International Journal of Computer Vision
https://doi.org/10.1007/s11263-024-02133-4

RNAS-CL: Robust Neural Architecture Search by Cross-Layer Knowledge Distillation

Utkarsh Nath[1] · Yancheng Wang[1] · Pavan Turaga[2] · Yingzhen Yang[1]

Received: 10 October 2023 / Accepted: 17 May 2024

Abstract
Deep Neural Networks are often vulnerable to adversarial attacks. Neural Architecture Search (NAS), one of the tools for developing novel deep neural architectures, demonstrates superior performance in prediction accuracy in various machine learning applications. However, the performance of a neural architecture discovered by NAS against adversarial attacks has not been sufficiently studied, especially under the regime of knowledge distillation. Given the presence of a robust teacher, we investigate if NAS would produce a robust neural architecture by inheriting robustness from the teacher. In this paper, we propose Robust Neural Architecture Search by Cross-Layer knowledge distillation (RNAS-CL), a novel NAS algorithm that improves the robustness of NAS by learning from a robust teacher through cross-layer knowledge distillation. Unlike previous knowledge distillation methods that encourage close student-teacher output only in the last layer, RNAS-CL automatically searches for the best teacher layer to supervise each student layer. Experimental results demonstrate the effectiveness of RNAS-CL and show that RNAS-CL produces compact and adversarially robust neural architectures. Our results point to new approaches for finding compact and robust neural architecture for many applications. The code of RNAS-CL is available at https://github.com/Statistical-Deep-Learning/RNAS-CL.

Keywords Adversarial attacks · Neural architecture search · Cross-layer knowledge distillation

1 Introduction

Neural Architecture Search (NAS) has become a highly regarded tool for driving new advancements in deep neural networks, improving state-of-the-art (SOTA) performance in various tasks, including computer vision and natural language processing. NAS has been attracted a lot of attention in recent years. NAS automatically searches for a neural architecture according to user-specified criteria without human intervention, thus avoiding the time-consuming and burdensome manual design of neural architectures. Earlier studies in NAS are based on Evolutionary Algorithms (EA) (Real et al., 2017) and Reinforcement Learning (RL) (Zoph and Le, 2017; Tan et al., 2019). However, despite their performance, they are computationally expensive. For instance, some these methods take order of 3000 GPU days to achieve state-of-the-art performance on the ImageNet dataset. Most recent studies (Liu et al., 2019; Cai et al., 2019; Wu et al., 2019; Wan et al., 2020; Nath et al., 2020) encode architectures as a weight-sharing supernet and optimize the weights using gradient descent. Architectures found by NAS exhibit two significant advantages. First, they achieve SOTA performance for various computer vision tasks. Second, the architectures found by NAS are efficient in terms of speed and size. Both advantages make NAS incredibly useful for real-world applications. However, most NAS methods are designed to optimize accuracy, parameters, or FLOPs. It is not clear how these architectures perform against adversarial attacks, which is an important dimension for deploying

Communicated by Zhouchen Lin.

✉ Yingzhen Yang
yingzhen.yang@asu.edu

Utkarsh Nath
unath@asu.edu

Yancheng Wang
ywan1053@asu.edu

Pavan Turaga
Pavan.Turaga@asu.edu

[1] School of Computing and Augmented Intelligence, Arizona State University, 699 S Mill Ave, Tempe, AZ 85281, USA

[2] School of Electrical, Computer and Energy Engineering, Arizona State University, 950 S Forest Mall, Tempe, AZ 85281, USA

Published online: 24 June 2024

secure and robust machine learning methods. Limited works (Yue et al., 2022; Ning et al., 2020; Xie et al., 2023) have studied NAS from the perspective of optimizing both adversarial accuracy and efficiency. In this paper, we propose RNAS-CL, a NAS method that jointly optimizes accuracy, latency, and robustness against adversarial attacks by inheriting robustness from a robust teacher.

Adversarial samples are constructed by modifying the inputs, for example, by adding small sophisticated perturbations to a clean image, such that the model misclassifies the given image. It has been established that nearly all deep neural networks are susceptible to adversarial attacks (Szegedy et al., 2014). Therefore, it is critical to analyze the robustness of models against adversarial attacks. Models that are robust to adversarial attack are crucial for high-stakes applications such as autonomous vehicles, health care, and physical security.

Adversarial training (Goodfellow et al., 2015; Madry et al., 2018; Kannan et al., 2018; Tramèr et al., 2018; Zhang et al., 2019) is a common approach to help create a more robust defense mechanism against adversarial attacks. In this case, models are trained on adversarial examples, which are often generated by fast gradient sign method (FGSM) (Goodfellow et al., 2015) or projected gradient descent (PGD) (Madry et al., 2018). Other types of defense mechanisms include models trained by loss functions or regularizations (Cissé et al., 2017; Hein & Andriushchenko, 2017; Yan et al., 2018; Pang et al., 2020), transforming inputs before feeding to model (Dziugaite et al., 2016; Guo et al., 2018; Xie et al., 2019), and using model ensemble (Kurakin et al., 2018; Liu et al., 2018).

Complementary to these methods, recent research (Madry et al., 2018; Guo et al., 2020; Su et al., 2018; Xie & Yuille, 2020; Huang et al., 2021) has found an intrinsic influence of network architecture on adversarial robustness. Motivated by these findings, we propose Robust Neural Architecture Search by Cross-Layer knowledge distillation (RNAS-CL). We use knowledge distilled from a robust teacher model to find a robust student architecture. Knowledge distillation transfers knowledge from a competent and complicated teacher model to a small student model. In standard knowledge distillation (Hinton et al., 2015), outputs from the teacher model are used as the "soft labels" to train the student model. However, apart from the final teacher outputs, intermediate layers can contain rich attention information. Different intermediate layers attend to different parts of the input object (Zagoruyko & Komodakis, 2017).

The central question of our investigation is: *can a robust teacher improve the robustness of the student model by providing information about where to look, that is, where to pay attention?* The proposed RNAS-CL method provides affirmative answers to the above question. In RNAS-CL, apart from learning from the output of the robust teacher model, each layer in the student learns "where to look" from the layers in the teacher model. Since the teacher and student often have different numbers of layers, a student layer should identify the potential best teacher layer that it should learn from. In RNAS-CL, apart from searching for the architecture of the student model, we search for the perfect tutor (teacher) layer for each student layer.

Furthermore, motivated by recent advances in self-supervised and semi-supervised learning that enforce consistency between predictions from different augmented views, we propose a novel Confidence-Aware Consistency loss, or CAC loss, that maximizes the prediction consistency between the adversarial view and the original view of input data. A wide range of adversarial training objectives, such as TRADES, are compatible with CAC. In our experimental section, we report that RNAS-CL significantly outperforms most existing models trained without adversarial training, in terms of robust accuracy on the CIFAR-10 dataset. Adversarially training RNAS-CL models with CAC and TRADES further significantly improves the robustness. RNAS-CL also renders promising result on the large-scale ImageNet dataset.

1.1 Contributions

Our contributions are presented as follows.

First, we propose RNAS-CL, a new approach which searches for a robust and efficient neural architecture that optimizes the tradeoff between robustness and prediction accuracy in a differentiable manner. To the best of our knowledge, RNAS-CL is the first work which shows that a student model can inherit robustness from a robust teacher model through cross-layer knowledge distillation and neural architecture search, and the student model can be trained potentially without robust training. The searched neural architecture can be adversarially trained using the loss function such as the loss in TRADES (Zhang et al., 2019) so as to further improve the adversarial robustness of the searched neural network. While there are existing works, such as AdvRush (Mok et al., 2021), which use neural architecture search to find adversarially robust neural architecture, RNAS-CL shows that a student model can inherit robustness from a robust teacher model by a novel cross-layer knowledge distillation method and enjoy better adversarial robustness than AdvRush (Mok et al., 2021). Leveraging the penalty on model size and inference cost, the neural architecture found by RNAS-CL is compact compared to competing NAS methods. We compare RNAS-CL with other computationally efficient and robust models (Sehwag et al., 2020; Ye et al., 2019; Gui et al., 2019; Goldblum et al., 2020; Dong et al., 2020; Huang et al., 2021). Compared to these models, similar-sized RNAS-CL models achieve significantly higher clean and PGD accuracy on the CIFAR-10 dataset, as shown in Fig. 1, and the ImageNet dataset.

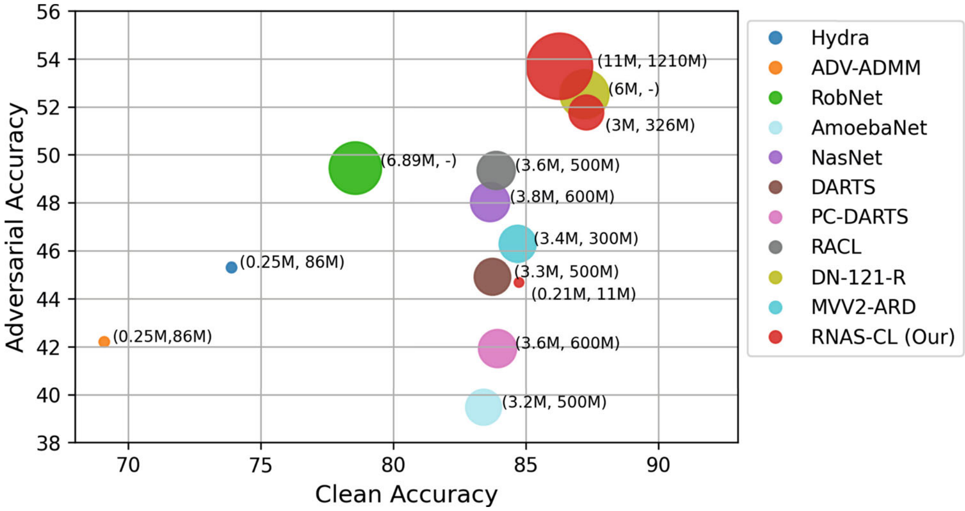


Fig. 1 The figure compares various SOTA efficient and robust methods on CIFAR-10. Clean Accuracy represents top-1 accuracy on clean images. Adversarial Accuracy represents top-1 accuracy on images perturbed by PGD attack. Larger marker size indicates larger architecture. The numbers in brackets represent the number of parameters and MACs, respectively

Second, our work advances the research of Knowledge Distillation (KD) using NAS. In particular, while conventional KD only uses fixed connections between teacher and student models to guide the student model, RNAS-CL extends the teaching scheme to learnable connections between layers of the teacher and the student models. Such observation is of independent interest and it potentially guides the design of the future adversarially robust NAS methods.

Our research also reveals an interesting observation that that there may be only a few layers in the teacher network that are more robust to adversarial attacks, which are referred to as the robust teacher layers and discussed in detail in Sect. 5.1. RNAS-CL identifies such robust teacher layers and uses these robust layers to teach the student network.

2 Related Work

2.1 Knowledge Distillation

Knowledge Distillation (KD) transfers knowledge from a large, cumbersome model to a small model. (Hinton et al., 2015) proposed the teacher-student model, where they use the soft targets from the teacher to train the student model. KD forces the student to generalize, similar to the teacher model. Since the work (Hinton et al., 2015) was proposed, numerous KD variants (Romero et al., 2015; Yim et al., 2017; Zagoruyko & Komodakis, 2017; Li et al., 2019; Tian et al., 2020; Sun et al., 2019) which are based on feature map, attention map, or contrastive learning have been proposed. (Romero et al., 2015) introduced intermediate-level hints from the teacher model to guide the student model training. (Romero et al., 2015) trained the student model in two stages. First, they trained the student model such that the student's middle layer predicts the output of the teacher's middle layer (hint layer). Next, they fine-tuned the pre-trained student model using the standard KD optimization function. Thanks to the intermediate hint, the student model achieved better performance with fewer parameters. Moving a step further, (Yim et al., 2017; Zagoruyko & Komodakis, 2017) and (Li et al., 2019) used information from multiple teacher layers to guide students' training. (Yim et al., 2017) computed Gramian matrix between the first and the last layer's output features to represent the flow of problem-solving. (Yim et al., 2017) transferred knowledge by minimizing the distance between student and teacher's flow matrix. (Li et al., 2019) calculated the inter-layered Gramian matrix and inter-class Gramian matrix to find the most representative layer and then minimized the distance between a few of the most representative student and teacher layers. (Zagoruyko & Komodakis, 2017) minimized the distance between teacher and student attention maps at the various block. (Li et al., 2020) distilled knowledge from teachers' blocks to supervise students' block-wise architecture search. In contrast with the above methods which map few teacher-student layers or blocks, we propose to map every student layer to a teacher layer. To this end, we propose RNAS-CL to search for the perfect tutor layer for each student layer. Similar to (Zagoruyko & Komodakis, 2017), we minimize the distance between mapped student-teacher attention maps.

