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

Anonymous ACL submission

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 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 A 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 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 of the subspaces shared between heads, thereby hindering the comparison and interpretation of similarities between

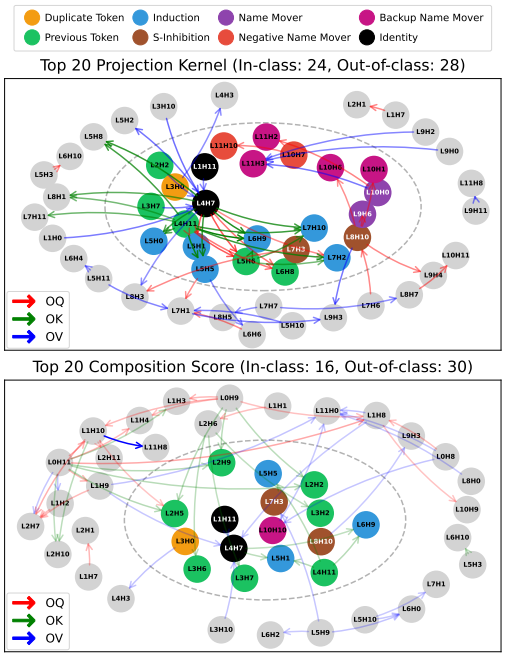


Figure 1: For the attention-head weights of GPT2-small, we compute similarities between heads using (top) the Projection Kernel (PK) and (bottom) the Composition Score (CS), and visualize the top 20 scores as edges for each weight pairing type (OQ, OK, and OV). Edge opacity corresponds to the score normalized by the maximum score. The seven head classes excluding 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.

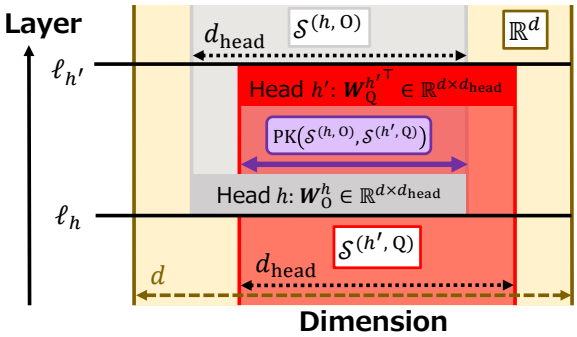


Figure 2: Illustration of measuring similarity using the Projection Kernel (PK) in (9) between the d_{head}-dimensional subspace $\mathcal{S}^{(h,\mathrm{O})} \subset \mathbb{R}^d$ spanned by the output weights $\boldsymbol{W}_{\mathrm{O}}^h \in \mathbb{R}^{d\times d_{\text{head}}}$ of head h and the d_{head}-dimensional subspace $\mathcal{S}^{(h',\mathrm{Q})} \subset \mathbb{R}^d$ spanned by the query weights $\boldsymbol{W}_{\mathrm{Q}}^{h'\top} \in \mathbb{R}^{d\times d_{\text{head}}}$ of head h'. The layer indices of heads h and h' are denoted by ℓ_h and ℓ'_h with $\ell_h < \ell'_h$. A larger $\mathrm{PK}(\mathcal{S}^{(h,\mathrm{O})}, \mathcal{S}^{(h',\mathrm{Q})})$ 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'.

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 spanned by 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 (Ham 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 simple attention-only model. Section 4 introduces a method for interpreting heads based on projection onto head-specific subspaces. Section 6 compares PK and CS, and Section 7 evaluates similarity metrics using functional groups of heads identified in the IOI task (hereafter, *head classes*). Section 8 discusses how the hub head L4H7 in Fig. 1 is an Identity Head and a method for comparing PK distributions, and finally we conclude.

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 its limitations. See Appendix B for details.

2.1 QK and OV matrix

Let d be the model dimension and d_{head} the head dimension. For head h, let $\boldsymbol{W}_{\mathrm{Q}}^h, \boldsymbol{W}_{\mathrm{K}}^h, \boldsymbol{W}_{\mathrm{V}}^h \in \mathbb{R}^{d_{\text{head}}\times d}$ be the query, key, and value weight matrices, and let $\boldsymbol{W}_{\mathrm{O}}^h \in \mathbb{R}^{d\times d_{\text{head}}}$ be the output weight matrix.

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

$$\boldsymbol{W}_{\mathrm{QK}}^h = \boldsymbol{W}_{\mathrm{Q}}^{h\top}\boldsymbol{W}_{\mathrm{K}}^h \in \mathbb{R}^{d\times d}, \tag{1}$$

$$\boldsymbol{W}_{\mathrm{OV}}^h = \boldsymbol{W}_{\mathrm{O}}^h\boldsymbol{W}_{\mathrm{V}}^h \in \mathbb{R}^{d\times d}. \tag{2}$$

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 later-layer head h'. We refer to this metric as the Composition Score (CS)[1]. For each pairing, we define $\mathrm{CS}_{\mathrm{OQ}}$, $\mathrm{CS}_{\mathrm{OK}}$, and $\mathrm{CS}_{\mathrm{OV}}$ as follows[2]:

$$\mathrm{CS}_{\mathrm{OQ}}(h,h') = \mathrm{CS}\left(\boldsymbol{W}_{\mathrm{QK}}^{h'\top}, \boldsymbol{W}_{\mathrm{OV}}^{h}\right), \tag{3}$$

$$\mathrm{CS}_{\mathrm{OK}}(h,h') = \mathrm{CS}\left(\boldsymbol{W}_{\mathrm{QK}}^{h'}, \boldsymbol{W}_{\mathrm{OV}}^{h}\right), \tag{4}$$

$$\mathrm{CS}_{\mathrm{OV}}(h,h') = \mathrm{CS}\left(\boldsymbol{W}_{\mathrm{OV}}^{h'}, \boldsymbol{W}_{\mathrm{OV}}^{h}\right). \tag{5}$$

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

$$\mathrm{CS}(\boldsymbol{W}^{h'}, \boldsymbol{W}^h) = \frac{\|\boldsymbol{W}^{h'}\boldsymbol{W}^h\|_{\mathrm{F}}}{\|\boldsymbol{W}^{h'}\|_{\mathrm{F}}\|\boldsymbol{W}^h\|_{\mathrm{F}}} \tag{6}$$

where $\|\boldsymbol{A}\|_{\mathrm{F}} = \sqrt{\mathrm{tr}(\boldsymbol{A}^\top\boldsymbol{A})}$ denotes the Frobenius norm. Thus, CS in (6) normalizes the Frobenius norm of the product $\boldsymbol{W}^{h'}\boldsymbol{W}^h$ by the Frobenius norms of $\boldsymbol{W}^{h'}$ and $\boldsymbol{W}^h$. Moreover, for example in (3), replacing $\boldsymbol{W}_{\mathrm{OV}}^h$ with $\boldsymbol{W}_{\mathrm{QK}}^h$ yields analogous definitions such as $\mathrm{CS}_{\mathrm{QQ}}$. CS is a head-to-head similarity metric, and tends to take larger values for pairs of heads that are more strongly related.

[1]The term *Composition Score* is not used in Elhage et al. (2021), and follows Merullo et al. (2024).

[2]Elhage et al. (2021) instead describe them as the scores for Q-, K-, and V-composition.

