Supplementary Material for "DOTS: Decoupling Operation and Topology in Differentiable Architecture Search"

Yu-Chao Gu[1*] Li-Juan Wang[1*] Yun Liu[1] Yi Yang[2] Yu-Huan Wu[1]
Shao-Ping Lu[1] Ming-Ming Cheng[1]

[1]TKLNDST, CS, Nankai University [2]Zhejiang University

Abstract

In this supplementary, we provide more detailed information for DOTS, including:

- ***Detailed experimental settings.***
- ***Discussion about the operation search.***
- ***Visualization of the searched cells.***

1. Detailed Experimental Settings

CIFAR. The whole search process on CIFAR10/100 takes 70 epochs, *i.e.*, 30 epochs for the operation search and 40 for the topology search. We pretrain network weights in the first half epochs for both stages by only updating network weights. The network is composed of 8 cells for the operation search and 20 cells for the topology search. The SGD optimizer is adopted to optimize the network weight w with an initial learning rate of 0.025 (cosine decaying to 0.001 in 70 epochs), weight decay of $3e$-4, and momentum of 0.9. For updating the operation weight α and edge combination weight β, we use the Adam optimizer with a constant learning rate of $1e$-4 and weight decay of $1e$-3. We set the initial temperature $T_0 = 10$ and decay rate $\theta = 0.72$ for annealing the edge combination weight in the topology search.

ImageNet. We randomly sample 10% and 2.5% images from ImageNet to build the training and validation set, following PC-DARTS [16]. The search schedule of ImageNet follows CIFAR experiment. The SGD optimizer is used to optimize the network weight w with an initial learning rate of 0.25 (cosine decaying to $1e$-2 in 70 epochs), weight decay of $3e$-4, and momentum of 0.9. The batch size of SGD is set to 512. For updating the operation weight α and edge combination weight β, we use the Adam optimizer with a constant learning rate of $3e$-3 and weight decay of $1e$-3 for both stages. We set the initial temperature $T_0 = 10$ and decay rate $\theta = 0.72$ for annealing the edge combination weight in the topology search.

2. Discussion about the Operation Search

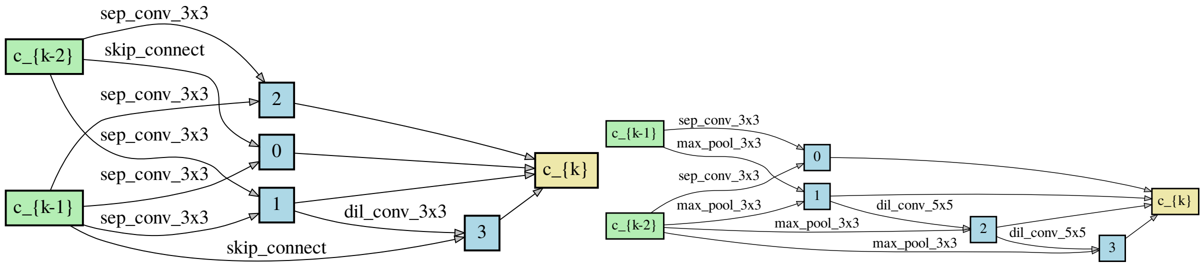
DOTS introduces two operation search strategies, *i.e.*, 1) incorporating existing gradient-based methods, and 2) searching from scratch using the group strategy. The first strategy suffers from inheriting instability in gradient-based methods [4, 2], and thus the retained operations may be suboptimal. Furthermore, the first strategy ignores that some operations are related to the topology, which is better to make them involved in the topology search. Recent research [3] reveals that the *Skip-Connection* operation severs two roles: 1) an operation in the cell and 2) a connection to stabilize the network. The latter role makes skip-connection related to the network topology. The *Zero* operation is proposed in DARTS [13] to scale edge importance in the search stage, which is also related to the network topology. Pruning these topology-related operations in the operation search phase eliminates the potential topology choices in the topology search.

