Lemma 3.

ComRoPE: Scalable and Robust Rotary Position Embedding Parameterized by Trainable Commuting Angle Matrices

Supplementary Material

A. Theorems and proofs

A.1. Proof of the main theorem

To prove our main theorem (*i.e.*, Theorem 1), we first propose some lemmas and prove them.

Lemma 1. *Matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ commute if and only if $e^{\mathbf{A}x}e^{\mathbf{B}y} = e^{\mathbf{A}x+\mathbf{B}y}$ for all $x, y \in \mathbb{R}$.*

Proof.
1) Necessity (⇒). By the definition of $e^{\mathbf{A}}$, we have:

$$\begin{aligned} e^{\mathbf{A}+\mathbf{B}} &= \sum_{n=0}^{\infty}\frac{(\mathbf{A}+\mathbf{B})^n}{n!} \\ &= \sum_{n=0}^{\infty}\frac{\sum_{k=0}^{n}\binom{n}{k}\mathbf{A}^k\mathbf{B}^{n-k}}{n!} \\ &= \sum_{n=0}^{\infty}\sum_{k=0}^{n}\frac{\mathbf{A}^k\mathbf{B}^{n-k}}{k!(n-k)!} \\ &= \left(\sum_{k=0}^{\infty}\frac{\mathbf{A}^k}{k!}\right)\left(\sum_{m=0}^{\infty}\frac{\mathbf{B}^m}{m!}\right) \\ &= e^{\mathbf{A}}e^{\mathbf{B}}. \end{aligned} \tag{13}$$

Substituting $\mathbf{A}, \mathbf{B}$ with $\mathbf{A}x, \mathbf{B}y$, we obtain:

$$e^{\mathbf{A}x+\mathbf{B}y} = e^{\mathbf{A}x}e^{\mathbf{B}y}. \tag{14}$$

2) Sufficiency (⇐). We have:

$$\begin{aligned} e^{\mathbf{A}t}e^{\mathbf{B}t} &= \left(\sum_{n=0}^{\infty}\frac{t^n\mathbf{A}^n}{n!}\right)\left(\sum_{m=0}^{\infty}\frac{t^m\mathbf{B}^m}{m!}\right) \\ &= \mathbf{I} + t(\mathbf{A}+\mathbf{B}) + t^2\cdot\frac{\mathbf{A}^2+2\mathbf{AB}+\mathbf{B}^2}{4} + o(t^2), \end{aligned} \tag{15}$$

and

$$\begin{aligned} e^{(\mathbf{A}+\mathbf{B})t} &= \sum_{n=0}^{\infty}\frac{((\mathbf{A}+\mathbf{B})t)^n}{n!} \\ &= \mathbf{I} + t(\mathbf{A}+\mathbf{B}) + t^2\cdot\frac{(\mathbf{A}+\mathbf{B})^2}{4} + o(t^2). \end{aligned} \tag{16}$$

Let $t^2 f(t)$ be the difference between the two expressions above. Thus, we obtain:

$$\begin{aligned} f(t) &= \frac{e^{\mathbf{A}t}e^{\mathbf{B}t} - e^{(\mathbf{A}+\mathbf{B})t}}{t^2} \\ &= \frac{\mathbf{A}^2+2\mathbf{AB}+\mathbf{B}^2}{4} - \frac{(\mathbf{A}+\mathbf{B})^2}{4} + o(1) \\ &= \frac{\mathbf{AB}-\mathbf{BA}}{4} + o(1). \end{aligned} \tag{17}$$

Taking the limit as $t \to 0$, we have:

$$\lim_{t\to 0} f(t) = \frac{\mathbf{AB}-\mathbf{BA}}{4}. \tag{18}$$

Since $e^{\mathbf{A}x}e^{\mathbf{B}y} = e^{\mathbf{A}x+\mathbf{B}y}$, we have $f(t) = 0$, which implies $\mathbf{AB} = \mathbf{BA}$.

■

Lemma 2. *Matrices $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m \in \mathbb{R}^{n\times n}$ $(m > 1)$ pairwise commute if and only if:*

$$e^{\mathbf{A}_1x_1}e^{\mathbf{A}_2x_2}\cdots e^{\mathbf{A}_mx_m} = e^{\mathbf{A}_1x_1+\mathbf{A}_2x_2+\cdots+\mathbf{A}_mx_m} \tag{19}$$

for all $x_1, x_2, \ldots, x_m \in \mathbb{R}$.

