

Measuring Affinity between Attention-Head Weight Subspaces via the Projection Kernel

Hiroaki Yamagiwa^{1*}

¹ SB Intuitions

Yusuke Takase²

² Kyoto University

Hidetoshi Shimodaira^{2,3}

³ RIKEN AIP

hiroaki.yamagiwa@sbintuitions.co.jp, y.takase@sys.i.kyoto-u.ac.jp,
shimo@i.kyoto-u.ac.jp

Abstract

Understanding relationships between attention heads is essential for interpreting the internal structure of Transformers, yet existing metrics do not capture this structure well. We focus on the subspaces spanned by attention-head weight matrices and quantify head-to-head relationships using the Projection Kernel (PK), a principal-angle-based measure of subspace similarity. Experiments on GPT2-small, BLOOM-560M, and Qwen2-0.5B-Instruct show that PK reproduces known head-to-head interactions on the IOI task more clearly than prior metrics such as the Composition Score. We further introduce a framework to quantify the informativeness of PK distributions by comparing them with a reference distribution derived from random orthogonal subspaces. As an application, we analyze a directed graph constructed from PK and show that, in GPT2-small, L4H7 acts as a hub by functioning as an Identity Head.

1 Introduction

Mechanistic interpretability (Olah, 2022) is a research field that aims to understand the internal computations of language models. Recently, many studies have identified functional modules by probing models with specific input sentences, exemplified by the Indirect Object Identification (IOI) task (Wang et al., 2023) (see Appendix B for related work). Such analyses capture model behavior at a microscopic level, but the resulting insights depend on the particular inputs.

In contrast, weight-based analyses can reveal global properties of a model without relying on any specific input (Ameisen et al., 2025). In particular, Elhage et al. (2021) introduced the Composition Score (CS), which quantifies the similarity between heads in different layers based on the Frobenius

*Work done while at Kyoto University.

Our code is available at <https://github.com/ymgw55/proj-kernel-attn-subspace>.

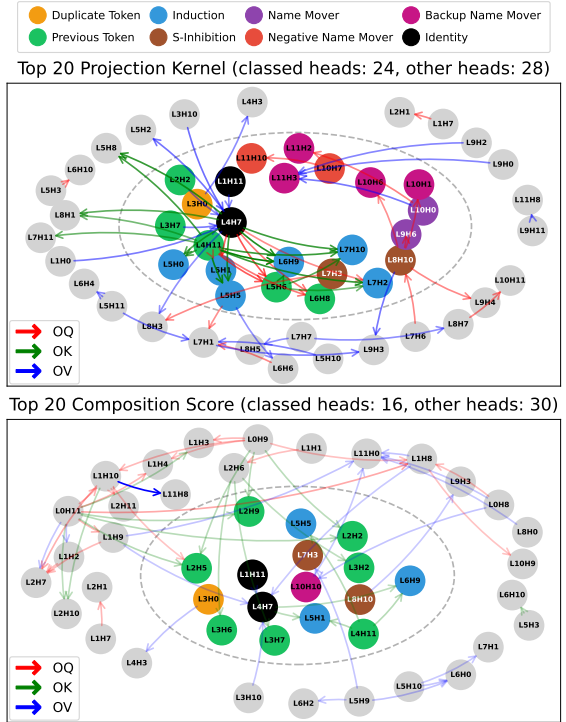


Figure 1: For the attention-head weights of GPT2-small, we compute similarities between heads using (top) the Projection Kernel (PK) and (bottom) the Composition Score (CS), and visualize the top 20 scores as edges for each weight pairing (OQ, OK, and OV). Edge opacity corresponds to the score normalized by the maximum score. The annotations for the seven head classes other than Identity Heads follow Wang et al. (2023). With PK, many heads from IOI-relevant head classes appear, and edges concentrate among these heads. In contrast, this tendency is weaker with CS; instead, many edges connect heads that do not belong to any head class.

norm of the attention-head weight matrices. Since their method is derived from the attention computation, CS tends to be large for pairs of heads that are functionally related. However, CS is a correlation-like measure that normalizes the Frobenius norm of a matrix product by the product of the Frobenius norms. This normalization removes the overall matrix scale, but it retains direction-dependent scaling

effects and relatively emphasizes directions that dominate the product, namely directions associated with large singular values. As a result, CS can emphasize the alignment of a few dominant directions more than the overall overlap between the subspaces associated with the two heads. This emphasis can hinder the comparison and interpretation of similarities between heads.

Therefore, in this work, we focus on the d_{head} -dimensional subspace spanned by the weights of each head, and characterize relationships between heads not by the magnitudes of the weights but solely by the directions of these subspaces. Specifically, we interpret the amount of information that attention heads can share during reading and writing as the *overlap between their subspaces*, and quantify it using the principal-angle-based Projection Kernel (Hamm and Lee, 2008) (PK) (Fig. 2). Figure 1 compares PK with CS in analyzing relationships between attention heads in GPT2-small (Radford et al., 2019). The contrast observed in this figure suggests that PK has properties distinct from CS when capturing functional relationships between heads.

We now outline the paper. Section 2 reviews CS, a weight-based similarity metric, and summarizes its limitations. Section 3 defines PK and examines its behavior using a toy example and a two-layer attention-only model. Section 4 introduces a method for interpreting heads based on projection onto head-specific subspaces. Section 5 describes the experimental setup using GPT2-small, and Section 6 compares PK and CS. Section 7 evaluates similarity metrics using functional groups of heads identified in the IOI task, hereafter referred to as *head classes*. Section 8 shows that the hub head L4H7 in Fig. 1 is an Identity Head and introduces a method for comparing PK distributions. It also shows that PK is effective for comparing annotated heads on the IOI task in BLOOM-560M (BigScience Workshop, 2024) and Qwen2-0.5B-Instruct (Yang et al., 2024), which have architectures different from that of GPT2-small. Finally, Section 9 concludes the paper.

2 Background

This section briefly reviews the QK and OV matrices introduced by Elhage et al. (2021) and the Composition Score (CS), which quantifies relationships between attention heads. We then summarize the limitations of CS. See Appendix C for details.

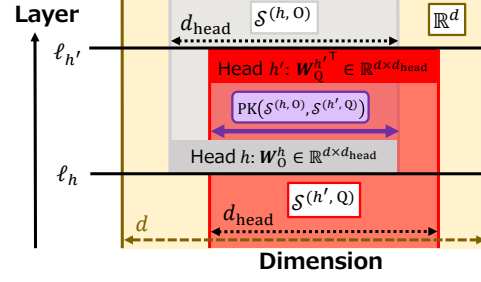


Figure 2: Illustration of measuring similarity using the Projection Kernel (PK) in (9) between the d_{head} -dimensional subspace $\mathcal{S}^{(h,O)} \subset \mathbb{R}^d$ spanned by the output weights $\mathbf{W}_O^h \in \mathbb{R}^{d \times d_{\text{head}}}$ of head h and the d_{head} -dimensional subspace $\mathcal{S}^{(h',Q)} \subset \mathbb{R}^d$ spanned by the query weights $\mathbf{W}_Q^{h'} \in \mathbb{R}^{d \times d_{\text{head}}}$ of head h' . The layer indices of heads h and h' are denoted by ℓ_h and $\ell_{h'}$ with $\ell_h < \ell_{h'}$. A larger $\text{PK}(\mathcal{S}^{(h,O)}, \mathcal{S}^{(h',Q)})$ indicates that the subspaces share more directions with small principal angles in (7), and thus that the output of head h is more strongly incorporated into the query of head h' .

2.1 QK and OV matrices

Let d be the model dimension and d_{head} the head dimension. For head h , let $\mathbf{W}_Q^h, \mathbf{W}_K^h, \mathbf{W}_V^h \in \mathbb{R}^{d_{\text{head}} \times d}$ be the query, key, and value weight matrices, and let $\mathbf{W}_O^h \in \mathbb{R}^{d \times d_{\text{head}}}$ be the output weight matrix.

Based on the attention computation, Elhage et al. (2021) defined the QK and OV matrices as

$$\mathbf{W}_{\text{QK}}^h = \mathbf{W}_Q^{h\top} \mathbf{W}_K^h \in \mathbb{R}^{d \times d}, \quad (1)$$

$$\mathbf{W}_{\text{OV}}^h = \mathbf{W}_O^h \mathbf{W}_V^h \in \mathbb{R}^{d \times d}. \quad (2)$$

Intuitively, the QK matrix captures *where* information is moved from, whereas the OV matrix captures *what* information is moved (Nanda and Bloom, 2022).

2.2 Composition Score

Using the QK and OV matrices, Elhage et al. (2021) quantified how the output of head h contributes to the query, key, and value computations of a head h' in a subsequent layer. They refer to these quantities as the scores for Q-, K-, and V-composition, respectively. In this paper, following Merullo et al. (2024), we refer to this metric as the Composition Score (CS), as we use the same form to capture head-to-head relationships more generally, rather than restricting it to the relationships between the output and the query, key, or value.

For $\mathbf{W}^{h'}, \mathbf{W}^h \in \mathbb{R}^{d \times d}$, we define

$$\text{CS}(\mathbf{W}^{h'}, \mathbf{W}^h) = \frac{\|\mathbf{W}^{h'} \mathbf{W}^h\|_F}{\|\mathbf{W}^{h'}\|_F \|\mathbf{W}^h\|_F}, \quad (3)$$

where $\|A\|_F = \sqrt{\text{tr}(A^\top A)}$ denotes the Frobenius norm. Thus, CS in (3) normalizes the Frobenius norm of the product $W^{h'} W^h$ by the Frobenius norms of $W^{h'}$ and W^h .

Using the definition in (3), we define CS_{OQ} , CS_{OK} , and CS_{OV} as the respective counterparts of the scores for Q-, K-, and V-composition described by [Elhage et al. \(2021\)](#)¹:

$$\text{CS}_{\text{OQ}}(h, h') = \text{CS}\left(W_{\text{QK}}^{h'}{}^\top, W_{\text{OV}}^h\right), \quad (4)$$

$$\text{CS}_{\text{OK}}(h, h') = \text{CS}\left(W_{\text{QK}}^{h'}, W_{\text{OV}}^h\right), \quad (5)$$

$$\text{CS}_{\text{OV}}(h, h') = \text{CS}\left(W_{\text{OV}}^{h'}, W_{\text{OV}}^h\right). \quad (6)$$

For example, in (4), replacing W_{OV}^h with W_{QK}^h yields analogous definitions such as CS_{QQ} . We use CS as a normalized measure of head-to-head interaction, which tends to take larger values for pairs of heads that are more strongly related.

2.3 Limitations

While CS provides a reasonable way to compare heads using the QK and OV matrices that arise in the attention computation, we argue that it has several limitations. First, CS retains the direction-dependent scaling present in the weight matrices. Specifically, in (3), the overlap between the output singular directions of W^h and the input singular directions of $W^{h'}$ is weighted according to the magnitudes of the corresponding singular values (see Appendix C.4). Consequently, rather than capturing the geometric overlap of the entire subspaces uniformly, CS places relatively greater emphasis on directions associated with large singular values.

Second, although any invertible $A \in \mathbb{R}^{d \times d}$ yields the factorization $W^{h'} W^h = W^{h'} A A^{-1} W^h$, in general $\|W^{h'}\|_F \|W^h\|_F \neq \|W^{h'} A\|_F \|A^{-1} W^h\|_F$. Therefore, CS is not invariant to this factorization, and typically $\text{CS}(W^{h'}, W^h) \neq \text{CS}(W^{h'} A, A^{-1} W^h)$.

Finally, we observe layerwise variation in the Frobenius norms of QK and OV matrices, a tendency for CS_{OV} to decrease as its denominator increases (see Appendix C.5), and substantial changes in CS under weight preprocessing ([Nanda and Bloom, 2022](#)) that simplifies the weights (see Appendix O). Taken together, these observations suggest that CS can be sensitive to variations in weight scaling and parameterization.

¹Here, OQ, OK, and OV indicate that the source side is the output weight of head h , while the target side is the query, key, or value weight of head h' , respectively.

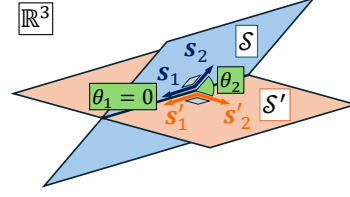


Figure 3: Principal angles $\theta_1 = 0$ and θ_2 between two planes $\mathcal{S}, \mathcal{S}' \subset \mathbb{R}^3$ with $\dim \mathcal{S} = \dim \mathcal{S}' = 2$, and the corresponding principal vectors $s_1, s_2 \in \mathcal{S}$ and $s'_1, s'_2 \in \mathcal{S}'$. Inspired by [Mandolesi \(2021, Fig. 1\)](#).

3 Quantifying overlap between subspaces spanned by weights

In this section, we describe how to quantify the overlap between the subspaces spanned by attention-head weights using the Projection Kernel (PK) based on principal angles. We then examine the behavior and usefulness of PK through experiments on a toy subspace example and a two-layer attention-only Transformer.

3.1 Principal angles

Let $\mathcal{S}, \mathcal{S}'$ be m -dimensional subspaces of $\mathbb{R}^d$. For $i = 1, \dots, m$, we define

$$\cos \theta_i = \max_{\substack{s \in \mathcal{S}, \|s\|=1, s \perp s_1, \dots, s_{i-1} \\ s' \in \mathcal{S}', \|s'\|=1, s' \perp s'_1, \dots, s'_{i-1}}} s^\top s' \quad (7)$$

and call θ_i the i -th principal angle ([Björck and Golub, 1973](#)). The vectors $s_i \in \mathcal{S}$ and $s'_i \in \mathcal{S}'$ that attain the maximum are called the i -th principal vectors ([Dahlquist and Björck, 2008](#)). As an example, Fig. 3 illustrates principal angles and principal vectors between two planes in $\mathbb{R}^3$.

Let $U, U' \in \mathbb{R}^{d \times m}$ be orthonormal basis matrices of $\mathcal{S}$ and $\mathcal{S}'$, and let $\sigma_1 \geq \dots \geq \sigma_m$ be the singular values of $U^\top U' \in \mathbb{R}^{m \times m}$. Then the i -th singular value σ_i corresponds to $\cos \theta_i$ ([Dahlquist and Björck, 2008](#)) (see Appendix D).

3.2 Projection Kernel

To quantify the overlap between subspaces $\mathcal{S}$ and $\mathcal{S}'$ based on principal angles, we use the Projection Kernel (PK) ([Hamm and Lee, 2008](#))²

$$\text{PK}(\mathcal{S}, \mathcal{S}') = \sum_{i=1}^m \cos^2 \theta_i, \quad (8)$$

²Variants of PK include $\sqrt{\text{PK}(\mathcal{S}, \mathcal{S}')}$ ([Soltanolkotabi and Candés, 2012](#)) and $\sqrt{\text{PK}(\mathcal{S}, \mathcal{S}')/m}$, a normalized version ([Soltanolkotabi et al., 2014](#)).

which satisfies³ $0 \leq \text{PK}(\mathcal{S}, \mathcal{S}') \leq m$. In particular, $\text{PK}(\mathcal{S}, \mathcal{S}') = m$ when $\mathcal{S} = \mathcal{S}'$. By algebraic manipulation (see Appendix D), we obtain

$$\text{PK}(\mathcal{S}, \mathcal{S}') = \left\| \mathbf{U}^\top \mathbf{U}' \right\|_{\text{F}}^2 = \text{tr}(\mathbf{P} \mathbf{P}'), \quad (9)$$

where $\mathbf{P} = \mathbf{U} \mathbf{U}^\top \in \mathbb{R}^{d \times d}$ and $\mathbf{P}' = \mathbf{U}' \mathbf{U}'^\top \in \mathbb{R}^{d \times d}$ are the orthogonal projection matrices onto $\mathcal{S}$ and $\mathcal{S}'$. From (9), PK can be computed directly from the basis matrices. From the definitions of PK and CS (see Appendix D), we have

$$\text{PK}(\mathcal{S}, \mathcal{S}') = (m \text{CS}(\mathbf{P}', \mathbf{P}))^2. \quad (10)$$

Therefore, PK is CS applied to projection matrices, which have no direction-dependent scaling.

3.3 Toy example of subspaces

As a toy example, let $\mathbf{b}_1, \dots, \mathbf{b}_d \in \mathbb{R}^d$ be an orthonormal basis of $\mathbb{R}^d$. Choose m basis vectors indexed by $\mathcal{I}, \mathcal{I}' \subset \{1, \dots, d\}$ to form orthonormal basis matrices $\mathbf{F}, \mathbf{F}' \in \mathbb{R}^{d \times m}$, and use PK to measure the overlap between the subspaces $\mathcal{F}, \mathcal{F}' \subset \mathbb{R}^d$ they span. By construction, the dimension of the intersection $\mathcal{F} \cap \mathcal{F}'$ is $|\mathcal{I} \cap \mathcal{I}'|$.

Using the Kronecker delta $\delta_{ii'}$, we have $\mathbf{b}_i^\top \mathbf{b}_{i'} = \delta_{ii'}$, and therefore

$$\begin{aligned} \text{PK}(\mathcal{F}, \mathcal{F}') &= \left\| \mathbf{F}^\top \mathbf{F}' \right\|_{\text{F}}^2 = \sum_{i \in \mathcal{I}, i' \in \mathcal{I}'} (\mathbf{b}_i^\top \mathbf{b}_{i'})^2 \\ &= \sum_{i \in \mathcal{I} \cap \mathcal{I}'} 1 + \sum_{i \in \mathcal{I}, i' \in \mathcal{I}', i \neq i'} 0 = |\mathcal{I} \cap \mathcal{I}'|. \end{aligned} \quad (11)$$

The PK score therefore coincides with the dimension of the intersection and gives a natural measure of overlap between the subspaces.

3.4 Measuring overlap between weight subspaces with PK

We use PK to measure overlap between attention-head weight subspaces. We define the set of weight matrices

$$\mathcal{W} = \bigcup_{h \in \mathcal{H}} \left\{ \mathbf{W}_Q^h, \mathbf{W}_K^h, \mathbf{W}_V^h, \mathbf{W}_O^h \right\}, \quad (12)$$

where $\mathcal{W} \subset \mathbb{R}^{d \times d_{\text{head}}}$ and $\mathcal{H}$ is the set of all heads. For weights $\mathbf{W}^r, \mathbf{W}^{r'} \in \mathcal{W}$, let $\mathcal{S}^r$ and $\mathcal{S}^{r'}$ denote the subspaces spanned by their column vectors⁴. In this paper, we assume $\dim \mathcal{S}^r = \dim \mathcal{S}^{r'} = d_{\text{head}}$,

³Since $-1 \leq \cos \theta_i \leq 1$, we have $0 \leq \cos^2 \theta_i \leq 1$.

⁴For example, when $r = (h, Q)$ we have $\mathbf{W}^r = \mathbf{W}_Q^h$.

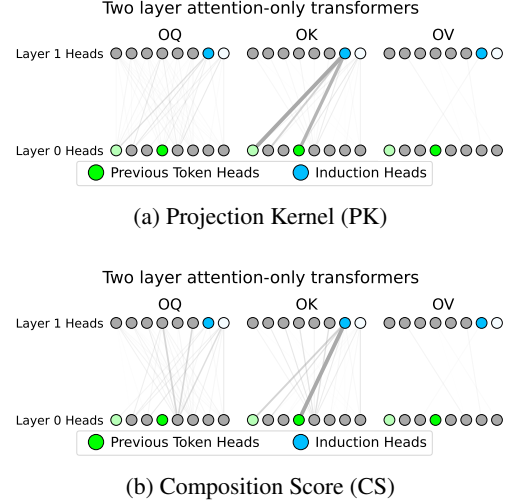


Figure 4: For a two-layer attention-only Transformer, we visualize head-to-head similarity strengths using (a) PK and (b) CS. We show the top two Previous Token Heads in the first layer and the top two Induction Heads in the second layer, coloring them by their head scores (see Appendix F for details).

and measure similarity between attention-head weight subspaces using $\text{PK}(\mathcal{S}^r, \mathcal{S}^{r'})$ (see Fig. 2). As with CS, for example, we denote the PK score between $\mathbf{W}^{(h, Q)}$ and $\mathbf{W}^{(h', Q)}$ by $\text{PK}_{Q, Q}(h, h')$.

As an example, consider the OQ pairing to compare PK and CS. Unlike CS_{OQ} , which uses the QK and OV matrices in (1) and (2), PK_{OQ} does not involve $\mathbf{W}^{(h, K)}$ or $\mathbf{W}^{(h', V)}$. Moreover, since PK is computed from orthonormal bases, it is not directly affected by weight norms.

3.5 Measuring similarity in a toy model

Elhage et al. (2021) showed that, in a two-layer attention-only Transformer, CS_{OK} takes large values between Previous Token Heads and Induction Heads. We therefore run the same experiment with PK and CS using the attn-only-2l model in TransformerLens (Nanda and Bloom, 2022)⁵.

Figure 4 shows the results. In Fig. 4a, PK_{OK} is large only for the two pairs consisting of the two Previous Token Heads and the top Induction Head, and is small for all other pairs. In contrast, in Fig. 4b, among these two pairs only one has a large CS_{OK} score and the other has a small score. Moreover, CS is noisier and tends to assign relatively larger values to many other pairs compared to PK. These results indicate that PK captures the relationships between Previous Token Heads and Induction Heads more clearly than CS.

⁵This model includes pre-norm and final LayerNorm.

4 Interpreting head subspaces

Section 3 introduced PK to measure the similarity between attention-head weight subspaces. In this section, we identify which vocabulary tokens are strongly associated with each attention head from a subspace perspective. Specifically, we project the unembedding matrix onto the subspace spanned by each head, and identify tokens whose projected norms are large as representative vocabulary items that are strongly read and written by that head.

In the Logit Lens (nostalgebraist, 2020), logits are computed by applying the unembedding matrix after the final layer normalization (LN), denoted LN_{final} . Following this idea, for $\mathbf{W}^r \in \mathcal{W}$ we define $\tilde{\mathbf{W}}^r := \text{LN}_{\text{final}}(\mathbf{W}^r) \in \mathbb{R}^{d \times d_{\text{head}}}$ and let $\tilde{\mathcal{S}}^r$ be the subspace spanned by the columns of $\tilde{\mathbf{W}}^r$. The orthogonal projection matrix onto $\tilde{\mathcal{S}}^r$ is $\tilde{\mathbf{P}}^r = \tilde{\mathbf{W}}^r (\tilde{\mathbf{W}}^{r\top} \tilde{\mathbf{W}}^r)^{-1} \tilde{\mathbf{W}}^{r\top} \in \mathbb{R}^{d \times d}$.

Let $\mathcal{T}$ be the vocabulary. For the unembedding matrix $\mathbf{E}_{\text{out}} \in \mathbb{R}^{d \times |\mathcal{T}|}$, let $\mathbf{e}_t \in \mathbb{R}^d$ denote its column for token t , which we call the unembedding vector. After preprocessing $\mathbf{e}_t$ with a function f , we project it onto $\tilde{\mathcal{S}}^r$ using $\tilde{\mathbf{P}}^r$, and treat the norm of the projected vector $\tilde{\mathbf{P}}^r f(\mathbf{e}_t) \in \mathbb{R}^d$ as a logit.

Empirically, we find that normalization shrinks the norms of less interpretable tokens, and that centering before normalization preserves the intrinsic statistical properties related to token generation probabilities (see Appendix G). Accordingly, for $\mathbf{W}^r \in \mathcal{W}$, letting $\bar{\mathbf{e}} \in \mathbb{R}^d$ be the mean unembedding vector, we compute the logit for token t as

$$\left\| \tilde{\mathbf{P}}^r \frac{\mathbf{e}_t - \bar{\mathbf{e}}}{\|\mathbf{e}_t - \bar{\mathbf{e}}\|} \right\|. \quad (13)$$

5 Experimental setup

Following Wang et al. (2023), we use GPT2-small (Radford et al., 2019) ($d = 768$, $d_{\text{head}} = 64$) as the base model. As in Fig. 4, to analyze similarities between heads based on their functions, we need to specify annotations for head classes.

We adopt the IOI-task annotations from Wang et al. (2023). However, we found that both Previous Token Heads and Induction Heads are substantially more numerous than Duplicate Token Heads⁶ (see Fig. 18 in Appendix H). We therefore recompute head scores for these two classes and assign the Previous Token Head or Induction Head label to

⁶Moreover, Wang et al. (2023) identify L0H1, L0H10, and L3H0 as Duplicate Token Heads, but in our replication we replace L0H10 with L0H5, which has the largest head score.

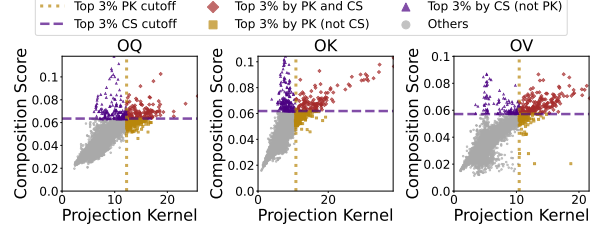


Figure 5: We compute PK and CS scores for OQ, OK, and OV and plot them as scatter plots. Spearman’s ρ values over all pairs are 0.885, 0.844, and 0.866, indicating strong correlations. In contrast, when we restrict to pairs that belong to the top 3% in either metric (the $\diamond$, $\square$, and $\triangle$ symbols), the coefficients drop to -0.251 , -0.110 , and 0.186 , and the correlations become weak. In particular, for pairs that belong only to the top 3% in CS (the $\triangle$ symbol), many have small PK scores. For visibility, we plot values up to the 99.99th percentile.

unannotated heads among the top 10 heads in each class. The full list of annotations and details are provided in Appendix H.

6 Comparing PK and CS

For each pair consisting of a head in an earlier layer and a head in a later layer, we compare PK and CS scores for OQ, OK, and OV.

6.1 Scatter plots

Figure 5 shows scatter plots of PK and CS scores for OQ, OK, and OV. The head pairs ranked highly by PK differ substantially from those ranked highly by CS. Although the overall correlation is strong, it is weaker when we consider only head pairs in the top 3% of either metric. This result indicates that PK and CS behave differently in the high-score regime.

6.2 Wiring diagrams of attention heads

In Fig. 4, we compared PK and CS scores between attention heads in a two-layer model. We perform the same analysis for GPT2-small, which has 12 layers with 12 heads per layer. We compare directed graphs constructed from head-to-head similarity scores, which we call “wiring diagrams”.

6.2.1 Setup

For each of OQ, OK, and OV, we visualize the top 20 scores. We also annotate each head h with the top 10 tokens that represent $\mathbf{W}_O^h$ using the projection method in Section 4.

6.2.2 Results for PK

Figure 6 shows the PK wiring diagram.

GPT2-small: Top 20 Projection Kernel for OQ, OK, OV

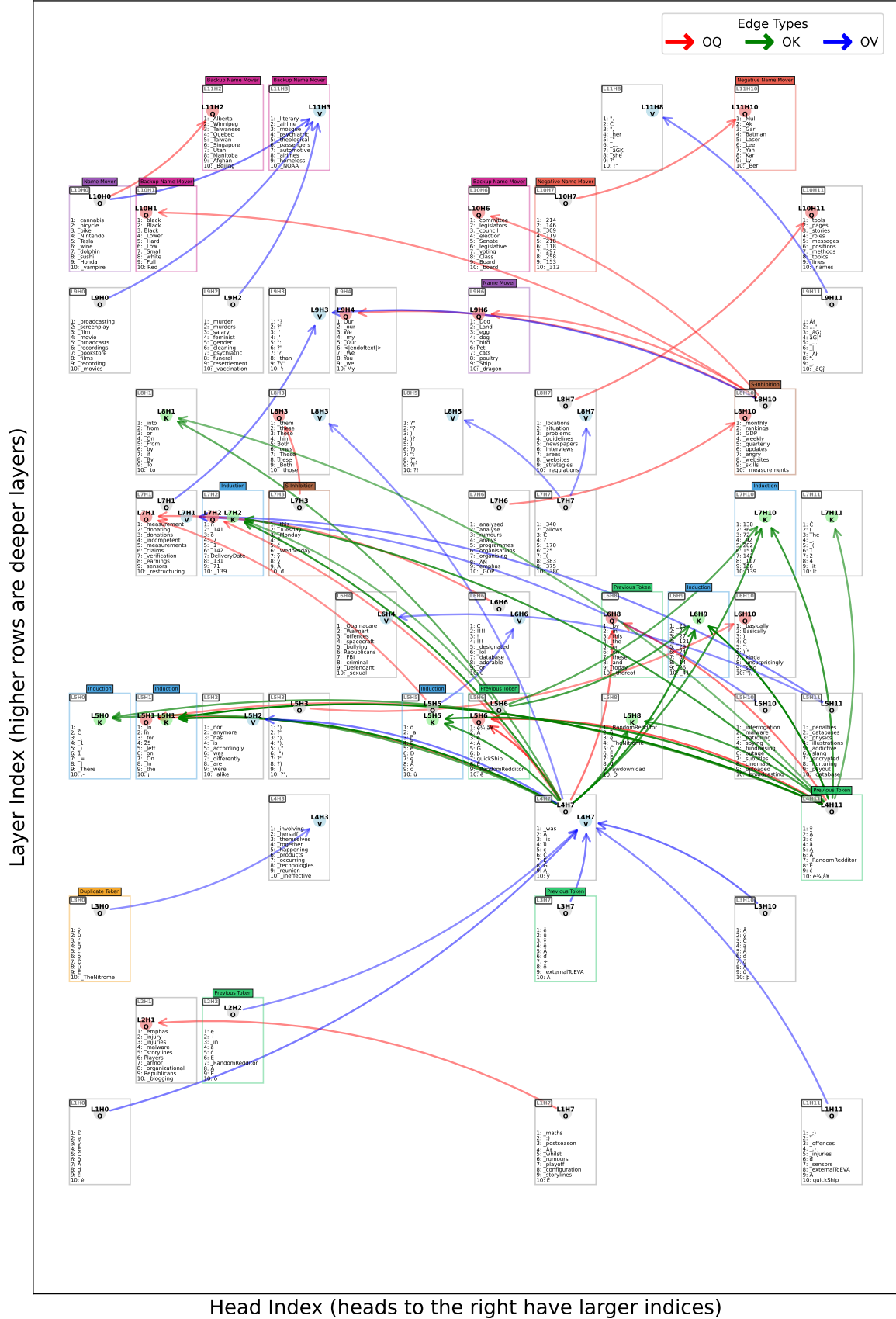


Figure 6: Visualization of the top 20 scores of PK_{OQ} , PK_{OK} , and PK_{OV} . For each head h , we show the top 10 tokens that represent W_O^h using the projection method in Section 4. For readability, we replace $\bar{G}$ in tokens with $_$. The L4H7 head is an Identity Head (see Section 8). Appendix J.1 shows results for the top 10 and top 30 scores.

OQ. We observe high scores from the S-Inhibition Head L8H10 to the Name Mover Head L9H6 and the Backup Name Mover Heads L10H1 and L10H6. This is consistent with the findings of Wang et al. (2023). In addition, scores are large among Name-Mover-related heads, such as from L10H0 to L11H2 and from L10H7 to L11H10.

OK. As in Fig. 4, we observe high scores from Previous Token Heads, such as L4H11 and L5H6, to Induction Heads. Moreover, when L4H7 or L4H11 is the source head, scores are large for many target heads. We discuss such outlet heads in Section 8.

OV. Except for the edge from L10H0 to L11H3, we find no edges where both the source and target belong to head classes, which makes interpretation difficult. Notably, many source heads yield high scores with L4H7 or L11H3 as the target. We discuss such inlet heads in Section 8.

6.2.3 Results for CS

In the CS wiring diagram (Fig. 22 in Appendix J.1), many unannotated heads appear in early layers. As a result, the Name-Mover-related heads, which are prominent under PK and are located mainly in late layers, become less visible. Moreover, CS often assigns large scores to head pairs that are widely separated across layers. For example, CS_{OV} scores from layer 1 to layer 11 are high, which makes the layerwise flow of information harder to see than with PK. As in Fig. 4, CS also captures relationships between Previous Token Heads and Induction Heads, although fewer such pairs appear than with PK.

6.2.4 Interpreting top tokens of heads

In middle and late layers, the top tokens of the heads are relatively easy to interpret. For example, those of L10H6 include many politics-related tokens. Across head classes, they are mainly nouns for Name-Mover-related heads and often numbers for Induction Heads. Some heads outside the head classes are also interpretable, such as L9H4, corresponding to personal pronouns, and L8H5, corresponding to punctuation such as “?” and “!”. Heads in early layers are harder to interpret, consistent with observations from the Logit Lens.

7 Evaluating similarity metrics

Using the annotated heads, we evaluate PK against baseline methods. We consider the OQ, OK, and

	Method	OQ	OK	OV	Avg.
PR-AUC	PK	0.446	0.451	0.289	0.395
	CS	0.227	0.352	0.215	0.265
	Simple-CS	0.245	<u>0.359</u>	0.305	0.303
	Linear CKA	<u>0.337</u>	0.323	<u>0.409</u>	<u>0.356</u>
	Proc. Sim.	0.328	0.269	0.428	0.342

Table 1: PR-AUC scores for detecting annotated heads in GPT2-small. We treat heads in head classes for the IOI task (36 out of 144) as positives, rank head pairs by similarity, and compute PR-AUC from head-level precision and recall. The best score is shown in bold and the second best score is underlined.

Method	PR-AUC	ROC-AUC
PK	0.047	0.809
CS	0.013	0.785
Simple-CS	0.016	<u>0.792</u>
Linear CKA	0.021	<u>0.715</u>
Proc. Sim.	<u>0.038</u>	0.787

Table 2: Mean PR-AUC and ROC-AUC, averaged over the same-type pairings (QQ, KK, VV, and OO) and head classes for the IOI task in GPT2-small. For each pairing type and head class, we treat within-class head pairs as positives and compute PR-AUC and ROC-AUC based on similarities, then average the scores. The best score is shown in bold and the second best is underlined. See Appendix K.2 for details.

OV pairings from Section 6, and also compare the same-type pairings QQ, KK, VV, and OO.

7.1 Baselines

Elhage et al. (2021) used $CS(\mathbf{W}_{QK}^{h'\top}, \mathbf{W}_{OV}^h)$ to measure, for example, the relationship between outputs and queries. We also include as a baseline a simpler method that directly computes $CS(\mathbf{W}_Q^{h'\top}, \mathbf{W}_O^h)$, which we call Simple-CS. In addition, as weight-based similarity methods, we adopt Linear CKA (Centered Kernel Alignment) (Kornblith et al., 2019) and a similarity based on the normalized Procrustes distance (Rohlf, 1999), which we refer to as Procrustes Similarity. See Appendix I for details.

7.2 Head detection for OQ, OK, and OV

As in Section 6, we compute similarity for each pair of a head in an earlier layer and a head in a later layer (9,504 pairs). We treat heads in head classes (36 out of 144) as positives and compare detection performance of the similarity metrics. We evaluate each weight pairing type OQ, OK, and OV. Specifically, we rank head pairs by similarity, compute precision and recall based on whether the heads are positive, and report PR-AUC.

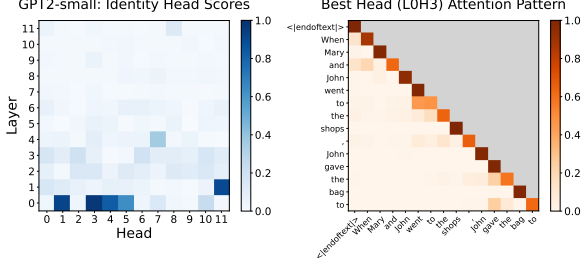


Figure 7: For GPT2-small, (left) Identity Head scores for each head and (right) an example attention pattern of the L0H3 head, which has the largest score. In particular, several heads in early layers and L4H7 have large scores. See Appendix H for details.

Table 1 shows the results. PK outperforms CS for all pairing types, and achieves the best performance for OQ and OK. The relatively low performance of PK for OV is consistent with the difficulty of interpreting OV in Fig. 6.

7.3 Head classification for same-type pairings QQ, KK, VV, and OO

We expect that, when comparing weights of the same type, similarity will be higher for heads within the same head class. We therefore evaluate classwise classification performance for each similarity metric. We consider the $\binom{144}{2} = 10,296$ directed head pairs whose source is in an earlier layer than the target or in the same layer as the target. We treat head pairs that belong to the same head class as positives, rank pairs by similarity, and compute PR-AUC and ROC-AUC. We compare results for the pairing types QQ, KK, VV, and OO.

Table 2 reports the mean scores over QQ, KK, VV, and OO. In this highly imbalanced setting, where the number of positive pairs is very small⁷, PK achieves the highest mean PR-AUC and ROC-AUC and shows strong classification performance. These results suggest that heads in the same head class have large overlaps between the subspaces spanned by their weights and therefore handle similar information in the residual stream.

8 Discussion

In this section, we first examine the function of L4H7 in GPT2-small, whose hub-like behavior was observed in Section 6.2.2, and then discuss heads that act as hubs. We then compare PK distributions across weight pairing types in GPT2-small. Finally,

⁷For example, there are only 15 pairs of Induction Heads.

		Top 1	Top 2	Top 3	Top 4	Top 5
Inlet	OQ	L10H7 ^{N-NM}	L5H6 ^{Prev}	L7H1	L8H10 ^{S-Inh}	L10H11
	OK	L5H1 ^{Ind}	L7H7	L5H5 ^{Ind}	L11H3 ^{B-NM}	L11H8
	OV	L4H7 ^{Id}	L11H3 ^{B-NM}	L9H3	L5H2	L11H8
Outlet	OQ	L7H6	L1H1	L4H7 ^{Id}	L8H10 ^{S-Inh}	L5H2
	OK	L4H7 ^{Id}	L4H11 ^{Prev}	L0H9	L1H11 ^{Id}	L5H10
	OV	L0H1 ^{Dup, Id}	L3H0 ^{Dup, Id}	L4H7 ^{Id}	L5H10	L5H11

Table 3: Top five heads in GPT2-small by inlet and outlet scores for OQ, OK, and OV, defined in (83) and (85). We treat the top 10 heads by Identity Head score in Fig. 7 as Identity Heads and highlight them in bold. For heads that belong to head classes for the IOI task (including Identity Heads), we also show the class labels. See Appendix L for details.

we show that PK is also effective for comparing annotated heads on the IOI task in BLOOM-560M and Qwen2-0.5B-Instruct, which have architectures different from that of GPT2-small.

8.1 L4H7 is an Identity Head

In Fig. 6, L4H7 in GPT2-small plays the role of a hub. Our experiments show that L4H7 is an Identity Head (Gopalani et al., 2024) whose attention pattern is close to a diagonal matrix. Figure 7 shows that several heads in early layers and L4H7 have large Identity Head scores (see Appendix H).

8.2 Detecting hub heads

To detect hub heads such as L4H7, we focus on (i) target heads that receive the maximum PK from source heads in earlier layers and (ii) source heads that send the maximum PK to target heads in later layers. For each of OQ, OK, and OV, we quantify these as inlet and outlet scores, defined in (83) and (85) in Appendix L.

Table 3 lists the top five heads by inlet and outlet scores. For inlet scores, heads in middle and late layers tend to rank highly, and for OV, L4H7 has the largest value. In contrast, for outlet scores, heads in early to middle layers tend to rank highly, and for OK, L4H7 has the largest value. Many Identity Heads also appear among the top-ranked heads.

Although Identity Heads are widely known (Bonino et al., 2025; de Llano et al., 2026), their simplicity means they have received less scrutiny than Previous Token Heads or Induction Heads. Our results may therefore offer a new perspective on Identity Heads. A more detailed analysis of Identity Heads, including whether L4H7 is special, is left for future work.

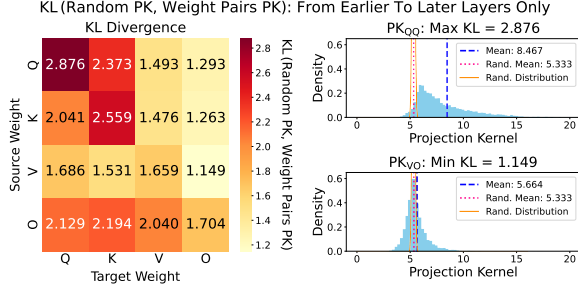


Figure 8: (Left) KL divergence between the PK distribution for random orthogonal matrices in (14) and the PK distributions measured for each weight pairing type in GPT2-small. (Right) The PK distributions for QQ, which has the largest KL divergence, and VO, which has the smallest KL divergence. In the left panel, the KL scores are large for OQ, OK, and OV, for same-type pairings, and for query and key pairings, and are small otherwise. In the right panel, the QQ distribution differs substantially from the random distribution, whereas the VO distribution is close to the random one.

8.3 Comparing PK distributions

To compare weight pairing types, we quantify the informativeness of PK distributions. For random orthogonal matrices, using expectations of products of their entries (Meckes, 2019), we derive the following normal distribution as an approximation to the PK distribution between such matrices (see Appendix M for details):

$$\mathcal{N}\left(\frac{d_{\text{head}}^2}{d}, \frac{2d_{\text{head}}^2(d - d_{\text{head}})^2}{d^2(d - 1)(d + 2)}\right). \quad (14)$$

The Kullback–Leibler divergence (KL) between this distribution and an empirical PK distribution can be viewed as a measure of how informative the PK distribution is relative to the random baseline.

Figure 8 shows a heatmap of KL scores for the 16 weight pairing types. The KL score is large for (i) pairings with output weights as sources, (ii) same-type pairings, and (iii) query and key pairings used to compute attention patterns (QK and KQ). For other pairing types, such as those with output weights as targets, the KL scores are small. These results indicate that our framework captures meaningful differences across weight pairing types.

8.4 PK for comparing annotated heads on the IOI task in other models

To examine whether our analysis based on GPT2-small generalizes to other models, we apply the same analysis to BLOOM-560M and Qwen2-0.5B-Instruct, using the IOI task head classes from

	Method	OQ	OK	OV	Avg.
PR-AUC	PK	0.358	0.283	0.164	0.268
	CS	0.151	0.157	0.114	0.141
	Simple-CS	<u>0.266</u>	<u>0.261</u>	0.145	<u>0.224</u>
	Linear CKA	0.135	0.136	0.193	0.155
	Proc. Sim.	0.133	0.140	<u>0.169</u>	0.147

(a) BLOOM-560M

	Method	OQ	OK	OV	Avg.
PR-AUC	PK	0.247	0.369	0.137	0.251
	CS	0.140	0.275	0.156	0.190
	Simple-CS	<u>0.152</u>	0.379	0.135	<u>0.222</u>
	Linear CKA	0.096	0.093	0.088	0.092
	Proc. Sim.	0.089	0.101	0.084	0.091

(b) Qwen2-0.5B-Instruct

Table 4: Detection performance for annotated heads on the IOI task in BLOOM-560M and Qwen2-0.5B-Instruct, evaluated by PR-AUC as in Table 1.

Method	BLOOM-560M		Qwen2-0.5B-Instruct	
	PR-AUC	ROC-AUC	PR-AUC	ROC-AUC
PK	0.027	0.866	0.032	0.781
CS	0.003	0.844	0.005	0.804
Simple-CS	<u>0.015</u>	0.869	0.004	<u>0.794</u>
Linear CKA	0.001	0.533	<u>0.024</u>	0.509
Proc. Sim.	0.001	0.568	0.024	0.557

Table 5: Classification performance for annotated heads on the IOI task in BLOOM-560M and Qwen2-0.5B-Instruct, evaluated by PR- and ROC-AUC as in Table 2.

Zhang et al. (2025). These two models differ from GPT2-small in both size and architecture. See Appendix N for experimental details.

For both models, PK achieves the best performance in the annotated-head detection task (Table 4) and in the classification task evaluated by PR-AUC (Table 5)⁸. Furthermore, in BLOOM-560M, as in GPT2-small in Table 3, we find Identity Heads with large inlet and outlet scores⁹.

9 Conclusion

We use the Projection Kernel (PK) to quantify overlap between attention-head subspaces and capture head relationships. We reveal structure within head classes, identify L4H7 as an Identity Head, and interpret heads via unembedding projections. We also introduce hub detection scores and a framework to quantify PK-distribution informativeness. We expect these results to contribute to a deeper understanding of language models.

⁸Table 5 corresponds to a highly imbalanced setting with very few positive examples, so we emphasize PR-AUC rather than ROC-AUC.

⁹We do not analyze Qwen2-0.5B-Instruct here because it uses grouped-query attention (GQA) (Ainslie et al., 2023), which produces many ties in the inlet scores in (83).

