(Manifest AI)
Scaling Context Requires Rethinking Attention

[ICLR 2025 preprint]

Hankyo Jeong

December 18, 2025

Seoul National University

KV Cache

- Autoregressive inference at step t uses past $\{k_1, \ldots, k_t\}$ and $\{v_1, \ldots, v_t\}$.
- Recomputing past K, V each step increases cost.
- Store past K, V as KV cache.

KV-cache state for length T : state $\sim O(cT)$, for some contant c

Causal Self-Attention

- $q_i, k_i \in \mathbb{R}^d$, $v_i \in \mathbb{R}^v$.

$$Q = \begin{bmatrix} q_1^\top \\ \vdots \\ q_t^\top \end{bmatrix} \in \mathbb{R}^{t \times d}, \quad K = \begin{bmatrix} k_1^\top \\ \vdots \\ k_t^\top \end{bmatrix} \in \mathbb{R}^{t \times d}, \quad V = \begin{bmatrix} v_1^\top \\ \vdots \\ v_t^\top \end{bmatrix} \in \mathbb{R}^{t \times v}.$$

$$M \in \mathbb{R}^{t \times t}, \qquad M_{ij} = \mathbf{1}_{i \geq j}.$$

$$\mathrm{attn}_{\mathrm{exp}}(Q, K, V) = \big(\exp(QK^\top) \odot M\big) V.$$

- Compute $t \times t$ interactions $\Rightarrow$ cost scales with t^2.
- KV cache grows with t.

Two Limits of Long Context

- Exponential attention:

  compute $\sim O(t^2 \cdot d)$, state $\sim O(t)$.

- Linear attention:

  compute $\sim O(t)$, state $\sim O(1)$ (fixed feature size).

- Fixed-size state can limit performance at long context.

Goal: larger state capacity + GPU-parallel execution form.

Linear Attention

Linear Attention

- Feature map: $\phi : \mathbb{R}^d \to \mathbb{R}^D$.
- Row-wise application:

$$\phi(Q) = \begin{bmatrix} \phi(q_1)^\top \\ \vdots \\ \phi(q_t)^\top \end{bmatrix} \in \mathbb{R}^{t \times D} \quad \text{and} \quad \phi(K) \in \mathbb{R}^{t \times D}.$$

$$\mathrm{attn}_{\mathrm{lin}}^{\phi}(Q, K, V) = \big(\phi(Q)\phi(K)^\top \odot M\big) V.$$

Linear Attention: Recurrent Form and Linear Cost

- Index i denotes the i-th row (time step).

$$S_i := \sum_{j=1}^{i} v_j \, \phi(k_j)^\top \in \mathbb{R}^{v \times D}.$$

$$\mathrm{attn}_{\mathrm{lin}}^{\phi}(Q, K, V)_i = \sum_{j=1}^{i} \phi(q_i)^\top \phi(k_j) \, v_j = \left(\sum_{j=1}^{i} v_j \, \phi(k_j)^\top \right) \phi(q_i) = S_i \, \phi(q_i).$$

- $v_i \phi(k_i)^\top : O(vD), \qquad S_i \phi(q_i) : O(vD).$
- Total: O(tDv).

State Size Limit in Linear Attention

- Linear attention replaces KV cache with $S_i \in \mathbb{R}^{v \times D}$.
- State capacity depends on D.
- If D is fixed, state capacity stays fixed as t increases.

Problem: fixed $D \rightarrow$ fixed state capacity at long context.

Power Attention

Power Attention: from $\exp(QK^\top)$ to $(QK^\top)^{\odot p}$

$$\mathrm{attn}_{\exp}(Q,K,V) = \big(\exp(S) \odot M\big)V = \big(\exp(QK^\top) \odot M\big)V,$$

$$\mathrm{attn}^{p}_{\mathrm{pow}}(Q,K,V) = \big(S^{\odot p} \odot M\big)V = \big((QK^\top)^{\odot p} \odot M\big)V.$$

$A^{\odot p}$: elementwise p-th power.

Choose ϕ by Tensor Power (TPOW)

Lemma 1

Let $p \in \mathbb{N}$. define $\mathrm{TPOW}_p : \mathbb{R}^d \to \mathbb{R}^{d^p}$ by

$$\phi \;=\; \mathrm{TPOW}_p(x) \;=\; \begin{bmatrix} x_1 \cdots x_1 \\ x_1 \cdots x_2 \\ \vdots \\ x_d \cdots x_d \end{bmatrix} \;=\; \left[\prod_k x_{i_k} \right]_{(i_1,\ldots,i_p) \in \mathbb{N}_d^{\times p}}.$$

Then for any $q, k \in \mathbb{R}^d$, $\mathrm{TPOW}_p(q)^\top \mathrm{TPOW}_p(k) = (q^\top k)^p$.

