Information Fusion 108 (2024) 102387

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

DBFFT: Adversarial-robust dual-branch frequency domain feature fusion in vision transformers

Jia Zeng [a], Lan Huang [a,b], Xingyu Bai [a], Kangping Wang [a,b,*]

[a] College of Computer Science and Technology, Jilin University, Changchun, 130012, China
[b] Key Laboratory of Symbolic Computation and Knowledge Engineering of the Ministry of Education, Jilin University, Changchun, 130012, China

ARTICLE INFO

Keywords:
Frequency domain
Domain fusion
Scale fusion
Vision transformer
Adversarial learning

ABSTRACT

Vision transformers (ViTs) have been successful in image recognition. However, it is difficult for ViTs to capture comprehensive information and resist adversarial perturbations by learning features from the spatial domain alone. Features with frequency domain information also play an important role in image classification and robustness improvement. In particular, the relative importance of spatial and frequency domain feature representations should vary depending on the encoding stage. Previous studies lack consideration of the flexible fusion of feature representations from different domains. To address this limitation, we propose a novel dual-branch adaptive frequency domain feature fusion architecture for Transformers with good classification ability and strong adversarial robustness, namely DBFFT. In each layer, we design two parallel Fourier transform and self-attention branches to learn hidden representations from the frequency domain and spatial domain, respectively. These are then adaptively weighted and fused according to their learned importance. Moreover, we further propose a dual-branch patch embedding fusion module. The module introduces different convolutional paths to extract input image features at different scales. The features are then embedded and combined into more informative tokens. Our DBFFT architecture can make full use of diverse domain and scale information, which benefits the image classification and enhances robustness against adversarial interference. Experimental results show that our DBFFT achieves promising performance and robustness in many image classification datasets and robustness benchmarks with favorable accuracy-complexity trade-offs.

1. Introduction

Vision Transformer (ViT) [1] has achieved remarkable results in the field of computer vision. Following the ViT, substantial variants have been developed to improve image classification performance [2–6] and enhance robustness [7–9]. Specifically, ViT models apply the multi-head attention mechanism to capture long-term dependencies from the spatial domain. However, for visual tasks, it is not sufficient to encode the hidden representation from a single domain [1,2,10]. The model is expected to understand more complex image information from both spatial domain and frequency domain. The model is less likely to be confused by adversarial noise in any one source since it can use the information from the other source to compensate. Therefore, the structure that integrates diverse domain features needs to be explored.

One primary challenge is to intelligently fuse hidden features across multiple domains. In computer vision, the spatial domain and frequency domain provide different perspectives to observe image features. The frequency of an image is a measure of the gray change intensity of visual features. Some features show more clear patterns in the frequency domain than the spatial domain and vice versa. In the frequency domain, the edges and textures can be better identified by looking for high-frequency information and repetitive patterns. In the spatial domain, the pixel values and the image objects can be better observed. Therefore, the model can have a more comprehensive ability to understand the images if it fuses information from different domains. The model can also be more adaptable against various image disturbances. The ViT is good at capturing information from the spatial domain, but it is not very powerful at processing features from the frequency domain. Recently, there has been some work [10,11] to introduce frequency domain information into vision transformer style models, improving the efficiency and adversarial robustness. These studies have replaced the original spatial attention layers either entirely [10] or partially [11] with the spectrum filters to learn the long and short term dependencies from the frequency domain. Unfortunately, these approaches only model the hidden feature representation from one domain in each layer. For example, ViT only focuses on spatial

* Corresponding author at: College of Computer Science and Technology, Jilin University, Changchun, 130012, China.
E-mail addresses: zengjia22@mails.jlu.edu.cn (J. Zeng), huanglan@jlu.edu.cn (L. Huang), bxy21@mails.jlu.edu.cn (X. Bai), wangkp@jlu.edu.cn (K. Wang).

https://doi.org/10.1016/j.inffus.2024.102387
Received 13 October 2023; Received in revised form 24 January 2024; Accepted 25 March 2024
Available online 28 March 2024

domain information, while GFNet [10] only focuses on frequency domain information. Although SpectFormer [11] combines the frequency domain and spatial domain, its serial fusion manner still causes to select one domain information and discard the other in one layer. To make the best use of diverse information, the model should have the integrative ability to learn hidden representations from both the frequency domain and the spatial domain at the same time. Furthermore, the contributions of frequency domain and spatial domain information to the model is not always equal during the encoding phase. SpectFormer [11] manually selects the domain in each layer. But we hypothesize that the fusion of domains should be a more flexible process that can automatically adjust the importance of different domains. Motivated by the above, an adaptive parallel combination is necessary to enable each layer to learn and trade off the feature representations from both the frequency domain and spatial domain.

In this paper, we design a novel dual-branch frequency domain feature fusion Transformer (DBFFT) to aggregate hidden representations from different domains adaptively. In each layer, one branch learns hidden feature representations in the spatial domain by self-attention and the other branch learns from the frequency domain perspective by spectrum transform in a parallel manner. The frequency branch consists of a Fourier transform, a frequency gating unit and an inverse Fourier transform. Moreover, the weights assigned to the two branches are learned to balance the importance of information from two domains. The diverse patterns are leveraged and combined to construct a more complete view, which can help DBFFT to boost the accuracy and adversarial robustness.

In addition to fusing hidden features, another objective is to design a mixer for the input features. For the input layer of ViT, the image is typically split into a sequence of non-overlapping patches and projected into token embeddings by a convolutional layer. Images are embedded using large convolution kernels and strides, which has a large receptive field and introduces more shape bias [12]. However, such a straightforward one-shot non-overlapping embedding will cause local information discontinuity. To solve this problem, a small stack of stride-two 3×3 convolutional kernels is proposed as the network's stem for progressive embedding [13]. Small kernel convolutions are easier to optimize, but they have relatively small receptive fields. To implement the advantages of each other, we construct a multi-scale feature fusion structure which combines one-shot and progressive embedding strategies as the input layer. This structure consists of two convolutional branches with different kernel sizes and strides, which can extract mixed-scale features, such as edges and shapes, from the original input image. These features are then adaptively fused into token embeddings to produce more general representations.

In order to more clearly demonstrate the contribution and originality of our DBFFT, we utilize a sketch to illustrate the difference between our DBFFT and other related methods in Fig. 1. Our contributions can be summarized as below:

1. To address the limitation that hidden representations in each layer only depend on a single domain, we design a parallel dual-branch fusion structure to adaptively weight and fuse hidden representations from both spatial and frequency domains in each layer for the improvement of modeling capabilities and adversarial robustness.

2. We construct a dual-branch structure for patch embedding which leverages diverse convolution designs to extract multi-scale input features. The features are then adaptively fused to enrich the scale information.

3. Extensive experiments on ImageNet and many robust benchmarks show our DBFFT achieves both good classification accuracy and strong robustness, outperforming recent vision transformer models and similar spectrum approaches. For example, DBFFT-H-L achieves 84.2% accuracy with $54M$ parameters. DBFFT-S improves +13.7% robustness over GFNet [10] on ImageNet-A adversarial dataset. Our DBFFT also performs well on transfer learning tasks and has good generalization ability.

This paper is organized as below: Related work are revisited in Section 2. The detail designs of DBFFT are explained in Section 3. Section 4 presents comprehensive experiments on image classification, adversarial robustness, ablation studies, transfer learning, generalization ability and visualization to evaluate the effectiveness of DBFFT. The conclusion is in Section 5.

2. Related work

2.1. Vision transformers

Dosovitskiy et al. [1] firstly apply the Transformer [14] blocks to vision tasks and design the vision transformer model. ViT has become a competitive alternative to convolutional neural networks (CNNs). Inspired by the success of ViT, many studies aim to improve the ViT models [2,4,5,15,16]. DeiT [2] improves the training strategies of ViT and achieves good performance on image classification with less data and computing resources. CaiT [15] proposes LayerScale and class attention to train deeper transformer networks, which increases the stability of the optimization at deeper scales. The above models are in a vanilla manner. Recently, some work are inspired by the hierarchical structure of CNNs [17–19] to build hierarchical Transformers. CvT [16] introduces convolutions to vision transformers, which brings more inductive bias to transformers. Wang et al. [4] design Pyramid Vision Transformer (PVT) which is a shrinking pyramid structure. In PVT, the length of sequence is reduced progressively as the network deepens to reduce the computation cost. Swin Transformer [5] carries out self-attention operations in windows and realizes global information interactions by moving windows, which reduces the model complexity. These improvements focus on the overall architecture design of ViT. Our dual-branch fusion structure can potentially apply to many of these backbones to improve performance.

2.2. Adversarial robustness in vision transformers

In addition to the accuracy, the adversarial robustness of ViTs is also investigated. Shao et al. [20] and Paul et al. [21] demonstrate that ViTs have stronger adversarial robustness than CNNs benefited from the self-attention mechanism. There are many work to improve the robustness by data augmentation [9,22] and structure design [7,8]. Herrmann et al. [9] present pyramid adversarial training method. It applies structured perturbations and flexing perturbations on images, leading to improving performance on out-of-distribution datasets. Qin et al. [22] transform images by adding patch-base perturbations as negatively augmented images to regularize training to improve robustness. Mao et al. [7] design a kind of discrete tokens and add them to the input layer to learn invariant and robust information. RVT [8] analyzes the contributions of components in ViT to robustness and combines those robust components. Our method enables the model to learn underlying structures more comprehensively by fusing information from different domains and scales, reducing the fragility against perturbations.

