ON ADVERSARIAL ROBUSTNESS: A NEURAL ARCHITECTURE SEARCH PERSPECTIVE

Chaitanya Devaguptapu[1], **Devansh Agarwal**[1,*], **Gaurav Mittal**[2*], **Vineeth N Balasubramanian**[1]
[1]Indian Institute of Technology, Hyderabad [2]Microsoft
{cs19mtech11025,es16btech11009,vineethnb}@iith.ac.in
gaurav.mittal@microsoft.com

ABSTRACT

Adversarial robustness of deep learning models has gained much traction in the last few years. While many approaches have been proposed to improve adversarial robustness, one promising direction for improving adversarial robustness is un-explored, i.e., the complex topology of the neural network architecture. In this work, we empirically understand the effect of architecture on adversarial robustness by experimenting with different hand-crafted and NAS based architectures. Our findings show that, for small-scale attacks, NAS-based architectures are more robust for small-scale datasets and simple tasks than hand-crafted architectures. However, as the dataset's size or the task's complexity increases, hand-crafted architectures are more robust than NAS-based architectures. We perform the first large-scale study to understand adversarial robustness purely from an *architectural perspective*. Our results show that random sampling in the search space of DARTS (a popular NAS method) with simple ensembling can improve the robustness to PGD attack by nearly 12%. We show that NAS, which is popular for SoTA accuracy, can provide adversarial accuracy as a *free add-on* without any form of adversarial training. We also introduce a metric that can be used to calculate the trade-off between clean accuracy and adversarial robustness.

1 INTRODUCTION AND RELATED WORK

Topology of a neural network plays a crucial role in the performance of any deep learning system. Zoph & Le (2016) introduced Neural Architecture Search (NAS) to automate the painstaking task of hand-designing network architectures by searching for the network topology that maximizes performance. Since then, several NAS-based algorithms have been introduced (Yan et al., 2019; Chen et al., 2019; Pham et al., 2018) with NAS-based architectures achieving state-of-the-art (SoTA) performance across a wide spectrum of computer vision tasks.

An adversarial attack refers to subjecting a neural network to images, which have been perturbed with humanly imperceptible noise, in order to fool the network to wrongly classify them with high confidence. Adversarial attacks can be white-box, where the attackers have access to the network architecture and parameters, or black-box, where the attacker is oblivious to architecture and/or parameters. Several defenses have been proposed to adversarially train the model to boost its robustness to such attacks. Given that a white-box attack can be a function of the network topology, it raises an important question, *Can the complex topology of a neural network architecture provide adversarial robustness without any form of adversarial training?*. We attempt to answer this question in our work focusing on white-box attacks (and also providing some results on black-box attacks in the Appendix). For more details on existing work on NAS, adversarial attacks/defenses, please refer to Chakraborty et al. (2018); Ren et al. (2020).

To evaluate adversarial robustness purely from an **architectural perspective**, we attempt to answer,

- How do NAS-based architectures compare with hand-crafted architectures (like ResNets, DenseNets, *etc.*) in terms of architectural robustness?
- Does an increase in the number of parameters of the architecture help improve robustness?
- Where does the source of adversarial vulnerability lie for NAS? Is it in the search space or in the way the current methods are performing the search?

[*]equal contribution

Previous works (Madry et al., 2017; Guo et al., 2020; Vargas et al., 2019) attempt to shed light on how architectural aspects like dense connections, parameter count, *etc.* affect robustness. But in these works, the use of adversarial training makes it difficult to assess the role of network topology itself in robustness. Moreover, no study compares/evaluates the robustness of existing NAS approaches (which, as per our study, are already quite robust without any explicit adversarial training). To the best of our knowledge, our work is the first attempt to understand robustness purely from an architectural perspective without using any form of adversarial training. We also introduce two simple metrics, *Harmonic Robustness Score (HRS)* and *Per-parameter HRS (PP-HRS)* that combine (1) accuracy on both clean and perturbed samples and (2) parameter count, to convey how robust and deployment-ready a given model is when no adversarial training is performed.

