Neural Architecture Search with Loss Flatness-aware Measure

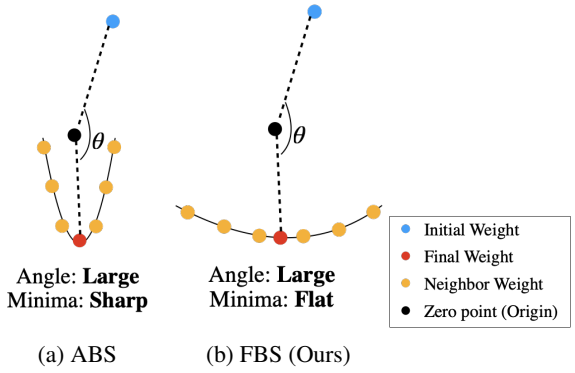
Joonhyun Jeong [1] **Joonsang Yu** [1 2] **Dongyoon Han** [2] **YoungJoon Yoo** [1 2]

Abstract

We propose a new proxy measure for Neural Architecture Search (NAS) focusing on the flatness of loss surface. One step forward to the existing NAS studies utilizing the validation-set accuracy or angle which measures convergence speed during training, we claim that the flatness of the loss surface can be a promising proxy for predicting the generalization capability of neural network architectures. To validate the claim, we formulate a novel approach of capturing the depth and flatness of the loss surface around local minima of a given network architecture. We demonstrate the effectiveness of the proposed method by performing experiments on various search spaces (NAS-Bench-201, DARTS), diverse datasets (CIFAR, ImageNet, MS-COCO), and various tasks such as object detection.

1. Introduction

Recently, Neural Architecture Search (NAS) (Baker et al., 2016; Liu et al., 2018; Real et al., 2019; Tan et al., 2019) has evolved to achieve remarkable accuracy along with the development of human-designed networks (Dosovitskiy et al., 2020; He et al., 2016; Tan & Le, 2019) on image recognition task. Several NAS methods (Chu et al., 2020; Hong et al., 2022; Zhang et al., 2021; Zoph et al., 2018) further demonstrated generalization ability (generalizability) of these automatically designed networks with high test accuracy performance and even with simply transferring the found architecture onto the other datasets. For the widespread leverage of architectures found by NAS on the other various tasks such as object detection (Lin et al., 2014) and segmentation (Cordts et al., 2016) (task-generalizability), investigating generalizability of each architecture candidate is a prerequisite and indispensable. Despite its importance, quantitative measuring of generalizability during architecture search process is still an open problem. In this paper, we discover an optimal proxy measure to discriminate generalizable architectures during search process.



(a) ABS (b) FBS (Ours)

Figure 1. Shape of local loss minima found by angle-based searching (ABS) and flatness-based searching (FBS). (a) Architecture found by ABS can not guarantee to be located within flat local minima. (b) FBS searches for architectures with flat local minima by inspecting loss values of local neighborhood weights.

Comparison	Kendall's Tau		
	CIFAR-10	CIFAR-100	ImageNet16-120
Angle & Flatness	0.4302	0.4724	0.4097
Accuracy & Flatness	0.7923	0.7568	0.7620

Table 1. Rank correlation between searching metrics on NAS-Bench-201 search space, showing relatively low-correlation between angle and flatness. We evaluated validation accuracy, angle, and flatness of all architectures and compared Kendall's Tau (Kendall, 1938) rank correlation between these searching metrics on CIFAR-10, CIFAR-100, and ImageNet16-120 (Chrabaszcz et al., 2017) dataset.

Most previous NAS algorithms including the pioneering differentiable search method, DARTS (Liu et al., 2018) and evolutionary search method, SPOS (Guo et al., 2020) use validation performance as a proxy measure for the generalizability as follows:

$$a^* = \underset{a \in A}{\operatorname{argmax}}\, S(a), \tag{1}$$

where a and A denote an architecture candidate and the entire search space, and $S(\cdot)$ represents a measurement function indicating the validation performance. Here, the measurement function is defined by accuracy (Guo et al., 2020), negative of loss value (Liu et al., 2018) on a validation dataset (For detailed formulation, see Appendix A.1).

[1]Image Vision, NAVER CLOVA [2]NAVER AI Lab. Correspondence to: YoungJoon Yoo <youngjoon.yoo@navercorp.com>.

DyNN workshop at the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, 2022.

These performance-based searching (PBS) methods suffer from overfitting on validation set, resulting in poor test-set generalization (Guo et al., 2020; Oymak et al., 2021; Zela et al., 2019; Zhang et al., 2021). The lack of generalizability for these PBS methods hinders broader usage of resultant found architecture on various tasks and datasets.

To explicitly search out generalizable architectures, recent literatures (Shu et al., 2019; Zhang et al., 2021) empirically observed that architectures with fast convergence during training tend to have high correlation with better test generalizability. Based on the empirical connection between convergence speed and generalization, RLNAS (Zhang et al., 2021) proposed an Angle-Based Searching (ABS) method, which uses angle (i.e. convergence speed) as a proxy performance measure during searching process. The ABS method defines the score function $S(a)$ in Eq (1) by measuring the angle between the initial weight parameters $W^0(a)$ and final weight parameters $W^f(a)$ of the architecture a. Consequently, they search for the architecture a^* which maximizes the angle (For detailed formulation, see Appendix A.2). ABS empirically demonstrated superior performance compared to PBS methods on various search spaces and datasets (Zhang et al., 2021). However, we argue that ABS still has a large headroom for better generalization in terms of flat (wide) local minima, which has been considered as one of the key signals for inspecting generalizability of a trained network (Cha et al., 2020; He et al., 2019; Keskar et al., 2016; Pereyra et al., 2017; Zhang et al., 2018).

