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
{zhizzhang, culan, yanlu, bainguo}@microsoft.com
hzp1104@mail.ustc.edu.cn zhazj@ustc.edu.cn

Abstract

Recent vision transformers, large-kernel CNNs and MLPs have attained remarkable successes in broad vision tasks thanks to their effective information fusion in the global scope. However, their efficient deployments, especially on mobile devices, still suffer from noteworthy challenges due to the heavy computational costs of self-attention mechanisms, large kernels, or fully connected layers. In this work, we apply conventional convolution theorem to deep learning for addressing this and reveal that adaptive frequency filters can serve as efficient global token mixers. With this insight, we propose Adaptive Frequency Filtering (AFF) token mixer. This neural operator transfers a latent representation to the frequency domain via a Fourier transform and performs semantic-adaptive frequency filtering via an elementwise multiplication, which mathematically equals to a token mixing operation in the original latent space with a dynamic convolution kernel as large as the spatial resolution of this latent representation. We take AFF token mixers as primary neural operators to build a lightweight neural network, dubbed AFFNet. Extensive experiments demonstrate the effectiveness of our proposed AFF token mixer and show that AFFNet achieve superior accuracy and efficiency trade-offs compared to other lightweight network designs on broad visual tasks, including visual recognition and dense prediction tasks.

1. Introduction

Remarkable progress has been made in ever-changing vision network designs to date, wherein effective token mixing in the global scope is constantly highlighted. Three existing dominant network families, *i.e*., Transformers, CNNs and MLPs achieve global token mixing with their respective ways. Transformers [17, 67, 43, 13, 83] mix tokens with self-attention mechanisms where pairwise correlations between query-key pairs are taken as mixing weights. CNNs achieve competitive performance with transformers by scaling up their kernel sizes [54, 16, 41, 10]. MLPs [64, 28, 38] provide another powerful paradigm via fully connections across all tokens. All of them are effective but computationally expensive, imposing remarkable challenges in practical deployments, especially on edge devices.

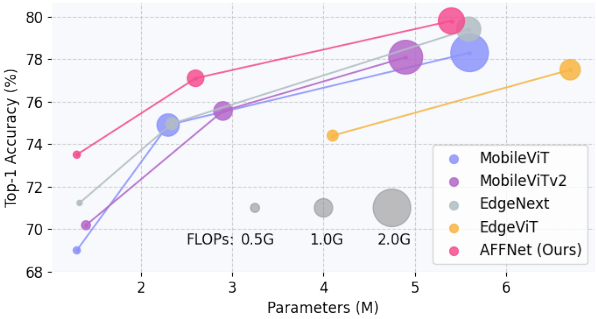


Figure 1. Comparison of Top-1 accuracy on ImageNet-1K [58] between our proposed AFFNet to some state-of-the-art lightweight networks that have global token mixing. The bubble size corresponds to FLOPs.

Recently, there is increased attention on improving the efficiency of token mixing in transformers. Some works [33, 43, 13, 25, 48, 53, 34] squeeze the scope of token mixing in different ways to compromise the representation capacities of neural networks for their efficiencies. Other works reduce the complexity of the matrix operations in self-attention by making use of the associativity property of matrix products [32] or low-rank approximation methods [23, 78]. These methods all sacrifice the expressiveness of neural networks and lead to unsatisfactory performance of efficient network designs. A general-purpose global token mixing for lightweight networks is still less explored. Better trade-off between accuracy and efficiency for global-scope token mixing is worthy of further study.

In this work, we reveal that *adaptive frequency filters can serve as efficient global token mixers*, inspired by the convolution theorem [47, 56, 51] widely used in conventional signal processing. This theorem states that a convolution

*This work was done when Zhipeng Huang was an intern at Microsoft Research Asia.

in one domain mathematically equals the Hadamard product (also known as elementwise product) in its corresponding Fourier domain. This equivalence allows us to frame global token mixing as a large-kernel convolution in the latent space and efficiently implement this convolution with a Hadamard product operation in the frequency domain by performing Fourier transforms on tokens in the latent space.

Besides large scopes, the adaptability to semantics also matters for token mixing as studied in [14, 9, 74, 1, 76]. This means that the weights for token mixing should be instance-adaptive. Moreover, different semantic attributes of the learned latent representations distribute in different channels [1, 77]. This property poses requirements for channel-specific token mixing wherein the weights of token mixing vary across different channels. From the perspective of framing global adaptive token mixing as a convolution, the kernel of this convolution operation should be not only large but also spatially dynamic. However, it is well known that dynamic convolutions are computationally expensive in common. Large-kernel dynamic convolutions seem extremely prohibitive for efficient/lightweight network designs. In this paper, we propose to adopt frequency filtering in the Fourier domain with learned instance-adaptive masks as a mathematical equivalent of token mixing using large-kernel dynamic convolutions by making use of the aforementioned convolution theorem. This equivalent could reduce the complexity of token mixing from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$ thanks to adopting Fast Fourier Transforms (FFT), which is more computationally efficient.

With the key insight above, we propose Adaptive Frequency Filtering (AFF) token mixer. In this neural operator, the latent representations (*i.e.*, a set of tokens) are transferred from its original latent space to a frequency space via a 2D discrete Fourier transform applied spatially. In this way, we get the frequency representations whose spatial positions correspond to different frequency components. We adopt an extremely lightweight network to learn instance-adaptive masks from these frequency representations, and then calculate the Hadamard product between the learned masks and the frequency representations for adaptive frequency filtering. The filtered representations are transferred back to the original latent space via an inverse Fourier transform. The features after this inverse transform could be viewed as the results of token mixing with depthwise convolution kernels whose spatial dimensions are as large as those of latent representations (*i.e.*, the token set). According to the convolution theorem [47], our proposed operation mathematically equals to taking the tensors of applying an inverse Fourier transform to the learned masks in the Fourier domain as the corresponding kernel weights and perform convolution with this kernel in the original domain. Detailed introduction, demonstration and analysis are given in subsequent sections.

Furthermore, we take the proposed AFF token mixer as the primary neural operator and assemble it into an AFF block together with a plain channel mixer. AFF blocks serve as the basic units for constructing efficient vision backbone, dubbed AFFNet. We evaluate the effectiveness and efficiency of our proposed AFF token mixer by conducting extensive ablation study and comparison across diverse vision tasks and model scales.

Our contributions can be summarized in the following:

- We reveal that adaptive frequency filtering in the latent space can serve as efficient global token mixing with large dynamic kernels, and propose Adaptive Frequency Filtering (AFF) token mixer.
- We conduct theoretical analysis and empirical study to compare our proposed AFF token mixer with other related frequency-domain neural operators from the perspective of information fusion for figuring out what really matters for the effects of token mixing.
- We take AFF token mixer as the primary neural operator to build a lightweight vision backbone AFFNet. AFFNet achieves the state-of-the-art accuracy and efficiency trade-offs compared to other lightweight network designs across a broad range of vision tasks. An experimental evidence is provided in Fig.1.

2. Related Work

2.1. Token Mixing in Deep Learning

Mainstream neural network families, *i.e.*, CNNs, Transformers, MLPs, differ in their ways of token mixing, as detailed in [75]. CNNs [52] mix tokens with the learned weights of convolution kernels where the spatial kernel size determines the mixing scope. Commonly, the weights are deterministic and the scope is commonly a local one. Transformers [70, 17] mix tokens with pairwise correlations between query and key tokens in a local [43, 13] or global[17, 67] range. These weights are semantic-adaptive but computationally expensive due to the $\mathcal{O}(N^2)$ complexity. MLPs commonly mix tokens with deterministic weights in manually designed scopes [6, 66, 65, 85] wherein the weights are the network parameters. This work aims to design a generally applicable token mixer for lightweight neural networks with three merits: computation-efficient, semantic-adaptive and effective in the global scope.

