Supplementary Material: Adaptive Frequency Filters As Efficient Global Token Mixers

Zhipeng Huang[1,2*] Zhizheng Zhang[2†] Cuiling Lan[2] Zheng-Jun Zha[1] Yan Lu[2] Baining Guo[2]
[1]University of Science and Technology of China [2]Microsoft Research Asia

1. Detailed Network Architectures

As introduced in our manuscript, we build three versions of our proposed hierarchical backbone AFFNet with different channel dimensions, namely AFFNet, AFFNet-T and AFFNet-ET, respectively. Here, we provide the detailed model configurations of them in Table 1. Specifically, following commonly used designs [10, 11], we adopt a convolution stem for tokenization, which consists of a 3×3 convolution layer with a stride of 2, followed by four MBConv layers. MBConv is short for the Mobile Convolution Block in [13] with a kernel size of 3. After tokenization, three stages are cascaded as the main body of AFFNet, where each stage is composed of a MBConv layer with stride 2 for down-sampling in spatial and N_i AFF Block. Specifically, we set $N_1 = 2$, $N_2 = 4$ and $N_3 = 3$.

2. Detailed Introduction for Dataset

ImageNet [12] is a large-scale dataset with over 1.2 million images and 1000 object categories for the visual recognition challenge. It serves as the most widely used dataset for image classification. The images in this dataset are of varying sizes and resolutions, and include various objects in diverse backgrounds. We train our models on Imagenet-1k dataset from scratch to illustrate the effectiveness and efficiency of our proposed models on image classification.

MS-COCO [8] (abbreviated as COCO) is a widely used benchmark dataset for object detection, instance segmentation, and keypoint detection tasks. It contains more than 200,000 images and 80 object categories, annotated with bounding boxes, masks, and keypoints. The objects in this dataset are diverse and challenging, including people, animals, vehicles, household items, *etc.*.

ADE20k [20] is a dataset consisting of 20,210 images covering a wide range of indoor and outdoor scenes. The images in this dataset are annotated with pixel-level labels for 150 semantic categories, such as sky, road, person and so on. This dataset is widely used for evaluating the performance of deep models on semantic segmentation and scene understanding.

PASCAL VOC 2012 [5] (abbreviated as VOC) is a widely used benchmark for object recognition, object detection, and semantic segmentation. It consists of 20 object categories and contains more than 11,000 images with pixel-level annotations for object boundaries and semantic categories. This dataset is challenging due to the large variability in object appearances and the presence of occlusions and clutter within it.

3. Detailed Experiment Settings

We provide detailed experiment settings for different tasks in Table 2, including the detailed configurations for model, data and training.

4. More Experiment Results

4.1. Quantitative Results

Running speed evaluation. We report the model speeds of our proposed AFFNet models on mobile devices (iPhone) and GPUs, and compare them with other advanced lightweight models that incorporate global token mixers in Table 3. Models with similar Top-1 accuracy are grouped together for clear comparison. The latency results are equivalently measured by CoreML[1] on an iPhone with a batch size of 1. The throughput results are measured with TorchScript[2] on an A100 GPU (batch size = 128). As shown in Table 3, thanks to the AFF token mixer, AFFNet outperforms other network designs by a clear margin across different model scales. On GPUs (NVIDIA A100), AFFNet achieves 0.4% Top-1 accuracy improvement with 179 image/s lager throughput compared to the second fastest model EdgeNext-S. On the mobile device (iPhone), AFFNet also surpasses the second fastest model mobilevitv2 by 1.7% Top-1 accuracy with 0.3 ms less latency. These results reflect high effectiveness and efficiency of our proposed method.

*This work was done when Zhipeng Huang was an intern at MSRA.
†Correspondence to *zhizzhang@microsoft.com*.

[1] https://github.com/apple/coremltools
[2] https://github.com/pytorch/pytorch/blob/master/torch/csrc/jit/OVERVIEW.md

Layer / Block	Resolution	Down-sample Ratio	Number of Blocks	Number of Channels		
				AFFNet-ET	AFFNet-T	AFFNet
Image	256^2	-	1	16	16	16
Conv Stem	128^2	$\downarrow 2$	1	32	32	32
	64^2	$\downarrow 2$	4	48	48	64
Down-sampling	32^2	$\downarrow 2$	1	64	96	128
AFF Block	32^2	-	2	64	96	128
Down-sampling	16^2	$\downarrow 2$	1	104	160	256
AFF Block	16^2	-	4	104	160	256
Down-sampling	8^2	$\downarrow 2$	1	144	192	320
AFF Block	8^2	-	3	144	192	320
Parameters	-	-	-	1.4M	2.6M	5.5M
FLOPs	-	-	-	0.4G	0.8G	1.5G

Table 1. Detailed model configurations. The resolution and the number of channels in above table correspond to the output representations for each layer/block.

