Supplementary material for differentiable architecture search with random features

1. Algorithm

We utilize the first-order approximation of DARTS for the architecture gradient to accelerate the search stage. The overall procedure of RF-DARTS is outlined in Alg. 1.

Algorithm 1 RF-DARTS: Differentiable Architecture Search with Random Features

Require: Supernet weights $w_{\text{conv}}, w_{\text{bn}}$;
Architecture parameters α.
Ensure: Searched architecture parameters α.

Random initialize weights $w_{\text{conv}}^{\text{init}}$;
while *not converged* **do**
  Update α by descending $\nabla_{\alpha}\mathcal{L}_{\text{val}}(w_{\text{conv}}^{\text{init}}, w_{\text{bn}}, \alpha)$.
  Update w_{bn} by descending $\nabla_{w_{\text{bn}}}\mathcal{L}_{\text{train}}(w_{\text{conv}}^{\text{init}}, w_{\text{bn}}, \alpha)$.
end while

2. Experiment setting

2.1. Search space

NAS-Bench-201. The skeleton of NAS-Bench-201 [2] supernet consists of four parts: 1) a stem layer, 2) three stacked stages and each stage includes 5 search cells, 3) two residual blocks with stride 2, 4) a global average pooling layer and a classifier layer. The search cell is represented as a densely-connected directed acyclic graph (DAG). There are four nodes and six edges in the DAG. Each edges has five candidate operations: (1) none, (2) skip-connection, (3) 1×1 convolution, (4) 3×3 convolution and (5) 3×3 average pooling. With the assumption of shared cell topology, there are total 15625 candidate architectures in the NAS-Bench-201 search space.

DARTS search space. The main body of DARTS [4] supernet consists of three parts: 1) a stem layer, 2) eight stacked search cells, and 3) a global average pooling layer and a classifier layer. Specifically, DARTS supernet includes two search cell types, namely normal cell and reduction cell. The reduction cells are located at 1/3 and 2/3 of the supernet depth, and the other search cells are called normal cells. There are six nodes and fourteen edges in both normal cells and reduction cells. Each edge has eight candidate operations: (1) none, (2) 3×3 average pooling, (3) 3×3 max pooling, (4) skip-connection, (5) 3×3 SepConv, (6) 5×5 SepConv, (7) 3×3 DilConv, (8) 5×5 DilConv. With the assumption of the shared normal cell and reduction cell topology, there are total 10^{18} candidate architectures in the DARTS search space. In search phase, there are 8 search cells both on CIFAR and ImageNet. In evaluation phase, the number of cell is increased from 8 to 20 on CIFAR, and is increased to 14 on ImageNet.

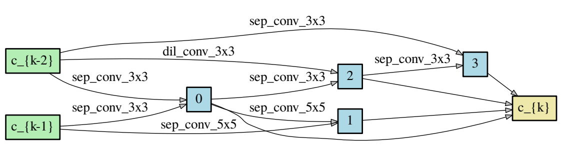
2.2. Datasets and training settings

CIFAR. CIFAR [3] consists of CIFAR-10 and CIFAR-100. Both CIFAR-10 and CIFAR-100 contains 50K training images and 10K test images. CIFAR-10 has 10 image categories and CIFAR-100 has 100 image categories. The image resolution in CIFAR-10 and CIFAR-100 is 32×32. In the search stage, the original training dataset is splitted into training dataset and validation dataset with equal size. The new training dataset is used to train supernet weights and validation dataset is used to optimize architecture parameters. We use similar search training setting in both NAS-Bench-201 and DARTS search space as vanilla DARTS. The number of training epochs is 50. We use SGD optimizer with a cosine learning rate scheduler initialized with 0.025, a momentum of 0.9, a weight decay of 1e-3 (14e-4 in NAS-Bench-201) and a gradient clip of 5. We also use an Adam optimizer with a constant learning rate 3e-4, a beta of (0.5, 0.999), and a weight decay of 1e-3. There is no evaluation stage in NAS-Bench-201 because of the providing ground truth accuracy. For the evaluation stage in DARTS search space, we retrain searched architectures 600 epochs on both CIFAR-10 and CIFAR-100. Besides, the depth of searched architecture is increased from 8 to 20 and the number of initial channel is increased from 16 to 36. Other training settings keep the same as the ones of supernet weight optimization in the search stage.