2.2 Neural Architecture Search

Neural Architecture Search (NAS) is a technique that automatically designs neural architecture without human intervention. Given a search space, we can find the best architecture by training all architectures from scratch to convergence, however, it is computationally impractical. Earlier studies in NAS were based on RL (Zoph and Le, 2017; Tan et al., 2019) and EA (Real et al., 2017), but they still required lots of computation resources. Most recent studies

(Liu et al., 2019; Cai et al., 2019; Wu et al., 2019) encoded architectures as a weight-sharing super-network. Specifically, they trained an over-parameterized network containing all candidate paths. During training, they introduced weights corresponding to each path. These weights were optimized using gradient descent to select a single network in the end. The selected network is then trained in a standard fashion. Typically, NAS incurs substantial training time. Recent studies (Yang et al., 2020, 2021; Lian et al., 2019) have concentrated on devising efficient training schemes to mitigate this challenge. Some NAS approaches incorporate knowledge distillation into architecture search. For instance, (Peng et al., 2020) transfers knowledge between architectures without an external teacher, allowing its subnetworks to learn in a collaborative manner. The majority of these NAS studies concentrate on searching for CNN-based architectures. More recently, there has been a surge in NAS work addressing the search for transformer-based models, as seen in (Chen et al., 2021; Mo et al., 2022). Despite the state-of-the-art (SOTA) performance achieved by these methods across various classification tasks, their resilience against adversarial attacks remains uncertain. (Devaguptapu et al., 2021; Guo et al., 2020; Li et al., 2021; Madry et al., 2018; Su et al., 2018; Xie & Yuille, 2020; Huang et al., 2021) found an intrinsic influence of network architecture on adversarial robustness. (Devaguptapu et al., 2021) observed handcrafted architectures are more robust against adversarial attacks as compared to NAS models. Furthermore, they empirically observed that an increase in model size increased the robustness of the model against adversarial attacks. (Guo et al., 2020) discovered that densely connected architectures are more robust to adversarial attacks. As a result, they proposed a NAS method that conducts adversarial training on supernet and then selects the architecture with dense connections. (Li et al., 2021) dilated the backbone network to preserve its standard accuracy and then optimized the architecture and parameters using adversarial training. Despite its good performance, a major drawback is that adversarial training is highly time-consuming and decreases the performance on standard (clean) images. Our RNAS-CL can optimize robustness and prediction accuracy without adversarial training.

2.3 Efficient and Robust Models

The deep learning research community has extensively studied building efficient and adversarially robust models individually. However, few works combine both domains, that is, building an efficient model which is also adversarially robust. (Sehwag et al., 2020) proposed to make the pruning technique aware of the robust training objective. They formulate pruning as an empirical risk minimization (ERM) problem and integrate it with a robust training objective. (Huang et al., 2021) investigated the impact of network width and depth configurations on the robustness of adversarial-trained DNNs. They observed that reducing capacity at last blocks improves adversarial robustness. (Goldblum et al., 2020), proposed Adversarially Robust Distillation (ARD), where they encouraged student networks to mimic their teacher's output within an ϵ-ball of training samples. Furthermore, there are few NAS methods (Yue et al., 2022; Ning et al., 2020; Xie et al., 2023) that jointly optimize accuracy, latency, and robustness. (Ning et al., 2020) trained a multi-shot NAS method to search for adversarially robust architectures. They interpolate multiple one-shot methods to find architecture at the targeted capacity. (Xie et al., 2023; Yue et al., 2022) proposed a one-shot NAS method that selects an efficient model from the adversarially trained supernet. Compared to these methods, similar-sized RNAS-CL models achieve higher accuracy for both clean and and advarsarial images.

3 Robust Knowledge Distillation for Neural Architecture Search

We use knowledge distilled from a robust teacher model to search for a robust and efficient architecture. Knowledge distillation is the transfer of knowledge from a large teacher model to a small student model. In standard knowledge distillation, outputs from the teacher model are used as the “soft labels” to train the student model. However, apart from the final teacher outputs, intermediate features constitute important attention information. Different intermediate layers "attend" to different parts of the input object. In RNAS-CL, apart from learning from the teacher's soft labels, the student model learns where to pay attention among intermediate teacher layers. That is, each student layer is mapped to a robust teacher layer so that the attention maps of the student layer and the robust teacher layer are similar to each other. We define the attention maps in Sect. 3.1. We hypothesize that learning where to pay attention from a robust teacher would inherently make the student model more robust to adversarial attacks. RNAS-CL searches for a tutor layer for each student layer. In addition to increasing the robustness, we are also interested in searching for an efficient student architecture. In Sect. 3.2 and 3.3, we discuss our tutor and architecture search algorithm. Similar to other state-of-the-art NAS methods (Liu et al., 2019; Wu et al., 2019; Wan et al., 2020), RNAS-CL consists of the searching and the training phase. In the search phase, we optimize the neural architecture weights. In the training phase, we train the architecture sampled from the search phase in a standard fashion. In Sect. 3.4, we introduce our searching and training optimization objectives. While RNAS-CL can find the neural architecture which is already robust for the student, we use adversarial training to further increase the robustness of the student model. In Sect. 3.5, we propose a regularization term, Confidence-Aware Adversarial Consistency Loss (or CAC), which can be used with any adversarial objective, such as TRADES and FastAT (Wong et al., 2020), to increase the robustness of the student model.

3.1 Attention Map

We are interested in learning where to pay attention from a robust teacher model. Let us consider a convolution layer with the activation tensor $A \in \mathbb{R}^{C\times H\times W}$ where C is the number of channels, and H and W are spatial dimensions. We define a mapping function $\mathcal{F} : \mathbb{R}^{C\times H\times W} \longrightarrow \mathbb{R}^{H\times W}$ that takes A as input and outputs an attention map $\mathcal{F}(A) \in \mathbb{R}^{H\times W}$ by $[\mathcal{F}(A)]_{hw} = \sum_{c=1}^{C} A^2_{c,h,w}$, where $A_{c,h,w}$ represents the element of A with channel coordinate c and spatial coordinates h and w. We use the activation-based mapping function $\mathcal{F}$ as proposed in (Zagoruyko & Komodakis, 2017). The mapping function $\mathcal{F}$ is applied to the activation tensor after each convolution layer to generate an attention map. Several sample attention maps are illustrated in Fig. 2. RNAS-CL aims to find a teacher layer, referred to as a tutor, for each student layer such that the student layer's attention map is similar to that of its tutor in the teacher model. The student attention map may have different dimensions from that of its tutor. To address this issue, we interpolate all the attention maps to common dimensions.

3.2 Tutor Search

As described above, we aim to find a tutor (teacher layer) for each student layer, which teaches the student layer where to pay attention. Each student layer may choose any teacher layer as its tutor, resulting in an exponentially large search space. For example, the search space for a student model with 20 layers and a teacher model with 50 layers is of size 50^{20}. In order to make the tutor searching process computationally efficient, we employ Gumbel-Softmax (Jang et al., 2017) to search for the tutor for each student layer in a differentiable manner. Given network parameter $v = [v_1, \dots, v_n]$ and a constant τ, the Gumbel-Softmax function is defined as $g(v) = [g_1, \dots, g_n]$ where $g_i = \frac{\exp[(v_i+\epsilon_i)/\tau]}{\sum_i \exp[(v_i+\epsilon_i)/\tau]}$, $\epsilon_i \sim N(0, 1)$ is the uniform random noise which is also referred to as Gumbel noise. When $\tau \to 0$, Gumbel-Softmax tends to the arg max function. Gumbel-Softmax is a "re-parametrization trick", that can be regarded as a differentiable approximation to the argmax function.

Now consider a teacher T and student S model with n_t and n_s number of layers, respectively. A^i_t and A^i_s are the i-th activation tensors of teacher and student layers. In RNAS-CL, each student layer (i) is associated with n_t Gumbel weights (g_i) such that $g_i \in R^{1\times n_t}$. Let g_{ij} be the Gumbel weight associated with the i-th student and the j-th teacher layer. Then the attention loss is defined as

$$L_{\text{Attn}}(A_t, A_s) = \frac{1}{n_s n_t}\sum_{i=0}^{n_s}\sum_{j=0}^{n_t} g_{ij} \| \frac{\mathcal{F}(A^i_s)}{||\mathcal{F}(A^i_s)||_2} - \frac{\mathcal{F}(A^j_t)}{||\mathcal{F}(A^j_t)||_2} \|_2, \tag{1}$$

where A_s and A_t are the activation tensors of all the student and the teacher convolution layers. $\mathcal{F}$ is the mapping function defined in Sect. 3.1, $\|\cdot\|_2$ is the ℓ^2-norm. We exponentially decay the temperature τ of Gumbel-Softmax during the searching process, leading to an encoding close to a one-hot vector.

3.3 Architecture Search

Apart from searching for the tutor for each layer, we are interested in searching for an efficient architecture with low latency for the student model. Inspired by FBNetV2 (Wan et al., 2020), we search for the optimal number of filters, or the number of output channels, for each convolution block. Let $A = \{f_1, f_2, ..., f_n\}$ be the choices of filters and $\{z_1, z_2, ..., z_n\}$ be their corresponding outputs for a convolution block. Then the output of the convolution block is defined as $Z = \sum_{i=1}^{n} g^{(i)}_w z_i$, where $g^{(i)}_w$ is the Gumbel weight corresponding to i-th filter choice. Let FLOP(i) be the number of floating point operations for the i-th filter choice, then the number of FLOPs at the convolution block is $\sum_{i=1}^{n} g^{(i)}_w \text{FLOP}(i)$, which can be optimized in a differential manner using SGD. Similar to tutor search, temperature is exponentially decayed to obtain an encoding that is close to a one-hot vector, that is, only one of the Gumbel weights $\{g^{(i)}_w\}_{i=1}^{n}$ is close to 1 and the others are close to 0. The filter choice corresponding to the maximum Gumbel weight decides the number of filters of the convolution block in the searched architecture for the student model. Figure 12 in the appendix illustrates the architecture search process by FBNetV2.

3.4 RNAS-CL Loss

Following the convention of state-of-the-art NAS methods (Liu et al., 2019; Wu et al., 2019), RNAS-CL has the search and the training phases. In the search phase, we update the Gumbel weights and other model parameters at each epoch. Here the Gumbel weights refer to the Gumbel weights $\{g_{ij}\}$ for the student-teacher connection in (1) and the Gumbel weights $\left\{g^{(i)}_w\right\}$ for the filter choices described in Sect. 3.3. The weights are optimized using our RNAS-CL search loss to be described below.

RNAS-CL search loss Let y be the ground-truth one-hot encoded vector, p and q be output probabilities of the student and teacher network, and A_s, A_t be activation tensors for all student and teacher convolution layers. Then the RNAS-CL search loss function is defined as

$$\begin{aligned} L(y, p, q, A_t, A_s) =& (-y \log p + \text{KL}(p, q) \\ & + \gamma_s L_{\text{Attn}}(A_t, A_s)) n_f(G), \end{aligned} \tag{2}$$

where $\text{KL}(p, q) = \sum_i p_i \log \frac{p_i}{q_i}$ is the Kullback-Leibler (KL) divergence between two probability measures. L_{Attn} is the attention loss as defined in (1) and γ_s is a normalization constant. The latency penalty term $n_f(G)$ measures the latency of a neural network during the architecture search process, where $G = [g_w(1), \ldots, g_w(L)]$ are the Gumbel weights with $g_w(\ell)$ being the Gumbel weight vector for the ℓ-th layer. $n_f(G)$ is the Gumbel weighted FLOPs for the searched network, that is,

$$n_f(G) = \sum_{\ell=1}^{L} \sum_{i=1}^{m_i} g_w^{(i)}(\ell) \text{FLOP}(\ell, i), \tag{3}$$

where m_i denotes the total number of filter choices at the ℓ-th layer, $\text{FLOP}(\ell, i)$ is the number of floating point operations at the ℓ-th layer corresponding to the i-th filter choice. In this manner, larger architecture corresponding to higher FLOPs has a larger value of $n_f(G)$, so that the optimization of the search loss (2) encourages smaller and more compact neural architecture with less latency.

