Rotary Position Embedding for Vision Transformer

Byeongho Heo Song Park Dongyoon Han Sangdoo Yun

NAVER AI Lab

Abstract. Rotary Position Embedding (RoPE) performs remarkably on language models, especially for length extrapolation of Transformers. However, the impacts of RoPE on computer vision domains have been underexplored, even though RoPE appears capable of enhancing Vision Transformer (ViT) performance in a way similar to the language domain. This study provides a comprehensive analysis of RoPE when applied to ViTs, utilizing practical implementations of RoPE for 2D vision data. The analysis reveals that RoPE demonstrates impressive extrapolation performance, i.e., maintaining precision while increasing image resolution at inference. It eventually leads to performance improvement for ImageNet-1k, COCO detection, and ADE-20k segmentation. We believe this study provides thorough guidelines to apply RoPE into ViT, promising improved backbone performance with minimal extra computational overhead. Our code and pre-trained models are available at `https://github.com/naver-ai/rope-vit`

1 Introduction

Transformers [34] have become popular due to their strong performance across various tasks in language and computer vision domains [5, 6]. The transformer treats input data as a sequence of tokens. The tokens equally interact with others through a self-attention mechanism [34]. Since the self-attention mechanism is independent of the token index or positions (*i.e.*, permutation invariance), the transformer requires additional position information, usually injected by position embedding [5, 23, 27, 34]. The position embeddings give the position information to input tokens with specific embedding designed for the transformer. They uniquely differentiate tokens based on their locations rather than their contents. Thus, the position information of self-attention heavily depends on the position embedding, which is a crucial component in designing transformer architectures.

There are two primary methods in position embedding for Vision Transformers: Absolute Positional Embedding (APE) [5,6] and Relative Position Bias (RPB) [17, 23, 27]. APE utilizes the absolute position of tokens for position embedding through sinusoidal or learnable embedding. Otherwise, RPB enables relative positions between tokens by adding relative biases to the attention matrix of the self-attention layers. In general, APE is used for traditional ViT architecture [6], and RPE is preferred to hierarchical ViT like Swin Transformer [17].

Although both position embeddings are effective for the transformer on fixed-resolution settings, they struggle with resolution changes, requiring flexibility and extrapolation in position embeddings. Considering that the resolution of pre-training is usually smaller than that of downstream dense prediction, it might degrade ViT performance in various applications, such as multi-resolution recognition, object detection, and segmentation.

This paper aims to improve position embedding for vision transformers by applying an extended Rotary Position Embedding (RoPE) [29]. RoPE is a relative position embedding that is specially designed for extrapolation in language domains. Despite the remarkable success of RoPE in Large-Language Models [12, 26, 33], its effectiveness in vision tasks has not been validated due to limited investigation. In this paper, we provide a comprehensive investigation of RoPE for transformers in vision recognition tasks. Our investigation starts with 1D to 2D expansion of RoPE to cope with images rather than original language inputs. Although 2D RoPE using axial frequencies was used in pioneer works [7, 18, 19], we argue that it lacks the ability to handle diagonal directions, which are preferred in convolution networks by the square kernel. To cope with the diagonal direction of RoPE, we propose to use mixed axis frequencies for 2D RoPE, named RoPE-Mixed. Since RoPE-Mixed uses frequencies for both axes as learnable network parameters, it effectively handles diagonal direction and is more suitable for ViT's attention than Axial 2D RoPE.

In experiments, we apply variants of 2D RoPE to representative transformer architectures, ViT and Swin Transformer, and validate the effects of 2D RoPE in various tasks, including multi-resolution classification on ImageNet-1k [4], object detection on MS-COCO [16], and semantic segmentation on ADE20k [40,41]. The results show that 2D RoPE is a beneficial option for position embedding in transformers with impressive performance improvements on high-resolution images, i.e., extrapolation of images. We believe our study demonstrates the significant impact of 2D RoPE in vision domains and contributes to future research by suggesting a beneficial option in position embedding for vision transformers.

2 Related Works

Position embedding. ViT [6] introduces a transformer [34] architecture for visual inputs, employing Absolute Positional Embedding (APE) [5,6]. APE with learnable parameters effectively injects spatial positions of each token to be used for the self-attention mechanism. Hierarchical ViT such as Swin Transformer [17] increase the spatial length of tokens at early layers using pooling. To handle a large number of tokens with limited position embeddings, Relative Position Bias (RPB) [17, 23, 27] is preferred by the hierarchical ViTs. Studies have been conducted to improve position embedding for ViT based on these two major position embeddings. iRPE [36] proposes an improved RPB by applying relative position embedding as multiplication with query vector. CPE [3] finds that a convolution network can effectively inject relative position information to tokens and utilizes 3×3 depth-wise convolution [10] as conditional position embedding.

LaPE [38] shows that simple scaling with adaptive layer-norm can improve the positional embedding of various networks.

RoPE in vision modeling. Pioneering studies introduced RoPE to ViT-related architectures. Hybrid X-former [11] applies 1D RoPE to ViT variants named Vision X-formers; it is the first attempt at the application of RoPE in ViT to our knowledge. However, 1D RoPE is insufficient to demonstrate performance, and evaluation is limited to small datasets such as CIFAR [13] and Tiny ImageNet [14]. EVA-02 [7] introduces 2D Axial RoPE to a new language-aligned vision model EVA-02, like CLIP [22]. Unified-IO 2 [18] uses 2D RoPE for new multi-modal modeling; 2D Axial RoPE is applied to non-text modalities, including vision, audio, and video. In diffusion modeling [25], FiT [19] applies 2D Axial RoPE for their new diffusion model. In these studies, 2D Axial RoPE was used to improve new model performance on language-related or generation tasks, which differs from our goal of challenging classification, detection, and segmentation tasks. Exploring the impacts of 2D RoPE implementations in basic architectures with general training recipes could benefit diverse vision researchers.

