SpRePE: A Spherical Geometry-Aware Position Embedding scheme for Vision Transformers

Qilong Jia
Tsinghua University, Beijing
jql23@mails.tsinghua.edu.cn

Yi Xiao
Tsinghua University, Beijing
y-xiao22@mails.tsinghua.edu.cn

Wei Xue[✉]
Tsinghua University, Beijing
xuewei@tsinghua.edu.cn

Abstract

Vision Transformers are increasingly applied to data defined on the sphere, particularly in physics, meteorology, and related scientific domains. Position embeddings provide the spatial information that attention itself does not encode, making their design central to the expressive power of Transformers. However, most existing position embedding schemes are designed for Cartesian grids and therefore do not naturally handle longitude periodicity, polar singularities, or geodesic relations on spherical domains. This mismatch limits the ability of standard attention to model spherical geometry without specialized architectural modifications.

We propose Spherical Reflection Position Embedding (SpRePE), a drop-in position embedding scheme for Vision Transformers on spherical data. SpRePE encodes each absolute spherical position by applying Householder reflections to query and key representations. Although each token is encoded only from its own absolute coordinates, the resulting attention inner products induce an explicit spherical relative-position term with a clear geometric interpretation. This formulation injects sphere-aware relative geometric information into standard attention without constructing pairwise attention-bias matrices, introducing task-specific modules, or modifying the backbone architecture. It also avoids the quadratic overhead of relative position biases and preserves the same asymptotic overhead as RoPE.

We evaluate SpRePE on spherical image classification, panoramic depth estimation, and global weather forecasting. Across these tasks, SpRePE achieves competitive performance compared with strong position embedding baselines, with particularly clear gains in settings where spherical geometry is important.

1 Introduction

The Transformer architecture [1] was first introduced for natural language processing tasks and has since demonstrated a strong non-linear modeling capacity across numerous benchmarks [1, 2, 3]. Its vision counterpart, Vision Transformer (ViT), applies the same attention principle to sequences of image patches and achieves competitive or superior performance compared to convolutional neural networks, particularly for modeling long-range dependencies [4, 5].

Since the attention mechanism itself is position-agnostic, a Transformer model without position embedding (PE) treats all tokens as interchangeable, severely limiting representational ability [6]. Injecting positional information is therefore indispensable for capturing token order or spatial layout, and preserving accuracy for Transformer [1, 3, 7, 8] as well as Vision Transformer [4, 9].

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Spherical data are widely used across many scientific domains, including geoscience [10], astronomy [11], and meteorology [12]. In recent years, Transformer-based models have also been introduced to address problems involving spherical data, with notable examples including Pangu-Weather [13] for weather forecasting and WenHai [14] for oceanography. Unlike planar Cartesian coordinates, the spherical domain is non-Euclidean. Spherical coordinates obey periodic boundary conditions in longitude and exhibit coordinate singularities at the poles. These properties pose specific challenges for positional encoding when applying Vision Transformers on the sphere.

Existing approaches to positional modeling on spherical domains can be broadly grouped into two categories. The first inherits positional embedding schemes originally developed for standard Transformers, including additive absolute position embeddings [1, 3, 7, 15], relative attention biases [5, 8, 16, 17, 18, 19], and rotary position embedding [9, 20]. These methods are typically defined on Euclidean grids, planar coordinates, or discrete token offsets, and therefore do not naturally capture longitudinal periodicity, polar singularities, or geodesic relations on the sphere. Several studies further adapt these formulations to spherical data [13, 21], but they still capture only limited aspects of spherical geometry. The second category explicitly incorporates spherical or Earth-geometry priors through sphere-specific attention kernels or spherical operators [22, 23, 24]. These approaches provide stronger spherical inductive biases, but they usually rely on specialized grids, geometric kernels, spherical operators, or additional modules.

To address the above challenges and the shortcomings of current methods, we introduce SpRePE, a reflection-matrix-based spherical position embedding scheme for Vision Transformer. SpRePE encodes absolute positions on the sphere using reflection matrices and directly injects an explicit spherical relative-position term into the attention mechanism. The reflection-matrix formulation endows this relative term with a clear geometric interpretation, thereby respecting spherical geometric constraints. SpRePE adopts a drop-in design that operates solely within the attention module, requiring neither substantial additional parameters nor modifications to the overall model architecture. In practice, the extra computation overhead is negligible compared with the attention computation.

To validate the effectiveness of SpRePE, we conduct experiments on representative spherical tasks: spherical image classification, panoramic depth estimation and global weather forecasting. The experiments are implemented within the same Vision Transformer framework to ensure fair and consistent evaluation. We compare SpRePE with well-known position embedding methods. The results show that SpRePE performs strongly across tasks, with particularly pronounced advantages when conventional position embedding are affected by geometric distortions. These results demonstrate that SpRePE is an efficient and practical position embedding method for attention-based networks in spherical modeling tasks.

2 Related Work

Transformer position embeddings and spherical variants. Position information is commonly injected into Vision Transformers through absolute position embeddings or relative position biases/embeddings [4, 5]. Later studies propose positional encoding schemes with better transfer or extrapolation ability, such as ALiBi-style biases [19] and RoPE [20]. In vision tasks, RoPE is further extended to one-dimensional rotations along token order, two-dimensional rotations factorized along the height and width axes, and fused two-dimensional rotations [9, 25, 26]. However, these methods are mostly defined on Euclidean grids, planar coordinates, discrete token-index offsets, or abstract transformations over coordinate domains. When directly applied to spherical data represented by equirectangular projection, they do not naturally capture longitudinal periodicity, the seam at the prime meridian, polar coordinate singularities, or geodesic distortions caused by non-uniform area elements on the sphere.

Several recent models adapt positional encoding to spherical data, especially in global weather forecasting. For example, Pangu-Weather introduces an Earth-specific positional bias [13], and ARROW proposes Ring Positional Encoding [21]. These studies show that spherical geometric information is important for global modeling. However, they mainly model circular periodicity, ring-like structures, or task-specific geographic biases, rather than providing a general formulation of spherical relative position. In contrast, SpRePE constructs position-dependent orthogonal transformations directly on S^2 through Householder reflections. For two spherical positions p_i and p_j, the relative term induced by $R_n(p_i)^\top R_n(p_j)$ has a clear geometric interpretation on the sphere. Thus, SpRePE

Table 1: Comparison of position embedding schemes.

Property	**APE**	**RPE**	**RoPE**	**SpRePE**
Where applied	stem stage	attention score	q/k vectors	q/k vectors
Compute-efficient	✓	×	✓	✓
Relative info	×	✓	✓	✓
Sphere-aware	×	×	×	✓

retains the drop-in form of RoPE-like query/key transformations while injecting spherical relative positional information into standard attention, without constructing pairwise attention-bias matrices or introducing task-specific modules.

Spherical representations, spherical attention, and spherical operators. Another line of work models data directly on S^2 through spherical basis functions, spherical grids, attention kernels, or operators. Representative methods include DFS and related sinusoidal or harmonic position encodings [27, 28, 29], location encoders such as Sphere2Vec and spherical-harmonic-based representations [30, 31], geometry-aware attention designs such as EGFormer, Attention on the Sphere, and CirT [22, 23, 32], and spherical operator-learning frameworks such as SFNO [24]. These methods usually obtain stronger spherical inductive biases by redesigning the representation, attention kernel, numerical integration rule, or backbone architecture. As a result, they often require specialized spherical transforms, geometric kernels, spherical operators, or additional modules.

SpRePE is complementary to this line of work. It does not aim to replace spherical Transformers, geometry-aware attention kernels, or spherical neural operators. Instead, it provides an architecture-preserving position embedding that acts only on query and key representations in standard attention. This design injects spherical relative geometric information while keeping the overall Transformer backbone unchanged.

3 Method

We propose **Spherical Reflection Position Embedding (SpRePE)**, a position embedding scheme for spherical domain. SpRePE uses absolute embeddings to induce spherical relative information through inner products between position-encoded queries and keys. In particular, **SpRePE** requires only the absolute position of a single token at the embedding stage, while the attention mechanism expresses pairwise relative positional information between arbitrary tokens.

More concretely, when a Vision Transformer processes spherical data, let $p_1 = (\theta_{p_1}, \phi_{p_1})$ and $p_2 = (\theta_{p_2}, \phi_{p_2})$ be two arbitrary points on the unit sphere, where $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$ denote the polar angle and the azimuth angle, respectively. SpRePE defines a mapping $f(\,\cdot\,,\cdot)$ such that, for a query vector q and a key vector k,

$$\big\langle f(q, p_1),\, f(k, p_2) \big\rangle \;=\; g\big(q,\, k,\, \mathrm{pos}(p_1, p_2)\big), \tag{1}$$

where $\mathrm{pos}(p_1, p_2)$ denotes a function of the *relative* position between p_1 and p_2 on the sphere. Table 1 compares SpRePE with well-known PE schemes in terms of several key properties. The RoPE-like design of SpRePE enables the incorporation of spherical relative positional information while keeping the computational overhead negligible.

3.1 Derivation of SpRePE in Three Dimensions

We begin with the special case in which the embedding dimension is $d = 3$. Following the RoPE-style formulation, we seek position-dependent linear transformations that preserve feature norms and encode relative position through the query–key inner product. The target relation in Eq. 1 admits multiple constructions. We first describe the geometric constraints and remaining freedom, and then introduce the Householder construction used by SpRePE.

