Discrete Cosine Transform-Based Decorrelated Attention for Vision Transformers

Hongyi Pan[1] , **Emadeldeen Hamdan**[2] , **Xin Zhu**[2] , **Ahmet Enis Cetin**[2] , **Ulas Bagci**[1]
[1]Machine and Hybrid Intelligence Lab, Northwestern University, Chicago, USA
[2]Department of Electrical and Computer Engineering, University of Illinois Chicago, Chicago, USA
{hongyi.pan, ulas.bagci}@northwestern.edu, {ehamda3, xzhu61, aecyy}@uic.edu,

Abstract

Self-attention is central to the success of Transformer architectures; however, learning the query, key, and value projections from random initialization remains challenging and computationally expensive. In this paper, we propose two complementary methods that leverage the Discrete Cosine Transform (DCT) to enhance the efficiency and performance of Vision Transformers. First, we address the initialization problem by introducing a simple yet effective DCT-based initialization strategy for self-attention, where projection weights are initialized using DCT coefficients. This structure-preserving approach consistently improves classification accuracy on the CIFAR-10 and ImageNet-1K benchmarks. Second, we propose a DCT-based attention compression technique that exploits the decorrelation properties of the frequency domain. By observing that high-frequency DCT coefficients typically correspond to noise, we truncate high-frequency components of the input patches, thereby reducing the dimensionality of the query, key, and value projections without sacrificing accuracy. Experiments on Swin Transformer models demonstrate that the proposed compression method achieves a substantial reduction in computational overhead while maintaining comparable performance. Code: https://github.com/NUBagciLab/DCT-Transformer.

1 Introduction

In the last decade, convolutional neural networks (CNNs) have dominated image-processing tasks. However, inspired by the success of transformer architectures in natural language processing, Vision Transformers (ViTs) have redefined the norms of visual data interpretation in recent years. Self-attention mechanism, a key component of transformers, is pivotal in reshaping how models perceive and analyze visual information [Vaswani *et al.*, 2017].

Unlike CNNs, ViTs view an image as a sequence of informative patches, adopting a more holistic approach. By treating image patches like "visual words", ViTs can leverage powerful transformer architectures, originally designed for natural language processing, to learn global context and long-range dependencies. This novel approach has led to remarkable performance gains on a wide range of computer vision tasks, demonstrating the potential of ViTs to revolutionize the field.

The initialization of self-attention weights is a critical step significantly impacting the training dynamics and performance of the Transformers. The core of Transformers is the *self-attention* mechanism, which includes three weight matrices: queries ($\mathbf{W}_Q$), keys ($\mathbf{W}_K$), and values ($\mathbf{W}_V$). Proper initialization of these matrices ensures that the model starts from a stable state, facilitating effective learning. The most common approach is to initialize the weights randomly, often using distributions like the normal or uniform distribution. For example, Xavier initialization [Glorot and Bengio, 2010] and He initialization [He *et al.*, 2015] are popular choices. They are designed to keep the scale of the gradients roughly the same in all layers. Although less common, in some cases, biases or certain parts of the weight matrices are initialized to zeros [Masood and Chandra, 2012; Zhao *et al.*, 2022]. This is less common for queries, keys, and values matrices, as it can lead to symmetry-breaking issues during training [Hu *et al.*, 2019a].

Efficiency is an ongoing concern in Transformers. While conventional initialization strategies are often suitable for generic applications, new algorithms are increasingly tailored to the specific characteristics and needs of Transformers. These advancements in initialization techniques contribute to the ongoing improvements in the performance and efficiency of Transformer-based models. Orthogonal initialization [Saxe *et al.*, 2013; Hu *et al.*, 2019b], sparse initialization [Martens and others, 2010], variance scaling [Zhang *et al.*, 2019], pre-trained initialization [Weiss *et al.*, 2016; Bozinovski, 2020; Xu *et al.*, 2023], layer-wise, domain-specific and attention-specific initialization [Trockman and Kolter, 2023] are some of the new trends in initialization approaches. In this study, we propose a new family of algorithms for initialization strategy in the intersection of attention and domain-specific approaches: *Discrete Cosine Transform-Based Decorrelated Attention*.

Can DCT in Vision Transformer be a game changer? DCT has been widely used in signal processing and data compression applications. It reduces the dimensionality of the input data while preserving its essential features. Transform-

ing the input data into the frequency domain helps capture the most important features of the input data. In this paper, we propose two methods that apply DCT to the attention for vision transformers. First, we initialize the weight matrices for the queries, keys, and values computation as the DCT matrix. Our DCT-based initialization covers the entire spectrum utilizing the DCT basis vectors, and each weight vector has a different initial bandwidth. As a comparison, to cover the entire spectrum in the classical initialization, all the weight matrices are initialized as random numbers due to their independent generation. These random numbers can be treated as white noise. All the weight initialization starts with the same spectra. Second, we introduce a DCT-based compressed attention for vision transformers. With our compression methodology, compared to the vanilla Swin Transformers, the number of parameters and computational overhead of the Swin Transformers are reduced. Meanwhile, the accuracy of the models is comparable.

2 Related Works

Vision Transformer Transformer architectures have merged as a dominant standard for natural language processing tasks [Vaswani *et al.*, 2017]. Vision Transformer (ViT) [Dosovitskiy *et al.*, 2020] built a bridge between language and vision. It shows that pure transformers can also be applied directly to sequences of image patches for classification tasks. In other words, it partitions images into patches and treats them as word sequences. Then, a shifted windowing scheme was proposed in Swin Transformer [Liu *et al.*, 2021] to obtain greater efficiency. This limits self-attention computation to non-overlapping local windows while allowing for cross-window connection. Later, to overcome the limitations of the heavyweights in ViTs, MobileViT [Mehta and Rastegari, 2021] combined the strengths of MobileNets [Sandler *et al.*, 2018] and ViTs. It is a lightweight and low-latency network for mobile vision tasks. Moreover, gSwin [Go and Tachibana, 2023] combined MLP-based architecture with Swin Transformer and attains superior accuracy with a smaller model size.

