Latent RoPE for DeepSeek MLA

March 2025

1 Introduction

Generative language models based on transformers use the KV cache to accelerate inference. DeepSeek's Multi-Latent Attention (MLA) addresses the excessive memory demand for long sequence caching by applying low - rank compression and then recovery to Q, K, V in transformers, also speeding up training and inference. However, position encoding can't be directly applied to Q and K in MLA as it prevents the upper projection matrices from being mutually absorbed. So, an extra branch is introduced to represent sequence positions, increasing computation. We present a RoPE method in latent space. By restricting the upper projection matrix, RoPE is moved to latent space, removing the need for an extra branch. Also, the upper projection matrices of Q and K can still be mutually absorbed.

2 Notation

Our notation for original MLA is given in table 2.

3 Methodology

Below is a complete explanation from scratch, including the motivation, the method (with all the matrix operations), and a full proof showing that our learnable upper-projection matrix W commutes with the positional (RoPE) rotation operator $M(t)$, i.e. that

$$W\,M(t) = M'(t)\,W,$$

where $M'(t)$ is the high-dimensional counterpart of $M(t)$.

3.1 Motivation: Why Require $WM = MW$?

In our modified RoPE method, we aim to incorporate position-dependent rotations into a low-dimensional space and then "lift" the result to a higher-dimensional key space via an upper-projection matrix W. The key idea is that

Notation	Definition
i	Index of attention head
d	Embedding dimension
n_h	Number of attention heads
d_h	Dimension of each head
$\mathbf{h}_t$	Attention input of the t-th token at an attention layer, $\mathbf{h}_t \in \mathbb{R}^d$
W^{DKV}	Learnable weight matrix for keys and values down projection
W^{DQ}	Learnable weight matrix for queries down projection
W^{UQ}	Learnable weight matrix for queries up projection
W^{UK}	Learnable weight matrix for keys up projection
W^{UV}	Learnable weight matrix for values up projection
W^{QR}	Matrix for decouples queries
W^{KR}	Matrix for decouples keys
$\mathbf{c}_t^Q$	Compressed latent vector for queries, $\mathbf{c}_t^Q = W^{DQ}\mathbf{h}_t$
$\mathbf{c}_t^{KV}$	Compressed latent vector for keys and values, $\mathbf{c}_t^{KV} = W^{DKV}\mathbf{h}_t$
$\mathbf{q}_t^C$	Queries matrix, $\mathbf{q}_t^C = W^{UQ}\mathbf{c}_t^Q$
$\mathbf{k}_t^C$	Keys matrix, $\mathbf{k}_t^C = W^{UK}\mathbf{c}_t^{KV}$
$\mathbf{v}_t^C$	Values matrix, $\mathbf{v}_t^C = W^{UV}\mathbf{c}_t^{KV}$
$\mathbf{q}_t^R$	Matrix carries RoPE for queries, $\mathbf{q}_t^R = \text{RoPE}(W^{QR}\mathbf{c}_t^Q)$
$\mathbf{k}_t^R$	Matrix carries RoPE for keys, $\mathbf{k}_t^R = \text{RoPE}(W^{KR}\mathbf{h}_t)$
$\mathbf{q}_{t,i}$	Matrix of queries, $\mathbf{q}_{t,i} = [\mathbf{q}_{t,i}^C; \mathbf{q}_{t,i}^R]$
$\mathbf{k}_{t,i}$	Matrix of keys, $\mathbf{k}_{t,i} = [\mathbf{k}_{t,i}^C; \mathbf{k}_{t,i}^R]$
$\mathbf{o}_{t,i}$	Attention Matrix, $\mathbf{o}_{t,i} = \sum_{j=1}^{t} \text{Softmax}_j(\frac{\mathbf{q}_{t,i}^T \mathbf{k}_{j,i}}{\sqrt{d_h + d_h^R}})\mathbf{v}_{t,i}^C$
W^O	Learnable linear projection matrix for the final output
$\mathbf{u}_t$	output $W^O = [\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; \ldots; \mathbf{o}_{t,n_h}]$