Power attention is linear attention (with $\phi = \mathrm{TPOW}_p$)

- Linear attention with feature map $\phi : \mathbb{R}^d \to \mathbb{R}^D$:

$$\mathrm{attn}^{\phi}_{\mathrm{lin}}(Q, K, V) := \big(\phi(Q)\phi(K)^\top \odot M\big) V, \quad \phi(Q) = \begin{bmatrix} \phi(q_1)^\top \\ \vdots \\ \phi(q_t)^\top \end{bmatrix}.$$

- With $\phi = \mathrm{TPOW}_p$, Lemma1 gives, entrywise for all i, j:

$$\big(\phi(Q)\phi(K)^\top\big)_{ij} = \phi(q_i)^\top \phi(k_j) = (q_i^\top k_j)^p = \big((QK^\top)^{\odot p}\big)_{ij}.$$

$$\begin{aligned} \mathrm{attn}^{p}_{\mathrm{pow}}(Q, K, V) &= \big((QK^\top)^{\odot p} \odot M\big) V \\ &= \big(\phi(Q)\phi(K)^\top \odot M\big) V \\ &= \mathrm{attn}^{\phi}_{\mathrm{lin}}(Q, K, V). \end{aligned}$$

From TPOW to SPOW: permutation duplicates

- **Example ($p = 2$, $x = [a, b, c]^\top$).**

  TPOW produces a symmetric tensor

$$\mathrm{TPOW}_2(x) = \begin{bmatrix} aa & ab & ac \\ ab & bb & bc \\ ac & bc & cc \end{bmatrix}$$

  SPOW produces unique elements of that tensor

  $\mathrm{SPOW}_2(x) = [aa, ab, ac, bb, bc, cc]\top$.

SPOW

- scaled by a coefficient that depends on the permutation count.

$$\mathrm{SPOW}_2\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1 x_1 \\ \sqrt{2}\, x_1 x_2 \\ x_2 x_2 \end{bmatrix}, \qquad \mathrm{SPOW}_3\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = \begin{bmatrix} x_1 x_1 x_1 \\ \sqrt{3}\, x_1 x_1 x_2 \\ \sqrt{3}\, x_1 x_2 x_2 \\ x_2 x_2 x_2 \end{bmatrix}.$$

Which gives:

1. For $x, y \in \mathbb{R}^d$:

$$\langle \mathrm{SPOW}_p(x), \mathrm{SPOW}_p(y) \rangle \;=\; (x^\top y)^p \;=\; \langle \mathrm{TPOW}_p(x), \mathrm{TPOW}_p(y) \rangle.$$

2. Dimension:

$$D \;=\; \binom{d+p-1}{p}.$$

Recap: Power attention via ϕ

- Power attention:

  $$\mathrm{attn}^p_{\mathrm{pow}}(Q,K,V) = \left((QK^\top)^{\odot p} \odot M\right) V.$$

- If ϕ satisfies $\phi(q)^\top \phi(k) = (q^\top k)^p$, then

  $$\mathrm{attn}^p_{\mathrm{pow}}(Q,K,V) = \mathrm{attn}^\phi_{\mathrm{lin}}(Q,K,V).$$

  $$\phi = \mathrm{TPOW}_p \Rightarrow D = d^p, \qquad \phi = \mathrm{SPOW}_p \Rightarrow D = \binom{d+p-1}{p}.$$

GPU Parallelism via Chunking

Chunking for GPU Execution

- Recurrent update: $S_i \leftarrow S_{i-1}$ creates a dependency chain.
- GPU throughput is tied to batched matmul kernels.
- Chunking rewrites computation into chunk-level operations.
- Let chunk size be c and chunk index be n.
- Use only boundary states: $S_0, S_c, S_{2c}, \ldots$.

Chunked Form

- Treat Q, K as feature-space inputs (after ϕ) to match the chunk derivation.

$$Y_{nc+m} = S_{nc} Q_{nc+m} + \sum_{j=nc+1}^{nc+m} \left(Q_{nc+m}^{\top} K_j\right) V_j, \qquad m \in \{1, \ldots, c\}.$$

$$S_{c(n+1)} = S_{cn} + V_{(n)c} K_{(n)c}^{\top}.$$

- Cost: $O(tDv + tcd)$.

Conclusion

- Exponential attention: compute scales with t^2, KV-cache state scales with t.
- Linear attention: recurrent state $S_i \in \mathbb{R}^{v \times D}$, compute $O(tDv)$.
- Chunking: rewrite recurrent computation into GPU matmul form, cost $O(tDv + tcd)$.
- Power attention: score $(q^\top k)^p$ with a feature map satisfying $\phi(q)^\top \phi(k) = (q^\top k)^p$.
- TPOW gives $D = d^p$; SPOW gives $D = \binom{d+p-1}{p}$.

Thank you!