After the search phase, a tutor is selected as the j^* teacher layer with $j^* = \text{argmax}_j \, g_{ij}$ for each student layer i. In addition, the filter choices described in Sect. 3.3 for the student's neural architecture are decided as the ones corresponding to the maximum Gumbel weights for each convolution block. We then start the training phase, where the searched architecture is trained using the RNAS-CL training loss defined below.

RNAS-CL training loss. Let y be the ground-truth one-hot encoded vector, p and q be output probabilities of the student and teacher network, and A_t, A_s be the activation tensors of all the student and the teacher convolution layers. Then the RNAS-CL training loss function is

$$\begin{aligned} L(y, p, q, A_t, A_s) =& L_{\text{CE}}(y, p) + \text{KL}(p, q) \\ & + \gamma_t L_{\text{Attn}}(A_t, A_s), \end{aligned} \tag{4}$$

where $L_{\text{CE}}(y, p) = -y \log p$ is the cross-entropy loss, $\text{KL}(p, q)$ is the KL-divergence, γ_t is a normalization constant. Note that g_i in L_{Attn} is a one-hot vector. As a result, each student attention map is optimized with respect to a single tutor layer.

3.5 Confidence-Aware Adversarial Consistency Loss

Inspired by recent works in self-supervised learning (Zhai et al., 2019) and semi-supervised learning (Berthelot et al., 2019) that enforce consistency between predictions from different augmented views, we propose a consistency loss that maximizes the prediction consistency between the adversarial view and original view of input data. The optimization is only performed on samples that have high confidence in the prediction by the adversarial view. For an input image x, we first generate its adversarial view x_{adv} and obtain the predictions of x and x_{adv} with the student network as p and p_{adv}. Next, we take the average of p and p_{adv} as $\bar{p} = \frac{p+p_{adv}}{2}$. Then we sharpen the average prediction $\bar{p}$ by $\tilde{p}_j = \bar{p}_j^{\frac{1}{\tau}} / \sum_{k=1}^{K} \bar{p}_k^{\frac{1}{\tau}}$ where K is the number of classes, $\tilde{p}_j$ is the j-th element of $\tilde{p}$. $\tau \in (0, 1]$ is the sharpening factor. $\tilde{p}$ is close to one-hot distribution with small τ. The sharpened $\tilde{p}$ is regarded as a pseudo label for x based on the predictions by both x and x_{adv}. We aim to enforce consistency between p and p_{adv} by minimizing their distances to $\tilde{p}$. Therefore, the confidence-aware adversarial consistency loss is defined as

$$L_{\text{CAC}}(x) = \mathbb{1}(\max(\bar{p}) \geq \gamma) \left(\text{KL}(\tilde{p}, p) + \text{KL}(\tilde{p}, p_{adv})\right), \tag{5}$$

where $\mathbb{1}(\cdot)$ is an indicator function, $\gamma \in [0, 1)$ is the confidence threshold. In L_{CAC}, the consistency between the prediction of an image and its adversarial view will be optimized only when the maximum element of its prediction $\bar{p}$ is not less than γ, indicating that L_{CAC} imposes consistency only on images with confident predictions. The optimization of L_{CAC} reduces the negative impact of the noisy adversarial view and improves the robustness of the student network. We adversarially train our model with L_{CAC} and existing adversarial objectives such as TRADES (Zhang et al., 2019) and FastAT (Wong et al., 2020). The training loss for adversarial training with TRADES and L_{CAC} is defined by

$$L_{\text{ADV}} = L_{\text{CAC}} + L_{\text{TRADES}} + L_{\text{KL}} + \gamma_t L_{\text{Attn}}, \tag{6}$$

where L_{TRADES} is TRADES optimization objective and L_{KL}, γ_t, L_{Attn} are the same as those in (4).

4 Experiments

In this section, we conduct experiments on real-world datasets to show the effectiveness of the proposed framework. The experiments section is organized as follows. In Sect. 4.1, we discuss our experimental setup and implementation details. In Sect. 4.2, we compare models trained by RNAS-CL against state-of-the-art efficient and robust models on CIFAR-10. In Sect. 4.3, we compare RNAS-CL against various knowledge distillation methods. In Sect. 4.4, we compare RNAS-CL against stronger attacks such as such as CW_∞ (Carlini & Wagner, 2017) and AutoAttack (Croce & Hein, 2020). In Sect. 4.5 and 4.7, we compare RNAS-CL models trained on ImageNet and ImageNet-100 datasets. In Sect. 4.9, we empirically show the effectiveness of cross-connections and Confidence-Aware Adversarial Consistency Loss (CAC) in improving the adversarial robustness of the model.

4.1 Implementation Details

In this paper, we evaluate RNAS-CL on three public benchmarks for image classification: (1) CIFAR-10, a collection of $60k$ images in 10 classes (Krizhevsky, 2009); (2) ImageNet, an image classification dataset (Russakovsky et al., 2015) with 1000 classes and about 1.2M images; (3) ImageNet-100, a subset of ImageNet-1k dataset (Russakovsky et al., 2015) with 100 classes and about $130k$ images (Tian et al., 2020). We use standard data augmentation techniques for each dataset, such as random-resize cropping and random flipping. On each dataset, we first perform the searching step. We train our model using RNAS-CL search loss (2), and we search for the channel number and the connected teacher layer for each student layer. We conduct experiments with different search spaces and various robust teacher models. In this section, we refer to our model by RNAS-CL-X-T where X represents our search space, and T represents the robust teacher model. Detailed search space is provided in Table 13 and Table 14. We use 4 robust teacher model, ResNet-50, ResNet-18, WideResNet-50, and WideResNet-34, which are referred to as R-50, R-18, WRT-50, and WRT-34 respectively. For example, RNAS-CL-S3-R-18 represents a model trained in the S3 search space using ResNet-18 as the adversarially robust teacher model.

We use the SGD optimizer for all the three datasets. The default values of momentum and weight decay are set to 0.9 and 4×10^{-5} for ImageNet and ImageNet-100. The batch size is 256. The learning rate is initialized as 0.1 and annealed down to zero following a cosine schedule. After the search stage which takes 100 epochs, the searched architecture is trained from scratch using RNAS-CL training loss (4) for 200 epochs. For CIFAR-10, default values of momentum and weight decay are set to 0.9 and 2×10^{-4}. The batch size is 128. We train our model for 100 epochs in both the searching and training phases. The learning rate is initialized as 0.1 and reduced by a factor of 10 after the 75-th and the 90-th epoch. Following the settings of FBNetV2, the temperature (τ) in Gumbel-Softmax is initialized as 5.0 and exponentially annealed by $e^{-0.045}$ at every epoch in the search phase. The hyper-parameters λ_s and λ_t are set to 1.0 for all experiments. In the search phase, we use 80% of the data in each batch to optimize the model weights and the remaining 20% data to optimize the architecture weights, which are Gumbel weights described in Sect. 3.4. For robustness evaluation, we choose five powerful attacks, including FGSM (Goodfellow et al., 2015), MI-FGSM (Dong et al., 2018), PGD (Madry et al., 2018), CW (Carlini & Wagner, 2017), and AutoAttack (Croce & Hein, 2020). To be consistent with the adversarial literature (Madry et al., 2018; Zhang et al., 2019), the adversarial perturbation is considered under the ℓ_∞ norm with a total perturbation scale of $8/255(= 0.031)$.

4.2 Comparison with Efficient and Robust CIFAR-10 Models

In this section, we compare the robustness of our method against other SOTA efficient and robust models. In Table 1, we compare RNAS-CL to both efficient models trained with and without adversarial training. All RNAS-CL models are trained with robust WideResNet-34 (Rice et al., 2020) as the teacher model. It can be observed from Table 1 that RNAS-CL significantly outperforms all models trained without adversarial training in terms of adversarial accuracy. While being significantly smaller, our RNAS-CL models achieve significantly higher adversarial accuracy when compared to models trained without adversarial training. For example, RNAS-CL-S7-WRT-34 achieves more than 28% higher PGD accuracy compared to most of the other similar-sized models.

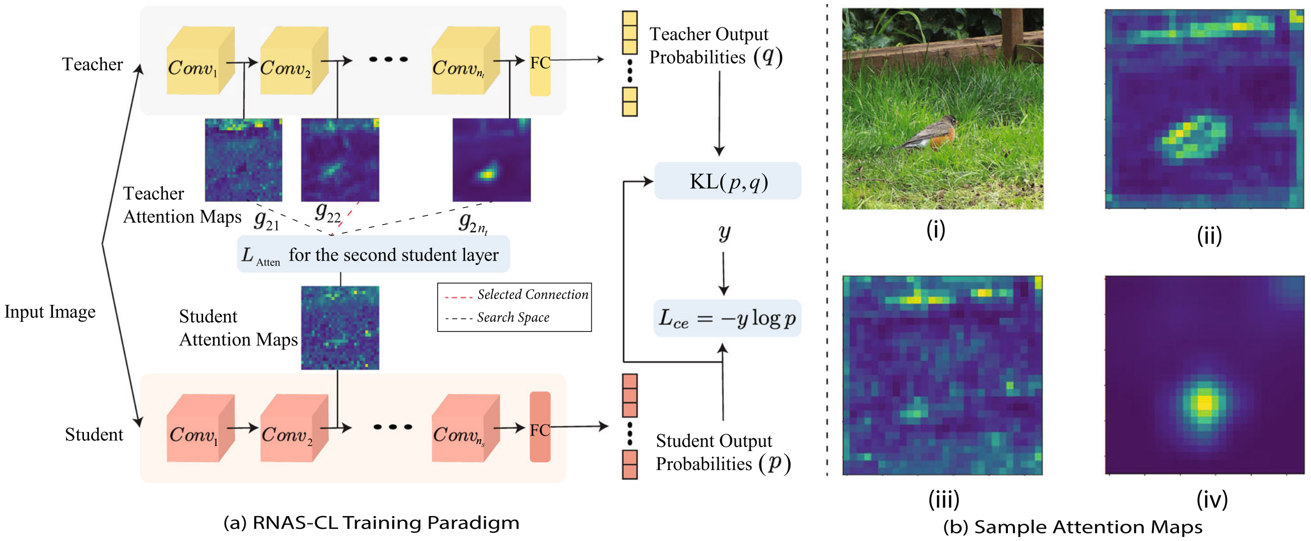
Next, we compare RNAS-CL against adversarially trained robust models. For a fair comparison, after the training stage, we further train our RNAS-CL models with our adversarial training loss (6) for 20 epochs. Adversarially training RNAS-CL models improves their adversarial accuracy. RNAS-CL models achieve similar or higher adversarial accuracy compared to other adversarially trained models. Furthermore, RNAS-CL models are much smaller and achieve significantly higher clean accuracy. For example, in Table 1, RNAS-CL-M-WRT-34 achieves similar or higher adversarial accuracy than most other methods while being smaller and significantly exceeding in terms of clean accuracy. We also obtain much smaller models using RNAS-CL. The Tiny RNAS-CL model, RNAS-CL-S5-WRT-34, exceed its counterpart Hydra ResNet 34 (Sehwag et al., 2020) by more than $\sim 12\%$ in terms of clean accuracy with the same model size. Similar results can also be visualized in Fig. 1. In Fig. 1, RNAS-CL models are on the top right corner of the plot, representing the models with the highest clean and adversarial accuracy. Results for RNAS-CL models trained with different robust teachers have been discussed in Sect. 5.2.