Multi-resolution inference. Unlike ConvNets [8], ViT [6] requires a transformation in position embedding for multi-resolution inference. Some studies investigated a multi-resolution inference method for ViT. CAPE [15] analyzes ViT's position embedding in resolution changes and finds that augmenting position embedding improves the multi-resolution performance of ViT. Thus, they propose a new training recipe that includes continuous augmenting of position embedding (CAPE). ResFormer [30] shows that relative position embedding based on depth-wise convolution layer benefits multi-resolution inference. Using this property, the study proposes an improved ViT architecture with global and local depth-wise conv embedding. It substantially improves multi-resolution performance with multi-resolution self-distillation learning recipes. In contrast to conventional multi-resolution, FlexiViT [1] proposes a ViT with flexible patch sizes that can replace multi-resolution inference. In FlexiViT, ViT increases the patch size instead of increasing input resolution. By training with a multi-patch-size training scheme and distillation using ViT-B/8 [28], FlexiViT exhibits remarkable performance for various patch-size, which corresponds to multi-resolution in computation cost aspect.

These studies require special training methods, which make them difficult to combine with other training recipes, potentially reducing general applicability. RoPE improves multi-resolution performance while using existing training recipes as is, offering generally applicable and easy-to-use compared to others.

3 Method

Rotary Position Embedding (RoPE) [29] was introduced to apply to key and query in self-attention layers as channel-wise multiplications, which is distinct from conventional position embeddings - APE is added to the stem layer; RPB is added to an attention matrix. We first present conventional position embeddings, including RoPE [29] in language model at §3.1, and provide feasible expansion of RoPE to 2D inputs for transformers in the vision domain in subsequent §3.2. In §3.3, we describe the characteristics of RoPE compared to other position embedding and analysis for 2D RoPE.

3.1 Preliminary: Introducing Position Embeddings

Absolute Positional Embedding (APE) [5, 6, 34] is the most common position embedding for Vision Transformer (ViT). APE is generally added to the feature right after the patchification layer computes tokens from 16 × 16 or 32 × 32 patch images. For patch tokens $\mathbf{x}_0 \in \mathbb{R}^{N\times d}$, APE $\mathbf{E}_{APE} \in \mathbb{R}^{N\times d}$ gives the position information for each token by addition:

$$\mathbf{x}'_0 = \mathbf{x}_0 + \mathbf{E}_{APE}. \tag{1}$$

The tokens with APE $\mathbf{x}'_0$ are fed to transformer blocks and utilized as a feature merged with the absolute positional information. There are two variants on how to build $\mathbf{E}_{APE}$: sinusoidal and learnable embedding. Sinusoidal embedding uses axial sinusoidal functions as APE. When APE for position $\mathbf{p}_n = (p^x_n, p^y_n)$ is denoted as $\mathbf{E}_{APE}(\mathbf{p}_n) \in \mathbb{R}^d$, t-th dim of sinusoidal embedding $\mathbf{E}_{APE}(\mathbf{p}_n, t)$ is

$$\begin{aligned}\mathbf{E}_{APE}(\mathbf{p}_n, 4t) = \sin(p^x_n/10^{4t/\lfloor \frac{d}{4}\rfloor}),\ \mathbf{E}_{APE}(\mathbf{p}_n, 4t+1) = \cos(p^x_n/10^{4t/\lfloor \frac{d}{4}\rfloor}), \quad (2)\\ \mathbf{E}_{APE}(\mathbf{p}_n, 4t+2) = \sin(p^y_n/10^{4t/\lfloor \frac{d}{4}\rfloor}),\ \mathbf{E}_{APE}(\mathbf{p}_n, 4t+3) = \cos(p^y_n/10^{4t/\lfloor \frac{d}{4}\rfloor}).\end{aligned}$$

Note that we use 0-base numbers for indexes p^x_n, p^y_n, and t. The other implementation of APE is to use learnable parameters and train them with the training process. $N \times d$ learnable parameters are randomly initialized and are used as Eq. 1. It is the simplest way for APE, and supervised learning recipes use APE with learnable parameters [6, 31, 32]. Since learnable APE is commonly used for ViT, we refer to it as the default option for APE.

Relative Position Bias (RPB) [17, 23] is a popular way to inject relative distances to the ViT architectures. APE is not suitable for handling tokens based on their relative positions, as it relies solely on absolute positions in the image $\mathbf{p}_n = (p^x_n, p^y_n)$. It is necessary to use different types of position embedding that utilize relative positions $\tilde{\mathbf{p}}_{nm} = (\tilde{p}^x_{nm}, \tilde{p}^y_{nm}) = (p^x_n - p^x_m, p^y_n - p^y_m)$. RPB is widely used relative position embedding for ViT. In contrast to learnable APE, which has learnable parameters for each absolute position, RPB uses learnable parameters for each relative position. i.e., Relative Position Bias (RPB) table T is defined as learnable parameters for every possible relative position:

$$T = \{T_{\tilde{p}^x \tilde{p}^y} \in \mathbb{R} \mid \tilde{p}^x \in \{-W, \ldots, 0, \ldots, W\}, \tilde{p}^y \in \{-H, \ldots, 0, \ldots, H\}\}. \tag{3}$$

While APE is added to network features, RPB is directly applied to the attention matrix of every self-attention layer since it is the only position that can handle relative relations in transformer architecture. The attention matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ with the query and key of a head denoted by $\mathbf{q}, \mathbf{k} \in \mathbb{R}^{N\times d_{head}}$, is calculated

$$\mathbf{A} = \mathrm{SoftMax}(\mathbf{q}\mathbf{k}^T/\sqrt{d_{head}}). \tag{4}$$

To fit with the attention matrix, the RPB table $T \in \mathbb{R}^{2W \times 2H}$ is rearranged to RPB embedding $\mathbf{E}_{RPB} \in \mathbb{R}^{N \times N}$ where (n, m)-th component $\mathbf{E}_{nm}^{RPB}$ is

$$\mathbf{E}_{nm}^{RPB} = T_{\tilde{p}_{nm}^{x} \tilde{p}_{nm}^{y}} = T_{(p_n^x - p_m^x)(p_n^y - p_m^y)}. \tag{5}$$

Then, RPB is added to the attention matrix in Eq. 4 as

$$\mathbf{A} = \mathrm{SoftMax}(\mathbf{q}\mathbf{k}^T / \sqrt{d_{head}}) + \mathbf{E}_{RPB}. \tag{6}$$

By RPB, self-attention handles relative positions. Note that we describe RPB for a head in a multi-head self-attention layer. Thus, in practice, RPB parameters and addition are repeated for each head in multi-head attention.

