Analysis and Design of Positional Encoding Methods in Vision Transformers

Kyuho Lee[1] Euntae Kim[1] A hyun Jung[2] Jeongwon Lee[2]

[1]Korea University, department of Computer Science and Engineering
[2]Korea University, department of Biomedical Engineering

A. Introduction

Vision Transformer (ViT) excels at capturing global dependencies across an image via the self-attention mechanism. however, its computation is inherently permutation-invariant, preventing it from distinguishing the order of input tokens [1]. As a result, without explicit positional information (Positional Encoding, PE), ViT cannot learn the spatial arrangement or the inherent 2D structure of images. To address this issue, approaches such as 2D sinusoidal PE and RoPE (Rotary Positional Embedding) [2] are widely adopted. however, these methods typically interpret images using a simple(x, y) grid coordinate system. Such formulations are limited in capturing diagonal relationships or complex geometric structures within an image, and prior studies have noted that they overlook the center bias characteristic of natural images, where primary objects tend to appear near the image center [3,4].

In this work, inspired by the radial processing characteristics of the human visual system and the center bias inherent in natural image distributions, we propose Polar RoPE, a positional encoding method that overcomes the limitations of existing grid-coordinate–based approaches. The proposed method decomposes the position of each image patch into its radial distance(r)and angular direction(θ)from the image center, and applies rotary transformations by independently encoding each component within dedicated subspaces of the embedding dimensions. This design induces ViT to internalize the center–periphery structure and directional cues, providing a strong positional inductive bias aligned with real-world data distributions. Through CIFAR-10 image classification experiments, we demonstrate that Polar RoPE achieves faster convergence and superior generalization performance compared to existing methods, and further analyze the contribution of each component $(r, \theta)$to the overall improvement.

B. Methods

B.1. Baseline PE method

Sinusoidal positional encoding represents absolute position pos using sine and cosine functions, thereby injecting order information into the Transformer. Because the self-attention architecture does not directly utilize the sequence order [1], positional information must be added to the token embeddings in vector form. To construct Sinusoidal PE, for each position pos, a set of $(\sin, \cos)$ pairs at multiple frequencies is generated by applying sine functions to the even dimensions and cosine functions to the odd dimensions.

$$\begin{aligned} PE(\text{pos}, 2i) &= \sin\left(\frac{\text{pos}}{10000^{2i/d}}\right), \\ PE(\text{pos}, 2i+1) &= \cos\left(\frac{\text{pos}}{10000^{2i/d}}\right). \end{aligned} \tag{1}$$

By representing each position pos as a set of 2D vectors with different frequencies, Sinusoidal PE simultaneously captures long-range positional variation through low-frequency components and local positional variation through high-frequency components. For any given frequency, changes in position pos correspond to a rotation in the embedding space. At a specific frequency $\omega = 10000^{-2i/d}$, the embedding vector for position pos can be written as follows.

$$v(\text{pos}) = \begin{bmatrix} \sin(\text{pos}\,\omega) \\ \cos(\text{pos}\,\omega) \end{bmatrix} \tag{2}$$

If pos increases by 1, the corresponding embedding vector can be expressed as follows.

$$v(\text{pos}+1) = \begin{bmatrix} \sin\left((\text{pos}+1)\,\omega\right) \\ \cos\left((\text{pos}+1)\,\omega\right) \end{bmatrix} \tag{3}$$

Applying the addition identities to each component gives:

$$\begin{aligned} \sin\left((\text{pos}+1)\,\omega\right) &= \sin(\text{pos}\,\omega)\cos(\omega) + \cos(\text{pos}\,\omega)\sin(\omega), \\ \cos\left((\text{pos}+1)\,\omega\right) &= \cos(\text{pos}\,\omega)\cos(\omega) - \sin(\text{pos}\,\omega)\sin(\omega). \end{aligned} \tag{4}$$

Therefore, we obtain the following relationship between pos+1 and pos.

$$v(\text{pos}+1) = R(\omega)\,v(\text{pos}) \tag{5}$$

Here, R denotes the rotation matrix.

$$R(\omega) = \begin{bmatrix} \cos(\omega) & \sin(\omega) \\ -\sin(\omega) & \cos(\omega) \end{bmatrix}$$

Thus, for a given frequency, each increase in position corresponds to applying the same rotation matrix, allowing the positional shift from any previous location to be expressed as successive powers of this rotation matrix.

$$v(\text{pos}+k) = R(\omega)^k\,v(\text{pos}) \tag{6}$$

In self-attention, the query and key vectors are produced by applying linear projections to the sum of the token embedding and the positional embedding.