Comparison against various perturbation budgets. To further illustrate the effectiveness of RNAS-CL, we compare RNAS-CL with previously proposed defense mechanisms against various perturbation budgets. In Fig. 5, we compare various methods against PGD and FGSM attacks. For both

Table 1 The table shows the performance of various efficient and robust methods on the CIFAR-10 dataset

Method	Clean Acc	FGSM	PGD^{20}	MI-FGSM	Params (M)	MACs (M)
Without adversarial training						
DARTS (Liu et al., 2019)	97.03	42.48	7.09	0.28	3.3	500*
PC-DARTS (Xu et al., 2020)	97.05	49.18	9.84	1.21	3.6	600*
RACL (Dong et al., 2020)	97.44	50.53	1.93	4.68	3.6	500*
AmoebaNet (Real et al., 2019)	97.39	44.79	0.25	0.80	3.2	500*
NasNet (Zoph et al., 2018)	97.37	47.53	0.42	1.01	3.8	600*
MVV2-ARD (Goldblum et al., 2020)	76.13	–	38.21	–	3.4	300
E2RNAS-C16 (Yue et al., 2022)	93.97	–	6.76	–	0.44	–
RNAS-CL-S3-WRT-34 (our)	**89.4**	44.95	**34.3**	38.92	**0.11**	**6.64**
RNAS-CL-S5-WRT-34 (our)	90.4	46.72	35.59	40.57	0.21	11.02
RNAS-CL-S7-WRT-34 (our)	90.62	48.93	37.24	42.27	0.32	15.58
RNAS-CL-M-WRT-34 (our)	92.46	50.51	39.84	44.54	3	326
RNAS-CL-L-WRT-34 (our)	**92.6**	**52.37**	**41.9**	**46.66**	11	1210
With adversarial training						
Hydra ResNet 18 (Sehwag et al., 2020)	69	–	41.6	–	0.11	37.63
Hydra ResNet 34 (Sehwag et al., 2020)	71.8	–	44.4	–	0.21	75.43
Hydra ResNet 50 (Sehwag et al., 2020)	73.9	–	45.3	–	0.25	85.92
ADV-ADMM ResNet 18 (Ye et al., 2019)	58.7	–	36.1	–	0.11	37.63
ADV-ADMM ResNet 34 (Ye et al., 2019)	68.8	–	41.5	–	0.21	75.43
ADV-ADMM ResNet 50 (Ye et al., 2019)	69.1	–	42.2	–	0.25	85.92
RobNet-Small (Guo et al., 2020)	78.05	53.93	48.32	48.98	4.41	–
RobNet-Medium (Guo et al., 2020)	78.33	54.55	49.13	49.34	5.66	–
RobNet-Large (Guo et al., 2020)	78.57	54.98	49.44	49.92	6.89	–
AmoebaNet (Real et al., 2019)	83.41	56.40	39.47	47.60	3.2	500*
NasNet (Zoph et al., 2018)	83.66	55.67	48.02	53.05	3.8	600*
DARTS (Liu et al., 2019)	83.75	55.75	44.91	51.63	3.3	500*
PC-DARTS (Xu et al., 2020)	83.94	52.67	41.92	49.09	3.6	600*
RACL (Dong et al., 2020)	83.89	57.44	49.34	54.73	3.6	500*
VGG-11-R (Huang et al., 2021)	79.63	57.35	43.93	–	5.83	–
DN-121-R (Huang et al., 2021)	87.22	**67.12**	52.52	–	6	–
DARTS-R (Huang et al., 2021)	87.2	66.74	52.36	–	2.53	–
MVV2-ARD (Goldblum et al., 2020)	84.70	–	46.28	–	3.4	300
MSRobNet-1000 (Ning et al., 2020)	84.5	59.6	52.7	–	3.16	–
MSRobNet-2000 (Ning et al., 2020)	85.7	60.6	53.6	–	6.46	–
$S^{2}_{8/255}$ (Xie et al., 2023)	76.54	–	31.83	–	1.68	–
AdvRush (Mok et al., 2021)	87.30	60.87	53.07	–	4.2	659
RNAS-CL-S3-WRT-34 (ours)	83.11	50.67	43.41	43.98	**0.11**	**6.64**
RNAS-CL-S5-WRT-34 (ours)	84.81	51.99	45.34	46.3	0.21	11.02
RNAS-CL-S7-WRT-34 (ours)	85.06	49.11	43.88	45.53	0.32	15.58
RNAS-CL-M-WRT-34 (ours)	**87.29**	59.71	51.76	53.43	3	326
RNAS-CL-M2-WRT-34 (ours)	87.17	60.98	53.14	54.64	5.6	604
RNAS-CL-L-WRT-34 (ours)	86.28	61.12	**53.69**	**55.07**	11	1210

Clean Acc represents top-1 accuracy on clean images. FGSM, PGD^{20}, MI-FGSM represent top-1 accuracies on images perturbed by the corresponding attacks. PGD^{20} represents 20 step PGD attack. $*$ represents approximate values. Columns with unreported values are represented by '–'



(a) RNAS-CL Training Paradigm

(b) Sample Attention Maps

Fig. 2 **a** Training paradigm based on RNAS-CL. We connect attention maps from each student layer to each robust teacher layer. For each student layer, we search for the optimum teacher layer. g_{ij} represents the Gumbel weight associated with the i-th student layer and the jth teacher layer. RNAS-CL induces robustness of the student model by searching for the optimum teacher layer for each student layer. We also search for the number of filters in each convolution block of the student model to build an efficient model inspired by FBNetV2 (Wan et al., 2020). **b** Sample attention maps corresponding to the input image (i) from low-level (ii), mid-level (iii), and high-level (iv) convolution layers

attacks, RNAS-CL outperforms its counterparts at all perturbations. RNAS-CL significantly outperforms other methods as perturbation size increases. For $\epsilon = 0.1$, RNAS-CL exceeds other methods by ∼20% for both PGD and FGSM attacks.

4.3 Comparison Against KD Variants

In this section, we compare our methods against various knowledge distillation methods (Park et al., 2019; Ahn et al., 2019; Tung & Mori, 2019; Tian et al., 2020; Passalis & Tefas, 2018). We use Robust WRT-34 as the teacher model for all KD methods and train three different student architectures: RNAS-CL-S3, RNAS-CL-S5, and RNAS-CL-S7. In Fig. 3, models trained using our paradigm are explicitly on the upper right-most part of the graph, demonstrating the effectiveness of intermediate cross-connections. RNAS-CL-S3 architecture trained using RKD (Park et al., 2019) performs similarly to the model trained using our method. Apart from this, all models trained using RNAS-CL significantly outperform all other methods in terms of clean and adversarial accuracy.

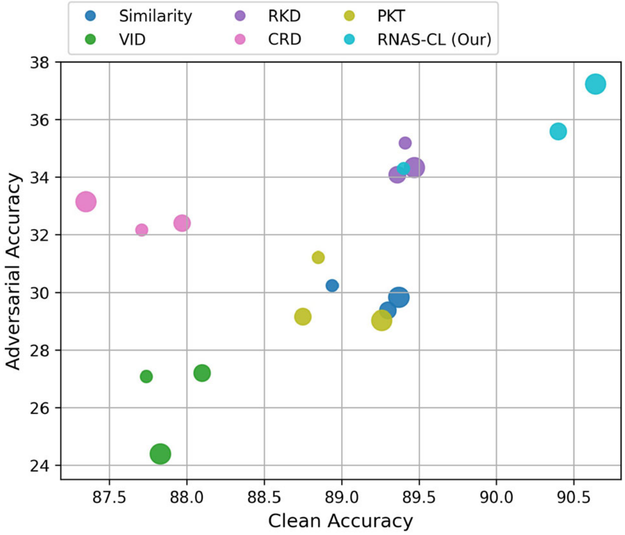


Fig. 3 The figure compares various knowledge distillation variants (Similarity (Tung & Mori, 2019), VID (Ahn et al., 2019), RKD (Park et al., 2019), CRD (Tian et al., 2020), PKT (Passalis & Tefas, 2018)) against RNAS-CL on the CIFAR-10 dataset. Adversarial Accuracy represents top-1 Accuracy on images perturbed by 20 step PGD attack. Clean Accuracy represents top-1 Accuracy on clean images. Larger marker size indicates larger architecture. For each method, RNAS-CL-S3, RNAS-CL-S5, and RNAS-CL-S7 are represented by increasing marker size

4.4 Compare CIFAR-10 Model Against CW and AutoAttack

In this section, we compare RNAS-CL and (Huang et al., 2021) against recent attacks including CW_∞ (Carlini & Wagner, 2017) and AutoAttack (Croce & Hein, 2020) on the CIFAR-10 dataset. CW attacks were proposed to defeat defensive distillation. In Table 2, we use the ℓ_∞ version of the CW attack optimized by PGD with maximum perturba-

Table 2 Comparison between the performance of (Huang et al., 2021) and RNAS-CL against CW_∞ (Carlini & Wagner, 2017) and AutoAttack (Croce & Hein, 2020) on the CIFAR-10 dataset

Method	CW_∞	AA
VGG-R (Huang et al., 2021)	46.49	38.44
DN-121-R (Huang et al., 2021)	53.07	47.75
RNAS-CL-S3-WRT-34 (our)	47.07	37.17
RNAS-CL-S5-WRT-34 (our)	48.33	39.28
RNAS-CL-S7-WRT-34 (our)	47.91	38.36
RNAS-CL-M-WRT-34 (our)	**53.52**	46.89
RNAS-CL-L-WRT-34 (our)	52.63	**48.49**

tion budget set to $\epsilon = 8/255$. AutoAttack is a parameter-free ensemble attack currently considered to be one of the most reliable and widely acknowledged evaluation benchmark in Adversarial Defences.

4.5 Results for ImageNet

In this section, we compare our model against the SOTA compact and efficient methods (Huang et al., 2021; Guo et al., 2020) on the ImageNet (ILSVRC-12) dataset (Russakovsky et al., 2015). The searching phase of RNAS-CL model is conducted on the ImageNet-100 dataset. The searched model is then adversarially trained on the full ImageNet dataset using FastAT (Wong et al., 2020) or Free Training (Shafahi et al., 2019). We compare RNAS-CL to the robust methods in (Huang et al., 2021; Guo et al., 2020) against 10 step PGD attack with $\epsilon = 4/255$ on the ImageNet dataset. The robust models in (Huang et al., 2021) are adversarially trained using FastAT whereas (Guo et al., 2020) employs Free Training (Shafahi et al., 2019). In the training phase of RNAS-CL, we train the models using FastAT/Free Training and CAC to enhance their robustness. As shown in Table 3, the RNAS-CL models significantly outperform (Huang et al., 2021) in terms of clean accuracy, robust accuracy, and the number of parameters. Although RNAS-CL achieves lower PGD accuracy compared to RobNet, it excels in clean accuracy with a 30% smaller model. Furthermore, RNAS-CL exhibits notably reduced training times in comparison to RobNet models, as indicated in Table 6.

4.6 Confidence-Aware Adversarial Consistency Loss (CAC)

In this section, we conduct an ablation study on the choice of the objective function used during adversarial training of RNAS-CL models on the CIFAR-10 dataset. For all three models, we observe that models trained using CAC have higher robust accuracy than models trained using TRADES as shown in Table 4. However, models trained using TRADES have higher clean accuracy. Therefore, we train RNAS-CL models with both TRADES and CAC to get a better trade-off. Models trained using both objective functions have higher clean accuracy than CAC and higher robust accuracy than TRADES. For example, the RNAS-CL-S3 model trained using both objective functions has 1.08% higher clean accuracy compared to CAC and 0.34% higher robust accuracy compared to TRADES. We see similar results for all other models. Thus, throughout the paper, we used both CAC and TRADES for adversarially training our RNAS-CL models unless otherwise mentioned.

4.7 Compare Efficient and Robust ImageNet-100 Models

We compare RNAS-CL to adversarially robust pruning methods on the ImageNet-100 dataset, with results shown in Table 5. We train RNAS-CL models with three robust teachers, ResNet-18, ResNet-50, and WideResNet-50 (Engstrom et al., 2019). RNAS-CL models consistently exceed other models by a large margin in terms of clean accuracy while exhibiting similar or higher adversarial robustness. In Table 5, Hydra and LWM are adversarially trained using TRADES (Zhang et al., 2019). For a fair comparison, we retrain our RNAS-CL models with the TRADES optimization objective after the regular training. We replace the cross-entropy term in (4) with the TRADES optimization objective. With such training, RNAS-CL achieves similar or higher adversarial accuracy while significantly outperforming Hydra and LWM in clean accuracy with only a fraction of MACs.

We further study adversarial accuracy at various perturbation budgets for three different teacher models. As illustrated in Fig. 4, RNAS-CL exceeds its counterpart in adversarial accuracy at various perturbation budgets for all teacher models on the ImageNet-100 dataset. This demonstrates the significance of cross-layer connections in RNAS-CL.

4.8 Training Time Comparison

We compare the training time between our proposed RNAS-CL and previous state-of-the-art robust and efficient methods. As shown in Table 6, RNAS-CL achieves higher adversarial accuracy than competing methods with significantly less training time than most baselines. Compared with DN-121-R, RNAS-CL-M2-WRT-34 takes a slightly longer training time. However, it exceeds DN-121-R in terms of adversarial accuracy and exhibits fewer parameters and MACs.