Limitations

- Our analysis is limited to small language models with fewer than 1B parameters. We have not analyzed large language models (LLMs) with more modern architectures. Specifically, we mainly focus on GPT2-small (Radford et al., 2019), and Section 8.4 demonstrates the effectiveness of PK on the IOI task for BLOOM-560M (BigScience Workshop, 2024) and Qwen2-0.5B-Instruct (Yang et al., 2024) (see Appendix N.1 for differences among these model architectures).
- This study implicitly assumes a multi-head attention (MHA) architecture (Vaswani et al., 2017) in which each head has its own query, key, value, and output weights. For example, for Qwen2-0.5B-Instruct, which uses grouped-query attention (GQA) (Ainslie et al., 2023), the inlet and outlet scores can still be computed by repeating the key and value weights, but this produces many ties in the inlet scores in (83)¹⁰. Recent LLMs commonly use attention variants such as GQA, multi-query attention (MQA) (Shazeer, 2019), and multi-head latent attention (MLA) (DeepSeek-AI, 2024), rather than MHA, to improve inference speed. Therefore, as in the case of Qwen2-0.5B-Instruct, our approach may not be directly applicable to LLMs with these attention architectures.
- In addition, our evaluation of similarity metrics is based on how well they reproduce known head-to-head relationships reported for the IOI task (Wang et al., 2023). Although PK captures many of these relationships, CS and other baselines may capture other types of head-to-head relationships that are not represented by the IOI-based annotations. Future work should therefore evaluate these metrics on a broader range of tasks with known circuits, such as the greater-than task (Hanna et al., 2023) and the docstring completion task (Heimersheim and Janiak, 2023).
- As in Elhage et al. (2021), our method ignores feed-forward layers, and therefore cannot fully account for the internal computations of language models.
- Except for the interpretation of head subspaces proposed in Section 4, our method does not account for layer normalization (LN) (Ba et al., 2016), as in Elhage et al. (2021). Moreover, recent LLMs often use root mean square layer normalization (RMSNorm) without bias (Zhang and Sennrich, 2019). A careful comparison that accounts for the presence and type of normalization layers is an important direction for future work.
- For preprocessing before projecting the unembedding matrix, we restrict our comparison to four choices, namely the identity map, centering, normalization, and normalization after centering, focusing on the bias term of the final-layer LN (see Appendix G). We adopt normalization after centering, but other preprocessing choices may be more appropriate. Moreover, recent LLMs often use RMSNorm without bias, so our analysis based on the bias term of the final-layer LN may not be directly transferable to such architectures.
- PK provides a similarity measure between attention heads for each weight pairing, but it does not directly identify head functions. Nonetheless, using wiring diagrams based on PK, we discovered that L4H7 acts as a hub head. Guided by this observation, we further verified experimentally that L4H7 is an Identity Head. In this sense, PK can be useful not only for validating known head-to-head relationships but also for discovering heads with previously unnoticed functions.
- We do not provide a detailed analysis of how Identity Heads relate to hub heads. However, the following observations led us to identify L4H7 in GPT2-small as an Identity Head. In the early stage of our experiments, we noticed from Fig. 6 that L4H7 has a hub-like structure, with a high inlet score for OV and high outlet scores for OQ, OK, and OV. This structure suggests that the value weight W_V^{L4H7} is aligned with directions in the residual stream to which many heads in preceding layers write information. It further suggests that the output weight W_O^{L4H7} writes to directions in the residual stream that are likely to be read by

¹⁰For this reason, in Section 8.4, we analyze the inlet and outlet scores only for BLOOM-560M to avoid complicating the interpretation of the hub-head analysis for Qwen2-0.5B-Instruct.

subsequent heads in their query, key, and value computations. This matches the role of an Identity Head. Such a head attends to its own position, reading the residual stream only at that position through its value weights, and can then relay this information to subsequent layers. We therefore hypothesized that L4H7 might be an Identity Head, and confirmed this by measuring the Identity Head score. However, information can also be transmitted across layers through the residual stream itself. Thus, further analysis and discussion are needed to distinguish the contribution of L4H7 from information transmission through the residual stream itself.

- In Fig. 6, we visualize only the top 20 edges for each of OQ, OK, and OV. From the perspective of interpretability, there may exist better strategies for selecting edges. In practice, reducing the number of edges to the top 10 (PK: Fig. 19, CS: Fig. 21 in Appendix J.1) yields a sparser diagram, whereas increasing it to the top 30 (PK: Fig. 20, CS: Fig. 23 in Appendix J.1) makes the diagram much denser. Moreover, as the numbers of layers and heads in LLMs increase, selecting edges for visualization may become increasingly difficult when interpreting wiring diagrams of attention heads.
- PK between the subspaces spanned by head weights captures global interactions between heads, but it cannot capture token-level relationships. For example, on the IOI task, it is known that the output of an S-Inhibition Head affects Name-Mover-related heads via the query vector of the final token (Wang et al., 2023). Such behavior is not directly visible from Fig. 6.
- Even with the PK wiring diagram showing the top 30 edges for each pairing (Fig. 20 in Appendix J.1), some head-to-head relationships reported by Wang et al. (2023) are not visualized. For example, the PK values between the output weights of Duplicate Token Heads or Induction Heads and the key weights of the S-Inhibition Head are small, and therefore do not appear as edges.
- Although many heads exhibit multiple functions (Ahmad et al., 2025), we assign each

head to a single head class in our analysis. For example, we label L10H7 as a Negative Name Mover Head, but it also has a high Induction Head score, as shown in Table 8d in Appendix H¹¹.

- The inlet and outlet scores defined in (83) and (85) in Appendix L do not correct for frequency biases due to layer position. For example, in inlet scores, heads in layer 1 have only heads in layer 0 as source candidates, which can lead to relatively smaller scores. Normalizing by the number of candidate heads is possible, but we did not adopt it because it inflated inlet scores for early-layer heads.
- The token logits obtained by projecting the unembedding matrix onto each head subspace are partly interpretable in terms of the meanings of the top tokens, especially in middle and late layers, but it remains difficult to infer the specific function of a head from these tokens. For example, L10H6 has many politics-related tokens among the top tokens, but it is not obvious whether this head actually increases the generation probabilities of such tokens. To address this, we aggregated token generation probabilities for each head using the OpenWebText Corpus (Gokaslan et al., 2019), based on the attention output in (16) in Appendix C, in a manner analogous to the Logit Lens¹². However, since many contexts are unrelated to the top tokens of each head, the results did not yield meaningful interpretations. Using carefully designed inputs, as in the IOI task, may improve interpretability, but a detailed analysis is left for future work.
- In GPT2-small, the embedding and unembedding matrices are tied, but we do not provide a detailed analysis of how weight tying affects the projection results.
- In this work, we adopt a theoretical approximation of the PK distribution between random orthogonal matrices as a common baseline that does not depend on the weight pairing types or the training states. This baseline provides a consistent reference distribution for comparison. However, as shown in

¹¹In addition, L10H7 also has a copy suppression function (McDougall et al., 2024).

¹²We used <https://huggingface.co/datasets/NeelNanda/openwebtext-tokenized-9b>.

Fig. 8 (right), even the VO distribution with the smallest KL divergence, while closer to the theoretical distribution in shape than the QQ distribution with the largest KL divergence, still shows a noticeable mismatch in spread. Therefore, significance tests that treat this theoretical distribution as the null may not directly assess differences across weight pairing types, and should be interpreted with caution. To test differences across weight pairing types more directly, it is desirable to introduce an appropriate baseline based on the observed weight matrices.

- In our discussion of PK, we implicitly assume that each weight matrix $\mathbf{W}^r \in \mathcal{W}$ has full rank¹³, where $\mathcal{W} \subset \mathbb{R}^{d \times d_{\text{head}}}$ is defined in (12). In Appendix P, we assess the validity of this assumption by estimating the rank of each weight matrix from its singular values, and find that the assumption is reasonably well supported.

Acknowledgments

This work was partially supported by JSPS KAKENHI JP22H05106, JP23K28045, JP26H02495 (to HS), JP25K24366 (to HY), JST CREST JPMJCR21N3 (to HS), JST Moonshot JPMJMS263I (to HS), and JST BOOST JPMJBS2407 (to YT). We thank the developers of TransformerLens (Nanda and Bloom, 2022) for making their implementation publicly available, which facilitated our analysis.

Use of AI assistants

In the camera-ready version, we used generative AI assistants for language polishing, coding support, drafting parts of the manuscript based on experimental code and results, and checking mathematical expressions. All outputs were reviewed and verified by the authors.

References

Areeb Ahmad, Abhinav Joshi, and Ashutosh Modi. 2025. [Beyond components: Singular vector-based interpretability of transformer circuits](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. [GQA: Training generalized multi-query transformer models from multi-head checkpoints](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermy, Andy Jones, and 8 others. 2025. [Circuit tracing: Revealing computational graphs in language models](#). *Transformer Circuits Thread*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. [Layer normalization](#). *Preprint*, arXiv:1607.06450.

BigScience Workshop. 2024. [Bloom: A 176b-parameter open-access multilingual language model](#). *Journal of Machine Learning Research*, 25(422):1–74.

Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. 2021. [Too much in common: Shifting of embeddings in transformer language models and its implications](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5117–5130, Online. Association for Computational Linguistics.

Åke Björck and Gene H Golub. 1973. [Numerical methods for computing angles between linear subspaces](#). *Mathematics of computation*, 27(123):579–594.

Matteo Bonino, Giorgia Ghione, and Giansalvo Cirrincione. 2025. [The geometry of bert](#). *Preprint*, arXiv:2502.12033.

Ingwer Borg and Patrick JF Groenen. 2005. *Modern multidimensional scaling: Theory and applications*. Springer.

Yasuko Chikuse. 2003. [Statistics on special manifolds](#), volume 174. Springer Science & Business Media.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. [What does BERT look at? an analysis of BERT’s attention](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. [Towards automated circuit discovery for mechanistic interpretability](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 16318–16352. Curran Associates, Inc.

¹³Equivalently, the subspace $\mathcal{S}^r \subset \mathbb{R}^d$ spanned by $\mathbf{W}^r$ has dimension $\dim \mathcal{S}^r = d_{\text{head}}$.

- Germund Dahlquist and Åke Björck. 2008. *Numerical methods in scientific computing, volume 1*. SIAM.
- Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2023. [Analyzing transformers in embedding space](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16124–16170, Toronto, Canada. Association for Computational Linguistics.
- Enrique Queipo de Llano, Alvaro Arroyo, Federico Barbero, Xiaowen Dong, Michael M. Bronstein, Yann LeCun, and Ravid Shwartz-Ziv. 2026. [Attention sinks and compression valleys in LLMs are two sides of the same coin](#). In *The Fourteenth International Conference on Learning Representations*.
- DeepSeek-AI. 2024. [Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model](#). Preprint, arXiv:2405.04434.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. [An image is worth 16x16 words: Transformers for image recognition at scale](#). In *International Conference on Learning Representations*.
- Morris L. Eaton. 2007. [Multivariate statistics: A vector space approach](#). *Lecture Notes-Monograph Series*, 53:i–512.
- Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. [A mathematical framework for transformer circuits](#). *Transformer Circuits Thread*.
- Kawin Ethayarajh. 2019. [How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.
- Javier Ferrando and Elena Voita. 2024. [Information flow routes: Automatically interpreting language models at scale](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17432–17445, Miami, Florida, USA. Association for Computational Linguistics.
- Gabriel Franco and Mark Crovella. 2024. [Sparse attention decomposition applied to circuit tracing](#). *CoRR*, abs/2410.00340.
- Gabriel Franco, Carson Loughridge, and Mark Crovella. 2026. [Singular vectors of attention heads align with features](#). In *Forty-third International Conference on Machine Learning*.
- Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus. <http://Skylion007.github.io/OpenWebTextCorpus>.
- Pulkit Gopalani, Ekdeep Singh Lubana, and Wei Hu. 2024. [Abrupt learning in transformers: A case study on matrix completion](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. 2024. [Successor heads: Recurring, interpretable attention heads in the wild](#). In *The Twelfth International Conference on Learning Representations*.
- Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2025. [When attention sink emerges in language models: An empirical view](#). In *The Thirteenth International Conference on Learning Representations*.
- Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. 2024. [Universal neurons in GPT2 language models](#). *Transactions on Machine Learning Research*.
- Jihun Hamm and Daniel D. Lee. 2008. [Grassmann discriminant analysis: a unifying view on subspace-based learning](#). In *Proceedings of the 25th International Conference on Machine Learning, ICML '08*, page 376–383, New York, NY, USA. Association for Computing Machinery.
- Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. [How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Michael Hanna, Mateusz Piotrowski, Jack Lindsey, and Emmanuel Ameisen. 2025. [Circuit-tracer: A new library for finding feature circuits](#). In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 239–249, Suzhou, China. Association for Computational Linguistics.
- Jonathan Hayase, Alisa Liu, Yejin Choi, Sewoong Oh, and Noah Smith. 2024. [Data mixture inference attack: Bpe tokenizers reveal training data compositions](#). In *Advances in Neural Information Processing*.

- Systems*, volume 37, pages 8956–8983. Curran Associates, Inc.
- N. Heckert, J. Filliben, C. Croarkin, B. Hembree, W. Guthrie, P. Tobias, and J. Prinz. 2002. [Handbook 151: Nist/sematech e-handbook of statistical methods](#). Nist interagency/internal report (nistir), National Institute of Standards and Technology, Gaithersburg, MD.
- Stefan Heimersheim. 2024. [You can remove GPT2’s layernorm by fine-tuning](#). In *Interpretable AI: Past, Present and Future*.
- Stefan Heimersheim and Jett Janiak. 2023. [A circuit for Python docstrings in a 4-layer attention-only transformer](#). AI Alignment Forum.
- Dan Hendrycks and Kevin Gimpel. 2016. [Gaussian error linear units \(gelus\)](#). *Preprint*, arXiv:1606.08415.
- Aliyah R. Hsu, Georgia Zhou, Yeshwanth Cherapanamjeri, Yaxuan Huang, Anobel Odisho, Peter R. Carroll, and Bin Yu. 2025. [Efficient automated circuit discovery in transformers using contextual decomposition](#). In *The Thirteenth International Conference on Learning Representations*.
- Go Kamoda, Benjamin Heinzerling, Tatsuro Inaba, Keito Kudo, Keisuke Sakaguchi, and Kentaro Inui. 2025. [Weight-based analysis of detokenization in language models: Understanding the first stage of inference without inference](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6339–6358, Albuquerque, New Mexico. Association for Computational Linguistics.
- Prannay Kaul, Chengcheng Ma, Ismail Elezi, and Jiankang Deng. 2025. [From attention to activation: Unraveling the enigmas of large language models](#). In *The Thirteenth International Conference on Learning Representations*.
- Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. 2019. [Similarity of neural network representations revisited](#). In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR.
- André LG Mandolesi. 2021. [Blade products and angles between subspaces](#). *Advances in Applied Clifford Algebras*, 31:69.
- David Mareček and Rudolf Rosa. 2019. [From balustrades to pierre vinken: Looking for syntax in transformer self-attentions](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 263–275, Florence, Italy. Association for Computational Linguistics.
- Callum Stuart McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. 2024. [Copy suppression: Comprehensively understanding a motif in language model attention heads](#). In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 337–363, Miami, Florida, US. Association for Computational Linguistics.
- Elizabeth S Meckes. 2019. *The random matrix theory of the classical compact groups*, volume 218. Cambridge University Press.
- Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2024. [Talking heads: Understanding inter-layer communication in transformer language models](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Paul Michel, Omer Levy, and Graham Neubig. 2019. [Are sixteen heads really better than one?](#) In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Evan Miller. 2023. [Attention is off by one](#).
- Beren Millidge and Sid Black. 2022. [The Singular Value Decompositions of Transformer Weight Matrices are Highly Interpretable](#). AI Alignment Forum.
- Jiaqi Mu and Pramod Viswanath. 2018. [All-but-the-top: Simple and effective postprocessing for word representations](#). In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.
- Neel Nanda and Joseph Bloom. 2022. [Transformerlens](#). <https://github.com/TransformerLensOrg/TransformerLens>.
- nostalgebraist. 2020. [interpreting GPT: the logit lens](#). AI Alignment Forum.
- Chris Olah. 2022. [Mechanistic interpretability, variables, and the importance of interpretable bases](#). *Transformer Circuits Thread*.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. [In-context learning and induction heads](#). *Transformer Circuits Thread*.
- Xu Pan, Aaron Philip, Ziqian Xie, and Odelia Schwartz. 2024. [Dissecting query-key interaction in vision transformers](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Hossein Pishro-Nik. 2014. [Introduction to probability, statistics, and random processes](#).
- Rudolph W. Preisendorfer and Curtis D. Mobley. 1982. [Data intercomparison theory. III. S-phase and T-phase tests for spatial pattern and temporal evolution](#). NOAA Technical Memorandum ERL PMEL-40, Pacific Marine Environmental Laboratory.

- Ofir Press, Noah Smith, and Mike Lewis. 2022. [Train short, test long: Attention with linear biases enables input length extrapolation](#). In *International Conference on Learning Representations*.
- Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2020. [Learning to deceive with attention-based explanations](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793, Online. Association for Computational Linguistics.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). *OpenAI blog*.
- F James Rohlf. 1999. [Shape statistics: Procrustes superimpositions and tangent spaces](#). *Journal of classification*, 16(2):197–223.
- Olivier Roy and Martin Vetterli. 2007. [The effective rank: A measure of effective dimensionality](#). In *15th European Signal Processing Conference, EUSIPCO 2007, Poznan, Poland, September 3-7, 2007*, pages 606–610. IEEE.
- Jessica Rumbelow and Matthew Watkins. 2023. [Solid-GoldMagikarp \(plus, prompt generation\)](#). AI Alignment Forum.
- Peter H Schönemann. 1966. [A generalized solution of the orthogonal procrustes problem](#). *Psychometrika*, 31(1):1–10.
- Sofia Serrano and Noah A. Smith. 2019. [Is attention interpretable?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.
- Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeffrey Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Isaac Bloom, Stella Biderman, Adrià Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Mary Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, William Saunders, and 10 others. 2025. [Open problems in mechanistic interpretability](#). *Transactions on Machine Learning Research*. Survey Certification.
- Noam Shazeer. 2019. [Fast transformer decoding: One write-head is all you need](#). *CoRR*, abs/1911.02150.
- Noam Shazeer. 2020. [Glu variants improve transformer](#). *Preprint*, arXiv:2002.05202.
- Aaditya K. Singh, Ted Moskovitz, Felix Hill, Stephanie C. Y. Chan, and Andrew M. Saxe. 2024. [What needs to go right for an induction head? A mechanistic study of in-context learning circuits and their formation](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.
- Mahdi Soltanolkotabi and Emmanuel J. Candès. 2012. [A geometric analysis of subspace clustering with outliers](#). *The Annals of Statistics*, 40(4):2195 – 2238.
- Mahdi Soltanolkotabi, Ehsan Elhamifar, and Emmanuel J. Candès. 2014. [Robust subspace clustering](#). *The Annals of Statistics*, 42(2):669 – 699.
- Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. [Roformer: Enhanced transformer with rotary position embedding](#). *Neurocomput.*, 568(C).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Junxuan Wang, Xuyang Ge, Wentao Shu, Zhengfu He, and Xipeng Qiu. 2026. [Dimensional collapse in transformer attention outputs: A challenge for sparse dictionary learning](#). In *Forty-third International Conference on Machine Learning*.
- Junxuan Wang, Xuyang Ge, Wentao Shu, Qiong Tang, Yunhua Zhou, Zhengfu He, and Xipeng Qiu. 2025. [Towards universality: Studying mechanistic similarity across language model architectures](#). In *The Thirteenth International Conference on Learning Representations*.
- Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. [Interpretability in the wild: a circuit for indirect object identification in GPT-2 small](#). In *The Eleventh International Conference on Learning Representations*.
- Matthew Watkins and Jessica Rumbelow. 2023. [Solid-GoldMagikarp II: technical details and more recent findings](#). AI Alignment Forum.
- Sarah Wiegrefe and Yuval Pinter. 2019. [Attention is not not explanation](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. [Efficient streaming language models with attention sinks](#). In *The Twelfth International Conference on Learning Representations*.

Hiroaki Yamagiwa, Momose Oyama, and Hidetoshi Shimodaira. 2025. [Revisiting cosine similarity via normalized ICA-transformed embeddings](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7423–7452, Abu Dhabi, UAE. Association for Computational Linguistics.

Hiroaki Yamagiwa and Hidetoshi Shimodaira. 2025. [Norm of mean contextualized embeddings determines their variance](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7778–7808, Abu Dhabi, UAE. Association for Computational Linguistics.

Hiroaki Yamagiwa, Yusuke Takase, and Hidetoshi Shimodaira. 2024. [Axis tour: Word tour determines the order of axes in ICA-transformed embeddings](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 477–506, Miami, Florida, USA. Association for Computational Linguistics.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. [Qwen2 technical report](#). *Preprint*, arXiv:2407.10671.

Biao Zhang and Rico Sennrich. 2019. [Root mean square layer normalization](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Ruochen Zhang, Qinan Yu, Matianyu Zang, Carsten Eickhoff, and Ellie Pavlick. 2025. [The same but different: Structural similarities and differences in multilingual language modeling](#). In *The Thirteenth International Conference on Learning Representations*.

A Overview of the appendices

Appendix A provides an overview of the subsequent appendices. Since each appendix in this paper contains relatively independent material, this overview is intended to help readers directly identify and refer to the sections relevant to their interests. In addition, some appendices provide detailed derivations of the equations required for the discussion, together with the necessary background, so that they can be read independently.

Appendix B discusses related work. Appendix B.1 reviews prior work on circuit analysis, Appendix B.2 reviews analyses of model weights, and Appendix B.3 reviews analyses of attention patterns.

Appendix C provides details of the Composition Score (CS). As background, following Elhage

et al. (2021), we first derive the QK and OV matrices in Appendix C.1 and then describe products of these matrices in Appendix C.2. Appendix C.3 describes the matrices used to compute CS. Next, Appendix C.4 discusses the direction-dependent scaling inherent in CS, and Appendix C.5 examines the properties of the Frobenius norms of the QK and OV matrices used in CS for GPT2-small. Finally, Appendix C.6 derives the range of CS.

Appendix D provides details of the Projection Kernel (PK). First, Appendix D.1 shows that the optimization problem for the cosine similarities corresponding to the principal angles can be solved analytically. Next, Appendix D.2 derives PK using orthonormal bases, and Appendix D.3 derives it using orthogonal projections. Appendix D.4 then establishes the relationship between PK and CS in (10), and finally, Appendix D.5 shows that PK is invariant to rotations.

Appendix E shows that PK depends only on directions by fixing the weight pairing type and computing PK from the matrices used to compute CS¹⁴. Appendix E.1 rewrites PK in a form that allows direct comparison with the square of CS. Appendix E.2 then compares the summands in this form for the top five PK and CS pairs of each of the OQ, OK, and OV weight pairing types in GPT2-small, empirically illustrating the distinct characteristics of CS and PK.

Appendix F provides details of the wiring diagram for the two-layer attention-only Transformer shown in Fig. 4 in Section 3.5.

Appendix G provides details of the projections of the unembedding matrix in GPT2-small. First, Appendix G.1 shows that tokens with lower generation probabilities, which we regard as less interpretable, tend to have larger unembedding norms. Next, Appendix G.2 shows that normalizing the unembedding vectors before projecting them onto the subspace $\mathcal{S}^r$ reduces the projected norms of less interpretable tokens. Finally, Appendix G.3 explains why centering is necessary before normalization.

Appendix H first provides details of the computation of the head scores in Appendix H.1, and then describes the head class annotations for GPT2-small used in our experiments in Appendix H.2.

Appendix I provides details of the baselines used

¹⁴For example, for the OQ weight pairing type, CS uses the matrices $\mathbf{W}_{\text{QK}}^{h'\top}$ and $\mathbf{W}_{\text{OV}}^h$, whereas PK uses $\mathbf{W}_{\text{Q}}^{h'\top}$ and $\mathbf{W}_{\text{O}}^h$. Appendix E derives the subspaces spanned by $\mathbf{W}_{\text{Q}}^{h'\top}$ and $\mathbf{W}_{\text{O}}^h$ from $\mathbf{W}_{\text{QK}}^{h'\top}$ and $\mathbf{W}_{\text{OV}}^h$.

in this work: Simple-CS (Appendix I.1), Linear CKA (Appendix I.2), and Procrustes Similarity (Appendix I.3).

Appendix J presents additional experimental results for each method on GPT2-small. Appendix J.1 presents wiring diagrams for PK and CS with varying numbers of displayed edges, as well as wiring diagrams for the baseline methods. Next, Appendix J.2 shows heatmaps for each method for the 16 weight pairing types. Finally, Appendix J.3 compares the layerwise mean values of each method for the QQ, KK, VV, and OO weight pairing types.

Appendix K provides details of head detection and head classification for GPT2-small in Appendix K.1 and Appendix K.2, respectively.

Appendix L provides details of the inlet and outlet scores for GPT2-small.

Appendix M describes approximate distributions of PK between random orthogonal matrices. First, Appendix M.1 simplifies the setting using the rotation invariance of PK. Next, Appendix M.2 derives the PK distribution under a loose approximation in which the entries of random orthogonal matrices are approximated by Gaussian distributions. Finally, Appendix M.3 derives the PK distribution under a tighter approximation using the distributions of the entries of random orthogonal matrices.

Appendix N provides details of additional experiments using BLOOM-560M and Qwen2-0.5B-Instruct. First, Appendix N.1 compares the architectures of these two models with that of GPT2-small, and Appendix N.2 reports their head scores and describes the resulting head class annotations. Next, Appendix N.3 provides details of head detection and classification. Finally, Appendix N.4 provides details of hub heads in BLOOM-560M.

Appendix O compares PK and CS using preprocessed weights of GPT2-small. First, Appendix O.1 provides background on weight preprocessing. For both PK and CS, we compare the original and preprocessed weights using wiring diagrams in Appendix O.2 and quantitatively evaluate their differences using MSE and Pearson’s r in Appendix O.3. For the preprocessed weights, Appendix O.4 shows heatmaps of PK and CS for the 16 weight pairing types, and Appendix O.5 compares their layerwise mean values for the QQ, KK, VV, and OO weight pairing types.

Appendix P revisits the assumption made in Section 3.4 that each attention weight matrix $\mathbf{W}^r \in \mathcal{W}$ of GPT2-small has full column rank.

B Related work

B.1 Circuit analysis

Mechanistic interpretability (Olah, 2022; Sharkey et al., 2025) aims to interpret the internal computations of language models, and circuit analysis is a common approach. As foundational work, Elhage et al. (2021) formalized the attention computation and, in a two-layer attention-only model, identified a circuit consisting of Previous Token Heads and Induction Heads. For Induction Heads, Olsson et al. (2022) connected them to in-context learning, and Singh et al. (2024) studied how they emerge during training.

As a standard benchmark for circuit analysis, Wang et al. (2023) proposed the Indirect Object Identification (IOI) task and analyzed head classes and circuits in GPT2-small (Radford et al., 2019) in detail. Subsequent work studies more fine-grained functions of a single head (e.g., L10H7) (McDougall et al., 2024), compares similarities and differences of internal structures in multilingual models (Zhang et al., 2025), and identifies head-to-head circuits from a single forward pass (Franco and Crovella, 2024).

To scale circuit identification, recent work proposes frameworks that automatically search for and extract important paths (Conmy et al., 2023; Ferrando and Voita, 2024; Hsu et al., 2025). Libraries that support feature-based circuit tracing have also been proposed to facilitate verification (Hanna et al., 2025).

There is also work examining the extent to which circuits and head functions are universal across model families (Wang et al., 2025). In addition, Gould et al. (2024) showed that successor heads, which map tokens in ordered sequences such as numbers and days of the week to their immediate successors, appear in multiple large language models.

B.2 Model weight analysis

In contrast to input-dependent analyses, some studies characterize model properties directly from weights. Elhage et al. (2021) introduced Q-, K-, and V-composition to characterize interactions between attention heads through compositions of the QK and OV matrices. Weight preprocessing methods that simplify weights without changing the output have been proposed (Nanda and Bloom, 2022), and are useful when implementing methods that remove layer normalization (Heimersheim,

2024). Moreover, Ameisen et al. (2025) analyzed the global weights of a replacement model using interference-corrected virtual weights. In addition, Kamoda et al. (2025) analyzed the 0th-layer QK matrices of GPT2-small to explain detokenization without running inference.

Another active line of work interprets mechanisms within Transformers by decomposing weight matrices using singular value decomposition (SVD). Merullo et al. (2024) applied SVD to the QK and OV matrices and analyzed information transmission between attention heads via a small number of singular directions. Ahmad et al. (2025) decomposed the QK and OV matrices as well as MLP weights using SVD, and analyzed multiple functions encoded in these weights in terms of individual singular directions. Pan et al. (2024) analyzed cosine similarities between singular vectors of QK matrices in Vision Transformers (Dosovitskiy et al., 2021), while Franco et al. (2026) theoretically and empirically studied the correspondence between singular vectors of QK matrices and feature directions in activation space. Relatedly, Wang et al. (2026) applied SVD to attention output activations rather than weights and analyzed their low-dimensional structure.

Another line of work interprets tokens by relating the unembedding matrix to singular vectors of OV matrices (Millidge and Black, 2022) or more generally to weight matrices (Millidge and Black, 2022; Dar et al., 2023). Our method is related to these approaches, but identifies representative tokens by orthogonally projecting the unembedding matrix onto the subspace associated with a specific attention head.

B.3 Attention pattern analysis

Besides circuit identification, analyses based on visualizations and statistics of attention matrices (patterns) have long been used to characterize attention-head functions. For example, prior work analyzed attention patterns in BERT (Devlin et al., 2019) and examined their correspondence to linguistic properties (Clark et al., 2019; Mareček and Rosa, 2019). It has also been reported that many heads are redundant, and pruning them causes little performance degradation (Michel et al., 2019).

However, treating attention patterns as explanations requires caution. The validity of explanations based on attention patterns has been extensively discussed (Wiegrefe and Pinter, 2019; Serrano and Smith, 2019), and explanations can be intention-

ally manipulated (Pruthi et al., 2020). Moreover, excessive attention to specific tokens, especially the first token, has been reported, and mitigation methods have been proposed (Miller, 2023; Xiao et al., 2024; Gu et al., 2025; Kaul et al., 2025).

C Details of the Composition Score (CS)

As background, following Elhage et al. (2021), we first derive the QK and OV matrices from the attention computation in a single layer and then explain the matrix products that arise from the attention computation across two layers. We then describe the matrices used to compute CS and discuss the direction-dependent scaling inherent in CS. Next, using GPT2-small, we examine layerwise variation in the QK and OV Frobenius norms and the relationship between CS scores and their denominators. Finally, we show that CS lies in $[0, 1]$.

C.1 Background: deriving the QK and OV matrices from single-layer attention

We represent n token embeddings as a matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$. Following Elhage et al. (2021), we describe the attention computation in a single layer with input $\mathbf{X}$.

For head h , the attention pattern $a^h(\mathbf{X}) \in [0, 1]^{n \times n}$ for input $\mathbf{X}$ is computed from the query matrix $\mathbf{W}_Q^h \mathbf{X} \in \mathbb{R}^{d_{\text{head}} \times n}$ and the key matrix $\mathbf{W}_K^h \mathbf{X} \in \mathbb{R}^{d_{\text{head}} \times n}$ as

$$\begin{aligned} a^h(\mathbf{X}) &= \sigma \left(\left(\mathbf{W}_Q^h \mathbf{X} \right)^\top \mathbf{W}_K^h \mathbf{X} \right) \\ &= \sigma \left(\mathbf{X}^\top \mathbf{W}_Q^{h\top} \mathbf{W}_K^h \mathbf{X} \right) = \sigma \left(\mathbf{X}^\top \mathbf{W}_{\text{QK}}^h \mathbf{X} \right), \end{aligned} \quad (15)$$

where $\sigma(\cdot)$ denotes the softmax that includes normalization by $\sqrt{d_{\text{head}}}$ and the causal mask¹⁵. The expression in (15) contains the QK matrix $\mathbf{W}_{\text{QK}}^h \in \mathbb{R}^{d \times d}$ defined in (1). For simplicity, we write $\mathbf{A}^h = a^h(\mathbf{X}) \in [0, 1]^{n \times n}$.

Next, we consider the write from head h to the residual stream. The residual stream accumulates the outputs written by the model components through residual connections (Elhage et al., 2021). We denote the contribution written by head h as $o^h(\mathbf{X}) \in \mathbb{R}^{d \times n}$. Using the attention pattern $\mathbf{A}^h$, the value matrix $\mathbf{W}_V^h \mathbf{X} \in \mathbb{R}^{d_{\text{head}} \times n}$, and the output

¹⁵Here $\sigma(\mathbf{Y})$ denotes the row-wise softmax of $\mathbf{Y} / \sqrt{d_{\text{head}}} + \mathbf{M}_{\text{mask}}$ for $\mathbf{Y} \in \mathbb{R}^{n \times n}$, where $\mathbf{M}_{\text{mask}} \in \mathbb{R}^{n \times n}$ satisfies $(\mathbf{M}_{\text{mask}})_{ij} = -\infty$ for $j > i$ and $(\mathbf{M}_{\text{mask}})_{ij} = 0$ otherwise.

weight matrix $\mathbf{W}_O^h \in \mathbb{R}^{d \times d_{\text{head}}}$, it is computed as

$$o^h(\mathbf{X}) = \mathbf{W}_O^h \mathbf{W}_V^h \mathbf{X} \mathbf{A}^h{}^\top = \mathbf{W}_{OV}^h \mathbf{X} \mathbf{A}^h{}^\top. \quad (16)$$

The expression in (16) contains the OV matrix $\mathbf{W}_{OV}^h \in \mathbb{R}^{d \times d}$ defined in (2).

C.2 Background: products of QK and OV matrices from two-layer attention

Following Elhage et al. (2021), we describe the attention computation across two layers¹⁶ with input $\mathbf{X}$. Let $\mathcal{H}_\ell$ be the set of heads in layer ℓ . The contribution of the attention heads in layer ℓ to the residual stream is

$$\mathbf{O}^\ell = \sum_{h \in \mathcal{H}_\ell} o^h(\mathbf{X}) \in \mathbb{R}^{d \times n}. \quad (17)$$

We consider the input to a head h' in layer $\ell' (> \ell)$ to be the residual stream after the attention layer in layer ℓ , namely $\mathbf{X} + \mathbf{O}^\ell$.

As in (15), the attention pattern $\mathbf{A}^{h'} \in [0, 1]^{n \times n}$ is computed as

$$\begin{aligned} \mathbf{A}^{h'} &= a^{h'}(\mathbf{X} + \mathbf{O}^\ell) \\ &= \sigma \left((\mathbf{X} + \mathbf{O}^\ell)^\top \mathbf{W}_{QK}^{h'} (\mathbf{X} + \mathbf{O}^\ell) \right) \in [0, 1]^{n \times n}. \end{aligned} \quad (18)$$

The argument of $\sigma(\cdot)$ in (18), $(\mathbf{X} + \mathbf{O}^\ell)^\top \mathbf{W}_{QK}^{h'} (\mathbf{X} + \mathbf{O}^\ell) \in \mathbb{R}^{n \times n}$, can be expanded as

$$\begin{aligned} &(\mathbf{X} + \mathbf{O}^\ell)^\top \mathbf{W}_{QK}^{h'} (\mathbf{X} + \mathbf{O}^\ell) \\ &= \left(\mathbf{X} + \sum_{h \in \mathcal{H}_\ell} o^h(\mathbf{X}) \right)^\top \\ &\quad \mathbf{W}_{QK}^{h'} \left(\mathbf{X} + \sum_{h \in \mathcal{H}_\ell} o^h(\mathbf{X}) \right) \quad (\because (17)) \\ &= \mathbf{X}^\top \mathbf{W}_{QK}^{h'} \mathbf{X} + \sum_{h \in \mathcal{H}_\ell} o^h(\mathbf{X})^\top \mathbf{W}_{QK}^{h'} \mathbf{X} \\ &\quad + \sum_{h \in \mathcal{H}_\ell} \mathbf{X}^\top \mathbf{W}_{QK}^{h'} o^h(\mathbf{X}) \\ &\quad + \left(\sum_{h \in \mathcal{H}_\ell} o^h(\mathbf{X}) \right)^\top \mathbf{W}_{QK}^{h'} \left(\sum_{h \in \mathcal{H}_\ell} o^h(\mathbf{X}) \right). \end{aligned} \quad (19)$$

To examine weight relationships between heads h and h' , we focus on the summand in the second

¹⁶Note that we ignore the layer normalization and feed-forward layers.

term, which uses $o^h(\mathbf{X})$ and $\mathbf{X}$ as inputs to the query and key computations, respectively, and the summand in the third term, which uses $\mathbf{X}$ and $o^h(\mathbf{X})$.

For the summand $o^h(\mathbf{X})^\top \mathbf{W}_{QK}^{h'} \mathbf{X} \in \mathbb{R}^{n \times n}$ in the second term of (19), we have

$$\begin{aligned} &o^h(\mathbf{X})^\top \mathbf{W}_{QK}^{h'} \mathbf{X} \\ &= \left(\mathbf{W}_{OV}^h \mathbf{X} \mathbf{A}^h{}^\top \right)^\top \mathbf{W}_{QK}^{h'} \mathbf{X} \quad (\because (16)) \\ &= \mathbf{A}^h \mathbf{X}^\top \underbrace{\mathbf{W}_{OV}^h{}^\top \mathbf{W}_{QK}^{h'}}_{\text{head-}h \text{ } \textcircled{O} \rightarrow \text{head-}h' \text{ } \textcircled{Q}} \mathbf{X}. \end{aligned} \quad (20)$$

Thus, the relationship from the output of head h to the query of head h' is represented by the matrix product $(\mathbf{W}_{OV}^h{}^\top \mathbf{W}_{QK}^{h'})^\top = \mathbf{W}_{QK}^{h'}{}^\top \mathbf{W}_{OV}^h$.

For the summand $\mathbf{X}^\top \mathbf{W}_{QK}^{h'} o^h(\mathbf{X}) \in \mathbb{R}^{n \times n}$ in the third term of (19), we have

$$\begin{aligned} &\mathbf{X}^\top \mathbf{W}_{QK}^{h'} o^h(\mathbf{X}) \\ &= \mathbf{X}^\top \underbrace{\mathbf{W}_{QK}^{h'} \mathbf{W}_{OV}^h}_{\text{head-}h' \text{ } \textcircled{K} \leftarrow \text{head-}h \text{ } \textcircled{O}} \mathbf{X} \mathbf{A}^h{}^\top \quad (\because (16)). \end{aligned} \quad (21)$$

Thus, the relationship from the output of head h to the key of head h' is represented by the matrix product $\mathbf{W}_{QK}^{h'} \mathbf{W}_{OV}^h$.

As in (16), the write of head h' to the residual stream, $o^{h'}(\mathbf{X} + \mathbf{O}^\ell) \in \mathbb{R}^{d \times n}$, is computed as

$$\begin{aligned} &o^{h'}(\mathbf{X} + \mathbf{O}^\ell) \\ &= \mathbf{W}_{OV}^{h'} \left(\mathbf{X} + \sum_{h \in \mathcal{H}_\ell} o^h(\mathbf{X}) \right) \mathbf{A}^{h'}{}^\top \quad (\because (17)) \\ &= \mathbf{W}_{OV}^{h'} \mathbf{X} \mathbf{A}^{h'}{}^\top + \sum_{h \in \mathcal{H}_\ell} \underbrace{\mathbf{W}_{OV}^{h'} \mathbf{W}_{OV}^h}_{\text{head-}h' \text{ } \textcircled{V} \leftarrow \text{head-}h \text{ } \textcircled{O}} \mathbf{X} \mathbf{A}^h{}^\top \mathbf{A}^{h'}{}^\top \\ &\quad (\because (16)). \end{aligned} \quad (22)$$

Thus, the relationship from the output of head h to the value of head h' is represented by the matrix product $\mathbf{W}_{OV}^{h'} \mathbf{W}_{OV}^h$.

C.3 Matrices used to compute CS

In Appendix C.2, we saw that the matrix products $\mathbf{W}_{QK}^{h'}{}^\top \mathbf{W}_{OV}^h$ for OQ, $\mathbf{W}_{QK}^{h'} \mathbf{W}_{OV}^h$ for OK, and $\mathbf{W}_{OV}^{h'} \mathbf{W}_{OV}^h$ for OV represent weight relationships. Based on these products, Elhage et al. (2021) compute CS in (3) as the Frobenius norm of each product normalized by the product of the Frobenius

norms of the original matrices, which yields CS_{OQ} in (4), CS_{OK} in (5), and CS_{OV} in (6).

While Elhage et al. (2021) compute CS only for the OQ, OK, and OV pairings, we compute CS for all $4 \times 4 = 16$ combinations of query, key, value, and output weights for heads h and h' to enable a direct comparison with PK. As shown in Appendix C.2, whether a transpose appears depends on the multiplication order in the attention computation. Accordingly, we define the matrices associated with the query, key, value, and output of head h as $\mathbf{W}_{\text{QK}}^h$, $\mathbf{W}_{\text{QK}}^{h\top}$, $\mathbf{W}_{\text{OV}}^h$, and $\mathbf{W}_{\text{OV}}^{h\top}$, and the matrices associated with the query, key, value, and output of head h' as $\mathbf{W}_{\text{QK}}^{h'}$, $\mathbf{W}_{\text{QK}}^{h'\top}$, $\mathbf{W}_{\text{OV}}^{h'}$, and $\mathbf{W}_{\text{OV}}^{h'\top}$. We compute CS between head weights based on these matrices¹⁷.

C.4 Details of direction-dependent scaling in CS

Section 2.3 identified direction-dependent scaling as one of the limitations of CS. In this section, we first rewrite the square of CS in (3) using SVD to show explicitly that CS retains direction-dependent scaling. We then interpret the Frobenius norm in terms of isotropic inputs and show that CS does not account for directional biases in the actual input distribution.

C.4.1 Interpretation of CS using SVD

For the matrices¹⁸ $\mathbf{W}^h, \mathbf{W}^{h'} \in \mathbb{R}^{d \times d}$ used to compute CS in (3), consider their compact SVDs

$$\mathbf{W}^h = \mathbf{U}^h \mathbf{\Sigma}^h \mathbf{V}^{h\top}, \quad (23)$$

$$\mathbf{W}^{h'} = \mathbf{U}^{h'} \mathbf{\Sigma}^{h'} \mathbf{V}^{h'\top}. \quad (24)$$

Here, $\mathbf{U}^h, \mathbf{U}^{h'} \in \mathbb{R}^{d \times d_{\text{head}}}$ and $\mathbf{V}^h, \mathbf{V}^{h'} \in \mathbb{R}^{d \times d_{\text{head}}}$ are column-orthonormal matrices whose columns are the left and right singular vectors, corresponding to the output-side and input-side directions, respectively. The matrices $\mathbf{\Sigma}^h, \mathbf{\Sigma}^{h'} \in$

¹⁷Our notation represents inputs as $d \times n$ matrices whose columns are token vectors, whereas many implementations use $n \times d$ inputs. Accordingly, the multiplication order of weight matrices and the use of transposes depend on these conventions.

¹⁸In our setting, in particular, $\mathbf{W}^h \in \{\mathbf{W}_{\text{QK}}^h, \mathbf{W}_{\text{QK}}^{h\top}, \mathbf{W}_{\text{OV}}^h, \mathbf{W}_{\text{OV}}^{h\top}\}$. From (1) and (2), each of $\mathbf{W}_{\text{QK}}^h$ and $\mathbf{W}_{\text{OV}}^h$ can be expressed as the product of a $d \times d_{\text{head}}$ matrix and a $d_{\text{head}} \times d$ matrix. Furthermore, in Section 3.4, we assume that these $d \times d_{\text{head}}$ and $d_{\text{head}} \times d$ matrices have rank d_{head} , so $\mathbf{W}^h$ also has rank d_{head} . The same argument applies to $\mathbf{W}^{h'}$.

$\mathbb{R}^{d_{\text{head}} \times d_{\text{head}}}$ are diagonal matrices with nonnegative singular values on their diagonals. We denote these singular values by $\sigma_1^h \geq \dots \geq \sigma_{d_{\text{head}}}^h \geq 0$ and $\sigma_1^{h'} \geq \dots \geq \sigma_{d_{\text{head}}}^{h'} \geq 0$. In particular, let $\mathbf{u}_1^h, \dots, \mathbf{u}_{d_{\text{head}}}^h \in \mathbb{R}^d$ denote the columns of $\mathbf{U}^h$, and let $\mathbf{v}_1^{h'}, \dots, \mathbf{v}_{d_{\text{head}}}^{h'} \in \mathbb{R}^d$ denote the columns of $\mathbf{V}^{h'}$.

Substituting (23) and (24) into the square of CS in (3), and using the column orthonormality of the singular-vector matrices, we obtain

$$\begin{aligned} & \text{CS}(\mathbf{W}^{h'}, \mathbf{W}^h)^2 \\ &= \frac{\|\mathbf{\Sigma}^{h'} \mathbf{V}^{h'\top} \mathbf{U}^h \mathbf{\Sigma}^h\|_{\text{F}}^2}{\|\mathbf{\Sigma}^{h'}\|_{\text{F}}^2 \|\mathbf{\Sigma}^h\|_{\text{F}}^2} \\ &= \frac{\sum_{i=1}^{d_{\text{head}}} \sum_{j=1}^{d_{\text{head}}} \sigma_i^{h'^2} \sigma_j^{h^2} (\mathbf{v}_i^{h'\top} \mathbf{u}_j^h)^2}{\left(\sum_{i'=1}^{d_{\text{head}}} \sigma_{i'}^{h'^2}\right) \left(\sum_{j'=1}^{d_{\text{head}}} \sigma_{j'}^{h^2}\right)} \\ &= \sum_{i=1}^{d_{\text{head}}} \sum_{j=1}^{d_{\text{head}}} \left\{ \left(\frac{\sigma_i^{h'}}{\sqrt{\sum_{i'=1}^{d_{\text{head}}} \sigma_{i'}^{h'^2}}} \mathbf{v}_i^{h'} \right)^\top \left(\frac{\sigma_j^h}{\sqrt{\sum_{j'=1}^{d_{\text{head}}} \sigma_{j'}^{h^2}}} \mathbf{u}_j^h \right) \right\}^2. \end{aligned} \quad (25)$$

In (25), each singular vector is weighted by a coefficient proportional to its corresponding singular value. Consequently, directions associated with larger singular values contribute more strongly to CS. This is precisely the direction-dependent scaling retained by CS.