We examine adversarial robustness of multiple hand-crafted and NAS-based architectures on datasets of different sizes/complexities. There is indeed a correlation between network topology and adversarial robustness (Figure 1). We find that traditional hand-crafted network topologies like ResNet and DenseNet are more robust than NAS-based architectures. Our study can be helpful to design architectures that provide higher robustness out-of-the-box in addition to clean SoTA performance. We also show that the popular way of increasing parameters to increase robustness (Madry et al., 2017; Xie & Yuille, 2019) holds only up to a certain threshold—increasing parameters beyond that hurt both adversarial and clean accuracy.

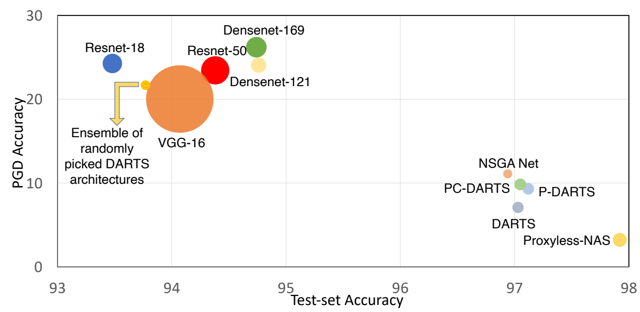

Figure 1: Comparison of test-set accuracy and PGD accuracy of NAS and hand-crafted architectures on CIFAR-10 dataset. Bubble size represents the number of parameters

2 ROBUSTNESS OF NAS MODELS: A STUDY

Datasets and Architectures: To assess architectural robustness on datasets of varying scale and complexity, we perform experiments on CIFAR-10/100 (Krizhevsky et al., a;b), fine-grained 102-Flowers (Nilsback & Zisserman, 2008) and large-scale ImageNet (Deng et al., 2009) dataset.

Based on datasets supported, we evaluate on five handcrafted architectures and popular NAS methods that include DARTS (Liu et al., 2018), P-DARTS (Chen et al., 2019), ProxylessNAS (Cai et al., 2019), NSGA-Net (Lu et al., 2018), PC-DARTS (Xu et al., 2020) and DenseNAS (Fang et al., 2019).

To understand the contribution of ensemble-based architectures to adversarial robustness, in Section 3.3 we build an ensemble using randomly sampled cells (zero search cost) from the DARTS search space. Each of these architectures is independently trained using the standard training protocol. We ensure the number of cells in the total ensemble is equal to the number of cells in any standard DARTS architecture. Outputs of different models in the ensemble are combined using a simple linear model, which is just trained for two epochs. The difference between standard DARTS and ensembling by sampling from DARTS search space is visually shown in Appendix Figure 3.

Adversarial Attacks and metrics: We test against standard adversarial attacks including FGSM (Goodfellow et al., 2014), PGD (Madry et al., 2017) and F-FGSM(Wong et al., 2020) using perturbation of $8/255(3e^{-2})$ (maximum noise added to any image pixel), step size of $2/255(7e^{-3})$ and 10 attack iterations (Pang et al., 2020; Fan & Li, 2020; Wong et al., 2020). All architectures are trained using standard training protocal with no adversarial training.

Evaluation Metrics: We use Clean Accuracy and Adversarial Accuracy as evaluation metrics that represent the accuracy on undisturbed and adversarially perturbed test set respectively. There is a trade-off between clean and adversarial accuracy in that clean accuracy is higher for non-adversarially-trained models than adversarially-trained models and vice versa in the case of adversarial accuracy. There is no metric that captures this trade-off, and so to capture it, we introduce a new metric called *Harmonic Robustness Score (HRS)* as the harmonic mean of clean accuracy, C and adversarial accuracy, P against PGD attack, HRS $= \frac{2CP}{C+P}$. To compare across network architectures within a model family, we further define per-parameter harmonic robustness score (PP-HRS).

In a model family $\mathcal{F}$ with baseline model m_b having p_b parameters, for a model $m_i \in \mathcal{F}$ with p_i parameters, PP-HRS is computed as PP-HRS $=$ HRS $* \frac{p_b}{p_i}$. HRS and PP-HRS help measure the usefulness of a model along with its resilience in the absence of any adversarial training.

