GFNet: Global Filter Networks for Visual Recognition

Yongming Rao, *Student Member, IEEE*, Wenliang Zhao, *Student Member, IEEE*, Zheng Zhu, *Member, IEEE*, Jie Zhou, *Senior Member, IEEE*, and Jiwen Lu, *Senior Member, IEEE*

***Abstract*—Recent advances in self-attention and pure multi-layer perceptrons (MLP) models for vision have shown great potential in achieving promising performance with fewer inductive biases. These models are generally based on learning interaction among spatial locations from raw data. The complexity of self-attention and MLP grows quadratically as the image size increases, which makes these models hard to scale up when high-resolution features are required. In this paper, we present the Global Filter Network (GFNet), a conceptually simple yet computationally efficient architecture, that learns long-term spatial dependencies in the frequency domain with log-linear complexity. Our architecture replaces the self-attention layer in vision Transformers with three key operations: a 2D discrete Fourier transform, an element-wise multiplication between frequency-domain features and learnable global filters, and a 2D inverse Fourier transform. Based on this basic design, we develop a series of isotropic models with a Transformer-style simple architecture and CNN-style hierarchical models with better performance. Isotropic GFNet models exhibit favorable accuracy/complexity trade-offs compared to recent vision Transformers and pure MLP models. Hierarchical GFNet models can inherit successful designs in CNNs and be easily scaled up with larger model sizes and more training data, showing strong performance on both image classification (e.g., 85.0% top-1 accuracy on ImageNet-1 k without any extra data or supervision, and 87.4% accuracy with ImageNet-21 k pre-training) and dense prediction tasks (e.g., 54.3 mIoU on ADE20 k val). Our results demonstrate that GFNet can be a very competitive alternative to Transformer-based models and CNNs in terms of efficiency, generalization ability and robustness. Code is available at https://github.com/raoyongming/GFNet.**

***Index Terms*—Image classification, object detection, representation learning, semantic segmentation.**

I. Introduction

THE Transformer architecture, originally designed for the natural language processing (NLP) tasks [67], has shown promising performance on various vision problems recently [7], [20], [46], [65], [79]. Different from convolutional neural networks (CNNs), vision Transformer models use self-attention layers to capture long-term dependencies, which are able to learn more diverse interactions between spatial locations. The pure multi-layer perceptrons (MLP) models [63], [64] further simplify the vision Transformers by replacing the self-attention layers with MLPs that are applied across spatial locations. Since fewer inductive biases are introduced, these two kinds of models have the potential to learn more generic and flexible interactions among spatial locations from raw data.

One primary challenge of applying self-attention and pure MLP models to vision tasks is the considerable computational complexity that grows quadratically as the number of tokens increases. Therefore, typical vision Transformer style models usually consider a relatively small resolution for the intermediate features (e.g., 14×14 tokens are extracted from the input images in both ViT [20] and MLP-Mixer [63]). This design may limit the applications of downstream dense prediction tasks like detection and segmentation. A possible solution is to replace the global self-attention with several local self-attention like Swin Transformer [46]. Despite the effectiveness in practice, local self-attention brings quite a few hand-made choices (e.g., window size, padding strategy, *etc*.) and limits the receptive field of each layer.

In this paper, we present a new conceptually simple yet computationally efficient architecture called Global Filter Network (*GFNet*), which follows the trend of removing inductive biases from vision models while enjoying the log-linear complexity in computation. The basic idea behind our architecture is to learn the interactions among spatial locations in the frequency domain. Different from the self-attention mechanism in vision Transformers and the fully connected layers in MLP models, the interactions among tokens are modeled as a set of learnable *global filters* that are applied to the spectrum of the input features. Since the global filters are able to cover all the frequencies, our model can capture both long-term and short-term interactions. The filters are directly learned from the raw data without introducing human priors. Our architecture is largely based on the vision Transformers only with some minimal modifications. We replace the self-attention sub-layer in vision Transformers with three key operations: a 2D discrete Fourier transform to convert the input spatial features to the frequency domain, an element-wise multiplication between frequency-domain features and the global filters, and a 2D inverse Fourier transform to map the features back to the spatial domain. Since

Manuscript received 23 April 2022; revised 24 December 2022; accepted 26 March 2023. Date of publication 3 April 2023; date of current version 4 August 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62125603. Recommended for acceptance by H. Pirsiavash. *(Corresponding author: Jiwen Lu.)*

The authors are with the Beijing National Research Center for Information Science and Technology (BNRist), Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: raoyongming95@gmail.com; zhaowl20@mails.tsinghua.edu.cn; zhengzhu@ieee.org; jzhou@tsinghua.edu.cn; lujiwen@tsinghua.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2023.3263824, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2023.3263824

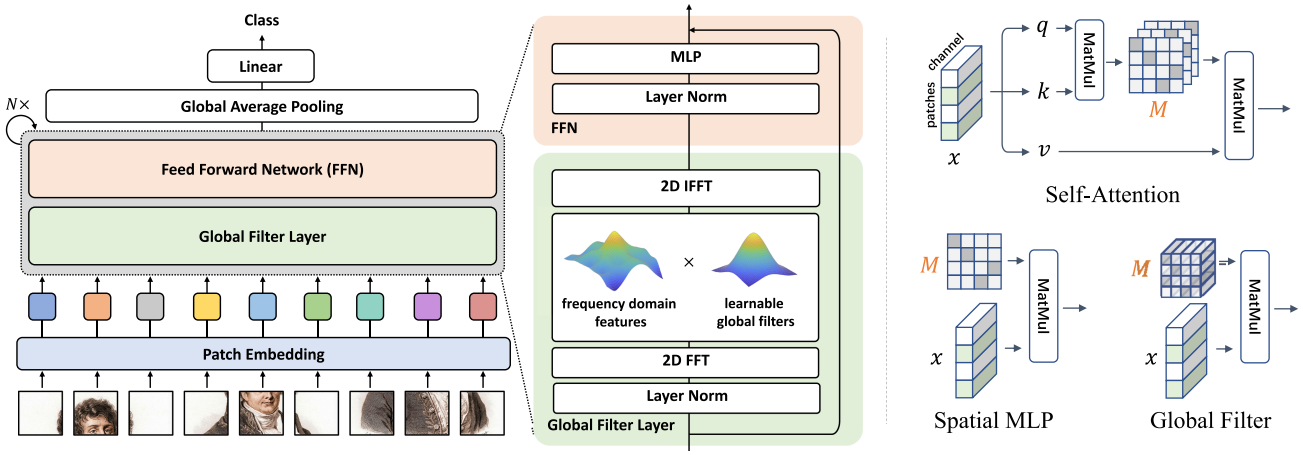


Fig. 1. *The overall architecture of the Global Filter Network*. Our architecture is based on Vision Transformer (ViT) models with some minimal modifications. We replace the self-attention sub-layer with the proposed *global filter layer*, which consists of three key operations: a 2D discrete Fourier transform to convert the input spatial features to the frequency domain, an element-wise multiplication between frequency-domain features and the global filters, and a 2D inverse Fourier transform to map the features back to the spatial domain. The efficient fast Fourier transform (FFT) enables us to learn arbitrary interactions among spatial locations with log-linear complexity. The connections and differences among the proposed global filter layer, self-attention and spatial MLP are also summarized in the right part (see Section III-C for more details).

TABLE I
COMPARISONS OF THE PROPOSED *GLOBAL FILTER* WITH PREVALENT OPERATIONS IN DEEP VISION MODELS. H, W AND D ARE THE HEIGHT, WIDTH AND THE NUMBER OF CHANNELS OF THE FEATURE MAPS. k IS THE KERNEL SIZE OF THE DEPTH-WISE CONVOLUTION OPERATION [10]. THE PROPOSED GLOBAL FILTER IS MUCH MORE EFFICIENT THAN SELF-ATTENTION [67] AND SPATIAL MLP [63] WHILE ENABLING GLOBAL INTERACTIONS AMONG LOCAL FEATURES

	Complexity (FLOPs)	# Parameters	Global Interactions
Depth-wise Convolution [10]	$\mathcal{O}(k^2HWD)$	k^2D	✗
Self-Attention [67]	$\mathcal{O}(HWD^2 + H^2W^2D)$	$4D^2$	✓
Spatial MLP [63]	$\mathcal{O}(H^2W^2D)$	H^2W^2	✓
Global Filter	$\mathcal{O}\left(HWD\lceil\log_2(HW)\rceil + HWD\right)$	HWD	✓

the Fourier transform is used to mix the information of different tokens, the global filter is much more efficient compared to the self-attention and MLP thanks to the $\mathcal{O}(L \log L)$ complexity of the fast Fourier transform algorithm (FFT) [12]. Benefiting from this, the proposed global filter layer is less sensitive to the token length L and thus is compatible with larger feature maps and CNN-style hierarchical architectures without modifications. The overall architecture of GFNet is illustrated in Fig. 1. We also compare our global filter with prevalent operations in deep vision models in Table I.

Apart from the Transformer-style *isotropic* architecture, our new global filter layer is also compatible with the successful *hierarchical* design in CNNs [25], [37] and modified vision Transformers [46], [69]. By combining the Transformer-like basic block (i.e., a spatial mixing layer followed by a feed-forward network) and the hierarchical macro architecture with multiple feature resolutions, we show that our GFNet models can achieve better performance with lower computation complexity. While models based solely on global filters and MLP layers can achieve better efficiency than state-of-the-of Transformers like Swin [46], we further improve the hierarchical models with two new micro designs: 1) we develop a two-path layer that separately and explicitly captures global and local spatial relations with our global filter operation and the conventional local convolution respectively; 2) we introduce the input-dependent weights to our models inspired by the success of Transformer architectures [20], [46] and dynamic networks [8], [23], [29].

Our experiments on multiple visual recognition tasks and datasets verify the effectiveness of GFNet. With a similar architecture, our model outperforms the recent vision Transformer and MLP models including DeiT [65], ResMLP [64] and gMLP [45]. When using the hierarchical architecture, GFNet can further enlarge the gap. With better efficiency, GFNet models also enjoy the similar scalability to vision Transformers and can be easily scaled up with larger model sizes (from 7 M to 251 M parameters) and more training data (from ImageNet-1 k to ImageNet-21k [15]). Our models work well on both ImageNet classification and downstream transfer learning, semantic segmentation, and object detection tasks, showing strong performance on multiple benchmarks (e.g., 85.0% top-1 accuracy on ImageNet-1 k without any extra data or supervision, 87.4% accuracy with ImageNet-21 k pre-training, and 54.3 mIoU on ADE20 k val). Our results demonstrate that GFNet can be a very competitive alternative to Transformer-style models and CNNs in terms of efficiency, generalization ability and robustness.