The second strategy, *i.e.*, the operation search with the group strategy, helps stabilize the operation search and preserve more potential topology choices. We have compared two group strategies in the manuscript, *i.e.*, Group-V1 strategy and Group-V2 strategy. The Group-V1 strategy [10] considers the multicollinearity of similar operations by dividing operations into four groups:

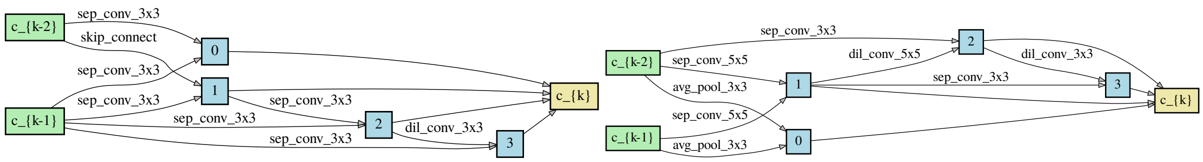
- Group1: *Skip-Connection*
- Group2: *Max-Pooling*, *Avg-Pooling*
- Group3: *SepConv3×3*, *SepConv5×5*
- Group4: *DilConv3×3*, *DilConv5×5*

The Group-V2 strategy [8] considers the *Matthew Effect* in the operation search. Specifically, the operations with learnable parameters are under-performing at the beginning of the search and thus be punished by lowering their importance. The lower importance makes these operations update slower, resulting in even smaller importance. Hence, the Group-V2 strategy divides operations into two groups based on whether they have learnable parameters:

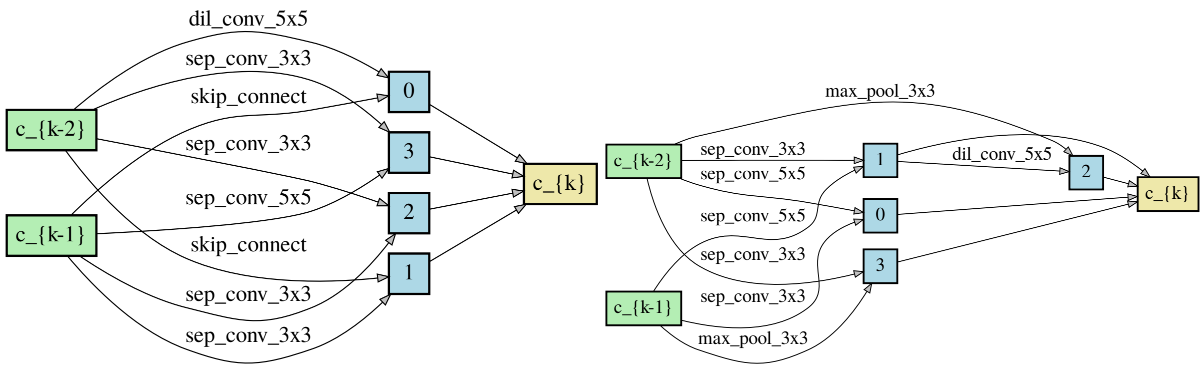
- Group1: *Zero*, *Skip-Connection*, *Max-Pooling*, *Avg-Pooling*



(a) Normal cell (left) and reduction cell (right) of CIFAR10



(b) Normal cell (left) and reduction cell (right) of CIFAR100



(c) Normal cell (left) and reduction cell (right) of ImageNet

Figure 1: Visualization of the best searched cells of DOTS.

Backbone	#Param (M)	FLOPs (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-50 [7]	25.6	4120	0.363	0.553	0.386	0.193	0.400	0.488
MobileNet-V2 [14]	3.4	300	0.283	0.467	0.293	0.148	0.307	0.381
SinglePath NAS [6]	4.3	365	0.307	0.498	0.322	0.154	0.339	0.416
MobileNet-V3 [9]	5.4	219	0.299	0.493	0.308	0.149	0.333	0.411
MnasNet [15]	4.8	340	0.305	0.502	0.320	0.166	0.341	0.411
FairDARTSC [4]	5.0	386	0.319	0.519	0.330	0.174	0.353	0.430
DOTS	5.3	596	0.357	0.552	0.378	0.199	0.393	0.478

Table 1: Evaluation of object detection on the MS-COCO 2017 dataset [12].

