarial training (AT) [28], defensive distillation [31], and data compression [13], have been proposed to protect DNNs

Methods	Evaluation Metric(s)	Training-Free?
RACL [9]	Lipschitz Constant	×
RobNet [14]	Flow of Solution Procedure Matrix	×
NADAR [24]	Neural Architecture Dilation	×
AdvRush [29]	Input Loss Landscape Smoothness	×
DSRNA [19]	Jacobian Regularization	×
ABanditNAS [38]	Upper /Lower Confidence Bounds	×
LRNAS [10]	Shapley Statistical Value	×
CRoZe [15]	Network Character Consistency	√
ZCPRob [11]	Robust Neural Tangent Kernel	√
TRNAS	Activation Capability and Consistency	√

Table 1. Comparison of TRNAS with existing robust NAS methods in terms of their used evaluation metrics and training type.

from adversarial attacks. Among them, AT [28] is the most widely used method to enhance the robustness of DNNs. This method uses adversarial examples from multiple perturbations to improve the generalization of DNNs against adversarial attacks. However, the effectiveness of AT can be affected by the network architecture [10]. Low-quality architectures result in reduced accuracy and robustness. To alleviate this, robust NAS methods are adopted.

2.2. Neural Architecture Search

Early NAS methods often require training each subnet from scratch [2], which consumes thousands of GPU days. To improve the efficiency of NAS, ENAS [34] and One-shot NAS [3] introduce a weight-sharing mechanism, thereby significantly reducing the search time to within 1 GPU day [17, 27]. Subsequently, training-free NAS methods use zero-cost proxies based on interpretable theories to evaluate the performance of DNNs, which reduces search time to under 1 GPU hour [23]. Early zero-cost proxies [1] explore various theories, such as $grad\_norm$, $snip$, and $synflow$. However, the performance of these proxies is still limited. Recently, some training-free methods [33, 39] have introduced novel interpretable theories, such as neural tangent kernel and linear activation capability [5], to improve the quality of evaluation. However, previous methods focus on clean accuracy and overlook robustness. Therefore, TRNAS proposes a training-free robust NAS to alleviate the above challenge.

2.3. A Brief Review of Robust NAS

Currently, several research studies have used NAS to design robust networks. We analyze them from evaluation metrics and training-free search. First, RACL [9] was the first robust NAS method that used the Lipschitz constant constraint to improve network robustness. Then, RobNet [14] maintains a robust supernet and directly searches for architectures using the solution procedure matrix flow. Subsequently, NADAR [24] further improves adversarial robustness through a neural architecture dilation mechanism. At the same time, AdvRush [29] believes that the input loss landscape of the network is highly related to intrinsic robustness, which can be effectively used for evaluation. Then, DSRNA [19] finds that certified lower bounds and Jacobian norm bounds remain very effective in intense adversarial scenarios. ABanditNAS [38] further explores the application of upper and lower confidence bounds in robust NAS. In addition, LRNAS [10] uses Shapley values to quantify the contribution of each operation in the network, which maintains the network's robustness and compactness. However, the previous methods still require maintaining a robust supernet, and face challenges such as weak ranking consistency and long search times. Therefore, CRoZe [15] introduces a character consistency proxy, which reduces the search time to just 0.2 GPU days. Subsequently, ZCPRob [11] integrates robust NTK theory with the NAS method, which improves ranking consistency and search efficiency.

As shown in Table 1, few studies consider the training-free robust metrics. In this work, we propose a training-free proxy (R-Score) and introduce a multi-objective selection (MOS) strategy to ensure both efficiency and stability without relying on handcrafted design.

3. Methods

The search process of TRNAS can be mathematically formulated as the following optimization problem:

$$\alpha^* = \arg\max_{\alpha \in \mathcal{A}} F_{(x,x',\alpha)}\left[f_x^{std}(\alpha), f_{x,x'}^{adv}(\alpha, W_\alpha, W'_\alpha)\right] \tag{1}$$

where F is used to evaluate the performance of architecture α. Specifically, f_x^{std} measures the clean accuracy of α under the standard input x. Then, $f_{x,x'}^{adv}$ represents the robust accuracy of α under the adversarial attack input x'. In addition, W represents the random weights for architecture α, while W' is the weights after simple interference. Each architecture is sampled from the search space $\mathcal{A}$. TRNAS designs the architecture α^* suitable for complex scenarios by maximizing F.

3.1. The Complete Framework of TRNAS

The overall framework of TRNAS is illustrated in Fig. 2. After population initialization, it includes three key steps: reproduction, evaluation, and selection. First, Step 1 (Reproduction) undergoes multi-point crossover and mutation to generate new offspring P_{off}, which must satisfy the parameter capacity constraint. Next, Step 2 (Evaluation) evaluates the robust performance (R-Score) of architectures in both the parent and offspring populations. Finally, Step 3 (Selection) selects the superior architectures using a MOS selection strategy based on R-Score and Params (the number of parameters).