Table 1: Notation Table

if the positional rotation $M(t)$ and the projection W commute, then we can "absorb" the rotation into the cached representations or delay its application, which leads to significant efficiency gains during inference. In other words, by ensuring

$$W\, M(t) = M'(t)\, W,$$

we guarantee that whether we apply the rotation before or after the projection, the final result is the same. So that

$$\begin{aligned} M'(t)W^{UK}\mathbf{c}_t^{KV} &= W^{UK}M(t)\mathbf{c}_t^{KV} = W^{UK}\mathbf{c}_{\text{rot}}^{KV}, \\ M'(t)W^{UQ}\mathbf{c}_t^{Q} &= W^{UQ}M(t)\mathbf{c}_t^{Q} = W^{UQ}\mathbf{c}_{\text{rot}}^{Q}, \\ \mathbf{q}_{t,i}^T\mathbf{k}_{t,j}^T &= (\mathbf{c}_{\text{rot}}^Q)^T[(W^{UQ})^T W^{UK}]\mathbf{c}_{\text{rot}}^{KV}. \end{aligned}$$

Thus we can still absorb $[(W^{UQ})^T W^{UK}]$. Notably, in original MLA, the RoPE

Notation	Definition
W	Best upper projection matrix under our restrictions to recover original upper projection matrix
d_{in}	Dimension of compressed Q or K.
d_{out}	Dimension of output, $d_{out} = n_h d_h$.
m	Number of dim-2 subspaces for RoPE.
$M'(t)$	High-dimensional RoPE Matrix (RoPE in original space)
$M(t)$	Low-dimensional RoPE Matrix (RoPE in latent space)
C, B	Orthogonal basis
c_{rot}	Latent representation of QK with RoPE.

Table 2: Notation Table 2

matrices lie between W^{UQ} and W^{UK}. As a result, W^{UK} cannot be absorbed into W^{UQ} because the RoPE matrix should be computed again based on new token's position.

3.2 The Method

3.2.1 Low-Dimensional QK Space

Let the compressed (or "down-sampled") key vector be

$$c \in \mathbb{R}^{d_{\text{in}}}.$$

We design the system such that

$$d_{\text{in}} = 2m,$$

meaning that we partition the low-dimensional space into m two-dimensional subspaces. We define an orthogonal basis matrix for this space:

$$B \in \mathbb{R}^{d_{\text{in}} \times (2m)},$$

which we view as being composed of m blocks:

$$B = \big[B_1,\ B_2,\ \ldots,\ B_m\big],$$

with each block

$$B_i \in \mathbb{R}^{d_{\text{in}} \times 2}.$$

3.2.2 High-Dimensional QK Space

Let the target (or "up-sampled") key space have dimension

$$d_{\text{out}} = n_h\, d_h.$$

We similarly partition this high-dimensional space into m two-dimensional subspaces by defining an orthogonal basis

$$C \in \mathbb{R}^{d_{\text{out}} \times (2m)},$$

which is divided into blocks:

$$C = \big[C_1,\, C_2,\, \ldots,\, C_m\big],$$

where each

$$C_i \in \mathbb{R}^{d_{\text{out}} \times 2}.$$

3.3 Applying RoPE in the Low-Dimensional Space

For each two-dimensional subspace i, we define a rotation matrix that depends on the position t:

$$R_i(t) = \begin{pmatrix} \cos(\alpha_i\, t) & -\sin(\alpha_i\, t) \\ \sin(\alpha_i\, t) & \cos(\alpha_i\, t) \end{pmatrix}.$$

Here, α_i is a (pre-defined or optionally learnable) frequency parameter for the ith block. We then define the overall low-dimensional RoPE operator as:

$$M(t) = \sum_{i=1}^{m} B_i\, R_i(t)\, B_i^{\top}.$$

Thus, given a compressed key vector c, its position-encoded version is

$$c_{\text{rot}} = M(t)\, c.$$

In effect, each sub-block $B_i^{\top} c \in \mathbb{R}^2$ is rotated by $R_i(t)$ and then mapped back by B_i.