C.4.2 Interpretation of the Frobenius norm using isotropic inputs

The Frobenius norm can be interpreted as measuring the magnitude of the output of a linear map averaged over isotropic reference inputs. Let $\mathbf{I}_d \in \mathbb{R}^{d \times d}$ denote the identity matrix, and let a random vector $\mathbf{x} \in \mathbb{R}^d$ satisfy $\mathbb{E}_{\mathbf{x}}[\mathbf{x}\mathbf{x}^\top] = \mathbf{I}_d$. Then, for a matrix $\mathbf{A}$,

$$\begin{aligned} \mathbb{E}_{\mathbf{x}} [\|\mathbf{A}\mathbf{x}\|^2] &= \text{tr} \left(\mathbf{A}^\top \mathbf{A} \mathbb{E}_{\mathbf{x}}[\mathbf{x}\mathbf{x}^\top] \right) \\ &= \text{tr} \left(\mathbf{A}^\top \mathbf{A} \right) = \|\mathbf{A}\|_{\text{F}}^2. \end{aligned} \quad (26)$$

Thus, $\|\mathbf{A}\|_{\text{F}}$ can be interpreted as the root mean square of $\|\mathbf{A}\mathbf{x}\|$ over isotropic reference inputs $\mathbf{x}$.

Applying this interpretation to CS, the effect of the direction-dependent scaling retained in $\|\mathbf{W}^{h'} \mathbf{W}^h\|_{\text{F}}$ can be interpreted in terms of the

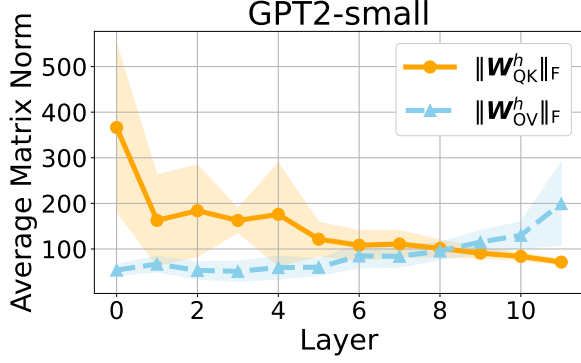


Figure 9: For the QK matrix in (1) and the OV matrix in (2), we plot the Frobenius norms across layers in GPT2-small. For each layer, we also show the 1σ range.

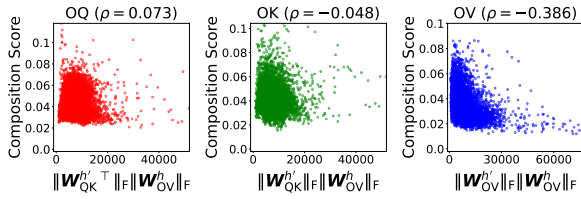


Figure 10: For each pair consisting of a head in an earlier layer and a head in a later layer in GPT2-small, totaling 9,504 pairs, we plot the OQ, OK, and OV CS scores against the corresponding denominator values. Each panel also reports Spearman’s ρ between these quantities.

magnitude of $\|\mathbf{W}^{h'} \mathbf{W}^h \mathbf{x}\|$ for isotropic reference inputs $\mathbf{x}$. However, contextualized representations in Transformers are known to exhibit anisotropy (Ethayarajh, 2019; Yamagiwa and Shimodaira, 2025). Therefore, this interpretation based on isotropic reference inputs does not necessarily reflect the actual activation distribution. Thus, in addition to retaining direction-dependent scaling, CS does not explicitly account for directional biases in the actual activations.

C.5 Properties of the Frobenius norms of the QK and OV matrices used in CS for GPT2-small

In this section, we examine the properties of the Frobenius norms of the QK and OV matrices used in CS for GPT2-small. Specifically, we first examine their layerwise trends (Appendix C.5.1) and then investigate the relationship between CS scores and their denominators (Appendix C.5.2).

C.5.1 Layerwise trends in Frobenius norms

Figure 9 plots the Frobenius norms of the QK and OV matrices across layers in GPT2-small. The QK norm is large at layer 0 and decreases with depth.

In contrast, the OV norm gradually increases with depth. We conjecture that these large layerwise changes in weight norms are one reason why CS in (3) can be difficult to use for quantifying differences between heads.

C.5.2 CS scores can vary with the magnitude of the denominator

Figure 10 plots the OQ, OK, and OV CS scores computed from head pairs in GPT2-small against their corresponding denominator values. For OV, the CS score tends to decrease as the denominator increases, with a Spearman’s rank correlation coefficient of $\rho = -0.386$.

CS in (3) is designed to remove differences in overall matrix scale by normalizing the Frobenius norm of the matrix product in the numerator by the product of the Frobenius norms in the denominator. However, as observed for OV in GPT2-small, the numerator does not necessarily increase proportionally with the denominator. In such cases, larger denominator values can suppress the CS score, making the score sensitive to the magnitudes of the Frobenius norms.

C.6 Range of CS

In this section, assuming $\|\mathbf{W}^{h'}\|_F \|\mathbf{W}^h\|_F \neq 0$, we show that CS in (3) satisfies

$$0 \leq \text{CS}(\mathbf{W}^{h'}, \mathbf{W}^h) = \frac{\|\mathbf{W}^{h'} \mathbf{W}^h\|_F}{\|\mathbf{W}^{h'}\|_F \|\mathbf{W}^h\|_F} \leq 1. \quad (27)$$

When $\|\mathbf{W}^{h'}\|_F = 0$ or $\|\mathbf{W}^h\|_F = 0$, we define $\text{CS}(\mathbf{W}^{h'}, \mathbf{W}^h) = 0$.

The vectors $\mathbf{v}_i^{h'}$ and $\mathbf{u}_j^h$ defined in Appendix C.4.1 are columns of the column-orthonormal matrices $\mathbf{V}^{h'}$ and $\mathbf{U}^h$, respectively, and therefore have unit norm. Thus, by the Cauchy-Schwarz inequality,

$$0 \leq \left| \mathbf{v}_i^{h'} \top \mathbf{u}_j^h \right| \leq \left\| \mathbf{v}_i^{h'} \right\| \left\| \mathbf{u}_j^h \right\| = 1. \quad (28)$$

Applying (28) to (25) gives

$$0 \leq \text{CS}(\mathbf{W}^{h'}, \mathbf{W}^h)^2 \leq 1. \quad (29)$$

Since $\text{CS}(\mathbf{W}^{h'}, \mathbf{W}^h)$ is nonnegative, taking the square root yields (27).

D Details of the Projection Kernel (PK)

In this section, we first solve the optimization problem that defines principal angles (Björck and Golub, 1973) following Dahlquist and Björck

(2008). Using this result, we show that PK can be expressed in two equivalent forms, one using orthonormal bases and the other using orthogonal projections. We then derive (10), which characterizes the relationship between PK and CS, and also present the rotation invariance of PK, which will be used in Appendix M.

D.1 Solving the cosine similarity optimization for principal angles

Let $U, U' \in \mathbb{R}^{d \times m}$ be orthonormal basis matrices of m -dimensional subspaces $\mathcal{S}, \mathcal{S}' \subset \mathbb{R}^d$, and let

$$M = U^\top U' \in \mathbb{R}^{m \times m} \quad (30)$$

have singular values $\sigma_1 \geq \dots \geq \sigma_m \geq 0$. Based on Dahlquist and Björck (2008, Theorem 8.1.20), we show that the cosine similarity $\cos \theta_i$ of the i -th principal angle θ_i between $\mathcal{S}$ and $\mathcal{S}'$ corresponds to the i -th singular value σ_i of M .

We begin with the optimization problem for the first principal angle. The first principal angle θ_1 is defined by

$$\cos \theta_1 = \max_{\substack{s \in \mathcal{S}, \|s\|=1 \\ s' \in \mathcal{S}', \|s'\|=1}} s^\top s'. \quad (31)$$

Since U and U' are orthonormal basis matrices of $\mathcal{S}$ and $\mathcal{S}'$, any $s \in \mathcal{S}$ and $s' \in \mathcal{S}'$ can be written using coefficients $g, g' \in \mathbb{R}^m$ as

$$s = Ug, s' = U'g'. \quad (32)$$

For s in (32),

$$\begin{aligned} \|s\|^2 &= s^\top s = (Ug)^\top Ug \\ &= g^\top U^\top Ug = g^\top g = \|g\|^2, \end{aligned} \quad (33)$$

where we used $U^\top U = I_m \in \mathbb{R}^{m \times m}$ since the columns of U are orthonormal. Therefore, $\|s\| = 1$ implies $\|g\| = 1$, and similarly $\|s'\| = 1$ implies $\|g'\| = 1$. For s, s' in (32), we obtain

$$\begin{aligned} s^\top s' &= (Ug)^\top U'g' \\ &= g^\top U^\top U'g' = g^\top M g'. \end{aligned} \quad (34)$$

Using $\|g\| = \|g'\| = 1$ and (34), (31) can be rewritten as

$$\cos \theta_1 = \max_{\|g\|=\|g'\|=1} g^\top M g'. \quad (35)$$

Let M in (30) be written as $M = U_M \Sigma_M V_M^\top$ via the SVD. Here, the diagonal entries of Σ_M are

the singular values $\sigma_1 \geq \dots \geq \sigma_m \geq 0$, and $U_M, V_M \in \mathbb{R}^{m \times m}$ are orthogonal matrices. For $g^\top M g'$ in (35),

$$\begin{aligned} g^\top M g' &= g^\top U_M \Sigma_M V_M^\top g' \\ &= (U_M^\top g)^\top \Sigma_M (V_M^\top g'). \end{aligned} \quad (36)$$

Define

$$\xi := U_M^\top g, \xi' := V_M^\top g' \in \mathbb{R}^m. \quad (37)$$

Since U_M and V_M are orthogonal and $\|g\| = \|g'\| = 1$, we have $\|\xi\| = \|\xi'\| = 1$. Writing $\xi = (\xi_1, \dots, \xi_m)^\top$ and $\xi' = (\xi'_1, \dots, \xi'_m)^\top$, (36) becomes

$$g^\top M g' = \xi^\top \Sigma_M \xi' = \sum_{j=1}^m \sigma_j \xi_j \xi'_j. \quad (38)$$

Since σ_1 is the largest singular value and $\|\xi\| = \|\xi'\| = 1$, the Cauchy–Schwarz inequality gives

$$\begin{aligned} &\sum_{j=1}^m \sigma_j \xi_j \xi'_j \\ &\leq \left(\sum_{j=1}^m \sigma_j^2 \xi_j^2 \right)^{1/2} \left(\sum_{j=1}^m \xi'^2_j \right)^{1/2} \\ &\leq \sigma_1 \left(\sum_{j=1}^m \xi_j^2 \right)^{1/2} \left(\sum_{j=1}^m \xi'^2_j \right)^{1/2} = \sigma_1. \end{aligned} \quad (39)$$

Equality holds when ξ and ξ' equal the first standard basis vector $e_1^{\text{Std}} = (1, 0, \dots, 0)^\top \in \mathbb{R}^m$. Therefore, for (35) we obtain

$$\cos \theta_1 = \max_{\|g\|=\|g'\|=1} g^\top M g' = \sigma_1. \quad (40)$$

Let $\xi_1 = \xi'_1 = e_1^{\text{Std}} \in \mathbb{R}^m$ be the maximizers, and let $g_1, g'_1 \in \mathbb{R}^m$ be the corresponding coefficients. From (37), we have

$$g_1 = U_M \xi_1 = U_M e_1^{\text{Std}} = u_{M,1}, \quad (41)$$

$$g'_1 = V_M \xi'_1 = V_M e_1^{\text{Std}} = v_{M,1}, \quad (42)$$

where $u_{M,1}$ and $v_{M,1}$ are the first columns of U_M and V_M . We call the maximizers $s = s_1$ and $s' = s'_1 \in \mathbb{R}^d$ in (31) the principal vectors (Dahlquist and Björck, 2008). By (32), the principal vectors are

$$s_1 = U g_1 = U u_{M,1}, \quad (43)$$

$$s'_1 = U' g'_1 = U' v_{M,1}. \quad (44)$$

Next, the second principal angle θ_2 is defined by

$$\cos \theta_2 = \max_{\substack{s \in \mathcal{S}, \|s\|=1, s^\top s_1=0 \\ s' \in \mathcal{S}', \|s'\|=1, s'^\top s'_1=0}} s^\top s'. \quad (45)$$

From (32) we have $s = Ug$, and from (43) we have $s_1 = Ug_1$; therefore,

$$\begin{aligned} s^\top s_1 &= (Ug)^\top Ug_1 \\ &= g^\top U^\top Ug_1 = g^\top g_1. \end{aligned} \quad (46)$$

Similarly, $s'^\top s'_1 = g'^\top g'_1$. Combining these with $\|g\| = \|g'\| = 1$ and (34), we can rewrite (45) as

$$\cos \theta_2 = \max_{\substack{\|g\|=1, g^\top g_1=0 \\ \|g'\|=1, g'^\top g'_1=0}} g^\top M g'. \quad (47)$$

Moreover, from (37) and (41),

$$\begin{aligned} g^\top g_1 &= (U_M \xi)^\top U_M \xi_1 = \xi^\top U_M^\top U_M \xi_1 \\ &= \xi^\top \xi_1 = \xi^\top e_1^{\text{Std}} = \xi_1. \end{aligned} \quad (48)$$

Similarly, $g'^\top g'_1 = \xi'_1$. Using $\|\xi\| = \|\xi'\| = 1$, (38), and (48), we can rewrite (47) as

$$\begin{aligned} \cos \theta_2 &= \max_{\substack{\|\xi\|=1, \xi_1=0 \\ \|\xi'\|=1, \xi'_1=0}} \xi^\top \Sigma_M \xi' \\ &= \max_{\substack{\|\xi\|=1, \xi_1=0 \\ \|\xi'\|=1, \xi'_1=0}} \sum_{j=2}^m \sigma_j \xi_j \xi'_j. \end{aligned} \quad (49)$$

By the same argument as in (39), the maximum is attained at $\xi = \xi' = e_2^{\text{Std}}$, and we obtain $\cos \theta_2 = \sigma_2$. Repeating the same steps for the i -th principal angle¹⁹ yields

$$\cos \theta_i = \sigma_i. \quad (50)$$

¹⁹It is known that the i -th canonical correlation is equal to the cosine of the i -th principal angle between the column spaces of the data matrices (Björck and Golub, 1973). We verify this correspondence below. Let the data matrices be $\mathbf{X} \in \mathbb{R}^{n \times p}$ and $\mathbf{Y} \in \mathbb{R}^{n \times q}$, and assume that their columns are centered. Define

$$\mathbf{S}_{XX} = \frac{1}{n} \mathbf{X}^\top \mathbf{X}, \quad \mathbf{S}_{YY} = \frac{1}{n} \mathbf{Y}^\top \mathbf{Y}, \quad \mathbf{S}_{XY} = \frac{1}{n} \mathbf{X}^\top \mathbf{Y}.$$

The i -th canonical correlation is defined as

$$\rho_i := \max_{\substack{\mathbf{a}^\top \mathbf{S}_{XX} \mathbf{a}=1, \mathbf{a}^\top \mathbf{S}_{XX} \mathbf{a}_1=\dots=\mathbf{a}^\top \mathbf{S}_{XX} \mathbf{a}_{i-1}=0, \\ \mathbf{b}^\top \mathbf{S}_{YY} \mathbf{b}=1, \mathbf{b}^\top \mathbf{S}_{YY} \mathbf{b}_1=\dots=\mathbf{b}^\top \mathbf{S}_{YY} \mathbf{b}_{i-1}=0}} \mathbf{a}^\top \mathbf{S}_{XY} \mathbf{b},$$

where $\mathbf{a}_i \in \mathbb{R}^p$ and $\mathbf{b}_i \in \mathbb{R}^q$ are the vectors that attain the i -th canonical correlation. We assume $\text{rank}(\mathbf{X}) = p$ and $\text{rank}(\mathbf{Y}) = q$. For rank-deficient matrices, it suffices to restrict the analysis to the subspaces corresponding to nonzero singular values.

Let the SVDs of $\mathbf{X}$ and $\mathbf{Y}$ be

$$\mathbf{X} = \mathbf{U}_X \Sigma_X \mathbf{V}_X^\top, \quad \mathbf{Y} = \mathbf{U}_Y \Sigma_Y \mathbf{V}_Y^\top,$$

Moreover, as in (43) and (44), if $\mathbf{S}, \mathbf{S}' \in \mathbb{R}^{d \times m}$ are matrices whose columns are the principal vectors, then

$$\mathbf{S} = \mathbf{U} \mathbf{U}_M, \quad \mathbf{S}' = \mathbf{U}' \mathbf{V}_M. \quad (51)$$

D.2 Deriving PK using orthonormal bases

For $\text{PK}(\mathcal{S}, \mathcal{S}')$ in (8), using $\cos \theta_i = \sigma_i$ in (50), we can derive $\text{PK}(\mathcal{S}, \mathcal{S}') = \|\mathbf{U}^\top \mathbf{U}'\|_F^2$ in (9) as follows:

$$\begin{aligned} \text{PK}(\mathcal{S}, \mathcal{S}') &= \sum_{i=1}^m \cos^2 \theta_i = \sum_{i=1}^m \sigma_i^2 = \text{tr}(\Sigma_M^\top \Sigma_M) \\ &= \text{tr}(\mathbf{V}_M^\top \mathbf{V}_M \Sigma_M^\top \mathbf{U}_M^\top \mathbf{U}_M \Sigma_M) \\ &= \text{tr}(\mathbf{V}_M \Sigma_M^\top \mathbf{U}_M^\top \mathbf{U}_M \Sigma_M \mathbf{V}_M^\top) \\ &= \text{tr}\left(\left(\mathbf{U}_M \Sigma_M \mathbf{V}_M^\top\right)^\top \mathbf{U}_M \Sigma_M \mathbf{V}_M^\top\right) \\ &= \text{tr}(\mathbf{M}^\top \mathbf{M}) = \|\mathbf{M}\|_F^2 = \|\mathbf{U}^\top \mathbf{U}'\|_F^2. \end{aligned} \quad (52)$$

D.3 Deriving PK using orthogonal projections

For $\text{PK}(\mathcal{S}, \mathcal{S}')$ in (8), using $\text{PK}(\mathcal{S}, \mathcal{S}') = \|\mathbf{U}^\top \mathbf{U}'\|_F^2$ in (52), we can derive $\text{PK}(\mathcal{S}, \mathcal{S}') = \text{tr}(\mathbf{P} \mathbf{P}')$ in (9) as follows:

$$\begin{aligned} \text{PK}(\mathcal{S}, \mathcal{S}') &= \|\mathbf{U}^\top \mathbf{U}'\|_F^2 = \|\mathbf{M}\|_F^2 = \text{tr}(\mathbf{M}^\top \mathbf{M}) \\ &= \text{tr}\left((\mathbf{U}^\top \mathbf{U}')^\top \mathbf{U}^\top \mathbf{U}'\right) = \text{tr}(\mathbf{U}'^\top \mathbf{U} \mathbf{U}^\top \mathbf{U}') \\ &= \text{tr}(\mathbf{U} \mathbf{U}^\top \mathbf{U}' \mathbf{U}'^\top) = \text{tr}(\mathbf{P} \mathbf{P}'). \end{aligned} \quad (53)$$

where $\mathbf{U}_X \in \mathbb{R}^{n \times p}$, $\Sigma_X \in \mathbb{R}^{p \times p}$, $\mathbf{V}_X \in \mathbb{R}^{p \times p}$, $\mathbf{U}_Y \in \mathbb{R}^{n \times q}$, $\Sigma_Y \in \mathbb{R}^{q \times q}$, and $\mathbf{V}_Y \in \mathbb{R}^{q \times q}$. Under the change of variables

$$\begin{aligned} \mathbf{a} &= \sqrt{n} \mathbf{V}_X \Sigma_X^{-1} \boldsymbol{\alpha}, & \mathbf{a}_j &= \sqrt{n} \mathbf{V}_X \Sigma_X^{-1} \alpha_j, \\ \mathbf{b} &= \sqrt{n} \mathbf{V}_Y \Sigma_Y^{-1} \boldsymbol{\beta}, & \mathbf{b}_j &= \sqrt{n} \mathbf{V}_Y \Sigma_Y^{-1} \beta_j, \end{aligned}$$

we have

$$\begin{aligned} \mathbf{a}^\top \mathbf{S}_{XX} \mathbf{a} &= \|\boldsymbol{\alpha}\|^2, & \mathbf{a}^\top \mathbf{S}_{XX} \mathbf{a}_j &= \alpha^\top \alpha_j, \\ \mathbf{b}^\top \mathbf{S}_{YY} \mathbf{b} &= \|\boldsymbol{\beta}\|^2, & \mathbf{b}^\top \mathbf{S}_{YY} \mathbf{b}_j &= \beta^\top \beta_j, \end{aligned}$$

and

$$\mathbf{a}^\top \mathbf{S}_{XY} \mathbf{b} = \boldsymbol{\alpha}^\top \mathbf{U}_X^\top \mathbf{U}_Y \boldsymbol{\beta}.$$

Therefore,

$$\rho_i = \max_{\substack{\|\boldsymbol{\alpha}\|=1, \boldsymbol{\alpha} \perp \boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_{i-1}, \\ \|\boldsymbol{\beta}\|=1, \boldsymbol{\beta} \perp \boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_{i-1}}} \boldsymbol{\alpha}^\top \mathbf{U}_X^\top \mathbf{U}_Y \boldsymbol{\beta}.$$

Since $\mathbf{U}_X$ and $\mathbf{U}_Y$ are orthonormal bases of $\text{col}(\mathbf{X})$ and $\text{col}(\mathbf{Y})$, respectively, the right-hand side is equal to the cosine of the i -th principal angle.

Furthermore, let $\mathcal{S}_\perp, \mathcal{S}'_\perp \subset \mathbb{R}^d$ denote the orthogonal complements of $\mathcal{S}$ and $\mathcal{S}'$, respectively. Then, $\text{PK}(\mathcal{S}_\perp, \mathcal{S}'_\perp)$ can be readily computed from $\text{PK}(\mathcal{S}, \mathcal{S}')$. Since the orthogonal projection matrices onto $\mathcal{S}_\perp$ and $\mathcal{S}'_\perp$ are $\mathbf{I}_d - \mathbf{P}$ and $\mathbf{I}_d - \mathbf{P}' \in \mathbb{R}^{d \times d}$, respectively, we have

$$\begin{aligned} & \text{PK}(\mathcal{S}_\perp, \mathcal{S}'_\perp) \\ &= \text{tr}((\mathbf{I}_d - \mathbf{P})(\mathbf{I}_d - \mathbf{P}')) \\ &= \text{tr}(\mathbf{I}_d) - \text{tr}(\mathbf{P}) - \text{tr}(\mathbf{P}') + \text{tr}(\mathbf{P}\mathbf{P}') \\ &= d - 2d_{\text{head}} + \text{PK}(\mathcal{S}, \mathcal{S}'). \end{aligned} \quad (54)$$

Here, we used the result²⁰ $\text{tr}(\mathbf{P}) = \text{tr}(\mathbf{P}') = d_{\text{head}}$.

D.4 Relationship between PK and CS in (10)

Based on the definition of CS in (3), we show (10).

First, for the orthogonal projection matrix $\mathbf{P} = \mathbf{U}\mathbf{U}^\top$ onto $\mathcal{S}$, we have

$$\begin{aligned} \|\mathbf{P}\|_{\text{F}}^2 &= \text{tr}(\mathbf{P}^\top \mathbf{P}) = \text{tr}((\mathbf{U}\mathbf{U}^\top)^\top \mathbf{U}\mathbf{U}^\top) \\ &= \text{tr}(\mathbf{U}\mathbf{U}^\top \mathbf{U}\mathbf{U}^\top) = \text{tr}(\mathbf{U}\mathbf{I}_m \mathbf{U}^\top) \\ &= \text{tr}(\mathbf{U}^\top \mathbf{U}) = \text{tr}(\mathbf{I}_m) = m. \end{aligned} \quad (55)$$

Here we used $\mathbf{U}^\top \mathbf{U} = \mathbf{I}_m$ since $\mathbf{U}$ is an orthonormal basis matrix of $\mathcal{S}$. Similarly, for the projection matrix $\mathbf{P}'$ onto $\mathcal{S}'$, we have $\|\mathbf{P}'\|_{\text{F}}^2 = m$.

Next, for $\mathbf{P}$ and $\mathbf{P}'$,

$$\begin{aligned} \|\mathbf{P}'\mathbf{P}\|_{\text{F}}^2 &= \text{tr}((\mathbf{P}'\mathbf{P})^\top \mathbf{P}'\mathbf{P}) = \text{tr}(\mathbf{P}^\top \mathbf{P}'^\top \mathbf{P}'\mathbf{P}) \\ &= \text{tr}(\mathbf{P}\mathbf{P}'\mathbf{P}'\mathbf{P}) = \text{tr}(\mathbf{P}\mathbf{P}\mathbf{P}'\mathbf{P}') \\ &= \text{tr}(\mathbf{P}\mathbf{P}') = \text{PK}(\mathcal{S}, \mathcal{S}') \quad (\because (53)). \end{aligned} \quad (56)$$

Here we used $\mathbf{P}^\top = \mathbf{P}$ and $\mathbf{P}^2 = \mathbf{P}$ for an orthogonal projection matrix $\mathbf{P}$, and the same holds for $\mathbf{P}'$.

²⁰Let $\mathbf{U} \in \mathbb{R}^{d \times d_{\text{head}}}$ and $\mathbf{U}_\perp \in \mathbb{R}^{d \times (d - d_{\text{head}})}$ be orthonormal bases of the d_{head} -dimensional subspace $\mathcal{S}$ and the $(d - d_{\text{head}})$ dimensional subspace $\mathcal{S}_\perp$, respectively. The orthogonal matrix $\mathbf{\Upsilon} = [\mathbf{U}, \mathbf{U}_\perp] \in \mathbb{R}^{d \times d}$ satisfies $\mathbf{P}\mathbf{\Upsilon} = \mathbf{U}\mathbf{U}^\top [\mathbf{U}, \mathbf{U}_\perp] = [\mathbf{U}, \mathbf{O}_{d \times (d - d_{\text{head}})}]$. Therefore,

$$\begin{aligned} \mathbf{\Upsilon}^\top \mathbf{P}\mathbf{\Upsilon} &= \begin{bmatrix} \mathbf{U}^\top \\ (\mathbf{U}_\perp)^\top \end{bmatrix} [\mathbf{U} \quad \mathbf{O}_{d \times (d - d_{\text{head}})}] \\ &= \begin{bmatrix} \mathbf{U}^\top \mathbf{U} & \mathbf{O}_{d_{\text{head}} \times (d - d_{\text{head}})} \\ (\mathbf{U}_\perp)^\top \mathbf{U} & \mathbf{O}_{(d - d_{\text{head}}) \times (d - d_{\text{head}})} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_{d_{\text{head}}} & \mathbf{O}_{d_{\text{head}} \times (d - d_{\text{head}})} \\ \mathbf{O}_{(d - d_{\text{head}}) \times d_{\text{head}}} & \mathbf{O}_{(d - d_{\text{head}}) \times (d - d_{\text{head}})} \end{bmatrix}. \end{aligned}$$

Thus, $\text{tr}(\mathbf{P}) = \text{tr}(\mathbf{P}\mathbf{\Upsilon}\mathbf{\Upsilon}^\top) = \text{tr}(\mathbf{\Upsilon}^\top \mathbf{P}\mathbf{\Upsilon}) = \text{tr}(\mathbf{I}_{d_{\text{head}}}) = d_{\text{head}}$. The same argument applies to $\mathbf{P}'$.

By (55) and (56),

$$\text{CS}(\mathcal{S}', \mathcal{S})^2 = \frac{\|\mathbf{P}'\mathbf{P}\|_{\text{F}}^2}{\|\mathbf{P}'\|_{\text{F}}^2 \|\mathbf{P}\|_{\text{F}}^2} = \frac{\text{PK}(\mathcal{S}, \mathcal{S}')}{m^2}, \quad (57)$$

which proves (10).

D.5 Rotation invariance of PK

Let $\mathbf{\Psi} \in \mathbb{R}^{d \times d}$ be an orthogonal matrix, and let $\mathcal{S}_\Psi$ and $\mathcal{S}'_\Psi$ be the subspaces spanned by $\mathbf{\Psi}\mathbf{U}$ and $\mathbf{\Psi}\mathbf{U}'$. The PK between $\mathcal{S}_\Psi$ and $\mathcal{S}'_\Psi$ satisfies

$$\begin{aligned} \text{PK}(\mathcal{S}_\Psi, \mathcal{S}'_\Psi) &= \text{tr}(\mathbf{\Psi}\mathbf{U}(\mathbf{\Psi}\mathbf{U})^\top \mathbf{\Psi}\mathbf{U}'(\mathbf{\Psi}\mathbf{U}')^\top) \\ &= \text{tr}(\mathbf{\Psi}\mathbf{U}\mathbf{U}^\top \mathbf{\Psi}^\top \mathbf{\Psi}\mathbf{U}'\mathbf{U}'^\top \mathbf{\Psi}^\top) \\ &= \text{tr}(\mathbf{U}\mathbf{U}^\top \mathbf{U}'\mathbf{U}'^\top) = \text{tr}(\mathbf{P}\mathbf{P}') \\ &= \text{PK}(\mathcal{S}, \mathcal{S}') \quad (\because (53)). \end{aligned} \quad (58)$$

Therefore, PK is unchanged by left multiplication of $\mathbf{U}$ and $\mathbf{U}'$ by any orthogonal matrix $\mathbf{\Psi} \in \mathbb{R}^{d \times d}$.

In Appendix M, leveraging this rotation invariance, we simplify the setting of the PK distribution between random orthogonal matrices and derive an approximate distribution of PK.

E Computing PK from the matrices used for CS with a fixed weight pairing type

For a fixed weight pairing type, we derive the subspaces required to compute PK from the matrices used to compute CS, and show that the PK between these subspaces depends only on their directions. We then examine the squared inner products that constitute the summands of PK and squared CS, where PK has no direction-dependent scaling whereas CS does.

E.1 Deriving PK from the matrices used to compute CS

We first derive the subspaces $\mathcal{S}'$ and $\mathcal{S}$ spanned by the matrices $\mathbf{W}^{r'}, \mathbf{W}^r \in \mathcal{W} \subset \mathbb{R}^{d \times d_{\text{head}}}$ used to compute PK, starting from the matrices $\mathbf{W}^{h'}, \mathbf{W}^h \in \mathbb{R}^{d \times d}$ used to compute CS in (3). We then compute $\text{PK}(\mathcal{S}', \mathcal{S})$ in (8) between these subspaces²¹.

E.1.1 Example setting for OQ

As an example, consider OQ as the weight pairing type. From (4), the matrices $\mathbf{W}^{h'}, \mathbf{W}^h \in \mathbb{R}^{d \times d}$

²¹From (9), $\text{PK}(\mathcal{S}, \mathcal{S}') = \text{PK}(\mathcal{S}', \mathcal{S})$. For simplicity, we consider $\text{PK}(\mathcal{S}', \mathcal{S})$ throughout this section.

used to compute CS in (3) are

$$\mathbf{W}^{h'} = \mathbf{W}_{\text{QK}}^{h'\top} = \mathbf{W}_{\text{K}}^{h'\top} \mathbf{W}_{\text{Q}}^{h'}, \quad (59)$$

$$\mathbf{W}^h = \mathbf{W}_{\text{OV}}^h = \mathbf{W}_{\text{O}}^h \mathbf{W}_{\text{V}}^h. \quad (60)$$

In contrast, the matrices $\mathbf{W}^{r'}, \mathbf{W}^r \in \mathcal{W} \subset \mathbb{R}^{d \times d_{\text{head}}}$ used to compute PK are

$$\mathbf{W}^{r'} = \mathbf{W}_{\text{Q}}^{h'\top}, \quad (61)$$

$$\mathbf{W}^r = \mathbf{W}_{\text{O}}^h. \quad (62)$$

E.1.2 Deriving the subspaces used to compute PK for OQ

Let $\text{col}(\mathbf{A})$ denote the subspace spanned by the columns of $\mathbf{A}$ and $\text{row}(\mathbf{A})$ denote the subspace spanned by its rows. From the SVDs in (23) and (24), for $\mathcal{S}' = \text{col}(\mathbf{W}^{r'})$, we have²²

$$\begin{aligned} \text{col}(\mathbf{W}^{r'}) &= \text{col}(\mathbf{W}_{\text{Q}}^{h'\top}) = \text{row}(\mathbf{W}_{\text{Q}}^{h'}) \\ &= \text{row}(\mathbf{W}_{\text{QK}}^{h'\top}) = \text{row}(\mathbf{W}^{h'}) \\ &= \text{row}(\mathbf{V}^{h'\top}) = \text{col}(\mathbf{V}^{h'}). \end{aligned} \quad (63)$$

Similarly, for $\mathcal{S} = \text{col}(\mathbf{W}^r)$,

$$\begin{aligned} \text{col}(\mathbf{W}^r) &= \text{col}(\mathbf{W}_{\text{O}}^h) \\ &= \text{col}(\mathbf{W}_{\text{O}}^h \mathbf{W}_{\text{V}}^h) = \text{col}(\mathbf{W}_{\text{OV}}^h) \\ &= \text{col}(\mathbf{W}^h) = \text{col}(\mathbf{U}^h). \end{aligned} \quad (64)$$

Thus, from (63) and (64), for OQ, the subspaces used to compute PK can be derived from the matrices used to compute CS.

E.1.3 Computing PK between the derived subspaces

The same argument applies to the other weight pairing types, yielding

$$\text{col}(\mathbf{W}^{r'}) = \text{col}(\mathbf{V}^{h'}), \quad (65)$$

$$\text{col}(\mathbf{W}^r) = \text{col}(\mathbf{U}^h). \quad (66)$$

²²The equality $\text{row}(\mathbf{W}_{\text{Q}}^{h'}) = \text{row}(\mathbf{W}_{\text{QK}}^{h'\top})$ follows from the fact that $\mathbf{W}_{\text{K}}^{h'\top} \in \mathbb{R}^{d \times d_{\text{head}}}$ has full column rank. Indeed, since $\mathbf{W}_{\text{QK}}^{h'\top} = \mathbf{W}_{\text{K}}^{h'\top} \mathbf{W}_{\text{Q}}^{h'}$, we have $\text{row}(\mathbf{W}_{\text{QK}}^{h'\top}) \subseteq \text{row}(\mathbf{W}_{\text{Q}}^{h'})$. Conversely, because $\mathbf{W}_{\text{K}}^{h'\top}$ has full column rank, there exists a left inverse $\mathbf{L}$ satisfying $\mathbf{L} \mathbf{W}_{\text{K}}^{h'\top} = \mathbf{I}_{d_{\text{head}}}$. Therefore, $\mathbf{W}_{\text{Q}}^{h'} = \mathbf{L} \mathbf{W}_{\text{QK}}^{h'\top}$, which gives $\text{row}(\mathbf{W}_{\text{Q}}^{h'}) \subseteq \text{row}(\mathbf{W}_{\text{QK}}^{h'\top})$. Thus, the two row spaces are identical. Similarly, from (24), since $\mathbf{U}^{h'} \Sigma^{h'}$ has full column rank, we have $\text{row}(\mathbf{W}^{h'}) = \text{row}(\mathbf{V}^{h'\top})$.

Therefore,

$$\begin{aligned} \text{PK}(\mathcal{S}', \mathcal{S}) &= \text{PK}(\text{col}(\mathbf{V}^{h'}), \text{col}(\mathbf{U}^h)) \\ &= \left\| \mathbf{V}^{h'\top} \mathbf{U}^h \right\|_{\text{F}}^2 \\ &= \sum_{i=1}^{d_{\text{head}}} \sum_{j=1}^{d_{\text{head}}} \left(\mathbf{v}_i^{h'\top} \mathbf{u}_j^h \right)^2. \end{aligned} \quad (67)$$

Unlike the squared CS in (25), which scales the singular vectors in proportion to their singular values, PK in (67) can be interpreted as a score that depends only on the directions of the singular vectors.

E.2 Comparison of summands in PK and squared CS

In this section, we examine the summands of PK and squared CS using empirical data and observe the effect of direction-dependent scaling. Specifically, we focus on the summand of PK in (67),

$$\left(\mathbf{v}_i^{h'\top} \mathbf{u}_j^h \right)^2, \quad (68)$$

and the summand of squared CS in (25),

$$\left\{ \left(\frac{\sigma_i^{h'} \mathbf{v}_i^{h'}}{\sqrt{\sum_{i'=1}^{d_{\text{head}}} \sigma_{i'}^{h'^2}}} \right)^\top \left(\frac{\sigma_j^h \mathbf{u}_j^h}{\sqrt{\sum_{j'=1}^{d_{\text{head}}} \sigma_{j'}^{h^2}}} \right) \right\}^2. \quad (69)$$

We first plot the singular values of the QK and OV matrices for all 144 heads in GPT2-small in Figs. 11 and 12 to examine their overall trends. The scales of the singular values vary substantially across layers and heads. In particular, some heads have large singular values in early layers for the QK matrices and in late layers for the OV matrices²³.

To select head pairs for comparing the summands, we follow the same procedure as for the wiring diagrams in Section 6 and select the top five head pairs according to PK and CS for each of OQ, OK, and OV. For these pairs, we sort the summands of PK in (68) and squared CS in (69) in descending order and plot their values and the cumulative fraction of their sums in Figs. 13 and 14, respectively. The horizontal axis is shown on a logarithmic scale.

For the summands of squared CS in (69), which include direction-dependent scaling by the singular values, the summands at higher ranks, typically

²³Since $\|\mathbf{W}^h\|_{\text{F}} = \|\Sigma^h\|_{\text{F}}$, the tendency for the norms of the QK matrices to be large in layer 0 and gradually decrease with depth, while those of the OV matrices gradually increase with depth, is consistent with Fig. 9.

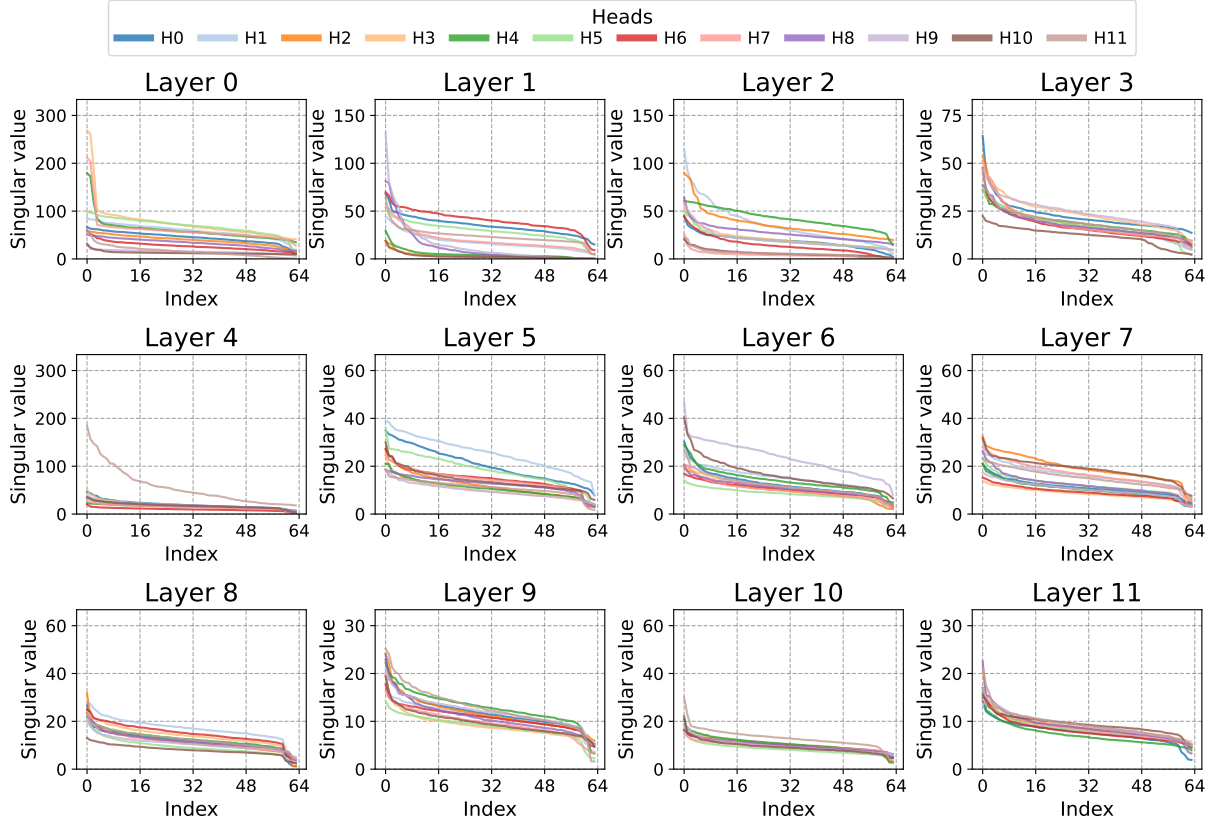


Figure 11: Singular values of GPT2-small QK matrices W_{QK}^h .

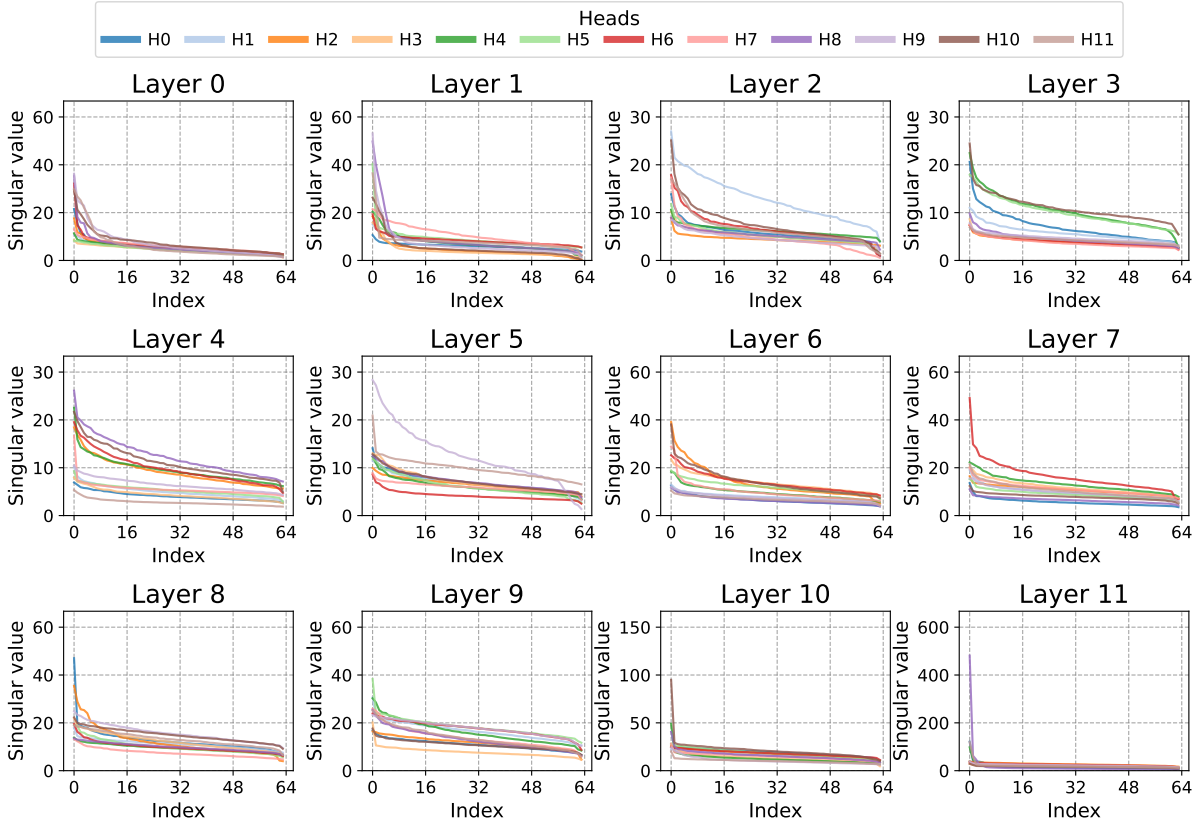


Figure 12: Singular values of GPT2-small OV matrices W_{OV}^h .

