FFT-Based Dynamic Token Mixer for Vision

Yuki Tatsunami\textsuperscript{1,2}, Masato Taki\textsuperscript{1}
\textsuperscript{1}Rikkyo University
\textsuperscript{2}AnyTech Co., Ltd.
y.tatsunami@rikkyo.ac.jp, taki_m@rikkyo.ac.jp

Abstract
Multi-head-self-attention (MHSA)-equipped models have achieved notable performance in computer vision. Their computational complexity is proportional to quadratic numbers of pixels in input feature maps, resulting in slow processing, especially when dealing with high-resolution images. New types of token-mixer are proposed as an alternative to MHSA to circumvent this problem: an FFT-based token-mixer involves global operations similar to MHSA but with lower computational complexity. However, despite its attractive properties, the FFT-based token-mixer has not been carefully examined in terms of its compatibility with the rapidly evolving MetaFormer architecture. Here, we propose a novel token-mixer called Dynamic Filter and novel image recognition models, DFFormer and CDFFormer, to close the gaps above. The results of image classification and downstream tasks, analysis, and visualization show that our models are helpful. Notably, their throughput and memory efficiency when dealing with high-resolution image recognition is remarkable. Our results indicate that Dynamic Filter is one of the token-mixer options that should be seriously considered.

The code is available at https://github.com/okojoalg/dfformer

Introduction
A transformer architecture was propelled to the forefront of investigations in the computer vision field. The architecture locates the center in various visual recognition tasks, including not only image classification (Dosovitskiy et al. 2021; Touvron et al. 2021; Yuan et al. 2021; Zhou et al. 2021; Wang et al. 2021; Beyer et al. 2023) but also action recognition (Neimark et al. 2021; Bertasius, Wang, and Torresani 2021), even point cloud understanding (Guo et al. 2021; Zhao et al. 2021; Wei et al. 2022). Vision Transformer (ViT) (Dosovitskiy et al. 2021) and its variants triggered this explosion. ViT was inspired by Transformer in NLP and is equipped with a multi-head self-attention (MHSA) mechanism (Vaswani et al. 2017) as a critical ingredient. MHSA modules are low-pass filters (Park and Kim 2022). Hence they are suitable for recognizing information about an entire image. While MHSA modules have been a success, it has faced challenges, especially in aspects of quadratic computational complexity due to global attention design.

This problem is not agonizing in ImageNet classification but in dense tasks like semantic segmentation since we often deal with high-resolution input images. This problem can be tackled by using local attention design (Liu et al. 2021; Zhang et al. 2021; Chu et al. 2021; Chen, Panda, and Fan 2022), but the token-to-token interaction is limited, which means that one of the selling points of Transformers is mislaid.

GFNet (Rao et al. 2021) is one of the architectures induced by ViT and has two charming properties: (1) Global filter, the principal component in GFNet, multiplies features and complex parameters in frequency domains to increase/decrease a specific frequency. It is similar to a large kernel convolution (Ding et al. 2022b; Liu et al. 2023) but has the attractive aspect of reducing theoretical computational costs (Rao et al. 2021). In addition, GFNet has less difficulty training than a large kernel convolution, whereas large kernel convolutions employ a tricky training plan. Moreover, a computational complexity of GFNet is \( O(\log_2(HW)) + HW^2 C^2 ) \), which is superior to of self-attention \( O(HW^2 + (HW)^2 C) \) where \( H \) is height, \( W \) is width, and \( C \) is channel. In other words, the higher the resolution of the input image, the more complexity GFNet has relative dominance. (2) A global filter is expected to serve as a low-pass filter like MHSA because a global filter is a global operation with room to capture low-frequency information. We follow (Park and Kim 2022) and compare the relative log amplitudes of Fourier transformed between ViT and GFNet, which is a non-hierarchical architecture similar to ViT, and show in Figure 1. The result demonstrates that GFNet retains the same low-frequency information near 0 as ViT. Thus, we need to address the potential for the global filter alternative to MHSA due to its computational advantage when handling high-resolution input images and its similarity to MHSA. Global filters and MHSA are different in certain respects, even though they have low-pass filter characteristics like they are different in certain respects. For example, MHSA is a data-dependent operation generated from data and applied to data. In contrast, global filters multiply data and parameters and hence are less data-dependent operations. In recent works, MLP-based models, which have been less data-dependent, employ data-dependent operations (Wang et al. 2022; Rao et al. 2022), achieved accuracy improvements. It is inferred from...
the findings that introducing a data-dependent concept to global filters will improve accuracy.

