Global Filter Networks for Image Classification

Anonymous Author(s)
Affiliation
Address
email

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, 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. We exhibit favorable accuracy/complexity trade-offs of our models on ImageNet, which achieve competitive performance with vision transformers while maintaining the high efficiency of MLP models.

1 Introduction

The transformer architecture, originally designed for the natural language processing (NLP) tasks [33], has shown promising performance on various vision problems recently [7, 31, 21, 40, 3]. 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 [29, 30] 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 [7] and MLP-Mixer [29]). 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 [21]. 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, which may hurt the generality of this kind of models.

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

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

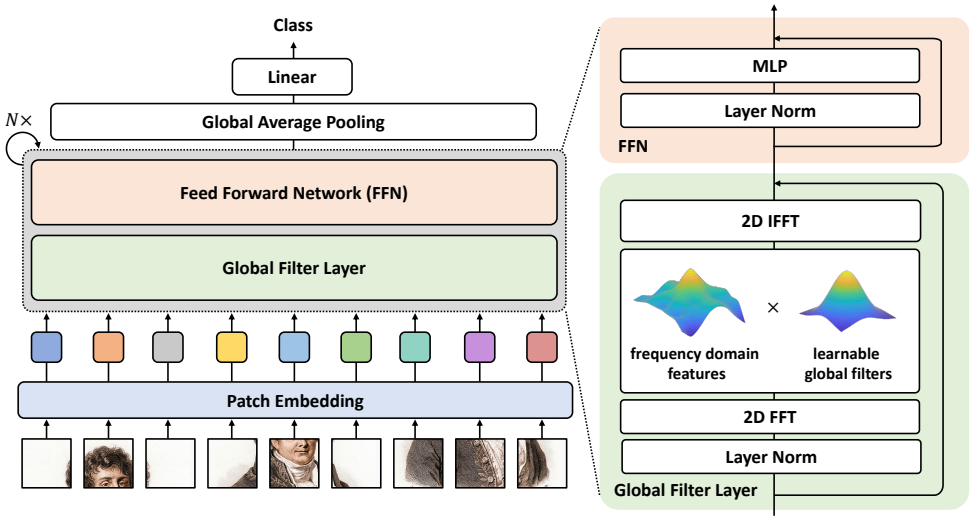


Figure 1: **The overall architecture of the Global Filter Network**. Our architecture is based on Vision Transformer (ViT) models with some minimal modifications. We replace the self-attention sub-layer with the proposed *global filter layer*, which consists of three key operations: a 2D discrete Fourier 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.

	Complexity (FLOPs)	# Parameters
Depthwise Convolution	k^2HWD	k^2D
Self-Attention	$4HWD^2 + 2H^2W^2D$	$4D^2$
Spatial MLP	H^2W^2D	H^2W^2
Global Filter	$HWD\lceil\log_2(HW)\rceil + HWD$	HWD

Table 1: 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 convolution operation. The proposed global filter is much more efficient than self-attention and spatial 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 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) [4]. 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. The overall architecture of GFNet is illustrated in Figure 1. We also compare the proposed global filter with prevalent operations in deep vision models in Table 1.

Our experiments on ImageNet verify the effectiveness of GFNet. Our method achieves competitive performance with DeiT [31] while maintaining the high efficiency of MLP-like models [30]. With a similar architecture, our model outperform the recent MLP models including ResMLP [30] and gMLP [20]. When using the hierarchical architecture, GFNet can further enlarge the gap. GFNet also works well on downstream transfer learning datasets. Our results demonstrate that GFNet can be a very competitive alternative to vision transformers and MLP-like models in accuracy/complexity trade-offs.

2 Related works

Vision transformers. Since Dosovitskiy *et al*. [7] introduce transformers to the image classification and achieve a competitive performance compared to CNNs, transformers begin to exhibit their potential in various vision tasks [2, 3, 40]. Recently, there are a large number of works which aim to improve the transformers [31, 32, 21, 34, 13, 8, 36]. These works either seek for better training strategies [31, 8] or design better architectures [21, 34, 36] or both [32, 8]. However, most of the architecture modification of the transformers [34, 13, 21, 36] introduces additional inductive biases similar to CNNs. In this work, we only focus on the standard transformer architecture [7, 31] and our goal is to replace the heavy self-attention layer ($\mathcal{O}(L^2)$) to an more efficient operation which can still model the interactions among different spatial locations without introducing the inductive biases associated with CNNs.