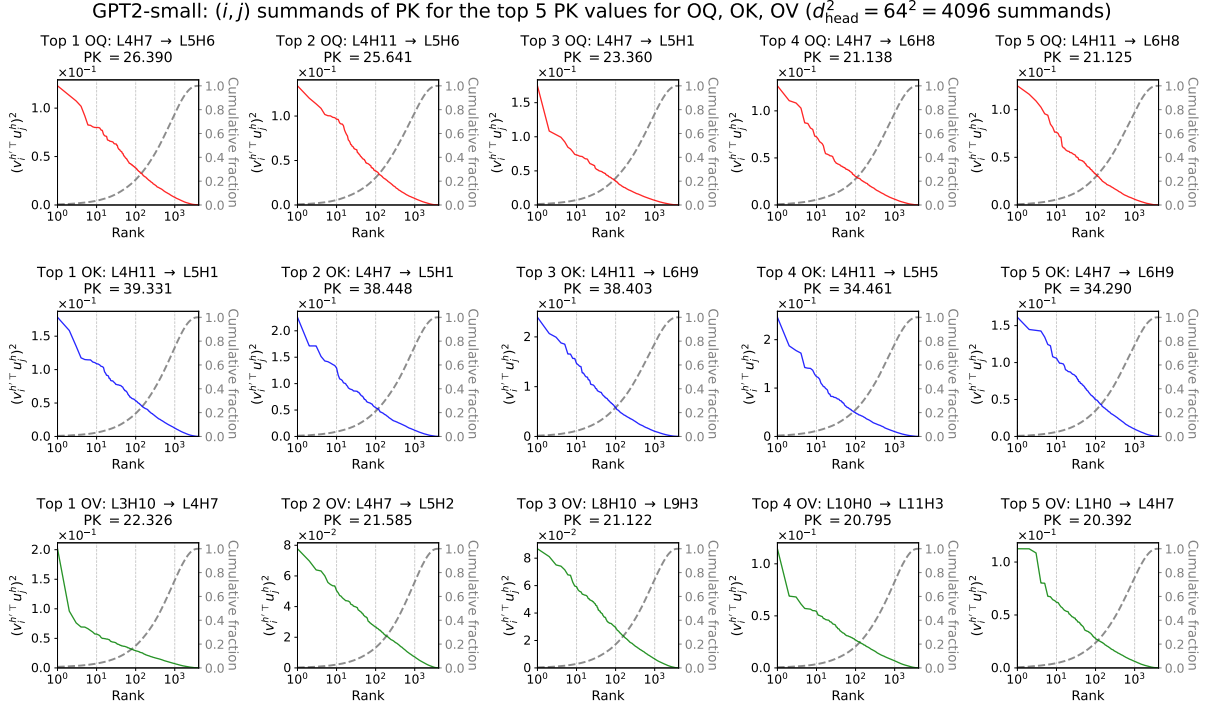


Figure 13: Summands of PK for the top 5 PK head pairs for each of OQ, OK, and OV (rows) in GPT2-small; the columns correspond to the top five head pairs. Each panel shows the $d_{\text{head}}^2 = 64^2 = 4096$ summands $(\mathbf{v}_i^{h'} \mathbf{u}_j^h)^2$, which add up to the PK value. Here $\mathbf{v}_i^{h'}$ and $\mathbf{u}_j^h$ are the leading d_{head} right and left singular vectors of the matrices used to compute the corresponding CS. The summands are sorted in descending order (solid line, left axis), with their cumulative fraction of the total shown by the dashed line (right axis).

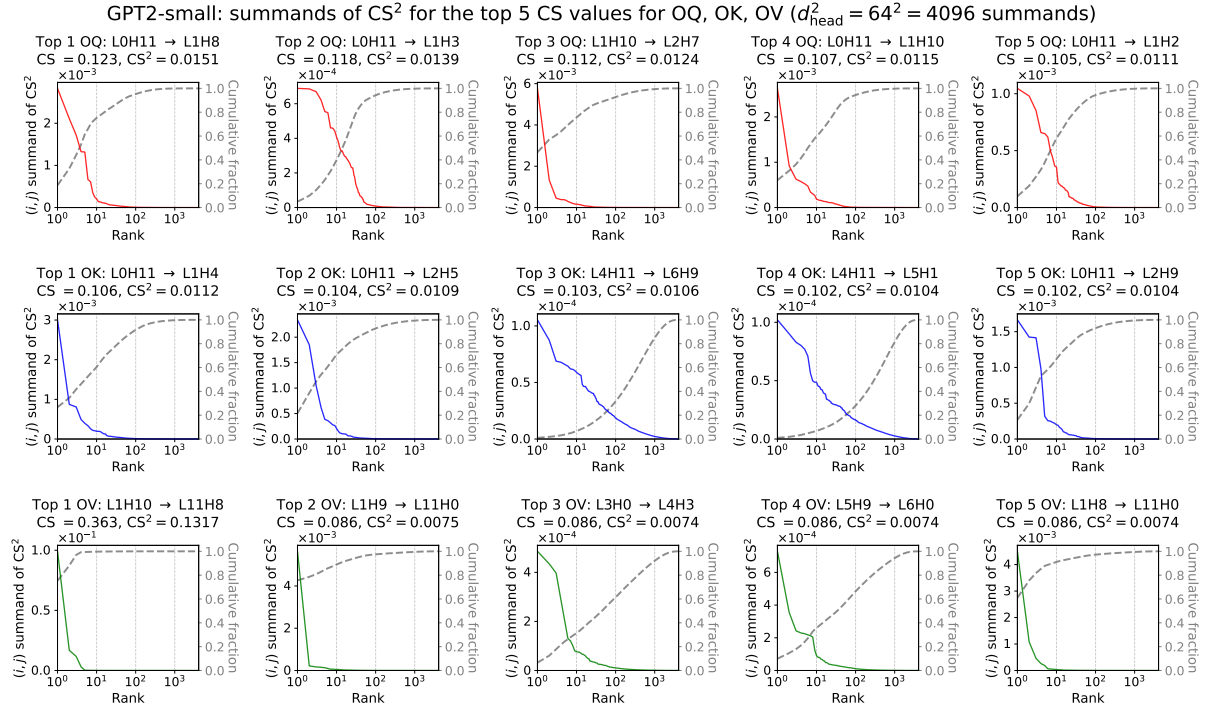


Figure 14: Summands of squared CS for the top 5 CS head pairs for each of OQ, OK, and OV (rows) in GPT2-small; the columns correspond to the top five head pairs. Each panel shows the $d_{\text{head}}^2 = 64^2 = 4096$ summands $\{(\sigma_i^{h'} \mathbf{v}_i^{h'} / \|\Sigma^{h'}\|_F)^\top (\sigma_j^h \mathbf{u}_j^h / \|\Sigma^h\|_F)\}^2$. Here $\sigma_i^{h'}$, $\mathbf{v}_i^{h'}$ and σ_j^h , $\mathbf{u}_j^h$ are the singular values and the right and left singular vectors of the source and target matrices of CS, respectively. The summands are sorted in descending order (solid line, left axis), with their cumulative fraction of the total shown by the dashed line (right axis).

within the first 10^1 to 10^2 ranks, are large for many head pairs, whereas those at lower ranks become nearly negligible. Consequently, a small number of summands at higher ranks account for most of the cumulative fraction²⁴. In contrast, the summands of PK in (68), which do not include such scaling, decrease more gradually than those of squared CS. Consequently, the cumulative fraction also increases more gradually. These results show that CS strongly reflects the influence of a small number of directions associated with large singular values, whereas PK treats the directions of the singular vectors uniformly.

F Details of the wiring diagram for a two-layer attention-only Transformer

In Fig. 4 in Section 3.5, using the attn-only-21 model in TransformerLens (Nanda and Bloom, 2022) (two layers with eight heads per layer), we showed that PK captures the relationship between Previous Token Heads and Induction Heads more clearly than CS. In this section, we briefly describe how we constructed Fig. 4.

For visibility, following Elhage et al. (2021), we subtract a random bias from the similarity computed for each weight pairing. This is because PK and CS can be nonzero even for unrelated weights, which shifts the baseline. Specifically, we generate $8^2 = 64$ pairs of random matrices whose entries follow the standard normal distribution, estimate the bias as the mean similarity over these pairs, and subtract it²⁵. After subtraction, we clip values at 0 and normalize by the maximum value. In the figures, edge width and opacity are proportional to this normalized similarity.

²⁴For squared CS, however, the summands decrease more gradually for the third- and fourth-ranked OK pairs involving the QK matrices of L6H9 and L5H11, and for the third- and fourth-ranked OV pairs involving the OV matrices of L4H3 and L6H0. We attribute this to the more uniform singular values of QK matrices in the middle layers than in early layers, and of OV matrices in the middle layers than in late layers, as shown in Figs. 9, 11, and 12. Furthermore, Merullo et al. (2024) showed that CS can be high between specific heads that function in the IOI task, such as from S-Inhibition Heads to Name Mover Heads, even when only a small number of singular vectors are used.

²⁵Elhage et al. (2021) describe this step as *subtract off the empirical expected amount for random matrices of the same shapes*, so the exact distribution is unclear. However, our CS results (Fig. 4b) roughly reproduce their findings.

G Details of projections of the unembedding matrix in GPT2-small

In Section 4, to interpret heads in GPT2-small, we project preprocessed unembedding vectors onto the subspace $\mathcal{S}^r$ spanned by the weights $\widehat{\mathbf{W}}^r$ after the final LN, treat the norm of the projected vector as a logit, and interpret representative tokens. In particular, we use normalization after centering as preprocessing for the unembedding vectors. In this section, we first discuss properties of unembedding vectors, then describe the effect of normalization, and finally show why centering before normalization is necessary.

G.1 Tokens with lower generation probabilities (and lower interpretability) tend to have larger unembedding norms

We treat the norm of the projected vector as a logit. Therefore, for less interpretable tokens, it is preferable that the projected norm is small. We accordingly regard tokens with low generation probabilities as less interpretable. For this purpose, we need a tokenwise quantity that represents how easy a token is to generate, independent of the input prompt.

To define such a quantity, we first review the standard logit computation. In language models such as GPT2, logits are computed by applying layer normalization (LN) to the final layer output and then multiplying by the unembedding matrix. Let this final LN be LN_{final} . Using the scaling weights $\gamma_{\text{final}} \in \mathbb{R}^d$ and the bias $\beta_{\text{final}} \in \mathbb{R}^d$, we write

$$\text{LN}_{\text{final}}(\mathbf{x}) = \gamma_{\text{final}} \odot \frac{\mathbf{x} - \text{Mean}(\mathbf{x})\mathbf{1}_d}{\text{Std}(\mathbf{x})} + \beta_{\text{final}}, \quad (70)$$

where $\odot$ denotes the Hadamard product, $\mathbf{1}_d \in \mathbb{R}^d$ is the d -dimensional all-ones vector, and $\text{Mean}(\cdot)$ and $\text{Std}(\cdot)$ denote the mean and standard deviation.

Based on LN_{final} , we use the inner product between the unembedding vector $\mathbf{e}_t$ and the bias term β_{final} in (70), namely $\mathbf{e}_t^\top \beta_{\text{final}}$, as a token-specific statistic related to generation probabilities²⁶ (Nanda and Bloom, 2022). In fact, the logit for token t is computed as the inner product between the unembedding vector $\mathbf{e}_t$ and $\text{LN}_{\text{final}}(\mathbf{x})$

²⁶See https://transformerlensorg.github.io/TransformerLens/generated/demos/Main_Demo.html for details.

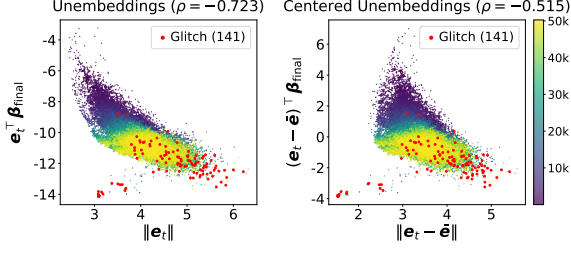


Figure 15: Scatter plots of the norm of unembedding vectors in GPT2-small and the inner product with the bias term β_{final} of the final layer normalization in (70), (left) without centering and (right) with centering. Since GPT2 uses byte pair encoding (BPE), the token ID can be viewed as a proxy for frequency, where larger token IDs tend to be less frequent (Hayase et al., 2024) (see Fig. 16). Without centering (left), the norm and the inner product with β_{final} show a strong negative correlation. With centering (right), the correlation becomes slightly weaker but the same trend remains.

in (70),

$$\begin{aligned} & e_t^\top \text{LN}_{\text{final}}(\mathbf{x}) \\ &= e_t^\top \left(\gamma_{\text{final}} \odot \frac{\mathbf{x} - \text{Mean}(\mathbf{x})\mathbf{1}_d}{\text{Std}(\mathbf{x})} \right) + e_t^\top \beta_{\text{final}}, \end{aligned} \quad (71)$$

where the first term on the right-hand side of (71) depends on the final-layer output $\mathbf{x}$, whereas the second term $e_t^\top \beta_{\text{final}}$ is an input-independent, token-specific statistic (i.e., a scalar bias term). In particular, tokens with more negative $e_t^\top \beta_{\text{final}}$ tend to have smaller probabilities under the softmax.

To understand the behavior of $e_t^\top \beta_{\text{final}}$, we examine its relationship with the unembedding norm. For comparison, we also consider centering the unembedding vectors using their mean $\bar{\mathbf{e}} \in \mathbb{R}^d$. Figure 15 shows scatter plots of the unembedding norm and the inner product, with and without centering. In Fig. 15, as examples of less interpretable tokens, we label glitch tokens (Rumbelow and Watkins, 2023; Watkins and Rumbelow, 2023), which are extremely low-frequency tokens whose embedding parameters are poorly trained²⁷. With or without centering, larger unembedding norms correspond to more negative inner products with β_{final} , suggesting smaller generation probabilities and lower interpretability. In fact, many tokens with large norms are low-frequency tokens, including glitch tokens²⁸.

²⁷We use the list of 141 glitch tokens provided by Rumbelow and Watkins (2023) in Table 6.

²⁸Note that some glitch tokens also have small norms.

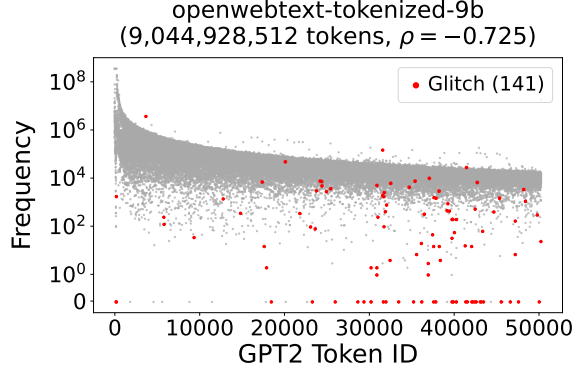


Figure 16: Scatter plot of GPT2 token IDs versus token frequencies in the OpenWebText Corpus (Gokaslan et al., 2019) (9B tokens). Despite differences between OpenWebText and the undisclosed GPT2 tokenizer training corpus, larger token IDs tend to be less frequent, consistent with Hayase et al. (2024) (Spearman’s $\rho = -0.725$). Since token frequencies vary across datasets, we use token ID as a dataset-agnostic proxy for frequency. Many of the 141 glitch tokens identified by Rumbelow and Watkins (2023) in Table 6 have zero frequency in OpenWebText.

G.2 Normalization before projection reduces projected norms for less interpretable tokens

We consider projecting preprocessed unembedding vectors onto the subspace $\tilde{\mathcal{S}}^r$ spanned by $\tilde{\mathbf{W}}^r$, where $\tilde{\mathbf{W}}^r$ is obtained by applying LN_{final} to $\mathbf{W}^r \in \mathcal{W}$. As candidates for preprocessing f , we consider the identity map, centering, normalization, and normalization after centering. Table 7 reports, for these four choices, the correlation between the norm of the (centered or uncentered) unembedding vector and the norm of its projection onto $\tilde{\mathcal{S}}^r$. Without normalization, the pre- and post-projection norms show a strong positive correlation, whereas with normalization we observe a weak negative correlation.

From Fig. 15 and Table 7, we obtain the following. Without normalization, less interpretable tokens that have large unembedding norms and strongly negative inner products with β_{final} tend to also have large projected norms. With normalization, such tokens instead tend to have small projected norms. Therefore, to treat the projected norm as a logit, normalization appears necessary.

G.3 Why centering before normalization is necessary

Since the inner product between an unembedding vector and β_{final} can be regarded as a token-specific

[illegible]

Table 6: The 141 glitch tokens in GPT2 identified by [Rumbelow and Watkins \(2023\)](#). Here, _ denotes a leading space, and \xNN a non-printable byte with hexadecimal code NN (e.g., \x00 for ID 188 and \x7F for ID 221).

$f(e_t)$	Unembed. Norm	Query	Key	Value	Output	All
e_t	$\ e_t\ $	0.750 ± 0.049	0.757 ± 0.045	0.745 ± 0.063	0.670 ± 0.091	0.731 ± 0.074
$e_t - \bar{e}$	$\ e_t - \bar{e}\ $	0.605 ± 0.084	0.625 ± 0.078	0.670 ± 0.108	0.547 ± 0.147	0.612 ± 0.116
$e_t / \ e_t\ $	$\ e_t\ $	-0.392 ± 0.092	-0.363 ± 0.086	-0.382 ± 0.127	-0.402 ± 0.145	-0.385 ± 0.116
$(e_t - \bar{e}) / \ e_t - \bar{e}\ $	$\ e_t - \bar{e}\ $	-0.198 ± 0.138	-0.168 ± 0.139	-0.054 ± 0.226	-0.163 ± 0.185	-0.146 ± 0.184

Table 7: For the 144 heads in GPT2-small, we report the mean and standard deviation of Spearman’s ρ between the unembedding norm and the norm of the vector projected onto $\tilde{S}^r$ after preprocessing f , separately for each weight type. We also report the results when ignoring weight types (All). Without normalization ($f(e_t) = e_t$ or $f(e_t) = e_t - \bar{e}$), the pre- and post-projection norms show a strong positive correlation. With normalization ($f(e_t) = e_t / \|e_t\|$ or $f(e_t) = (e_t - \bar{e}) / \|e_t - \bar{e}\|$), we instead observe a weak negative correlation.

statistic related to generation probabilities, regardless of whether we center²⁹, it is desirable that this inner product remains correlated even after normal-

²⁹The inner product between the centered unembedding vector $e_t - \bar{e}$ and β_{final} equals $e_t^\top \beta_{\text{final}} - \bar{e}^\top \beta_{\text{final}}$, and $\bar{e}^\top \beta_{\text{final}}$ is a token-independent constant. Moreover, when computing token probabilities, the softmax is invariant to adding the same constant shift to all logits, and thus is unaffected by centering the unembedding matrix. See Appendix O.1 for details.

izing the unembedding vectors. Figure 17 plots how the inner product with β_{final} changes before and after normalization, with and without centering. With normalization alone, the correlation almost vanishes. With normalization after centering, the inner products remain strongly correlated.

To analyze the effect of centering, we examine cosine similarities of unembedding vectors.

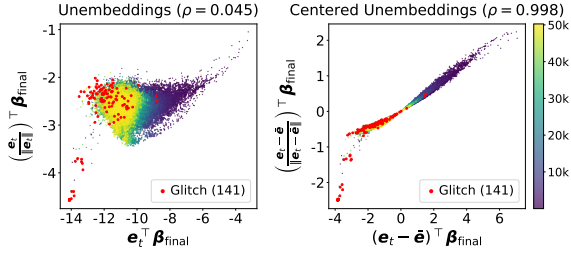


Figure 17: Scatter plots of the inner product with β_{final} before and after normalization, (left) without centering and (right) with centering. As in Fig. 15, we indicate token IDs as a proxy for frequency, where larger token IDs tend to be less frequent, and we label glitch tokens. Without centering (left), the inner products become nearly uncorrelated after normalization. With centering (right), the inner products remain strongly correlated before and after normalization.

The mean cosine similarity between e_t and the mean vector $\bar{e} \in \mathbb{R}^d$, namely $\cos(e_t, \bar{e})$, is 0.518. The mean cosine similarity with β_{final} , namely $\cos(e_t, \beta_{\text{final}})$, is -0.225 . These results³⁰ suggest that the unembedding vectors are anisotropic³¹ toward the direction of $\bar{e}$, and that this dominant direction is somewhat opposite to β_{final} .

Therefore, centering can be viewed as an operation analogous to all-but-the-top (Mu and Viswanath, 2018), which removes a dominant direction that is somewhat opposite to β_{final} . Based on these results³² and the above discussion, when interpreting top tokens of attention heads, we use unembedding vectors that are normalized after centering.

H Details of head scores and head class annotations in GPT2-small

In this section, we describe how we compute head scores using TransformerLens³³ (Nanda and Bloom, 2022) following Wang et al. (2023), and provide details of our head class annotations.

H.1 Computing head scores

We first describe how we construct the dataset. Specifically, we sample 100 tokens uniformly at

random from token IDs 0 to 20,000 to form a sentence³⁴, and repeat this sentence to obtain a length-200 token sequence. We create 128 such sequences, compute head scores for each sequence, and average them to obtain the final head scores.

For Identity Heads, we define the head score as the mean of the diagonal entries of the attention pattern. For example, in L0H3, which attains the largest Identity Head score and is shown in the right panel of Fig. 7, attention weights concentrate on the diagonal. For the Identity Head scores shown in the left panel of Fig. 7, Table 8a lists the top 10 Identity Heads. Except for L4H7, which we discussed in Section 8, these heads mainly appear in early layers.

To revisit the head class annotations, we recompute head scores for Duplicate Token, Previous Token, and Induction Heads following Wang et al. (2023, Appendix F), and report the results in Fig. 18. Duplicate Token Heads mainly appear in early layers, whereas Previous Token Heads mainly appear in middle layers, and Induction Heads span middle to late layers. Moreover, for each head class, the attention pattern of the top-scoring head captures the characteristic behavior of that class, which supports the validity of these scores. Tables 8b, 8c, and 8d list the top 10 heads and their head scores for each class. For Duplicate Token Heads, the top three heads have much larger scores than the rest, whereas for Previous Token and Induction Heads, the scores decrease more gradually. Based on these results, we next describe our head class annotations.

H.2 Head class annotations

We adopt the IOI task head class annotations from Wang et al. (2023, Fig. 17). Based on Appendix H.1, we revisit the annotations for Duplicate Token, Previous Token, and Induction Heads.

For Duplicate Token Heads, Wang et al. (2023) annotate L0H1, L0H10, and L3H0. Based on Table 8b, we replace L0H10, which does not appear in the table, with L0H5, which attains the largest score³⁵.

³⁰For reference, $\cos(\bar{e}, \beta_{\text{final}}) = -0.431$.

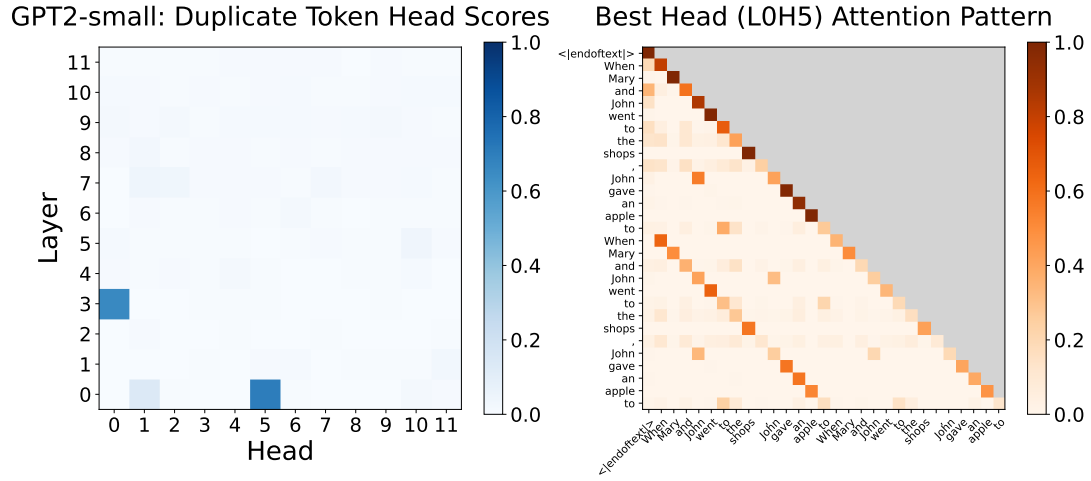
³¹Biś et al. (2021) link the anisotropy of the unembedding vectors to training dynamics induced by optimizing the softmax cross-entropy loss.

³²After centering, the mean cosine similarities of $e_t - \bar{e}$ with $\bar{e}$ and β_{final} are -0.005 and 0.001 , respectively.

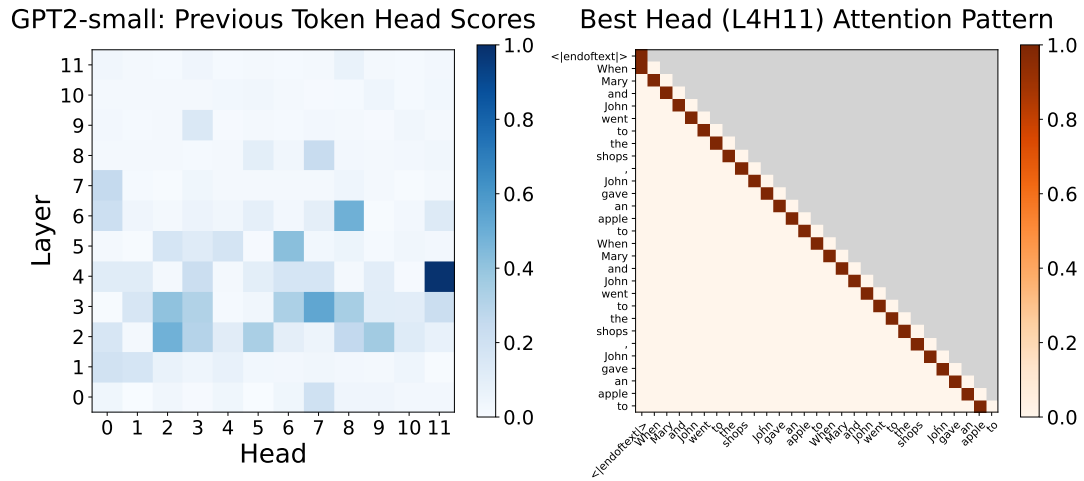
³³https://github.com/TransformerLensOrg/TransformerLens/blob/main/demos/Exploratory_Analysis_Demo.ipynb

³⁴TransformerLens (Nanda and Bloom, 2022) samples token IDs from 100 to 20,000. Since GPT2 has token IDs from 0 to 50,256 and higher IDs tend to be rarer (Hayase et al., 2024), we also restrict the range to 20,000. In addition, since token IDs from 0 to 100 include alphabetic characters and digits, we include 0 to 20,000 for simplicity.

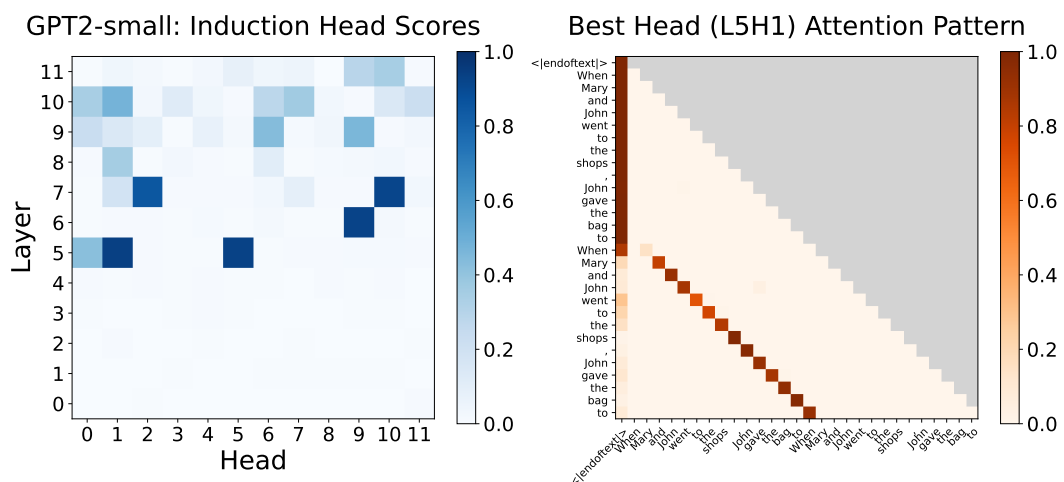
³⁵In fact, Wang et al. (2023, Fig. 12) show the head scores for Duplicate Token Heads, where L0H5 has a large score and L0H10 has a small score.



(a) Duplicate Token Heads



(b) Previous Token Heads



(c) Induction Heads

Figure 18: For Duplicate Token, Previous Token, and Induction Heads, we show (left) head scores for all heads in GPT2-small and (right) an example attention pattern for the head with the largest score. The head-score patterns broadly reproduce the results of Wang et al. (2023, Fig. 12). For each head class, the attention pattern of the top-scoring head captures the characteristic behavior of that class.

Head	Score	Head	Score	Head	Score	Head	Score
L0H3	0.966	L0H5	0.703	L4H11	0.988	L5H1	0.936
L0H1	0.918	L3H0	0.660	L3H7	0.533	L5H5	0.927
L1H11	0.900	L0H1	0.133	L6H8	0.490	L6H9	0.925
L0H4	0.826	L7H1	0.045	L2H2	0.485	L7H10	0.914
L0H5	0.639	L5H10	0.042	L5H6	0.422	L7H2	0.849
L4H7	0.358	L7H2	0.039	L3H2	0.408	L10H1	0.474
L0H10	0.235	L1H11	0.030	L2H9	0.362	L9H9	0.458
L3H2	0.197	L8H1	0.026	L3H8	0.344	L9H6	0.437
L3H6	0.192	L7H7	0.025	L2H5	0.339	L5H0	0.425
L3H0	0.163	L9H0	0.023	L3H6	0.330	L10H7	0.367
(a) Identity		(b) Duplicate Token		(c) Previous Token		(d) Induction	

Table 8: For Identity, Duplicate Token, Previous Token, and Induction Heads in GPT2-small, we list the top 10 heads and their head scores. See Appendix H.1 for how we compute these scores.

Head Classes	Count	Heads
Duplicate Token Heads	3	L0H1, L0H5*, L3H0
Previous Token Heads	10	L2H2, L2H5*, L2H9, L3H2*, L3H6*, L3H7*, L3H8*, L4H11, L5H6*, L6H8*
Induction Heads	6	L5H0*, L5H1*, L5H5, L6H9, L7H2*, L7H10*
Name Mover Heads	3	L9H6, L9H9, L10H0
Negative Name Mover Heads	2	L10H7, L11H10
Backup Name Mover Heads	8	L9H7, L10H1, L10H2, L10H6, L10H10, L11H2, L11H3, L11H9
S-Inhibition Heads	4	L7H3, L7H9, L8H6, L8H10

Table 9: Head classes and their corresponding heads in our experimental setup for GPT2-small. Heads that are not included in Wang et al. (2023, Fig. 17) are marked with “*”. The total number of annotated heads is 36.

As shown in Tables 8c and 8d, there are many more Previous Token Heads and Induction Heads than Duplicate Token Heads. We therefore assign the Previous Token Head or Induction Head label to unannotated heads among the top 10 heads for each of these two classes.

Table 9 lists the heads in each head class under our setup. All Previous Token Heads annotated by Wang et al. (2023), namely L2H2, L2H9, and L4H11, are included in Table 9. In contrast, among the Induction Heads annotated by Wang et al. (2023), namely L5H5, L5H8, L5H9, and L6H9, only L5H5 and L6H9 are included in Table 9.

I Details of baseline methods

In this section, we define and describe Simple-CS, Linear CKA (Kornblith et al., 2019), and Procrustes Similarity.

I.1 Simple-CS

We explain Simple-CS using the example of the weights associated with an output and a query. In CS_{OQ} in (4), we compute $\text{CS}(\mathbf{W}_{\text{QK}}^{h'\top}, \mathbf{W}_{\text{OV}}^h)$ using

$\mathbf{W}_{\text{OV}}^h, \mathbf{W}_{\text{QK}}^{h'\top} \in \mathbb{R}^{d \times d}$. Thus, even when measuring similarity between weights associated with an output and a query, CS uses the OV matrix $\mathbf{W}_{\text{OV}}^h$ as the output side and the QK matrix $\mathbf{W}_{\text{QK}}^{h'\top}$ as the query side. As a result, $\mathbf{W}_{\text{OV}}^h$ in (2) includes the value weight matrix $\mathbf{W}_{\text{V}}^h$, and $\mathbf{W}_{\text{QK}}^{h'\top}$ in (1) includes the key weight matrix $\mathbf{W}_{\text{K}}^{h'}$.

As a baseline, we therefore consider directly comparing the output weight $\mathbf{W}_{\text{O}}^h$ and the query weight $\mathbf{W}_{\text{Q}}^{h'\top}$ by computing $\text{CS}(\mathbf{W}_{\text{Q}}^{h'\top}, \mathbf{W}_{\text{O}}^h)$, and refer to this method as Simple-CS. That is, for $\mathbf{W}^r, \mathbf{W}^{r'} \in \mathcal{W}$, we define

$$\text{Simple-CS}(r, r') = \text{CS}(\mathbf{W}^{r'}, \mathbf{W}^r). \quad (72)$$

I.2 Linear CKA

We define Linear CKA following Kornblith et al. (2019). For $\mathbf{W}^r = (\mathbf{w}_1^r, \dots, \mathbf{w}_{d_{\text{head}}}^r) \in \mathcal{W} \subset \mathbb{R}^{d \times d_{\text{head}}}$, we define the columnwise mean vector as

$$\mathbf{w}_{\text{Mean}}^r = \frac{1}{d_{\text{head}}} \sum_{i=1}^{d_{\text{head}}} \mathbf{w}_i^r \in \mathbb{R}^d. \quad (73)$$

We then define the centered matrix $\mathbf{W}_c^r$ as³⁶

$$\mathbf{W}_c^r = \mathbf{W}^r - \mathbf{w}_{\text{Mean}}^r \mathbf{1}_{d_{\text{head}}}^\top \in \mathbb{R}^{d \times d_{\text{head}}}. \quad (74)$$

Similarly, for $\mathbf{W}^{r'} \in \mathcal{W}$ we define $\mathbf{W}_c^{r'}$. We define Linear CKA for $\mathbf{W}_c^r$ and $\mathbf{W}_c^{r'}$ as

$$\begin{aligned} & \text{Linear-CKA}(r, r') \\ &= \frac{\left\| \mathbf{W}_c^r \mathbf{W}_c^{r'}^\top \right\|_F^2}{\left\| \mathbf{W}_c^r \mathbf{W}_c^r^\top \right\|_F \left\| \mathbf{W}_c^{r'} \mathbf{W}_c^{r'}^\top \right\|_F}. \end{aligned} \quad (75)$$

To verify that $0 \leq \text{Linear-CKA}(r, r') \leq 1$, we note the following inequality³⁷

$$\begin{aligned} & \left\| \mathbf{W}_c^r \mathbf{W}_c^{r'}^\top \right\|_F^2 \\ &= \text{tr} \left(\left(\mathbf{W}_c^r \mathbf{W}_c^{r'}^\top \right)^\top \mathbf{W}_c^r \mathbf{W}_c^{r'}^\top \right) \\ &= \text{tr} \left(\mathbf{W}_c^{r'} \mathbf{W}_c^r^\top \mathbf{W}_c^r \mathbf{W}_c^{r'}^\top \right) \\ &= \text{tr} \left(\mathbf{W}_c^r \mathbf{W}_c^r^\top \mathbf{W}_c^r \mathbf{W}_c^{r'}^\top \mathbf{W}_c^{r'} \right) \\ &= \text{tr} \left(\left(\mathbf{W}_c^r \mathbf{W}_c^r^\top \right)^\top \mathbf{W}_c^{r'} \mathbf{W}_c^{r'}^\top \right) \\ &= \langle \mathbf{W}_c^r \mathbf{W}_c^r^\top, \mathbf{W}_c^{r'} \mathbf{W}_c^{r'}^\top \rangle_F \\ &\leq \left\| \mathbf{W}_c^r \mathbf{W}_c^r^\top \right\|_F \left\| \mathbf{W}_c^{r'} \mathbf{W}_c^{r'}^\top \right\|_F \\ &= \left\| \mathbf{W}_c^r \mathbf{W}_c^r^\top \right\|_F \left\| \mathbf{W}_c^{r'} \mathbf{W}_c^{r'}^\top \right\|_F, \end{aligned} \quad (76)$$

where the inequality follows from the Cauchy–Schwarz inequality for the Frobenius inner product, and equality holds if and only if $\mathbf{W}_c^r \mathbf{W}_c^r^\top = \lambda \mathbf{W}_c^{r'} \mathbf{W}_c^{r'}^\top$ for some $\lambda \geq 0$. Therefore, when $\left\| \mathbf{W}_c^r \mathbf{W}_c^r^\top \right\|_F \left\| \mathbf{W}_c^{r'} \mathbf{W}_c^{r'}^\top \right\|_F \neq 0$, we have

$$0 \leq \text{Linear-CKA}(r, r') \leq 1. \quad (77)$$

³⁶Using the centering matrix $\mathbf{C}_{d_{\text{head}}}$ in (126), we have $\mathbf{W}_c^r = \mathbf{W}^r \mathbf{C}_{d_{\text{head}}}$.

³⁷By the definition of the Frobenius norm and the cyclic invariance of the trace, we have

$$\begin{aligned} \left\| \mathbf{W}_c^r \mathbf{W}_c^r^\top \right\|_F^2 &= \text{tr} \left(\mathbf{W}_c^r \mathbf{W}_c^r^\top \mathbf{W}_c^r \mathbf{W}_c^r^\top \right) \\ &= \text{tr} \left(\mathbf{W}_c^r \mathbf{W}_c^r \mathbf{W}_c^r^\top \mathbf{W}_c^r \right) \\ &= \left\| \mathbf{W}_c^r \mathbf{W}_c^r^\top \right\|_F^2. \end{aligned}$$

Therefore, $\left\| \mathbf{W}_c^r \mathbf{W}_c^r^\top \right\|_F = \left\| \mathbf{W}_c^r \mathbf{W}_c^r^\top \right\|_F$. Similarly, $\left\| \mathbf{W}_c^{r'} \mathbf{W}_c^{r'}^\top \right\|_F = \left\| \mathbf{W}_c^{r'} \mathbf{W}_c^{r'}^\top \right\|_F$.

I.3 Procrustes Similarity

For weights $\mathbf{W}^r, \mathbf{W}^{r'} \in \mathcal{W} \subset \mathbb{R}^{d \times d_{\text{head}}}$, the orthogonal Procrustes problem (Schönemann, 1966) is

$$\begin{aligned} & \text{minimize}_{\substack{\Phi \in \mathbb{R}^{d \times d}, \\ \Phi^\top \Phi = \Phi \Phi^\top = \mathbf{I}_d}} \left\| \Phi \mathbf{W}^r - \mathbf{W}^{r'} \right\|_F. \end{aligned} \quad (78)$$

This problem admits an analytic solution. Let $\mathbf{W}^{r'} \mathbf{W}^r^\top = \mathbf{U}^{r,r'} \Sigma^{r,r'} \mathbf{V}^{r,r'}^\top$ be the SVD, where $\mathbf{U}^{r,r'}, \mathbf{V}^{r,r'} \in \mathbb{R}^{d \times d}$ are orthogonal and $\Sigma^{r,r'} \in \mathbb{R}^{d \times d}$ is diagonal with nonnegative entries. Then (78) is minimized by $\Phi^* = \mathbf{U}^{r,r'} \mathbf{V}^{r,r'}^\top \in \mathbb{R}^{d \times d}$ (Schönemann, 1966). We refer to $\left\| \Phi^* \mathbf{W}^r - \mathbf{W}^{r'} \right\|_F$ as the Procrustes distance (Rohlf, 1999).

To compare with PK and CS, we convert the Procrustes distance into a normalized similarity. For $\Phi^* \mathbf{W}^r - \mathbf{W}^{r'} \in \mathbb{R}^{d \times d_{\text{head}}}$, we have

$$\begin{aligned} & \left\| \Phi^* \mathbf{W}^r - \mathbf{W}^{r'} \right\|_F^2 \\ &= \text{tr} \left(\left(\Phi^* \mathbf{W}^r - \mathbf{W}^{r'} \right)^\top \left(\Phi^* \mathbf{W}^r - \mathbf{W}^{r'} \right) \right) \\ &= \text{tr} \left(\left(\Phi^* \mathbf{W}^r \right)^\top \Phi^* \mathbf{W}^r \right) + \text{tr} \left(\mathbf{W}^{r'}^\top \mathbf{W}^{r'} \right) \\ &\quad - 2 \text{tr} \left(\mathbf{W}^{r'}^\top \Phi^* \mathbf{W}^r \right) \\ &= \left\| \Phi^* \mathbf{W}^r \right\|_F^2 + \left\| \mathbf{W}^{r'} \right\|_F^2 - 2 \text{tr} \left(\mathbf{W}^{r'}^\top \Phi^* \mathbf{W}^r \right) \\ &\leq \left\| \Phi^* \mathbf{W}^r \right\|_F^2 + \left\| \mathbf{W}^{r'} \right\|_F^2, \end{aligned} \quad (79)$$

where the inequality uses

$$\begin{aligned} & \text{tr} \left(\mathbf{W}^{r'}^\top \Phi^* \mathbf{W}^r \right) \\ &= \text{tr} \left(\left(\mathbf{W}^{r'}^\top \Phi^* \mathbf{W}^r \right)^\top \right) \\ &= \text{tr} \left(\mathbf{W}^r \mathbf{W}^r^\top \Phi^*^\top \mathbf{W}^{r'} \right) = \text{tr} \left(\mathbf{W}^{r'} \mathbf{W}^r \mathbf{W}^r^\top \Phi^*^\top \right) \\ &= \text{tr} \left(\mathbf{U}^{r,r'} \Sigma^{r,r'} \mathbf{V}^{r,r'}^\top \left(\mathbf{U}^{r,r'} \mathbf{V}^{r,r'}^\top \right)^\top \right) \\ &= \text{tr} \left(\mathbf{U}^{r,r'} \Sigma^{r,r'} \mathbf{V}^{r,r'}^\top \mathbf{V}^{r,r'} \mathbf{U}^{r,r'}^\top \right) \\ &= \text{tr} \left(\Sigma^{r,r'} \right) \geq 0, \end{aligned} \quad (80)$$

where we use $\mathbf{U}^{r,r'} \mathbf{U}^{r,r'}^\top = \mathbf{V}^{r,r'} \mathbf{V}^{r,r'}^\top = \mathbf{I}_d$ and the cyclic invariance of the trace. In addition, the inequality is tight if and only if $\text{tr}(\Sigma^{r,r'}) = 0$, equivalently, $\Sigma^{r,r'} = \mathbf{O}$ (that is, $\mathbf{W}^r \mathbf{W}^{r'}^\top = \mathbf{O}$). Therefore, when $\left\| \Phi^* \mathbf{W}^r \right\|_F^2 + \left\| \mathbf{W}^{r'} \right\|_F^2 \neq 0$, we have

$$0 \leq \frac{\left\| \Phi^* \mathbf{W}^r - \mathbf{W}^{r'} \right\|_F^2}{\left\| \Phi^* \mathbf{W}^r \right\|_F^2 + \left\| \mathbf{W}^{r'} \right\|_F^2} \leq 1. \quad (81)$$

	Method	WP	Heads			Edges			Figure
			#Total	#Annot.	#Unannot.	#Total	#Both Annot.	#Either Annot.	
Top10	PK		30	15 (50.0%)	15 (50.0%)	30	10 (33.3%)	22 (73.3%)	Fig. 19
	CS		30	9 (30.0%)	21 (70.0%)	30	3 (10.0%)	9 (30.0%)	Fig. 21
Top20	PK	✓	52	22 (42.3%)	30 (57.7%)	60	18 (30.0%)	42 (70.0%)	Fig. 6
			51	21 (41.2%)	30 (58.8%)	60	17 (28.3%)	41 (68.3%)	Fig. 37
	CS	✓	46	14 (30.4%)	32 (69.6%)	60	3 (5.0%)	20 (33.3%)	Fig. 22
			34	6 (17.6%)	28 (82.4%)	60	0 (0.0%)	9 (15.0%)	Fig. 38
	Simple-CS		51	16 (31.4%)	35 (68.6%)	60	5 (8.3%)	26 (43.3%)	Fig. 24
	Linear CKA		24	12 (50.0%)	12 (50.0%)	60	22 (36.7%)	51 (85.0%)	Fig. 25
	Procrustes Similarity		31	12 (38.7%)	19 (61.3%)	60	19 (31.7%)	47 (78.3%)	Fig. 26
Top30	PK		58	24 (41.4%)	34 (58.6%)	90	22 (24.4%)	61 (67.8%)	Fig. 20
	CS		61	20 (32.8%)	41 (67.2%)	90	10 (11.1%)	37 (41.1%)	Fig. 23

Table 10: Number of heads and edges appearing in the wiring diagrams of GPT2-small. For each of OQ, OK, and OV, a score is computed for each pair of a head in an earlier layer and a head in a later layer (9,504 pairs), and the top- k pairs are drawn as edges. A head is counted if it is an endpoint of at least one edge. Annotated heads are those registered in the head classes, and an edge is counted in the “Both Annot.” column when both of its endpoints are annotated and in the “Either Annot.” column when at least one of its endpoints is annotated. The symbol ✓ in the “WP” column indicates that the method is applied to the preprocessed weights described in Appendix O.