Because the attention score includes the inner product between the query at position p and the key at position q, expanding the expression shows that the final term captures the positional interaction between the two locations.

$$(W_Q PE(p)) \cdot (W_K PE(q)) \tag{7}$$

This approach has the advantage of introducing no learnable parameters while providing multiscale positional information through a structured and stable formulation. However, it is limited in that it cannot explicitly encode the relative distances between embedding vectors, and its performance becomes unstable in tasks where the number of patches varies.

B.2. Rotary Positional Embedding (RoPE)

RoPE [2] is a positional encoding method designed to inject positional information into self-attention in a relative form. Let the token sequence of length L be denoted as $\{x_m\}_{m=0}^{L-1}$, and define the query and key at positions m and n as follows.

$$q_m = f_q(x_m, m), \qquad k_n = f_k(x_n, n) \tag{8}$$

Our objective is to make the query–key inner product depend on the relative position $m-n$, rather than the absolute positions m, n. Formally, this can be viewed as the problem of finding a function that satisfies the following condition.

$$\langle f_q(x_m, m),\, f_k(x_n, n)\rangle = g\big(x_m, x_n,\, m-n\big) \tag{9}$$

RoPE achieves this objective by representing queries and keys as vectors rotated according to their positions. To illustrate the idea intuitively, we begin with the 2D (complex-valued) case. Definition of the content embedding at each position:

$$u_m = W_q x_m, \qquad v_n = W_k x_n \tag{10}$$

Definition of the query at position m and the key at position n:

$$f_q(x_m, m) = u_m\, e^{im\theta}, \qquad f_k(x_n, n) = v_n\, e^{in\theta} \tag{11}$$

Here, $e^{im\theta} = \cos(m\theta) + i\sin(m\theta)$ is a complex number representing a rotation by angle $m\theta$ The real part of the query–key inner product after applying RoPE is given by:

$$\begin{aligned} g(x_m, x_n, m-n) &= \Re[\, f_q(x_m, m)\, f_k(x_n, n)^* \,] \\ &= \Re\Big[\, u_m\, v_n^*\, e^{i(m-n)\theta} \,\Big] \end{aligned} \tag{12}$$

Since the rotation terms combine as $e^{im\theta}(e^{in\theta})^* = e^{i(m-n)\theta}$, the positional component naturally depends only on the relative position $m-n$. In practical implementations, complex numbers are replaced with 2D rotation matrices to generalize the idea. By grouping every adjacent pair of dimensions in $x_m \in \mathbb{R}^d$ into a single 2D vector, we obtain:

$$\mathbf{x}_{m,i} = \begin{bmatrix} x_{m,2i} \\ x_{m,2i+1} \end{bmatrix} \in \mathbb{R}^2, \qquad i = 0, \dots, \frac{d}{2} - 1 \tag{13}$$

For each pair of dimensions, the frequency scale is defined as follows:

$$\omega_i = B^{-2i/d}, \qquad B = 10000 \tag{14}$$

The rotation angle used for the i-th dimensional pair at position m is

$$\theta_{m,i} = m\,\omega_i \tag{15}$$

and the corresponding 2 × 2 rotation matrix is given by

$$R(\theta_{m,i}) = \begin{bmatrix} \cos\theta_{m,i} & -\sin\theta_{m,i} \\ \sin\theta_{m,i} & \cos\theta_{m,i} \end{bmatrix} \tag{16}$$

Using this matrix, the $i-th$ dimensional pair of the RoPE-applied vector can be obtained.

$$\tilde{\mathbf{x}}_{m,i} = R(\theta_{m,i})\, \mathbf{x}_{m,i} \tag{17}$$

After performing this operation for all dimension pairs and concatenating the results, the full vector with RoPE applied becomes:

$$\tilde{x}_m = \mathrm{RoPE}(x_m) = \big(\tilde{\mathbf{x}}_{m,0}^\top, \tilde{\mathbf{x}}_{m,1}^\top, \dots, \tilde{\mathbf{x}}_{m,d/2-1}^\top\big)^\top \in \mathbb{R}^d \tag{18}$$

Assuming that RoPE is applied to the query q_m at position m and the key k_n at position n, we can use the rotation-matrix property for each dimenstional pair, $R(\alpha)^\top R(\beta) = R(\beta - \alpha)$, to express the query-key inner product after RoPE as follows.