For convenience, we first convert the spherical coordinates to Cartesian form: $p_1 = \big(\sin\theta_{p_1}\cos\phi_{p_1}, \sin\theta_{p_1}\sin\phi_{p_1}, \cos\theta_{p_1}\big)$, $p_2 = \big(\sin\theta_{p_2}\cos\phi_{p_2}, \sin\theta_{p_2}\sin\phi_{p_2}, \cos\theta_{p_2}\big)$. We consider the class of embeddings in which f acts by multiplication with an orthogonal matrix; that is,

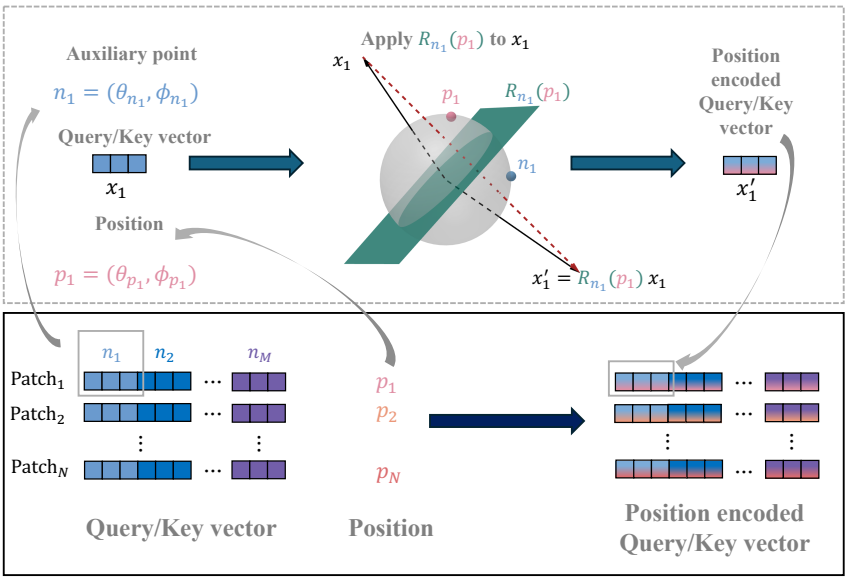

Figure 1: **Implementation of Spherical Reflection Position Embedding (SpRePE).** The embedding space is partitioned into $d/3$ disjoint 3D subspaces, each of which is processed independently by SpRePE. For each subspace, we select an auxiliary point $n_m = (\theta_{n_m}, \phi_{n_m})$ on the unit sphere. Given a patch located at $p_i = (\theta_{p_i}, \phi_{p_i})$, the Householder reflection matrix $R_{n_m}(p_i)$ is applied to the corresponding query/key vector x_i, yielding the position-encoded vector $x_i' = R_{n_m}(p_i)\, x_i$. This scheme ensures that the embedding stage relies only on the *absolute* coordinates of p_i. The upper part of the figure illustrates the special case with $m = 1$ and $i = 1$.

$f(q, p) = R(p)\, q$. Even under this constraint, the choice of $R(p)$ is not unique. Then,

$$g\big(q, k, \mathrm{pos}(p_1, p_2)\big) \;=\; \big\langle f(q, p_1),\, f(k, p_2) \big\rangle \;=\; q^\top R(p_1)^\top R(p_2)\, k \;=\; q^\top R(p_1, p_2)\, k, \quad (2)$$

where $R(p_1, p_2) = R(p_1)^\top R(p_2)$. Because both $R(p_1)$ and $R(p_2)$ are orthogonal and satisfy $\det R(p_1) = \det R(p_2)$, we have $\det R(p_1, p_2) = \det R(p_1)\, \det R(p_2) = 1$. Hence, $R(p_1, p_2)$ is an orthogonal matrix with unit determinant, that is, a three-dimensional rotation.

We further impose the anchoring condition

$$p_1{}^\top R(p_1, p_2)\, p_2 \;=\; p_1{}^\top R(p_1)^\top R(p_2)\, p_2 \;=\; p_1{}^\top p_1 \;=\; p_2{}^\top p_2 \;=\; 1. \quad (3)$$

Accordingly, $R(p_1, p_2)$ is a rotation matrix that maps p_2 onto p_1. This mapping constraint still leaves a one-parameter family of admissible rotations. For $p_1 \neq \pm p_2$, the principal rotation angle γ satisfies $\gamma \in \big[\arccos\langle p_1, p_2\rangle, \pi\big]$. Once γ is fixed, the rotation axis can be chosen from at most two alternatives.

However, directly choosing this pair-dependent rotation angle does not provide a position-wise map $R(p)$ for the embedding scheme, in which each position must be encoded independently. We therefore seek a parameterization that determines $R(p)$ from a single position.

Instead, we parameterize the construction by an auxiliary point n and encode absolute position through a Householder reflection $R_n(p)$. Reflections are preferred because, for a fixed pair (n, p), $R_n(p)$ is uniquely determined. In contrast, the set of rotations that map n to p forms a one-parameter family, which introduces axis-angle ambiguity. Moreover, the induced relative term,

$$R_n(p_1, p_2) = R_n(p_1)^\top R_n(p_2) \quad (4)$$

has a clear geometric interpretation: it is a rotation whose axis is normal to the affine plane through n, p_1, and p_2, and its rotation angle is uniquely defined as the arc length between p_1 and p_2 along the circle obtained by intersecting that plane with the sphere, divided by the radius of the circle. A more complete proof is provided in the Appendix C.

This explicit geometric interpretation is useful in practice because it shows that the degree of freedom associated with n can be redundant: any two choices of n that generate the same plane yield identical

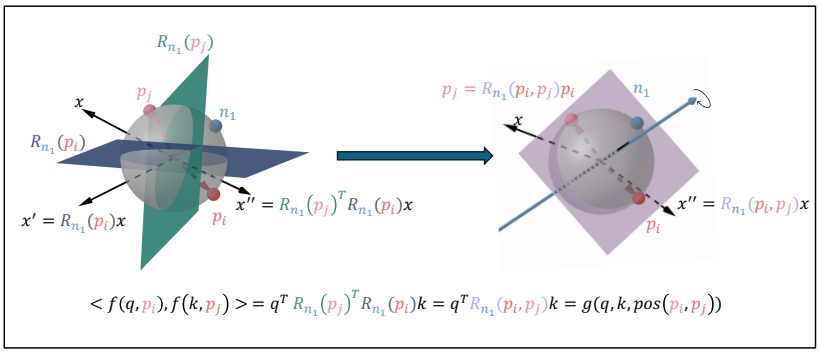

Figure 2: **Illustration of how SpRePE encodes *relative* position information through *absolute* reflections and an inner product ($d = 3$).** The Householder matrix is applied to the query/key vector x_i, yielding the position-encoded vector $x'_i = R_{n_1}(p_i)\, x_i$. Given two patches p_i and p_j, the product $R_{n_1}(p_i)^\top R_{n_1}(p_j)$ is a rotation whose axis is the normal of the plane defined by n_1, p_i, and p_j. Consequently, the inner product $\langle f(q, p_1),\ f(k, p_2)\rangle = q^\top R_n(p_1)^\top R_n(p_2)k$ depends only on q, k, and the relative position $\mathrm{pos}(p_1, p_2)$, thereby realizing Eq. 1 without observing both coordinates simultaneously at the embedding stage.

embeddings. To avoid *representational degeneracy*, we enforce that, for every pair (p_1, p_2), any two auxiliary points n_1 and n_2 must define distinct spherical sections. Under this constraint, distinct choices of n lead to distinct embeddings. Ablation results in Appendix F further indicate that degeneracy degrades the performance of SpRePE, whereas SpRePE remains robust to the choice of n once this constraint is satisfied. Because the input grid is fixed *a priori*, the token locations are also fixed by the grid. This constraint can therefore be enforced straightforwardly in practice, making degeneracy easy to avoid.

Algorithm 1 SpRePE ($d = 3$)

Input: query/key vector $x \in \{q, k\}$,
token position p,
auxiliary point n
Output: position-aware vector $\hat{x}$

$v \leftarrow \dfrac{p - n}{\|p - n\|}$
$R \leftarrow I - 2vv^\top$
$\hat{x} \leftarrow Rx$
return $\hat{x}$

Algorithm 2 Attention score with SpRePE ($d = 3$)

Input: $Q = [q_1^\top, q_2^\top, \ldots, q_N^\top]$, $K = [k_1^\top, k_2^\top, \ldots, k_N^\top]$,
token positions $P = [p_1, p_2, \ldots, p_N]$, auxiliary point n
Output: attention score matrix Attn

for $i = 1, 2, \ldots, N$ **do**
  $\hat{q}_i \leftarrow$ **SpRePE**(q_i, p_i, n), $\hat{k}_i \leftarrow$ **SpRePE**(k_i, p_i, n)
end for
$\hat{Q} \leftarrow [\hat{q}_1^\top, \hat{q}_2^\top, \ldots, \hat{q}_N^\top]$, $\hat{K} \leftarrow [\hat{k}_1^\top, \hat{k}_2^\top, \ldots, \hat{k}_N^\top]$
Attn $\leftarrow \mathrm{Softmax}(\hat{Q}\hat{K}^\top)$
return Attn

It is worth emphasizing that the proposed embedding scheme imposes no assumption on the grid and it imposes no restriction on the backbone architecture; the only requirement is the presence of an attention mechanism.