Task	Image Classification			Object Detection	Semantic Segmentation	
Model	AFFNet-ET	AFFNet-T	AFFNet	AFFNet	AFFNet	AFFNet
EMA	✓	✓	✓	✓	✓	✓
Weight Initialization	Kaiming normal	Kaiming normal	Kaiming normal	ImageNet-1k pretrain	ImageNet-1k pretrain	ImageNet-1k pretrain
Dataset	ImageNet-1k	ImageNet-1k	ImageNet-1k	COCO	ADE20k	PASCAL VOC
Resolution	256^2	256^2	256^2	320^2	512^2	512^2
RandAug	✗	✗	✓	✗	✗	✗
CutMix	✗	✗	✓	✗	✗	✗
MixUp	✗	✗	✓	✗	✗	✗
Random Resized Crop	✓	✓	✓	✗	✓	✓
Random Horizontal Flip	✓	✓	✓	✗	✓	✓
Random Erase	✗	✗	✓	✗	✗	✗
Gaussian Noise	✗	✗	✗	✗	✓	✓
Label Smoothing	✓	✓	✓	✗	✗	✗
Loss	CE	CE	CE	Ssd Multibox	CE	CE
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Weight Decay	0.008	0.02	0.05	0.05	0.05	0.05
Warm-up Iterations	20 k	20 K	20 k	500	500	500
LR Scheduler	Cosine	Cosine	Cosine	Cosine	Cosine	Cosine
Base LR	0.009	0.0049	0.002	0.0007	0.0005	0.0005
Minimal LR	0.0009	0.00049	0.0002	0.00007	1.00E-06	1.00E-06
Number of Epochs	300	300	300	200	120	50
Batch Size	1024	1024	1024	128	16	128

Table 2. Detailed training configurations of AFFNet, AFFNet-T, and AFFNet-ET models on different tasks. "LR" denotes the learning rate and "EMA" is short for Exponential Moving Average. For object detection and semantic segmentation tasks, AFFNet-T and AFFNet-ET use the same training configuration as AFFNet.

Evaluation on more downstream task frameworks. For the experiments reported in our main paper (*e.g.*, Table **??**), we adopt the most commonly used task frameworks, *i.e.*, SSD and Deeplabv3, in accordance with recent studies [13, 10, 11, 9] on general-purpose lightweight backbone design to ensure a fair comparison. Moreover, to evaluate the compatibility of AFFNet with more downstream task frameworks, we incorporated AFFNet into more down-

Model	Param. (M)	FLOPs (G)	Latency (ms)	Throughput (images/s)	Top-1
MViT-XXS [10]	1.3	0.4	4.8	6803	69.0
MViTv2-0.5 [11]	1.4	0.5	1.6	7021	70.2
EdgeNext-XXS [9]	1.3	0.3	1.7	7768	71.2
AFFNet-ET	1.4	0.4	1.4	8196	73.0
MViT-XS [10]	2.3	1.0	7.0	4966	74.8
MViTv2-0.75 [11]	2.9	1.0	2.4	5150	75.6
EdgeNext-XS [9]	2.3	0.5	2.6	5307	75.0
AFFNet-T	2.6	0.8	2.1	5412	77.0
CycleMLP-B1 [2]	15.2	2.1	15.2	3073	79.1
PoolFormer-S12 [19]	11.9	1.8	5.3	3922	77.2
MFormer-294 [3]	11.8	0.3	40.7	2790	77.9
MViT-S [10]	5.6	2.0	9.9	3703	78.4
MViTv2-1.0 [11]	4.9	1.8	3.4	3973	78.1
EdgeNext-S [9]	5.6	1.3	6.4	4023	79.4
AFFNet	5.5	1.5	3.1	4202	79.8

Table 3. Results of model speed evaluation. Here, the latency results are equivalently measured using CoreML on an iPhone with a batch size of 1. The throughput results are measured using TorchScript on an A100 GPU with a batch size of 128.

stream task frameworks [6, 14, 7, 18] as their encoders. These frameworks involve multi-stage/scale feature interactions via some task-specific architecture designs. By utilizing AFFNet as the encoders, these models perform consistently better compared to their vanilla versions in mAP@COCO and mIOU@ADE20K, as presented in Table 4. There results further demonstrate that our proposed AFFNet is compatible with diverse downstream task frameworks and generally applicable.

Task	Detection(mAP)		Segmentation(mIOU)	
Framework From	yolox [6]	efficientdet [14]	van [7]	moat [18]
w. Origin Encoder	32.8	40.2	38.5	41.2
w. AFFNet Encoder	35.9	41.6	43.2	41.5

Table 4. Performance evaluation on more downstream task frameworks. Our proposed AFFNet are integrated into them as their encoders to compare with their original ones.

Comparisons of different frequency transforms. We investigate the effectiveness of adopting different frequency transforms in implementing our proposed AFF token mixer. Specifically, we compare using FFT and using wavelet transform or Discrete Cosine Transform (DCT). The comparison results are in Table 5. We observe that adopting the wavelet transform also attains improvements compared to the baseline model without any frequency transforms, but it is clearly inferior to adopting FFT as we recommend. This is because the wavelet transform is a low-frequency transformation that performs our proposed filtering operation in a local space, which limits the benefits of our AFF token mixer as a global token mixer. Moreover, DCT is slightly inferior to FFT since that DCT is a Fourier-related transform with coarser transform basis. It thus leads to more information loss when mixing tokens. Besides, DCT only performs transformation only on real numbers.