$$\tilde{q}_m^\top \tilde{k}_n = \sum_i \mathbf{q}_{m,i}^\top\, R\big(\theta_{m,i} - \theta_{n,i}\big)\, \mathbf{k}_{n,i}. \tag{19}$$

Since $\theta_{m,i} - \theta_{n,i} = (m-n)\omega_i$, the positional component for each dimensional pair is expressed as a rotation angle proportional to the relative position difference $m - n$. In other words, RoPE can be viewed as a method that injects relative positional bias into the self-attention scores in a natural way by simply “rotating’’ the query and key vectors according to their positions.

In Vision Transformers, 2D image patches are flattened into a 1D sequence and assigned position indices m.By applying the RoPE transformation to the query and key of each patch, the model can directly encode relative positional information within the self-attention mechanism. In the following section, we introduce Polar RoPE, which extends this basic RoPE formulation into a polar coordinate system.

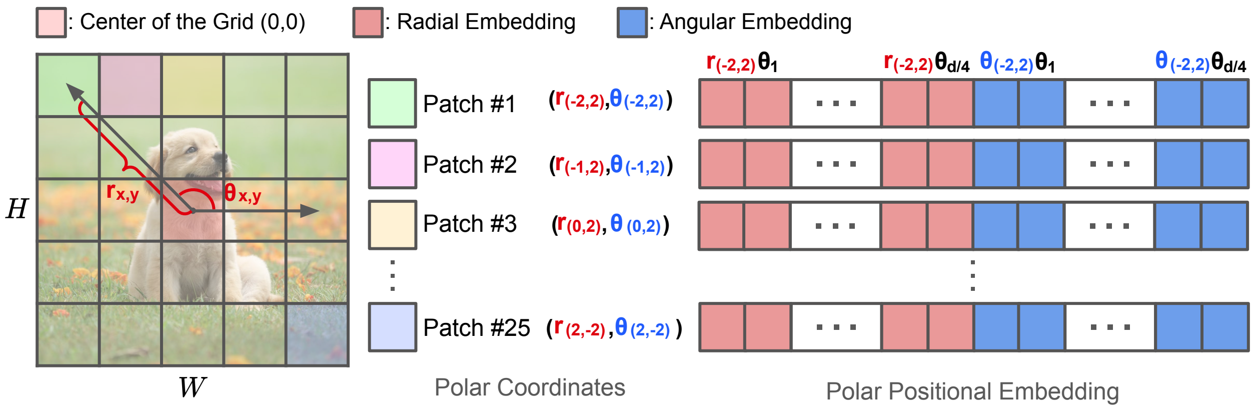
: Center of the Grid (0,0)
: Radial Embedding
: Angular Embedding
H
W
$r_{x,y}$
$\theta_{x,y}$
Patch #1
Patch #2
Patch #3
Patch #25
$(r_{(-2,2)}, \theta_{(-2,2)})$
$(r_{(-1,2)}, \theta_{(-1,2)})$
$(r_{(0,2)}, \theta_{(0,2)})$
$(r_{(2,-2)}, \theta_{(2,-2)})$
$r_{(-2,2)}\theta_1$
$r_{(-2,2)}\theta_{d/4}$
$\theta_{(-2,2)}\theta_1$
$\theta_{(-2,2)}\theta_{d/4}$
Polar Coordinates
Polar Positional Embedding

Figure 1. **Overview of the proposed Polar RoPE.** The spatial coordinates of image patches are first transformed from Cartesian indices to Polar coordinates (r, θ) with respect to the grid center (Left). Correspondingly, the embedding dimension is decomposed into a **radial component** (red) and an **angular component** (blue). Rotary positional embeddings are then applied independently to each subspace to explicitly incorporate distance and directional information into the self-attention mechanism (Right).

B.3. Proposed Method: Polar RoPE

In this paper, we propose Polar RoPE, a positional encoding method based on the polar coordinate system, rather than the conventional Cartesian coordinate system, in order to reflect the center bias of objects in images and the foveal processing characteristics of the human visual system (see Fig. 1). Polar RoPE decomposes the position of each patch into its radial distance (r) and angular direction (θ) from the image center and encodes them into distinct subspaces of the embedding dimensions. This design enables ViT to explicitly learn center–periphery structures and radial directional cues.