MLP-like models. More recently, there are several works that question the importance of self-attention in the vision transformers and propose to use MLP to replace the self-attention layer in the transformers [29, 30, 20]. The MLP-Mixer [29] employs MLPs to perform token mixing and channel mixing alternatively in each block. ResMLP [30] adopts a similar idea but substitutes the Layer Normalization with an Affine transformation for acceleration. The recently proposed gMLP [20] 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.

Applications of Fourier transform in vision. Fourier transform has been an important tool in digital image processing for decades [25, 1]. With the breakthroughs of CNNs in vision [10, 9], there are a variety of works that start to incorporate Fourier transform in some deep learning method [19, 35, 6, 17]. 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 [17, 35], while others utilize the convolution theorem to accelerate the CNNs via fast Fourier transform (FFT) [19, 6]. 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 [25]. We also take advantage of some properties of FFT to reduce the computational costs and the number of parameters.

3 Method

3.1 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:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)kn} := \sum_{n=0}^{N-1} x[n] W_N^{kn} \tag{3.1}$$

where j is the imaginary unit and $W_N = e^{-j(2\pi/N)}$. The formulation of DFT in Equation (3.1) can be derived from the Fourier transform for continuous signal by sampling in both the time domain and the frequency domain (see Appendix A for details). Since $X[k]$ repeats on intervals of length N, it is suffice 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):

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j(2\pi/N)kn}. \tag{3.2}$$

Algorithm 1 Pseudocode of Global Filter Layer.

```
# x: the token features, B x H x W x D (where N = H * W)
# K: the filter kernel, H x W_hat x D (where W_hat = W // 2 + 1, see Section 3.2 for details)

X = rfft2(x, dim=(1, 2))
X_tilde = X * K
x = x + irfft2(X_tilde, dim=(1, 2))
```

`rfft2/irfft2`: 2D FFT/IFFT for real signal

For real input $x[n]$, it can be proved that (see Appendix A) 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 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 (3.2), 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:

$$X[u, v] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x[m, n] e^{-j2\pi\left(\frac{um}{M} + \frac{vn}{N}\right)}. \tag{3.3}$$

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.

3.2 Global Filter Networks

Overall architecture. Recent advances in vision transformers [7, 31] 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 [30, 29] 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 architecture of our model is depicted in Figure 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 [7, 31]. The output tokens of the last block are fed into a global average pooling layer followed by a linear classifier.

Global filter layer. We propose 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 3.1) along the spatial dimensions to convert $\boldsymbol{x}$ to the frequency domain:

$$\boldsymbol{X} = \mathcal{F}[\boldsymbol{x}] \in \mathbb{C}^{H \times W \times D}, \tag{3.4}$$

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 $\boldsymbol{X}$:

$$\tilde{\boldsymbol{X}} = \boldsymbol{K} \odot \boldsymbol{X}, \tag{3.5}$$

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 with $\boldsymbol{X}$, 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 via a residual connection [10],

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \mathcal{F}^{-1}[\tilde{\boldsymbol{X}}]. \tag{3.6}$$

The formulation of the global filter layer is motivated by the frequency filters in the digital image processing [25], 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 Appendix A) that the global filter layer is equivalent to a depthwise *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 relation in the frequency domain instead of spatial domain (see Figure 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 depthwise 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:

$$\boldsymbol{X}_r = \boldsymbol{X}[:, 0 : \widehat{W}] := \mathcal{F}_r[\boldsymbol{x}], \quad , \widehat{W} = \lceil W/2 \rceil, \tag{3.7}$$

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 [24]), as is shown in Algorithm 1. The FFT and ITTF 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.

Relationship to vision transformers and pure MLP models. The GFNet follows the line of research about the exploration of approaches to mix the tokens. Compared to existing architectures like vision transformers and pure MLP models, we exhibit that GFNet has several favorable properties: 1) GFNet is more efficient. The complexity of both the vision transformers [7, 31, 32] and the MLP models [29, 30] is $\mathcal{O}(L^2)$. Different from them, global filter layer only consists an FFT ($\mathcal{O}(L \log L)$), an element-wise multiplication ($\mathcal{O}(L)$) and an IFFT ($\mathcal{O}(L)$), which means the total computational complexity is $\mathcal{O}(L \log L)$. 2) 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 length. 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.