3 Analysis and Results

3.1 How do NAS based models compare with hand-crafted models in terms of architectural robustness?

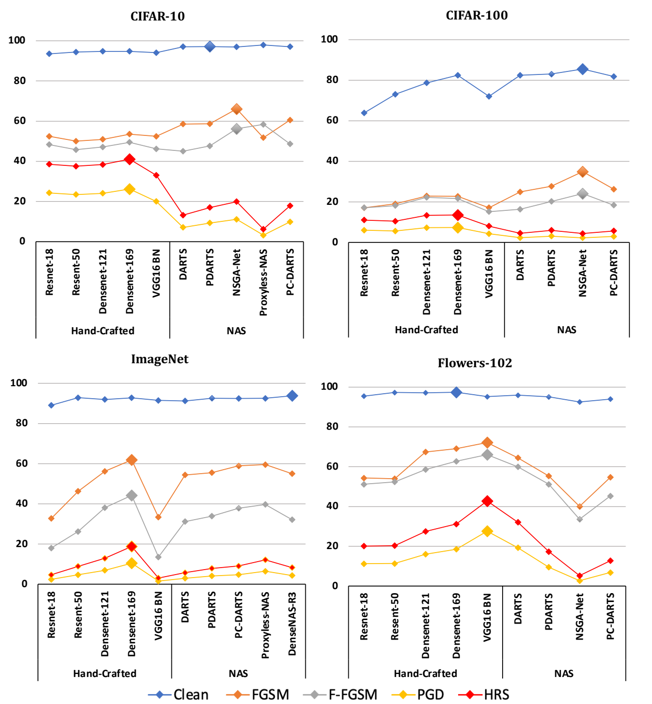

Figure 2: Comparison of robustness and clean accuracy of different architectures; As the difficulty of the task or the scale of the dataset increases hand-crafted architectures are more robust; (best performance is indicated by diamond symbol)

Figure 4 compares the HRS score, clean and adversarial accuracy for various hand-crafted and NAS architectures for CIFAR-10/100, 102-Flowers and Imagenet dataset. For smaller datasets like CIFAR-10/100, NAS-based architectures achieve significantly higher adversarial accuracy than hand-crafted architectures on attacks like FGSM and F-FGSM but perform significantly lower on stronger attacks like PGD. As a result, the HRS score of the best-performing hand-crafted model is 21% and 7% higher than best-performing NAS model on CIFAR-10 and CIFAR-100 respectively.

For large-scale ImageNet and fine-grained 102-Flowers dataset, hand-crafted models are more robust than NAS-based architectures for all adversarial attacks. We infer that as the dataset size or task complexity increases, hand-crafted models start performing better for all adversarial attacks considered. For stronger attacks like PGD, hand-crafted models are more robust than NAS-based architectures across all dataset sizes/complexities. While NAS-based architectures can achieve SoTA clean accuracy in general, their adversarial robustness without explicit adversarial training is unreliable, particular for large and complex datasets.

3.2 Does an increase in the number of parameters of architecture help improve robustness?

Within the same family of architectures, increasing the number of network parameters helps improve robustness (Su et al. (2018); Madry et al. (2017)). To study this claim, we compare the robustness of different families of architectures on the ImageNet dataset . We use PGD accuracy along with the PP-HRS (defined in Section 2) as the evaluation metrics. The family of architectures we consider are 1. EfficientNet (which are mix of both NAS and hand-crafted method like compound scaling), 2. DenseNAS 3. ResNets and DenseNets.

