Supplementary Materials for "Joint-DetNAS: Upgrade Your Detector with NAS, Pruning and Dynamic Distillation"

Lewei Yao[1*] Renjie Pi[1*] Hang Xu[2†] Wei Zhang[2] Zhenguo Li[2] Tong Zhang[1]
[1]Hong Kong University of Science and Technology [2]Huawei Noah's Ark Lab

A. Implementation Detail

A.1. NAS

A.1.1 Search Space

We adopt the ResNet-based detectors as the search space due to its popularity in the detection community. Specifically, the backbone architecture is divided into four stages, where the feature resolution halves and the number of output channels doubles at the beginning of each stage. Basic block is used for R18-based students, while Bottleneck Block is used for other students and the teacher pool. In the following sections, "layer" and "block" are used interchangeably.

A.1.2 Student Morphism

The student's action spaces contains four actions: (1) **Channel Pruning**, (2) **Layer Pruning**, (3) **Add-Layer** and (4) **Rearrange**.

We specify the definition of f_{evolve} for each action.

- **Pruning** The parameters are first ranked globally by an importance measure, then the least important ones are removed while the rest are inherited. For **Channel Pruning**, the importance measure is the magnitude of each BN's channel weights. For **Layer Pruning**, the importance measure is the parameter's L1 norm.
- **Add-Layer** aims to introduce extra capacity into the detector while maintain the performance of the predecessor. This is realized by initializing the block as an identity mapping. Specifically, for each block in ResNet whose output can represented as $H(x) = F(x) + x$, we make $F(x)$ equal to 0 by applying Dirac initialization [8] to the CONV layers and zero-initializing the last BN layer[4, 3]. The new layer is appended to the end of the selected stage.
- **Rearrange**, a stage is firstly selected, then the layer at the beginning or the end of the stage is moved to its neighboring stage by modifying its stride, the parameters can then be directly inherited.

A.1.3 Elastic Teacher Pool (ETP)

Subnet Space. In our implementation of the ETP, the super-network is set to have the same depth and 1.5x width as ResNet101. Specifically, the depths and the width coefficients are [3, 4, 23, 3] and [1.5, 1.5, 1.5, 1.5] at each stage, respectively. During our integrated progressive shrinking training, the subnet space is gradually expanded to include smaller subnets. At the final phase, the smallest subnet in the space has depths [2,2,2,2] and width coefficients [1.0, 1.0, 1.0, 1.0] at each stage, all the subnets in between can be sampled and trained. The width coefficients can be 1.0, 1.25 or 1.5.

Dynamic Resolution. We use 512×512, 800×600, 1080×720 and 1333×800 as the predefined resolutions, from which one is randomly sampled during each training iteration.

Phases of integrated progressive shrinking. (1) **Training the super-network:** the super-network is firstly trained with dynamic resolution, which is later used as the teacher detector to distill other subnets. (2) **First shrinking phase:** the depths and widths of the subnet space are expanded to [3,4,12-23,3] and [1.25-1.5,1.25-1.5,1.25-1.5,1.25-1.5] , respectively. (3) **Second shrinking phase:** the depths and widths of the subnet space are expanded to [2-3,2-4,2-23,2-3] and [1.0-1.5,1.0-1.5,1.0-1.5,1.0-1.5] , respectively. During (2) and (3), one subnet is randomly sampled from the subnet space and trained in each training iteration. Dynamic resolution is adopted throughout the training process.

Training details. The teacher pool is trained from scratch on 32 GPUs with batch size 2×32 (2 for each GPU). Synchronized BN is adopted to normalize input distribution across multiple nodes, which addresses the issue cause by small batch size. Step learning rate schedule is used throughout training. The initial learning rate and training epochs for the 3 phases are described in Table 1.

*Equal contribution
†Corresponding author: xbjxh@live.com

Phase	Initial learning rate	Epochs
Super-net training	0.12	48
Shrinking phase 1	0.04	24
Shrinking phase 2	0.04	36

Table 1: Training schedule at each phase of our ETP.

