

Appendix

Implementation details

Fractionally-strided convolution: An important detail when implementing a fractionally-strided convolution with a transposed convolution in PyTorch is to flip the kernel spatially about the center, and swap the input and output channel dimensions of the kernel tensor before performing the convolution. These modifications are necessary because a transposed convolution is essentially implemented as a standard convolution with a flipped kernel striding over the output feature rather than the input.

Random option fine-tuning: In this work, we fine-tune for 15 epochs. For the dynamic ResNet-50 model, we use SGD and start with a learning rate of 0.001 and decay it to 0.0001 after 10 epochs. We also set momentum to 0.9 and weight decay to 0.0001.

Dynamic layer locations: For the ResNet-50 model, we make the 3×3 convolutional layer in the first Bottleneck module in stages $C2$, $C3$, $C4$, and $C5$ dynamic. ResNeXt-50 is very similar, but the layers that we modify are group convolutions. In MobileNetV2, we allow the 3×3 depth-wise convolutions in all four Bottlenecks that have a default stride of 2 to be dynamic. DenseNet-121 is slightly different because it does not use strided convolution to down-sample feature resolution. Rather, it uses three strided pooling layers (*i.e.* *Transition* layers). Thus, we implement the dynamic stride by changing the stride of the average pool layers in the three *Transition* layers as well as the stride in the max pool layer in the stem. We implement dynamic dilation and kernel size in the first *Dense Layer* in each of the four *Dense Blocks*.

Additional figures

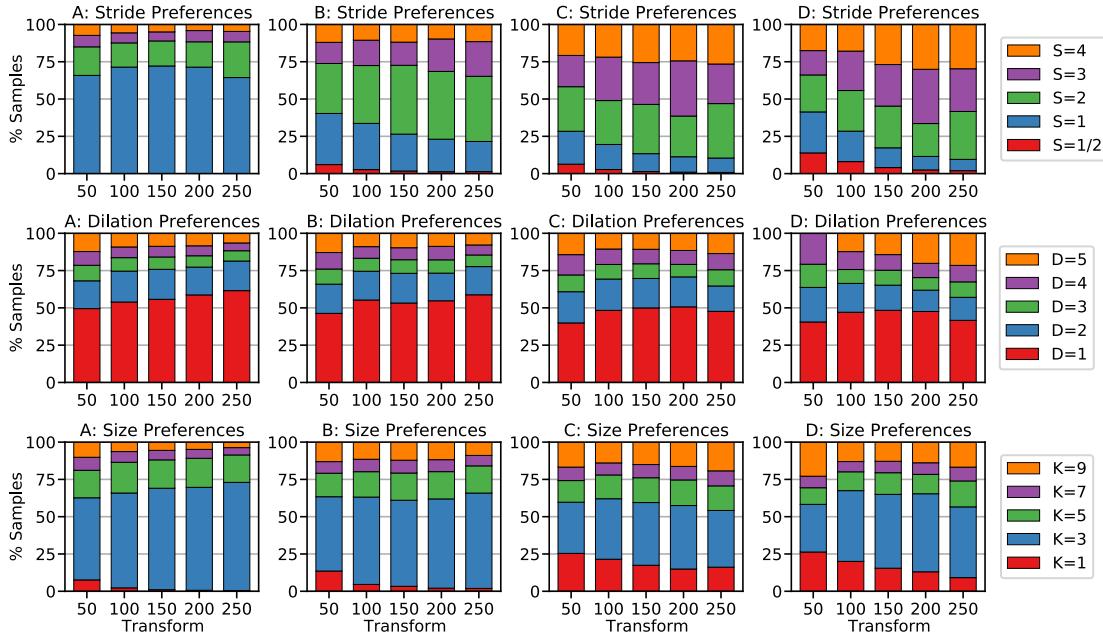


Figure 7: The effect of global context on model preference. The distribution of model preference is compared for each dynamic layer over a series of crop transforms to each input image. Note that the transform 200 refers to center-cropping a 200×200 region of the resized input. The three rows correspond to ResNet-50 stride-, dilation-, and size-dynamic models, respectively.

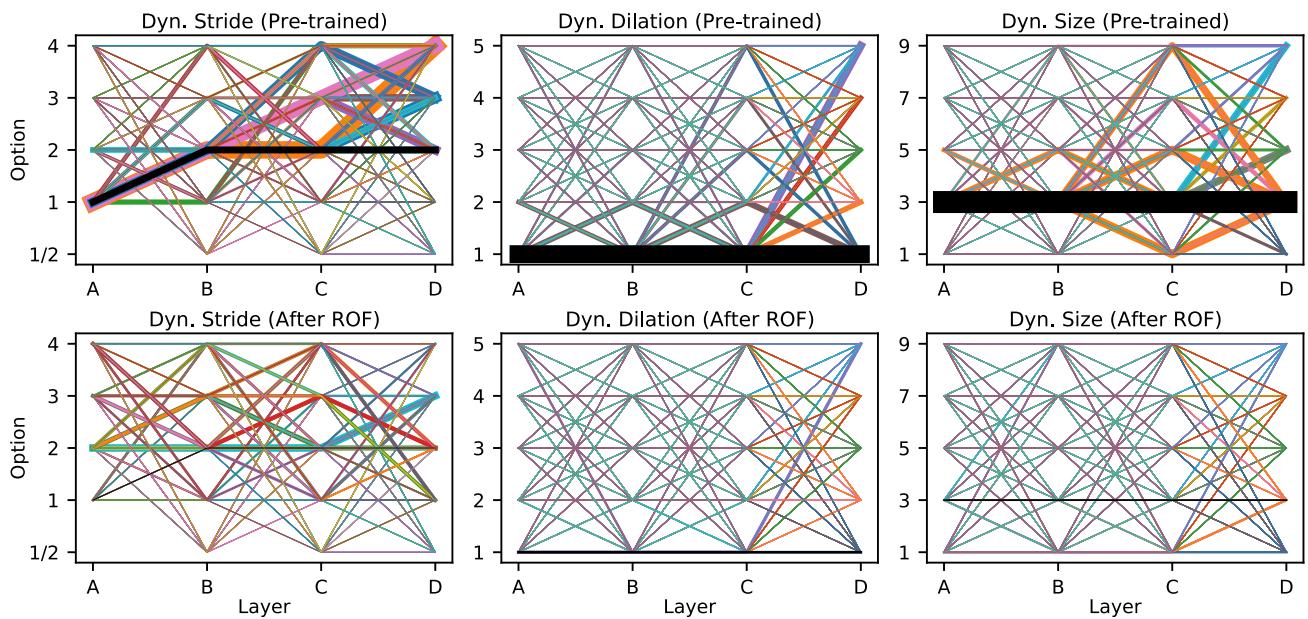


Figure 8: Preference paths of ResNet-50 (ABCD). Similar to Figure 1, this plot is an intuitive visualization of the option permutation preferences of a ResNet-50 (ABCD) model. Each different colored line stretches from layer A to D and represents a different possible option sequence over the depth of the model. Line thickness encodes the proportion of the validation samples that most prefer that permutation of options. The lines corresponding to the default permutations are colored black.