2.2. Lightweight Neural Networks

Lightweight neural network designs have been of high values for practical deployments. CNNs, Transformers, and MLPs have their own efficient designs. MobileNets series [30, 59, 29] introduce depthwise and pointwise convolutions as well as modified architectures for improving the efficiency. Shufflenet series [88, 44] further improve pointwise convolution via shuffle operations. MobileViT

[48] combines lightweight MobileNet block and multi-head self-attention blocks. Its follow-up versions further improve it with a linear-complexity self-attention method [49]. Besides, there are many works reducing the complexity of self-attention via reducing the region of token mixing [43, 13, 53, 34] or various mathematical approximations [23, 78, 45]. Many efficient MLPs limit the scope of token mixing to horizontal and vertical stripes [86, 28, 63] or a manually designed region [7].

2.3. Frequency-domain Deep Learning

Frequency-domain analysis has been a classical tool for conventional signal processing [2, 55] for a long time. Recently, frequency-domain methods begin to be introduced to the field of deep learning for analyzing the optimization [80, 82] and generalization [71, 79] capabilities of Deep Neural Networks (DNNs). Besides these, frequency-domain methods are integrated into DNNs to learn non-local [12, 57, 37, 21] or domain-generalizable [39] representations. Our proposed method might be similar to them at first glance but actually differs from them in both modelling perspectives and architecture designs. These five works propose different frequency-domain operations by introducing convolutions [12], elementwise multiplication with trainable network parameters [57], matrix multiplication with trainable parameters [37], groupwise MLP layers [21] and elementwise multiplication with spatial instance-adaptive masks [39] to frequency-domain representations, respectively. All of them are not designed for the same purpose with ours. We provide detailed mathematical analysis on their shortcomings as token mixers and conduct extensive experimental comparisons in the following sections.

3. Method

We first describe a unified formulation of token mixing, then introduce our proposed Adaptive Frequency Filtering (AFF) token mixer. We further analyze what properties matter for a frequency-domain operation in terms of its effects on token mixing. We finally introduce AFFNet which is a lightweight backbone with AFF token mixer as its core.

3.1. Unified Formulation of Token Mixing

Token mixing is of high importance since learning non-local representations is critical for visual understanding [73, 17, 67]. In most mainstream neural networks, the input image is firstly patchified into a feature tensor $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ whose spatial resolution is $H \times W$ and the number of channels is C. This feature tensor could be viewed as a set of tokens, in which each token can be denoted as $\mathbf{x} \in \mathbb{R}^{1\times 1\times C}$. The updated token for a query $\mathbf{x}^q$ after token mixing in its contextual region $\mathcal{N}(\mathbf{x}^q)$ can be formulated in a unified form:

$$\hat{\mathbf{x}^q} = \sum_{i\in\mathcal{N}(\mathbf{x}^q)} \boldsymbol{\omega}^{i\to q} \times \phi(\mathbf{x}^i), \tag{1}$$

where $\hat{\mathbf{x}^q}$ refers to the updated $\mathbf{x}^q$ and $\mathbf{x}^i$ refers to the tokens in $\mathcal{N}(\mathbf{x}^q)$. $\phi(\cdot)$ denotes the embeding functions. $\boldsymbol{\omega}^{i\to q}$ represents the weights of information fusion from token $\mathbf{x}^i$ to the updated $\mathbf{x}^q$. The symbol $\times$ could be Hadamard product or matrix multiplication.

We revisit the prevailing token mixing methods in different types of network architectures in terms of their effectiveness and efficiency. For CNNs, tokens are mixed by matrix multiplication with deterministic network parameters as the mixing weights. Here, the kernel sizes of convolutions determine the scopes of token mixing. This makes mixing in a global scope quite costly due to the quadratically increased parameters and FLOPs as the kernel size increases. Transformers mix tokens with pairwise correlations between query and key tokens. Its computational complexity is $\mathcal{O}(N^2)$ (N is the total number of tokens), limiting its applications in lightweight networks. Like CNNs, MLPs also mix tokens with deterministic network parameters. The scope of token mixing in advanced MLPs [6, 66, 65, 85] are commonly manually design, where the globality comes at the cost of huge computational complexity. They are all not specifically designed for lightweight neural networks.

This work aims to deign a *computationally efficient*, *semantically adaptive* and *global-scope* token mixer for lightweight networks. This requires a large $\mathcal{N}(\mathbf{x}^q)$ and instance-adaptive $\boldsymbol{\omega}^{i\to q}$ with less network parameters and low computation costs as possible.

3.2. Adaptive Frequency Filtering Token Mixer

We apply the convolution theorem [47, 56, 51] to deep learning for designing a token mixer with aforementioned merits for lightweight neural networks. Based on this theorem, we reveal that *adaptive frequency filters can serve as efficient global token mixers*. In the following, we introduce its mathematical modelling, architecture design and the equivalence between them for our proposed token mixer.

Modelling. To simplify understanding, we frame token mixing in the form of global convolution, succinctly denoted by $\hat{\mathbf{X}} = \mathcal{K}*\mathbf{X}$. For the query token at position (h, w), *i.e.*, $\mathbf{X}(h, w)$, Eq.(1) can be reformulated as:

$$\hat{\mathbf{X}}(h,w) = \sum_{h'=-\lfloor\frac{H}{2}\rfloor}^{\lfloor\frac{H}{2}\rfloor} \sum_{w'=-\lfloor\frac{W}{2}\rfloor}^{\lfloor\frac{W}{2}\rfloor} \mathcal{K}(h',w')\mathbf{X}(h-h',w-w'), \tag{2}$$

where $\hat{\mathbf{X}}(h, w)$ represents the updated token for $\mathbf{X}(h, w)$ after token mixing. H and W are the height and weight of the input tensor, respectively. $\mathcal{K}(h', w')$ denotes the weights