A.1.4 Details for search process

The student's architecture is fixed during the first 5 search iterations to make the search more stable. At the beginning of each search iteration, one student-teacher pair is sampled from the topk list according to the score ranking. The size of topk list is set to 5. In f_{score}, β is set to 0.8 for all base detector; α is set to 0.1 for X101 to encourage higher performance, while it is set to 0.4 for other base detectors. During fast evaluation phase, $\left\{S_{new}^{\theta'}, T_{new}\right\}$ is trained for 3 epochs under cosine learning rate schedule, where the initial learning rate is set to 0.01; the batch size is 4; synchronized BN is adopted.

A.2. Knowledge Distillation

Adaptation function. The adaptation function $f_{adap}(\cdot)$ is implemented as a 3x3 Conv layer to match the feature dimensions of the student-teacher pair. The output dimension is set to 256 and the stride is set to 1.

Proposal matching. The student and the teacher have different proposals, leading to unmatched outputs which cannot be directly distilled. We solve this by sharing student's proposals with the teacher.

A.3. Pruning

The existence of skip connections constrain the blocks in the same stage to have identical output dimensions. Thus, the channels can not be arbitrarily pruned. To address this issue, the BN's weights in projection mapping (the skip connection of the stage's first block) are used to prune the output channel of all blocks in the stage. The other channels inside the block are determined by the weights of the two BN modules at the middle.

To encourage channel sparsity, we enforce a regularization term on the weights of BN. We set the loss weight λ to be 1×10^{-5} in our implementation.

B. Encoding of the Searched Architecture

The student's backbone architecture is encoded as the output channels of each convolutional layer in each block at every stage. Blocks and stages are separated by "-" and "], [", respectively. We list out the encodings of students obtained with different base detectors and the corresponding input resolutions

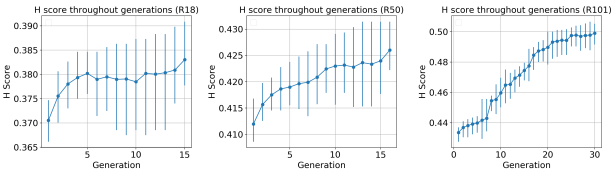

Figure 1: The H score of sampled student detectors throughout generation. Joint-DetNAS can consistently optimize the performance-complexity tradeoff for various base detectors. Weight inheritance strategy consistently improve the student's score throughout the search.

R18. Student: [(64, 64)], [(128, 128)-(128, 128)], [(256, 256)-(256, 256)], [(512, 512)-(512, 512); **Input size:** 1080×720.

R50. Student: (58, 59, 205)-(60, 64, 205)-(63, 62, 205)], [(127, 128, 314)-(109, 122, 314)-(127, 123, 314)-(125, 124, 314)], [(256, 255, 591)-(243, 245, 591)-(237, 247, 591)-(243, 246, 591)-(252, 244, 591)-(252, 254, 591)], [(509, 507, 1856)-(509, 506, 1856)-(508, 507, 1856)]; **Input size:** 1080×720.

R101. Student: [(49, 62, 202)-(35, 33, 202)-(56, 62, 202)], [(123, 128, 300)-(57, 90, 300)-(117, 113, 300)-(124, 117, 300)], [(255, 254, 321)-(65, 127, 321)-(32, 47, 321)-(32, 63, 321)-(120, 161, 321)-(132, 181, 321)-(162, 232, 321)-(175, 241, 321)-(143, 237, 321)-(199, 246, 321)-(210, 238, 321)-(201, 225, 321)-(210, 215, 321)-(211, 222, 321)-(201, 208, 321)-(198, 206, 321)-(220, 213, 321)-(226, 221, 321)-(234, 221, 321)-(237, 222, 321)], [(249, 229, 321)-(245, 231, 321)-(511, 478, 2031)-(507, 503, 2031)-(491, 477, 2031)]; **Input size:** 1080×720.