Meanwhile, one should note that global filters have not been based on the most up-to-date and effective architectures relative to the well-researched MHSA-based and their relatives’ architectures. MetaFormer (Yu et al. 2022a,b) pays attention to the overall architecture of the Transformer. MetaFormer has a main block consisting of an arbitrary token-mixer and a channel mixer. It also makes sense to consider this framework for global filters because it is one of the most sophisticated architectures. It is necessary to close these differentials of architectural design if global filters can be a worthy alternative to MHSA modules.

We propose DFFormer, an FFT-based network to fill the above gap. Our architectures inherit MetaFormer and equip modules that can dynamically generate a global filter for each pair of feature channels in the image, considering their contents. Our experimental results show that the proposed methods achieve promising performance among vision models on the ImageNet-1K dataset, except for MHSA. DFFormer-B36 with 115M parameters has achieved a top-1 accuracy of 84.8%. CDFFormer, a hybrid architecture with convolutions and FTFs architecture, is even better. Moreover, when dealing with high-resolution images, the proposed method realizes higher throughput than the MHSA-based architecture and architecture using both CNN and MHSA. We also found that the dynamic filter tends to learn low frequency rather than MHSA on a hierarchical Metaformer.

Our main contributions are summarized as follows: First, we have introduced the dynamic filter, a dynamic version of the traditional global filter. Second, we found that injecting Metaformer-like architecture using the dynamic filter can close the gap between the accuracy of the global filter architecture and of SOTA models. Finally, through comprehensive analysis, we have demonstrated that the proposed architecture is less costly for downstream tasks of higher resolution, like dense prediction.

Figure 1: Relative log amplitudes of Fourier transformed feature maps. \(\Delta\) Log amplitude means relative logarithmic amplitude concerning the logarithmic amplitude at normalized frequency 0. The yellower the color, the deeper the layer.

Related Work

**Vision Transformers and Metaformers** Transformer (Vaswani et al. 2017), which was proposed in NLP, has become a dominant architecture in computer vision as well, owing to the success of ViT (Dosovitskiy et al. 2021) and DETR (Carion et al. 2020). Soon after, MLP-Mixer (Tolstikhin et al. 2021) demonstrates that MLP can also replace MHSA in Transformer. MetaFormer (Yu et al. 2022a), as an abstract class of Transformer and MLP-Mixer, was proposed as a macro-architecture. The authors tested the hypothesis using MetaFormer with pooling. Generally, the abstract module corresponding to MHSA and MLP is called a token-mixer. It emphasizes the importance of MetaFormer that new classes of MetaFormer, such as Sequencer (Tatsunami and Taki 2022) using RNNs, Vision GNN (Han et al. 2022) using graph neural networks, and RMT (Fan et al. 2023) using retention mechanism that is variant of the attention mechanism, have emerged. Moreover, a follow-up study (Yu et al. 2022b) has shown that even more highly accurate models can be developed by improving activation and hybridizing with multiple types of tokens. We retain the macro-architecture of (Yu et al. 2022b). The FFT-based module is a global operation like MHSA and can efficiently process high-resolution images without much loss of accuracy.