We then define Procrustes Similarity as

$$\text{Procrustes-Similarity}(r, r') = 1 - \frac{\|\Phi^* \mathbf{W}^r - \mathbf{W}^{r'}\|_F^2}{\|\Phi^* \mathbf{W}^r\|_F^2 + \|\mathbf{W}^{r'}\|_F^2}. \quad (82)$$

J Additional experimental details for each method

J.1 Additional wiring diagrams

In this section, we show wiring diagrams for PK and CS while varying the number of top edges displayed. We also show wiring diagrams with the top 20 edges for the baseline methods in Section 7, namely Simple-CS, Linear CKA, and Procrustes Similarity. Table 10 summarizes the numbers of heads and edges appearing in these wiring diagrams³⁸.

Figures 19 and 20 show the PK wiring diagrams with the top 10 and top 30 edges, respectively. Although Fig. 19 is sparse, the OK PK scores remain large between Previous Token Heads and Induction Heads. It also shows large OQ PK scores among Name-Mover-related heads and confirms that L4H7 acts as a hub. In contrast, Fig. 20 is denser and contains more information, but is harder to interpret. Compared with Fig. 6, which shows the top 20 edges, it also includes more heads in the middle layers that do not belong to any head class. For these reasons, in Section 6, we set the number of displayed edges to 20.

³⁸Identity Heads are excluded from the head classes used to define annotated heads in Table 10, but are included among the head classes in Fig. 1.

Figures 21, 22, and 23 show the CS wiring diagrams with the top 10, top 20, and top 30 edges, respectively. Compared with the PK wiring diagrams, more heads in early layers appear, and many edges connect early and late layers. Even in the top 30 diagram, Name-Mover-related heads appear less frequently than under PK. Moreover, the OQ edges from the S-Inhibition Head L8H10 to Name-Mover-related heads, which appear under PK, are absent under CS.

Figures 24, 25, and 26 show the wiring diagrams with the top 20 edges for Simple-CS, Linear CKA, and Procrustes Similarity, respectively. For Simple-CS in Fig. 24, as with CS, many heads that do not belong to any head class appear in early layers. In particular, Simple-CS yields the largest number of such heads among the methods under the top-20 setting (see Table 10). For Linear CKA in Fig. 25 and Procrustes Similarity in Fig. 26, the displayed heads are concentrated in late layers, with dense connections among them. Consequently, unlike PK in Fig. 6, these methods do not capture relationships involving heads in early layers.

J.2 Score heatmaps for 16 weight pairing types

In this section, we examine the score patterns between heads for 16 weight pairing types using heatmaps for each method. Figures 27, 28, 29, 30, and 31 show the score heatmaps for PK, CS, Simple-CS, Linear CKA, and Procrustes Similarity, respectively. The color bar of each heatmap is

GPT2-small: Top 10 Projection Kernel for OQ, OK, OV

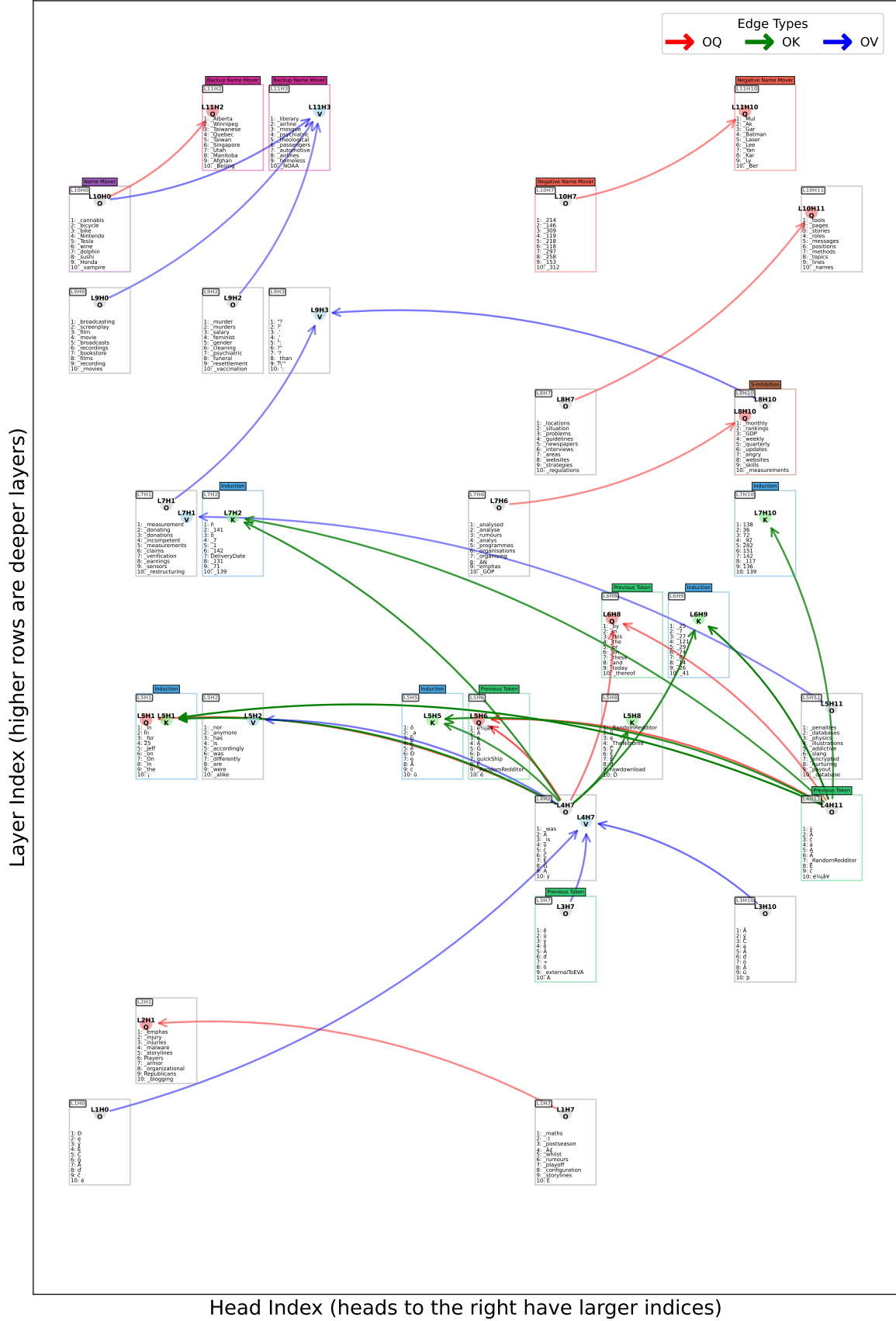


Figure 19: For GPT2-small, we visualize the top 10 scores of PK_{OQ} , PK_{OK} , and PK_{OV} . For each head h , we show the top 10 tokens that represent W_O^h using the projection method in Section 4. For readability, we replace $\tilde{G}$ in tokens with $_$.

GPT2-small: Top 30 Projection Kernel for OQ, OK, OV

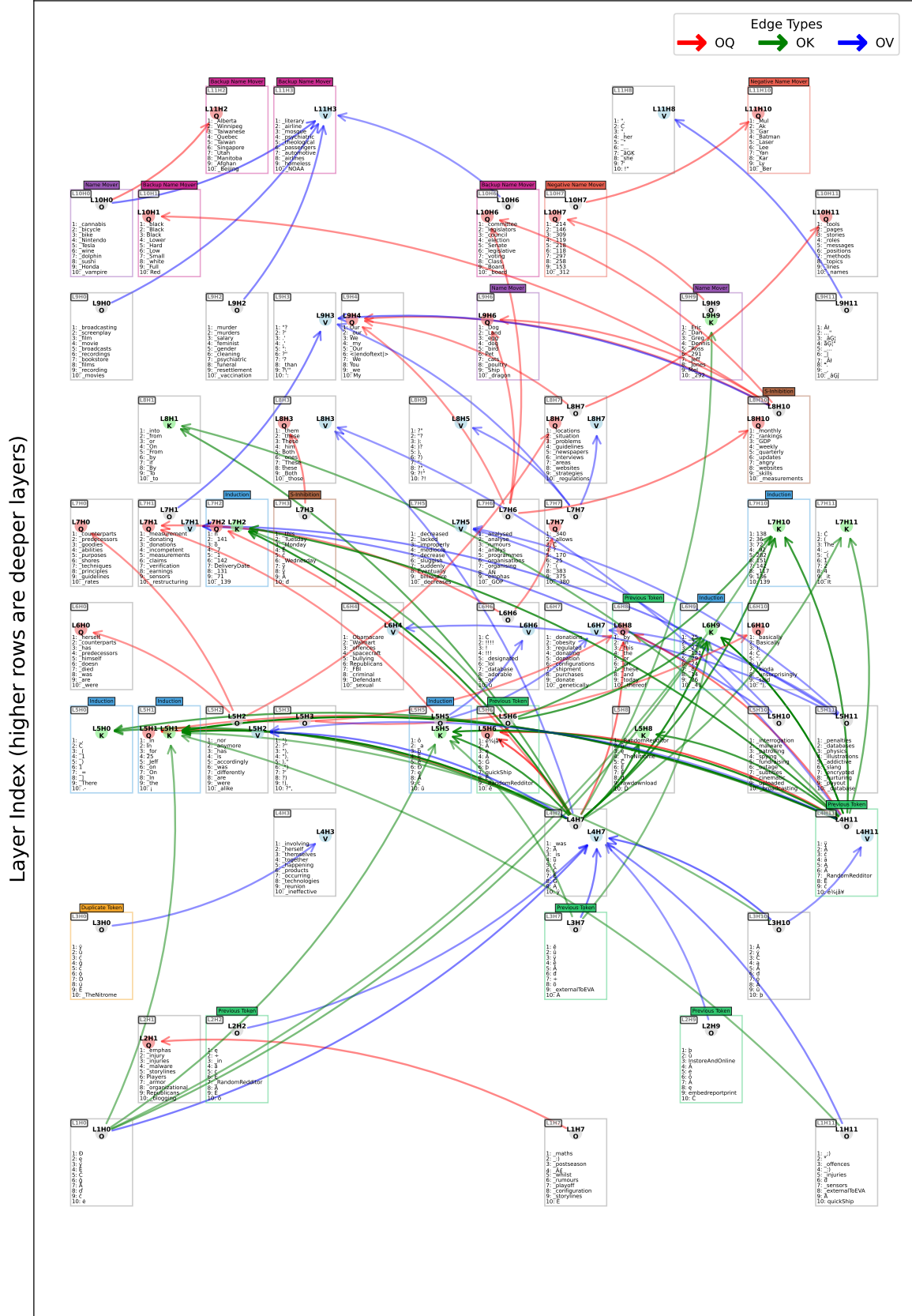


Figure 20: For GPT2-small, we visualize the top 30 scores of PK_{OQ} , PK_{OK} , and PK_{OV} . For each head h , we show the top 10 tokens that represent W_O^h using the projection method in Section 4. For readability, we replace $\tilde{G}$ in tokens with $_$.

GPT2-small: Top 10 Composition Score for OQ, OK, OV

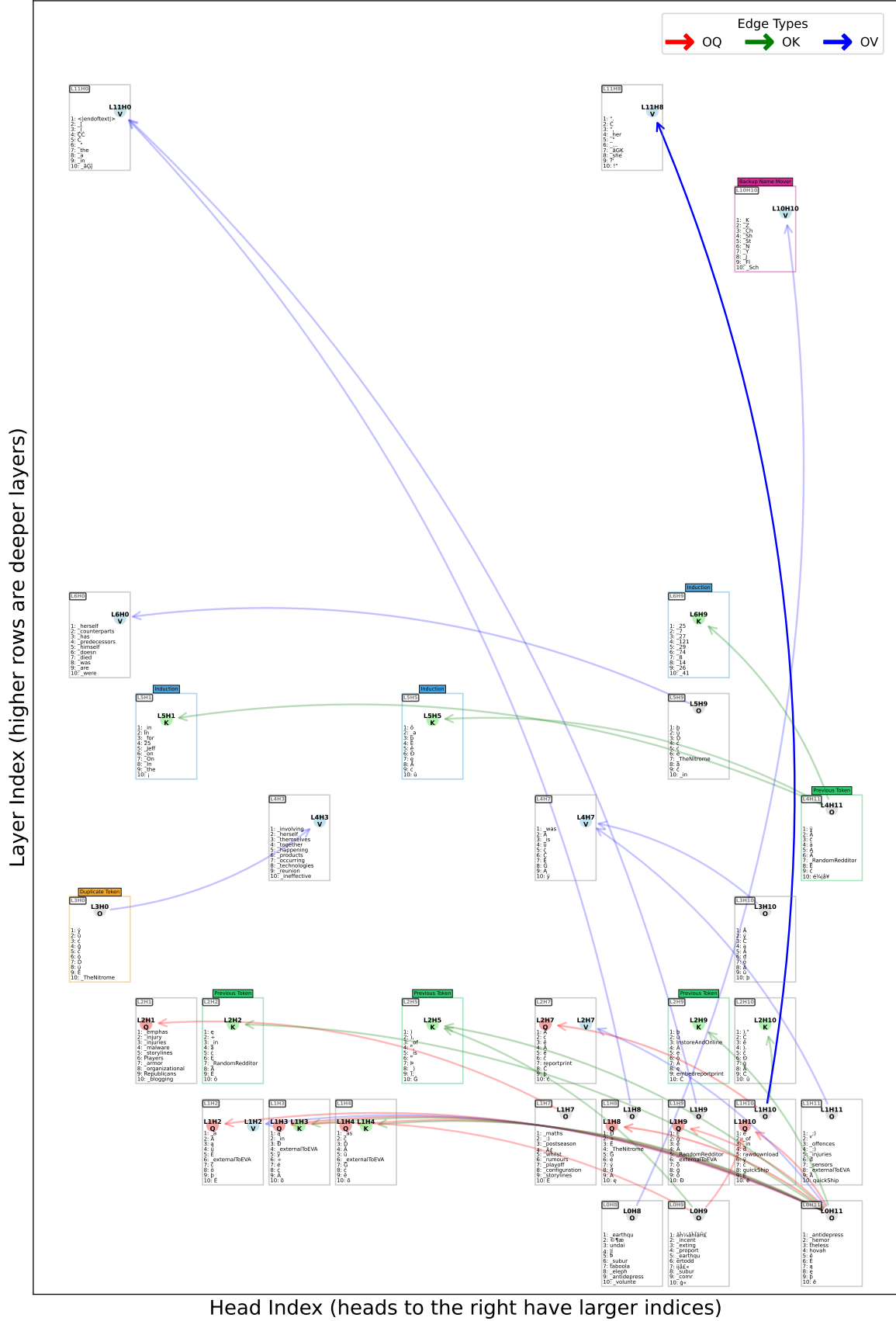


Figure 21: For GPT2-small, we visualize the top 10 scores of CS_{OQ} , CS_{OK} , and CS_{OV} . For each head h , we show the top 10 tokens that represent W_O^h using the projection method in Section 4. For readability, we replace $\tilde{G}$ in tokens with $_$.

GPT2-small: Top 20 Composition Score for OQ, OK, OV

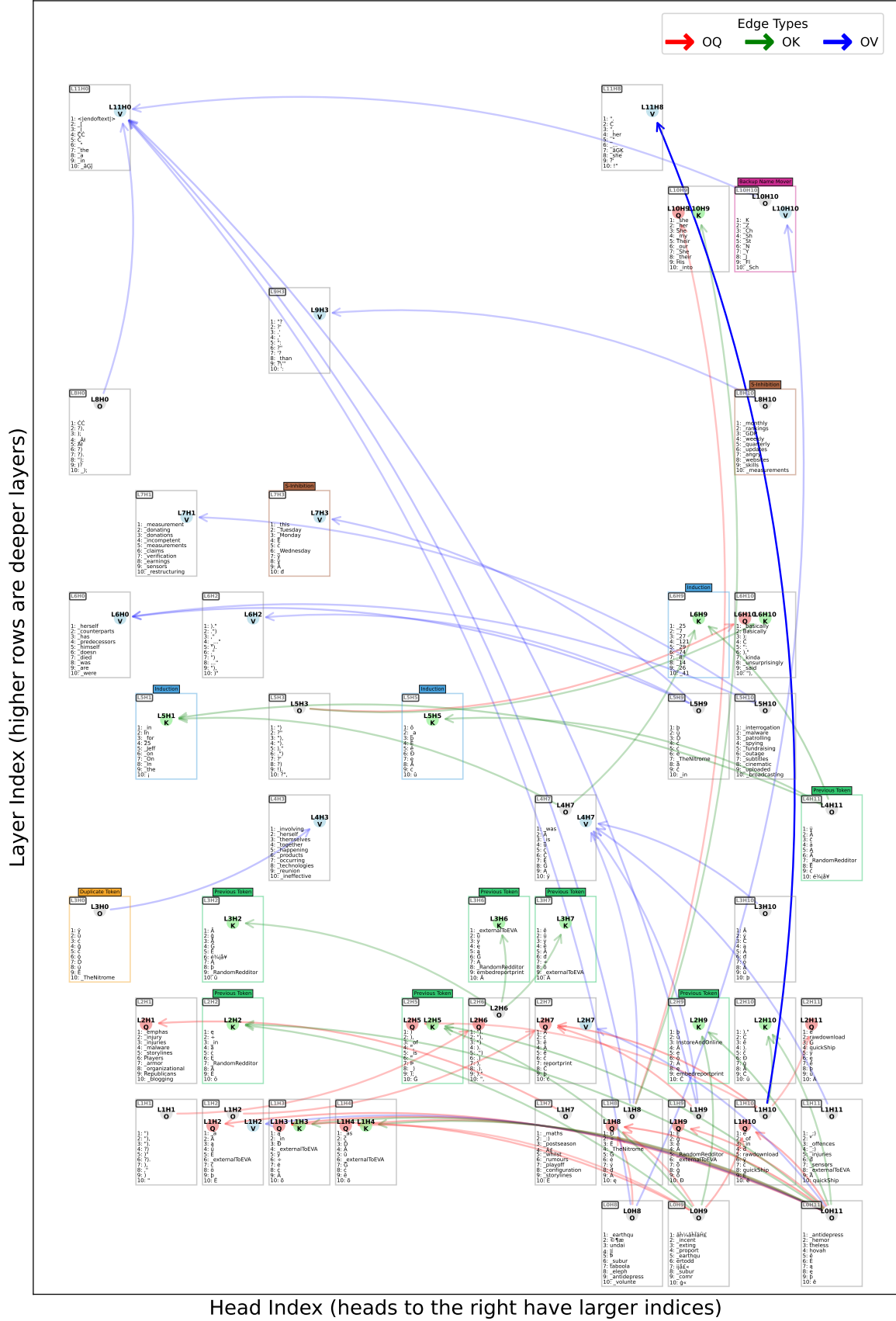


Figure 22: For GPT2-small, we visualize the top 20 scores of CS_{OQ} , CS_{OK} , and CS_{OV} . For each head h , we show the top 10 tokens that represent W_O^h using the projection method in Section 4. For readability, we replace $\tilde{G}$ in tokens with $_$.

GPT2-small: Top 30 Composition Score for OQ, OK, OV

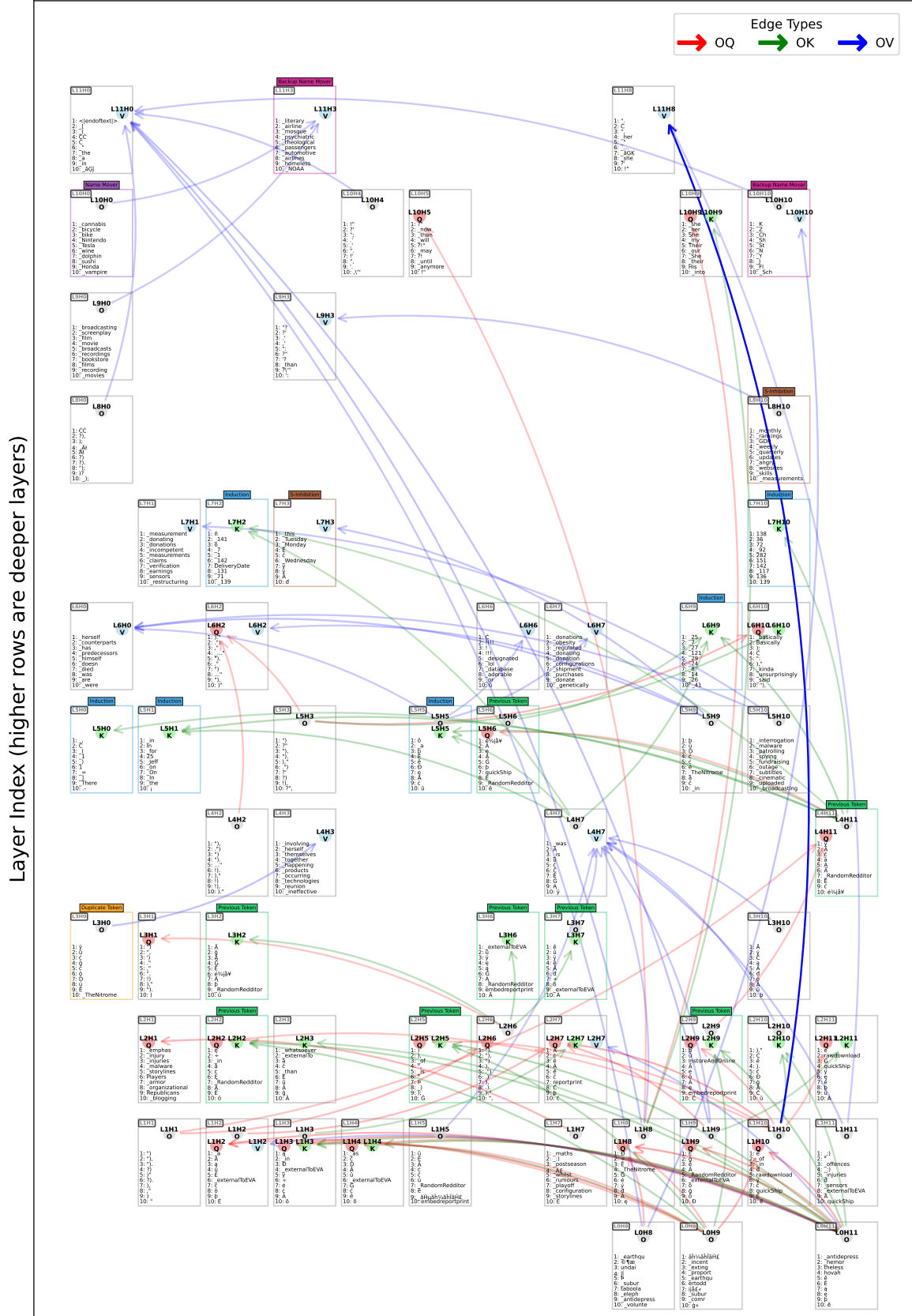


Figure 23: For GPT2-small, we visualize the top 30 scores of CS_{OQ} , CS_{OK} , and CS_{OV} . For each head h , we show the top 10 tokens that represent W_O^h using the projection method in Section 4. For readability, we replace $\tilde{G}$ in tokens with $_$.

GPT2-small: Top 20 Simple-CS for OQ, OK, OV

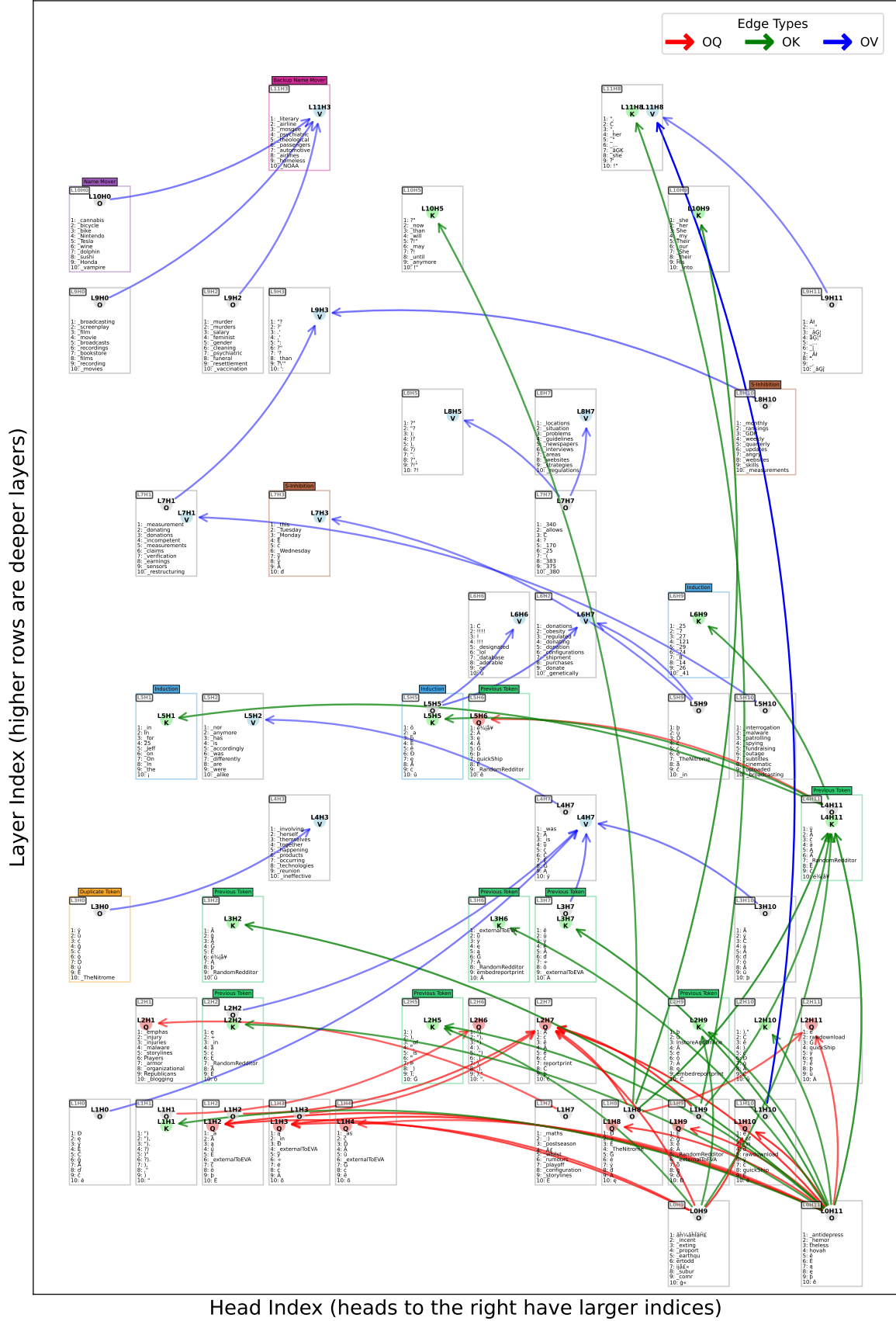


Figure 24: For GPT2-small, we visualize the top 20 scores of Simple-CS for the pairing types OQ, OK, and OV. For each head h , we show the top 10 tokens that represent $\mathbf{W}_O^h$ using the projection method in Section 4. For readability, we replace $\hat{G}$ in tokens with $_$.

GPT2-small: Top 20 Linear CKA for OQ, OK, OV

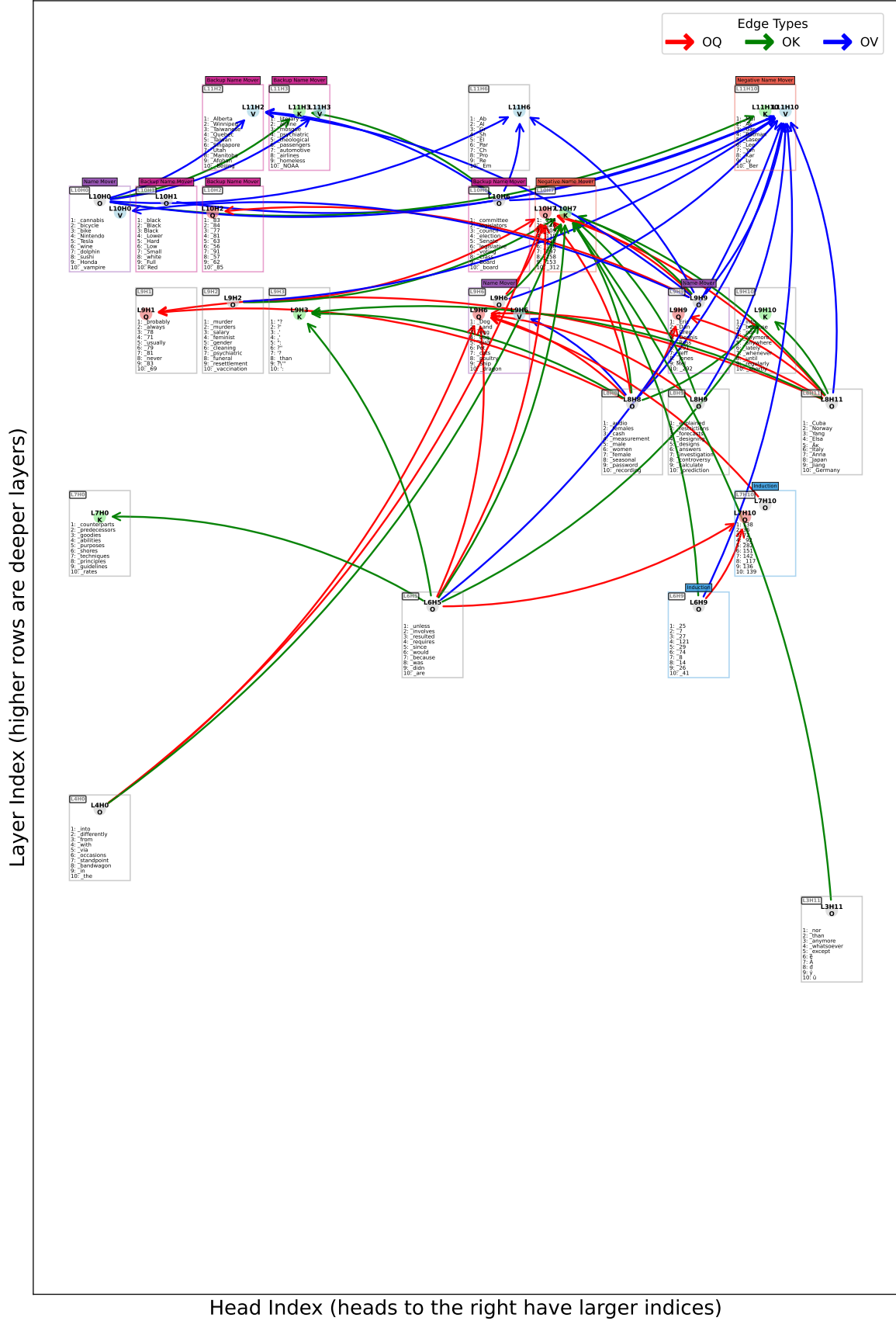


Figure 25: For GPT2-small, we visualize the top 20 scores of Linear CKA for the pairing types OQ, OK, and OV. For each head h , we show the top 10 tokens that represent W_O^h using the projection method in Section 4. For readability, we replace $\hat{G}$ in tokens with $_$.

GPT2-small: Top 20 Procrustes Similarity for OQ, OK, OV

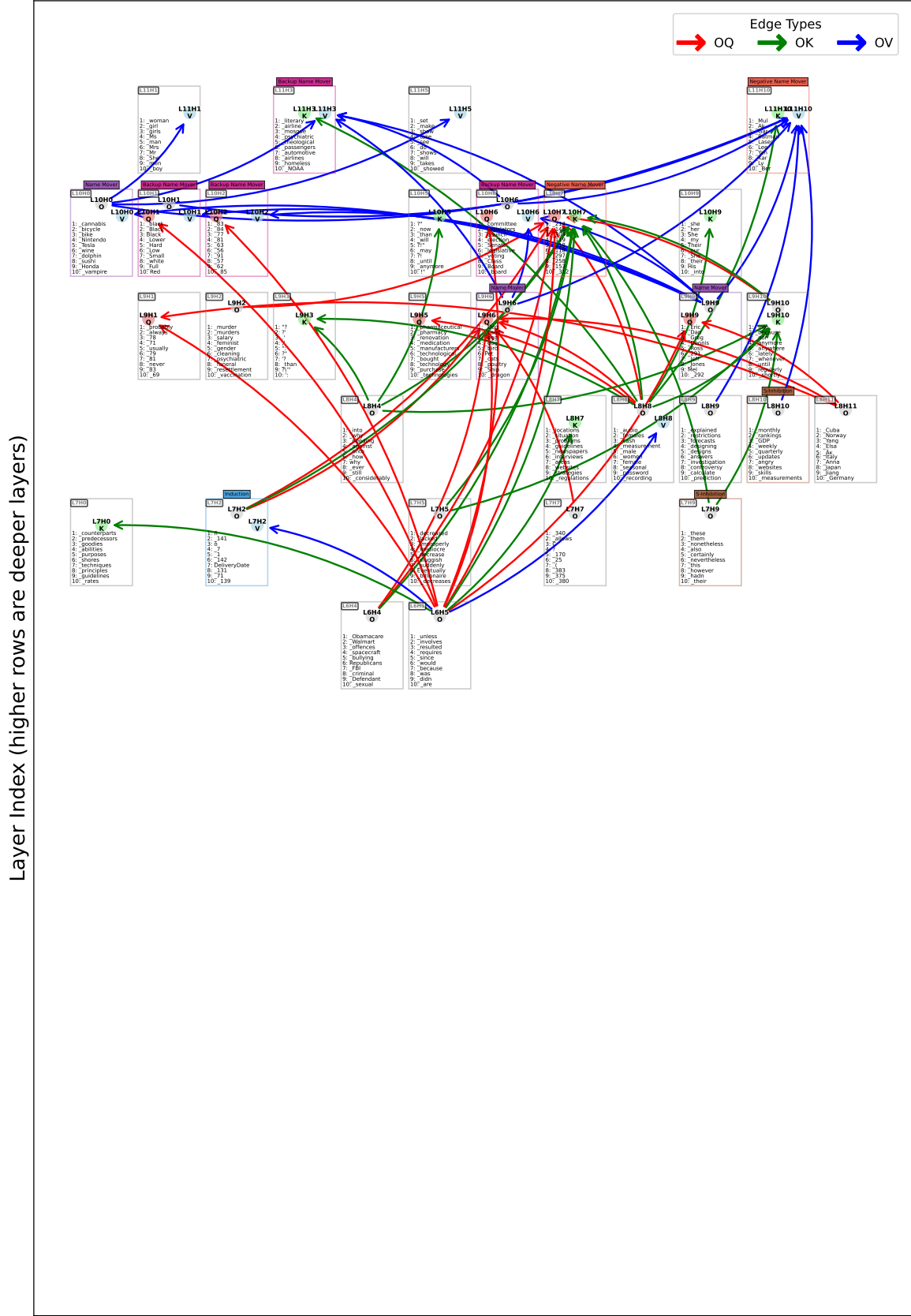


Figure 26: For GPT2-small, we visualize the top 20 scores of Procrustes Similarity for the pairing types OQ, OK, and OV. For each head h , we show the top 10 tokens that represent $\mathbf{W}_O^h$ using the projection method in Section 4. For readability, we replace $\tilde{G}$ in tokens with $_$.

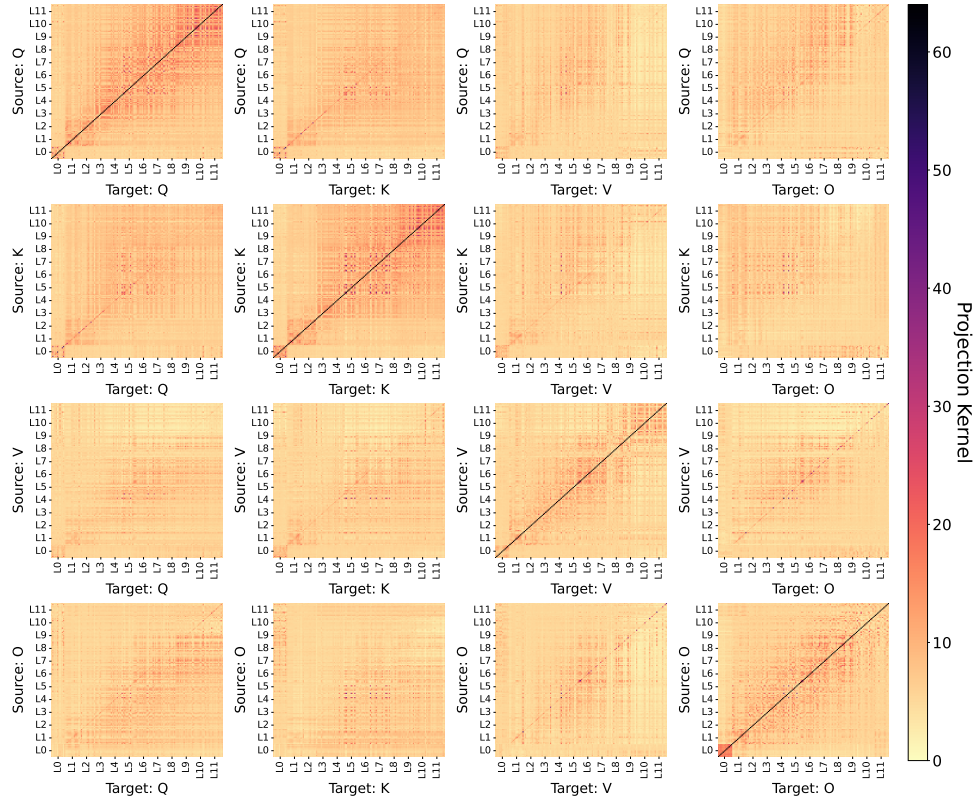


Figure 27: Projection Kernel (PK) between all head pairs of GPT2-small for each weight pairing type (source, target) of Q, K, V, and O, where the tick label $L\ell$ denotes layer ℓ .

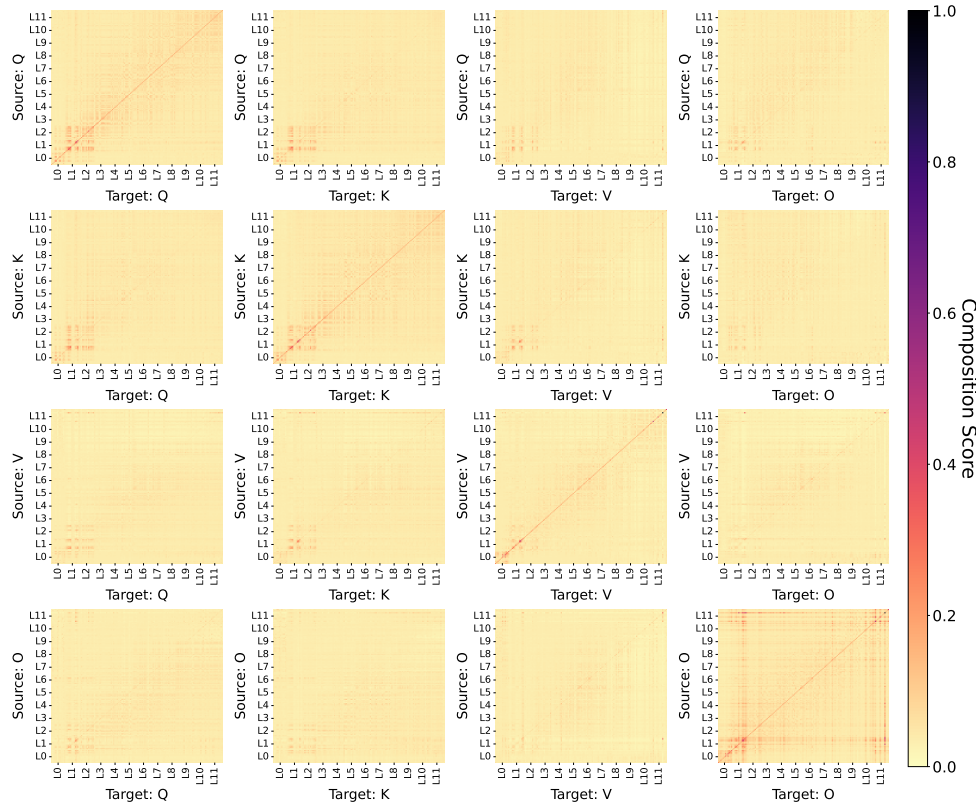


Figure 28: Composition Score (CS) between all head pairs of GPT2-small for each weight pairing type (source, target) of Q, K, V, and O, where the tick label $L\ell$ denotes layer ℓ .

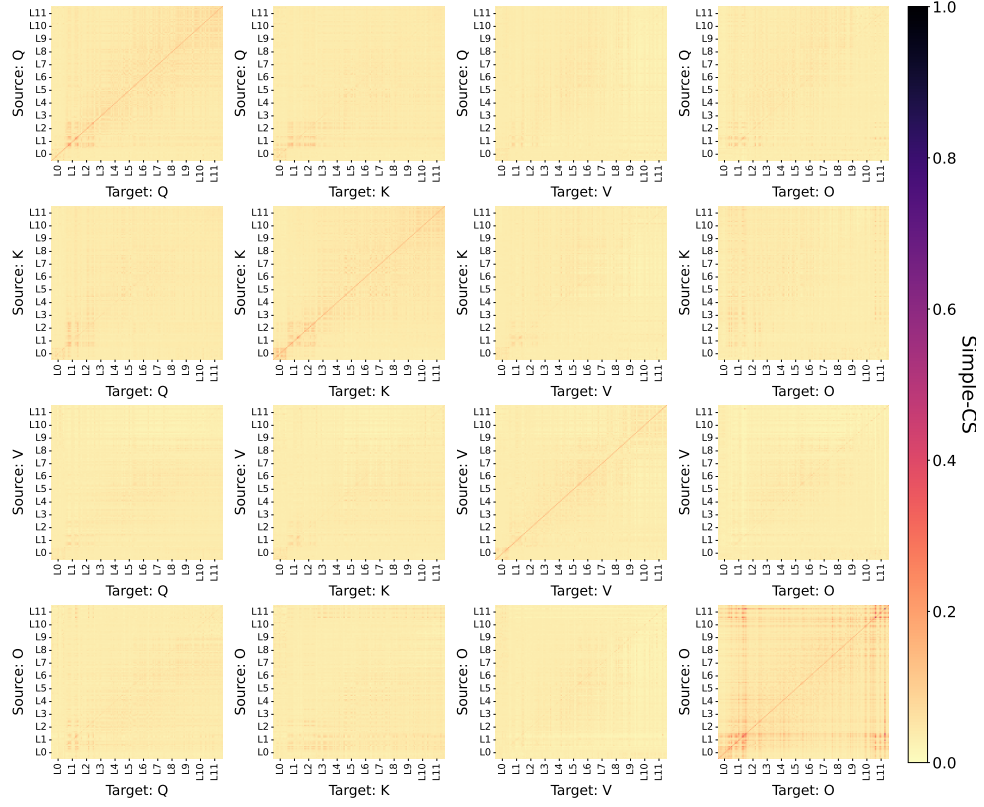


Figure 29: Simple-CS between all head pairs of GPT2-small for each weight pairing type (source, target) of Q, K, V, and O, where the tick label $L\ell$ denotes layer ℓ .

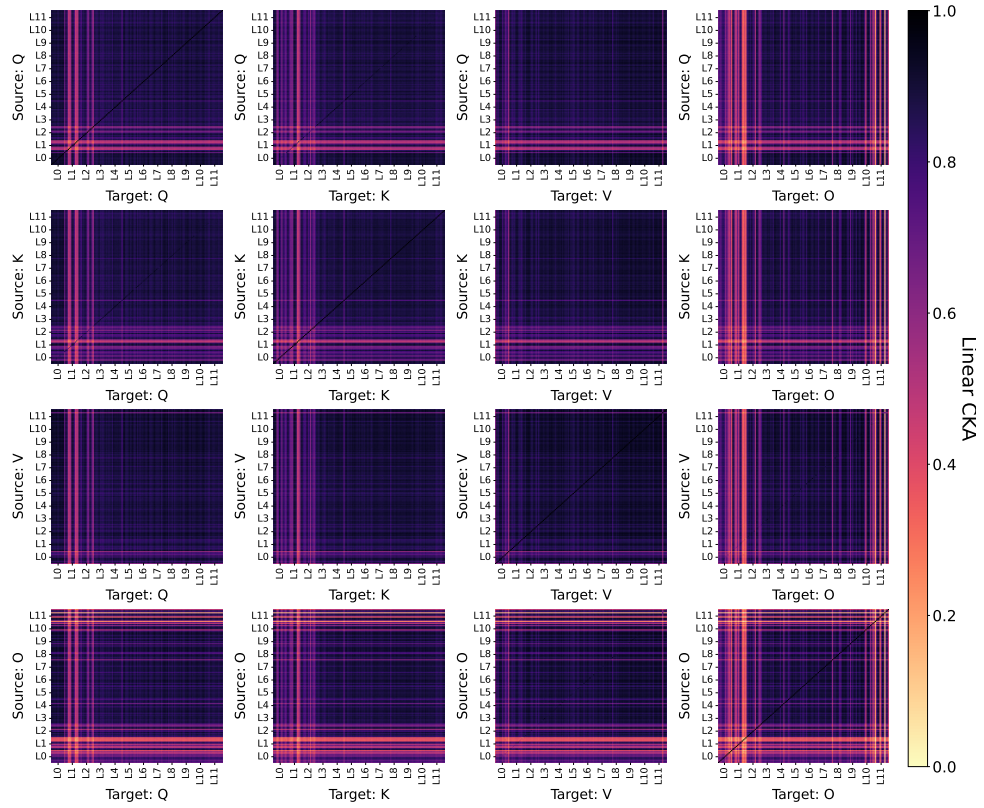


Figure 30: Linear CKA between all head pairs of GPT2-small for each weight pairing type (source, target) of Q, K, V, and O, where the tick label $L\ell$ denotes layer ℓ .

