Are Labels Necessary for Neural Architecture Search? (Supplementary Material)

Chenxi Liu[1], Piotr Dollár[2], Kaiming He[2], Ross Girshick[2], Alan Yuille[1], and Saining Xie[2]

[1] Johns Hopkins University
[2] Facebook AI Research

A Additional Details for Sample-Based Experiments

In sampling the 500 DARTS architectures, we only keep those whose #Params and #FLOPs both fall into [80%, 120%] that of ResNet-56. Doing so prevents the influence of model capacity on the rank correlations, but the downside is that these architectures may not be representative of the entire search space. Therefore, in sampling the 500 NAS-Bench-101 architectures, we used a complementary strategy where we first rank all architectures based on the released accuracy, and then sample 50 architectures from each 10th percentile.

B Additional Details for Search-Based Experiments

B.1 Search Phase

Recall that in the search phase of NAS-`DARTS`/UnNAS-`DARTS`, we consider 3 datasets: ImageNet-1K (IN1K), ImageNet-22K (IN22K), Cityscapes.

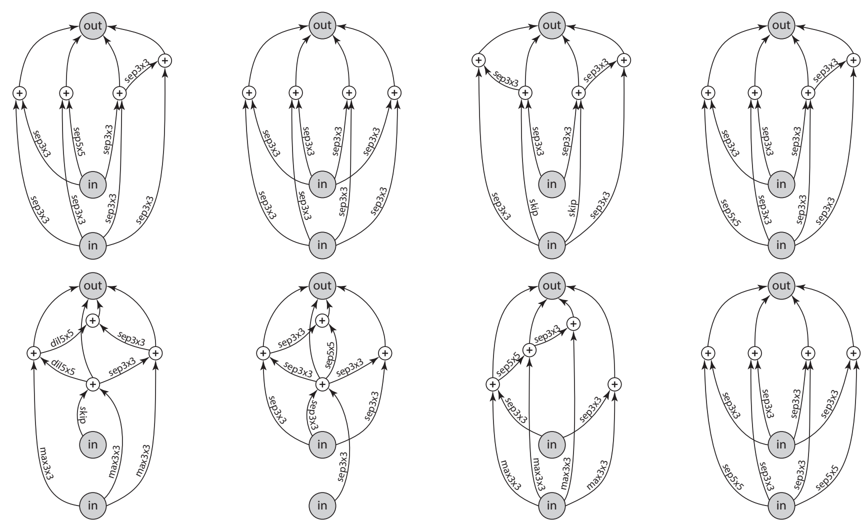
For IN1K, we follow [3,2] to postpone updating architecture parameters α for half of the total search epochs. For IN1K/IN22K, the search phase lasts 2 epochs for IN1K and 1 epoch for IN22K. Since IN22K is approximately 10 times larger than IN1K (~14M vs ~1.2M), the search on IN22K is approximately 5 times longer than the search on IN1K. Batch size is 64, learning rate is 0.1 (cosine schedule), weight decay is 0.00003. For Cityscapes, the search phase lasts 400 epochs. Batch size is 32, learning rate is 0.1 (cosine schedule), weight decay is 0.0003. Other than those listed here and in the main paper, all hyperparameters follow those used in the original DARTS. In a few settings where the batch size above will exceed 32GB GPU memory, we divide batch size and learning rate (both for weights and for architecture parameters α) by 2.

B.2 Evaluation Phase

When evaluating an architecture on Cityscapes semantic segmentation, since the task is pixel-level classification instead of image-level classification, we need to make minimal but necessary modifications. Different from the network used in IN1K classification, we (1) replace the last stride 2 layer with stride 1, to increase spatial resolution; and (2) remove the global average pooling and replace the fully connected classifier with the Atrous Spatial Pyramid Pooling (ASPP) module [1] followed by bilinear upsampling to produce per-pixel classification at the original resolution. These are the same modifications the segmentation framework DeepLabv3 [1] made to a ResNet backbone.

C NAS-`DARTS` and UnNAS-`DARTS` Architectures

Here we visualize all the NAS-`DARTS` and UnNAS-`DARTS` cell architectures: searched on ImageNet-1K (Figure 1), ImageNet-22K (Figure 2), Cityscapes (Figure 3).