**FFT-based Networks** In recent years, neural networks using Fourier transforms have been proposed. FNet (Lee-Thorp et al. 2022), designed for NLP, contains modules using a discrete or fast Fourier Transform to extract only the real part. Accordingly, they have no parameters. Fast-FNet (Sevim et al. 2023) removes waste and streamlines FNet. GFNet (Rao et al. 2021) is an FFT-based network designed for vision with global filters. The global filter operates in frequency space by multiplying features with a Fourier filter, equivalent to a cyclic convolution. The filter is a parameter, i.e., the same filter is used for all samples. In contrast, our dynamic filter can dynamically generate the Fourier filter using MLP. (Guibas et al. 2022) is one of the most related works and proposed AFNO. Instead of the element-wise product, the MLP operation of complex weights is used to represent an adaptive and non-separable convolution. AFNO is not a separable module. As a result, the computational cost is higher than the separable module. On the contrary, our proposed method can realize a dynamic separable Fourier filter, and our models do not differ much from the global filter in terms of FLOPs. We will mention these throughputs in subsection . Concurrent work (Vo et al. 2023) also attempts to generate dynamic filters, but the structure is such that only the filter coefficients are changed. Although the amplitude and frequency of the filter are dynamic, the major property of the filter cannot be changed (e.g., a high-pass filter cannot be changed to a low-pass filter). In contrast, a dynamic filter can represent high-pass and low-pass filters by first-order coupling.

**Dynamic Weights** There have been some suggestive studies on the generation of dynamic weights. (Jia et al. 2016) realizes the filter parameters to change dynamically, where the filter weights of the convolution were model parameters. A filter generation network generates the filters. (Ha,
Dai, and Le 2016) is a contemporaneous study. It proposed HyperNetworks, neural networks to create weights of another neural network. They can generate weights of CNN and RNN. (Yang et al. 2019; Chen et al. 2020) are at the same time worked on, both of which make the filters of convolution dynamic. These works influence our dynamic filter: whereas (Yang et al. 2019; Chen et al. 2020) predict the real coefficients of linear combination for a real filter basis of standard convolution, our work predicts the real coefficients of linear combination for a complex parameter filter basis.

This method can be considered an equivalent operation in forward propagation from the fast Fourier transform’s linearity; thus, our work is similar to these works. In backpropagation, however, whether the proposed dynamic filter can be learned is nontrivial. Our architecture does not also need the training difficulties cited in these studies, such as restrictions on batch size or adjustment of softmax temperatures. Outside of convolution, Synthesizer (Tay et al. 2021), DynaMixer (Wang et al. 2022), and AMixer (Rao et al. 2022) have MLP-Mixer-like token-mixer, but with dynamic generation of weights. In particular, AMixer includes linear combinations related to our work.

**Method**

**Preliminary: Global Filter**

We will look back at a discrete Fourier transform before introducing a global filter (Rao et al. 2021). We discuss a 2D discrete Fourier transform (2D-DFT). For given the 2D signal \( x(h, w) \), we define the 2D-DFT \( \tilde{x}(h', w') \) as following:

\[
\tilde{x}(h', w') = \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} x(h, w)e^{-2\pi j \left( \frac{h}{H}h' + \frac{w}{W}w' \right) / \sqrt{HW}}
\]

(1)

where \( H, W \in \mathbb{N}, h, h' \in \{ z \in \mathbb{Z} | 0 < z < H \} \), and \( w, w' \in \{ z \in \mathbb{Z} | 0 < z < W \} \). Its inverse transformation exists and is well-known as a 2D inverse discrete Fourier transform (2D-IDFT). Generally, \( \tilde{x}(h', w') \) is complex and periodic to \( h' \) and \( w' \). We assume that \( x(h', w') \) is a real number, then a complex matrix \( \tilde{X}(h', w') \) associated with \( \tilde{x}(h', w') \) is Hermitian. A space to which \( \tilde{x}(h', w') \) belongs is known as a frequency domain and is available for analyzing frequencies. In addition, the frequency domain has a significant property: Multiplication in the frequency domain is equivalent to a cyclic convolution in the original domain, called convolution theorem (Proakis 2007). The 2D-DFT is impressive but has a complexity of \( O(H^2W^2) \). Therefore, a 2D-FFT is proposed and is often used in signal processing. It is improved with complexity \( O(HW \log_2 (HW)) \).