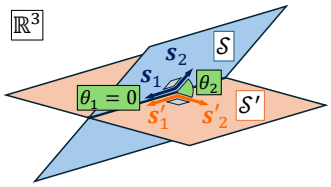


Figure 3: Principal angles $\theta_1 = 0, \theta_2$ and principal vectors $\boldsymbol{s}_1, \boldsymbol{s}_2 \in \mathcal{S}$ and $\boldsymbol{s}'_1, \boldsymbol{s}'_2 \in \mathcal{S}'$ for 2D subspaces $\mathcal{S}, \mathcal{S}' \subset \mathbb{R}^3$. (Inspired by Mandolesi (2021, Fig. 1))

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, the Frobenius norm measures the average output magnitude of a linear map under an isotropic input distribution [3]. Consequently, CS implicitly assumes an isotropic input distribution and retains direction-dependent scaling, thereby relatively emphasizing directions associated with large singular values, and thus cannot capture biases in the actual input distribution. Second, although any invertible $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ yields the factorization $\boldsymbol{W}^{h'}\boldsymbol{W}^{h} = \boldsymbol{W}^{h'}\boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{W}^{h}$, in general $\|\boldsymbol{W}^{h'}\|_{\mathrm{F}}\|\boldsymbol{W}^{h}\|_{\mathrm{F}} \neq \|\boldsymbol{W}^{h'}\boldsymbol{A}\|_{\mathrm{F}}\|\boldsymbol{A}^{-1}\boldsymbol{W}^{h}\|_{\mathrm{F}}$. Therefore, CS is not invariant to this factorization, and typically $\mathrm{CS}(\boldsymbol{W}^{h'}, \boldsymbol{W}^{h}) \neq \mathrm{CS}(\boldsymbol{W}^{h'}\boldsymbol{A}, \boldsymbol{A}^{-1}\boldsymbol{W}^{h})$. Taken together, these issues[4] suggest that CS can be insensitive to meaningful differences between heads.

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

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

and call θ_i the i-th principal angle (Björck and Golub, 1973). The vectors $\boldsymbol{s}_i \in \mathcal{S}$ and $\boldsymbol{s}'_i \in \mathcal{S}'$ 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 $\boldsymbol{U}, \boldsymbol{U}' \in \mathbb{R}^{d\times m}$ be orthonormal basis matrices of $\mathcal{S}$ and $\mathcal{S}'$, and let $\sigma_1 \geq \ldots \geq \sigma_m$ be the singular values of $\boldsymbol{U}^\top\boldsymbol{U}' \in \mathbb{R}^{m\times m}$. Then the i-th singular value σ_i corresponds to $\cos\theta_i$ (Dahlquist and Björck, 2008) (see Appendix C).

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) (Ham and Lee, 2008)[5]

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

which satisfies[6] $0 \leq \mathrm{PK}(\mathcal{S}, \mathcal{S}') \leq m$. In particular, $\mathrm{PK}(\mathcal{S}, \mathcal{S}') = m$ when $\mathcal{S} = \mathcal{S}'$. By algebraic manipulation (see Appendix C), we obtain

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

where $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$ and $\boldsymbol{P}' = \boldsymbol{U}'\boldsymbol{U}'^\top$ are the orthogonal projection matrices onto $\mathcal{S}$ and $\mathcal{S}'$. Equation (9) shows that PK can be computed directly from the basis matrices. From the definitions of PK and CS (see Appendix C), we have

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

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 $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d \in \mathbb{R}^d$ be an orthonormal basis of $\mathbb{R}^d$. Choose m basis vectors indexed by $\mathcal{I}, \mathcal{I}' \subset \{1, \ldots, d\}$ to form orthonormal basis matrices $\boldsymbol{F}, \boldsymbol{F}' \in \mathbb{R}^{d\times m}$, 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 δ_{ij}, $\boldsymbol{b}_i{}^\top\boldsymbol{b}_j = \delta_{ij}$, so

$$\begin{aligned}\mathrm{PK}(\mathcal{F}, \mathcal{F}') &= \left\|\boldsymbol{F}^\top\boldsymbol{F}'\right\|_{\mathrm{F}}^2 \\ &= \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{I}'}(\boldsymbol{b}_i{}^\top\boldsymbol{b}_j)^2 = \left|\mathcal{I} \cap \mathcal{I}'\right|. \quad (11)\end{aligned}$$

[3] In general, for a random vector $\boldsymbol{x}$ satisfying $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = I$, we have $\mathbb{E}_{\boldsymbol{x}}[\|\boldsymbol{A}\boldsymbol{x}\|^2] = \mathrm{tr}(\boldsymbol{A}^\top\boldsymbol{A}\,\mathbb{E}_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{x}^\top]) = \mathrm{tr}(\boldsymbol{A}^\top\boldsymbol{A}) = \|\boldsymbol{A}\|_{\mathrm{F}}^2$. Thus, $\|\boldsymbol{A}\|_{\mathrm{F}}$ is the average magnitude of $\boldsymbol{A}\boldsymbol{x}$ under the assumption that $\boldsymbol{x}$ is isotropically distributed.

[4] In practice, QK/OV norms vary by layer (Appendix B), and CS changes markedly under preprocessing (Appendix K).

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

[6] Since $-1 \leq \cos\theta_i \leq 1$, we have $0 \leq \cos^2\theta_i \leq 1$.

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 now use PK to measure similarity between attention-head weights. We define the set of weight matrices $\mathcal{W} = \bigcup_{h \in \mathcal{H}} \left\{ \boldsymbol{W}_{\mathrm{Q}}^{h\top}, \boldsymbol{W}_{\mathrm{K}}^{h\top}, \boldsymbol{W}_{\mathrm{V}}^{h\top}, \boldsymbol{W}_{\mathrm{O}}^{h} \right\} \subset \mathbb{R}^{d \times d_{\mathrm{head}}}$, where $\mathcal{H}$ is the set of all heads. For weights $\boldsymbol{W}^r, \boldsymbol{W}^{r'} \in \mathcal{W}$, let $\mathcal{S}^r$ and $\mathcal{S}^{r'}$ denote the subspaces spanned by their column vectors[7]. In this paper, we assume $\dim \mathcal{S}^r = \dim \mathcal{S}^{r'} = d_{\mathrm{head}}$, and measure similarity between attention-head weight subspaces using $\mathrm{PK}(\mathcal{S}^r, \mathcal{S}^{r'})$ (see Fig. 2). As with CS, for example, we denote the PK score between $\boldsymbol{W}^{(h,\,\mathrm{O})}$ and $\boldsymbol{W}^{(h',\,\mathrm{Q})}$ by $\mathrm{PK}_{\mathrm{OQ}}(h, h')$.

As an example, consider the OQ pairing to compare PK and CS. Unlike $\mathrm{CS}_{\mathrm{OQ}}$, which uses the QK and OV matrices in (1) and (2), $\mathrm{PK}_{\mathrm{OQ}}$ does not involve $\boldsymbol{W}^{(h,\mathrm{K})}$ or $\boldsymbol{W}^{(h',\mathrm{V})}$. 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, $\mathrm{CS}_{\mathrm{OK}}$ 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)[8].

Figure 4 shows the results. In Fig. 4a, $\mathrm{PK}_{\mathrm{OK}}$ 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 $\mathrm{CS}_{\mathrm{OK}}$ 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.

4 Interpreting head subspaces

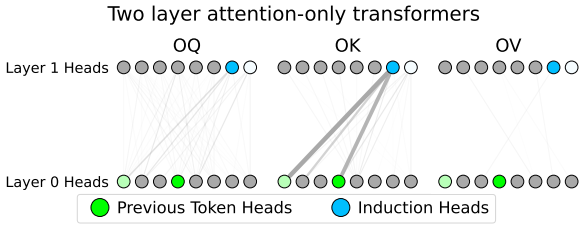
In Section 3, we use PK to measure the similarity between attention-head weight subspaces. In this section, we identify which vocabulary tokens are strongly associated with each attention head from a subspace perspective. Specifically, we project the unembedding matrix onto the subspace spanned by each 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) layer $\mathrm{LN}_{\mathrm{final}}$. Following this idea, for $\boldsymbol{W}^r \in \mathcal{W}$ we define $\widetilde{\boldsymbol{W}}^r := \mathrm{LN}_{\mathrm{final}}(\boldsymbol{W}^r) \in \mathbb{R}^{d \times d_{\mathrm{head}}}$ and let $\widetilde{\mathcal{S}}^r$ be the subspace spanned by the columns of $\widetilde{\boldsymbol{W}}^r$. The orthogonal projection matrix onto $\widetilde{\mathcal{S}}^r$ is $\widetilde{\boldsymbol{P}}^r = \widetilde{\boldsymbol{W}}^r \left(\widetilde{\boldsymbol{W}}^{r\top} \widetilde{\boldsymbol{W}}^r \right)^{-1} \widetilde{\boldsymbol{W}}^{r\top} \in \mathbb{R}^{d \times d}$.