Coordinate Transformation. Let (x, y) denote the index of each patch in an $H \times W$ patch grid G.he center coordinates (c_x, c_y) of the image are defined as follows:

$$c_x = \frac{W-1}{2}, \quad c_y = \frac{H-1}{2} \tag{20}$$

Based on this center, the coordinates of each patch are centered as $(x', y') = (x - c_x, y - c_y)$, and subsequently converted into polar coordinates (r, θ)

$$r_{x,y} = \sqrt{(x')^2 + (y')^2}, \quad \theta_{x,y} = \text{atan2}(y', x') \tag{21}$$

Here, $r_{x,y}$ denotes the Euclidean distance from the image center to the corresponding patch, and $\theta_{x,y} \in (-\pi, \pi]$ represents the angular direction of the patch with respect to the center.

Subspace Decomposition & Encoding.

Unlike conventional RoPE, which applies a single positional index (e.g., a 1D sequence index or x, y grid coordinates) to the entire embedding dimension or to paired dimensions, Polar RoPE divides the head dimension d into two subspaces and encodes r and θ separately. Specifically, the query vector q (and likewise the key vector k) is split into two sub-vectors, $q^{(r)}$ and $q^{(\theta)}$, each of dimension $d/2$

$$q = \left[q^{(r)} ; q^{(\theta)} \right], \quad q^{(r)}, q^{(\theta)} \in \mathbb{R}^{d/2} \tag{22}$$

The first subspace, $q^{(r)}$, is used to encode the radial distance r, while the second subspace, $q^{(\theta)}$ encodes the angular information θ. To achieve this, separate frequency bases are defined for each subspace.

$$\omega_j = 10000^{-2j/(d/2)}, \quad j = 0, \dots, \frac{d}{4} - 1 \tag{23}$$

The rotation angles for each component are computed as $\phi_j^{(r)} = r \cdot \omega_j$ and $\phi_j^{(\theta)} = \theta \cdot \omega_j$, respectively.

Polar Rotary Injection. Finally, rotary transformations equivalent to those used in standard RoPE are independently applied to each sub-vector, after which they are concatenated back together. The resulting query vector $\tilde{q}$ with Polar RoPE applied is expressed as follows.

$$\tilde{q} = \left[\text{RoPE}(q^{(r)}, r) \, ; \, \text{RoPE}(q^{(\theta)}, \theta) \right] \tag{24}$$

The attention score produced through this process can be decomposed as follows.

$$\text{Score} \propto \underbrace{(q^{(r)})^T R(r_i - r_j) k^{(r)}}_{\text{Radial relative dist.}} + \underbrace{(q^{(\theta)})^T R(\theta_i - \theta_j) k^{(\theta)}}_{\text{Angular relative dist.}} \tag{25}$$

In other words, Self-Attention equipped with Polar RoPE inherently exploits relative positional information based on the difference in *radial distance* and *difference in angular direction* between two patches. This allows the model to independently and jointly process how far a patch is from the center and in which direction it is oriented, thereby providing an inductive bias that aligns with the geometric characteristics of natural images.

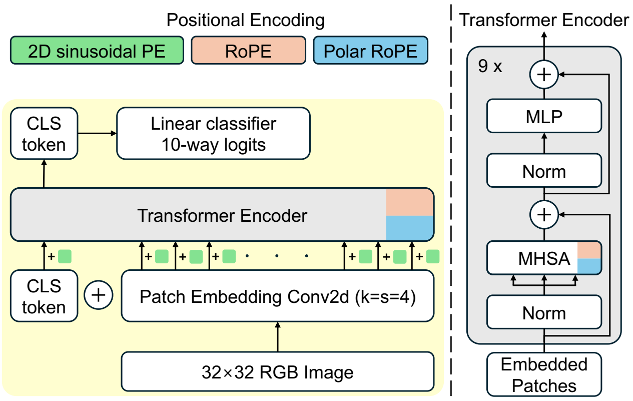

Figure 2. Overview of the ViT architecture used in our experiments. MHSA denotes Multi-Head Self-Attention.

C. Experiment Setup

To validate that the proposed Polar RoPE provides a positional inductive bias more closely aligned with real image statistics than conventional 2D sinusoidal positional encoding or standard RoPE—thereby improving training stability and generalization—we construct models that apply sinusoidal PE, RoPE, and the proposed Polar RoPE on top of a shared Vision Transformer architecture (see Fig. 2). We further evaluate these models on image classification benchmarks by comparing accuracy, loss curves, convergence behavior, and conducting component-wise ablation analyses.