Appendix

Appendix A: Chunked Form Setup

- Chunk size c, chunk index i.
- Chunk matrices: $Q_{(i)c}, K_{(i)c}, V_{(i)c}$.
- Boundary state: S_{ci}.

$$Y_{(i)c} = S_{ci} Q_{(i)c} + V_{(i)c} \big(Q_{(i)c} K_{(i)c}^{\top} \odot M \big), \tag{1}$$

$$S_{c(i+1)} = S_{ci} + V_{(i)c} K_{(i)c}^{\top}. \tag{2}$$

Appendix A: Position-wise Expansion

$$Y_i = S_{c\lfloor i/c \rfloor} Q_i + \sum_{j=\lfloor i/c \rfloor c+1}^{i} (Q_i K_j^\top) V_j \tag{3}$$

$$= \left(S_{c(\lfloor i/c \rfloor - 1)} + V_{(i)c} K_{(i)c}^\top\right) Q_i + \sum_{j=\lfloor i/c \rfloor c+1}^{i} (Q_i K_j^\top) V_j \tag{4}$$

$$= S_{c(\lfloor i/c \rfloor - 1)} Q_i + \sum_{j=\lfloor (i-1)/c \rfloor c+1}^{\lfloor i/c \rfloor c} V_j K_j^\top Q_i + \sum_{j=\lfloor i/c \rfloor c+1}^{i} (Q_i K_j^\top) V_j \tag{5}$$

$$= S_{c(\lfloor i/c \rfloor - 1)} Q_i + \sum_{j=\lfloor (i-1)/c \rfloor c+1}^{\lfloor i/c \rfloor c} (K_j^\top Q_i) V_j + \sum_{j=\lfloor i/c \rfloor c+1}^{i} (Q_i K_j^\top) V_j. \tag{6}$$

Appendix A: Telescoping to Attention Form

$$Y_i = S_{c(\lfloor i/c \rfloor - 1)} Q_i + \sum_{j=\lfloor (i-1)/c \rfloor c+1}^{\lfloor i/c \rfloor c} (Q_i K_j^\top) V_j + \sum_{j=\lfloor i/c \rfloor c+1}^{i} (Q_i K_j^\top) V_j \tag{7}$$

$$= S_{c(\lfloor i/c \rfloor - 1)} Q_i + \sum_{j=\lfloor (i-1)/c \rfloor c+1}^{i} (Q_i K_j^\top) V_j \tag{8}$$

$$= \cdots = S_0 Q_i + \sum_{j=1}^{i} (Q_i K_j^\top) V_j \tag{9}$$

$$= \sum_{j=1}^{i} (Q_i K_j^\top) V_j \qquad (\text{if } S_0 = 0). \tag{10}$$

Appendix B: Tensor Power Identity

$$\mathrm{TPOW}(x, p) = \left[\prod_{k=1}^{p} x_{i_k}\right]_{(i_1,\ldots,i_p)\in[d]^p} \tag{11}$$

$$\mathrm{TPOW}(x, p)^\top \mathrm{TPOW}(y, p) = \sum_{(i_1,\ldots,i_p)\in[d]^p} x_{i_1}\cdots x_{i_p}\, y_{i_1}\cdots y_{i_p} \tag{12}$$

$$= \left(\sum_{i_1\in[d]} x_{i_1} y_{i_1}\right)\left(\sum_{i_2\in[d]} x_{i_2} y_{i_2}\right)\cdots\left(\sum_{i_p\in[d]} x_{i_p} y_{i_p}\right) \tag{13}$$

$$= (x^\top y)^p. \tag{14}$$

- With $\phi = \mathrm{TPOW}(\cdot, p)$, $(q^\top k)^p = \phi(q)^\top \phi(k)$.
- Power attention fits linear attention with state expansion $D = d^p$.

Appendix B.2: Symmetric power (SPOW) definition

Index SPOW by a multi-index $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$ with $|\alpha| := \sum_{i=1}^d \alpha_i = p$.

$$\mathrm{SPOW}_p(x)_\alpha := \sqrt{\frac{p!}{\alpha_1! \cdots \alpha_d!}} \prod_{i=1}^d x_i^{\alpha_i}, \qquad \alpha \in \mathbb{N}_0^d,\ |\alpha| = p. \tag{27}$$

- The number of such α is $D = \binom{d+p-1}{p}$.
- This removes permutation duplicates of ordered indices.

Appendix B.2: Inner product identity (multinomial theorem)

Let $x, y \in \mathbb{R}^d$. Then

$$\langle \mathrm{SPOW}_p(x), \mathrm{SPOW}_p(y) \rangle = \sum_{|\alpha|=p} \left(\sqrt{\frac{p!}{\alpha!}} \prod_{i=1}^{d} x_i^{\alpha_i} \right) \left(\sqrt{\frac{p!}{\alpha!}} \prod_{i=1}^{d} y_i^{\alpha_i} \right) \tag{28}$$

$$= \sum_{|\alpha|=p} \frac{p!}{\alpha!} \prod_{i=1}^{d} (x_i y_i)^{\alpha_i} \tag{29}$$

$$= \left(\sum_{i=1}^{d} x_i y_i \right)^p \tag{30}$$

$$= (x^\top y)^p. \tag{31}$$

Here $\alpha! := \alpha_1! \cdots \alpha_d!$.

Appendix B.2: Conclusion

$$\langle \mathrm{SPOW}_p(x), \mathrm{SPOW}_p(y) \rangle = (x^\top y)^p = \langle \mathrm{TPOW}_p(x), \mathrm{TPOW}_p(y) \rangle. \tag{32}$$

- SPOW_p can replace TPOW_p in power attention.
- SPOW_p reduces the feature dimension from d^p to $\binom{d+p-1}{p}$.