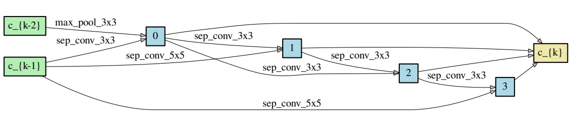
ImageNet. ImageNet-1K [1] consists of 1.28M training images and 50K validation images. The image resolution keeps a defaut setting with 224×224. We follow the search and evaluation training settings provided in PC-DARTS [5]. The depth of DARTS supernet is also 8 cells. However, with limited GPU memory, the DARTS supernet use three convolution layer with stride 2 to down-sample feature resolution from 224×224 to 28×28. In the search phase, data subsets 10% and 2.5% of the images from each class are randomly sampled from training dataset. The former (10% of the training images) is used to train supernet weights and the latter subset (5% of the training images) is used to optimize architecture parameters. We train supernet with 50 epochs. For the first 35 epochs, we only train BN affine weights. Then, we jointly optimize BN affine weights and architecture parameters in a iterative way. For BN affine weights optimization, we use SGD optimizer with a cosine learning rate scheduler initialized with 0.5, batch size 1024, a momentum of 0.9, a weight decay of 1e-3 and a gradient clip of 5. As for architecture parameters, we use an Adam optimizer with a constant learning rate 6e-3, a beta of (0.5, 0.999), and a weight decay of 1e-3. After search, we build the searched architecture with 14 cells and 48 initial channels. We evaluate the architecture with 250 training epochs and a SGD optimizer with a momentum of 0.9, an initial learning rate of 0.5 (decayed down to zero linearly), and a weight decay of 3e-5. Label smoothing with confidence 0.9 and an auxiliary loss tower are adopted during training. We warm-up learning rate for the first 5 epochs.

3. Visualization of architectures

Here we visualize the searched cell architectures: RF-DARTS searched on CIFAR-10 (Figure 1), RF_PCDARTS searched on ImageNet-1K (Figure 2), and RF-DARTS searched on CIFAR-10, CIFAR-100, and SVHN across RobustDARTS S1-S4 (Figure 3 to Figure 14).