Frequency-based Neural Network Orthogonal transforms such as the Fourier transform [Chi *et al.*, 2020; Buchholz and Jug, 2022; Patro *et al.*, 2025] and Hadamard transform [Pan *et al.*, 2023] have been used to improve the performance of the neural networks. In the orthogonal transform family, DCT is a mathematical technique commonly used in signal processing and image and video compression. Compared to the Fourier transform, which is the standard frequency representation, DCT does not generate complex-valued numbers. Therefore, it is computationally more efficient. DCT-Former [Scribano *et al.*, 2023], for instance, applied DCT compression on the input sequence along the length channel before feeding it into the self-attention layers to reduce the computational overhead in the NLP problems. Furthermore, inspired by JPEG compression, DCFormer [Li *et al.*, 2023] highlighted the visually significant signals by leveraging the input information compression on its frequency domain representation. These two DCT-based works and our DCT-based compression all aim to reduce the computational overhead. Therefore, we mainly compare our DCT-based compression method with them. DCT-Former [Scribano *et al.*, 2023] is designed for the NLP transformers and takes DCT along the width and height of the input tensor patches, while our method is designed for the vision transformers and it compresses the tensor along the channel axis. DCFormer [Li *et al.*, 2023] averages DCT coefficients via average pooling, which does not take advantage of the property that DCT components in low-frequency bands contain more important information. On the contrary, our DCT-based compression method retains the low-frequency components very accurately and discards the unnecessary extremely high-frequency components corresponding to noise. Under the same input image size, compared to the vanilla transformer models, DCFormer reduces the FLOPs and retains the number of parameters while sacrificing accuracy. Our DCT-based compression method, on the other hand, saves both trainable parameters and FLOPs and obtains on-par or better performance in addition to other benefits that our DCT-based compression method has. Furthermore, we introduce a DCT-based initialization and we have not found any related initialization in the literature.

Self-Attention Initialization DS-Init [Zhang *et al.*, 2019] improves convergence in deep Transformers for NLP by reducing parameter variance during initialization and minimizing output variance of residual connections, facilitating better gradient flow. It pairs with a merged attention sublayer (MAtt) that combines average-based self-attention with encoder-decoder attention to reduce the computational cost. Additionally, Mimetic initialization [Trockman and Kolter, 2023] is a compute-free method that initializes the weights of self-attention layers by simulating the patterns found in pre-trained weights, resulting in improved Transformer performance. Moreover, weight selection [Xu *et al.*, 2023] enables the initialization of smaller models by selecting a subset of weights from a larger pre-trained model, facilitating efficient knowledge transfer and faster training. This method enhances small model performance, reduces training time, and integrates with knowledge distillation, making it particularly valuable for resource-constrained environments. Furthermore, structured initialization [Zheng *et al.*, 2024] tackles the challenge of training vision transformers on small-scale datasets by introducing a structured initialization strategy inspired by the architectural biases of Convolutional Neural Networks (CNNs). While these methods focus on mimicking pre-trained patterns, selecting weights from larger models, or using CNN-inspired biases, our DCT-based initialization utilizes well-established DCT matrices to initialize attention weights.

Efficient Self-Attention A variety of approaches have been proposed to reduce the memory and computational costs of dot-product attention, including linear-complexity attention mechanisms [Shen *et al.*, 2021], increasing attention heads as in Hydra Attention [Bolya *et al.*, 2022], and architectural rethinking exemplified by MetaFormer [Yu *et al.*, 2022], which shows that overall architecture often matters more than the specific token mixer. MicroViT [Setyawan *et al.*, 2025] further targets resource-constrained settings by introducing an

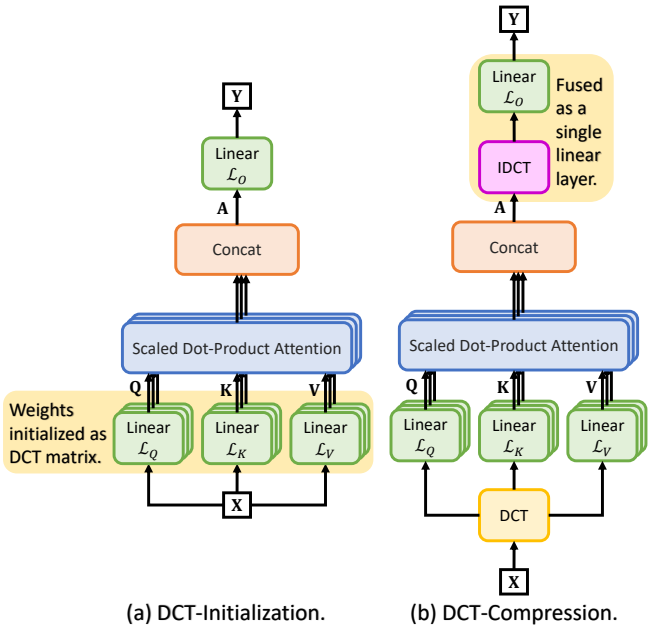


Figure 1: DCT-based initialization and compressed MSA.

Efficient Single Head Attention mechanism that reduces self-attention cost through group convolution and partial channel processing. Despite these advances, existing methods mainly pursue efficiency through linearization, head scaling, or architectural design, while largely ignoring frequency-domain transformations. We propose DCT-based methods to enhance both efficiency and expressiveness in vision transformers.

3 Methodology

3.1 Background: Multi-Head Self-Attention

To handle a 2D image tensor $\mathbf{I} \in \mathbb{R}^{H\times\ W\times C}$, the multi-head self-attention (MSA) module first splits it into patches $\mathbf{X} \in \mathbb{R}^{N\times M^2\times C}$, where $H \times W$ are the resolution of the image and C is the number of channels, $M \times M$ is the resolution of each image patch and $N = \frac{WH}{M^2}$ is the resulting number of patches. After that, three linear layers ($\mathcal{L}_Q$, $\mathcal{L}_K$, $\mathcal{L}_V$) are applied on $\mathbf{X}$ to compute the queries, keys, and values $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N\times M^2\times C}$:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q^T + \mathbf{b}_Q, \mathbf{K} = \mathbf{X}\mathbf{W}_K^T + \mathbf{b}_K, \mathbf{V} = \mathbf{X}\mathbf{W}_V^T + \mathbf{b}_V, \quad (1)$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{C\times C}$ and $\mathbf{b}_Q, \mathbf{b}_K, \mathbf{b}_V \in \mathbb{R}^{1\times C}$ are trainable weights and biases. Then, to perform P-head MSA, $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are split into P slices as $\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i \in \mathbb{R}^{N\times M^2\times d}$ for $i = 0, \ldots, P-1$, $d = \frac{C}{P}$. The scaled dot-product attention $\mathbf{A}_i \in \mathbb{R}^{N\times M^2\times d}$ is computed as:

$$\mathbf{A}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i) = \sigma\left(\frac{\mathbf{Q}_i\mathbf{K}_i^T}{\sqrt{d}} + \mathbf{B}_i\right)\mathbf{V}_i, \quad (2)$$

where $\sigma(\cdot)$ is the *Softmax* function. $\mathbf{B}_i \in \mathbb{R}^{M^2\times M^2}$ is the relative position bias. Since the relative position along each axis lies in $[-M+1, M-1]$, values in B_i are taken from a parameterized smaller-sized bias matrix $\hat{\mathbf{B}}_i \in \mathbb{R}^{(2M-1)\times(2M-1)}$. After that, $\{\mathbf{A}_i\}$ are concatenated to $\mathbf{A} \in \mathbb{R}^{N\times M^2\times C}$ to feed into a linear layer $\mathcal{L}_O$ to obtain the attention output $\mathbf{Y} \in \mathbb{R}^{N\times M^2\times C}$:

$$\mathbf{Y} = \mathbf{A}\mathbf{W}_O^T + \mathbf{b}_O. \quad (3)$$

Finally, $\mathbf{Y}$ is reversed into the image shape $\mathbf{J} \in \mathbb{R}^{H\times W\times C}$.

3.2 DCT-based Initialization for Attention

In the attention layer, the first three linear layers ($\mathcal{L}_Q$, $\mathcal{L}_K$, and $\mathcal{L}_V$) extract different features of the input patches $\mathbf{X}$. However, their weights $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ are trained simultaneously. Such training from scratch is difficult, especially when the dataset is insufficient and small-scale. On the other hand, DCT is a well-designed feature extractor. In our proposed DCT initialization strategy, one of $\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V \in \mathbb{R}^{C\times C}$ is initialized using the orthogonal type-II DCT matrix $\mathcal{D}$ [Ahmed *et al.*, 1974]:

$$\{\mathbf{X}\mathcal{D}^T\}[k] = \begin{cases} \frac{\sqrt{2}}{C}\sum_{n=0}^{C-1}\mathbf{X}[n], & k = 0, \\ \frac{2}{C}\sum_{n=0}^{C-1}\mathbf{X}[n]\cos\frac{(2n+1)k\pi}{2nC}, & k = 1, \ldots, C-1. \end{cases} \quad (4)$$

$\mathcal{D}$ can be obtained from applying DCT on an identity matrix $\mathbf{I}_{C\times C}$ because DCT is a linear transform. The remaining weight matrices and other trainable parameters are still initialized regularly. We don't initialize multiple weight matrices to avoid the weight symmetry problem [Hu *et al.*, 2019a] caused by initializing the multiple weights as the same values, and our ablation study has verified this point.

The DCT-based initialization can be treated as applying DCT instead of linear projection on $\mathbf{X}$ to obtain $\mathbf{Q}$, $\mathbf{K}$, or $\mathbf{V}$. Some DCT coefficients capture low-frequency information, while others capture high-frequency details. Therefore, the DCT matrix is a good starting point for the weight matrices.

3.3 DCT-Based Compressed Attention

Data compression using DCT is common in signal, image, and video processing for energy concentration, compression efficiency, and noise reduction. Usually, higher-frequency DCT coefficients contribute less to perceptual quality and contain more noise. Preserving essential information in lower frequencies while discarding less critical details in higher frequencies results in efficient signal representation and storage. The specific truncation strategy depends on the application's requirements and the acceptable level of information loss.

We observed that the attention layer matrices are correlated with each other, so that we can compress them using DCT. Fig. 1(b) shows the method of DCT-based compression for the attention layer. First, the input patches $\mathbf{X} \in \mathbb{R}^{N\times M^2\times C}$ are encoded by DCT along the channel of C with the high-frequency components truncated. If we keep the first τC coefficients ($\tau < 1$), the essential information of $\mathbf{X}$ is still preserved, while the input and output dimensions of the linear layers ($\mathcal{L}_Q, \mathcal{L}_K$, and $\mathcal{L}_V$) are reduced from $\mathbb{R}^{N\times M^2\times C}$ to $\mathbb{R}^{N\times M^2\times \tau C}$. In this way, the computational cost from $\mathcal{L}_Q, \mathcal{L}_K$, and $\mathcal{L}_V$ is reduced significantly. In our experiments,

Operation	Vanilla MSA	DCT-Compressed MSA
DCT	-	τNM^2C^3
$\mathcal{L}_{Q/K/V}$	$3NM^2C^3$	$3\tau^2NM^2C^3$
$\frac{\mathbf{QK}^T}{\sqrt{d}}+\mathbf{B}$	NM^4C^2	$\tau^2NM^4C^2$
Multiply $\mathbf{V}$	NM^4C^2	$\tau^2NM^4C^2$
$\mathcal{L}_O$	NM^2C^3	τNM^2C^3
Total	$4NM^2C^3$ $+2NM^4C^2$	$(2\tau+3\tau^2)NM^2C^3$ $+2\tau^2NM^4C^2$

Table 1: Multiplications in MSA on $\mathbf{X}\in\mathbb{R}^{N\times M^2\times C}$. $\tau<1$.