2.3. Frequency domain transform in vision

Frequency domain analysis is an important tool in digital image processing [23,24]. There are many methods to convert features from the spatial domain to the frequency domain, such as Fourier transform, discrete cosine transform (DCT) [25,26], wavelet transform [27] and so on. In computer vision, many work use frequency domain transform methods to process image features in the frequency domain to reduce computation costs and boost model performance. For CNN style models, FFC [28] designs a fast Fourier convolution with non-local receptive fields. FcaNet [29] considers the attention mechanism in CNNs from the perspective of frequency domain. FcaNet proposes a frequency-domain channel attention module for CNNs, which processes frequency components through DCT. For ViT style models, Guibas et al. design

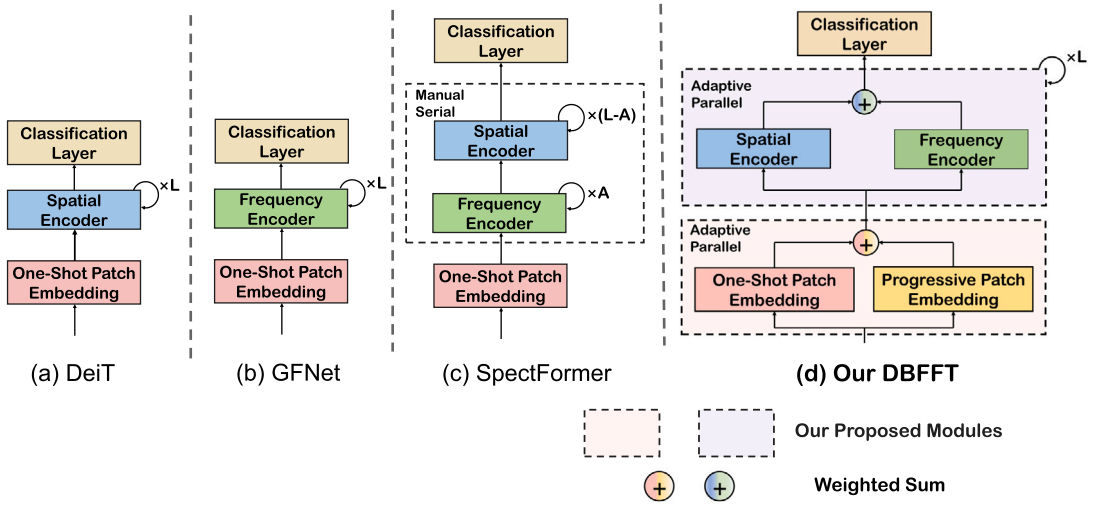

Fig. 1. A sketch of DBFFT. We explore a novel fusion strategy of combining spatial and frequency domain information, and multi-scale information. DeiT encodes the spatial hidden representations, and GFNet encodes the frequency hidden representations. SpectFormer fuses the spatial and frequency domain information in a serial manner and selects the kind of encoder and the number of encoders in a manual way, specifically assigning A frequency encoders and (L - A) spatial encoders. To integrate the domain and scale information at the same time more flexibly, our DBFFT fuses the information in an adaptive and parallel manner and fuses the multi-scale input features.

an AFNO [30] operator as an adaptive token mixer in Fourier domain. AFNO introduces a group-wise MLP (Multi-Layer Perceptron) to adaptively adjust Fourier frequency features. FourierFormer [31] views the attention as the non-parametric kernel regression and replaces query-key dot-product kernels to Fourier integral kernels. For lightweight networks, AFFNet [32] designs an adaptive frequency filter which can learn instance-adaptive representations in Fourier domain. DC-Former [33] explores different down-sampling strategies in frequency domain and learns semantics from DCT-based frequency domain representations. GFNet [10] replaces all the self-attention layers with global filter layers in ViT. The global filter layer introduces Fourier transform and learnable parameters to learn dependencies from the frequency domain. GFNet obtains the favorable accuracy and good robustness with log-linear complexity. SpectFormer [11] only replaces the self-attention in the initial layers with Fourier spectrum filters, which combines the spectral layers and self-attention layers in series. Inspired by GFNet and SpectFormer, which have shown both frequency domain dependencies and spatial domain dependencies play important roles in ViTs, we consider the information fusion of frequency domain and spatial domain to learn more representations and improve the performance. Our model is distinct from GFNet and SpectFormer in three aspects. (1) GFNet only models the interactions from the frequency domain, while our approach takes into account both the frequency domain and spatial domain. (2) The serial combination in SpectFormer means that each layer can only focus on either the frequency domain or the spatial domain, neglecting the other part. However, our parallel dual-branch structure can integrate the features of the two domains in each layer. (3) The number and order of frequency domain layers and self-attention layers are determined by manual design in SpectFormer. Our method can learn the fusion weights of frequency branch and spatial branch and adaptively combine features according to the importance of both.

2.4. Patch strategy in ViTs

The input to the standard Transformer architecture is a sequence of token embeddings. However, if the pixels of the original image are arranged into a token sequence, the sequence will become too long to compute. In order to reduce the amount of computation, it is necessary to split the image into small patches and take patches as the embedding and projection unit. In ViT [1], large-size convolution kernels (such as a 16×16 kernel) and strides are applied for convolutional projection. In paper [13], a convolutional stem consisting of a series of 3×3 convolution stacks is designed to replace the patchify stem of the original ViT to avoid optimization instability. CeiT [34] designs the image-to-token module, which uses small kernel convolution and pooling operations to replace the original patch embedding operations in ViT. CvT [16] replaces the non-overlapping convolution of the original ViT with a layer of overlapping convolution. Chen et al. [35] propose diversity regularization to regularize the patch embedding, which can eliminate redundancy of patch embedding and enhance diversity. LIT [36] and LITv2 [37] introduce deformable convolution [38] to sample more informative patches. However, these patch embedding approaches are designed to replace the original patchify stem. We suggest that it is a good choice to retain the original patch stem in combination with a progressive stem. This allows the model to capture both global and local features through different stem methods so that tokens can contain rich semantic information and the representation ability can be improved.

3. Method

This section is organized as below: The preliminaries of the Fourier transform are introduced in Section 3.1. The overall vanilla architecture of DBFFT is shown in Section 3.2. In Sections 3.3 and 3.4, the details of modules in DBFFT are explained. In Section 3.5, another architecture variant, hierarchical architecture, is designed to demonstrate the universality of our DBFFT strategy. Moreover, the detailed configurations of vanilla and hierarchical variants are summarized.

3.1. Preliminaries: Fourier transform

Fourier transform is a classical method of frequency domain transform. Since image features in computer vision are discrete signals, the discrete Fourier transform (DFT) is generally used in the field of digital image processing. Considering the 1-D DFT, let $x[n]$ be a finite-length sequence of length N, $n = 0, 1, \ldots, N-1$. The $x[n]$ sequence can be converted to the frequency domain by:

$$\begin{aligned} X[k] &= \text{1-}D\ DFT(x[n]) \\ &= \sum_{n=0}^{N-1} x[n] e^{-j(\frac{2\pi}{N})kn} \quad k = 0, 1, 2, \ldots, N-1, \end{aligned} \tag{1}$$

where j is the imaginary unit. It can be seen that the 1-D DFT output is also a discrete sequence. The output corresponding to each k is the weighted sum of all the input sequences. In addition, the inverse

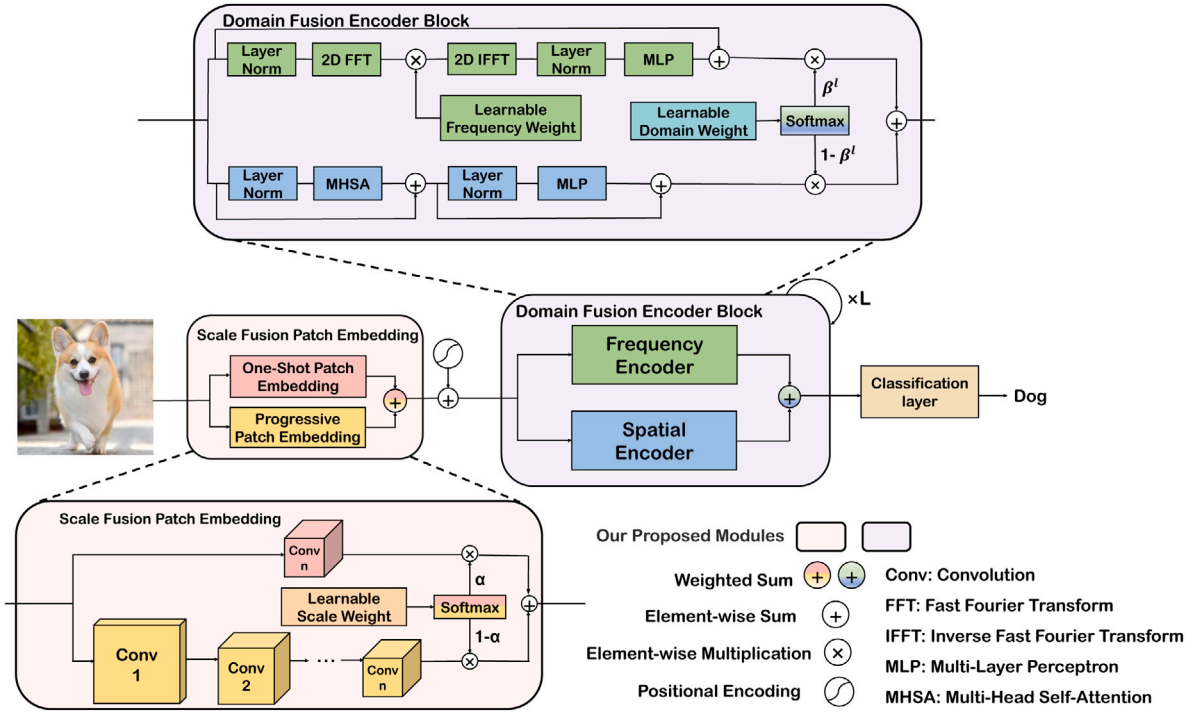

Fig. 2. The overall vanilla architecture of DBFFT. The details of two main modules, Domain Fusion Encoder Block and Scale Fusion Patch Embedding, are also displayed. The weighted sum symbols denote the adaptive fusion of two branches in color.

Fourier transform can be used to convert the frequency signal $X[k]$ into the spatial signal:

$$\begin{aligned} x[n] &= 1\text{-}D\ IDFT(X[k]) \\ &= \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j(\frac{2\pi}{N})kn} \quad n = 0, 1, 2, \ldots, N-1. \end{aligned} \tag{2}$$

Considering the case of 2-D signal, the 1-D DFT can be extended to the 2-D DFT. Given a 2-D finite signal sequence $x[m, n]$, $m = 0, 1, \ldots, M-1$ and $n = 0, 1, \ldots, N-1$, the 2-D DFT can be formulated as:

$$\begin{aligned} X[u, v] &= 2\text{-}D\ DFT(x[m, n]) \\ &= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x[m, n]e^{-j2\pi(\frac{um}{M}+\frac{vn}{N})} \\ & u = 0, 1, 2, \ldots, M-1, \quad v = 0, 1, 2, \ldots, N-1. \end{aligned} \tag{3}$$

Given the frequency sequence $X[u, v]$, the 2-D inverse Fourier transform can be formulated as:

$$\begin{aligned} x[m, n] &= 2\text{-}D\ IDFT(X[u, v]) \\ &= \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} X[u, v]e^{j2\pi(\frac{um}{M}+\frac{vn}{N})} \\ & m = 0, 1, 2, \ldots, M-1, \quad n = 0, 1, 2, \ldots, N-1. \end{aligned} \tag{4}$$

According to the formula, 2-D DFT can be decomposed into two steps of 1-D DFT.

The computational complexity of DFT is $O(N^2)$. In order to reduce the complexity, the fast Fourier transform (FFT) [39] is proposed. FFT can exploit the symmetry and periodicity of DFT formula to simplify and accelerate computation. FFT reduces the computational complexity to $O(N\,Log\,N)$. In our DBFFT, FFT is used to process frequency features in the frequency domain branch.