This paper is an extended version of our conference paper [57]. We make several new contributions: 1) We propose an enhanced version of GFNet by improving the micro designs of our basic block and introducing dynamic weights to our architecture inspired by the successful designs in CNNs and

modified vision Transformers; 2) We explore larger models and train our models on a larger dataset. We also evaluate our architecture on more tasks beyond image classification including semantic segmentation, object detection, and instance segmentation, where our models achieve very competitive performance; 3) We provide a unified view of vision Transformers, MLP-like models and GFNet to better understand our new operation. We also add more results, analysis and visualization of our models.

II. Related Work

In this section, we briefly review recent progress in several related topics: vision Transformers, MLP-like models, recent convolution-based models, and applications of Fourier transform in vision.

Vision Transformers. Since Dosovitskiy et al. [20] introduce Transformers [67] to image classification and achieve a competitive performance compared to CNNs, Transformers begin to exhibit their potential in various vision tasks [5], [7], [79]. Recently, there are a large number of works which aim to improve the Transformers for vision tasks [21], [33], [46], [65], [66], [71], [77]. These works either seek for better training strategies [21], [65] or design better architectures [46], [71], [77] or both [21], [66]. Most of the architecture modification of the Transformers [33], [46], [71], [77] introduces additional inductive biases similar to CNNs. There are also many promising methods to improve the efficiency of Transformers by modifying the self-attention operation for lower time complexity and memory access like Linformer [68], Performer [11], FlashAttention [14] and SOFT [49]. In this work, we propose a different solution by designing a new and more efficient operation to replace the heavy self-attention layer ($\mathcal{O}(L^2)$) while maintaining the ability to model the global interactions among tokens.

MLP-Like Models. Recently, there is some work that questions the importance of self-attention in the vision Transformers and proposes to use a simple MLP to replace the self-attention layer in the Transformers [45], [63], [64]. This line of work later is extended by using highly efficient operations to model spatial interactions and building models only with the operation and MLPs [6], [75], [78]. As the first deep all-MLP model for large-scale vision tasks, MLP-Mixer [63] employs MLPs to perform token mixing and channel mixing alternatively in each block. ResMLP [64] adopts a similar idea but substitutes the Layer Normalization with an Affine transformation for acceleration. gMLP [45] uses a spatial gating unit to re-weight tokens in the spatial dimension. However, all of the above models include MLPs to mix the tokens spatially, which brings two drawbacks: (1) like the self-attention in the Transformers, the spatial MLP still requires computational complexity quadratic to the length of tokens. (2) unlike Transformers, MLP models are hard to scale up to higher resolution since the weights of the spatial MLPs have fixed sizes. Our work follows this trend and successfully resolves the above issues in MLP-like models. The proposed GFNet enjoys log-linear complexity and can be easily scaled up to any resolution.

Recent Convolution-Based Models. The past decade has witnessed the remarkable advances driven by the progress of convolutional neural networks (CNNs) for various computer vision tasks [25], [37], [60], [61]. While classic CNN models like LeNet [38], VGG [60] and ResNet usually consider the relative small kernel size (typically 3×3), inspired by the recent success of vision Transformers and MLP-like models, some concurrent work of GFNet [57] explores new visual encoders with large-kernel convolutions as the basic operation for modeling spatial interactions [23]. Most recently, Liu et al. [47] propose a new series of new CNN models based on 7×7 convolutions and Transformer-style basic block. Ding et al. [18] show that CNN models with very large kernels and re-parameterization technique can also achieve competitive performance with Swin [46]. Different from these models with carefully designed kernel sizes, we show that the simple global filter designs with minimal inductive biases can attain comparable performance with state-of-the-art hierarchical vision Transformers like Swin. We also propose a new way to combine conventional local convolution and the proposed global filter and demonstrate good generalization ability across tasks and model sizes.

Applications of Fourier Transform in Vision. Fourier transform has been an important tool in digital image processing for decades [2], [54]. With the breakthroughs of CNNs in vision [24], [25], there are a variety of works that start to incorporate Fourier transform in some deep learning method [17], [39], [41], [74]. Some of these works employ discrete Fourier transform to convert the images to the frequency domain and leverage the frequency information to improve the performance in certain tasks [39], [74], while others utilize the convolution theorem to accelerate the CNNs via fast Fourier transform (FFT) [17], [41]. In this work, we propose to use learnable filters to interchange information among the tokens in the Fourier domain, inspired by the frequency filters in the digital image processing [54]. We also take advantage of some properties of FFT to reduce computational costs and the number of parameters.

III. Method

In this section, we will introduce the details of *GFNet*. We first review the key techniques in discrete Fourier transform that are used in our models in Secion III-A. Then, we elaborate on the detailed designs of the transformer-style GFNet in Section III-B. We also provide a unified view of vision Transformers, pure MLP models and GFNet to highlight the motivation of our models in Section III-C. Lastly, we present the hierarchical GFNet models that combine the successful designs of CNNs and our global filter layer in Section III-D, which can achieve better accuracy-complexity trade-off and be easily adapted to various downstream tasks.

A. Preliminaries: Discrete Fourier Transform

We start by introducing the discrete Fourier transform (DFT), which plays an important role in the area of digital signal processing and is a crucial component in our GFNet. For clarity, We first consider the 1D DFT. Given a sequence of N complex numbers $x[n], 0 \leq n \leq N-1$, the 1D DFT converts the sequence into the frequency domain by

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)kn} := \sum_{n=0}^{N-1} x[n] W_N^{kn}, \qquad (1)$$

where j is the imaginary unit and $W_N = e^{-j(2\pi/N)}$. The formulation of DFT in (1) can be derived from the Fourier transform for continuous signal by sampling in both the time domain and the frequency domain (see Supplementary Material for details), (available online). Since $X[k]$ repeats on intervals of length N, it suffices to take the value of $X[k]$ at N consecutive points $k = 0, 1, \ldots, N-1$. Specifically, $X[k]$ represents to the spectrum of the sequence $x[n]$ at the frequency $\omega_k = 2\pi k/N$.

It is also worth noting that DFT is a one-to-one transformation. Given the DFT $X[k]$, we can recover the original signal $x[n]$ by the inverse DFT (IDFT)

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j(2\pi/N)kn}. \tag{2}$$

For real input $x[n]$, it can be proved that (see Supplementary Material), available online, its DFT is conjugate symmetric, i.e., $X[N-k] = X^*[k]$. The reverse is true as well: if we perform IDFT to $X[k]$ which is conjugate symmetric, a real discrete signal can be recovered. This property implies that the half of the DFT $\{X[k] : 0 \leq k \leq \lceil N/2 \rceil\}$ contains the full information about the frequency characteristics of $x[n]$.

DFT is widely used in modern signal processing algorithms for mainly two reasons: (1) The input and output of DFT are both discrete and thus can be easily processed by computers; (2) There exist efficient algorithms for computing the DFT. The *fast Fourier transform* (FFT) algorithms take advantage of the symmetry and periodicity properties of W_N^{kn} and reduce the complexity to compute DFT from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$. The inverse DFT (2), which has a similar form to the DFT, can also be computed efficiently using the inverse fast Fourier transform (IFFT).

The DFT described above can be extend to 2D signals. Given the 2D signal $X[m, n], 0 \leq m \leq M-1, 0 \leq n \leq N-1$, the 2D DFT of $x[m, n]$ is given by

$$X[u, v] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m, n] e^{-j2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)}. \tag{3}$$

The 2D DFT can be viewed as performing 1D DFT on the two dimensions alternatively. Similar to 1D DFT, 2D DFT of real input $x[m, n]$ satisfied the conjugate symmetry property $X[M-u, N-v] = X^*[u, v]$. The FFT algorithms can also be applied to 2D DFT to improve computational efficiency.

B. Global Filter Networks

Recent advances in vision Transformers [20], [65] demonstrate that models based on self-attention can achieve competitive performance even without the inductive biases associated with the convolutions. Henceforth, there are several works [63], [64] that exploit approaches (e.g., MLPs) other than self-attention to mix the information among the tokens. The proposed Global Filter Networks (GFNet) follows this line of work and aims to replace the heavy self-attention layer ($\mathcal{O}(N^2)$) with a simpler and more efficient one.

The overall framework of our model is depicted in Fig. 1. Our model takes as an input $H \times W$ non-overlapping patches and projects the flattened patches into $L = HW$ tokens with dimension D. The basic building block of GFNet consists of: 1) a *global filter layer* that can exchange spatial information efficiently ($\mathcal{O}(L \log L)$); 2) a feedforward network (FFN) as in [20], [65]. The output tokens of the last block are fed into a global average pooling layer followed by a linear classifier.

Algorithm 1: Pseudocode of Global Filter Layer.

```
# x: the token features, B x H x W x D
 (where N = H * W)
# K: the frequency-domain filter, H x
 W_hat x D (where W_hat = W // 2 + 1,
 see Section III-B for details)
X = rfft2(x, dim=(1, 2))
X_tilde = X * K
x = irfft2(X_tilde, dim=(1, 2))
rfft2/irfft2: 2D FFT/IFFT for real signal
```

Global Filter Layer. We propose the global filter layer as an alternative to the self-attention layer which can mix tokens representing different spatial locations. Given the tokens $\boldsymbol{x} \in \mathbb{R}^{H \times W \times D}$, we first perform 2D FFT (see Section III-A) along the spatial dimensions to convert $\boldsymbol{x}$ to the frequency domain

$$\boldsymbol{X} = \mathcal{F}[\boldsymbol{x}] \in \mathbb{C}^{H \times W \times D}, \tag{4}$$

where $\mathcal{F}[\cdot]$ denotes the 2D FFT. Note that $\boldsymbol{X}$ is a complex tensor and represents the spectrum of $\boldsymbol{x}$. We can then modulate the spectrum by multiplying a learnable filter $\boldsymbol{K} \in \mathbb{C}^{H \times W \times D}$ to the $\boldsymbol{X}$

$$\tilde{\boldsymbol{X}} = \boldsymbol{K} \odot \boldsymbol{X}, \tag{5}$$

where $\odot$ is the element-wise multiplication (also known as the Hadamard product). The filter $\boldsymbol{K}$ is called the *global filter* since it has the same dimension as $\boldsymbol{X}$, which can represent an arbitrary filter in the frequency domain. Finally, we adopt the inverse FFT to transform the modulated spectrum $\tilde{X}$ back to the spatial domain and update the tokens

$$\boldsymbol{x} \leftarrow \mathcal{F}^{-1}[\tilde{\boldsymbol{X}}]. \tag{6}$$

The formulation of the global filter layer is motivated by the frequency filters in the digital image processing [54], where the global filter $\boldsymbol{K}$ can be regarded as a set of learnable frequency filters for different hidden dimensions. It can be proved (see Supplementary Material), available online, that the global filter layer is equivalent to a depth-wise *global circular convolution* with the filter size $H \times W$. Therefore, the global filter layer is different from the standard convolutional layer which adopts a relatively small filter size to enforce the inductive biases of the locality. We also find although the proposed global filter can also be interpreted as a spatial domain operation, the filters learned in our networks exhibit more clear patterns in the frequency domain than the spatial domain, which indicates our models tend to capture relations in the frequency domain instead of spatial domain (see Fig. 4). Note that the global filter implemented in the frequency domain is also much more efficient compared to the spatial domain, which enjoys a complexity of $\mathcal{O}(DL \log L)$

while the vanilla depth-wise global circular convolution in the spatial domain has $\mathcal{O}(DL^2)$ complexity. We will also show that the global filter layer is better than its local convolution counterparts in the experiments.