τ is chosen from 25%, 50%, and 75%. After obtaining the attention $\mathbf{A}\in\mathbb{R}^{N\times M^2\times\tau C}$ from the queries $\mathbf{Q}$, keys $\mathbf{K}$, and values $\mathbf{V}$, we apply zero-padding along the last channel to make the dimension back to $\mathbb{R}^{N\times M^2\times C}$. Before being fed into the linear layer $\mathcal{L}_O$, these padded patches are decoded by the inverse DCT (IDCT):

$$\{\mathbf{A}(\mathcal{D}^{-1})^T\}[n]=\frac{\sqrt{2}}{2}\mathbf{A}[0]+\sum_{k=1}^{C-1}\mathbf{A}[k]\cos\frac{(2n+1)k\pi}{2C}, n=0,1,\ldots,C-1.. \tag{5}$$

Because $\mathcal{D}$ is orthogonal, *i.e.* $\mathcal{D}\mathcal{D}^T=\mathbf{I}_{C\times C}$, the IDCT matrix can be obtained directly from the transpose of the DCT matrix. Although DCT and IDCT can be implemented in a fast manner with the complexity of $O(C\log_2 C)$ using the butterfly algorithm [Chen *et al.*, 1977], such an implementation is not officially supported in PyTorch currently. Therefore, in this work, we implement the DCT and IDCT via matrix multiplication between the input tensor and the truncated DCT and IDCT matrices $\bar{\mathcal{D}}\in\mathbb{R}^{\tau C\times C}$ and $\bar{\mathcal{D}}^{-1}\in\mathbb{R}^{C\times\tau C}$. These transform matrices are obtained by truncating the DCT and IDCT matrices along rows or columns to keep τ coefficients or zero padding, respectively. Then, DCT with truncation or IDCT with zero padding is implemented as one matrix multiplication to reduce memory pressure. Furthermore, as no non-linearity exists between IDCT and the final linear layer $\mathcal{L}_O$, these two steps can be implemented as a single linear layer whose weight matrix is $\mathbf{W}_O\bar{\mathcal{D}}^{-1}$ and the bias is still $\mathbf{b}_O$. However, such a combination is not applied on DCT and $\mathbf{W}_Q$, $\mathbf{W}_K$, or $\mathbf{W}_V$. This is because it may not always reduce the computational cost for them: To compute $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, with such a combination, $3kNM^2C^3$ multiplications are needed. In contrast, if they are computed separately, there are τNM^2C^3 multiplications from DCT and $\tau^2NM^2C^3$ multiplications from each linear layer. As a result, there are a total of $(\tau+3\tau^2)NM^2C^3$ multiplications. Hence, combing DCT with $\mathbf{W}_Q$, $\mathbf{W}_K$ and $\mathbf{W}_V$ brings computational cost saving only when $\tau>\frac{2}{3}$. To maintain the consistency of the framework, we do not combine DCT with $\mathbf{W}_Q$, $\mathbf{W}_K$, or $\mathbf{W}_V$.

Computational cost comparison of the vanilla versus the proposed DCT-compressed attentions is presented in Table 1. We evaluate both naive and simplified implementations of the DCT-compressed attention with the vanilla attention. The simplification is implemented by fusing IDCT into the linear layer $\mathcal{L}_O$. Computational overhead from the Softmax function is omitted because the Softmax function is based on exponential terms. Despite this, the Softmax function in the DCT-compressed attention requires less computational cost than in the vanilla attention as the number of entries in $\left(\frac{\mathbf{QK}^T}{\sqrt{d}}+\mathbf{B}\right)$ in the DCT-compressed attention is τ^2 times ($\tau<1$) as in the vanilla attention.

3.4 Frequency Domain Motivation of the DCT-based Initialization

In traditional random initialization, all the weight matrices are initialized as random numbers that are independently generated. As a result, the initial matrices can be considered as a realization of white noise covering the entire spectrum. A good initialization should take advantage of the full bandwidth to cover the entire spectrum of the input. The DCT-based initialization covers the entire frequency domain: The set of DCT basis vectors $\{\frac{\sqrt{2}}{2},\cos\frac{(2n+1)k\pi}{2C}\}$ is the class of discrete Chebyshev polynomials:

$$T_k[n]=\begin{cases}\frac{\sqrt{2}}{2}, & k=0,\\ \cos\frac{(2n+1)k\pi}{2C}, & k=1,\ldots,C-1,\end{cases} \tag{6}$$

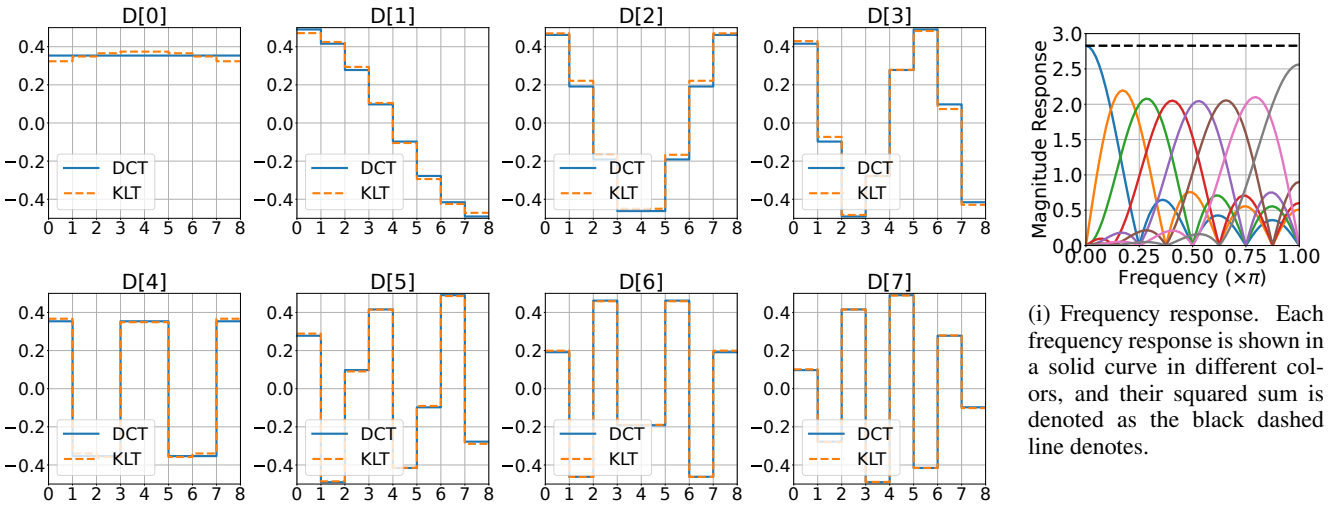
for $n=0,1,\ldots,C-1$. The basis member $\cos\frac{(2n+1)k\pi}{2nC}$ is the k-th Chebyshev polynomial $T_k[\xi]$ evaluated at the n-th zero of $T_n[\xi]$ [Ahmed *et al.*, 1974]. Fig. 2 draws the basis vectors in Eq. (6) for an 8×8 DCT matrix and the magnitude of their discrete Fourier transform ($\mathcal{F}\{\cdot\}$):

$$\hat{T}_l[k]=\mathcal{F}\{T_l[n]\}, l=0,1,\ldots,C-1. \tag{7}$$

The entire spectrum is covered as $\sqrt{\sum_{l=0}^{C-1}|\hat{T}_l[k]|^2}$ has perfect full-band coverage, and each weight vector in the DCT-based initialization covers a different bandwidth. On the other hand, in the traditional random initialization, all of the weights start with the same spectra. This is why our DCT-based initialization is superior to the traditional random initialization for the attention layer.