We next derive the explicit form of $R_n(p_1)$. Let p_1 and n be two unit vectors on the sphere. The Householder reflection that maps p_1 onto n is

$$R_n(p_1) = I - 2vv^\top, \qquad v = \frac{n - p_1}{\|n - p_1\|} = (v_1, v_2, v_3)^\top. \tag{5}$$

In summary, the above derivation gives the complete three-dimensional form of SpRePE. The resulting position-embedding procedure is outlined in Algorithm 1. After applying this reflection independently to all query and key vectors, the reflected matrices $\hat{Q}$ and $\hat{K}$ are used to compute the attention score matrix. This attention computation with SpRePE is outlined in Algorithm 2.

3.2 General Form

In the general setting where the embedding dimension is a multiple of three, we partition the d-dimensional space into $M = d/3$ three-dimensional subspaces and apply an independent reflection

to each subspace. Formally, the spherical position embedding acts as $f(x, p) = R(p)\, x$, where

$$R(p) = \mathrm{diag}(R_{n_1}(p), R_{n_2}(p), \ldots, R_{n_M}(p)). \tag{6}$$

Each $R_{n_i}(p)$ is the Householder matrix that reflects p onto the auxiliary point n_i. Distinct choices of the auxiliary points $\{n_i\}_{i=1}^{M}$ therefore yield a family of block-diagonal reflection matrices that act independently on disjoint triplets of feature dimensions.

Using the block-diagonal structure of $R(p)$ in Eq. 6, we do not explicitly materialize the full matrix $R(p)$. Instead, each three-dimensional block is applied through the Householder transformation

$$x'_m = x_m - 2v_{(n_m,p)}(v_{(n_m,p)}^{\top} x_m), \tag{7}$$

where $x_m \in \mathbb{R}^3$ denotes the m-th feature triplet and $v_{(n_m,p)}$ is the Householder vector determined by the auxiliary point n_m and position p. This implementation only requires element-wise operations and reductions within each triplet, and therefore has linear overhead in the embedding dimension.

3.3 Reducing Density-Induced Degeneracy via Partial SpRePE

To allow the contribution of spherical positional transformations to be learned at the channel level, we augment SpRePE with a learnable mask:

$$Q = (1 - M) \circ Q_0 + M \circ \mathrm{SpRePE}(Q_0, p),$$
$$K = (1 - M) \circ K_0 + M \circ \mathrm{SpRePE}(K_0, p),$$

where Q_0 and K_0 are the original content-dependent query and key projections, M is the learnable channel-wise mask, and $\circ$ denotes element-wise multiplication. This gating allows the model to regulate the strength of positional modulation while providing a direct pathway for the original projections.

3.4 Computational Overhead

Table 2 compares the additional computational cost introduced by each position embedding scheme. We consider a generic ViT with L layers, hidden dimension d, h attention heads, and batch size B, where each sample contains $N = H \times W$ patch tokens. For clarity, we report only leading-order terms and omit constants and lower-order factors.

Compared with common position embedding schemes, RPE introduces substantially higher computational and memory overhead. This overhead mainly comes from explicitly constructing an $N \times N$ attention-bias matrix for each head in every layer, which makes both computation and memory scale quadratically with the sequence length N. Such overhead becomes increasingly prohibitive in high-resolution settings and forms a critical bottleneck when scaling to high-resolution weather forecasting tasks.

In contrast, APE introduces only a one-time cost at the input embedding stage, with both time complexity and memory usage scaling linearly with the sequence length N. Since the dominant attention operations, namely $QK^{\top}$ and $\mathrm{softmax}(\cdot)V$, have time complexity $\mathcal{O}(N^2 d)$ and memory complexity $\mathcal{O}(hN^2 + Nd)$, this cost is negligible relative to the overall computation of the ViT backbone. Similarly, RoPE and SpRePE apply positional encoding to each token at every layer, with overhead that scales linearly with the sequence length N, hidden dimension d, and number of layers L. Notably, although SpRePE introduces stronger geometric constraints through reflections, it preserves the same asymptotic complexity as RoPE. We further provide end-to-end wall-clock time and peak memory usage for different PE schemes under the same experimental setting in Appendix J.

This efficiency allows SpRePE to introduce explicit relative positional information on the sphere with only a small additional cost beyond standard attention. As a result, SpRePE maintains low overhead while retaining strong positional expressiveness, making it well suited for spherical domains.

4 Experiment

In our main experiments, we integrate SpRePE into a standard ViT [4] and compare performance by varying only the positional embedding scheme while keeping all other hyperparameters identical.

Table 2: Asymptotic inference overhead of different position embeddings.

PE scheme	Time Complexity	Space Complexity
APE	$\mathcal{O}(BNd)$	$\mathcal{O}(Nd)$
RPE	$\mathcal{O}(BhN^2L)$	$\mathcal{O}(hN^2L)$
RoPE	$\mathcal{O}(BNdL)$	$\mathcal{O}(NdL)$
SpRePE	$\mathcal{O}(BNdL)$	$\mathcal{O}(NdL)$

PE scheme	Accuracy (%) ↑
NoPE	86.83
APE	97.40
RPE	87.48
Sphere2Vec	97.95
RoPE	97.88
RingPE	94.78
SpRePE	97.84
SpRePE(LM)	97.60

Table 3: **Spherical MNIST Top-1 accuracy (%) on test set without rotation.**

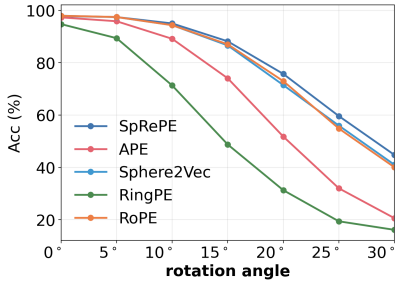

Figure 3: **Spherical MNIST Top-1 accuracy (%) on test set under the rotated setting.**

We evaluate SpRePE on representative tasks: panoramic depth estimation on Structured3D[33] and weather forecasting on ERA5 [12]. The baselines include No Position Embedding (NoPE), learnable Absolute Position Embedding (APE) [4], learnable Relative Position Bias (RPE) [5], Sphere2Vec [30], Ring Position Embedding (RingPE) [21] and Rotary Position Embedding (RoPE) [9]. In addition, to assess the effectiveness of our degeneracy-mitigation strategy, we report SpRePE(LM), which adopts a *learnable*, channel-wise mask applied along the hidden dimension and *shared across tokens* within each layer. Unless otherwise noted, the auxiliary-point set is also learned end-to-end in our experiments. An ablation study of mask selection and auxiliary-point selection is provided in Appendix F. To assess the applicability of SpRePE beyond the standard ViT backbone, we additionally evaluate it with EGFormer for panoramic depth estimation and S2Transformer for weather forecasting, with results reported in Appendix H.

4.1 Spherical image classification

We evaluate SpRePE on spherical image classification using the Spherical MNIST dataset, which places handwritten digits near the South Pole to stress correct handling of spherical geometry. Model and training hyper-parameters are provided in the Appendix D.

Figure 3 reports the Top-1 accuracy of different positional encoding schemes under rotated inputs, where each input is rigidly rotated around the axis normal to the prime-meridian plane. A rotation angle of 0 corresponds to the unrotated evaluation setting (result in Table 3), which matches the input scenario used during training. The results show that when the test inputs are consistent with the training scenario, methods such as APE, RoPE, and SpRePE all achieve high accuracy. As the rotation angle deviates from 0, the inputs still represent handwritten digits on the sphere, but their distribution gradually shifts away from the training data. Under this more challenging setting for spherical geometry, SpRePE maintains better accuracy than the other encoding schemes, demonstrating its ability to capture spherical geometric structure.

4.2 Panoramic Depth Estimation

We evaluate the effectiveness of SpRePE for depth prediction on the Structure3D dataset. Due to resource constraints, we downsample Structure3D to a resolution of 256×512 and construct the data splits following the prescribed protocol. We train the model using a geometry-weighted ℓ_2 loss. Model and training hyper-parameters are provided in the Appendix D. Following prior work on depth prediction, we report the AbsRel and δ metrics on the test set.

Table 4: **Results on panoramic depth estimation under the non-rotated setting.** Performance is evaluated by AbsRel (lower is better) and threshold accuracies δ_1, δ_2, and δ_3 (higher is better). δ_t denotes the rate of pixels satisfying $\max(\hat{d}/d,\ d/\hat{d}) < 1.25^t$.

PE scheme	AbsRel ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
NoPE	0.3147	0.5655	0.8322	0.9370
APE	0.2627	0.6881	0.8837	0.9520
RPE	0.3119	0.5728	0.8351	0.9374
Sphere2Vec	0.1296	0.8993	0.9637	0.9802
RoPE	0.1084	0.9250	0.9715	0.9843
RingPE	0.1532	0.8744	0.9524	0.9754
SpRePE	0.1296	0.8964	0.9652	0.9818
SpRePE(LM)	0.1438	0.8880	0.9614	0.9794

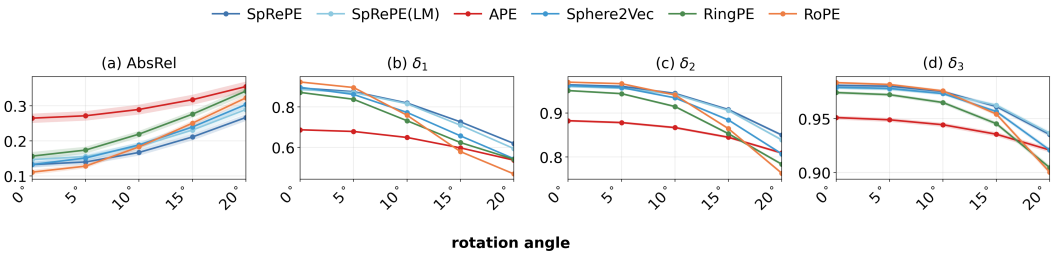

Figure 4: **Results on panoramic depth estimation under the rotated setting.**

Table 4 reports the test results on the original data, while Figure 4 shows how the AbsRel (with image-level standard errors) and δ metrics change when the input is rigidly rotated around the axis normal to the prime-meridian plane. Here, a rotation angle of 0 corresponds to the original data. The results in Table 4 show that SpRePE remains competitive with the baselines on the original data, namely at rotation angle 0, although depth prediction alone does not strongly expose spherical structure. As the rotation angle increases, as shown in Figure 4, understanding spherical geometry becomes increasingly important, and SpRePE achieves a clear advantage over the baselines.

