Group 2506
Corte Riccardo
Miotto Alessandro
Rizzi Lorenzo

University of Padua
Physics of Data

Academic Year: 2024-25

Null space of the attention mechanism

We begin by examining the attention mechanism in GPT-2, with the aim of understanding how the attention heads within each layer interact. Specifically, we investigate whether the heads develop shared interpretations of the token buffer, for example, in terms of contextual meaning, syntactic structure, punctuation, and other linguistic features, and to what extent different heads collaborate.

Our analysis focuses on a fixed layer, where we study all 12 attention heads. For a given head h, the attention score matrix $A^{(h)} \in \mathbb{R}^{N\times N}$ is computed using the token buffer $X \in \mathbb{R}^{N\times 768}$, along with the query and key weight matrices $W_Q^{(h)}, W_K^{(h)} \in \mathbb{R}^{768\times 64}$, and their corresponding bias vectors $\boldsymbol{b}_Q^{(h)}, \boldsymbol{b}_K^{(h)} \in \mathbb{R}^{64}$. Each element of the attention score matrix is given by the scaled dot product between the query and key vectors:

$$A_{ij}^{(h)}(\boldsymbol{x}_i, \boldsymbol{x}_j) \propto \boldsymbol{q}_i^{(h)} \cdot \boldsymbol{k}_j^{(h)}$$

where $\boldsymbol{q}_i^{(h)}$, $\boldsymbol{k}_j^{(h)} \in \mathbb{R}^{64}$ are the query and key vectors corresponding to tokens $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively. These are computed as follows[1]:

$$\underbrace{Q^{(h)}}_{N\times 64} = \underbrace{X}_{N\times 768}\underbrace{W_Q^{(h)}}_{768\times 64} + \underbrace{\mathbf{1}}_{N\times 1}\underbrace{\boldsymbol{b}_Q^{(h)}}_{1\times 64} \quad \longrightarrow \quad \boldsymbol{q}_i^{(h)} = \boldsymbol{x}_i W_Q^{(h)} + \boldsymbol{b}_Q^{(h)}$$

$$\underbrace{K^{(h)}}_{N\times 64} = \underbrace{X}_{N\times 768}\underbrace{W_K^{(h)}}_{768\times 64} + \underbrace{\mathbf{1}}_{N\times 1}\underbrace{\boldsymbol{b}_K^{(h)}}_{1\times 64} \quad \longrightarrow \quad \boldsymbol{k}_i^{(h)} = \boldsymbol{x}_i W_K^{(h)} + \boldsymbol{b}_K^{(h)}$$

Each attention head projects the 768-dimensional token embeddings into a lower-dimensional space of $768/12 = 64$ dimensions. Consequently, the corresponding weight matrices $W_Q^{(h)}$ and $W_K^{(h)}$ are low-rank, with $\text{rank}(W) \leq 64$ and $\text{Nullity}(W) \geq 704$. This implies the existence of nonzero vector $\boldsymbol{v}$ and $\boldsymbol{w}$ in the null spaces of $W_Q^{(h)}$ and $W_K^{(h)}$, respectively, such that their addition to any input tokens leaves the attention score matrix unchanged. Formally:

$$\boxed{A_{ij}^{(h)}(\boldsymbol{x}_i + \boldsymbol{v}, \boldsymbol{x}_j + \boldsymbol{w}) = A_{ij}^{(h)}(\boldsymbol{x}_i, \boldsymbol{x}_j) \iff \boldsymbol{v} \in \text{Null}\left(W_Q^{(h)}\right), \boldsymbol{w} \in \text{Null}\left(W_K^{(h)}\right)}$$

[1] We have verified the implementation details by inspecting the official GPT-2 source code https://github.com/openai/gpt-2/blob/master/src/model.py. It confirms that each attention head receives the full token embedding buffer $X \in \mathbb{R}^{N\times 768}$ as input. This buffer is then linearly projected into a lower-dimensional space $\mathbb{R}^{N\times 68}$ using a head-specific weight matrix W.