Second, we define the global filter for the feature \( X \in \mathbb{R}^{C \times H \times W} \). The global filter \( G \) formulating the following:

\[
G(x) = F^{-1}(K \odot F(x))
\]

(2)

where \( \odot \) is the element-wise product, \( K \in \mathbb{C}^{C \times H \times \lceil \frac{W}{2} \rceil} \) is a learnable filter, and \( F \) is the 2D-FFT of which redundant components are reduced (i.e., \( x \text{ rfft2} \)) since \( F(x) \) is Hermitian. Note that the operation is equivalent to cyclic convolution of the filter \( F^{-1}(x) \) based on the convolution theorem.
We prepare four sizes of models for codes for details in the appendix.

A schematic diagram is shown on the right side of Figure 3. We construct DFFormer and CDFFormer complying with MetaFormer blocks, such as DFFormer and CDFFormer, which mainly consists of DFFormer blocks, and CDFFormer, which is a hybrid model of DFFormer blocks and ConvFormer blocks. Each model is equipped with an MLP classifier with Squared ReLU (So et al. 2021) as the activation. A detailed model structure is shown in Table 1.

### Experiments

We conduct experiments on ImageNet-1K benchmark (Krizhevsky, Sutskever, and Hinton 2012) and perform further experiments to confirm downstream tasks such as ADE20K (Zhou et al. 2017). Finally, we will conduct ablation studies about design elements.