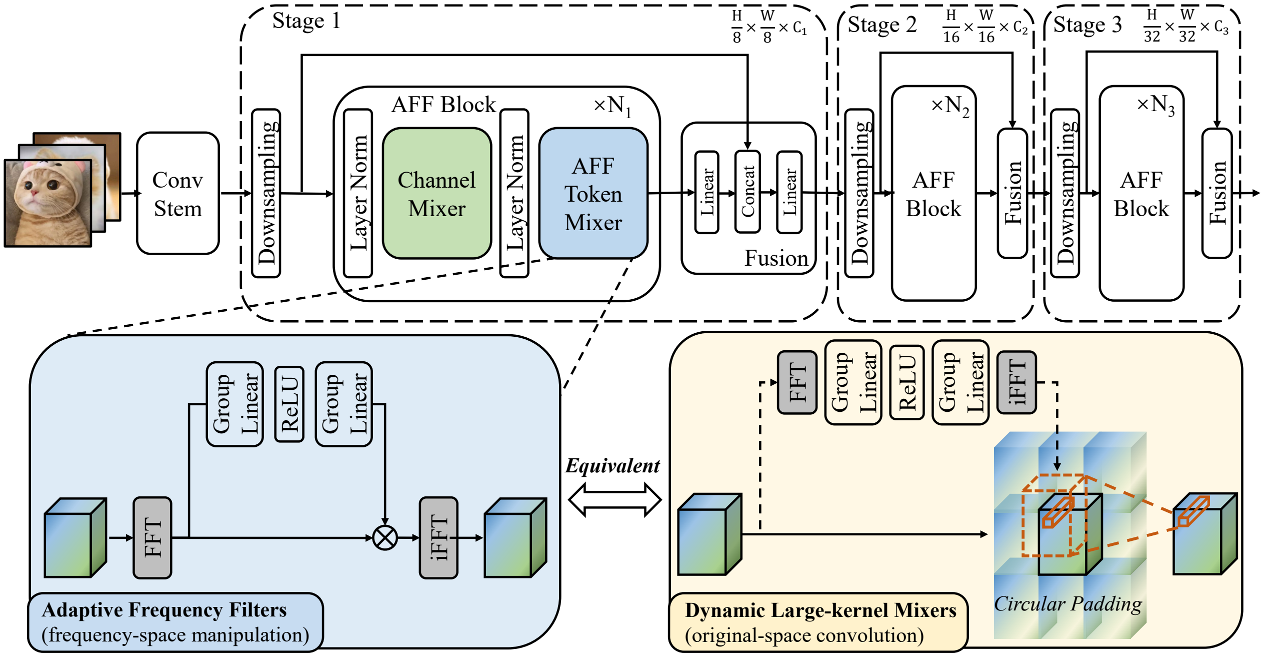


Figure 2. Illustration of our proposed AFF token mixer and its corresponding network AFFNet. The AFF token mixer is implemented by adaptive frequency filters at the bottom left and mathematically equals to the mixing operation at the bottom right. This operation can be viewed as token mixing with a large-kernel dynamic convolution where the kernel weights are inferred by the sub-network as shown in the bottom right sub-figure.

for token mixing, implemented by a global convolution kernel which has the same spatial size with $\mathbf{X}$. The padding operation for $\mathbf{X}$ is omitted here for simplicity and the specific padding method is introduced in the subsequent parts.

With the expectation for our proposed token mixer as a *semantic-adaptive* and *global-scope* one, the weights $\mathcal{K}$ for token mixing should be adaptive to $\mathbf{X}$ and of large spatial size. As illustrated by the lower right subfigure in Fig.2, a straightforward way for enabling $\mathcal{K}$ adaptive to $\mathbf{X}$ is to implement it with a dynamic convolution kernel [31, 9, 26, 87], *i.e.*, inferring weights of $\mathcal{K}$ with $\mathbf{X}$ as the inputs of a sub-network. However, adopting dynamic convolutions is usually computational costly, even more so, when using large-kernel ones. This thus imposes big challenges in designing an efficient token mixer for lightweight networks along this way. Next, we introduce an efficient method as its equivalent implementation by making use of the convolution theorem [47].

Architecture. The convolution theorem [47, 51, 56] for inverse Fourier transform states that a convolution in one domain mathematically equals the Hadamard product in its corresponding Fourier domain. This inspires us to propose a lightweight and fast architecture (illustrated by the lower left part of Fig.2) as an extremely efficient implementation of our modelling above.

Given feature $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$, *i.e.*, a set of tokens in the latent space, we adopt Fast Fourier Transform (FFT) to obtain the corresponding frequency representations $\mathbf{X}_F$ by $\mathbf{X}_F = \mathcal{F}(\mathbf{X})$. The detailed formulation of $\mathcal{F}(\cdot)$ is:

$$\mathbf{X}_F(u, v) = \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} \mathbf{X}(h, w) e^{-2\pi i(uh+vw)}. \tag{3}$$

As indicated by Eq.(3), features of different spatial positions in $\mathbf{X}_F$ correspond to different frequency components of $\mathbf{X}$. They incorporate global information from $\mathbf{X}$ with a transform of $\mathcal{O}(N \log N)$ complexity.

We apply the aforementioned convolution theorem to achieve efficient global token mixing for $\mathbf{X}$ by filtering its frequency representation $\mathbf{X}_F$ with a learnable instance-adaptive mask. We further adopt inverse FFT to the filtered $\mathbf{X}_F$ for getting the updated feature representations $\hat{\mathbf{X}}$ in the original latent space. This process can be formulated as:

$$\hat{\mathbf{X}} = \mathcal{F}^{-1}[\mathcal{M}(\mathcal{F}(\mathbf{X})) \odot \mathcal{F}(\mathbf{X})], \tag{4}$$

where $\mathcal{M}(\mathcal{F}(\mathbf{X}))$ is the mask tensor learned from $\mathbf{X}_F$, which has the same shape with $\mathbf{X}_F$. As shown in the lower left subfigure in Fig.2, to make the network lightweight as possible, $\mathcal{M}(\cdot)$ is efficiently implemented by a group 1×1 convolution (linear) layer, followed by a ReLU function and another group linear layer. $\odot$ denotes Hadamard product, also known as elementwise multiplication, and $\mathcal{F}^{-1}(\cdot)$ denotes inverse Fourier transform. Here, $\hat{\mathbf{X}}$ can be viewed as the results of global adaptive token mixing for $\mathbf{X}$, which is mathematically equivalent to adopting a large-size dynamic convolution kernel as the weights for token mixing. The equivalence is introduced in the following.

Equivalence. The convolution theorem still applies to the latent representations of neural networks. The multiplication of two signals in the Fourier domain equals to the Fourier transform of a convolution of these two signals in their original domain. When applying this to the frequency-domain multiplication in Fig.(2), we know that:

$$\mathcal{M}(\mathcal{F}(\mathbf{X})) \odot \mathcal{F}(\mathbf{X}) = \mathcal{F}\{\mathcal{F}^{-1}[\mathcal{M}(\mathcal{F}(\mathbf{X}))] * \mathbf{X}\}. \quad (5)$$

Combining Eq.(4) and Eq.(5), it is easy to get that:

$$\hat{\mathbf{X}} = \mathcal{F}^{-1}[\mathcal{M}(\mathcal{F}(\mathbf{X}))] * \mathbf{X}, \quad (6)$$

where $\mathcal{F}^{-1}(\mathcal{M}(\mathcal{F}(\mathbf{X})))$ is a tensor of the same shape with $\mathbf{X}$, which could be viewed as a dynamic depthwise convolution kernel as large as $\mathbf{X}$ in spatial. This kernel is adaptive to the contents of $\mathbf{X}$. Due to the property of Fourier transform [47], a circular padding is adopted to $\mathbf{X}$ here as shown in Fig.2. So far, we understand why the operation in Eq.(4) mathematically equals to a global-scope token mixing operation with semantic-adaptive weights.

3.3. Analysis

As introduced in Sec.2.3, there have been some studies applying frequency-domain methods to DNN for learning non-local or domain-generalizable representations in previous works [12, 57, 37, 21, 39]. They are all designed for different purposes with ours. In this section, we revisit the frequency-domain operations in these works from the perspective of token mixing and compare our design with them.

FFC [12] and AFNO [21] adopt linear (also known as 1×1 convolution) layers with non-linear activation functions to the representations in the frequency domain. Specifically, AFNO [21] adopt a linear layer followed by a ReLU function, another linear layer and a SoftShrink[1] function to the frequency representations after Fourier transforms, which can be briefly described as *FFT→Linear→ReLU→Linear→SoftShrink→iFFT*. Here, linear layer and Fourier transform are in fact commutative, *i.e.*, $\text{Linear}(\mathcal{F}(\mathbf{X})) = \mathcal{F}(\text{Linear}(\mathbf{X}))$, which can be proved with the distributive property of matrix multiplication by:

$$\begin{aligned} &\mathbf{W}_{Linear} \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} \mathbf{X}(h,w) e^{-2\pi i(uh+vw)} \\ &= \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} \left(\mathbf{W}_{Linear}\mathbf{X}(h,w)\right) e^{-2\pi i(uh+vw)}, \end{aligned} \quad (7)$$

where $\mathbf{W}_{Linear}$ denotes the parameters of a linear layer. We know that successive Fourier transform and its inverse transform equal to an identity function. Thus, the architecture of AFNO could be rewrote as: *FFT→Linear→ReLU→(iFFT→FFT)→Linear→SoftShrink→iFFT*. Upon the commutative law proved in Eq.(7), we can know this architecture is in fact equivalent to *Linear→**FFT→ReLU→iFFT**→Linear→**FFT→SoftShrink→iFFT***. Now, it is easy to find that only ReLU and SoftShrink functions remain in the Fourier domain. These two deterministic functions cannot achieve semantic-adaptive filtering as our proposed AFF token mixer does. The same problem also exists in FFC [12].