Proof. For $m = 2$, the theorem holds by Lemma 1. Suppose the theorem holds for all $2 \le m \le k$. We prove it for $m = k+1$.
1) Necessity (⇒). Assuming:

$$e^{\mathbf{A}_1x_1}e^{\mathbf{A}_2x_2}\cdots e^{\mathbf{A}_kx_k} = e^{\mathbf{A}_1x_1+\mathbf{A}_2x_2+\cdots+\mathbf{A}_kx_k}, \tag{20}$$

we split $\mathbf{A}_1x_1 + \mathbf{A}_2x_2 + \cdots + \mathbf{A}_{k+1}x_{k+1}$ into two parts:

$$\begin{aligned} &\mathbf{A}_1x_1 + \mathbf{A}_2x_2 + \cdots + \mathbf{A}_{k+1}x_{k+1} \\ &= (\mathbf{A}_1x_1 + \mathbf{A}_2x_2 + \cdots + \mathbf{A}_kx_k) + (\mathbf{A}_{k+1}x_{k+1}). \end{aligned} \tag{21}$$

Since $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{k+1}$ commute in pairs, $\mathbf{A}_1x_1 + \mathbf{A}_2x_2 + \cdots + \mathbf{A}_kx_k$ and $\mathbf{A}_{k+1}x_{k+1}$ also commute. Thus:

$$\begin{aligned} &e^{\mathbf{A}_1x_1+\mathbf{A}_2x_2+\cdots+\mathbf{A}_{k+1}x_{k+1}} \\ &= e^{(\mathbf{A}_1x_1+\mathbf{A}_2x_2+\cdots+\mathbf{A}_kx_k)}e^{\mathbf{A}_{k+1}x_{k+1}} \\ &= e^{\mathbf{A}_1x_1}e^{\mathbf{A}_2x_2}\cdots e^{\mathbf{A}_{k+1}x_{k+1}}. \end{aligned} \tag{22}$$

2) Sufficiency (⇐). Let $x_{k+1} = 0$. Then:

$$e^{\mathbf{A}_1x_1}e^{\mathbf{A}_2x_2}\cdots e^{\mathbf{A}_kx_k} = e^{\mathbf{A}_1x_1+\mathbf{A}_2x_2+\cdots+\mathbf{A}_kx_k}, \tag{23}$$

implying that $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k$ commute in pairs.

Position Encoding Method	Perturbation Intensity	Evaluation Resolution								
		112	128	192	224	256	320	384	448	512
APE	1	48.10	55.25	76.50	93.10	76.70	71.48	74.36	62.23	53.18
	0	45.54	55.18	76.48	92.87	75.91	70.70	73.70	60.43	48.79
Vanilla RoPE	1	47.28	54.96	75.69	93.79	77.66	72.53	74.72	65.19	57.34
	0	48.12	55.21	76.47	94.12	76.72	71.59	74.47	62.28	53.99
LieRE	1	48.97	56.15	77.33	$\underline{94.43}$	78.35	73.20	$\underline{77.33}$	65.74	58.23
	0	48.75	55.46	78.16	94.24	78.91	72.92	76.86	65.35	56.85
ComRoPE-AP	1	**50.14**	55.63	77.47	94.37	79.27	73.56	76.66	**67.68**	$\underline{59.34}$
	0	48.06	55.63	75.72	94.26	75.75	70.93	74.72	64.91	57.98
ComRoPE-LD	1	$\underline{49.89}$	**56.60**	**79.21**	94.24	**80.27**	**74.22**	**78.60**	$\underline{67.46}$	**60.39**
	0	48.70	$\underline{56.46}$	$\underline{78.30}$	**94.48**	$\underline{79.27}$	$\underline{74.58}$	76.86	66.02	57.68

Table 3. Accuracy of 3D classification on UCF-101. Models are trained at a resolution of 224×224 and evaluated at varying resolutions.