4.9 Ablation Study

This ablation study demonstrates the significance of student-teacher cross-layer connections in RNAS-CL. We compare three types of training paradigms. In the first training

Table 3 Performance of various efficient and robust methods on the ImageNet dataset

Method	Objective	Clean	PGD^{10}	Params (M)	GFLOPs
ResNet-50-R (Huang et al., 2021)	FastAT	56.63	31.14	25.5	4*
RobNet-large (Guo et al., 2020)	Free Training	61.26	37.17	12.76	–
RNAS-CL-IL-WRT-50	FastAT	61.7	32.5	8.5	0.35
RNAS-CL-IL-WRT-50	FastAT + CAC	61.5	33.5	8.5	0.35

Clean and PGD are the same as that in Table 1. ∗ represents approximate values

Table 4 Ablation study on the objective function used during adversarial training on the CIFAR-10 dataset

Method	Objective function	Clean	PGD^{20}
RNAS-CL-S3	TRADES	83.45	43.07
	CAC	82.03	43.53
	TRADES + CAC	83.11	43.41
RNAS-CL-S5	TRADES	84.75	44.68
	CAC	–	–
	TRADES + CAC	84.81	45.34
RNAS-CL-S7	TRADES	85.81	43.24
	CAC	82.35	44.07
	TRADES + CAC	85.06	43.88

Clean and PGD are the same as that in Table 1

paradigm, we conduct searching and training using cross-entropy loss without any teacher model. We refer to this as standard. In the second paradigm, we conduct searching and training by minimizing the cross-entropy loss and standard KL Divergence with a robust teacher model. We refer to the corresponding models as KL-X-T, where X represents the search space, and T represents the robust teacher model. Finally, the third model type is RNAS-CL, where we include all three terms, cross-entropy loss, KL Divergence, and cross-layer student-teacher connections.

In Fig. 5A, we compare the attention maps from student models trained using RNAS-CL-I-R-50 against students trained using KL-I-R-50. We compare attention maps for various convolution layers at regular intervals. As expected, adding cross-layer connections obtains attention maps from the student model closer to the teacher model. Each student layer learns where to pay attention from its connected teacher layer. For example, in column (b), the KL-I-R-50 layer attends to various parts of the image, whereas the RNAS-CL layer learning from the 28-th teacher layer pays more attention to the informative central part of the image. Similarly, in column (c), the RNAS-CL layer learns from

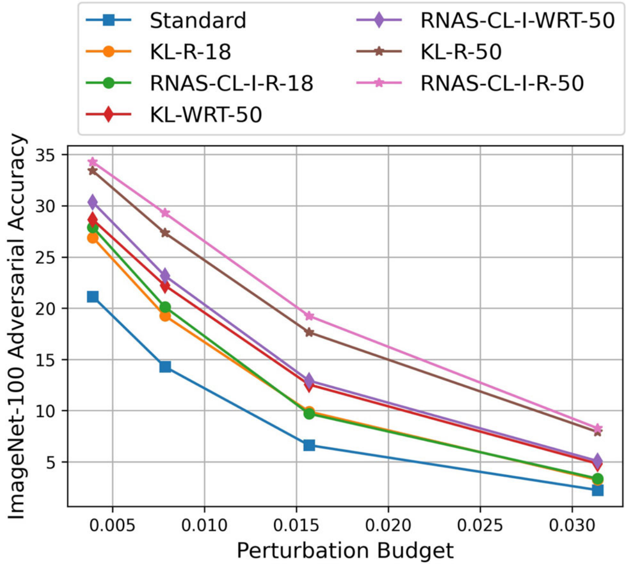


Fig. 4 Illustration of the adversarial accuracy of various models at various perturbation budgets on the ImageNet-100 dataset. Please refer to Sect. 4.9 for the definition of models named “Standard” and models named with the format of “KL-X-T”

Table 5 Performance of various efficient and robust methods on the ImageNet-100 dataset

Method	Clean	PGD^{20}	# Params (M)	MACs (M)
Hydra (ResNet-18)—90% (Sehwag et al., 2020)	59.96	29.79	**1.1**	1200
LWM (ResNet-18)—90% (Han et al., 2015)	59.02	27.67	1.1	1200
RNAS-CL-I-R-18	85.22	3.36	3.94	**241.98**
RNAS-CL-I-R-50	**85.98**	8.3	3.96	244.76
RNAS-CL-I-WRT-50	85.46	5.08	4.01	255.37
RNAS-CL-I-R-18 + TRADES	78.94	28.06	3.94	**241.98**
RNAS-CL-I-R-50 + TRADES	79.95	**32.44**	3.96	244.76
RNAS-CL-I-WRT-50 + TRADES	79.42	29.02	4.01	255.37

Clean and PGD are the same as that in Table 1. All MACs are calculated without special hardware (Han et al., 2016) or special software (Park et al., 2017)

Table 6 Training time (in GPU hours) comparison on the CIFAR-10 dataset

Method	Clean	PGD^{20}	# Params (M)	MACs (M)	Training Time (h)
RoboNet-Small (Guo et al., 2020)	78.05	48.32	4.4	1260	130
RACL (Dong et al., 2020)	83.89	49.34	3.3	500	44
MSRobNet-1000 (Ning et al., 2020)	84.5	52.7	3.16	1018	48.7
MVV2-ARD (Goldblum et al., 2020)	84.7	46.28	3.4	300	41.1
DN-121-R (Huang et al., 2021)	87.22	52.52	6	900	16.6
AdvRush (Mok et al., 2021)	87.30	53.07	4.2	659	71.74
RNAS-CL-M2-WRT-34	87.17	53.14	5.6	604	22.5
RNAS-CL-L-WRT-34	86.28	53.69	11	1210	28.7

We calculate the training time on a single Tesla V100 card with 16 G memory

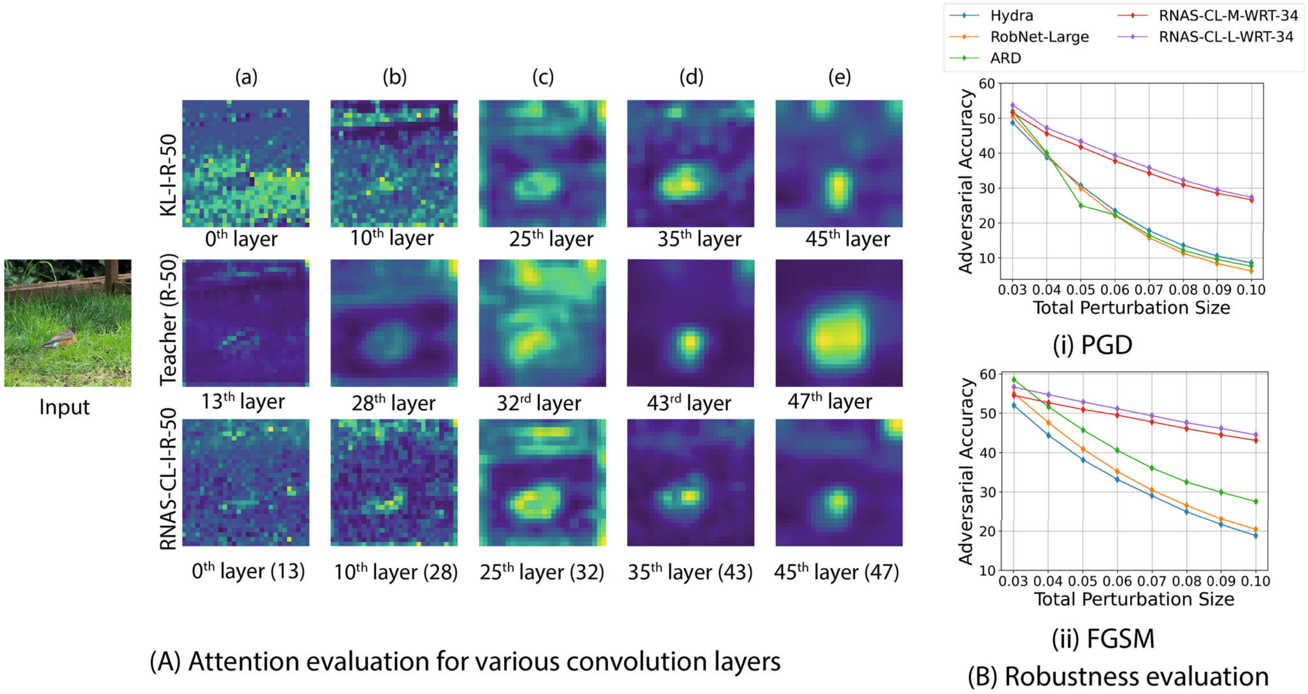


Fig. 5 **A** KL-I-R-50 represents attention maps from a model trained using cross-entropy loss and knowledge distillation without any cross-layer connections. Teacher and RNAS-CL represent the attention maps from the robust teacher (ResNet-50) and the RNAS-CL model. The name for each RNAS-CL layer includes its connected teacher layer. For example, in the 0th layer (13), 13 represents the corresponding teacher layer. RNAS-CL drives the attention maps of the student layers closer to that of their corresponding teacher layers. **B** Robustness evaluation under different perturbation sizes for the PGD and FGSM attacks on the CIFAR-10 dataset

the teacher model and pays more attention to the central and upper portions of the image. In Table 7, we compare the performance of various components of RNAS-CL. We observe that under both training schemes, KL and ICC (Intermediate Cross-Connections) significantly increase the robustness compared to the standard network. Finally, combining KL and ICC, that is, RNAS-CL, outperforms its counterparts. In Fig. 6, we compare RNAS-CL models to KL-X-T and standard models for PGD attacks at various perturbation budgets on the CIFAR-10 dataset.

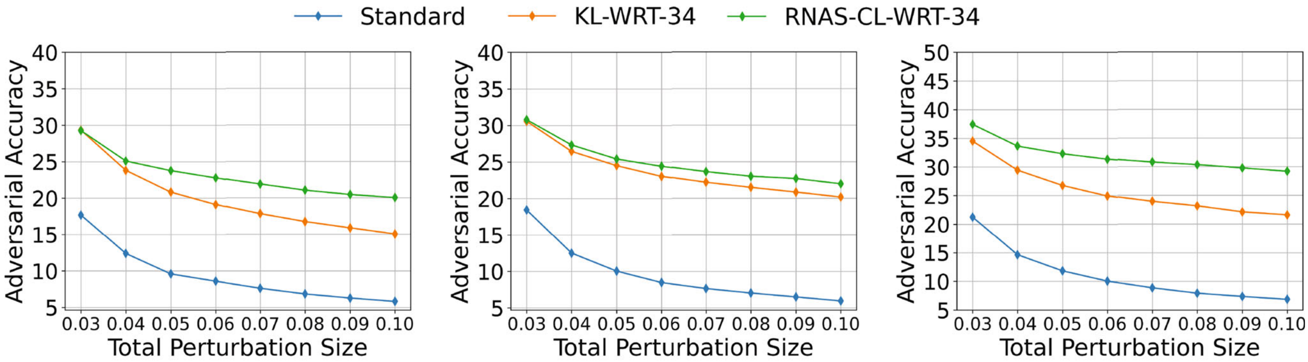
4.10 Comparison Against Layerwise KL-Divergence

In this paper, we use the attention loss (1) to search for the tutor layer for each student layer. It is interesting to investigate different attention losses, such as the KL-divergence. We replace the attention loss in (4) with the layer-wise KL-divergence between the feature maps of the tutor layers in the teacher model and the student layers in the student model, with the results reported in Table 8. We observe that there is a trade-off between clean accuracy and robust accuracy. Compared to layer-wise KL-divergence, the original atten-

Table 7 Ablation study on various components used during RNAS-CL training on the CIFAR-10 dataset with RNAS-CL-S7-WRT-34 as the base model

Training type	Objective function	Clean	PGD^{20}
Without adversarial training	CE	90.98	19.3
	CE + KL	90.76	36.3
	CE + ICC	90.33	35.54
	CE + KL + ICC	90.62	37.24
With adversarial training	CE	80.85	39.67
	CE + KL	85.07	41.63
	CE + ICC	82.45	41.03
	CE + KL + ICC	85.06	43.88

CE represents models trained using Cross-Entropy Loss. CE + KL represents models trained by minimizing the Cross-Entropy loss and standard KL Divergence with a robust teacher model. CE + ICC represents models trained by minimizing the Cross-Entropy loss and Intermediate Cross-Connections (ICC). Clean and PGD are the same as that in Table 1



(a) RNAS-CL-S3 (b) RNAS-CL-S5 (c) RNAS-CL-L

Fig. 6 Adversarial accuracy of various models at various perturbation budgets on the CIFAR-10 dataset

Table 8 Ablation study on the attention loss on the CIFAR-10 dataset

Method	Objective function	Clean	PGD^{20}
RNAS-CL-S3	Attention loss (our)	89.4	34.3
	Layer-wise KL-Div	90.21	33.05
RNAS-CL-S5	Attention loss (our)	90.4	35.59
	Layer-wise KL-Div	90.77	35.39
RNAS-CL-S7	Attention loss (our)	90.64	37.24
	Layer-wise KL-Div	90.97	36.22

Table 9 Ablation study on using various distillation methods, including PKT (Passalis & Tefas, 2018) and RKD (Park et al., 2019), in RNAS-CL on the CIFAR-10 dataset

Method	Distillation method	Clean	PGD^{20}
RNAS-CL-S3	Standard	89.4	34.3
	PKT	89.92	34.61
	RKD	89.79	36.25
RNAS-CL-S5	Standard	90.4	35.59
	PKT	90.65	35.61
	RKD	90.25	36.97
RNAS-CL-S7	Standard	90.64	37.24
	PKT	91.13	36.63
	RKD	91.17	37.58

tion loss (1) often achieves lower clean accuracy but higher robust accuracy.

