PA&DA: Jointly Sampling PAth and DAta for Consistent NAS - Supplementary Materials

Shun Lu[1, 2], Yu Hu[1, 2]*, Longxing Yang[1, 2], Zihao Sun[1, 2], Jilin Mei[1], Jianchao Tan[3], Chengru Song[3]
[1] Research Center for Intelligent Computing Systems,
Institute of Computing Technology, Chinese Academy of Sciences
[2] School of Computer Science and Technology, University of Chinese Academy of Sciences
[3] Kuaishou Technology
{lushun19s, huyu, yanglongxing20b, sunzihao18z, meijilin}@ict.ac.cn,
{jianchaotan, songchengru}@kuaishou.com

A. Detailed calculation of the supernet gradient variance

During the supernet training, we record the gradient d_w of each parameter $w \in \mathcal{W}$ after each training step, using the gradient generated by the normal back-propagation. After an epoch of training, we utilize the recorded information to compute the gradient variance σ_w^2 of each parameter $w \in \mathcal{W}$ as below,

$$\sigma_w^2 = \frac{1}{S}\sum_{s=1}^{S}(d_{w_s} - \mu_w) \tag{14}$$

where S is the sampled times for the parameter w and μ_w stands for the average gradient of w during updates.

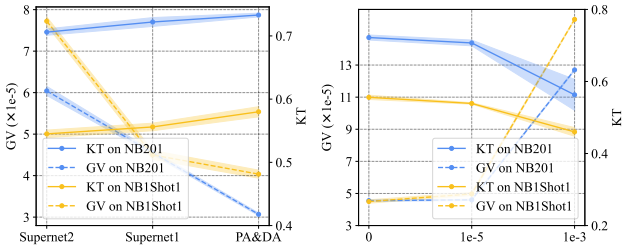
By collecting the gradient variance of each parameter $w \in \mathcal{W}$, we calculate the average value to represent the supernet gradient variance $\sigma_{\mathcal{N}}^2$, which can be formulated as

$$\sigma_{\mathcal{N}}^2 = \mathbb{E}_{w\in\mathcal{W}}[\sigma_w^2] \tag{15}$$

B. Additional experiments

B.1. More evidence for GV-KT

We conduct two more ablation experiments on NAS-Bench-201 [3] and the search space 3 of NAS-Bench-1Shot1 [12] to provide more evidence for the relationship of GV and KT. Firstly, we freeze the supernet operations and adopt different weight-sharing extent as CLOSE [14] to construct two supernets ($S1$ and $S2$). All candidate operations share the same copy of weights for each cell in $S2$, while each operation has its own weights in $S1$. As a result, $S2$ has an obviously higher weight-sharing extent than $S1$ with other factors fixed such as candidate operations and the search space. We use the same configuration to train these two supernets and record the KT and GV in Fig.6 (a). We can see that $S2$ suffers from higher GV than $S1$, resulting in lower KT than $S1$ on both benchmarks, showing that larger GV harms the supernet ranking consistency. Furthermore, we conduct another experiment by freezing the supernet operations and weight-sharing extent and only adding small noise ($1e^{-5}$∼$1e^{-3}$) to gradients of candidate operations during training, which increases GV and deteriorates the KT. Results are shown in Fig.6 (b) and larger GV hampers the supernet training when compared with the baseline without noise.



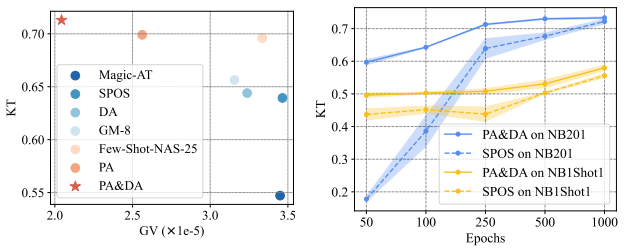
(a) Effect of weight-sharing extent (b) Effect of adding noise

Figure 6. (a) GV and KT with different weight-sharing extents. (b) GV and KT with different noises.

B.2. GV-KT results of more methods

We calculate the GV and KT of different methods on NAS-Bench-201 using the CIFAR-10 dataset. The results are summarized in Fig.7 (a). Note that with GV getting larger, KT generally decreases, demonstrating that larger GV corresponds to lower KT.

*Corresponding author.



(a) GV-KT results (b) Longer training results

Figure 7. (a) GV-KT results of different methods on NAS-Bench-201. (b) Results of longer training epochs on NAS-Bench-201 and NAS-Bench-1Shot1.