Since ABS only concerns the convergence speed of an architecture regardless of its shape of local minima, architectures found by ABS with large angles can not be guaranteed to have flat local minima, as can be seen in figure 1. Meanwhile, architectures not chosen by ABS (i.e. small angle) might have better generalizability based on the flat property of loss minima. Table 1 corroborates that angle is indeed in short of correlation with flatness of local minima. Inspired by this weak implicit connection between angle and flatness of local minima, we propose a flatness-based searching method, namely FBS, that can either replace or further enhance ABS in terms of generalizability. The proposed FBS finds local minima having deep and flat loss surface, and we note that the inspection of flatness should also be re-examined as a key factor for securing architectures with better generalizability on NAS domain, as previous literatures showed its strong empirical connections between flatness of local minima and actual test generalization performance (Cha et al., 2020; Goyal et al., 2017; Hoffer et al., 2017; Jastrzębski et al., 2017; Keskar et al., 2016; Masters & Luschi, 2018; Smith & Le, 2017).

Based on the reportings, in this paper, we propose a novel flatness-based NAS framework, namely GeNAS, for better discriminating generalizability of architectures during searching. We demonstrate the superior generalizability of architectures found by our GeNAS on various datasets and downstream tasks such as object detection.

2. Method

2.1. GeNAS: Generalization-aware NAS with Flatness of local minima

Since SPOS (Guo et al., 2020) can flexibly embrace a new architecture search proxy measure owing to the decoupled training and searching process unlike gradient-based NAS such as DARTS, we construct our proposed search framework based on SPOS, dubbed GeNAS. GeNAS is aimed to search for network architectures with better generalization performance. To this end, we introduce a procedure for quantitatively estimating flatness of an architecture's converged minima as a search proxy measure $F_{val}(\cdot)$ as follows:

$$a^* = \underset{a \in A}{\operatorname{argmax}}\, F_{val}(W_A^*(a)). \tag{2}$$

From the previous studies (Cha et al., 2020; Zhang et al., 2018) empirically investigating the landscape of converged local minima, the neural networks having flat local minima where the changes of the validation loss around the local minima are relatively small show better generalization performance at test phase. Based on these simple but effective empirical connections, we introduce a novel method that searches for the architecture with maximal loss flatness around converged minima which can be formulated as:

$$F_{val}(\theta) = \Big(\sum_{i=1}^{t-1} \frac{L(\theta + N(\sigma_{i+1})) - L(\theta + N(\sigma_i))}{\sigma_{i+1} - \sigma_i}\Big)^{-1}, \tag{3}$$

where $L(\theta)$ denotes validation loss value given weight parameter θ abbreviating $W_A^*(a)$, and $N(\sigma_i)$ denotes random Gaussian noise with its mean and standard deviation being 0 and σ_i, respectively. The hyper-parameters σ denotes the range for inspection of flatness around the converged local minima, and t denotes the number of perturbations with Gaussian noise. We set different σ for each target search dataset since the optimal range of flatness around the local minima is different for each dataset. Empirically, we set the number of iterations t to three, which is enough to achieve a good trade-off between test performance and search costs. To perturb the weight parameters, we use unidirectional random noise, much simpler than recent flatness measuring approaches using Hessian (Yao et al., 2019) and bidirectional random noise (He et al., 2019) which can induce considerable amount of computational cost. We observed that our choice is sufficient to discriminate architectures with high test generalization performance as shown in Section 3.1. We point out that most similar works (Chen & Hsieh, 2020; Zela et al., 2019) to our method tackle to alleviate fluctuating loss surface and accuracy caused by the architecture parameters

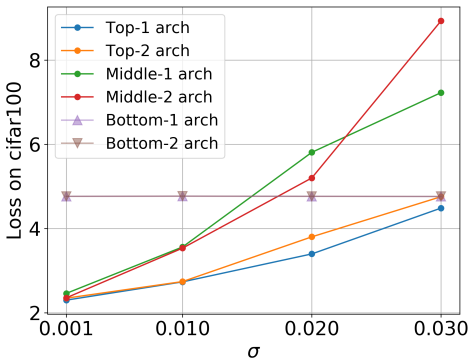


Figure 2. Validation loss curvatures of architectures sorted by the ground-truth test accuracy which is given by NAS-Bench-201 (Dong & Yang, 2020) on CIFAR-100.

from DARTS (Liu et al., 2018). Meanwhile, our method can be applied on any architecture search frameworks without dependence on architecture parameters of DARTS, such as evolutionary-based searching algorithm.

Eq (2) and (3) would find architecture a^* having the flattest local minima in the entire search space, but a^* can have sub-optimal local minima far from the global minimum. In Figure 2, two of the bottom architectures with the lowest ground-truth test accuracy given by NAS-Bench-201 show the flattest local minima with relatively large loss values compared to the middle and top architectures. Therefore, naive investigation of the flatness of an architecture possibly comes to achieve a sub-optimal architecture in terms of loss value. Note that the top architectures have the lowest loss values compared to middle and bottom architectures, equipped with flatness near converged minima. Correspondingly, considering both flatness of loss landscape and the depth of minima is essential for excavating a generalizable architecture. To implement the supposition, we add an additional term on Eq (3) to search for architectures with deep minima, along with flatness as follows:

$$F_{val}(\theta) = \Big(\sum_{i=1}^{t-1} \Big|\frac{L(\theta + N(\sigma_{i+1})) - L(\theta + N(\sigma_i))}{\sigma_{i+1} - \sigma_i}\Big| + \alpha\Big|\frac{L(\theta + N(\sigma_1))}{\sigma_1}\Big|\Big)^{-1} \tag{4}$$

Here, σ_1 denotes the smallest perturbation degree among σ, hence the second term inspects how low the loss value nearest converged minima is. The term α denotes the balancing coefficient term between flat and deep minima, which is set to 1 unless specified.