These null-space vectors define directions in input space that are invisible to head h. We refer to the null spaces $\mathcal{N}_Q^{(h)} := \mathrm{Null}(W_Q^{(h)})$ and $\mathcal{N}_K^{(h)} := \mathrm{Null}(W_K^{(h)})$ as *indifference subspace* for that specific head for queries and keys, respectively. Conversely, the orthogonal complements, the row spaces $\mathcal{R}_Q^{(h)} := \mathrm{Row}(W_Q^{(h)})$, $\mathcal{R}_K^{(h)} := \mathrm{Row}(W_K^{(h)})$, constitute the *attention subspace*, defining the directions in input space to which each head is actually sensitive.

For each head h, the 768-dimensional token embedding space $\mathbb{R}^{768}$ can be decomposed as a direct sum of the attention and indifference subspaces:

$$\mathbb{R}^{768} = \mathcal{R}_Q^{(h)} \oplus \mathcal{N}_Q^{(h)} = \mathcal{R}_K^{(h)} \oplus \mathcal{N}_K^{(h)}$$

This means that any input token vector $\boldsymbol{x} \in \mathbb{R}^{768}$ can be uniquely expressed as the sum of two orthogonal components: $\boldsymbol{x} = \boldsymbol{x}_{\text{att}}^{(h)} + \boldsymbol{x}_{\text{ind}}^{(h)}$, where only $\boldsymbol{x}_{\text{att}}^{(h)}$ contributes to the computation of the attention score for a specific head h.

A useful interpretation is that each head attends to a different subspace of the embedding space, effectively capturing different learned features. Measuring the overlap between these attention subspaces can help quantify how similar or distinct the heads are in terms of what they focus on. A major advantage of this approach is its token-agnostic nature: it depends only on the projection matrices, not on the specific input tokens. The validity of this low-rank subspace model can be empirically tested by averaging the differences between attention score matrices across a variety of input tokens.

Preliminary results and implementations

Both the attention subspace $\mathcal{R}$ and the indifference (null) subspace $\mathcal{N}$ can be derived using the singular value decomposition (SVD) $W = U\Sigma V^T$, where $V \in \mathbb{R}^{768\times768}$ is an orthogonal matrix, the row space $\mathcal{R}$ of W is spanned by the first $r := \mathrm{rank}(W)$ columns of V, which form an orthonormal basis. The remaining $768 - r = \mathrm{nullity}(W)$ columns of V span the null space.

$$\mathcal{R} = \mathrm{Span}(V_1, ..., V_r), \quad \mathcal{N} = \mathrm{Span}(V_{r+1}, ..., V_{768})$$

```python
def null_row_basis(A : np.ndarray) -> (np.ndarray, np.ndarray):
    """
    Computes the null space and row space of a matrix A.
    Args:
        A: np.ndarray, input matrix (n, m)
    Returns:
        null_space: np.ndarray (m, r), orthonormal basis for the null space of A
        row_space: np.ndarray (m, m - r), orthonormal basis for the row space of A
    """
    if A.shape[0] < A.shape[1]:
        A = A.T
    rank = np.linalg.matrix_rank(A)
    _, _, Vt = np.linalg.svd(A)
    null_space = Vt[rank:, :].T
    row_space = Vt[:rank, :].T
    return null_space, row_space
```

Once we obtain orthonormal bases for the attention ($\mathcal{N}$) and indifference ($\mathcal{R}$) subspaces using SVD, we can begin to analyze the geometric relationship between the query and key subspaces. In particular, we are interested in how the row spaces of the query and key projection matrices relate to one another within the same head, and how these relationships vary across different heads within the same layer.

In the future, our goal is to define a method for combining the query and key subspaces into a unified representation, whether through their union, intersection, or another structured operation, to facilitate comparison between attention heads. However, for now, we analyze the query and key subspaces separately to gain initial insight and assess the validity of this approach.

To assess the similarity between query and key subspaces, we examine their geometric alignment by computing the *principal angles* between them. These angles capture the degree of overlap between two subspaces: small angles indicate high alignment, while larger angles suggest orthogonality.

```
layer, head = 0, 0

NQ, RQ = null_row_basis(WQ[layer, head])
NK, RK = null_row_basis(WK[layer, head])

from scipy.linalg import subspace_angles
angles_NQ_NK = np.degrees(subspace_angles(NQ, NK))
angles_RQ_RK = np.degrees(subspace_angles(RQ, RK))
```

From the plots in Figure 1 and Figure 2, we can infer the relative positioning of the subspaces that collectively span the full embedding space. The strong concentration of principal angles near 0° between the null spaces of the queries $\mathcal{N}_Q$ and keys $\mathcal{N}_K$ suggests a large overlap of approximately 640 dimensions.

