
UNIVERSITÀ
DEGLI STUDI
DI PADOVA

# Analysis of Transformer's internal states

A study of internal representations using Random Matrix Theory and Differential Geometry

F. Bezzi, W. Conte, E. D'Amore, G. Gasparotto

July 24, 2025

# Table of Contents

# Table of Contents

# What is a Transformer?



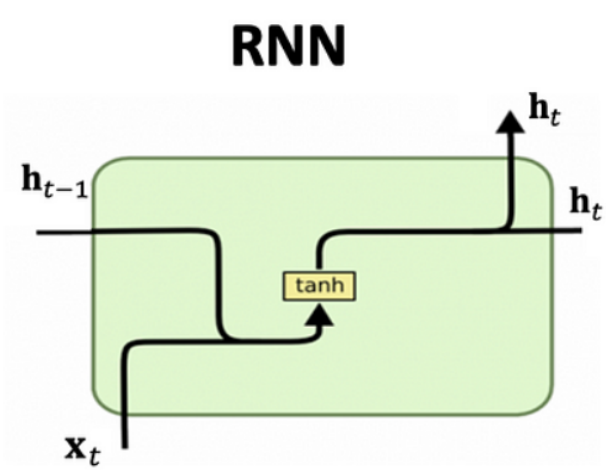
## Generalities

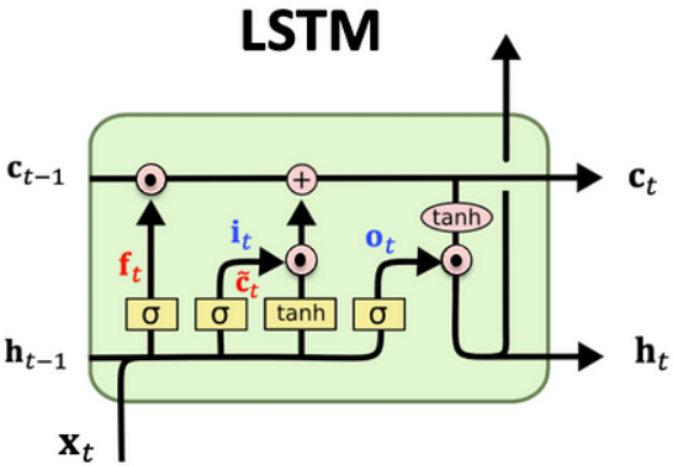
- Introduced by Vaswani et al. in 2017 — "Attention is All You Need".
- Replaces recurrence (RNN/LSTM) with self-attention.
- Processes entire input in parallel instead of step-by-step.

# Why Transformers?

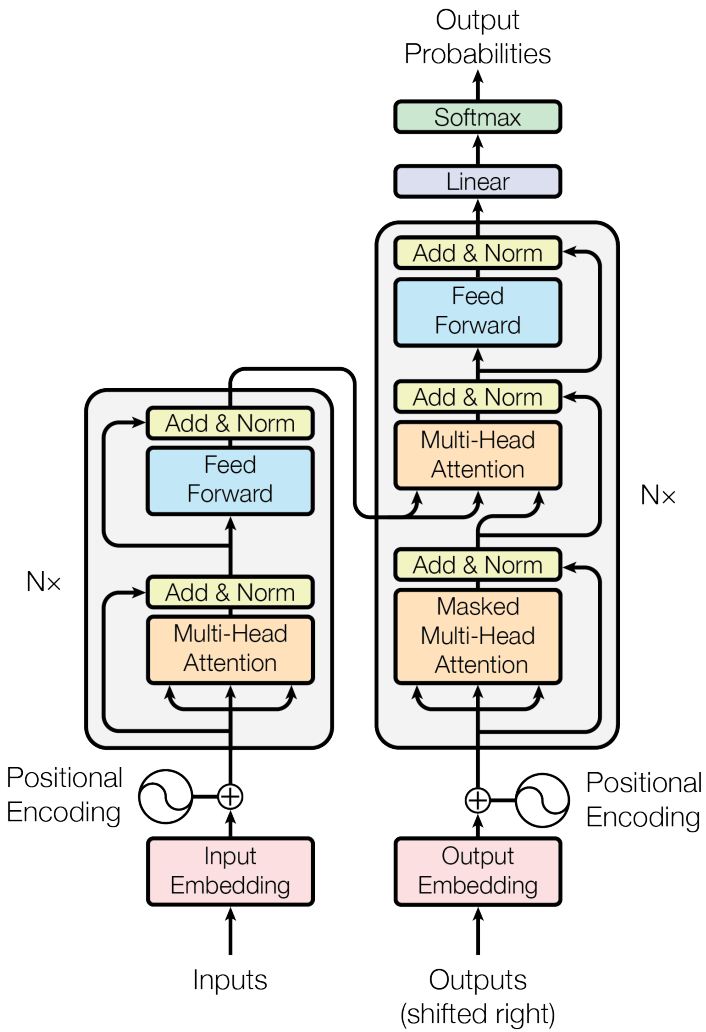


- Previous models struggled with long sequences and context.
- Slow training due to sequential processing.
- Transformers enable:
  - Better long-range dependency modeling.
  - Efficient parallelization.



# Transformer Architecture

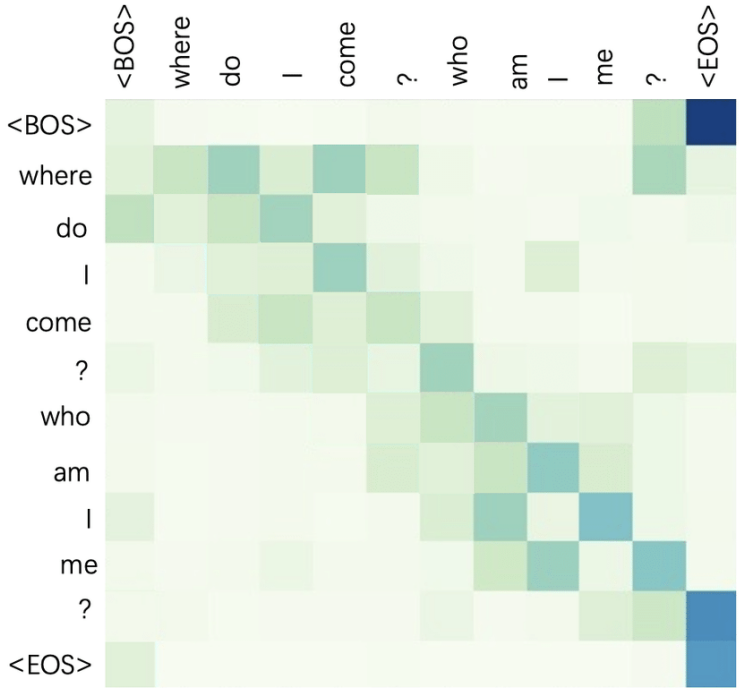



- Encoder-Decoder architecture (or just one side like GPT).
- Key components:
  - **Self-Attention** (organzied in **heads**)
  - **Feed-Forward Networks**
  - **Positional Encoding**
- Stacked layers form deep understanding of sequences.

# Self-Attention: Core Idea



- Each word attends to all others in the sequence.
- Learns contextual importance - what to "pay attention" to.
- Captures syntax and semantics.



# Terminology and notation



- We will use the model **GPT-2** $\rightarrow$ **12 decoders** and dimension of embedding space $\mathbf{d = 768}$
- We will refer to each decoder as a single **"block"**
- **Buffer:** $n \times d$ matrix of token embeddings (Often called $\boldsymbol{B}$ in the following), where $n$ is the number of tokens. Each row represents a token's vector in embedding space
- **Token:** A minimal unit of text used as input to a model

# Table of Contents

# Introduction



Our goal is to understand how the geometry of the buffer states at the output of each block evolves throughout the dynamics of the transformer algorithm, using various metrics:

- Volume of the buffer state
- Cosine similarity
- Grassmann distance
- Fraction of explored space

# log(Volume)



1. Singular value decomposition (SVD) on the buffer matrix ($B$)

$$B = U\Sigma V^T$$

2. Volume of B obtained by multiplying the resulting singular values ($\sigma_i$):

$$V_B = \prod_{i=1}^{rank(B)} \sigma_i$$

$\rightarrow$ **Useful to understand how the buffer changes its structure**

# Cosine similarity



- For each $B$, we tracked the *n-th* row $\rightarrow$ predicted token in the sequence
- Calling it as $\boldsymbol{x_{final}}$, at each step "$i$" in the dynamics, the cosine similarity is computed as:

$$\boldsymbol{S_C}(\boldsymbol{x_i}, \boldsymbol{x}_{\text{final}}) = \frac{\boldsymbol{x_i} \cdot \boldsymbol{x}_{\text{final}}}{\|\boldsymbol{x_i}\| \cdot \|\boldsymbol{x}_{\text{final}}\|}$$

where $\boldsymbol{x_i}$ is the last vector of the *i-th* buffer state.

$\rightarrow$ **Useful to understand how the predicted token evolves**

# Grassmann Distance



## Definition (Grassmann distance)

The **Grassmann distance** measures the distance between two linear subspaces of the same dimension.
Given two subspaces $\mathcal{U}, \mathcal{V} \subseteq \mathbb{R}^n$ of dimension $k$, the Grassmann distance is defined via the principal angles $\theta_1, \theta_2, \ldots, \theta_k$ between them:

$$d_G(\mathcal{U}, \mathcal{V}) = \left( \sum_{i=1}^{k} \theta_i^2 \right)^{\frac{1}{2}}$$

$\rightarrow$ **Understanding the changes in orientation through the dynamics**

# Visual intuition

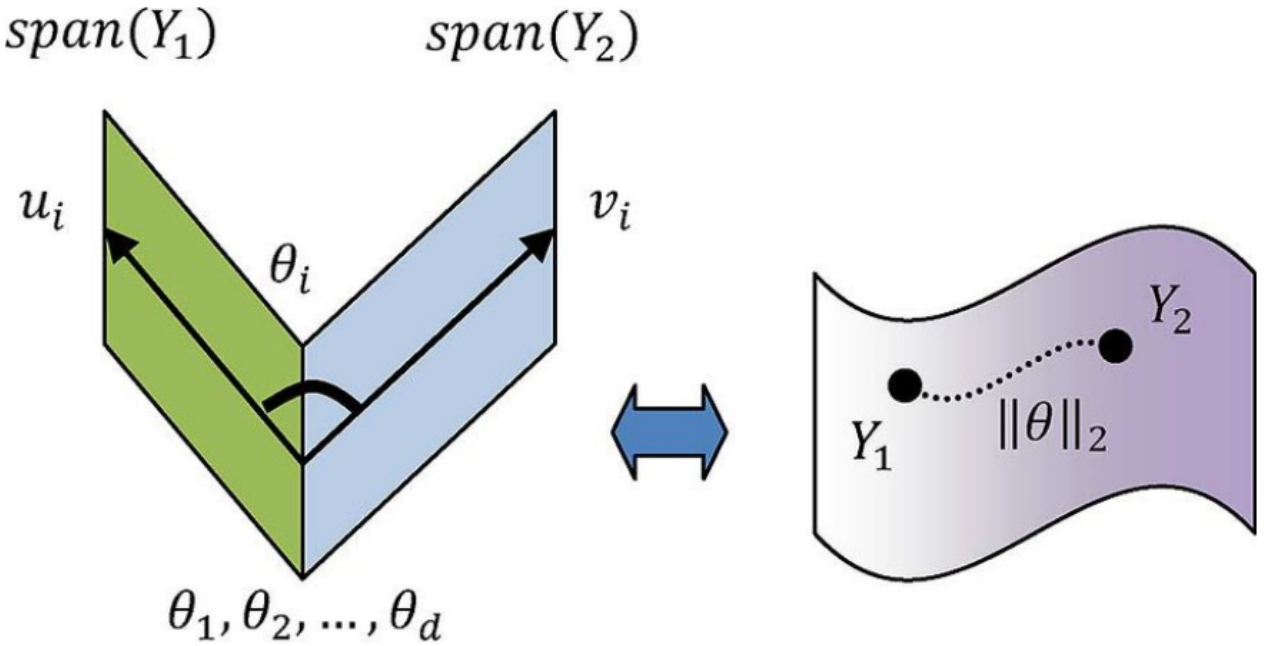
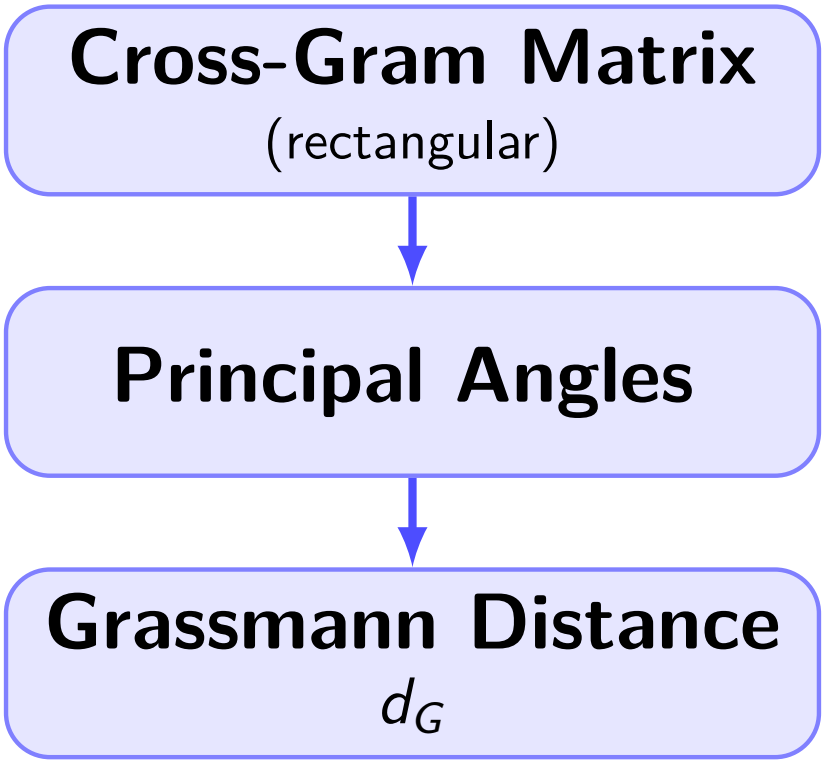



Figure: Left panel shows principal angles $\theta_i$ between the subspaces span($Y_1$) and span($Y_2$), right panel illustrates the corresponding geodesic distance $\|\theta\|_2$ between subspaces on the Grassmann manifold.