3.2. Overall

An overview of our DBFFT vanilla architecture is depicted in Fig. 2. Our model is composed of scale fusion patch embedding, domain fusion encoder blocks and classification head. The scale fusion patch embedding consists of two branches: (1) one-shot patch embedding; (2) progressive patch embedding. The domain fusion encoder block consists of frequency encoder and spatial encoder. The information in these branches are aggregated by element-wise weighted sum. The input image is first fed into the scale fusion patch embedding to obtain tokens with different scale features integrated. The positional encoding is added to the tokens. The tokens are then fed into a stack of domain fusion encoder blocks which encode and fuse the hidden representations learned from the frequency and spatial domains. Finally, the output of the encoder blocks passes through the classification head to get the classification result. According to the order in DBFFT, we will explain the scale fusion patch embedding and domain fusion encoder block, respectively, as below.

3.3. Scale fusion patch embedding module

Scale fusion patch embedding focuses on scale fusion of input features and builds the interactions between global and local features. The mixed-scale features can realize complement in face of image recognition and different kinds of adversarial attacks. In this module, the original image is fed into the one-shot patch embedding branch and progressive patch embedding branch, respectively. In the one-shot patch embedding branch, the original image $I \in R^{H\times W\times C}$ is split into non-overlapping patches of size $P \times P \times C$. Then $L = \frac{H}{P} \times \frac{W}{P}$ patches are embedded and flattened into tokens with dimension D. In particular, the above can be implemented by a $P \times P$ convolution with stride P. Then, a 1×1 convolution is used to mix the features along the channel. Finally, the flatten operation is used to obtain the tokens $T_{one-shot} \in R^{L\times D}$. These can be expressed as:

$$F_{one-shot} = GELU(BN(Conv_{P\times P}(I))), \tag{5}$$

$$T_{one-shot} = Flatten(Conv_{1\times 1}(F_{one-shot})), \tag{6}$$

where BN represents the BatchNorm operation [40] and GELU represents the GELU nonlinear activation function [41]. The input image is

embedded in one step by a large convolution kernel and stride in the one-shot branch, which enlarges the effective receptive field and allows the embedding layer to capture global features of the image.

Different from the one-shot embedding, we adopt a progressive embedding strategy in the progressive patch embedding branch to capture the local features. Concretely, four to six convolution blocks are stacked to progressively embed the input image I into a series of tokens. Each convolutional layer applies a 3×3 convolution kernel and a stride of 1 or 2. When the stride is 1, the resolution of the feature map and the number of output channels remain unchanged. However, when the stride is 2, the resolution is halved and the number of output channels is increased. Such small convolution kernels and overlapping convolution settings can capture detailed features and maintain continuity of image features. Symbolically, let the operator $O^n_{prog} = GELU \circ BN \circ Conv_{3\times3}$ represent the nth embedding operator stacked by the operations of 3×3 convolution, BatchNorm and GELU, $n = 1, 2, \ldots$. The output $T_{prog} \in R^{L\times D}$ of progressive patch embedding branch can be computed by:

$$F_{prog} = O^i_{prog}(...O^2_{prog}(O^1_{prog}(I))),\ i = 4, 5\ or\ 6 \tag{7}$$

$$T_{prog} = Flatten(Conv_{1\times1}(F_{prog})). \tag{8}$$

After patch embedding, the information of different scales is extracted through various convolutional settings of the two branches. The scale weight W_A is a learnable adaptive trade-off parameter and applies softmax to get α and $1-\alpha$ which control the relative importance of $T_{one-shot}$ and T_{prog}, respectively.

$$\alpha, 1-\alpha = Softmax(W_A). \tag{9}$$

Then, the features $T_{one-shot}$ and T_{prog} of the two branches are fused to $T_{scale-fusion} \in R^{L\times D}$ by:

$$T_{scale-fusion} = \alpha \times T_{one-shot} + (1-\alpha) \times T_{prog}. \tag{10}$$

The multi-scale features of the two branches are integrated by an element-wise weighted sum to achieve the scale fusion.

3.4. Domain fusion encoder block

The objective of the domain fusion encoder block is to achieve the domain information fusion of hidden representations. To this end, the frequency encoder captures frequency information of token features and learns long-term and short-term dependencies from the frequency domain. The spatial encoder learns the interactions among token features by self-attention from the perspective of spatial domain. The fusion of two kinds of domain information enriches the hidden feature representations and promotes the prediction ability and robustness.

In the frequency encoder branch, LayerNorm (LN) [42] is first applied on the input token features X^l in layer l by:

$$X^l_{norm} = LN(X^l). \tag{11}$$

The key idea of this branch is to mix frequency information of tokens. First, the dimension $R^{L\times D}$ of the token features X^l_{norm} will be reshaped to $R^{\frac{H}{P}\times\frac{W}{P}\times D}$. Then the hidden features are transformed from the spatial domain to the frequency domain by 2-D FFT along the spatial dimension. A learnable weight gating unit W^l_{gate} with the same dimension as the frequency features in the layer l is then set and multiplied by the frequency features element by element. The frequency gating unit can adaptively adjust various frequency components and determine the importance of each component. Finally, the frequency features are transformed back the spatial domain by 2-D inverse FFT (IFFT). The dimension will be reshaped back to $R^{L\times D}$. The 2-D FFT and 2-D IFFT operations can be implemented by the fft library provided by PyTorch to facilitate forward- and back-propagation on CPU and GPU. The frequency encoder of each block employs the above operations, so that a specific frequency-domain pattern can be learned in each block. The above can be expressed as:

$$\widetilde{T^l} = 2\text{-}D\ FFT(Reshape(X^l_{norm})) \odot W^l_{gate}, \tag{12}$$

$$T^l = Reshape(2\text{-}D\ IFFT(\widetilde{T^l})). \tag{13}$$

Following these, there are an LN layer and a three-layer MLP with GELU activation:

$$T^l_{freq} = X^l + MLP(LN(T^l)). \tag{14}$$

For another branch, spatial encoder, the standard multi-head self-attention (MHSA) and MLP are employed to learn the relationships among tokens in the spatial domain. Specifically, the LN layer and the MHSA layer is applied to the input X^l of layer l for token mixing, with a residual connection. Then an MLP is followed for channel mixing:

$$Z^l = X^l + MHSA(LN(X^l)), \tag{15}$$

$$T^l_{spatial} = Z^l + MLP(LN(Z^l)). \tag{16}$$

In the two domain branches, the hidden representations from both the frequency domain and the spatial domain are encoded in each block. The learnable domain weight parameter W_{B^l} in the layer l then applies softmax to get β^l and $1-\beta^l$. β^l represents the importance of hidden representations in the frequency domain branch in the l layer, while $1-\beta^l$ represents the importance of hidden representations in the spatial domain branch.

$$\beta^l, 1-\beta^l = Softmax(W_{B^l}). \tag{17}$$

To realize the adaptive fusion, the element-wise weighted sum is applied to different hidden representations in each layer:

$$X^{l+1} = \beta^l T^l_{freq} + (1-\beta^l) T^l_{spatial}. \tag{18}$$

After training, the β^l values in different layers are different, allowing different attentions to be paid to the frequency domain branch and the spatial branch in different layers. Therefore, the domain information is adjusted and fused more finely. We use two scalars to represent the importance of the branches since we treat the frequency encoder and the spatial encoder as two wholes. The use of scalars has better interpretation and is more simple and direct, which is convenient for network optimization.

3.5. Architecture variants

In this paper, we explore two variants of DBFFT architecture: vanilla architecture and hierarchical architecture. The above mainly describes the vanilla architecture. The vanilla architecture processes the fixed number of tokens in each encoder block, as shown in Fig. 2. Another popular hierarchical architecture is also considered. The hierarchical architecture borrows from the hierarchical pyramid design of CNNs. It is divided into a set of stages. The tokens are progressively downsampled in stages so the number of tokens is decreased in stages. We also design a dual-branch frequency domain fusion mechanism for PVT-like hierarchical variants, which is illustrated in Fig. 3.

Our DBFFT hierarchical variants consist of four stages. Each stage contains the patch embedding and the encoder block. In the stage 1, patch embedding adopts the scale fusion patch embedding mechanism similar to which in vanilla variants. The input image is passed through two branches. A 4×4 non-overlapping convolution block is designed in the one-shot patch embedding branch to embed tokens with large scale features. The four to five 3×3 overlapping convolution blocks are stacked in the progressive patch embedding branch to progressively embed the tokens with small scale features. Then the output tokens of the two branches are adaptively fused together as the input to the encoder block. However, there is no dual-branch design in the

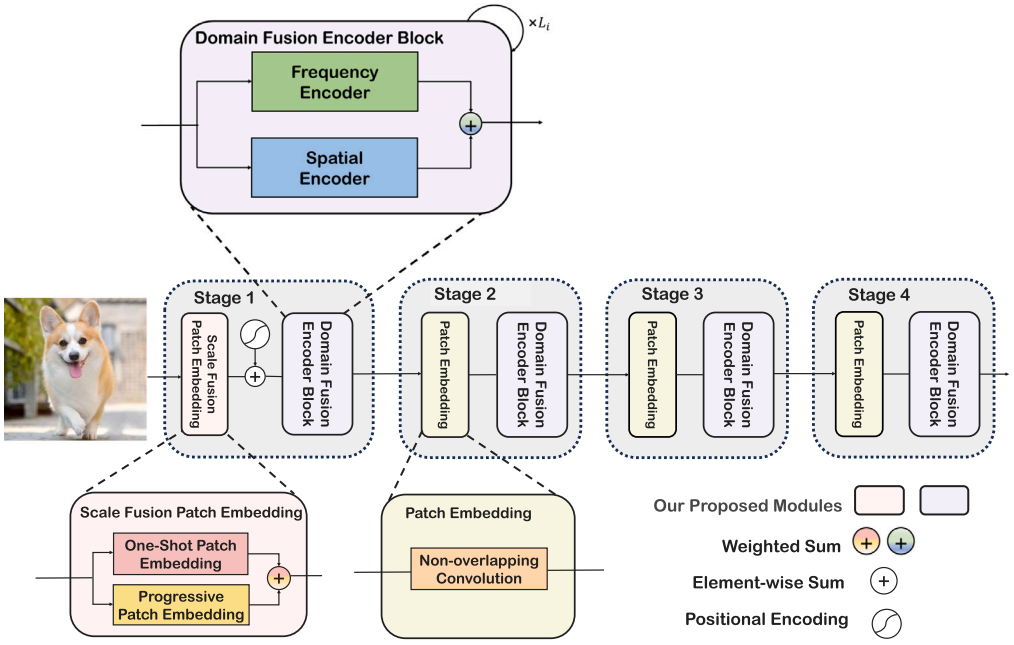

Fig. 3. The architecture of DBFFT hierarchical variants.