Rotary Position Embedding (RoPE) [29] is a recent method in the line of relative position embedding studies. Although RPB delivers relative position to the attention, simple addition as bias may limit interaction with attention weights, which causes limited utilization of relative position. Thus, RoFormer [29] proposes a novel relative position embedding method: Rotary Position Embedding (RoPE). Note that this section explains the original RoPE designed for language modeling. We will explain our RoPE for 2D images in §3.2

Limitations of RPB emerge from the addition to the attention matrix. Since RPB is applied to the attention matrix after query-key multiplication, it cannot affect and contribute to the query-key similarity, which is the core operation of self-attention. To resolve this limitation, RoPE introduces the multiplication of Euler's formula ($e^{i\theta}$) to key and query vectors as relative position embedding. i.e., when n, m-th query and key is $\mathbf{q}_n, \mathbf{k}_m \in \mathbb{R}^{1 \times d_{head}}$, RoPE is applied as

$$\mathbf{q}'_n = \mathbf{q}_n e^{in\theta},\ \mathbf{k}'_m = \mathbf{k}_m e^{im\theta}. \tag{7}$$

Then, (n, m)-th component of attention matrix is calculated as

$$\mathbf{A}'_{(n,m)} = \mathrm{Re}[\mathbf{q}'_n \mathbf{k}'^{*}_m] = \mathrm{Re}[\mathbf{q}_n \mathbf{k}^{*}_m e^{i(n-m)\theta}], \tag{8}$$

where $\mathrm{Re}[\cdot]$ denotes real part of complex number and * means complex conjugates. By multiplying complex rotation $e^{i\theta n}, e^{i\theta m}$ depending on token position (n, m), RoPE injects relative positions $(n - m)$ to the attention matrix in rotation form. In practical implementation, RoPE converts $\mathbf{q}_n, \mathbf{k}_m \in \mathbb{R}^{1 \times d_{head}}$ to complex vector $\bar{\mathbf{q}}_n, \bar{\mathbf{k}}_m \in \mathbb{C}^{1 \times (d_{head}/2)}$ by considers $(2t)$-th dim as real part and $(2t + 1)$-th dim as imaginary part. It produces the same attention value as $\mathbf{q}_n \mathbf{k}_m^T = \mathrm{Re}[\bar{\mathbf{q}}_n \bar{\mathbf{k}}_m^*]$ but reduces computational wastes. Also, RoPE utilizes multiple frequencies θ_t using channel dimensions of key and query as

$$\theta_t = 10000^{-t/(d_{head}/2)},\ \text{where}\, t \in \{0, 1, ..., d_{head}/2\}. \tag{9}$$

In summary, a rotation matrix $\mathbf{R} \in \mathbb{C}^{N \times (d_{head}/2)}$ is defined as

$$\mathbf{R}(n, t) = e^{i\theta_t n} \tag{10}$$

and applied to query and key with the Hadamard product $\circ$ as

$$\bar{\mathbf{q}}' = \bar{\mathbf{q}} \circ \mathbf{R}, \quad \bar{\mathbf{k}}' = \bar{\mathbf{k}} \circ \mathbf{R}, \quad \mathbf{A}' = \mathrm{Re}[\bar{\mathbf{q}}' \bar{\mathbf{k}}'^{*}]. \tag{11}$$

Note that the attention matrix with RoPE $\mathbf{A}'$ implies relative position in rotation form $e^{i(n-m)\theta_t}$ for $(d_{head}/2)$ number of frequencies, which gives a lot of performance beneficial to the transformer, especially for extrapolation on inference stage based on periodic functions.

3.2 RoPE for 2D images

RoPE exhibits remarkable performance in the language domain. However, only a few studies have explored using RoPE in the vision domain with 2D input, as it was designed solely for 1D input. This section introduces feasible implementations of 2D RoPE for input images: axial and learnable frequency.

Axial frequency. A typical way to expand 1D position embedding to 2D is repeating 1D operation for each axis. Similar to 2D sinusoidal embedding in Eq. 2, axial frequency is to divide embedding dimensions into two and apply position embedding for the x-axis and y-axis separately. It is straightforward because it is technically the same as repeating 1D embedding twice.

First, we need to change the 1D token index n in RoPE to a 2D token position $\mathbf{p}_n = (p^x_n, p^y_n)$ where $p^x_n \in \{0, 1, ..., W\}$, $p^y_n \in \{0, 1, ..., H\}$ for token width W and height H. Thus, the rotation matrix $\mathbf{R} \in \mathbb{C}^{N \times (d_{head}/2)}$ in Eq. 10 is changed as

$$\mathbf{R}(n, 2t) = e^{i\theta_t p^x_n}, \ \ \mathbf{R}(n, 2t+1) = e^{i\theta_t p^y_n}. \tag{12}$$

Also, the range of position indexes (p^x_n, p^y_n) is reduced by square root. It is natural to reduce RoPE frequencies θ_t in Eq. 9 by square root as

$$\theta_t = 100^{-t/(d_{head}/4)}, \ \text{where}\, t \in \{0, 1, ..., d_{head}/4\}. \tag{13}$$

Note that θ_t for vision is often larger than that of language, and the number of frequencies is halved to cover both (x, y) dimensions with d_{head} as well. This axial frequency has been used in a few pioneering works [7, 18, 19] to further improve the performance of a new ViT architecture.

Mixed learnable frequency. The axial frequency is a simple but effective way to expand RoPE for the vision domain. However, it is unable to handle diagonal directions since the frequencies only depend on a single axis. RoPE injects relative positions in the form of Euler's formula $(e^{i\theta_t(n-m)})$. Thus, with axial frequencies, the relative positions are applied as axial directions $e^{i\theta_t(p^x_n - p^x_m)}$ or $e^{i\theta_t(p^y_n - p^y_m)}$, which cannot be converted to mixed frequency $e^{i(\theta^x_t \tilde{p}^x_{nm} + \theta^y_t \tilde{p}^y_{nm})}$. In the case of sinusoidal APE in Eq. 2, the sinusoidal functions can be mixed with another axis through query-key multiplication in the self-attention layer. However, RoPE already spends query-key multiplication for position subtraction for relative distance. There is no way to mix axial frequencies for diagonal direction.