3.4 Constructing the Learnable Upper-Projection Matrix W

To "lift" the rotated low-dimensional representation to the high-dimensional key space, we define a learnable projection:

$$W = \sum_{i=1}^{m} C_i\, \Lambda_i\, B_i^{\top},$$

where each $\Lambda_i \in \mathbb{R}^{2\times 2}$ is a learnable matrix. In our design, we parameterize each Λ_i as:

$$\Lambda_i = \lambda_i\, I^{2\times 2},$$

with: $\lambda_i \in \mathbb{R}$ a scaling factor, and $I^{2\times 2}$ an 2×2 identity matrix. The overall projection then maps the low-dimensional rotated vector c_{rot} to the high-dimensional key k:

$$k = W\, c_{\text{rot}} = \sum_{i=1}^{m} C_i\, \Lambda_i\, B_i^{\top}\, M(t)\, c.$$

3.5 Proof of Commutativity: $W\,M(t) = M'(t)\,W$

We now show that our design ensures the commutativity between W and $M(t)$. Define the high-dimensional rotation operator as:

$$M'(t) = \sum_{i=1}^{m} C_i\, R_i(t)\, C_i^\top.$$

Our goal is to prove:

$$W\,M(t) = M'(t)\,W.$$

Proof. By definition,

$$W\,M(t) = \Big(\sum_{i=1}^{m} C_i\,\Lambda_i\,B_i^\top\Big)\Big(\sum_{j=1}^{m} B_j\,R_j(t)\,B_j^\top\Big).$$

Because the columns of B are partitioned into orthogonal subspaces, for $i \neq j$ we have:

$$B_i^\top\,B_j = 0.$$

Thus, only terms with $i = j$ survive, so we have:

$$W\,M(t) = \sum_{i=1}^{m} C_i\,\Lambda_i\,\big(B_i^\top B_i\big)\,R_i(t)\,B_i^\top.$$

Since $B_i^\top B_i = I_2$ (the 2×2 identity), this simplifies to:

$$W\,M(t) = \sum_{i=1}^{m} C_i\,\Lambda_i\,R_i(t)\,B_i^\top. \tag{1}$$

Recall that each Λ_i is parameterized as:

$$\Lambda_i = \lambda_i\,I^{2\times 2},$$

and note that in 2D, rotation matrices commute:

$$\lambda_i I^{2\times 2}\,R_i(t) = R_i(t)\,\lambda_i I^{2\times 2}.$$

Thus, for each i we have:

$$\Lambda_i\,R_i(t) = \lambda_i I^{2\times 2}\,R_i(t) = R_i(t)\,\lambda_i I^{2\times 2} = R_i(t)\,\Lambda_i.$$

Substitute this into the expression from equation 1:

$$W\,M(t) = \sum_{i=1}^{m} C_i\,R_i(t)\,\Lambda_i\,B_i^\top.$$

By definition,

$$M'(t)\,W = \Big(\sum_{i=1}^{m} C_i\,R_i(t)\,C_i^\top\Big)\Big(\sum_{j=1}^{m} C_j\,\Lambda_j\,B_j^\top\Big).$$

Again, by the orthogonality of the blocks C_i (i.e. $C_i^\top C_j = 0$ for $i \neq j$), only terms with $i = j$ survive. Hence,

$$M'(t)\,W = \sum_{i=1}^{m} C_i\,R_i(t)\,C_i^\top\,C_i\,\Lambda_i\,B_i^\top. \tag{2}$$

Since $C_i^\top C_i = I_2$, this simplifies to:

$$M'(t)\,W = \sum_{i=1}^{m} C_i\,R_i(t)\,\Lambda_i\,B_i^\top. \tag{3}$$

Comparing the results from equation 2 and 3, we find:

$$W\,M(t) = \sum_{i=1}^{m} C_i\,R_i(t)\,\Lambda_i\,B_i^\top = M'(t)\,W.$$

□

In our approach, if the upper projection matrix meets our requirements, that is, it can be expressed as

$$W = \sum_{i=1}^{m} C_i\,\Lambda_i\,B_i^\top,$$

then applying RoPE on recovered features and latent features are strictly equivalent. Thus, all properties of RoPE are fully inherited by our method.