Properties	FFC	AFNO	GFNet	FNO	DFF	Ours
Semantic-adaptive	✗	✗	✗	✗	✓	✓
Variable-size input	✓	✓	✗	✗	✓	✓
Channel-wise mix	✓	✓	✓	✓	✗	✓

Table 1. Comparisons of our proposed AFF token mixer with other frequency-domain neural operators in terms of three important properties for token mixing.

GFNet [57] and FNO [37] multiply the representations after Fourier transforms with trainable network parameters. GFNet [57] adopts elementwise multiplication while FNO [37] uses matrix multiplication. Both of them are not semantic-adaptive since the masks implemented by network parameters are shared over different instances and fixed after training. Besides, they cannot support for variable-size inputs since the shapes of these masks are fixed, leading to the lack of flexibility in their practical using.

DFF [39] learns a spatial mask to filter out frequency components that are not conductive to domain generalization. It is proposed for domain generalization problems in which only spatial mask is needed as studied in [39] since different spatial position of the features after a Fourier transform correspond to different frequency components. However, it is not competent as a token mixer since the learned mask is shared along the channel dimension. This means that the weights for its equivalent token mixing are shared for different channels. However, different channels commonly represent different semantic attributes [1, 77], thus requiring adaptive weights in token mixing.

We summarize the comparisons of different frequency-domain designs in terms of three important properties for token mixing in Table 1. The results of experimental verification are in Table 5 as follows.

3.4. Network Architectures

With our AFF token mixer as the core neural operator, we introduce its corresponding module and network design.

AFF Block For the output $\mathbf{X}^{l-1}$ of the $(l-1)$-th AFF Block, we adopt the commonly used module MBConv [49,

[1] https://pytorch.org/docs/stable/generated/torch.nn.Softshrink.html

60, 61, 48, 68] with Layer Normalization (LN) for channel mixing, then feed it to our proposed AFF token mixer for global token mixing to get the output of l-th AFF block. Skip-connections for channel mixing and token mixing are adopted to facilitate model training. The entire architecture of AFF Block can be formulated as:

$$\begin{aligned} \hat{\mathbf{X}}^l &= \text{MBConv}^l\left(\text{LN}\left(\mathbf{X}^{l-l}\right)\right) + \mathbf{X}^{l-l} \\ \mathbf{X}^l &= \text{AFF}^l\left(\text{LN}\left(\hat{\mathbf{X}}^l\right)\right) + \hat{\mathbf{X}}^l \end{aligned} \tag{8}$$

AFFNet We stack multiple AFF blocks for constructing a lightweight backbone network, namely AFFNet, as shown in Fig.2. Following the common practices [48, 49], we employ a convolution stem for tokenization and a plain fusion for combining local and global features at each stage. We build three versions of AFFNet with different numbers of channels, yielding different parameter scales. AFFNet and its tiny (AFFNet-T) and extremely tiny (AFFNet-ET) versions have 5.5M, 2.6M and 1.4M parameters, respectively. Their detailed configurations are in the Supplementary.

4. Experiments

We evaluate our proposed AFF token mixer by conducting comparisons with the state-of-the-art lightweight networks and extensive ablation studies for its design.

4.1. Image Classification

Settings. We train different versions of our proposed lightweight networks AFFNet as backbones on ImageNet-1k dataset [58] from scratch. All models are trained for 300 epochs on 8 NVIDIA V100 GPUs with a batch size of 1024. More implementation details are in the Supplementary.

Results. We report the comparison results between our proposed AFFNet and other SOTA lightweight models in Table 2. We observe that our AFFNet outperforms other lightweight networks with comparable model sizes in Top-1 accuracy. The AFFNet reaches 79.8% Top-1 accuracy with 5.5M parameters and 1.5G FLOPs. Our extremely tiny model AFFNet-ET attains 73% Top-1 accuracy with sorely 1.4M and 0.4G FLOPs. As a result, AFFNet achieves the best trade-offs between accuracy and efficiency. To show the comparison results more intuitively, we illustrate the accuracy and efficiency trade-offs of our AFFNet and some advanced lightweight models with global token mixers in Fig. 1. Thanks to AFF token mixer, AFFNet is superior to them by a clear margin across different model scales. Its superiority is especially significant when the model is extremely tiny, which demonstrates the effectiveness of AFF token mixer on information fusion at very low costs. AFFNet, AFFNet-T, and AFFNet-ET models achieve 4202, 5304, and 7470 images/s thoughtput on ImageNet-1K tested with one NVIDIA A100 GPU, respectively, which is 13.5%, 8.2%, and 14.9% faster than MobileViT-S/XS/XXS. More detailed results are in the Supplementary.

Model	Pub.	Res.	Param. (M)	FLOPs (G)	Top-1
MNetv1-0.5 [30]	arXiv17	224^2	1.3	0.2	63.7
MViT-XXS [48]	ICLR22	256^2	1.3	0.4	69.0
EdgeNext-XXS [46]	ECCV22	256^2	1.3	0.3	71.2
MViTv2-0.5 [49]	TMLR23	256^2	1.4	0.5	70.2
AFFNet-ET	-	256^2	1.4	0.4	73.0
MNetv3-L-0.5 [29]	ICCV19	224^2	2.6	0.1	68.8
MFormer-52 [8]	CVPR22	224^2	3.6	0.1	68.7
PVTv2-B0 [72]	CVM22	224^2	3.7	0.6	70.5
MViT-XS [48]	ICLR22	256^2	2.3	1.0	74.8
EdgeNext-XS [46]	ECCV22	256^2	2.3	0.5	75.0
EFormer-S0 [35]	arXiv22	224^2	3.5	0.4	75.7
MViTv2-0.75 [49]	TMLR23	256^2	2.9	1.0	75.6
AFFNet-T	-	256^2	2.6	0.8	77.0
MNetv2 [59]	CVPR18	224^2	6.9	0.6	74.7
ShuffleNetV2 [44]	ECCV18	224^2	5.5	0.6	74.5
MNetv3 [29]	ICCV19	224^2	5.4	0.2	75.2
T2T-ViT [84]	ICCV21	224^2	6.9	1.8	76.5
DeiT-T [67]	ICML21	224^2	5.7	1.3	72.2
CoaT-Lite-T [15]	ICCV21	224^2	5.7	1.6	77.5
LeViT-128 [20]	ICCV21	224^2	9.2	0.4	78.6
GFNet-Ti [57]	NeurIPS21	224^2	7.0	1.3	74.6
EFormer-L1 [36]	NeurIPS22	224^2	12.3	1.3	79.2
EFormer-S1 [35]	arXiv22	224^2	6.1	0.7	79.0
Mformer [8]	CVPR22	224^2	9.4	0.2	76.7
EfficientViT [3]	arXiv22	224^2	7.9	0.4	78.6
EdgeViT-XS [11]	ECCV22	256^2	6.7	1.1	77.5
MOne-S3 [69]	arXiv22	224^2	10.1	1.9	78.1
MViT-S [48]	ICLR22	256^2	5.6	2.0	78.4
EdgeNext-S [46]	ECCV22	256^2	5.6	1.3	79.4
MViTv2-1.0 [49]	TMLR23	256^2	4.9	1.8	78.1
AFFNet	-	256^2	5.5	1.5	79.8

Table 2. Comparisons of our proposed AFFNet with other state-of-the-art lightweight networks on ImageNet-1K classification over different model scales (*i.e.*, <2M, 2M $\sim$ 4M and $>$ 4M). For conciseness, Pub., Res., Param., MNet, MOne, MFormer, EFormer and MViT are short for Publication, Resolution, Parameters, MobileNet, MobileOne, MobileFormer, EfficientFormer and MobileViT, respectively.