We conjecture that it might degrade RoPE's potential performance and make sub-optimal axial frequency choices in the vision domain.

To handle mixed frequencies, we propose to use a rotation matrix in Eq. 10 in mixed axis form as

$$\mathbf{R}(n,t) = e^{i(\theta_t^x p_n^x + \theta_t^y p_n^y)}. \tag{14}$$

By using two frequencies for each axis, RoPE allows handling of the diagonal axis. The RoPE attention matrix in Eq. 8 is changed by mixed frequency as

$$\mathbf{A}'_{(n,m)} = \mathrm{Re}[\mathbf{q}_n \mathbf{k}_m^* e^{i(\theta_t^x (p_n^x - p_m^x) + \theta_t^y (p_n^y - p_m^y))}]. \tag{15}$$

This formulation is identical to the axial frequency implementation as θ_t^x or θ_t^y goes to zero. Thus, mixed frequency RoPE is a generalized version of axial frequency RoPE. Different from fixed frequencies in language RoPE and axial frequency, we let the network learn frequencies (θ_t^x, θ_t^y) for $t \in \{0, 1, ..., d_{head}/2\}$ as learnable parameters. Our mixed learnable frequency implementation enables diagonal direction handling to RoPE and makes RoPE learnable, like conventional positional embedding in the vision domain. Like RPB, we use separate sets of learnable frequencies for each head and every self-attention layer. It produces d learnable parameters per self-attention layer. However, it is negligible since it requires only $\sim$0.01% of network parameters in ViT-B.

3.3 Discussion

2D Fourier analysis. We design a 2D Fourier analysis to demonstrate the representational difference between RoPE-Axial and RoPE-Mixed. When all 2D frequencies are utilized, a 2D Fast Fourier Transform (FFT) followed by an inverse Fast Fourier Transform (iFFT) perfectly reconstructs the input. However, the number of RoPE frequencies is limited to $\frac{d_{head}}{2}$ as in Eq. 9. $\frac{d_{head}}{2}$ (= 32 for ViT-B) frequencies are insufficient to cover all 2D frequencies, resulting in imperfect reconstructions. This imperfect reconstruction reflects the expressiveness and representation pattern of the frequencies. In Fig. 1, we compare 2D FFT-iFFT results of RoPE-Axial and RoPE-Mixed frequencies. Note that we use RoPE-Mixed frequencies from ViT-B trained on ImageNet-1k. The results show a significant difference: Axial frequencies exhibit artifacts along axial lines, impairing precise positional representation, whereas Mixed frequencies utilize diverse 2D frequencies to produce sharper locations. Thus, we claim that the mixed frequencies are necessary for precise localization in the attention, which explains why RoPE-Mixed performs better than RoPE-Axial in §4.

On image resolution changes. Vision models use diverse image resolutions depending on the goal of target tasks. For example, image classification uses 224 × 224 as the standard resolution for comparison but utilizes small resolutions [32, 35] for training efficiency and enlarges resolutions to boost the performance additionally. Furthermore, object detection and segmentation prefer larger resolutions to capture small objects. Thus, transformers for vision should support resolution changes, which is linked to the necessity of resolution change

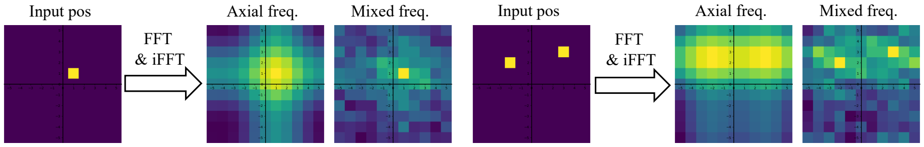

Fig. 1: 2D Fourier reconstruction with RoPE frequencies. We perform a Fast Fourier Transform (FFT) followed by an inverse FFT with only RoPE frequencies to evaluate the representation capabilities of RoPE frequencies

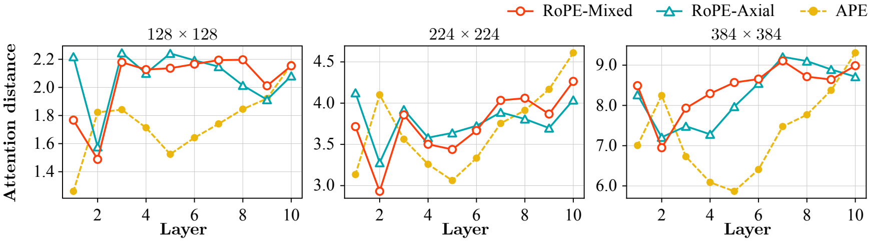

Fig. 2: Attention distances of ViT-B for APE/RoPE. We measure the average distance of attention interaction by computing the distance between query-key tokens from attention probabilities. We average the distance across the validation set.

in position embedding. RoPE makes an extended position embedding based on sinusoidal function for large resolution. Different from zero-padding in RPB, the rotation matrix in Eq. 12 and Eq. 14 can produce values for extended positions since it is based on periodic functions, which has proven its effectiveness for extrapolation [26, 33]. We expect that the advantage of RoPE in extrapolation will also be effective for multi-resolution benchmark in §4.1 and dense prediction tasks in §4.2 and §4.3.

Phase shift in RoPE. In sinusoidal representation, phase shift such as ϕ in $e^{i(m-n)\theta+i\phi}$ is an important ability to control activation area. This phase shift ability is already included in $\mathbf{W}_q$ and $\mathbf{W}_k$ of the self-attention layer. Based on Eq. 8, when we apply $e^{i(m-n)\theta+i\phi}$ and $\mathbf{q}_n = \mathbf{x}_n\mathbf{W}_q$, the equation is

$$\mathbf{x}_n\mathbf{W}_q e^{i(n-m)\theta+i\phi}\mathbf{k}_m^* = \mathbf{x}_n\mathbf{W}_q e^{i\phi}\mathbf{k}_m^* e^{i(n-m)\theta} = \mathbf{x}_n\mathbf{W}_q'\mathbf{k}_m^* e^{i(n-m)\theta}. \tag{16}$$

Thus, RoPE does not need additional parameters for phase shift ϕ since learnable parameters $\mathbf{W}_q$ and $\mathbf{W}_k$ can do the same role in network training.