3.6 Optimization target

We make assumptions about the upper projection matrix and express it as

$$W = \sum_{i=1}^{m} C_i\,\Lambda_i\,B_i^\top,$$

where

$$\Lambda_i = \lambda_i\,I^{2\times 2}.$$

However, not all matrices can be expressed in such a form. Therefore, to ensure the commutativity between the position encoding and the upper projection, we can only find the optimal recovery matrix for the upper projection under this constraint. Here we denote original upper projection matrix as W^U. Thus we define our optimization goal as:

$$\arg\min_{W} \|W\mathbf{c}_t - W^U\mathbf{c}_t\|_2 \tag{4}$$

If we assume that the distribution of x is uniformly distributed in all directions such that

$$\mathbb{E}[xx^T] = \sigma^2 I,$$

then we can eliminate x from the expression by taking the expectation. In this case, we have

$$\mathbb{E}\|Wx - W'x\|_2^2 = \mathbb{E}\left[x^T(W-W')^T(W-W')x\right] = \operatorname{tr}\Big((W-W')^T(W-W')\,\mathbb{E}[xx^T]\Big).$$

Substituting $\mathbb{E}[xx^T] = \sigma^2 I$ into the equation, we get

$$\mathbb{E}\|Wx - W'x\|_2^2 = \sigma^2 \operatorname{tr}\Big((W-W')^T(W-W')\Big) = \sigma^2 \|W-W'\|_F^2,$$

where $\|W - W'\|_F$ denotes the Frobenius norm of $W - W'$. Here, we do SVD decompose on W^U, and acquire: $W^U = USV^\top$. Here, W and $W^U \in \mathbb{R}^{d_{out}\times d_{in}}$. As mentioned before, $B \in \mathbb{R}^{d_{in}\times 2m}$ where $d_{in} = 2m$. $C \in \mathbb{R}^{d_{out}\times 2m}$. Thus, we set:

$$B = V, \tag{5}$$

$$C = U_{:,1:2m}. \tag{6}$$

Thus, the optimization target is equals to:

$$\begin{aligned} \arg\min_{\lambda_i, 1\le i\le m} &\left\| \sum_{i=1}^{m} C_i \Lambda_i B_i - \sum_{i=1}^{m} C_i \operatorname{diag}(\sigma_{2i}, \sigma_{2i+1}) B_i \right\|_F \\ &= \left\| \sum_{i=1}^{m} C_i (\Lambda_i - \operatorname{diag}(\sigma_{2i}, \sigma_{2i+1})) B_i \right\|_F \end{aligned} \tag{7}$$

Here, $S \in \mathrm{R}^{d_{out}\times d_{in}}$ can be written as:

$$S = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{2m} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} \Sigma \\ \mathbf{0} \end{bmatrix}$$

We arrange λ_i in order to form a 2×2 diagonal matrix and extent the matrix with zeros to match the dimensions of S:

$$\Lambda = \begin{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \end{bmatrix} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix} \lambda_2 & 0 \\ 0 & \lambda_2 \end{bmatrix} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \begin{bmatrix} \lambda_m & 0 \\ 0 & \lambda_m \end{bmatrix} \end{bmatrix}$$

and we extend it by zeros:

$$\Lambda' = \begin{bmatrix} \Lambda \\ \mathbf{0} \end{bmatrix}$$

Thus, we have:

$$\begin{aligned}
&\arg \min_{\lambda_i, 1 \le i \le m} \left\| \sum_{i=1}^{m} C_i(\Lambda_i - \operatorname{diag}(\sigma_{2i}, \sigma_{2i+1}))B_i \right\|_F \\
&= \left\| U(\Lambda' - S)V^\top \right\|_F \\
&= \sqrt{\operatorname{Tr}(V^\top(\Lambda' - S)^\top U^\top U(\Lambda' - S))V} \\
&= \sqrt{\operatorname{Tr}(V^\top(\Lambda' - S)^\top(\Lambda' - S))V} \\
&= \sqrt{\operatorname{Tr}((\Lambda' - S)^\top(\Lambda' - S))VV^\top} \\
&= \sqrt{\operatorname{Tr}((\Lambda' - S)^\top(\Lambda' - S))} \\
&= \left\| (\Lambda' - S) \right\|_F \\
&= \sum_{i=1}^{m} (\lambda_i - \sigma_{2i})^2 + (\lambda_i - \sigma_{2i+1})^2
\end{aligned} \tag{8}$$

Thus optimized $\lambda_i = (\sigma_{2i} + \sigma_{2i+1})/2$. Therefore, we can actually obtain the desired parameters directly from the pre-trained parameters through a closed-form solution.

4 Model Compression Pipeline

This section introduces how to transform a MHA to MLA(new).