For any $p \in \{1, 2, \dots, k\}$, set all $x_i = 0$ except for x_p and x_{k+1}. This yields:

$$e^{\mathbf{A}_p x_p} e^{\mathbf{A}_{k+1} x_{k+1}} = e^{\mathbf{A}_p x_p + \mathbf{A}_{k+1} x_{k+1}}, \tag{24}$$

which implies $\mathbf{A}_p$ and $\mathbf{A}_{k+1}$ commute. Thus, $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_{k+1}$ commute in pairs.

■

Matrices $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_m \in \mathbb{R}^{n\times n}$ $(m > 1)$ *pairwise commute if and only if there exists a function* $f : \mathbb{R}^m \to \mathbb{R}^{n\times n}$ *such that:*

$$\begin{aligned} &f(x_1 + y_1, x_2 + y_2, \dots, x_m + y_m) \\ &= e^{\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \cdots + \mathbf{A}_m x_m} e^{\mathbf{A}_1 y_1 + \mathbf{A}_2 y_2 + \cdots + \mathbf{A}_m y_m} \end{aligned} \tag{25}$$

for all $x_1, y_1, x_2, y_2, \dots, x_m, y_m \in \mathbb{R}$.

Proof. **1) Necessity ($\Rightarrow$).** By Lemma 2, we can easily verify that the following f satisfies the condition:

$$\begin{aligned} &f(x_1 + y_1, x_2 + y_2, \dots, x_m + y_m) \\ &= e^{\mathbf{A}_1(x_1+y_1) + \mathbf{A}_2(x_2+y_2) + \cdots + \mathbf{A}_m(x_m+y_m)}. \end{aligned} \tag{26}$$

2) Sufficiency ($\Leftarrow$). From Equation 31, let x_k be replaced with $x_k + y_k$ and y_k with 0. We obtain:

$$\begin{aligned} &f(x_1 + y_1, x_2 + y_2, \dots, x_m + y_m) \\ &= e^{\mathbf{A}_1(x_1+y_1) + \cdots + \mathbf{A}_m(x_m+y_m)} e^{\mathbf{A}_1 \cdot 0 + \cdots + \mathbf{A}_m \cdot 0} \\ &= e^{\mathbf{A}_1(x_1+y_1) + \cdots + \mathbf{A}_m(x_m+y_m)}. \end{aligned} \tag{27}$$

Comparing this with Equation equation 31, we get:

$$\begin{aligned} &e^{\mathbf{A}_1 x_1 + \mathbf{A}_2 x_2 + \cdots + \mathbf{A}_m x_m} e^{\mathbf{A}_1 y_1 + \mathbf{A}_2 y_2 + \cdots + \mathbf{A}_m y_m} \\ &= e^{\mathbf{A}_1(x_1+y_1) + \cdots + \mathbf{A}_m(x_m+y_m)}. \end{aligned} \tag{28}$$

For any $i, j \in \{1, 2, \dots, m\}$, set $x_k = 0$ for all $k \neq i$ and $y_k = 0$ for all $k \neq j$. This leads to:

$$e^{\mathbf{A}_i x_i} e^{\mathbf{A}_j y_j} = e^{\mathbf{A}_i x_i + \mathbf{A}_j y_j}. \tag{29}$$

By Lemma 1, this implies that $\mathbf{A}_i$ and $\mathbf{A}_j$ commute. Therefore, matrices $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_m \in \mathbb{R}^{n\times n}$ $(m > 1)$ pairwise commute.

■

Proof of Theorem 1. Recall that $e^{\mathbf{A}}$ is an orthogonal matrix if $\mathbf{A}$ is skew-symmetric, which implies $\mathbf{R}(\boldsymbol{x}; \mathcal{A})^\top = \mathbf{R}(\boldsymbol{x}; \mathcal{A})^{-1} = \mathbf{R}(-\boldsymbol{x}; \mathcal{A})$. Thus, we have:

$$\begin{aligned} &\mathbf{R}(\boldsymbol{x}; \mathcal{A})^\top \mathbf{R}(\boldsymbol{y}; \mathcal{A}) \\ &= e^{-\mathbf{A}_1 x_1 - \mathbf{A}_2 x_2 - \cdots - \mathbf{A}_N x_N} e^{\mathbf{A}_1 y_1 + \mathbf{A}_2 y_2 + \cdots + \mathbf{A}_N y_N}. \end{aligned} \tag{30}$$

By Lemma 3 and Equation 30, $\mathcal{A}$ pairwise commute if and only if there exists a function $f : \mathbb{R}^N \to \mathbb{R}^{d\times d}$ such that:

$$\begin{aligned} &f(y_1 - x_1, y_2 - x_2, \dots, y_N - x_N) \\ &= \mathbf{R}(\boldsymbol{x}; \mathcal{A})^\top \mathbf{R}(\boldsymbol{y}; \mathcal{A}). \end{aligned} \tag{31}$$

Therefore, the theorem holds.