The complete pseudo code of TRNAS can be found in **Algorithm 1**. TRNAS initializes a population P of size N and generates a high-performance population through

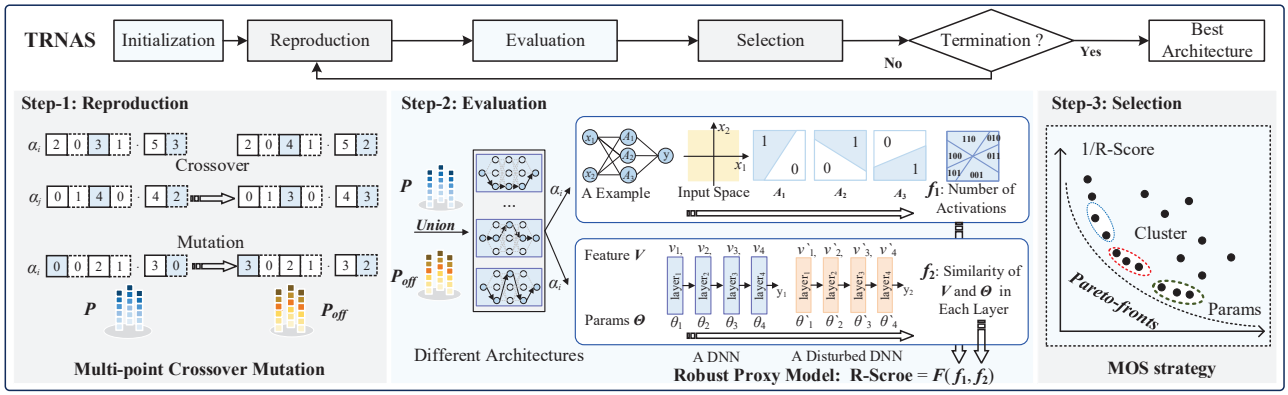


Figure 2. The framework of TRNAS.

Algorithm 1: Evolutionary Search for TRNAS

Input: Population size N;
Output: The final population of architectures P;

$P = Initialization(N)$;
while *termination criterion is not met* **do**
  while $|P| < 2 \times N$ **do**
    // Multi-point Crossover Mutation
    $P_{off} = Reproduction(P)$;
    $P = P_{off} \cup P$;
  // Robust Proxy Mode Evaluation
  $F_P = Evaluation(P)$;
  // Multi-objective Selection Strategy
  $P = Selection(P, F_P)$;

iteration. In each iteration, TRNAS performs multi-point crossover mutation on the high-quality population, which generates new offspring and merges them with the original population. Then, the robust proxy model (R-Score) is used to evaluate the population, which predicts the robustness of each architecture. Finally, a multi-objective selection (MOS) strategy is employed to update the population, which selects robust and diverse architectures.

3.2. Search Space of TRNAS

We adopt the DARTS [26] search space, consistent with previous robust NAS methods [11, 15], to ensure fair validation, as shown in Fig. 3. The DARTS space comprises normal and reduction cells containing four network nodes. Additionally, each cell has 14 edges, of which only eight edges are retained. Each edge offers a choice of 8 candidate operations, resulting in approximately 10^{20} possible architectures.

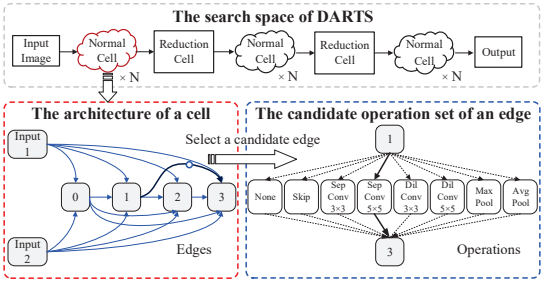


Figure 3. Overview of the search space.