We also notice recently a concurrent work FNet [18] 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 implement the FNet and show that our model outperforms it.

Architecture variants. Due to the limitation from the quadratic complexity in the self-attention, vision transformers [7, 31] are usually designed to process a relatively small feature map (*e.g.*, 14×14). However, our GFNet, which enjoys log-linear complexity, avoids that problem. Since in our GFNet the computational costs do not grow significantly when the feature map size increases, we can adopt a hierarchical architecture inspired by the success of CNNs [16, 10]. Generally speaking,

we can start from a large feature map (*e.g.*, 56×56) and gradually perform downsampling after a few blocks. In this paper, we mainly investigate three variants of GFNet: GFNet-12 that has a similar architecture to DeiT-S [31] and ResMLP-12 [30], GFNet-H18 and GFNet-H24 that are two hierarchical architectures with higher efficiency and performance respectively. The detailed architecture can be found in Appendix B.

Training techniques. In our experiments, we find the default training procedure introduced for DeiT [31] can already yield good performance for our GFNet. However, as is analyzed in [8], transformers-like architectures may suffer from over-smoothing due to the global receptive field in every layer. Inspired by [38], we employ the token-labeling [13] technique to provide each token a supervision signal. For the sake of simplicity, we implement it by using a well-trained model (LV-ViT [13]) as the teacher model and perform token-wise distillation (see Appendix B). Experiments show that this can overcome over-smoothing and further enhance the performance of GFNet.

4 Experiments

We conduct extensive experiments to verify the effectiveness of our GFNet architecture on the image classification task. We present the main results on ImageNet [5] and compare them with various architectures. We also test our models on the downstream transfer learning datasets including CIFAR-10/100 [15], Stanford Cars [14] and Flowers-102 [23]. Finally, we investigate the effects of the proposed global filters and provide visualization to have an intuitive understanding of our method.

4.1 ImageNet results

Setups. We conduct our main experiments on ImageNet [5], which is a widely used large-scale benchmark for image classification. ImageNet contains roughly 1.2M images from 1,000 categories. Following common practice [10, 31], 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 [31, 30], we follow the most training details for our models. We train our models for 300 epochs using the AdamW optimizer [22]. We set the initial learning rate as $\frac{\text{batch size}}{1024} \times 0.001$ and decay the learning rate to $1e^{-5}$ using the cosine schedule. We use a linear warm-up learning rate in the first 5 epochs and apply gradient clipping to stabilize the training process. For GFNet-12, we faithfully follow the regularization strategies used in DeiT [31] including Mixup [39], CutMix [37], EMA model [26], RandomEarse [41] and repeated augmentation [11]. For GFNet-H18 and GFNet-24, we follow most settings of [31] except for the EMA model and repeated augmentation as suggested in [21]. We set the stochastic depth coefficient [12] to 0, 0.1 and 0.3 for GFNet-12, GFNet-H18 and GFNet-H24 respectively. During finetuning at the higher resolution, we use the hyper-parameters suggested by the implementation of [31] and train the model for 30 epochs. All of our models are trained on a single machine with 8 GPUs.

Main results. The main results are presented in Table 2. We compare our method with different architectures for image classification including convolutional networks, vision transformers, and MLP-like models that have similar complexity and number of parameters. One can find that our method can clearly outperform recent MLP-like models like ResMLP [30] and gMLP [20]. Specifically, GFNet-12 outperforms ResMLP-12 by 1.9% while having slightly fewer FLOPs. GFNet-12 also achieves competitive results compared to gMLP-S and has 1.7 GFLOPs less computation. Our hierarchical models even have more significant advantages. Benefiting from the log-linear complexity, GFNet-H18 is 2.6% more accurate than ResMLP-12 while having 36% fewer FLOPs. These results show the high efficiency of the global filter.

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 finetuning the model trained at 224×224 resolution to higher resolution following the practice in vision transformers [31]. As shown in Table 2, our model can easily adapt to higher resolution with only 30 epoch finetuning and achieve better performance.