4.2. Object Detection

Settings. We conduct object detection experiments on MS-COCO dataset [40], Following the common practices in [30, 59, 48, 49, 46], we compare different lightweight backbones upon the Single Shot Detection (SSD) [42] framework wherein separable convolutions are adopted to replace the standard convolutions in the detection head for evaluation in the lightweight setting. In the training, we load ImageNet-1K pre-trained weights as the initialization of the backbone network, and fine-tune the entire model on the training set of MS-COCO with the AdamW optimizer for 200 epochs. The input resolution of the images is 320×320. Detailed introduction for the used dataset and

Model	Detection		Segmentation		
	Param.	mAP(%) COCO	Param.	mIOU(%) ADE20K	VOC
MViT-XXS [48]	1.9	18.5	1.9	-	73.6
MViTv2-0.5 [49]	2.0	21.2	3.6	31.2	75.1
AFFNet-ET	1.9	21.8	2.2	33.0	76.1
MViT-XS [48]	2.7	24.8	2.9	-	77.1
MViTv2-0.75 [49]	3.6	24.6	6.2	34.7	75.1
AFFNet-T	3.0	25.3	3.5	36.9	77.8
ResNet-50 [27]	22.9	25.2	68.2	36.2	76.8
MNetv1 [30]	5.1	22.2	11.2	-	75.3
MNetv2 [59]	4.3	22.1	18.7	34.1	75.7
MViT-S [48]	5.7	27.7	6.4	-	79.1
MViTv2-1.0 [49]	5.6	26.5	9.4	37.0	78.9
EdgeNext [46]	6.2	27.9	6.5	-	80.2
AFFNet	5.6	28.4	6.9	38.4	80.5

Table 3. Comparisons of our AFFNet with other state-of-the-art models for object detection on COCO dataset, and segmentation on ADE20k and VOC dataset. Here, Param., MNet and MViT are short for Paramters, MobileNet and MobileViT, respectively.

more implementation details are in the Supplementary.

Results. As shown in Table 3, the detection models equipped with AFFNet consistently outperforms other lightweight CNNs or transformers based detectors in mAP across different model scales. Specifically, AFFNet surpasses the second-best EdgeNext [46] by 0.5% in mAP with 0.6M fewer parameters, and surpasses the model with ResNet-50 backbone by 3.2% in mAP using about 1/4 of parameters. Our smallest model AFFNet-ET outperforms the second-best model with comparable parameters MobileViTv2-0.5 [49] by 0.6% in mAP with fewer parameters. These results demonstrate the effectiveness of our proposed method on capturing spatial location information required by the task of object detection at low costs.

4.3. Semantic Segmentation

Settings. We conduct semantic segmentation experiments on two benchmarks datasets ADE20k [89] and PASCAL VOC 2012 [18] (abbreviated as VOC). For the experiments on VOC dataset, we follow the common practices in [5, 50] to extend the training data with more annotations and data from [24] and [40], respectively. The widely used semantic segmentation framework DeepLabv3 [5] is employed for experiments with different backbones. The input resolution of the images is set to 512×512 and the ImageNet-1K pretrained weights are loaded as model initialization. All models were trained for 120 and 50 epochs on ADE20K and VOC dataset, respectively. Please see our Supplementary for more detailed introduction.

Results. As the results shown in Table 3, AFFNet performs clearly better than other lightweight networks on these two datasets. Our AFFNet outperforms the second-best lightweight network MobileViTv2-1.0 [49] by 1.4% in mIOU on ADE20K, and outperforms the second-best lightweight model EdgeNext [46] by 0.3% in mIOU on VOC. Besides, it achieves large improvements (2.2% mIOU on ADE20K, 3.7% mIOU on VOC) relative to the representative CNN model (*i.e.*, ResNet-50) with about 10% of the parameters of ResNet-50. These exhibit the effectiveness of our proposed method on dense prediction tasks.

Method	Param (M)	FLOPs (G)	Top-1
Base.	5.2	1.3	77.9
Base. + Conv-mixer (3×3)	10.7	2.7	78.6
Base. + AFF w/o FFT	5.5	1.5	78.4
Base. + AFF (Our AFFNet)	5.5	1.5	79.8

Table 4. Comparisons of our proposed model with baseline (no spatial token mixer) and models with other token mixers in the original domain on ImageNet-1K classification. "Base." denotes the baseline model discarding all AFF token mixers. "Conv-Mixer (3×3)" refers to adopting token mixers implemented by 3×3 convolutions in the original space. "AFF w/o FFT" denotes performing adaptive filtering in the original space with the same networks by discarding the Fourier transforms where "w/o" and "AFF" are short for "without" and "AFF token mixer", respectively.

4.4. Ablation Study

Effectiveness and complexity of AFF token mixer. We analyze the effectiveness and complexity of our proposed AFF token mixer by comparing AFFNet with the *Base.* model in which all AFF token mixers are replaced with identity functions. As shown in Table 4, all AFF token mixers in AFFNet only requires 0.3M parameter increase ($< 6\%$) and 0.2G FLOPs increase ($\sim 15\%$) relative to the baseline and improves the Top-1 accuracy on ImageNet-1K by 1.9%. Comparing to the model with one 3×3 convolution layer as the token mixer, *i.e.*, *Base.+Conv-Mixer (3×3)*, AFFNet delivers 1.2% Top-1 accuracy improvements with about only half of parameters and FLOPs. This strongly demonstrates the effectiveness and efficiency of our proposed method for token mixing in lightweight networks.

Original vs. frequency domain. We compare applying the same adaptive filtering operations in original domain and in frequency domain. We discard the all Fourier and inverse Fourier transforms and remain others the same as AFFNet, *i.e.*, *Base.+AFF w/o FFT* in Table 4. Our AFFNet clearly outperforms it by 1.4% Top-1 accuracy with the same model complexity. Applying adaptive filtering in the original domain is even weaker than convolutional token mixer, which indicates that only adaptive ***frequency*** filters can serve as effeicient global token mixers.

Comparisons of different frequency operations. We compare the frequency operation design in AFF token mixer with those in previous works [37, 57, 12, 39, 21] in terms of

Method	Param (M)	FLOPs (G)	Top-1
Base.	5.2	1.3	77.9
Base. + AFNO [21]	5.5	1.5	78.8
Base. + GFN [57]	6.5	1.5	79.1
Base. + FFC [12]	7.7	1.7	79.1
Base. + DFF [39]	7.7	1.7	79.3
Base. + FNO [37]	141.0	1.5	79.7
Base. + AFF w. SUM	5.5	1.5	78.8
Base. + AFF (AFFNet)	5.5	1.5	79.8

Table 5. Comparisons of our design for AFF token mixer and other frequency-domain operations in previous works [37, 57, 12, 39, 21] in terms of their roles for token mixing on ImageNet-1K. “AFF w. SUM” denotes replacing the Hadamard product with a summation operation, “w.” is short for “with”.

their effects as token mixers. The results are in Table 5. As analyzed in Sec.3.3, FFC [12] and AFNO [21] actually perform filtering with deterministic functions, resulting in the lack of the adaptivity to semantics. The frequency-domain operations in them are both obviously inferior to ours. Moreover, our operation design is also clearly better than those in GFN [57] and FNO [37] since they perform filtering with network parameters implemented masks. These masks are fixed after training and lead to a large increase in parameters (*Base.+FNO* has more than 25× parameters as ours). Note that the implementation of FNO [37] with unshared fully connected layers for each frequency component results in a significant increase in the number of parameters. DFF [39] is designed for filtering out the frequency components adverse to domain generalization, thus requiring a spatial mask only. Our AFFNet is superior to *Base.+DFF* by 0.5% with fewer parameters and FLOPs, demonstrating the importance of channel-wise mixing. This will be further verified with a fairer comparison. These existing frequency-domain operations might be similar with our proposed one at the first glance, but they are designed for different purposes and perform worse than ours as token mixers. When replacing the Hadamard product in our method with a summation operation, the Top-1 accuracy drops by 1.0% since the equivalence introduced in Sec.3.2 no longer holds.