Analyzing attention. We analyze the attention matrix of RoPE ViT compared to the ViT with APE. Following attention analysis in literature [9, 20], we measure attention distances and entropy on the ImageNet-1k validation set with various resolutions. Attention distance refers to the average spatial distance involved in attention interaction. Attention entropy represents the entropy values of attention probabilities, indicating the sharpness of attention. The averaged attention distances are shown in Fig. 2. In training resolution 224 × 224, RoPEs increase attention distance at the middle layers but decrease it in the second and later layers. In other resolutions, the pattern is similar, but the difference is

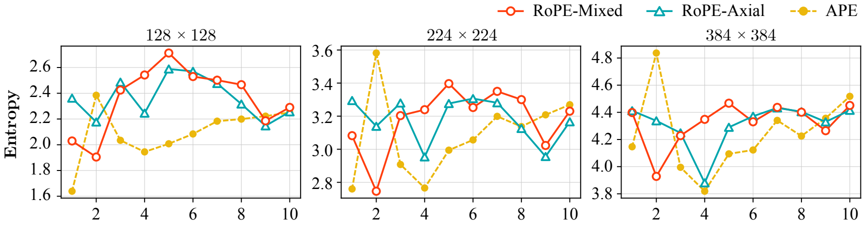

Fig. 3: Entropy of attention in ViT-B with APE or RoPE. Entropy of attention probability is measured for every self-attention of ViT-B. A high entropy value indicates that a large number of tokens are involved in the attention interaction.

more significant than the training resolution. In short, RoPEs increase attention distance at the middle layers, which becomes substantial at resolution changes. The entropy results are reported in Fig. 3. Interestingly, the pattern is similar to attention distance. Entropy of RoPE is larger than that of APE at the middle layers. These analysis results imply that RoPE makes attention interact with long-range (attention distance) and various tokens (entropy). We speculate that these differences in attention contributed to the performance improvement of RoPE observed in §4.

Computation costs. Although RoPE has an involved formulation compared with APE and RPB, its computation cost is negligible to the overall computation. The rotation matrix in Eq. 12 and 14 is pre-computed before inference. The Hadamard product in Eq. 11 is the only computation required for inference - 1.8M FLOPs for ViT-B and accounts for only 0.01% of ViT-B's 17.6G FLOPs.

4 Experiments

We apply 2D RoPE to two representative ViT architectures: ViT [6] and Swin Transformer [17]. Note that ViT uses APE, whereas Swin Transformer uses RPB. Thus, our experiment can verify the performance of RoPE when it replaces APE or RPB. RoPE in ViT and Swin Transformer is validated for image recognition, including multi-resolution classification (§4.1) on ImageNet-1k [4], object detection (§4.2) on MS-COCO [16], and semantic segmentation (§4.3) on ADE20k [40, 41]. We compare the conventional position embeddings (APE, RPB) with two variants of 2D RoPE RoPE-Axial (Eq. 12) and RoPE-Mixed (Eq. 14). Our experiments will exhibit the remarkable performance of 2D RoPE across all tasks, particularly with a significant margin in extrapolation.

4.1 Multi-resolution classification

Robustness on multi-resolution inputs is an essential factor of ViT performance, as it is closely related to their downstream performance in dense prediction tasks. In language models [12, 26, 33], RoPE exhibited strong extrapolation performance, i.e., text sequence longer than training samples. 2D RoPE might be

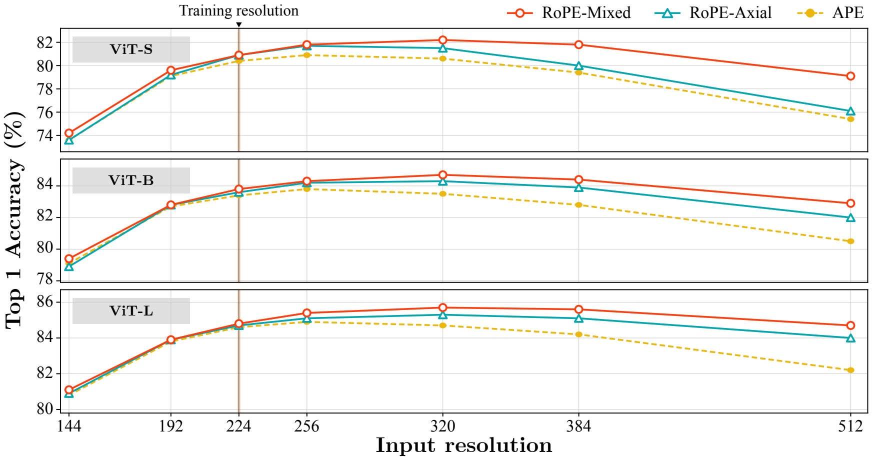

Fig. 4: Multi-resolution performance of ViTs. We apply two variants of 2D RoPE, RoPE-Axial, and RoPE-Mixed, to the ViT architectures. All ViTs are trained on ImageNet-1k [4] with DeiT-III [32]'s 400 epochs training recipe.

suitable for large-resolution images, leveraging its extrapolation capabilities. We train ViTs and Swin Transformers on ImageNet-1k [4] training set with high-performance training recipes [17,32]. We report the accuracy on the ImageNet-1k validation set as varying image sizes. Note that we use the ImageNet-1k standard image resolution 224 × 224 for training. Thus, a resolution larger than 224 can be considered as extrapolation.

Vision Transformer (ViT). We apply 2D RoPE to ViT-S, ViT-B, and ViT-L. We train ViT with a strong supervised learning training recipe for ImageNet-1k, DeiT-III 400 epochs training recipe. When applying RoPE to ViT, we remove APE from ViT by default. Thus, 2D RoPE is the only position embedding for RoPE ViT. We denote ViT uses both RoPE and APE as RoPE+APE.