# Our approximation



*Subspace dimension might not be constant throughout the dynamics*



# $d_G$ calculation outline



1. Consider two consecutive buffer states **B** and **B'**
2. **SVD** retaining 90% of the variance $\rightarrow$ $\mathbf{B}_{\text{reduced}}$ (rank $r$) and $\mathbf{B}'_{\text{reduced}}$ (rank $r'$)
3. **Reduced QR decomposition** on the **transpose** of both low-rank matrices $\rightarrow$ **Q** and **Q'** orthonormal bases ($\dim(Q) = d \times r$, $\dim(Q') = d \times r'$)
4. **Cross-Gram Matrix (CGM)**:

$$\text{CGM} = Q^\top Q'$$

$$\dim(\text{CGM}) = r \times r'$$

5. **Singular values** of the CGM $\rightarrow$ **cosines** of the principal angles between the two (consecutive) subspaces
6. Inverse cosine $\rightarrow$ principal angles $\rightarrow$ **Grassmann distance** $d_G$

# Fraction of explored space



## Meaning

With the expression "**Fraction of explored space**" we mean how many different directions of the embedding space the buffer has explored through the dynamics

Calculation outline:

1. Given $Q_i \rightarrow$ **horizontally-stack** all of them in $Q_{total}$:

$$Q_{total} = [Q_0, Q_1, \ldots, Q_{12}]$$

2. Compute the rank of $Q_{total} \rightarrow$ dimension of the space "spanned" by the buffer
3. Performing $\frac{rank(Q_{total})}{d} \rightarrow$ **fraction of explored space**

$\rightarrow$ **Fraction of embedding regions explored by the buffer before making its prediction**

# Results of the analysis

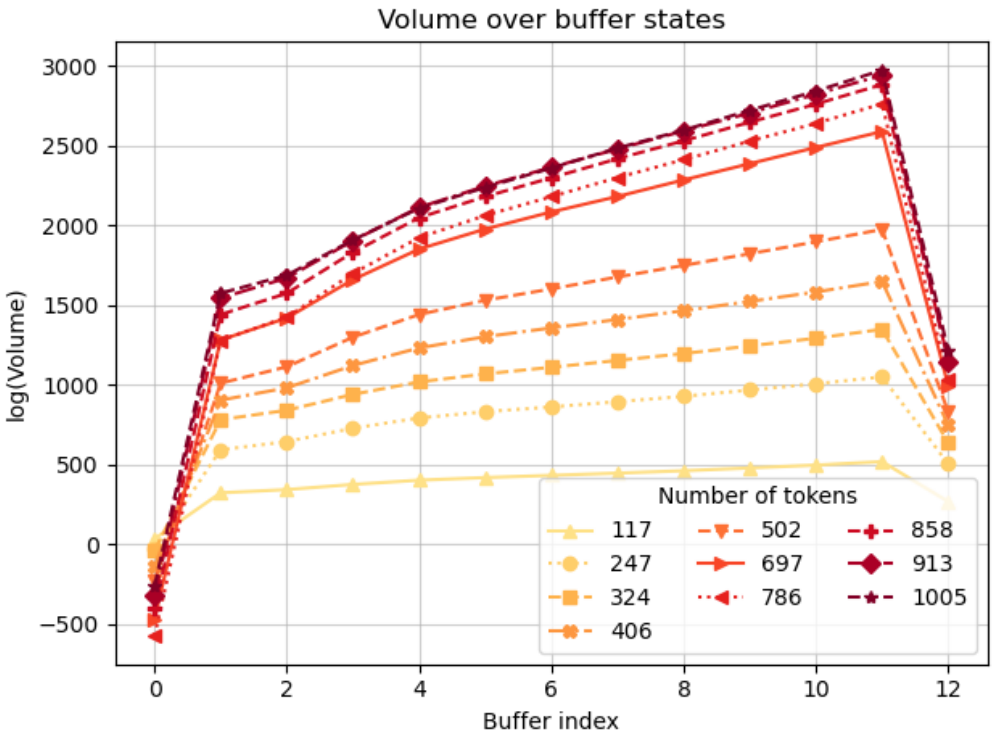


To test our metrics, we will focus on three different input prompt situations:

- 100-1000 tokens prompts;
- Around 768 tokens prompts;
- Around 100 tokens prompts from different contexts.

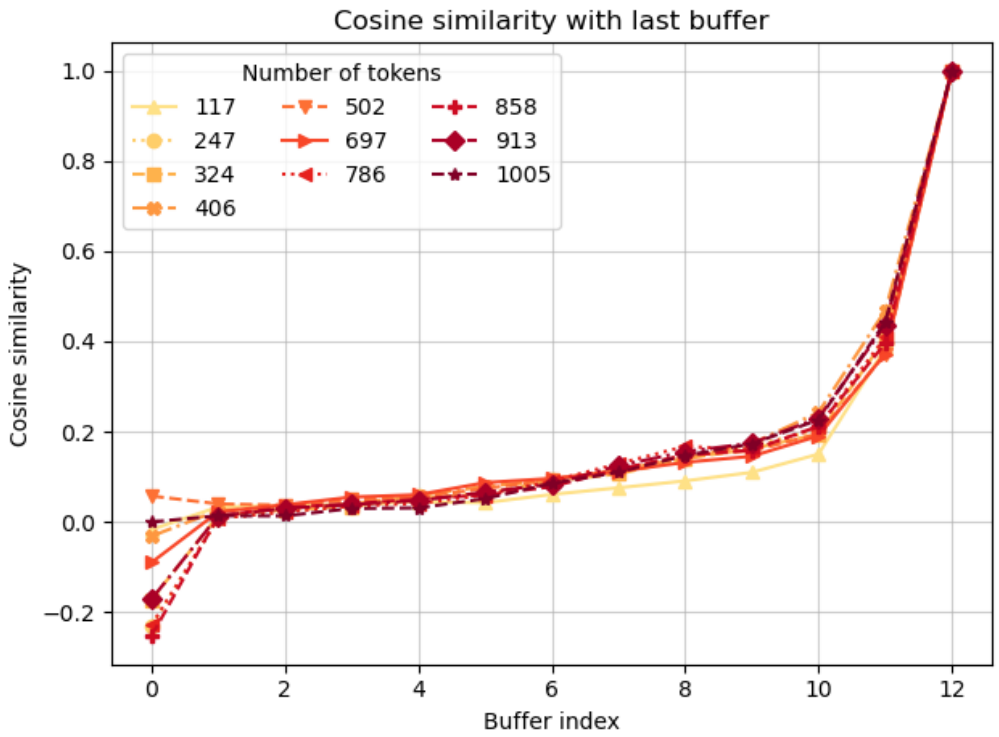
# 100 to 1000 tokens

# log(Volume)



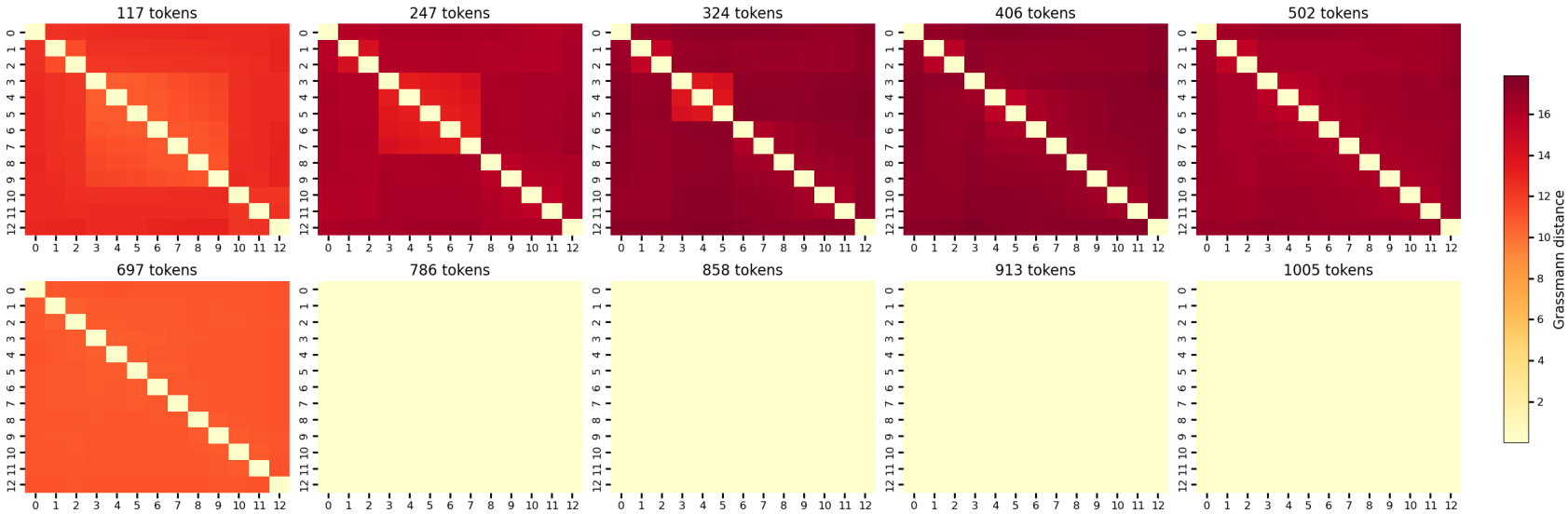

- Noticeable increase following the embedding step;
- Nearly linear growth across the middle blocks;
- Substantial volume reduction after the final block.

# Cosine similarity




- Gradually converge to the final predicted vector;
- Nearly orthogonal throughout the middle blocks → **geometric constraints and space exploration**;
- At the end there is convergence to the final predicted vector.

# Grassmann distance

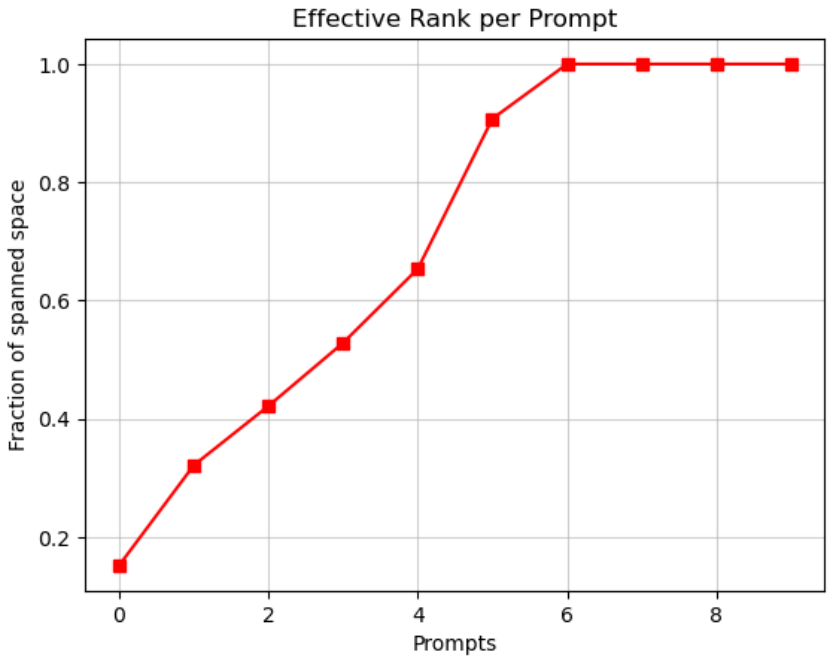



- For a small number of tokens, $d_G$ increases as more tokens are added; however, after reaching a certain point, it begins to decrease and stabilize, eventually approaching zero.

→ **A more detailed investigation of the 'zero approach' is presented in the next part of the analysis**

# Explored space






**Short prompts:**

- Each token introduces novel semantic content;
- Rapid increase in effective rank as more dimensions of the embedding space are explored.

**Longer prompts:**

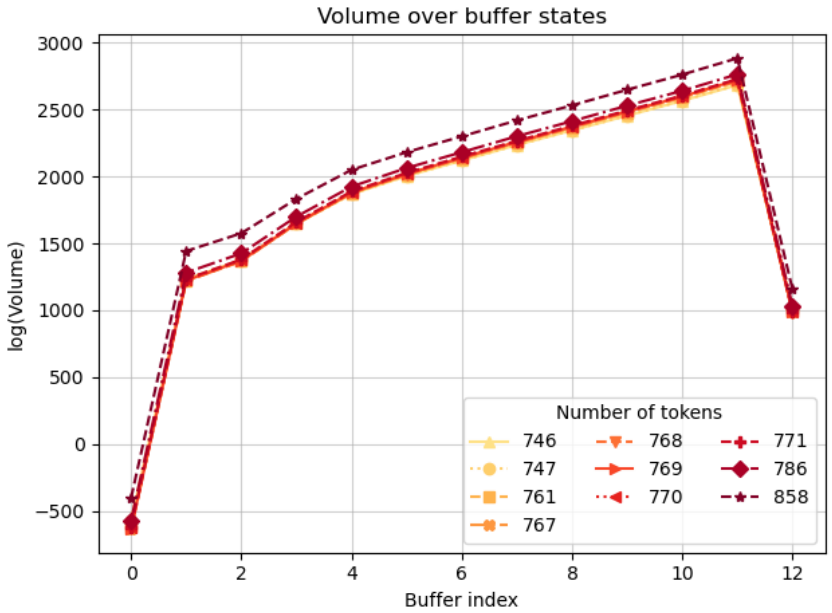
- New tokens become increasingly redundant or semantically similar;
- The model reuses existing representational directions $\rightarrow$ rank saturation.