Table 1
Detailed configurations of various vanilla DBFFT variants. We show the number of channels in progressive patch embedding branch. We also show the details of the parallel frequency encoder and the spatial encoder. The expansion ratio of MLP layers in the spatial encoder is 4. C denotes the channel dimension and H refers to the number of heads.

Layer name	DBFFT-Ti	DBFFT-S	DBFFT-B
Progressive patch embedding channels	[64, 96, 128, 192]	[64, 128, 128, 192, 384]	[64, 128, 128, 256, 256, 512]
Frequency encoder	$[C = 192] \times 12$	$[C = 384] \times 12$	$[C = 512] \times 12$
Spatial encoder	$\begin{bmatrix} C = 192 \\ H = 3 \end{bmatrix} \times 12$	$\begin{bmatrix} C = 384 \\ H = 6 \end{bmatrix} \times 12$	$\begin{bmatrix} C = 512 \\ H = 8 \end{bmatrix} \times 12$

patch embedding of the stage 2, 3 and 4. Instead, only a 2 × 2 non-overlapping convolution with stride 2 is employed for down-sampling in these stages. The encoder block in each stage of hierarchical variants is similar to the domain fusion encoder designed in vanilla variants, which applies a frequency domain branch and a spatial domain branch to adaptively fuse the hidden representations. The method of weighted sum is also the same as that of vanilla variants.

We design various vanilla and hierarchical DBFFT variants with different model sizes. For vanilla variants, the DeiT-like architecture is adopted. We obtain variants of different model sizes by adjusting the number of embedding dimensions. These variants (DBFFT-Ti, DBFFT-S, DBFFT-B) have similar computational costs to the GFNet and SpectFormer vanilla architecture variants. The details of vanilla architectures are summarized in Table 1.

On the other hand, for hierarchical variants, we obtain variants of different sizes by adjusting the number of encoders in each stage and the number of embedding dimensions. The details of hierarchical architectures variants (DBFFT-H-Ti, DBFFT-H-S, DBFFT-H-B and DBFFT-H-L) are summarized in Table 2.

4. Experiments

We conduct extensive experiments on various datasets to evaluate our DBFFT. (1) Training vanilla DBFFT and hierarchical DBFFT from scratch on ImageNet-1K image classification dataset [43] and comparing the accuracy with other similar methods. (2) Conducting the robust experiments to evaluate adversarial robustness on Fast Gradient Sign Method (FGSM) [44] and Projected Gradient Descent (PGD) [45] attackers, ImageNet-A adversarial dataset [46] and AdvDrop [47] attackers using DBFFT pretrained on ImageNet-1K. (3) Conducting a series of ablation studies. (4) Transfer learning on CIFAR-10 and CIFAR-100 datasets [48] using DBFFT pretrained on ImageNet-1K. (5) Conducting generalization experiments on ImageNet-V2 [49] using DBFFT pretrained on ImageNet-1K.

4.1. ImageNet-1K classification

4.1.1. Dataset and setups

ImageNet-1K is a widely used image classification dataset. The training set contains 1.28 million images, and the validation set contains 50 000 images, with a total of 1000 categories. In order to make a fair comparison with previous similar methods, our model follows the most training details in GFNet [10]. The model was trained 300 epochs from scratch on 8 NVIDIA RTX 3090 GPUs. The AdamW optimizer [50] with 0.9 momentum is applied in the experiment. The initial learning rate is set to $\frac{batchsize}{512} \times 0.0005$ and is decayed by cosine schedule [51]. A linear warm-up learning rate is used for the first 5 epochs. In the experiment, gradient clipping to 1 is applied to stabilize the model optimization process. For data augmentation, random cropping, mixup [52], CutMix [53], label smoothing [54] and other strategies are applied. It is worth noting that we do not employ EMA (Exponential Moving Average) [55], repeated augmentation [56] and RandomErase [57] like papers [10,11]. LayerScale [15] is also used to stably train deep models for DBFFT-H-S, DBFFT-H-B and DBFFT-H-L. The stochastic depth coefficient [58] is set to 0, 0.15 and 0.25 for DBFFT-Ti, DBFFT-S and DBFFT-B and 0.1, 0.2, 0.3 and 0.4 for DBFFT-H-Ti, DBFFT-H-S, DBFFT-H-B and DBFFT-H-L like [10]. The evaluation metric is classification accuracy. The accuracy is higher, the classification performance is better.

Table 2
Detailed configurations of various hierarchical DBFFT variants. Since the scale fusion patch embedding is adopted in the stage 1, we show the number of channels in progressive patch embedding branch in the stage 1. In the each stage, we also show the details of the parallel frequency encoder and the spatial encoder. The expansion ratio of MLP layers in the spatial encoder is 4. In the stage i, C_i denotes the channel dimension and H_i refers to the number of heads.

Stage	Output size	Layer name	DBFFT-H-Ti	DBFFT-H-S	DBFFT-H-B	DBFFT-H-L
Stage 1	$\frac{H}{4} \times \frac{W}{4}$	Progressive patch embedding channels	[32, 32, 64, 64]	[32, 32, 64, 64]	[32, 32, 64, 64, 64]	[48, 48, 96,96,96]
		Frequency encoder	$[C_1 = 64] \times 2$	$[C_1 = 64] \times 3$	$[C_1 = 64] \times 3$	$[C_1 = 96] \times 3$
		Spatial encoder	$\begin{bmatrix} C_1 = 64 \\ H_1 = 1 \end{bmatrix} \times 2$	$\begin{bmatrix} C_1 = 64 \\ H_1 = 1 \end{bmatrix} \times 3$	$\begin{bmatrix} C_1 = 64 \\ H_1 = 1 \end{bmatrix} \times 3$	$\begin{bmatrix} C_1 = 96 \\ H_1 = 3 \end{bmatrix} \times 3$
Stage 2	$\frac{H}{8} \times \frac{W}{8}$	Frequency encoder	$[C_2 = 128] \times 2$	$[C_2 = 128] \times 4$	$[C_2 = 128] \times 4$	$[C_2 = 192] \times 4$
		Spatial encoder	$\begin{bmatrix} C_2 = 128 \\ H_2 = 2 \end{bmatrix} \times 2$	$\begin{bmatrix} C_2 = 128 \\ H_2 = 2 \end{bmatrix} \times 4$	$\begin{bmatrix} C_2 = 128 \\ H_2 = 2 \end{bmatrix} \times 4$	$\begin{bmatrix} C_2 = 192 \\ H_2 = 6 \end{bmatrix} \times 4$
Stage 3	$\frac{H}{16} \times \frac{W}{16}$	Frequency encoder	$[C_3 = 320] \times 3$	$[C_3 = 320] \times 6$	$[C_3 = 320] \times 12$	$[C_3 = 384] \times 18$
		Spatial encoder	$\begin{bmatrix} C_3 = 320 \\ H_3 = 5 \end{bmatrix} \times 3$	$\begin{bmatrix} C_3 = 320 \\ H_3 = 5 \end{bmatrix} \times 6$	$\begin{bmatrix} C_3 = 320 \\ H_3 = 5 \end{bmatrix} \times 12$	$\begin{bmatrix} C_3 = 384 \\ H_3 = 12 \end{bmatrix} \times 18$
Stage 4	$\frac{H}{32} \times \frac{W}{32}$	Frequency encoder	$[C_4 = 512] \times 2$	$[C_4 = 512] \times 3$	$[C_4 = 512] \times 3$	$[C_4 = 512] \times 3$
		Spatial encoder	$\begin{bmatrix} C_4 = 512 \\ H_4 = 8 \end{bmatrix} \times 2$	$\begin{bmatrix} C_4 = 512 \\ H_4 = 8 \end{bmatrix} \times 3$	$\begin{bmatrix} C_4 = 512 \\ H_4 = 8 \end{bmatrix} \times 3$	$\begin{bmatrix} C_4 = 512 \\ H_4 = 16 \end{bmatrix} \times 3$

Table 3
Comparisons with vanilla architectures on ImageNet-1K. We report the number of parameters, FLOPs, and the Top-1 and Top-5 accuracy on the validation set of ImageNet-1K. All of our DBFFT models are trained with 224 × 224 image resolution. We use the notation "↑ 384" to indicate models that are fine-tuned for 30 epochs on 384 × 384 image resolution. Our models are indicated in bold.

Model	Params (M)	FLOPs (G)	Top-1 Acc.	Top-5 Acc.
DeiT-Ti [2]	5	1.2	72.2%	91.1%
$ViT_C - 1GF$ [13]	5	1.1	75.3%	–
FourierFormer [31]	–	–	73.3%	91.7%
GFNet-Ti [10]	7	1.3	74.6%	92.2%
SpectFormer-T [11]	9	1.8	76.9%	93.4%
DBFFT-Ti	**7**	**1.5**	**78.6%**	**94.3%**
DeiT-S [2]	22	4.6	79.8%	95.0%
$ViT_C - 4GF$ [13]	18	4.0	81.4%	–
AFNO [30]	16	15.3	80.9%	95.4%
GFNet-S [10]	25	4.5	80.0%	94.9%
SpectFormer-S [11]	32	6.6	81.7%	95.6%
DBFFT-S	**25**	**5.5**	**82.2%**	**95.9%**
DeiT-B [2]	86	17.5	81.8%	95.6%
GFNet-B [10]	43	7.9	80.7%	95.1%
SpectFormer-B [11]	57	11.5	82.1%	95.7%
DBFFT-B	**45**	**9.6**	**82.7%**	**95.9%**
GFNet-S↑384 [10]	28	13.2	81.7%	95.8%
GFNet-B↑384 [10]	47	23.3	82.1%	95.8%
SpectFormer-S↑384 [11]	33	22.0	83.0%	96.3%
SpectFormer-B↑384 [11]	57	37.3	82.9%	96.1%
DBFFT-S↑384	**27**	**16.1**	**83.5%**	**96.5%**
DBFFT-B↑384	**47**	**28.1**	**83.6%**	**96.4%**

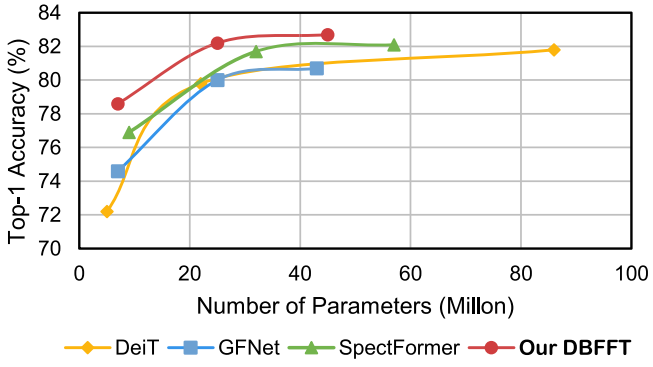

Fig. 4. The comparison of accuracy on ImageNet-1K vs. number of parameters of models. The upper left side of this figure represents the lower the complexity and the higher the accuracy.