3.3. Robust Proxy Model R-Score

TRNAS proposes the robust proxy (R-Score), inspired by the research on the expressivity [5, 23, 33] and consistency [15] theories of DNNs. As shown in step-2 of Fig. 2, the R-Score evaluates the network's robustness by measuring the activation patterns and the architecture's feature consistency. From the upper half, the number of linear regions represents how the network $\mathcal{N}$ can divide the input space (spanned by x_1 and x_2) into different parts. Assuming each layer in the network contains a single ReLU function, the approximate activation pattern $A_{\mathcal{N},\Theta}$ can be evaluated, as shown in Eq. (2).

$$A_{\mathcal{N},\Theta} = \left\{ \mathbf{p}^{(i)} : \mathbf{p}^i = \mathbf{1}\left(p_s^i\right)_{s=1}^{S}, i \in \{1, \ldots, I\} \right\} \quad (2)$$

where p_s^i denotes the binary activation value of the s-th sample in the i-th layer, and $\mathbf{p}^i$ represents the activation pattern of all training samples S in the i-th layer. The DNN's parameter Θ is randomly initialized. The final DNN's expressivity value $\Psi_{\mathcal{N},\Theta}$ is obtained by de-duplicating and normalizing $A_{\mathcal{N},\Theta}$.

Then, the second half of Step 2 describes the evaluation metric $\Phi_{\mathcal{N}}$ for architectural consistency. Initially, we generate attack examples x' through FGSM [12] to perturb the neural network $\mathcal{N}$. After gradient updates, we obtain the

adversarial network $\mathcal{N}'$ and the perturbation parameter θ'. Subsequently, we evaluate the feature consistency V of network $\mathcal{N}$ before and after the attack using cosine similarity, as shown in Eq. (3).

$$V_i(\mathcal{N}(x), \mathcal{N}'(x')) = 1 + \frac{\mathbf{v}_i \cdot \mathbf{v}'_i}{\|\mathbf{v}_i\| \|\mathbf{v}'_i\|} \tag{3}$$

Next, the consistency of DNN's parameter Θ is evaluated, as shown in Eq. (4).

$$\Theta_i(\theta, \theta') = 1 + \frac{\theta_i \cdot \theta'_i}{\|\theta_i\| \|\theta'_i\|} \tag{4}$$

where i in the Eqs. (3) and (4) represents the number of layers in the neural network, while v and θ represent the feature vector and parameter vector of the i-th layer, respectively. Then, we calculate the total feature consistency value Φ by multiplying the feature and parameter consistency values layer by layer and then summing the results, as shown in Eq. (5).

$$\Phi_{\mathcal{N},\Theta} = \sum_{i=1}^{I} V_i \times \Theta_i \tag{5}$$

Finally, we can obtain the R-Score of the final robust architecture α by integrating the network's expressivity Ψ and consistency Φ, as shown in Eq. (6).

$$F(\mathcal{N}(\alpha)) = \Psi_{\mathcal{N},\Theta}^{\frac{1}{2}} \times \Phi_{\mathcal{N},\Theta} \tag{6}$$

3.4. Multi-objective Selection Strategy

TRNAS employs the multi-objective selection (MOS) strategy to refine the architectural population P, with the goal of discovering robust, compact, and diverse architectures. This strategy is inspired by previous multi-objective protective selection [41, 42, 44]. In the first stage, MOS adopts a decomposition method [42] to select α from P, aiming to balance robustness and complexity as formalized in Eq. (7):

$$\begin{gathered} \min g^{te}(\alpha \mid I, Z^*) = \max_{1 \le j \le m} \left\{ I_j \cdot \left| f_j(\alpha) - Z_j^* \right| \right\} \\ \text{s. t. } \alpha \in \Omega \end{gathered} \tag{7}$$

The performance of each α is saved in a vector $f(\alpha) = [1/\text{R-Score}, \text{Params}]$, which is normalized via min-max scaling to ensure that the proxy score and the lightweight constraint are considered. This equation minimizes the largest weighted deviation between the architecture's performance and the ideal vector Z^*, where I_j denotes the importance of the j-th objective. After ranking all $2N$ architectures, the top $(N - e)$ architectures are stored in S_{top}, while the next $(N + e)$ architectures are assigned to S_{clu} for diversity preservation.

Then, MOS applies a k-means clustering method [25] to perform clustering based on the Euclidean distance between the performance vectors of each architecture on S_{clu}, as shown in Eq. (8). Then, the objective of MOS is to minimize the within-cluster variance.

$$cluster = \min_{\{C_1, \ldots, C_k\}} \sum_{i=1}^{e} \sum_{F \in C_i} \|F - \mu_i\|^2 \tag{8}$$

where μ_i denotes the centroid of the i-th cluster C_i. MOS clusters S_{clu} into e clusters. Next, TRNAS selects the nearest architecture to each μ_i from cluster C_i, keeping e architectures in S_{clu}. These are then combined with S_{top} to form the final architectural population P of size N. Through this strategy, TRNAS can find lightweight and robust DNNs.

4. Experiments

4.1. Experimental Dataset

As demonstrated by previous robust validation experiments [11, 15, 29], the CIFAR-10, CIFAR-100 [22], and ImageNet [8] datasets are employed to evaluate the robustness and generalization ability of the architectures found by TRNAS. In addition, we use RobustBench [11] and NAS-Rob-Bench-201 [40] to evaluate the clean and robust accuracies of multiple training-free NAS methods across the DARTS and NAS-Bench-201 spaces.