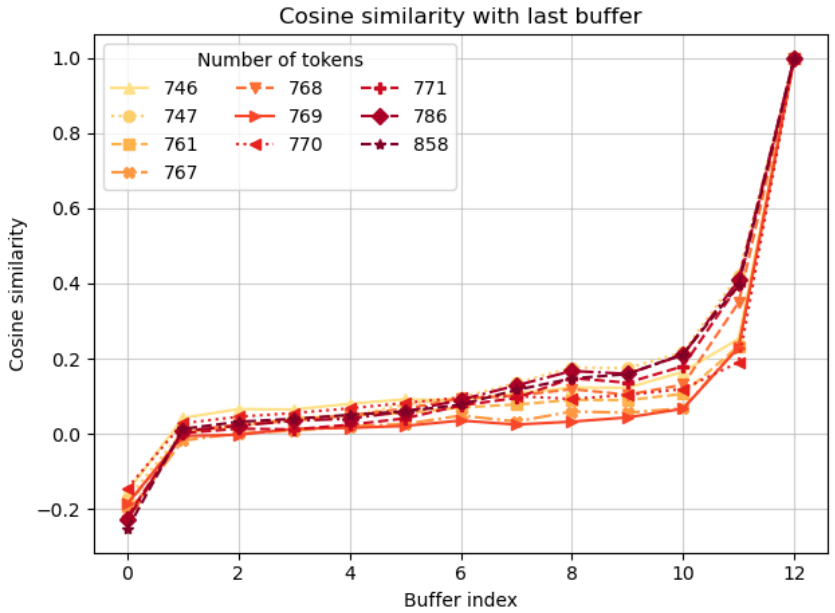
# Around 768 tokens

# log(Volume) & Cosine similarity

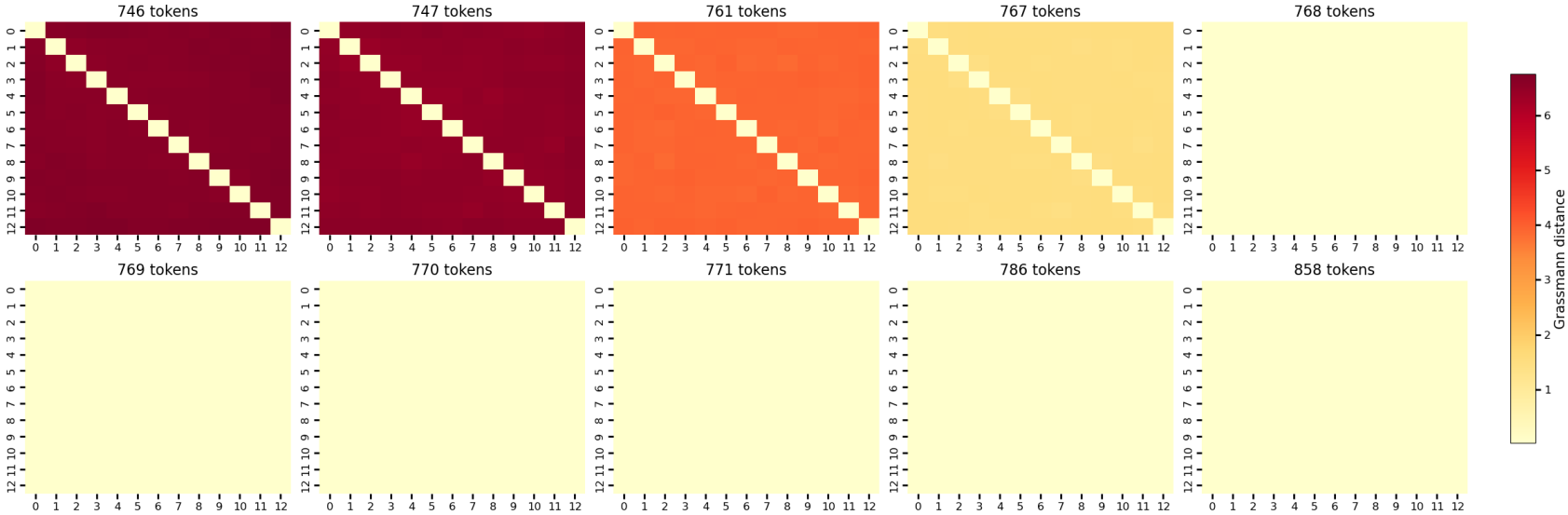


Consistent behavior observed, as previously discussed:



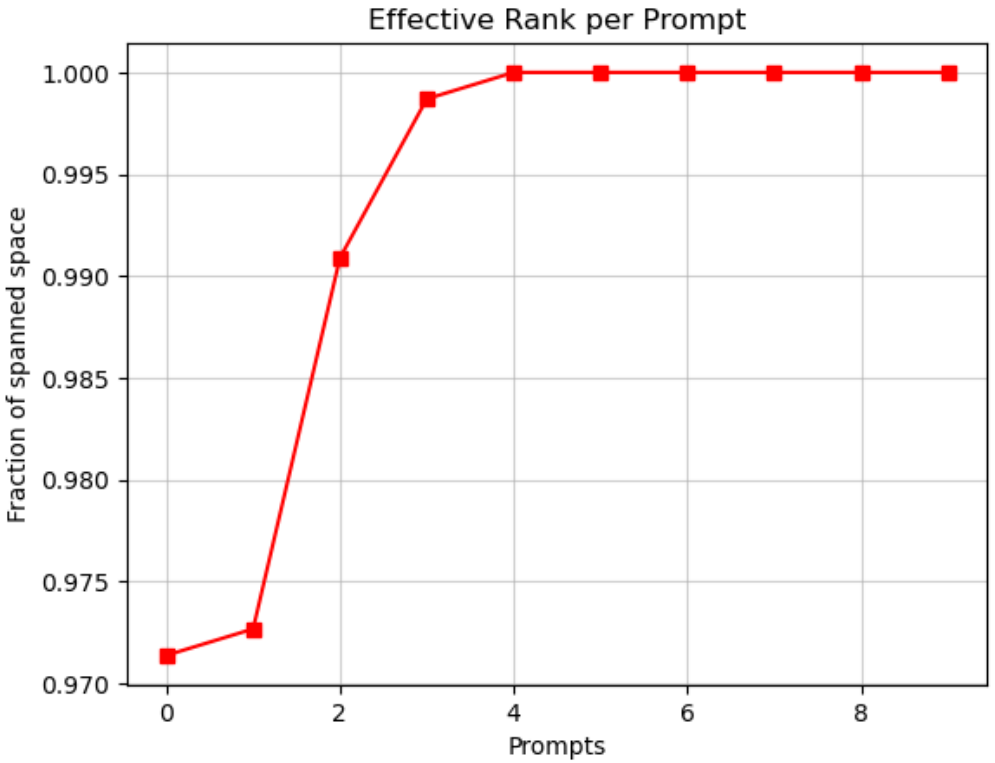



# Grassmann distance




- As the number of tokens approaches the embedding dimensionality, $d_G$ begins to decrease, and beyond that threshold, it becomes zero.

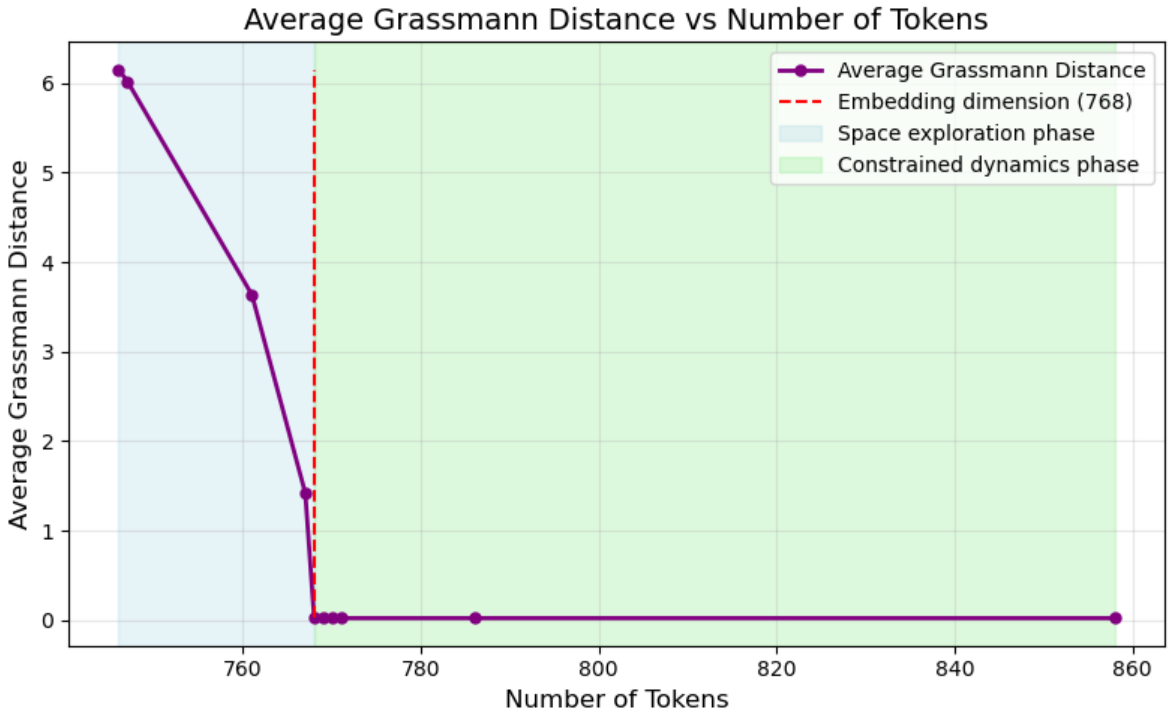
# Explored space




- Consistent with the behavior observed previously

→ **Saturation reached exactly at n = d!**

# Why does avg $d_G$ drop to zero?




- **Critical transition** at 768 tokens (embedding dimension)
- **Two regimes:** Exploration phase $\rightarrow$ Saturation phase
- **Implication:** Effective representational capacity = architectural constraint

**Model hits representational wall at embedding dimension**

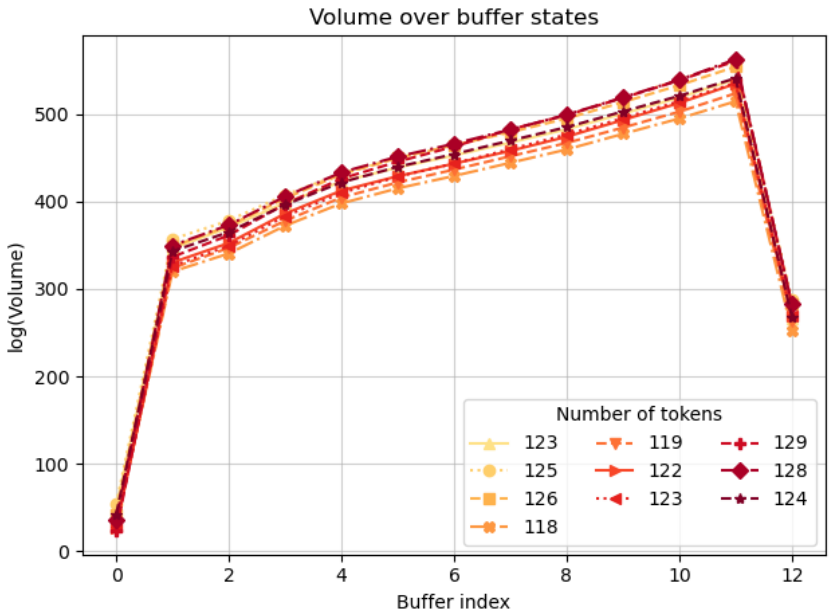
# Different context prompts

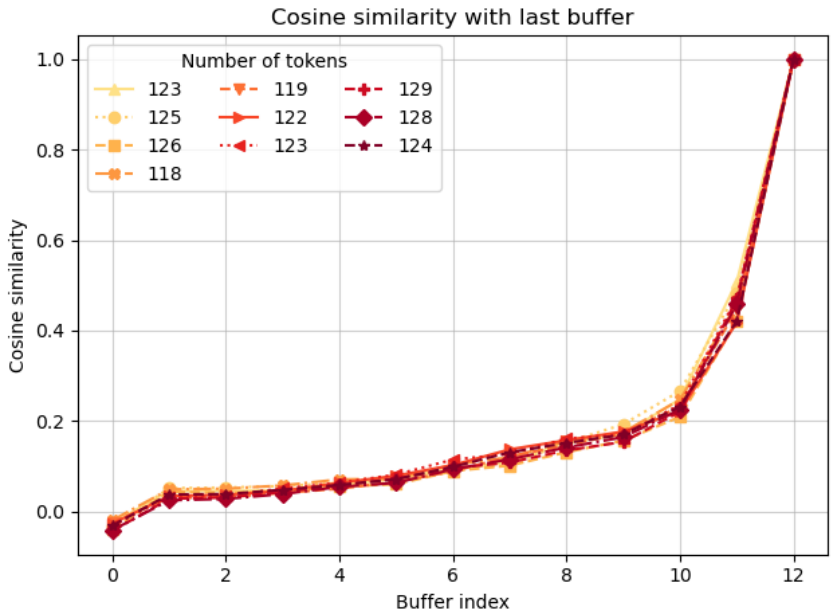
- 0 - Art
- 1 - Sport
- 2 - Physics
- 3 - Literature
- 4 - Philosophy
- 5 - Anatomy
- 6 - Politics
- 7 - Economy
- 8 - Cinema
- 9 - Mathematics

# log(Volume) & Cosine similarity



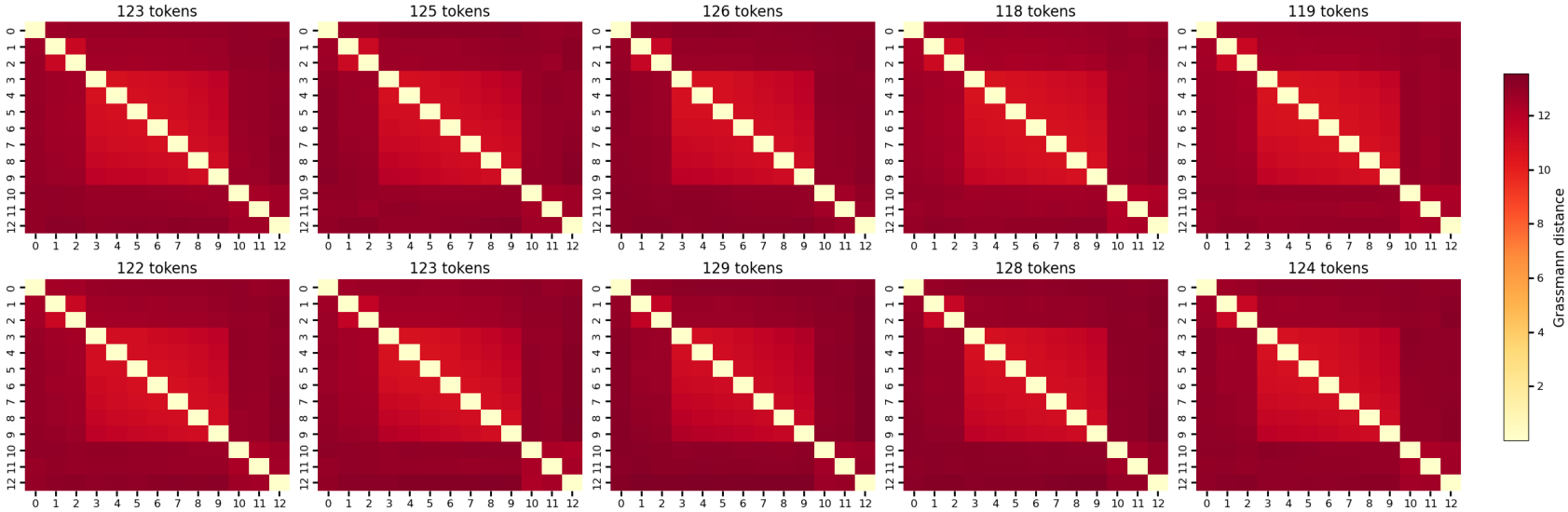
Again, the same behavior as before...