4.1.2. Comparison with similar architectures

In the comparison of vanilla architectures, we report the classification accuracy on the ImageNet-1K image classification dataset with the parameters and floating point operations (FLOPs) in Table 3. Our DBFFT is compared to DeiT [2], ViT_C architecture [13] which employs multi-layer convolutions instead of the original patch embedding, FourierFormer [31], AFNO [30], GFNet [10] and SpectFormer [11], both of which introduce Fourier frequency domain feature processing to the Transformer, similar to DBFFT. The results show that our DBFFT vanilla architectures achieve better accuracy for all model sizes. Compared to GFNet [10] and the latest similar SpectFormer [11], our DBFFT models improve the accuracy by +4%/ + 2.2%/ + 2% and +1.7%/ + 0.5%/ + 0.6%, respectively, with favorable parameter sizes. Moreover, to illustrate the accuracy and complexity trade-offs of the proposed DBFFT model and the other relevant models more intuitively, we show the accuracy vs. complexity in Fig. 4. From the figure, it can be seen that our DBFFT achieves better the accuracy and complexity trade-offs. This demonstrates that our parallel adaptive fusion method, which focuses on both the frequency and spatial characteristics of each layer at the same time, performs better than methods that only consider self-attention (DeiT [2]), only consider frequency filters (GFNet [10]), or serially combine spectral and attention layers (SpectFormer [11]). The comparisons of important properties in these methods are summarized in detail on Table 4.

In addition, we show the fine-tuning performance of our models at higher resolution in Table 3. The models are trained on 224 × 224 and fine-tuned on 384 × 384. To accommodate larger image resolutions, we employ interpolation to adjust the dimensions of weights of trained parameters with varying resolutions during fine-tuning like GFNet [10]. We observe that our DBFFT achieves good performance in higher resolution.

We also carry out the ImageNet-1K classification experiments on the hierarchical variants, and the results are shown in Table 5. Our DBFFT hierarchical variants are compared to various kinds of hierarchical models. ResNet [18] and RegNet [59] are classic CNN baselines. FcaNet [29] introduces frequency domain attention mechanism by DCT in CNNs. In addition to CNNs, the comparisons of Transformer architectures are also considered. PVT [4] and Swin Transformer [5] are widely used hierarchical Transformer architectures. CvT [16] designs a layer of overlapping convolution in patch embedding. DCFormer [33] learns semantics from DCT-based frequency domain representations. LIT [36] architecture introduces deformable convolution in patch embedding and combines pure MLP and self-attention. LITv2 [37] designs deformable convolution in patch embedding and encodes tokens with different attention windows. GFNet-H [10] is the hierarchical variant of GFNet. It is worth mentioning that SpectFormer [11] only reports models that introduce class attention [15]. Therefore, for a fair and comprehensive comparison, we also train DBFFT-H-S with class attention, denoted as DBFFT-H-S*. Our DBFFT-H variants outperform

Table 4
Comparisons of our DBFFT with other similar methods in terms of properties.

Properties	DeiT [2]	GFNet [10]	SpectFormer [11]	Our DBFFT
Frequency information	✗	✓	✓	✓
Spatial information	✓	✗	✓	✓
Fusion method	✗	✗	Manual serial fusion	Adaptive parallel fusion
Patch embedding fusion	✗	✗	✗	✓

the above hierarchical models, which shows our dual-branch fusion mechanism is also applicable to hierarchical architectures.

Although hierarchical architectures have already considered both global and local features by progressively down-sampling tokens at each stage, the global and local features fusion strategy of our multi-scale patch embedding module is different. Our strategy focuses on the fusion from the perspective of input features, which is straightforward and simple. It involves designing two convolutional branches with different kernel sizes in the patch embedding part of the model. Our multi-scale feature fusion approach extracts and embeds original image features from the input stage, capturing more internal information related to the original image than hierarchical architectures. This enables our model to represent the image's overall structure and details more effectively.

4.2. Robustness

4.2.1. Benchmarks and setups

We evaluate the robustness of our DBFFT on four benchmarks. To measure the adversarial robustness, we choose two white-box adversarial attack algorithms including FGSM [44] and PGD [45], one adversarial dataset ImageNet-A [46] and one frequency domain adversarial attack AdvDrop [47] as the evaluation benchmarks. For white-box adversarial attack algorithms, FGSM [44] is a one-step algorithm which generates adversarial samples by moving along the direction of the gradient. PGD [45] carries out multi-step perturbations. ImageNet-A [46] collects 7500 unmodified natural adversarial examples which are likely to confuse the model. AdvDrop [47] is a frequency domain adversarial attack method, in which details of images are discarded in the frequency domain to generate adversarial samples.

For FGSM [44] on ImageNet-1K, the max attack magnitude ϵ is set to 1. The steps are $t = 5$ and the step size is $\alpha = 0.5$ in PGD [45]. For AdvDrop attack, 2000 validation images are randomly selected from ImageNet-1K [47]. The attack step is set to 30. The evaluation metrics are the Top-1 classification accuracy against these attackers on ImageNet-1K and adversarial examples on ImageNet-A. The higher accuracy represents the stronger robustness.

4.2.2. Results

We first compare our DBFFT with other models in FGSM, PGD and ImageNet-A benchmarks and the results are summarized in Table 6. Our DBFFT shows strong adversarial robustness in different parameter sizes against FGSM, PGD attackers. Our DBFFT models are also resistant to the natural adversarial images in ImageNet-A. It demonstrates that our dual-branch parallel fusion mechanism can take into account additional frequency information and different scale information, which has the advantages on handling adversarial images. Specifically, the fusion of frequency domain features and multi-scale features can capture richer image information. The image contains complex frequency component patterns and multi-scale information. By fusing the frequency domain feature representation and scale features, the model can focus on these details, leading to a more comprehensive understanding of the image. Moreover, our DBFFT model consists of two branches and each branch encodes the hidden representation of different domains. Each branch generates its own gradient and loss. However, FGSM and PGD attack methods perturb image pixels according to the gradient and loss of the whole model. They cannot effectively attack the model by only considering the overall gradient and loss. This decomposition makes it more challenging for adversarial attacks to manipulate the model, as they need to simultaneously consider the gradients and losses from both branches. Our adaptive fusion mechanism also allows for efficient and effective integration, further enhancing robustness. Our fusion branches have good complementarity, so the DBFFT model can resist this global disturbance well. In addition, we hypothesize that although FGSM, PGD attack methods are optimized to accommodate specific target network, they all directly change image pixels during the attack, which is more related to the spatial domain. The frequency component may be less sensitive to these perturbations because the frequency information focuses on the overall structure and texture information of the image. Therefore, our model with frequency domain information is more robust against FGSM and other attacks.

To further evaluate the robustness quality of features, we carry out the feature visualization experiment on the FGSM attack and ImageNet-A dataset by t-Distributed Stochastic Neighbour Embedding (t-SNE), respectively. The features are extracted from the linear layer before the

Table 5
Comparisons with hierarchical architectures included CNNs [18,59], spectral CNN [29], Transformers [4,5,16,36,37], spectral transformers [10,11,33] and our DBFFT-H models, which have similar number of parameters and FLOPs. We report the Top-1 and Top-5 accuracy on the validation set of ImageNet-1K. Our models are indicated in bold.

Model	Params (M)	FLOPs (G)	Top-1 Acc.	Top-5 Acc.
ResNet-18 [18]	12	1.8	69.8%	89.1%
RegNetY-1.6GF [59]	11	1.6	78.0%	–
PVT-Ti [4]	13	1.9	75.1%	–
GFNet-H-Ti [10]	15	2.1	80.1%	95.1%
DBFFT-H-Ti	**14**	**2.6**	**80.8%**	**95.4%**
ResNet-50 [18]	26	4.1	76.1%	92.9%
RegNetY-4.0GF [59]	21	4.0	79.4%	–
FcaNet-S [29]	28	4.1	78.6%	94.1%
CvT-13 [16]	20	4.5	81.6%	–
PVT-S [4]	25	3.8	79.8%	–
Swin-Ti [5]	29	4.5	81.3%	–
GFNet-H-S [10]	32	4.6	81.5%	95.6%
DCFormer-SW-T [33]	28	4.5	81.2%	–
LIT-S [36]	27	4.1	81.5%	–
LITv2-S [37]	28	3.7	82.0%	–
DBFFT-H-S	**23**	**4.3**	**82.8%**	**96.2%**
SpectFormer-H-S[a] [11]	22	3.9	83.1%	96.3%
DBFFT-H-S[a]	**26**	**4.4**	**83.3%**	**96.4%**
ResNet-101 [18]	45	7.9	77.4%	93.5%
RegNetY-8.0GF [59]	39	8.0	79.9%	–
FcaNet-M [29]	49	7.9	79.6%	94.6%
CvT-21 [16]	32	7.1	82.5%	–
PVT-M [4]	44	6.7	81.2%	–
Swin-S [5]	50	8.7	83.0%	–
GFNet-H-B [10]	54	8.6	82.9%	96.2%
DCFormer-SW-S [33]	50	8.7	82.8%	–
LIT-M [36]	48	8.6	83.0%	–
LITv2-M [37]	49	7.5	83.3%	–
DBFFT-H-B	**31**	**6.0**	**83.5%**	**96.5%**
RegNetY-12GF [59]	52	12.1	80.3%	–
FcaNet-L [29]	67	11.6	80.0%	94.9%
PVT-L [4]	61	9.8	81.7%	–
Swin-B [5]	88	15.4	83.5%	–
DCFormer-SW-B [33]	88	15.4	83.1%	–
LIT-B [36]	86	15	83.4%	–
LITv2-B [37]	87	13.2	83.6%	–
DBFFT-H-L	**54**	**12.4**	**84.2%**	**96.8%**

[a] Denotes the model additionally trained with the class attention [15].

Table 6
The adversarial robustness performance of DBFFT compared to CNNs [18,59–61], Transformers [2–5,62], frequency-style models [10,11], adversarial robust models [7,9,22] on three robustness benchmarks. On FGSM [44], PGD [45] and ImageNet-A [46] adversarial robustness benchmarks, the higher accuracy is better. Our models are indicated in bold.