C.1. Dataset

For all image classification experiments, we use the CIFAR-10 benchmark dataset. CIFAR-10 consists of 60,000 RGB images of size 32×32 from 10 classes (6,000 images per class), with an official split of 50,000 training images and 10,000 test images.

In our setup, we construct a validation set by stratifiedly sampling 5,000 images (10% of the training split) and use the remaining 45,000 images for training. For the training set, we apply standard data augmentation with `RandomCrop(32, padding=4)` and `RandomHorizontalFlip()`, followed by `ToTensor()` and normalization using CIFAR-10 statistics. Validation and test images are processed only with `ToTensor()` and the same normalization. In experiments on robustness to resolution changes, we additionally evaluate by resizing the CIFAR-10 test images to 48×48 at inference time.

C.2. Model architecture

We build three Vision Transformer (ViT) models that differ only in their positional encoding: **2D sinusoidal PE**, **RoPE**, and the proposed **Polar RoPE**. All models take 32×32 CIFAR-10 images as input and follow a standard ViT pipeline (see Fig. 2): the image is tokenized into patches, positional information is injected, the token sequence is processed by multiple Transformer encoder blocks, and classification is performed using a class token.

The input image is partitioned into non-overlapping patches of size 4×4 using a convolutional layer with kernel size and stride 4, converting a 32×32 RGB image into $8 \times 8 = 64$ patch tokens. The number of output channels of this layer is set to 192, which we use as the embedding dimension. A learnable class token is prepended to the 64 patch tokens to form a sequence of length 65, after which the selected positional encoding is applied and the sequence is fed into the Transformer encoder.

2D sinusoidal PE. For 2D sinusoidal positional encoding, we use the discrete coordinates (x, y) of the 8×8 patch grid to encode two-dimensional positional information. The embedding dimension is split into two halves: the first half encodes the horizontal coordinate x and the second half encodes the vertical coordinate y, with sinusoidal functions of varying frequencies applied along each axis. The class token receives a zero positional vector, and the 2D sinusoidal encoding is added only to the 64 patch tokens.

RoPE (Rotary Positional Embedding). In the RoPE variant, no explicit positional vectors are added to the input tokens. Instead, positional information is injected by applying position-dependent rotations to the query and key vectors inside each self-attention block. Patch positions on the 8×8 grid are represented as integer coordinates (x, y), and the head dimension D is split so that the first half encodes frequency components for x and the second half for y. Using precomputed $\cos$ and $\sin$ tables derived from these phase values, we rotate the query and key of all patch tokens during self-attention, while keeping the CLS token unrotated so that it remains a position-independent global summary.

Polar RoPE. Polar RoPE retains the overall RoPE structure but redefines the positional parameters using polar coordinates (r, θ) relative to the image center instead of Cartesian coordinates (x, y). For each patch on the 8×8 grid, we first shift the grid so that its center (c_x, c_y) becomes the origin, compute $x' = x - c_x$ and $y' = y - c_y$, and then obtain

$$r = \sqrt{x'^2 + y'^2}, \qquad \theta = \operatorname{atan2}(y', x').$$

The head dimension D is again split into two halves: the first encodes frequency components for the radial term r and the second for the angular term θ. As in RoPE, we construct $\cos$ and $\sin$ tables from these components and apply rotary transformations to the query and key vectors of all patch tokens, excluding the CLS token. This way, Polar RoPE differs from standard RoPE only in how positional parameters are defined, while aligning its positional representation with the radial structure and center bias observed in natural images.

C.3. Training hyperparameters

Across all experiments, we use the same Vision Transformer architecture, fixing the patch size to 4×4,the embedding dimension to 192, the depth to 9 blocks, the number of heads to 12, and the MLP ratio to 4. The three positional encoding variants (sinusoidal, RoPE, and the proposed Polar RoPE) differ only in the `pe_method`. all other architectural components and training configurations remain identical.

We split the CIFAR-10 training set into 45,000 training images and 5,000 validation images using stratified sampling, and we use the standard 10,000 images for the test set. The batch size is set to 128, and the total number of training epochs is 50. We adopt AdamW as the optimizer, with an initial learning rate of 1×10^{-3} and a weight decay of 1×10^{-4} applied to most parameters. Positional embeddings, the class token, bias terms, and normalization-layer parameters are excluded from weight decay by assigning them to separate parameter groups. For learning rate scheduling, we employ a cosine annealing schedule (`CosineAnnealingLR`,$T_{\max} = 50$) over the full training duration. The loss function used is the standard cross-entropy loss for multi-class classification.