4.2. Parameter Settings

In the search stage, we use the **Algorithm 1** to search for robust architectures. Search cost is measured in GPU days [11, 15]. First, the search space consists of 8 stacked cells in the DARTS framework. Secondly, TRNAS performs a total of 20 searches during this stage. The population size is 50, and the e value used in the clustering process is 20. After the search, the selected best architecture undergoes further adversarial training. In the training stage, we follow the previous robust NAS [11, 29] settings and use a 7-step PGD for adversarial training. In this phase, the neural architecture consists of 20 stacked cells. Specifically, the learning rate is 0.01, and the total perturbation range is 8/255. Then, we use SGD to optimize the network parameters. The learning rate is initially set to 0.1 and decays to 0.01 at the 100th training epoch. Finally, the training batch size is 64, and the number of training epochs is 120. In the testing stage, we choose FGSM [12], PGD [28], and AutoAttack [7] adversarial attacks to test the adversarial robustness of DNNs after the training. The total perturbation size of the above attacks is 8/255. In addition, the single-step perturbation scale of the PGD attack is 2/255 and we evaluate it using both 20-step and 100-step versions. All experiments were conducted on an RTX 4090 GPU using the PyTorch 2.0 framework.

4.2.1. Some Compared Algorithms

We select competitive robust NAS algorithms that are widely used in prior studies. For manually designed networks, we include ResNet-18 [16] and DenseNet-121 [20].

Table 2. Comparison of State-of-the-Art NAS Methods on CIFAR-10

Methods	Category	Params	FLOPs	Clean Acc	FGSM	PGD^{20}	PGD^{100}	AutoAttack	GPU Days	Training-Free?
ResNet-18 [16]	Manual	11.2	**37.67**	84.09	54.64	45.86	45.53	43.22	-	-
DenseNet-121 [20]		7.0	59.83	**85.95**	58.46	50.49	49.92	47.46	-	-
DARTS [26]	Standard	3.3	547.44	85.17	58.74	50.45	49.28	46.79	1.0	×
PDARTS [6]		3.4	550.75	85.37	59.12	51.32	50.91	48.52	0.3	×
SWAP [33]		4.4	680.31	85.45	60.84	53.90	53.52	50.68	**0.01**	√
RoBoT [18]		3.0	492.75	85.03	59.36	52.87	52.39	49.78	0.6	√
RACL [9]	Robust	3.6	568.86	83.97	59.29	52.13	51.72	48.59	0.5	×
RobNet [14]		5.6	800.40	85.00	59.22	52.09	51.14	48.56	3.3	×
AdvRush [29]		4.2	668.53	85.59	59.98	52.76	52.55	49.28	0.7	×
DSRNA [19]		**2.0**	336.23	80.93	54.49	49.11	48.89	44.87	0.4	×
LRNAS [10]		2.2	346.10	84.26	59.89	50.20	49.90	49.07	0.4	×
CRoZe [15]		5.5	841.00	83.30	58.47	52.63	52.36	49.37	0.2	√
ZCPRob [11]		3.4	555.54	85.60	60.20	52.75	52.51	49.97	0.02	√
TRNAS	Robust	4.3	673.28	85.66	**61.38**	**54.56**	**53.95**	**51.76**	0.02	√

For standard NAS, we consider DARTS [26], PDARTS [6], SWAP [33], and RoBoT [18], with the last two being training-free NAS methods. For robust NAS, we evaluate RACL [9], RobNet [14], AdvRush [29], DSRNA [19], LRNAS [10], CRoZe [15], and ZCPRob [11], where the last two are training-free robust NAS methods.

4.3. Results and Analysis

4.3.1. White-Box Attacks

TRNAS performs an architecture search on CIFAR-10. Then, we conduct adversarial training on the searched architecture. Subsequently, we evaluate these architectures using adversarial attacks such as FGSM, PGD, and AutoAttack. The results of the white-box attack experiments on CIFAR-10 and CIFAR-100 are shown in Tables 2 and 3.