	Model	Params (M)	FLOPs (G)	FGSM	PGD	ImageNet-A
CNN	ResNet-50 [18]	26	4.1	12.2%	0.9%	0.0%
	Inception v3 [60]	27	5.7	22.5%	3.1%	10.0%
	RegNetY-4GF [59]	21	4.0	15.4%	2.4%	8.9%
	ResNeXt50-32×4d [61]	25	4.3	34.7%	13.5%	10.7%
Transformer	DeiT-Ti [2]	5	1.2	22.3%	6.2%	7.3%
	PVT-Ti [4]	13	1.9	10.0%	0.5%	7.9%
	GFNet-Ti [10]	7	1.3	23.6%	7.6%	6.3%
	DBFFT-Ti	**7**	**1.5**	**37.4%**	**15.8%**	**15.7%**
	DeiT-S [2]	22	4.6	40.7%	16.7%	18.9%
	PVT-S [4]	25	3.8	26.6%	3.1%	18.0%
	Swin-T [5]	28	4.5	33.7%	7.3%	21.6%
	TNT-S [62]	24	5.2	33.2%	4.2%	24.7%
	T2T-ViT t-14 [3]	22	6.1	40.9%	11.4%	23.9%
	GFNet-S [10]	25	4.5	42.6%	21.0%	14.3%
	SpectFormer-S [11]	32	6.6	45.9%	23.8%	21.2%
	DBFFT-S	**25**	**5.5**	**48.3%**	**24.1%**	**28.0%**
	DeiT-B [2]	86	17.5	46.4%	21.3%	27.4%
	PVT-L [4]	61	9.8	33.1%	7.3%	26.6%
	Swin-B [5]	88	15.4	49.2%	21.3%	35.8%
	T2T-ViT t-24 [3]	64	15.0	46.7%	17.5%	28.9%
	GFNet-B [10]	43	7.9	43.7%	20.7%	16.9%
	ViT-B+Discrete [7]	86	–	–	–	6.5%
	Hybrid ViT-B+Discrete [7]	–	–	–	–	17.2%
	ViT-B/16+PyramidAT [9]	86	–	–	–	23.0%
	ViT-B/16+Patch-wise [22]	86	–	–	–	8.6%
	DBFFT-B	**45**	**9.6**	**49.1%**	**25.8%**	**31.2%**

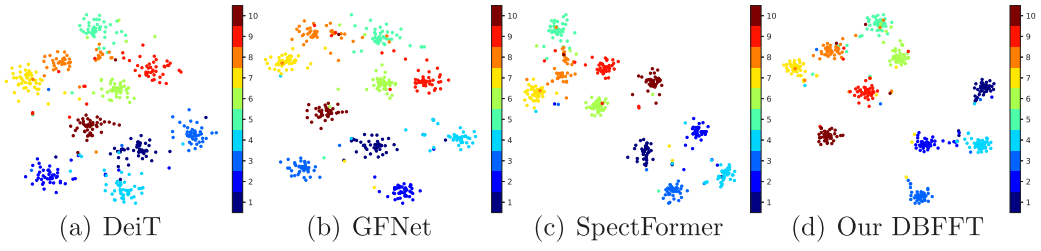

Fig. 5. t-SNE visualizations on FGSM attack. Classes are distinguished by color.

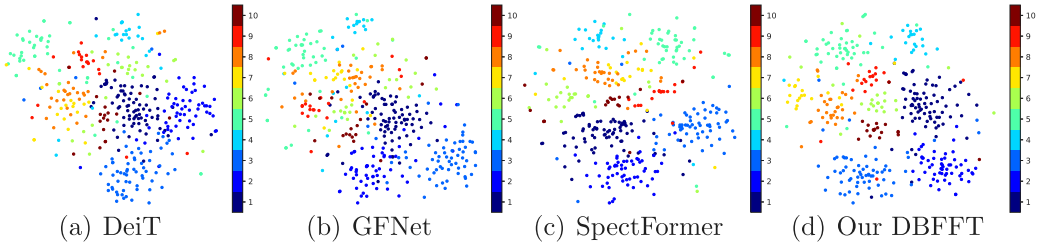

Fig. 6. t-SNE visualizations on ImageNet-A. Classes are distinguished by color.

classification head on DeiT-S, GFNet-S, SpectFormer-S and our DBFFT-S model. For t-SNE in FGSM, we randomly select 10 classes from ImageNet-1K validation set. For t-SNE in ImageNet-A, we randomly select 10 classes from ImageNet-A. Figs. 5 and 6 shows the inter-class and intra-class structure of various features on FGSM attack and ImageNet-A dataset. Compared to these models, the feature structure presents a better distribution in our DBFFT. The robust features learned by our DBFFT are gathered more closely within classes and are more separate between classes. It can reveal that our DBFFT has stronger robustness.

We also evaluate the robustness of our DBFFT when the model is against frequency domain adversarial attack method. We compare our DBFFT with similar models in the AdvDrop benchmark. As shown in Table 7, our DBFFT achieves higher accuracy against AdvDrop. The robustness of our frequency domain fusion DBFFT model is stronger. It demonstrates that our adaptive parallel frequency domain information fusion strategy has advantages in dealing with the frequency domain adversarial attack.

4.3. Ablation studies and analysis

We conduct a series of ablation studies to analyze factors contributing to DBFFT's performance of accuracy and robustness and understand DBFFT better. All the ablation experiments are carried out based on DBFFT-Ti. The standard accuracy on ImageNet-1K and robust accuracy against PGD attack are reported. The experimental setups are the same as Sections 4.1 and 4.2.

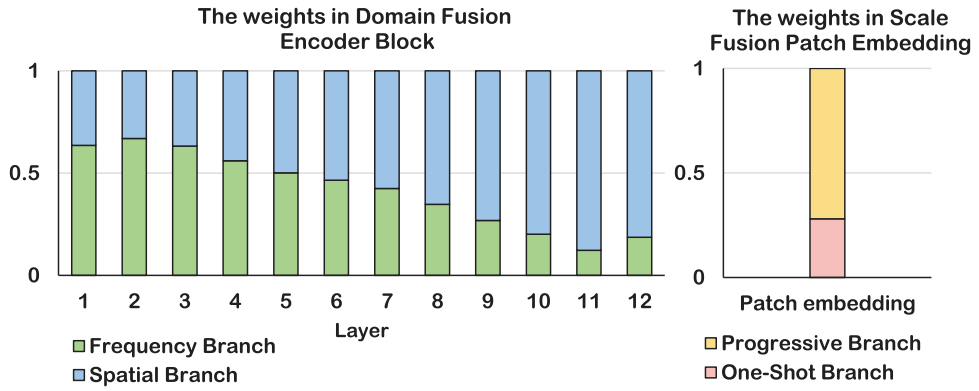

Fig. 7. The detailed weights of the two branches in domain fusion encoder block and scale fusion patch embedding module.

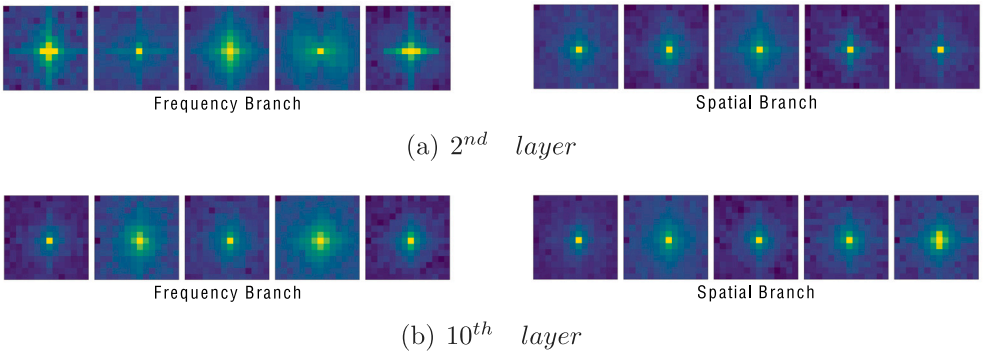

Fig. 8. The spectrum from 5 channels of the frequency branch and spatial branch in the 2nd and 10th layer. The magnitude of the spectrum is averaged over 64 samples. The lighter color means the magnitude is stronger. The closer to center point of the spectrum represents the lower frequency.

Table 7
The comparison of adversarial robustness performance against AdvDrop [47]. The higher accuracy represents the stronger robustness. Our model is indicated in bold.

Model	Params (M)	FLOPs (G)	AdvDrop
DeiT-S [2]	22	4.6	3.3%
GFNet-S [10]	25	4.5	6.3%
SpectFormer-S [11]	32	6.6	7.7%
DBFFT-S	**25**	**5.5**	**8.5%**

4.3.1. Dual-branch structure

In order to explore the validity of each branch in our DBFFT, we retain only the individual branch in the encoder block and the patch embedding module, respectively, and observe the results. We enable only frequency branch or spatial branch in domain fusion encoder block and report the results in the top half part of Table 8, where ✓and ✗ indicate whether the branch is enabled or not, respectively. It is worth mentioning that the standard and robust accuracy of frequency-only model are relatively low because the number of parameters of frequency branch in DBFFT is too few. The effectiveness and robustness of the frequency encoder with a suitable parameter size has been demonstrated by GFNet [10] and other papers. As shown in the table, the model of parallel combining the spatial and frequency branches outperforms which of retaining only a single branch, demonstrating the effectiveness of our domain fusion encoder. Table 8 also shows the results of retaining only the one-shot branch or progressive branch in the patch embedding module in the bottom half part. The model with only progressive branch achieves better accuracy than the model with only one-shot branch, demonstrating that introducing more fine-grained inductive bias is beneficial for model optimization. The low-level features also impose a positive effect on robustness. In addition, we observe that the performance of the dual-branch model is better than these of the single-branch models. This shows that our fusion design in patch embedding fully integrates features of different scales and achieves a balance between extracting fine- and coarse-grained image features.

Moreover, since SpectFormer only applies one-shot patch embedding in the patch embedding module, we additionally design a one-shot patch embedding model with more similar parameter size to SpectFormer-T to evaluate the effectiveness of our parallel frequency fusion method in more detail. We increase the number of parameters in the one-shot model from 6.5M to 9.9M by increasing the dimensions of the encoders from 192 to 240 dimensions. In the Table 8, our model outperforms SpectFormer-T, which demonstrates the effectiveness of our adaptive parallel fusion strategy.

4.3.2. Fusion method

We have investigated the macro fusion methods of the frequency domain layer and the self-attention layer (serial fusion in [11] or parallel fusion in DBFFT) in the previous Section 4.1.2. In this section, we explore the micro fusion methods of the two branches. One is the additive fusion method, which means the outputs of the two branches are element-wise added together, as in our DBFFT model. The other is the concatenation fusion method. Concretely, considering that the concatenation operation will make the dimension double and greatly increase the final parameter sizes, we split the input of each layer in half along the channel dimension for model size restraint. One half of the input is processed by the frequency domain branch, while the other half is processed by the spatial self-attention branch. The outputs of the two branches are then concatenated together and fed into a three-layer MLP to exchange information. Table 9 shows the comparative results of the two fusion methods. It can be seen that the element-wise add fusion method is better than the concatenation method. This may be because: (1) with similar model scales, the additive fusion method maintains the original feature dimension and contains more information than

Table 8
Ablation study of domain fusion and scale fusion.