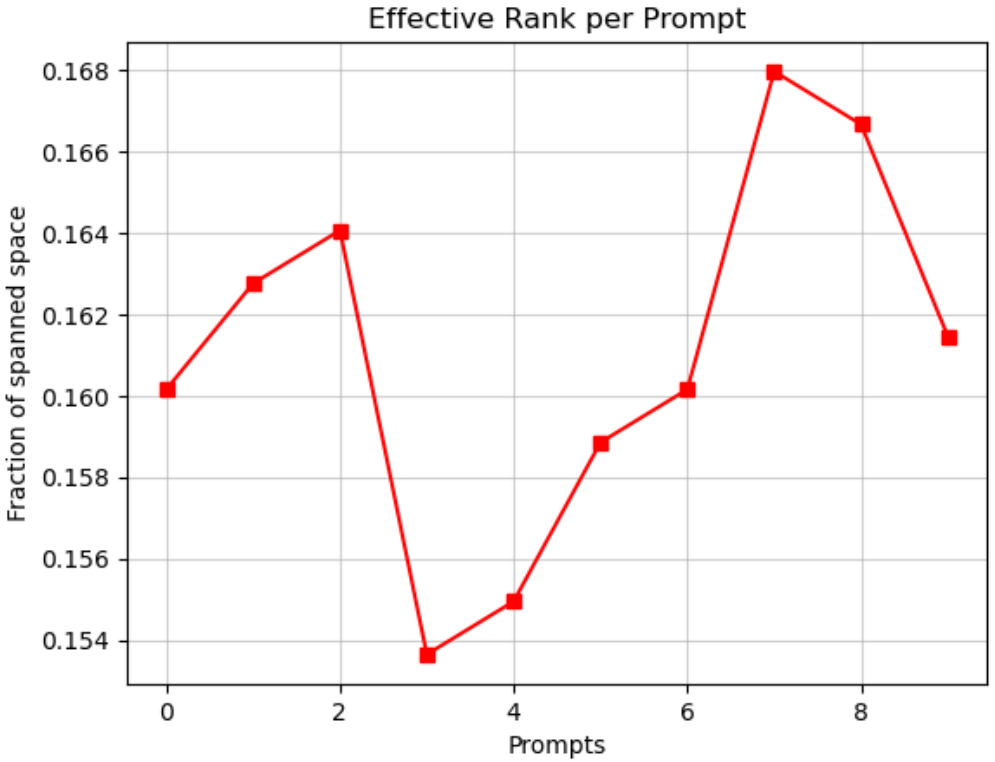
# Grassmann distance






- Despite varying contexts, the $d_G$ heatmaps appear remarkably similar
  $\rightarrow$ Indicates a consistent pattern of motion during the exploration phase.

# Explored space




- The fraction of explored space increases with the number of tokens;
- It is likely that even if the total explored space remains similar, the directions differ across prompts.

# Main Takeaways



1. The pattern of motion across decoders appears highly regular according to our metrics;
2. When $n = d$, the representational capacity of the embedding space reaches its limit;
3. Depending on the context, the fraction of explored space may remain the same, but different directions within the embedding space are likely used.

# Table of Contents

# Motivation: Introduction



From "buffer-space" to "**model-space**":

- How attention and MLP layers contribute to the geometrical transformation of the buffer

Spectral analysis of those weight matrices:

- **Specific directions** are used to **operate** within the transformer [1], [2].

Our objective is to formalize and visualize the **action** of the weight matrices in the **Attention** and **MLP** layers.

# Motivation: Looking for Linearity



```
def attn(x, ...):   # ATTENTION layer
    ...
    c = conv1d(x, 'c_attn', n_state*3)  # projection to q, k, v matrices
    q, k, v = map(split_heads, tf.split(c, 3, axis=2))
    ...
    a = multihead_attn(q, k, v)
    a = merge_heads(a)
    a = conv1d(a, 'c_proj', n_state)
    return a, present

def mlp(x, ...):    # MLP layer
    ...
    h = gelu(conv1d(x, 'c_fc', n_state))    # up-projection
    h2 = conv1d(h, 'c_proj', nx)    # down-projection
    return h2
```

# Motivation: Looking for Linearity



**Weights are linear maps** that act on a buffer-like object
$X_{in} \mapsto WX_{in}$

```
def conv1d(x, ...):     # Linear Operation X' = WX + b
    ...
    w = tf.get_variable('w', [1, nx, nf], ...)
    b = tf.get_variable('b', [nf], ...)
    c = tf.reshape(
        tf.matmul(
            tf.reshape(x, [-1, nx]),
            tf.reshape(w, [-1, nf])
        ) + b,
        start + [nf])
    return c
```

# Theoretical Framework



1. From a **set of prompts** we compute the covariance matrix of the actions $Y \equiv V^T X_{in}$, where $V^T$ comes from the SVD of $W$.
2. Eigen-decomposition gives the **principal directions** of $Y$, namely the matrix of eigenvectors $\mathcal{F}_Y$.
3. We show that the eigenvectors in $\mathcal{F}_Y$ are the **projections** of the principal directions of all data points of the **set of buffers** on the principal directions of $W$.
4. We claim that **significant action** shows for spikes in the values for those projections.
5. To assess statistical validity we compute the Marchenko-Pastur distribution (MP) of the singular values of $W$. We expect to find deviations, i.e. **outliers** from MP bounds, for **spikes in the projections** indicating **non-random action** of $W$.

# Theory: The SVD of the Weight Matrix



The linear transformation $W$ can be decomposed in a **rotation**, **scaling** and a second **rotation "back"**.
For a matrix $W \in \mathbb{R}^{m_\ell \times n_\ell}$.
SVD given by $W = U\Sigma V^T$, where

- $V \in \mathbb{R}^{n_\ell \times n_\ell}$ has columns $\{v_k\}_{k=1}^{n_\ell}$, the right singular vectors.
- $\Sigma = \mathrm{diag}(s_1 \geq s_2 \geq \cdots \geq s_{r_\ell})$ are the singular values.
- $U \in \mathbb{R}^{m_\ell \times m_\ell}$ has the left singular vectors.
- $r_\ell = \min(m_\ell, n_\ell)$ is the numerical rank.

The **principal directions** used by the Transformer to **operate** are the **singular vectors** of the weight matrix.

# Theory: The Activation Covariance Matrix



Set of $p$ buffer matrices $X \in \mathbb{R}^{n_\ell \times m}$, each of $m$ tokens and $n_\ell$ buffer-space dimensionality.
Run the prompt batches and extract the input and to each sub-layer and its weight.

| **Sublayer** | **Module** | **Input dim** $n_\ell$ | **Output dim** $m_\ell$ |
|---|---|---|---|
| Query (q) | `attn.c_attn[:, :d]` | $n_q = d$ | $m_q = d$ |
| Key (k) | `attn.c_attn[:, d:2*d]` | $n_k = d$ | $m_k = d$ |
| Value (v) | `attn.c_attn[:, 2*d:3*d]` | $n_v = d$ | $m_v = d$ |
| Attn-out (a) | `attn.c_proj` | $n_a = d$ | $m_a = d$ |
| MLP-up (u) | `mlp.c_fc` | $n_u = d$ | $m_u = 4d$ |
| MLP-down (d) | `mlp.c_proj` | $n_d = 4d$ | $m_d = d$ |

**Table:** Mapping of GPT-2's sub-layer and their input/output dimensions.

# Theory: The Activation Covariance Matrix



The Activation Covariance Matrix (ACM) elements:

$$F_{ab} = \frac{1}{p \cdot m} \sum_{i}^{p} \sum_{t}^{m} (X_{i,t,a} - \bar{X}_a)(X_{i,t,b} - \bar{X}_b)$$

with $a, b = 1, ..., n$ the **embedding dimension** and $\bar{X}_a$ is the **sample mean activation** of the buffer-space dimension $a$.

- $F_{aa}$ entries are the empirical variances over buffer-space dimensions $a$.
- $F_{ab}$ are the empirical covariances (correlations) across dimensions a and b

$F \in \mathbb{R}^{n_\ell \times n_\ell}$ captures how the buffer's embedding-dimensions **co-vary** across all prompts.

# Theory: Eigen-decomposition of the ACM



- Data points: the set of buffer vectors $x_{i,t}^{(\ell)} \in \mathbb{R}^{n_\ell}, \quad i = 1, \ldots, p, \quad t = 1, \ldots, m$ , i.e. one $n_\ell$-dimensional vector in buffer-embedding space for every token of every prompt.
- **Cloud** in $\mathbb{R}^{n_\ell}$: the full collection of data points gives us a set of $N = p \cdot m$ points in $\mathbb{R}^{n_\ell}$.
- Eigen-decomposition of the ACM and principal directions: Eigenvalues $\lambda_1 \geq \cdots \geq \lambda_{n_\ell} \geq 0$ are the variances along each **principal axis** $f_1, \ldots, f_{n_\ell}$, that best describe variability of the cloud of data points.

# Theory: Change-of-Basis Perspective



We define the Action as the transformation:

$$Y \equiv V^T X, \in \mathbb{R}^{n_\ell \times m}.$$

For each $k = 1, \ldots, n_\ell$, we ask how much the $k$-th (action) axis (i.e. $v_k$) **aligns** with the buffer samples principal directions, i.e. the **set of ACM eigenvectors** $\{f_X^{(j)}\}_{j=1}^{n_\ell}$.

# Theory: Change-of-Basis Perspective



- The action $Y = V^T X$
- In $Y$-coordinates, the ACM is $F_Y = \mathrm{COV}(Y) = V^T F V$, a similarity of $F$.
- **Eigen-pairs carry over** resulting in $\boxed{\mathcal{F}_Y = V^T \mathcal{F}_X}$ with $\mathcal{F}_X$ the matrix of eigenvectors $\{f_X^{(j)}\}_{j=1}^{n_\ell}$ of the buffer samples cloud.
- The eigenvectors of $F_Y$ are the **projection of buffer samples principal directions on the space spanned by the right singular vectors** of $W$. We call $\mathcal{F}_Y$ the Projection Matrix (PM)

Explicit PM:

$$\mathcal{F}_Y = V^T \mathcal{F}_X = \begin{pmatrix} \vec{v}_1^T \vec{f}_X^{(1)} & \vec{v}_1^T \vec{f}_X^{(2)} & \cdots & \vec{v}_1^T \vec{f}_X^{(n)} \\ \vdots & & & \vdots \\ \vec{v}_n^T \vec{f}_X^{(1)} & \vec{v}_n^T \vec{f}_X^{(2)} & \cdots & \vec{v}_n^T \vec{f}_X^{(n)} \end{pmatrix} = \begin{pmatrix} \vec{O}_1^T \\ \vec{O}_2^T \\ \vdots \\ \vec{O}_n^T \end{pmatrix},$$

with the **overlap** vectors $\vec{O}_k$ quantifying the projection.
Finally, we define the overlap as:

$$\boxed{O_k = \max_{1 \leq j \leq n_\ell} \left| \langle v_k, f_X^{(j)} \rangle \right|,}$$

- $O_k \approx 1$: **one activation-axis $f_X^{(j)}$ lies essentially on $v_k$.**
- $O_k \approx 0$: $v_k$ is almost orthogonal to all principal data-axes, i.e. an unused direction.

# Theory: The Projection Matrix



- SVD of $W$: 1. rotation (action), 2. **scaling** (singular values) and 3. second rotation "back".
- The Transformer uses the principal directions $v_k$ to operate on the buffer by a process of **projection** (on the principal directions in **model-space**) and re-scaling (**expansion** / **contraction**) before projecting back to the buffer-space.

# Theory: Marchenko-Pastur Law



Separate bulk (random-like) from **signal** (encoded structure):

- Need a **null model** for the distribution of the singular-values if $W$ were isotropic noise in $\mathbb{R}^{n_\ell \times m_\ell}$.
- Singular values that **lie outside** the theoretical MP bulk can be flagged as outliers, i.e. learned, data-driven directions (significant action).

# Theory: Marchenko-Pastur Law



Given our weight matrices, the Marchenko–Pastur distribution is:

$$p_s(s) = \frac{1}{\pi\, \sigma_\ell^2\, q_\ell\, s_\ell} \sqrt{(s_-^2 - s_\ell^2)\,(s_\ell^2 - s_+^2)},$$

supported on the range of singular values $s_\ell \in [s_-, s_+]$, where $q = \frac{m_\ell}{n_\ell}$.

- Any $s_k \in [s_-, s_+]$ is **indistinguishable** from what a random matrix of the same size and variance would produce
- Any $s_k > s_+$ correspond to directions that $W$ amplifies more strongly than chance (**learned feature axes**).
- Any $s_k < s_-$ is a direction that $W$ that the model has learned to **attenuate** (redundancy).

# Results

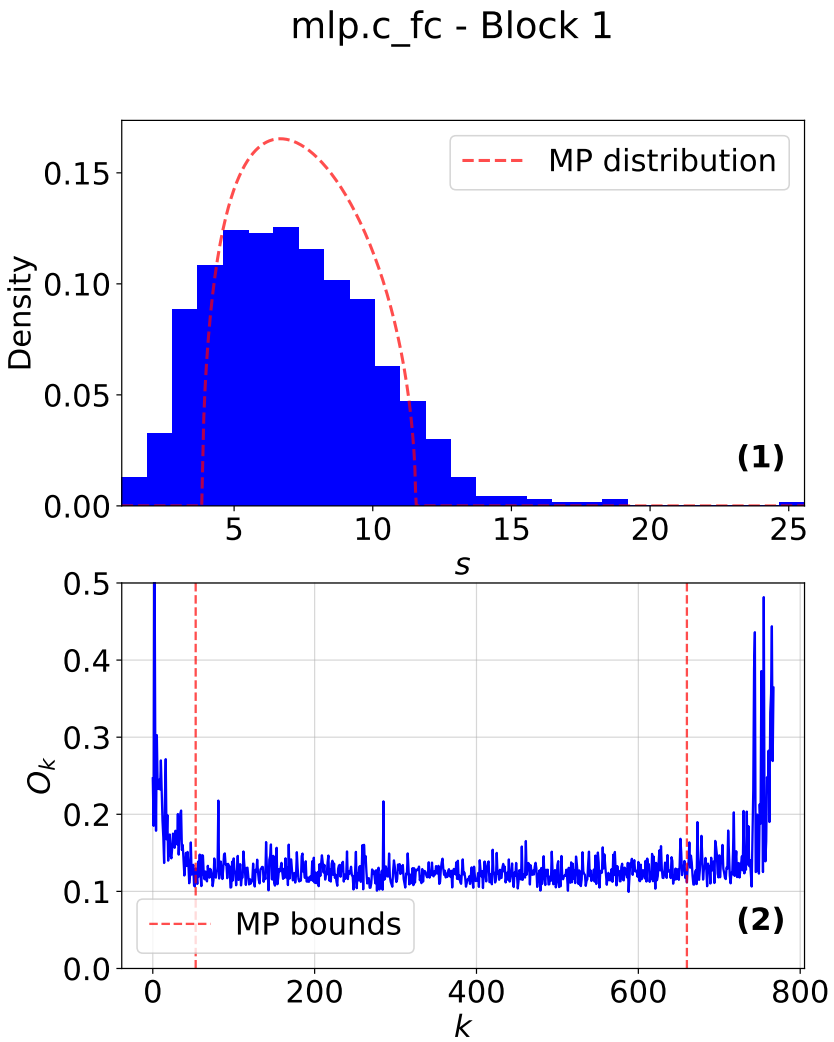


We performed the analysis using two sets of different prompts:

- `random`: 100 prompts composed each of 100 tokens sampled at random from the GPT2 internal dictionary
- `eco`: 100 prompts composed each of 100 tokens taken from an Umberto Eco's essay

In this way, we can assess whether the semantical structure of the prompts influences the way the model interacts with the buffer, considering the randomly generated prompts as a sort of "null model".