Additionally, the principal angles between the row spaces $\mathcal{R}_Q$ and $\mathcal{R}_K$ are mostly concentrated near 90°, indicating that these subspaces are nearly orthogonal and do not intersect. This leads to the natural decomposition of the full space into a direct sum of mutually orthogonal components:

$$\mathbb{R}^{768} = \mathcal{R}_Q \oplus \mathcal{R}_K \oplus (\mathcal{N}_Q \cap \mathcal{N}_K)$$

This decomposition highlights that the query and key row spaces are disjoint, and both are embedded within the shared null spaces of the other, making the intersection of the null spaces the only remaining component to complete the basis of the full embedding space.

This topological structure is also visually supported by Figure 3, where we sampled vectors from each subspace and applied t-SNE for dimensionality reduction to 2D. The resulting projection clearly shows that the null spaces are highly overlapping, while the row spaces form separate, distinguishable clusters.

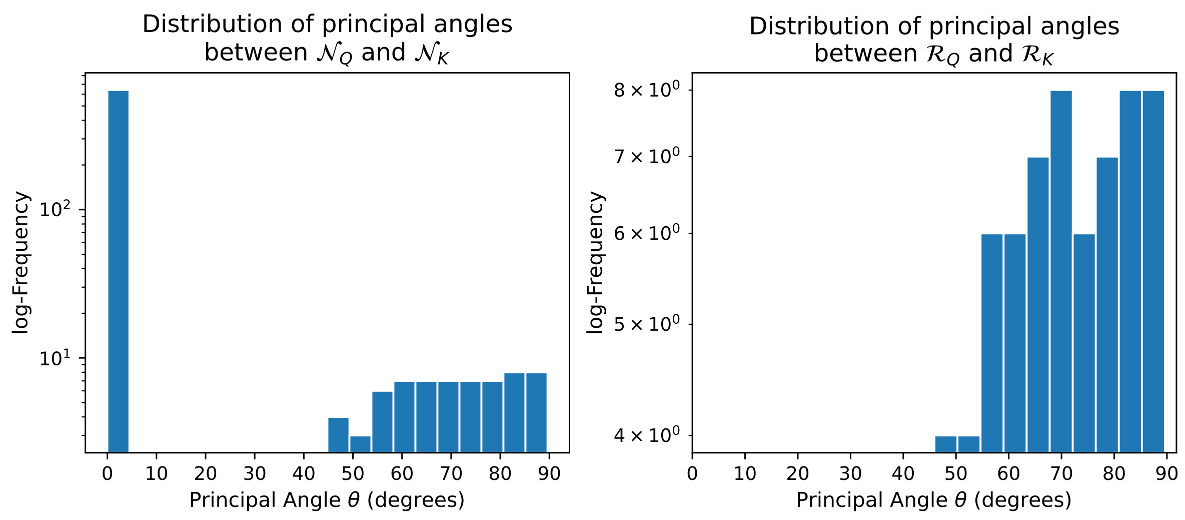

Figure 1: Distribution of principal angles between query and key subspaces. The null spaces $\mathcal{N}_Q$ and $\mathcal{N}_K$ exhibit a higher concentration of angles near $0°$, indicating a significant overlap between these indifference subspaces.

