# Spectral Dynamics of Attention Caches:
## A Numerical Linear Algebra Perspective on Transformer Inference

**Clément Richefort (c.richefort@fz-juelich.de)**, *Xinzhe Wu*, *Edoardo Di Napoli*
*Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany*
*SciCoCo.nrw 2026 · RWTH Aachen · 21–23 September*

# Theoretical Framework & Compression Strategy

### KV cache growth

Transformers store past keys and values. With $P$ layers, $d$ embeddings and context size $L$,

$$\text{memory}(KV) = 2PLd.$$

For $L = 10^5$, $P = 32$, $d = 4096$: $\approx 26$ GB. After projection,

$$Y_Q = XW_Q, \quad Y_K = XW_K, \quad Q = \text{RoPE}(Y_Q), \quad K = \text{RoPE}(Y_K), \quad V = XW_V.$$

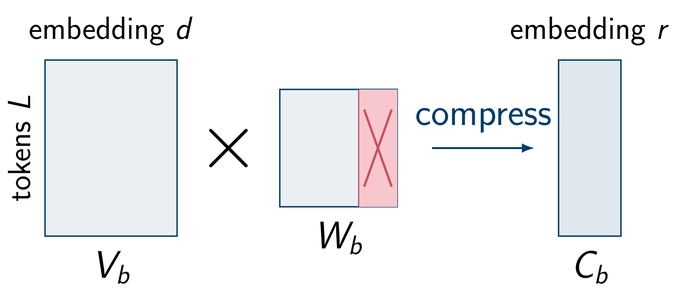
Decoding only appends a block $\Delta X$:

$$V_b^\top = \begin{bmatrix} V_a^\top & \Delta V^\top \end{bmatrix}, \qquad \Delta V \in \mathbb{R}^{\Delta L \times d}.$$

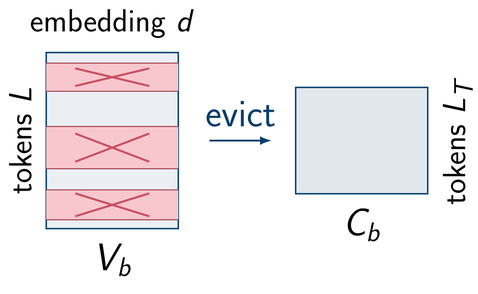
The new-token output uses the full key/value cache:

$$\Delta O = \text{softmax}\left(\frac{\Delta Q}{\sqrt{d}} K_b^\top\right) V_b.$$

### Embedding Compression




### Token eviction




### Gram matrix viewpoint

The thin SVD $V_a = U_a \Sigma_a W_a^\top$ is equivalent to the fixed-size SPD eigenproblem

$$G_{V,a} = V_a^\top V_a = W_a \Lambda_a W_a^\top, \qquad \Lambda = \Sigma^2.$$

Energy threshold $\tau$ selects the rank

$$r = \text{argmin}_{\hat{r}} \left\{ \frac{\sum_{i=1}^{\hat{r}} \sigma_i}{\sum_j \sigma_j} \geq \tau \right\}.$$

Appending tokens is a rank-$\Delta L$ update:

$$G_b = G_a + \Delta V^\top \Delta V.$$

After compression only $C_a$ and $W_{a,r}$ are stored, so

$$\widetilde{G}_{V,b} = W_{a,r} \Sigma_{a,r}^2 {W_{a,r}}^\top + \Delta V^\top \Delta V.$$

**Subspace Iteration**

1: **for** $k = 1, 2, \ldots$ **do**
2: $\quad W_k \leftarrow G W_{k-1}$
3: $\quad \epsilon \leftarrow \|W_k - W_{k-1} W_{k-1}^\top W_k\|_F / \|G\|_F$
4: $\quad QR \leftarrow W_k; \quad W_k \leftarrow Q$
5: $\quad$ **if** $\epsilon \leq$ `tol` **stop**
6: **end for**
7: **return** $W_k$

$\rightarrow$ Warm-start with $W_{a,r}$ instead of random $W_0$.