Results for this experiments are shown in Appendix (Table 6 and Figure 4). Excluding EfficientNets, an increase in parameters increases both clean and adversarial accuracy. The maximum value of the parameter count in these families is nearly 26 million. A similar trend is also observed in EfficientNets but only put a parameter count of 20 million. Beyond 20 million, increasing parameters *alone* results in a decrease of both clean and adversarial accuracy; this is probably why EfficientNet considers different image sizes for each of the eight networks. *"In what family of architectures, the increase in parameter count is helping the performance?"*, to better understand this, we report

PP-HRS in Appendix (Table 6). In the case of DenseNAS models developed using MobileNet-V2 search space, an increase in parameters from DenseNAS-A to DenseNAS-Large is improving both clean accuracy and adversarial robustness; which as a result led to improved PP-HRS score. Excluding the EfficientNet family, for all the other family of architectures in our study, the increased parameter count does not give a significant and sufficient improvement in the PP-HRS score and adversarial robustness. In summary, adversarial robustness can be improved by increasing the number of parameters, but this holds only to an extent. Beyond a certain point (approximately 20-25 million as per our results), increasing parameters alone cannot improve adversarial robustness.

3.3 WHAT IS THE SOURCE OF ADVERSARIAL VULNERABILITY LIE FOR NAS? IS IT IN THE SEARCH SPACE OR THE SEARCH ALGORITHM?

In Section 3.1, we observe that although NAS-based architectures are more robust than hand-crafted networks for small datasets and simpler attacks, they perform poorly against standard attacks like PGD even for a small dataset like CIFAR-10. Most of the NAS algorithms work in two stages of searching a network cell on CIFAR-10 or subset of ImageNet and then stacking the discovered cell to train on other datasets. To further investigate whether the problem lies in the search space or search algorithm, we perform two simple experiments.

Motivated by Yang et al. (2020), which shows that a randomly sampled cell in the DARTS search space gives as good a clean accuracy as a searched cell, we conduct our first experiment to test this for adversarial robustness. We sample random cells from DARTS search space, stack and train them using standard procedure, and test their robustness against PGD attack for CIFAR-10. Table 1 tabulates the results of this experiment. Due to the random sampling, we report an average of 4 different runs. We observe that even with random sampling, we can achieve a better adversarial accuracy against PGD attack on average. But the variance is very high, suggesting that relying on randomly sampled architectures for better adversarial robustness is not a good idea.

Model	# cells	Params (M)	Clean %	PGD
DARTS Liu et al. (2018)	20	3.35	97.03	7.09
P-DARTS Chen et al. (2019)	20	3.43	**97.12**	9.31
PC-DARTS Xu et al. (2020)	20	3.63	97.05	9.84
RANDOM*	20	2.73 ± 0.49	95.57 ± 0.40	14.47 ± 4.70
ENSEMBLE†	20	2.74 ± 0.41	93.77 ± 0.39	$\mathbf{21.68 \pm 0.35}$

Table 1: Adversarial accuracy comparison of DARTS based architectures on CIFAR-10 dataset. $\star$ Randomly picked architectures from DARTS search-space. † Ensemble of small, randomly picked architectures from DARTS search space.

Therefore, in our second experiment, we randomly sample cells from DARTS search space to build small (weak) models and train them independently. We then ensemble the models by combining their outputs via a small linear network that consists of 2 linear layers with BatchNorm and a classification layer; this network is fine-tuned for just two epochs. For a fair comparison, we ensure that the ensemble overall has the same number of cells as a standard DARTS network. Using the entire ensemble as a single network, we generate adversarial examples via PGD to compute adversarial accuracy. Due to random sampling, we again report the average accuracy over 4 runs and restrict experiments to CIFAR-10 and DARTS search space due to computational cost. For full procedure, see Fig. 3 in Appendix. As observed in Table 1, this simple ensemble of randomly sampled architectures can improve accuracy of DARTS against PGD by nearly 12% with significantly low variance.

Now, this leads to two interesting conclusions; (1) Learning to build a simple network to combine the outputs of randomly sampled architectures can give clean accuracy with adversarial robustness as an add-on. In this case, we used a simple linear model; replacing this with a searched NAS based architecture can improve the results further. (2) Using NAS to search for an ensemble of architectures can be a potential way to achieve adversarial robustness as an add-on to SoTA clean accuracy. In this case, the NAS objective should be modified to find small models that can complement each other. We plan to explore this in our future work.