2.2. Searching with Combined Metrics

Recent works (Hosseini et al., 2021; Mellor et al., 2021) adopted a combined search metric for enhancing the performance of the resultant architecture. Hosseini et al. (2021) employed an integrated search metric where the conventional cross-entropy loss over a clean image is combined with approximately measured adversarial robustness lower bound to enhance test accuracy of both clean images and adversarially attacked images. Inspired by the weak correlation between existing search metrics (e.g. angle, validation accuracy) and flatness (Table 1), we target to explicitly fulfill the large headroom of conventional search metrics to find better generalizable architectures in terms of our proposed flatness-based search measure (Eq (4)). Formally, we combine existing metrics with flatness as a search proxy measure as follows:

$$a^* = \underset{a \in A}{\operatorname{argmax}}\, S(W_A^*(a)) + \gamma\beta F_{val}(W_A^*(a)) \tag{5}$$

where S denotes conventional search metrics such as angle and validation accuracy, γ is a balancing parameter between existing metric and flatness, and β is a normalization term, which is fixed as σ_1^{-1}, for matching scale of flatness term with existing search metric.

Searching Metric	Params (M)	FLOPs (G)	Top-1 Acc (%)	Top-5 Acc (%)
Angle	5.4	0.61	75.00	92.31
Accuracy	5.4	0.60	75.37	92.23
Flatness	**5.2**	**0.58**	**76.05**	**92.64**
Angle + Accuracy	5.6 (+0.2)	0.62 (+0.01)	75.53 (+0.53)	92.61 (+0.30)
Angle + Flatness	5.4 (+0.0)	0.60 (-0.01)	75.72 (+0.72)	92.46 (+0.15)
Accuracy + Flatness	5.4 (+0.0)	0.60 (+0.00)	75.85 (+0.48)	92.74 (+0.51)

Table 2. Transferability of various searching metrics from CIFAR-100 onto ImageNet. The quantities in the parentheses denote the amount of change induced by the integrated metric from the baseline metric. Improvements from integrating $Flatness$ term is denoted with blue color.

3. Experiments

In this section, we evaluate our proposed GeNAS framework on various search spaces including DARTS (Liu et al., 2018) and NAS-Bench-201 (Dong & Yang, 2020) (See Appendix C.1) with widely-used benchmark datasets such as CIFAR-10/100 and ImageNet. We also tested transferability of our excavated architectures onto other task domain, object detection, with MS-COCO (Lin et al., 2014) dataset.

3.1. Search Results on The DARTS Search Space

In Table 2, we analyze transferability of architectures found on small datasets such as CIFAR onto ImageNet, with DARTS search space. The results show that flatness consistently reports significantly superior searching performance even with fewer flops and parameters compared to ABS or PBS metrics, about 1.05% and 0.68% better top-1 accuracy, respectively. Furthermore, when flatness is combined with angle and accuracy as a search proxy measure, top-1 accuracy increases by 0.72% and 0.48%, respectively, which

Search Dataset	Method	Params (M)	FLOPs (G)	Top-1 Acc (%)	Top-5 Acc (%)
CIFAR-10	DARTS (Liu et al., 2018)	4.7	0.57	73.30	91.30
	PC-DARTS (Xu et al., 2019)	5.3	0.59	74.90	92.20
	FairDARTS-B (Chu et al., 2020)	4.8	0.54	75.10	92.50
	SPOS (Guo et al., 2020)	5.4	0.60	75.32	92.20
	SDARTS-RS (Chen & Hsieh, 2020)	5.5	0.61	75.52	92.66
	SDARTS-ADV (Chen & Hsieh, 2020)	5.5	0.62	75.61	92.39
	P-DARTS (Chen et al., 2019)	4.9	0.56	75.60	92.60
	RLNAS (Zhang et al., 2021)	5.3	0.59	75.70	92.45
	DropNAS† (Hong et al., 2022)	5.4	0.60	75.98	92.80
	GeNAS (Flatness)	5.6	0.61	75.95	92.74
	GeNAS (Angle + Flatness)	**5.3**	**0.59**	**76.06**	**92.77**
CIFAR-100	PC-DARTS (Xu et al., 2019)	5.3	0.59	74.75	92.16
	RLNAS (Zhang et al., 2021)	5.4	0.61	75.00	92.31
	DropNAS† (Hong et al., 2022)	5.1	0.57	75.07	92.33
	P-DARTS (Chen et al., 2019)	5.1	0.58	75.30	92.50
	SPOS (Guo et al., 2020)	5.4	0.60	75.37	92.23
	GeNAS (Flatness)	**5.2**	**0.58**	**76.05**	**92.64**
	GeNAS (Angle + Flatness)	5.4	0.60	75.72	92.46

Table 3. ImageNet performance comparison of SOTA NAS methods searched with DARTS search space on CIFAR-10 and CIFAR-100 dataset. † denotes that SE (Hu et al., 2018) module is excluded for fair comparison with other methods.

Method	Params (M)	FLOPs (G)	AP	AP_{50}	AP_{70}	AP_S	AP_M	AP_L
PC-DARTS (Xu et al., 2019)	5.3	0.59	35.56	55.50	37.45	19.85	38.80	47.70
RLNAS (Zhang et al., 2021)	5.4	0.61	35.98	55.78	38.22	20.80	39.72	47.90
SPOS (Guo et al., 2020)	5.4	0.60	36.04	56.30	38.08	20.01	39.49	47.76
DropNAS (Hong et al., 2022)	5.1	0.57	36.39	56.14	38.45	**21.88**	39.82	48.20
GeNAS (Flatness)	5.2	0.58	**37.05**	**56.92**	**39.19**	20.70	**40.68**	**49.74**
GeNAS (Angle + Flatness)	5.4	0.60	36.59	56.37	38.79	21.43	39.94	49.02

Table 4. Object detection performance comparison of SOTA NAS methods on MS COCO dataset.

was consistently shown in case of searching with CIFAR-10 (See Appendix C.2).