In Fig. 4, we compare 2D RoPE variants with APE for ViT position embedding. Both 2D RoPE, RoPE-Axial, and RoPE-Mixed implementations outperform APE for resolutions larger than 224, i.e., extrapolation cases. As expected, the strong extrapolation performance of RoPE can be extended to image recognition tasks. In comparison between RoPE-Axial and RoPE-Mixed, RoPE-Mixed performs better than RoPE-Axial in all input resolutions, meaning learnable frequencies for mixed axes are beneficial for classification.

We measure the performance of RoPE-Mixed when it is used with APE. The left side of Fig. 6 shows the performance of RoPE-Mixed with APE (RoPE-Mixed + APE) compared to RoPE-Mixed and APE. Note that we report accuracy improvement over APE for RoPE models to improve visualization. When used with RoPE, APE is beneficial for interpolation (res $<$ 224) but reduces improvement on extrapolation (res $>$ 224). RoPE+APE is almost double the improvement of RoPE-Mixed in interpolation, while the disadvantage in extrapolation is comparably small. Thus, RoPE+APE is a considerable choice for applying RoPE to ViT-based architectures on the target resolution of the tasks.

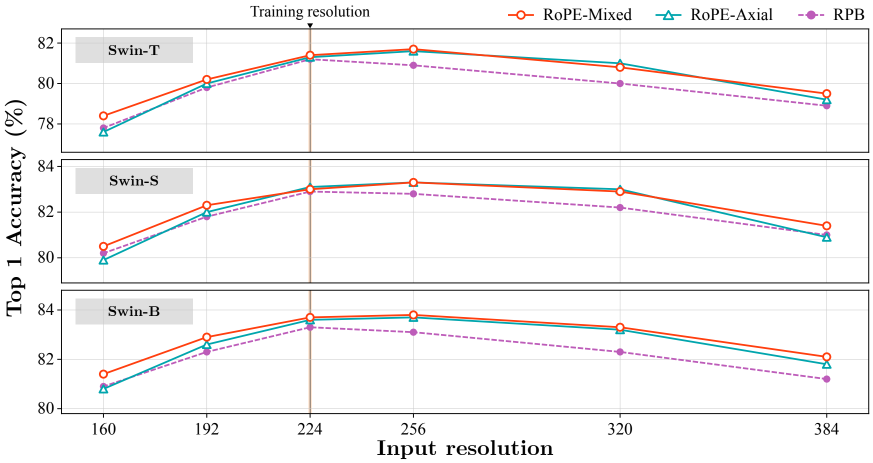

Fig. 5: Multi-resolution performance of Swin Transformers. We replace RPB in Swin Transformers with 2D RoPE variants: RoPE-Axial and RoPE-Mixed. Various Swin Transformers are trained with their 300 epochs training recipe [17]. For multi-resolution inference, we change the window size of the window attention.

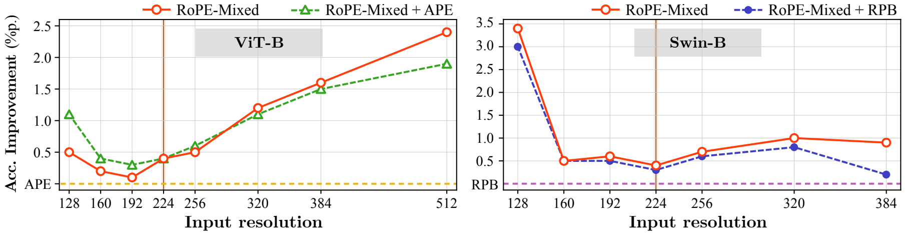

Fig. 6: RoPE with conventional position embedding. We report multi-resolution performance when RoPE is used with conventional position embeddings: APE and RPB. Performance improvement over baseline is reported to improve visualization.

Swin Transformer 2D RoPE variants are applied to Swin Transformers, a milestone work in hierarchical ViT with relative position embedding RPB. The experiment in Swin Transformer investigates whether RoPE can replace RPB or work efficiently in a hierarchical ViT. We train Swin-T, Swin-S, and Swin-B on ImageNet-1k with 300 epochs of Swin Transformer training recipe [17]. Similar to ViT, we replace RPB with 2D RoPE for comparison. Thus, RoPE Swin (*i.e.* Swin Transformer armed with RoPE) does not use RPB by default. A Swin Transformer using both position embedding is dubbed RoPE+RPE.

Fig 5 shows the multi-resolution performance of various Swin Transformers with different position embeddings. Two variants of 2D RoPE show remarkable performance improvements for extrapolation cases (res $>$ 224). Even in interpolation (res $<$ 224), RoPE-Mixed outperforms RPB by a large margin. It means that RoPE-Mixed is a more suitable option than RPB for Swin Transformers. When comparing RoPE-Mixed with RoPE-Axial, RoPE-Mixed outperforms in most resolutions. RoPE-Axial is especially weak in interpolation and significant extrapolation (res$=$ 384) cases.

Table 1: MS-COCO object detection with DINO-ViTDet. The table shows MS-COCO [16] detection performance (box AP). DINO [39] is trained with DINO-ViTDet 12 epochs setting [24]. RoPE is applied to the backbone ViT, which is pre-trained on ImageNet-1k with DeiT-III 400epochs recipe.

Backbone	APE	RoPE			
		Axial	Mixed	Axial+APE	Mixed+APE
ViT-B	49.4	50.8(+1.4)	**51.2(+1.8)**	50.7(+1.3)	51.1(+1.7)
ViT-L	51.1	52.2(+1.1)	**52.9(+1.8)**	52.5(+1.4)	52.8(+1.7)

Table 2: MS-COCO object detection with DINO-Swin. MS-COCO [16] detection performance (box AP) is reported for Swin Transformer with RoPE. DINO [39] is trained with DINO Swin 12 epochs setting [24]. Swin Transformers with RoPE or RPE are pre-trained on ImageNet-1k with Swin Transformer 300epochs recipe.