# Results: How to read the plots

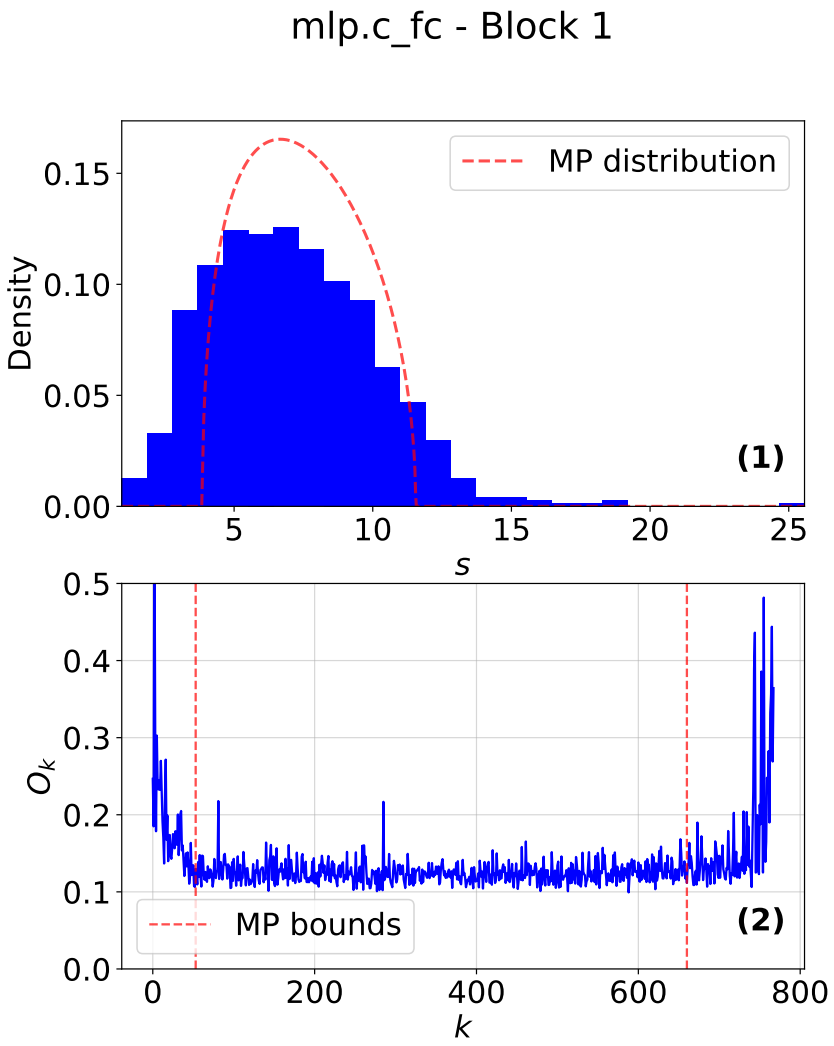



We have produced two different types of plot:

1 - an histogram visualization of the distribution of the singular values of a given weight matrix

2 - a line plot that represents the overlap $O_k$ for $k = 1, \ldots, n_\ell$

# Results: How to read the plots

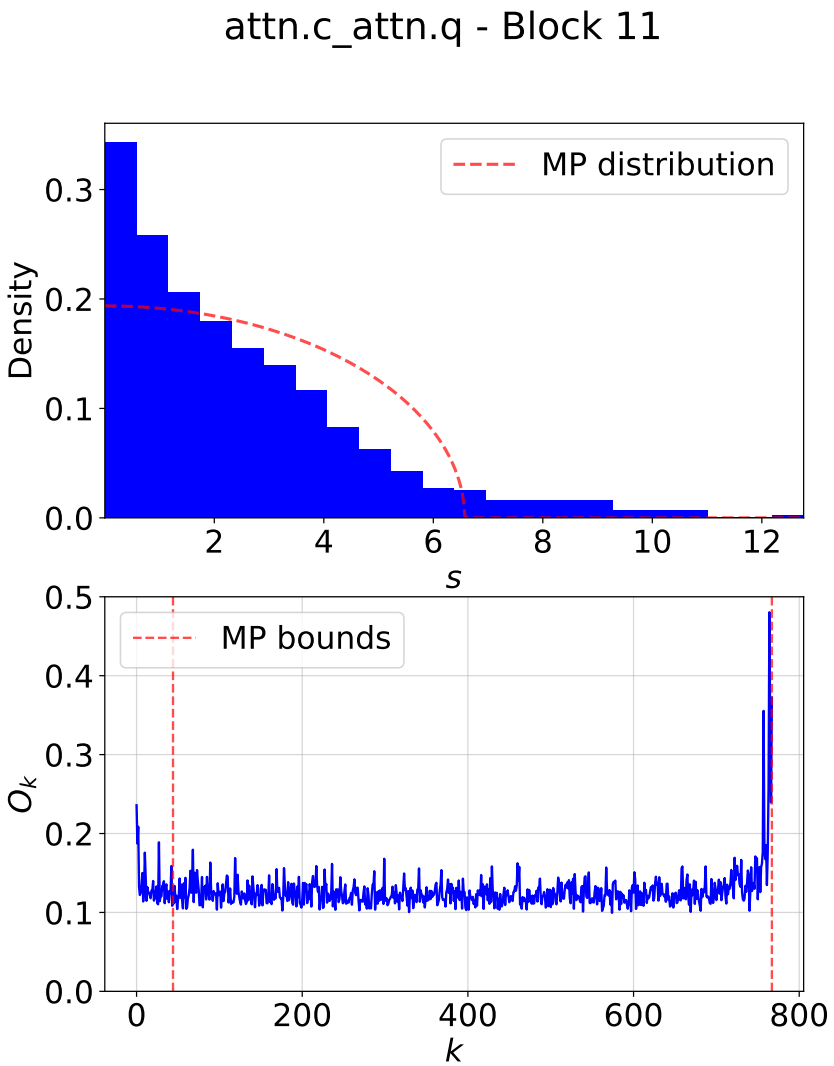



In both of these plots we highlighted the Marchenko-Pastur distribution. In particular:

1 - we plotted the MP distribution corresponding to a random matrix with the same dimentions as the weight matrix

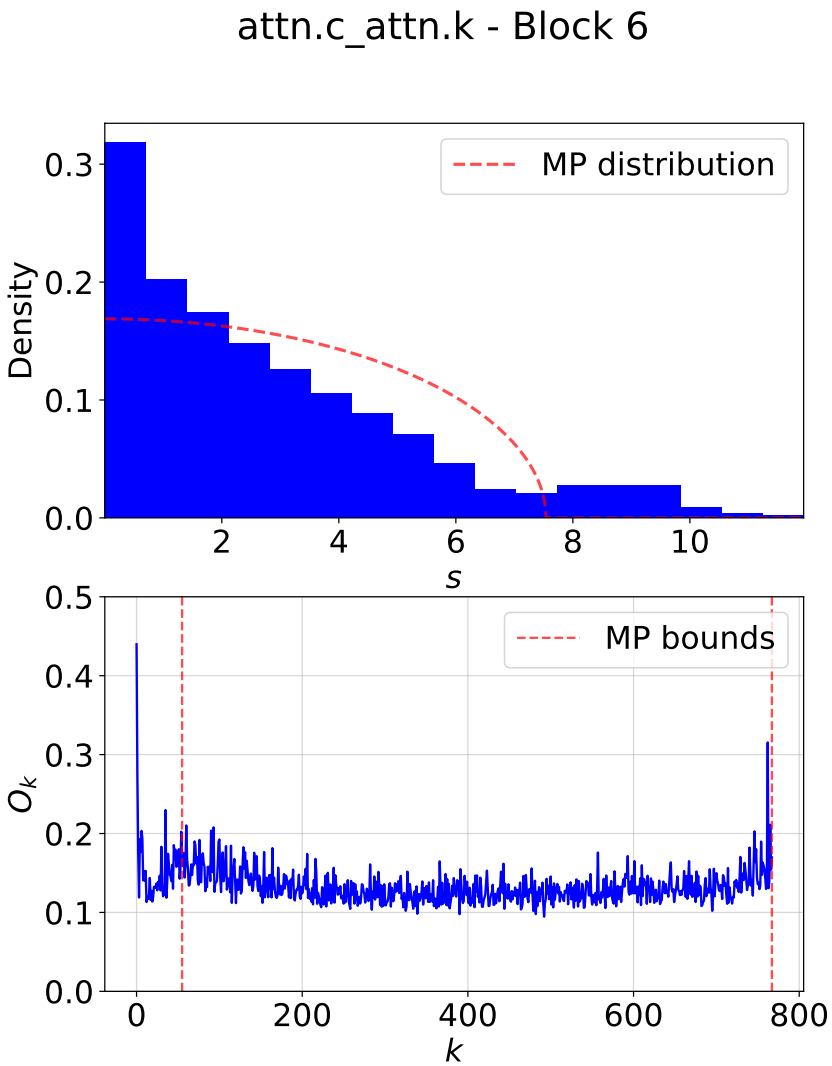
2 - we highlighted the right singular vector indices for which the corresponding singular values are just outside of the MP bounds (i.e. $s_-$ and $s_+$)

# Results: Attention layer



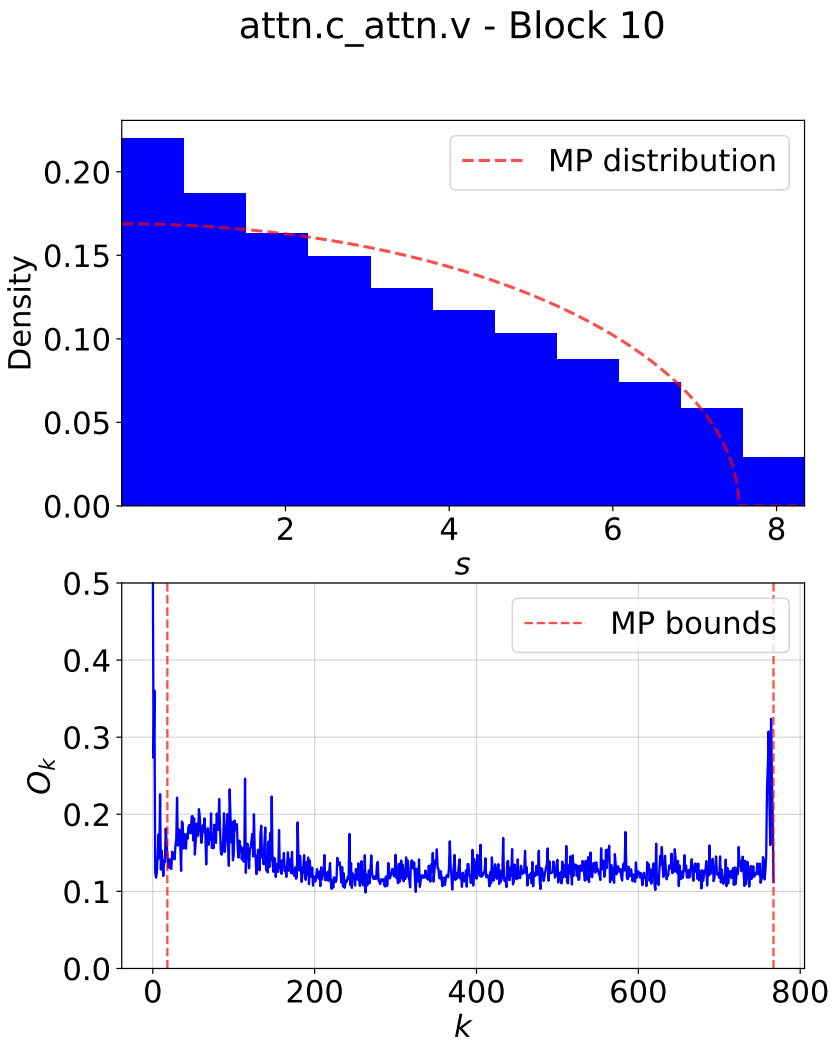

- Generally, we observe higher than average overlap only outside, or close to the bounds, of the Marchenko-Pastur distribution
- $Q$ displays these overlap peaks in the small singular value region, while $K$ and $V$ also in the large singular value region

# Results: Attention layer



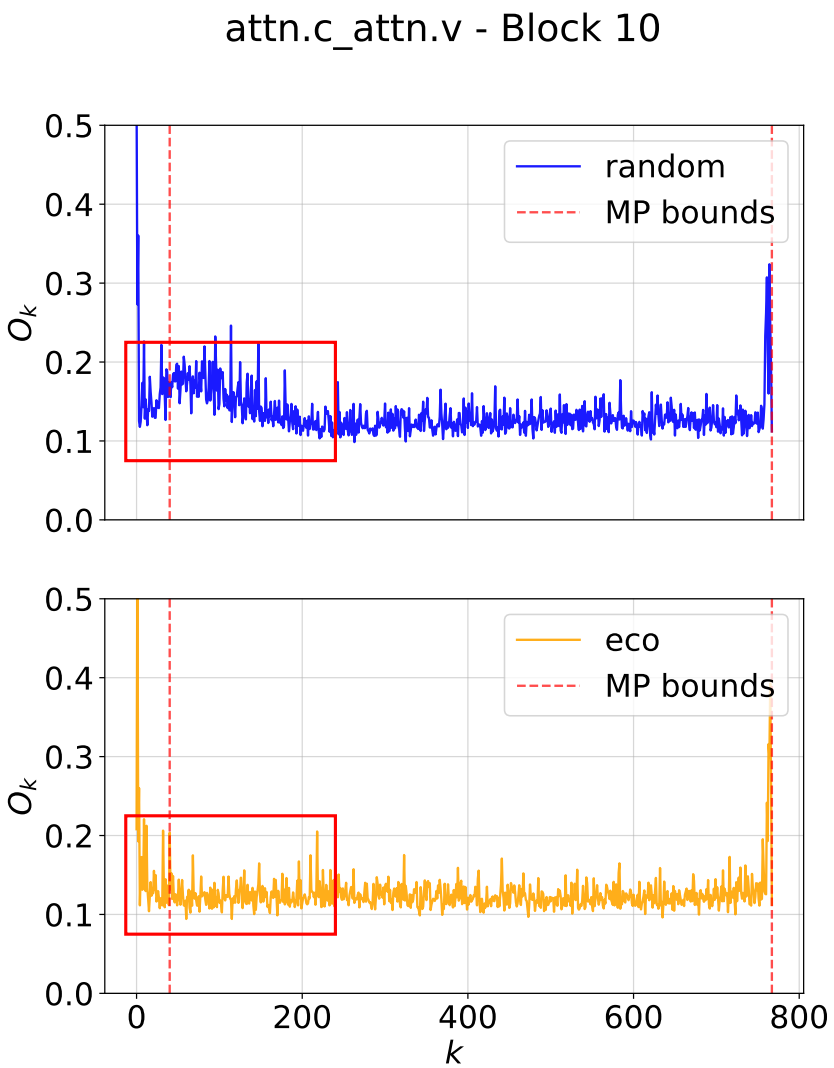

- Generally, we observe higher than average overlap only outside, or close to the bounds, of the Marchenko-Pastur distribution
- $Q$ displays these overlap peaks in the small singular value region, while $K$ and $V$ also in the large singular value region

# Results: Attention layer




- Generally, we observe higher than average overlap only outside, or close to the bounds, of the Marchenko-Pastur distribution
- $Q$ displays these overlap peaks in the small singular value region, while $K$ and $V$ also in the large singular value region