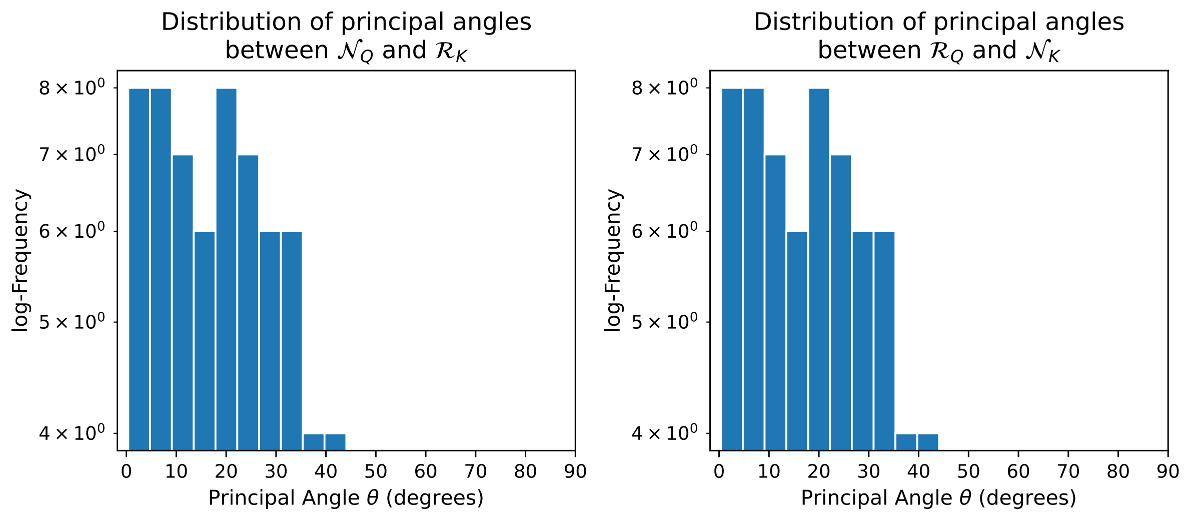

Figure 2: Distribution of principal angles between query and key subspaces. The angles between null spaces $\mathcal{N}$ and row spaces $\mathcal{R}$ exhibit a higher concentration for smaller angles, indicating a significant overlap between the attention subspace of queries and the indifference subspace of keys, and the vice versa.

Based on these considerations, we define the attention subspace of a given head as the union of its query and key row spaces $\mathcal{R}$. The remaining component, $\mathcal{N} := \mathcal{N}_Q \cap \mathcal{N}_K$, represents the indifference subspace, the directions in the embedding space that leave the attention score matrix of the head invariant.

To assess the similarity between two attention heads i and j, we compute the distance between their respective attention subspaces, $\mathcal{R}^{(i)}$ and $\mathcal{R}^{(j)}$. A larger distance indicates that the two heads attend to distinct and largely disjoint regions of the embedding space, potentially reflecting specialization in different syntactic or semantic features. Conversely, heads that capture

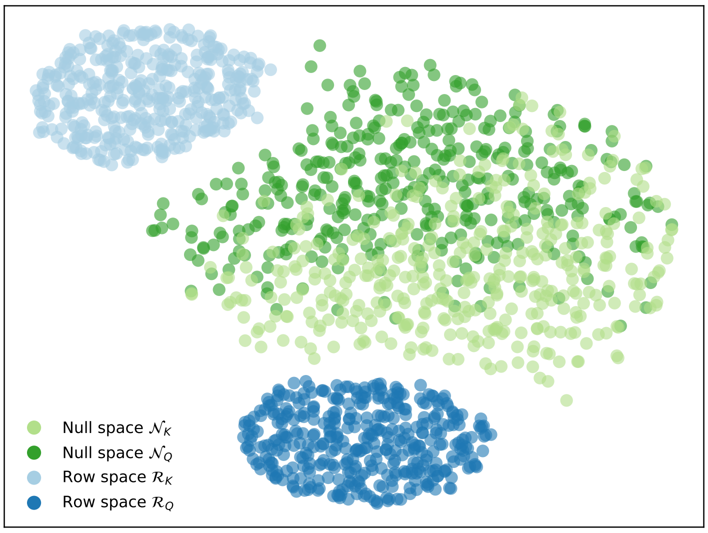

Figure 3: t-SNE visualization of query and key subspaces. The projection reveals a strong overlap between the null spaces $\mathcal{N}_Q$ and $\mathcal{N}_K$, consistent with their shared principal directions. In contrast, the row spaces $\mathcal{R}_Q$ and $\mathcal{R}_K$ appear more distinct, supporting the hypothesis that these attention subspaces capture orthogonal or complementary features.

similar structures are expected to have closer or overlapping attention subspaces.

