Higher-order Tensor Product Attention

TPA Authors

Abstract

Higher-order Tensor Product Attention.

1 Higher-Order Tensor Product Attention

All prior discussions in Zhang et al. (2025) have focused on a *second-order* factorization in which each rank-R_Q (and similarly R_K, R_V) component is the outer product of two vectors: one in $\mathbb{R}^h$ (the "head" dimension) and one in $\mathbb{R}^{d_h}$. We now generalize this by introducing an additional latent factor, yielding a *third-order* (or higher) factorization reminiscent of canonical polyadic (CP) decomposition. Concretely, for a single token t, we write

$$\mathbf{Q}_t = \frac{1}{R_Q} \sum_{r=1}^{R_Q} \mathbf{a}_r^Q(\mathbf{x}_t) \otimes \operatorname{vec}\big(\mathbf{b}_r^Q(\mathbf{x}_t) \otimes \mathbf{c}_r^Q(\mathbf{x}_t)\big),$$

where the newly introduced factor $\mathbf{c}_r^Q(\mathbf{x}_t) \in \mathbb{R}^{d_\mathbf{c}}$ can be viewed as a learnable gate or modulation term. Analogous expansions apply to $\mathbf{K}_t$ and $\mathbf{V}_t$. In practice, these triple (or higher-order) products still collapse into a matrix in $\mathbb{R}^{h \times d_h}$. One straightforward way to achieve this collapse is to split the feature dimension d_h such that $d_b \times d_c = d_h$,

$$\mathbf{b}_r^Q(\mathbf{x}_t) \in \mathbb{R}^{d_b}, \quad \mathbf{c}_r^Q(\mathbf{x}_t) \in \mathbb{R}^{d_c}, \quad \operatorname{vec}\big(\mathbf{b}_r^Q(\mathbf{x}_t) \otimes \mathbf{c}_r^Q(\mathbf{x}_t)\big) \in \mathbb{R}^{d_h}.$$

This additional factor can enhance expressiveness without necessarily increasing the base rank. Conceptually, it can act as a learnable nonlinearity or gating mechanism. One could also tie or share $\mathbf{c}_r^Q$ across queries, keys, and values, to reduce parameter overhead.

A similar setup holds for keys (with rank R_K) and values (with rank R_V). Although this extra dimension adds to the parameter count, it can reduce the required rank to achieve a certain level of representational power.

From a memory perspective, higher-order TPA still leverages factorized KV caching: only the factors $\mathbf{a}(\mathbf{x}_t), \mathbf{b}(\mathbf{x}_t)$, and $\mathbf{c}(\mathbf{x}_t)$ for each past token are cached. As usual, a trade-off arises between model capacity and the overhead of memory and computing. Nonetheless, moving from a rank-$\big(R_Q, R_K, R_V\big)$ matrix factorization to a higher-order tensor decomposition can provide additional flexibility and increased capacity.

1.1 RoPE Compatibility in Higher-Order TPA

Rotary positional embeddings (RoPE) remain compatible even under higher-order factorizations. In second-order TPA, RoPE can be treated as an invertible blockwise linear map acting on the last dimension of $\mathbf{Q}_t$ or $\mathbf{K}_t$. The same argument carries over when a third factor $\mathbf{c}_r^Q(\mathbf{x}_t)$ is present. Suppose RoPE acts on the $\mathbf{b}_r^Q(\mathbf{x}_t)$ portion (of dimension size d_b), we have the following theorem.