Token-wise distillation. We address the over-smoothing problem by introducing token-wise distillation supervision. This strategy improves the performance of our models by around 1% without

Table 2: **Main results on ImageNet.** We compare different architectures for image classification including convolutional networks, vision transformers, MLP-like models and our method that have comparable FLOPs and 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 "↑384" to represent models finetuned on 384×384 images for 30 epochs. † indicates the models trained with token-wise distillation.

	Model	Params (M)	FLOPs (G)	Res.	Top-1 Acc. (%)	Top-5 Acc. (%)
ConvNets	EfficientNet-B3 [28]	12	1.8	300	81.6	95.7
	EfficientNet-B4 [28]	19	4.2	380	82.9	96.4
	EfficientNet-B5 [28]	30	9.9	456	83.6	96.7
	RegNetY-4GF [27]	21	4.0	224	80.0	-
	RegNetY-8GF [27]	39	8.0	224	81.7	-
	RegNetY-16GF [27]	84	16.0	224	82.9	-
Transformers	ViT-B/16 [7]	86	55.4	384	77.9	-
	ViT-L/16 [7]	307	190.7	384	76.5	-
	DeiT-Ti [31]	5	1.2	224	72.2	91.1
	DeiT-S [31]	22	4.6	224	79.8	95.0
	DeiT-B [31]	86	17.5	224	81.8	95.6
MLP-like Models	Mixer-B/16	59	12.7	224	76.4	-
	ResMLP-12 [30]	15	3.0	224	76.6	-
	ResMLP-24 [30]	30	6.0	224	79.4	-
	ResMLP-36 [30]	45	8.9	224	79.7	-
	gMLP-Ti [20]	6	1.4	224	72.0	-
	gMLP-S [20]	20	4.5	224	79.4	-
	gMLP-B [20]	73	15.8	224	81.6	-
Ours	GFNet-12	16	2.8	224	78.5	94.0
	GFNet-H18	17	1.9	224	79.2	94.5
	GFNet-H24	37	5.6	224	81.0	95.2
	GFNet-12↑384	18	8.4	384	80.5	95.1
	GFNet-H18↑384	20	5.5	384	80.8	95.3
	GFNet-H24↑384	42	16.5	384	82.0	95.6
	GFNet-12†	16	2.9	224	79.7	94.7
	GFNet-H18†	17	1.9	224	79.9	94.9
	GFNet-H24†	37	5.7	224	82.0	95.8
	GFNet-12↑384†	18	8.6	384	81.5	95.7
	GFNet-H18↑384†	20	5.6	384	81.5	95.7
	GFNet-H24↑384†	43	16.6	384	83.1	96.4

bringing significantly large computation during inference. By incorporating both token-wise distillation and larger image resolution, the performance can be further improved.

4.2 Transfer learning results

Setups. 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 [15], CIFAR-100 [15], Stanford Cars [14] and Flowers-102 [23]. We follow the setting of previous works [28, 7, 31, 30], where the model is initialized by the ImageNet pre-trained weights and finetuned on the new datasets. During finetuning, we use the SGD optimizer and set the weight decay to $1e^{-4}$. We use batch size 512 and a smaller initial learning rate of 0.0001 with cosine decay. Linear learning rate warm-up in the first 5 epochs and gradient clipping with a max norm of 1 are also applied to stabilize the training. We keep most of the regularization methods unchanged except for removing Random Erasing and stochastic depth following [31].

Table 3: **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	CIFAR-10	CIFAR-100	Flowers-102	Cars-196
ResNet50 [10]	4.1G	26M	-	-	96.2	90.0
EfficientNet-B7 [28]	37G	66M	98.9	91.7	98.8	94.7
ViT-B/16 [7]	55.4G	86M	98.1	87.1	89.5	-
ViT-L/16 [7]	190.7G	307M	97.9	86.4	89.7	-
Deit-B/16 [31]	17.5G	86M	99.1	90.8	98.4	92.1
ResMLP-12 [30]	3.0G	15M	98.1	87.0	97.4	84.6
ResMLP-24 [30]	6.0G	30M	98.7	89.5	97.9	89.5
GFNet-12	2.8G	16M	98.6	89.0	98.1	92.4
GFNet-H24	5.6G	37M	98.8	89.3	98.5	93.1