We quantify this distance using the *Grassmann distance.* It is derived from the principal angles $\{\theta_1, ..\theta_k\}$ between the subspaces and is computed as:

$$d_{\mathrm{Gr}}(\mathcal{R}^{(i)}, \mathcal{R}^{(j)}) = \left(\sum_{l=1}^{k} \theta_l^2\right)^{1/2}$$

```python
def grassmann_distance(A : np.ndarray, B : np.ndarray) -> float:
    """
    Computes the Grassmann distance between two subspaces A and B.
    Args:
        A: np.ndarray, orthonormal basis for subspace A
        B: np.ndarray, orthonormal basis for subspace B
    Returns:
        float, Grassmann distance between the two subspaces
    """
    from scipy.linalg import subspace_angles
    angles = subspace_angles(A, B)
    return np.sqrt(np.sum(angles**2))
```

In future, to test the relevance of this theoretical framework, we could sample different token buffers and compute the pairwise differences between the resulting attention matrices across heads and layers. We then compared these empirical differences to the Grassmann distances computed between the corresponding attention subspaces (Figure 4).

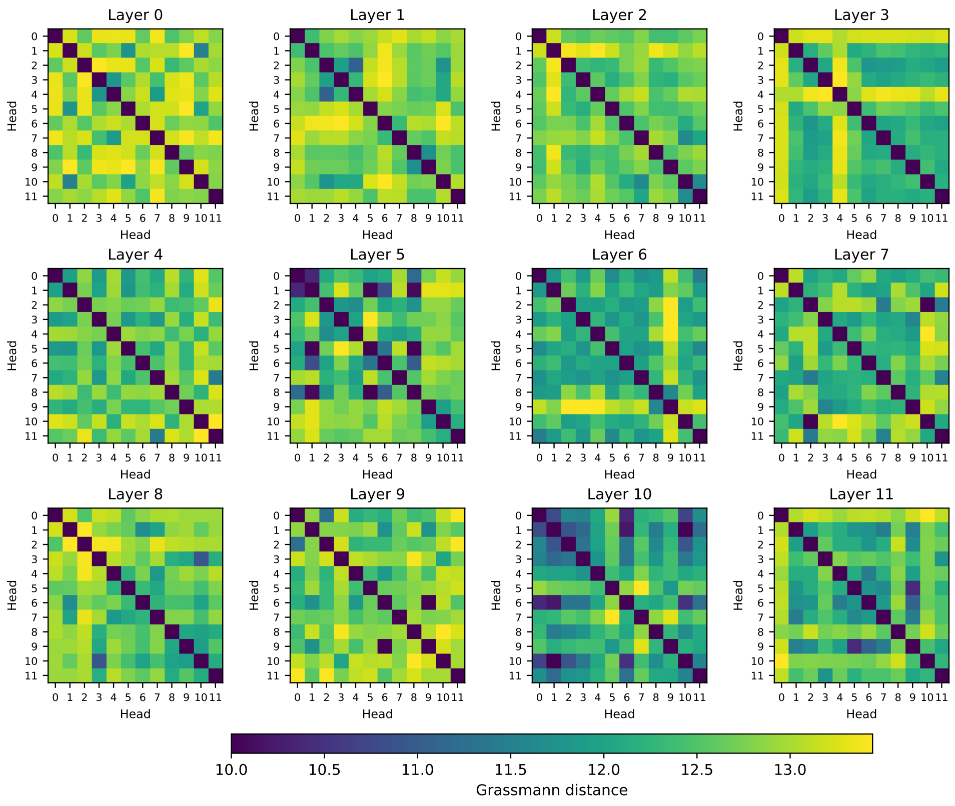

Figure 4: Grassmann distances between attention subspaces across heads, layer by layer. Each heatmap corresponds to one of the 12 layers in GPT-2, showing the pairwise Grassmann distances between the attention subspaces of the 12 heads in that layer. The attention subspace of each head is defined as the union of its query and key row spaces. Lower values (darker colors) indicate higher similarity or overlap between attention subspaces, while higher values (brighter colors) suggest more orthogonal. This visualization reveals how attention head diversity evolves across layers and highlights patterns of specialization.

```python
grassmann_layers = []    # storing the 12x12 matricies for each layer
for layer in range(12):
    layers_att = []      # save the attention space (R) basis
    for head in range(12):
        _, RQ = null_row_basis(WQ[layer, head])
        _, RK = null_row_basis(WK[layer, head])
        attention_space = union_subspace(RQ, RK)
        layers_att.append(attention_space)

    grassman = np.zeros((12, 12))

    for i in range(12):
        for j in range(12):
            if i > j:    # exploiting symmetricity
                grassman[i, j] = grassmann_distance(layers_att[i], layers_att[j])
                grassman[j, i] = grassman[i, j]

    grassmann_layers.append(grassman)
```