

## A Enhanced Expressive Power of MLA with Decoupled RoPE

### A.1 Introduction

This section provides a theoretical analysis to demonstrate that Multi-Head Latent Attention (MLA) with decoupled Rotary Position Embedding (RoPE), as described in Section 3.3 of the main paper, possesses greater expressive power than Grouped-Query Attention (GQA) (Section 3.2). This analysis assumes **comparable KV cache sizes and number of query heads**.

Our primary argument focuses on the core projection mechanisms that generate queries, keys, and values, abstracting away from the specifics of RoPE application initially. We first present the following proposition concerning the relative expressiveness of these core mechanisms:

**Proposition 1.** *Given the same KV cache size and number of query heads, the expressiveness of the core attention projection mechanisms follows the order:  $\text{GQA} < \text{MLA}_{\text{Factorized}} < \text{MQA}$ .*

Here,  $\text{MLA}_{\text{Factorized}}$  refers to an attention mechanism employing low-rank factorization for its key and value projections, representing the content-processing aspect of the full MLA. It is important to note that in the proposition, the query projection in  $\text{MLA}_{\text{Factorized}}$  does not undergo low-rank factorization; this differs from the full MLA, where the query is also factorized. After proving this proposition, we will discuss how the full MLA architecture, which incorporates such an  $\text{MLA}_{\text{Factorized}}$  core for its content components and an MQA core for its decoupled RoPE components, is thereby more expressive than GQA. For this analysis, we primarily consider the impact of the architectural structure on representational capacity, **setting aside the direct effects of RoPE itself** on the expressiveness comparison between the fundamental GQA, MLA-Factorized, and MQA structures.

### A.2 Proof of Proposition 1

Let  $D$  be the hidden dimension of the input token  $\mathbf{x}_t \in \mathbb{R}^D$ ,  $h$  be the number of query heads, and  $d = D/h$  be the dimension per head. In GQA, query heads are divided into  $g$  groups. For fair KV cache comparison, the latent dimension for keys and values in  $\text{MLA}_{\text{Factorized}}$  ( $r_{kv}$ ) and the head dimension of MQA will be related to  $gd$ . Specifically, if the KV cache per token in GQA is  $2gd$  for both keys and values, then in  $\text{MLA}_{\text{Factorized}}$ ,  $r_{kv} = 2gd$ , and in MQA, the head dimension is also  $2gd$ ; this ensures the KV cache sizes are aligned.

#### A.2.1 $\text{GQA} \leq \text{MLA}_{\text{Factorized}}$

In GQA, query head  $\mathbf{q}_{t,i}$  attends to key  $\mathbf{k}_{j, \lceil i/(h/g) \rceil}$  and value  $\mathbf{v}_{j, \lceil i/(h/g) \rceil}$ . The GQA key projection  $W^K \in \mathbb{R}^{gd \times D}$  produces  $g$  distinct key vectors  $[\mathbf{k}_{t,1}; \dots; \mathbf{k}_{t,g}]$ . Similarly,  $W^V \in \mathbb{R}^{gd \times D}$  produces value vectors. We define effective per-query-head projection matrices  $W'^K \in \mathbb{R}^{hd \times D}$  and  $W'^V \in \mathbb{R}^{hd \times D}$  for GQA:

$$W'^K = \begin{pmatrix} W_1'^K \\ \vdots \\ W_h'^K \end{pmatrix}, \text{ where } W_i'^K = W_{\lceil i/(h/g) \rceil}^K, \quad (21)$$

$$W'^V = \begin{pmatrix} W_1'^V \\ \vdots \\ W_h'^V \end{pmatrix}, \text{ where } W_i'^V = W_{\lceil i/(h/g) \rceil}^V. \quad (22)$$

Here,  $W_k^K$  is the  $k$ -th  $d \times D$  block of  $W^K$ . Thus,  $\mathbf{k}'_{j,i} = W_i'^K \mathbf{x}_j = \mathbf{k}_{j, \lceil i/(h/g) \rceil}$ , and similarly for values. The matrices  $W'^K$  and  $W'^V$  have ranks at most  $gd$ .

An  $\text{MLA}_{\text{Factorized}}$  mechanism generates keys via  $\mathbf{k}_{j,i} = (W^{UK}(W^{DKV} \mathbf{x}_j))_i$ , where  $W^{DKV} \in \mathbb{R}^{r_{kv} \times D}$  and  $W^{UK} \in \mathbb{R}^{hd \times r_{kv}}$ . A similar formulation applies for values with  $W^{UV} \in \mathbb{R}^{hd \times r_{kv}}$ .

To demonstrate expressive capability,  $\text{GQA} \leq \text{MLA}_{\text{Factorized}}$ , we set  $r_{kv} = 2gd$ . Let  $W^{DKV} = \begin{pmatrix} W^K \\ W^V \end{pmatrix} \in \mathbb{R}^{2gd \times D}$ . We seek  $W^{UK}, W^{UV} \in \mathbb{R}^{hd \times 2gd}$  such that  $W'^K = W^{UK} W^{DKV}$ ,  $W'^V =$

867  $W^{UV}W^{DKV}$ . This is achieved by setting  $W_i^{UK}, W_i^{UV} \in \mathbb{R}^{d \times 2gd}$  (the block for head  $i$ ) as selector  
868 matrices:

$$W_i^{UK} = [\underbrace{\mathbf{0}_{d \times d}, \dots, \mathbf{0}_{d \times d}}_{k-1 \text{ blocks}}, \mathbf{I}_{d \times d}, \underbrace{\mathbf{0}_{d \times d}, \dots, \mathbf{0}_{d \times d}}_{2g-k \text{ blocks}}], \quad (23)$$

$$W_i^{UV} = [\underbrace{\mathbf{0}_{d \times d}, \dots, \mathbf{0}_{d \times d}}_{g+k-1 \text{ blocks}}, \mathbf{I}_{d \times d}, \underbrace{\mathbf{0}_{d \times d}, \dots, \mathbf{0}_{d \times d}}_{g-k \text{ blocks}}], \quad (24)$$

869 where  $k = \lceil i/(h/g) \rceil$ . Thus, GQA's key/value generation can be replicated by an  $\text{MLA}_{\text{Factorized}}$   
870 model with  $r_{kv} = 2gd$  and specific sparse structures for  $W^{UK}$  and  $W^{UV}$ . The KV cache size  
871  $2gd \times (\text{sequence length})$  is preserved since we will be caching  $\mathbf{c}_j^{KV} = W^{DKV} \mathbf{x}_j \in \mathbb{R}^{2gd}$ . On that  
872 account, the theoretical expressive power of GQA is less than or equal to that of  $\text{MLA}_{\text{Factorized}}$  given  
873 the same KV cache size.