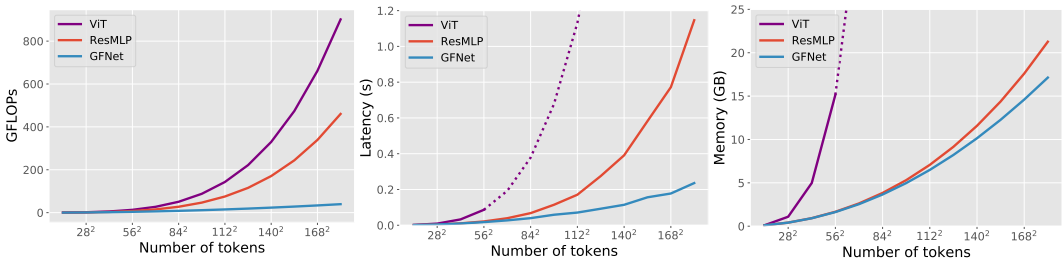


Figure 2: Comparisons among GFNet, ViT [7] and ResMLP [30] in **(a)** FLOPs **(b)** latency and **(c)** GPU memory with respect to the number of tokens (feature resolution). The dotted lines indicate the estimated values when the GPU memory has run out. The latency and GPU memory is measured using a single NVIDIA RTX 3090 GPU with batch size 32 and feature dimension 384.

Results. We evaluate the transfer learning performance of our basic model and best model. The results are presented in Table 3. The proposed models generally work well on downstream datasets. GFNet-H24 outperforms ResMLP-24 on three out of four datasets and achieves very close accuracy on the left one dataset. Our models also show competitive performance compared to state-of-the-art CNNs and vision transformers.

4.3 Analysis and visualization

Efficiency of GFNet. We demonstrate the efficiency of our GFNet in Figure 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 complexity and accuracy trade-offs of various architectures in Figure 3. GFNet achieves the best trade-off among kinds of models.

Ablation study on the global filter. To more clearly show the effectiveness of the proposed global filters, we compare GFNet-12 with several baseline models that are equipped with different token mixing operations. The results are presented in Table 4. 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 [18] 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 4, our method outperforms all baseline methods except DeiT-S that has 64% higher FLOPs.

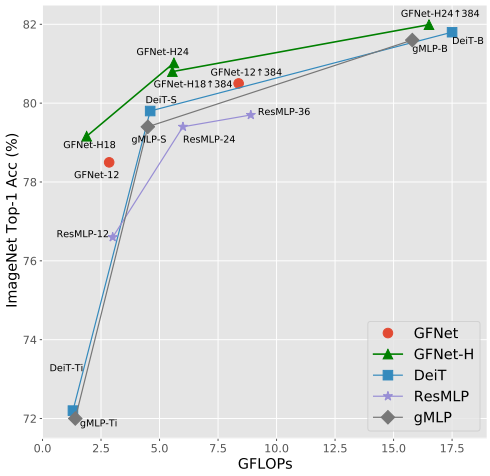
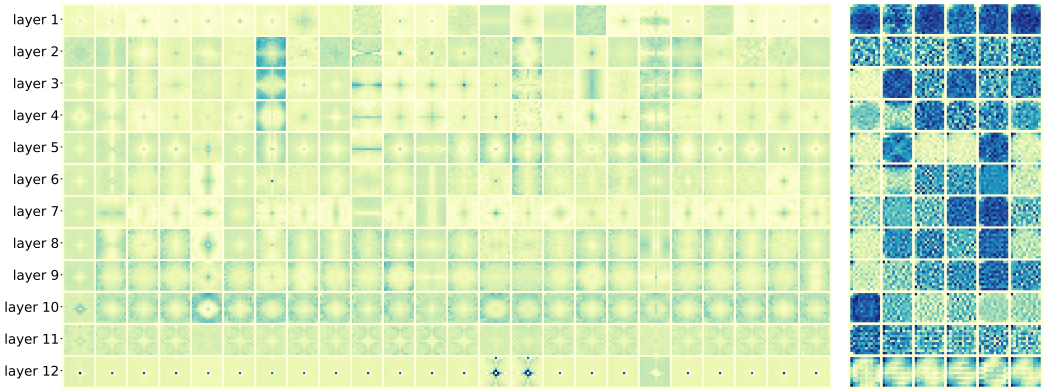


Figure 3: ImageNet acc. *vs* model complexity.