■

A.2. Explanation of rotation matrix and its exponential representation

Following the definition in [9], we first demonstrate the definition of rotation group and rotation matrix:

Definition 7 (Rotation Group and Rotation Matrix)**.** *A* ***special orthogonal group*** *in* $\mathbb{R}^n$*, denoted* $SO(n)$*, is the set of all* $n \times n$ *orthogonal matrices with determinant 1,* i.e.,

$$SO(n) = \{\mathbf{R} \in \mathbb{R}^{n\times n} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\}.$$

*We use the terms **rotation group** and **special orthogonal group** interchangeably. Any matrix in the rotation group is called a **rotation matrix**.*

To establish Definition 5, there is a necessary proposition to ensure the correctness of the exponential representation of a rotation matrix:

Proposition 4. *Any rotation matrix* $\mathbf{R}$ *can be represented by* $\exp(\mathbf{A})$ *where* $\mathbf{A}$ *is a skew-symmetric matrix.*

Proposition 4 is a well-known result in Lie theory, as detailed in [8]. Specifically, the matrix $\mathbf{R}$ in Proposition 4 belongs to the Lie group $SO(n)$. The associated Lie algebra of this group is $\mathfrak{so}(n)$, within which the skew-symmetric matrix $\mathbf{A}$ resides.

B. More experiments

B.1. 3D classification

To assess the ability to handle higher dimensions beyond 2D, we conduct a 3D classification task on UCF-101 [23]. The details of the model and configuration can be found in Appendix C.

The results shown in Table 3 demonstrate similar results in 2D experiments, that ComRoPE performs best when resolution increases beyond the training resolution, displaying the resolution robustness of ComRoPE.

B.2. Fine-tune on pre-trained model

Recall that we represent the RoPE function parameterized by angle matrices as defined in Equation 3. If all elements in $\mathcal{A} = \{\mathbf{A}_i\}_{i=1}^N$ are initialized as zero matrices (i.e., $\forall i, \mathbf{A}_i = \mathbf{O}$), the behavior of this RoPE function degenerates into a standard attention mechanism. This is because, in this case, $\mathbf{R}(\boldsymbol{x}; \mathcal{A}) = \exp(\mathbf{O}) = \mathbf{I}$ for any input $\boldsymbol{x}$.

On the other hand, if $\mathcal{A} = \{\mathbf{A}_i\}_{i=1}^N$ is initialized as described in Appendix D, the RoPE function reduces to the vanilla RoPE formulation.

These observations demonstrate that our method can represent both the standard attention mechanism and various common RoPE attention variants. Therefore, during fine-tuning, standard attention or vanilla RoPE can be replaced with our method. Pre-trained weights can be loaded and fine-tuned under this new paradigm seamlessly, even if ComRoPE was not applied during the pre-training phase.

As an example, we fine-tune the Vision Transformer pre-trained in CLIP [19] on ImageNet by simply replacing the standard attention mechanism with each RoPE method. Specifically, we fine-tune the model for 4 epochs using a batch size of 3456 and a learning rate of 3×10^{-4}.

The results, presented in Table 4, show that ComRoPE-LD achieves the best performance. An interesting observation is that vanilla RoPE exhibits the lowest accuracy among all five methods. This is likely because its fixed and manually defined parameters cannot be loaded seamlessly. In other words, it must adapt the pre-trained latent space during fine-tuning to effectively complete the task, which may result in suboptimal performance.

Method	Accuracy
APE	79.91
Vanilla RoPE	79.82
LieRE	<u>80.12</u>
ComRoPE-AP (ours)	80.11
ComRoPE-LD (ours)	**80.17**

Table 4. Accuracy of fine-tuned models with different positional encoding methods on ImageNet.

C. Details of configuration

C.1. Configuration of 2D classification

Configuration of 2D classfication task is shown in Table 5.