4.3 Weather Forecasting

We verify SpRePE in weather forecasting on ERA5 dataset [12]. Due to limited resource, we train our models on a subset of atmospheric variables, sub-sampled at a temporal frequency of 6 hours and 128×256 spatial resolution of the ERA5 dataset. The input sample has 70 channels corresponding to the surface and upper-air variables at different vertical layers. The variables used are listed in Appendix D. We use six years of ERA5 data (2015-2020): 2015-2018 are used for training, 2019 is used for validation, and 2020 is held out as out-of-sample test set. During training, the network's input is ERA5 fields at time t and is optimized to predict the fields at $t + 6\,\mathrm{h}$, with geometry-weighted RMSE. For validation and testing, the model is initialized with the analysis fields at $t = 0$ and then run auto-regressively for 40 steps, producing forecasts at lead times $+6\,\mathrm{h}, +12\,\mathrm{h}, \ldots, +240\,\mathrm{h}$.

Table 5 summarizes auto-regressive forecasting results on key variables, including geopotential at 500hPa (Z500), temperature at 850hPa (T850) and 2-meter-height air temperature (T2M). At +24h, RoPE attains the best ACC on Z500, whereas SpRePE(LM) is highest on T2M and T850. For long-range skill, at +120h, SpRePE(LM) widens the gap over RoPE and RingPE on T2M and T850 while remaining slightly lower on Z500; by +240h it surpasses RoPE and RingPE across all variables, indicating that the advantage of our method grows from +24h to +120h and culminates in a complete overtake at +240h.

We argue that spherical geometry has a relatively limited effect at short lead times, but becomes increasingly important as the forecast horizon extends. Figure 5 reports the average differences in absolute relative error between SpRePE(LM) and the two baselines, RingPE and RoPE, on the test set after 40-step autoregressive inference. The results show that SpRePE(LM) yields noticeably lower errors than both RoPE and RingPE near the polar regions. By incorporating an explicit sphere-aware

Table 5: **Auto-regressive forecasting results on the ERA5 dataset using different position embedding schemes.** Performance is evaluated by Anomaly Correlation Coefficient (ACC, higher is better). We measure ACC at the 4th(+24h), 20th (+120h) and 40th (+240h) steps. **Bold** denotes the best score; underline marks the second best.

PE scheme	ACC at 24h ↑			ACC at +120h ↑			ACC at +240h ↑		
	$Z500$	$T2M$	$T850$	$Z500$	$T2M$	$T850$	$Z500$	$T2M$	$T850$
NoPE	0.9931	0.9632	0.9755	0.8199	0.8251	0.8044	0.5286	0.6767	0.5770
APE	0.9954	0.9746	0.9795	0.8818	0.9087	0.8653	0.5971	0.8119	0.6539
RPE	0.9963	0.9659	0.9782	0.9000	0.8560	0.8636	0.6037	0.7260	0.6377
Sphere2Vec	0.9954	0.9666	0.9783	0.8813	0.8621	0.8523	0.5911	0.7440	0.6368
RoPE	**0.9971**	0.9683	0.9810	**0.9172**	0.8741	0.8811	0.6215	0.7502	0.6591
RingPE	0.9964	0.9689	0.9808	0.9164	0.9087	0.8903	0.6383	0.8060	0.6780
SpRePE	0.9966	0.9671	0.9790	0.9069	0.8772	0.8732	0.6149	0.7665	0.6573
SpRePE(LM)	0.9966	**0.9760**	**0.9816**	0.9148	**0.9234**	**0.8913**	**0.6414**	**0.8294**	**0.6865**

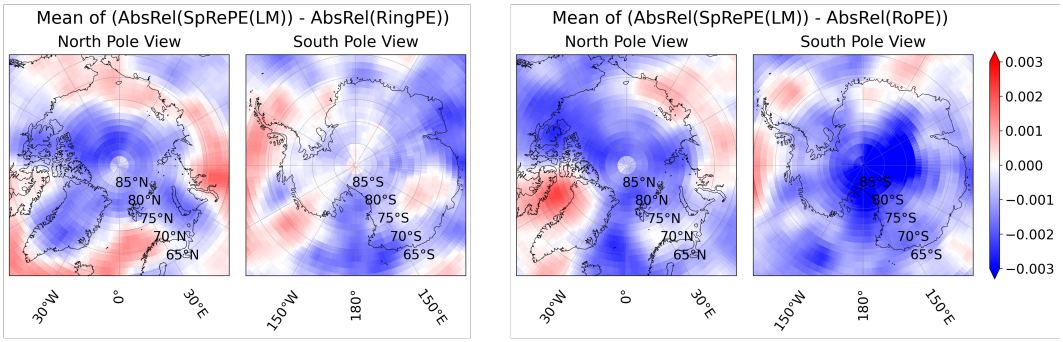

Figure 5: **Polar views of the average difference in absolute relative error of z500 between SpRePE(LM) and RingPE/RoPE after 40-step autoregressive inference on the test set.** Each pixel shows $\mathrm{Avg}\big(\mathrm{AbsRel}_{\mathrm{SpRePE(LM)}} - \mathrm{AbsRel}_{\mathrm{RingPE/RoPE}}\big)$, where $\mathrm{AbsRel} = \frac{|y_{\mathrm{pred}} - y_{\mathrm{gt}}|}{|y_{\mathrm{gt}}|}$. Blue denotes regions where SpRePE(LM) is more accurate (lower AbsRel); red indicates the opposite.

relative-position term into attention, our method better preserves spatial coherence at long lead times and consequently achieves higher ACC. We also provide an illustrative comparison of the attention patterns induced by RoPE and SpRePE on the sphere in Appendix E.

5 Limitations and Future Work

Although SpRePE provides a drop-in spherical geometry-aware position embedding scheme, its current formulation is tailored to positions on the spherical surface S^2 and therefore does not directly generalize to more general geometric domains. In addition, our current implementation and empirical evaluation mainly focus on standard softmax attention in ViT-style backbones, leaving the compatibility between SpRePE and linear attention mechanisms to be further investigated. Future work may extend the reflection-based construction from spherical surfaces to solid balls and other geometric domains, enabling positional structures that capture both angular and radial components. Another promising direction is to adapt SpRePE to linear attention, so that sphere-aware relative geometric information can be retained while preserving the linear complexity of efficient attention mechanisms.

6 Conclusion

We present **SpRePE**, a geometry-aware position embedding scheme for spherical data that operates entirely within the attention module. SpRePE encodes absolute positional information via reflection matrices and augments attention with an explicit spherical relative-position term, while preserving the computational footprint of standard attention and requiring no changes to the backbone. Across

a range of representative tasks, SpRePE demonstrates competitive performance under a matched ViT setup. Its advantages are particularly pronounced in settings where spherical geometry plays a consequential role, such as long-range weather forecasting and polar-region modeling. Taken together, these results establish SpRePE as a simple, efficient, and effective position embedding for Transformer-based modeling on the sphere.

Acknowledgments and disclosures

This work was supported by the National Natural Science Foundation of China (No. U2242210).

References

[1] 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.

[2] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. `'https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf'`, 2019. Accessed: 2025-07-04.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[5] 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.

[6] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions?, 2020.

[7] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2020.

[8] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations, 2018.

[9] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer, 2024.

[10] Patrick Alken, Erwan Thébault, Ciarán D Beggan, Hagay Amit, J Aubert, Julien Baerenzung, TN Bondar, WJ Brown, S Califf, A Chambodut, et al. International geomagnetic reference field: The thirteenth generation. *Earth, Planets and Space*, 73(1):49, 2021.