	One-shot patch embedding	Progressive patch embedding	Frequency encoder	Spatial encoder	Params (M)	FLOPs (G)	Top-1 Acc.	PGD Acc.
Branches in domain	✓	✓	✗	✓	6.2	1.4	76.8%	11.1%
fusion encoder block	✓	✓	✓	✗	1.6	0.4	57.2%	0.5%
Branches in scale fusion	✗	✓	✓	✓	6.7	1.4	78.3%	14.1%
patch embedding	✓	✗	✓	✓	6.5	1.1	75.3%	12.9%
module	✓	✗	✓	✓	9.9	1.7	77.3%	15.5%
Our dual-branch DBFFT	✓	✓	✓	✓	**6.9**	**1.5**	**78.6%**	**15.8%**

Table 9
Ablation study of micro fusion methods.

Fusion method	Params (M)	FLOPs (G)	Top-1 Acc.	PGD Acc.
Element-wise add	**7**	**1.5**	**78.6%**	**15.8%**
Concatenation	6	1.3	76.1%	6.5%

Table 10
Ablation study of middle dimension in frequency branch.

Dimension	Params (M)	FLOPs (G)	Top-1 Acc.	PGD Acc.
$input\_dim \div 8$	6.8	1.4	78.4%	15.0%
$input\_dim \div 4$	6.9	1.5	78.6%	15.8%
$input\_dim \times 4$	10.3	2.1	79.5%	17.7%

Table 11
Ablation study of fusion weight.

Weight	Params (M)	FLOPs (G)	Top-1 Acc.	PGD Acc.
Both 0.5	7	1.5	78.4%	14.9%
Softmax	**7**	**1.5**	**78.6**%	**15.8**%

Table 12
Ablation study of weighted fusion method.

Weight	Params (M)	FLOPs (G)	Top-1 Acc.	PGD Acc.
MLP weighted fusion	7	1.5	77.6%	13.3%
Softmax weighted fusion	**7**	**1.5**	**78.6**%	**15.8**%

the concatenation method; (2) the concatenation method requires an MLP fusion module, which equivalently adds more layers to the model, potentially hindering model optimization.

4.3.3. Middle dimension in frequency branch

Since the middle dimension of the MLP is always 4 times the input dimensions in the encoder of most Transformer-based models, this convention is also followed in the spatial branch of DBFFT. In this section, we explore the optimal middle layer dimension of the MLP in the frequency branch. The dimension size determines the number of parameters in the frequency branch and further controls the size of the overall model. The dimension of the middle layer is denoted by its quantitative relation to the input dimension. In our DBFFT model, the middle dimension is set to $input\_dim \div 4$ to maintain a proper model size. On this basis, we conduct ablation studies by reducing or increasing the middle dimension to seek for trade-off between model size and accuracy. The results are summarized in the Table 10. The standard accuracy and robustness accuracy of the model with $input\_dim \div 8$ middle dimension is 0.2% and 0.8% lower than that of DBFFT with similar model parameters. When the middle dimension reaches $input\_dim \times 4$, the accuracy is slightly higher than that of DBFFT with $input\_dim \div 4$. However, the number of parameters and FLOPs in this model is greatly increased. Therefore, our DBFFT with $input\_dim \div 4$ middle dimension achieves the best balance between cost and accuracy.

4.3.4. Adaptive fusion

In our DBFFT, softmax parameters are introduced to determine the importance weights of branches. The outputs are then integrated by weighted sum for adaptive fusion. We believe that different concentrations should be given to the branches depending on the stages. To justify this hypothesis, we investigate the effect of the importance weights: (1) the same concentration is paid to the branches, which means the weights of both branches are set to 0.5; (2) softmax is used to adaptively adjust the importance weights. The results are shown in Table 11. From the table, we observe that our softmax weighted sum performs better than equal weighting on standard and robust accuracy. It indicates that our DBFFT can effectively balance the various information in the different stages. We further explore the weights of branches in each layer. As shown in Fig. 7, for the encoder block, the frequency domain branch receives more attention in the early layers, but the spatial attention branch becomes more important in the deeper layers. The architecture of SpectFormer [11], which only replaces the initial attention layers with spectral layers, illustrates a similar condition. However, our fusion method considers the adaptive fusion. For the patch embedding module, the progressive branch has a higher ratio.

To evaluate the effectiveness of softmax weighted sum method, we also design another weighted approach to carry out ablation studies. We design a more sophisticated weighted approach, which utilizes two small three-layer MLPs to learn the weights of one-shot and progressive patch embedding branch, frequency and spatial encoder. The comparison results are shown in the Table 12. From the table, our softmax weighted sum method performs better than the MLP weighted fusion method. The reason may be that the MLP-style weighted fusion method introduces multiple MLPs, which increases the complexity of the model and makes model optimization more difficult.

4.4. Transfer learning

4.4.1. Datasets and setups

In order to evaluate the generality of our DBFFT, we conduct transfer learning experiments on the CIFAR-10 and CIFAR-100 datasets. CIFAR-10 contains 60 000 color images across 10 classes. There are 50 000 images in training set and 10 000 images in test set. CIFAR-100 has 100 classes, each containing 600 images, with 500 in training set and 100 in test set. The models pretrained on ImageNet-1K are loaded and are fine-tuned on the CIFAR-10 and CIFAR-100 datasets for 200 epochs like GFNet [10]. The initial learning rate and the weight decay are set to 0.0001. The evaluation metric is Top-1 classification accuracy.

4.4.2. Results

We compare the transfer learning performance of our DBFFT to other models, and the results are shown in Table 13. From the table, our DBFFT also performs well on the CIFAR-10 and CIFAR-100 downstream datasets. Compared to CNN-style [63], Transformer-style [1,2], MLP-style [64], frequency-style models [10,11], our DBFFT achieves better or similar accuracy on transfer learning.

4.5. Generalization

4.5.1. Datasets and setups

In order to test the generalization ability of our DBFFT, we conduct experiments on ImageNet-V2 dataset [49]. Different from the original

Table 13
Accuracy on transfer learning. Our models are indicated in bold.

Model	Params (M)	FLOPs (G)	CIFAR-10	CIFAR-100
EfficientNet-B7 [63]	66	37	98.9%	91.7%
ViT-B/16 [1]	86	55.4	98.1%	87.1%
ViT-L/16 [1]	307	190.7	97.9%	86.4%
DeiT-B/16 [2]	86	17.5	99.1%	90.8%
ResMLP-12 [64]	15	3.0	98.1%	87.0%
ResMLP-24 [64]	30	6.0	98.7%	89.5%
GFNet-H-B [10]	54	8.6	99.0%	90.3%
SpectFormer-B [11]	57	11.5	98.9%	90.3%
DBFFT-H-B	**31**	**6.0**	**99.1%**	**90.3%**
DBFFT-B	**45**	**9.6**	**99.1%**	**90.4%**

Table 14
The accuracy on ImageNet-V2. The generalization ability is measured on ImageNet-V2. Our model is indicated in bold.

Model	Params (M)	FLOPs (G)	Top-1 Acc.
ResNet-50 [18]	26	4.1	67.4%
ResNeXt50-32×4d [61]	25	4.3	68.2%
DeiT-S [2]	22	4.6	68.4%
ResMLP-12 [64]	15	3.0	64.4%
GFNet-S [10]	25	4.5	68.5%
DBFFT-S	**25**	**5.5**	**71.2%**

Imagenet-1K, Imagenet-V2 is a recollected validation set by repeating the collection process of ImageNet-1K from a larger data source. ImageNet-V2 will test how well the model generalizes to new data, with the hope that it can perform well on new images collected from the similar sources. The ImageNet-V2 dataset has 1000 classes and contains 10 000 images. The models pretrained on ImageNet-1K are loaded and are evaluated on ImageNet-V2. The evaluation metric is the Top-1 classification accuracy. The classification accuracy is higher, the generalization ability is better.

4.5.2. Results

The experimental results are shown in Table 14. Compared to other models, our DBFFT model performs better on ImageNet-V2. It demonstrates that our DBFFT model has better generalization ability.

4.6. Visualization

In order to explore different schemes of the spatial and frequency branches, inspired by LITv2 [37], we visualize the spectrum of the output feature maps from the two branches in Fig. 8. We take the 2nd and 10th layers of the DBFFT-Ti model as examples.

We observe the spatial self-attention branch is good at capturing low-frequency components, while the frequency branch has the better ability to capture more mid-frequency and high-frequency components due to its frequency mixing mechanism. Moreover, in the initial layers, the frequency branch has the higher weights, which encourages it to learn more diverse frequency schemes, such as more mid-frequency and high-frequency signals, than the spatial branch. In the higher layers, the frequency branch is with the lower weights, which causes it to capture more similar frequency components to the spatial branch. The results reflect that the initial layers tends to capture the fine details of images while the higher layers focus on learning more general features.

5. Conclusion

We present DBFFT, a novel dual-branch frequency fusion Transformer that couples self-attention and frequency layers in a parallel manner, which realizes the adaptive fusion of hidden representations from the spatial domain and frequency domain. We also design a dual-branch patch embedding module containing two different convolutional paths to fuse different-scale input features. Our DBFFT can adaptively and flexibly integrate both domain and scale information according to the importance of each branch. Extensive experiments on ImageNet-1K demonstrate that our DBFFT outperforms widely used and similar models in terms of accuracy-complexity trade-off. Moreover, our DBFFT exhibits outstanding adversarial robustness against attacks and natural adversarial images. On transfer learning tasks, our DBFFT also works well on downstream datasets, such as CIFAR-10 and CIFAR-100. Our DBFFT also shows good generalization ability. In future work, we plan to explore more complex and effective combination mode by neural architecture search (NAS). Additionally, we aim to analyze and research attacks from the perspective of frequency domain and develop new techniques to attack the model.

CRediT authorship contribution statement

Jia Zeng: Conceptualization, Formal analysis, Methodology, Software, Writing – original draft, Writing – review & editing. **Lan Huang:** Conceptualization, Resources. **Xingyu Bai:** Software. **Kangping Wang:** Conceptualization, Resources, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62072212), the Development Project of Jilin Province of China (No. 20240101364JC, 20230201065GX), and the Jilin Provincial Key Laboratory of Big Data Intelligent Cognition (No. 20210504003GH).

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: 9th International Conference on Learning Representations, ICLR, 2021.

[2] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: Proceedings of the 38th International Conference on Machine Learning, ICML, Vol. 139, 2021, pp. 10347–10357.

[3] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F.E.H. Tay, J. Feng, S. Yan, Tokens-to-token ViT: Training vision transformers from scratch on ImageNet, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 538–547, http://dx.doi.org/10.1109/ICCV48922.2021.00060.