# Embedding Compression

### Compressing along the embeddings

Compress the embedding, keep the sequence:

$$V \approx C W_r^\top, \qquad C = V W_r \in \mathbb{R}^{L \times r}, \quad r < d.$$

Projected active space when $\Delta V$ arrives:

$$A = \Delta V W_{a,r}, \qquad B = \Delta V - A {W_{a,r}}^\top.$$

Orthonormalize the residual and enlarge the basis:

$$QR = B^\top, \qquad W = [W_{a,r}\ Q].$$

The reduced Gram in this basis is

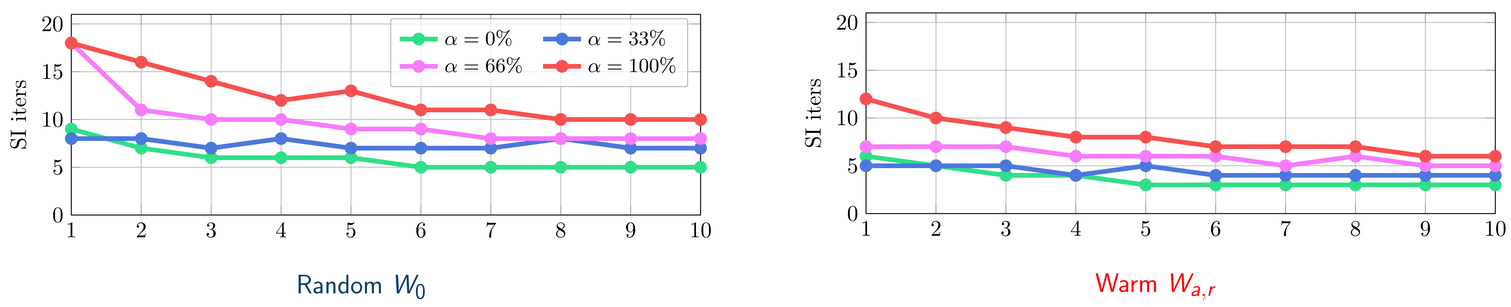
$$W^\top \widetilde{G} W = \begin{bmatrix} \Sigma_{a,r}^2 + A^\top A & A^\top R^\top \\ RA & RR^\top \end{bmatrix}.$$

If $\Delta V \in \text{range}(W_{a,r})$ then $B = 0$ and Subspace Iteration converges in one iter.

$$C_b = \begin{bmatrix} C_a Z_1 \\ A Z_1 + R^\top Z_2 \end{bmatrix},$$

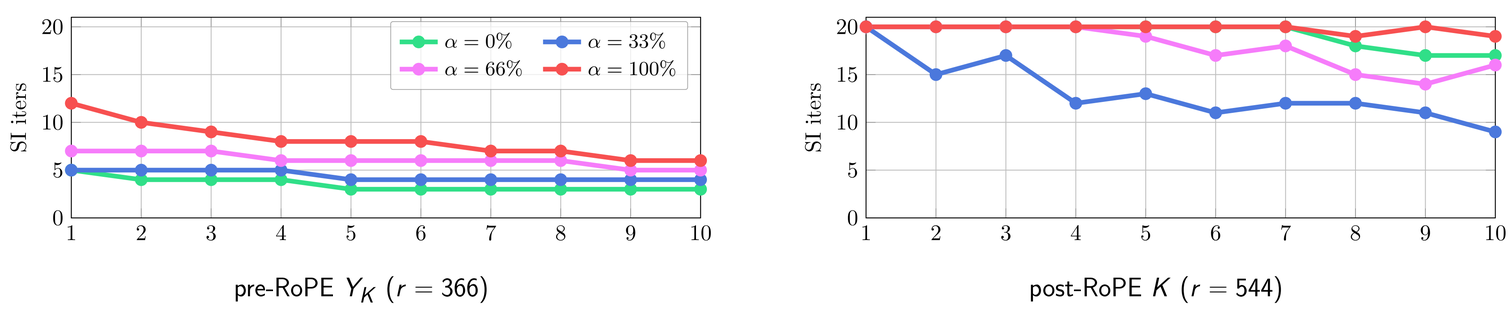
with $Z_r = [Z_1; Z_2]$ the eigenvectors of the reduced Gram matrix.