[11] Planck Collaboration, Aghanim, N., Akrami, Y., Arroja, F., Ashdown, M., Aumont, J., Baccigalupi, C., Ballardini, M., Banday, A. J., Barreiro, R. B., Bartolo, N., Basak, S., Battye, R., Benabed, K., Bernard, J.-P., Bersanelli, M., Bielewicz, P., Bock, J. J., Bond, J. R., Borrill, J., Bouchet, F. R., Boulanger, F., Bucher, M., Burigana, C., Butler, R. C., Calabrese, E., Cardoso, J.-F., Carron, J., Casaponsa, B., Challinor, A., Chiang, H. C., Colombo, L. P. L., Combet, C., Contreras, D., Crill, B. P., Cuttaia, F., de Bernardis, P., de Zotti, G., Delabrouille, J., Delouis, J.-M., Désert, F.-X., Di Valentino, E., Dickinson, C., Diego, J. M., Donzelli, S., Doré, O., Douspis, M., Ducout, A., Dupac, X., Efstathiou, G., Elsner, F., Enßlin, T. A., Eriksen, H. K., Falgarone, E., Fantaye, Y., Fergusson, J., Fernandez-Cobos, R., Finelli, F., Forastieri, F., Frailis, M., Franceschi, E., Frolov, A., Galeotta, S., Galli, S., Ganga, K., Génova-Santos, R. T., Gerbino,

M., Ghosh, T., González-Nuevo, J., Górski, K. M., Gratton, S., Gruppuso, A., Gudmundsson, J. E., Hamann, J., Handley, W., Hansen, F. K., Helou, G., Herranz, D., Hildebrandt, S. R., Hivon, E., Huang, Z., Jaffe, A. H., Jones, W. C., Karakci, A., Keihänen, E., Keskitalo, R., Kiiveri, K., Kim, J., Kisner, T. S., Knox, L., Krachmalnicoff, N., Kunz, M., Kurki-Suonio, H., Lagache, G., Lamarre, J.-M., Langer, M., Lasenby, A., Lattanzi, M., Lawrence, C. R., Le Jeune, M., Leahy, J. P., Lesgourgues, J., Levrier, F., Lewis, A., Liguori, M., Lilje, P. B., Lilley, M., Lindholm, V., López-Caniego, M., Lubin, P. M., Ma, Y.-Z., Macías-Pérez, J. F., Maggio, G., Maino, D., Mandolesi, N., Mangilli, A., Marcos-Caballero, A., Maris, M., Martin, P. G., Martinelli, M., Martínez-González, E., Matarrese, S., Mauri, N., McEwen, J. D., Meerburg, P. D., Meinhold, P. R., Melchiorri, A., Mennella, A., Migliaccio, M., Millea, M., Mitra, S., Miville-Deschênes, M.-A., Molinari, D., Moneti, A., Montier, L., Morgante, G., Moss, A., Mottet, S., Münchmeyer, M., Natoli, P., Nørgaard-Nielsen, H. U., Oxborrow, C. A., Pagano, L., Paoletti, D., Partridge, B., Patanchon, G., Pearson, T. J., Peel, M., Peiris, H. V., Perrotta, F., Pettorino, V., Piacentini, F., Polastri, L., Polenta, G., Puget, J.-L., Rachen, J. P., Reinecke, M., Remazeilles, M., Renault, C., Renzi, A., Rocha, G., Rosset, C., Roudier, G., Rubiño-Martín, J. A., Ruiz-Granados, B., Salvati, L., Sandri, M., Savelainen, M., Scott, D., Shellard, E. P. S., Shiraishi, M., Sirignano, C., Sirri, G., Spencer, L. D., Sunyaev, R., Suur-Uski, A.-S., Tauber, J. A., Tavagnacco, D., Tenti, M., Terenzi, L., Toffolatti, L., Tomasi, M., Trombetti, T., Valiviita, J., Van Tent, B., Vibert, L., Vielva, P., Villa, F., Vittorio, N., Wandelt, B. D., Wehus, I. K., White, M., White, S. D. M., Zacchei, A., and Zonca, A. Planck 2018 results - i. overview and the cosmological legacy of planck. *AA*, 641:A1, 2020.

[12] Hans Hersbach, Bill Bell, Paul Berrisford, Gionata Biavati, András Horányi, Joaquín Muñoz Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Iryna Rozum, et al. Era5 hourly data on pressure levels from 1979 to present. `https://doi.org/10.24381/cds.bd0915c6`, 2018. Accessed: 2025-07-05.

[13] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

[14] Yingzhe Cui, Ruohan Wu, Xiang Zhang, Ziqi Zhu, Bo Liu, Jun Shi, Junshi Chen, Hailong Liu, Shenghui Zhou, Liang Su, et al. Forecasting the eddying ocean with a deep neural network. *Nature Communications*, 16(1):2268, 2025.

[15] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators, 2020.

[16] Zhiheng Huang, Davis Liang, Peng Xu, and Bing Xiang. Improve transformer models with better relative position embeddings. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3327–3335, Online, November 2020. Association for Computational Linguistics.

[17] Guolin Ke, Di He, and Tie-Yan Liu. Rethinking positional encoding in language pre-training, 2021.

[18] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2021.

[19] Ofir Press, Noah A. Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation, 2022.

[20] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[21] Jindong Tian, Yifei Ding, Ronghui Xu, Hao Miao, Chenjuan Guo, and Bin Yang. Arrow: An adaptive rollout and routing method for global weather forecasting. *arXiv preprint arXiv:2510.09734*, 2025.

[22] Ilwi Yun, Chanyong Shin, Hyunku Lee, Hyuk-Jae Lee, and Chae Eun Rhee. Egformer: Equirectangular geometry-biased transformer for 360 depth estimation, 2023.

[23] Boris Bonev, Max Rietmann, Andrea Paris, Alberto Carpentieri, and Thorsten Kurth. Attention on the sphere. *arXiv preprint arXiv:2505.11157*, 2025.

[24] Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical fourier neural operators: Learning stable dynamics on the sphere. In *International conference on machine learning*, pages 2806–2823. PMLR, 2023.

[25] Pranav Jeevan and Amit Sethi. Resource-efficient hybrid x-formers for vision. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2982–2990, 2022.

[26] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, September 2024.

[27] Steven A Orszag. Fourier series on spheres. *Monthly weather review*, 102(1):56–75, 1974.

[28] Gengchen Mai, Krzysztof Janowicz, Bo Yan, Rui Zhu, Ling Cai, and Ni Lao. Multi-scale representation learning for spatial feature distributions using grid cells, 2020.

[29] Gengchen Mai, Krzysztof Janowicz, Yingjie Hu, Song Gao, Bo Yan, Rui Zhu, Ling Cai, and Ni Lao. A review of location encoding for geoai: methods and applications. *International Journal of Geographical Information Science*, 36(4):639–673, January 2022.

[30] Gengchen Mai, Yao Xuan, Wenyun Zuo, Yutong He, Jiaming Song, Stefano Ermon, Krzysztof Janowicz, and Ni Lao. Sphere2vec: A general-purpose location representation learning over a spherical surface for large-scale geospatial predictions, 2023.

[31] Marc Rußwurm, Konstantin Klemmer, Esther Rolf, Robin Zbinden, and Devis Tuia. Geographic location encoding with spherical harmonics and sinusoidal representation networks, 2024.

[32] Yang Liu, Zinan Zheng, Jiashun Cheng, Fugee Tsung, Deli Zhao, Yu Rong, and Jia Li. Cirt: Global subseasonal-to-seasonal forecasting with geometry-inspired transformer. *arXiv preprint arXiv:2502.19750*, 2025.

[33] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Proceedings of The European Conference on Computer Vision (ECCV)*, 2020.

[34] Carlos Esteves, Jean-Jacques Slotine, and Ameesh Makadia. Scaling spherical CNNs. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 9396–9411. PMLR, 23–29 Jul 2023.

[35] Carlos Esteves, Ameesh Makadia, and Kostas Daniilidis. Spin-weighted spherical cnns. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8614–8625. Curran Associates, Inc., 2020.

[36] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pages 606–610. IEEE, 2007.

A Reproducibility statement

We provide an anonymous repository at `https://github.com/jiqingsong/SpRePE`, which contains (i) the core implementation used in our experiments and (ii) environment specifications required to run them. The repository further includes configuration files, scripts for data preparation, training and evaluation entry points.

Appendix D provides a complete description of the datasets used in the experiments. After downloading the raw data, the datasets can be generated using the scripts we provide in the repository.

B Implementation Details of Position Embeddings

B.1 Absolute Position Embedding (APE)

Given an input sequence $\mathbf{X} = [x_1, \ldots, x_n] \in \mathbb{R}^{n \times d}$, APE adds a unique positional vector e_k^{APE} to each token at the stem stage:

$$\widetilde{f}(x_1, \ldots, x_n) = f\big(x_1 + e_1^{\text{APE}},\ x_2 + e_2^{\text{APE}},\ \ldots,\ x_n + e_n^{\text{APE}}\big).$$

APE is commonly implemented in two ways: *sinusoidal* and *learnable*. For ViT on a $H \times W$ grid, let the k-th token have 2D coordinates (p_k^x, p_k^y). The sinusoidal APE of dimension d is defined as

$$\begin{aligned}
e_k^{\text{APE}}(4t) &= \sin\left(\frac{p_k^x}{10^{\,4t/(\lfloor d \rfloor/4)}}\right), \\
e_k^{\text{APE}}(4t+1) &= \cos\left(\frac{p_k^x}{10^{\,4t/(\lfloor d \rfloor/4)}}\right), \\
e_k^{\text{APE}}(4t+2) &= \sin\left(\frac{p_k^y}{10^{\,4t/(\lfloor d \rfloor/4)}}\right), \\
e_k^{\text{APE}}(4t+3) &= \cos\left(\frac{p_k^y}{10^{\,4t/(\lfloor d \rfloor/4)}}\right),
\end{aligned} \tag{8}$$

for $t = 0, 1, \ldots$. Alternatively, $e_k^{\text{APE}} \in \mathbb{R}^d$ can be set as a trainable parameter and learned end-to-end. In our experiments, this learnable APE serves as the APE baseline.