[4] W. Wang, E. Xie, X. Li, D. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao, Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 548–558, http://dx.doi.org/10.1109/ICCV48922.2021.00061.

[5] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: 2021 IEEE/CVF International Conference on Computer Vision, 2021, pp. 9992–10002, http://dx.doi.org/10.1109/ICCV48922.2021.00986.

[6] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, J. Gao, Focal attention for long-range interactions in vision transformers, in: Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS, 2021, pp. 30008–30022.

[7] C. Mao, L. Jiang, M. Dehghani, C. Vondrick, R. Sukthankar, I. Essa, Discrete representations strengthen vision transformer robustness, in: The Tenth International Conference on Learning Representations, ICLR, 2022.

[8] X. Mao, G. Qi, Y. Chen, X. Li, R. Duan, S. Ye, Y. He, H. Xue, Towards robust vision transformer, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 12032–12041, http://dx.doi.org/10.1109/CVPR52688.2022.01173.

[9] C. Herrmann, K. Sargent, L. Jiang, R. Zabih, H. Chang, C. Liu, D. Krishnan, D. Sun, Pyramid adversarial training improves ViT performance, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 13409–13419, http://dx.doi.org/10.1109/CVPR52688.2022.01306.

[10] Y. Rao, W. Zhao, Z. Zhu, J. Lu, J. Zhou, Global filter networks for image classification, in: Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems, NeurIPS, 2021, pp. 980–993.

[11] B.N. Patro, V.P. Namboodiri, V.S. Agneeswaran, SpectFormer: Frequency and attention is what you need in a vision transformer, 2023, http://dx.doi.org/10.48550/arXiv.2304.06446, CoRR abs/2304.06446.

[12] X. Ding, X. Zhang, J. Han, G. Ding, Scaling up your kernels to 31×31: Revisiting large kernel design in CNNs, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 11953–11965, http://dx.doi.org/10.1109/CVPR52688.2022.01166.

[13] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, R.B. Girshick, Early convolutions help transformers see better, in: Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems, NeurIPS, 2021, pp. 30392–30400.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, 2017, pp. 5998–6008.

[15] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, H. Jégou, Going deeper with image transformers, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 32–42, http://dx.doi.org/10.1109/ICCV48922.2021.00010.

[16] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, L. Zhang, CvT: Introducing convolutions to vision transformers, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 22–31, http://dx.doi.org/10.1109/ICCV48922.2021.00009.

[17] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems, 2012, pp. 1106–1114.

[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778, http://dx.doi.org/10.1109/CVPR.2016.90.

[19] T. Lin, P. Dollár, R.B. Girshick, K. He, B. Hariharan, S.J. Belongie, Feature pyramid networks for object detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 936–944, http://dx.doi.org/10.1109/CVPR.2017.106.

[20] R. Shao, Z. Shi, J. Yi, P. Chen, C. Hsieh, On the adversarial robustness of vision transformers, Trans. Mach. Learn. Res. 2022 (2022).

[21] S. Paul, P. Chen, Vision transformers are robust learners, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, the Twelveth Symposium on Educational Advances in Artificial Intelligence, 2022, pp. 2071–2081, http://dx.doi.org/10.1609/aaai.v36i2.20103.

[22] Y. Qin, C. Zhang, T. Chen, B. Lakshminarayanan, A. Beutel, X. Wang, Understanding and improving robustness of vision transformers through patch-based negative augmentation, in: NeurIPS, 2022.

[23] G.A. Baxes, Digital Image Processing - Principles and Applications, Wiley, 1994.

[24] I. Pitas, Digital Image Processing Algorithms and Applications, Wiley, 2000.

[25] N. Ahmed, T.R. Natarajan, K.R. Rao, Discrete cosine transform, IEEE Trans. Comput. 23 (1) (1974) 90–93, http://dx.doi.org/10.1109/T-C.1974.223784.

[26] K.R. Rao, P.C. Yip, Discrete cosine transform - Algorithms, advantages, applications, 1990, http://dx.doi.org/10.1016/c2009-0-22279-3.

[27] S. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, IEEE Trans. Pattern Anal. Mach. Intell. 11 (7) (1989) 674–693, http://dx.doi.org/10.1109/34.192463.

[28] L. Chi, B. Jiang, Y. Mu, Fast Fourier convolution, in: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS, 2020.

[29] Z. Qin, P. Zhang, F. Wu, X. Li, FcaNet: Frequency channel attention networks, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 763–772, http://dx.doi.org/10.1109/ICCV48922.2021.00082.

[30] J. Guibas, M. Mardani, Z. Li, A. Tao, A. Anandkumar, B. Catanzaro, Efficient token mixing for transformers via adaptive Fourier neural operators, in: The Tenth International Conference on Learning Representations, ICLR, 2022.

[31] T. Nguyen, M. Pham, T. Nguyen, K. Nguyen, S.J. Osher, N. Ho, FourierFormer: Transformer meets generalized Fourier integral theorem, in: NeurIPS, 2022.

[32] Z. Huang, Z. Zhang, C. Lan, Z. Zha, Y. Lu, B. Guo, Adaptive frequency filters as efficient global token mixers, 2023, http://dx.doi.org/10.48550/arXiv.2307.14008, CoRR abs/2307.14008.

[33] X. Li, Y. Zhang, J. Yuan, H. Lu, Y. Zhu, Discrete cosin TransFormer: Image modeling from frequency domain, in: IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023, 2023, pp. 5457–5467, http://dx.doi.org/10.1109/WACV56688.2023.00543.

[34] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, W. Wu, Incorporating convolution designs into visual transformers, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 559–568, http://dx.doi.org/10.1109/ICCV48922.2021.00062.

[35] T. Chen, Z. Zhang, Y. Cheng, A. Awadallah, Z. Wang, The principle of diversity: Training stronger vision transformers calls for reducing all levels of redundancy, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 12010–12020, http://dx.doi.org/10.1109/CVPR52688.2022.01171.

[36] Z. Pan, B. Zhuang, H. He, J. Liu, J. Cai, Less is more: Pay less attention in vision transformers, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI , Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI , the Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI, 2022, pp. 2035–2043, http://dx.doi.org/10.1609/aaai.v36i2.20099.

[37] Z. Pan, J. Cai, B. Zhuang, Fast vision transformers with HiLo attention, in: NeurIPS, 2022.

[38] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: IEEE International Conference on Computer Vision, ICCV, 2017, pp. 764–773, http://dx.doi.org/10.1109/ICCV.2017.89.

[39] J. Cooley, J. Tukey, An algorithm for the machine calculation of complex fourier series, Math. Comp. 19 (1965) 297–301.

[40] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proceedings of the 32nd International Conference on Machine Learning, ICML, in: JMLR Workshop and Conference Proceedings, Vol. 37, 2015, pp. 448–456.

[41] D. Hendrycks, K. Gimpel, Bridging nonlinearities and stochastic regularizers with Gaussian error linear units, 2016, CoRR abs/1606.08415.

[42] L.J. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, 2016, CoRR abs/1607.06450.

[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M.S. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252, http://dx.doi.org/10.1007/s11263-015-0816-y.

[44] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: 3rd International Conference on Learning Representations, ICLR, 2015.

[45] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: 6th International Conference on Learning Representations, ICLR, 2018.

[46] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, D. Song, Natural adversarial examples, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 15262–15271, http://dx.doi.org/10.1109/CVPR46437.2021.01501.

[47] R. Duan, Y. Chen, D. Niu, Y. Yang, A.K. Qin, Y. He, AdvDrop: Adversarial attack to DNNs by dropping information, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, 2021, pp. 7486–7495, http://dx.doi.org/10.1109/ICCV48922.2021.00741.

[48] A. Krizhevsky, G. Hinton, Learning Multiple Layers of Features from Tiny Images (Master's thesis), Department of Computer Science, University of Toronto, 2009.

[49] B. Recht, R. Roelofs, L. Schmidt, V. Shankar, Do ImageNet classifiers generalize to ImageNet? in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, ICML, Vol. 97, 2019, pp. 5389–5400.

[50] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: 7th International Conference on Learning Representations, ICLR, 2019.

[51] I. Loshchilov, F. Hutter, SGDR: stochastic gradient descent with warm restarts, in: 5th International Conference on Learning Representations, ICLR, 2017.

[52] H. Zhang, M. Cissé, Y.N. Dauphin, D. Lopez-Paz, Mixup: Beyond empirical risk minimization, in: 6th International Conference on Learning Representations, ICLR, 2018.

[53] S. Yun, D. Han, S. Chun, S.J. Oh, Y. Yoo, J. Choe, CutMix: Regularization strategy to train strong classifiers with localizable features, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV, 2019, pp. 6022–6031, http://dx.doi.org/10.1109/ICCV.2019.00612.

[54] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 2818–2826, http://dx.doi.org/10.1109/CVPR.2016.308.

[55] B.T. Polyak, A.B. Juditsky, Acceleration of stochastic approximation by averaging, SIAM J. Control Optim. 30 (4) (1992) 838–855.

[56] E. Hoffer, T. Ben-Nun, I. Hubara, N. Giladi, T. Hoefler, D. Soudry, Augment your batch: Improving generalization through instance repetition, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 8126–8135, http://dx.doi.org/10.1109/CVPR42600.2020.00815.

[57] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random erasing data augmentation, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI , the Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI , the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI, 2020, pp. 13001–13008, http://dx.doi.org/10.1609/aaai.v34i07.7000.

[58] G. Huang, Y. Sun, Z. Liu, D. Sedra, K.Q. Weinberger, Deep networks with stochastic depth, in: Computer Vision - ECCV 2016 - 14th European Conference, Vol. 9908, 2016, pp. 646–661, http://dx.doi.org/10.1007/978-3-319-46493-0_39.

[59] I. Radosavovic, R.P. Kosaraju, R.B. Girshick, K. He, P. Dollár, Designing network design spaces, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 10425–10433, http://dx.doi.org/10.1109/CVPR42600.2020.01044.

[60] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 2818–2826, http://dx.doi.org/10.1109/CVPR.2016.308.

[61] S. Xie, R.B. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 5987–5995, http://dx.doi.org/10.1109/CVPR.2017.634.

[62] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, Y. Wang, Transformer in transformer, in: Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems, 2021, pp. 15908–15919.

[63] M. Tan, Q.V. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: Proceedings of the 36th International Conference on Machine Learning, ICML, Vol. 97, 2019, pp. 6105–6114.

[64] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, G. Izacard, A. Joulin, G. Synnaeve, J. Verbeek, H. Jégou, ResMLP: Feedforward networks for image classification with data-efficient training, IEEE Trans. Pattern Anal. Mach. Intell. 45 (4) (2023) 5314–5321, http://dx.doi.org/10.1109/TPAMI.2022.3206148.