It is also worth noting that in the implementation, we make use of the property of DFT to reduce the redundant computation. Since $\boldsymbol{x}$ is a real tensor, its DFT $\boldsymbol{X}$ is conjugate symmetric, i.e., $\boldsymbol{X}[H-u, W-v, :] = \boldsymbol{X}^*[H, W, :]$. Therefore, we can take only the half of the values in the $\boldsymbol{X}$ but preserve the full information at the same time

$$\boldsymbol{X}_r = \boldsymbol{X}[:, 0:\widehat{W}] := \mathcal{F}_r[\boldsymbol{x}], \quad \widehat{W} = \lceil W/2 \rceil, \tag{7}$$

where $\mathcal{F}_r$ denotes the 2D FFT for real input. In this way, we can implement the global filter as $\boldsymbol{K}_r \in \mathbb{C}^{H \times \widehat{W} \times D}$, which can reduce half the parameters. This can also ensure $\mathcal{F}_r^{-1}[\boldsymbol{K}_r \odot \boldsymbol{X}_r]$ is a real tensor, thus it can be added directly to the input $\boldsymbol{x}$. The global filter layer can be easily in modern deep learning frameworks (e.g., PyTorch [53]), as shown in Algorithm 1. The FFT and IFFT are well supported by GPU and CPU thanks to the acceleration libraries like `cuFFT` and `mkl-fft`, which makes our models perform well on hardware.

Architecture Variants. To better compare with previous methods, we construct our basic GFNet models by using the identical overall architecture to ViT [20], MLP-Mixer [63] and ResMLP [64], and only modify the self-attention/MLP sublayers to the proposed global filter layer in each block. Different from widely used CNNs, this type of architecture adopts an isotropic design as the practice in NLP tasks [16], [67], where a fixed size of feature maps is used in all layers. We begin with a 12-layer model (*GFNet-XS*) with a similar architecture to DeiT-S and ResMLP-12. Then, we further develop 3 variants of the model (*GFNet-Ti*, *GFNet-S* and *GFNet-B*) by simply adjusting the depth to $\{12, 19, 19\}$ and the embedding dimension to $\{256, 384, 512\}$, which have similar complexity with the widely used ResNet-18, 50 and 101 models [25]. Another small modification to the architecture is that we use a single residual connection in each block for GFNet models since it will lead to 0.2% top-1 accuracy improvement on ImageNet.

C. An Unified View of Vision Transformers, Pure MLP Models and GFNet

GFNet follows the line of research about the exploration of approaches to mix the tokens. In this section, we provide an in-depth analysis of vision Transformers, pure MLP models, and GFNet, and show the connections and differences among these token mixing strategies from a unified perspective.

Given the input feature $\boldsymbol{x} \in \mathbb{R}^{L \times D}$, the general formulation that unifies the three operations can be written as

$$\begin{aligned} &\boldsymbol{x} \leftarrow \text{Mixer}(\boldsymbol{x}) = \text{Concat}_{1 \le h \le H}(\mathbf{M}_h(\boldsymbol{x}\mathbf{W}_h^V))\mathbf{W}^C, \\ &\mathbf{M}_h \in \mathbb{R}^{L \times L}, \mathbf{W}_h^V \in \mathbb{R}^{D \times D_h}, \mathbf{W}^C \in \mathbb{R}^{D \times D}, \end{aligned} \tag{8}$$

where we consider the multi-head mixing scenario with H heads. For the h-head, the input feature is first transformed with a linear projection $\mathbf{W}_h$. Then, we perform spatial interactions with mixing matrix $\mathbf{M}_h$ and concatenate the mixing results along the feature dimension. Lastly, we transform the aggregated feature with another linear projection $\mathbf{W}^C$. For vision Transformers [20], the widely used multi-head self-attention (MHSA) mechanism [67] can be derived from (8) by generating $\{\mathbf{M}_h\}_{h=1}^H$ with dot product and Softmax

$$\begin{aligned} &\mathbf{M}_h = \text{Softmax}\left(\frac{\boldsymbol{x}\mathbf{W}_h^Q(\boldsymbol{x}\mathbf{W}_h^K)^\top}{\sqrt{D_h}}\right), \\ &\mathbf{W}_h^Q \in \mathbb{R}^{D \times D_h}, \mathbf{W}_h^K \in \mathbb{R}^{D \times D_h}, \end{aligned} \tag{9}$$

where the $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V$ and $\mathbf{W}^C$ are all learnable parameters for linear projections. For all-MLP models [63], [64], the mixing process is simplified as a single learnable linear projection along the spatial dimension

$$H = 1, \mathbf{M}_1 = \mathbf{M}, \mathbf{W}_1^V = \mathbf{I}, \mathbf{W}^C = \mathbf{I}, \tag{10}$$

where $\mathbf{I} \in \mathbb{R}^{D \times D}$ is a identity matrix and $\mathbf{M}$ is the learnable parameters. Different from the all-MLP models, the proposed global filter layer provide a more efficient way for spatial mixing with more diverse mixing policies. Our global filter layer can be viewed as

$$H = D, \mathbf{M}_h = \mathcal{C}(\mathcal{F}^{-1}(\boldsymbol{K}_h)), \mathbf{W}_h^V = \mathbf{I}[:, h], \mathbf{W}^C = \mathbf{I}, \tag{11}$$

where $\mathbf{M}_h$ is the equivalent mixing matrix of the hth global filter, $\mathcal{C}(\cdot)$ is the function that transforms the IFFT of a frequency-domain filter to the corresponding circulant matrix, and $\mathbf{W}_h^V$ is the hth column of the identity matrix to select the hth channel of the input feature.

Compared to vision Transformers and pure MLP models, we exhibit that our solution has several favorable properties: 1) GFNet is more efficient. The complexity of both the vision Transformers [20], [65], [66] and the MLP models [63], [64] is $\mathcal{O}(L^2)$. Different from them, the global filter layer only consists of an FFT ($\mathcal{O}(L \log L)$), an element-wise multiplication ($\mathcal{O}(L)$), and an IFFT ($\mathcal{O}(L \log L)$), which means the total computational complexity is $\mathcal{O}(L \log L)$; 2) GFNet offers more diverse mixing policies with acceptable parameters, computation, and memory consumption. While self-attention and spatial MLP layers share a mixing matrix for a group of channels to save parameters and memory, our method mixes features of different channels with independent mixing policies. By leveraging the symmetric structure in FFT, we can scale the number of heads to D without explicitly saving $L \times L \times D$ parameters or computing $L \times L \times D$ mixing matrices; 3) GFNet can be easily adapted to different resolutions. Although pure MLP models are simpler compared to Transformers, it is hard to fine-tune them on higher resolution (e.g. , from 224×224 resolution to 384×384 resolution) since they can only process a fixed number of tokens. As opposed to pure MLP models, we will show that our GFNet can be easily scaled up to higher resolution. Our model is more flexible since both the FFT and the IFFT have no learnable parameters and can process sequences with arbitrary lengths. We can simply interpolate the global filter $\boldsymbol{K}$ to $\boldsymbol{K}' \in \mathbb{C}^{H' \times W' \times D}$ for different inputs, where $H' \times W'$ is the target size. The interpolation is reasonable due to the property

of DFT. Each element of the global filter $\boldsymbol{K}[u, v]$ corresponds to the spectrum of the filter at $\omega_u = 2\pi u/H, \omega_v = 2\pi v/W$ and thus, the global filter $\boldsymbol{K}$ can be viewed as a sampling of a continuous spectrum $\boldsymbol{K}(\omega_u, \omega_v)$, where $\omega_u, \omega_v \in [0, 2\pi]$. Hence, changing the resolution is equivalent to changing the sampling interval of $\boldsymbol{K}(\omega_u, \omega_v)$. Therefore, we only need to perform interpolation to shift from one resolution to another. The connections and differences among these three types of models are also summarized in the right part of Fig. 1.

We also notice recently a concurrent work FNet [40] leverages Fourier transform to mix tokens. Our work is distinct from FNet in three aspects: (1) FNet performs FFT to the input and directly adds the real part of the spectrum to the input tokens, which blends the information from different domains (spatial/frequency) together. On the other hand, GFNet draws motivation from the frequency filters, which is more reasonable. (2) FNet only keeps the real part of the spectrum. Note that the spectrum of real input is conjugate symmetric, which means the real part is exactly symmetric and thus contains redundant information. Our GFNet, however, utilizes this property to simplify the computation. (3) FNet is designed for NLP tasks, while our GFNet focuses on vision tasks. In our experiments, we also show that GFNet can outperform FNet by a large margin on vision tasks.

D. Hierarchical Global Filter Networks

Due to the limitation from the quadratic complexity in the self-attention, vision Transformers [20], [65] and all-MLP models [45], [63], [64] are usually designed to process a relatively small feature map (typically 14×14). However, our GFNet, which enjoys log-linear complexity, avoids that problem. Since in our GFNet the computational costs do not grow such significantly when the feature map size increases, we can adopt a hierarchical architecture inspired by the success of CNNs [25], [37]. Generally speaking, we can start from a large feature map (e.g., 56×56) and gradually perform downsampling after a few blocks. Therefore, we develop a series of CNN-style hierarchical GFNet models (*GFNet-H*) with only the proposed global filter layers, MLPs and the downsampling layers to show the potential of our global filter layer to process features with multiple resolutions. We use 4×4 patch embedding to form the input tokens and use a non-overlapping convolution layer (i.e., Patch Merging layer in [46]) to downsample tokens following [46], [69]. The high efficiency of our GFNet makes it possible to *directly* process a high-resolution feature map in the early stages (e.g., $H/4 \times W/4$) without introducing any handcraft structures like Swin [46]. Therefore, we directly apply our building block on different stages without any modifications.

While the direct application of our global filter layer yields a better accuracy-complexity trade-off than the isotropic ViTs and GFNet, we propose to further improve our models by introducing the successful experiences in CNNs and vision Transformers.