B.3. Benchmark results of more training epochs

We provide the results of more training epochs on NAS-Bench-201 and the search space 3 of NAS-Bench-1Shot1 following CLOSE [14] in Fig.7 (b). We can see that PA&DA converges faster and constantly performs better than the baseline SPOS [4]. KT of PA&DA at 250 epochs is close to SPOS at 1K epochs on NAS-Bench-201, significantly saving time in practice. The highest KT of PA&DA on NAS-Bench-201 and NAS-Bench-1Shot1-3 are 0.733 and 0.591 using 3.5 (500 epochs) and 10 (1K epochs) GPU hours respectively. Note that CLOSE ranks the non-isomorphic 6466 sub-models of NAS-Bench-201 and needs to fine-tune the number of curriculums and GLOW blocks, while PA&DA ranks all 15625 sub-models and no longer requires time-consuming fine-tuning. When setting the path smoothing parameter $\delta = 0.5$ and ranking the same sub-models as CLOSE did, we can achieve higher KT than CLOSE (0.772 vs. 0.762). Furthermore, CLOSE explores the dynamic weight-sharing extent during training, which is orthogonal to our method.

C. Discussion with related works

[7] and [8] both use a biased sampling for the supernet training. The former sampled the architectures in proportion to the model sizes, which is not as accurate as our gradient norm to locate the models with insufficient training. The latter proposed de-isomorphic sampling to mitigate the estimation bias and provided many meaningful insights. However, it's intractable to pick out the isomorphic ones in a huge search space and not applicable to the search spaces without isomorphic architectures, for example, the NAS-Bench-1Shot1-3. By contrast, our method is more convenient and general. [11] attributed the failure of NAS methods to weight-sharing and did not give a solution. However, we find that an importance-based sampling can alleviate this issue, thus we propose to jointly optimize path and data sampling distributions during training to improve the supernet consistency.

D. Re-training configuration

D.1. Settings in the DARTS search space

The experimental settings are consistent with previous works [6, 10] to ensure a fair comparison. By stacking the searched normal and reduction cells, the final architecture consists of 20 layers and 36 channels. The final architecture is re-trained on a single GPU [1] by a total of 600 epochs using the training dataset and evaluated on the test dataset to get the top-1 accuracy. The initial learning rate is 2.5e-2 and is then decayed to zero via a cosine strategy. We use the SGD optimizer with the weight decay 3e-4, momentum 0.9, and the training batch size 96. The auxiliary head with a weight of 0.4 and the drop path [5] with a probability of 0.2 are both adopted to mitigate over-fitting. We use the Cutout [2] technique with the length 16 to augment the training data. Besides, we set the threshold of the gradient norm clipping as 5 for all trainable parameters.

D.2. Settings in the ProxylessNAS search space

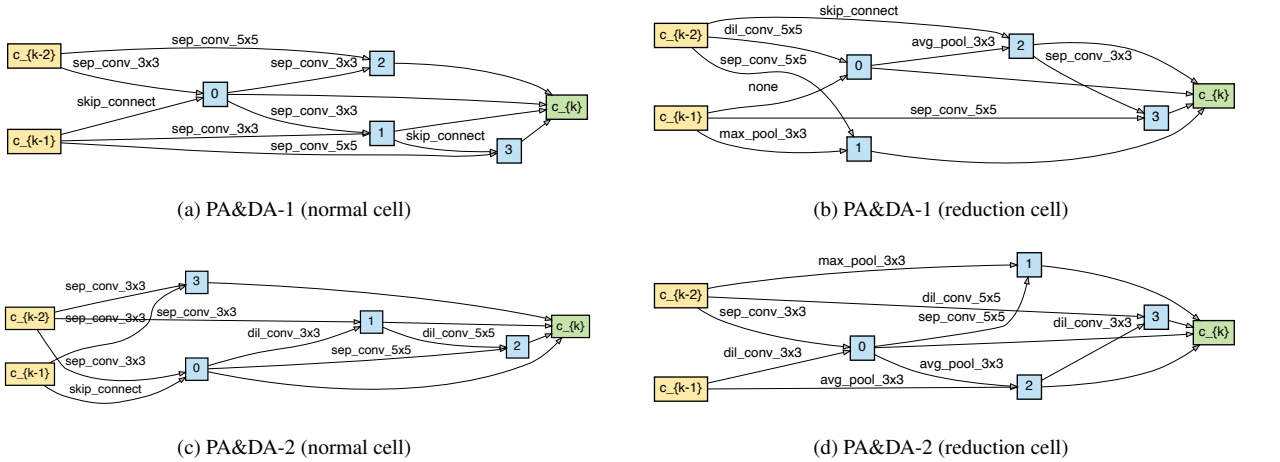
The final architecture contains 21 layers, one of which is the Identity layer. We use 8 GPUs [1] in parallel to re-train our searched architecture on the ImageNet training dataset for 450 epochs and evaluate its performance on the validation dataset. We use the RMSpropTF optimizer with an initial learning rate of 0.16 and a step decay scheduler, which decays the learning rate per 2.4 epochs with a reduction rate of 0.97. The weight decay is 1e-5 and the momentum is 0.9. To mitigate over-fitting, we adopt the AutoAug [1] and RE [13] for the data augmentation, and utilize both the drop path [5] and Dropout [9] with the same rate 0.2. At the initial stage of the training, we utilize a small learning rate of 1e-6 for warm-up by 3 epochs. During training, the moving average technique is employed to smooth the model weights with a rate of 0.9999.