Mask Shape	Param (M)	FLOPs (G)	Top-1
1×H×W	5.5	1.5	79.3
C×H×W	5.5	1.5	79.8

Table 6. Experiments of verifying the importance of channel-specific token mixing on ImageNet-1K. Here, we adopt an average pooling operation along the channel dimension of the masks learned in AFFNet, yielding the mask with a shape of 1×H×W. This mask is shared across channels.

Spatial K-Size	N_{group}	Param. (M)	FLOPs (G)	Top-1
1×1	C	5.3	1.4	79.4
1×1	1	7.7	2.0	79.9
3×3	8	7.9	2.0	79.8
1×1	8	5.5	1.5	79.8

Table 7. Comparisons of different hyper-parameter choices in the sub-network for learning the filtering masks in AFFNet on ImageNet-1K. “Spatial K-Size” refers to the spatial size of convolution kernels. N_{group} denotes the number of groups for group linear or convolution layers. C is the total number of channels.

The importance of channel-specific token mixing. We have preliminarily demonstrated this by comparing the frequency-domain operations in DFF [39] and ours. Considering their masks are learned with different networks, here, we conduct a fairer comparison by adopting an average pooling along the channels of the learned masks in AFF token mixer. As shown in Table 6, frequency filtering with the masks of a shape of 1×H×W lead to 0.5% accuracy drop with the same model complexities, verifying the importance of channel-specific token mixing. This is because different semantic attributes of the learned latent representations distribute in different channels [1, 77], thus requiring channel-specific weights for token mixing. Besides, it delivers the same accuracy with *Base.+DFF* in Table 5. This indicates that the network architectures here are in fact not dominating factors for the effects of token mixing, allowing us to use a lightweight one.

Comparisons of hyper-parameter choices. As shown in Fig.2, we adopt two group linear layers (also known as 1×1 convolution layers) with ReLU to learn the masks for our proposed adaptive frequency filtering. As shown in Table 7, improving the kernel size cannot further improve the performance but leads to larger model complexities. Moreover, we keep the spatial kernel size as 1×1 while using different group numbers. When N_{group}=C, the Top-1 accuracy drops by 0.4%, in which depthwise convolutions are used so that the contexts among different channels are under-exploited for inferring the weights of token mixing. When N_{group}=1, it means that regular convolution/linear layers are used, which slightly improve the Top-1 accuracy by 0.1% at the expense of 40% parameters increase and 33.3% FLOPs increase. This setting explores more contexts but results in a worse accuracy and efficiency trade-off.

5. Conclusion

In this work, we reveal that *adaptive frequency filters can serve as efficient global token mixers* in a mathematically equivalent manner. Upon this, we propose Adaptive Frequency Filtering (AFF) token mixer to achieve low-cost adaptive token mixing in the global scope. Moreover, we take AFF token mixers as primary neural operators to build a lightweight backbone network, dubbed AFFNet. AFFNet achieves SOTA accuracy and efficiency trade-offs compared to other lightweight network designs across multiple vision tasks. Besides, we revisit the existing frequency-domain neural operations for figuring out what matters in their designs for token mixing. We hope this work could inspire more interplay between conventional signal processing and deep learning technologies.

References

[1] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *PNAS*, 2020. 2, 5, 8

[2] Gregory A Baxes. *Digital image processing: principles and applications*. John Wiley & Sons, Inc., 1994. 3

[3] Han Cai, Chuang Gan, and Song Han. Efficientvit: Enhanced linear attention for high-resolution low-computation visual recognition. *arXiv preprint arXiv:2205.14756*, 2022. 6

[4] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. In *ICLR*, 2022. 14

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 7

[6] Shoufa Chen, Enze Xie, Chongjian GE, Runjian Chen, Ding Liang, and Ping Luo. CycleMLP: A MLP-like architecture for dense prediction. In *ICLR*, 2022. 2, 3

[7] Shoufa Chen, Enze Xie, Chongjian Ge, Ding Liang, and Ping Luo. Cyclemlp: A mlp-like architecture for dense prediction. In *ICLR*, 2022. 3, 14

[8] 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. 6, 14

[9] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *CVPR*, 2020. 2, 4

[10] Yukang Chen, Jianhui Liu, Xiaojuan Qi, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Scaling up kernels in 3d cnns. *arXiv preprint arXiv:2206.10555*, 2022. 1

[11] Zekai Chen, Fangtian Zhong, Qi Luo, Xiao Zhang, and Yanwei Zheng. Edgevit: Efficient visual modeling for edge computing. In *WASA*, 2022. 6

[12] Lu Chi, Borui Jiang, and Yadong Mu. Fast fourier convolution. In *NeurIPS*, 2020. 3, 5, 7, 8

[13] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *NeurIPS*, 2021. 1, 2, 3

[14] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2

[15] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. In *NeurIPS*, 2021. 6

[16] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *CVPR*, 2022. 1

[17] 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. 1, 2, 3, 14

[18] 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. 7, 12

[19] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021. 12, 14

[20] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *ICCV*, 2021. 6

[21] John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers. *ICLR*, 2022. 3, 5, 7, 8

[22] 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. 12, 14

[23] Qipeng Guo, Xipeng Qiu, Xiangyang Xue, and Zheng Zhang. Low-rank and locality constrained self-attention for sequence modeling. *IEEE Trans Audio Speech Lang Process*, 2019. 1, 3

[24] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 7

[25] Ali Hassani and Humphrey Shi. Dilated neighborhood attention transformer. *arXiv preprint arXiv:2209.15001*, 2022. 1

[26] Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multiscale filters for semantic segmentation. In *ICCV*, 2019. 4

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[28] Qibin Hou, Zihang Jiang, Li Yuan, Ming-Ming Cheng, Shuicheng Yan, and Jiashi Feng. Vision permutator: A permutable mlp-like architecture for visual recognition. *TPAMI*, 2022. 1, 3

[29] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019. 2, 6

[30] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2, 6, 7

[31] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *NeurIPS*, 2016. 4

[32] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, 2020. 1

[33] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020. 1

[34] Jiashi Li, Xin Xia, Wei Li, Huixia Li, Xing Wang, Xuefeng Xiao, Rui Wang, Min Zheng, and Xin Pan. Next-vit: Next generation vision transformer for efficient deployment in realistic industrial scenarios. *arXiv preprint arXiv:2207.05501*, 2022. 1, 3

[35] Yanyu Li, Ju Hu, Yang Wen, Georgios Evangelidis, Kamyar Salahi, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Rethinking vision transformers for mobilenet size and speed. *arXiv preprint arXiv:2212.08059*, 2022. 6

[36] Yanyu Li, Geng Yuan, Yang Wen, Eric Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. In *NeurIPS*, 2022. 6

[37] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *ICLR*, 2021. 3, 5, 7, 8

[38] Dongze Lian, Zehao Yu, Xing Sun, and Shenghua Gao. As-mlp: An axial shifted mlp architecture for vision. *ICLR*, 2022. 1

[39] Shiqi Lin, Zhizheng Zhang, Zhipeng Huang, Yan Lu, Cuiling Lan, Peng Chu, Quanzeng You, Jiang Wang, Zicheng Liu, Amey Parulkar, et al. Deep frequency filtering for domain generalization. *arXiv preprint arXiv:2203.12198*, 2022. 3, 5, 7, 8