X101. Student: [(128, 128, 256)-(112, 112, 256)-(124, 124, 256)], [(256, 256, 512)-(256, 256, 512)-(256, 256, 512)-(256, 256, 512)], [(512, 512, 1024)-(448, 448, 1024)-(480, 480, 1024)-(496, 496, 1024)-(512, 512, 1024)-(464, 464, 1024)-(416, 416, 1024)-(416, 416, 1024)-(416, 416, 1024)-(416, 416, 1024)-(432, 432, 1024)-(496, 496, 1024)-(400, 400, 1024)-(400, 400, 1024)-(464, 464, 1024)-(464, 464, 1024)-(432, 432, 1024)-(352, 352, 1024)-(400, 400, 1024)-(384, 384, 1024)-(272, 272, 1024)-(384, 384, 1024)-(384, 384, 1024)], [(384, 384, 1024)-(352, 352, 1024)-(1024, 1024, 2048)-(864, 864, 2048)-(384, 384, 2048)]; **Input size:** 1333×800.

C. Illustration of the search process

In Figure 1, we show the H score (defined in Section 3.1.4 of the paper) of sampled student detectors throughout generation. The results verify that Joint-DetNAS can consistently optimize the performance-complexity tradeoff for various base detectors. In addition, weight inheritance strategy enables the student's score to be consistently improved throughout the search. We excluded 512×512 input resolution from the plot since it presents a clear performance

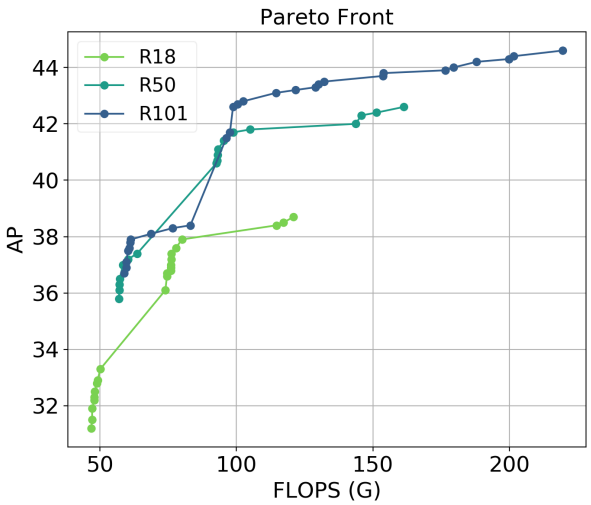

Figure 2: The Pareto optimal of various base detectors. As can be seen, R101 almost dominates both R18 and R50, indicating that given the same score function, starting with a larger base detector can often achieve better result.

Base model	Group	Input size	FLOPS (G)	FPS	AP
R18-FPN	baseline	1333 × 800	160.5	28.2	36.0
	ours	1080 × 720	**117.3**$^{-27\%}$	**33.0**$^{+17\%}$	**38.5↑39.8**
R50-FPN	baseline	1333 × 800	215.8	20.5	39.5
	ours	1080 × 720	**145.7**$^{-32\%}$	**25.4**$^{+24\%}$	**42.3↑43.2**
R101-FPN	baseline	1333 × 800	295.7	15.9	41.4
	ours	1080 × 720	**153.9**$^{-48\%}$	**23.3**$^{+47\%}$	**43.9↑44.3**
X101-FPN	baseline	1333 × 800	286.9	13.2	42.9
	ours	1333 × 800	**266.3**$^{-7\%}$	**14.0**$^{+6\%}$	**45.7↑46.0**

Table 2: The performance of found detector with post-search fine-tuning for various input base detectors. The fine-tuning lasts for 16 epochs; cosine learning rate schedule is adopted, with initial learning rate set to 0.01. The value on the left and right of ↑ are the searched detector's performance and its fine-tuned performance, respectively.

gap with other resolutions.