# Results: Attention layer




- For the `random` prompt set, we observe a small increment in the overlap values, with respect to the average, when looking closely at the $s_+$ bound
- This phenomenon disappears when considering the `eco` prompt set

# Results: Attention layer

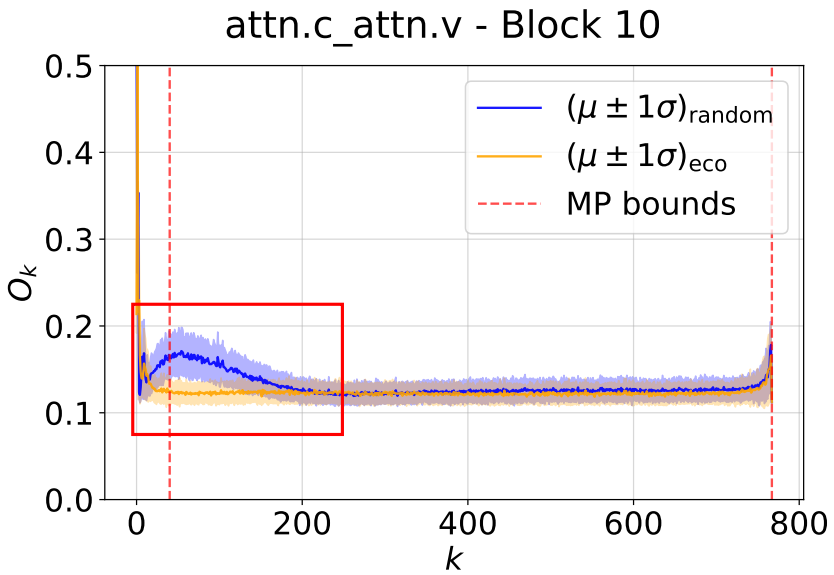
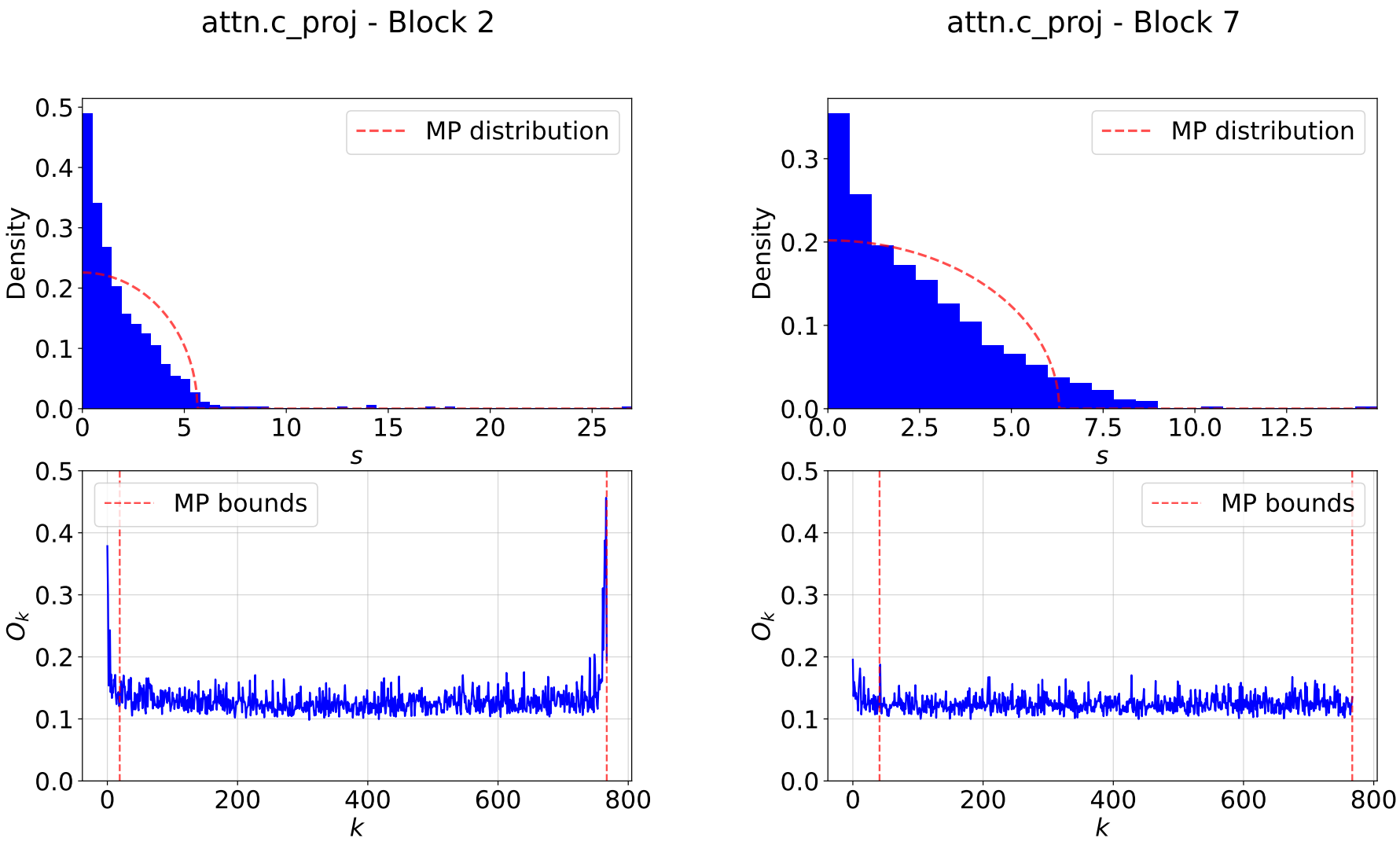


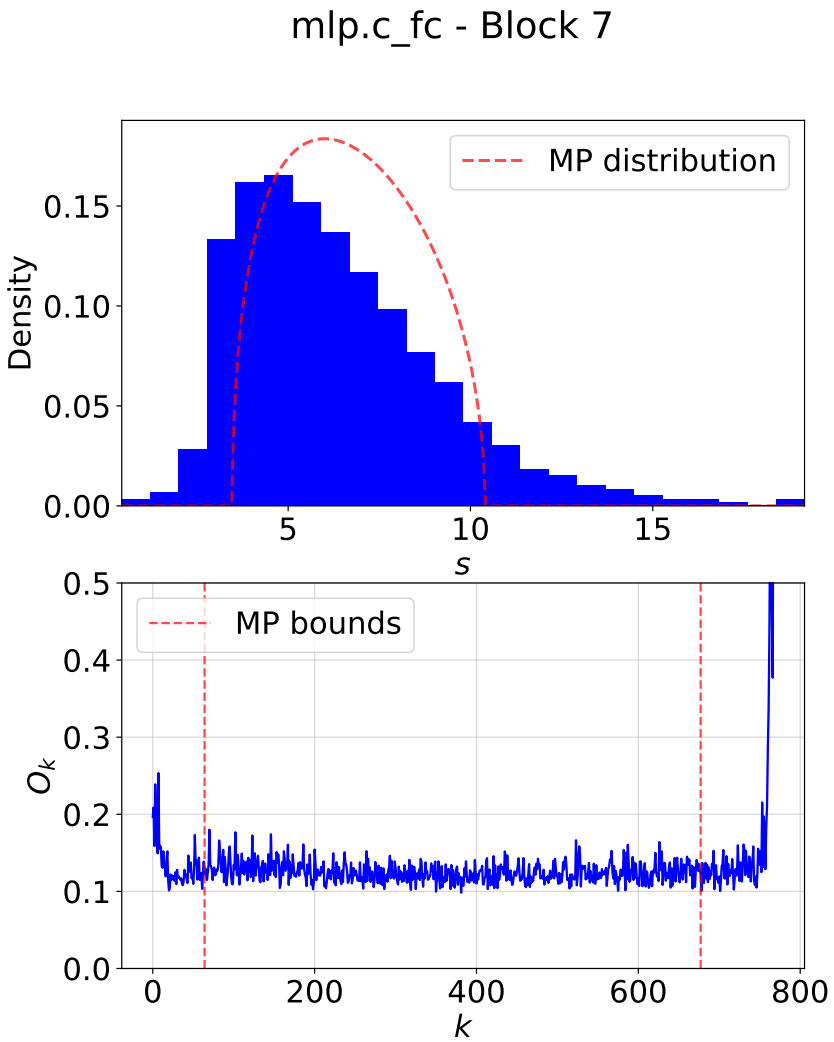

Figure: Average overlaps over 100 realizations

- For the `random` prompt set, we observe a small increment in the overlap values, with respect to the average, when looking closely at the $s_+$ bound
- This phenomenon disappears when considering the `eco` prompt set

# Results: Attention layer



# Results: MLP layer




- The MLP-up projection displays large overlaps in the small singular values region
- Some blocks display differences in the overlaps when comparing the `random` and the eco sets, similarly to the Attention layer

# Results: MLP layer

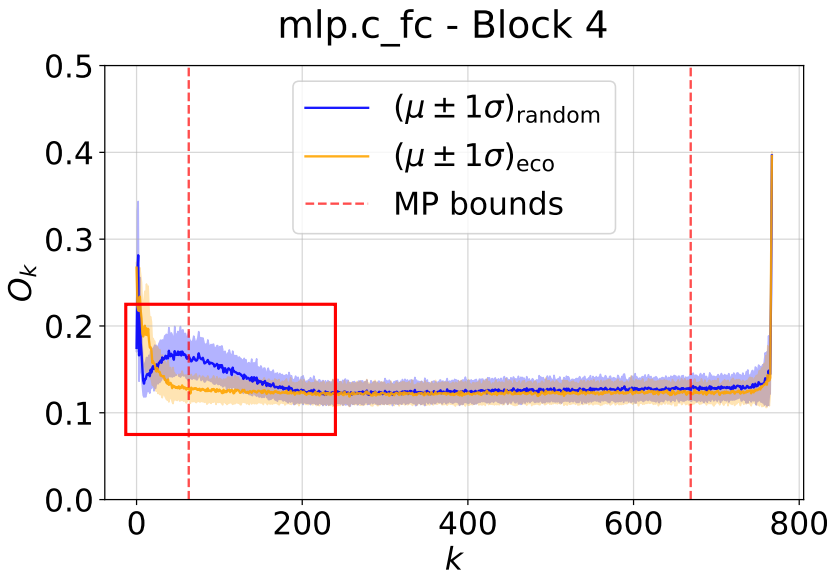
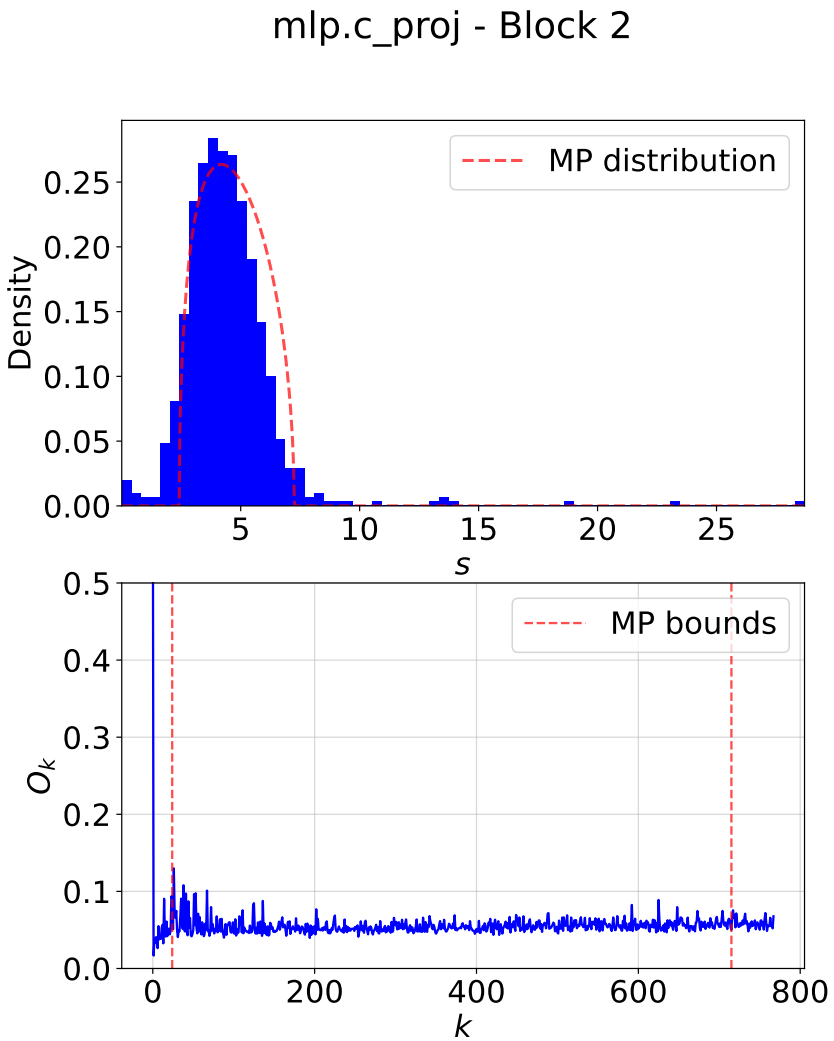



Figure: Average overlaps over 100 realizations

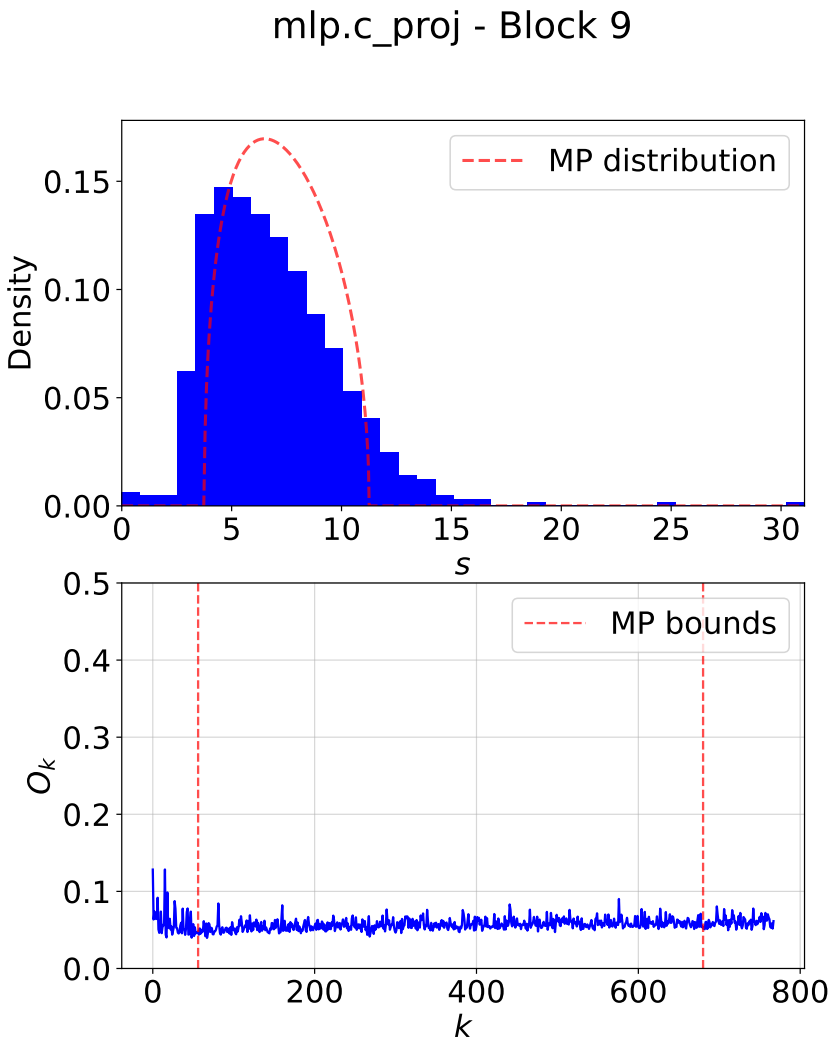
- The MLP-up projection displays large overlaps in the small singular values region
- Some blocks display differences in the overlaps when comparing the `random` and the `eco` sets, similarly to the Attention layer