### Image Classification

DFFormers and CDFFormers have experimented on ImageNet-1K (Krizhevsky, Sutskever, and Hinton 2012), one of the most renowned data sets in computer vision for image classification. It has 1000 classes and contains 1,281,167 training images and 50,000 validation images. Our training strategy is mainly according to (Touvron et al. 2021) and is detailed as follows. For data augmentation methods, we apply MixUp (Zhang et al. 2018), CutMix (Yun et al. 2019), random erasing (Zhong et al. 2020), and RandAugment (Cubuk et al. 2020). Stochastic depth (Huang et al. 2016) and label smoothing (Szegedy et al. 2016) are used to regularize. We employ AdamW (Loshchilov and Hutter 2019) optimizer for 300 epochs with a batch size of 1024. The base learning rate of $\frac{\text{batch size}}{412} \times 5 \times 10^{-4}$, 20 epochs of linear warm-up, cosine decay for learning rate, and weight decay of 0.05 are used. The implementation is based on PyTorch (Paszke et al. 2019) and timm (Wightman 2019). The details of the hyperparameters are presented in the appendix. We compare the proposed models with various family models, including CNN-based like ConvNeXt (Liu et al. 2022) and ConvFormer (Yu et al. 2022b), attention-based like DeiT (Touvron et al. 2021), CSwin (Dong et al. 2022), MViTv2 (Li et al. 2022), Di-NAT (Hassani and Shi 2022), DuViT (Ding et al. 2022a), GCViT (Hatamizadeh et al. 2023), and MaxViT (Tu et al. 2022), Retention-based like RMT (Fan et al. 2023), MLP-based like DynaMixer (Wang et al. 2022) and AMixer (Rao et al. 2022), FFT-based like GFNet (Rao et al. 2021), and hybrid models like CAFomer (Yu et al. 2022b). Table 2 shows the results. We can see that DFFormers and CDFFormers perform top-1 accuracy among models except for models using attention or retention, with comparable parameters. DFFormers are particularly state-of-the-art among traditional FFT-based models. They perform more than 0.5% better than other FFT-based models and have been ahead of MLP-based models to which FFT-based tend to compare. DFFormers also have better cost performance than DFFormers due to using conjunction with convolution. The largest model, CDFFormer-B36, outperforms DFFormer-B36. The above performance comparison of DFFormer and CDFFormer indicates that dynamic filters are promising modules for image recognition.
<table>
<thead>
<tr>
<th>Model</th>
<th>Ty.</th>
<th>Prm. (M)</th>
<th>FLOP (G)</th>
<th>Thrp. (img/s)</th>
<th>Top-1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvNeXt-T</td>
<td>C</td>
<td>29</td>
<td>4.5</td>
<td>1471</td>
<td>82.1</td>
</tr>
<tr>
<td>ConvFormer-S18</td>
<td>C</td>
<td>27</td>
<td>3.9</td>
<td>756</td>
<td>83.0</td>
</tr>
<tr>
<td>MViTv2-T</td>
<td>A</td>
<td>23</td>
<td>4.3</td>
<td>340</td>
<td>82.7</td>
</tr>
<tr>
<td>DiNaT-T</td>
<td>A</td>
<td>24</td>
<td>4.7</td>
<td>624</td>
<td>82.3</td>
</tr>
<tr>
<td>DaViT-Tiny</td>
<td>A</td>
<td>28</td>
<td>4.5</td>
<td>1121</td>
<td>82.8</td>
</tr>
<tr>
<td>GCViT-T</td>
<td>A</td>
<td>28</td>
<td>4.7</td>
<td>566</td>
<td>83.5</td>
</tr>
<tr>
<td>MaxViT-T</td>
<td>R</td>
<td>41</td>
<td>5.6</td>
<td>527</td>
<td>83.6</td>
</tr>
<tr>
<td>AMixer-T</td>
<td>M</td>
<td>26</td>
<td>4.5</td>
<td>724</td>
<td>82.0</td>
</tr>
<tr>
<td>GFNet-H-S</td>
<td>F</td>
<td>32</td>
<td>4.6</td>
<td>952</td>
<td>81.5</td>
</tr>
<tr>
<td>DFFormer-S18</td>
<td>F</td>
<td>30</td>
<td>3.8</td>
<td>535</td>
<td>83.2</td>
</tr>
<tr>
<td>CAFormer-S18</td>
<td>CA</td>
<td>26</td>
<td>4.1</td>
<td>741</td>
<td>83.6</td>
</tr>
<tr>
<td>CDFFormer-S18</td>
<td>CF</td>
<td>30</td>
<td>3.9</td>
<td>567</td>
<td>83.1</td>
</tr>
</tbody>
</table>

Table 2: Performance comparison of models pre-trained on ImageNet-1K at the resolution of 224^2. Throughput has been benchmarked on a V100 with 16GB memory at a batch size of 16. C is Convolution, A is Attention, R is Retention, M is MLP, F is FFT, CA is a hybrid of Convolution and Attention, and CF is a hybrid of Convolution and FFT.

Semantic Segmentation on ADE20K
We train and test our models on ADE20K (Zhou et al. 2017) dataset for a semantic segmentation task to evaluate the performance in dense prediction tasks. We employ Semantic FPN (Kirillov et al. 2019) in mmseg (Contributors 2020) as a base framework. See the appendix for setup details. Table 3 shows that DFFormer-based and CDFFormer-based models equipped with Semantic FPN for semantic segmentation are effective for the semantic segmentation task. They are superior to those based on other models, including PoolFormer-based models. From these results, it can be seen that DFFormer-S36 has 5.5 points higher mIoU than PoolFormer-S36. In addition, CDFFormer-M36 achieves 48.6 mIoU. As a result, we also verify the effectiveness of DFFormer and CDFFormer for semantic segmentation.

Ablation Studies
We experiment with ablation studies on the ImageNet-1K dataset. Let us discuss ablation from several perspectives.

Table 3: Performance comparison of models using Semantic FPN trained on ADE20K. The settings follow (Yu et al. 2022a) and the results of compared models are cited from (Yu et al. 2022a).