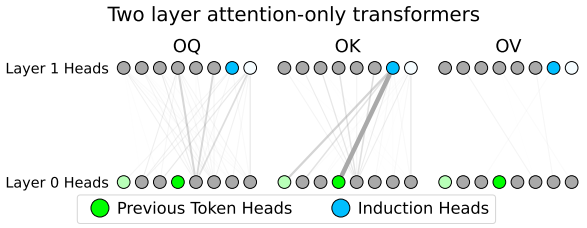
Let $\mathcal{T}$ be the vocabulary. For the unembedding matrix $\boldsymbol{E}_{\mathrm{out}} \in \mathbb{R}^{d \times |\mathcal{T}|}$, let $\boldsymbol{e}_t \in \mathbb{R}^d$ denote its column for token t, which we call the unembedding vector. After preprocessing $\boldsymbol{e}_t$ with a function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

$$\left\| \widetilde{\boldsymbol{P}}^r \frac{\boldsymbol{e}_t - \bar{\boldsymbol{e}}}{\|\boldsymbol{e}_t - \bar{\boldsymbol{e}}\|} \right\|. \tag{12}$$



(a) Projection Kernel (PK)



(b) Composition Score (CS)

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

[7] For example, when $r = (h, \mathrm{Q})$ we have $\boldsymbol{W}^r = \boldsymbol{W}_{\mathrm{Q}}^{h\top}$.

[8] This model uses LayerNorm as pre-norm and final layer.

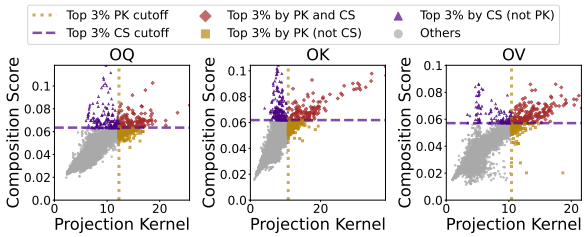


Figure 5: We compute PK and CS scores for OQ, OK, and OV and plot them as scatter plots. The Spearman's ρ values over all pairs are 0.885, 0.844, and 0.866, indicating strong correlations. In contrast, when we restrict to pairs that belong to the top 3% in either metric (the ♦, ■, and ▲ symbols), the 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 ▲ symbol), many have small PK scores. For visibility, we plot values up to the 99.99th percentile.

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 there are many Previous Token Heads and Induction Heads compared with head classes such as Duplicate Token Heads[9] (see Fig. 17 in Appendix G). We therefore recompute head scores for these two classes and assign the Previous Token Head or Induction Head label to unannotated heads among the top 10 heads in each class. The full list of annotations and details are provided in Appendix G.

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 largely 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.

[9]Moreover, 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.

6.2 Wiring diagrams between 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 $\boldsymbol{W}_{\text{O}}^{h}$ using the projection method in Section 4.

6.2.2 Results for PK

Figure 6 shows the PK wiring diagram.

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

OK. As in Fig. 4, scores are large 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, when L4H7 or L11H3 is the target head, scores are large from many source heads. We discuss such inlet heads in Section 8.

6.2.3 Results for CS

In the CS wiring diagram (Figure 13 in Appendix D), many unannotated heads appear in shallow layers. As a result, the Name-Mover-related heads that are prominent under PK, which are primarily found in deeper 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 deep layers, top tokens of heads are relatively easy to interpret. For example, L10H6

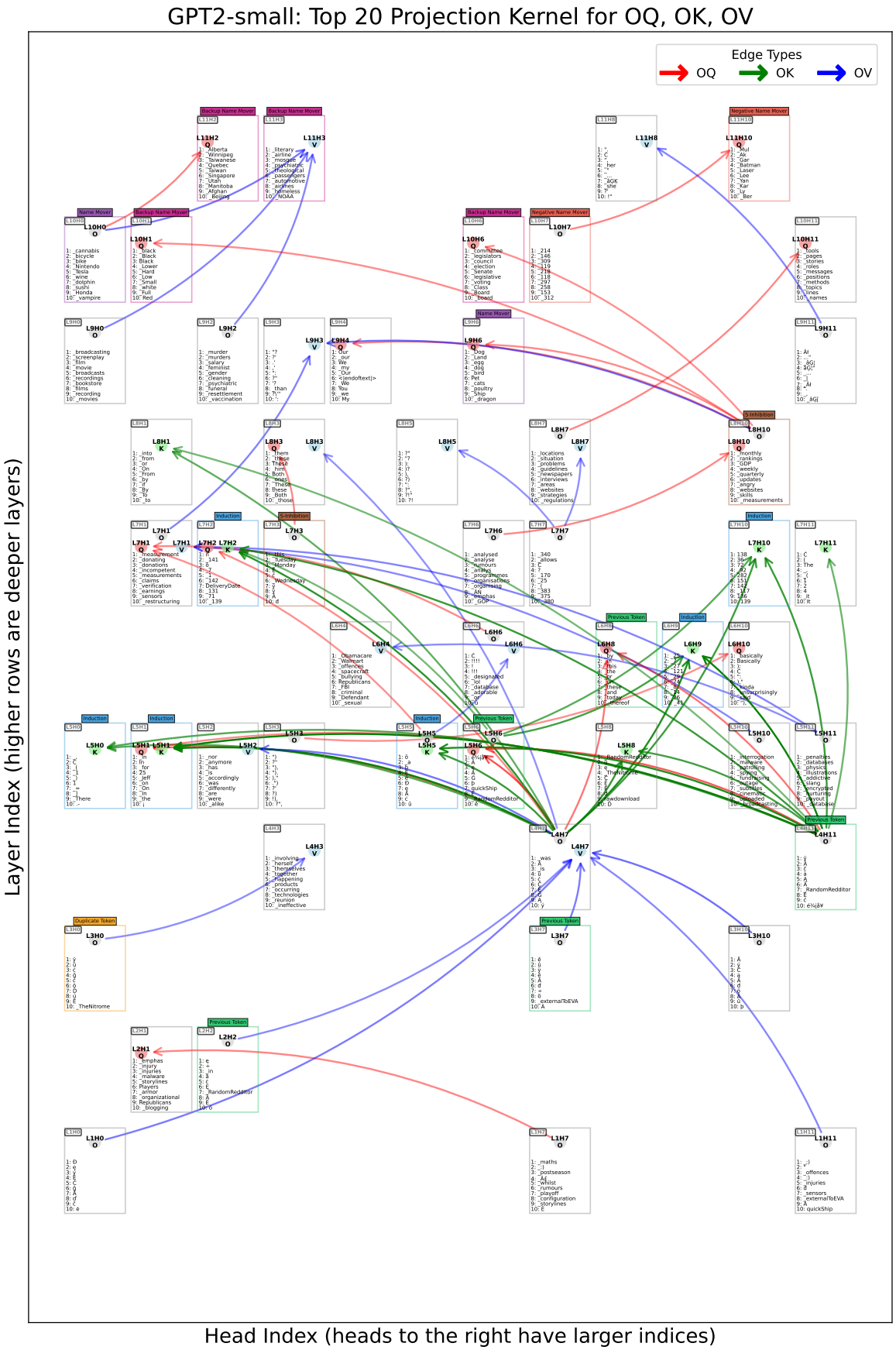


Figure 6: Visualization of the top 20 scores of $\mathrm{PK}_{\mathrm{OQ}}$, $\mathrm{PK}_{\mathrm{OK}}$, and $\mathrm{PK}_{\mathrm{OV}}$. For each head h, we show the top 10 tokens that represent $\boldsymbol{W}_{\mathrm{O}}^{h}$ using the projection method in Section 4. For readability, we replace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

	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	0.359	0.305	0.303
	Linear CKA	0.337	0.323	0.409	0.356
	Proc. Sim.	0.328	0.269	**0.428**	0.342

Table 1: PR-AUC scores for detecting annotated heads. We treat heads in head classes (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 in underline.