As shown in Table 2, TRNAS has satisfactory robustness and clean accuracy on CIFAR-10. The final architecture's performance surpasses that of most similar robust NAS methods. In terms of clean accuracy, TRNAS achieves a performance of 85.66%. This result is better than previous robust NAS methods like LRNAS [10], CRoZe [15], and ZCPRob [11]. In comparison, its clean accuracy is only slightly lower than that of DenseNet-121 [20]. Regarding robustness, TRNAS performs exceptionally well under four types of adversarial attacks. The adversarial accuracy of TRNAS is 61.38%, 54.56%, 53.95%, and 51.76%, respectively. These results outperform other robust NAS methods. For example, compared to ZCPRob [11], TRNAS has improved accuracy under all adversarial attacks. In addition, TRNAS's computational efficiency is outstanding, completing search in just 0.02 GPU days. Training-based robust NAS methods usually take more than 0.4 GPU days [10, 14, 29]. In training-free robust NAS, TRNAS's search speed is 10× faster than CRoZe's [15], and it also outperforms ZCPRob [11] in robustness.

In Table 3, we further evaluate the performance of TRNAS on the CIFAR-100 dataset. From a clean accuracy perspective, TRNAS achieves 61.32%. This result surpasses other robust NAS methods, demonstrating its transferability to complex data. In terms of adversarial accuracy, TRNAS also achieves excellent results. Compared to ZCPRob [11], TRNAS improves the accuracy by 2.01%, 1.52%, 1.44%, and 1.73% under FGSM, PGD^{20}, PGD^{100}, and AutoAttack, respectively. Overall, TRNAS achieves a better balance between robustness and efficiency.

Table 3. Comparison of SOTA NAS Methods on CIFAR-100

Methods	Clean Acc	FGSM	PGD^{20}	PGD^{100}	AutoAttack
DARTS [26]	59.14	30.35	25.66	25.40	22.65
SWAP [33]	59.63	33.91	29.49	29.17	26.51
RoBoT [18]	60.47	33.19	29.21	28.86	26.29
RACL [9]	59.18	32.04	26.61	26.20	22.92
LRNAS [10]	57.42	32.19	27.93	27.71	24.99
CRoZe [15]	58.22	32.65	28.77	28.66	25.96
ZCPRob [11]	60.53	33.04	29.08	28.92	25.59
TRNAS	**61.32**	**35.03**	**30.60**	**30.36**	**27.32**

Table 4. Transfer-based Black-box Attacks on CIFAR-10

Source \ Target	SWAP	RoBoT	CRoZe	ZCPRob	TRNAS
SWAP [33]	-	64.76	63.53	63.46	**64.87**
RoBoT [18]	65.14	-	63.69	64.30	**65.44**
CRoZe [15]	64.55	64.52	-	63.24	**65.01**
ZCPRob [11]	64.90	65.34	63.90	-	**65.57**
TRNAS	**63.82**	**63.90**	**63.28**	**62.93**	-

4.3.2. Black-Box Attacks

We perform transfer-based attacks to investigate the performance of TRNAS under black-box attacks [11, 15]. We employ adversarial examples generated by the source model to perform an attack on the target model, and the results are shown in Tables 4 and 5.

Table 5. Transfer-based Black-box Attacks on CIFAR-100

Source \ Target	SWAP	RoBoT	CRoZe	ZCPRob	TRNAS
SWAP [33]	-	39.57	38.88	37.38	**40.18**
RoBoT [18]	39.74	-	38.71	38.35	**40.91**
CRoZe [15]	39.70	39.40	-	37.66	**40.71**
ZCPRob [11]	39.49	40.09	38.94	-	**41.56**
TRNAS	39.17	39.27	38.70	37.82	-

Table 6. Comparison of SOTA NAS methods on ImageNet

Methods	Clean ACC	FGSM	PGD^{20}	PGD^{100}	AutoAttack
SWAP [33]	54.19	20.44	11.36	10.30	8.81
RoBoT [18]	51.97	17.84	10.34	9.64	7.93
CRoZe [15]	49.52	16.28	9.41	8.87	7.12
ZCPRob [11]	52.93	18.86	10.75	9.92	8.25
TRNAS	**55.10**	**20.56**	**11.73**	**10.81**	**9.08**

TRNAS exhibits powerful adversarial attack capabilities. In each column, when TRNAS serves as the source model, the target model often achieves the lowest accuracy (highlighted in bold gray), indicating that the adversarial examples generated by TRNAS possess the strongest transfer-based black-box attack effectiveness. Moreover, TRNAS also demonstrates strong robustness. As shown in the last column, TRNAS achieves the highest robustness against adversarial examples generated by multiple robust architectures.

In Table 4, TRNAS's attack success rate against ZCPRob is 37.07% when used as the attack model. In contrast, the success rate of ZCPRob's attack on TRNAS is 34.43%. This result indicates that TRNAS has stronger robustness when facing transfer-based black-box attacks. A similar phenomena is presented in Table 5. On the CIFAR-100 dataset, TRNAS's attack success rate against ZCPRob is 62.18%, while ZCPRob's against TRNAS is 58.44%.