3.2. Comparison with SOTA NAS methods

In Table 3, our GeNAS clearly represents large headroom compared to the other SOTA NAS methods. Especially in comparison with SDARTS (Chen & Hsieh, 2020) which is a similar approach to GeNAS by using an implicit regularization for smoothing accuracy landscape, our GeNAS outperforms with comparable number of FLOPs. Table 2 and 3 results show that our proposed flatness search metric indeed serves as a powerful search proxy measure for finding well-transferable architectures and also enhances the other search metrics to have stronger ability to find architectures with better test generalization performance.

3.3. Task Generalization Ability

We evaluate the generalization capability of architectures found by GeNAS on downstream task, specifically object detection. We firstly re-train architectures found on CIFAR-100 onto ImageNet, and finetune on MS-COCO (Lin et al., 2014) dataset. For training, we adopt default training strategy of RetinaNet (Lin et al., 2017) from Detectron2 (Wu et al., 2019). We only replace the backbone network of RetinaNet for analyzing sole impact of architectures found by each NAS method in terms of generalization ability across various task domains. Table 4 shows that our GeNAS framework guided by flatness measure achieves the best AP scores without bells and whistles. In case of RLNAS (angle) combined with flatness as a search metric, AP is enhanced by about 0.6%, without increase of FLOPs or number of parameters.

4. Conclusion

This paper demonstrates that flatness of local minima can be directly employed as a proxy of discriminating and searching for generalizable architectures. Based on quantitative benchmark experiments on various search spaces and datasets, we demonstrate the superior generalizability of our flatness-based searching over conventional search metrics, while showing comparable or even better searching performance compared to recent state-of-the-art NAS frameworks. We further analyze insufficient generalizability of conventional search metrics in terms of flatness of local minima. Consequently, integrating conventional search metrics with our proposed flatness measure can further lead to significantly boosting the searching performance. We also demonstrate superior generalization capability of GeNAS on the downstream object detection task, compared to other search metrics and SOTA NAS methods.

References

Baker, B., Gupta, O., Naik, N., and Raskar, R. Designing neural network architectures using reinforcement learning. arXiv preprint arXiv:1611.02167, 2016. 1

Cha, S., Hsu, H., Hwang, T., Calmon, F. P., and Moon, T. Cpr: Classifier-projection regularization for continual learning. arXiv preprint arXiv:2006.07326, 2020. 2

Chen, X. and Hsieh, C.-J. Stabilizing differentiable architecture search via perturbation-based regularization. In International conference on machine learning, pp. 1554–1565. PMLR, 2020. 2, 4

Chen, X., Xie, L., Wu, J., and Tian, Q. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1294–1303, 2019. 4

Chrabaszcz, P., Loshchilov, I., and Hutter, F. A downsampled variant of imagenet as an alternative to the cifar datasets. arXiv preprint arXiv:1707.08819, 2017. 1, 7

Chu, X., Zhou, T., Zhang, B., and Li, J. Fair darts: Eliminating unfair advantages in differentiable architecture search. In European conference on computer vision, pp. 465–480. Springer, 2020. 1, 4

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3213–3223, 2016. 1

Dong, X. and Yang, Y. Nas-bench-201: Extending the scope of reproducible neural architecture search. arXiv preprint arXiv:2001.00326, 2020. 3

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020. 1

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv preprint arXiv:1706.02677, 2017. 2

Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., and Sun, J. Single path one-shot neural architecture search with uniform sampling. In European Conference on Computer Vision, pp. 544–560. Springer, 2020. 1, 2, 4, 7, 8

He, H., Huang, G., and Yuan, Y. Asymmetric valleys: Beyond sharp and flat local minima. Advances in neural information processing systems, 32, 2019. 2

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016. 1

Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. Advances in neural information processing systems, 30, 2017. 2

Hong, W., Li, G., Zhang, W., Tang, R., Wang, Y., Li, Z., and Yu, Y. Dropnas: Grouped operation dropout for differentiable architecture search. arXiv preprint arXiv:2201.11679, 2022. 1, 4

Hosseini, R., Yang, X., and Xie, P. Dsrna: Differentiable search of robust neural architectures. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6196–6205, 2021. 3

Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132–7141, 2018. 4

Jastrzębski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. Three factors influencing minima in sgd. arXiv preprint arXiv:1711.04623, 2017. 2

Kendall, M. G. A new measure of rank correlation. Biometrika, 30(1/2):81–93, 1938. 1

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. arXiv preprint arXiv:1609.04836, 2016. 2

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In European conference on computer vision, pp. 740–755. Springer, 2014. 1, 3, 4

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pp. 2980–2988, 2017. 4

Liu, H., Simonyan, K., and Yang, Y. Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055, 2018. 1, 3, 4, 8

Masters, D. and Luschi, C. Revisiting small batch training for deep neural networks. arXiv preprint arXiv:1804.07612, 2018. 2

Mellor, J., Turner, J., Storkey, A., and Crowley, E. J. Neural architecture search without training. In International Conference on Machine Learning, pp. 7588–7598. PMLR, 2021. 3

Oymak, S., Li, M., and Soltanolkotabi, M. Generalization guarantees for neural architecture search with train-validation split, 2021. 2

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., and Hinton, G. Regularizing neural networks by penalizing confident output distributions. arXiv preprint arXiv:1701.06548, 2017. 2

Real, E., Aggarwal, A., Huang, Y., and Le, Q. V. Regularized evolution for image classifier architecture search. In Proceedings of the aaai conference on artificial intelligence, volume 33, pp. 4780–4789, 2019. 1