To ensure a fair comparison across the three positional encoding methods, we train all models (sinusoidal, RoPE, and Polar RoPE) under identical random seed settings for each global seed. To mitigate the possibility of results being influenced by seed-specific randomness, we repeat the entire experimental procedure using multiple global seeds (e.g., $0, 42, 3407$). The validation split is generated once using a fixed random seed and is kept identical across all repeated experiments.

C.4. Ablation settings for Polar RoPE

In this study, we conduct ablation experiments to quantitatively assess the individual contributions of the radial component r and the angular component θ within the proposed Polar RoPE design. All configurations use the same Vision Transformer backbone, CIFAR-10 dataset, and training hyperparameters as in the main experiments. the only differences lie in how the r and θ components are incorporated into the RoPE rotation within Polar RoPE.

The experimental configurations are defined as follows. First, in Radius-only Polar RoPE, positional rotation is applied using only the radial component r derived from the polar coordinates. To achieve this, we split the head dimension into two halves: the first half applies RoPE frequency components based on r, while the second half disables RoPE by setting $\cos = 1, \sin = 0$. Because the dimensions without RoPE preserve their original values, the model architecture and representational capacity remain identical to those of the full Polar RoPE, ensuring that only the r component contributes to the rotation.

Method	Val Acc (%)	Test Acc (%)
2D sinusoidal PE	79.07 ± 1.82	78.56 ± 1.20
RoPE	81.42 ± 0.55	81.24 ± 0.42
Polar RoPE (ours)	**82.76 ± 0.81**	**82.63 ± 0.32**

Table 1. Classification performance (Mean ± Std over 3 seeds) of different positional encoding methods on CIFAR-10. All models share the same ViT backbone and training hyperparameters.

Second, in Angle-only Polar RoPE, only the angular component θ is used to encode directional (orientation) information. Symmetrically to the Radius-only setting, the first half of the head dimensions does not apply RoPE (i.e., remains identity), while the second half applies RoPE frequency components derived fromθ. This design preserves the model architecture while ensuring that only the θ component contributes to the relative rotational encoding within attention.

Third, in Full Polar RoPE, the head dimension is split into two halves as in the default design, with the first half corresponding to the radial component r and the second half to the angular component θ. RoPE rotation is applied to both components, yielding a complete Polar RoPE representation that incorporates both distance information based on r and directional information based on θ.

In all three experimental settings, the CLS token is excluded from the RoPE rotation, and Polar RoPE is applied only to the spatial tokens corresponding to the 8×8 patch grid. Training is conducted for 50 epochs using the AdamW optimizer and a cosine annealing learning-rate schedule. For each variant, we measure validation and test accuracy and compare the results against full Polar RoPE and 2D sinusoidal PE. This analysis allows us to quantify how the radial component r and the angular component θ individually contribute to the performance improvements achieved by Polar RoPE.

D. Results

D.1. Overall Classification Performance

We first quantitatively compare three positional encoding methods—2D sinusoidal PE, RoPE, and the proposed Polar RoPE—on CIFAR-10 image classification performance.

Table 1 summarizes the final validation and test accuracies of the three methods. Overall, RoPE consistently achieves higher accuracy than 2D sinusoidal PE, and the proposed Polar RoPE further surpasses both baselines. In particular, Polar RoPE attains the highest validation and test accuracies, demonstrating that simply changing the positional encoding while keeping the backbone architecture fixed yields a notable performance improvement.