E. Visualization

E.1. Searched cells in DARTS search space

Our best-searched cells have been shown in Fig.3 of our main text and we present the other two searched cells in Fig.8. Although these searched cells have different operations and typologies, we can find a common characteristic of them: all the searched normal cells have many $sep\_conv\_3\times3$ operations and have one $skip\_connect$ operation from the input node to one of the intermediate nodes. We conjecture that these merits lead to the superior performance of the searched cells.

[1] All of our experiments were conducted on the NVIDIA Tesla V100 GPU.



(a) PA&DA-1 (normal cell)

(b) PA&DA-1 (reduction cell)

(c) PA&DA-2 (normal cell)

(d) PA&DA-2 (reduction cell)

Figure 8. Our searched cells in the DARTS search space.

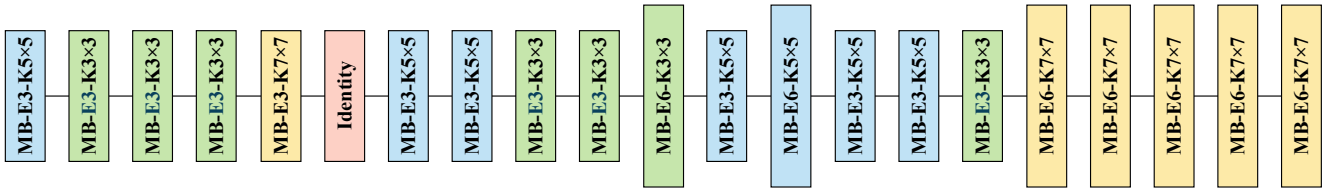


Figure 9. Our searched architecture in the ProxylessNAS search space. The expansion rate of short squares is 3 while being 6 for long squares. Colors of green, blue, and yellow denote the kernel size 3×3, 5×5 and 7×7 of the depth-wise convolution in the MobileNet block, respectively. The red square stands for the layer with the Identity operation.

E.2. Searched architectures in ProxylessNAS search space

As shown in Fig.9, slimmer channels and smaller receptive fields are preferable at the beginning of the network, thus our searched architecture adopts smaller expansion rates and kernel sizes in shallow layers. On the contrary, the last 5 layers all choose the expansion rate 6 with the largest kernel size 7, demonstrating that more channels and larger receptive fields are necessary for encoding the semantic information.

References

[1] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. In *CVPR*, 2019. 2

[2] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 2

[3] Xuanyi Dong and Yi Yang. Nas-bench-201: Extending the scope of reproducible neural architecture search. In *ICLR*, 2020. 1

[4] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *ECCV*, 2020. 2

[5] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. In *ICLR*, 2017. 2

[6] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *ICLR*, 2019. 2

[7] Renqian Luo, Tao Qin, and Enhong Chen. Understanding and improving one-shot neural architecture optimization. *arXiv preprint arXiv:1909.10815*, 44, 2019. 2

[8] Xuefei Ning, Changcheng Tang, Wenshuo Li, Zixuan Zhou, Shuang Liang, Huazhong Yang, and Yu Wang. Evaluating efficient performance estimators of neural architectures. In *NeurIPS*, 2021. 2

[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 2

[10] Ruochen Wang, Minhao Cheng, Xiangning Chen, Xiaocheng Tang, and Cho-Jui Hsieh. Rethinking architecture selection in differentiable nas. In *ICLR*, 2021. 2

[11] Kaicheng Yu, Christian Sciuto, Martin Jaggi, Claudiu Musat, and Mathieu Salzmann. Evaluating the search phase of neural architecture search. In *ICLR*, 2020. 2

[12] Arber Zela, Julien Siems, and Frank Hutter. Nas-bench-1shot1: Benchmarking and dissecting one-shot neural architecture search. In *ICLR*, 2020. 1

[13] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020. 2

[14] Zixuan Zhou, Xuefei Ning, Yi Cai, Jiashu Han, Yiping Deng, Yuhan Dong, Huazhong Yang, and Yu Wang. Close: Curriculum learning on the sharing extent towards better one-shot nas. In *ECCV*, 2022. 1, 2