3.5 Motivation of the DCT-based Compression

The mechanism behind the DCT-based compression mimics the Karhunen-Loeve transform (KLT), which is also known as principal component analysis (PCA), a mathematical technique used for dimensionality reduction in signal processing [Ji *et al.*, 2022]. Neighboring image patches are highly correlated with each other. During training, the attention matrices are inter-correlated because they are obtained using similar gradient vectors. Therefore, they can be decorrelated and compressed using DCT. When the input data is highly correlated, the covariance matrix can be approximated by a Teoplitz matrix ϕ of $[1,\rho,\rho^2,\cdots,\rho^{C-1}]$ with $\rho<1$ close to 1 [Akansu and Torun, 2012]. This is the covariance matrix of an autoregressive model random process AR(1) with the correlation coefficient ρ [Testa and Magli, 2016]. Fig. 2 shows that DCT basis vectors approximate the KLT of a first-order Markov process with high correlation among the neighboring samples [Dony and others, 2001; Radünz *et al.*, 2022]. We observe that $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ have parameters with high correlations among themselves. We can directly use DCT in data compression since its matrix approximates the corresponding KLT matrix [Ahmed *et al.*,



(i) Frequency response. Each frequency response is shown in a solid curve in different colors, and their squared sum is denoted as the black dashed line denotes.

Figure 2: Basis vectors of an 8×8 DCT matrix and their frequency response. The DCT basis vectors provide a good approximation to the eigenvectors of the Toeplitz matrix with $\rho = 0.9$ (KLT) and cover the entire frequency spectrum.

1974]. Although KLT can also be utilized in data compression, it requires massive computation for the covariance matrices and the corresponding eigenvectors of $\mathbf{X}$. On the other hand, DCT is very successful in decorrelating the data without computing the covariance matrix and eigenvectors. As a consequence, it is also used in JPEG image [Wallace, 1991] and MPEG video [Ebrahimi and Horne, 2000] compression.

4 Experimental Results

We conduct experiments on the CIFAR-10 and ImageNet-1K classification tasks and COCO 2017 detection and segmentation tasks. Swin Transformers [Liu *et al.*, 2021] were employed as the backbone network. Models were trained on a server with 8 NVIDIA A100 80GB GPUs using Python with the PyTorch framework.

4.1 Implementation Details

Image Classification Training models for the image classification tasks from scratch mostly followed the settings in [Liu *et al.*, 2021]. We employed AdamW optimizer for 300 epochs using a cosine decay learning rate scheduler and 20 epochs of linear warm-up. We set the batch size as 1024, the initial learning rate as 0.001, and the weight decay as 0.05. Data augmentation was performed as follows: The CIFAR-10 augmentation followed [He *et al.*, 2016]. First, the training images were padded with 4 pixels. Then, they were randomly cropped to get 32 by 32. Finally, the images were randomly flipped horizontally. The images were normalized with the means of [0.4914, 0.4822, 0.4465] and the standard deviations of [0.2023, 0.1994, 02010]. The ImageNet-1K augmentation followed PyTorch Torchvision's GitHub repository [Paszke *et al.*, 2019]. As a start, images were resized to make the smaller edge of the image 232 for Swin-T and 246 for Swin-S. Next, they were cropped to 224 by 224 images and randomly flipped horizontally. Techniques such as Autoaugment [Cubuk *et al.*, 2018], mixup [Zhang *et al.*, 2018], cutmix [Yun *et al.*, 2019], and random erasing [Zhong *et al.*, 2020] were applied. Images were normalized with the means of [0.485, 0.456, 0.406] and the standard deviations of [0.229, 0.224, 0.225]. During the training, the best models were stored based on the top-1 accuracy on the CIFAR-10 test dataset and the ImageNet-1K validation dataset.

COCO Detection and Segmentation Training models for the COCO tasks followed the settings in [Liu *et al.*, 2021]. Models started from ImageNet-1K pre-trained weights. We applied AdamW optimizer for 36 epochs with an initial learning rate of 0.0001, weight decay of 0.05, batch size of 16, $3\times$ schedule, and multi-scale training.

4.2 Image Classification Benchmark

CIFAR-10 To demonstrate the benefits of using our DCT-based decorrelated attention on the small-scale dataset, we conduct Swin Transformer Tiny (Swin-T) from-scratch CIFAR-10 training. Table 2 presents the performance comparison on the CIFAR-10 test dataset, where the number of parameters only counts trainable parameters. The baseline vanilla Swin-T obtained 90.51% accuracy, and its accuracy was always improved by the DCT-based initialization regardless of which attention weight was initialized as the DCT matrix. When the weights to compute keys ($\mathbf{K}$) were initialized as the DCT matrices, the accuracy was improved to 90.83%. For DCT-compressed attention, when we only keep 25% of DCT coefficients, trainable parameters were reduced from 27.53M to 21.45M (22.1% off), while the accuracy only dropped 0.05%. If we kept more DCT coefficients, the accuracy even improved. 75% of DCT coefficients increased accuracy from 90.51% to 90.90%. This is because although the number of trainable parameters is reduced, transforming the input data into the frequency domain using DCT can boost the extraction of important features.

