Appendices

A. Related works on object detection

Recently, two-stage detectors [10, 25] have been performance state-of-the-art. They first generate class-independent region proposals by using the region proposal network, then classify them by using detection heads. However, they have the drawbacks of long inference time and complex model architecture. To cope with this drawback, one-stage detectors [24, 19] directly predict object categories and bounding boxes (that is, anchors) at each location of feature maps that are generated by the backbone network. Although this end-to-end approach has the advantage of faster inference, it requires hyper-parameter tuning to find suitable anchors and complex model architecture for increasing the number of anchors.

B. Training details

Stage	Search space, (D,W,E)
1	$\{0,1,2\} \times \{1.0\} \times \{0.35\}$
2	$\{0,1,2\} \times \{0.8,1.0\} \times \{0.35\}$
3	$\{0,1,2\} \times \{0.65,0.8,1.0\} \times \{0.35\}$
4	$\{0,1,2\} \times \{0.65,0.8,1.0\} \times \{0.25,0.35\}$
5	$\{0,1,2\} \times \{0.65,0.8,1.0\} \times \{0.2,0.25,0.35\}$

Table A1. Search space for OFA PS.

Stage	1	2	3	4	5
Epochs	70	5	65	5	65

Table A2. Training schedule for OFA progressive shrinking.

The search space and training schedule for each OFA progressive shrinking (PS) stage are detailed in Table A1 and Table A2. For training, we used the Adam optimizer [18]. Settings for each dataset are detailed as follows.

Pascal VOC: The initial learning rate was set to 5e-4, with the step scheduler for learning rate decay. The learning rate was decayed by 0.1 at 45 and 60 epochs. The training epochs for fullnet were 70. The training batch size was 32.

COCO: The initial learning rate was set to 5e-4, with the cosine scheduler for learning rate decay. The fullnet training epochs were 140. The training batch size was 64.

C. Evalution of path filter

Our path filter is designed to predict the relative performance of paths. It is more flexible than the path filter proposed in the prior work [15], i.e., once the path filter is trained, a different pruning ratio can be applied. Here, we

Pruning ratio	Accuracy	Precision	Recall
0.2	0.940	0.850	0.180
0.3	0.910	0.869	0.252
0.4	0.890	0.864	0.368

Table A3. Path filter performance for predicting the weakest $r_{\text{path}}\%$ paths on different pruning ratios. Our path filter can be used for different pruning ratios.

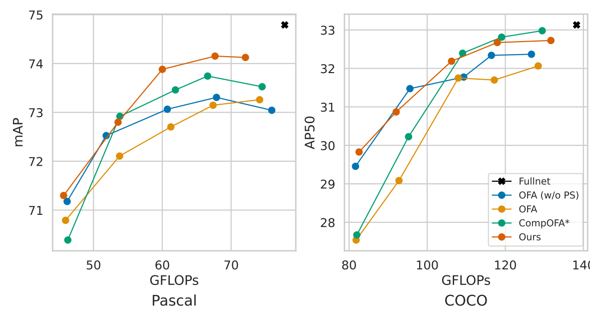

Figure A1. The performance of the optimal architecture a^*, for object detection on the Pascal VOC (left) and the COCO (right) dataset. Our method outperforms OFA (w/o progressive shrinking, PS) across all given FLOP bounds. CompOFA* performs well only for larger FLOPs.

demonstrate that our path filter performs well when different pruning ratios are adopted. Table A3 presents the path filter performance to predict the weakest $r_{\text{path}}\%$ paths. For all pruning ratios, the precision is more than 0.7. The performance, especially for precision and recall, improves for larger pruning ratios because classification is easier when the number of positive and negative samples is similar. The results confirm the utility of our path filter for different pruning ratios.

D. Comparision with prior NAS approaches under the same GPU costs

Method	Epochs (Pascal VOC)	Epochs (COCO)
OFA (w/o PS)	147	147
CompOFA*	147	147
Ours	70	140

Table A4. Training schedule for comparison under the same GPU costs.

Figure A1 presents the evolution search results for training the supernet with OFA (w/o PS) and CompOFA* for the same GPU costs with the proposed method. The training schedule is summarized in Table A4. OFA (w/ PS) is trained for the same schedule as results in Figure 5 and presented for reference. For both Pascal VOC and COCO, training more epochs improves the accuracy of small paths for OFA, however, it degrades the accuracy for large paths. Moreover, the accuracy of small paths for CompOFA* is smaller than that of ours. This infers the limitation of hand-crafted search space pruning.