### 874 A.2.2 $\text{MLA}_{\text{Factorized}} \leq \text{MQA}$

875 Consider an  $\text{MLA}$ -Factorized model where queries are  $\mathbf{q}_{t,i} = W_i^Q \mathbf{x}_t$  (assuming  $W_i^Q \in \mathbb{R}^{d \times D}$  is the  
876  $i$ -th block of  $W^Q$ ) and keys are  $\mathbf{k}_{j,i} = (W_i^{UK}(W^{DKV} \mathbf{x}_j))$ . The attention score for head  $i$  involves  
877  $\mathbf{q}_{t,i}^\top \mathbf{k}_{j,i}$ :

$$\mathbf{q}_{t,i}^\top \mathbf{k}_{j,i} = (W_i^Q \mathbf{x}_t)^\top (W_i^{UK}(W^{DKV} \mathbf{x}_j)). \quad (25)$$

878 This can be rewritten as:

$$\mathbf{q}_{t,i}^\top \mathbf{k}_{j,i} = (\underbrace{(W_i^{UK})^\top W_i^Q}_{W_i'^Q} \mathbf{x}_t)^\top (W^{DKV} \mathbf{x}_j). \quad (26)$$

879 Let  $\hat{\mathbf{q}}_{t,i} = W_i'^Q \mathbf{x}_t \in \mathbb{R}^{2gd}$  and  $\mathbf{c}_j^{KV} = W^{DKV} \mathbf{x}_j \in \mathbb{R}^{2gd}$ . The computation of attention output  
880 becomes:

$$\mathbf{o}_{t,i} = \sum_j \text{softmax}_j \left( \frac{\hat{\mathbf{q}}_{t,i}^\top \mathbf{c}_j^{KV}}{\sqrt{d}} \right) W_i^{UV} \mathbf{c}_j^{KV}, \quad (27)$$

$$\begin{aligned} \mathbf{y}_t &= W^O [\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; \dots; \mathbf{o}_{t,h}] \\ &= \underbrace{W^O \begin{pmatrix} W_1^{UV} & & & \\ & W_2^{UV} & & \\ & & \ddots & \\ & & & W_h^{UV} \end{pmatrix}}_{W'^O} \begin{pmatrix} \text{softmax}_j \left( \frac{\hat{\mathbf{q}}_{t,1}^\top \mathbf{c}_j^{KV}}{\sqrt{d}} \right) \mathbf{c}_j^{KV} \\ \vdots \\ \text{softmax}_j \left( \frac{\hat{\mathbf{q}}_{t,h}^\top \mathbf{c}_j^{KV}}{\sqrt{d}} \right) \mathbf{c}_j^{KV} \end{pmatrix}. \end{aligned} \quad (28)$$

881 This is an MQA formulation where each modified query  $\hat{\mathbf{q}}_{t,i}$  (now of dimension  $2gd$ ) attends to a  
882 shared key and value  $\mathbf{c}_j^{KV}$ . This indicates that the computations within  $\text{MLA}_{\text{Factorized}}$  can be  
883 structured to use shared intermediate key and value representations akin to MQA's core. Thus, any  
884  $\text{MLA}$ -Factorized model can be represented as an MQA model with a shared key/value of dimension  
885  $2gd$ .

### 886 A.2.3 Strict Inequalities: $\text{GQA} < \text{MLA}_{\text{Factorized}} < \text{MQA}$

887 The relationships are strict:

888  **$\text{GQA} < \text{MLA}_{\text{Factorized}}$**  When GQA is represented as an  $\text{MLA}_{\text{Factorized}}$  model, the up-projection ma-  
889 trices  $W^{UK}$  and  $W^{UV}$  must adopt specific sparse, block-selector structures. A general  $\text{MLA}_{\text{Factorized}}$   
890 model imposes no such constraints;  $W^{UK}$  and  $W^{UV}$  are typically dense and fully learnable. This  
891 allows a general  $\text{MLA}_{\text{Factorized}}$  to create  $h$  distinct key (and value) vectors by combining features  
892 from the  $r_{kv}$ -dimensional latent space in complex ways. GQA is restricted to  $g$  unique key (and  
893 value) vectors that are merely replicated  $h/g$  times. If  $h > g$ ,  $\text{MLA}_{\text{Factorized}}$  can generate a richer  
894 set of interaction patterns. Thus,  $\text{MLA}_{\text{Factorized}}$  has strictly greater expressive power.

895 **MLA<sub>Factorized</sub> < MQA** Consider the bilinear form  $\mathbf{x}_t^\top \mathbf{M} \mathbf{x}_j$  in the attention score. In  $\text{MLA}_{\text{Factorized}}$ ,  
 896 for head  $i$ ,  $\mathbf{M}_{\text{MLA},i} = (W_i^Q)^\top W_i^{UK} W^{DKV}$ . The maximum rank of the transformation is  
 897 determined by the smallest one among the ranks of  $W_i^Q \in \mathbb{R}^{d \times D}$ ,  $W_i^{UK} \in \mathbb{R}^{d \times 2gd}$ , and  
 898  $W^{DKV} \in \mathbb{R}^{2gd \times D}$ , which is at most  $d$ .

899 However, in the MQA form derived from  $\text{MLA}_{\text{Factorized}}$ , the rank of the interaction matrix here,  
 900  $(W_i^{Q'})^\top W^{DKV}$ , is determined by the smallest one among the ranks of  $W_i^{Q'} \in \mathbb{R}^{2gd \times D}$  and  
 901  $W^{DKV} \in \mathbb{R}^{2gd \times D}$ , which is at most  $2gd$ .

902 Since  $2gd \geq d$ , MQA allows for a potentially higher-rank interaction between the (modified) query  
 903 and the shared key representations compared to the per-head effective rank in  $\text{MLA}_{\text{Factorized}}$ 's  
 904 original formulation. This indicates that MQA has a greater representational capacity for the scoring  
 905 mechanism.

### 906 A.3 Expressiveness of MLA with Decoupled RoPE

907 The full MLA architecture, as defined in Section 3.3 (main paper), employs a decoupled RoPE  
 908 strategy. The query  $\mathbf{q}_{t,i}$  and key  $\mathbf{k}_{t,i}$  for head  $i$  (in the MHA-like training paradigm, Equation 9) are:

$$\mathbf{q}_{t,i} = [\mathbf{q}_{t,i}^C; \mathbf{q}_{t,i}^R] \quad (29)$$

$$\mathbf{k}_{t,i} = [\mathbf{k}_{t,i}^C; \mathbf{k}_t^R] \quad (30)$$

909 where  $\mathbf{k}_t^R$  is a shared RoPE key component across all heads for token  $t$ . The bilinear attention score  
 910 (numerator of the softmax argument) for head  $i$  between query at  $t$  and key at  $j$  is:

$$(\mathbf{q}_{t,i}^C)^\top \mathbf{k}_{j,i}^C + (\mathbf{q}_{t,i}^R)^\top \mathbf{k}_j^R \quad (31)$$