Backbone	RPB	RoPE			
		Axial	Mixed	Axial+RPB	Mixed+RPB
Swin-T	51.3	51.6(+0.3)	**51.8(+0.5)**	51.7(+0.4)	51.6(+0.3)
Swin-S	53.0	53.1(+0.1)	53.3(+0.3)	53.5(+0.5)	**53.6(+0.6)**
Swin-B	54.2	54.4(+0.2)	54.5(+0.3)	**54.7(+0.5)**	54.5(+0.3)

We also measure performance when RoPE-Mixed is used together with RPB. The right side of Fig. 6 shows the results. Different from RoPE+APE in ViT, RoPE+RPB has no performance advantage compared to RoPE-Mixed in all resolutions. This implies that RoPE-Mixed effectively replaces RPB as a relative position embedding. Note that the gap between RoPE-Mixed and RoPE+RPB is significant when input resolution is far different from training resolution, demonstrating the advantage of RoPE-Mixed on resolution changes.

4.2 Object detection

We verify 2D RoPE in object detection on MS-COCO [16]. DINO [39] detector is trained using ViT and Swin as backbone network. We use ImageNet-1k weights from §4.1 for pre-trained weights, and RoPE is only applied to the backbone. We use Detrex [24] codebase for detection training. DINO-ViTDet 12 epochs setting and DINO-Swin 12 epochs setting are used for DINO training.

Table 1 shows the DINO-ViTDet results in bounding box AP. We report four variants of RoPEs: Axial, Mixed, Axial+APE, and Mixed+APE; all demonstrate remarkable performance improvements. DINO-ViTDet achieves AP improvement of more than +1.0pp by changing positional embedding to RoPE. Among RoPE variants, RoPE-Mixed shows the best improvement at +1.8pp. AP in ViT-B and ViT-L. DINO-ViTDet uses ViT backbone with window-block attention, but still, a few layers remain as global attention. We believe that RoPE is highly effective due to the extrapolation in global attention.

The performance of DINO-Swin is reported in Table 2. Like DINO-ViTDet, four variants are reported: Axial, Mixed, Axial+RPB, and Mixed+RPB. RoPE

Table 3: ADE20k semantic segmentation using the UperNet [37] head. UperNet is trained with ViT backbone following ViT training recipe [21]. The table reports performance as mIoU metric. We report single-scale and multi-scale evaluation results.

	Multi-scale	APE	RoPE			
			Axial	Mixed	Axial+APE	Mixed+APE
ViT-B	-	47.7	49.0(+1.3)	49.6(+1.9)	49.5(+1.8)	**50.0(+2.3)**
	✔	48.4	49.9(+1.5)	50.7(+2.3)	50.5(+2.1)	**50.9(+2.5)**
ViT-L	-	50.8	51.8(+1.0)	51.5(+0.7)	51.6(+0.8)	**52.0(+1.2)**
	✔	51.6	**52.6(+1.0)**	52.3(+0.7)	52.4(+0.8)	**52.6(+1.0)**

Table 4: ADE20k semantic segmentation with Swin-Mask2Former [2]. Mask2Former model for semantic segmentation is trained using Swin Transformer. The table shows segmentation performance in mIoU metric.

Backbone	RPB	RoPE			
		Axial	Mixed	Axial+RPB	Mixed+RPB
Swin-S	50.2	50.4(+0.2)	51.1(+0.9)	**51.2(+1.0)**	50.9(+0.7)
Swin-B	51.5	**52.0(+0.5)**	**52.0(+0.5)**	50.0(-1.5)	51.4(-0.1)

outperforms RPB for all variants. RoPE-Mixed performs better than RoPE-Axial. +RPB is beneficial for Axial but has limited effect on Mixed. Performance improvement is smaller than DINO-ViTDet since DINO-Swin maintains a window size of the pre-trained backbone, i.e., DINO-Swin has no extrapolation. However, RoPE achieves meaningful gains and has room for improvement by increasing the Swin Transformer's window size for the detection backbone.

4.3 Semantic segmentation

We train 2D RoPE ViT and Swin for semantic segmentation on ADE20k [40, 41]. For ViT, we use UperNet [37] with ViT training recipe [21]. For Swin, Mask2Former [2] for segmentation is used with the Swin. ImageNet-1k pre-trained weights from §4.1 are used for pre-trained weights. Also, RoPE is only applied to the backbone. The networks are trained for 160k iterations.

Table 3 shows ViT-UperNet performances. RoPE-based models achieve impressive performance improvement in all cases. It is noteworthy that Mixed+APE achieves +2.3 and +2.5 mIoU improvement with only position embedding changes. The improvement might originate from the extrapolation performance of RoPE since the ViT-UperNet setting uses 512 × 512 images for inputs. Among the three variants of RoPE, Mixed+APE shows the best performance in all cases, which is different from detection results. As shown in Fig. 6, Mixed+APE has an advantage at interpolation while degrading performance at extrapolation. These results suggest that the use of APE in a RoPE-based ViT should be adjusted based on the target task. Swin-Mask2Former performances are shown in Table 4. RoPE also improves the performance of Swin-based segmentation. RoPE-Mixed shows impressive performance, while +RPB is only beneficial in limited cases.

Table 5: Multi-resolution comparison with ResFormer [30]. The table shows a comparison of RoPE-Mixed based ViTs with ResFormer-S trained for 224 × 224 resolution. RoPE ViT outperforms ResFormer on extrapolation, resolution > 224, and shows comparable performance at small resolutions.