4 CONCLUSION

We present a detailed analysis of the adversarial robustness of NAS and hand-crafted models and show how the complex topology of neural networks can be leveraged to achieve a good amount of adversarial robustness without any form of adversarial training. We also introduce a metric that can be used to calculate the trade-off between clean and adversarial accuracy within and across different families of architectures. Finally, we show that using NAS to find an ensemble of architectures can be one potential way to build robust and reliable models without any form of adversarial training.

References

Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In *International Conference on Learning Representations*, 2019. URL `https://arxiv.org/pdf/1812.00332.pdf`.

Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial Attacks and Defences: A Survey. *arXiv e-prints*, art. arXiv:1810.00069, September 2018.

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 International Conference on Computer Vision*, pp. 1294–1303, 2019.

Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive Differentiable Architecture Search: Bridging the Depth Gap between Search and Evaluation. *arXiv e-prints*, art. arXiv:1904.12760, April 2019.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Jiameng Fan and Wenchao Li. Adversarial Training and Provable Robustness: A Tale of Two Objectives. *arXiv e-prints*, art. arXiv:2008.06081, August 2020.

Jiemin Fang, Yuzhu Sun, Qian Zhang, Yuan Li, Wenyu Liu, and Xinggang Wang. Densely Connected Search Space for More Flexible Neural Architecture Search. *arXiv e-prints*, art. arXiv:1906.09607, June 2019.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *arXiv e-prints*, art. arXiv:1412.6572, December 2014.

Minghao Guo, Yuzhe Yang, Rui Xu, and Ziwei Liu. When nas meets robustness: In search of robust architectures against adversarial attacks. *CVPR*, 2020.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). a. URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research). b. URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable Architecture Search. *arXiv e-prints*, art. arXiv:1806.09055, June 2018.

Zhichao Lu, Ian Whalen, Vishnu Boddeti, Yashesh Dhebar, Kalyanmoy Deb, Erik Goodman, and Wolfgang Banzhaf. NSGA-Net: Neural Architecture Search using Multi-Objective Genetic Algorithm. *arXiv e-prints*, art. arXiv:1810.03522, October 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv e-prints*, art. arXiv:1706.06083, June 2017.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of Tricks for Adversarial Training. *arXiv e-prints*, art. arXiv:2010.00467, October 2020.

Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient Neural Architecture Search via Parameter Sharing. *arXiv e-prints*, art. arXiv:1802.03268, February 2018.

Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346 – 360, 2020. ISSN 2095-8099. doi: https://doi.org/10.1016/j.eng.2019.12.012. URL `http://www.sciencedirect.com/science/article/pii/S209580991930503X`.

Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is Robustness the Cost of Accuracy? – A Comprehensive Study on the Robustness of 18 Deep Image Classification Models. *arXiv e-prints*, art. arXiv:1808.01688, August 2018.

Danilo Vasconcellos Vargas, Shashank Kotyan, and SPM IIIT-NR. Evolving robust neural architectures to defend from adversarial attacks. *arXiv preprint arXiv:1906.11667*, 2019.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv e-prints*, art. arXiv:2001.03994, January 2020.

Cihang Xie and Alan Yuille. Intriguing properties of adversarial training at scale. *arXiv e-prints*, art. arXiv:1906.03787, June 2019.

Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. Pc-darts: Partial channel connections for memory-efficient architecture search. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJlS634tPr.

Shen Yan, Biyi Fang, Faen Zhang, Yu Zheng, Xiao Zeng, Hui Xu, and Mi Zhang. HM-NAS: Efficient Neural Architecture Search via Hierarchical Masking. *arXiv e-prints*, art. arXiv:1909.00122, August 2019.

Antoine Yang, Pedro M. Esperança, and Fabio M. Carlucci. Nas evaluation is frustratingly hard. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HygrdpVKvr.

Barret Zoph and Quoc V. Le. Neural Architecture Search with Reinforcement Learning. *arXiv e-prints*, art. arXiv:1611.01578, November 2016.