Frequency Transformations	Param (M)	FLOPs (G)	Top-1
Baseline	5.5	1.5	78.4
Wavelet	5.5	1.5	78.6
DCT	5.5	1.5	79.6
FFT (Ours)	5.5	1.5	79.8

Table 5. Comparisons of adopting different frequency transforms in implementating our proposed method. "Baseline" denotes the model without any frequency transforms, "Wavelet" denotes the wavelet transforms with the Haar filters, and "DCT" is short for Discrete Cosine transform.

Order	Param (M)	FLOPs (G)	Top-1
Token-mixing first	5.5	1.5	79.7
Channel-mixing first (Ours)	5.5	1.5	79.8

Table 6. Investigation results of the effects of the order of token-mixing and channel-mixing in AFF Block. "Token-mixing first" denotes performing token mixing before channel mixing while "Channel-mixing first" is an opposite order.

The order of token-mixing and channel-mixing. We study the effect of the order of token mixing and channel mixing in backbone design. As shown in Table 6, *channel-mixing first* design is slightly superior to the *token-mixing first* design, indicating it would be better to perform within-token refinement before token mixing. Overall, they deliver very close results.

The design of channel mixer. In this paper, we focus on the design of token mixer while the channel mixer is not the main point of this work. Thus, we employ a plain channel mixer implemented by Mobilenet Convolution Block (MBConv) [13] following prior works [17, 1, 15, 18]. Here, we compare two dominated designs of the channel mixer in Table 7 for a detailed empirical study. Feed-Forward Network (FFN) [16, 4] adopts two cascaded linear layers while MBConv adds a depth-wise 3×3 convolution layer between two linear layers. We find MBConv is more powerful as the channel mixer in lightweight neural network design than FFN, in which their computational costs are almost the same.

5. Visualization Results

We present the qualitative results of AFFNet on object detection and semantic segmentation in Fig. 1 and Fig. 2, respectively. These qualitative results demonstrate that our proposed AFFNet is capable of precisely localizing and classifying objects in the dense prediction tasks with diverse object scales and complex backgrounds as a lightweight

Channel-mixing Design	Param (M)	FLOPs (G)	Top-1
FFN	5.5	1.5	79.5
MBConv (Ours)	5.5	1.5	79.8

Table 7. Comparisons of two mainstream designs for channel mixers. They are FFN (Feed-Forward Network) and MBConv (Mobilenet Convolution Block) as channel mixer. Note that the design of channel mixers is not the focus of our work, and we adopt MBConv as token mixers in our proposed method.

network design. And this demonstrates the effectiveness of our proposed AFF token mixer in preserving the spatial structure information during token mixing.

6. Limitations

Although we show the superiority of AFFNet in the running speed, We have to point out that there is still a gap between the current running speed and the theoretical upper limit of the speed it can achieve, as the speed optimization in engineering implementation of frequency transformations such as FFT/iFFT has not been fully considered yet. Besides, this work only focuses on the vision domain currently. We are looking forwards to its further extension to other research fields.

References

[1] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. In *ICLR*, 2022. 3

[2] Shoufa Chen, Enze Xie, Chongjian Ge, Ding Liang, and Ping Luo. Cyclemlp: A mlp-like architecture for dense prediction. In *ICLR*, 2022. 3

[3] Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. In *CVPR*, pages 5270–5279, 2022. 3

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

[5] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015. 1

[6] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 3

[7] Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network. *arXiv preprint arXiv:2202.09741*, 2022. 3

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[9] Muhammad Maaz, Abdelrahman Shaker, Hisham Cholakkal, Salman Khan, Syed Waqas Zamir, Rao Muhammad Anwer, and Fahad Shahbaz Khan. Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications. In *ECCV Workshops*, 2023. 2, 3

[10] Sachin Mehta and Mohammad Rastegari. Mobilevit: lightweight, general-purpose, and mobile-friendly vision transformer. In *ICLR*, 2022. 1, 2, 3

[11] Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers. *TMLR*, 2022. 1, 2, 3

[12] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 1

[13] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 1, 2, 3

[14] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, pages 10781–10790, 2020. 3

[15] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *ECCV*, 2022. 3

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 3

[17] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *CVM*, 2022. 3

[18] Chenglin Yang, Siyuan Qiao, Qihang Yu, Xiaoding Yuan, Yukun Zhu, Alan Yuille, Hartwig Adam, and Liang-Chieh Chen. Moat: Alternating mobile convolution and attention brings strong vision models. In *ICLR*, 2023. 3

[19] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *CVPR*, 2022. 3

[20] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. In *IJCV*, 2019. 1

Figure 1. Qualitative results of the detection model with our AFFNet as the backbone on the validation set of COCO dataset.

Figure 2. Qualitative results of the segmentation model with our AFFNet as the backbone on unseen validation set of COCO dataset. This model is trained on the Pascal VOC dataset with 20 segmentation classes.