From multi-seed experiments, we compare the stability

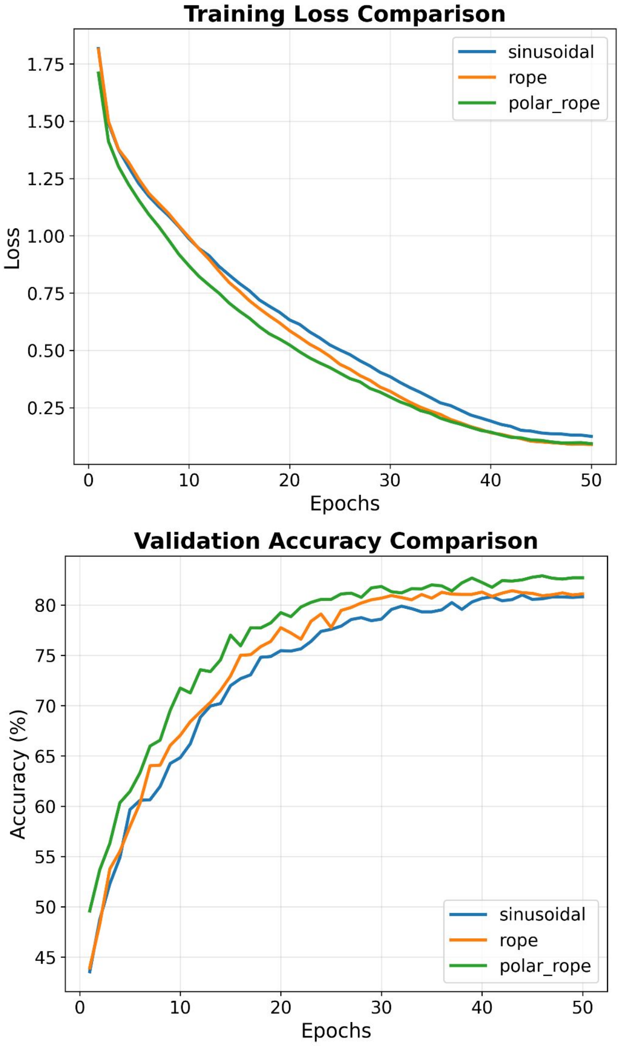

Figure 3. Training loss and validation accuracy curves. Comparison of ViT models trained with 2D sinusoidal PE, RoPE, and the proposed Polar RoPE on CIFAR-10.

of each positional encoding method. 2D sinusoidal PE exhibits the largest standard deviation in both validation and test accuracy, indicating that it is the most sensitive to the choice of random seed and shows the highest variability. In contrast, RoPE and Polar RoPE consistently show much smaller variance, suggesting more stable training behavior.

Considering both validation and test accuracy, 2D sinusoidal PE exhibits the largest standard deviation, indicating the highest sensitivity to random seeds. RoPE shows the smallest variance on validation accuracy, while Polar RoPE remains clearly more stable than 2D sinusoidal PE and achieves the lowest variance on test accuracy, providing the most consistent generalization across runs.

D.2. Convergence Analysis

We compare the convergence behavior of the three positional encodings using the training curves in Fig. 3. Overall, Polar RoPE adapts faster from the early stages of training. For example, when we align the epoch at which the validation accuracy reaches 70%, 2D sinusoidal PE requires 14 epochs and RoPE 13 epochs, whereas Polar RoPE reaches the same performance in only 10 epochs. This suggests that polar-coordinate positional information provides more structured spatial signals to self-attention, enabling more effective pattern learning from the beginning of training.

In terms of training loss, Polar RoPE maintains the lowest loss throughout training, with the gap to the other methods being most pronounced in the mid-training phase (approximately epochs 0–30), where it reaches a lower-loss region more quickly at the same epoch. This behavior indicates that separating radius (r) and angle (θ) and encoding them independently leads to more efficient optimization and faster loss reduction.

Inspecting training loss and validation accuracy together, we do not observe clear signs of overfitting for any of the three positional encodings: training loss steadily decreases, while validation accuracy increases and then saturates without a noticeable late-epoch drop. Under this setup, the choice of positional encoding does not substantially affect the degree of overfitting, but Polar RoPE yields the fastest early convergence and the most efficient loss reduction, supporting the view that it provides a beneficial positional inductive bias for ViT training dynamics.

D.3. Ablation Study on Polar RoPE

To analyze the contribution of the radius r and angle θ components in the proposed Polar RoPE, we design three ablation settings: (1) *Radius-only Polar RoPE*, which encodes only the center–periphery distance using r; (2) *Angle-only Polar RoPE*, which encodes only directional information (orientation) using θ; and (3) *Full Polar RoPE*, which leverages both components. All three settings share the same ViT backbone and training configuration, and differ only in the positional information used for the RoPE rotation inside self-attention.

As shown in Table 2, the Radius-only setting yields the lowest performance, with a validation accuracy of 76.04% and a test accuracy of 75.77%, indicating that distance alone provides only a limited positional bias. In contrast, the Angle-only setting significantly improves performance, achieving 81.82% validation accuracy and 80.93% test accuracy, suggesting that orientation carries important structural cues in natural images such as CIFAR-10.