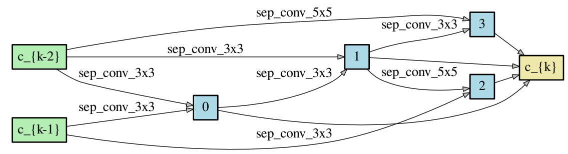


(a) normal cell

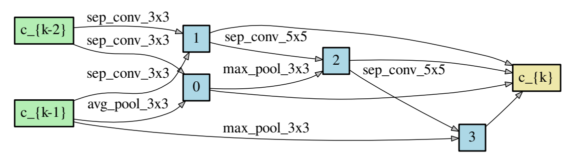


(b) reduction cell

Figure 1. RF-DARTS searched on CIFAR-10

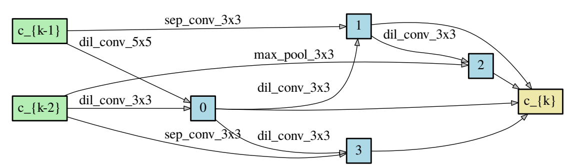


(a) normal cell

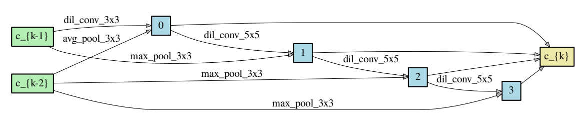


(b) reduction cell

Figure 2. RF-PCDARTS searched on ImageNet-1K

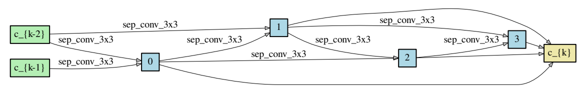


(a) normal cell

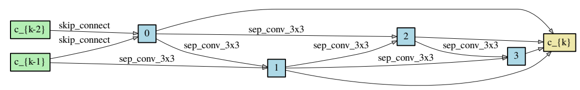


(b) reduction cell

Figure 3. RF-DARTS (S1) searched on CIFAR-10

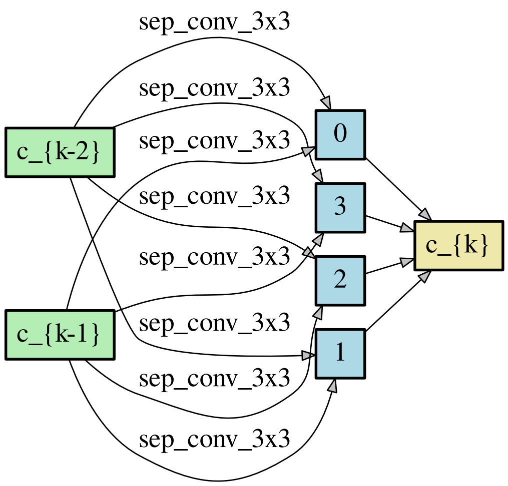


(a) normal cell

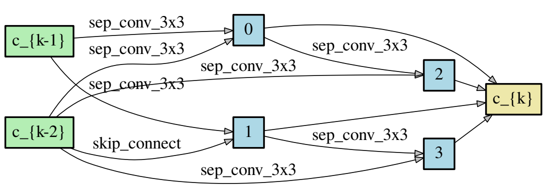


(b) reduction cell

Figure 4. RF-DARTS (S2) searched on CIFAR-10

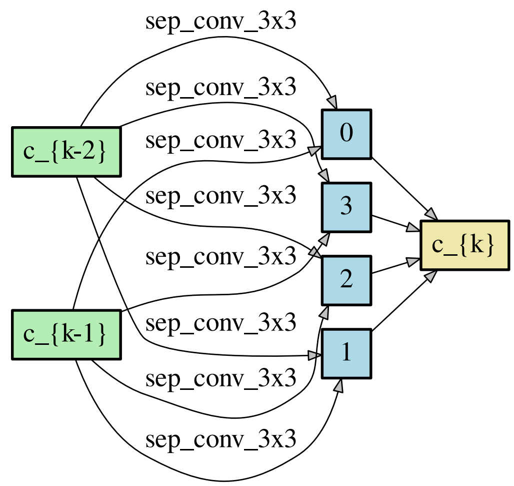


(a) normal cell

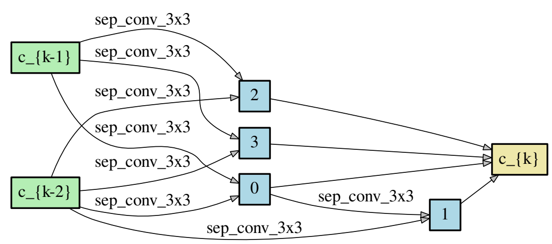


(b) reduction cell

Figure 5. RF-DARTS (S3) searched on CIFAR-10

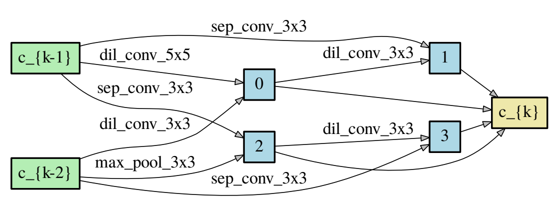


(a) normal cell

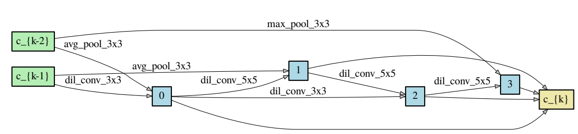


(b) reduction cell

Figure 6. RF-DARTS (S4) searched on CIFAR-10

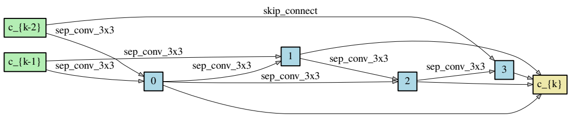


(a) normal cell

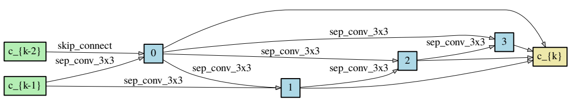


(b) reduction cell

Figure 7. RF-DARTS (S1) searched on CIFAR-100

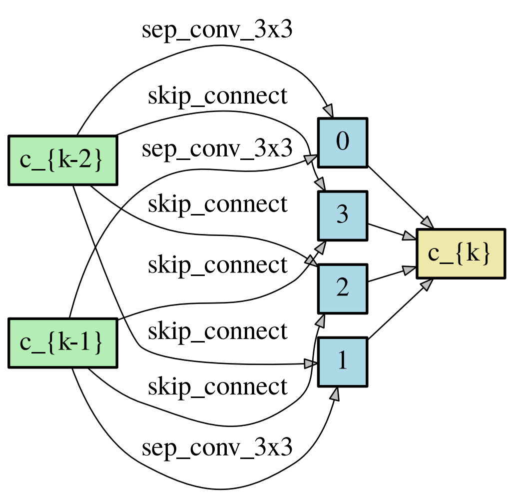


(a) normal cell

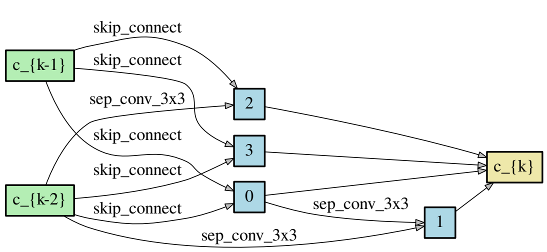