4.11 RNAS-CL with Other Knowledge Distillation Methods

In this section, we conduct experiments using two different knowledge distillation methods in RNAS-CL, including RKD (Park et al., 2019) and PKT (Passalis & Tefas, 2018), with the results reported in Table 9. RKD minimizes the structural relationship between the outputs, emphasizing the relationship rather than the distance between individual outputs. On the other hand, PKT matches the probability distribution of the data in the feature space, rather than aligning with the actual representation. We substitute the KL term in the RNAS-CL search loss function (2) and training loss function (4) with the KL objectives in RKD and PKT respectively. We observe that using RKD or PKT distillation in

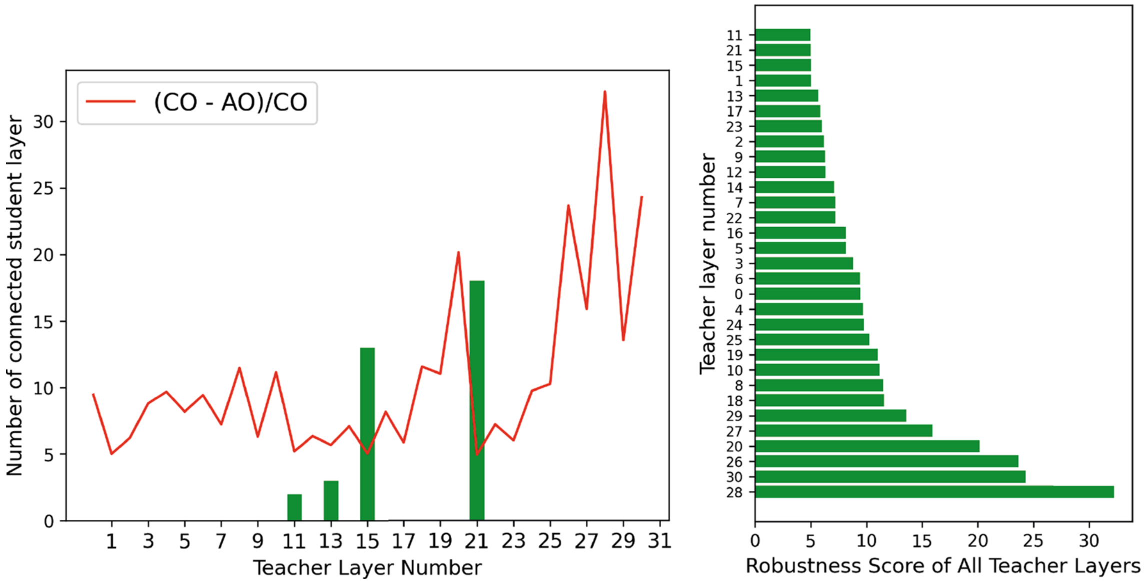


Fig. 7 (Left) Illustration of the number of student layers connected to each teacher layer on the CIFAR-10 for RNAS-CL-S5-WRT-34. (Right) Illustration of the robustness score of all the teacher layers. We have sorted teacher layers from the lowest scores to the highest (lower robustness score suggests more robustness)

RNAS-CL enhances the clean and robust accuracy compared to the standard knowledge distillation. This interesting observation evidences the potential of the proposed RNAS-CL framework, because researchers in the communities of NAS and adversarial training can use our RNAS-CL with more advanced distillation methods to obtain better results in the future.

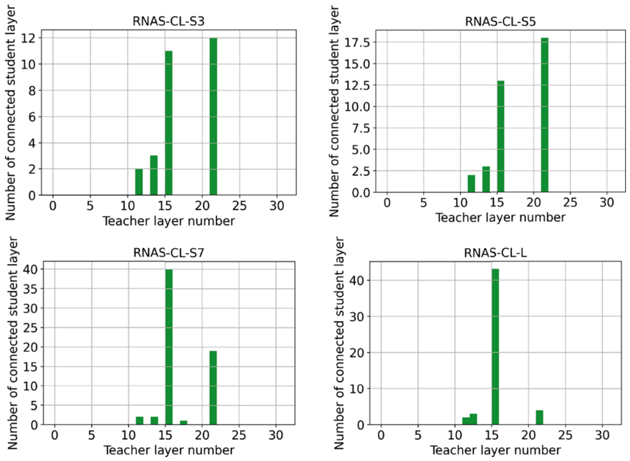


Fig. 8 Illustrations of the number of student layers connected to each teacher layer in RNAS-CL for various student models on the CIFAR-10 dataset. We choose adversarially trained Wide-ResNet-34 as the robust teacher model for all four student models, with one plot for each student model. All student architectures are described in Table 13

5 Discussions

5.1 Robust Teacher Layers

In this section, we discuss robustness inducing capacity of teacher layers. We hypothesize that there are teacher layers which are more robust than others and thus should induce more robustness to the student models. RNAS-CL identifies such robust teacher layers and uses these robust layers to teach the student network. In RNAS-CL, each student layer is associated with a teacher layer. Figures 8 and 10 illustrate the number of student layers connected to each robust teacher layer on the CIFAR-10 and the ImageNet-100 datasets. For all the student models on CIFAR-10, we observe that layers 15 and 21 of the robust teacher model have significantly more intermediate connections with the student models. Similarly, for ImageNet-100, layers 18, 32, and 40 are the dominant robust layers. In Figs. 9 and 11, we visualize the most robust teacher layers on CIFAR-10 and ImageNet-100, respectively. For both datasets, there are teacher layers which have significantly more intermediate connections for several models, suggesting that such chosen teacher layers have higher robustness-inducing capacity than other layers.

We conduct more experiments to explore the robustness of each teacher layer. We define the robustness of a teacher layer as the difference between the output corresponding to a clean and perturbed image. Let X be a clean input image and X_{adv} be the perturbed image. Then, CO_j and AO_j represent the output corresponding to the clean and the perturbed image at the j-th teacher layer. We define the robustness score of the j-th teacher layer as $|\text{CO}_j - \text{AO}_j|/|\text{CO}_j|$. Intuitively, a

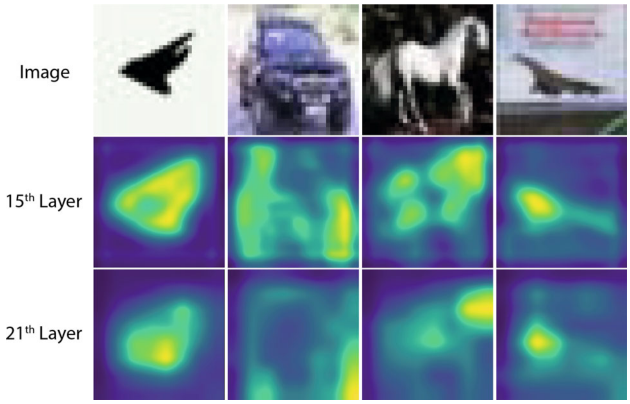


Fig. 9 Attention map for the most robust teacher layers on CIFAR-10 dataset. We chose the same robust teacher model as that in Fig. 8. The illustrated layers are the two teacher layers with the maximum number of intermediate connections for various RNAS-CL models (as described in Fig. 8)

more robust layer would lead to closer values of CO_j and AO_j with j being the index of that layer, so a lower robustness score suggests more robustness. In Fig. 7, we observe that layers with the most student connections also often have the minimum robustness loss. All the four teacher layers (11, 13, 15, 21) connected to the student model are within the top-5 layers with the lowest robustness scores. This result indicates that teacher layers with significantly more intermediate connections have more robustness-inducing capacity than others using the definition of robustness score.

5.2 Teacher's Influence on Student's Performance

In this section, we discuss how the teacher influences the student's performance. We conduct experiments using three different robust teacher models, which are adversarially trained WRT-34 (Rice et al., 2020), ResNet-50 (Engstrom et al., 2019), and ResNet-18 (Sehwag et al., 2021) on the

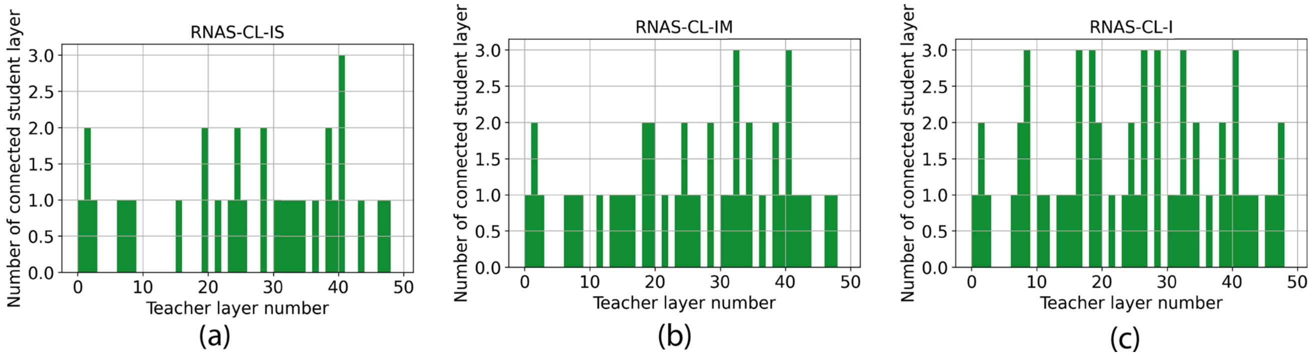


Fig. 10 Illustrations of the number of student layers connected to each teacher layer in RNAS-CL for various student models on the ImageNet-100 dataset. We choose adversarially trained Wide-ResNet-50 as the robust teacher for all the three student models, with one plot for each student model. All RNAS-CL architectures are described in Table 14

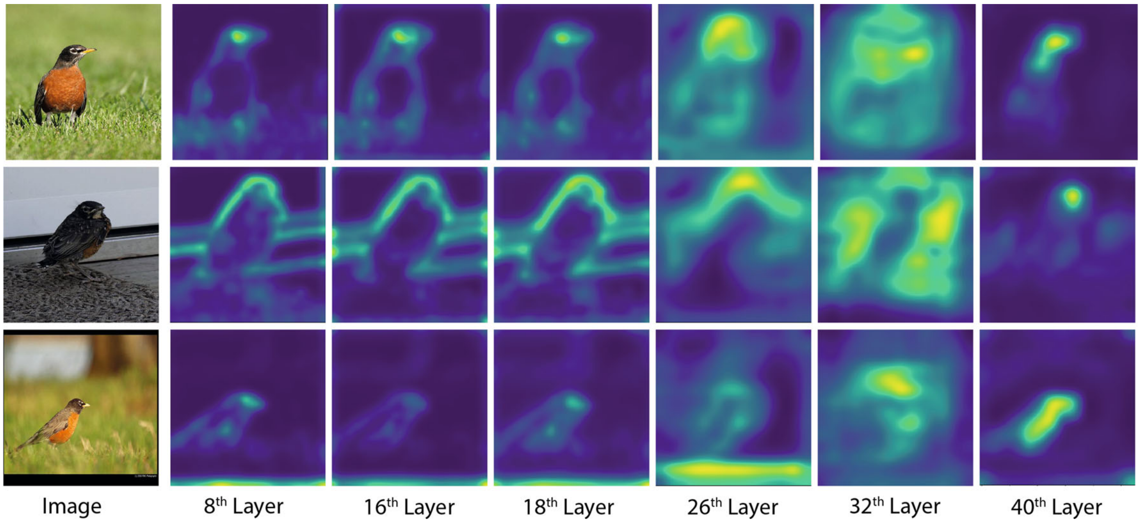


Fig. 11 Attention maps for the most robust teacher layers on ImageNet-100 dataset. We chose the same robust teacher model as in Fig. 10. The illustrated layers are the teacher layers with maximum number of intermediate connections for various RNAS-CL models (as described in Fig. 10)