A APPENDIX

Model	Clean %	FGSM	F-FGSM	PGD	HRS
ResNet-18	93.48	52.43	48.33	24.27	38.53
ResNet-50	94.38	50.05	45.78	23.45	37.57
DenseNet-121	94.76	50.94	47.14	24.06	38.38
DenseNet-169	94.74	53.53	49.47	**26.21**	**41.06**
VGG16 BN	94.07	52.42	46.16	20.03	33.03
DARTS	97.03	58.53	45.03	7.09	13.21
PDARTS	**97.12**	58.67	47.62	9.31	16.99
NSGA Net	96.94	**66.08**	56.16	11.1	19.92
Proxyless-NAS	97.92	51.73	**58.38**	3.22	6.23
PC-DARTS	97.05	60.55	48.65	9.84	17.87

Table 2: Comparison of clean accuracy and adversarial robustness on CIFAR-10 dataset (Top-1 Accuracy)

Model	Clean %	FGSM	F-FGSM	PGD	HRS
ResNet-18	63.87	17.08	17.12	6.05	11.05
ResNet-50	73.09	19	18.12	5.63	10.45
DenseNet-121	78.71	22.9	22.22	7.28	13.33
DenseNet-169	82.44	22.73	21.66	**7.37**	**13.53**
VGG16 BN	72.05	17.09	15.15	4.27	8.06
DARTS	82.43	24.91	16.34	2.32	4.51
PDARTS	83.07	27.69	20.23	3.09	5.96
NSGA Net	**85.44**	**34.93**	**24.1**	2.26	4.40
PC-DARTS	81.83	26.22	18.35	2.93	5.66

Table 3: Comparison of clean accuracy and adversarial robustness on CIFAR-100 dataset (Top-1 Accuracy)

Model	Clean %	FGSM	F-FGSM	PGD	HRS
ResNet18	89.08	32.75	18.03	2.41	4.70
ResNet50	92.86	46.28	26.22	4.68	8.90
DenseNet121	91.97	56.20	38.11	6.932	12.89
DenseNet169	92.81	**61.89**	**44.22**	**10.46**	**18.80**
VGG16	91.52	33.34	13.54	1.55	3.05
DARTS	91.26	54.41	31.18	2.94	5.70
P-DARTS	92.61	55.53	33.87	4.11	7.86
PC-DARTS	92.49	58.90	37.86	4.75	9.04
Proxyless-NAS	92.54	59.56	39.69	6.48	12.11
DenseNAS-Large	92.80	47.91	27.25	2.97	5.76
DenseNAS-R3	**93.81**	54.99	32.11	4.32	8.25

Table 4: Comparison of clean accuracy and adversarial robustness on ImageNet dataset (Top-5 Accuracy)

Model	Clean %	FGSM	F-FGSM	PGD	HRS
ResNet-18	95.48	54.33	51.16	11.23	20.10
ResNet-50	97.31	53.97	52.38	11.36	20.34
DenseNet-121	97.19	67.4	58.61	16	27.48
DenseNet-169	**97.44**	69.11	62.76	18.56	31.18
VGG16 BN	95.24	**72.16**	**66.06**	**27.59**	**42.78**
DARTS Liu et al. (2018)	95.97	64.47	59.95	19.29	32.12
PDARTS Chen et al. (2019)	95.12	55.31	51.16	9.52	17.31
NSGA Net Lu et al. (2018)	92.55	40.05	33.58	2.69	5.23
PC-DARTS Xu et al. (2020)	94.02	54.7	45.3	6.84	12.75

Table 5: Comparison of clean accuracy and adversarial robustness on Flowers-102 dataset (Top-1 Accuracy)