911 Let's analyze the two components of this score:

- 912 1. **Content Component Interaction:**  $(\mathbf{q}_{t,i}^C)^\top \mathbf{k}_{j,i}^C$ . The content keys  $\mathbf{k}_{j,i}^C$  are derived from  
 913  $W^{UK}(W^{DKV} \mathbf{x}_j)$ . This key generation mechanism for  $\mathbf{k}_{j,i}^C$  is precisely that of the  
 914  $\text{MLA}_{\text{Factorized}}$  model discussed in Section A.1. As established,  $\text{MLA}_{\text{Factorized}}$  is strictly  
 915 more expressive than GQA for the non-positional part of the representation.
- 916 2. **Positional Component Interaction:**  $(\mathbf{q}_{t,i}^R)^\top \mathbf{k}_j^R$ . This interaction, where  $h$  distinct query-  
 917 side RoPE components  $\mathbf{q}_{t,i}^R$  attend to a single, shared key-side RoPE component  $\mathbf{k}_j^R$ , is an  
 918 MQA structure specifically for the positional information. As shown in Section A.2.3, MQA  
 919 is strictly more expressive than  $\text{MLA}_{\text{Factorized}}$ , and by extension, GQA.

920 In summary, we have demonstrated that the expressive power of MLA with decoupled RoPE is  
 921 stronger than that of the traditional GQA. However, it is worth noting that in the previously proven  
 922 proposition, the  $\text{MLA}_{\text{Factorized}}$  does not have a low-rank decomposition on the query; this differs  
 923 from DeepSeek MLA. In the full MLA architecture, the query is also decomposed.

## 924 B Proof of RoPE Inner Product Invariance under Orthogonal 925 Transformation

926 In this subsection, we provide a rigorous proof of Equation 19, namely:

$$\sum_{l=1}^{d/2} \left( \left[ \mathbf{U}_l \hat{\mathbf{q}}_{t,i}^{[2l-1::d]}; \mathbf{U}_l \hat{\mathbf{q}}_{t,i}^{[2l::d]} \right] \right)^{R^\top} \left( \left[ \mathbf{U}_l \hat{\mathbf{k}}_j^{[2l-1::d]}; \mathbf{U}_l \hat{\mathbf{k}}_j^{[2l::d]} \right] \right)^R = \hat{\mathbf{q}}_{t,i}^{R^\top} \hat{\mathbf{k}}_j^R.$$

927 Here,  $d$  is the dimension of each original attention head. The notation  $\mathbf{q}_{t,i}^{[2l-1::d]}$  (and similarly for  
 928 other terms) refers to an  $h$ -dimensional vector collecting the  $(2l-1)$ -th components from each of the  
 929  $h$  original attention heads. The matrix  $\mathbf{U}_l$  is an  $h \times h$  orthogonal matrix. The superscript  $R$  denotes  
 930 the application of RoPE.

931 *Proof.* For the sake of convenience, we omit all  $i, j, k$  and let  $\mathbf{q}_{x,l} = \mathbf{q}_{t,i}^{[2l-1::]}$  and  $\mathbf{q}_{y,l} = \mathbf{q}_{t,i}^{[2l::]}$ .  
 932 These are  $h$ -dimensional vectors. Similarly, let  $\mathbf{k}_{x,l} = \mathbf{k}_j^{[2l-1::]}$  and  $\mathbf{k}_{y,l} = \mathbf{k}_j^{[2l::]}$ .

933 The RoPE transformation, as defined by Equations (17) and (18) in the main text, applies as follows  
 934 for a query vector at position  $t$  and key vector at position  $j$  within the  $l$ -th subspace:

935 For the query vector components:

$$\begin{aligned}(\mathbf{q}_{x,l})^R &= \mathbf{q}_{x,l} \cos(t\theta_l) - \mathbf{q}_{y,l} \sin(t\theta_l) \\ (\mathbf{q}_{y,l})^R &= \mathbf{q}_{x,l} \sin(t\theta_l) + \mathbf{q}_{y,l} \cos(t\theta_l)\end{aligned}$$

936 For the key vector components:

$$\begin{aligned}(\mathbf{k}_{x,l})^R &= \mathbf{k}_{x,l} \cos(j\theta_l) - \mathbf{k}_{y,l} \sin(j\theta_l) \\ (\mathbf{k}_{y,l})^R &= \mathbf{k}_{x,l} \sin(j\theta_l) + \mathbf{k}_{y,l} \cos(j\theta_l)\end{aligned}$$

937 We use the shorthand  $c_t = \cos(t\theta_l)$ ,  $s_t = \sin(t\theta_l)$ ,  $c_j = \cos(j\theta_l)$ , and  $s_j = \sin(j\theta_l)$ .

938 The right-hand side (RHS) of Equation (19) is given by the definition of the RoPE inner product:

$$\begin{aligned}\mathbf{q}_{t,i}^{R\top} \mathbf{k}_j^R &= \sum_{l=1}^{d/2} [(\mathbf{q}_{x,l})^R; (\mathbf{q}_{y,l})^R]^\top [(\mathbf{k}_{x,l})^R; (\mathbf{k}_{y,l})^R] \\ &= \sum_{l=1}^{d/2} (((\mathbf{q}_{x,l})^R)^\top (\mathbf{k}_{x,l})^R + ((\mathbf{q}_{y,l})^R)^\top (\mathbf{k}_{y,l})^R)\end{aligned}$$

939 Let  $S_l$  be the  $l$ -th term in this sum:

$$\begin{aligned}S_l &= (c_t \mathbf{q}_{x,l} - s_t \mathbf{q}_{y,l})^\top (c_j \mathbf{k}_{x,l} - s_j \mathbf{k}_{y,l}) + (s_t \mathbf{q}_{x,l} + c_t \mathbf{q}_{y,l})^\top (s_j \mathbf{k}_{x,l} + c_j \mathbf{k}_{y,l}) \\ &= c_t c_j \mathbf{q}_{x,l}^\top \mathbf{k}_{x,l} - c_t s_j \mathbf{q}_{x,l}^\top \mathbf{k}_{y,l} - s_t c_j \mathbf{q}_{y,l}^\top \mathbf{k}_{x,l} + s_t s_j \mathbf{q}_{y,l}^\top \mathbf{k}_{y,l} \\ &\quad + s_t s_j \mathbf{q}_{x,l}^\top \mathbf{k}_{x,l} + s_t c_j \mathbf{q}_{x,l}^\top \mathbf{k}_{y,l} + c_t s_j \mathbf{q}_{y,l}^\top \mathbf{k}_{x,l} + c_t c_j \mathbf{q}_{y,l}^\top \mathbf{k}_{y,l} \\ &= (c_t c_j + s_t s_j)(\mathbf{q}_{x,l}^\top \mathbf{k}_{x,l} + \mathbf{q}_{y,l}^\top \mathbf{k}_{y,l}) + (s_t c_j - c_t s_j)(\mathbf{q}_{x,l}^\top \mathbf{k}_{y,l} - \mathbf{q}_{y,l}^\top \mathbf{k}_{x,l}) \\ &= \cos((t-j)\theta_l)(\mathbf{q}_{x,l}^\top \mathbf{k}_{x,l} + \mathbf{q}_{y,l}^\top \mathbf{k}_{y,l}) + \sin((t-j)\theta_l)(\mathbf{q}_{x,l}^\top \mathbf{k}_{y,l} - \mathbf{q}_{y,l}^\top \mathbf{k}_{x,l}).\end{aligned}$$

940 Now, let's analyze the left-hand side (LHS) of Equation (19). Let  $\mathbf{q}'_{x,l} = \mathbf{U}_l \mathbf{q}_{x,l}$  and  $\mathbf{q}'_{y,l} = \mathbf{U}_l \mathbf{q}_{y,l}$ .