Method	PR-AUC	ROC-AUC
PK	**0.047**	**0.809**
CS	0.013	0.785
Simple-CS	0.016	0.792
Linear CKA	0.021	0.715
Proc. Sim.	0.038	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 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 F.2 for details.

contains many tokens related to politics. Across head classes, Name-Mover-related heads mainly attend to nouns, whereas Induction Heads often focus on numbers. Heads outside the head classes are also interpretable, such as L9H4, which corresponds to personal pronouns, and L8H5, which corresponds to punctuation such as "?" and "!". Heads in shallow 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 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 $\mathrm{CS}\left(\boldsymbol{W}_{\mathrm{QK}}^{h'\top}, \boldsymbol{W}_{\mathrm{OV}}^{h}\right)$ to measure, for example, the relationship between outputs and queries. We also include as a baseline a simpler method that directly computes $\mathrm{CS}(\boldsymbol{W}_{\mathrm{Q}}^{h'\top}, \boldsymbol{W}_{\mathrm{O}}^{h})$, 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 F.1 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.

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

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 directed head pairs whose source is in an earlier or the same layer and whose target is in the same or a later layer[10]. 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 small[11], 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 observed in Section 6.2.2, then discuss heads that act as hubs. We then compare PK distributions across weight pairing types.

8.1 L4H7 is an Identity Head

In Fig. 6, L4H7 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

[10] We exclude pairs of identical heads, leaving 10,296 pairs.
[11] For example, there are only 15 pairs of Induction Heads.

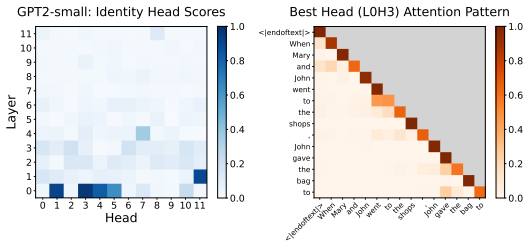


Figure 7: For GPT2-small, (left) Identity Head scores for each head and (right) an example attention pattern of the L0H3 head, which has the largest score. In particular, several heads in shallow layers and L4H7 have large scores. See Appendix G for details.

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

Table 3: Top five heads by inlet and outlet scores for OQ, OK, and OV, defined in (67) and (69). 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 (including Identity Heads), we also show the class labels. See Appendix I for details.

that several heads in shallow layers and L4H7 have large identity head scores (see Appendix G).

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 (equations (67) and (69) in Appendix I).

Table 3 lists the top five heads by inlet and outlet scores. For inlet scores, heads in middle and deep layers tend to rank highly, and for OV, L4H7 has the largest value. In contrast, for outlet scores, heads in shallow 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., 2025), 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.

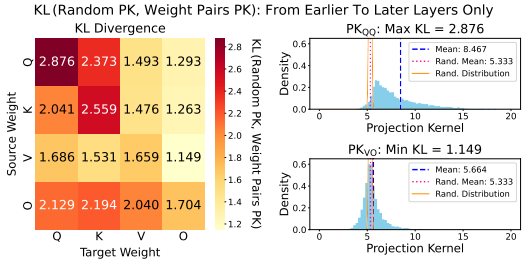


Figure 8: (Left) KL divergence between the PK distribution for random orthogonal matrices in (13) and the PK distributions measured for each weight pairing type, and (right) the PK distributions for QQ, which attains the maximum, and VO, which attains the minimum. In the left panel, the KL scores are 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 largely 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 J for details):

$$\mathcal{N}\left(\frac{d_{\text{head}}^2}{d}, \frac{2d_{\text{head}}^2(d-d_{\text{head}})^2}{d^2(d-1)(d+2)}\right) \tag{13}$$

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.

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.

Code availability

Our code will be available on GitHub. The URL is hidden during peer review.

Limitations

- Following Wang et al. (2023), we focus our analysis on GPT2-small, a standard baseline model in mechanistic interpretability. As a result, we do not evaluate larger models such as BLOOM (BigScience Workshop et al., 2023) and Qwen2 (Yang et al., 2024), whose head functions have been studied by Zhang et al. (2025). Extending our analysis to such models is left for future work. However, as a comparison study on the original weights of GPT2-small, we also apply weight preprocessing (Nanda and Bloom, 2022) that simplifies pretrained weights without changing model outputs. We find that PK is largely unchanged, whereas CS changes substantially. See Appendix K for details.

- As in Elhage et al. (2021), our method does not account for Layer Normalization (LN) (Ba et al., 2016). Moreover, recent large models often use Root Mean Square Layer Normalization 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.

- As in Elhage et al. (2021), our method also ignores feed-forward layers, and therefore cannot fully account for the internal computations of language models.

- PK provides similarity between attention heads for weight pairings, 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.

- 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. 10, CS: Fig. 12 in Appendix D) yields a sparser diagram, whereas increasing it to the top 30 (PK: Fig. 11, CS: Fig. 14 in Appendix D) makes the diagram much denser.

- 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. 11 in Appendix D), 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.

- 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). While PK captures many of these relationships, CS and other baselines may capture different types of head-to-head relationships that are not represented by the IOI-based annotations.

- Although many heads exhibit multiple functions, 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 7d in Appendix G[12].

- The inlet and outlet scores defined in Appendix I (equations (67) and (69)) 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 shallow-layer heads.

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

- 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 deep 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 (equation (15) in Appendix B), in a manner analogous to the Logit Lens[13]. 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.

- 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 E). We adopt normalization after centering, but other preprocessing choices may be more appropriate.

- 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.

- GPT2-small uses absolute positional embeddings, whereas many recent large language models use relative positional encodings such as RoPE (Su et al., 2024). RoPE applies a position-dependent rotation to the query and key vectors before computing attention scores, but we do not evaluate whether PK works well for models that use RoPE.

- 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 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.

[13] We use https://huggingface.co/datasets/NeelNanda/openwebtext-tokenized-9b.

References

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 Jermyn, 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, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, and 374 others. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

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.

Mohammed Dahleh, Munther A. Dahleh, and George C. Verghese. 2011. Lectures on dynamic systems and control. MIT OpenCourseWare.

Germund Dahlquist and Åke Björck. 2008. *Numerical methods in scientific computing, volume I*. 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, Álvaro Arroyo, Federico Barbero, Xiaowen Dong, Michael Bronstein, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Attention sinks and compression valleys in llms are two sides of the same coin. *Preprint*, arXiv:2510.06477.

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.

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*.

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.

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 Ham and Daniel D. Lee. 2008. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 376–383. ACM.

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 A. Smith. 2024. Data mixture inference attack: BPE tokenizers reveal training data compositions. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

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*.

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 6324–6343, 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. In *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. In *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*.

Hossein Pishro-Nik. 2014. Introduction to probability, statistics, and random processes.

Rudolph W Preisendorfer. 1982. *Data Intercomparison Theory*, volume 38. US Department of Commerce, National Oceanic and Atmospheric Administration, Environmental Research Laboratories.

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.

Jessica Rumbelow and Matthew Watkins. 2023. Solid-goldmagikarp (plus, prompt generation). In *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.

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. Candés. 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).

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. In *AI Alignment Forum*.

Sarah Wiegreffe 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, 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 Related Work

A.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 methods propose 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 on how universal circuits and head functions are 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.

A.2 Model weight analysis

In contrast to input-dependent analyses, some studies characterize model properties directly from weights. Elhage et al. (2021) introduced the Composition Score, which quantifies head-to-head relationships using the QK and OV matrices and their compositions. 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) performed an input independent analysis of a replacement model by using a correction that accounts for interference.

Another line of work interprets tokens associated with large components using the SVD of the OV matrix, or using products of weight matrices and the unembedding matrix (Millidge and Black, 2022; Dar et al., 2023). Our method, which identifies representative tokens by orthogonally projecting the unembedding matrix onto a specific attention-head subspace, is related to these approaches. In addition, Kamoda et al. (2025) analyzed the first-layer weights of GPT-2 to explain detokenization without running inference.

A.3 Attention pattern analysis

In characterizing attention-head functions, in addition to circuit identification, analyses based on visualizations and statistics of attention matrices (patterns) have long been used. 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 (Wiegreffe and Pinter, 2019; Serrano and Smith, 2019), and explanations can be intentionally 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).

B 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, show that CS lies in $[0, 1]$, and analyze how the Frobenius norms of the QK and OV matrices vary across layers in GPT2-small.