Key	**Value**
Layers	12
Image Size	224
Patch Size	16
Hidden Dimension	768
Attention Heads	12
Batch Size	6144
Optimizer	AdamW
Weight Decay	0.01
Learning Rate	10^{-3}
LR Scheduler	cosine
Warmup Ratio	0.02
Epochs	200

Table 5. Model and training configuration of 2D classification experiments.

C.2. Configuration of 3D classification

Because the vanilla RoPE and ComRoPE-AP require that the head dimension be a multiple coordinate dimension, standard ViT-Base is not applicable. We modified the model parameters to make it possible to conduct experiments on all of the five positional encoding methods. Besides, because the data size of UCF-101 is not too large, using a smaller model is more appropriate. All the details are shown in Table 6.

Key	Value
Layers	8
Image Size	224
Frame Count	8
Patch Size	16
Hidden Dimension	384
Attention Heads	8
Batch Size	768
Optimizer	AdamW
Weight Decay	0.01
Learning Rate	1.2×10^{-4}
LR Scheduler	cosine
Warmup Ratio	0.02
Epochs	80

Table 6. Model and training configuration of 3D classification experiments.

D. Reformulation of baseline RoPE methods in detail

D.1. Vanilla RoPE

Firstly, note that we can represent a 2D rotation matrix in the exponential form:

$$\begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} = \exp(\begin{pmatrix} 0 & -\alpha \\ \alpha & 0 \end{pmatrix}) \tag{32}$$

The solution proposed by RoFormer, which we call vanilla RoPE here, can be regarded as a special type of ComRoPE-AP with block size 2 and non-trainable $\mathbf{P}_j$ in Equation 8 where:

$$\begin{aligned} \mathbf{P}_j &= \begin{pmatrix} \cos(m\theta^{\frac{2N}{d}\cdot j}) & -\sin(m\theta^{\frac{2N}{d}\cdot j}) \\ \sin(m\theta^{\frac{2N}{d}\cdot j}) & \cos(m\theta^{\frac{2N}{d}\cdot j}) \end{pmatrix} \\ &= \exp(m\theta^{\frac{2N}{d}\cdot j}\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}) \end{aligned} \tag{33}$$

In practice, RoFormer adopts $\theta = 10000^{-1}$ as the hyper-parameter of the rotation base.

D.2. LieRE

For LieRE, the blocks are independent and trainable. Hence, we directly define B_{ij} in Equation 6 as:

$$\mathbf{B}_{ij} = \mathbf{P}_{ij} - \mathbf{P}_{ij}^{\top}, \tag{34}$$

where $\mathbf{P}_{ij}$ is a trainable matrix.

E. Analysis and comparison of complexity and extra consumption

Table 7 presents an overview of the properties of the positional encoding methods evaluated in this work. Specifically, the table highlights their commutativity (*i.e.*, the commutativity of angle matrices when represented in the RoPE form parameterized by angle matrices), the number of additional parameters, and the extra time complexity introduced by the positional encoding module.

E.1. APE

For a Transformer that takes n embeddings with d features as inputs, the extra parameters of position encoding are the tensors in the position code book, i.e., $n \times d$. The extra computation is to add position embeddings onto the original features. Therefore, the extra time complexity is $O(n \times d)$.

E.2. RoPE parameterized by angle matrices

We unify RoPE with angle matrices whose rotation process is presented in Algorithm 1, where n, h, d, b, N represents sequence length, number of heads, dimension of hidden states, block size, and number of axes respectively. In this part, we focus on extra parameters and time complexity on each layer.

Algorithm 1 Rotation of query and key matrices

```
In 1: query matrix $\mathbf{Q}$ with shape $(n, h, \frac{d}{h})$
In 2: key matrix $\mathbf{K}$ with shape $(n, h, \frac{d}{h})$
In 3: angle base matrix $\mathbf{A}$ with shape $(N, h, \frac{d}{hb}, b, b)$
In 4: patch positions $\mathbf{P}$ with shape $(n, N)$
Out: rotated query and key matrices $\hat{\mathbf{Q}}, \hat{\mathbf{K}}$

for $axis = 1$ to $N$ do
    $\mathbf{M}_{axis} \leftarrow \mathbf{A}_{axis} \odot \mathbf{P}_{axis}$
end for
$\mathbf{M} \leftarrow \sum \mathbf{M}_{axis}$ where $M$ has a shape of $(n, h, \frac{d}{hb}, b, b)$
$\mathbf{R} \leftarrow \mathrm{diag}(e^{\mathbf{M}}, \mathrm{dim} = 2)$ with shape $(n, h, \frac{d}{h}, \frac{d}{h})$
$\hat{\mathbf{Q}} \leftarrow \mathbf{RQ}, \hat{\mathbf{K}} \leftarrow \mathbf{RK}$

return $\hat{\mathbf{Q}}, \hat{\mathbf{K}}$
```