ImageNet-1K Table 3 presents the performance comparison on the ImageNet-1K dataset. Same as on the CIFAR-10

Model	Param(M)	Top-1 Acc
Swin-T	27.53	90.51%
Swin-T-DCT-Q	27.53	90.78%
Swin-T-DCT-K	27.53	90.83%
Swin-T-DCT-V	27.53	90.81%
Swin-T-DCT-0.25	21.45	90.46%
Swin-T-DCT-0.5	22.67	90.67%
Swin-T-DCT-0.75	24.69	90.90%

Table 2: CIFAR-10 test dataset evaluation. "-DCT-" followed by Q/K/V in model names indicates which attention projection weights are initialized with the DCT matrix, while numerical values denote the proportion of coefficients retained in DCT-compressed attention. Models with higher or lower top-1 accuracy are highlighted in pink or cyan, respectively.

dataset, DCT-based initialization consistently improves performance on both Swin-T and Swin-S models, regardless of which attention weight was initialized as the DCT matrix. For example, when we initialized the weights to compute keys ($\mathbf{K}$) as the DCT weights, the accuracy of Swin-T was improved from 81.474% to 81.728%, and Swin-S was increased from 83.196% to 83.378%, respectively. We then evaluated the DCT-compressed attention with different truncating rates and compared the results with DCFormer [Li *et al.*, 2023], a DCT-based approach designed to reduce computational cost. When 75% of DCT coefficients were kept, we obtained a comparable accuracy result on Swin-T and a slightly better accuracy result on Swin-S, while reducing the trainable parameters from 28.29M to 25.45M (13.1% off) and from 49.61M to 44.45M (10.4% off), respectively. Flops were reduced from 4.49G to 4.18G (6.9% off) for Swin-T and 8.13G to 8.74G (7.0% off) for Swin-S. Retaining fewer coefficients yields further savings at the cost of accuracy. In contrast, DCFormer-SW-T/S with $\tau = 2$ ($4\times$ frequency compression) substantially reduces FLOPs but suffers notable performance degradation, as it treats low- and high-frequency components equally. We further validated our methods on the ViT-B-32 backbone [Dosovitskiy *et al.*, 2020]. DCT-based initialization of the key projection improves accuracy from 73.120% to 73.614%. Applying DCT-compressed attention with 75% coefficient retention further improves accuracy to 73.226%, while reducing parameters from 88.22M to 75.83M (14.0%) and FLOPs from 4.41G to 4.05G (8.2%).

4.3 Ablation Study

Non-Trainable DCT Weights Initialization This study investigates the effectiveness of DCT weights as a principled starting point for attention projection initialization. Specifically, we fix the projection weights as DCT matrices, under which the computation of queries, keys, and values becomes equivalent to applying fixed shifts in the DCT representation of the input features. Table 4 shows that our method achieved accuracy comparable to the baseline on the CIFAR-10 dataset while requiring fewer trainable parameters. On the ImageNet-1K dataset, non-trainable DCT weights for computing $\mathbf{Q}$ or $\mathbf{K}$ led to only a marginal drop in top-1 accuracy (less than 0.3%). In this work, the DCT was implemented via matrix multiplication, incurring the same computational cost as standard attention; however, this overhead can be further reduced using butterfly-based DCT implementations.

Method	Param/Flops	Top-1/Top-5 Acc
Swin-T	28.29M/4.49G	81.474%/95.776%
PoolFormer-S24	21M/3.5G	80.3%/-
DCFormer-SW-T(τ=1)	28.29M/4.5G	81.2%/-
DCFormer-SW-T(τ=2)	28.29M/1.3G	79.2%/-
gSwin-T	22/3.6	81.715%/-
Swin-T-DCT-Q	28.29M/4.49G	81.568%/95.862%
Swin-T-DCT-K	28.29M/4.49G	81.728%/95.862%
Swin-T-DCT-V	28.29M/4.49G	81.528%/95.850%
Swin-T-DCT-0.25	22.21M/3.24G	79.580%/94.936%
Swin-T-DCT-0.5	23.43M/3.64G	80.762%/95.520%
Swin-T-DCT-0.75	25.45M/4.18G	81.474%/95.728%
Swin-S	49.61M/8.74G	83.196%/96.360%
PoolFormer-M36	56M/8.8G	82.1%/-
DCFormer-SW-S(τ=1)	49.61M/8.7G	82.8%/-
DCFormer-SW-S(τ=2)	49.61M/2.7G	80.9%/-
gSwin-S	40M/7.0G	83.014%/-
Swin-S-DCT-Q	49.61M/8.74G	83.298%/96.298%
Swin-S-DCT-K	49.61M/8.74G	83.378%/ 96.398%
Swin-S-DCT-V	49.61M/8.74G	83.200%/96.364%
Swin-S-DCT-0.25	38.54M/6.28G	81.840%/95.704%
Swin-S-DCT-0.5	40.76M/7.07G	82.622%/96.036%
Swin-S-DCT-0.75	44.45M/8.13G	83.202%/96.388%
ViT-B-32	88.22M/4.41G	73.120%/90.306%
ViT-B-32-DCT-Q	88.22M/4.41G	73.278%/90.574%
ViT-B-32-DCT-K	88.22M/4.41G	73.614%/90.870%
ViT-B-32-DCT-V	88.22M/4.41G	73.306%/90.502%
ViT-B-32-DCT-0.5	66.97M/3.51G	72.948%/90.390%
ViT-B-32-DCT-0.75	75.83M/4.05G	73.226%/90.642%

Table 3: ImageNet-1K validation dataset evaluation.

Multiple DCT Initializations This study shows why we only initialize one instead of multiple projection weights. Table 5 presents an ablation study on initializing multiple projection matrices with the DCT matrix. Although this approach still improved top-1 accuracy for Swin-T, the gains were reduced compared to initializing only a single matrix. For example, initializing both $\mathbf{W}_Q$ and $\mathbf{W}_K$ resulted in a top-1 accuracy of 81.540%, lower than 81.568% by initializing only $\mathbf{W}_Q$ and 81.728% by initializing only $\mathbf{W}_K$. This degradation was likely due to the weight symmetry problem [Hu *et al.*, 2019a], which often arises when multiple weight matrices are initialized with identical values. In the context of

DCT	CIFAR-10		ImageNet-1K	
	Param	Top-1 Acc	Param	Top-1/Top-5 Acc
-	27.53M	90.51%	28.29M	81.474%/95.776%
Q	25.37M	90.47%	26.13M	81.214%/95.674%
K	25.37M	90.64%	26.13M	81.302%/95.788%
V	25.37M	90.75%	26.13M	80.486%/95.236%
QK	23.21M	90.70%	23.98M	80.608%/95.446%
QV	23.21M	90.50%	23.98M	80.234%/95.208%
KV	23.21M	90.62%	23.98M	80.228%/95.276%

Table 4: Ablation study on Swin-T with non-trainable DCT weights.