# Results: MLP layer



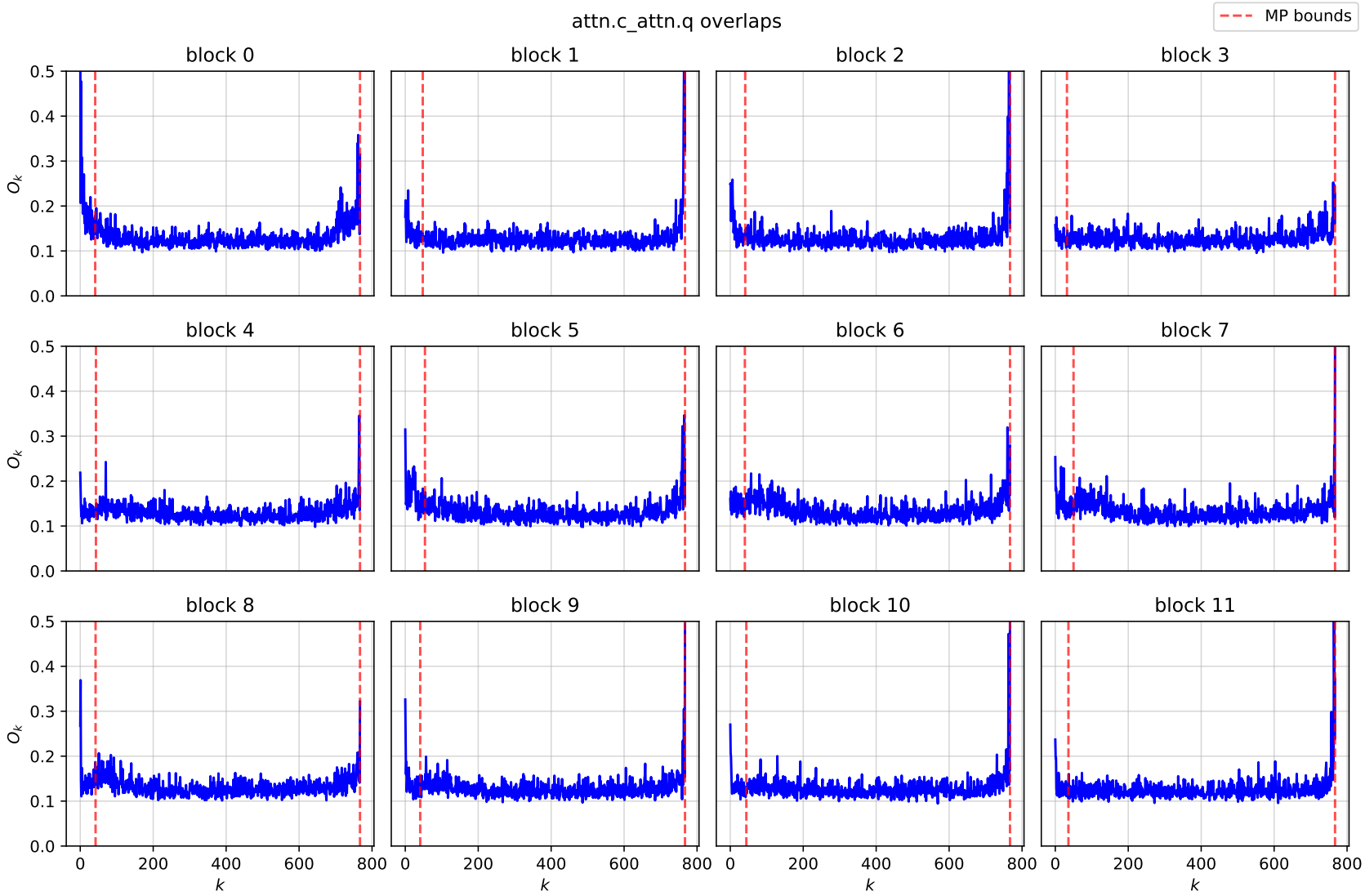

- The MLP-down projection displays lower average overlap and also shows the peak overlap decreasing as the blocks progress

# Results: MLP layer

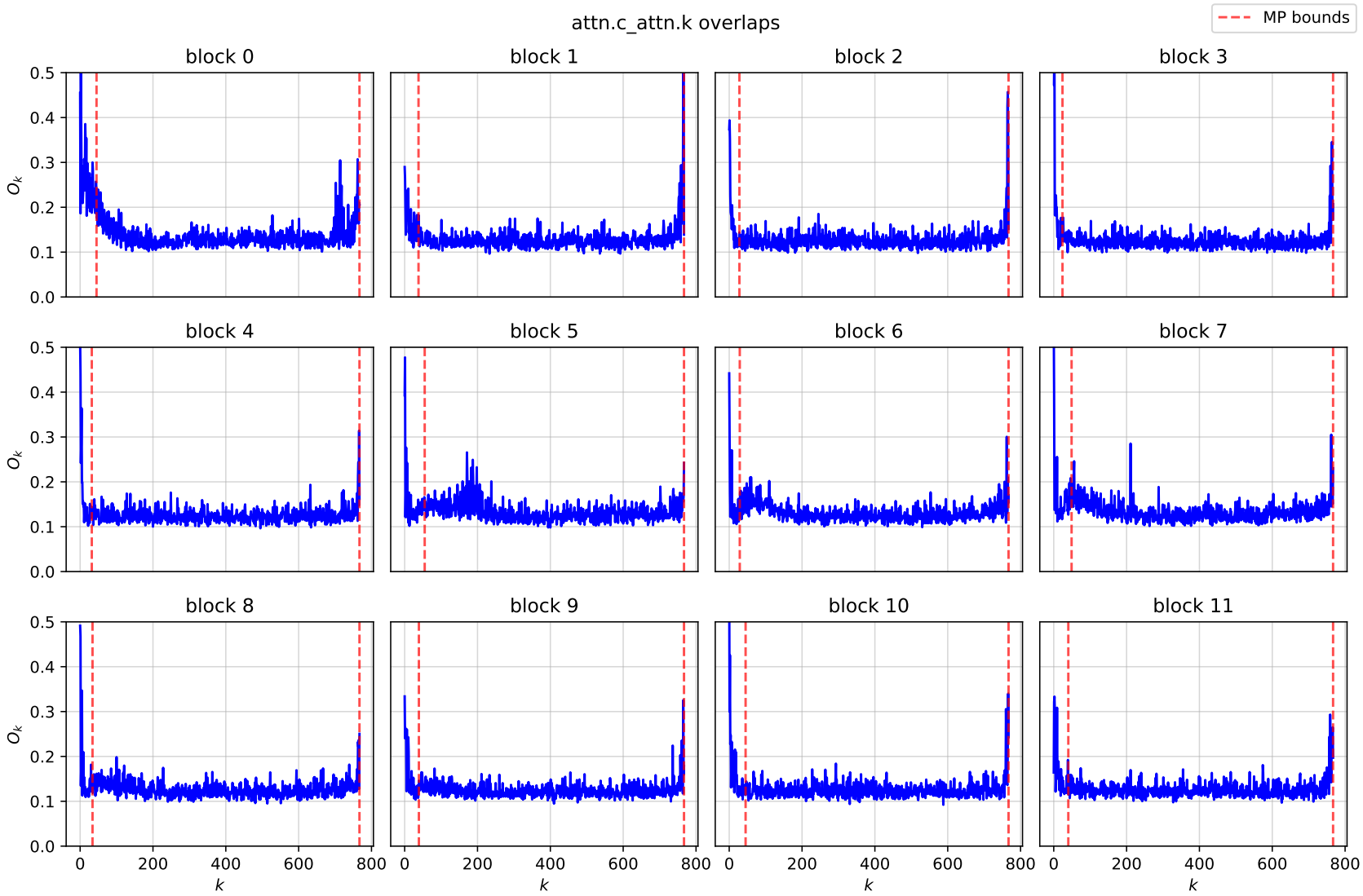



- The MLP-down projection displays lower average overlap and also shows the peak overlap decreasing as the blocks progress

