Supplementary Materials for Disentangled Differentiable Network Pruning

A Choice of p Given Different Datasets and Architectures.

Dataset	CIFAR-10		ImageNet			
Architecture	ResNet-56	MobileNet-V2	ResNet-34	ResNet-50	MobileNet-V2	MobileNet-V3 small
p	0.48	0.56	0.55	0.38/0.31	0.67	0.75

Table A1: Choice of p for different models. p is the **remained** FLOPs divided by the total FLOPs.

We present the choice of p for all experiments in Tab A1. For ResNet-50, p is set to 0.38 when we prune 55.0% FLOPs, and p is set to 0.31 when we prune 62.0% FLOPs (results in Tab. A4)

B Architectures of g_s and g_k.

Inputs x_{sl}, $l = 1, \cdots, L$
GRU(64,128), WeightNorm, ReLU $\text{FC}_l(128, C_l)$, WeightNorm, $l = 1, \cdots, L$
Outputs $\bar{s}^l$, $l = 1, \cdots, L$

Table A2: The architecture of g_s used in our method.

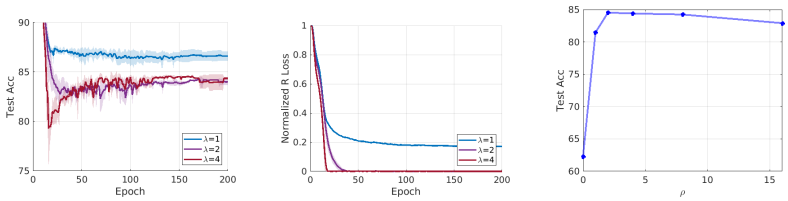
Inputs x_k,
$\text{FC}_l(32, 64)$, WeightNorm, ReLU $\text{FC}_l(64, L)$, WeightNorm
Outputs $\bar{\mathbf{k}} = [\bar{k}_1, \cdots, \bar{k}_L]$

Table A3: The architecture of g_k used in our method.

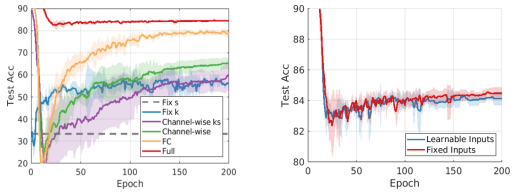
The architecture of g_s is shown in Tab. A2. The forward calculation of g_s is:

$$\begin{aligned} o_l, h_l &= \text{GRU}(x_{sl}, h_{l-1}) \\ \bar{s}^l &= \text{FC}_l(o_l), \end{aligned} \tag{1}$$

$\bar{s}^l$ is the importance score before the final output function. In most experiments, we use the sigmoid function to produce final importance score: $s^l = \text{sigmoid}(\bar{s}^l)$.



(a) Test Acc given λ (b) $\mathcal{R}$ Loss given λ (c) Effect of ρ



(d) More settings (e) Different inputs

Fig. A1: (a,b): Test accuracy and normalized $\mathcal{R}$ loss during training given different λ. (c): Test accuracy given different slection of ρ. (d): Impact to pruning with additional settings. (e): Learnable inputs vs. fixed inputs. We run each setting three times and use shaded areas to represent variance. All experiments are done on CIFAR-10 with ResNet-56.

The experiments in Figure 4 (d,b) of the main text employees two other types of output functions, and they are absolute value function (used in weight pruning papers [3]): $s^{AF} =\mid \bar{s} \mid$ and square function: $s^{SF} = \bar{s}^2$.

We also present the architecture of g_k in Tab. A3. Here, we use $\bar{k}$ to represent outputs before normalization. Given a certain layer l, the final k_l is obtained by:

$$k_l = \text{sigmoid}(\bar{k}_l + \text{b}), \tag{2}$$

where b is a positive constant to ensure we start pruning from a whole network, and $b = 3.0$ for all experiments. This is also discussed in section 5.1 of the main text.