Full Polar RoPE, which combines both components, achieves the best results with 82.70% validation accuracy and 82.47% test accuracy. This indicates that r and θ act in

Variant	Val Acc (%)	Test Acc (%)
Radius-only Polar RoPE	76.04	75.77
Angle-only Polar RoPE	81.82	80.93
Full Polar RoPE ($r + \theta$)	82.70	82.47

Table 2. Ablation study on the components of Polar RoPE. All models share the same ViT backbone; only the radial (r) and angular (θ) components used in the positional encoding differ.

a complementary manner, providing a richer positional inductive bias to self-attention than either component alone.

E. Discussion

Which PE Works Best and Why?

The experimental results show that the proposed Polar RoPE achieves the best overall performance and convergence behavior. The conventional 2D sinusoidal PE has a rigid structure that cannot differentiate positional importance, while standard RoPE incorporates relative positional information but remains limited in capturing the intrinsic geometric characteristics of 2D images. In contrast, Polar RoPE reparameterizes positional information using (r, θ) , enabling the self-attention mechanism to directly learn the center bias and structural patterns present in natural images. By leveraging both distance and directional cues, Polar RoPE provides a richer inductive bias, leading to further performance gains over standard RoPE.

Does Polar RoPE Help Early Convergence?

From a convergence perspective, Polar RoPE exhibits the fastest loss reduction during the early stages of training. This behavior suggests that self-attention is able to exploit meaningful relative positional structures from the outset, thereby reducing unnecessary parameter exploration by the optimizer and enabling more efficient convergence toward a good optimum. In particular, because Polar RoPE explicitly differentiates relationships between central and peripheral patches, it can rapidly capture the common “central object + surrounding background’’ pattern found in natural images. This leads to a noticeably steeper loss decline during the initial epochs.

When Does RoPE Still Outperform Sinusoidal PE?

Although Polar RoPE delivers the strongest overall performance and stability, understanding the relative characteristics of RoPE and 2D sinusoidal PE is equally important. Under conditions involving resolution changes (32×32 → 48×48) or substantial shifts in input positions, RoPE consistently demonstrates stronger robustness than 2D sinusoidal PE. This advantage arises because RoPE encodes positional information through rotations based on the relative position difference $(m - n)$ inside self-attention, allowing it to preserve meaningful inter-patch relationships even when absolute coordinates or grid scales change. In contrast, 2D sinusoidal PE relies on frequency patterns tied to absolute coordinates; therefore, when input resolution or spatial alignment shifts, the semantics of the encoded positional signals change substantially, often leading to performance degradation.

F. Conclusion

In this work, we propose Polar RoPE, a polar-coordinate–based positional encoding method designed to enhance the positional representation capability of Vision Transformers by incorporating the foveal characteristics of human vision and the center bias commonly observed in natural images. Polar RoPE decomposes each patch location into radial(r)and angular(θ)components and encodes them independently in separate subspaces of the embedding dimensions, enabling the model to effectively learn center–periphery structure and radial directional patterns. Experiments on the CIFAR-10 dataset show that Polar RoPE achieves the highest classification accuracy (82.47%) and the fastest early-stage convergence compared to both 2D sinusoidal PE and standard RoPE. Furthermore, the ablation study demonstrates that the radial and angular components contribute complementarily to the overall performance improvement. These findings suggest that, rather than relying solely on grid-based coordinates, selecting a coordinate system that better matches the geometric distribution of the data plays a crucial role in enhancing the representational learning capability of Vision Transformers. For future work, we aim to validate the effectiveness of Polar RoPE on larger-scale datasets and higher-resolution images such as ImageNet. Another promising direction is to extend the method by introducing a learnable center, allowing the model to adapt the reference point dynamically to accommodate diverse object distributions rather than relying on a fixed center.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017. 1

[2] J. Su, M. Ahmed, Y. Lu, S. Pan, B. Wen, and Y. Liu, “RoFormer: Enhanced transformer with rotary position embedding,” *Neurocomputing*, vol. 568, p. 127063, 2024. 1, 2

[3] Q. Wu, Y. Li, V. Gómez, L. Zhang, and Y. Xu, “Rethinking and improving relative position encoding for vision transformer,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10 033–10 042. 1

[4] W. Heo, J. Park, J. Han, and S. Yun, “Rotary position embedding for vision transformer,” in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2024, pp. 684–701. 1