B.1 Background: Deriving the QK and OV matrices from single-layer attention

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

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

$\boldsymbol{W}_{\mathrm{K}}^h \boldsymbol{X} \in \mathbb{R}^{d_{\text{head}} \times n}$ as

$$
\begin{aligned}
& a^h(\boldsymbol{X}) = \sigma\left(\left(\boldsymbol{W}_{\mathrm{Q}}^h \boldsymbol{X}\right)^\top \boldsymbol{W}_{\mathrm{K}}^h \boldsymbol{X}\right) \\
& = \sigma\left(\boldsymbol{X}^\top \boldsymbol{W}_{\mathrm{Q}}^{h\top} \boldsymbol{W}_{\mathrm{K}}^h \boldsymbol{X}\right) = \sigma\left(\boldsymbol{X}^\top \boldsymbol{W}_{\mathrm{QK}}^h \boldsymbol{X}\right),
\end{aligned} \tag{14}
$$

where $\sigma(\cdot)$ denotes the softmax that includes normalization by $\sqrt{d_{\text{head}}}$ and the causal mask[14]. Equation (14) contains the QK matrix $\boldsymbol{W}_{\mathrm{QK}}^h \in \mathbb{R}^{d \times d}$ in (1). For simplicity, we write $\boldsymbol{A}^h = a^h(\boldsymbol{X})$.

Next, the write from head h to the residual stream, defined as the sum of layer outputs from the input embeddings to the final layer (Elhage et al., 2021), is $o^h(\boldsymbol{X}) \in \mathbb{R}^{d \times n}$. Using the attention pattern $\boldsymbol{A}^h$, the value matrix $\boldsymbol{W}_{\mathrm{V}}^h \boldsymbol{X} \in \mathbb{R}^{d_{\text{head}} \times n}$, and the output weight matrix $\boldsymbol{W}_{\mathrm{O}}^h \in \mathbb{R}^{d \times d_{\text{head}}}$, it is computed as

$$
o^h(\boldsymbol{X}) = \boldsymbol{W}_{\mathrm{O}}^h \boldsymbol{W}_{\mathrm{V}}^h \boldsymbol{X} \boldsymbol{A}^h = \boldsymbol{W}_{\mathrm{OV}}^h \boldsymbol{X} \boldsymbol{A}^h. \tag{15}
$$

Equation (15) contains the OV matrix $\boldsymbol{W}_{\mathrm{OV}}^h \in \mathbb{R}^{d \times d}$ in (2).

B.2 Background: Products of QK and OV matrices from two-layer attention

Following Elhage et al. (2021), we describe the attention computation across two layers[15] with input $\boldsymbol{X}$. 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

$$
\boldsymbol{O}^\ell = \sum_{h \in \mathcal{H}_\ell} o^h(\boldsymbol{X}) \in \mathbb{R}^{d \times n}. \tag{16}
$$

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

As in (14), the attention pattern $\boldsymbol{A}^{h'} \in [0,1]^{n \times n}$ is computed as

$$
\begin{aligned}
& \boldsymbol{A}^{h'} = a^{h'}(\boldsymbol{X} + \boldsymbol{O}^\ell) \\
& = \sigma\left((\boldsymbol{X} + \boldsymbol{O}^\ell)^\top \boldsymbol{W}_{\mathrm{QK}}^{h'} (\boldsymbol{X} + \boldsymbol{O}^\ell)\right) \in [0,1]^{n \times n}.
\end{aligned} \tag{17}
$$

The argument of $\sigma(\cdot)$ in (17), $(\boldsymbol{X} + \boldsymbol{O}^\ell)^\top \boldsymbol{W}_{\mathrm{QK}}^{h'} (\boldsymbol{X} + \boldsymbol{O}^\ell) \in \mathbb{R}^{n \times n}$, can be expanded as

$$
\begin{aligned}
& (\boldsymbol{X} + \boldsymbol{O}^\ell)^\top \boldsymbol{W}_{\mathrm{QK}}^{h'} (\boldsymbol{X} + \boldsymbol{O}^\ell) \\
& = \left(\boldsymbol{X} + \sum_{h \in \mathcal{H}_\ell} o^h(\boldsymbol{X})\right)^\top \\
& \quad \boldsymbol{W}_{\mathrm{QK}}^{h'} \left(\boldsymbol{X} + \sum_{h \in \mathcal{H}_\ell} o^h(\boldsymbol{X})\right) \quad (\because (16)) \\
& = \boldsymbol{X}^\top \boldsymbol{W}_{\mathrm{QK}}^{h'} \boldsymbol{X} + \sum_{h \in \mathcal{H}_\ell} o^h(\boldsymbol{X})^\top \boldsymbol{W}_{\mathrm{QK}}^{h'} \boldsymbol{X} \\
& + \sum_{h \in \mathcal{H}_\ell} \boldsymbol{X}^\top \boldsymbol{W}_{\mathrm{QK}}^{h'} o^h(\boldsymbol{X}) \\
& + \left(\sum_{h \in \mathcal{H}_\ell} o^h(\boldsymbol{X})\right)^\top \boldsymbol{W}_{\mathrm{QK}}^{h'} \left(\sum_{h \in \mathcal{H}_\ell} o^h(\boldsymbol{X})\right).
\end{aligned} \tag{18}
$$

To examine weight relationships between heads h and h', we focus on the summand in the second term, which uses $o^h(\boldsymbol{X})$ and $\boldsymbol{X}$ as inputs to the query and key computations, and the summand in the third term, which uses $\boldsymbol{X}$ and $o^h(\boldsymbol{X})$.

For the summand $o^h(\boldsymbol{X})^\top \boldsymbol{W}_{\mathrm{QK}}^{h'} \boldsymbol{X} \in \mathbb{R}^{n \times n}$ in the second term of (18), we have

$$
\begin{aligned}
& o^h(\boldsymbol{X})^\top \boldsymbol{W}_{\mathrm{QK}}^{h'} \boldsymbol{X} \\
& = \left(\boldsymbol{W}_{\mathrm{OV}}^h \boldsymbol{X} \boldsymbol{A}^h\right)^\top \boldsymbol{W}_{\mathrm{QK}}^{h'} \boldsymbol{X} \quad (\because (15)) \\
& = \boldsymbol{A}^{h\top} \boldsymbol{X}^\top \underbrace{\boldsymbol{W}_{\mathrm{OV}}^{h\top} \boldsymbol{W}_{\mathrm{QK}}^{h'}}_{\text{head-}h\ \mathrm{O} \to \text{head-}h'\ \mathrm{Q}} \boldsymbol{X}.
\end{aligned} \tag{19}
$$

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

For the summand $\boldsymbol{X}^\top \boldsymbol{W}_{\mathrm{QK}}^{h'} o^h(\boldsymbol{X}) \in \mathbb{R}^{n \times n}$ in the third term of (18), we have

$$
\begin{aligned}
& \boldsymbol{X}^\top \boldsymbol{W}_{\mathrm{QK}}^{h'} o^h(\boldsymbol{X}) \\
& = \boldsymbol{X}^\top \underbrace{\boldsymbol{W}_{\mathrm{QK}}^{h'} \boldsymbol{W}_{\mathrm{OV}}^h}_{\text{head-}h'\ \mathrm{K} \leftarrow \text{head-}h\ \mathrm{O}} \boldsymbol{X} \boldsymbol{A}^h \quad (\because (15)).
\end{aligned} \tag{20}
$$

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

As in (15), the write of head h' to the residual

[14] $\sigma(\boldsymbol{Y}) = \mathrm{softmax}(\boldsymbol{Y}/\sqrt{d_{\text{head}}} + \boldsymbol{M}_{\text{mask}})$ for $\boldsymbol{Y} \in \mathbb{R}^{n \times n}$, where $\boldsymbol{M}_{\text{mask}} \in \mathbb{R}^{n \times n}$ is a mask matrix whose upper triangular entries excluding the diagonal are $-\infty$.