941 Similarly, let  $\mathbf{k}'_{x,l} = \mathbf{U}_l \mathbf{k}_{x,l}$  and  $\mathbf{k}'_{y,l} = \mathbf{U}_l \mathbf{k}_{y,l}$ . The  $l$ -th term of the LHS sum, denoted  $S'_l$ , is:

$$S'_l = (((\mathbf{q}'_{x,l})^R)^\top (\mathbf{k}'_{x,l})^R + ((\mathbf{q}'_{y,l})^R)^\top (\mathbf{k}'_{y,l})^R).$$

942 This has the same structure as  $S_l$ , just with primed variables:

$$S'_l = \cos((t-j)\theta_l)((\mathbf{q}'_{x,l})^\top \mathbf{k}'_{x,l} + (\mathbf{q}'_{y,l})^\top \mathbf{k}'_{y,l}) + \sin((t-j)\theta_l)((\mathbf{q}'_{x,l})^\top \mathbf{k}'_{y,l} - (\mathbf{q}'_{y,l})^\top \mathbf{k}'_{x,l}).$$

943 We need to show that the dot product terms involving primed variables are equal to their unprimed  
 944 counterparts. Consider the first coefficient term:

$$\begin{aligned}(\mathbf{q}'_{x,l})^\top \mathbf{k}'_{x,l} + (\mathbf{q}'_{y,l})^\top \mathbf{k}'_{y,l} &= (\mathbf{U}_l \mathbf{q}_{x,l})^\top (\mathbf{U}_l \mathbf{k}_{x,l}) + (\mathbf{U}_l \mathbf{q}_{y,l})^\top (\mathbf{U}_l \mathbf{k}_{y,l}) \\ &= \mathbf{q}_{x,l}^\top \mathbf{U}_l^\top \mathbf{U}_l \mathbf{k}_{x,l} + \mathbf{q}_{y,l}^\top \mathbf{U}_l^\top \mathbf{U}_l \mathbf{k}_{y,l} \\ &= \mathbf{q}_{x,l}^\top \mathbf{k}_{x,l} + \mathbf{q}_{y,l}^\top \mathbf{k}_{y,l}.\end{aligned}$$

945 The last equation holds because  $\mathbf{U}_l$  is an orthogonal matrix. This matches the corresponding term in  
 946  $S_l$ .

947 The same applies to the second coefficient term. In this way, we have proven that  $S'_l = S_l$  for each  
 948  $l \in \{1, \dots, d/2\}$ . This implies that the LHS of Equation (19) is equal to its RHS:

$$\sum_{l=1}^{d/2} \left( [\mathbf{U}_l \mathbf{q}_{t,i}^{[2l-1::]}; \mathbf{U}_l \mathbf{q}_{t,i}^{[2l::]}] \right)^{R\top} \left( [\mathbf{U}_l \mathbf{k}_j^{[2l-1::]}; \mathbf{U}_l \mathbf{k}_j^{[2l::]}] \right)^R = \mathbf{q}_{t,i}^{R\top} \mathbf{k}_j^R.$$

949 This completes the proof, demonstrating that the orthogonal transformation  $\mathbf{U}_l$  applied to the  $h$ -  
 950 dimensional vectors representing the  $l$ -th 2D subspace components across heads preserves the  
 951 RoPE-based inner product structure.  $\square$

In practice, we take advantage of this property to find such orthogonal matrices  $U_l$  that concentrate the principal components of keys into the first head. In this way, when we remove the principal components of most heads in queries and keys, we can retain most of the positional information in the first head, significantly reducing the loss caused by removing RoPE.

Specifically, we proceed as follows: First, we run the model on a calibration dataset (Wikitext-2) to obtain the keys at each layer. Then, we perform PCA on the key activations across the dimensions corresponding to different heads. The resulting matrix of eigenvectors (ordered by the magnitude of their corresponding eigenvalues, from largest to smallest) is used as the orthogonal matrix  $U_l$  in this context. As a result, after this rotation, the principal components of the keys in each dimension are concentrated in the first few heads.

## C FreqFold: Detailed Mechanism, Example, and PCA Efficiency

This appendix provides a detailed explanation of the FreqFold technique, illustrates its operation with a concrete example, and formally connects its benefits to a general principle of Principal Component Analysis (PCA) concerning structured data. This justification clarifies FreqFold’s role in minimizing transformation loss towards decoupled RoPE within the RoRoPE framework (Section 4.2).

### C.1 Detailed Explanation of FreqFold and RoRoPE’s PCA

In the RoRoPE framework, Rotary Position Embedding (RoPE) is applied. RoPE encodes positional information by rotating pairs of feature dimensions. For each RoPE frequency index  $l \in \{1, \dots, d/2\}$ , the corresponding pair of dimensions ( $[2l - 1 :: d]$ ,  $[2l :: d]$ ) from query and key vectors are rotated. When multiple original attention heads are used (say,  $g$  heads), and their key/query projection outputs are concatenated, the RoPE operation for a specific frequency index  $l$  applies to a  $2g$ -dimensional vector segment (formed by concatenating the  $l$ -th 2D RoPE subspace from each of the  $g$  heads). RoRoPE then applies PCA via matrices  $\{U_l\}_{l=1}^{d/2}$  to these  $2g$ -dimensional segments, independently for each frequency index  $l$ .

The core idea of FreqFold is to approximate numerically similar RoPE base frequencies as being effectively identical. For instance, if RoPE uses original base frequencies  $\theta_{l_1}, \theta_{l_2}, \dots, \theta_{l_M}$  that are close in value, MD-FreqFold might treat them all as a single, representative frequency  $\theta^*$ .

This approximation has a significant implication for how PCA is applied in RoRoPE:

- **Without FreqFold (Standard RoRoPE PCA):** For each distinct RoPE frequency index  $l$ , a separate PCA transformation  $U_l$  is learned and applied to the corresponding  $2g$ -dimensional key/query segments.
- **With FreqFold:** If  $M$  original RoPE frequency indices (say  $l_1, \dots, l_M$ ) are grouped together by FreqFold due to their frequency similarity, the  $M$  corresponding  $2g$ -dimensional segments are effectively concatenated. Instead of  $M$  separate PCAs on  $2g$ -dimensional vectors, a single PCA is performed on the resulting  $M \cdot 2g$ -dimensional vectors.