Shu, Y., Wang, W., and Cai, S. Understanding architectures learnt by cell-based neural architecture search. arXiv preprint arXiv:1909.09569, 2019. 2

Smith, S. L. and Le, Q. V. A bayesian perspective on generalization and stochastic gradient descent. arXiv preprint arXiv:1710.06451, 2017. 2

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning, pp. 6105–6114. PMLR, 2019. 1

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V. Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2820–2828, 2019. 1

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 4

Xu, Y., Xie, L., Zhang, X., Chen, X., Qi, G.-J., Tian, Q., and Xiong, H. Pc-darts: Partial channel connections for memory-efficient architecture search. arXiv preprint arXiv:1907.05737, 2019. 4, 8

Yao, Z., Gholami, A., Keutzer, K., and Mahoney, M. Pyhessian: Neural networks through the lens of the hessian. arXiv preprint arXiv:1912.07145, 2019. 2, 12

Zela, A., Elsken, T., Saikia, T., Marrakchi, Y., Brox, T., and Hutter, F. Understanding and robustifying differentiable architecture search. arXiv preprint arXiv:1909.09656, 2019. 2

Zhang, X., Hou, P., Zhang, X., and Sun, J. Neural architecture search with random labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10907–10916, 2021. 1, 2, 4, 7, 8

Zhang, Y., Xiang, T., Hospedales, T. M., and Lu, H. Deep mutual learning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4320–4328, 2018. 2

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8697–8710, 2018. 1

A. Formulation of PBS and ABS

This section describes the detailed formulation of PBS method and ABS method, respectively.

A.1. PBS: Single Path One-Shot NAS.

In order to decouple the biased connection between weight parameters of SuperNet and its architecture parameters, Single Path One-Shot NAS (SPOS) (Guo et al., 2020) sequentially optimizes the weight parameters of SuperNet and select SubNet that shows the **highest validation accuracy** where a SubNet is subsampled from SuperNet. Formally, entire training and search processes are described as:

$$W_A^* = \underset{W_A}{\operatorname{argmin}}\, \mathbb{E}_{a\sim U(A)} L_{train}(a, W_A), \tag{6}$$

$$a^* = \underset{a\in A}{\operatorname{argmax}}\, S_{val}(W_A^*(a)), \tag{7}$$

where a denotes the SubNet architecture inherited from the SuperNet architecture A, where W_A denotes the weight parameters of SuperNet. $W_A(a)$ denotes the weight of the architecture a inherited from the SuperNet weight W_A. $U(A)$ denotes the uniform distribution for sampling a from A. In Eq (6), the SubNet weight parameters $W_A(a)$ selected by random-uniformly are optimized, giving all the SubNets $a \sim U(A)$ to be uniformly optimized. In Eq (7), given the trained SuperNet weight parameters W_A^*, each SubNet candidate $a \in A$ is evaluated by the architecture score measurement function $S_{val}(\cdot)$, here defined as accuracy on validation set, and consequently the architecture having highest accuracy a^* is selected. Specifically, SPOS utilizes evolutionary searching algorithm for Eq (7), where top-K populations sorted by validation accuracy repeatedly *Cross-Over* with each other and *Mutate* itself to search for the architecture having better validation accuracy. As SPOS decouples the SuperNet training and architecture searching process, it becomes flexible in using a new search proxy measure for Eq (7).

A.2. ABS: Random Label NAS.

Random Label NAS (RLNAS) (Zhang et al., 2021) proposed a new architecture score measurement function $S_{angle}()$, which uses an angle as a search proxy measure rather than validation accuracy, as follows:

$$S_{angle}(W_A^*(a)) = acos(W_A^0(a) \cdot W_A^f(a) / (\parallel W_A^0(a) \parallel_2 \parallel W_A^f(a) \parallel_2)), \tag{8}$$

where $W_A^0(a)$ and $W_A^f(a)$ denotes initial weight parameters of SubNet a before training and final weight parameters of a after training is finished, respectively. The symbol $\cdot$ denotes the inner product. For angle estimation, $W_A^0(a)$ and $W_A^f(a)$ are both vectorized by flattening all the weight parameters of $W_A(a)$ into one-dimensional vectors and concatenating these weight vectors. Therefore, $S_{angle}()$ indicates angle between initial and converged weight parameters, which also means the convergence speed of each architecture. RLNAS claimed that the high convergence speed is correlated with the test generalization performance, so angle can be used for the proxy measure of architecture evaluation. However, we point out that RLNAS still lacks awareness of generalization in terms of flatness of local minima (Figure 1 and Table 1). Therefore, we conjecture that the generalizability of RLNAS can be further enriched through explicitly combining the angle metric with a flatness-aware measure.

B. Training and Searching Setups

This section describes detailed experimental setups for training and searching on NAS-Bench-201 and DARTS search space.

B.1. NAS-Bench-201 search space.

NAS-Bench-201 provides a relatively small search space where 5 edges with 6 possible operation candidates compose a directed acyclic graph cell, thus the number of architecture candidates from the entire search space is $5^6 = 15625$. Using the ground truth test accuracy of all of the candidate architectures from NAS-Bench-201, we measure Kendall's Tau score by comparing rank correlation between search proxy measure and those from NAS-Bench-201. We use the equivalent settings to NAS-Bench-201 for constructing training / validation / test set of CIFAR-10, CIFAR-100, and ImageNet16-120 (Chrabaszcz et al., 2017). For training SuperNet, we use the same training settings (e.g. SGD optimizer with $5e^{-4}$ weight decay factor,

250 training epochs, cosine learning rate scheduling annealed from 0.025 to 0.001) from RLNAS (Zhang et al., 2021). During evolutionary searching, we set the entire size of population as 100 with 20 evolution iterations, following RLNAS. For investigating the loss landscape near converged minima, we set $\sigma = \{2e-3, 1e-2, 2e-2\}$ as default unless specified.