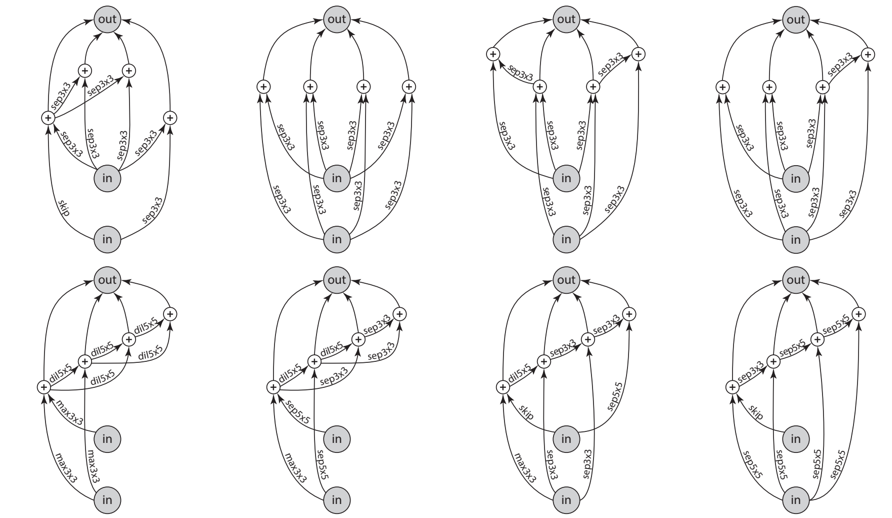


(a) NAS `Supv.Cls` (b) UnNAS `Rot` (c) UnNAS `Color` (d) UnNAS `Jigsaw`

Figure 1: Cell architectures (normal and reduce) searched on ImageNet-1K.

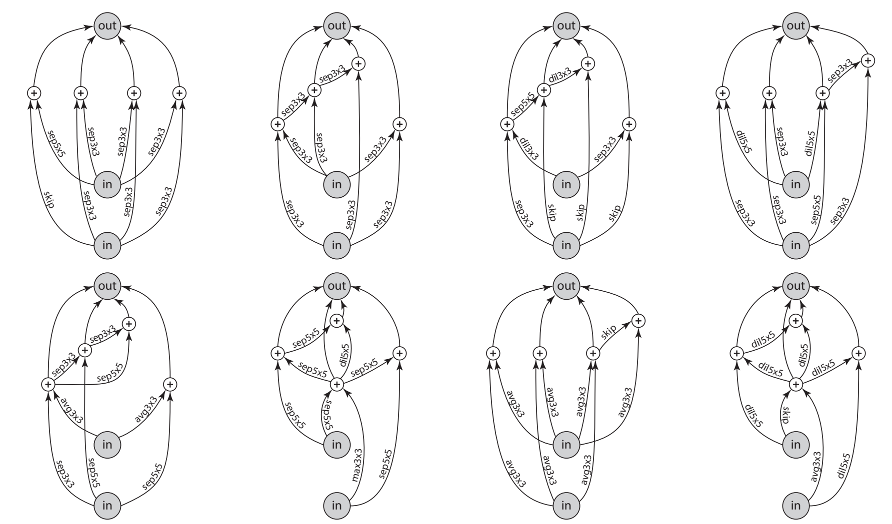
References

1. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv:1706.05587 (2017)
2. Chen, X., Xie, L., Wu, J., Tian, Q.: Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In: ICCV (2019)
3. Liu, C., Chen, L.C., Schroff, F., Adam, H., Hua, W., Yuille, A.L., Fei-Fei, L.: Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In: CVPR (2019)



(a) NAS `Supv.Cls` (b) UnNAS `Rot` (c) UnNAS `Color` (d) UnNAS `Jigsaw`

Figure 2: Cell architectures (normal and reduce) searched on ImageNet-22K.



(a) NAS `Supv.Seg` (b) UnNAS `Rot` (c) UnNAS `Color` (d) UnNAS `Jigsaw`

Figure 3: Cell architectures (normal and reduce) searched on Cityscapes.