#### C.1.1 Illustrative Example of FreqFold

Let’s consider a scenario with  $g = 2$  key heads, and each head has  $d_{head} = 8$  dimensions. Thus, there are  $d/2 = 8/2 = 4$  distinct RoPE frequency indices per head, which we denote as  $\phi_1, \phi_2, \phi_3, \phi_4$ . The total number of dimensions is  $2 \times 8 = 16$ . The RoPE angles for these 16 dimensions could be conceptualized as follows (repeating for each pair, and across heads):

- **Head 1 (dims 1-8):**  $(\phi_1, \phi_1), (\phi_2, \phi_2), (\phi_3, \phi_3), (\phi_4, \phi_4)$
- **Head 2 (dims 9-16):**  $(\phi_1, \phi_1), (\phi_2, \phi_2), (\phi_3, \phi_3), (\phi_4, \phi_4)$

**Case 1: RoRoPE without FreqFold** For each frequency index  $\phi_l$ , RoRoPE groups the corresponding dimensions from all  $g = 2$  heads. Each such group forms  $2g = 2 \times 2 = 4$ -dimensional vectors (across  $N$  samples).

- **Group for  $\phi_1$ :** Dimensions  $\{1, 2\}$  from Head 1 and  $\{9, 10\}$  from Head 2. PCA is applied to these  $N$  samples of 4D vectors.

- 999 • Group for  $\phi_2$ : Dimensions  $\{3, 4\}$  from Head 1 and  $\{11, 12\}$  from Head 2. PCA is applied  
1000 to these  $N$  samples of 4D vectors.
- 1001 • Group for  $\phi_3$ : Dimensions  $\{5, 6\}$  from Head 1 and  $\{13, 14\}$  from Head 2. PCA is applied  
1002 to these  $N$  samples of 4D vectors.
- 1003 • Group for  $\phi_4$ : Dimensions  $\{7, 8\}$  from Head 1 and  $\{15, 16\}$  from Head 2. PCA is applied  
1004 to these  $N$  samples of 4D vectors.

1005 Here, RoRoPE performs 4 separate PCA operations.

1006 **Case 2: RoRoPE with 2D-FreqFold** 2D-FreqFold implies we are pairing up original frequencies.  
1007 Suppose FreqFold approximates  $\phi_1 \approx \phi_2$  (calling this effective frequency  $\Phi_A = \phi_1$ ) and  $\phi_3 \approx \phi_4$   
1008 (calling this  $\Phi_B = \phi_3$ ).

- 1009 • **Effective Group for  $\Phi_A$ :** This group now includes all dimensions originally associated with  
1010  $\phi_1$  OR  $\phi_2$ .
  - 1011 – Original  $\phi_1$ -dimensions:  $\{1, 2\}$  from Head 1;  $\{9, 10\}$  from Head 2. (Forms a 4D  
1012 segment  $S_{\phi_1}$ )
  - 1013 – Original  $\phi_2$ -dimensions:  $\{3, 4\}$  from Head 1;  $\{11, 12\}$  from Head 2. (Forms a 4D  
1014 segment  $S_{\phi_2}$ )

1015 With FreqFold, these segments  $S_{\phi_1}$  and  $S_{\phi_2}$  are concatenated. PCA is now applied to the  $N$   
1016 samples of  $(4 + 4) = 8$ -dimensional vectors formed by  $[S_{\phi_1}, S_{\phi_2}]$ . Effectively, dimensions  
1017  $\{1, 2, 3, 4\}$  from Head 1 are combined with  $\{9, 10, 11, 12\}$  from Head 2.

- 1018 • **Effective Group for  $\Phi_B$ :** Similarly, this group includes dimensions originally for  $\phi_3$  OR  
1019  $\phi_4$ .

- 1020 – Original  $\phi_3$ -dimensions:  $\{5, 6\}$  from Head 1;  $\{13, 14\}$  from Head 2. (Forms  $S_{\phi_3}$ )
- 1021 – Original  $\phi_4$ -dimensions:  $\{7, 8\}$  from Head 1;  $\{15, 16\}$  from Head 2. (Forms  $S_{\phi_4}$ )

1022 PCA is applied to the  $N$  samples of 8-dimensional vectors formed by  $[S_{\phi_3}, S_{\phi_4}]$ .

1023 Here, RoRoPE with FreqFold performs 2 PCA operations, but each operates on larger, 8-dimensional  
1024 vectors which are concatenations of what were previously separate PCA targets.

## 1025 C.2 Formalizing the Benefit of FreqFold in PCA

1026 The example above illustrates that FreqFold causes a re-grouping and concatenation of data segments  
1027 prior to PCA. The benefit of this concatenation is explained by the following proposition. It states  
1028 that performing PCA jointly on these concatenated segments (as FreqFold enables) is more effective  
1029 at preserving variance (and thus minimizing loss) than the alternative of performing separate PCAs  
1030 on the original, smaller segments and then notionally combining their outcomes.

1031 Consider one such FreqFold merge: suppose  $M$  original RoPE frequency indices  $l_1, \dots, l_M$  are  
1032 deemed equivalent by FreqFold. Without FreqFold, each  $l_p$  would correspond to a dataset  $X_p$  (e.g.,  
1033  $N$  samples of  $2g$ -dimensional key segments). With FreqFold, these  $M$  datasets are concatenated into  
1034 a single larger dataset  $X_{merged} = [X_1, X_2, \dots, X_M]$ , and PCA is applied to  $X_{merged}$ .

1035 **Proposition 2.** *Let  $M$  distinct groups of key segments  $X_1, X_2, \dots, X_M$  be identified. Each  $X_p \in$   
1036  $\mathbb{R}^{N \times d'}$  (where  $p \in \{1, \dots, M\}$ ) consists of  $N$  samples of  $d'$ -dimensional vectors. Assume data in  
1037 each  $X_p$  is mean-centered. Let  $S_p = \frac{1}{N-1} X_p^T X_p \in \mathbb{R}^{d' \times d'}$  be its covariance matrix. FreqFold  
1038 causes these  $M$  groups to be merged for a single PCA operation.*

1039 *Define  $V_1 = \sum_{p=1}^M \lambda_{p,1}$ , where  $\lambda_{p,1}$  is the largest eigenvalue of  $S_p$ . This  $V_1$  represents the sum of  
1040 variances if each of the  $M$  original groups  $X_p$  were individually reduced to its single most dominant  
1041 dimension.*