### Subspace Iterations on $V$ and $K$




Random $W_0$ Warm $W_{a,r}$

Figure: Small-GPT2 ($d = 768$), $\tau = 0.95$. $\Delta X = \alpha\,\text{Random} + (1-\alpha)\Delta X_0$. Random vs. Warm.

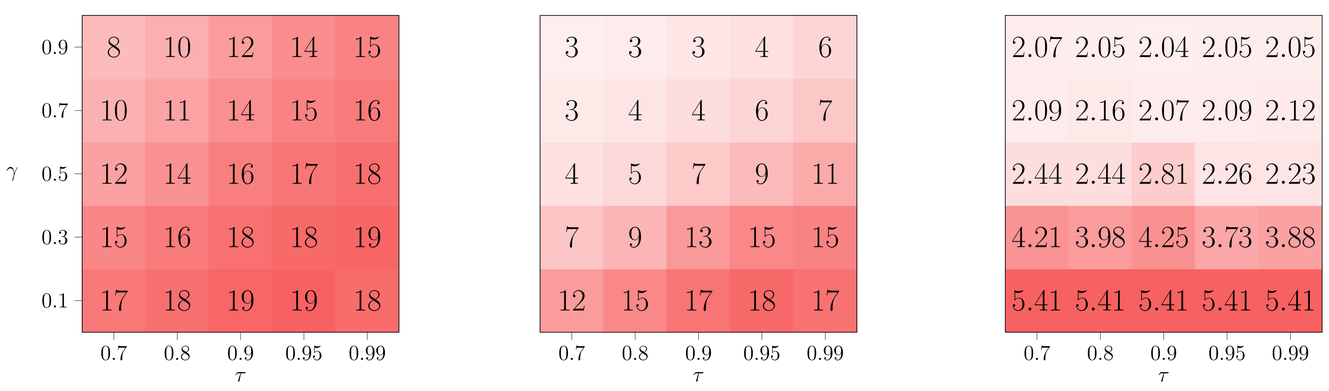



pre-RoPE $Y_K$ ($r = 366$) post-RoPE $K$ ($r = 544$)

Figure: Small-GPT2 ($d = 768$), $\tau = 0.95$. $\Delta X = \alpha\,\text{Random} + (1-\alpha)\Delta X_0$.

# Token Eviction

### Subspace Iteration vs. Perplexity measurement




Subspace Iterations (random $W_0$) Subspace Iterations (recycled) mean PPL

Figure: Qwen2-0.5B-Instruct; dense baseline PPL = 2.06. Axes: $\tau$ (bottom), keep fraction $\gamma$ (left / right). Compression ranks $r(\tau)$: 10, 14, 22, 30, 46 for $\tau = 0.7, 0.8, 0.9, 0.95, 0.99$ (independent of $\gamma$). Recycled Subspace Iteration needs fewer iterations than random $W_0$.

### Scoring the tokens on the trailing space

Embedding compression can drop semantics carried by small singular values. Eviction keeps the full embedding $d$ and shortens the sequence to a set $\mathcal{T}$ of size $L_T = \gamma L$. With selection $P_a$,

$$C_a = P_a V_a \in \mathbb{R}^{L_T \times d}, \quad L_T < L.$$

Appending new tokens and re-selecting gives

$$V_b^\top = \begin{bmatrix} C_a^\top & \Delta V^\top \end{bmatrix}, \qquad C_b = P_b V_b.$$

Outliers are tokens poorly captured by the dominant space $W_{b,r}$ of $V_b$. The low-rank residual is

$$E = V_b - V_b W_{b,r} {W_{b,r}}^\top = V_b W_{b,s} {W_{b,s}}^\top.$$

**Token eviction based on Subspace Recycling**

1: Compute $G_b = G_a + (\Delta V^\top \Delta V - D_a^\top D_a)$
2: Factorize $LL^\top = G_b$
3: Compute $W_{b,s} =$ Subspace Iteration ($L$, $W^{(0)} = W_{a,s}$)
4: Set outliers in $\mathcal{T}$ based on scores $\|E_i\|_2^2 = \left\|[V_b]_{i,:}\, W_{b,s}\right\|_2^2$
5: Compress by eviction $C_b = P_b V_b$