B.2 Relative Position Embedding (RPE)

APE is straightforward and efficient but does not encode relative positions. A common remedy is to add a *relative positional bias* (RPB) directly to the attention logits:

$$\text{Attn}'(i,j) \;=\; \text{Attn}(i,j) \;+\; M_{i,j}^{\text{RPB}},$$

where $M_{i,j}^{\text{RPB}} = e^{\text{RPB}}(\Delta p_{ij})$ and

$$\Delta p_{ij} \;=\; \big(p_i^x - p_j^x,\ p_i^y - p_j^y\big).$$

For a $H \times W$ grid, $e^{\text{RPB}} \in \mathbb{R}^{(2H-1)\times(2W-1)}$ stores a bias value for each possible relative offset. The table e^{RPB} can be either learnable or predefined (e.g., as a deterministic function of inter-token distance). In all comparisons, we adopt a learnable RPB implementation as the RPE baseline.

C Geometric interpretation of SpRePE

In this appendix we make explicit the geometric meaning of SpRePE and explain how it guides the choice of auxiliary points. Recall that for each token position $p_k \in \mathbb{S}^2$ and each auxiliary point $n_\ell \in \mathbb{S}^2$, SpRePE applies a Householder reflection

$$R_{n_\ell}(p_k) \;=\; I - 2\, v_{n_\ell}(p_k)\, v_{n_\ell}(p_k)^\top, \qquad v_{n_\ell}(p_k) \;=\; \frac{n_\ell - p_k}{\big\|n_\ell - p_k\big\|}. \tag{9}$$

The attention inner product between positions p_i and p_j in the subspace associated with n_ℓ involves the matrix $R_{n_\ell}(p_i)^\top R_{n_\ell}(p_j)$. We now show that this matrix admits a precise geometric interpretation as a 3D rotation that encodes the relative position between p_i and p_j on the sphere.

Let $n, p_i, p_j \in \mathbb{S}^2$ be three non-collinear points. They determine a unique plane

$$P_{\{p_i,p_j,n\}} \;=\; \big\{\, n + \alpha\, v_n(p_i) + \beta\, v_n(p_j) : \alpha, \beta \in \mathbb{R} \,\big\}, \tag{10}$$

spanned by $v_n(p_i)$ and $v_n(p_j)$. A unit normal of this plane is given by

$$u \;=\; \frac{v_n(p_i) \times v_n(p_j)}{\big\| v_n(p_i) \times v_n(p_j) \big\|_2}, \tag{11}$$

which is well-defined because $v_n(p_i)$ and $v_n(p_j)$ are linearly independent.

We can now state the main geometric result.

Proposition (Geometric form of the SpRePE encoding). Let $n, p_i, p_j \in \mathbb{S}^2$ be non-collinear points and define the Householder reflections

$$R_n(p_i) \;=\; I - 2\, v_n(p_i)\, v_n(p_i)^\top, \qquad R_n(p_j) \;=\; I - 2\, v_n(p_j)\, v_n(p_j)^\top. \tag{12}$$

Then the matrix $R_n(p_i)^\top R_n(p_j)$ satisfies:

1. It is a proper rotation in $\mathbb{R}^3$ (orthogonal with determinant $+1$);
2. Its rotation axis is the line spanned by u, i.e., the normal vector to $P_{\{p_i,p_j,n\}}$;
3. Restricted to the circle $P_{\{p_i,p_j,n\}} \cap \mathbb{S}^2$, this rotation maps p_j to p_i, and the rotation angle equals the central angle between p_i and p_j along that circle.

Proof. We first show that u is a rotation axis of $R_n(p_i)^\top R_n(p_j)$. Since u is orthogonal to both $v_n(p_i)$ and $v_n(p_j)$, we have

$$u^\top v_n(p_i) = u^\top v_n(p_j) = 0. \tag{13}$$

For $m \in \{i, j\}$,

$$R_n(p_m)\, u \;=\; \big(I - 2\, v_n(p_m) v_n(p_m)^\top\big) u \;=\; u, \tag{14}$$

so u is an eigenvector of both $R_n(p_i)$ and $R_n(p_j)$ with eigenvalue 1, and hence

$$R_n(p_i)^\top R_n(p_j)\, u \;=\; R_n(p_i)^\top (R_n(p_j)\, u) \;=\; R_n(p_i)^\top u \;=\; u. \tag{15}$$

Thus u is an eigenvector of $R_n(p_i)^\top R_n(p_j)$ with eigenvalue 1, so the line spanned by u is a candidate rotation axis.

Each $R_n(p_m)$ is a Householder reflection and therefore an orthogonal matrix with determinant -1. Their product $R_n(p_i)^\top R_n(p_j)$ is thus orthogonal with determinant $+1$. An orthogonal 3×3 matrix with determinant $+1$ and a real eigenvalue 1 must be a proper rotation about the eigenspace associated with eigenvalue 1. This establishes items (1) and (2).

For item (3), note that $R_n(p_j)$ is a reflection across the plane orthogonal to $v_n(p_j)$ and passing through the origin. Geometrically, this reflection swaps n and p_j while preserving the circle $P_{\{p_i,p_j,n\}} \cap \mathbb{S}^2$. Similarly, $R_n(p_i)$ swaps n and p_i and also preserves the same circle. Consequently, the composition $R_n(p_i)^\top R_n(p_j)$ maps p_j to p_i along that circle. Since the axis is orthogonal to $P_{\{p_i,p_j,n\}}$, the restriction of $R_n(p_i)^\top R_n(p_j)$ to $P_{\{p_i,p_j,n\}} \cap \mathbb{S}^2$ is a planar rotation taking p_j to p_i, and its angle is exactly the central angle between p_i and p_j on that circle. □

This proposition has two immediate consequences for SpRePE.

First, it provides a clear geometric interpretation of the encoding. For each auxiliary point n_ℓ, the associated SpRePE subspace effectively represents the relative position between tokens p_i and p_j as a 3D rotation: the rotation axis is the normal of the plane $P_{\{p_i,p_j,n_\ell\}}$ determined by p_i, p_j, and n_ℓ, and the rotation angle is the central angle between p_i and p_j along the circle $P_{\{p_i,p_j,n_\ell\}} \cap \mathbb{S}^2$. In this sense, relative positions on the sphere are encoded not by arbitrary linear transforms but by geometrically interpretable rotations.

Second, the same geometric characterization directly motivates our rule for choosing auxiliary points. If two auxiliary points n_ℓ and n_q yield the same plane $P_{\{p_i,p_j,n\}}$ for given p_i and p_j, then

$$R_{n_\ell}(p_i)^\top R_{n_\ell}(p_j) \;=\; R_{n_q}(p_i)^\top R_{n_q}(p_j),$$

so the corresponding subspaces carry redundant positional information. Accordingly, our selection scheme is designed so that, given all positions p in advance, each auxiliary point is chosen to ensure that the resulting planes $P_{\{p_i,p_j,n_\ell\}}$ are always distinct, thereby avoiding the above degeneracies. An ablation experiment on the auxiliary-point selection scheme is provided in Appendix F.

Table 6: Variables used in our experiments. The "Type" column indicates whether the variable represents a static property, a time-varying single-level property (e.g., surface variables are included), or a time-varying atmospheric property.

Type	Variable name	Short name	Role
Atmospheric	Geopotential	Z	Input/Predicted
Atmospheric	Specific humidity	Q	Input/Predicted
Atmospheric	Temperature	T	Input/Predicted
Atmospheric	U component of wind	U	Input/Predicted
Atmospheric	V component of wind	V	Input/Predicted
Single	2 metre temperature	T2M	Input/Predicted
Single	2 metre Specific humidity	Q2M	Input/Predicted
Single	10 metre u wind component	10U	Input/Predicted
Single	10 metre v wind component	10V	Input/Predicted
Single	Mean sea level pressure	msl	Input/Predicted
Static	Land-sea mask	lsm	Input
Static	Geopotential at surface	Z	Input
Clock	Zenith Angle	n/a	Input

D Experiment setting

Computing environment. Experiments are conducted on a single node with 8× NVIDIA A100-PCIe-40GB GPUs. The NVIDIA driver version is 550.54.14 with CUDA 12.4. The host CPU is a dual-socket Intel Xeon Gold 6248R @ 3.0 GHz (24 cores per socket; physical cores / 96 threads in total), organized into two NUMA nodes with AVX/AVX2/AVX-512 support.

Spherical image classification Dataset preparation follows [34, 35]. After downloading the original MNIST dataset, the Spherical MNIST variant can be generated using the scripts provided in our repository. We use a standard Vision Transformer (ViT) with patch size 2×2, length 8, a hidden dimension of 384, and 8 attention heads. Training uses a batch size of 32 and an initial learning rate of 3×10^{-4}. We optimize with Adam ($\beta_1 = 0.9,\ \beta_2 = 0.999$) and apply a cosine-annealing learning-rate scheduler for 30 epochs.

Panoramic Depth Estimation In experiments, we use a standard Vision Transformer (ViT) with patch size 8×8, length 8, a hidden dimension of 384, and 4 attention heads. Training uses a batch size of 16 and an initial learning rate of 3×10^{-4}. We optimize with Adam ($\beta_1 = 0.9,\ \beta_2 = 0.999$) and apply a cosine-annealing learning-rate scheduler for 20 epochs.

Weather Forecasting Table6 lists all variables used in the experiments. For all atmospheric variables, we use 13 pressure levels: 50, 100, 150, 200, 250, 300, 400, 500, 600, 700, 850, 925, and 1000 hPa.