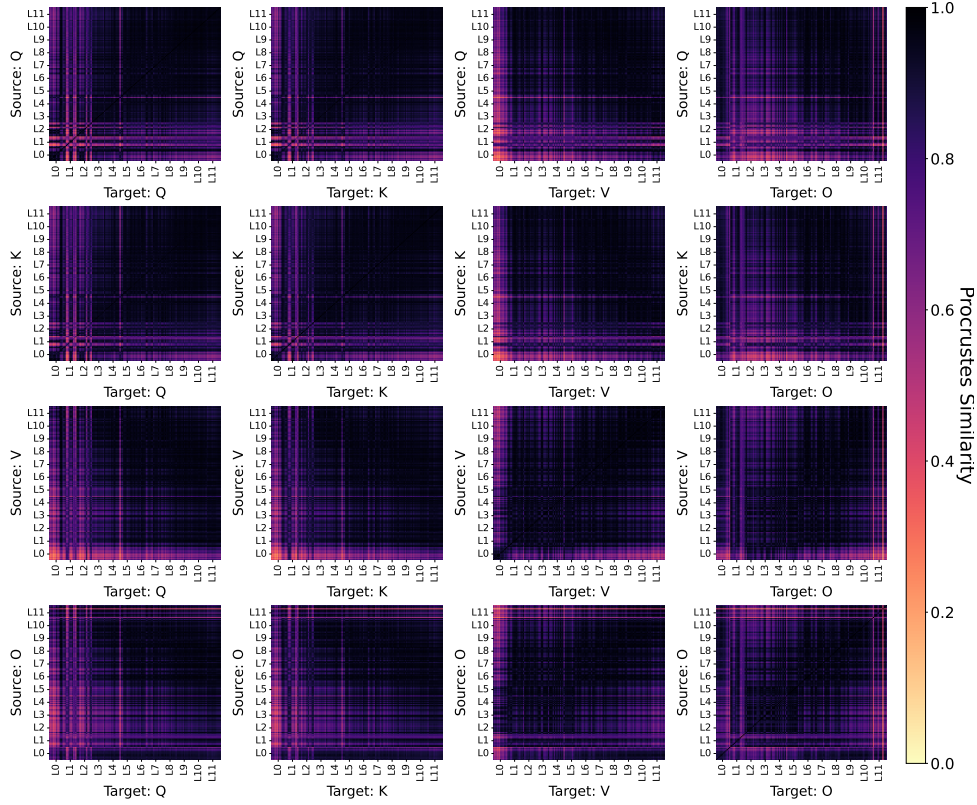


Figure 31: Procrustes Similarity between all head pairs of GPT2-small for each weight pairing type (source, target) of Q, K, V, and O, where the tick label $L\ell$ denotes layer ℓ .

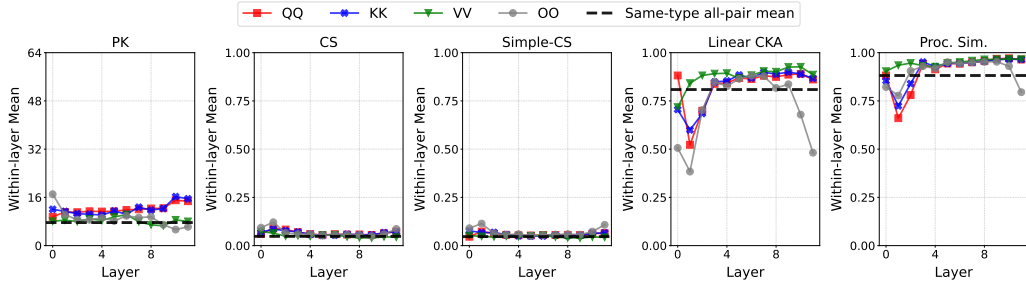


Figure 32: Layerwise PK, CS, Simple-CS, Linear CKA, and Procrustes Similarity of GPT2-small, where each panel shows the mean over all 66 head pairs within each layer (all pairwise combinations of the 12 heads) for the weight pairing types QQ, KK, VV, and OO. The dashed line marks the mean over all $\binom{144}{2} = 10,296$ head pairs for each of the four weight pairing types, yielding a total of $4 \times \binom{144}{2} = 41,184$ comparisons: PK = 7.65, CS = 0.05, Simple-CS = 0.05, Linear CKA = 0.81, and Proc. Sim. = 0.88.

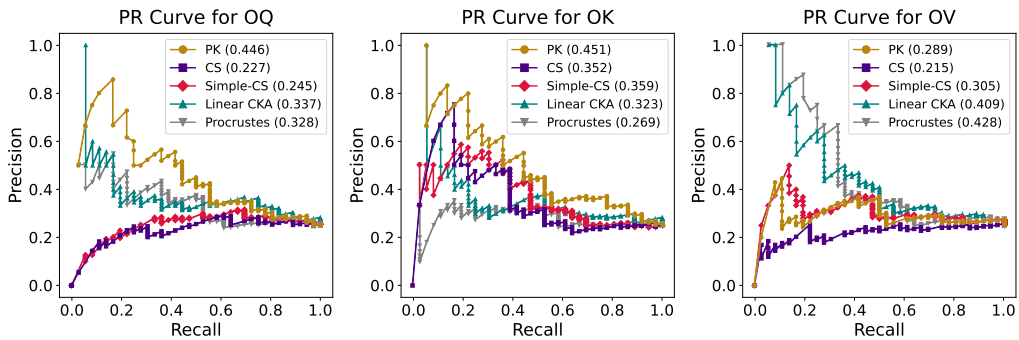


Figure 33: PR curves of head detection for OQ, OK, and OV in GPT2-small, shown for each method.

scaled to the range of the corresponding method³⁹.

In the early layers, PK, CS, and Simple-CS tend to yield high within-layer scores for same-type weight pairings, namely QQ, KK, VV, and OO. The PK heatmaps exhibit greater contrast than those of CS and Simple-CS, indicating that PK scores span a broader portion of their range. Furthermore, for the OQ, OK, and OV heatmaps of PK, scores tend to be high in the bottom-right region, corresponding to pairs of a head in an earlier layer and a head in a later layer. This pattern is consistent with the relationships being readily visible in the PK wiring diagrams. In contrast, although CS and Simple-CS have a range from 0 to 1, most of their scores are close to 0 and exhibit limited variation.

Although Linear CKA and Procrustes Similarity also have a range from 0 to 1, their scores are close to 1 in most cases and therefore show similarly limited variation. However, we find that the scores become extremely small for several specific heads. A more detailed analysis of these methods and specific heads is left for future work.

J.3 Layerwise averages for same-type weight pairings

Building on the analysis in Appendix J.2, Fig. 32 shows, for each method and each same-type weight pairing (QQ, KK, VV, and OO), the mean score within each layer together with the overall mean across the considered head pairs as a baseline.

For PK, the within-layer mean is higher than the baseline in most cases. Furthermore, the mean scores for QQ and KK increase with depth, whereas those for VV and OO decrease. For CS and Simple-CS, the within-layer means remain close to the baseline, although slight variations are observed in the early layers and the final layer. For Linear CKA and Procrustes Similarity, the scores are generally high, as also observed in the heatmaps. In the early layers, however, the scores are relatively low, and those for QQ, KK, and OO drop sharply in layer 1, that is, the second layer. From the middle layers onward, the mean scores for all four pairing types exceed the baseline, except that OO drops sharply again toward the final layers, falling below the baseline. A more detailed layerwise analysis is left for future work.

³⁹Specifically, the range of PK is from 0 to d_{head} , whereas those of the other methods are from 0 to 1. Rescaling the PK range to 0 to 1 does not change the colors in the heatmap.

K Details of the evaluation of each method on GPT2-small

K.1 Details of head detection performance for OQ, OK, and OV

Figure 33 shows the PR curves used to compute the PR-AUC values in Table 1. For OQ and OK, the PK curve lies above the other curves in almost all cases, resulting in larger PR-AUC values. In contrast, for OV, the Linear CKA and Procrustes Similarity curves lie above the PK curve, resulting in higher PR-AUC values than that of PK.

K.2 Details of classwise classification performance for QQ, KK, VV, and OO

Table 11 reports, for each head class, the mean PR-AUC and ROC-AUC scores used to compute Table 2, with the mean taken over QQ, KK, VV, and OO. It also reports PR-AUC and ROC-AUC computed separately for each of the four weight pairing types QQ, KK, VV, and OO.

Focusing on the mean scores over QQ, KK, VV, and OO, we find that PK achieves high classification performance for Duplicate Token, Induction, Negative Name Mover, Backup Name Mover, and S-Inhibition Heads, indicating that PK performs well for many head classes. In contrast, CS outperforms PK for Previous Token Heads, and Procrustes Similarity outperforms PK for Name Mover Heads.

The results for individual weight pairing types further show that classwise classification performance depends strongly on the weight pairing type as well as the method. For Previous Token Heads, PK underperforms CS for QQ, KK, and VV, whereas PK attains the best performance for OO. For Name Mover Heads, PK underperforms Procrustes Similarity for VV and OO, whereas PK attains the best performance for QQ and KK. A more detailed analysis of the relationships among the method, weight pairing type, and head class is left for future work.

L Details of inlet and outlet scores in GPT2-small

Motivated by the wiring diagram based on PK in Fig. 6, where L4H7 acts as a hub, we aim to detect such hub heads in GPT2-small. To this end, we characterize the hub role in terms of inlet and outlet behavior, and propose the inlet score and outlet

	Method	Dup (3)	Prev (10)	Ind (6)	NM (3)	N-NM (2)	B-NM (8)	S-Inh (4)	Avg
QQ	PK	0.006	0.012	0.199	0.055	<u>0.001</u>	0.090	0.009	0.053
	CS	0.001	0.052	0.023	0.005	0.000	0.020	0.003	0.015
	Simple-CS	<u>0.002</u>	<u>0.021</u>	0.045	0.016	0.001	<u>0.041</u>	<u>0.005</u>	<u>0.019</u>
	Linear CKA	0.001	0.003	0.036	0.003	0.001	0.006	0.001	0.007
	Proc. Sim.	0.001	0.003	<u>0.049</u>	<u>0.019</u>	0.012	0.039	0.001	0.018
KK	PK	0.006	0.010	0.685	0.031	0.003	0.063	0.008	0.115
	CS	0.001	0.111	0.066	0.006	0.001	0.022	0.003	0.030
	Simple-CS	<u>0.002</u>	<u>0.055</u>	<u>0.137</u>	<u>0.011</u>	0.002	<u>0.031</u>	<u>0.004</u>	0.034
	Linear CKA	0.001	0.003	0.002	0.001	<u>0.008</u>	0.003	0.003	0.003
	Proc. Sim.	0.000	0.004	0.003	0.002	0.250	0.012	0.002	<u>0.039</u>
VV	PK	0.000	0.018	<u>0.009</u>	0.008	<u>0.009</u>	0.015	0.002	0.009
	CS	0.000	0.021	0.007	0.002	0.003	0.007	0.002	0.006
	Simple-CS	<u>0.000</u>	<u>0.021</u>	0.010	0.006	0.006	0.011	<u>0.002</u>	0.008
	Linear CKA	0.000	0.004	0.001	<u>0.034</u>	0.004	<u>0.036</u>	0.001	<u>0.011</u>
	Proc. Sim.	0.000	0.007	0.003	0.070	0.022	0.045	0.002	0.021
OO	PK	0.002	0.038	0.024	0.002	0.019	0.002	0.004	0.013
	CS	<u>0.000</u>	0.012	0.002	0.000	0.000	0.002	0.001	0.003
	Simple-CS	0.000	0.009	0.002	0.000	0.000	0.002	0.001	0.002
	Linear CKA	0.000	0.006	0.004	<u>0.430</u>	0.000	<u>0.010</u>	0.001	<u>0.064</u>
	Proc. Sim.	0.000	<u>0.014</u>	<u>0.010</u>	0.461	<u>0.001</u>	0.018	<u>0.002</u>	0.072
Mean over QQ, KK, VV, OO	PK	0.003	0.019	0.229	0.024	<u>0.008</u>	0.042	0.006	0.047
	CS	0.001	0.049	0.024	0.003	0.001	0.013	0.002	0.013
	Simple-CS	0.001	<u>0.027</u>	<u>0.048</u>	0.008	0.002	0.021	<u>0.003</u>	0.016
	Linear CKA	0.000	0.004	0.011	<u>0.117</u>	0.003	0.014	0.001	0.021
	Proc. Sim.	0.000	0.007	0.016	0.138	0.071	<u>0.029</u>	0.002	<u>0.038</u>

(a) PR-AUC

	Method	Dup (3)	Prev (10)	Ind (6)	NM (3)	N-NM (2)	B-NM (8)	S-Inh (4)	Avg
QQ	PK	0.453	0.748	0.967	0.997	<u>0.952</u>	<u>0.922</u>	0.953	<u>0.856</u>
	CS	0.416	0.898	0.898	0.973	0.902	0.895	0.911	0.842
	Simple-CS	0.454	<u>0.842</u>	0.943	<u>0.992</u>	0.921	0.921	<u>0.931</u>	0.858
	Linear CKA	0.800	0.265	<u>0.971</u>	0.923	0.945	0.779	0.730	0.773
	Proc. Sim.	<u>0.538</u>	0.433	0.974	0.989	0.996	0.967	0.714	0.802
KK	PK	0.360	0.732	0.998	0.995	0.985	0.926	0.964	0.851
	CS	0.361	0.932	0.977	0.977	0.945	0.891	0.902	<u>0.855</u>
	Simple-CS	0.365	<u>0.897</u>	<u>0.991</u>	<u>0.987</u>	0.979	<u>0.899</u>	0.924	0.863
	Linear CKA	0.739	0.252	0.702	0.791	<u>0.994</u>	0.643	0.893	0.716
	Proc. Sim.	<u>0.394</u>	0.468	0.806	0.908	1.000	0.895	0.730	0.743
VV	PK	0.418	0.874	0.709	0.821	<u>0.994</u>	0.753	<u>0.850</u>	0.774
	CS	0.531	0.889	0.725	0.777	0.981	0.725	0.829	0.780
	Simple-CS	<u>0.520</u>	0.885	<u>0.734</u>	0.792	0.992	0.749	0.847	0.788
	Linear CKA	0.287	0.458	<u>0.454</u>	<u>0.996</u>	0.989	0.932	0.745	0.694
	Proc. Sim.	0.307	0.698	0.746	0.998	0.998	<u>0.903</u>	0.856	<u>0.786</u>
OO	PK	<u>0.562</u>	0.914	<u>0.842</u>	0.712	0.997	0.332	0.910	<u>0.753</u>
	CS	0.601	0.831	0.685	0.512	0.902	0.336	0.779	0.663
	Simple-CS	0.557	0.789	0.714	0.506	0.881	0.364	0.794	0.658
	Linear CKA	0.327	0.651	0.791	<u>0.999</u>	0.611	<u>0.660</u>	0.693	0.676
	Proc. Sim.	0.369	<u>0.851</u>	0.893	1.000	<u>0.922</u>	0.802	<u>0.877</u>	0.816
Mean over QQ, KK, VV, OO	PK	0.448	0.817	0.879	0.881	0.982	0.733	0.919	0.809
	CS	<u>0.477</u>	0.887	0.821	0.810	0.932	0.712	0.855	0.785
	Simple-CS	0.474	<u>0.853</u>	0.845	0.819	0.943	0.733	<u>0.874</u>	<u>0.792</u>
	Linear CKA	0.538	0.407	0.729	<u>0.927</u>	0.885	<u>0.754</u>	0.765	0.715
	Proc. Sim.	0.402	0.612	<u>0.855</u>	0.974	<u>0.979</u>	0.892	0.794	0.787

(b) ROC-AUC

Table 11: PR-AUC and ROC-AUC for each head class in GPT2-small, computed separately for the weight pairing types QQ, KK, VV, and OO, together with the mean over the four weight pairing types for each head class. These values are used to compute the summary reported in Table 2. The maximum is shown in bold and the second largest is underlined. We consider the $\binom{144}{2} = 10,296$ directed head pairs whose source is in an earlier layer than the target; for heads in the same layer, the head with the smaller index is treated as the source. Dup, Prev, Ind, NM, N-NM, B-NM, and S-Inh denote Duplicate Token, Previous Token, Induction, Name Mover, Negative Name Mover, Backup Name Mover, and S-Inhibition heads, respectively.

Top	OQ		OK		OV	
1	L10H7 (N-NM)	161.02	L5H1 (Ind)	140.81	L4H7 (Id)	228.05
2	L5H6 (Prev)	135.27	L7H7	123.09	L11H3 (B-NM)	180.96
3	L7H1	100.09	L5H5 (Ind)	105.05	L9H3	145.73
4	L8H10 (S-Inh)	65.27	L11H3 (B-NM)	79.57	L5H2	135.06
5	L10H11	60.13	L11H8	76.27	L11H8	133.63
6	L9H0	52.52	L9H4	70.04	L7H1	63.83
7	L7H7	51.84	L10H11	65.57	L8H7	62.65
8	L2H1	51.22	L6H8 (Prev)	51.08	L6H8 (Prev)	55.31
9	L1H8	50.87	L6H6	48.90	L6H6	51.27
10	L11H10 (N-NM)	49.37	L5H0 (Ind)	47.84	L7H4	47.76

(a) Inlet Head Scores

Top	OQ		OK		OV	
1	L7H6	243.47	L4H7 (Id)	263.86	L0H1 (Dup, Id)	201.87
2	L1H1	178.36	L4H11 (Prev)	191.91	L3H0 (Dup, Id)	91.87
3	L4H7 (Id)	129.33	L0H9	142.50	L4H7 (Id)	78.90
4	L8H10 (S-Inh)	119.35	L1H11 (Id)	135.82	L5H10	78.79
5	L5H2	114.06	L5H10	106.17	L5H11	77.93
6	L5H3	109.91	L1H1	95.06	L3H10	71.73
7	L6H6	100.38	L0H11	80.11	L1H6	67.85
8	L0H9	85.10	L1H9	80.02	L5H5 (Ind)	60.81
9	L3H10	76.71	L5H3	77.17	L0H4 (Id)	53.09
10	L7H4	56.06	L6H0	54.17	L1H1	40.85

(b) Outlet Head Scores

Table 12: For OQ, OK, and OV in GPT2-small, we report (a) the top 10 heads and their inlet scores in (83) and (b) the top 10 heads and their outlet scores in (85). When a head belongs to the head classes in Table 9 or to the top 10 heads by Identity Head score in Table 8a, we also report this information. Dup, Prev, Ind, NM, N-NM, B-NM, and S-Inh denote Duplicate Token, Previous Token, Induction, Name Mover, Negative Name Mover, Backup Name Mover, and S-Inhibition heads, respectively.

score to identify heads that play each role⁴⁰. This section describes the definitions and results.

First, for each of OQ, OK, and OV, we identify inlet heads by quantifying the extent to which a target head receives the maximum PK score from source heads in earlier layers. For a head h in layer 1 or later, and for $\omega \in \{\text{“Q”}, \text{“K”}, \text{“V”}\}$, we define the PK inlet score as

$$\text{PK-Inlet}_{O\omega}(h) = \sum_{h' \in \bigcup_{\ell' < \ell_h} \mathcal{H}_{\ell'}} \text{PK}((h', O), (h, \omega)) \mathbb{I} \left(\text{PK}((h', O), (h, \omega)) = \max_{h'' \in \bigcup_{\ell_{h'} < \ell''} \mathcal{H}_{\ell''}} \text{PK}((h', O), (h'', \omega)) \right), \quad (83)$$

⁴⁰Although inlet and outlet scores can be defined for similarity measures other than PK, such as CS, we focus on PK in this paper.

where

$$\mathbb{I}(\text{condition}) := \begin{cases} 1 & \text{if the condition holds,} \\ 0 & \text{otherwise,} \end{cases} \quad (84)$$

where ℓ_h denotes the layer index of head h , and $\mathcal{H}_\ell$ denotes the set of heads in layer ℓ . Here and throughout this section, we use shorthand notation for PK⁴¹. Fix a target head h in layer ℓ_h , and consider any source head h' in an earlier layer $\ell_{h'} < \ell_h$. For this fixed h' , we compute $\text{PK}((h', O), (h'', \omega))$ for all target heads h'' in later layers $\ell_{h'} < \ell''$. If $\text{PK}((h', O), (h, \omega))$ attains this maximum, we add its value to the score. Summing these contributions over all such source heads h' gives the inlet score for the target head h .

Next, for each of OQ, OK, and OV, we identify outlet heads by quantifying the extent to which a source head yields the maximum PK score for target heads in subsequent layers. For a head h in any layer before the final layer, and for $\omega \in$

⁴¹For example, $\text{PK}((h', O), (h, \omega))$ abbreviates $\text{PK}(\mathcal{S}^{(h', O)}, \mathcal{S}^{(h, \omega)})$.

{“Q”, “K”, “V”}, we define the PK outlet score as

$$\begin{aligned} \text{PK-Outlet}_{\text{O}\omega}(h) &= \sum_{h' \in \bigcup_{\ell_h < \ell'} \mathcal{H}_{\ell'}} \text{PK}((h, \text{O}), (h', \omega)) \\ \mathbb{I} \left(\text{PK}((h, \text{O}), (h', \omega)) &= \max_{h'' \in \bigcup_{\ell'' < \ell_{h'}} \mathcal{H}_{\ell''}} \text{PK}((h'', \text{O}), (h', \omega)) \right), \end{aligned} \quad (85)$$

where $\mathbb{I}(\cdot)$, ℓ_h , and $\mathcal{H}_\ell$ are defined as above. Fix a source head h in layer ℓ_h , and consider any target head h' in a later layer $\ell_h < \ell_{h'}$. For this fixed h' , we compute $\text{PK}((h'', \text{O}), (h', \omega))$ for all source heads h'' in earlier layers $\ell'' < \ell_{h'}$. If $\text{PK}((h, \text{O}), (h', \omega))$ attains this maximum, we add its value to the score. Summing these contributions over all such target heads h' gives the outlet score for the source head h .

Table 12 reports the top 10 heads and their scores for OQ, OK, and OV based on the inlet and outlet scores in (83) and (85). L4H7 attains the largest inlet score for OV, and its value exceeds the largest inlet scores for OQ and OK. In contrast, no other Identity Head appears among the top-ranked inlet heads. Moreover, L4H7 also attains the largest outlet score for OK, and its value exceeds the largest outlet scores for OQ and OV. L4H7 also ranks third for the outlet scores of OQ and OV. In addition, Identity Heads such as L0H1, L3H0, L1H11, and L0H4 have large outlet scores. A more detailed study of whether L4H7 is special, and of the relationship between Identity Heads and inlet and outlet scores, is left for future work.

M Approximate distribution of PK between random orthogonal matrices

In Section 8, we introduced a framework that quantifies the informativeness of PK distributions for weight pairings by using an approximate distribution of PK for random orthogonal matrices and measuring the Kullback–Leibler (KL) divergence between the two distributions. In this section, we derive the approximate distribution of PK for random orthogonal matrices used in this framework.

Concretely, we sample $\mathbf{U}_A, \mathbf{U}_B \in \mathcal{V}_m(\mathbb{R}^d)$ from the uniform distribution⁴² on the Stiefel manifold

⁴²Following Chikuse (2003, Theorem 2.2.1 (iii)) and Eaton (2007, Proposition 7.2), we define the uniform distribution on $\mathcal{V}_m(\mathbb{R}^d)$ as follows. We sample $\mathbf{W} \in \mathbb{R}^{d \times m}$ with i.i.d. en-

of m orthonormal vectors in $\mathbb{R}^d$,

$$\mathcal{V}_m(\mathbb{R}^d) := \{\mathbf{U} \in \mathbb{R}^{d \times m} : \mathbf{U}^\top \mathbf{U} = \mathbf{I}_m\}, \quad (86)$$

and derive an approximate theoretical distribution of PK between the subspaces $\mathcal{S}_A$ and $\mathcal{S}_B$ spanned by the columns of $\mathbf{U}_A$ and $\mathbf{U}_B$. When $m = d$, $\mathcal{V}_d(\mathbb{R}^d)$ coincides with the orthogonal group in $\mathbb{R}^d$,

$$\mathcal{O}(\mathbb{R}^d) = \{\mathbf{R} \in \mathbb{R}^{d \times d} : \mathbf{R}^\top \mathbf{R} = \mathbf{R}\mathbf{R}^\top = \mathbf{I}_d\}. \quad (87)$$

M.1 Simplifying the setting using rotation invariance of PK

By the rotation invariance of PK (see Appendix D.5), letting $\mathbf{e}_i^{\text{Std}} \in \mathbb{R}^d$ denote the i -th standard basis vector, we may fix

$$\mathbf{U}_A = (\mathbf{e}_1^{\text{Std}}, \dots, \mathbf{e}_m^{\text{Std}}) = \begin{pmatrix} \mathbf{I}_m \\ \mathbf{0} \end{pmatrix} \in \mathcal{V}_m(\mathbb{R}^d), \quad (88)$$

without loss of generality, where $\mathbf{I}_m \in \mathbb{R}^{m \times m}$ is the m -dimensional identity matrix.

Next, we sample a random orthogonal matrix $\mathbf{R} \in \mathcal{O}(\mathbb{R}^d)$ from the uniform distribution on $\mathcal{O}(\mathbb{R}^d)$ and define⁴³

$$\mathbf{U}_B = \mathbf{R}\mathbf{U}_A = \mathbf{R}_{[:,1:m]}, \quad (89)$$

where $\mathbf{R}_{[:,1:m]}$ denotes the $d \times m$ submatrix of $\mathbf{R}$ consisting of its first m columns, so that $\mathbf{U}_B \in \mathcal{V}_m(\mathbb{R}^d)$.

Since $\mathbf{R}$ is random, the PK value between the subspaces $\mathcal{S}_A$ and $\mathcal{S}_B$ spanned by $\mathbf{U}_A$ and $\mathbf{U}_B$ is a random variable. Letting this random variable be Z , from (52), (88), and (89), we obtain

$$\begin{aligned} Z = \text{PK}(\mathcal{S}_A, \mathcal{S}_B) &= \left\| \mathbf{U}_A^\top \mathbf{U}_B \right\|_{\text{F}}^2 \\ &= \left\| \mathbf{U}_A^\top \mathbf{R} \mathbf{U}_A \right\|_{\text{F}}^2 = \left\| \mathbf{R}_{[1:m], [1:m]} \right\|_{\text{F}}^2 \\ &= \sum_{i=1}^m \sum_{j=1}^m R_{ij}^2, \end{aligned} \quad (90)$$

where $\mathbf{R}_{[1:m], [1:m]}$ denotes the top-left $m \times m$ block of $\mathbf{R}$, R_{ij} is its (i, j) entry, and $\|\cdot\|_{\text{F}}$ is the Frobenius norm.

tries from $\mathcal{N}(0, 1)$, and orthonormalize its columns using the Gram–Schmidt procedure (or equivalently QR decomposition or SVD). We use the resulting matrix $\mathbf{U}$ as a sample from the uniform distribution.

⁴³The resulting $\mathbf{U}_B$ follows the uniform distribution on $\mathcal{V}_m(\mathbb{R}^d)$; see Chikuse (2003, Section 2.5.1).

M.2 Loose approximation: approximating the distribution of R_{ij} as Gaussian and applying the central limit theorem

The entries R_{ij} of the random orthogonal matrix $\mathbf{R}$ sampled in Appendix M.1 are not independent due to orthogonality constraints. As a loose approximation, we consider an approximate distribution of Z under the assumption that the entries behave independently.

For each entry R_{ij} , Meckes (2019, Corollary 2.6) showed that⁴⁴

$$\sqrt{d} R_{ij} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1) \quad (d \rightarrow \infty). \quad (91)$$

Consequently, for large d , R_{ij} is approximately distributed as

$$\mathcal{N}\left(0, \frac{1}{d}\right). \quad (92)$$

We therefore approximate R_{ij} by independent random variables

$$X_{ij} \sim \mathcal{N}\left(0, \frac{1}{d}\right). \quad (93)$$

For $d = 768$ and $m = 64$, which match the dimensions of attention weights in GPT2-small, we sample $\mathbf{U}_B = \mathbf{R}_{[:,1:m]} \in \mathcal{V}_m(\mathbb{R}^d)$ and find that the density of $\mathcal{N}(0, 1/d)$ provides a good fit to the histogram of R_{ij} (see Fig. 34a)⁴⁵.

For X_{ij}^2 , let $Y_{ij} := \sqrt{d} X_{ij}$. Then $Y_{ij} \sim \mathcal{N}(0, 1)$, and hence Y_{ij}^2 follows a chi-square distribution with one degree of freedom, χ_1^2 . Since $X_{ij}^2 = Y_{ij}^2/d$, it follows that X_{ij}^2 follows a Gamma distribution⁴⁶ with shape $k = 1/2$ and scale $\theta = 2/d$,

$$X_{ij}^2 \sim \text{Gamma}\left(\frac{1}{2}, \frac{2}{d}\right). \quad (94)$$

⁴⁴Here, $\xrightarrow{\mathcal{D}}$ denotes convergence in distribution (weak convergence): for random variables X_d and X , $X_d \xrightarrow{\mathcal{D}} X$ means that the cumulative distribution functions satisfy $F_{X_d}(\tau) \rightarrow F_X(\tau)$ as $d \rightarrow \infty$ for every τ at which F_X is continuous, where $F_Y(\tau) := \mathbb{P}(Y \leq \tau)$ (Pishro-Nik, 2014, Section 7.2.4). Here, $\mathbb{P}(\cdot)$ denotes probability under the underlying probability measure. Equivalently, we write $X_d \xrightarrow{\mathcal{D}} \mathcal{L}$ if there exists a random variable X such that $X_d \xrightarrow{\mathcal{D}} X$ and X has probability distribution $\mathcal{L}$.

⁴⁵This $\mathcal{N}(0, 1/d)$ distribution also appears in a related high-dimensional limit: when d is sufficiently large, the cosine similarity between two d -dimensional random vectors is approximately distributed as $\mathcal{N}(0, 1/d)$, provided that each vector has mean-zero components with finite variance. The component variances are shared within each vector but may differ across the two vectors; see Yamagiwa et al. (2024, 2025).

⁴⁶In general, χ_ν^2 coincides with $\text{Gamma}(\nu/2, 2)$. Moreover, if $Y \sim \text{Gamma}(\nu/2, 2)$ and $X = cY$ with $c > 0$, then $X \sim \text{Gamma}(\nu/2, 2c)$.

Similarly, for $d = 768$ and $m = 64$, we sample $\mathbf{U}_B = \mathbf{R}_{[:,1:m]} \in \mathcal{V}_m(\mathbb{R}^d)$ and find that the density of $\text{Gamma}(1/2, 2/d)$ provides a good fit to the histogram of R_{ij}^2 (see Fig. 34b). From the mean and variance formulas of the Gamma distribution (Heckert et al., 2002, Section 8.1.6.5), the mean and variance of X_{ij}^2 are given by

$$\mathbb{E}[X_{ij}^2] = k\theta = \frac{1}{d}, \quad (95)$$

$$\text{Var}(X_{ij}^2) = k\theta^2 = \frac{2}{d^2}. \quad (96)$$

Using the i.i.d. Gaussian variables X_{ij} introduced in (93), we define the corresponding $d \rightarrow \infty$ approximation to the Projection Kernel by

$$Z^{(\infty)} := \sum_{i=1}^m \sum_{j=1}^m X_{ij}^2. \quad (97)$$

Since $\{X_{ij}\}$ are i.i.d., the family $\{X_{ij}^2\}$ is also i.i.d. and satisfies $\text{Var}(X_{ij}^2) < \infty$. Hence, by the central limit theorem,

$$\frac{Z^{(\infty)} - m^2 \mathbb{E}[X_{ij}^2]}{\sqrt{m^2 \text{Var}(X_{ij}^2)}} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1) \quad (m \rightarrow \infty). \quad (98)$$

Substituting (95) and (96) into (98), we obtain

$$\frac{Z^{(\infty)} - \frac{m^2}{d}}{\sqrt{\frac{2m^2}{d^2}}} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1) \quad (m \rightarrow \infty). \quad (99)$$

Consequently, for large m , the distribution of $Z^{(\infty)}$ is well approximated by

$$\mathcal{N}\left(\frac{m^2}{d}, \frac{2m^2}{d^2}\right). \quad (100)$$

For $d = 768$ and $m = 64$, sampling⁴⁷ $\mathbf{U}_A, \mathbf{U}_B \in \mathcal{V}_m(\mathbb{R}^d)$ and overlaying the density of $\mathcal{N}(m^2/d, 2m^2/d^2)$ on the histogram of Z shows a rough match (Fig. 34c). However, due to both the large- d approximation in (92) and the large- m approximation in (100), the fit is weaker than for the approximations of R_{ij} and R_{ij}^2 (Figs. 34a and 34b).

M.3 Tight approximation: deriving the mean and variance of PK and applying the central limit theorem

In Appendix M.2, we derived an approximate distribution of Z from the approximate mean (95) and

⁴⁷In the simulation, instead of using the identity-based $\mathbf{U}_A$ in (88), we generated $\mathbf{U}_A$ and $\mathbf{U}_B$ by first sampling $d \times m$ matrices whose entries are i.i.d. standard normal $\mathcal{N}(0, 1)$, and then taking their SVD to obtain orthonormal basis matrices.

approximate variance (96) under the independence assumption in (93). In this section, we derive a tighter approximate distribution of Z by computing its mean and variance from the distribution of R_{ij} . To this end, we first formulate $\mathbb{E}[Z]$ and $\text{Var}(Z)$, compute them, and then derive an approximate distribution based on the central limit theorem.

M.3.1 Formulating the mean $\mathbb{E}[Z]$ and variance $\text{Var}(Z)$ of Z

From (90), the mean of Z is

$$\mathbb{E}[Z] = \sum_{i=1}^m \sum_{j=1}^m \mathbb{E}[R_{ij}^2], \quad (101)$$

and the variance of Z is

$$\text{Var}(Z) = \mathbb{E}[(Z - \mathbb{E}[Z])^2] = \mathbb{E}[Z^2] - \mathbb{E}[Z]^2. \quad (102)$$

Thus, to compute $\text{Var}(Z)$, it suffices to compute $\mathbb{E}[Z]$ and $\mathbb{E}[Z^2]$.

Expanding Z^2 yields

$$Z^2 = \sum_{i=1}^m \sum_{j=1}^m \sum_{i'=1}^m \sum_{j'=1}^m R_{ij}^2 R_{i'j'}^2, \quad (103)$$

so

$$\mathbb{E}[Z^2] = \sum_{i=1}^m \sum_{j=1}^m \sum_{i'=1}^m \sum_{j'=1}^m \mathbb{E}[R_{ij}^2 R_{i'j'}^2]. \quad (104)$$

M.3.2 Computing the mean $\mathbb{E}[Z]$

The density and mean of R_{ij} are given by Preisendorfer and Mobley (1982, Appendix E):

$$f_{R_{ij}}(x) = \frac{1}{B\left(\frac{1}{2}, \frac{d-1}{2}\right)} (1-x^2)^{\frac{d-3}{2}} \quad x \in [-1, 1], \quad (105)$$

$$\mathbb{E}[R_{ij}] = 0, \quad (106)$$

where B is the Beta function. The density in (105) closely matches the histogram of R_{ij} in Fig. 34a.

The distribution and mean of R_{ij}^2 are given by Preisendorfer and Mobley (1982, Appendix E):

$$R_{ij}^2 \sim \text{Beta}\left(\frac{1}{2}, \frac{d-1}{2}\right), \quad (107)$$

$$\mathbb{E}[R_{ij}^2] = \frac{1}{d}. \quad (108)$$

The density of $\text{Beta}(1/2, (d-1)/2)$ also closely matches the histogram of R_{ij}^2 in Fig. 34b. From

(108), we have $\text{Var}(R_{ij}) = \mathbb{E}[R_{ij}^2] - \mathbb{E}[R_{ij}]^2 = 1/d$.

Using (108) in (101), we obtain⁴⁸

$$\mathbb{E}[Z] = \sum_{i=1}^m \sum_{j=1}^m \mathbb{E}[R_{ij}^2] = m^2 \mathbb{E}[R_{ij}^2] = \frac{m^2}{d}. \quad (109)$$

This matches the mean of the loose approximation in (100) for $Z^{(\infty)}$ in (97). Moreover, when $m = d$, we have $\mathbf{U}_A, \mathbf{U}_B \in \mathcal{V}_m(\mathbb{R}^d) = \mathcal{O}(\mathbb{R}^d)$, so both are full-rank orthogonal matrices. In this case, the corresponding subspaces satisfy $\mathcal{S}_A = \mathcal{S}_B = \mathbb{R}^d$, and hence $Z = \text{PK}(\mathcal{S}_A, \mathcal{S}_B) = d$. Consistently, $m^2/d = d^2/d = d$.

M.3.3 Computing $\mathbb{E}[Z^2]$ and $\text{Var}(Z)$

Since $\mathbf{R} \in \mathcal{O}(\mathbb{R}^d)$ is a random orthogonal matrix, the entries R_{ij} are dependent within each row and column. To compute the summand $\mathbb{E}[R_{ij}^2 R_{i'j'}^2]$ in (104), we consider the following four cases⁴⁹:

(i) If $i' = i$ and $j' = j$, then $\mathbb{E}[R_{ij}^2 R_{i'j'}^2] = \mathbb{E}[R_{ij}^4]$, and there are m^2 such terms.

(ii) If $i' = i$ and $j' \neq j$, then $\mathbb{E}[R_{ij}^2 R_{i'j'}^2] = \mathbb{E}[R_{ij}^2 R_{ij'}^2]$, and there are $m^2(m-1)$ such terms.

(iii) If $i' \neq i$ and $j' = j$, then $\mathbb{E}[R_{ij}^2 R_{i'j'}^2] = \mathbb{E}[R_{ij}^2 R_{i'j}^2]$, and there are $m^2(m-1)$ such terms.

(iv) If $i' \neq i$ and $j' \neq j$, then $\mathbb{E}[R_{ij}^2 R_{i'j'}^2]$ appears, and there are $m^2(m-1)^2$ such terms.

Therefore, we can write

$$\begin{aligned} \mathbb{E}[Z^2] &= m^2 \mathbb{E}[R_{ij}^4] + m^2(m-1) \mathbb{E}[R_{ij}^2 R_{ij'}^2] \\ &\quad + m^2(m-1) \mathbb{E}[R_{ij}^2 R_{i'j}^2] \\ &\quad + m^2(m-1)^2 \mathbb{E}[R_{ij}^2 R_{i'j'}^2]. \end{aligned} \quad (110)$$

For entries $R_{ij}, R_{rs}, R_{\alpha\beta}, R_{\lambda\mu} \in \mathbb{R}$ of a random orthogonal matrix $\mathbf{R} \in \mathcal{O}(\mathbb{R}^d)$, letting

⁴⁸The expected Projection Kernel between subspaces spanned by random orthogonal matrices, m^2/d , can also be understood naturally through the following discrete analogy. Uniformly at random, choose m of d boxes and place a red ball in each selected box, and independently repeat the procedure with blue balls. Each box is selected in both trials with probability $(m/d)^2$, so the expected number of boxes containing both a red and a blue ball is m^2/d . If each box is identified with a vector in a fixed orthonormal basis, selecting boxes corresponds to choosing basis vectors to form subspaces (see Section 3.3).

⁴⁹These four cases exhaust all m^4 pairs $((i, j), (i', j'))$: $m^2 + 2m^2(m-1) + m^2(m-1)^2 = m^4$.

δ_{ij} be the Kronecker delta, the fourth moment $\mathbb{E}[R_{ij}R_{rs}R_{\alpha\beta}R_{\lambda\mu}]$ is given by Meckes (2019, Lemma 2.2):

$$\begin{aligned} & \mathbb{E}[R_{ij}R_{rs}R_{\alpha\beta}R_{\lambda\mu}] \\ &= -\frac{1}{(d-1)d(d+2)}(\delta_{ir}\delta_{\alpha\lambda}\delta_{j\beta}\delta_{s\mu} \\ & \quad + \delta_{ir}\delta_{\alpha\lambda}\delta_{j\mu}\delta_{s\beta} + \delta_{i\alpha}\delta_{r\lambda}\delta_{j\beta}\delta_{s\mu} \\ & \quad + \delta_{i\alpha}\delta_{r\lambda}\delta_{j\mu}\delta_{s\beta} + \delta_{i\lambda}\delta_{r\alpha}\delta_{j\beta}\delta_{s\mu} \\ & \quad + \delta_{i\lambda}\delta_{r\alpha}\delta_{j\mu}\delta_{s\beta}) \\ & \quad + \frac{d+1}{(d-1)d(d+2)}(\delta_{ir}\delta_{\alpha\lambda}\delta_{j\beta}\delta_{s\mu} \\ & \quad + \delta_{i\alpha}\delta_{r\lambda}\delta_{j\beta}\delta_{s\mu} + \delta_{i\lambda}\delta_{r\alpha}\delta_{j\mu}\delta_{s\beta}). \end{aligned} \quad (111)$$

Setting $r = i$, $s = j$, $\alpha = \lambda = i'$, and $\beta = \mu = j'$ in (111) yields

$$\begin{aligned} & \mathbb{E}[R_{ij}^2 R_{i'j'}^2] \\ &= -\frac{2}{(d-1)d(d+2)}(\delta_{jj'} + \delta_{ii'} + \delta_{ii'}\delta_{jj'}) \\ & \quad + \frac{d+1}{(d-1)d(d+2)}(1 + 2\delta_{ii'}\delta_{jj'}). \end{aligned} \quad (112)$$

Here we used $\delta_{\alpha\beta}^2 = \delta_{\alpha\beta}$ and $\delta_{\alpha\beta} = \delta_{\beta\alpha}$.

Setting $i' = i$ and $j' = j$ in (112), we obtain

$$\mathbb{E}[R_{ij}^4] = \frac{3}{d(d+2)}. \quad (113)$$

Similarly, setting $i' = i$ and $j \neq j'$ yields

$$\mathbb{E}[R_{ij}^2 R_{ij'}^2] = \frac{1}{d(d+2)}. \quad (114)$$

Setting $i' \neq i$ and $j' = j$ yields

$$\mathbb{E}[R_{ij}^2 R_{i'j}^2] = \frac{1}{d(d+2)}. \quad (115)$$

Finally, setting $i' \neq i$ and $j' \neq j$ yields

$$\mathbb{E}[R_{ij}^2 R_{i'j'}^2] = \frac{d+1}{(d-1)d(d+2)}. \quad (116)$$

Note that (114) and (115) imply⁵⁰ $\mathbb{E}[R_{ij}^2 R_{i'j'}^2] = \mathbb{E}[R_{ij}^2 R_{i'j}^2]$.

Substituting (113), (114), (115), and (116) into (110), we obtain

$$\begin{aligned} & \mathbb{E}[Z^2] \\ &= \frac{m^2}{(d-1)d(d+2)} \{(m^2 + 2)d + m(m-4)\}. \end{aligned} \quad (117)$$

⁵⁰This also follows from symmetry.

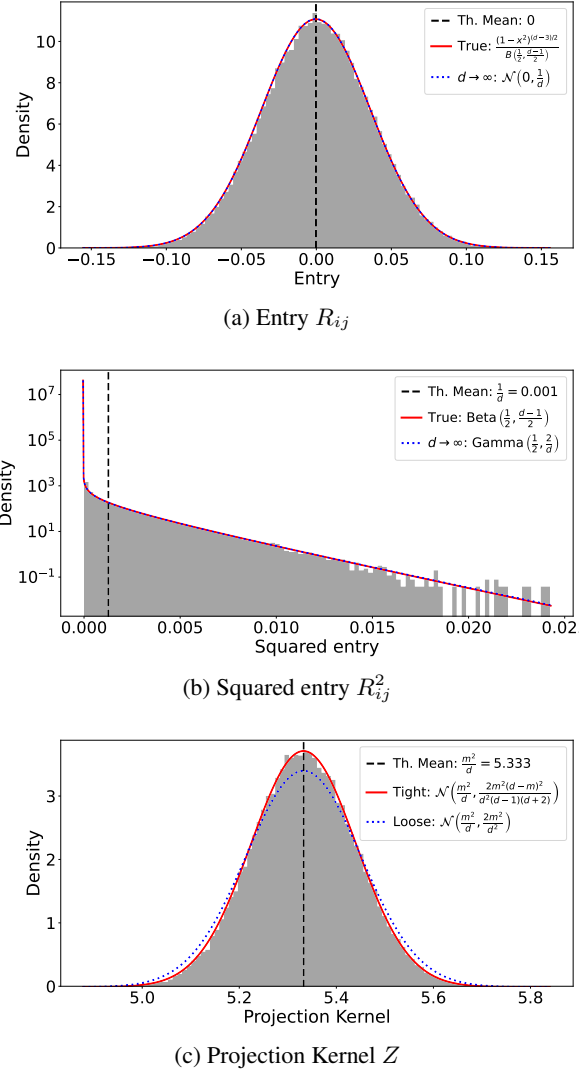


Figure 34: For random orthogonal matrices in $\mathcal{V}_m(\mathbb{R}^d)$, we show (a) a histogram of entry values, (b) squared entry values, and (c) PK values between pairs of random orthogonal matrices. We use $d = 768$ and $m = 64$, following GPT2-small; (a) and (b) are based on 100,000 randomly sampled entries, and (c) on 100,000 randomly sampled pairs. For (a) and (b), we also plot the corresponding density and its asymptotic approximation as $d \rightarrow \infty$. For (c), we plot two approximations to the PK distribution: the tight approximation in Appendix M.3 and the loose approximation in Appendix M.2. The two approximations are nearly indistinguishable for (a) and (b), while for (c) the tight approximation matches the histogram better, especially near the mean.