[15] Note that we ignore the Layer Normalization and feed-forward layers.

stream, $o^{h'}(\boldsymbol{X}+\boldsymbol{O}^{\ell}) \in \mathbb{R}^{d\times n}$, is computed as

$$
\begin{aligned}
& o^{h'}(\boldsymbol{X}+\boldsymbol{O}^{\ell}) \\
&= \boldsymbol{W}_{\mathrm{OV}}^{h'}\left(\boldsymbol{X}+\sum_{h\in\mathcal{H}_{\ell}} o^{h}(\boldsymbol{X})\right)\boldsymbol{A}^{h'} \quad (\because (16)) \\
&= \boldsymbol{W}_{\mathrm{OV}}^{h'}\boldsymbol{X}\boldsymbol{A}^{h'} + \sum_{h\in\mathcal{H}_{\ell}} \underbrace{\boldsymbol{W}_{\mathrm{OV}}^{h'}\boldsymbol{W}_{\mathrm{OV}}^{h}}_{\text{head-}h'\ \mathrm{V}\ \leftarrow\ \text{head-}h\ \mathrm{O}} \boldsymbol{X}\boldsymbol{A}^{h}\boldsymbol{A}^{h'} \\
&\quad (\because (15)). \qquad (21)
\end{aligned}
$$

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

B.3 Matrices used to compute CS

In Section B.2, we saw that the matrix products ${\boldsymbol{W}_{\mathrm{QK}}^{h'}}^{\top}\boldsymbol{W}_{\mathrm{OV}}^{h}$ for OQ, $\boldsymbol{W}_{\mathrm{QK}}^{h'}\boldsymbol{W}_{\mathrm{OV}}^{h}$ for OK, and $\boldsymbol{W}_{\mathrm{OV}}^{h'}\boldsymbol{W}_{\mathrm{OV}}^{h}$ for OV represent weight relationships. Based on these products, Elhage et al. (2021) compute CS in (6) as the Frobenius norm of each product normalized by the product of the Frobenius norms of the original matrices, which yields $\mathrm{CS}_{\mathrm{OQ}}$ in (3), $\mathrm{CS}_{\mathrm{OK}}$ in (4), and $\mathrm{CS}_{\mathrm{OV}}$ in (5).

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 Section B.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 $\boldsymbol{W}_{\mathrm{QK}}^{h}$, ${\boldsymbol{W}_{\mathrm{QK}}^{h}}^{\top}$, ${\boldsymbol{W}_{\mathrm{OV}}^{h}}^{\top}$, and $\boldsymbol{W}_{\mathrm{OV}}^{h}$, and the matrices associated with the query, key, value, and output of head h' as ${\boldsymbol{W}_{\mathrm{QK}}^{h}}^{\top}$, $\boldsymbol{W}_{\mathrm{QK}}^{h}$, $\boldsymbol{W}_{\mathrm{OV}}^{h}$, and ${\boldsymbol{W}_{\mathrm{OV}}^{h}}^{\top}$. We compute CS between head weights based on these matrices[16].

B.4 Range of CS

In this section, assuming $\|\boldsymbol{W}^{h'}\|_{\mathrm{F}}\|\boldsymbol{W}^{h}\|_{\mathrm{F}} \neq 0$, we show that CS in (6) satisfies

$$
0 \le \mathrm{CS}(\boldsymbol{W}^{h'},\boldsymbol{W}^{h}) = \frac{\|\boldsymbol{W}^{h'}\boldsymbol{W}^{h}\|_{\mathrm{F}}}{\|\boldsymbol{W}^{h'}\|_{\mathrm{F}}\|\boldsymbol{W}^{h}\|_{\mathrm{F}}} \le 1. \qquad (22)
$$

[16]Our 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.

When $\|\boldsymbol{W}^{h'}\|_{\mathrm{F}} = 0$ or $\|\boldsymbol{W}^{h}\|_{\mathrm{F}} = 0$, we define $\mathrm{CS}(\boldsymbol{W}^{h'},\boldsymbol{W}^{h}) = 0$.

For $\boldsymbol{W}^{h'} \in \mathbb{R}^{d\times d}$, let $\|\boldsymbol{W}^{h'}\|_2$ be the operator norm:

$$
\|\boldsymbol{W}^{h'}\|_2 := \sup_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{W}^{h'}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}, \qquad (23)
$$

where $\|\cdot\|$ denotes the Euclidean norm for vectors in $\mathbb{R}^d$. Let $\sigma_1^{h'} \ge \ldots \ge \sigma_d^{h'} \ge 0$ be the singular values of $\boldsymbol{W}^{h'}$. Since $\|\boldsymbol{W}^{h'}\|_2 = \sigma_1^{h'}$ (Dahleh et al., 2011, Theorem 4.2) and $\|\boldsymbol{W}^{h'}\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^{d}\sigma_i^{h'2}}$ (Dahleh et al., 2011, (4.23)), we have $\|\boldsymbol{W}^{h'}\|_2 \le \|\boldsymbol{W}^{h'}\|_{\mathrm{F}}$. Moreover, when $\|\boldsymbol{W}^{h'}\|_{\mathrm{F}} \neq 0$, the equality $\|\boldsymbol{W}^{h'}\|_2 = \|\boldsymbol{W}^{h'}\|_{\mathrm{F}}$ holds if and only if $\sigma_2^{h'} = \cdots = \sigma_d^{h'} = 0$, equivalently $\mathrm{rank}(\boldsymbol{W}^{h'}) = 1$.

Let $\boldsymbol{W}^{h} \in \mathbb{R}^{d\times d}$, and denote its i-th column by $\boldsymbol{w}_i^{h} \in \mathbb{R}^{d}$. Then

$$
\begin{aligned}
\|\boldsymbol{W}^{h'}\boldsymbol{W}^{h}\|_{\mathrm{F}}^2 &= \sum_{i=1}^{d}\|\boldsymbol{W}^{h'}\boldsymbol{w}_i^{h}\|^2 \\
&\le \sum_{i=1}^{d}\|\boldsymbol{W}^{h'}\|_2^2\|\boldsymbol{w}_i^{h}\|^2 = \|\boldsymbol{W}^{h'}\|_2^2\|\boldsymbol{W}^{h}\|_{\mathrm{F}}^2 \\
&\le \|\boldsymbol{W}^{h'}\|_{\mathrm{F}}^2\|\boldsymbol{W}^{h}\|_{\mathrm{F}}^2. \qquad (24)
\end{aligned}
$$

Equality in the inequality $\|\boldsymbol{W}^{h'}\boldsymbol{w}_i^{h}\| \le \|\boldsymbol{W}^{h'}\|_2\|\boldsymbol{w}_i^{h}\|$ holds if and only if $\boldsymbol{w}_i^{h} = \boldsymbol{0}$ or $\boldsymbol{w}_i^{h}$ lies in the right singular subspace of $\boldsymbol{W}^{h'}$ associated with the largest singular value $\sigma_1^{h'}$ (including multiplicity)[17]. Therefore, the equality

[17]Assume $\boldsymbol{w}_i^{h} \neq \boldsymbol{0}$ and let $\boldsymbol{W}^{h'} = \boldsymbol{U}^{h'}\boldsymbol{\Sigma}^{h'}{\boldsymbol{V}^{h'}}^{\top}$ be the SVD, where $\boldsymbol{U}^{h'},\boldsymbol{V}^{h'} \in \mathbb{R}^{d\times d}$ are orthogonal and $\boldsymbol{\Sigma}^{h'} \in \mathbb{R}^{d\times d}$ is diagonal with entries $\sigma_1^{h'},\ldots,\sigma_d^{h'}$. Let $\boldsymbol{v}_1^{h'},\ldots,\boldsymbol{v}_d^{h'} \in \mathbb{R}^{d}$ be the right singular vectors, that is, the columns of $\boldsymbol{V}^{h'}$. Let $\boldsymbol{\eta} = {\boldsymbol{V}^{h'}}^{\top}(\boldsymbol{w}_i^{h}/\|\boldsymbol{w}_i^{h}\|)$, so $\|\boldsymbol{\eta}\| = 1$. Then

$$
\begin{aligned}
\frac{\|\boldsymbol{W}^{h'}\boldsymbol{w}_i^{h}\|^2}{\|\boldsymbol{w}_i^{h}\|^2} &= \left\|\boldsymbol{U}^{h'}\boldsymbol{\Sigma}^{h'}{\boldsymbol{V}^{h'}}^{\top}\left(\frac{\boldsymbol{w}_i^{h}}{\|\boldsymbol{w}_i^{h}\|}\right)\right\|^2 \\
&= \|\boldsymbol{U}^{h'}\boldsymbol{\Sigma}^{h'}\boldsymbol{\eta}\|^2 = \|\boldsymbol{\Sigma}^{h'}\boldsymbol{\eta}\|^2 = \sum_{j=1}^{d}\sigma_j^{h'2}\eta_j^2 \\
&\le \sigma_1^{h'2}\sum_{j=1}^{d}\eta_j^2 = \sigma_1^{h'2} = \|\boldsymbol{W}^{h'}\|_2^2.
\end{aligned}
$$