B.2. DARTS search space.

DARTS (Liu et al., 2018) has a larger search space than NAS-Bench-201, which provides 8 edges with 7 possible operation candidates (excluding zero operation). Furthermore, reduction cell (stride $= 2$) is also included in search target, further broadening the search space and increasing the difficulty of searching. We evaluate each NAS method by searching architectures on proxy datasets such as CIFAR-10 and CIFAR-100. For the selected architectures, we train each model on ImageNet from scratch and measure the top-1 accuracy. Following RLNAS, we set the number of cells in SuperNet as 8 and train 250 epochs. We divide the original training set into training / validation set with equal size on CIFAR-10/100, as in DARTS (Liu et al., 2018) and PC-DARTS (Xu et al., 2019). During evolutionary searching, we set the entire size of population as 50 with 20 evolution iterations, following SPOS (Guo et al., 2020). We set $\sigma = \{1e-5, 5e-5, 1e-4\}$, $\{1e-3, 3e-3, 6e-3\}$ for searching on CIFAR-10 and CIFAR-100, respectively. For scratch training on ImageNet, we adjust initial channels of a target network to have FLOPs around 0.6G. We set the training hyper-parameters exactly same as PC-DARTS with 8 V100 GPUs.

C. Experiments

This section describes additional experiments on various search spaces such as NAS-Bench-201 and DARTS, with ablation studies for components of our proposed GeNAS framework.

C.1. Search Results on NAS-Bench-201

Searching Metric	Kendall's Tau		
	CIFAR-10	CIFAR-100	ImageNet16-120
Angle	0.6671	0.6942	0.6342
Accuracy	0.5701	0.5394	0.5411
Flatness	0.6047	0.5918	0.5800
Angle + Accuracy	0.7539 (+0.0868)	0.7004 (+0.0062)	0.6895 (+0.0553)
Angle + Flatness	0.7636 (+0.0965)	0.7619 (+0.0677)	0.7368 (+0.1026)
Accuracy + Flatness	0.6023 (+0.0322)	0.5658 (+0.0264)	0.5657 (+0.0246)

Table 5. Kendall's Tau of various searching proxy metrics on NAS-Bench-201. The quantities in the parentheses denote the amount of Kendall's Tau changed by the integrated metric from the baseline metric. Improvements from integrating *Flatness* term is denoted with blue color.

C.1.1. Stand-alone metric performance.

We compare search performance of various search proxy metrics such as angle, validation accuracy, and our proposed flatness of minima in Table 5. We only replaced the architecture fitness indicator with the above mentioned metrics during evolutionary searching, and measured Kendall's Tau considering the entire architecture candidates' rank correlation. In Table 5, ABS shows the best searching performance with the highest Kendall's Tau score, representing the powerful test generalizability on small scale datasets such as CIFAR and ImageNet16-120. However, we highlight that angle is still insufficient for achieving better test generalizability in terms of flatness on local minima (Table 1), and we observe significant improvement on searching performance when flatness is granted to the local-minima of the architecture found from ABS.

C.1.2. Combined metric performance.

Inspired by the weak connection between angle metric and flatness of local minima, shown in Table 1, we test a integrated search proxy measure where angle is explicitly combined with flatness for better generalization. In Table 5, angle metric combined with flatness as a search metric shows significantly enhanced Kendall's Tau rank correlation compared to that of stand-alone angle case on CIFAR-10/100 and ImageNet16-120. The improvements show that our FBS can drive ABS to better discriminate generalizable architectures by seeking not only its fast convergence speed but also flatness of curvature near converged local minima.

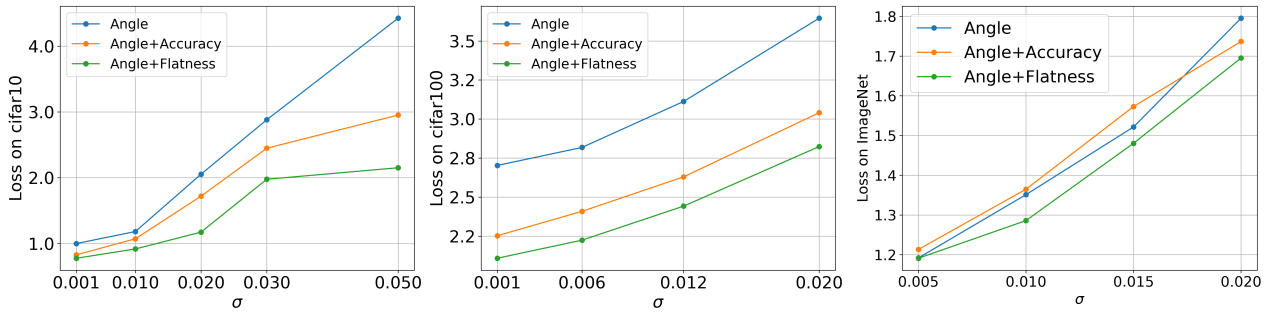


Figure 3. Test loss curvatures of architectures found by *Angle*, *Angle+Accuracy*, *Angle+Flatness* on CIFAR-10/100 (left, center) with the NAS-Bench-201 search space and ImageNet (right) with the DARTS search space. For ImageNet results, we transferred architectures found from CIFAR-10 onto ImageNet.