1042 *Let  $Z = [X_1, X_2, \dots, X_M] \in \mathbb{R}^{N \times (M \cdot d')}$  be the dataset formed by concatenating the features  
1043 (columns) of these  $M$  groups. Let  $S_{concat} = \frac{1}{N-1} Z^T Z \in \mathbb{R}^{(M \cdot d') \times (M \cdot d')}$  be its covariance matrix.*

1044 *Define  $V_2 = \sum_{j=1}^M \mu_j$ , where  $\mu_1 \geq \mu_2 \geq \dots \geq \mu_M$  are the  $M$  largest eigenvalues of  $S_{concat}$ . This  
1045  $V_2$  represents the variance captured if the concatenated data  $Z$  is reduced to  $M$  dimensions using  
1046 PCA.*

Then, the variance captured by the joint PCA on the FreqFold-merged data ( $V_2$ ) is greater than or equal to the sum of variances from optimally reducing each original group to one dimension ( $V_1$ ):

$$V_2 \geq V_1$$

This proposition explains that FreqFold’s strategy of enabling PCA over larger, concatenated segments (formed by merging data from RoPE frequencies deemed similar) is mathematically favored for variance preservation compared to separate, more fragmented PCAs.

### C.3 Proof of Proposition 2

The objective is to prove that  $V_2 \geq V_1$ , using the notation from Proposition 2. The proof strategy is to construct a specific  $M$ -dimensional subspace for the concatenated data  $Z$ . We show that the variance captured by projecting  $Z$  onto this particular subspace equals  $V_1$ . Since the PCA procedure yielding  $V_2$  finds the optimal  $M$ -dimensional subspace maximizing captured variance,  $V_2$  must be at least  $V_1$ .

Let  $\lambda_{p,1}$  be the largest eigenvalue of  $S_p$  (covariance of  $X_p$ ), and  $w_{p,1} \in \mathbb{R}^{d'}$  be its corresponding eigenvector. So,  $S_p w_{p,1} = \lambda_{p,1} w_{p,1}$  and  $w_{p,1}^T w_{p,1} = 1$ . The variance  $\lambda_{p,1} = w_{p,1}^T S_p w_{p,1}$ .  $V_1 = \sum_{p=1}^M \lambda_{p,1}$ .

For the concatenated data  $Z$ ,  $V_2 = \sum_{j=1}^M \mu_j$ . By Ky Fan’s theorem for matrix eigenvalues:

$$V_2 = \max_{\substack{U \in \mathbb{R}^{(M \cdot d') \times M} \\ U^T U = I_M}} \text{Tr}(U^T S_{\text{concat}} U)$$

where  $U$ ’s columns form an orthonormal basis for an  $M$ -dimensional subspace of  $\mathbb{R}^{M \cdot d'}$ .

Construct  $U^* = [u_1^*, \dots, u_M^*] \in \mathbb{R}^{(M \cdot d') \times M}$ . For  $p \in \{1, \dots, M\}$ , define  $u_p^* \in \mathbb{R}^{M \cdot d'}$ :

$$u_p^* = \begin{pmatrix} 0_{d' \times 1} \\ \vdots \\ w_{p,1} \\ \vdots \\ 0_{d' \times 1} \end{pmatrix} \quad (\text{as the } p\text{-th block of size } d')$$

The set  $\{u_1^*, \dots, u_M^*\}$  is orthonormal. The variance retained by projecting  $Z$  onto the subspace of  $U^*$  is:

$$\text{Tr}((U^*)^T S_{\text{concat}} U^*) = \sum_{p=1}^M (u_p^*)^T S_{\text{concat}} u_p^*$$

Let  $S_{qr}$  be the  $(q, r)$ -th block of  $S_{\text{concat}}$ , where  $S_{qr} = \frac{1}{N-1} X_q^T X_r$ . Note  $S_{pp} = S_p$ . Each term  $(u_p^*)^T S_{\text{concat}} u_p^* = w_{p,1}^T S_{pp} w_{p,1} = w_{p,1}^T S_p w_{p,1} = \lambda_{p,1}$ . So,  $\text{Tr}((U^*)^T S_{\text{concat}} U^*) = \sum_{p=1}^M \lambda_{p,1} = V_1$ . Since  $V_2$  is the maximum possible variance:

$$V_2 \geq \text{Tr}((U^*)^T S_{\text{concat}} U^*) = V_1$$

Thus,  $V_2 \geq V_1$ . This proves Proposition 2.

### C.4 Discussion on the Trade-off in FreqFold

While Proposition 2 demonstrates a clear benefit of FreqFold in terms of PCA efficiency—specifically, that merging  $M$  original frequency groups allows for greater variance preservation when reducing to  $M$  dimensions—it is crucial to acknowledge an inherent trade-off. The foundational assumption of FreqFold is the approximation of numerically similar RoPE base frequencies as effectively identical. This approximation, by its very nature, introduces a degree of deviation from the original, precise RoPE formulation.

The extent of this deviation, and thus the potential loss in the fidelity of positional encoding, typically correlates with how aggressively frequencies are grouped. A larger  $M$  or a looser criterion for



similarity when grouping frequencies can amplify this approximation error. Consequently, while increasing the dimensionality of vectors undergoing PCA is beneficial from the perspective of PCA variance capture as shown by the proposition, it may simultaneously increase the lossiness of the RoPE approximation itself. Therefore, the practical application of FreqFold requires a careful balancing act. The parameter  $M$  (representing the number of original RoPE frequencies treated as one effective frequency for PCA purposes) or the specific grouping strategy for frequencies must be chosen to optimize this trade-off.

## D Balancing Key-Value Norms and Low-Rank Approximation

This appendix elaborates on the Key-Value (KV) balancing technique and the subsequent joint low-rank approximation applied to the NoPE (No Positional Encoding) components of the keys and the values, as mentioned in Section 4.3 of the main paper. After the RoRoPE procedure (Section 4.2), the key projection matrix  $W^K$  is effectively split into two components:  $W_{\text{RoPE}}^{DK} \in \mathbb{R}^{d \times D}$  corresponding to the single head that retains RoPE, and  $W_{\text{NoPE}}^{DK} \in \mathbb{R}^{(g-1)d \times D}$  corresponding to the remaining  $g-1$  head components that do not use RoPE. The value projection matrix is denoted as  $W^{DV} \in \mathbb{R}^{gd \times D}$ .

### D.1 KV Balancing: Purpose and Formulation

**Purpose** The primary goal of KV balancing is to ensure that the principal component analysis (PCA), when applied jointly to the NoPE key and value activations, is not disproportionately influenced by components with larger norms. We observed that the activations derived from  $W_{\text{NoPE}}^{DK}$  (i.e.,  $\mathbf{k}_{\text{NoPE},t} = W_{\text{NoPE}}^{DK} \mathbf{x}_t$ ) often have a significantly larger average norm than those from  $W^{DV}$  (i.e.,  $\mathbf{v}_t = W^{DV} \mathbf{x}_t$ ). Without balancing, PCA would predominantly capture the variance within the NoPE key components, potentially neglecting important variations in the value components.