Global-Local Filter Layer. While the proposed global filter layer can efficiently capture the long-term relations for visual understanding, the over-smoothed features may lack locality that is critical for downstream vision tasks like object detection. Previous work [21] also suggests that the over-smoothing phenomenon in vision Transformers may affect the generalization ability on unseen samples. To tackle this problem, we propose a new two-path layer that separately and explicitly captures global and local spatial relations with our global filter operation and the conventional local convolution respectively. The resulting global-local filter enjoys both the advantages of our global filter that can efficiently mix spatial information from arbitrary ranges and the local filter that enhances the locality of the feature maps. Specifically, given the input feature $\boldsymbol{x} \in \mathbb{R}^{H \times W \times D}$ that is normalized by LayerNorm [1], we first divide the feature along the channel dimension to form two groups of feature maps with the same size $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^{H \times W \times \frac{D}{2}}$. Then, we use a global filter ϕ_{global} and a local filter ϕ_{local} to process these two groups of features separately

$$\boldsymbol{x} \leftarrow \text{Concat}(\phi_{\text{global}}(\boldsymbol{x}_1), \phi_{\text{local}}(\boldsymbol{x}_2)). \tag{12}$$

In practice, the local filter ϕ_{local} can be implemented as a depth-wise convolution [10] with a kernel size of 3×3. Note that although recent work [76] shows a simple average pooling layer can also serve as the local filter, our experimental results show that the parametric convolution layer actually leads to a significantly better performance with very limited additional computation and parameters.

Apart from the above modification in our basic block, we also conduct an extensive study on the combination of conventional local convolutions and the proposed global filter layer at the macro architecture level (see our results in Table IX). We find that using the small kernel convolutions in the early stages and global filters in the late stages will lead to better performance on both ImageNet and downstream tasks. Therefore, we use different basic blocks in different stages in our improved hierarchical models.

Dynamic Weights. The dynamic attention weights that are generated conditioned on the input sample are usually recognized as one of the key advantages of vision Transformers over the conventional static backbones like ResNet [25]. Inspired by the self-attention mechanism and previous work on dynamic neural networks, we propose to further enhance our GFNet models by introducing the input-adaptive weights following [8], [23], [29]. Specifically, we investigate the following three kinds of strategies to integrate dynamic weights into our architecture:

- *Dynamic filter generation* [23], [32]. Since the parameters of the global filter have the same size as the input feature maps, a straightforward way to obtain input-dependent weights is generating the weights from the features with a point-wise MLP layer φ_{gen}. In this case, the improved global filter layer can be written as

$$\boldsymbol{x} \leftarrow \mathcal{F}^{-1}[\mathcal{F}[\varphi_{\text{gen}}(\boldsymbol{x})] \odot \mathcal{F}[\boldsymbol{x}]], \tag{13}$$

  where the frequency-domain filter is obtained by applying FFT to the transformed features. Following previous work [23], [29], we adopt the bottleneck design for the MLP layer φ_{gen} with a reduction ratio of 8 to reduce the extra computation and parameters brought by the weight generation process.

- *Dynamic filter selection* [8]. Different from directly generating the filters, an alternative method is predicting a mixing policy with a small MLP φ_{sel} and generating the filter by linearly blending a set of learnable candidate filters $\{\boldsymbol{K}_i\}_{i=1}^{k}$

$$\boldsymbol{K} \leftarrow \sum_{i=1}^{k} \alpha_i \boldsymbol{K}_i, \quad \{\alpha_i\}_{i=1}^{k} = \varphi_{\text{sel}}(\mathcal{A}(\boldsymbol{x})), \tag{14}$$

  where $\mathcal{A}$ is an aggregation function like global average pooling to reduce the spatial dimension of $\boldsymbol{x}$ and k is the number of candidate filters.
- *Dynamic channel gating* [29], [30]. A simpler way to introduce input-adaptive weights is to re-weight channels based on the global representation of the input like [29], [30]. This strategy can be written as

$$\boldsymbol{x} \leftarrow \varphi_{\text{gate}}(\mathcal{A}(\boldsymbol{x})) \odot \boldsymbol{x}, \tag{15}$$

  where φ_{gate} is also an MLP layer with the bottleneck design and the element-wise multiplication is performed by expanding the spatial dimensions of the generated weights. Since the dynamic channel gating operation is not directly associated with the global filter layer, we also investigate the locations to add this operation and find that adding the weights to expanded features in FFN will lead to the best performance.

Our results (see Table IX) show that dynamic weights can generally improve the performance of our models with a similar level of complexity, which further confirms the value of dynamic structures in vision backbones. It is surprising that simple channel gating can achieve the best trade-off between accuracy and complexity. A possible explanation is that the final weights can be factorized as the input-dependent and input-agnostic components and a channel-wise re-weighting is sufficient to model the input-specific patterns. We believe that an in-depth analysis of the differences behind various dynamic structures can be an interesting future direction. Based on our experimental results, we use the channel gating strategy as the default configuration in our improved hierarchical models.

Architecture Variants. We develop two types of hierarchical GFNet models including 1) *GFNet-H* series that aims to examine the ability of the proposed global filter layer as the only operation to model spatial relations and directly process high-resolution features, and 2) *GFNet-H++* series that explores high-performance architectures equipped with the improved global-local filter layer and dynamic weights. For *GFNet-H* series, we design three models (*GFNet-H-Ti*, *GFNet-H-S* and *GFNet-H-B*) that have the three levels of complexity as ResNet-18,50 and 101 [25] following the practice in PVT [69]. For *GFNet-H++* series, we develop the counterparts *GFNet-H-Ti++*, *GFNet-H-S++* and *GFNet-H-B++*, and further verify the scaling ability of our architecture by constructing larger models *GFNet-H-L++* and *GFNet-H-XL++*. In hierarchical models, we use a similar strategy to ResNet [25] to increase network depth by fixing the number of blocks for the stage 1, 2, 4 to 3, and adjusting the number of blocks in stage 3. For small and larger hierarchical models, we adopt LayerScale normalization [66] to stabilize the training process. The detailed architectures of our hierarchical models are summarized in Table II.

IV. Experiments

We conduct extensive experiments to verify the effectiveness of our GFNet. We present the main results on ImageNet [15] and compare them with various architectures. We also test our models on the downstream transfer learning datasets and show the potential of our models on dense prediction tasks on commonly used semantic segmentation benchmark ADE20k [81] and object detection dataset COCO [44]. Lastly, we provide an in-depth analysis of our basic layer and architectural choices and investigate the efficiency and robustness of the proposed models.

A. ImageNet Classification

1) Setups: ImageNet-1k Training. We first test our models by training and evaluating on ImageNet-1k [15], which is a widely used large-scale benchmark for image classification. ImageNet-1 k contains roughly 1.2 M images from 1,000 categories. Following common practice [25], [65], we train our models on the training set of ImageNet and report the single-crop top-1 accuracy on 50,000 validation images. To fairly compare with previous works [64], [65], we follow the most training details for our models and do not add extra regularization methods like [33]. Different from [65], we do not use EMA model [55], RandomEarse [80] and repeated augmentation [28], which are important to train DeiT while sightly hurting the performance of our models. We train our models for 300 epochs using the AdamW optimizer [48]. All of our models are trained with 224×224 images. More details can be found in Supplementary Materials, available online.

ImageNet-22k Training. ImageNet-22k [15] is a larger dataset with >21 k classes and around 14 M images for pre-training. We use the subset suggested by [59], where roughly half categories and only 13% images are removed to reduce the classes with few samples. We only train our GFNet-H-L++ and GFNet-H-XL++ models on this dataset to evaluate the scaling ability of our models. We follow previous practice [46] to train our models for 90 epochs and use a similar data augmentation strategy as ImageNet-1 k experiments. We set the stochastic depth coefficient [31] to 0.2 and 0.3 for GFNet-H-L++ and GFNet-H-XL++, respectively. We set the initial learning rate as 0.001 with a batch size of 4090 and decay the learning rate using the cosine schedule. We also use a linear warm-up learning rate in the first 5 epochs.

ImageNet-1k Fine-Tuning. We fine-tune the models pre-trained on ImageNet-22 k to ImageNet-1 k for 30 epochs with a batch size of 1,024 and a cosine learning rate schedule with an initial learning rate of $5e^{-5}$. We set the weight decay to $1e^{-6}$ and use the data augmentation strategy in the ImageNet-1 k training experiments. We use the corresponding classifier weights for ImageNet-22 k classes to initialize the ImageNet-1 k classifier.

2) Comparisons With Transformer-Style Architectures: The results are presented in Table III. We compare our method with Transformer-style architectures for image classification including vision Transformers DeiT [65] and MLP-like models

TABLE II

THE DETAILED ARCHITECTURES OF HIERARCHICAL GFNET VARIANTS. WE ADOPT HIERARCHICAL ARCHITECTURES WHERE THE WE USE PATCH EMBEDDING LAYER TO PERFORM DOWNSAMPLING. "$\downarrow n$" INDICATES THE STRIDE OF THE DOWNSAMPLING IS n. "GFBLOCK(D)" REPRESENTS ONE BUILDING BLOCK OF GFNET WITH EMBEDDING DIMENSION D. WE SET THE MLP EXPANSION RATIO TO 4 FOR ALL THE FEEDFORWARD NETWORKS

	Output Size	GFNet-H-Ti/Ti++	GFNet-H-S/S++	GFNet-H-B/B++	GFNet-H-L++	GFNet-H-XL++
Stage1	$\frac{H}{4} \times \frac{W}{4}$	Patch Embed. $\downarrow 4$ GFBlock(64) $\times$ 3	Patch Embed. $\downarrow 4$ GFBlock(96) $\times$ 3	Patch Embed. $\downarrow 4$ GFBlock(96) $\times$ 3	Patch Embed. $\downarrow 4$ GFBlock(128) $\times$ 3	Patch Embed. $\downarrow 4$ GFBlock(196) $\times$ 3
Stage2	$\frac{H}{8} \times \frac{W}{8}$	Patch Embed. $\downarrow 2$ GFBlock(128) $\times$ 3	Patch Embed. $\downarrow 2$ GFBlock(192) $\times$ 3	Patch Embed. $\downarrow 2$ GFBlock(192) $\times$ 3	Patch Embed. $\downarrow 2$ GFBlock(256) $\times$ 3	Patch Embed. $\downarrow 2$ GFBlock(384) $\times$ 3
Stage3	$\frac{H}{16} \times \frac{W}{16}$	Patch Embed. $\downarrow 2$ GFBlock(256) $\times$ 10	Patch Embed. $\downarrow 2$ GFBlock(384) $\times$ 10	Patch Embed. $\downarrow 2$ GFBlock(384) $\times$ 27	Patch Embed. $\downarrow 2$ GFBlock(512) $\times$ 27	Patch Embed. $\downarrow 2$ GFBlock(768) $\times$ 27
Stage4	$\frac{H}{32} \times \frac{W}{32}$	Patch Embed. $\downarrow 2$ GFBlock(512) $\times$ 3	Patch Embed. $\downarrow 2$ GFBlock(768) $\times$ 3	Patch Embed. $\downarrow 2$ GFBlock(768) $\times$ 3	Patch Embed. $\downarrow 2$ GFBlock(1024) $\times$ 3	Patch Embed. $\downarrow 2$ GFBlock(1536) $\times$ 3
Classifier		Global Average Pooling, Linear				