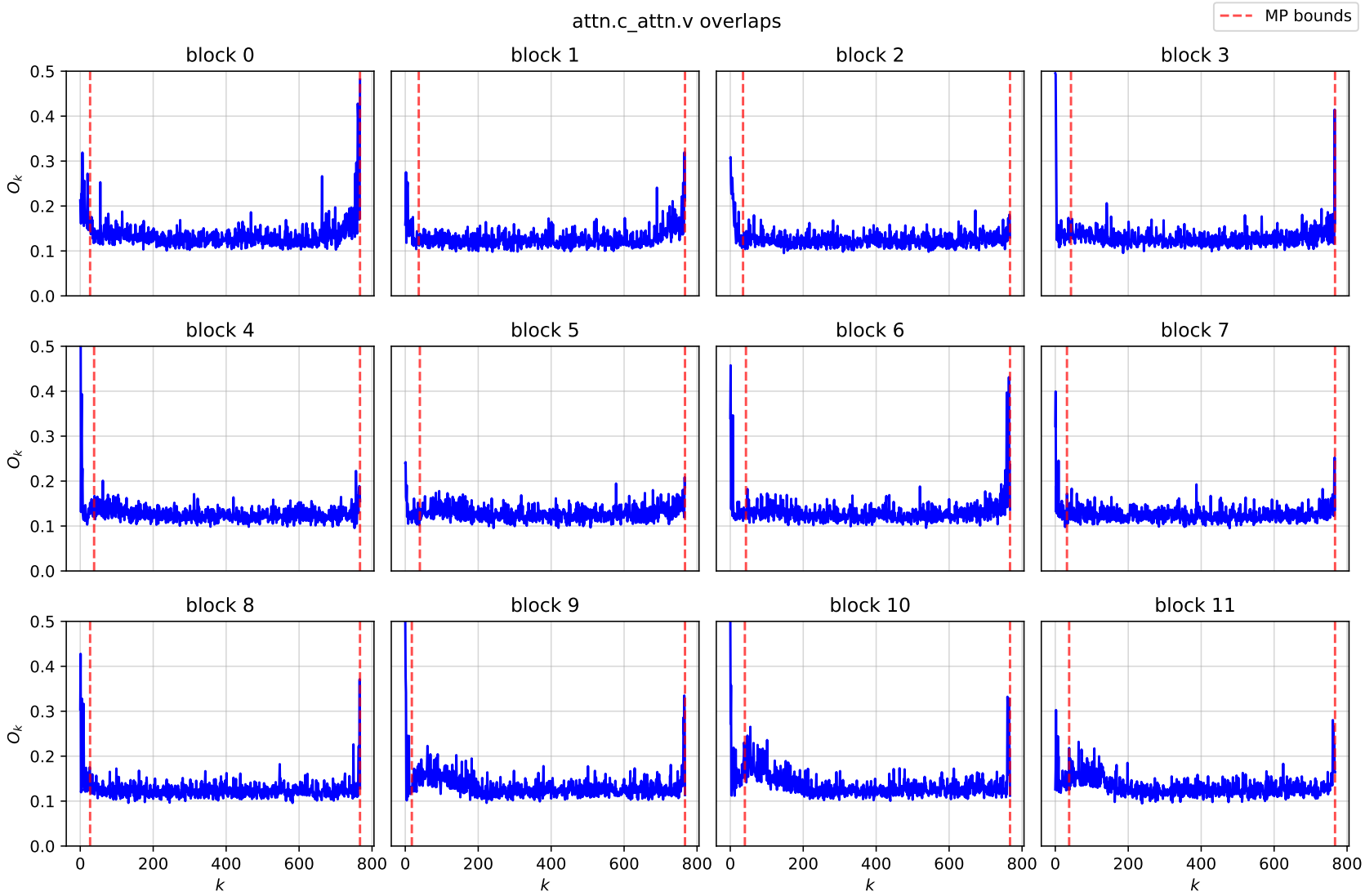
# Results: `random` overlaps



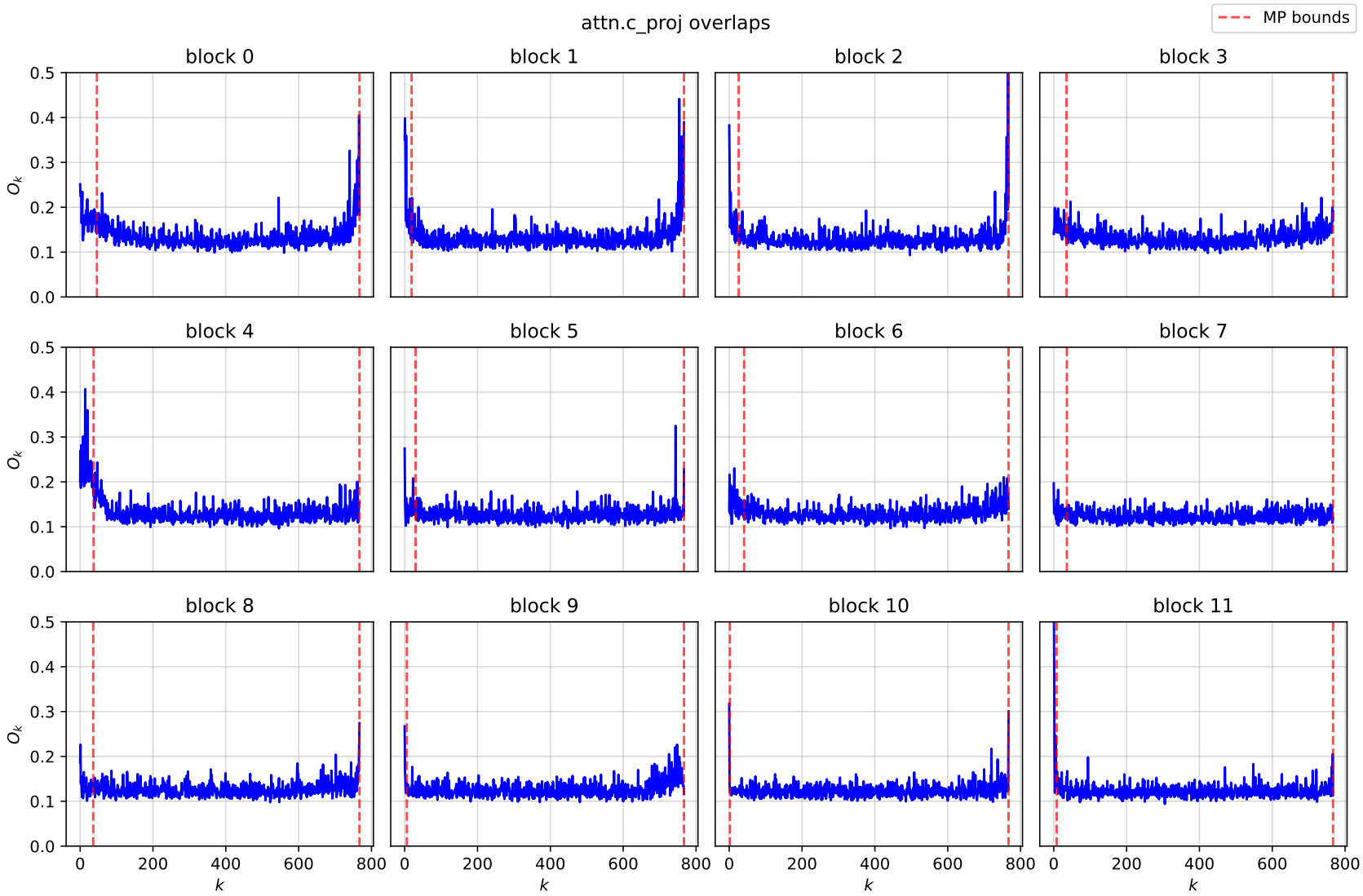
# Results: `random` overlaps



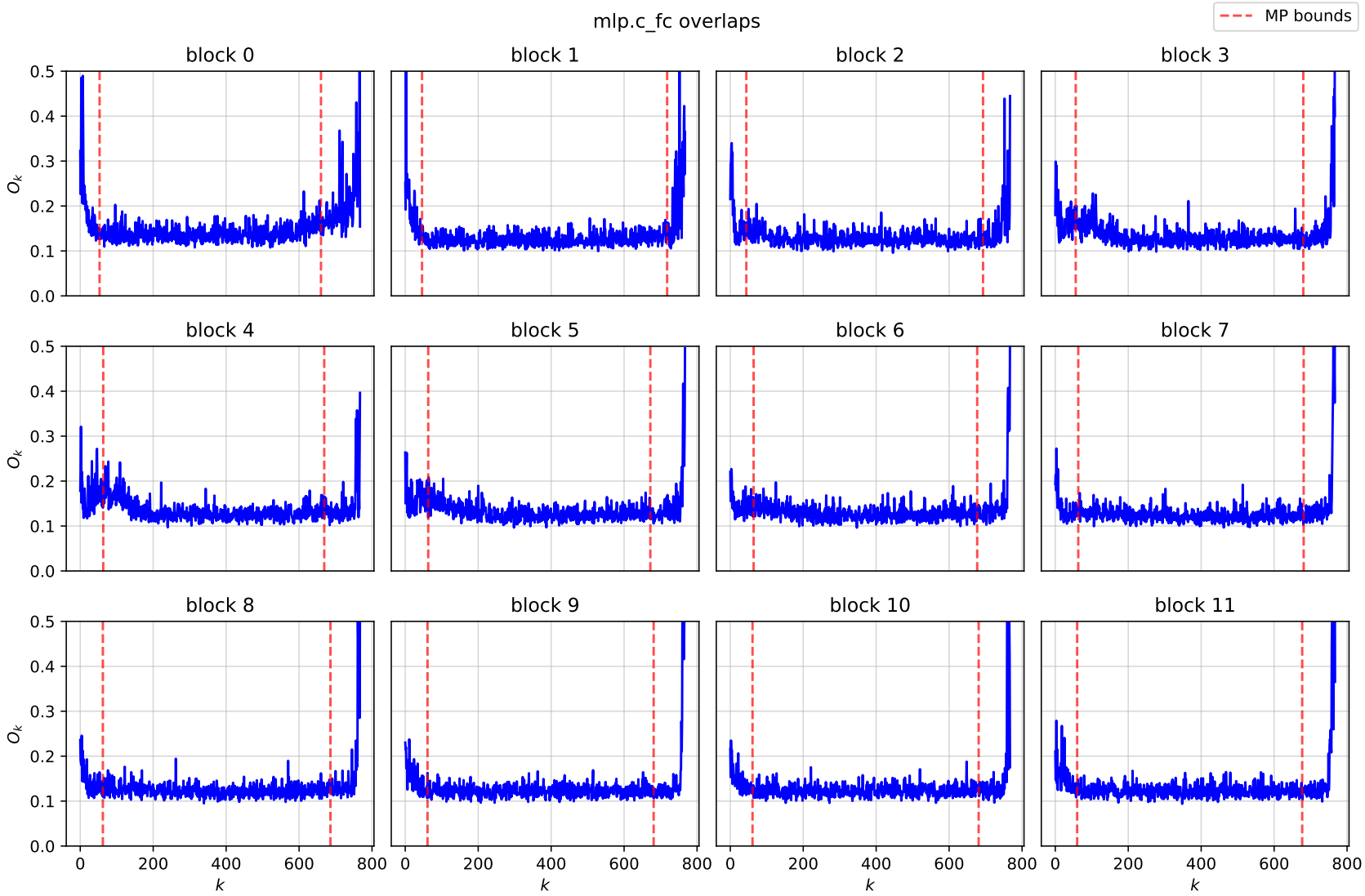
# Results: `random` overlaps



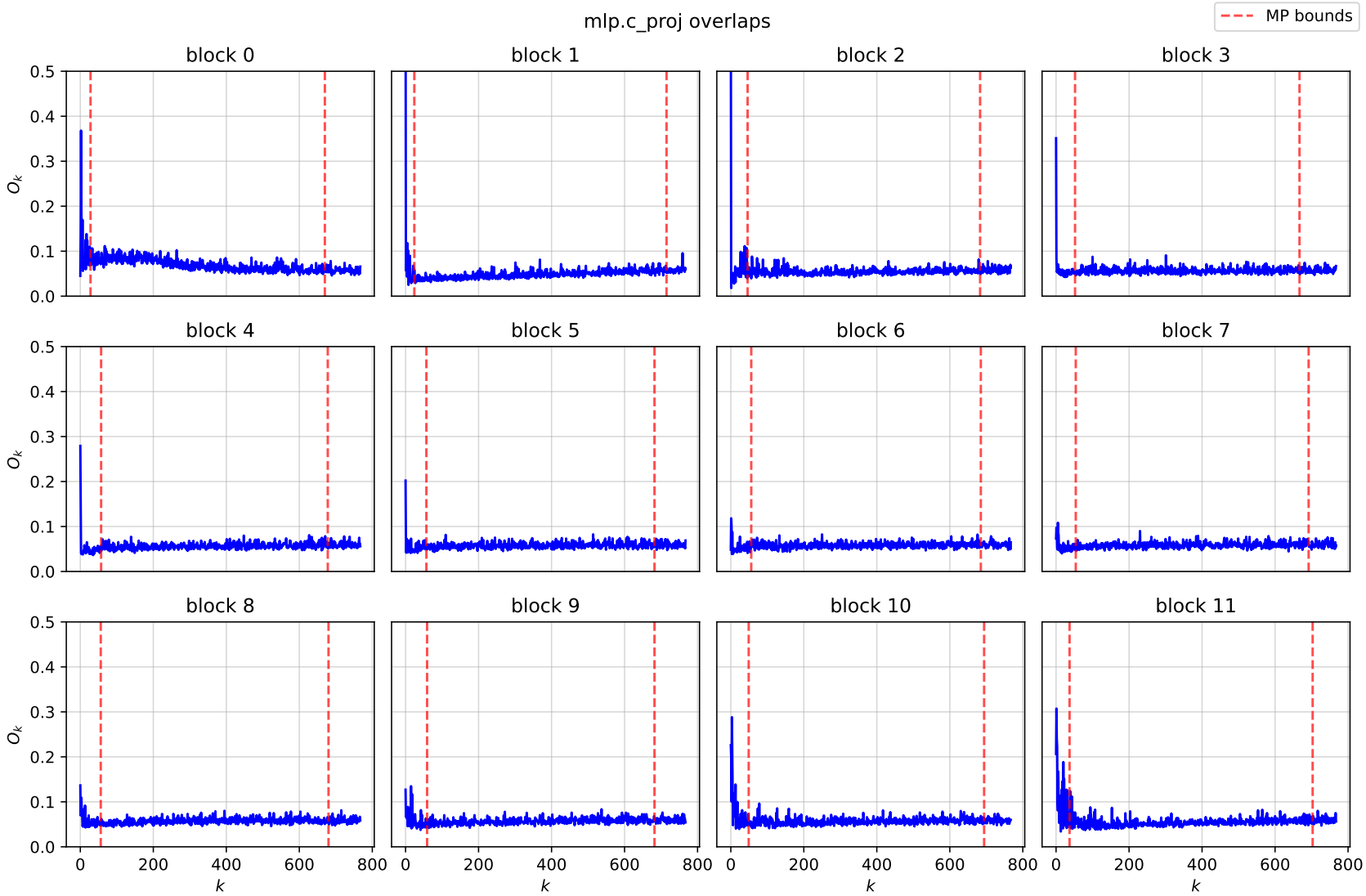
# Results: `random` overlaps



# Results: `random` overlaps



# Results: `random` overlaps



# References



[1] Beren Millidge and Sid Black. *The Singular Value Decompositions of Transformer Weight Matrices are Highly Interpretable*. Accessed: 2025-07-24. Nov. 2022. URL: https://www.alignmentforum.org/posts/mkbGjzxD8d8XqKHzA/the-singular-value-decompositions-of-transformer-weight.

[2] Max Staats, Matthias Thamm, and Bern Rosewon. "Small Singular Values Matter: A Random Matrix Analysis of Transformer Models". In: *arXiv preprint arXiv:2401.17770* (Feb. 2025). Preliminary work. Under review by the International Conference on Machine Learning (ICML).

Thank you!

# Appendix - Math. results (1)



## Definition

Given a matrix $A \in \mathbb{R}^{m \times n}$ with $m \geq n$, the **reduced QR decomposition** (or thin QR) factors $A$ as:

$$A = QR$$

where:

- $Q \in \mathbb{R}^{m \times n}$ has orthonormal columns ($Q^\top Q = I_n$),
- $R \in \mathbb{R}^{n \times n}$ is upper triangular.

# Appendix - Math. results (2)



## Definition

Given two matrices $A \in \mathbb{R}^{n \times d}$ and $B \in \mathbb{R}^{n \times d}$, whose columns represent orthonormal bases of two $d$-dimensional subspaces in $\mathbb{R}^n$, the **Cross-Gram matrix** is defined as:

$$G = A^\top B \in \mathbb{R}^{d \times d}.$$

**Properties:**

- Entries of $G$ represent inner products between the basis vectors of $A$ and $B$.
- If $A$ and $B$ are orthonormal, $G$ captures the relative orientation of their subspaces.
- The singular values of $G$ equal the **cosines of the principal angles** between the subspaces.

# Appendix - Math. results (3)



Given a data matrix $A \in \mathbb{R}^{m \times n}$, the **Singular Value Decomposition (SVD)** factors it as: $A = U\Sigma V^\top$, where:

- $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices,
- $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$.

## Low-Rank Approximation

By keeping only the top $k$ singular values (and corresponding vectors), we get:

$$A_k = U_k \Sigma_k V_k^\top,$$

Choose $k$ by thresholding the singular values to retain a desired variance percentage (e.g., 90%). This reduces dimensionality while preserving most of the data's structure.

# Appendix - MP Law Derivation for SVD (1)



Let $X \in^{m \times n}$ have i.i.d. entries with zero mean and variance $\sigma^2$. One can form the (scaled) sample covariance

$$C = \frac{1}{n} X X^T \in^{m \times m}.$$

As $m, n \to \infty$ with the ratio $q = \frac{m}{n}$ $(0 < q \leq 1)$ fixed, the empirical eigenvalue distribution of $C$ converges to the Marchenko–Pastur law with support $\lambda_{\pm}^{(\mathrm{cov})} = \sigma^2(1 \pm \sqrt{q})^2$, meaning that nearly all eigenvalues of $C$ lie in

$$[\sigma^2(1 - \sqrt{q})^2,\ \sigma^2(1 + \sqrt{q})^2].$$

# Appendix - MP Law Derivation for SVD (2)



The nonzero singular values of $X$ are the square-roots of the nonzero eigenvalues of $XX^T$, i.e., let $\{\lambda_i\}$ be the eigenvalues of $C = \frac{1}{n} XX^T$,then the corresponding singular values of $X$ are $s_i(X) = \sqrt{n\,\lambda_i}$ . Thus the support of the singular-value distribution of $X$ is

$$s_\pm = \sqrt{n\,\lambda_\pm^{(\mathrm{cov})}} = \sqrt{n\,\sigma^2(1 \pm \sqrt{q})^2} = \sigma\,\sqrt{n}\,(1 \pm \sqrt{q}).$$

Center $W$ and compute $\sigma^2 = \frac{1}{mn}\sum_{i,j} W_{ij}^2$ as the empirical variance of $W_{centered}$. Set

$$s_- = \sigma\left|\sqrt{n} - \sqrt{m}\right|, \quad s_+ = \sigma(\sqrt{n} + \sqrt{m}),$$

then any empirical singular values $s_k$ outside $[s_-, s_+]$ will be outliers relative to the random baseline for that weight matrix $W$.

# Appendix - MP Law Derivation for SVD (3)



Given that the (nonzero) eigenvalues $\{\lambda_i\}$ of $C$ in the large-$m, n$ limit have probability density function

$$p_C(\lambda) = \frac{1}{2\pi\,\sigma^2\,q\,\lambda}\sqrt{(\lambda_+^{(\mathrm{cov})} - \lambda)\,(\lambda - \lambda_-^{(\mathrm{cov})})},$$

and that the nonzero singular values $s_i$ of $W_{centered}$ relate by

$$s_i = \sqrt{\lambda_i},$$

so the density $p_s(s)$ satisfies

$$p_s(s)\,ds \;=\; p_C(\lambda)\,d\lambda \quad \text{with} \quad \lambda = s^2, \quad d\lambda = 2s\,ds.$$

# Appendix - MP Law Derivation for SVD (4)



We then have

$$\begin{aligned} p_s(s) &= p_C(s^2)\left|\frac{d\lambda}{ds}\right| \\ &= 2s\; p_C(s^2) \\ &= \frac{2s}{2\pi\,\sigma^2\,q\,s^2}\sqrt{(\lambda_+^{(\mathrm{cov})} - s^2)\,(s^2 - \lambda_-^{(\mathrm{cov})})}. \end{aligned}$$

Then the Marchenko–Pastur distribution for each weight matrix follows:

$$\boxed{p_s(s) = \frac{1}{\pi\,\sigma^2\,q\,s}\sqrt{(s_-^2 - s^2)\,(s^2 - s_+^2)}}$$

supported on $s \in [s_-, s_+]$.