**Formulation** To address this imbalance, we introduce a scaling factor  $\alpha$ . This factor is computed as the ratio of the expected L2 norms of the NoPE key activations to the value activations, based on a calibration dataset:

$$\alpha = \frac{\mathbb{E}_t[\|W_{\text{NoPE}}^{DK} \mathbf{x}_t\|_2]}{\mathbb{E}_t[\|W^{DV} \mathbf{x}_t\|_2]} \quad (32)$$

where  $\mathbf{x}_t \in \mathbb{R}^D$  is the  $t$ -th input token.

While the main paper states scaling  $W_{\text{NoPE}}^{DK}$  by  $1/\alpha$  and  $W^{UK}$  by  $\alpha$  for mathematical equivalence in the model’s output, for the purpose of deriving the PCA projection, we effectively use scaled NoPE key activations. That is, the activations used to compute the PCA basis are  $\mathbf{k}'_{\text{NoPE},t} = 1/\alpha \cdot W_{\text{NoPE}}^{DK} \mathbf{x}_t$  and  $\mathbf{v}_t = W^{DV} \mathbf{x}_t$ . This ensures that the PCA process considers features from keys and values on a more equitable footing with respect to their magnitudes. The subsequent low-rank decomposition will then be applied to  $W_{\text{NoPE}}^{DK}$  and  $W^{DV}$ , using the PCA basis derived from these balanced activations.

### D.2 Joint Low-Rank Approximation of NoPE Keys and Values using PCA

After determining the scaling factor  $\alpha$ , we proceed to compress the projection matrices associated with the NoPE keys ( $W_{\text{NoPE}}^{DK}$ ) and all values ( $W^{DV}$ ) jointly.

The process is as follows:

1. **Collect Calibrated Activations:** A small calibration dataset (WikiText-2) is used. For each input  $\mathbf{x}_t$  from this dataset, we compute the scaled NoPE key activations  $\mathbf{k}'_{\text{NoPE},t}$  and the value activations  $\mathbf{v}_t$ . These are concatenated to form combined activation vectors:

$$\mathbf{c}_{\text{NoPE},t} = \begin{pmatrix} \mathbf{k}'_{\text{NoPE},t} \\ \mathbf{v}_t \end{pmatrix} \in \mathbb{R}^{(2g-1)d} \quad (33)$$

2. **Perform PCA:** PCA is performed on the set of collected combined activation vectors  $\{\mathbf{c}_{\text{NoPE},t}\}$ . This involves computing the covariance matrix of these vectors and finding its principal components. The eigenvectors (corresponding to the largest eigenvalues) are selected to form the columns of a projection matrix  $R_{KV} \in \mathbb{R}^{((2g-1)d) \times r_{kv}}$ , where  $r_{kv}$  is the reduced rank. This matrix  $R_{KV}$  captures the directions of highest variance in the (balanced) combined NoPE key and value activation space.



Table 2: Composition of the training dataset.

Dataset	Sampling Weight
fineweb-edu-dedup	0.70
cosmopedia-v2	0.15
python-edu	0.06
open-web-math	0.08
stackoverflow	0.01

1112 **3. Low-Rank Decomposition of Projection Matrices:** Let  $W^{DKV} = \begin{pmatrix} W_{\text{NoPE}}^{DK} \\ W^{DV} \end{pmatrix} \in \mathbb{R}^{((2g-1)d) \times D}$   
1113 be the initial projection matrix that transforms the input  $\mathbf{x}_t$  into an intermediate NoPE Key and Value  
1114 representation  $\mathbf{c}_{\text{NoPE},t} = W^{DKV} \mathbf{x}_t$ . Further, let  $W^{UKV} = \begin{pmatrix} W_{\text{NoPE}}^{UK} & 0 \\ 0 & W^{UV} \end{pmatrix} \in \mathbb{R}^{2hd \times ((2g-1)d)}$   
1115 represent the subsequent collective projection matrix that takes  $\mathbf{c}_{\text{NoPE},t}$  and processes it to produce  
1116 the actual keys and values required by the attention mechanism for the NoPE components, where  
1117  $W_{\text{RoPE}}^{UK} \in \mathbb{R}^{hd \times gd}$  and  $W_{\text{NoPE}}^{UK} \in \mathbb{R}^{hd \times (g-1)d}$  are two parts of  $W^{UK}$  that participate in and do not  
1118 participate in the RoPE computation, respectively. The original sequence of operations for these  
1119 components can be expressed as  $W^{UKV} W^{DKV} \mathbf{x}_t \in \mathbb{R}^{2hd}$ , in which the first  $hd$  elements correspond  
1120 to the keys and the following  $hd$  elements correspond to the values.  
1121 To introduce a low-rank bottleneck, we modify both  $W^{DKV}$  and  $W^{UKV}$  using the PCA projection  
1122 matrix  $R_{KV}$ .

- 1123 • The initial projection matrix  $W^{DKV}$  is transformed into  $W^{DKV'} \in \mathbb{R}^{r_{kv} \times D}$ :

$$W^{DKV'} = R_{KV}^T W^{DKV} \quad (34)$$

1124 This new matrix  $W^{DKV'}$  takes the original input  $\mathbf{x}_t$  and projects it into a compressed  
1125  $r_{kv}$ -dimensional latent space, which is the actual content stored in the KV cache for the  
1126 NoPE components.

- 1127 • The subsequent projection matrix  $W^{UKV}$  is transformed into  $W^{UKV'} \in \mathbb{R}^{2hd \times r_{kv}}$ :

$$W^{UKV'} = W^{UKV} R_{KV} \quad (35)$$

1128 This new matrix  $W^{UKV'}$  now takes the compressed latent representation as input and  
1129 produces the final representations for the NoPE components that are used in the attention  
1130 calculation. As we can see,  $W^{UKV'}$  is actually the concatenated form of  $W^{UK}$  and  $W^{UV}$   
1131 in MLA:

$$W^{UKV'} = \begin{pmatrix} W^{UK} \\ W^{UV} \end{pmatrix} \quad (36)$$

1132 This joint decomposition allows for a more holistic compression by identifying shared latent structures  
1133 between NoPE keys and values, guided by the balanced PCA.

## 1134 E Experimental Settings of Fine-tuning

1135 **Datasets** Following the experimental setups of MHA2MLA, we fine-tune our models using the  
1136 pretraining corpus from SmolLM [7]. The dataset comprises FineWeb-Edu-Dedup [23], Cosmopedia-  
1137 v2 — a synthetic dataset generated by Mixtral [40], Python-Edu from StarCoder [24], Open-Web-  
1138 Math [29], and data from StackOverflow [34]. To ensure a fair comparison with the MHA2MLA  
1139 baseline, we constructed our training dataset using the same data composition strategy. Specifically,  
1140 we replicate the dataset mixing ratios used in the MHA2MLA setup to maintain experimental  
1141 consistency, which is shown in Table 2.