[40] 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. 6, 7, 12

[41] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *ICLR*, 2023. 1

[42] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 6

[43] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1, 2, 3

[44] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 2018. 2, 6

[45] Xuezhe Ma, Xiang Kong, Sinong Wang, Chunting Zhou, Jonathan May, Hao Ma, and Luke Zettlemoyer. Luna: Linear unified nested attention. In *NeurIPS*, 2021. 3

[46] 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. 6, 7, 12, 14

[47] Clare D McGillem and George R Cooper. *Continuous and discrete signal and system analysis*. Oxford University Press, USA, 1991. 1, 2, 3, 4, 5

[48] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. In *ICLR*, 2022. 1, 3, 6, 7, 12, 14

[49] Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers. *TMLR*, 2022. 3, 6, 7, 12, 14

[50] Sachin Mehta, Mohammad Rastegari, Linda Shapiro, and Hannaneh Hajishirzi. Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. In *CVPR*, 2019. 7

[51] Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999. 1, 3, 4

[52] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015. 2

[53] Zizheng Pan, Jianfei Cai, and Bohan Zhuang. Fast vision transformers with hilo attention. In *NeurIPS*, 2022. 1, 3

[54] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters–improve semantic segmentation by global convolutional network. In *CVPR*, 2017. 1

[55] Ioannis Pitas. *Digital image processing algorithms and applications*. John Wiley & Sons, 2000. 3

[56] Lawrence R Rabiner and Bernard Gold. Theory and application of digital signal processing. *Englewood Cliffs: Prentice-Hall*, 1975. 1, 3, 4

[57] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. In *NeurIPS*, 2021. 3, 5, 6, 7, 8

[58] 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, 6, 12

[59] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 2, 6, 7, 12, 14

[60] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 6

[61] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *ICML*, 2021. 6

[62] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, pages 10781–10790, 2020. 12, 14

[63] Chuanxin Tang, Yucheng Zhao, Guangting Wang, Chong Luo, Wenxuan Xie, and Wenjun Zeng. Sparse mlp for image recognition: Is self-attention really necessary? In *AAAI*, 2022. 3

[64] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. In *NeurIPS*, 2021. 1

[65] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. In *NeurIPS*, volume 34, 2021. 2, 3

[66] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021. 2, 3

[67] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 1, 2, 3, 6

[68] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *ECCV*, 2022. 6, 14

[69] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. An improved one millisecond mobile backbone. *arXiv preprint arXiv:2206.04040*, 2022. 6

[70] 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. 2, 14

[71] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *CVPR*, 2020. 3

[72] 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. 6, 14

[73] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 3

[74] Guoqiang Wei, Zhizheng Zhang, Cuiling Lan, Yan Lu, and Zhibo Chen. Active token mixer. In *AAAI*, 2023. 2

[75] Guoqiang Wei, Zhizheng Zhang, Cuiling Lan, Yan Lu, and Zhibo Chen. Active token mixer. *AAAI*, 2023. 2

[76] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *CVPR*, 2021. 2

[77] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *CVPR*, 2021. 2, 5, 8

[78] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *AAAI*, 2021. 1, 3

[79] Zhiqin John Xu. Understanding training and generalization in deep learning by fourier analysis. *arXiv preprint arXiv:1808.04295*, 2018. 3

[80] Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In *ICONIP*, 2019. 3

[81] 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. 12, 14

[82] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *NeurIPS*, 2019. 3

[83] 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. 1, 14

[84] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *CVPR*, pages 558–567, 2021. 6

[85] David Junhao Zhang, Kunchang Li, Yunpeng Chen, Yali Wang, Shashwat Chandra, Yu Qiao, Luoqi Liu, and Mike Zheng Shou. Morphmlp: A self-attention free, mlp-like backbone for image and video. *arXiv preprint arXiv:2111.12527*, 2021. 2, 3

[86] David Junhao Zhang, Kunchang Li, Yali Wang, Yunpeng Chen, Shashwat Chandra, Yu Qiao, Luoqi Liu, and Mike Zheng Shou. Morphmlp: An efficient mlp-like backbone for spatial-temporal representation learning. In *ECCV*, 2022. 3

[87] Lu Zhang, Shuigeng Zhou, Jihong Guan, and Ji Zhang. Accurate few-shot object detection with support-query mutual guidance and hybrid loss. In *CVPR*, 2021. 4

[88] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018. 2

[89] 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. 7, 12

Supplementary Material

6. 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 8. Specifically, following commonly used designs [48, 49], 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 [59] 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 $N_3 = 3$.

7. Detailed Introduction for Dataset

ImageNet [58] 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 [40] (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 [89] 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 [18] (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.

8. Detailed Experiment Settings

We provide detailed experiment settings for different tasks in Table 9, including the detailed configurations for model, data and training.

9. More Experiment Results

9.1. 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 10. Models with similar Top-1 accuracy are grouped together for clear comparison. The latency results are equivalently measured by CoreML[2] on an iPhone with a batch size of 1. The throughput results are measured with TorchScript[3] on an A100 GPU (batch size = 128). As shown in Table 10, 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.

Evaluation on more downstream task frameworks. For the experiments reported in our main paper (*e.g.*, Table 3), we adopt the most commonly used task frameworks, *i.e.*, SSD and Deeplabv3, in accordance with recent studies [59, 48, 49, 46] 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 downstream task frameworks [19, 62, 22, 81] 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 11. There results further demonstrate that our proposed AFFNet is compatible with diverse downstream task frameworks and generally applicable.

Comparisons of different frequency transforms. We investigate the effectiveness of adopting different frequency transforms in implementing our proposed AFF token mixer.

[2] https://github.com/apple/coremltools

[3] 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	↓ 2	1	32	32	32
	64^2	↓ 2	4	48	48	64
Down-sampling	32^2	↓ 2	1	64	96	128
AFF Block	32^2	-	2	64	96	128
Down-sampling	16^2	↓ 2	1	104	160	256
AFF Block	16^2	-	4	104	160	256
Down-sampling	8^2	↓ 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 8. 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 9. 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.

Specifically, we compare using FFT and using wavelet transform or Discrete Cosine Transform (DCT). The comparison results are in Table 12. 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

Model	Param. (M)	FLOPs (G)	Latency (ms)	Throughput (images/s)	Top-1
MViT-XXS [48]	1.3	0.4	4.8	6803	69.0
MViTv2-0.5 [49]	1.4	0.5	1.6	7021	70.2
EdgeNext-XXS [46]	1.3	0.3	1.7	7768	71.2
AFFNet-ET	1.4	0.4	1.4	8196	73.0
MViT-XS [48]	2.3	1.0	7.0	4966	74.8
MViTv2-0.75 [49]	2.9	1.0	2.4	5150	75.6
EdgeNext-XS [46]	2.3	0.5	2.6	5307	75.0
AFFNet-T	2.6	0.8	2.1	5412	77.0
CycleMLP-B1 [7]	15.2	2.1	15.2	3073	79.1
PoolFormer-S12 [83]	11.9	1.8	5.3	3922	77.2
MFormer-294 [8]	11.8	0.3	40.7	2790	77.9
MViT-S [48]	5.6	2.0	9.9	3703	78.4
MViTv2-1.0 [49]	4.9	1.8	3.4	3973	78.1
EdgeNext-S [46]	5.6	1.3	6.4	4023	79.4
AFFNet	5.5	1.5	3.1	4202	79.8

Table 10. 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.