Table 4: Comparisons among the GFNet and other variants based on the transformer-like architecture on ImageNet. We show that GFNet outperforms the ResMLP [30], FNet [18] and models with local depth-wise convolutions. We also report the number of parameters and theoretical complexity in FLOPs.

Model	Acc (%)	Param (M)	FLOPs (G)
DeiT-S [31]	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 [30]	76.6	15	3.0
FNet [18]	71.2	15	2.9
GFNet-12	78.5	16	2.8



(a) Frequency domain (b) Spatial domain

Figure 4: Visualization of the learned *global filters* in GFNet-12. 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 spatial domain.

Visualization. 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 Figure 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.

5 Conclusion

We have presented the Global Filter Network (*GFNet*), which is a conceptually simple yet computationally efficient architecture for image classification. Our model replaces the self-attention sub-layer in 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. Our experimental results demonstrated that GFNet can be a very competitive alternative to vision transformers and MLP-like models in accuracy/complexity trade-offs. Since we only focus on vision tasks in this paper, extending our method to broader applications like NLP and speech understanding would be interesting future directions.

References

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

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 3

[3] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, 2021. 1, 3

[4] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965. 2

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 6

[6] Caiwen Ding, Siyu Liao, Yanzhi Wang, Zhe Li, Ning Liu, Youwei Zhuo, Chao Wang, Xuehai Qian, Yu Bai, Geng Yuan, et al. Circnn: accelerating and compressing deep neural networks using block-circulant weight matrices. In *MICRO*, pages 395–408, 2017. 3

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 3, 4, 5, 7, 8

[8] Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Improve vision transformers training by suppressing over-smoothing. *arXiv preprint arXiv:2104.12753*, 2021. 3, 6

[9] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3, 4, 5, 6, 8

[11] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *CVPR*, pages 8129–8138, 2020. 6

[12] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, pages 646–661, 2016. 6

[13] Zihang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Xiaojie Jin, Anran Wang, and Jiashi Feng. Token labeling: Training a 85.5% top-1 accuracy vision transformer with 56m parameters on imagenet. *arXiv preprint arXiv:2104.10858*, 2021. 3, 6

[14] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, pages 554–561, 2013. 6, 7

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6, 7

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25:1097–1105, 2012. 5

[17] Jae-Han Lee, Minhyeok Heo, Kyung-Rae Kim, and Chang-Su Kim. Single-image depth estimation based on fourier domain analysis. In *CVPR*, pages 330–339, 2018. 3

[18] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021. 5, 8, 9

[19] Shaohua Li, Kaiping Xue, Bin Zhu, Chenkai Ding, Xindi Gao, David Wei, and Tao Wan. Falcon: A fourier transform based approach for fast and secure convolutional neural network predictions. In *CVPR*, pages 8705–8714, 2020. 3

[20] Hanxiao Liu, Zihang Dai, David R So, and Quoc V Le. Pay attention to mlps. *arXiv preprint arXiv:2105.08050*, 2021. 2, 3, 6, 7

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 1, 3, 6

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[23] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, pages 722–729, 2008. 6, 7

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 5

[25] Ioannis Pitas. *Digital image processing algorithms and applications*. John Wiley & Sons, 2000. 3, 5

[26] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. 6

[27] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *CVPR*, 2020. 7

[28] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114. PMLR, 2019. 7, 8

[29] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021. 1, 3, 4, 5

[30] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021. 1, 2, 3, 4, 5, 6, 7, 8, 9

[31] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 1, 2, 3, 4, 5, 6, 7, 8, 9

[32] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021. 3, 5

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 1

[34] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 3

[35] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *CVPR*, pages 4085–4095, 2020. 3

[36] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021. 3

[37] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019. 6

[38] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, Junsuk Choe, and Sanghyuk Chun. Re-labeling imagenet: from single to multi-labels, from global to localized labels. *arXiv preprint arXiv:2101.05022*, 2021. 6

[39] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 6

[40] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020. 1, 3

[41] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, volume 34, pages 13001–13008, 2020. 6

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We develop a general framework for image classification in this paper. Our model is not for specific applications.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We follow the common practice used in our baseline methods, where no error bars are reported.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See our implementation details in the experiment part.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]