Table 10 Performance of RNAS-CL method trained with various robust teacher models on the CIFAR-10 dataset

Method	Clean	PGD^{20}
Standard-S3	89.92	17.69
Standard-S5	90.76	18.44
Standard-S7	90.98	19.3
RNAS-CL-S3-WRT-34	89.4	34.3
RNAS-CL-S5-WRT-34	90.4	35.59
RNAS-CL-S7-WRT-34	90.62	37.24
RNAS-CL-S3-R50	89.39	35.76
RNAS-CL-S5-R50	90.53	37.32
RNAS-CL-S7-R50	90.41	37.98
RNAS-CL-S3-R18	88.47	26.35
RNAS-CL-S5-R18	88.77	25.49
RNAS-CL-S7-R18	89.47	27.96

"Standard" represents models searched and trained by cross-entropy loss without any teacher model

Table 11 Performance of RNAS-CL models trained with vulnerable and robust teacher on the CIFAR-10 dataset

Model	Clean	PGD^{20}
ResNet-50 (vulnerable teacher)	93.67	18.5
RNAS-CL-S3-R50	90.33	17.68
RNAS-CL-S5-R50	91.51	19.11
RNAS-CL-S7-R50	92.39	20.2
WRT-34 (robust teacher)	86.07	58.33
RNAS-CL-S3-WRT-34	89.4	34.3
RNAS-CL-S5-WRT-34	90.4	35.59
RNAS-CL-S7-WRT-34	90.62	37.24

Table 12 Performance of RNAS-CL models trained with robust transformer based teacher models on the CIFAR-10 dataset

Model	Clean	PGD^{20}
ViT	81.24	51.42
RNAS-CL-S3-ViT	88.1	33.39
RNAS-CL-S5-ViT	87.7	35.42
RNAS-CL-S7-ViT	87.0	34.3
DeiT Tiny	79.5	49.3
RNAS-CL-S3-Deit	87.0	30.8
RNAS-CL-S5-Deit	87.6	31.5
RNAS-CL-S7-Deit	88.2	32.1

CIFAR-10 dataset. All RNAS-CL models, while achieving similar clean accuracy, exceed its counterpart by more than 10% in PGD accuracy. RNAS-CL-R50 achieves higher robust accuracy than RNAS-CL-R18 and RNAS-CL-WRT-34. However, ResNet-50 has the lowest PGD accuracy among the teacher models, suggesting that the teacher's architecture influences the student's performance more than the teacher's performance. This result may be attributed to the fact that higher number of teacher layers allows more options for the student layer to learn from, leading to better robustness. We also observe similar results for ImagNet-100 in Fig. 4. The teacher models' performance is reported in Table 15 of the appendix.

5.3 Characteristics of Robust Architecture

After the searching phase of RNAS-CL, the number of channels, which is also the filter choice, at each convolution block of the network is decided. Consistent with the findings in (Huang et al., 2021), we observe that width reduction in the last few stages contributes to enhanced robust accuracy. For instances such as RNAS-CL-S3, RNAS-CL-S5, and RNAS-CL-S7, the filter choice is always made from the smallest two output channels within the search space for the last stage. The details about the search space of RNAS-CL for the CIFAR-10, the ImageNet, and the ImageNet-100 datasets are provided in Table 13 and Table 14 of the appendix.

5.4 Performance with Vulnerable Teacher

In this section, we examine the significance of robustness of the teacher model in RNAS-CL. We train RNAS-CL models using a vulnerable teacher with good clean accuracy but poor adversarial accuracy. In Table 11, we observe that RNAS-CL models trained with the vulnerable teacher model have significantly lower adversarial accuracy than models trained with a robust teacher. Here we use ResNet-50 as the vulnerable teacher and the adversarially trained WRT-34 as the robust teacher model.

5.5 Performance with Transformer-Based Models

In this section, we train RNAS-CL models using the transformer architecture with the results reported in Table 12. It is noteworthy to note that RNAS-CL models trained with the robust transformer-based models (Mo et al., 2022) as the teacher models often exhibit lower clean and robust accuracy compared to the RNAS-CL models trained with CNN-based robust teachers, which can be verified from Table 11 and Table 12. This decrease in robustness is attributed to the inherently lower adversarial robustness of transformer-based models. For example, the adversarially trained Vision Transformer (ViT) achieved a PGD accuracy of 51.42%, while its counterpart, WRT-34, attained a much higher PGD accuracy of 58.33%. However, it can be observed from Table 12 that RNAS-CL models trained with the robust transformer-based models still achieve a reasonable level of robustness. For example, with a much less robust teacher (ViT), RNAS-CL-

Table 13 The table describes the search space for CIFAR-10. Depth represents the depth of each stage

Search Space	Depth	Stage 1	Stage 2	Stage 3
RNAS-CL-S3	3-3-3	16, 12	32, 28, 24, 20	64, 60, 56, 52
RNAS-CL-S5	5-5-5	16, 12	32, 28, 24, 20	64, 60, 56, 52
RNAS-CL-S7	7-7-7	16, 12	32, 28, 24, 20	64, 60, 56, 52
RNAS-CL-M	9-7-1	80, 76	160, 156, 152, 148	128, 124, 120, 116
RNAS-CL-L	9-7-1	160, 156	320, 316, 312, 308	256, 252, 248, 244

For example, 3-3-3 represents three convolution blocks in each stage. All search spaces have three stages. Stage 1, Stage 2, and Stage 3 represent the filter choices for the corresponding stage. For example, at stage 3 of RNAS-CL-S3, we search among 4 output channels, (64, 60, 56, 52), for each convolution block

Table 14 The table describes the search space for ImageNet and ImageNet-100

Search space	Depth	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
RNAS-CL-IS	3-3-3	28, 24, 20, 16	40, 36, 32, 28	96, 88, 80, 72, 64, 56, 48		
RNAS-CL-IM	3-3-3-4	28, 24, 20, 16	40, 36, 32, 28	96, 88, 80, 72, 64, 56, 48	128 120, 108, 100, 92, 84, 76, 68	
NAS-CL-I	3-3-3-4-4	28, 24, 20, 16	40, 36, 32, 28	96, 88, 80, 72, 64, 56, 48	128 120, 108, 100, 92, 84, 76, 68	216, 208, 200, 192, 184,176, 168, 160, 152, 144,136, 128, 120, 108
RNAS-CL-IL	1-2-2-4-3	28, 24, 20, 16	40, 36, 32, 28	96, 88, 80, 72, 64, 56, 48	128 120, 108, 100, 92, 84, 76, 68	216, 208, 200, 192, 184,176, 168, 160, 152, 144,136, 128, 120, 108

Similar to Table 13, depth represents the depth of each stage. For ImageNet, we have up to 5 stages. Stage 1, Stage 2, Stage 3, Stage 4, and Stage 5 represent the filter choices for the corresponding stage. For example, at stage 1, we search among 4 output channel options, (28, 24, 20, 16), for each convolution block

S5-ViT still achieves a robust accuracy of 35.42%, which is very close to the robust accuracy of 35.59% achieved by RNAS-CL-S5-WRT-34 trained with a much more robust teacher (WRT-34).

Table 15 Robustness results for various teacher models on the CIFAR-10 dataset

Method	Clean	PGD^{20}
WRT-34	86.07	58.33
ResNet 18	84.59	55.54
ResNet 50	87.03	49.25

6 Conclusions

In this paper, we propose Robust Neural Architecture Search by Cross-Layer Knowledge Distillation (RNAS-CL), a novel NAS algorithm that improves the robustness of the student model by learning from a robust teacher through cross-layer knowledge distillation. RNAS-CL optimizes neural architecture to achieve a good tradeoff between robustness and clean accuracy in a differentiable manner either with or without robust training. The experiments show that the compact models trained by RNAS-CL outperform the competing models obtained without robust training in terms of adversarial robustness, and adding adversarial training can further increase the adversarial robustness of the RNAS-CL models. After robust training, RNAS-CL achieves better adversarial robustness compared to competing models obtained via robust training.

Appendix A Robust Teacher Models

In this section, we report the robustness of adversarially trained teacher models used throughout the paper on the CIFAR-10 dataset in Table 15.

Appendix B Architecture

In this section, we discuss architectures for various proposed supernets used in RNAS-CL for the CIFAR-10, the ImageNet-100 and the ImageNet datasets. Table 13 describes the supernets used for CIFAR-10. We use supernets with three blocks. Supernets used for ImageNet-100 and ImageNet are described in Table 14. For ImageNet-100, the number of blocks varies from 3 to 5.

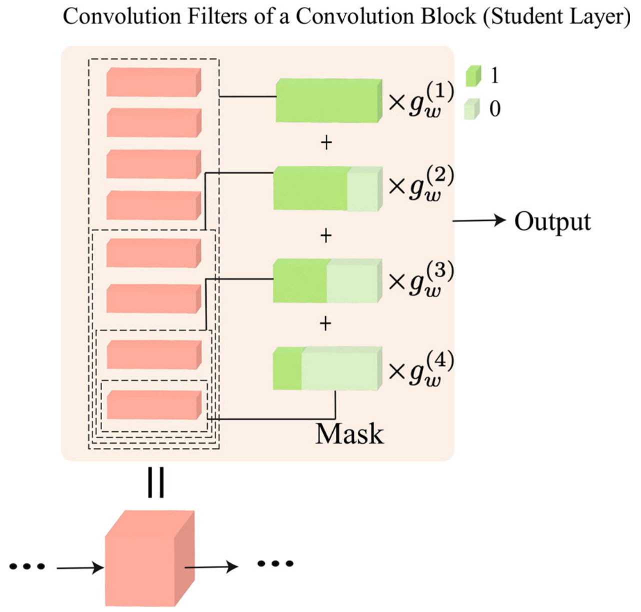


Fig. 12 Illustration of searching for the neural architecture of a convolution layer of a student model using the searching mechanism in FBNetV2. $\left\{g_w^{(i)}\right\}$ represents the Gumbel weights associated with different filter choices

Appendix C Architecture Search by FBNetV2

RNAS-CL builds an efficient and adversarially robust deep learning model. In this work, we use the training paradigm of FBNetV2 to search for efficient neural architecture. Figure 12 illustrates the searching process for the neural architecture of a single convolution layer. Each filter choice is associated with a Gumbel weight. These Gumbel weights are optimized to decide the best filter choice for the convolution layer.

Acknowledgements This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 17STQAC00001-07-00. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security. This work was supported by the 2023 Mayo Clinic and Arizona State University Alliance for Health Care Collaborative Research Seed Grant Programm, and this work was supported in part by NSF grant 2323086.

Data Availibility The following datasets are employed in the experiments of this paper. (1) CIFAR-10, a collection of 60k images in 10 classes (Krizhevsky, 2009), which is available at `https://www.cs.toronto.edu/~kriz/cifar.html`; (2) ImageNet (ILSVRC-12), an image classification dataset (Russakovsky et al., 2015) with 1000 classes and about 1.2M images, which is available at https://www.image-net.org/challenges/LSVRC/; (3) ImageNet-100, a subset of ImageNet-1k dataset (Russakovsky et al., 2015) with 100 classes and about 130k images (Tian et al., 2020), which is available at https://github.com/HobbitLong/CMC.

References

Ahn, S., Hu, S. X., Damianou, A., Lawrence, N. D., & Dai, Z. (2019). Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9163–9171).

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems, 32*.

Cai, H., Zhu, L., & Han, S. (2019). Proxylessnas: Direct neural architecture search on target task and hardware. In *7th international conference on learning representations, ICLR*.

Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 IEEE symposium on security and privacy* (pp. 39–57). IEEE

Chen, M., Peng, H., Fu, J., & Ling, H. (2021) Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 12270–12280).