<table>
<thead>
<tr>
<th>Backbone</th>
<th>Prm. (M)</th>
<th>mIoU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50</td>
<td>28.5</td>
<td>36.7</td>
</tr>
<tr>
<td>PVT-Small</td>
<td>28.2</td>
<td>39.8</td>
</tr>
<tr>
<td>PoolFormer-S24</td>
<td>23.2</td>
<td>40.3</td>
</tr>
<tr>
<td>DFFormer-S18</td>
<td>31.7</td>
<td>45.1</td>
</tr>
<tr>
<td>CDFFormer-S18</td>
<td>31.4</td>
<td>44.9</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>47.5</td>
<td>38.8</td>
</tr>
<tr>
<td>ResNeXt-101-32x4d</td>
<td>47.1</td>
<td>39.7</td>
</tr>
<tr>
<td>PVT-Medium</td>
<td>48.0</td>
<td>41.6</td>
</tr>
<tr>
<td>PoolFormer-S36</td>
<td>34.6</td>
<td>42.0</td>
</tr>
<tr>
<td>DFFormer-S36</td>
<td>47.2</td>
<td>47.5</td>
</tr>
<tr>
<td>CDFFormer-S36</td>
<td>46.5</td>
<td>46.7</td>
</tr>
<tr>
<td>PVT-Large</td>
<td>65.1</td>
<td>42.1</td>
</tr>
<tr>
<td>PoolFormer-M36</td>
<td>59.8</td>
<td>42.4</td>
</tr>
<tr>
<td>DFFormer-M36</td>
<td>66.4</td>
<td>47.6</td>
</tr>
<tr>
<td>CDFFormer-M36</td>
<td>65.2</td>
<td>48.6</td>
</tr>
</tbody>
</table>

Figure 5: Throughput vs. resolution. Throughput has been benchmarked on a V100 with 16GB memory at a batch size of four.

The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)
### Analysis

**Advantages at Higher Resolutions**

We observe how the throughput and peak memory at inference change on proposed and comparative models when resolution varies. For throughput, the proposed models have been inferior to CAFormer, an architecture that utilized MHSA, at a resolution of $224^2$ since Table 2 can comprehend the fact. The result conflicts with the computational complexity presented in the Introduction. Our interpretation of the issue is that the actual throughput would depend on the implementation of FFT (although we use cuFFT via PyTorch), hardware design, etc. Increasing the resolution, the theoretically computational complexity of MHSA comes into effect: Figure 5 shows the throughput for different resolutions. DFFormer and CDFFormer maintain throughput close to that of ConvFormer, while only CAFormer shows a significant decrease in throughput. The same is true for peak memory in Figure 6: Whereas the CAFormer peak memory increases, the DFFormer and CDFFormer peak memory is comparable to that of ConvFormer. Therefore, DFFormer and CDFFormer are beneficial in speed- and memory-constrained environments for tasks such as semantic segmentation, where high resolution is required.

**Representational Similarities**

Figure 7 shows the similarities between GFFormer-S18 and DFFormer-S18, ConvFormer-S18 and DFFormer-S18, and CAFormer-S18 and CDFFormer-S18 in the validation set of ImageNet-1K. In our analysis, we metric the similarity using mini-batch linear CKA (Nguyen, Raghu, and Kornblith 2021). torch.cka (Subramanian 2021) toolbox was used to implement the mini-batch linear CKA. The layers to be

---

**Table 4: Ablation for DFFormer-S18 on ImageNet-1K. Remind $N$ is the dimension of the dynamic filter basis, and $\rho$ is a ratio of MLP intermediate dimension to input dimension.**