4.1 Theories

Although DeepSeek's model employs MLA, other models do not utilize MLA. In this regard, we make the following extension to accelerate the inference process. For any model using MHA and RoPE, the attention computation is given by equation 9:

$$q = W^Q h \quad k = W^K h \quad v = W^K h. \tag{9}$$

We first decompose W^Q and W^V into 2 low rank matrices respectively. So that:

$$\begin{aligned}
W^Q &= W^{UQ} W^{DQ} \\
W^K &= W^{UK} W^{DK}
\end{aligned}$$

Then, we have transformed the MHA to MLA, which RoPE matrices are now between W^{UQ}, W^{UK}, formally:

$$\text{Attn}_{i,j} = \mathbf{c}_i^\top W^{UQ\top} R(i)^\top R(j) W^{UK} c_j.$$

Here, $R(i)$ and $R(j)$ denotes the RoPE on original output space, which are $d_{out} \times d_{out}$ block diagonal matrices. In our design, refering to Deepseek's work, we can only assign a subspace of output space RoPE information and let the rest of it carry no RoPE information. Thus, we first replace the RoPE matrices by partial RoPE matrices. Mathematically:

$$R_p = \begin{bmatrix} R_1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & R_2 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_m & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & I \end{bmatrix}$$

where each R_i represents a rotational matrix, an R_p only carry first $m(m = d_{in}/2)$ rotation matrix. We define the left top $2m \times 2m$ matrix which carries all the partial rotation information as R'. After fine-tuning, we use the formula $M'W = WM$, which means

$$CR_pC^\top C\Lambda B^\top \mathbf{c}_t = C\Lambda B^\top BR_pB^\top \mathbf{c}_t$$

Thus we can cache $BR_pB^\top \mathbf{c}_t$ instead of $\mathbf{c}_t$, where $C\Lambda B^\top$ is the RoPE part of matrix W^U. The no RoPE part can be cached directly without any modification, because it does not contains RoPE information but just latent model representations. However, the fine-tuned version apply RoPE as(only the RoPE part is considered now):

$$R_pC\Lambda B\mathbf{c}_t \neq CR_pC^\top C\Lambda B^\top \mathbf{c}_t$$

Then, the attention is:

$$\mathbf{c}_t^{Q\top} W_{rot}^{UQ\top} C^Q R_p^{Q\top} C^{Q\top} C^K R_p^K C^{K\top} W_{rot}^{UK} \mathbf{c}_t^K.$$

To maintain the positional invariance:

$$\mathbf{q}^\top R(s)^\top R(t)\mathbf{k} = \mathbf{q}^\top R(t-s)\mathbf{k},$$

we multiply the RoPE part of W^U on the left by matrix C to eliminate the orthogonal matrix between the two RoPE matrices. So that the attention is now

$$\mathbf{c}_t^{Q\top} W_{rot}^{UQ\top} C^Q R_p^{Q\top} R_p^K C^{K\top} W_{rot}^{UK} \mathbf{c}_t^K.$$

However, it is still different from original RoPE, because the expression now is equivalent to multiply the RoPE part of W^U by $C^\top$ before RoPE. Original:

$$\mathbf{c}_t^{Q\top} W_{rot}^{UQ\top} R_p^{Q\top} R_p^K W_{rot}^{UK} \mathbf{c}_t^K.$$

Thus, we need to fine-tune the model to adapt this modification.

4.2 Post Training

As mentioned, to transform a MHA model to MLA, we first apply SVD to compress the Q,K,V matrices to a latent low dimensional space then recover it. Formally, $W = W^U W^D$. Then, we replace RoPE matrix by R_p. After that, we can start to fine-tune the model. At each time step, before we do RoPE, we do SVD for temporary upper projection matrices to get U, S, V. Then, we set first $2m$ columns of U as C and multiply the RoPE part of W^U by $C^\top$ before RoPE.

Notably, this is actually equivalent as DeepSeek MLA, which apply RoPE on a low dimension projected space, and eventually concatenate no RoPE and RoPE part together. In our version, we also just use part of the ouput dimension to carry RoPE information. This part has the same dimensionality as the compressed representation of K and Q. Therefore, in practice, we are also performing RoPE in a compressed dimension. After the matrix is restored, it essentially becomes equivalent to directly applying RoPE only on a subset of the output dimensions.