Using (109) and (117), we compute $\text{Var}(Z)$ in (102) as

$$\text{Var}(Z) = \frac{2m^2(d-m)^2}{d^2(d-1)(d+2)}. \quad (118)$$

In particular, when $m \ll d$, we have $(d-m)^2 \approx d^2 \approx (d-1)(d+2)$ and thus $\text{Var}(Z) \approx 2m^2/d^2$,

	GPT2-small	BLOOM-560M	Qwen2-0.5B-Instruct
# parameters	124M	560M	494M
# layers	12	24	24
# query heads	12	16	14
# KV heads	12	16	2
Attention architecture	MHA	MHA	GQA
d	768	1024	896
d_{head}	64	64	64
FFN architecture	GELU MLP	GELU MLP	SwiGLU
d_{FFN}	3072	4096	4864
Normalization	LayerNorm	LayerNorm	RMSNorm
Positional encoding	learned absolute embeddings	ALiBi	RoPE

Table 13: Architectural configurations of the models used in our experiments. In particular, GPT2-small and BLOOM-560M use multi-head attention (MHA) (Vaswani et al., 2017), whereas Qwen2-0.5B-Instruct uses grouped-query attention (GQA) (Ainslie et al., 2023) and thus has fewer KV heads than query heads.

which matches the variance of the loose approximation in (100) for $Z^{(\infty)}$ in (97). Moreover, when $m = d$, we have seen that $Z = \text{PK}(\mathcal{S}_A, \mathcal{S}_B) = d$ for any $\mathbf{U}_A, \mathbf{U}_B \in \mathcal{V}_m(\mathbb{R}^d) = \mathcal{O}(\mathbb{R}^d)$, and hence $\text{Var}(Z) = 0$. This also follows from (118), since substituting $m = d$ yields $\text{Var}(Z) = 0$.

M.3.4 Approximating the distribution of Z using the central limit theorem

Although Z can be written as a sum of m^2 random variables R_{ij}^2 , these variables are not independent due to the orthogonality constraints on R_{ij} . Therefore, unlike Appendix M.2, we cannot neither treat Z as a sum of independent variables nor directly apply the central limit theorem.

To assess the strength of these dependencies, we compare the variance and covariances of R_{ij}^2 . From (108) and (113), the variance is

$$\begin{aligned} \text{Var}(R_{ij}^2) &= \mathbb{E}[R_{ij}^4] - \mathbb{E}[R_{ij}^2]^2 \\ &= \frac{2(d-1)}{d^2(d+2)} = O\left(\frac{1}{d^2}\right). \end{aligned} \quad (119)$$

For $j \neq j'$, from (108) and (114), the covariance is

$$\begin{aligned} \text{Cov}(R_{ij}^2, R_{ij'}^2) &= \mathbb{E}[R_{ij}^2 R_{ij'}^2] - \mathbb{E}[R_{ij}^2] \mathbb{E}[R_{ij'}^2] \\ &= -\frac{2}{d^2(d+2)} = O\left(\frac{1}{d^3}\right). \end{aligned} \quad (120)$$

The same order holds for $\text{Cov}(R_{ij}^2, R_{i'j}^2)$ when $i \neq i'$. For $i \neq i'$ and $j \neq j'$, from (108) and (116), we have

$$\begin{aligned} \text{Cov}(R_{ij}^2, R_{i'j'}^2) &= \mathbb{E}[R_{ij}^2 R_{i'j'}^2] - \mathbb{E}[R_{ij}^2] \mathbb{E}[R_{i'j'}^2] \\ &= \frac{2}{d^2(d-1)(d+2)} = O\left(\frac{1}{d^4}\right). \end{aligned} \quad (121)$$

Thus, the covariances in (120) and (121) are at least a factor of $1/d$ smaller than the variance in (119).

These estimates suggest that Z behaves like a sum of m^2 nearly independent random variables R_{ij}^2 . We therefore approximate the distribution of Z by a Gaussian distribution⁵¹. Using the mean in (109) and the variance in (118), we adopt

$$\begin{aligned} &\mathcal{N}(\mathbb{E}[Z], \text{Var}(Z)) \\ &= \mathcal{N}\left(\frac{m^2}{d}, \frac{2m^2(d-m)^2}{d^2(d-1)(d+2)}\right) \end{aligned} \quad (122)$$

as a tight approximate distribution of Z . Overlaying the density of (122) on the histogram of Z in Fig. 34c shows a better fit than the loose approximation in Appendix M.2.

N Details of experiments with BLOOM-560M and Qwen2-0.5B-Instruct

In Section 8.4, we used BLOOM-560M and Qwen2-0.5B-Instruct to conduct the same evaluation as in Section 7. This section provides further details of these experiments.

N.1 Differences in model architecture

Table 13 summarizes the architectural differences among GPT2-small (Radford et al., 2019), which is the primary model used in this work, and BLOOM-560M (BigScience Workshop, 2024) and Qwen2-0.5B-Instruct (Yang et al., 2024), which are used in Section 8.4.

⁵¹We do not verify conditions required for a rigorous application of the central limit theorem, such as independence or specific dependence structures, and instead adopt a Gaussian approximation as a practical model for the distribution of Z .

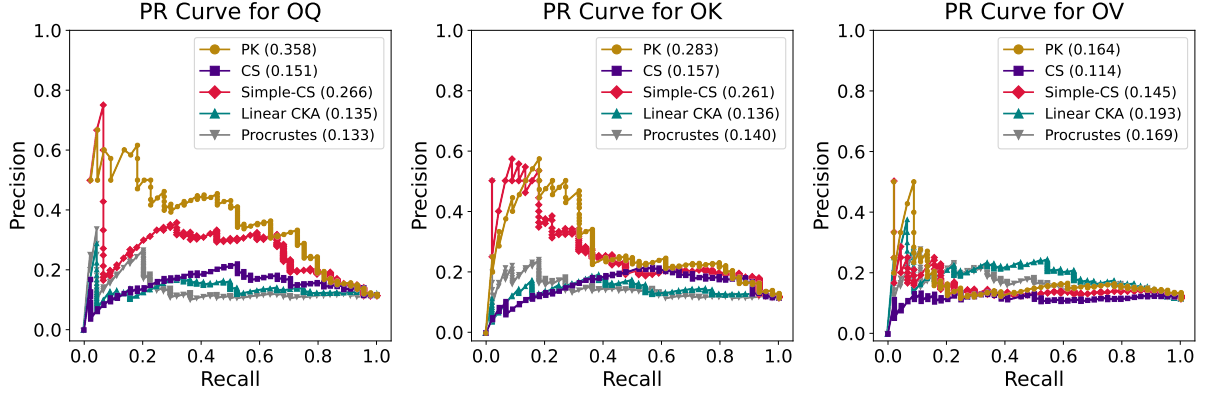


Figure 35: PR curves of head detection for OQ, OK, and OV in BLOOM-560M, shown for each method.

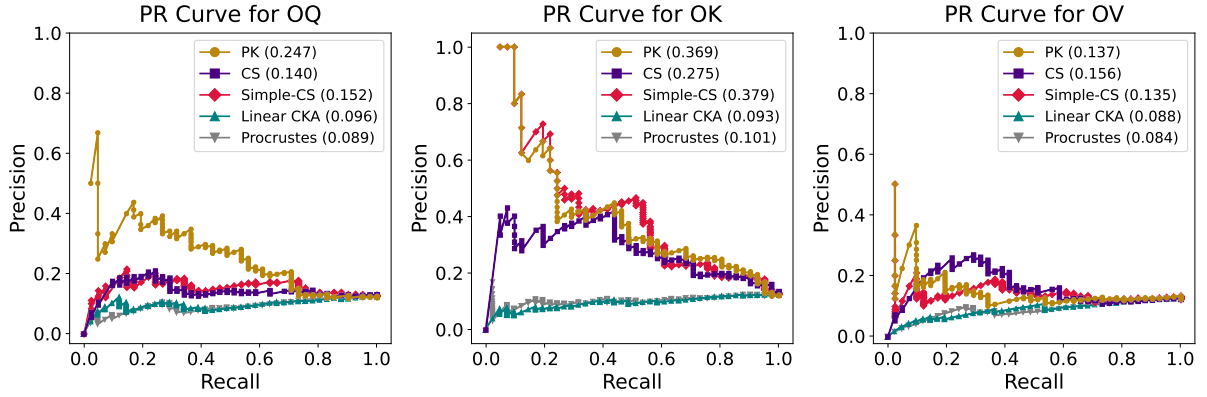


Figure 36: PR curves of head detection for OQ, OK, and OV in Qwen2-0.5B-Instruct, shown for each method.

GPT2-small and BLOOM-560M share many features of classical Transformer architectures, including multi-head attention (MHA) (Vaswani et al., 2017), a GELU (Hendrycks and Gimpel, 2016) MLP as the FFN, and layer normalization (Ba et al., 2016). However, they differ in positional encoding: GPT2-small uses learned absolute embeddings, whereas BLOOM-560M uses ALiBi (Press et al., 2022). Qwen2-0.5B-Instruct, on the other hand, has a more modern architecture based on grouped-query attention (GQA) (Ainslie et al., 2023), with SwiGLU (Shazeer, 2020) as its FFN and root mean square layer normalization (RMSNorm) (Zhang and Sennrich, 2019). It further uses RoPE (Su et al., 2024) for positional encoding.

A particularly important difference lies in the attention architecture. While GPT2-small and BLOOM-560M use MHA, Qwen2-0.5B-Instruct uses GQA and has only two key and value heads for 14 query heads in each layer. For Qwen2-0.5B-Instruct, we analyzed the GQA architecture within the same framework as MHA by simply repeating the shared key and value head weights.

N.2 Details of head scores and head class annotations

In this section, as in Appendix H, we assign heads in BLOOM-560M and Qwen2-0.5B-Instruct to head classes based on their head scores. Tables 14 and 15 show the top 10 head scores and the corresponding heads for Identity, Duplicate Token, Previous Token, and Induction Heads in BLOOM-560M and Qwen2-0.5B-Instruct, respectively. As with GPT2-small (Table 8 in Appendix H.1), multiple heads achieve high head scores for each head class. Based on these results and the head class annotations of Zhang et al. (2025), we define the head classes shown in Tables 16 and 17.

N.3 Details of the evaluation of each method

This section provides further details of Tables 4 and 5 in Section 8.4.

N.3.1 Details of head detection performance for OQ, OK, and OV

Figures 35 and 36 show the PR curves corresponding to the PR-AUC values in Table 4 for BLOOM-560M and Qwen2-0.5B-Instruct, respectively. As

Head	Score	Head	Score	Head	Score	Head	Score
L7H8	0.923	L6H14	0.323	L6H0	0.996	L10H10	0.972
L0H5	0.867	L11H10	0.209	L6H2	0.995	L8H13	0.963
L21H0	0.810	L1H14	0.187	L7H0	0.982	L7H12	0.957
L6H3	0.638	L3H12	0.161	L5H0	0.970	L8H11	0.955
L23H1	0.620	L11H9	0.054	L9H3	0.888	L10H12	0.949
L20H0	0.577	L13H13	0.051	L11H0	0.875	L10H13	0.936
L22H4	0.524	L11H11	0.046	L4H0	0.731	L9H14	0.878
L1H3	0.511	L4H15	0.043	L8H0	0.726	L13H12	0.873
L12H0	0.496	L0H11	0.034	L17H0	0.696	L18H13	0.864
L5H2	0.490	L6H13	0.034	L11H5	0.544	L8H14	0.858
(a) Identity		(b) Duplicate Token		(c) Previous Token		(d) Induction	

Table 14: For Identity, Duplicate Token, Previous Token, and Induction Heads in BLOOM-560M, we list the top 10 heads and their head scores. See Appendix H for how we compute these scores.

Head	Score	Head	Score	Head	Score	Head	Score
L10H7	1.000	L7H7	0.893	L8H10	0.906	L11H11	0.988
L23H0	0.998	L5H1	0.863	L0H5	0.786	L9H12	0.940
L16H11	0.981	L1H1	0.837	L10H13	0.773	L11H13	0.924
L8H6	0.958	L5H0	0.782	L15H4	0.724	L16H1	0.912
L17H10	0.935	L12H2	0.617	L1H5	0.717	L11H12	0.853
L17H5	0.905	L9H0	0.282	L5H5	0.665	L9H8	0.621
L12H0	0.892	L4H9	0.277	L20H9	0.604	L11H9	0.597
L10H0	0.863	L4H2	0.220	L4H0	0.545	L16H6	0.566
L6H3	0.830	L5H6	0.138	L2H3	0.542	L20H7	0.547
L14H1	0.802	L7H8	0.104	L0H7	0.524	L9H9	0.497
(a) Identity		(b) Duplicate Token		(c) Previous Token		(d) Induction	

Table 15: For Identity, Duplicate Token, Previous Token, and Induction Heads in Qwen2-0.5B-Instruct, we list the top 10 heads and their head scores. See Appendix H for how we compute these scores.

Head Classes	Count	Heads
Duplicate Token Heads	4	L1H14*, L3H12*, L6H14*, L11H10
Previous Token Heads	10	L4H0, L5H0, L6H0, L6H2, L7H0, L8H0*, L9H3*, L11H0*, L11H5*, L17H0*
Induction Heads	10	L7H12, L8H11, L8H13, L8H14*, L9H14*, L10H10, L10H12*, L10H13*, L13H12*, L18H13*
Name Mover Heads	11	L18H7, L18H8, L19H8, L19H11, L20H8, L20H9, L20H10, L20H11, L20H12, L21H9, L21H10
Negative Name Mover Heads	3	L17H9, L19H15, L21H6
S-Inhibition Heads	6	L12H6, L14H6, L16H5, L19H4, L21H2, L22H3

Table 16: Head classes and their corresponding heads in our experimental setup for BLOOM-560M. Heads that are not included in Zhang et al. (2025, Fig. 2) are marked with “*”. The total number of annotated heads is 44.

Head Classes	Count	Heads
Duplicate Token Heads	5	L1H1*, L5H0, L5H1*, L7H7, L12H2*
Previous Token Heads	10	L0H5*, L0H7*, L1H5, L2H3*, L4H0, L5H5, L8H10*, L10H13*, L15H4*, L20H9*
Induction Heads	9	L9H8*, L9H9, L9H12, L11H9*, L11H11, L11H12*, L11H13, L16H1*, L16H6*
Name Mover Heads	6	L16H0, L20H11, L21H5, L21H6, L21H11, L21H12
Negative Name Mover Heads	3	L20H7, L22H6, L23H12
S-Inhibition Heads	8	L13H8, L14H8, L14H10, L15H7, L16H5, L17H6, L21H0, L22H8

Table 17: Head classes and their corresponding heads in our experimental setup for Qwen2-0.5B-Instruct. Heads that are not included in Zhang et al. (2025, Fig. 9) are marked with “*”. The total number of annotated heads is 41.

Top	OQ		OK		OV	
1	L17H9 (N-NM)	340.46	L6H12	372.52	L21H5	262.06
2	L6H2 (Prev)	198.95	L17H6	264.11	L17H8	188.13
3	L19H10	182.18	L11H5 (Prev)	234.30	L5H2 (Id)	180.28
4	L19H15 (N-NM)	179.84	L5H9	180.51	L18H6	151.28
5	L5H0 (Prev)	144.04	L13H15	167.87	L11H4	131.25
6	L22H14	109.76	L7H15	137.67	L6H3 (Id)	111.10
7	L18H12	105.52	L23H12	107.98	L10H3	107.47
8	L10H9	96.50	L8H13 (Ind)	103.30	L7H3	104.10
9	L10H13 (Ind)	89.25	L15H13	99.56	L20H1	103.87
10	L10H8	86.51	L22H9	97.46	L9H4	98.08

(a) Inlet Head Scores

Top	OQ		OK		OV	
1	L6H13	323.58	L6H3 (Id)	627.11	L0H3	248.62
2	L12H2	263.44	L17H0 (Prev)	412.18	L13H13	144.67
3	L17H8	206.67	L0H5 (Id)	180.32	L6H13	120.39
4	L15H10	189.91	L13H13	128.17	L11H9	114.29
5	L5H2 (Id)	140.45	L9H0	106.13	L6H3 (Id)	110.92
6	L15H13	131.03	L11H4	100.69	L11H11	108.36
7	L17H9 (N-NM)	120.90	L1H13	92.80	L11H10 (Dup)	107.61
8	L11H4	117.98	L6H14 (Dup)	84.50	L1H3 (Id)	97.84
9	L9H4	106.55	L6H13	82.84	L3H9	89.75
10	L2H1	104.56	L11H10 (Dup)	81.41	L6H14 (Dup)	85.67

(b) Outlet Head Scores

Table 18: For OQ, OK, and OV in BLOOM-560M, we report (a) the top 10 heads and their inlet scores in (83) and (b) the top 10 heads and their outlet scores in (85). When a head belongs to the head classes in Table 16 or to the top 10 heads by Identity Head score in Table 14a, we also report this information. Dup, Prev, Ind, NM, N-NM, and S-Inh denote Duplicate Token, Previous Token, Induction, Name Mover, Negative Name Mover, and S-Inhibition heads, respectively.

with GPT2-small (Fig. 33 in Appendix K.1), for OQ and OK, the PR curve of PK dominates the other curves or achieves a comparable AUC. For OV, however, the PK curve is dominated by those of other methods, such as Linear CKA for BLOOM-560M and CS for Qwen2-0.5B-Instruct.

N.3.2 Details of classwise classification performance for QQ, KK, VV, and OO

Tables 19 and 20 report, for each head class in BLOOM-560M and Qwen2-0.5B-Instruct, respectively, the mean PR-AUC and ROC-AUC scores used to compute Table 5, with the mean taken over QQ, KK, VV, and OO. The tables also report PR-AUC and ROC-AUC separately for each of the four weight pairing types, QQ, KK, VV, and OO.

For PR-AUC, PK achieves the best average performance across head classes in Tables 19a and 20a for all five settings, namely QQ, KK, VV, OO, and their mean. Moreover, unlike for GPT2-small (Table 11a in Appendix K.2), PK achieves the best score or a score comparable to the best for many head classes across the five settings.

For ROC-AUC, Simple-CS outperforms PK for

BLOOM-560M in Table 19b, whereas CS outperforms PK for Qwen2-0.5B-Instruct in Table 20b. However, because the classification setting is extremely imbalanced⁵², we place greater emphasis on PR-AUC than on ROC-AUC.

N.4 Detecting hub heads in BLOOM-560M based on inlet and outlet scores

For BLOOM-560M, we examine whether, as in GPT2-small, heads that function as hub heads appear among those with high inlet and outlet head scores, and whether these heads are Identity Heads. Note that the repeated key and value head weights in GQA lead to many ties in the OK and OV inlet scores (83) in Appendix L. Therefore, we do not consider Qwen2-0.5B-Instruct in this analysis because its GQA architecture makes the hub analysis more difficult to interpret.

Table 18 shows the top 10 heads and their inlet and outlet head scores for OQ, OK, and OV. As with GPT2-small (Table 3 in Section 8.2), the

⁵²For example, Induction Head pairs account for only 45 of 73,536 pairs in BLOOM-560M and 36 of 56,280 pairs in Qwen2-0.5B-Instruct.

	Method	Dup (4)	Prev (10)	Ind (10)	NM (11)	N-NM (3)	S-Inh (6)	Avg
QQ	PK	0.001	0.041	0.090	0.183	<u>0.001</u>	<u>0.001</u>	0.053
	CS	0.000	0.005	0.011	0.021	<u>0.000</u>	<u>0.001</u>	0.006
	Simple-CS	<u>0.001</u>	<u>0.012</u>	<u>0.055</u>	<u>0.084</u>	0.021	0.001	<u>0.029</u>
	Linear CKA	0.000	0.001	0.000	0.001	0.000	0.000	0.000
	Proc. Sim.	0.000	0.001	0.001	0.001	0.000	0.000	0.000
KK	PK	0.001	0.014	0.140	0.095	0.001	0.001	0.042
	CS	0.000	0.002	0.005	0.010	0.000	0.001	0.003
	Simple-CS	<u>0.001</u>	<u>0.010</u>	<u>0.091</u>	<u>0.062</u>	<u>0.001</u>	<u>0.001</u>	<u>0.027</u>
	Linear CKA	0.000	0.001	0.000	0.001	0.000	0.000	0.000
	Proc. Sim.	0.000	0.001	0.000	0.001	0.000	0.000	0.000
VV	PK	0.000	0.012	0.006	<u>0.005</u>	<u>0.001</u>	<u>0.001</u>	0.004
	CS	0.001	0.007	0.004	0.002	0.001	0.001	0.003
	Simple-CS	<u>0.000</u>	<u>0.011</u>	<u>0.006</u>	0.003	0.001	0.001	<u>0.004</u>
	Linear CKA	0.000	0.000	0.001	0.011	0.000	0.002	0.002
	Proc. Sim.	0.000	0.001	0.001	0.004	0.000	0.001	0.001
OO	PK	0.000	0.035	0.008	0.001	0.001	<u>0.000</u>	0.008
	CS	0.001	0.005	0.002	0.001	0.000	0.000	0.002
	Simple-CS	<u>0.000</u>	<u>0.005</u>	0.003	<u>0.001</u>	<u>0.000</u>	0.000	<u>0.002</u>
	Linear CKA	0.000	0.001	<u>0.004</u>	0.001	0.000	0.001	0.001
	Proc. Sim.	0.000	0.003	0.003	0.001	0.000	0.000	0.001
Mean over QQ, KK, VV, OO	PK	0.001	0.026	0.061	0.071	<u>0.001</u>	0.001	0.027
	CS	<u>0.001</u>	0.005	0.006	0.008	0.000	0.001	0.003
	Simple-CS	0.000	<u>0.009</u>	<u>0.038</u>	<u>0.038</u>	0.006	0.001	<u>0.015</u>
	Linear CKA	0.000	0.001	0.001	0.003	0.000	<u>0.001</u>	0.001
	Proc. Sim.	0.000	0.001	0.001	0.002	0.000	0.000	0.001

(a) PR-AUC

	Method	Dup (4)	Prev (10)	Ind (10)	NM (11)	N-NM (3)	S-Inh (6)	Avg
QQ	PK	0.906	0.943	0.987	0.995	0.847	0.856	<u>0.923</u>
	CS	0.839	0.905	0.946	0.977	<u>0.898</u>	0.876	0.907
	Simple-CS	<u>0.877</u>	<u>0.929</u>	<u>0.978</u>	<u>0.991</u>	0.978	<u>0.875</u>	0.938
	Linear CKA	0.492	0.500	0.382	0.542	0.025	0.685	0.437
	Proc. Sim.	0.327	0.445	0.438	0.666	0.096	0.704	0.446
KK	PK	0.953	0.914	0.995	0.996	0.930	0.896	0.947
	CS	0.821	0.834	0.934	0.966	0.858	0.821	0.872
	Simple-CS	<u>0.919</u>	<u>0.886</u>	<u>0.989</u>	<u>0.993</u>	<u>0.915</u>	<u>0.866</u>	<u>0.928</u>
	Linear CKA	0.690	0.497	0.398	0.492	0.288	0.748	0.519
	Proc. Sim.	0.325	0.466	0.376	0.611	0.257	0.713	0.458
VV	PK	0.820	0.865	0.747	<u>0.858</u>	0.896	0.668	<u>0.809</u>
	CS	0.852	<u>0.880</u>	0.739	0.814	<u>0.910</u>	0.674	0.811
	Simple-CS	<u>0.828</u>	0.884	<u>0.744</u>	0.842	0.883	0.665	0.808
	Linear CKA	0.197	0.307	0.594	0.948	0.860	0.857	0.627
	Proc. Sim.	0.468	0.657	0.609	0.778	0.949	<u>0.773</u>	0.706
OO	PK	0.681	0.940	<u>0.825</u>	0.637	<u>0.909</u>	0.717	0.785
	CS	0.793	<u>0.910</u>	0.764	0.619	0.902	<u>0.734</u>	<u>0.787</u>
	Simple-CS	<u>0.748</u>	0.905	0.777	0.681	0.934	0.756	0.800
	Linear CKA	0.285	0.711	0.864	0.597	0.223	0.613	0.549
	Proc. Sim.	0.534	0.885	0.777	<u>0.644</u>	0.537	0.592	0.661
Mean over QQ, KK, VV, OO	PK	<u>0.840</u>	0.916	0.888	<u>0.871</u>	<u>0.896</u>	<u>0.784</u>	<u>0.866</u>
	CS	0.826	0.882	0.846	0.844	0.892	0.776	0.844
	Simple-CS	0.843	<u>0.901</u>	<u>0.872</u>	0.877	0.928	0.791	0.869
	Linear CKA	0.416	0.504	0.560	0.645	0.349	0.726	0.533
	Proc. Sim.	0.413	0.613	0.550	0.675	0.460	0.695	0.568

(b) ROC-AUC

Table 19: PR-AUC and ROC-AUC for each head class in BLOOM-560M, computed separately for the weight pairing types QQ, KK, VV, and OO, together with the mean over the four weight pairing types for each head class. These values are used to compute the summary reported in Table 5. The maximum is shown in bold and the second largest is underlined. We consider the $\binom{24 \times 16}{2} = \binom{384}{2} = 73,536$ directed head pairs whose source is in an earlier layer than the target; for heads in the same layer, the head with the smaller index is treated as the source. Dup, Prev, Ind, NM, N-NM, and S-Inh denote Duplicate Token, Previous Token, Induction, Name Mover, Negative Name Mover, and S-Inhibition heads, respectively.

	Method	Dup (5)	Prev (10)	Ind (9)	NM (6)	N-NM (3)	S-Inh (8)	Avg
QQ	PK	0.101	0.001	0.030	0.023	0.002	<u>0.003</u>	0.027
	CS	0.003	<u>0.001</u>	<u>0.020</u>	<u>0.013</u>	<u>0.001</u>	<u>0.002</u>	<u>0.007</u>
	Simple-CS	0.002	0.003	0.012	0.011	0.001	0.003	0.005
	Linear CKA	<u>0.004</u>	0.000	0.009	0.003	0.000	0.000	0.003
	Proc. Sim.	<u>0.001</u>	0.001	0.004	0.003	0.000	0.001	0.001
KK	PK	0.050	0.001	0.145	0.070	0.001	0.020	0.048
	CS	0.001	<u>0.001</u>	0.017	0.008	<u>0.001</u>	0.002	0.005
	Simple-CS	0.000	0.001	0.013	0.005	0.001	0.002	0.004
	Linear CKA	0.051	0.001	0.141	0.067	0.000	0.019	0.046
	Proc. Sim.	<u>0.050</u>	0.001	<u>0.142</u>	<u>0.067</u>	0.000	<u>0.019</u>	<u>0.046</u>
VV	PK	0.050	0.001	0.145	0.069	0.000	0.019	0.047
	CS	0.001	0.002	0.012	0.002	<u>0.000</u>	0.001	0.003
	Simple-CS	0.001	<u>0.001</u>	0.016	0.002	0.000	0.001	0.004
	Linear CKA	0.050	0.000	0.141	0.067	0.000	0.018	0.046
	Proc. Sim.	<u>0.050</u>	0.001	<u>0.141</u>	<u>0.067</u>	0.000	<u>0.019</u>	<u>0.046</u>
OO	PK	0.003	0.003	0.025	0.000	0.001	0.001	0.006
	CS	<u>0.003</u>	0.007	<u>0.009</u>	0.000	0.000	0.001	0.003
	Simple-CS	0.003	0.007	<u>0.007</u>	0.000	<u>0.001</u>	0.001	0.003
	Linear CKA	0.000	0.000	0.001	<u>0.001</u>	0.000	0.001	0.001
	Proc. Sim.	0.000	0.000	0.001	0.002	0.000	0.001	0.001
Mean over QQ, KK, VV, OO	PK	0.051	0.001	0.087	0.041	0.001	0.011	0.032
	CS	0.002	<u>0.003</u>	0.015	0.006	<u>0.001</u>	0.001	0.005
	Simple-CS	0.001	0.003	0.012	0.005	0.001	0.002	0.004
	Linear CKA	<u>0.026</u>	0.000	<u>0.073</u>	0.035	0.000	<u>0.010</u>	<u>0.024</u>
	Proc. Sim.	0.025	0.001	0.072	<u>0.035</u>	0.000	0.010	0.024

(a) PR-AUC

	Method	Dup (5)	Prev (10)	Ind (9)	NM (6)	N-NM (3)	S-Inh (8)	Avg
QQ	PK	0.875	0.494	0.887	0.980	0.988	<u>0.879</u>	0.850
	CS	0.935	<u>0.611</u>	0.939	<u>0.979</u>	<u>0.980</u>	0.874	<u>0.887</u>
	Simple-CS	<u>0.920</u>	0.683	0.918	0.976	0.962	0.889	0.891
	Linear CKA	0.889	0.252	0.731	0.721	0.935	0.537	0.677
	Proc. Sim.	0.713	0.381	0.593	0.759	0.560	0.553	0.593
KK	PK	0.612	<u>0.371</u>	0.922	0.959	0.954	<u>0.828</u>	0.774
	CS	0.717	0.404	0.976	0.975	0.966	0.818	0.809
	Simple-CS	0.610	0.360	<u>0.958</u>	<u>0.967</u>	<u>0.966</u>	0.843	<u>0.784</u>
	Linear CKA	0.855	0.335	0.773	0.299	0.147	0.570	0.497
	Proc. Sim.	<u>0.767</u>	0.359	0.867	0.494	0.399	0.581	0.578
VV	PK	0.810	0.653	0.857	0.915	<u>0.869</u>	0.509	0.769
	CS	0.879	0.731	<u>0.859</u>	0.935	0.875	0.500	0.797
	Simple-CS	<u>0.820</u>	0.656	0.863	<u>0.926</u>	0.856	0.517	<u>0.773</u>
	Linear CKA	0.426	0.217	0.361	0.370	0.227	<u>0.566</u>	0.361
	Proc. Sim.	0.579	0.371	0.622	0.588	0.608	0.610	0.563
OO	PK	0.865	0.708	0.922	0.375	<u>0.846</u>	<u>0.655</u>	<u>0.729</u>
	CS	0.949	0.781	0.860	0.308	0.804	0.642	0.724
	Simple-CS	<u>0.948</u>	<u>0.773</u>	<u>0.886</u>	0.292	0.858	0.617	0.729
	Linear CKA	0.209	0.118	0.658	<u>0.750</u>	0.567	0.707	0.501
	Proc. Sim.	0.207	0.119	0.743	0.763	0.619	0.525	0.496
Mean over QQ, KK, VV, OO	PK	0.790	0.557	0.897	0.807	0.914	0.718	0.781
	CS	0.870	0.632	0.909	<u>0.799</u>	0.906	0.709	0.804
	Simple-CS	<u>0.824</u>	<u>0.618</u>	<u>0.906</u>	0.790	<u>0.910</u>	<u>0.717</u>	<u>0.794</u>
	Linear CKA	0.595	0.230	0.631	0.535	0.469	0.595	0.509
	Proc. Sim.	0.566	0.307	0.706	0.651	0.547	0.567	0.557

(b) ROC-AUC

Table 20: PR-AUC and ROC-AUC for each head class in Qwen2-0.5B-Instruct, computed separately for the weight pairing types QQ, KK, VV, and OO, together with the mean over the four weight pairing types for each head class. These values are used to compute the summary reported in Table 5. The maximum is shown in bold and the second largest is underlined. We consider the $\binom{24 \times 14}{2} = \binom{336}{2} = 56,280$ directed head pairs whose source is in an earlier layer than the target; for heads in the same layer, the head with the smaller index is treated as the source. Dup, Prev, Ind, NM, N-NM, and S-Inh denote Duplicate Token, Previous Token, Induction, Name Mover, Negative Name Mover, and S-Inhibition heads, respectively.

Weight preprocessing	Module	Original	Preprocessed	Shape
Layer normalization folding	$\text{LN}(\mathbf{x})$	$\gamma \odot \frac{\mathbf{x} - \text{Mean}(\mathbf{x})\mathbf{1}_d}{\text{Std}(\mathbf{x})} + \beta$	$\frac{\mathbf{x} - \text{Mean}(\mathbf{x})\mathbf{1}_d}{\text{Std}(\mathbf{x})}$	$\mathbb{R}^d \rightarrow \mathbb{R}^d$
	Weight after LN	$\mathbf{W}_{\text{in}}$	$\mathbf{W}_{\text{in}}\mathbf{D}_\gamma\mathbf{C}_d$	$\mathbb{R}^{d_{\text{mid}} \times d}$
	Bias after LN	$\mathbf{b}_{\text{in}}$	$\mathbf{W}_{\text{in}}\beta + \mathbf{b}_{\text{in}}$	$\mathbb{R}^{d_{\text{mid}}}$
Centering writing weights	Weight before LN	$\mathbf{W}_{\text{out}}$	$\mathbf{C}_d\mathbf{W}_{\text{out}}$	$\mathbb{R}^{d \times d_{\text{mid}}}$
	Bias before LN	$\mathbf{b}_{\text{out}}$	$\mathbf{C}_d\mathbf{b}_{\text{out}}$	$\mathbb{R}^d$
Centering the unembedding matrix	Unembedding matrix	$\mathbf{E}_{\text{out}}$	$\mathbf{E}_{\text{out}}\mathbf{C}_T$	$\mathbb{R}^{d \times T}$
Folding attention bias	Bias in value	$\mathbf{b}_V^h$	$\mathbf{0}_{d_{\text{head}}}$	$\mathbb{R}^{d_{\text{head}}}$
	Bias in output	$\mathbf{b}_O^h$	$\mathbf{W}_O^h\mathbf{b}_V^h + \mathbf{b}_O^h$	$\mathbb{R}^d$

Table 21: Summary of weight preprocessing (Elhage et al., 2021; Nanda and Bloom, 2022). Let d be the model dimension and d_{mid} the intermediate dimension. Here, $\gamma, \beta \in \mathbb{R}^d$ are the LN scaling weight and bias, and $\mathbf{D}_\gamma \in \mathbb{R}^{d \times d}$ is the diagonal matrix with diagonal entries γ . The matrices $\mathbf{C}_d \in \mathbb{R}^{d \times d}$ and $\mathbf{C}_T \in \mathbb{R}^{T \times T}$ are the centering matrices in (126) for dimensions d and T , where T is the vocabulary size. Note that only $\text{LN}(\mathbf{x})$ is a function, whereas the other entries are parameters. See Appendix O.1 for more details.

Identity Heads L5H2 and L6H3 appear among the top-ranked heads for OV inlet scores. L5H2 also appears among the top-ranked heads for OQ outlet scores, while L6H3 appears among the top-ranked heads for both OQ and OV outlet scores. Thus, L5H2 and L6H3 in BLOOM-560M are Identity Heads that function as hubs, similarly to L4H7 in GPT2-small. This result suggests that the hub role of Identity Heads observed in Section 8.2 is not specific to GPT2-small, but may be a more general property.

Interestingly, as with GPT2-small, other Identity Heads also appear among the top-ranked heads for outlet scores in Table 18b. A more detailed analysis of the functions of Identity Heads is left for future work.

O Comparing PK and CS on preprocessed weights in GPT2-small

Pretrained weights can be simplified without changing the output of a language model (Elhage et al., 2021; Nanda and Bloom, 2022). Following Gurnee et al. (2024), we call this procedure *weight preprocessing*, and we call the resulting weights *preprocessed weights*.

In TransformerLens (Nanda and Bloom, 2022), which is widely used in mechanistic interpretability research, this weight preprocessing is applied by default when loading pretrained model weights from HuggingFace (Wolf et al., 2020). Therefore, when analyzing weights with TransformerLens using PK or CS, one should note that the weights may differ from the original ones.

In this section, we first describe the weight pre-

processing procedure based on Nanda and Bloom (2022). We then compare, for GPT2-small, PK and CS computed on the original weights with those computed on the preprocessed weights.

O.1 Background: weight preprocessing

Based on Nanda and Bloom (2022), we describe four types of weight preprocessing⁵³. Table 21 summarizes these procedures and compares the original weights and the preprocessed weights.

O.1.1 Layer normalization folding

By decomposing layer normalization (LN) (Ba et al., 2016) into four steps, we can fold the LN scaling weight and bias into the weight and bias immediately after LN, thereby simplifying LN.

Layer normalization. Using the scaling weight $\gamma \in \mathbb{R}^d$ and bias $\beta \in \mathbb{R}^d$, LN for an input $\mathbf{x} = (x_1, \dots, x_d)^\top \in \mathbb{R}^d$ is

$$\text{LN}(\mathbf{x}) = \gamma \odot \frac{\mathbf{x} - \text{Mean}(\mathbf{x})\mathbf{1}_d}{\text{Std}(\mathbf{x})} + \beta, \quad (123)$$

where $\odot$ denotes the Hadamard product, $\mathbf{1}_d \in \mathbb{R}^d$ is the all-ones vector, and $\text{Mean}(\cdot)$ and $\text{Std}(\cdot)$ are defined as

$$\text{Mean}(\mathbf{x}) = \frac{1}{d}\mathbf{1}_d^\top \mathbf{x}, \quad (124)$$

$$\text{Std}(\mathbf{x}) = \sqrt{\frac{1}{d} \sum_{i=1}^d (x_i - \text{Mean}(\mathbf{x}))^2}. \quad (125)$$

⁵³https://github.com/TransformerLensOrg/TransformerLens/blob/main/further_comments.md

Centering matrix. To simplify algebraic manipulations of LN, we use the following centering matrix (Borg and Groenen, 2005):

$$C_d = I_d - \frac{1}{d} \mathbf{1}_d \mathbf{1}_d^\top \in \mathbb{R}^{d \times d}, \quad (126)$$

where $I_d \in \mathbb{R}^{d \times d}$ is the identity matrix. From the definition of C_d in (126) and $\text{Mean}(\cdot)$ in (124), we have

$$\begin{aligned} C_d \mathbf{x} &= \mathbf{x} - \frac{1}{d} \mathbf{1}_d \mathbf{1}_d^\top \mathbf{x} \\ &= \mathbf{x} - \text{Mean}(\mathbf{x}) \mathbf{1}_d. \end{aligned} \quad (127)$$

The matrix C_d is an orthogonal projection matrix since it satisfies

$$C_d C_d = C_d, \quad (128)$$

$$C_d^\top = C_d. \quad (129)$$

Decomposing LN into four steps. LN can be decomposed into the following four steps⁵⁴:

$$\begin{aligned} \mathbf{x}^{(1)} &= \mathbf{x} - \text{Mean}(\mathbf{x}) \mathbf{1}_d \\ &= C_d \mathbf{x} \quad (\because (127)), \end{aligned} \quad (130)$$

$$\begin{aligned} \mathbf{x}^{(2)} &= \frac{\mathbf{x}^{(1)}}{\sqrt{\frac{1}{d} \sum_{i=1}^d (x_i - \text{Mean}(\mathbf{x}))^2}} \\ &= \sqrt{d} \frac{\mathbf{x}^{(1)}}{\sqrt{\sum_{i=1}^d (x_i^{(1)})^2}} = \sqrt{d} \frac{\mathbf{x}^{(1)}}{\|\mathbf{x}^{(1)}\|}, \end{aligned} \quad (131)$$

$$\mathbf{x}^{(3)} = \gamma \odot \mathbf{x}^{(2)} = D_\gamma \mathbf{x}^{(2)}, \quad (132)$$

$$\mathbf{x}^{(4)} = \mathbf{x}^{(3)} + \beta. \quad (133)$$

Here, $x_i^{(1)}$ is the i -th component of $\mathbf{x}^{(1)}$ in (130), and $D_\gamma \in \mathbb{R}^{d \times d}$ is the diagonal matrix whose diagonal entries are γ . By definition, $\text{LN}(\mathbf{x}) = \mathbf{x}^{(4)}$.

Folding into the weight and bias after LN. Based on the four-step decomposition, consider the affine map immediately after LN that takes

⁵⁴Since $\mathbf{1}_d^\top C_d = \mathbf{1}_d^\top (I_d - \mathbf{1}_d \mathbf{1}_d^\top / d) = \mathbf{0}^\top$, we have $\mathbf{1}_d^\top C_d \mathbf{x} = 0$ for any $\mathbf{x} \in \mathbb{R}^d$. Thus, C_d is the orthogonal projection onto the orthogonal complement of the one-dimensional subspace spanned by $\mathbf{1}_d$. In particular, for (130), we have $\mathbf{x}^{(1)} = C_d \mathbf{x} = \mathbf{x} - \mathbf{x}^\top (\mathbf{1}_d / \sqrt{d}) \mathbf{1}_d / \sqrt{d}$, so Nanda and Bloom (2022) view centering as removing the component of $\mathbf{x}$ in the $\mathbf{1}_d$ direction using the unit vector $\mathbf{1}_d / \sqrt{d}$. For the resulting $\mathbf{x}^{(1)}$, they obtain $\mathbf{x}^{(2)}$ in (131) by projecting it onto the sphere of radius $\sqrt{d}$ while preserving its direction.

$\text{LN}(\mathbf{x}) = \mathbf{x}^{(4)}$ as input. Let $\mathbf{W}_{\text{in}} \in \mathbb{R}^{d_{\text{mid}} \times d}$ and $\mathbf{b}_{\text{in}} \in \mathbb{R}^{d_{\text{mid}}}$ be its weight and bias⁵⁵. Then

$$\begin{aligned} &\mathbf{W}_{\text{in}} \mathbf{x}^{(4)} + \mathbf{b}_{\text{in}} \\ &= \mathbf{W}_{\text{in}} D_\gamma \frac{\sqrt{d} C_d \mathbf{x}}{\|\mathbf{x}^{(1)}\|} + \mathbf{W}_{\text{in}} \beta + \mathbf{b}_{\text{in}} \\ &\quad (\because (133), (132), (131), (130)) \\ &= \mathbf{W}_{\text{in}} D_\gamma C_d C_d \frac{\sqrt{d} \mathbf{x}}{\|\mathbf{x}^{(1)}\|} + \mathbf{W}_{\text{in}} \beta + \mathbf{b}_{\text{in}} \\ &\quad (\because (128)) \\ &= \mathbf{W}_{\text{in}} D_\gamma C_d \mathbf{x}^{(2)} + \mathbf{W}_{\text{in}} \beta + \mathbf{b}_{\text{in}} \\ &\quad (\because (130), (131)). \end{aligned} \quad (134)$$

Define

$$\text{LN}^{(2)}(\mathbf{x}) = \mathbf{x}^{(2)} = \frac{\mathbf{x} - \text{Mean}(\mathbf{x}) \mathbf{1}_d}{\text{Std}(\mathbf{x})}, \quad (135)$$

$$\mathbf{W}'_{\text{in}} = \mathbf{W}_{\text{in}} D_\gamma C_d, \mathbf{b}'_{\text{in}} = \mathbf{W}_{\text{in}} \beta + \mathbf{b}_{\text{in}}. \quad (136)$$

Then, from $\text{LN}(\mathbf{x}) = \mathbf{x}^{(4)}$ and (134), we obtain

$$\mathbf{W}_{\text{in}} \text{LN}(\mathbf{x}) + \mathbf{b}_{\text{in}} = \mathbf{W}'_{\text{in}} \text{LN}^{(2)}(\mathbf{x}) + \mathbf{b}'_{\text{in}}. \quad (137)$$

From (137), replacing LN in (123) with $\text{LN}^{(2)}(\mathbf{x})$ in (135), and replacing $\mathbf{W}_{\text{in}}$ and $\mathbf{b}_{\text{in}}$ with $\mathbf{W}'_{\text{in}}$ and $\mathbf{b}'_{\text{in}}$, does not change the output of the affine map. In particular, $\text{LN}^{(2)}(\mathbf{x})$ in (135) performs only centering and standard-deviation normalization, and is simpler than the original LN in (123). For attention heads, this folding can be applied to the reading weights $\mathbf{W}_Q^h$, $\mathbf{W}_K^h$, and $\mathbf{W}_V^h$.