Angle base matrix $\mathbf{A}$ is defined by the RoPE method, and the extra parameters are brought by the definition of $\mathbf{A}$. Time complexity of 1) calculating the element-wise product over each axis is $O(n \times h \times \frac{d}{hb} \times b^2) = O(ndb)$; 2) calculating sum of M is $O(N \times n \times h \times \frac{d}{hb} \times b^2) = O(ndbN)$; 3) calculating matrix exponential is $O(n \times h \times \frac{d}{hb} \times b^3 = O(ndb^2)$ based on [1]; 4) applying rotation is $O(n \times h \times (\frac{d}{h})^2) = O(\frac{nd^2}{h})$. Thus, the overall time complexity of rotation is $O(ndbN + ndb^2 + \frac{nd^2}{h})$.

E.2.1 Vanilla RoPE

No extra parameters are presented in vanilla RoPE, and the angle base matrix $\mathbf{A}$ can be calculated during pre-processing. Besides, in vanilla RoPE, block size $b = 2$,

Positional Encoding Method	Commutativity	Extra Parameters	Extra Time Complexity
APE	–	nd	$O(nd)$
Vanilla RoPE	Yes	0	$O(Lnd(bN+b^2+\frac{d}{h}))\approx O(\frac{Lnd^2}{h})$
LieRE	Commonly Not	$LNdb$	$O(Lnd(bN+b^2+\frac{d}{h}))$
ComRoPE-AP (ours)	Yes	Ldb	$O(Lnd(bN+b^2+\frac{d}{h}))$
ComRoPE-LD (ours)	Yes	$Ld(\frac{b}{N}+\frac{N}{b})$	$O(Lnd(bN+b^2+\frac{d}{h}))$

Table 7. Comparison of different types of positional encoding methods. n represents for count of patches (tokens), d represents for dimension of hidden states, L represents for count of layers, b represents for block size, N represents for count of axes, and h represents the count of attention heads.

so $\frac{d}{h} \gg bN + b^2 = 2N + 4$ in most cases. Thus, count of extra parameters are 0 and extra time complexity is $O(ndbN + ndb^2 + \frac{nd^2}{h}) \approx O(\frac{nd^2}{h})$ where $b = 2$.

E.2.2 LieRE

For LieRE, the angle base matrix can be formulated as $\mathbf{A} = \mathbf{P} - \mathbf{P}^\top$ where the parameters in $\mathbf{P}$ are all independent. The only extra step to get $\mathbf{A}$ from $\mathbf{P}$ is the subtraction whose time complexity is $O(Ndb)$. Thus, count of extra parameters are $N \times h \times \frac{d}{hb} \times b^2 = Ndb$ and extra time complexity is $O(ndbN+ndb^2+\frac{nd^2}{h}+ndb) = O(ndbN+ndb^2+\frac{nd^2}{h})$.

E.2.3 ComRoPE-AP

For ComRoPE-AP, we compose the angle base matrix $\mathbf{A}$ whose shape is $(N, h, \frac{d}{hb}, b, b)$ with matrices with shape $(N, h, \frac{d}{hbN}, b, b)$ by filling the blocks that are irrelevant to the corresponding coordinate axes with zeros. Thus, similarly, count of extra parameters are $N \times h \times \frac{d}{hbN} \times b^2 = db$ and extra time complexity is $O(ndbN + ndb^2 + \frac{nd^2}{h})$.