In Figure 2, we show the Pareto optimal of various base detectors. R101 almost dominates both R18 and R50, which indicates that given the same score function, starting with a larger base detector is often the better choice, because base detector with higher capacity can be adjusted more flexibly, thus derive a better performance-complexity tradeoff.

D. Post-search Fine-tuning Further Improves Performance

Although the obtained student detector can achieve competitive performance without additional training, we want to show that applying post-search fine-tuning to the student-teacher pair is able to further improve the student's performance. The results are demonstrated in Table 2.

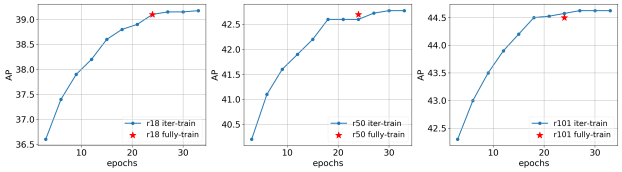

Figure 3: Comparison between iterative training and fully training. The super-net in the ETP is used as teacher for both iterative training and fully training. The result shows that: (1) the convergence speeds are comparable, and (2) the final performance of iterative training is on par with fully training.

Search Method	FLOPS	AP	#Searched architectures	Search cost (GPU days)
NAS-FPN (R50-7@256) [2]	281.3	39.9	10000	>>500
SP-NAS [5]	349.3	41.7	200	200
ours (ETP-R50)	149.1	41.9	200	**119**
ours (ETP-R101)	180.0	43	200	**120**
ours (Joint-DetNAS-R50)	**145.7**	**42.3**	100	185
ours (Joint-DetNAS-R101)	**153.9**	**43.9**	100	200

Table 3: Comparison between ETP search, our Joint-DetNAS and previous works. The results demostrate that, both ETP search and Joint-DetNAS outperform previous works: ETP search is more efficient, while Joint-DetNAS achieves higher performance.

E. Iterative Training Does Not Hurt Performance

In the framework, the student detector is trained iteratively in each search iteration during fast evaluation. Each iterative training process lasts for three epochs with cosine learning rate schedule. We comparing it with fully training in this experiment. Specifically, we fix the student detector and use the super-net in the ETP as teacher. Then we plot the change of AP with the training time for iterative training. Iterative training follows the same setting as mentioned in A.1.4. Fully training adopts 2x schedule and cosine learning rate decay, the initial learning rate is 0.02. The result in Figure 3 shows that: (1) the convergence speeds are comparable, and (2) the final performance of iterative training is on par with fully training.

F. Search with ETP

In fact, ETP can already serve as a search space, from which detectors can be directly sampled. We compare the search result of ETP with other NAS methods and our Joint-DetNAS in Table 3. The comparison shows that, both ETP search and Joint-DetNAS outperform previous works: ETP search is more efficient, while Joint-DetNAS achieves higher performance. Furthermore, the Joint-DetNAS framework is applicable for different student