1142 **Hyperparameters** The fine-tuning hyperparameters for models of all sizes are listed in Table 3. In  
1143 the table, entries with a slash (/) indicate a two-step training process.

Table 3: Training details across different models.

	<b>SmolLM 1B7</b>		<b>LLaMA2 7B</b>		
	<b>-68.75%</b>	<b>-87.50%</b>	<b>-68.75%</b>	<b>-87.50%</b>	<b>-92.97%</b>
Batch size	64	64	64	64 / 64	256 / 64
Learning rate	1e-4	1e-4	2e-5	2e-5 / 2e-5	1e-4 / 2e-5
Tokens	300M	1B	500M	2B / 1B	5B / 1B
Warmup ratio	0.03	0.08	0	0 / 0.03	0 / 0.03
lr scheduler	constant	constant	constant	constant / cosine	constant / cosine
Sequence length	2048	2048	4096	4096	4096

## F Detail Information for vLLM Benchmark

In Section 5.4, we demonstrated the speedup achieved by TransMLA—which compresses 92.97% of the KV cache—compared to the original LLaMA-2-7B model. This section provides a detailed analysis of throughput across various hardware configurations.

To account for the effects of both the prefilling and decoding stages, we adopt a setting where the input and output lengths are equal. For instance, with a total context length of 1k, we set the input length to 512 tokens and the output length to 512 tokens. Most experiments are conducted using 100 requests to compute the average throughput. However, for shorter context lengths such as 1k, inference is extremely fast, leading to some timing fluctuations. To mitigate this, we increase the number of requests to 1000 for more stable measurements.

While the original LLaMA-2-7B model supports a maximum context length of 4096 tokens, we extend this limit to 32k tokens in our evaluation. Detailed throughput results are presented in Table 4.

On a GPU with 165.2 TFLOPS of compute and 24GB of memory, the LLaMA-2-7B model runs out of memory when the context length reaches 16k tokens. In contrast, TransMLA sustains a throughput of 414.41 tokens per second under the same conditions. On a more powerful GPU with 320 TFLOPS and 64GB of memory, we employ a development version of the vLLM framework. We anticipate that the throughput of TransMLA will improve further with the release of future optimized versions of the framework tailored for this hardware.

Table 4: Throughput comparison between LLaMA-2-7b and TransMLA at varying input lengths and number of requests.

Context Length	Requests	Model	Throughput(output tokens/s)		
			165.2 TF 24GB	312 TF 40GB	320 TF 64GB
1K	1000	LLaMA-2-7b	653.81	1579.26	1249.13
		TransMLA	<b>3043.65</b>	<b>4062.43</b>	<b>1798.17</b>
2K	100	LLaMA-2-7b	352.85	850.14	789.31
		TransMLA	<b>2241.87</b>	<b>2577.01</b>	<b>1080.73</b>
4K	100	LLaMA-2-7b	173.09	441.37	442.63
		TransMLA	<b>1318.78</b>	<b>1926.15</b>	<b>1021.03</b>
8K	100	LLaMA-2-7b	85.80	218.51	216.66
		TransMLA	<b>832.69</b>	<b>1118.18</b>	<b>870.15</b>
16K	100	LLaMA-2-7b	OOM	110.58	112.13
		TransMLA	<b>414.41</b>	<b>601.36</b>	<b>483.22</b>
32K	100	LLaMA-2-7b	OOM	38.32	55.69
		TransMLA	OOM	<b>243.81</b>	<b>278.09</b>

## G Case Study

To provide an intuitive understanding of TransMLA’s impact on model performance, this section presents several examples from vLLM’s docs. We compare the outputs of three model variants: (1) a model with 92.97% of its KV cache compressed without any fine-tuning; (2) a model pretrained on

1166 6B tokens, as detailed in Table 1; and (3) a model fine-tuned for one epoch on the SmolTalk dataset,  
1167 following the setup described in [4]. The results are summarized in Table 5.

1168 As shown in Table 5, even without any additional training, the compressed model is still able to  
1169 produce coherent and meaningful responses. This demonstrates the effectiveness of techniques such  
1170 as RoRoPE, FreqFold, and BKV-PCA in significantly mitigating performance degradation. Moreover,  
1171 with a modest amount of pretraining or supervised fine-tuning (SFT), the model’s performance  
1172 improves substantially. These findings highlight TransMLA’s potential as a general framework for  
1173 converting various GQA models into MLA models, with promising prospects for aligning with the  
performance of advanced systems like DeepSeek R1.

Table 5: Examples from different model configurations. **Red** indicates input; black indicates output. “w/o Training” denotes the TransMLA-compressed model (92.97% KV cache) without further training. “Pre-Training” and “Fine-Tuning” show outputs after pretraining on a 6B-token corpus and SFT on SmolTalk [5], respectively.

Model	Prompt & Generated Text
w/o Training	<b>Hello, my name is</b> Katiu, my father’s dog, the pet of the 3600 year-old tribe, Kint. The Kangs were part of a race of reptiles. A small handful
Pre-Training	<b>Hello, my name is</b> Sasha and I am in third grade at Meadows. You may be wondering what this article is about. Well, I have been doing a lot of research on the water cycle and decided to write about it.
Fine-Tuning	<b>Hello, my name is</b> Emily, and I’m a 20-year-old college student. My hobbies include painting, writing, and photography. I also enjoy playing the guitar.
w/o Training	<b>The president of the United States is</b> elected by the legislature. The legislature controls the national armed forces, but only provides the funds to establishing a national guard.
Pre-Training	<b>The president of the United States is</b> elected to a four-year term by the people of each state in a general election held every four years on the Tuesday following the first Monday in November.
Fine-Tuning	<b>The president of the United States is</b> not a position to be taken lightly. This person is the chief executive of the United States of America, and has immense power and influence.
w/o Training	<b>The capital of France is</b> Paris. Its geographical position in the Iberian Plain of France, Spain, Spain, and Morocco are the four largest cities. This region is located in Asia, Spain and Morocco.
Pre-Training	<b>The capital of France is</b> a major business city and it is a favorite destination for businesses from all over the world. It has a strategic location in the heart of the European Union, which makes it one of the most popular cities in Europe.
Fine-Tuning	<b>The capital of France is</b> Paris, and it is one of the most popular tourist destinations in the world. It is a city that offers something for everyone, from art and history to food and fashion.
w/o Training	<b>The future of AI is</b> in serious risk to create a major breakthrough in this emerging phenomenon in the history of artificial intelligence.
Pre-Training	<b>The future of AI is</b> looking bright. With advancements in technology and the increasing availability of data, AI is expected to become more intelligent and capable of performing even more complex tasks.
Fine-Tuning	<b>The future of AI is</b> The future of AI is more nuanced and complex than we might think. Here are some potential developments that could shape the future of AI.

1174