TABLE III

COMPARISONS WITH TRANSFORMER-STYLE ARCHITECTURES ON IMAGENET. WE COMPARE DIFFERENT TRANSFORMER-STYLE ARCHITECTURES FOR IMAGE CLASSIFICATION INCLUDING VISION TRANSFORMERS [65], MLP-LIKE MODELS [45], [64] AND OUR MODELS THAT HAVE COMPARABLE FLOPS AND THE NUMBER OF PARAMETERS. WE REPORT THE TOP-1 ACCURACY ON THE VALIDATION SET OF IMAGENET AS WELL AS THE NUMBER OF PARAMETERS AND FLOPS. ALL OF OUR MODELS ARE TRAINED WITH 224×224 IMAGES. WE USE "$\uparrow$384" TO REPRESENT MODELS FINETUNED ON 384×384 IMAGES FOR 30 EPOCHS. OUR MODELS ARE HIGHLIGHTED IN GRAY

Model	Image Size	Params (M)	FLOPs (G)	Throughput (im/s)	Top-1 Acc. (%)
DeiT-Ti [65]	224^2	5	1.2	3231	72.2
gMLP-Ti [45]	224^2	6	1.4	1591	72.0
GFNet-Ti	224^2	7	1.3	3452	74.6
ResMLP-12 [64]	224^2	15	3.0	1864	76.6
GFNet-XS	224^2	16	2.8	1913	78.6
DeiT-S [65]	224^2	22	4.6	1328	79.8
gMLP-S [45]	224^2	20	4.5	592	79.4
GFNet-S	224^2	25	4.5	1299	80.0
ResMLP-36 [64]	224^2	45	8.9	591	79.7
GFNet-B	224^2	43	7.9	748	80.7
GFNet-XS$\uparrow$384	224^2	18	8.4	635	80.6
DeiT-B [65]	224^2	86	17.5	412	81.8
gMLP-B [45]	224^2	73	15.8	203	81.6
GFNet-S$\uparrow$384	224^2	28	13.2	414	81.7
GFNet-B$\uparrow$384	224^2	47	23.3	262	82.1

ResMLP [64] and gMLP [45] that have similar complexity and number of parameters. We see that our method can clearly outperform recent MLP-like models like ResMLP [64] and gMLP [45], and show similar performance with DeiT. Specifically, GFNet-XS outperforms ResMLP-12 by 2.0% while having slightly fewer FLOPs. GFNet-S also achieves better top-1 accuracy compared to gMLP-S and DeiT-S. Our tiny model is significantly better compared to both DeiT-Ti (+2.4%) and gMLP-Ti (+2.6%) with a similar level of complexity. Benefiting from the highly efficient implementation of FFT and IFFT on GPU, our models also achieve higher throughput than ViT and ResMLP architectures that have similar theoretical complexity. These results suggest that our global filter layer is a new hardware-friendly and feasible basic operation in many application scenarios.

3) Fine-Tuning at Higher Resolution: One prominent problem of MLP-like models is that the feature resolution is not adjustable. On the contrary, the proposed global filter is more flexible. We demonstrate the advantage of GFNet by fine-tuning the model trained at 224×224 resolution to higher resolution following the practice in vision Transformers [65]. As shown in Table III, our model can easily adapt to higher resolution with only 30-epoch fine-tuning and achieve better performance.

4) Comparisons With Hierarchical Architectures: We compare different kinds of hierarchical models [6], [13], [19], [25], [42], [46], [56], [70] in Table IV. we focus on representative baseline architectures including convolutional neural networks RegNet [56], hierarchical MLP-like model CycleMLP [6], hierarchical vision Transformers Swin [46] Benefiting from the log-linear complexity, GFNet-H models show significantly better performance than ResNet [25] and RegNet [56], and achieve similar performance with Swin while having a much simpler and more generic design. Equipped with the improved global-local filter layer and dynamic weights, the GFNet-H++ series achieves significantly better performance with various model sizes (from tiny models to large models) and resolutions (both 224×224 and 384×384). Compared to conventional CNN models, our models show strong scaling ability like the recent vision Transformers. Compared to hierarchical vision Transformers, small, base and large GFNet-H++ models outperform widely used hierarchical vision Transformer Swin [46] models with similar FLOPs by +1.3%, +0.5%, and +0.4%, respectively. These results clearly demonstrate our GFNet models are very competitive alternatives to state-of-the-art CNN and vision Transformers on image classification tasks.

5) Pre-Training on ImageNet-22k: To further test the scaling ability of our models, we scale the size of our hierarchical model to 34 G FLOP and over 200 M parameters by pre-training models on ImageNet-22 k. The results of pre-training on ImageNet-22 k, and fine-tuning and evaluating on ImageNet-1 k are presented in Table IV. We see our models can consistently outperform the Swin series when increasing the model size and image resolution with a similar pre-training strategy. Our results demonstrate that GFNet models can also generalize to the larger pre-training dataset and achieve better performance with more parameters and computation.

TABLE IV
COMPARISONS WITH HIERARCHICAL ARCHITECTURES ON IMAGENET. WE COMPARE DIFFERENT HIERARCHICAL ARCHITECTURES FOR IMAGE CLASSIFICATION, WHERE WE FOCUS ON REPRESENTATIVE BASELINE ARCHITECTURES INCLUDING CONVOLUTIONAL NEURAL NETWORKS REGNET [56], HIERARCHICAL MLP-LIKE MODEL CYCLEMLP [6], HIERARCHICAL VISION TRANSFORMERS SWIN [46] AND OUR HIERARCHICAL MODELS THAT HAVE COMPARABLE FLOPS AND NUMBER OF PARAMETERS. WE REPORT THE NUMBER OF PARAMETERS, FLOPS AND TOP-1 ACCURACY ON THE VALIDATION SET OF IMAGENET AS WELL AS THE THROUGHPUT MEASURED ON AN NVIDIA RTX 3090 GPU WITH A FIXED BATCH SIZE OF 128. ALL OUR MODELS ARE TRAINED AND TESTED WITH 224×224 IMAGES. WE USE "↑384" TO REPRESENT MODELS FINETUNED ON 384×384 IMAGES FOR 30 EPOCHS. ‡INDICATES THE MODEL IS PRE-TRAINED ON IMAGENET-22 K AND FINE-TUNED ON IMAGENET-1 K. OUR MODELS ARE HIGHLIGHTED IN GRAY

Model	Image Size	Params (M)	FLOPs (G)	Throughput (im/s)	Top-1 Acc. (%)
ResNet-18 [25]	224^2	12	1.8	-	69.8
ResNet-50 [25]	224^2	26	4.1	-	76.1
ResNet-101 [25]	224^2	45	7.9	-	77.4
CoAtNet-0 [13]	224^2	25	4.2	-	81.6
CoAtNet-1 [13]	224^2	42	8.4	-	83.3
CoAtNet-2 [13]	224^2	75	15.7	-	84.1
CoAtNet-3 [13]	224^2	168	34.7	-	84.5
CSWin-T [19]	224^2	23	4.3	-	82.7
CSWin-S [19]	224^2	35	6.9	-	83.6
CSWin-B [19]	224^2	78	15.0	-	84.2
PVTv2-B1 [70]	224^2	13	2.1	-	78.7
PVTv2-B2 [70]	224^2	25	4.0	-	82.0
PVTv2-B3 [70]	224^2	45	6.9	-	83.2
PVTv2-B4 [70]	224^2	63	10.1	-	83.6
PVTv2-B5 [70]	224^2	82	11.8	-	83.8
MViTv2-T [42]	224^2	24	4.7	-	82.3
MViTv2-S [42]	224^2	35	7.0	-	83.6
MViTv2-B [42]	224^2	52	10.2	-	84.4
RegNetY-1.6GF [56]	224^2	11	1.6	1285	78.0
CycleMLP-B1 [6]	224^2	15	2.1	1366	78.9
GFNet-H-Ti	224^2	15	2.0	1491	80.1
GFNet-H-Ti++	224^2	17	2.1	1801	81.2
GFNet-H-Ti++↑384	384^2	18	6.1	606	82.0
RegNetY-4.0GF [56]	224^2	21	4.0	862	80.0
CycleMLP-B2 [6]	224^2	27	3.9	823	81.6
Swin-Ti [46]	224^2	29	4.5	850	81.3
GFNet-H-S	224^2	32	4.5	887	81.5
GFNet-H-S++	224^2	37	4.6	992	82.5
Swin-Ti↑384 [46]	384^2	29	13.5	249	82.2
GFNet-H-S++↑384	384^2	37	13.6	326	83.5
RegNetY-8.0GF [56]	224^2	39	8.0	594	81.7
CycleMLP-B4 [6]	224^2	52	10.1	336	83.0
Swin-S [46]	224^2	50	8.7	518	83.0
GFNet-H-B	224^2	54	8.4	543	82.9
GFNet-H-B++	224^2	62	8.6	576	83.5
Swin-S↑384 [46]	384^2	50	26.1	154	83.9
GFNet-H-B++↑384	384^2	65	25.3	191	84.6
RegNetY-16.0GF [56]	224^2	84	16.0	401	82.9
CycleMLP-B5 [6]	224^2	76	12.3	309	83.2
Swin-B [46]	224^2	88	15.4	334	83.5
GFNet-H-L++	224^2	110	15.3	371	83.9
Swin-B↑384 [46]	384^2	88	47.1	103	84.5
GFNet-H-L++↑384	384^2	112	44.7	128	85.0
Swin-B‡ [46]	224^2	88	15.4	334	85.2
GFNet-H-L++‡	224^2	110	15.3	371	85.8
Swin-B↑384‡ [46]	384^2	88	47.1	103	86.4
GFNet-H-L++↑384‡	384^2	112	44.7	128	86.6
Swin-L‡ [46]	224^2	197	34.5	162	86.3
GFNet-H-XL++‡	224^2	247	34.2	185	86.5
Swin-L↑384‡ [46]	384^2	197	103.9	51	87.3
GFNet-H-XL++↑384‡	384^2	253	100.6	64	87.4