4.3.3. Transferability to Other Datasets

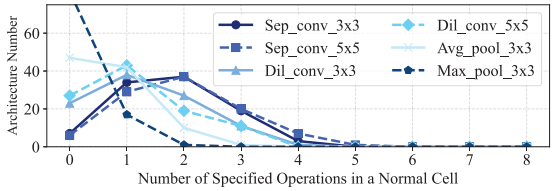
We transfer the searched architecture to ImageNet [8] to demonstrate its transferability. The experimental results presented in Table 6 clearly demonstrate a significant improvement in robust performance. Our architecture consistently outperforms SWAP [33], RoBoT [18], CRoZe [18], and ZCPRob [11] in terms of clean accuracy, FGSM accuracy, and PGD accuracy. In addition, under AutoAttack, we outperform the SOTA robust NAS methods by 9.08%.

4.3.4. Analysis on NAS-Rob-Bench-201

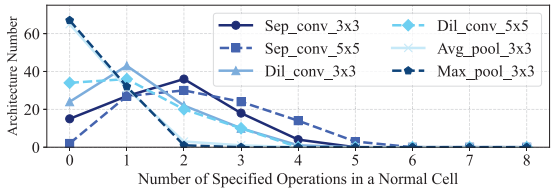
As shown in Table 7, we further test the latest NAS-Rob-Bench-201 [40] benchmark. We strictly followed the description of the comparison algorithms [10, 11, 15, 33] and uniformly used 1,000 evaluation numbers for the search. TRNAS can find a promising architecture in the benchmark within 130 evaluations, surpassing other SOTA methods.

Table 7. Results on NAS-Rob-Bench-201

Methods	Clean ACC	FGSM (3/255)	PGD (3/255)	FGSM (8/255)	PGD (8/255)
DARTS [26]	33.2	28.6	28.5	21.5	21.3
SWAP [33]	78.2	68.6	68.1	52.3	47.2
LRNAS [10]	72.9	63.7	63.2	48.5	44.3
CRoZe [15]	77.2	67.5	67.0	51.4	42.7
ZCPRob [11]	77.9	68.2	67.9	51.9	47.0
TRNAS	**79.6**	**69.7**	**69.2**	**53.5**	**48.1**



(a) The Top-100 Architectures Selected by ZCPRob



(b) The Top-100 Architectures Selected by TRNAS

Figure 4. Distribution of Architectural Operations

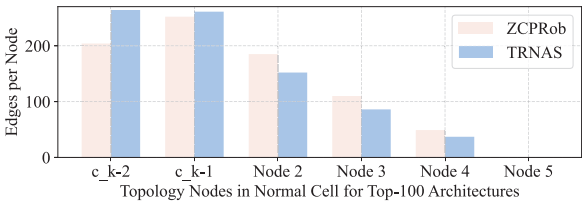


Figure 5. Topology Difference for the Top 100 Architectures

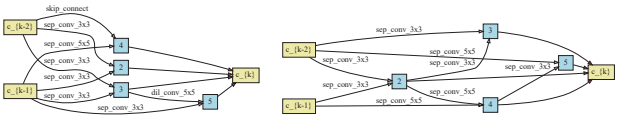
4.4. Architecture Analysis of Robust DNNs

TRNAS follows the reproducible random sampling method of ZCPRob [11], sampling 2,000 architectures. Fig. 4 shows the differences in the operational distribution of ZCPRob [11] and TRNAS in the top 100 architectures. There is a slight difference between the two in the distribution of most operations. However, TRNAS tends to use convolution to replace pooling layers with more learning parameters. In addition, TRNAS prefers max pooling layers and depthwise separable convolutions, thereby making the architecture robust.

The statistical analysis of the topology architectures in Fig. 5 reveals that TRNAS's high performance may stem from its higher topological density in the first half of the

Table 8. Efficiency Analysis of Training-Free Robust Methods

Methods	All Architectures	Single	Effective Rate
CRoZe [15]	4075 s	2.04 s	79.30 %
ZCPRob [11]	9322 s	4.66 s	98.84 %
TRNAS	4845 s	2.42 s	100.0 %



(a) Normal Cell (b) Reduction Cell

Figure 6. One of the High-quality Architectures of TRNAS.