Moreover, under the assumption $\boldsymbol{w}_i^{h} \neq \boldsymbol{0}$, equality holds if and only if $\eta_j = 0$ for all j with $\sigma_j^{h'} < \sigma_1^{h'}$. Equivalently, letting $\mathcal{I}_1 = \{j \in \{1,\ldots,d\} : \sigma_j^{h'} = \sigma_1^{h'}\}$, equality holds if and only if $\boldsymbol{\eta}$ is supported on $\mathcal{I}_1$, i.e., $\boldsymbol{\eta} \in \mathrm{span}\{\boldsymbol{e}_j^{\mathrm{std}} : j \in \mathcal{I}_1\}$, where $\boldsymbol{e}_j^{\mathrm{std}} \in \mathbb{R}^{d}$ denotes the j-th standard basis vector. Interpreting $\boldsymbol{\eta}$ as the coordinates of $\boldsymbol{w}_i^{h}/\|\boldsymbol{w}_i^{h}\|$ in the basis $\{\boldsymbol{v}_j^{h'}\}_{j=1}^{d}$, we have $\boldsymbol{w}_i^{h}/\|\boldsymbol{w}_i^{h}\| = \boldsymbol{V}^{h'}\boldsymbol{\eta}$; hence $\boldsymbol{\eta}$ being supported on $\mathcal{I}_1$ is equiv-

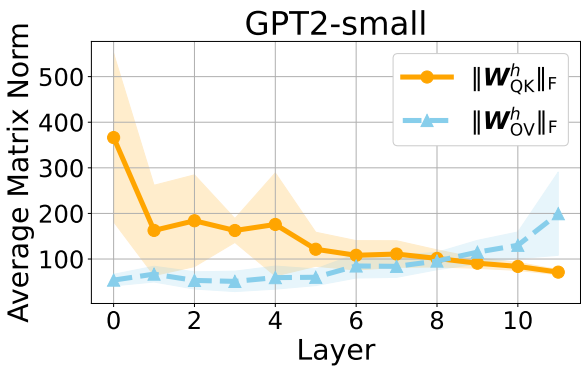


Figure 9: For the QK matrix in (1) and the OV matrix in (2), we plot the Frobenius norms across layers in GPT2-small. For each layer, we also show the 1σ range.

$\|\boldsymbol{W}^{h'}\boldsymbol{W}^h\|_F^2 \leq \|\boldsymbol{W}^{h'}\|_2^2\|\boldsymbol{W}^h\|_F^2$ holds if and only if for every i, either $\boldsymbol{w}_i^h = \boldsymbol{0}$ or $\boldsymbol{w}_i^h$ lies in this right singular subspace of $\boldsymbol{W}^{h'}$. Moreover, when $\|\boldsymbol{W}^{h'}\|_F \neq 0$, the equality $\|\boldsymbol{W}^{h'}\|_2^2 \leq \|\boldsymbol{W}^{h'}\|_F^2$ holds if and only if $\mathrm{rank}(\boldsymbol{W}^{h'}) = 1$.

Finally, dividing both sides of (24) by $\|\boldsymbol{W}^{h'}\|_F^2\|\boldsymbol{W}^h\|_F^2 \neq 0$ and taking square roots yields the range in (22).

B.5 Layerwise Frobenius norms of the QK and OV matrices in GPT2-small

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 (6) can be difficult to use for quantifying differences between heads.

C 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 J.

alent to $\boldsymbol{w}_i^h/\|\boldsymbol{w}_i^h\| \in \mathrm{span}\{\boldsymbol{v}_j^{h'} : j \in \mathcal{I}_1\}$. Since scaling is irrelevant for the equality condition, this is equivalent to $\boldsymbol{w}_i^h \in \mathrm{span}\{\boldsymbol{v}_j^{h'} : j \in \mathcal{I}_1\}$, the right singular subspace corresponding to $\sigma_1^{h'}$ (including multiplicity).

C.1 Solving the cosine similarity optimization for principal angles

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

$$\boldsymbol{M} = \boldsymbol{U}^\top \boldsymbol{U}' \in \mathbb{R}^{m\times m} \tag{25}$$

have singular values $\sigma_1 \geq \ldots \geq \sigma_m \geq 0$. 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 $\boldsymbol{M}$.

We begin with the optimization problem for the first principal angle. The following angle θ_1 is the first principal angle:

$$\cos\theta_1 = \max_{\substack{\boldsymbol{s}\in\mathcal{S},\ \|\boldsymbol{s}\|=1 \\ \boldsymbol{s}'\in\mathcal{S}',\ \|\boldsymbol{s}'\|=1}} \boldsymbol{s}^\top \boldsymbol{s}'. \tag{26}$$

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

$$\boldsymbol{s} = \boldsymbol{U}\boldsymbol{g},\ \boldsymbol{s}' = \boldsymbol{U}\boldsymbol{g}'. \tag{27}$$

For $\boldsymbol{s}$ in (27),

$$\begin{aligned}\|\boldsymbol{s}\|^2 &= \boldsymbol{s}^\top\boldsymbol{s} = (\boldsymbol{U}\boldsymbol{g})^\top\boldsymbol{U}\boldsymbol{g} \\ &= \boldsymbol{g}^\top\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{g} = \boldsymbol{g}^\top\boldsymbol{g} = \|\boldsymbol{g}\|^2,\end{aligned} \tag{28}$$

where we used $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}_m \in \mathbb{R}^{m\times m}$ since the columns of $\boldsymbol{U}$ are orthonormal. Therefore, $\|\boldsymbol{s}\| = 1$ implies $\|\boldsymbol{g}\| = 1$, and similarly $\|\boldsymbol{s}'\| = 1$ implies $\|\boldsymbol{g}'\| = 1$. For $\boldsymbol{s}, \boldsymbol{s}'$ in (27), we obtain

$$\begin{aligned}\boldsymbol{s}^\top\boldsymbol{s}' &= (\boldsymbol{U}\boldsymbol{g})^\top\boldsymbol{U}'\boldsymbol{g}' \\ &= \boldsymbol{g}^\top\boldsymbol{U}^\top\boldsymbol{U}'\boldsymbol{g}' = \boldsymbol{g}^\top\boldsymbol{M}\boldsymbol{g}'.\end{aligned} \tag{29}$$

Using $\|\boldsymbol{g}\| = \|\boldsymbol{g}'\| = 1$ and (29), (26) can be rewritten as

$$\cos\theta_1 = \max_{\|\boldsymbol{g}\|=\|\boldsymbol{g}'\|=1} \boldsymbol{g}^\top\boldsymbol{M}\boldsymbol{g}'. \tag{30}$$

Let $\boldsymbol{M}$ in (25) be written as $\boldsymbol{M} = \boldsymbol{U}_M\boldsymbol{\Sigma}_M{\boldsymbol{V}_M}^\top$ via the singular value decomposition (SVD). Here, the diagonal entries of $\boldsymbol{\Sigma}_M$ are the singular values $\sigma_1 \geq \ldots \geq \sigma_m \geq 0$, and $\boldsymbol{U}_M, \boldsymbol{V}_M \in \mathbb{R}^{m\times m}$ are orthogonal matrices. For $\boldsymbol{g}^\top\boldsymbol{M}\boldsymbol{g}'$ in (30),

$$\begin{aligned}\boldsymbol{g}^\top\boldsymbol{M}\boldsymbol{g}' &= \boldsymbol{g}^\top\boldsymbol{U}_M\boldsymbol{\Sigma}_M{\boldsymbol{V}_M}^\top\boldsymbol{g}' \\ &= ({\boldsymbol{U}_M}^\top\boldsymbol{g})^\top\boldsymbol{\Sigma}_M({\boldsymbol{V}_M}^\top\boldsymbol{g}').\end{aligned} \tag{31}$$