	$\mu = 0$	$\mu = 0.05$	$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.5$	$\mu = 1$
Kendall's Tau (CIFAR-10)	0.7539	0.7531	0.7180	0.6539	0.5592	0.5262

Table 6. Kendall's Tau trend on CIFAR-10 for the $Angle + Accuracy + Sharpness$ combination to investigate the impact of the inherent flatness characteristic of accuracy-based searching. μ denotes balancing coefficient for the sharpness term (i.e., the negation of the flattness metric), where $\mu = 0$ denotes $Angle + Accuracy$ case. The result shows that sharpening (unflattening) the loss surface harms the Kendall's Tau.

Meanwhile, it is worth noting that appending the validation accuracy metric onto the angle metric also improved Kendall's Tau on all the dataset. PBS (i.e., accuracy-based search) has shown strong correlation with FBS (Table 1), which means there exists a potential ability in PBS to discriminate flat or non-flat architectures on NAS-Bench-201 search space. Therefore, we conjecture that the possible reason for the improvements for angle combined with accuracy comes from this inherent flatness searching ability of PBS. To demonstrate this hypothesis, we explicitly harm the inherent flatness characteristic of PBS by adding sharpness term (the negation of the flatness term in Eq (4)) on *Angle+Accuracy*, and the searching performance becomes significantly degraded as the weight for sharpness term increases (Table 6). Furthermore, Figure 3 (left, center subfigure) shows that the found architecture from angle combined with accuracy indeed shows more smoothed test loss landscape near converged minima compared to that of angle case. We note that angle combined with flatness induces the flattest architecture equipped with the deepest minima compared to the angle or angle with accuracy case, achieving the largest performance gain as shown in Table 5.

C.2. Searching on CIFAR-10 with DARTS Search Space

We compare our proposed FBS with other search metrics on CIFAR-10 in Table 7. As a stand-alone search metric, flatness measure shows the best searching performance among the other metrics including accuracy and angle with comparable FLOPs ($\sim=$ 0.6G) and parameters, when transferring searched architecture from CIFAR-10 onto ImageNet. Furthermore, when angle is combined with flatness, loss landscape of found architecture becomes to be more flat and deeper as shown in right subfigure of Figure 3. As a result, searching performance is further improved by 0.36% top-1 accuracy without any increase of either FLOPs or parameters. Also, the accuracy-based proxy measure also achieved performance gain when flatness is combined. Meanwhile, it is noted that angle combined with accuracy only harms top-1 accuracy slightly. We conjecture that the inherent flatness property of PBS shown in small search spaces such as NAS-Bench-201 (Table 1) might not be consistently secured on large search space including DARTS as shown in right subfigure from Figure 3, resulting in slight performance degradation. In Table 3, GeNAS (*Flatness*, *Angle+Flatness*) show the best top-1 accuracy under FLOPs being $\sim=$ 0.6G setting, compared to the other state-of-the-art NAS methods.

Searching Metric	Params (M)	FLOPs (G)	Top-1 Acc (%)	Top-5 Acc (%)
Angle	5.3	0.59	75.70	92.45
Accuracy	5.4	0.60	75.32	92.20
Flatness	**5.6**	**0.61**	**75.95**	**92.74**
Angle + Accuracy	5.5 (+0.2)	0.61 (+0.02)	75.62 (-0.08)	92.55 (+0.10)
Angle + Flatness	5.3 (+0.0)	0.59 (+0.00)	76.06 (+0.36)	92.77 (+0.32)
Accuracy + Flatness	5.6 (+0.2)	0.61 (+0.01)	75.72 (+0.40)	92.59 (+0.39)

Table 7. Transferability of various searching metrics from CIFAR-10 onto ImageNet. The quantities in the parentheses denote the amount of change induced by the integrated metric from the baseline metric. Improvements from integrating $Flatness$ term is denoted with blue color.

C.3. Ablation Study

To better analyze our proposed FBS-based GeNAS framework, we conduct ablation study of each component and hyper-parameters consisting GeNAS.

C.3.1. Flatness range.

We analyze the effect of range of inspecting flatness near converged local minima in Table 8. The results demonstrate that searching flat-architectures within too small area near converged minima (1st row in Table 8) is not sufficient for discriminating generalizable architectures. When σ is set to $\{2e-3, 1e-2, 2e-2\}$, Kendall's Tau is considerably improved, while further widening the flatness inspection range (4th row in Table 8) only significantly degrades the searching performance on various datasets.

σ	Kendall's Tau		
	CIFAR-10	CIFAR-100	ImageNet16-120
$\{1e-6, 5e-6, 1e-5\}$	0.5756	0.5496	0.5524
$\{5e-4, 1e-3, 2e-3\}$	0.5770	0.5503	0.5531
$\{2e-3, 1e-2, 2e-2\}$	0.6047	0.5918	0.5800
$\{2e-3, 2e-2, 4e-2\}$	0.5416	0.3404	0.2364

Table 8. Kendall's Tau on the NAS-Bench-201 search space according to the perturbation range σ, inspecting the effect of flatness range near local minima.

C.3.2. Deep and low minima.

We further investigate the effect of searching architectures equipped with not only flatness but also depth of loss landscape near converged minima. Specifically, we adjust α in Eq (4), where $\alpha = 0$ denotes searching with only flatness of local minima. Results on Table 9 demonstrate that as α value increases from zero to one, searching performance is drastically enhanced, indicating the indispensability of searching with both flatness and depth of minima. Note that $\alpha = 0$ case can search out a sub-optimal architecture that has largely flat loss curvature but its loss values near local minima are too high, as shown in Figure 2. When α is further increased to $\alpha > 1$, Kendall's Tau rank correlation starts to decrease, denoting that searching with largely depending on depth of converged minima is not optimal for discriminating better generalizable architectures.

	$\alpha=0$	$\alpha=0.1$	$\alpha=0.5$	$\alpha=1$	$\alpha=2$	$\alpha=4$	$\alpha=8$	$\alpha=16$
Kendall's Tau (CIFAR-10)	0.1777	0.4026	0.5890	0.6047	0.5964	0.5898	0.5847	0.5820

Table 9. Kendall's Tau on CIFAR-10 with different α in Eq (4).