TABLE V
RESULTS ON TRANSFER LEARNING DATASETS. WE REPORT THE TOP-1 ACCURACY ON THE FOUR DATASETS AS WELL AS THE NUMBER OF PARAMETERS AND FLOPS

Model	FLOPs	Params	C10	C100	Flo.	Cars
ResNet50 [25]	4.1G	26M	-	-	96.2	90.0
EffNet-B7 [62]	37G	66M	98.9	91.7	98.8	94.7
ViT-B/16 [20]	55.4G	86M	98.1	87.1	89.5	-
ViT-L/16 [20]	190.7G	307M	97.9	86.4	89.7	-
Deit-B/16 [65]	17.5G	86M	99.1	90.8	98.4	92.1
ResMLP-12 [64]	3.0G	15M	98.1	87.0	97.4	84.6
ResMLP-24 [64]	6.0G	30M	98.7	89.5	97.9	89.5
GFNet-XS	2.8G	16M	98.6	89.1	98.1	92.8
GFNet-H-B	8.4G	54M	99.0	90.3	98.8	93.2

B. Downstream Tasks

1) Transfer Learning: To test the generality of our architecture and the learned representation, we evaluate GFNet on a set of commonly used transfer learning benchmark datasets including CIFAR-10 [36], CIFAR-100 [36], Stanford Cars [35] and Flowers-102 [52]. We follow the setting of previous works [20], [62], [64], [65], where the model is initialized by the ImageNet pre-trained weights and finetuned on the new datasets. We evaluate the transfer learning performance of our basic transformer-style model (GFNet-XS) and hierarchical model (GFNet-H-B). The results are presented in Table V. The proposed models generally work well on downstream datasets. GFNet models outperform ResMLP models by a large margin and achieve very competitive performance with state-of-the-art EfficientNet-B7. Our models also show competitive performance compared to state-of-the-art CNNs and vision Transformers.

2) Semantic Segmentation: We evaluate our GFNet on ADE20k [81], a challenging semantic segmentation dataset that is commonly used to test vision Transformers. We use the Semantic FPN framework [34] and follow the experiment settings in PVT [69]. We conduct two groups of experiments to evaluate our models. In the first group, we equip our basic hierarchical models (GFNet-H-Ti, S and B) that are built solely by global filters and MLP layers with the lightweight Semantic FPN [34] framework to show the potential of our global filter layer for dense prediction tasks. We train our model for 80 k steps with a batch size of 16 on the training set and report the mIoU on the validation set following common practice [69]. In the second group, we evaluate the scaling ability and compatibility of our model by applying our large model (GFNet-H-L++) to various popular segmentation systems including Semantic FPN [34], UperNet [72] and recent MaskFormer [9]. For semantic FPN, we use the same configuration as our first group of experiments for both Swin [46] and our model. For UperNet and MaskFormer, to fairly compare with state-of-the-art Swin model [46], we adopt similar training configurations in [46] and [9], and only adjust the weight decay and drop path rate to make the training recipe suitable for our model. The results are presented in Table VI. We observe that our GFNet works well on the dense prediction task and can achieve very competitive performance at different levels of complexity.

TABLE VI
SEMANTIC SEGMENTATION RESULTS ON ADE20 K. WE REPORT THE SINGLE-SCALE MIOU ON THE VALIDATION SET. FOR OUR TINY, SMALL AND BASE MODELS, WE EQUIP OUR MODELS WITH THE LIGHT-WEIGHT SEMANTIC FPN [34] FRAMEWORK AND TRAIN THE MODEL FOR 80 K ITERATIONS FOLLOWING [69] TO TEST THE EFFECTIVENESS OF OUR BACKBONE MODEL. FOR OUR LARGE MODEL, WE REPORT THE PERFORMANCE WITH VARIOUS SEGMENTATION FRAMEWORKS INCLUDING SEMANTIC FPN [34], UPERNET [72] AND MASKFORMER [9] TO SHOW THE SCALING ABILITY OF OUR MODEL. THE FLOPS ARE TESTED WITH 1024×1024 INPUT. ‡INDICATES THE BACKBONE IS PRE-TRAINED ON IMAGENET-22 K. OUR MODELS ARE HIGHLIGHTED IN GRAY

Backbone	Method	FLOPs (G)	Params (M)	mIoU (%)
ResNet-18 [25]	Semantic FPN	127	15.5	32.9
PVT-Ti [69]	Semantic FPN	123	17.0	35.7
GFNet-H-Ti	Semantic FPN	126	26.6	41.0
ResNet-50 [25]	Semantic FPN	183	28.5	36.7
PVT-S [69]	Semantic FPN	161	28.2	39.8
Swin-Ti [46]	Semantic FPN	182	31.9	41.5
GFNet-H-S	Semantic FPN	179	47.5	42.5
ResNet-101 [25]	Semantic FPN	260	47.5	38.8
PVT-M [69]	Semantic FPN	219	48.0	41.6
Swin-S [46]	Semantic FPN	274	53.2	45.2
GFNet-H-B	Semantic FPN	261	74.7	44.8
Swin-B [46]	Semantic FPN	422	91.2	46.0
GFNet-H-L++	Semantic FPN	407	114	48.8
Swin-B [46]	UperNet	1188	121	48.1
GFNet-H-L++	UperNet	1168	144	49.5
Swin-B‡ [46]	MaskFormer	448	102	52.7
GFNet-H-L++‡	MaskFormer	443	135	54.3

TABLE VII
OBJECT DETECTION AND INSTANCE SEGMENTATION RESULTS ON COCO. WE COMPARE OUR MODEL WITH THE STATE-OF-THE-ART SWIN TRANSFORMERS [46] USING THE PREVALENT RETINANET AND CASCADE MASK R-CNN FRAMEWORKS. WE REPORT THE MEAN BOUNDING BOX AP AND MASK AP ON THE MINI-VAL SET AS WELL AS THE NUMBER OF PARAMETERS AND FLOPS. OUR MODELS ARE HIGHLIGHTED IN GRAY

Model	FLOPs	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
RetinaNet							
Swin-T [46]	245G	41.5	62.1	44.2	-	-	-
GFNet-H-S++	246G	43.7	65.1	46.5	-	-	-
Swin-S [46]	335G	44.5	65.7	47.5	-	-	-
GFNet-H-B++	326G	44.8	65.9	47.8	-	-	-
Cascade Mask R-CNN							
Swin-T [46]	745G	48.1	67.1	52.2	41.7	64.4	45.0
GFNet-H-S++	744G	49.8	68.7	53.7	42.9	66.2	46.1
Swin-S [46]	838G	50.2	69.6	54.2	43.3	66.6	46.8
GFNet-H-B++	824G	51.6	70.4	56.1	44.7	67.9	48.5

3) Object Detection: We further evaluate our GFNet on object detection using two frameworks: RetinaNet [43] and Cascade Mask-RCNN [4], [24], where we directly use our GFNet-H-S/B++ models as the backbones. Our experiments are conducted on COCO 2017 dataset [44] and the results are from the validation set. For both the two frameworks, we train our model for 12 epochs using AdamW [48] and set the batch size as 16. Our results are summarized in Table VII, where we compare the performance of our GFNet++ with Swin [46] series. Note that the FLOPs are computed on 1280×800 image following common practice [46]. For the RetinaNet, we report the mean bounding box AP. We show our GFNet-H-S/B++ can achieve higher AP^{box} than Swin-T/S with similar or fewer FLOPs. For the Cascade Mask R-CNN, we report the mean bounding box AP and mask AP since both object detection and instance segmentation are performed in this framework. We find our models consistently outperform Swin [46] by 1.2∼1.7 on both AP^{box} and AP^{mask}. These results clearly show that our GFNet can also obtain better performance on object detection and instance segmentation.

TABLE VIII
COMPARISONS AMONG THE GFNET AND OTHER VARIANTS BASED ON THE TRANSFORMER-LIKE ARCHITECTURE ON IMAGENET. WE SHOW THAT GFNET OUTPERFORMS THE RESMLP [64], FNET [40] AND MODELS WITH LOCAL DEPTH-WISE CONVOLUTIONS. WE ALSO REPORT THE NUMBER OF PARAMETERS AND THEORETICAL COMPLEXITY IN FLOPS

Model	Top-1 Acc. (%)	Params (M)	FLOPs (G)
DeiT-S [65]	79.8	22	4.6
Local Conv (3×3)	77.7	15	2.8
Local Conv (5×5)	78.1	15	2.9
Local Conv (7×7)	78.2	15	2.9
ResMLP [64]	76.6	15	3.0
FNet [40]	71.2	15	2.9
GFNet-XS	78.6	16	2.8

C. Analysis and Visualization

Efficiency of GFNet. We demonstrate the efficiency of our GFNet in Fig. 2, where the models are compared in theoretical FLOPs, actual latency and peak memory usage on GPU. We test a single building block of each model (including one token mixing layer and one FFN) with respect to the different numbers of tokens and set the feature dimension and batch size to 384 and 32 respectively. The self-attention model quickly runs out of memory when feature resolution exceeds 56^2, which is also the feature resolution of our hierarchical model. The advantage of the proposed architecture becomes larger as the resolution increases, which strongly shows the potential of our model in vision tasks requiring high-resolution feature maps.

Complexity/Accuracy Trade-Offs. We show the computational complexity and accuracy trade-offs of various Transformer-style architectures in Fig. 3. It is clear that GFNet achieves the best trade-off among all models.

Ablation Study on the Global Filter. To more clearly show the effectiveness of the proposed global filters, we compare GFNet-XS with several baseline models that are equipped with different token mixing operations. The results are presented in Table VIII. All models have a similar building block (token mixing layer + FFN) and the same feature dimension of $D = 384$. We also implement the recent FNet [40] for comparison, where a 1D FFT on feature dimension and a 2D FFT on spatial dimensions are used to mix tokens. As shown in Table VIII, our method outperforms all baseline methods except DeiT-S which has 64% higher FLOPs.