<table>
<thead>
<tr>
<th>Ablation</th>
<th>Variant</th>
<th>Parameters (M)</th>
<th>FLOPs (G)</th>
<th>Throughput (img/sec)</th>
<th>Top-1 Acc. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>DFFormer-S18</td>
<td>$N = 4 \rightarrow 2$</td>
<td>30</td>
<td>3.8</td>
<td>535</td>
</tr>
<tr>
<td>Filter</td>
<td>DFFormer-S18</td>
<td>$\rho = 0.25 \rightarrow 0.125$</td>
<td>29</td>
<td>3.8</td>
<td>534</td>
</tr>
<tr>
<td></td>
<td>DFFormer-S18</td>
<td>$DF \rightarrow AFNO$</td>
<td>28</td>
<td>3.8</td>
<td>532</td>
</tr>
<tr>
<td></td>
<td>GFFormer-S18</td>
<td>$\rho = 0.25 \rightarrow 0.125$</td>
<td>30</td>
<td>3.8</td>
<td>575</td>
</tr>
<tr>
<td></td>
<td>DFFormer-S18</td>
<td>$DF \rightarrow AFNO$</td>
<td>30</td>
<td>3.8</td>
<td>575</td>
</tr>
</tbody>
</table>

DFS18 outperforms the replaced version of the model by more than 0.5%, demonstrating that StarReLU is also a valuable activation for dynamic filters. In this manner, StarReLU has a significant impact on the final performance. Therefore, to comprehend how much a dynamic filter would improve performance without StarReLU, we also experimented with a model version in which StartReLU and a dynamic filter are replaced by ReLU and AFNO, respectively. Comparing this result with the StarReLU $\rightarrow$ ReLU results shows us that the dynamic filter has enough impact even when the influence of StarReLU is removed.

---

**Figure 6: Peak memory vs. resolution. Peak memory has been benchmarked on a V100 with 16GB memory at a batch size of four.**

**Filter** We studied how changing filters would work. First, we change to the hyperparameters of dynamic filters. Specifically, we train and test models in which the dimension of dynamic filter base $N$ and the intermediate dimension $\rho C$ are modified in half of the cases, respectively. From Table 4, however, we found that the throughput and number of parameters remained almost the same, and the accuracy dropped by 0.2% and 0.1%, respectively. Next, we train GFFormer-S18, which is replaced by each dynamic filter of DFFormer-S18 with a global filter (i.e., static filter) to confirm the effectiveness of dynamic filters. Table 4 demonstrates DFFormer-S18 is superior to GFFormer-S18 0.3%. We also have studied a case where AFNOs replaced dynamic filters but found that the AFNO-based model degraded in accuracy more than the dynamic filter and that dynamic filters have better throughput than AFNOs.

**Activation** DFFormer and CDFFormer use StarReLU, but this is a relatively new activation; many MetaFormers employ Gaussian Error Linear Unit (GELU) (Hendrycks and Gimpel 2023), and earlier generations of convolutional neural networks (CNNs), such as ResNet (He et al. 2016), often used Rectified Linear Unit (ReLU) (Nair and Hinton 2010). Accordingly, we also experiment with a version of the model in which GELU and ReLU replace StarReLU in DFFormer-S18. As a result, Table 4 shows that DFFormer-S18 outperforms the replaced version of the model by more than 0.5%, demonstrating that StarReLU is also a valuable activation for dynamic filters. In this manner, StarReLU has a significant impact on the final performance. Therefore, to comprehend how much a dynamic filter would improve performance without StarReLU, we also experimented with a model version in which StarReLU and a dynamic filter are replaced by ReLUs and AFNOs, respectively. Comparing this result with the StarReLU $\rightarrow$ ReLU results shows us that the dynamic filter has enough impact even when the influence of StarReLU is removed.

---

**Figure 7: Representational Similarities between GFFormer-S18 and DFFormer-S18, ConvFormer-S18 and DFFormer-S18, and CAFormer-S18 and CDFFormer-S18 in the validation set of ImageNet-1K. In our analysis, we metric the similarity using mini-batch linear CKA (Nguyen, Raghu, and Kornblith 2021). torch.cka (Subramanian 2021) toolbox was used to implement the mini-batch linear CKA. The layers to be
Figure 7: The feature map similarities by CKA

(a) GF vs. DF

(b) CA vs. CDF

(c) Conv vs. DF

Figure 8: Relative log amplitudes of Fourier transformed feature maps on stage 3. A setup similar to that in Table 1.