Cissé, M., Bojanowski, P., Grave, E., Dauphin, Y. N., & Usunier, N. (2017). Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th international conference on machine learning, ICML. Proceedings of machine learning research* (Vol. 70, pp. 854–863). PMLR.

Croce, F., & Hein, M. (2020). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning* (pp. 2206–2216). PMLR

Devaguptapu, C., Agarwal, D., Mittal, G., Gopalani, P., & Balasubramanian, V. N. (2021). On adversarial robustness: A neural architecture search perspective. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 152–161).

Dong, M., Li, Y., Wang, Y., & Xu, C. (2020). Adversarially robust neural architectures. arXiv preprint arXiv:2009.00902

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., & Li, J. (2018). Boosting adversarial attacks with momentum. In *CVPR* (pp. 9185–9193). Computer Vision Foundation/IEEE Computer Society.

Dziugaite, G. K., Ghahramani, Z., & Roy, D. M. (2016). A study of the effect of jpg compression on adversarial images. arXiv preprint arXiv:1608.00853

Engstrom, L., Ilyas, A., Salman, H., Santurkar, S., & Tsipras, D. (2019). Robustness (python library). https://github.com/MadryLab/robustness

Goldblum, M., Fowl, L., Feizi, S., & Goldstein, T. (2020). Adversarially robust distillation. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 3996–4003).

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *3rd international conference on learning representations, ICLR*.

Gui, S., Wang, H., Yang, H., Yu, C., Wang, Z., & Liu, J. (2019). Model compression with adversarial robustness: A unified optimization framework. In *Annual conference on neural information processing systems* (pp. 1283–1294).

Guo, C., Rana, M., Cissé, M., & Maaten, L. (2018). Countering adversarial images using input transformations. In *6th international conference on learning representations, ICLR*.

Guo, M., Yang, Y., Xu, R., Liu, Z., & Lin, D. (2020). When NAS meets robustness: In search of robust architectures against adversarial attacks. In *IEEE/CVF conference on computer vision and pattern recognition, CVPR* (pp. 628–637).

Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M. A., & Dally, W. J. (2016). EIE: Efficient inference engine on compressed deep neural network. In *ISCA* (pp. 243–254). IEEE Computer Society.

Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. *Advances in Neural Information Processing Systems, 28*.

Hein, M., & Andriushchenko, M. (2017). Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Annual conference on neural information processing systems* (pp. 2266–2276).

Hinton, G. E., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. CoRR arXiv:1503.02531

Huang, H., Wang, Y., Erfani, S. M., Gu, Q., Bailey, J., & Ma, X. (2021). Exploring architectural ingredients of adversarially robust deep neural networks. In *Advances in neural information processing systems* (pp. 5545–5559).

Jang, E., Gu, S., & Poole, B. (2017). Categorical reparameterization with Gumbel–Softmax. In *5th international conference on learning representations, ICLR*.

Kannan, H., Kurakin, A., & Goodfellow, I. (2018). Adversarial logit pairing. arXiv preprint arXiv:1803.06373

Krizhevsky, A. (2009). *Learning multiple layers of features from tiny images*. Technical report, Univ. Toronto.

Kurakin, A., Goodfellow, I., Bengio, S., Dong, Y., Liao, F., Liang, M., Pang, T., Zhu, J., Hu, X., & Xie, C., *et al.* (2018). Adversarial attacks and defences competition. In *The NIPS'17 competition: Building intelligent systems* (pp. 195–231). Springer.

Li, H.-T., Lin, S.-C., Chen, C.-Y., & Chiang, C.-K. (2019). Layer-level knowledge distillation for deep neural network learning. *Applied Sciences, 9*(10).

Li, C., Peng, J., Yuan, L., Wang, G., Liang, X., Lin, L., & Chang, X. (2020). Block-wisely supervised neural architecture search with knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1989–1998).

Li, Y., Yang, Z., Wang, Y., & Xu, C. (2021). Neural architecture dilation for adversarial robustness. *Advances in Neural Information Processing Systems, 34*.

Lian, D., Zheng, Y., Xu, Y., Lu, Y., Lin, L., Zhao, P., Huang, J., & Gao, S. (2019). Towards fast adaptation of neural architectures with meta learning. In *International conference on learning representations*.

Liu, X., Cheng, M., Zhang, H., & Hsieh, C. (2018). Towards robust neural networks via random self-ensemble. In *European conference on computer vision, ECCV* (pp. 381–397).

Liu, H., Simonyan, K., & Yang, Y. (2019). Darts: Differentiable architecture search. In *7th international conference on learning representations, ICLR*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *6th international conference on learning representations, ICLR*.

Mok, J., Na, B., Choe, H., & Yoon, S. (2021). Advrush: Searching for adversarially robust neural architectures. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 12322–12332).

Mo, Y., Wu, D., Wang, Y., Guo, Y., & Wang, Y. (2022). When adversarial training meets vision transformers: Recipes from training to architecture. *Advances in Neural Information Processing Systems, 35*, 18599–18611.

Nath, U., Kushagra, S., & Yang, Y. (2020). Adjoined networks: A training paradigm with applications to network compression. arXiv preprint arXiv:2006.05624

Ning, X., Zhao, J., Li, W., Zhao, T., Zheng, Y., Yang, H., & Wang, Y. (2020). Discovering robust convolutional architecture at targeted capacity: A multi-shot approach. arXiv preprint arXiv:2012.11835

Pang, T., Xu, K., Dong, Y., Du, C., Chen, N., & Zhu, J. (2020). Rethinking SoftMax cross-entropy loss for adversarial robustness. In *8th international conference on learning representations, ICLR*.

Park, W., Kim, D., Lu, Y., & Cho, M. (2019). Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3967–3976).

Park, J., Li, S. R., Wen, W., Tang, P. T. P., Li, H., Chen, Y., & Dubey, P. (2017). Faster CNNs with direct sparse convolutions and guided pruning. In *5th international conference on learning representations, ICLR 2017, Toulon, France, April 24–26, 2017, conference track proceedings*. OpenReview.net.

Passalis, N., & Tefas, A. (2018). Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 268–284).

Peng, H., Du, H., Yu, H., Li, Q., Liao, J., & Fu, J. (2020). Cream of the crop: Distilling prioritized paths for one-shot neural architecture search. *Advances in neural information processing systems, 33*, 17955–17964.

Real, E., Aggarwal, A., Huang, Y., & Le, Q. V. (2019). Regularized evolution for image classifier architecture search. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 4780–4789).

Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y. L., Tan, J., Le, Q. V., & Kurakin, A. (2017). Large-scale evolution of image classifiers. In: Precup, D., Teh, Y. W. (Eds.), *Proceedings of the 34th international conference on machine learning, ICML. Proceedings of machine learning research* (Vol. 70, pp. 2902–2911). PMLR.

Rice, L., Wong, E., & Kolter, Z. (2020). Overfitting in adversarially robust deep learning. In *International conference on machine learning* (pp. 8093–8104). PMLR

Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., & Bengio, Y. (2015). Fitnets: Hints for thin deep nets. In *3rd international conference on learning representations (ICLR)*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision, 115*(3), 211–252.

Sehwag, V., Mahloujifar, S., Handina, T., Dai, S., Xiang, C., Chiang, M., & Mittal, P. (2021). Robust learning meets generative models: Can proxy distributions improve adversarial robustness? In *International conference on learning representations*.

Sehwag, V., Wang, S., Mittal, P., & Jana, S. (2020). HYDRA: pruning adversarially robust neural networks. In *Annual conference on neural information processing systems*.

Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., & Goldstein, T. (2019). Adversarial training for free! *Advances in Neural Information Processing Systems, 32*.

Su, D., Zhang, H., Chen, H., Yi, J., Chen, P., & Gao, Y. (2018). Is robustness the cost of accuracy? A comprehensive study on the robustness of 18 deep image classification models. In *European Conference on Computer Vision, ECCV* (pp. 644–661).

Sun, D., Yao, A., Zhou, A., & Zhao, H. (2019). Deeply-supervised knowledge synergy. In *IEEE conference on computer vision and pattern recognition, CVPR* (pp. 6997–7006).

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., & Fergus, R. (2014). Intriguing properties of neural networks. In *2nd international conference on learning representations, ICLR*.

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). MnasNet: Platform-aware neural architecture search for mobile. In *IEEE conference on computer vision and pattern recognition, CVPR* (pp. 2820–2828).

Tian, Y., Krishnan, D., & Isola, P. (2020). Contrastive multiview coding. In *European conference on computer vision* (pp. 776–794). Springer

Tian, Y., Krishnan, D., & Isola, P. (2020). Contrastive representation distillation. In *8th international conference on learning representations, ICLR*.

Tian, Y., Krishnan, D., & Isola, P. (2020). Contrastive representation distillation. In *International conference on learning representations*.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I. J., Boneh, D., & McDaniel, P. D. (2018). Ensemble adversarial training: Attacks and defenses. In *6th international conference on learning representations, ICLR*.

Tung, F., & Mori, G. (2019). Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1365–1374).

Wan, A., Dai, X., Zhang, P., He, Z., Tian, Y., Xie, S., Wu, B., Yu, M., Xu, T., Chen, K., Vajda, P., & Gonzalez, J.E. (2020). Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. In *IEEE/CVF conference on computer vision and pattern recognition, CVPR* (pp. 12962–12971).

Wong, E., Rice, L., & Kolter, J. Z. (2020). Fast is better than free: Revisiting adversarial training. In *International conference on learning representations (ICLR)*.

Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., Tian, Y., Vajda, P., Jia, Y., & Keutzer, K. (2019). Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *IEEE conference on computer vision and pattern recognition, CVPR* (pp. 10734–10742).

Xie, C., & Yuille, A. L. (2020). Intriguing properties of adversarial training at scale. In *8th international conference on learning representations, ICLR*.

Xie, G., Wang, J., Yu, G., Lyu, J., Zheng, F., & Jin, Y. (2023). Tiny adversarial multi-objective one-shot neural architecture search. *Complex and Intelligent Systems*.

Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., & Yuille, A. L. (2019). Improving transferability of adversarial examples with input diversity. In *IEEE conference on computer vision and pattern recognition, CVPR* (pp. 2730–2739).

Xu, Y., Xie, L., Zhang, X., Chen, X., Qi, G., Tian, Q., & Xiong, H. (2020). Pc-darts: Partial channel connections for memory-efficient architecture search. In *ICLR*. OpenReview.net.

Yan, Z., Guo, Y., & Zhang, C. (2018). Deep defense: Training DNNs with improved adversarial robustness. In *Annual conference on neural information processing systems* (pp. 417–426).

Yang, Y., You, S., Li, H., Wang, F., Qian, C., & Lin, Z. (2021). Towards improving the consistency, efficiency, and flexibility of differentiable neural architecture search. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6667–6676).

Yang, Y., Li, H., You, S., Wang, F., Qian, C., & Lin, Z. (2020). ISTA-NAS: Efficient and consistent neural architecture search by sparse coding. *Advances in Neural Information Processing Systems, 33*, 10503–10513.

Ye, S., Lin, X., Xu, K., Liu, S., Cheng, H., Lambrechts, J., Zhang, H., Zhou, A., Ma, K., & Wang, Y. (2019). Adversarial robustness vs. model compression, or both? In *IEEE/CVF international conference on computer vision, ICCV* (pp. 111–120).

Yim, J., Joo, D., Bae, J., & Kim, J. (2017). A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *IEEE conference on computer vision and pattern recognition, CVPR* (pp. 7130–7138).

Yue, Z., Lin, B., Zhang, Y., & Liang, C. (2022). Effective, efficient and robust neural architecture search. In *2022 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Zagoruyko, S., & Komodakis, N. (2017). Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *5th international conference on learning representations, ICLR*.

Zhai, X., Oliver, A., Kolesnikov, A., & Beyer, L. (2019). S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1476–1485).

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., El Ghaoui, L., & Jordan, M. I. (2019). Theoretically principled trade-off between robustness and accuracy. In *Proceedings of the 36th International Conference on Machine Learning, ICML. Proceedings of Machine Learning Research* (Vol. 97, pp. 7472–7482). PMLR.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., & Jordan, M. I. (2019). Theoretically principled trade-off between robustness and accuracy. In *ICML. Proceedings of machine learning research* (Vol. 97, pp. 7472–7482). PMLR.

Zoph, B., & Le, Q. V. (2017). Neural architecture search with reinforcement learning. In *5th international conference on learning representations, ICLR*.

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8697–8710).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.