Family	Variant	Params (M)	Clean %	PGD	PP-HRS
Efficient-Net	B0	5.29	91.36	8.11	**14.90**
	B1	7.79	88.89	5.47	7.00
	B2	9.11	92.77	11.40	11.79
	B3	12.23	**93.04**	13.37	10.11
	B4	19.34	92.73	**16.99**	7.86
	B5	30.39	90.95	9.37	2.96
	B6	43.04	91.86	11.71	2.55
	B7	**66.35**	91.57	11.20	1.59
DenseNAS	A	4.77	90.94	1.84	3.61
	B	5.58	91.89	2.13	3.56
	C	6.13	92.31	2.29	3.48
	Large	**6.48**	**92.80**	**2.97**	**4.24**
	R1	11.09	91.33	2.01	**3.93**
	R2	19.47	92.47	3.19	3.51
	R3	**24.66**	**93.81**	**4.32**	3.71
ResNet	18	11.69	89.08	2.41	**4.69**
	50	**25.56**	**92.86**	**4.68**	4.08
DenseNet	121	7.98	91.97	6.93	**12.89**
	169	**14.15**	**92.81**	**10.46**	10.60

Table 6: Comparison of parameter count vs Adversarial accuracy for five different family of architectures on ImageNet dataset

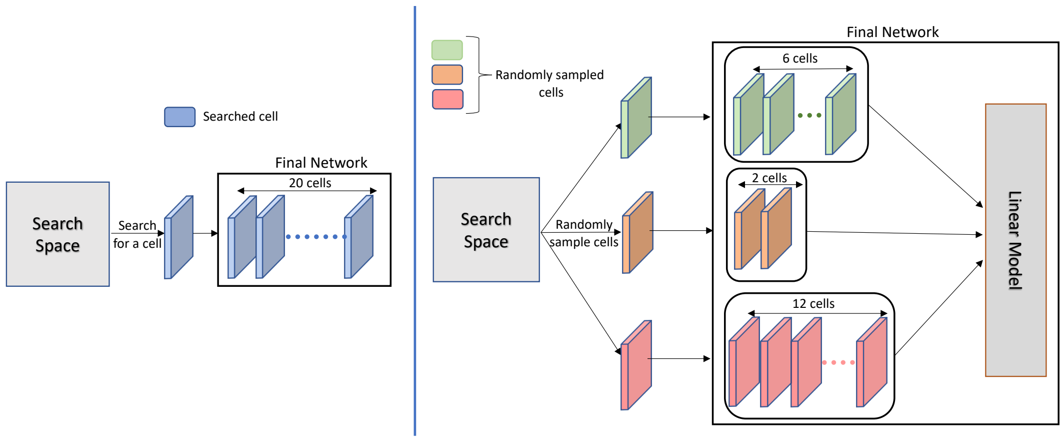

Figure 3: *Left:* Standard procedure for building architectures from DARTS search space; *Right:* Procedure for building ensembles using DARTS search space. 12, 6, 2 can be replaced with any values that sum to 20.

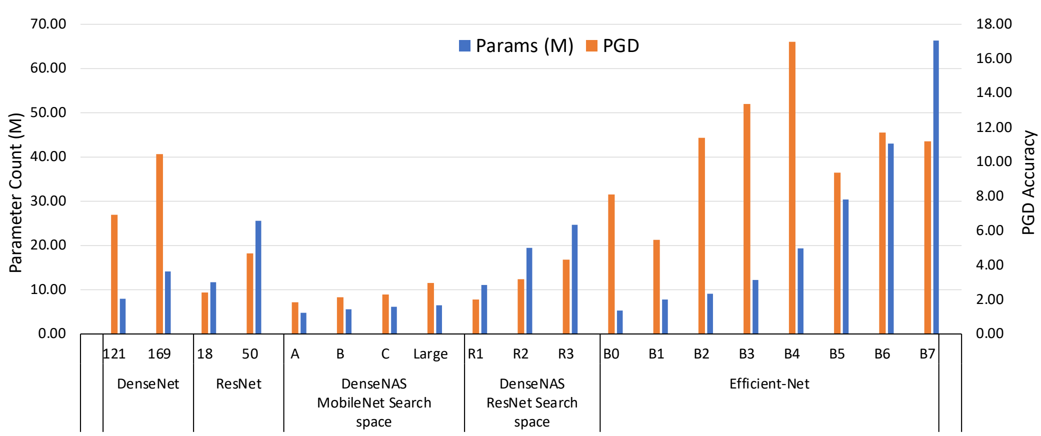

Figure 4: Comparison of PGD accuracy and Parameter count across different family of architectures