Theorem 1 (RoPE Compatibility in Higher-Order TPA)**.** Consider the higher-order (3-order) Tensor Product Attention (TPA) query factorization

$$\mathbf{Q}_t = \frac{1}{R_Q} \sum_{r=1}^{R_Q} \mathbf{a}_r^Q(\mathbf{x}_t) \otimes \operatorname{vec}\big(\mathbf{b}_r^Q(\mathbf{x}_t) \otimes \mathbf{c}_r^Q(\mathbf{x}_t)\big) \in \mathbb{R}^{h \times d_h},$$

where $\mathbf{a}_r^Q(\mathbf{x}_t) \in \mathbb{R}^h$, $\mathbf{b}_r^Q(\mathbf{x}_t) \in \mathbb{R}^{d_b}$, $\mathbf{c}_r^Q(\mathbf{x}_t) \in \mathbb{R}^{d_c}$, with $d_c = \frac{d_h}{d_b}$. Define the RoPE-transformed query as $\widetilde{\mathbf{Q}}_t = \mathrm{RoPE}_t\big(\mathbf{Q}_t\big) = \mathbf{Q}_t\mathbf{T}_t$, where

$$\mathbf{T}_t = \mathbf{R}_t \otimes \mathbf{I}_{d_c} = \begin{pmatrix} \mathbf{R}_t & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_t & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{R}_t \end{pmatrix} \in \mathbb{R}^{d_h \times d_h},$$

and $\mathbf{R}_t \in \mathbb{R}^{d_b \times d_b}$ ($d_b \in \mathbb{Z}_+$ is even) is a block-diagonal matrix composed of 2×2 rotation matrices:

$$\mathbf{R}_t = \begin{pmatrix} \cos(t\theta_1) & -\sin(t\theta_1) & & & & & \\ \sin(t\theta_1) & \cos(t\theta_1) & & & & & \\ & & \cos(t\theta_2) & -\sin(t\theta_2) & & & \\ & & \sin(t\theta_2) & \cos(t\theta_2) & & & \\ & & & & \ddots & & \\ & & & & & \cos(t\theta_{d_b/2}) & -\sin(t\theta_{d_b/2}) \\ & & & & & \sin(t\theta_{d_b/2}) & \cos(t\theta_{d_b/2}) \end{pmatrix},$$

for $t \in \{1, \ldots, T\}$ and $j \in \{1, \ldots, d_b/2\}$.

This construction ensures that RoPE rotates only the coordinates corresponding to $\mathbf{b}_r^Q(\mathbf{x}_t)$ while leaving $\mathbf{c}_r^Q(\mathbf{x}_t)$ unchanged. Under these conditions, the RoPE-transformed query $\mathrm{RoPE}_t\big(\mathbf{Q}_t\big)$ admits a higher-order TPA factorization of the same rank R_Q. Specifically, we have

$$\frac{1}{R_Q}\sum_{r=1}^{R_Q} \mathbf{a}_r^Q(\mathbf{x}_t) \otimes \mathrm{vec}\Big(\widetilde{\mathbf{b}}_r^Q(\mathbf{x}_t) \otimes \mathbf{c}_r^Q(\mathbf{x}_t)\Big) = \mathrm{RoPE}_t\big(\mathbf{Q}_t\big), \tag{1.1}$$

where $\widetilde{\mathbf{b}}_r^Q(\mathbf{x}_t) = \mathbf{R}_t\mathbf{b}_r^Q(\mathbf{x}_t)$.

Please see Appendix A for the proof. For fourth-order or higher, this result still holds.

References

Yifan Zhang, Yifeng Liu, Huizhuo Yuan, Zhen Qin, Yang Yuan, Quanquan Gu, and Andrew Chi-Chih Yao. Tensor product attention is all you need. *arXiv preprint arXiv:2501.06425*, 2025.

Appendix

A Proofs of Theorems

Proof of Theorem 1.