Model	Param/Flops	Top-1/Top-5 Acc
Swin-T	28.29M/4.49G	81.474%/95.776%
Swin-T-DCT-Q	28.29M/4.49G	81.568%/95.862%
Swin-T-DCT-K	28.29M/4.49G	81.728%/95.862%
Swin-T-DCT-V	28.29M/4.49G	81.528%/95.850%
Swin-T-DCT-QK	28.29M/4.49G	81.540%/95.754%
Swin-T-DCT-QV	28.29M/4.49G	81.548%/95.748%
Swin-T-DCT-KV	28.29M/4.49G	81.536%/95.770%
Swin-S	49.61M/8.74G	83.196%/96.360%
Swin-S-DCT-Q	49.61M/8.74G	83.298%/96.298%
Swin-S-DCT-K	49.61M/8.74G	83.378%/96.398%
Swin-S-DCT-QK	49.61M/8.74G	83.208%/96.378%

Table 5: Ablation study on ImageNet-1K for multiple weights initialization as the DCT matrix.

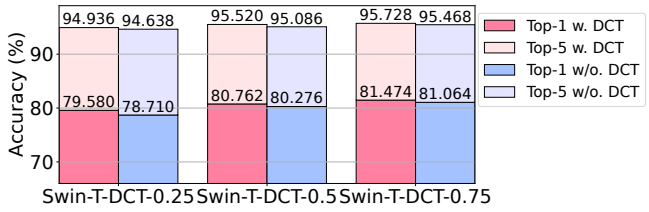


Figure 3: Ablation study on ImageNet-1K for removing DCT from DCT-compressed attention. Swin-T serves as the backbone model.

attention mechanisms, the query, key, and value projections play distinct roles in capturing relationships across the input sequence. If they are overly similar due to identical initialization, the model's ability to attend to relevant and diverse information is compromised. One of the key strengths of self-attention lies in its capacity to model complex dependencies between different positions in the input. Therefore, when $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ share the same initialization, this capacity is diminished, leading to reduced performance. We further verified this observation using Swin-S by applying the DCT-based initialization to both $\mathbf{W}_Q$ and $\mathbf{W}_K$. The results confirmed the same trend, supporting our conclusion that single-matrix DCT initialization is generally more effective.

Removing DCT in Compressed Attention This study discusses the necessity of DCT in the DCT-compressed attention. Specifically, removing DCT reduces the compression mechanism to a form of direct coefficient pruning in the spatial domain, without exploiting frequency-domain decorrelation. As Fig. 3 presents, when we removed the DCT in the DCT-compressed attention and retained other components, the accuracy of the revised Swin-T models on ImageNet-1K decreased across all compression ratios. These experiments show that DCT is essential because it transforms feature representations into the frequency domain, reducing redundancy and enhancing the model's ability to capture essential information. As a result, the attention mechanism benefits from improved expressiveness and robustness, even under compression. The accuracy drop observed in Fig. 3 confirms that DCT plays a crucial role in preserving discriminative features, demonstrating its necessity in the proposed approach.

Backbone	Param	Object Detection	Instance Segmentation
Swin-T	86M	50.3/69.1/54.3	43.7/ 66.6/47.3
DCFormer-T	86M	43.5/62.6/47.4	38.0/59.3/40.7
Swin-T-DCT-K	86M	50.4/69.2/54.3	43.9/66.6/47.6
Swin-T-DCT-0.75	83M	50.0/69.1/53.9	43.7/66.8/47.6
Swin-S	107M	51.8/70.4/56.3	44.7/67.9/ 48.5
DCFormer-S	107M	46.6/64.9/50.4	40.5/62.2/43.7
Swin-S-DCT-K	107M	52.4/70.8/57.3	45.1/68.3/49.1
Swin-S-DCT-0.75	102M	52.1/70.6/57.0	45.0/68.0/48.8

Table 6: COCO results. Results are AP^{Mask}/ $AP^{Mask}_{0.5}$/$AP^{Mask}_{0.75}$.

4.4 COCO 2017 Benchmark

We further evaluated the proposed methods on COCO 2017 object detection and instance segmentation tasks [Lin *et al.*, 2014]. We took Swin Transformers with ImageNet-1K pretrained weights as the backbone and the Cascade Mask R-CNN [Cai and Vasconcelos, 2018] as the pipeline. Based on the superior performance observed in the ImageNet-1K experiments when initializing $\mathbf{W}_K$ with the DCT matrix, we adopted this initialization for the Swin-T and Swin-S backbones in our evaluation. As shown in Table 6, both models achieved slightly higher average precision (AP) scores compared to their vanilla counterparts across both tasks. We also evaluated the DCT-compressed attention variant with the 0.75 compression ratio. This modification allowed Swin-T to maintain comparable AP performance while reducing the parameter count by approximately 3 million. It also enabled Swin-S to achieve improved AP scores with around 5 million fewer parameters.

5 Conclusion

In this work, we introduced two new methods using DCT to decorrelate the input in the attention for vision transformers. First, we presented a novel DCT-based initialization approach to address the challenges associated with training attention weights. We demonstrated a significant improvement over traditional initialization, providing a robust foundation for the attention mechanism. Second, we formulated a DCT-based compression method that can effectively reduce the computational overhead by truncating high-frequency components. It resulted in smaller weight matrices that compromised considerable accuracy. We validated the efficacy of these breakthroughs, showcasing their potential to enhance the performance and efficiency of Transformer models in classification tasks. These contributions underscore the importance of thoughtful initialization strategies and compression techniques in advancing the capabilities of the Vision Transformer architectures.

Acknowledgments

H. Pan and U. Bagci were supported by NIH grants R01-HL171376 and U01-CA268808. E. Hamdan, X. Zhu, and A. E. Cetin were supported by NSF 2531376.