C Additional Experiments

The test accuracy and normalized resource loss given different λ during training are shown in Fig. A1 (a,b). These figures show that a small λ may hinder the pruning process since the pruned model can not reach target FLOPs. Using a larger λ can solve this problem.

We also study the impact of ρ in Fig. A1 (c). We can see that the test accuracy of the pruned model is pretty low when $\rho = 0$, since learning of importance and width are completely disentangled. The performance of pruning also increases

Method	Base/Pruned Top-1	Base/Pruned Top-5	Δ Top-1	Δ Top-5	$\downarrow$ FLOPs
HRank [2]	76.15%/ 71.98%	92.87%/91.01%	−4.17%	−1.86%	62.1%
CC [1]	76.15%/ 74.54%	92.87%/92.25%	−1.61%	−0.62%	62.7%
CHIP [4]	76.15%/ 75.26%	92.87%/92.53%	−0.89%	−0.34%	62.8%
DDNP (ours)	76.13%/**75.56%**	92.86%/**92.68%**	**−0.57%**	**−0.18%**	62.0%

Table A4: Additional results with ResNet-50 on ImageNet when pruning more FLOPs.

when we increase ρ. But if we use a too large ρ, the performance decreases, indicating that entangling width and importance reduce pruning flexibility.

In Fig. A1 (d), we present more settings for pruning. "Fix S" means that importance s is fixed, and only the second term and third term of Eq. (9) are used. In this setting, data is no longer used for pruning, and L_1 norm is used as the channel importance. "Fix k" represents that width k is fixed, and all k_l are set to 0.5 since they are no longer learnable. "Channel-wise-ks" means that we use scalar parameterization (one variable per-channel and per-layer) for both width and importance. From the figure, we can see that parameterization for both channel importance and width can impact the pruning results, and the parameterization of channel importance has a larger impact. Moreover, "Fix S" or "Fix k" largely restrict the learning flexibility.

In Fig. A1 (e), we present the difference between learnable inputs and fixed inputs for g_k and g_s. For fixed inputs, we randomly generate x_s and x_k (in Tab. A2 and Tab. A3) before pruning and fix them during pruning. For learnedable inputs, we just randomly initialize x_s and x_k and include them as learnable parameters during pruning. It's clear that learnable inputs and fixed inputs do not have large difference.

In Tab. A4, we present additional results for ResNet-50 when the FLOPs pruning rate is larger (around 62%). Compared with state-of-the-art methods like CHIP [4] and CC [1], our method outperforms them with similar FLOPs pruning rate. The advantage of our method compared to HRank [2] is more obvious.

At last, we measure the additional costs bought by the importance and width generation networks with ResNet-56 on CIFAR-10 and ResNet-50 on ImageNet. On CIFAR-10, we measure the running time by averaging the time costs of the last 10 epochs. For ResNet-56, with scalar parameterization, the running time is 0.136 second/iteration. With importance and width generation networks, the time the running time is 0.143 second/iteration. On ImageNet, we measure the running time by averaging the time costs for the last 5 epochs. with scalar parameterization, the running time is 1.08 second/iteration. With importance and width generation networks, the running time is 1.15 second/iteration. From these results, we can see that the additional costs are trivial.

References

1. Li, Y., Lin, S., Liu, J., Ye, Q., Wang, M., Chao, F., Yang, F., Ma, J., Tian, Q., Ji, R.: Towards compact cnns via collaborative compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6438–6447 (2021)
2. Lin, M., Ji, R., Wang, Y., Zhang, Y., Zhang, B., Tian, Y., Shao, L.: Hrank: Filter pruning using high-rank feature map. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
3. Sehwag, V., Wang, S., Mittal, P., Jana, S.: Hydra: Pruning adversarially robust neural networks. In: NeurIPS (2020), `https://proceedings.neurips.cc/paper/2020/hash/e3a72c791a69f87b05ea7742e04430ed-Abstract.html`
4. Sui, Y., Yin, M., Xie, Y., Phan, H., Aliari Zonouz, S., Yuan, B.: Chip: Channel independence-based pruning for compact neural networks. Advances in Neural Information Processing Systems **34** (2021)