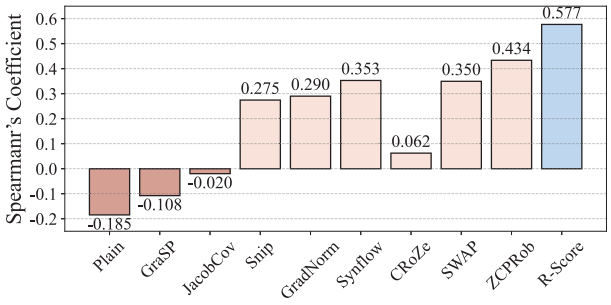


Figure 7. Spearmanr's Coefficient of Different Zero-cost Proxies on RobustBench Dataset.

cell (such as nodes c_k-2 and c_k-1). In contrast, ZCPRob [11] has more connections in the subsequent nodes (such as Node 2 and beyond) but with a sparser distribution. This indicates that TRNAS efficiently enhances overall performance by strengthening feature extraction capabilities in the network's earlier layers. At the same time, the contribution of topological complexity in the later stages of the architecture to robustness is relatively limited.

We further analyze the computational efficiency of existing robustness predictors, as shown in Table 8. We evaluate 2,000 architectures and find that CRoZe [15] has the fastest evaluation speed, requiring only 2.04 seconds per architecture. However, CRoZe [15] faces the problem of low evaluation effectiveness due to the tendency for gradient orthogonality when calculating gradient consistency. This results in 20.7% of high-performance architectures being considered extremely poor (0 scores). ZCProb [11] and TRNAS have relatively high evaluation effectiveness, but TRNAS is twice as fast in evaluation speed as ZCProb. Finally, TRNAS's high-quality architecture is shown in Fig. 6.

4.5. Ablation Experimental Analysis

4.5.1. Ablation of R-Score in RobustBench

To demonstrate the superiority of the training-free proxy R-Score, we compare it with SOTA proxy methods on RobustBench, as shown in Fig. 7. R-Score achieves the highest Spearman correlation coefficient, which indicates the strongest correlation between its predictions and adversarial robustness. Additionally, the ZCPRob [11], SWAP [33], and Synflow [37] methods also exhibit strong consistency. In contrast, CRoZe [15] exhibits lower prediction accuracy, primarily due to the gradient consistency theory used in this method, which tends to encounter gradient orthogonality issues when predicting deep networks. For specific architectures, CRoZe's gradient consistency may drop to zero, rendering it incapable of accurately predicting performance. Notably, some classic proxy models demonstrate a negative correlation with adversarial robustness. These models are designed based on clean loss environments, where clean loss often conflicts with adversarial loss.

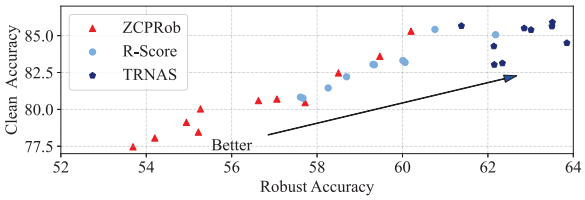


Figure 8. Ablation Analysis

4.5.2. Ablation Analysis of MOS strategy

As shown in Fig. 8, our baseline is ZCPRob [11]. This algorithm uses a robust NTK [11] to evaluate, and then combines it with a random search method. R-Score replaces NTK [11], while TRNAS adds a MOS strategy on the R-Score proxy for searching. TRNAS combines R-Score evaluation and the MOS selection strategy, which achieves the best performance with clean and robust accuracy. R-Score improves robust accuracy but slightly lowers clean accuracy, which indicates its effectiveness in robust architecture search. ZCPRob achieves high clean accuracy but low robust accuracy, which indicates insufficient attention to robustness. The results obtained by TRNAS validate the advantages of combining R-Score proxy and MOS strategy.

5. Conclusion

This paper proposes a training-free robust NAS method called TRNAS, which includes the R-Score and MOS strategy. R-Score aims to alleviate the high computational costs and unstable search problems in robust NAS. At the same time, the MOS strategy aims to produce diverse, superior, robust candidates for the search. TRNAS outperforms existing robust NAS methods on multiple benchmark datasets, which provides new insights for robustness research.

6. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 62376163, 62325307 and 62176161; in part by the Guangdong Regional Joint Foundation Key Project under Grant 2022B1515120076; in part by the Shenzhen Natural Science Foundation (the Stable Support Plan Program) under Grant 20231122104038002; and in part by the Shenzhen Science and Technology Program under Grants JCYJ20220531101411027, JCYJ20220531101614031, JCYJ20220818100005011, and KJZD20230923113801004.