In experiments, We use a standard Vision Transformer (ViT) with patch size 2×2, length 12, a hidden dimension of 384, and 8 attention heads. Training uses a batch size of 4 and an initial learning rate of 5×10^{-4}. We optimize with Adam ($\beta_1 = 0.9,\ \beta_2 = 0.999$) and apply a cosine-annealing learning-rate scheduler for 20 epochs.

E Attention Visualization Results of SpRePE

Figure 6 further illustrates the geometric effect of SpRePE by visualizing attention maps on the sphere. We consider two extreme query locations: one near the North Pole and one at the intersection of the equator and the prime meridian. When the query lies near the pole, RoPE produces a strip-like attention pattern along the top rows of the ERP grid, whereas SpRePE concentrates attention in a localized region around the pole that decays smoothly toward lower latitudes, consistent with geodesic distance. When the query is placed on the equator at the image boundary, RoPE largely ignores the wrap-around adjacency between the left and right edges, while SpRePE correctly assigns high attention to points that are far apart in the ERP image but adjacent on the sphere. These

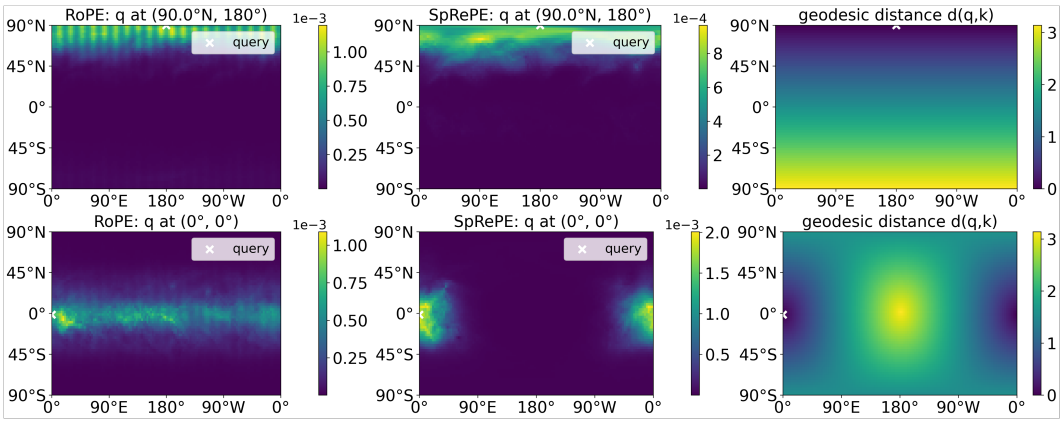

Figure 6: Attention maps of a representative attention head for queries at the North Pole (top) and at the equator–prime-meridian intersection (bottom), together with the corresponding geodesic distance. Across test samples, this head consistently shows that SpRePE aligns attention with spherical geometry more faithfully than RoPE.

Table 7: Auto-regressive forecasting results on the ERA5 dataset using different auxiliary-point selection strategies. Performance is evaluated by Anomaly Correlation Coefficient (ACC, higher is better). We measure ACC at the 4th(+24h), 20th (+120h) and 40th (+240h) steps.

PE scheme	**ACC at 24h** ↑			**ACC at +120h** ↑			**ACC at +240h** ↑		
	$Z500$	$T2M$	$T850$	$Z500$	$T2M$	$T850$	$Z500$	$T2M$	$T850$
Random	0.9962	0.9668	0.9793	0.9047	0.8738	0.8718	0.6181	0.7629	0.6591
Grid	0.9965	0.9671	0.9794	0.9095	0.8786	0.8767	0.6155	0.7674	0.6623
Learnable	0.9964	0.9661	0.9788	0.9089	0.8722	0.8736	0.6144	0.7557	0.6587
Identical	0.9962	0.9655	0.9782	0.8943	0.8642	0.8621	0.6050	0.7510	0.6475

visualizations provide an intuitive view of the "attention geometry" induced by SpRePE and support the interpretability of its geometry-aware design.

F Ablation Studies: Auxiliary-Point Selection and Masking

In this section, we conduct ablation studies on two aspects: the auxiliary-point selection strategy and the masking strategy.

Auxiliary-point selection. We compare the following four strategies:

- **Random**: In each layer, auxiliary points are sampled uniformly at random on the sphere and kept fixed;
- **Grid**: In each layer, auxiliary points are chosen from a fixed latitude–longitude grid and kept fixed;
- **Learnable**: In each layer, auxiliary points are initialized randomly and optimized jointly with the model parameters;
- **Identical**: In each layer, all auxiliary points coincide at the same location.

We adopt the same experimental setup as in the main-text experiments on the ERA5 dataset. The results are presented in Table 7.

The results show that the identical-selection strategy performs the worst, which is consistent with our geometric analysis: in this case, for arbitrary auxiliary points n_ℓ, n_q and positions p_i, p_j, the planes $P_{\{p_i,p_j,n_\ell\}}$ and $P_{\{p_i,p_j,n_q\}}$, determined by (p_i, p_j, n_ℓ) and (p_i, p_j, n_q) respectively, always satisfy

$$P_{\{p_i,p_j,n_\ell\}} = P_{\{p_i,p_j,n_q\}}. \tag{16}$$

Table 8: Auto-regressive forecasting results on the ERA5 dataset using different masking strategies. Performance is evaluated by Anomaly Correlation Coefficient (ACC, higher is better). We measure ACC at the 4th(+24h), 20th (+120h) and 40th (+240h) steps.

PE scheme	**ACC at 24h** ↑			**ACC at +120h** ↑			**ACC at +240h** ↑		
	$Z500$	$T2M$	$T850$	$Z500$	$T2M$	$T850$	$Z500$	$T2M$	$T850$
Fixed-ratio	0.9965	0.9758	0.9813	0.9121	0.9230	0.8886	0.6333	0.8258	0.6783
Density-adaptive	0.9964	0.9754	0.9809	0.9110	0.9226	0.8882	0.6237	0.8168	0.6705
Learnable	0.9966	0.9760	0.9816	0.9148	0.9234	0.8913	0.6414	0.8294	0.6865

As a result, different subspaces are not distinguished and the positional information becomes redundant. Random, grid-based, and learnable selection yield comparable performance. This indicates that as long as the auxiliary points satisfy a basic non-degeneracy condition, SpRePE is quite robust to the specific choice of auxiliary-point selection scheme.

Masking Strategy Beyond the masking scheme considered in the main text (learnable mask), we also evaluate two heuristic masking strategies.

- **Fixed-ratio mask**: SpRePE is applied on a constant fraction ρ of elements (e.g., $\rho = 7/8$), i.e., $M(j) = \mathbf{1}\{j \leq \lfloor \rho d \rfloor\}$.
- **Density-adaptive mask**: let $\rho_i \in [\rho_{\min}, \rho_{\max}]$ depend on token i's local sampling density. For equirectangular grids, we use the area weight $a(\theta_i) = \cos\theta_i$ ($\theta_i \in [-\pi/2, \pi/2]$, proportional to solid-angle per token) and set $\rho_i = \mathrm{clip}(\rho_{\max}\, a(\theta_i),\ \rho_{\min},\ \rho_{\max})$.

All experiments are conducted on ERA5 using the same backbone and training setup as in the main weather-forecasting experiments. The results are presented in Table 8.

Empirically, the learnable mask yields the best overall ACC, surpassing the fixed-ratio design by a small but consistent margin across all variables. Because the gate is shared across heads and tokens within each layer, it introduces virtually no additional computational or parameter overhead. We therefore adopt the learnable-mask configuration as our primary setting in the experiments, and include the other variants as ablations.

G Ablation Study: RoPE with Learnable Mask

To disentangle the effect of the learnable mask from the geometric effect of SpRePE, we further evaluate RoPE equipped with the same learnable masking strategy, denoted as RoPE(LM). Applying the same mask to RoPE therefore provides a controlled way to assess whether the observed gains mainly come from the masking mechanism itself or from the spherical reflection-based positional encoding.

We use the same ERA5 experimental setup as in the main text and report the results in Table 9. RoPE(LM) improves over RoPE, which suggests that learnable masking can be beneficial for query/key-based positional embeddings in general. However, the improvement is not sufficient to fully account for the performance of SpRePE(LM). In particular, SpRePE(LM) achieves the best ACC on most metrics at the longer forecast horizon, including all three variables at +240h. These results indicate that the learnable mask is a useful stabilization component, while the spherical reflection formulation still provides additional benefits beyond masking alone.

H Backbone Generalization

To evaluate whether SpRePE can be effectively integrated into Transformer architectures beyond the standard ViT backbone, we conduct additional experiments on two representative spherical modeling frameworks: EGFormer for panoramic depth estimation and S2Transformer for global weather forecasting. For EGFormer, we follow the official implementation and training pipeline, while for S2Transformer, which does not provide an ERA5 forecasting benchmark in its original setting, we adapt the model into our experimental framework and evaluate it under the same forecasting protocol.

Table 9: Auto-regressive forecasting results on the ERA5 dataset for evaluating the effect of the learnable masking. Performance is measured by the Anomaly Correlation Coefficient (ACC; higher is better). We report ACC at the 20th (+120h), and 40th (+240h) forecast steps.