(b) reduction cell

Figure 8. RF-DARTS (S2) searched on CIFAR-100

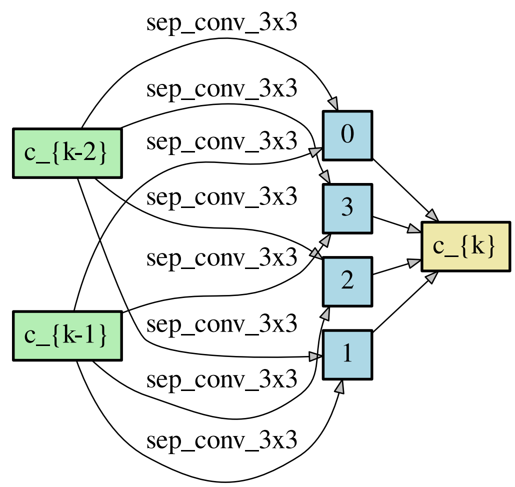


(a) normal cell

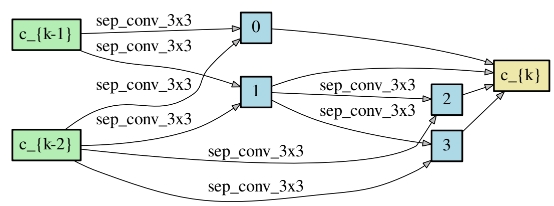


(b) reduction cell

Figure 9. RF-DARTS (S3) searched on CIFAR-100

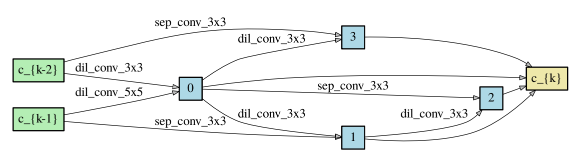


(a) normal cell

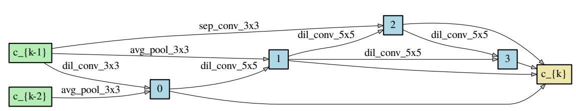


(b) reduction cell

Figure 10. RF-DARTS (S4) searched on CIFAR-100

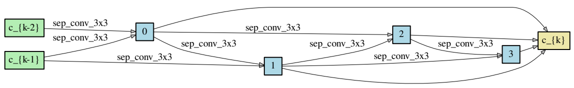


(a) normal cell

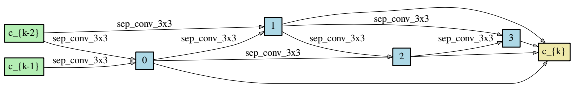


(b) reduction cell

Figure 11. RF-DARTS (S1) searched on SVHN

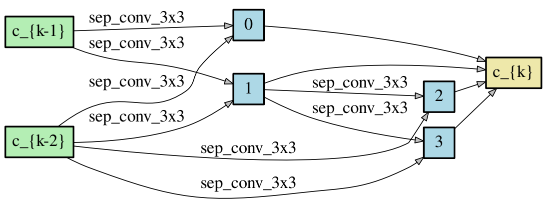


(a) normal cell

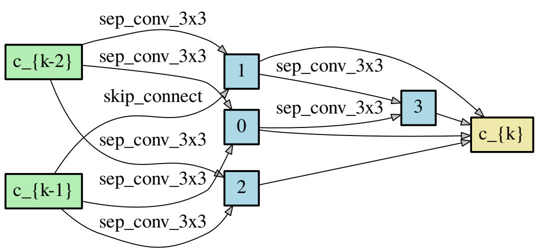


(b) reduction cell

Figure 12. RF-DARTS (S2) searched on SVHN

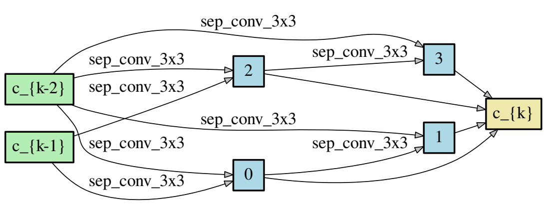


(a) normal cell

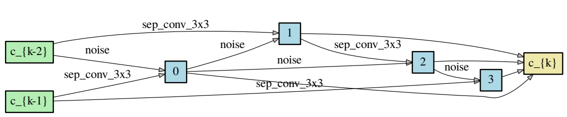


(b) reduction cell

Figure 13. RF-DARTS (S3) searched on SVHN



(a) normal cell



(b) reduction cell

Figure 14. RF-DARTS (S4) searched on SVHN

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 1

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

[3] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

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

[5] 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