Backbone	#Param (M)	FLOPs (G)	mIOU(%)	
			val	test
ResNet-18	14.1	20.1	74.8	74.7
Xception-39	1.9	4.1	69.0	68.4
MnasNet	6.8	11.0	76.8	74.2
DOTS	8.0	12.9	**79.3**	**77.6**

Table 2: Evaluation of semantic image segmentation on the Cityscapes dataset [5].

- Group2: *SepConv3×3*, *SepConv5×5*, *DilConv3×3*, *DilConv5×5*

In this paper, the Group-V2 strategy helps avoid the *Matthew Effect* and preserve more potential topology choices for the topology search. Therefore, DOTS uses the Group-V2 strategy as the default operation search method.

3. Applications

We apply the architecture searched by DOTS to object detection and semantic segmentation to validate its performance. We use the architecture searched on ImageNet as a drop-in replacement of the backbone of the baseline methods [11, 17]. Here, we compare with several manually-designed and automatically-searched mobile backbones.

Object Detection. The object detection benchmark is based on RetinaNet [11]. We use the MMDetection toolbox [1] for a fair comparison to [4]. All models are trained and evaluated on MS-COCO 2017 dataset [12] with the same settings as [4]. The results are summarized in Tab. 1. The proposed DOTS outperforms FairDARTSC [4] by 3.8% in terms of AP. DOTS has comparable performance to ResNet-50 with only 20.7% parameters and 14.5% FLOPs of ResNet-50.

Semantic Segmentation. The semantic segmentation benchmark is based on BiSeNet [17]. All models are trained and evaluated on the Cityscape dataset [5] with default settings in [17], respectively. We do not employ any complicated testing techniques, like multi-scale or multi-crop testing. From Tab. 2, we can observe that DOTS has a clear advantage over previous mobile backbones in lightweight semantic segmentation.

4. Visualization

We visualize the best searched cells of DOTS in Fig. 1.

References

[1] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 2

[2] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *ICCV*, pages 1294–1303, 2019. 1

[3] Xiangxiang Chu, Xiaoxing Wang, Bo Zhang, Shun Lu, Xiaolin Wei, and Junchi Yan. Darts-: Robustly stepping out of performance collapse without indicators. *arXiv preprint arXiv:2009.01027*, 2020. 1

[4] Xiangxiang Chu, Tianbao Zhou, Bo Zhang, and Jixiang Li. Fair darts: Eliminating unfair advantages in differentiable architecture search. *arXiv preprint arXiv:1911.12126*, 2019. 1, 2, 3

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 2, 3

[6] 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. *arXiv preprint arXiv:1904.00420*, 2019. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[8] Weijun Hong, Guilin Li, Weinan Zhang, Ruiming Tang, Yunhe Wang, Zhenguo Li, and Yong Yu. DropNAS: Grouped operation dropout for differentiable architecture search. In *IJCAI*, pages 2326–2332, 2020. 1

[9] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, pages 1314–1324, 2019. 2

[10] Guilin Li, Xing Zhang, Zitong Wang, Zhenguo Li, and Tong Zhang. StacNAS: Towards stable and consistent optimization for differentiable neural architecture search. *arXiv preprint arXiv:1909.11926*, 2019. 1

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *ICCV*, Oct 2017. 2

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014. 2

[13] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *ICLR*, 2019. 1

[14] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 2

[15] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, pages 2820–2828, 2019. 2

[16] 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 *ICLR*, 2020. 1

[17] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, September 2018. 2, 3