E.2.4 ComRoPE-LD

ForComRoPE-LD, the angle base matrices in $\mathbf{A}$ are pairwise linearly dependent on the first dimension (i.e., axis dimension). Therefore, it can be presented by a matrix with shape $(h, \frac{d}{hb}, b, b)$ and a multiplication factor with shape $(N, h, \frac{d}{hb})$ by a multiplication step with time complexity $O(N \times h \times \frac{d}{hb} \times b^2) = O(Ndb)$. Thus, count of extra parameters are $h \times \frac{d}{hbN} \times b^2 + N \times h \times \frac{d}{hb} = d(\frac{b}{N} + \frac{N}{b})$, and extra time complexity is $O(ndbN + ndb^2 + \frac{nd^2}{h})$.

F. Distribution of elements in angle matrices

In this section, we analyze the element distribution in angle matrices obtained from the 2D classification experiments. Specifically, we extract all elements from the upper triangular parts of the matrices. The standard deviations of these elements are summarized in Table 8, and their density plot is presented in Figure 8.

Method	Block Size	Standard Deviations
LieRE	2	0.326
ComRoPE-AP		0.271
ComRoPE-LD		0.384
LieRE	4	0.246
ComRoPE-AP		0.208
ComRoPE-LD		0.278
LieRE	8	0.195
ComRoPE-AP		0.171
ComRoPE-LD		0.238

Table 8. The standard deviations of elements in angle matrices obtained from the 2D classification experiments.

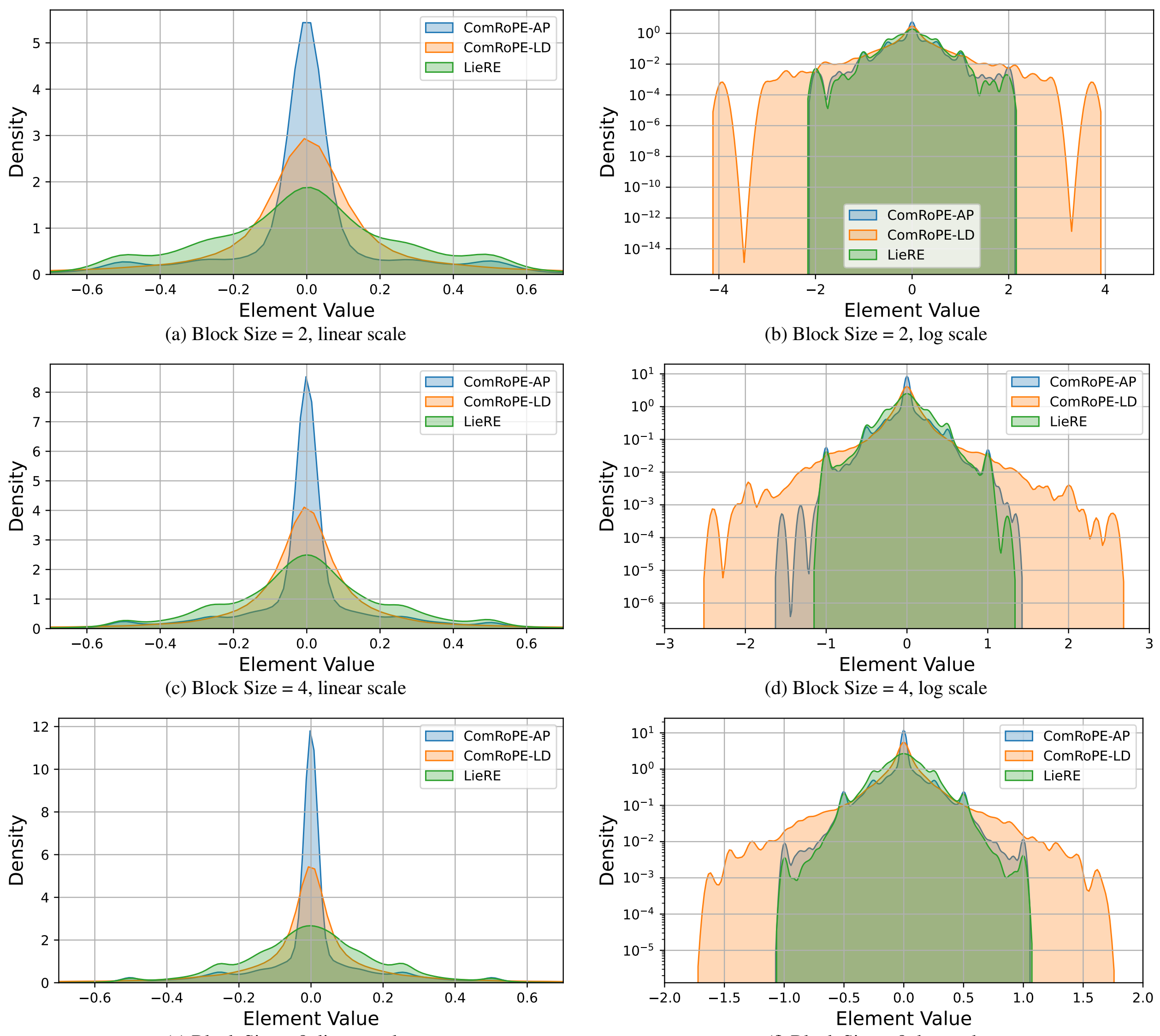
To provide a clearer view of the long-tail distribution, we present the density plot using both linear and logarithmic scales in Figure 8. From the linear scale plot, it can be observed that elements near zero exhibit the highest variance in the angle matrices of LieRE, while ComRoPE-AP demonstrates the most moderate variance. On the other hand, the logarithmic scale reveals notable differences in range. For instance, ComRoPE-LD retains a broader distribution at values farther from zero. Consequently, as indicated in Table 8, ComRoPE-LD exhibits the largest overall variance among the angle matrix elements. This phenomenon is likely due to the linear dependencies between angle matrices across different coordinate axes, which necessitate significant frequency differences to distinguish them effectively.