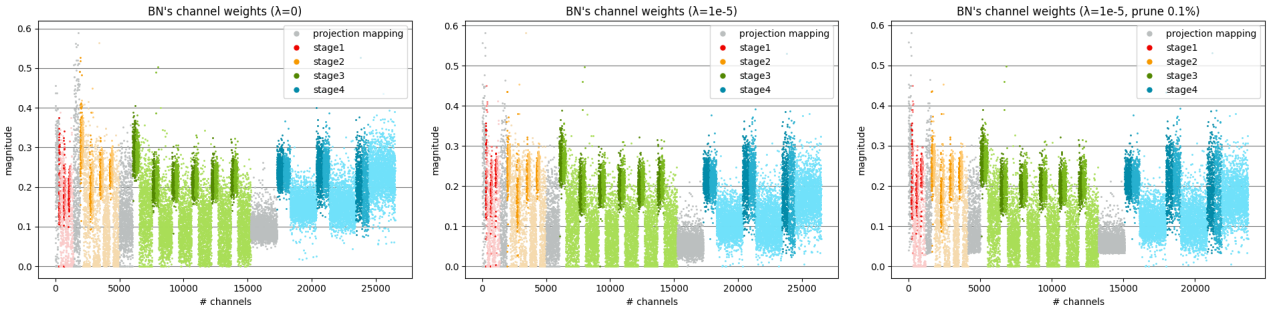

Figure 4: Analysis of BN's channel weights in the backbone. R50-FPN is used for analysis. The three graphs demonstrate the BN's weights of: **Left:** normally trained detector; **Middle:** detector trained with the regularization term, $\lambda = 1 \times 10^{-5}$; **Right:** pruning 10% channels from the backbone. More BN's channel weights are close to 0 after the regularization is enforced, which encourages sparsity.

Method	AP
Baseline R18	34.0
Whole Feature [1]	$35.2^{+1.2}$
Anchor Mask (fixed) [7]	$35.6^{+1.6}$
Gaussian Mask [6]	$35.4^{+1.4}$
Proposal Feature	$\mathbf{36.7}^{+2.7}$

Table 4: Comparison between different foreground attention mechanisms. Proposal feature outperforms the other mask based methods by a large margin. Thus, we adopt this approach in our framework. The student is trained under 1x schedule.

Proposal Feature	RCNN cls	RCNN bbox		AP
		original	class-aware	
-	-	-	-	34.0 (R18)
-	-	-	-	37.4 (R50)
✓				$36.7^{+2.7}$
	✓			$35.8^{+1.8}$
		✓		$34.8^{+0.8}$
			✓	$35.7^{+1.7}$
	✓		✓	$36.4^{+2.4}$
✓	✓		✓	$^{\dagger}\mathbf{37.9}^{+3.9}$

Table 5: Analysis on the effectiveness of each component in our KD framework. The first two rows are baseline APs of R18 and R50 FPN detectors; The student is trained under 1x schedule; † at the top left of AP indicates that the student outperforms the teacher under the same 1x training schedule.

architecture families without retraining the teacher pool, thus is more flexible and economical.

G. Ablation Study of Distillation for Object Detection

Comparison of different ways to distill feature level information. Most previous detection KD methods [1, 7, 6] aim to better distill teacher's feature level information. We compare the mask based methods with the adopted proposal feature distillation in Table 4 and found that the latter results in the most performance gain, while being the simplest to implement.

Analysis of each component in our KD framework. The ablation study of each component is shown in Table 5. Our experiments demonstrate that both feature level and prediction level distillation bring considerable improvement. We can also see that our proposed class-aware localization loss brings noticeable improvement relative to the original approach which directly distill the localization outputs. The student is R18-FPN and trained under 1x schedule, while the teacher is trained under 2x+ms schedule.

H. Ablation Study of Pruning for Object Detection

We analyze the effect of the regularization term as well as the pattern of all BNs' weights in the detector's backbone in Figure 4. As shown in the graph, more BN's channel weights are close to 0 after the regularization is enforced. In addition, BN's weights in the third projection mapping are smaller, thus causing the third stage to be pruned the most. This also indicates that the third stage contains the most redundancy.

References

[1] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems*, pages 742–751, 2017. 4

[2] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *CVPR*, 2019. 3

[3] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 1

[4] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 558–567, 2019. 1

[5] Chenhan Jiang, Hang Xu, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Sp-nas: Serial-to-parallel backbone search for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11863–11872, 2020. 3

[6] Ruoyu Sun, Fuhui Tang, Xiaopeng Zhang, Hongkai Xiong, and Qi Tian. Distilling object detectors with task adaptive regularization, 2020. 4

[7] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4933–4942, 2019. 4

[8] Sergey Zagoruyko and Nikos Komodakis. Diracnets: Training very deep neural networks without skip-connections. *arXiv preprint arXiv:1706.00388*, 2017. 1