Task	Detection(mAP)		Segmentation(mIOU)	
Framework From	yolox [19]	efficientdet [62]	van [22]	moat [81]
w. Origin Encoder	32.8	40.2	38.5	41.2
w. AFFNet Encoder	35.9	41.6	43.2	41.5

Table 11. Performance evaluation on more downstream task frameworks. Our proposed AFFNet are integrated into them as their encoders to compare with their original ones.

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 12. 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.

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.

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 13, *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.

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 13. 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.

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 14. 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.

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) [59] following prior works [72, 4, 68, 81]. Here, we compare two dominated designs of the channel mixer in Table 14 for a detailed empirical study. Feed-Forward Network (FFN) [70, 17] 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.

10. Visualization Results

We present the qualitative results of AFFNet on object detection and semantic segmentation in Fig. 3 and Fig. 4, 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 network design. And this demonstrates the effectiveness of our proposed AFF token mixer in preserving the spatial structure information during token mixing.

11. 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.

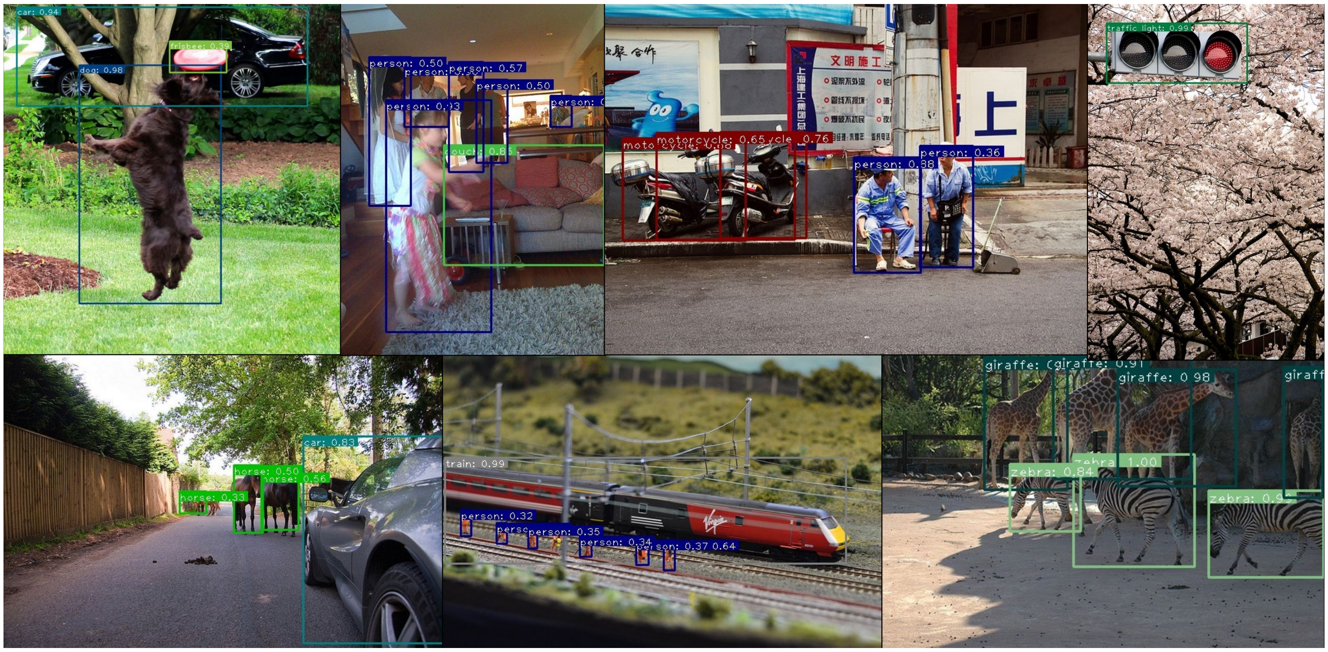
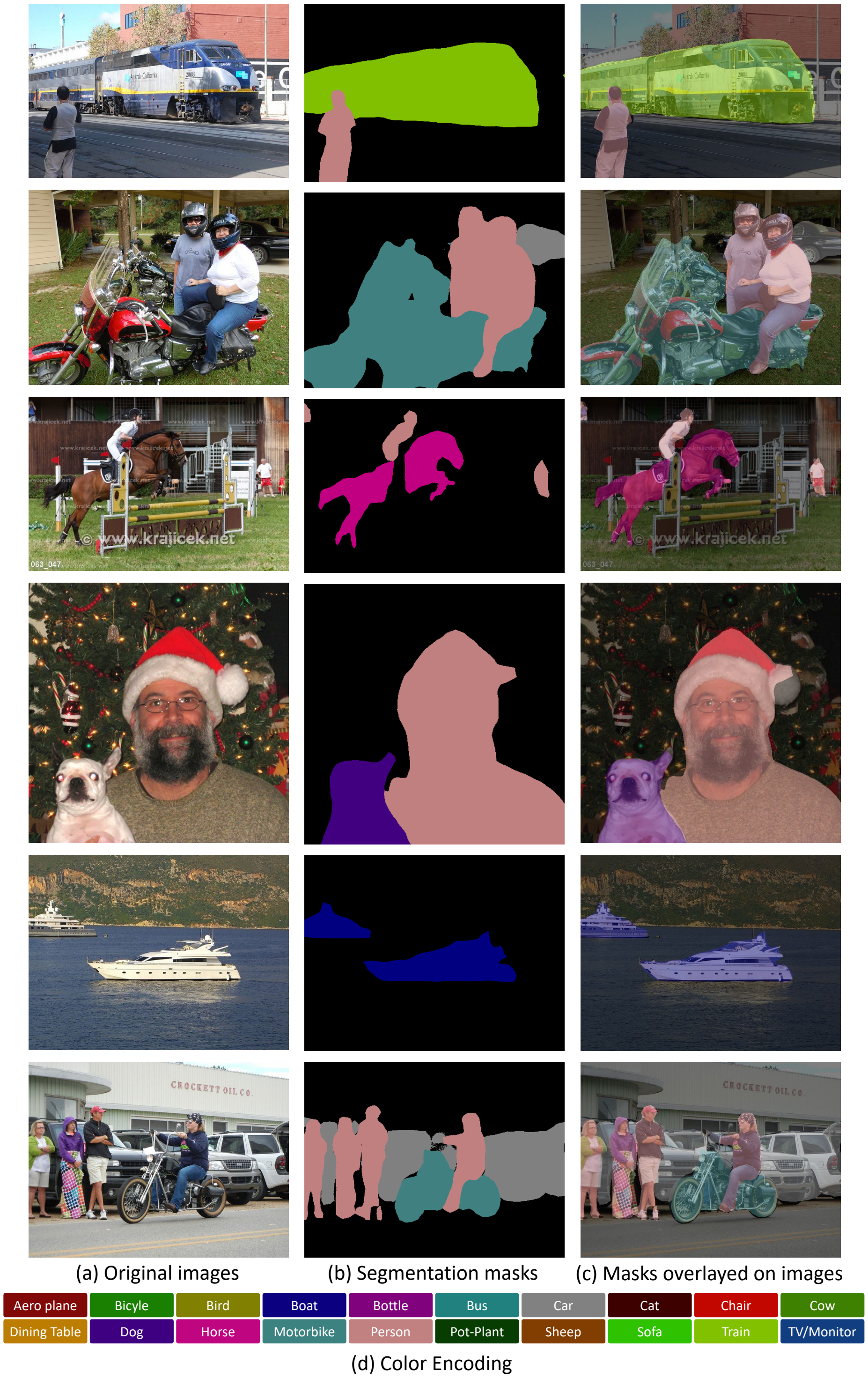


Figure 3. Qualitative results of the detection model with our AFFNet as the backbone on the validation set of COCO dataset.



(a) Original images (b) Segmentation masks (c) Masks overlayed on images

(d) Color Encoding

Figure 4. 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.