Proof. We begin by observing that each term $\mathbf{a}_r^Q(\mathbf{x}_t) \otimes \operatorname{vec}\big(\mathbf{b}_r^Q(\mathbf{x}_t) \otimes \mathbf{c}_r^Q(\mathbf{x}_t)\big)$ is an element of $\mathbb{R}^h \otimes \mathbb{R}^{d_h}$. Here, $\mathbf{b}_r^Q(\mathbf{x}_t) \in \mathbb{R}^{d_b}$, $\mathbf{c}_r^Q(\mathbf{x}_t) \in \mathbb{R}^{d_c}$, with $d_c = \frac{d_h}{d_b}$. Consequently, the tensor product $\mathbf{b}_r^Q(\mathbf{x}_t) \otimes \mathbf{c}_r^Q(\mathbf{x}_t)$ forms a $d_b \times d_c$ matrix, and its vectorization lies in $\mathbb{R}^{d_b \cdot d_c} = \mathbb{R}^{d_h}$.

Applying the RoPE transformation to a single summand yields

$$\operatorname{vec}\big(\mathbf{b}_r^Q(\mathbf{x}_t) \otimes \mathbf{c}_r^Q(\mathbf{x}_t)\big) \mapsto \mathbf{T}_t \operatorname{vec}\big(\mathbf{b}_r^Q(\mathbf{x}_t) \otimes \mathbf{c}_r^Q(\mathbf{x}_t)\big).$$

Since $\mathbf{T}_t$ is defined as the Kronecker product $\mathbf{R}_t \otimes \mathbf{I}_{d_c}$, where $\mathbf{R}_t \in \mathbb{R}^{d_b \times d_b}$ and $\mathbf{I}_{d_c}$ is the identity matrix of size $d_c \times d_c$, it follows that

$$\mathbf{T}_t \operatorname{vec}\big(\mathbf{b}_r^Q(\mathbf{x}_t) \otimes \mathbf{c}_r^Q(\mathbf{x}_t)\big) = \operatorname{vec}\big(\mathbf{R}_t \mathbf{b}_r^Q(\mathbf{x}_t) \otimes \mathbf{c}_r^Q(\mathbf{x}_t)\big).$$

This is because the Kronecker product with an identity matrix effectively applies the rotation $\mathbf{R}_t$ to the $\mathbf{b}_r^Q(\mathbf{x}_t)$ component while leaving $\mathbf{c}_r^Q(\mathbf{x}_t)$ unchanged.

Therefore, the RoPE transformation of a single summand becomes

$$\operatorname{RoPE}_t\Big(\mathbf{a}_r^Q(\mathbf{x}_t) \otimes \operatorname{vec}\big(\mathbf{b}_r^Q(\mathbf{x}_t) \otimes \mathbf{c}_r^Q(\mathbf{x}_t)\big)\Big) = \mathbf{a}_r^Q(\mathbf{x}_t) \otimes \operatorname{vec}\big(\mathbf{R}_t \mathbf{b}_r^Q(\mathbf{x}_t) \otimes \mathbf{c}_r^Q(\mathbf{x}_t)\big).$$

Importantly, this transformation does not mix the components $\mathbf{b}_r^Q(\mathbf{x}_t)$ and $\mathbf{c}_r^Q(\mathbf{x}_t)$; it solely rotates $\mathbf{b}_r^Q(\mathbf{x}_t)$ via $\mathbf{R}_t$.

Summing over all ranks $r = 1, \ldots, R_Q$, we obtain

$$\frac{1}{R_Q} \sum_{r=1}^{R_Q} \mathbf{a}_r^Q(\mathbf{x}_t) \otimes \operatorname{vec}\big(\mathbf{R}_t \mathbf{b}_r^Q(\mathbf{x}_t) \otimes \mathbf{c}_r^Q(\mathbf{x}_t)\big) = \operatorname{RoPE}_t\big(\mathbf{Q}_t\big),$$

which retains the same higher-order TPA structure with rank R_Q.

Thus, the RoPE transformation is fully compatible with higher-order TPA, preserving the factorization rank and maintaining the structure by only rotating the $\mathbf{b}_r^Q(\mathbf{x}_t)$ components while leaving $\mathbf{c}_r^Q(\mathbf{x}_t)$ unchanged. □