Test resolution	96	128	160	192	224	288	384	448	512
ResFormer-S	57.8	**71.4**	**77.0**	**79.6**	80.8	81.4	80.7	79.3	77.7
ViT-S	35.4	69.3	76.1	79.1	80.4	80.9	79.4	77.6	75.4
+ RoPE-Mixed	55.7	70.6	76.6	**79.6**	**80.9**	82.0	**81.8**	**80.9**	**79.1**
+ RoPE-M + APE	**58.5**	**71.4**	76.7	79.5	**80.9**	**82.3**	81.7	80.5	78.5

4.4 Comparison with multi-resolution methods

We compare 2D RoPE variants with recent ViT architecture designed for multi-resolution inference, namely ResFormer [30]. ResFormer uses depth-wise convolutions as the position embedding. It uses sinusoidal APE in Eq. 2 and depth-wise convolution after the patch-embed layer as Global Position Embedding (GPE). Also, another depth-wise convolution is used similar to skip-connection for every self-attention layer to add position embed as Local Position Embed (LPE). Using GPE and LPE, ResFormer is proposed as an improved ViT for multi-resolution inference. ResFormer is trained with multi-resolution training utilizing self-distillation loss. Since self-distillation with multi-resolution training is not a common recipe in ViT, we use ResFormer-S trained with fixed resolution 224 × 224 and compare it with RoPE-Mixed ViT-S in §4.1. Table 5 shows a multi-resolution comparison of RoPE-Mixed with ResFormer-S-224. RoPE-Mixed outperforms ResFormer with a meaningful margin for extrapolation ranges (res > 224), but RoPE-Mixed shows performance lower than ResFormer for significant interpolation ranges (res $\leq$ 160). To achieve comparable interpolation, RoPE-Mixed needs additional APE. Overall, the results show that RoPE-Mixed+APE outperforms ResFormer-S in multi-resolution inference.

5 Conclusion

Rotary Position Embedding (RoPE) is a novel method for relative position embedding with a lot of potential. However, it has been underexplored in vision modeling. In this paper, we have conducted a comprehensive investigation of 2D RoPE for Vision Transformer (ViT) and proposed an improved 2D RoPE, RoPE-Mixed, utilizing mixed axis frequency with learnable parameters. Our experiments show that 2D RoPE is an effective solution for multi-resolution classification for both ViT and Swin Transformers, particularly for large resolutions. 2D RoPE shows improved performance with a significant margin in downstream tasks, such as object detection and semantic segmentation. It is noteworthy that our RoPE-Mixed outperforms conventional 2D RoPE in various tasks, further enhancing the contribution of this research. We believe that our study will be useful for vision researchers looking for state-of-the-art performance by suggesting 2D RoPE as a solution for them.

References

1. Beyer, L., Izmailov, P., Kolesnikov, A., Caron, M., Kornblith, S., Zhai, X., Minderer, M., Tschannen, M., Alabdulmohsin, I., Pavetic, F.: Flexivit: One model for all patch sizes. In: CVPR. pp. 14496–14506 (2023)
2. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation (2022)
3. Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., Shen, C.: Conditional positional encodings for vision transformers. arXiv preprint arXiv:2102.10882 (2021)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255. Ieee (2009)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
7. Fang, Y., Sun, Q., Wang, X., Huang, T., Wang, X., Cao, Y.: Eva-02: A visual representation for neon genesis. arXiv preprint arXiv:2303.11331 (2023)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
9. Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J.: Rethinking spatial dimensions of vision transformers. In: ICCV. pp. 11936–11945 (2021)
10. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
11. Jeevan, P., Sethi, A.: Resource-efficient hybrid x-formers for vision. In: WACV. pp. 2982–2990 (2022)
12. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.d.l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al.: Mistral 7b. arXiv preprint arXiv:2310.06825 (2023)
13. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
14. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N **7**(7), 3 (2015)
15. Likhomanenko, T., Xu, Q., Synnaeve, G., Collobert, R., Rogozhnikov, A.: Cape: Encoding relative positions with continuous augmented positional embeddings. NeruIPS **34**, 16079–16092 (2021)
16. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755. Springer (2014)
17. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp. 10012–10022 (2021)
18. Lu, J., Clark, C., Lee, S., Zhang, Z., Khosla, S., Marten, R., Hoiem, D., Kembhavi, A.: Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action. arXiv preprint arXiv:2312.17172 (2023)
19. Lu, Z., Wang, Z., Huang, D., Wu, C., Liu, X., Ouyang, W., Bai, L.: Fit: Flexible vision transformer for diffusion model. arXiv preprint arXiv:2402.12376 (2024)

20. Park, N., Kim, W., Heo, B., Kim, T., Yun, S.: What do self-supervised vision transformers learn? arXiv preprint arXiv:2305.00729 (2023)
21. Peng, Z., Dong, L., Bao, H., Ye, Q., Wei, F.: Beit v2: Masked image modeling with vector-quantized visual tokenizers. arXiv preprint arXiv:2208.06366 (2022)
22. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763. PMLR (2021)
23. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR **21**(1), 5485–5551 (2020)
24. Ren, T., Liu, S., Li, F., Zhang, H., Zeng, A., Yang, J., Liao, X., Jia, D., Li, H., Cao, H., Wang, J., Zeng, Z., Qi, X., Yuan, Y., Yang, J., Zhang, L.: detrex: Benchmarking detection transformers (2023)
25. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
26. Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X.E., Adi, Y., Liu, J., Remez, T., Rapin, J., et al.: Code llama: Open foundation models for code. arXiv preprint arXiv:2308.12950 (2023)
27. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. In: ACL (2018)
28. Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., Beyer, L.: How to train your vit? data, augmentation, and regularization in vision transformers. arXiv preprint arXiv:2106.10270 (2021)
29. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
30. Tian, R., Wu, Z., Dai, Q., Hu, H., Qiao, Y., Jiang, Y.G.: Resformer: Scaling vits with multi-resolution training. In: CVPR. pp. 22721–22731 (2023)
31. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML. pp. 10347–10357. PMLR (2021)
32. Touvron, H., Cord, M., Jégou, H.: Deit iii: Revenge of the vit. In: ECCV. pp. 516–533. Springer (2022)
33. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
34. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeruIPS **30** (2017)
35. Wightman, R., Touvron, H., Jégou, H.: Resnet strikes back: An improved training procedure in timm. arXiv preprint arXiv:2110.00476 (2021)
36. Wu, K., Peng, H., Chen, M., Fu, J., Chao, H.: Rethinking and improving relative position encoding for vision transformer. In: ICCV. pp. 10033–10041 (2021)
37. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: ECCV. pp. 418–434 (2018)
38. Yu, R., Wang, Z., Wang, Y., Li, K., Liu, C., Duan, H., Ji, X., Chen, J.: Lape: Layer-adaptive position embedding for vision transformers with independent layer normalization. In: ICCV. pp. 5886–5896 (2023)
39. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection (2022)
40. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: CVPR. pp. 633–641 (2017)

41. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. IJCV **127**, 302–321 (2019)