Supplementary Material for
Decoupled Dynamic Filter Networks

I. Inference Latency

Table A compares the inference latency between convolution, depth-wise convolution, DDF module, and DDF operation for various feature resolutions. For all resolutions, the channel size is set to 256 and the batch size is 32. The inference latencies are evaluated on single GTX 2080 Ti GPU. We also measure the inference latency of only DDF operation (DDF Op) by omitting the time of filter generation. As we can see, DDF Op is always faster than convolution. Even the entire DDF module is faster than Conv for resolutions higher than 56 pixels, whereas the standard Conv layer is faster at smaller resolutions. This is mainly because the overhead of filter generation in DDF becomes more significant when operating at low resolutions. See the 7×7 and 14×14 inference latencies of DDF, the latency drop (0.03 ms) comes from the DDF operation, while both cases use 0.81 ms to generate filters.

Table A. **Inference latency under various resolutions.**

Resolution	Conv	DwConv	DDF	DDF Op
7×7	0.21 ms	0.05 ms	0.93 ms	0.12 ms
14×14	0.40 ms	0.09 ms	0.96 ms	0.15 ms
28×28	2.31 ms	0.22 ms	1.29 ms	0.48 ms
56×56	4.09 ms	0.79 ms	2.60 ms	1.80 ms
112×112	16.04 ms	3.08 ms	9.07 ms	7.30 ms
224×224	82.57 ms	11.97 ms	37.11 ms	28.62 ms

II. Comparison with WeightNet and Involution

WeightNet [2] provides a generalization of CondConv [5]. However, it still generates *spatially-shared* convolutional filters. In other words, the filters predicted in WeightNet, while being image/channel-adaptive, are content-agnostic across spatial dimension. In contrast, DDF introduces spatial dynamic filters into dynamic depth-wise convolution, making it both spatial-adaptive and image/channel-adaptive. Involution [1] proposes to use CARAFE-like [3] modules to extract features by setting a large group size. However, CARAFE/Involution is memory-consuming when group size is large. Thanks to the proposed decoupling strategy, DDF is more computation/memory efficient and naturally supports cross-modality tasks. Table B shows different aspects of these filters.

We compare DDF with depth-wise WeightNet (DwWeightNet) in Table 4 of the main paper. We note that DwWeightNet performs even worse than *"channel-only DDF"* (see Table 3(a) of the main paper). This is because DwWeigtNet adopts sigmoid activation in the filter generation branch. Note that DDF also gets similar accuracy when using sigmoid activation (see Table 3(b) of the main paper). In practice, we find that sigmoid will cause the gradient vanishing in the filter generation branch, while the proposed filter normalization can better propagate gradients in the filter generation branch.

Table B. **Comparisons between related filters.**

	WeightNet	Involution	DDF
Spatial-adaptive	×	✓	✓
Channel-adaptive	✓	×	✓
Overhead	low	medium	low

Table C. **Performance of DDF-ResNets50 with tricks.**

Methods	Top-1 Acc
DDF-ResNet50 (base)	79.1
+ wider network	79.8
+ ResNet-D structure	80.5
+ larger inference resolution	81.3

III. DDF-ResNet with Tricks

By using wider network as ResNeXt [4], ResNet-D structure, and larger inference resolution, DDF-ResNet50 reaches 81.3% Top-1 accuracy. Table C shows the details.

References

[1] Li, duo and hu, jie and wang, changhu and li, xiangtai and she, qi and zhu, lei and zhang, tong and chen, qifeng. In *CVPR*, 2021. 1

[2] Ningning Ma, Xiangyu Zhang, Jiawei Huang, and Jian Sun. Weightnet: Revisiting the design space of weight networks. In *ECCV*, 2020. 1

[3] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe: Content-aware reassembly of features. In *ICCV*, 2019. 1

[4] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 1

[5] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In *NeurIPS*, 2019. 1