References

[1] Philipp Bader, Sergio Blanes, and Fernando Casas. Computing the matrix exponential with an optimized taylor polynomial approximation. *Mathematics*, 7:1174, 2019. 4

[2] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1

[3] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao



(a) Block Size = 2, linear scale (b) Block Size = 2, log scale (c) Block Size = 4, linear scale (d) Block Size = 4, log scale (e) Block Size = 8, linear scale (f) Block Size = 8, log scale

Figure 8. Density distribution of elements in the upper triangular sections of angle matrices from 2D classification experiments. Subfigures (a-b), (c-d), and (e-f) show the distributions for different block sizes: 2, 4, and 8, respectively.

Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, march 2023. *URL https://lmsys. org/blog/2023-03-30-vicuna*, 3(5), 2023. 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. 1, 2

[6] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[7] 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, 2024. 2

[8] Jean H. Gallier. Basics of classical lie groups: The exponential map, lie groups, and lie algebras. 2001. 3

[9] Larry C. Grove. Classical groups and geometric algebra. 2001. 2

[10] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. *arXiv preprint arXiv:2403.13298*, 2024. 2

[11] Max Horn, Kumar Shridhar, Elrich Groenewald, and

Philipp FM Baumann. Translational equivariance in kernelizable attention. *arXiv preprint arXiv:2102.07680*, 2021. 2

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. *arXiv preprint arXiv: 1405.0312*, 2014. 7

[13] 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. 2

[14] Antoine Liutkus, Ondřej Cıfka, Shih-Lun Wu, Umut Simsekli, Yi-Hsuan Yang, and Gael Richard. Relative positional encoding for transformers with linear complexity. In *International Conference on Machine Learning*, pages 7067–7079. PMLR, 2021. 2

[15] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26439–26455, 2024. 2

[16] Sophie Ostmeier, Brian Axelrod, Michael E Moseley, Akshay Chaudhari, and Curtis Langlotz. Liere: Generalizing rotary position encodings. *arXiv preprint arXiv:2406.10322*, 2024. 2, 4

[17] A Radford. Improving language understanding by generative pre-training. 2018. 1

[18] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, 2021. 3

[20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 2

[21] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 2

[22] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018. 2

[23] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv: 1212.0402*, 2012. 3

[24] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 1, 2, 3

[25] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. *Advances in neural information processing systems*, 28, 2015. 2

[26] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[27] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 1, 2

[28] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10033–10041, 2021. 2

[29] Chunlong Xia, Xinliang Wang, Feng Lv, Xin Hao, and Yifeng Shi. Vit-comer: Vision transformer with convolutional multi-scale feature interaction for dense predictions. *Computer Vision and Pattern Recognition*, 2024. 7