C.3.3. Effect of FBS on ABS.

We analyze effect of integrating flatness on ABS. Specifically, we adjust γ in Eq (5), which balances coefficient concerning about the ratio of flatness to angle term. In Table 10, integrating flatness with a small proportion to angle mildly improves top-1 accuracy. As γ increases, top-1 accuracy of searched architecture gradually increases as to reach 0.72% improvement over $\gamma = 0$ (ABS) case, with comparable FLOPs and parameters.

γ	$Flatness$ (%)	Params (M)	FLOPs (G)	Top-1 Acc (%)	Top-5 Acc (%)
0	0	5.43	0.61	75.00	92.31
0.5	20	5.45 (+0.02)	0.60 (-0.01)	75.22 (+0.22)	92.39 (+0.08)
1.5	43	5.57 (+0.14)	0.61 (+0.00)	75.58 (+0.58)	92.44 (+0.13)
6	76	5.41 (-0.02)	0.60 (-0.01)	75.63 (+0.63)	92.54 (+0.23)
16	89	5.41 (-0.02)	0.60 (-0.01)	75.72 (+0.72)	92.46 (+0.15)

Table 10. Searching performance of $Angle + Flatness$ with different γ values, where searched on CIFAR-100 and transferred onto ImageNet. $Flatness$ (%) denotes the average ratio of $Flatness$ compared to $Angle$ during evaluation of architectures on evolutionary searching algorithm. The quantities in the parentheses denote the amount of change compared to the $\gamma = 0$ case.

C.3.4. Effect of FBS on PBS.

We analyze effect of integrating our proposed FBS on PBS in Table 11. Integrating flatness with a small proportion shows comparable top-1 and top-5 accuracy compared to PBS ($\gamma = 0$ case). As γ increases, top-1 accuracy of searched architecture also increases as to reach 0.48% improvement compared to PBS without any change of FLOPs and number of parameters.

γ	$Flatness$ (%)	Params (M)	FLOPs (G)	Top-1 Acc (%)	Top-5 Acc (%)
0	0	5.4	0.60	75.37	92.23
0.25	10	5.3 (-0.1)	0.59 (-0.01)	75.34 (-0.03)	92.37 (+0.14)
2	41	5.5 (+0.1)	0.61 (+0.01)	75.26 (-0.11)	92.34 (+0.11)
8	75	5.5 (+0.1)	0.60 (+0.00)	75.60 (+0.23)	92.36 (+0.13)
32	92	5.4 (+0.0)	0.60 (+0.00)	75.85 (+0.48)	92.74 (+0.51)

Table 11. Searching performance of $Accuracy + Flatness$ with different γ values, where searched on CIFAR-100 and transferred onto ImageNet. $Flatness$ (%) denotes the average ratio of $Flatness$ compared to $Accuracy$ during evaluation of architectures on evolutionary searching algorithm. The quantities in the parentheses denote the amount of change compared to the $\gamma = 0$ case.

C.3.5. Effect of PBS on ABS.

We further analyze effect of integrating PBS on ABS in Table 12. Integrating PBS with a small proportion on ABS improves top-1 accuracy of ABS. However, as the proportion of PBS increases, top-1 accuracy of searched architecture becomes to be comparable or even degraded compared to that of ABS ($\gamma_{Acc} = 0$ case).

γ_{Acc}	$Accuracy$ (%)	Params (M)	FLOPs (G)	Top-1 Acc (%)	Top-5 Acc (%)
0	0	5.4	0.61	75.00	92.31
0.1	12	5.6 (+0.2)	0.62 (+0.01)	75.32 (+0.32)	92.38 (+0.07)
0.5	41	5.3 (-0.1)	0.59 (-0.02)	74.69 (-0.31)	92.05 (-0.26)
2.5	78	5.5 (+0.1)	0.61 (+0.00)	74.26 (-0.74)	91.67 (-0.64)
10	93	5.5 (+0.1)	0.61 (+0.00)	75.05 (+0.05)	92.13 (-0.18)

Table 12. Searching performance of $Angle + Accuracy$ with different γ_{Acc} values (balancing parameter for $Accuracy$), where searched on CIFAR-100 and transferred onto ImageNet. $Accuracy$ (%) denotes the average ratio of $Accuracy$ compared to $Angle$ during evaluation of architectures on evolutionary searching algorithm. The quantities in the parentheses denote the amount of change compared to the $\gamma_{Acc} = 0$ case.

C.3.6. PERTURBATION METHODOLOGY.

To quantitatively measure flatness of loss landscape, all the weight parameters of a given network are perturbed with random direction following Gaussian distribution as in Eq (7) in the manuscript. Here, we investigate the effect of perturbation positions and directions. In Table 13, perturbing only weight parameters of target search cells (i.e. excluding stem *conv* layer and final *fully-connected* layer) only harms Kendall's Tau. Moreover, with regard to the perturbation directions, strongly perturbing the given models' parameters across the hessian eigen-vectors (Yao et al., 2019) suffers from a slight decrease of Kendall's Tau (Table 13) with large computational overhead induced by approximation of hessian.

Perturbation Position	Perturbation Direction	Kendall's Tau
All	Random	0.6047
Search Cells	Random	0.5612 (-0.0435)
All	Hessian	0.5908 (-0.0139)

Table 13. Ablation study of perturbation position and direction on CIFAR-10 with NAS-BENCH-201 search space. *All* denotes perturbing all the weight parameters of a given network, while *Search Cells* denotes perturbing only weight parameters of search cells. The quantities in the parentheses denote the amount of change compared to the default case (first row).