References

[Ahmed *et al.*, 1974] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

[Akansu and Torun, 2012] Ali N Akansu and Mustafa U Torun. Toeplitz approximation to empirical correlation matrix of asset returns: A signal processing perspective. *IEEE Journal of Selected Topics in Signal Processing*, 6(4):319–326, 2012.

[Bolya *et al.*, 2022] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, and Judy Hoffman. Hydra attention: Efficient attention with many heads. In *European Conference on Computer Vision*, pages 35–49. Springer, 2022.

[Bozinovski, 2020] Stevo Bozinovski. Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica*, 44(3), 2020.

[Buchholz and Jug, 2022] Tim-Oliver Buchholz and Florian Jug. Fourier image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1846–1854, 2022.

[Cai and Vasconcelos, 2018] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[Chen *et al.*, 1977] Wen-Hsiung Chen, CH Smith, and Sam Fralick. A fast computational algorithm for the discrete cosine transform. *IEEE Transactions on communications*, 25(9):1004–1009, 1977.

[Chi *et al.*, 2020] Lu Chi, Borui Jiang, and Yadong Mu. Fast fourier convolution. *Advances in Neural Information Processing Systems*, 33:4479–4488, 2020.

[Cubuk *et al.*, 2018] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[Dony and others, 2001] R Dony et al. Karhunen-loeve transform. *The transform and data compression handbook*, 1(1-34):29, 2001.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[Ebrahimi and Horne, 2000] Touradj Ebrahimi and Caspar Horne. Mpeg-4 natural video coding–an overview. *Signal Processing: Image Communication*, 15(4-5):365–385, 2000.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[Go and Tachibana, 2023] Mocho Go and Hideyuki Tachibana. gswin: Gated mlp vision model with hierarchical structure of shifted window. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Hu *et al.*, 2019a] Shell Xu Hu, Sergey Zagoruyko, and Nikos Komodakis. Exploring weight symmetry in deep neural networks. *Computer Vision and Image Understanding*, 187:102786, 2019.

[Hu *et al.*, 2019b] Wei Hu, Lechao Xiao, and Jeffrey Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks. In *International Conference on Learning Representations*, 2019.

[Ji *et al.*, 2022] Xiaozhong Ji, Boyuan Jiang, Donghao Luo, Guangpin Tao, Wenqing Chu, Zhifeng Xie, Chengjie Wang, and Ying Tai. Colorformer: Image colorization via color memory assisted hybrid-attention transformer. In *European Conference on Computer Vision*, pages 20–36. Springer, 2022.

[Li *et al.*, 2023] Xinyu Li, Yanyi Zhang, Jianbo Yuan, Hanlin Lu, and Yibo Zhu. Discrete cosin transformer: Image modeling from frequency domain. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5468–5478, 2023.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[Liu *et al.*, 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[Martens and others, 2010] James Martens et al. Deep learning via hessian-free optimization. In *ICML*, volume 27, pages 735–742, 2010.

[Masood and Chandra, 2012] Sarfaraz Masood and Pravin Chandra. Training neural network with zero weight ini-

tialization. In *Proceedings of the CUBE International Information Technology Conference*, pages 235–239, 2012.

[Mehta and Rastegari, 2021] Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations*, 2021.

[Pan *et al.*, 2023] Hongyi Pan, Xin Zhu, Salih Furkan Atici, and Ahmet Cetin. A hybrid quantum-classical approach based on the hadamard transform for the convolutional layer. In *International Conference on Machine Learning*, pages 26891–26903. PMLR, 2023.

[Paszke *et al.*, 2019] 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. *Advances in neural information processing systems*, 32, 2019.

[Patro *et al.*, 2025] Badri N Patro, Vinay P Namboodiri, and Vijay S Agneeswaran. Spectformer: Frequency and attention is what you need in a vision transformer. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 9543–9554. IEEE, 2025.

[Radünz *et al.*, 2022] Anabeth P Radünz, Fábio M Bayer, and Renato J Cintra. Low-complexity rounded klt approximation for image compression. *Journal of Real-Time Image Processing*, pages 1–11, 2022.

[Sandler *et al.*, 2018] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[Saxe *et al.*, 2013] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[Scribano *et al.*, 2023] Carmelo Scribano, Giorgia Franchini, Marco Prato, and Marko Bertogna. Dct-former: Efficient self-attention with discrete cosine transform. *Journal of Scientific Computing*, 94(3):67, 2023.

[Setyawan *et al.*, 2025] Novendra Setyawan, Chi-Chia Sun, Mao-Hsiu Hsu, Wen-Kai Kuo, and Jun-Wei Hsieh. Microvit: a vision transformer with low complexity self attention for edge device. In *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2025.

[Shen *et al.*, 2021] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3531–3539, 2021.

[Testa and Magli, 2016] Matteo Testa and Enrico Magli. Compressive estimation and imaging based on autoregressive models. *IEEE Transactions on Image Processing*, 25(11):5077–5087, 2016.

[Trockman and Kolter, 2023] Asher Trockman and J Zico Kolter. Mimetic initialization of self-attention layers. *arXiv preprint arXiv:2305.09828*, 2023.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wallace, 1991] Gregory K Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.

[Weiss *et al.*, 2016] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[Xu *et al.*, 2023] Zhiqiu Xu, Yanjie Chen, Kirill Vishniakov, Yida Yin, Zhiqiang Shen, Trevor Darrell, Lingjie Liu, and Zhuang Liu. Initializing models with larger ones. In *The Twelfth International Conference on Learning Representations*, 2023.

[Yu *et al.*, 2022] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10819–10829, 2022.

[Yun *et al.*, 2019] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[Zhang *et al.*, 2018] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[Zhang *et al.*, 2019] Biao Zhang, Ivan Titov, and Rico Sennrich. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 898–909, 2019.

[Zhao *et al.*, 2022] Jiawei Zhao, Florian Tobias Schaefer, and Anima Anandkumar. Zero initialization: Initializing neural networks with only zeros and ones. *Transactions on Machine Learning Research*, 2022.

[Zheng *et al.*, 2024] Jianqiao Zheng, Xueqian Li, and Simon Lucey. Structured initialization for attention in vision transformers. *arXiv preprint arXiv:2404.01139*, 2024.

[Zhong *et al.*, 2020] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.