Ablation Study on the Global-Local Filters and Dynamic Weights. To investigate the effects of our improved architectural designs introduced in Section III-D, we compare the baseline GFNet-H-Ti models with several variants that are equipped with the global-local filters and dynamic weights. The results are presented in Table IX. All models have a similar macro architecture based on GFNet-H-Ti. For designs that introduce

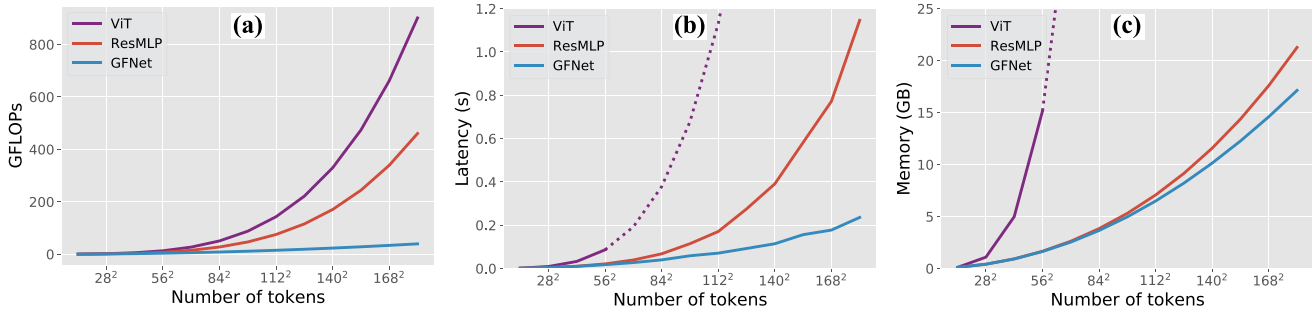


Fig. 2. Comparisons among GFNet, ViT [20] and ResMLP [64] in **(a)** FLOPs **(b)** latency and **(c)** GPU memory with respect to the number of tokens (feature resolution). The dotted lines indicate the estimated values when the GPU memory has run out. The latency and GPU memory is measured using a single NVIDIA RTX 3090 GPU with batch size 32 and feature dimension 384.

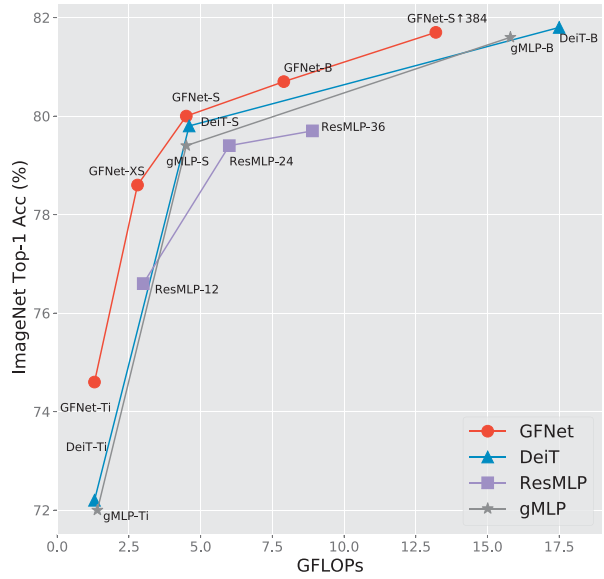


Fig. 3. ImageNet acc. *vs* model complexity of Transformer-style architectures. We compare our GFNet models with typical vision Transformer DeiT [65] and MLP-like models ResMLP [64] and gMLP [45]. All models are based on a similar meta architecture and training strategy.

TABLE IX
Ablation Study on the Global-Local Filters and Dynamic Weights. We Modify the Baseline GFNet-H-Ti Model With Different Strategies to Show the Effects of These Designs. For Designs That Introduce Significant Extra Computation, We Adjust the Number of Blocks in the Third Stage to Fairly Compare Them With the Baseline. We Apply the Global (G), Local (L) and Global-Local (GL) Filters to the 4 Stages ($S^1 - S^4$) and Introduce Dynamic Weights to the Last Two Stages to Test the Effects of These Designs

Model	Global-Local Filters				Dyn. Weights	Top-1 Acc. (%)
	S^1	S^2	S^3	S^4		
GFNet-H-Ti	G	G	G	G	-	80.1
A	L	G	G	G	-	80.3
B	G	L	G	G	-	80.4
C	G	G	L	G	-	79.7
D	G	G	G	L	-	79.8
E	L	L	G	G	-	80.5
GFNet-H-Ti+	L	L	GL	GL	-	80.8
F	G	G	G	G	φ_{gen}	80.1
G	G	G	G	G	φ_{sel}	80.2
H	G	G	G	G	φ_{gate}	80.6
GFNet-H-Ti++	L	L	GL	GL	φ_{gate}	81.2

TABLE X
Directly Adapting to Other Resolutions. We Report the Top-1 Accuracy (%) on the ImageNet Validation Set for GFNet-S and DeiT-S When Directly Adapting to Different Resolutions (From 96^2 to 448^2) Without Fine-Tuning the Model

Input Res.	96^2	128^2	192^2	224^2	256^2	384^2	448^2
DeiT-S	17.5	67.0	78.6	79.8	80.2	78.1	75.8
GFNet-S	52.1	69.9	78.3	80.1	80.7	79.8	78.7
Δ	+34.6	+2.9	-0.3	+0.3	+0.5	+0.7	+2.9

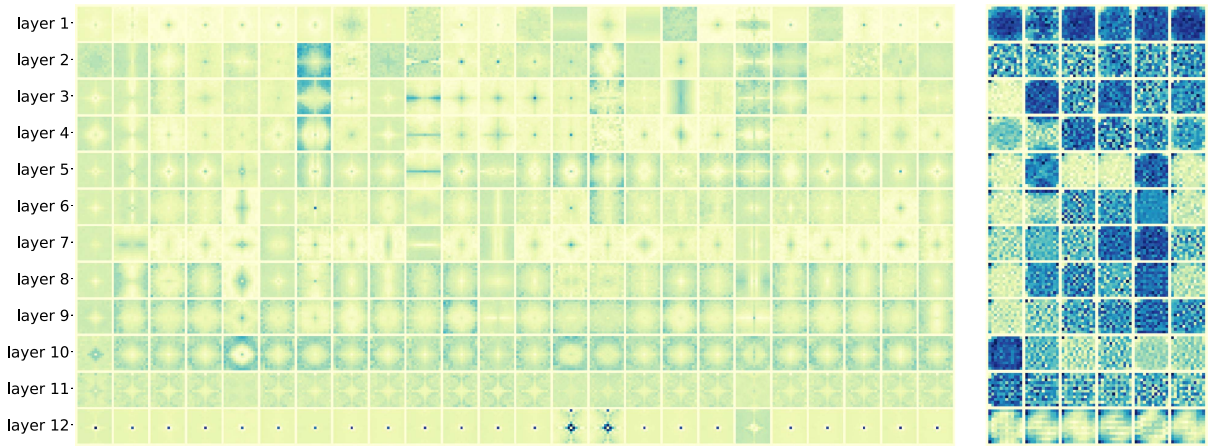
significant extra computation like the proposed dynamic weight generation method, we adjust the number of blocks in the third stage to fairly compare them with the baseline. We first replace the global filters in each stage of GFNet-H-Ti with local convolutions to search for the best configuration of global and local filters (model A-D). The results show that local convolutions in early stages (stage 1 and 2) are helpful while global filters in late stages (stage 3 and 4) play critical roles. Based on this observation, we replace global filters in all early stages with local convolutions (model E) and obtain a +0.4% improvement in accuracy. By further enhancing the global filter with the newly proposed global-local filter (GFNet-H-Ti+), we see a +0.7 boost over the baseline model. We then study the effects of dynamic weights by adding them to the last two stages since adding them to the early stages will largely reduce the throughput but bring no significant improvement. Although both three strategies can improve the performance, we find that the weight generation and selection methods (model F and G) bring limited improvements when we normalize the model sizes to perform a fair comparison. The channel gating method (model H) exhibits a significant improvement (+0.5%) with negligible extra computation. We combine both the global-local layer and dynamic weights to form our final model and obtain a +1.1% overall improvement.

Directly Adapting to Other Resolutions. As is discussed in Section III-B, one of the advantages of GFNet is the ability to deal with arbitrary resolutions. To verify this, we *directly* evaluate GFNet-S trained with 224×224 images on different resolutions (from 128 to 448) without fine-tuning the models. We show the accuracy of GFNet-S and DeiT-S in Table X. It can be observed that our GFNet can adapt to different resolutions with generally less performance drop than DeiT-S. GFNet shows

TABLE XI

Evaluation of Robustness and Generalization Ability. We Measure the Robustness From Different Aspects, Including the Adversarial Robustness by Adopting Adversarial Attack Algorithms Including FGSM and PGD and the Performance on Corrupted/Out-of-Distribution Datasets Including ImageNet-A [27] (Top-1 Accuracy) and ImageNet-C [26] (mCE, Lower is Better). The Generalization Ability is Evaluated on ImageNet-V2 [58] and ImageNet-Real [3]

Model	FLOPs	Params	ImageNet		Generalization		Robustness			
	(G)	(M)	Top-1↑	Top-5↑	IN-V2↑	IN-Real↑	FGSM↑	PGD↑	IN-C↓	IN-A↑
ResNet-50 [25]	4.1	26	76.1	92.9	67.4	85.8	12.2	0.9	76.7	0.0
ResNeXt50-32x4d [73]	4.3	25	79.8	94.6	68.2	85.2	34.7	13.5	64.7	10.7
DeiT-S [65]	4.6	22	79.8	95.0	68.4	85.6	40.7	16.7	54.6	18.9
ResMLP-12 [64]	3.0	15	76.6	93.2	64.4	83.3	23.9	8.5	66.0	7.1
GFNet-S	4.5	25	80.1	94.9	68.5	85.8	42.6	21.0	53.8	14.3



(a) Frequency domain (b) Spatial domain

Fig. 4. Visualization of the learned *global filters* in *GFNet-XS*. We visualize the original frequency domain global filters in (a) and show the corresponding spatial domain filters for the first 6 columns in (b). There are more clear patterns in the frequency domain than the spatial domain.

more significant advantages when the gap between training and inference test resolutions increases, which suggests our model is more robust to the resolution change during inference.

Robustness & Generalization Ability. Inspired by the [51], we further conduct experiments to evaluate the robustness and the generalization ability of the GFNet. For robustness, we consider ImageNet-A, ImageNet-C, FGSM and PGD. ImageNet-A [27] (IN-A) is a challenging dataset that contains natural adversarial examples. ImageNet-C [26] (IN-C) is used to validate the robustness of the model under various types of corruption. FGSM [22] and PGD [50] are two widely used algorithms that are targeted to evaluate the adversarial robustness of the model by the single-step attack and multi-step attack, respectively. For generalization ability, we adopt two variants of ImageNet validation set: ImageNet-V2 [58] (IN-V2) and ImageNet-Real [3] (IN-Real). ImageNet-V2 is a re-collected version of the ImageNet validation set following the same data collection procedure of ImageNet, while ImageNet-Real contains the same images as the ImageNet validation set but has reassessed labels. We compare GFNet-S with various baselines in Table XI including CNNs, Transformers and MLP-like architectures, and find the GFNet enjoys both favorable robustness and generalization ability.