PE scheme	**ACC at +120h ↑**			**ACC at +240h ↑**		
	$Z500$	$T2M$	$T850$	$Z500$	$T2M$	$T850$
RoPE	0.9172	0.8741	0.8811	0.6215	0.7502	0.6591
RoPE(LM)	**0.9174**	0.9189	**0.8927**	0.6376	0.8150	0.6799
SpRePE	0.9089	0.8722	0.8736	0.6144	0.7557	0.6587
SpRePE(LM)	0.9148	**0.9234**	0.8913	**0.6414**	**0.8294**	**0.6865**

Table 10: Evaluation of SpRePE with EGFormer on the Structured3D dataset. The comparison is conducted within the same EGFormer framework and training pipeline. Lower AbsRel is better, while higher δ values indicate better performance.

Method	AbsRel ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
EGFormer + original PE	0.0262	0.9884	0.9956	0.9976
EGFormer + SpRePE	**0.0253**	**0.9890**	**0.9958**	0.9976

Due to differences in backbone architectures and experimental pipelines, the absolute performance values are not directly comparable with those reported in the main experiments. Therefore, we focus on the relative improvements obtained by replacing the original positional encoding with SpRePE within each backbone architecture.

H.1 EGFormer on Structured3D

EGFormer [22] introduces a geometry-aware attention mechanism for panoramic depth estimation. To evaluate the compatibility of SpRePE with specialized attention architectures, we integrate SpRePE into EGFormer and conduct experiments on the Structured3D dataset under the official implementation and training protocol. Since the experiment follows native implementation and training pipeline of EGformer rather than the matched framework used in main experiments, its absolute values are not directly comparable with the Structured3D results reported in the manuscript.

Table 10 reports the comparison between the original positional encoding of EGFormer and SpRePE. By replacing only the positional encoding scheme while keeping the remaining architecture unchanged, SpRePE improves AbsRel from 0.0262 to 0.0253 and provides consistent improvements on δ_1 and δ_2. These results indicate that SpRePE can be effectively combined with geometry-aware attention mechanisms and provide additional spherical positional information beyond the original backbone design.

H.2 S2Transformer on ERA5

To further evaluate the applicability of SpRePE beyond the ViT backbone, we integrate SpRePE into S2Transformer for global weather forecasting. S2Transformer is designed specifically for spherical data modeling, providing a complementary evaluation setting with a different backbone architecture. We follow the same ERA5 forecasting setup as described in the main experiments and compare the original positional encoding with SpRePE under identical training and evaluation settings.

Table 11 reports autoregressive forecasting results measured by Anomaly Correlation Coefficient (ACC). Compared with the original positional encoding, replacing it with SpRePE substantially improves long-range forecasting performance.

I Regional Effective-Rank Analysis

We further examine the effect of the learnable mask on regional attention-score matrices in the ERA5 setting. This analysis complements the channel-wise positional modulation mechanism by comparing the effective ranks obtained with SpRePE and SpRePE(LM) in equatorial and polar regions.

Table 11: Autoregressive forecasting results on the ERA5 dataset using S2Transformer with different positional encoding schemes. Performance is evaluated by Anomaly Correlation Coefficient (ACC, higher is better). We report ACC at the 4th (+24 h) and 20th (+120 h) forecasting steps.

Architecture / PE	**ACC at +24 h ↑**			**ACC at +120 h ↑**		
	$Z500$	$T2M$	$T850$	$Z500$	$T2M$	$T850$
S2Transformer + original PE	0.9893	0.9380	0.9644	0.1626	0.3454	0.3034
S2Transformer + SpRePE	0.9956	0.9398	0.9701	0.8692	0.6281	0.8122

Using held-out test samples, we define the equatorial region by $|\mathrm{lat}| < 15^\circ$ and the polar region by $|\mathrm{lat}| > 75^\circ$. We sample the same number of tokens from the two regions, ensuring that the regional attention-score matrices have the same dimensions. For each Transformer layer, attention head, and test sample, we construct

$$S_{\mathcal{R}} = \frac{Q_{\mathcal{R}} K_{\mathcal{R}}^{\top}}{\sqrt{d_{\mathrm{head}}}}, \tag{17}$$

where $\mathcal{R}$ denotes the sampled tokens within a region, $Q_{\mathcal{R}}$ and $K_{\mathcal{R}}$ are their queries and keys after positional modulation, and d_{head} is the dimension of each attention head.

Following [36], we compute the entropy-based effective rank of $S_{\mathcal{R}}$ from its singular values $\{\sigma_i\}$:

$$p_i = \frac{\sigma_i}{\sum_j \sigma_j}, \qquad r_{\mathrm{eff}}(S_{\mathcal{R}}) = \exp\left(-\sum_i p_i \log p_i\right), \tag{18}$$

where $\log$ denotes the natural logarithm and $0 \log 0 = 0$. This statistic summarizes the concentration of the singular-value distribution, with a lower effective rank indicating greater concentration.

At each layer, we compute the effective rank separately for each attention head and test sample, and then average the resulting values within each region. We report the equator–pole effective-rank gap

$$\Delta r_{\mathrm{eff}}^{(\ell)} = \overline{r}_{\mathrm{eff}}^{(\ell,\mathrm{equator})} - \overline{r}_{\mathrm{eff}}^{(\ell,\mathrm{polar})}, \tag{19}$$

where ℓ indexes the Transformer layer and the overline denotes averaging over attention heads and test samples.

Table 12 reports the regional effective-rank gap across all 12 Transformer layers. For both variants, the gap is positive at every layer, indicating a lower mean effective rank in the polar region than in the equatorial region. SpRePE(LM) exhibits a smaller gap at every layer and reduces the layer-averaged gap from 12.3 to 5.4. These results show that, in this evaluation, incorporating the learnable mask makes the effective ranks of SpRePE's regional attention-score matrices more similar across the equatorial and polar regions.

J Runtime and Memory Comparison

To complement the asymptotic overhead analysis in the main text, we report empirical forward runtime and peak memory usage for different PE schemes under a controlled setting. The backbone follows the ViT architecture used in the weather forecasting experiments, and we vary the number of tokens N and the embedding dimension d while keeping the remaining configuration fixed. This setup isolates the practical cost introduced by each positional embedding mechanism.

Wall-clock time and peak memory usage depend on implementation details, kernel optimization, software stack, and hardware. Therefore, these measurements should be interpreted as empirical complements to the asymptotic analysis rather than implementation-independent constants. Under this setting, explicit RPE incurs substantially higher runtime and memory cost as N increases and runs out of memory at the largest resolution. In contrast, RoPE and SpRePE remain close to the NoPE and APE baselines, while SpRePE(LM) introduces additional runtime due to the learnable masking operation but keeps memory usage at a comparable scale.

Table 12: Equator–pole effective-rank gap Δr_{eff} in the ERA5 setting. Effective ranks are averaged over attention heads and test samples before computing the regional difference. Positive values indicate a lower mean effective rank in the polar region. The final row reports the average gap over all 12 layers.

Layer	**SpRePE**	**SpRePE(LM)**
0	9.4	4.1
1	4.8	2.9
2	7.8	3.2
3	14.2	6.7
4	14.3	4.1
5	14.8	6.2
6	15.4	4.9
7	16.8	2.8
8	13.3	10.4
9	10.5	9.2
10	12.5	6.8
11	13.9	3.7
Average	**12.3**	**5.4**

Table 13: Comparison of forward runtime(ms) for ViT models equipped with different positional embedding schemes under varying numbers of tokens N and embedding dimensions d. The model architecture follows that used in the weather forecasting experiments.

PE scheme	$N = 4096$			$N = 8192$			$N = 16384$		
$d =$	192	384	576	192	384	576	192	384	576
NoPE	50.49	65.05	85.07	170.92	210.28	258.60	684.84	802.02	927.29
APE	51.11	65.19	85.32	173.22	210.82	259.54	685.48	801.92	929.66
RPE	118.79	131.46	150.76	440.97	476.55	522.37	OOM	OOM	OOM
RoPE	52.58	67.09	87.64	175.31	213.99	263.46	689.81	804.18	938.39
SpRePE	52.95	68.14	89.67	175.50	217.59	268.58	691.60	811.19	950.46
SpRePE(LM)	65.20	83.37	106.07	192.53	243.24	313.10	725.09	864.09	1068.25

Table 14: Comparison of peak memory usage (GB) during the forward pass for ViT models equipped with different positional embedding schemes under varying numbers of tokens N and embedding dimensions d. The model architecture follows that used in the weather forecasting experiments.

PE scheme	$N = 4096$			$N = 8192$			$N = 16384$		
$d =$	192	384	576	192	384	576	192	384	576
NoPE	1.05	1.13	1.26	4.08	4.18	4.32	16.13	16.27	16.45
APE	1.06	1.14	1.26	4.08	4.19	4.34	16.14	16.29	16.49
RPE	3.06	3.14	3.26	12.09	12.19	12.33	OOM	OOM	OOM
RoPE	1.06	1.14	1.27	4.09	4.20	4.35	16.15	16.31	16.51
SpRePE(LM)	1.06	1.15	1.27	4.09	4.20	4.36	16.15	16.32	16.52

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in the Limitations and Future work section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the full set of assumptions and a complete proof in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and conclusions of the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: An repository is provided in Appendix A.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We specify the Experimental setting details in Appendix D

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper reports error bars for the rotated setting in the S3D experiment.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: For each experiment, we provide sufficient information on the computer resources needed to reproduce the experiments in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our work has significant value for AI for weather and AI for physics.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly credits the code and datasets used, explicitly states their licenses and terms of use, and appears to respect the corresponding usage requirements.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.