References

[1] Mohamed S Abdelfattah, Abhinav Mehrotra, Łukasz Dudziak, and Nicholas Donald Lane. Zero-cost proxies for lightweight nas. In *International Conference on Learning Representations*, 2021. 3

[2] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing neural network architectures using reinforcement learning. In *International Conference on Learning Representations*, 2016. 3

[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. 3

[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] Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. In *International Conference on Learning Representations*, 2021. 3, 4

[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. 6

[7] 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

[8] 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. 5, 7

[9] Minjing Dong, Yanxi Li, Yunhe Wang, and Chang Xu. Adversarially robust neural architectures. *arXiv preprint arXiv:2009.00902*, 2020. 1, 3, 6

[10] Yuqi Feng, Zeqiong Lv, Hongyang Chen, Shangce Gao, Fengping An, and Yanan Sun. LRNAS: Differentiable searching for adversarially robust lightweight neural architecture. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 1, 2, 3, 6, 7

[11] Yuqi Feng, Yuwei Ou, Jiahao Fan, and Yanan Sun. Zero-cost proxy for adversarial robustness evaluation. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2, 3, 4, 5, 6, 7, 8

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *stat*, 1050: 20, 2015. 2, 4, 5

[13] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. 2

[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. 1, 2, 3, 6

[15] Hyeonjeong Ha, Minseon Kim, and Sung Ju Hwang. Generalizable lightweight proxy for robust NAS against diverse perturbations. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 3, 4, 5, 6, 7, 8

[16] 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, 6

[17] Xin He, Jiangchao Yao, Yuxin Wang, Zhenheng Tang, Ka Chun Cheung, Simon See, Bo Han, and Xiaowen Chu. NAS-LID: efficient neural architecture search with local intrinsic dimension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7839–7847, 2023. 3

[18] Zhenfeng He, Yao Shu, Zhongxiang Dai, and Bryan Kian Hsiang Low. Robustifying and boosting training-free neural architecture search. In *The Twelfth International Conference on Learning Representations*, 2024. 6, 7

[19] 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. 1, 2, 3, 6

[20] 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. 5, 6

[21] 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. 1

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[23] Guihong Li, Duc Hoang, Kartikeya Bhardwaj, Ming Lin, Zhangyang Wang, and Radu Marculescu. Zero-shot neural architecture search: Challenges, solutions, and opportunities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1, 3, 4

[24] 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. 3

[25] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2):451–461, 2003. 5

[26] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. *International Conference on Learning Representations*, 2019. 2, 4, 6, 7

[27] Shun Lu, Yu Hu, Longxing Yang, Zihao Sun, Jilin Mei, Jianchao Tan, and Chengru Song. PA&DA: Jointly sampling path and data for consistent nas. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11940–11949, 2023. 3

[28] Aleksander Mądry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2017. 1, 2, 3, 5

[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, 5, 6

[30] Yuwei Ou, Yuqi Feng, and Yanan Sun. Towards accurate and robust architectures via neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5967–5976, 2024. 1, 2

[31] 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. 2

[32] 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

[33] Yameng Peng, Andy Song, Haytham M Fayek, Vic Ciesielski, and Xiaojun Chang. SWAP-NAS: Sample-wise activation patterns for ultra-fast nas. In *The Twelfth International Conference on Learning Representations*, 2024. 3, 4, 6, 7, 8

[34] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *International Conference on Machine Learning*, pages 4095–4104. PMLR, 2018. 3

[35] Jialiang Sun, Wen Yao, Tingsong Jiang, Chao Li, and Xiaoqian Chen. A3D: A platform of searching for robust neural architectures and efficient adversarial attacks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 1

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

[37] Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in Neural Information Processing Systems*, 33:6377–6389, 2020. 8

[38] Runqi Wang, Linlin Yang, Hanlin Chen, Wei Wang, David Doermann, and Baochang Zhang. Anti-bandit for neural architecture search. *International Journal of Computer Vision*, 131(10):2682–2698, 2023. 2, 3

[39] Yite Wang, Dawei Li, and Ruoyu Sun. NTK-SAP: Improving neural network pruning by aligning training dynamics. In *International Conference on Learning Representations*, 2023. 3

[40] Yongtao Wu, Fanghui Liu, Carl-Johann Simon-Gabriel, Grigorios G Chrysos, and Volkan Cevher. Robust nas under adversarial training: benchmark, theory, and beyond. *The Twelfth International Conference on Learning Representations*, 2024. 5, 7

[41] Zhaohui Yang, Yunhe Wang, Xinghao Chen, Boxin Shi, Chao Xu, Chunjing Xu, Qi Tian, and Chang Xu. CARS: Continuous evolution for efficient neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1829–1838, 2020. 5

[42] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007. 5

[43] Yuhang Zhou and Zhongyun Hua. Defense without forgetting: Continual adversarial defense with anisotropic & isotropic pseudo replay. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24263–24272, 2024. 2

[44] Qingling Zhu, Yeming Yang, Songbai Liu, Qiuzhen Lin, and Kay Chen Tan. SCGAN: Sampling and clustering-based neural architecture search for GANs. *IEEE Transactions on Emerging Topics in Computational Intelligence*, pages 1–12, 2025. 5