Visualization. The core operation in GFNet is the element-wise multiplication between frequency-domain features and the global filter. Therefore, it is easy to visualize and interpret. We visualize the frequency domain filters as well as their corresponding spatial domain filters in Fig. 4. The learned global filters have more clear patterns in the frequency domain, where different layers have different characteristics. Interestingly, the filters in the last layer particularly focus on the low-frequency component. The corresponding filters in the spatial domain are less interpretable for humans.

V. Conclusion

We have presented the Global Filter Network (*GFNet*), which is a conceptually simple yet computationally efficient architecture for visual recognition. Our model replaces the self-attention sub-layer in the vision Transformer with 2D FFT/IFFT and a set of learnable *global filters* in the frequency domain. Benefiting from the token mixing operation with log-linear complexity, our architecture is highly efficient. The proposed basic operation is easy-to-use, scalable, and compatible with different macro architectures and micro designs. Our experimental results demonstrated that GFNet can be a very competitive alternative to vision Transformers and CNNs in terms of efficiency, generalization ability and robustness.

REFERENCES

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[2] G. A. Baxes, *Digital Image Processing: Principles and Applications*. Hoboken, NJ, USA: Wiley, 1994.

[3] L. Beyer, O. J. Hénaff, A. Kolesnikov, X. Zhai, and A. van den Oord, "Are we done with ImageNet?" 2020, *arXiv:2006.07159*.

[4] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6154–6162.

[5] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 213–229.

[6] S. Chen, E. Xie, C. Ge, D. Liang, and P. Luo, "CycleMLP: A MLP-like architecture for dense prediction," 2021, *arXiv:2107.10224*.

[7] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8122–8131.

[8] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11030–11039.

[9] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 17864–17875.

[10] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.

[11] K. Choromanski et al., "Rethinking attention with performers," 2020, *arXiv:2009.14794*.

[12] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, no. 90, pp. 297–301, 1965.

[13] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "CoAtNet: Marrying convolution and attention for all data sizes," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 3965–3977.

[14] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, "FlashAttention: Fast and memory-efficient exact attention with IO-awareness," 2022, *arXiv:2205.14135*.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[17] C. Ding et al., "CirCNN: Accelerating and compressing deep neural networks using block-circulant weight matrices," in *Proc. 50th Annu. IEEE/ACM Int. Symp. Microarchitect.*, 2017, pp. 395–408.

[18] X. Ding, X. Zhang, Y. Zhou, J. Han, G. Ding, and J. Sun, "Scaling up your kernels to 31x31: Revisiting large kernel design in CNNs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11953–11965.

[19] X. Dong et al., "CSWin transformer: A general vision transformer backbone with cross-shaped windows," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12124–12134.

[20] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[21] C. Gong, D. Wang, M. Li, V. Chandra, and Q. Liu, "Improve vision transformers training by suppressing over-smoothing," 2021, *arXiv:2104.12753*.

[22] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[23] Q. Han et al., "On the connection between local attention and dynamic depth-wise convolution," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–14.

[24] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[26] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," 2019, *arXiv:1903.12261*.

[27] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15262–15271.

[28] E. Hoffer, T. Ben-Nun, I. Hubara, N. Giladi, T. Hoefler, and D. Soudry, "Augment your batch: Improving generalization through instance repetition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8129–8138.

[29] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[30] W. Hua, Y. Zhou, C. M. De Sa, Z. Zhang, and G. E. Suh, "Channel gating neural networks," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, Art. no. 169.

[31] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 646–661.

[32] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 667–675.

[33] Z. Jiang et al., "Token labeling: Training a 85.5% top-1 accuracy vision transformer with 56m parameters on ImageNet," 2021, *arXiv:2104.10858*.

[34] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6399–6408.

[35] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2013, pp. 554–561.

[36] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, Ontario, Canada, Tech. Rep., 2009.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[38] Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.

[39] J.-H. Lee, M. Heo, K.-R. Kim, and C.-S. Kim, "Single-image depth estimation based on Fourier domain analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 330–339.

[40] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon, "FNet: Mixing tokens with Fourier transforms," 2021, *arXiv:2105.03824*.

[41] S. Li et al., "Falcon: A Fourier transform based approach for fast and secure convolutional neural network predictions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8705–8714.

[42] Y. Li et al., "MViTv2: Improved multiscale vision transformers for classification and detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4804–4814.

[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[44] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2014, pp. 740–755.

[45] H. Liu, Z. Dai, D. R. So, and Q. V. Le, "Pay attention to MLPs," 2021, *arXiv:2105.08050*.

[46] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," 2021, *arXiv:2103.14030*.

[47] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11966–11976.

[48] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.

[49] J. Lu et al., "SOFT: Softmax-free transformer with linear complexity," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 21297–21309.

[50] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.

[51] X. Mao et al., "Rethinking the design principles of robust vision transformer," 2021, *arXiv:2105.07926*.

[52] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. 6th Indian Conf. Comput. Vis. Graph. Image Process.*, 2008, pp. 722–729.

[53] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, Art. no. 721.

[54] I. Pitas, *Digital Image Processing Algorithms and Applications*. Hoboken, NJ, USA: Wiley, 2000.

[55] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM J. Control Optim.*, vol. 30, no. 4, pp. 838–855, 1992.

[56] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10425–10433.

[57] Y. Rao, W. Zhao, Z. Zhu, J. Lu, and J. Zhou, "Global filter networks for image classification," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 980–993.
[58] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet classifiers generalize to ImageNet?," in *Proc. 36th Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 5389–5400.
[59] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "ImageNet-21k pretraining for the masses," 2021, *arXiv:2104.10972*.
[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[61] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[62] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
[63] I. Tolstikhin et al., "MLP-mixer: An all-MLP architecture for vision," 2021, *arXiv:2105.01601*.
[64] H. Touvron et al., "ResMLP: Feedforward networks for image classification with data-efficient training," 2021, *arXiv:2105.03404*.
[65] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," 2020, *arXiv:2012.12877*.
[66] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," 2021, *arXiv:2103.17239*.
[67] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[68] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," 2020, *arXiv:2006.04768*.
[69] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 548–558.
[70] W. Wang et al., "PVT v2: Improved baselines with pyramid vision transformer," *Comput. Vis. Media*, vol. 8, no. 3, pp. 415–424, 2022.
[71] H. Wu et al., "CVT: Introducing convolutions to vision transformers," 2021, *arXiv:2103.15808*.
[72] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 418–434.
[73] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1492–1500.
[74] Y. Yang and S. Soatto, "FDA: Fourier domain adaptation for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4085–4095.
[75] T. Yu, X. Li, Y. Cai, M. Sun, and P. Li, "S2-MLP: Spatial-shift MLP architecture for vision," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 297–306.
[76] W. Yu et al., "Metaformer is actually what you need for vision," 2021, *arXiv:2111.11418*.
[77] L. Yuan et al., "Tokens-to-token ViT: Training vision transformers from scratch on ImageNet," 2021, *arXiv:2101.11986*.
[78] D. J. Zhang et al., "MorphMLP: A self-attention free, MLP-like backbone for image and video," 2021, *arXiv:2111.12527*.
[79] S. Zheng et al., "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," 2020, *arXiv:2012.15840*.
[80] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 13001–13008.
[81] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 633–641.

Yongming Rao (Student Member, IEEE) received the BEng degree from the Department of Electronic Engineering, Tsinghua University, China, in 2018. Currently, he is working toward the PhD degree with the Department of Automation, Tsinghua University, advised by Prof. Jiwen Lu. His research interests include computer vision and deep learning. He has authored more than 20 conference papers in CVPR/ICCV/ECCV/NeurIPS and 4 journal papers in *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *International Journal of Computer Vision*. He serves as a reviewer for several international conferences and journals, where he was recognized as the outstanding reviewer of ECCV 2020, CVPR 2021 and ICME 2019. He is a recipient of the CCF-CV Academic Emerging Award, in 2019.

Wenliang Zhao (Student Member, IEEE) is currently working toward the PhD degree with the Department of Automation, Tsinghua University. His research interests include computer vision and deep learning. He has authored 5 conference papers in ICCV/NeurIPS/CVPR. He serves as a reviewer for several international conferences and journals including CVPR/ECCV/*IEEE Transactions on Image Processing*. He received the National Scholarship of China, and Qualcomm Scholarship of Tsinghua University.

Zheng Zhu (Member, IEEE) received the PhD degree from the Institute of Automation, Chinese Academy of Sciences, in 2019. He is currently a postdoctoral fellow with Tsinghua University. He served reviewers in various journals and conferences including *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Multimedia*, *IEEE Transactions on Circuits and Systems for Video Technology*, CVPR, ICCV, ECCV, ICLR. He has co-authored more than 40 journal and conference papers mainly on computer vision and robotics problems, such as face recognition, visual tracking, human pose estimation, and servo control. He has more than 3,000 Google Scholar citations to his work. He organized the Masked Face Recognition Challenge and Workshop in ICCV 2021. He ranked the 1st on NIST-FRVT Masked Face Recognition, won the COCO Keypoint Detection Challenge in ECCV 2020 and Visual Object Tracking (VOT) Real-Time Challenge in ECCV 2018.

Jie Zhou (Senior Member, IEEE) received the BS and MS degrees both from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the PhD degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), Wuhan, China, in 1995. From then to 1997, he served as a postdoctoral fellow with the Department of Automation, Tsinghua University, Beijing, China. Since 2003, he has been a full professor with the Department of Automation, Tsinghua University. His research interests include computer vision, pattern recognition, and image processing. In recent years, he has authored more than 300 papers in peer-reviewed journals and conferences. Among them, more than 100 papers have been published in top journals and conferences such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, and CVPR. He is an associate editor for *IEEE Transactions on Pattern Analysis and Machine Intelligence* and two other journals. He received the National Outstanding Youth Foundation of China Award. He is an IAPR fellow.

Jiwen Lu (Senior Member, IEEE) received the BEng degree in mechanical engineering and the MEng degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, in 2003 and 2006, respectively, and the PhD degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. He is currently an associate professor with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include computer vision and pattern recognition, where he has authored/co-authored more than 110 scientific papers in *IEEE Transactions on Pattern Analysis and Machine Intelligence/International Journal of Computer Vision*/CVPR/ICCV/ECCV. He serves as the co-editor-of-chief for *Pattern Recognition Letters*, an associate editor for *IEEE Transactions on Image Processing*, *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Biometrics, Behavior, and Identity Science*, and *Pattern Recognition*. He also serves as the general co-chair of ICME'2022, and the program co-chair of FG'2023, VCIP'2022, AVSS'2021 and ICME'2020. He received the National Outstanding Youth Foundation of China Award. He is an IAPR fellow.