O.1.2 Centering weights that write to the residual stream

Let $\zeta \in \mathbb{R}^{d_{\text{mid}}}$ be an intermediate representation, and suppose the affine output $\mathbf{W}_{\text{out}} \zeta + \mathbf{b}_{\text{out}} \in \mathbb{R}^d$ is written to the residual stream, where $\mathbf{W}_{\text{out}} \in \mathbb{R}^{d \times d_{\text{mid}}}$ and $\mathbf{b}_{\text{out}} \in \mathbb{R}^d$. Consider applying LN to $\mathbf{W}_{\text{out}} \zeta + \mathbf{b}_{\text{out}}$. As seen in (130), LN performs centering in the first step. Therefore, centering gives

$$\begin{aligned} &C_d(\mathbf{W}_{\text{out}} \zeta + \mathbf{b}_{\text{out}}) \\ &= C_d C_d(\mathbf{W}_{\text{out}} \zeta + \mathbf{b}_{\text{out}}) \quad (\because (128)) \\ &= C_d(C_d \mathbf{W}_{\text{out}} \zeta + C_d \mathbf{b}_{\text{out}}). \end{aligned} \quad (138)$$

Thus, defining

$$\mathbf{W}'_{\text{out}} = C_d \mathbf{W}_{\text{out}}, \mathbf{b}'_{\text{out}} = C_d \mathbf{b}_{\text{out}}, \quad (139)$$

we can replace $\mathbf{W}_{\text{out}}$ and $\mathbf{b}_{\text{out}}$ with $\mathbf{W}'_{\text{out}}$ and $\mathbf{b}'_{\text{out}}$ without changing the result of centering. For attention heads, this centering can be applied to the writing weights $\mathbf{W}_O^h$.

⁵⁵For example, $d_{\text{mid}} = d_{\text{head}}$ for an attention layer and $d_{\text{mid}} = 4d$ for a MLP layer.

O.1.3 Other preprocessing steps

When pretrained model weights are loaded using the default settings in TransformerLens (Nanda and Bloom, 2022), the unembedding matrix is also centered⁵⁶, and the bias term in the output computation of attention is also folded⁵⁷. These steps are unrelated to PK and CS.

O.2 Wiring diagrams based on PK and CS between preprocessed weights

We compare the top 20 wiring diagrams from the original GPT2-small weights (PK: Fig. 6, CS: Fig. 22) with those from the preprocessed weights described in Appendix O.1, with LN folding applied as in (136) and writing weights centered as in (139).

Figure 37 shows the top 20 PK edges computed from preprocessed weights for each of the OQ, OK, and OV pairings. This wiring diagram is almost unchanged from the original one in Fig. 6. In fact, among the 60 edges, 57 are shared (19 for OQ, 20 for OK, and 18 for OV)⁵⁸. This result indicates that

⁵⁶Let $\mathbf{E}_{\text{out}} \in \mathbb{R}^{d \times T}$ be the unembedding matrix and let $\tilde{\mathbf{x}} \in \mathbb{R}^d$ be the vector after the final LN in (70), where $T = |\mathcal{T}|$. Logits are computed as $\mathbf{E}_{\text{out}}^\top \tilde{\mathbf{x}} \in \mathbb{R}^T$, and token probabilities are computed by applying softmax. By definition of softmax, adding the same constant to all components of $\mathbf{E}_{\text{out}}^\top \tilde{\mathbf{x}}$ does not change the output. Let $\mathbf{C}_T \in \mathbb{R}^{T \times T}$ be the T -dimensional centering matrix, analogous to (126). Then centering logits using $\mathbf{C}_T$ does not affect the softmax output, so

$$\begin{aligned} \text{softmax}(\mathbf{E}_{\text{out}}^\top \tilde{\mathbf{x}}) &= \text{softmax}(\mathbf{C}_T \mathbf{E}_{\text{out}}^\top \tilde{\mathbf{x}}) \\ &= \text{softmax}((\mathbf{E}_{\text{out}} \mathbf{C}_T^\top)^\top \tilde{\mathbf{x}}) \\ &= \text{softmax}((\mathbf{E}_{\text{out}} \mathbf{C}_T)^\top \tilde{\mathbf{x}}) \in [0, 1]^T \quad (\cdot \text{ (129)}). \end{aligned}$$

Thus, defining $\mathbf{E}'_{\text{out}} = \mathbf{E}_{\text{out}} \mathbf{C}_T$, we can replace $\mathbf{E}_{\text{out}}$ with $\mathbf{E}'_{\text{out}}$ without changing the softmax output.

⁵⁷In $o^h(\mathbf{X})$ in (16), we omitted bias terms. Including the bias terms $\mathbf{b}_V^h \in \mathbb{R}^{d_{\text{head}}}$ and $\mathbf{b}_O^h \in \mathbb{R}^d$ in addition to $\mathbf{W}_V^h$ and $\mathbf{W}_O^h$ yields

$$\begin{aligned} o^h(\mathbf{X}) &= \mathbf{W}_O^h (\mathbf{W}_V^h \mathbf{X} \mathbf{A}^{h^\top} + \mathbf{b}_V^h) + \mathbf{b}_O^h \\ &= \mathbf{W}_O^h \mathbf{W}_V^h \mathbf{X} \mathbf{A}^{h^\top} + \mathbf{W}_O^h \mathbf{b}_V^h + \mathbf{b}_O^h \\ &= \mathbf{W}_O^h (\mathbf{W}_V^h \mathbf{X} \mathbf{A}^{h^\top} + \mathbf{0}_{d_{\text{head}}}) + \mathbf{W}_O^h \mathbf{b}_V^h + \mathbf{b}_O^h. \end{aligned}$$

Thus, defining $\mathbf{b}_V^{h'} = \mathbf{0}_{d_{\text{head}}}$ and $\mathbf{b}_O^{h'} = \mathbf{W}_O^h \mathbf{b}_V^h + \mathbf{b}_O^h$, we can replace $\mathbf{b}_V^h$ and $\mathbf{b}_O^h$ with $\mathbf{b}_V^{h'}$ and $\mathbf{b}_O^{h'}$ without changing the attention output.

⁵⁸Edges that appear only in the original wiring diagram are the OQ edge between L8H10 and L10H6, the OV edge between L5H5 and L6H6, and the OV edge between L7H7 and L8H5. Edges that appear only in the preprocessed wiring diagram are the OQ edge between L6H6 and L7H7, the OV edge between L3H10 and L4H11, and the OV edge between L5H10 and L7H5.

PK is robust to weight changes induced by weight preprocessing.

Figure 38 shows the top 20 CS edges computed from preprocessed weights for each of the OQ, OK, and OV pairings. This wiring diagram differs substantially from the original one in Fig. 22. In fact, among the 60 edges, only 16 are shared (1 for OQ, 4 for OK, and 11 for OV). This result indicates that CS is sensitive to weight changes induced by weight preprocessing. In particular, the change is largest for OQ and OK, and many head pairs now have edges in both pairings, which suggests that the preprocessing reduces the difference between the query and key weights as measured by CS. Moreover, edges between distant layers appear only for OV in the original wiring diagram, whereas they also appear for OQ and OK after weight preprocessing.

O.3 Quantitative comparison of scores computed from original and preprocessed weights for PK and CS

In this subsection, based on the results in Appendix O.2, where weight preprocessing changes the wiring diagrams of GPT2-small, we quantitatively compare these differences. Specifically, for PK and CS, we compare the mean squared error (MSE) and Pearson’s r between the values computed from the original weights and those computed from the preprocessed weights for each weight pairing type. We consider all $\binom{144}{2} = 10,296$ pairs among the 144 heads in GPT2-small, which has 12 layers with 12 heads per layer.

O.3.1 MSE comparison

In this section, we compare PK and CS in terms of the MSE between scores computed from the original weights and those computed from the preprocessed weights. To make PK comparable to CS, we normalize PK by d_{head} so that it lies on the same $[0, 1]$ scale as CS (see Appendix C.6 for details), and we refer to this quantity as Normalized PK.

Figure 39 reports the MSE for Normalized PK and CS across weight pairing types. Across the 16 weight pairings, Normalized PK yields MSE values that are smaller than or comparable to those of CS. This result indicates that PK is robust to changes induced by weight preprocessing, whereas CS is more sensitive to these changes, consistent with the findings in Appendix O.2.

We also find a common trend in both Normalized PK and CS. Pairings that involve only query

GPT2-small: Top 20 Weight-Preprocessing Projection Kernel for OQ, OK, OV

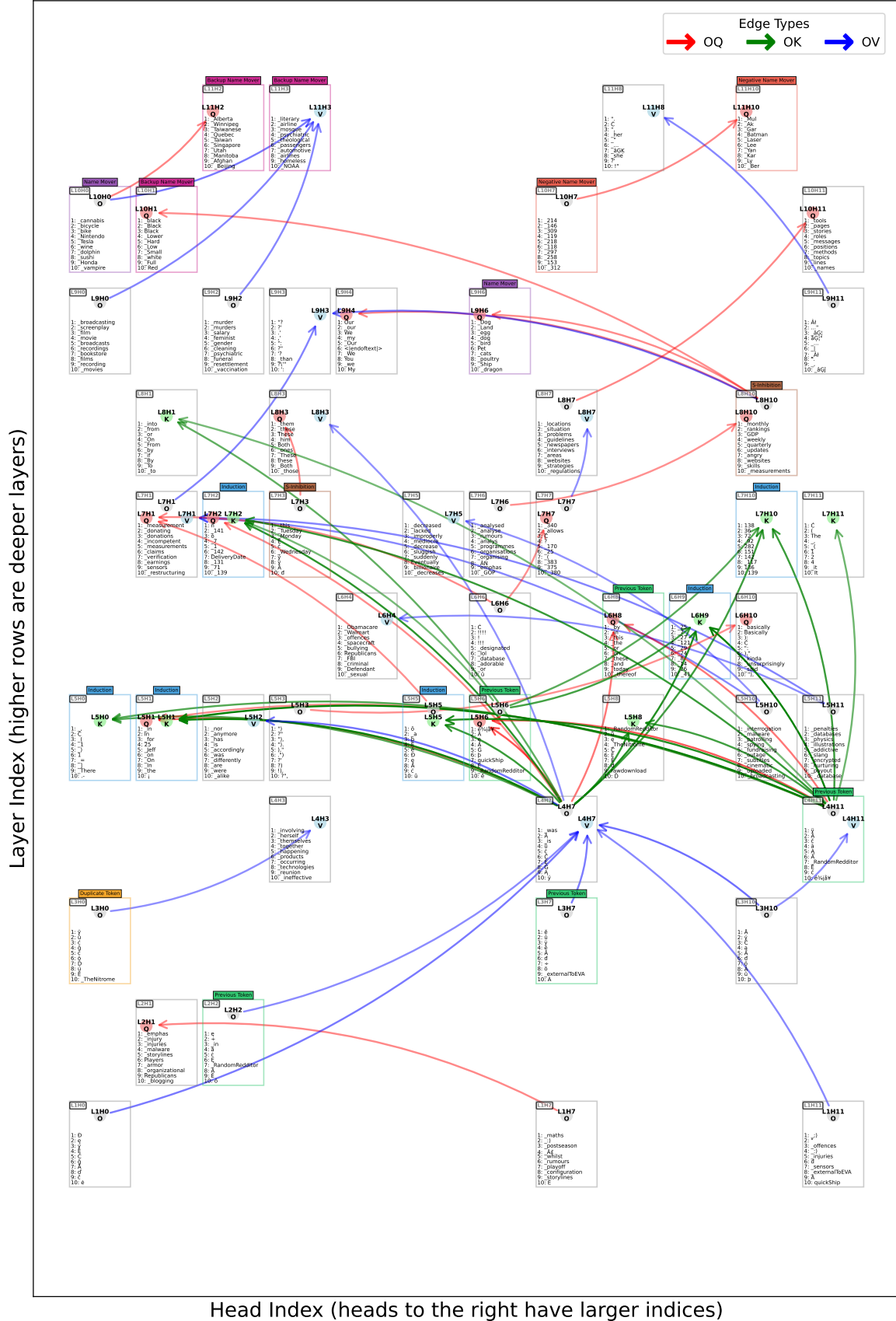


Figure 37: For GPT2-small, to compare with the wiring diagram constructed from PK on the original weights in Fig. 6, we visualize the top 20 scores of PK_{OQ}, PK_{OK}, and PK_{OV} computed on the preprocessed weights.

GPT2-small: Top 20 Weight-Preprocessing Composition Score for OQ, OK, OV

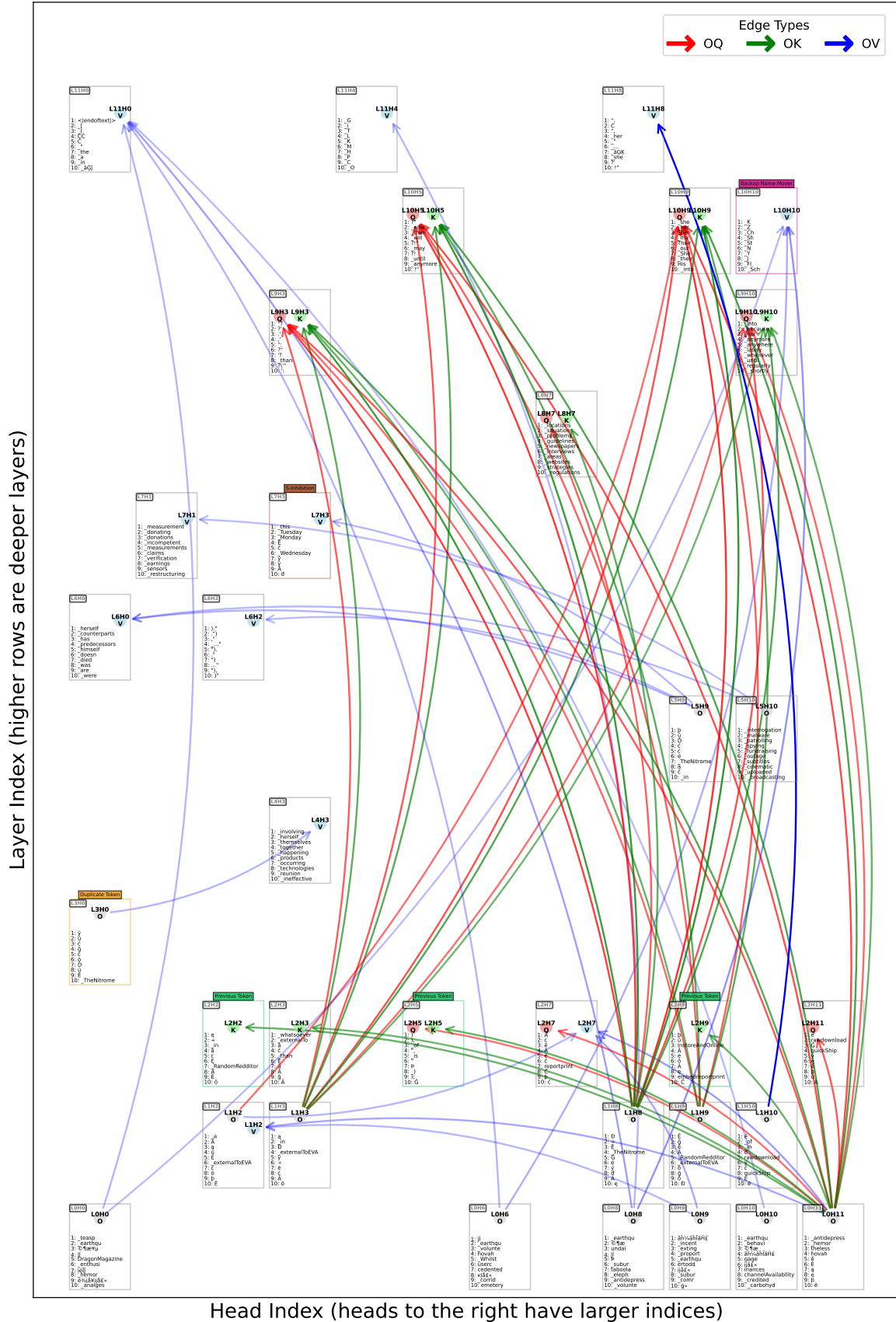


Figure 38: For GPT2-small, to compare with the wiring diagram constructed from CS on the original weights in Fig. 22, we visualize the top 20 scores of CS_{OQ} , CS_{OK} , and CS_{OV} computed on the preprocessed weights.

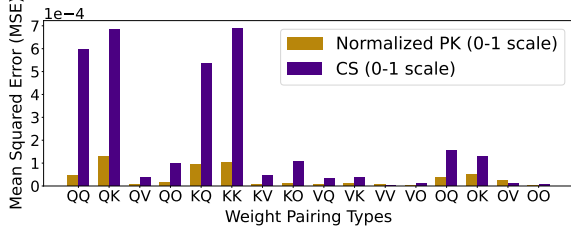


Figure 39: For GPT2-small, we report the MSE between scores from the original and preprocessed weights for PK and CS, by weight pairing type. Normalized PK denotes PK normalized by d_{head} , which maps it to the same $[0, 1]$ scale as CS.

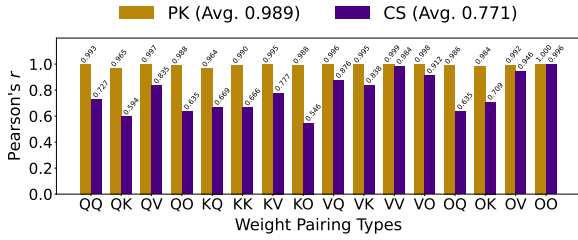


Figure 40: For GPT2-small, we report Pearson’s r between scores from the original and preprocessed weights for PK and CS, by weight pairing type.

and key weights (QQ, QK, KQ, and KK) exhibit larger MSE than pairings that involve only value or output weights (VV, VO, OV, and OO). Interestingly, although the query, key, and value weights all undergo the same layer normalization folding, the MSE for QQ or KK is larger than that for VV, and the difference is especially large for CS. A more detailed analysis of these results is left for future work.

O.3.2 Correlation comparison

In this section, we compare PK and CS in terms of Pearson’s r between scores computed from the original weights and those computed from the preprocessed weights.

Figure 40 reports Pearson’s r for PK and CS across the 16 weight pairing types. For PK, the correlation coefficient is close to 1 for all weight pairing types, with a mean of 0.989. In contrast, for CS, the correlation is lower for weight pairing types involving query or key weights. For example, KO exhibits the lowest correlation coefficient, with a value of 0.546. These results are also consistent with the findings in Appendix O.2 and Appendix O.3.1.

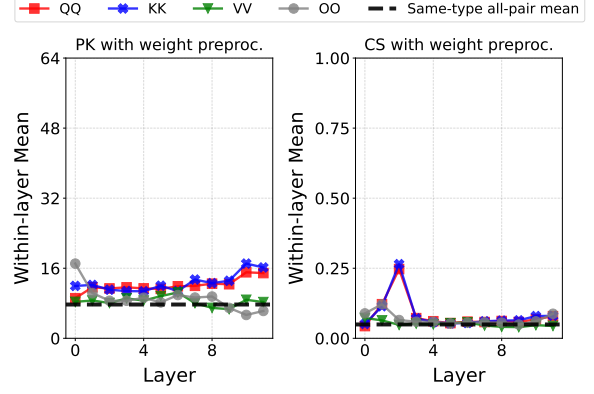


Figure 41: Layerwise PK and CS with weight preprocessing of GPT2-small, where each panel shows the mean over all 66 head pairs within each layer (all pairwise combinations of the 12 heads) for the weight pairing types QQ, KK, VV, and OO. The dashed line marks the mean over all $\binom{144}{2} = 10,296$ head pairs for each of the four weight pairing types, yielding a total of $4 \times \binom{144}{2} = 41,184$ comparisons: PK = 7.71 and CS = 0.05.

O.4 Score heatmaps for PK and CS after weight preprocessing

As with the score heatmaps in Appendix J.2, we examine heatmaps of scores between heads for PK and CS across the 16 weight pairing types after weight preprocessing. Figures 42 and 43 show the score heatmaps computed from the preprocessed weights for PK and CS, respectively.

For PK, the heatmap patterns are nearly identical to those computed from the original weights in Fig. 27 in Appendix J.2. This observation is consistent with the correlations between the corresponding PK scores being close to 1 (Fig. 40 in Appendix O.3.2).

In contrast, for CS, the heatmaps computed from the preprocessed weights differ from those computed from the original weights in Fig. 28 in Appendix J.2. In particular, for QQ, QK, KQ, and KK, the CS scores in layers 1 and 2 become larger after preprocessing, resulting in different heatmap patterns. This observation is also consistent with the relatively large MSE between the corresponding CS scores (Fig. 39 in Appendix O.3.1).

O.5 Layerwise averages of PK and CS for same-type weight pairings after weight preprocessing

Motivated by the results in Appendix O.4, we analyze the layerwise behavior of PK and CS after weight preprocessing, following Appendix J.3.

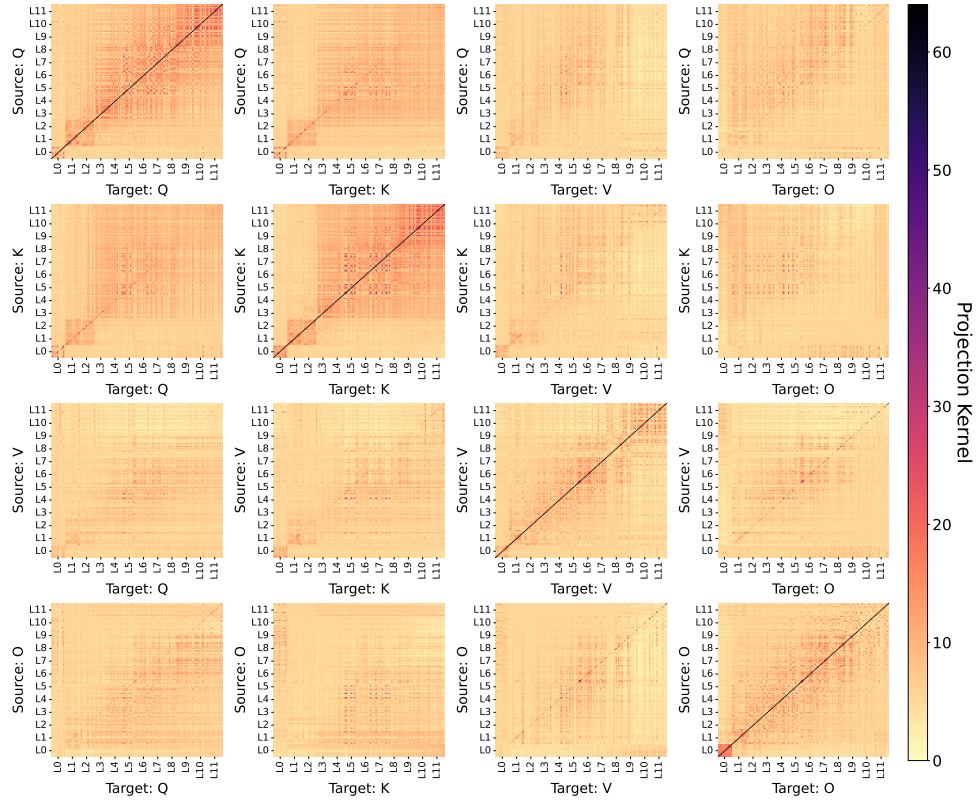


Figure 42: Projection Kernel (PK) with weight preprocessing between all head pairs of GPT2-small for each weight pairing type (source, target) of Q, K, V, and O, where the tick label $L\ell$ denotes layer ℓ .

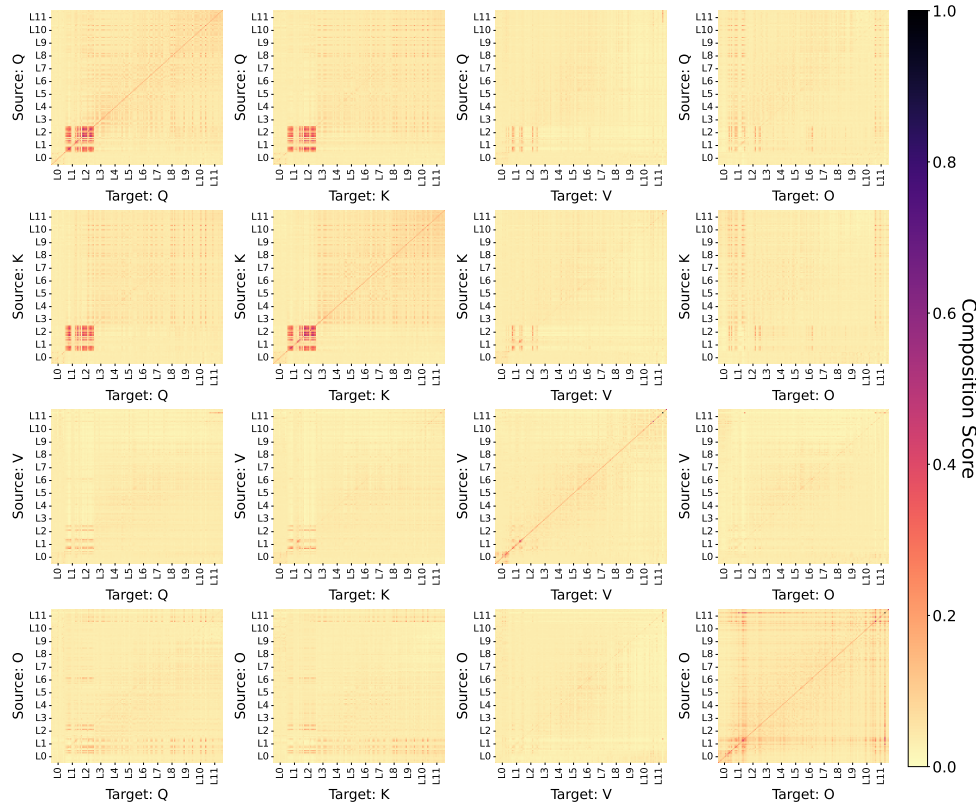


Figure 43: Composition Score (CS) with weight preprocessing between all head pairs of GPT2-small for each weight pairing type (source, target) of Q, K, V, and O, where the tick label $L\ell$ denotes layer ℓ .

Specifically, for the same-type weight pairings (QQ, KK, VV, and OO), we compute the layer-wise mean scores and report them together with the overall mean across the considered head pairs as a baseline in Fig. 41.

Compared with the results for the original weights in Fig. 32, PK exhibits nearly the same trends. In contrast, for CS, the QQ and KK scores become larger in layers 1 and 2, reflecting the heatmap patterns observed in Fig. 43 in Appendix O.4.

Although these results are limited to GPT2-small, they show that CS can change substantially with weight preprocessing for specific weight types, such as query and key weights, and in specific layers, even though the preprocessing does not change the model output. In contrast, PK remains nearly unchanged after weight preprocessing, suggesting that it may capture intrinsic properties of the heads and produce results nearly identical to those obtained from the original weights.

P Revisiting whether the attention-head weights of GPT2-small have full column rank

In Section 3.4, we assumed that the column space of each attention weight $\mathbf{W}^r \in \mathcal{W} \subset \mathbb{R}^{d \times d_{\text{head}}}$ in (12) has dimension d_{head} , implying that $\mathbf{W}^r$ has full column rank. In this section, we revisit this assumption.

First, Figs. 45, 46, 47, and 48 show the singular values of $\mathbf{W}_Q^h$, $\mathbf{W}_K^h$, $\mathbf{W}_V^h$, and $\mathbf{W}_O^h$, respectively, for each head in each layer of GPT2-small. For almost all heads, the singular values are visibly greater than zero, although a few are close to zero. Nevertheless, using `np.linalg.matrix_rank`⁵⁹, we confirmed that all of these weight matrices have rank $d_{\text{head}} = 64$.

Next, to account for differences in the magnitudes of the singular values, we measure the effective rank (Roy and Vetterli, 2007) as a continuous-valued notion of rank, following Wang et al. (2026)⁶⁰. For $\mathbf{W}^r \in \mathcal{W} \subset \mathbb{R}^{d \times d_{\text{head}}}$, let its singular values be $\sigma_1^r \geq \dots \geq \sigma_{d_{\text{head}}}^r > 0$. The effective

⁵⁹Note that `np.linalg.matrix_rank` determines the rank using a numerical tolerance on the singular values (see https://numpy.org/doc/stable/reference/generated/numpy.linalg.matrix_rank.html for more details).

⁶⁰Note that their study focuses on the dimensionality of activations output by Transformer blocks rather than on weights.

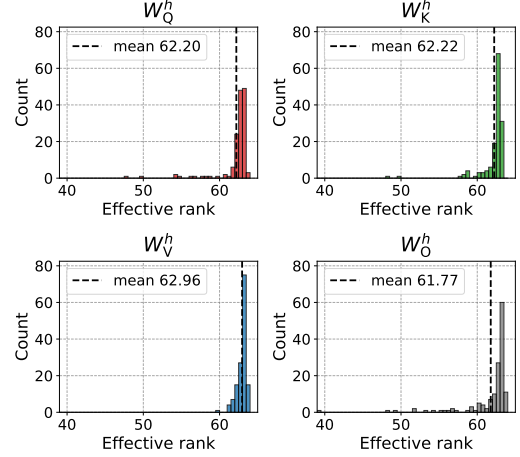


Figure 44: Effective ranks of $\mathbf{W}_Q^h$, $\mathbf{W}_K^h$, $\mathbf{W}_V^h$, and $\mathbf{W}_O^h$ across all 144 heads of GPT2-small. The maximum possible effective rank is $d_{\text{head}} = 64$. The black dashed line marks the mean of each panel.

rank is defined as follows⁶¹

$$\begin{aligned} \text{erank}(\mathbf{W}^r) &= \exp \left(- \sum_{i=1}^{d_{\text{head}}} \frac{\sigma_i^r}{\sum_{i'=1}^{d_{\text{head}}} \sigma_{i'}^r} \log \left(\frac{\sigma_i^r}{\sum_{i'=1}^{d_{\text{head}}} \sigma_{i'}^r} \right) \right). \end{aligned} \quad (140)$$

⁶¹As examples, we compute the effective rank when the singular values are uniform and when the largest singular value is sufficiently larger than the others. First, consider the case $\sigma_1^r = \dots = \sigma_{d_{\text{head}}}^r > 0$. Then, since $\sigma_i^r / \sum_{i'=1}^{d_{\text{head}}} \sigma_{i'}^r = 1/d_{\text{head}}$,

$$\begin{aligned} \text{erank}(\mathbf{W}^r) &= \exp \left(- \sum_{i=1}^{d_{\text{head}}} \frac{1}{d_{\text{head}}} \log \frac{1}{d_{\text{head}}} \right) \\ &= \exp \left(- \log \frac{1}{d_{\text{head}}} \right) = d_{\text{head}}. \end{aligned}$$

In contrast, consider the limit in which the largest singular value dominates the others, that is, $\sigma_i^r / \sigma_1^r \rightarrow 0$ for $i \neq 1$. In this limit, $\sigma_1^r / \sum_{i'=1}^{d_{\text{head}}} \sigma_{i'}^r \rightarrow 1$ and $\sigma_i^r / \sum_{i'=1}^{d_{\text{head}}} \sigma_{i'}^r \rightarrow 0$ for $i \neq 1$. Therefore,

$$\begin{aligned} &\frac{\sigma_i^r}{\sum_{i'=1}^{d_{\text{head}}} \sigma_{i'}^r} \log \left(\frac{\sigma_i^r}{\sum_{i'=1}^{d_{\text{head}}} \sigma_{i'}^r} \right) \\ &\rightarrow \begin{cases} 1 \log 1 = 0, & i = 1, \\ \lim_{x \rightarrow 0^+} x \log x = 0, & i \neq 1. \end{cases} \end{aligned}$$

Thus,

$$\text{erank}(\mathbf{W}^r) \rightarrow \exp \left(- \sum_{i=1}^{d_{\text{head}}} 0 \right) = 1.$$

Hence, the effective rank generally satisfies

$$1 \leq \text{erank}(\mathbf{W}^r) \leq d_{\text{head}}.$$

It equals d_{head} when the singular values are uniform, whereas it approaches 1 in the limit where a single singular value dominates. Thus, the effective rank continuously interpolates between these cases and can be interpreted as a measure of the effective dimensionality of the singular-value distribution.

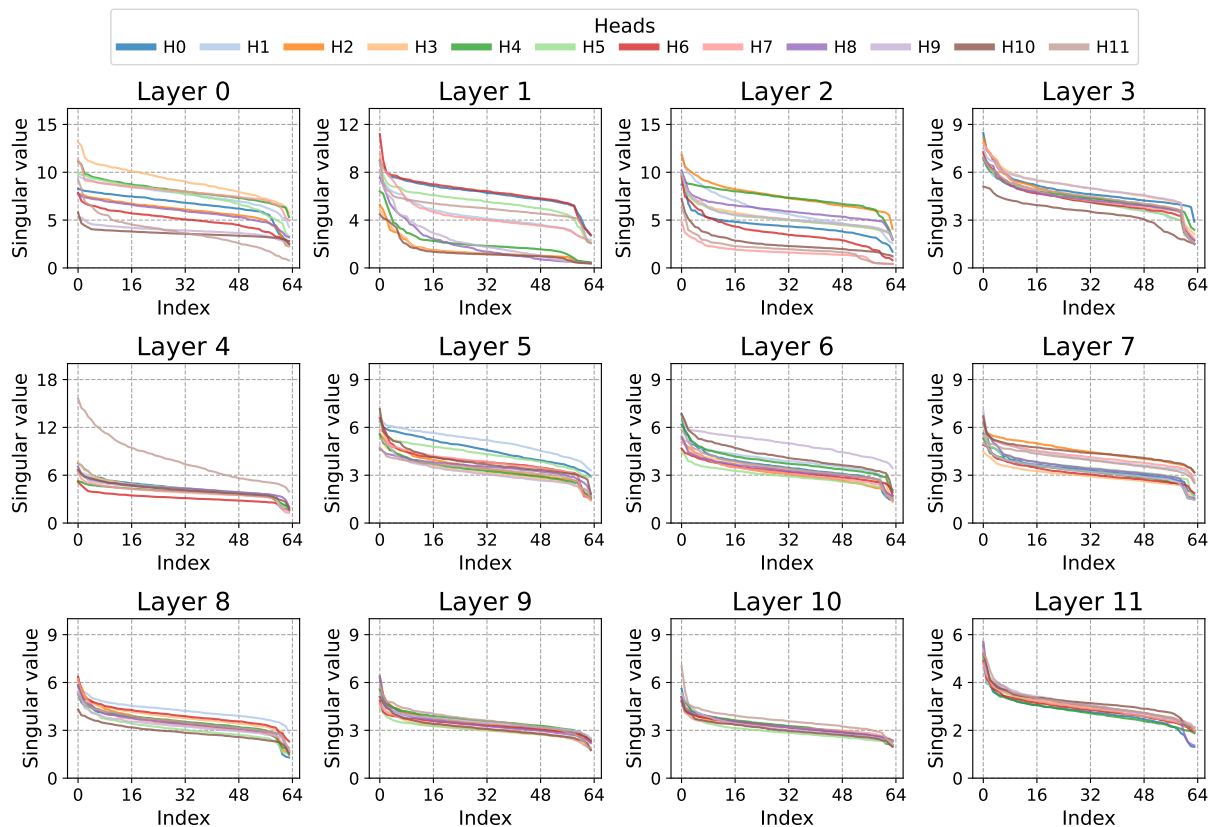


Figure 45: Singular values of GPT2-small attention-head weights $\mathbf{W}_Q^h$.

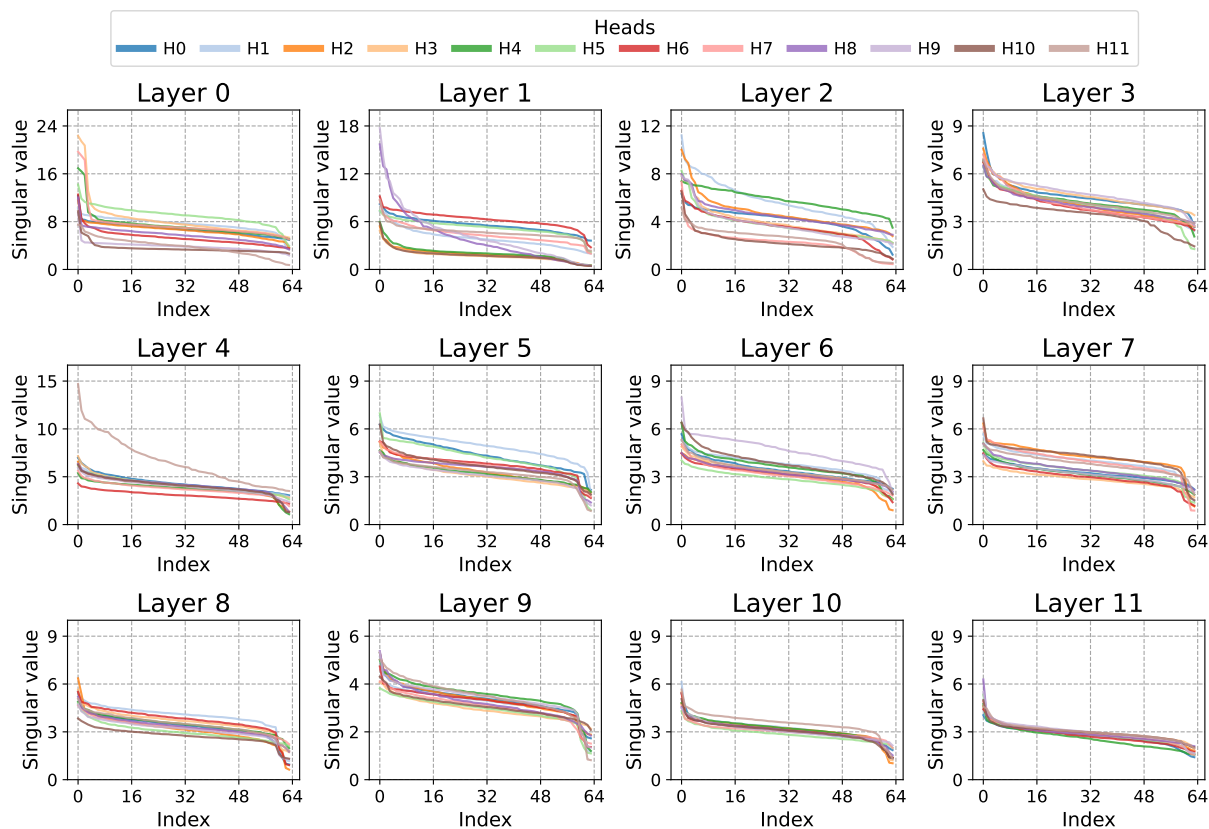


Figure 46: Singular values of GPT2-small attention-head weights $\mathbf{W}_K^h$.

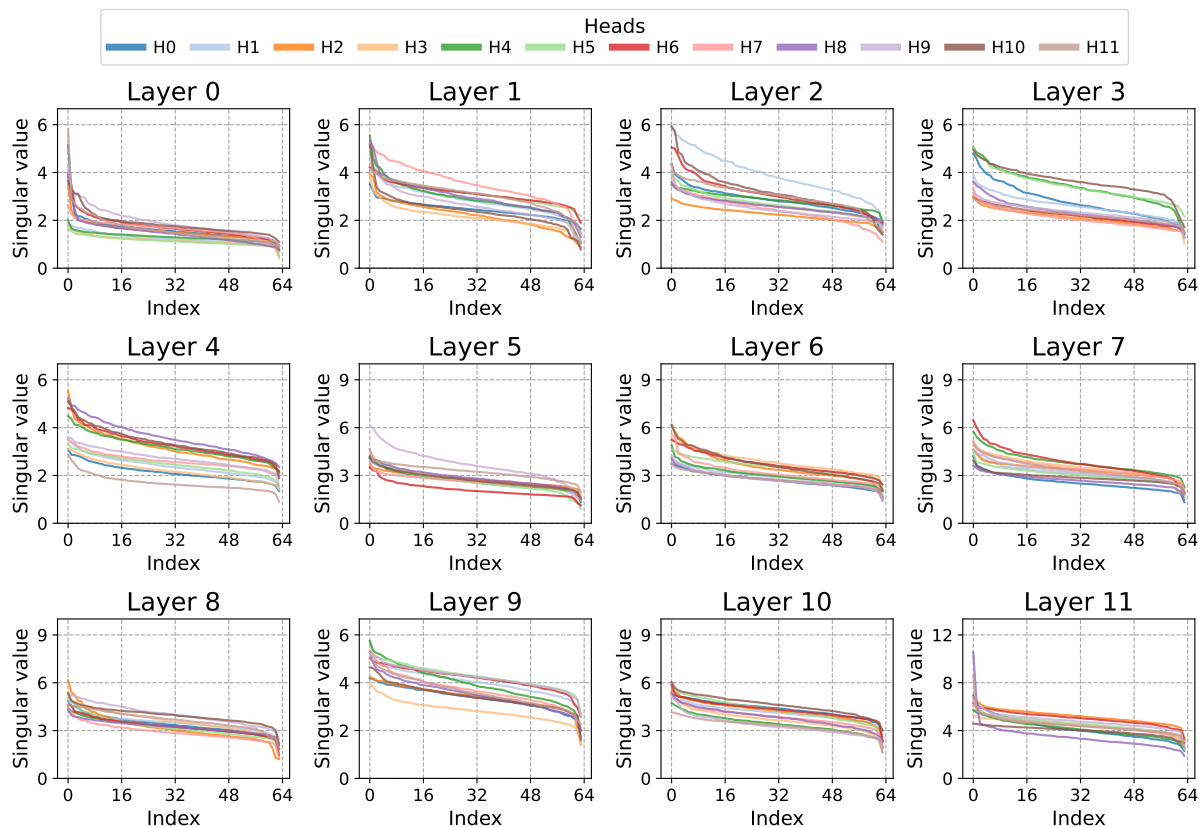


Figure 47: Singular values of GPT2-small attention-head weights W_V^h .

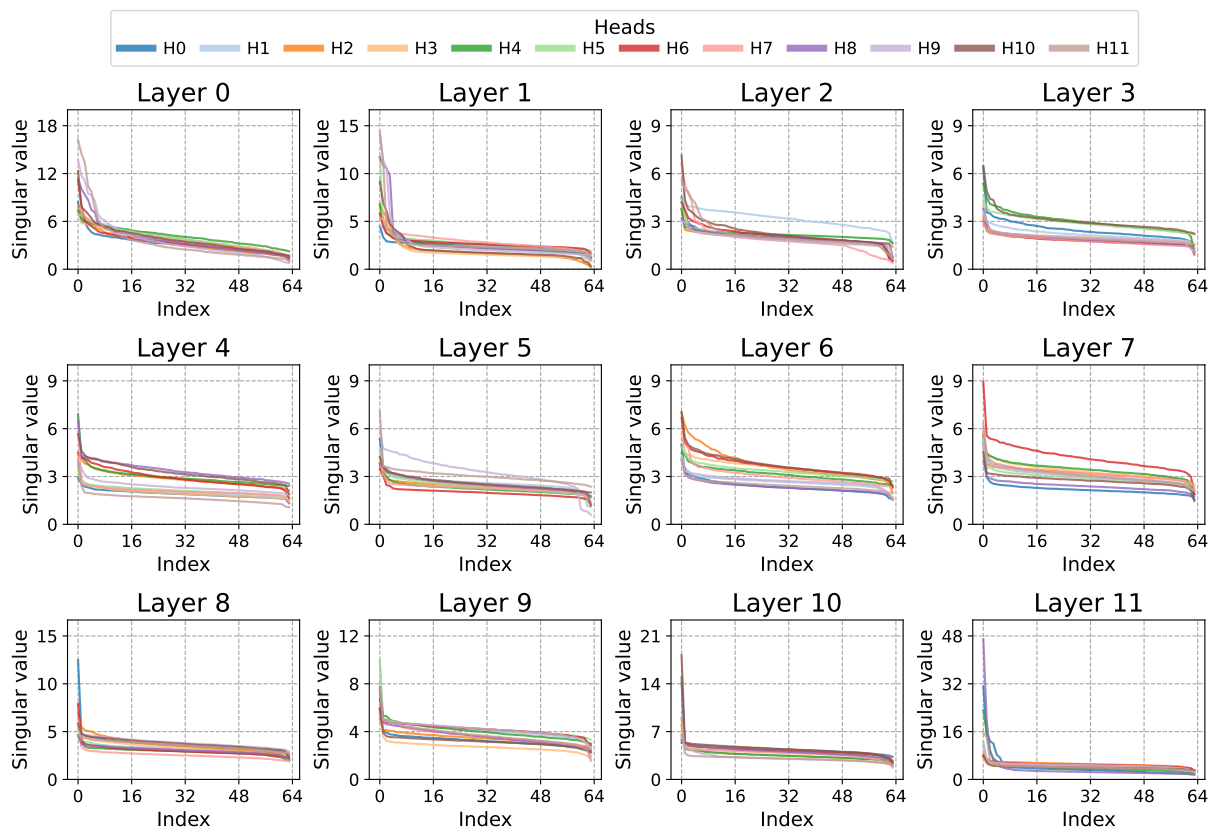


Figure 48: Singular values of GPT2-small attention-head weights W_O^h .

Figure 44 shows the results. For each weight type, some heads have effective ranks noticeably below d_{head} , particularly for $\mathbf{W}_O^h$, whereas the mean effective rank remains relatively close to d_{head} .

Thus, although the assumption in Section 3.4 may not hold exactly when interpreted in terms of effective dimensionality, these results provide some empirical support for its validity.