Figure 9: Filters in the frequency domain on DFFormer-S18

analyzed included four downsampled and 18 token-mixers residual blocks and 18 channel MLP residual blocks, and the indices in the figure are ordered from the shallowest layer. Figure 7a shows that GFFormer-S18 and DFFormer-S18 are very similar up to stage 3, although they are slightly different at stage 4. We also found the same thing about the similarity between ConvFormer-S18 and DFFormer-S18 from Figure 7c. On the contrary, Figure 7b shows that CAFormer-S18 and CDFFormer-S18 are similar up to stage 2 with convolution but become almost entirely different from stage 3 onward. Even though FFT-based token-mixers, like MHSA, are token-mixers in the global domain and can capture low-frequency features, they necessarily do not learn similar representations. The fact would hint that MHSA and the FFT-based token-mixer have differences other than spatial mixing. We perform a Fourier analysis on stage 3 to know the difference in stage 3. Figure 8 shows the difference between MHSA and FFT-based token-mixer results. Surprisingly, CAFormer-S18 acts as a high-pass filter, whereas the result for CDFFormer-S18 indicates more attenuated at other frequencies than around 0. In other words, MHSA incorporated in a hierarchical architecture can learn high-pass and low-pass filters while the dynamic filter attenuates the high frequencies. Only applying to stage 3 makes the analysis more interpretable than overall because this analysis drastically changes frequency by downsampling.

Analysis of Dynamic Filter Basis

The basis of complex parameters in the frequency domain can represent dynamic filters used in DFFormer and CDFFormer. Here, we visualize them for DFFormer-S18, which has four in each dynamic filter module. In our visualization, the center pixel means zero frequency and the yellower pixel means higher amplitudes. Figure 9 demonstrates that the redundancy of filters in the same layer is reduced compared to Figure 2 about GFNet-Ti. In addition, we can see high-pass, low-pass, and band-pass filters in many layers. Details of the visualization method and more visualization will be given in the appendix.

Conclusion

This paper studied the similarities and differences between the global filter and MHSA in vision models. According to the analysis, we proposed a novel dynamic filter responsible for dynamically generating global filters. Based on this module, we have developed new MetaFormer classes, DFFormer and CDFFormer, which achieve promising performance in MHSA-free MetaFormers. We have proven through a variety of experiments that the proposed models perform impressively not only in image classification but also in downstream tasks. In addition, our models can process high-resolution images faster and with less memory than MetaFormers using MHSA. We also found that the representation and properties of our models are not similar unexpectedly to that of models using MHSA.

Acknowledgments

This work was supported by the Rikkyo University Special Fund for Research.
References


Bertasius, G.; Wang, H.; and Torresani, L. 2021. Is space-time attention all you need for video understanding? In ICML.


Ding, X.; Zhang, X.; Zhou, Y.; Han, J.; Ding, G.; and Sun, J. 2022b. Scaling Up Your Kernels to 31x31: Revisiting Large Kernel Design in CNNs. In CVPR.


Ha, D.; Dai, A.; and Le, Q. V. 2016. Hypernetworks. In ICLR.

Han, K.; Wang, Y.; Guo, J.; Tang, Y.; and Wu, E. 2022. Vision gnn: An image is worth graph of nodes. In NeurIPS.


Lee-Thorp, J.; Ainslie, J.; Eckstein, I.; and Ontanon, S. 2022. Fnet: Mixing tokens with fourier transformers. In NAACL.


Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In ICLR.

Nair, V.; and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In ICML.


Nguyen, T.; Raghu, M.; and Kornblith, S. 2021. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In ICLR.