Define

$$\boldsymbol{\xi} := \boldsymbol{U}_M{}^\top \boldsymbol{g},\ \boldsymbol{\xi}' := \boldsymbol{V}_M{}^\top \boldsymbol{g}' \in \mathbb{R}^m. \tag{32}$$

Since $\boldsymbol{U}_M$ and $\boldsymbol{V}_M$ are orthogonal and $\|\boldsymbol{g}\| = \|\boldsymbol{g}'\| = 1$, we have $\|\boldsymbol{\xi}\| = \|\boldsymbol{\xi}'\| = 1$. Writing $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_m)^\top$ and $\boldsymbol{\xi}' = (\xi'_1, \ldots, \xi'_m)^\top$, (31) becomes

$$\boldsymbol{g}^\top \boldsymbol{M} \boldsymbol{g}' = \boldsymbol{\xi}^\top \boldsymbol{\Sigma}_M \boldsymbol{\xi}' = \sum_{j=1}^{m} \sigma_j \xi_j \xi'_j. \tag{33}$$

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

$$\begin{aligned} &\sum_{j=1}^{m} \sigma_j \xi_j \xi'_j \\ &\leq \left(\sum_{j=1}^{m} \sigma_j^2 {\xi_j}^2 \right)^{1/2} \left(\sum_{j=1}^{m} {\xi'_j}^2 \right)^{1/2} \\ &\leq \sigma_1 \left(\sum_{j=1}^{m} {\xi_j}^2 \right)^{1/2} \left(\sum_{j=1}^{m} {\xi'_j}^2 \right)^{1/2} = \sigma_1. \end{aligned} \tag{34}$$

Equality holds when $\boldsymbol{\xi}$ and $\boldsymbol{\xi}'$ equal the first standard basis vector $\boldsymbol{e}_1^{\text{std}} = (1, 0, \ldots, 0)^\top \in \mathbb{R}^m$. Therefore, for (30) we obtain

$$\cos\theta_1 = \max_{\|\boldsymbol{g}\|=\|\boldsymbol{g}'\|=1} \boldsymbol{g}^\top \boldsymbol{M} \boldsymbol{g}' = \sigma_1. \tag{35}$$

Let $\boldsymbol{\xi}_1 = \boldsymbol{\xi}'_1 = \boldsymbol{e}_1^{\text{std}} \in \mathbb{R}^m$ be the maximizers, and let $\boldsymbol{g}_1, \boldsymbol{g}'_1 \in \mathbb{R}^m$ be the corresponding coefficients. From (32), we have

$$\boldsymbol{g}_1 = \boldsymbol{U}_M \boldsymbol{\xi}_1 = \boldsymbol{U}_M \boldsymbol{e}_1^{\text{std}} = \boldsymbol{u}_{M,1}, \tag{36}$$

$$\boldsymbol{g}'_1 = \boldsymbol{V}_M \boldsymbol{\xi}'_1 = \boldsymbol{V}_M \boldsymbol{e}_1^{\text{std}} = \boldsymbol{v}_{M,1}, \tag{37}$$

where $\boldsymbol{u}_{M,1}$ and $\boldsymbol{v}_{M,1}$ are the first columns of $\boldsymbol{U}_M$ and $\boldsymbol{V}_M$. We call the maximizers $\boldsymbol{s} = \boldsymbol{s}_1$ and $\boldsymbol{s}' = \boldsymbol{s}'_1 \in \mathbb{R}^d$ in (26) the principal vectors (Dahlquist and Björck, 2008). By (27), the principal vectors are

$$\boldsymbol{s}_1 = \boldsymbol{U} \boldsymbol{g}_1 = \boldsymbol{U} \boldsymbol{u}_{M,1}, \tag{38}$$

$$\boldsymbol{s}'_1 = \boldsymbol{U}' \boldsymbol{g}'_1 = \boldsymbol{U}' \boldsymbol{v}_{M,1}. \tag{39}$$

Next, the following angle θ_2 is the second principal angle:

$$\cos\theta_2 = \max_{\substack{\boldsymbol{s}\in\mathcal{S},\ \|\boldsymbol{s}\|=1, \boldsymbol{s}^\top \boldsymbol{s}_1 = 0 \\ \boldsymbol{s}'\in\mathcal{S}',\ \|\boldsymbol{s}'\|=1, \boldsymbol{s}'^\top \boldsymbol{s}'_1 = 0}} \boldsymbol{s}^\top \boldsymbol{s}'. \tag{40}$$

From (27) we have $\boldsymbol{s} = \boldsymbol{U}\boldsymbol{g}$, and from (38) we have $\boldsymbol{s}_1 = \boldsymbol{U}\boldsymbol{g}_1$; therefore,

$$\begin{aligned} \boldsymbol{s}^\top \boldsymbol{s}_1 &= (\boldsymbol{U}\boldsymbol{g})^\top \boldsymbol{U}\boldsymbol{g}_1 \\ &= \boldsymbol{g}^\top \boldsymbol{U}^\top \boldsymbol{U} \boldsymbol{g}_1 = \boldsymbol{g}^\top \boldsymbol{g}_1. \end{aligned} \tag{41}$$

Similarly, $\boldsymbol{s}'^\top \boldsymbol{s}'_1 = \boldsymbol{g}'^\top \boldsymbol{g}'_1$. Combining these with $\|\boldsymbol{g}\| = \|\boldsymbol{g}'\| = 1$ and (29), we can rewrite (40) as

$$\cos\theta_2 = \max_{\substack{\|\boldsymbol{g}\|=1, \boldsymbol{g}^\top \boldsymbol{g}_1 = 0 \\ \|\boldsymbol{g}'\|=1, \boldsymbol{g}'^\top \boldsymbol{g}'_1 = 0}} \boldsymbol{g}^\top \boldsymbol{M} \boldsymbol{g}'. \tag{42}$$

Moreover, from (32) and (36),

$$\begin{aligned} \boldsymbol{g}^\top \boldsymbol{g}_1 &= (\boldsymbol{U}_M \boldsymbol{\xi})^\top \boldsymbol{U}_M \boldsymbol{\xi}_1 = \boldsymbol{\xi}^\top \boldsymbol{U}_M{}^\top \boldsymbol{U}_M \boldsymbol{\xi}_1 \\ &= \boldsymbol{\xi}^\top \boldsymbol{\xi}_1 = \boldsymbol{\xi}^\top \boldsymbol{e}_1^{\text{std}} = \xi_1. \end{aligned} \tag{43}$$

Similarly, $\boldsymbol{g}'^\top \boldsymbol{g}'_1 = \xi'_1$. Using $\|\boldsymbol{\xi}\| = \|\boldsymbol{\xi}'\| = 1$, (33), and (43), we can rewrite (42) as

$$\begin{aligned} \cos\theta_2 &= \max_{\substack{\|\boldsymbol{\xi}\|=1, \xi_1 = 0 \\ \|\boldsymbol{\xi}'\|=1, \xi'_1 = 0}} \boldsymbol{\xi}^\top \Sigma_M \boldsymbol{\xi}' \\ &= \max_{\substack{\|\boldsymbol{\xi}\|=1, \xi_1 = 0 \\ \|\boldsymbol{\xi}'\|=1, \xi'_1 = 0}} \sum_{j=2}^{m} \sigma_j \xi_j \xi'_j. \end{aligned} \tag{44}$$

By the same argument as in (34), the maximum is attained at $\boldsymbol{\xi} = \boldsymbol{\xi}' = \boldsymbol{e}_2^{\text{std}}$, and we obtain $\cos\theta_2 = \sigma_2$. Repeating the same steps for the i-th principal angle yields

$$\cos\theta_i = \sigma_i. \tag{45}$$

Moreover, as in (38), if $\boldsymbol{S}, \boldsymbol{S}' \in \mathbb{R}^{d\times m}$ are matrices whose columns are the principal vectors, then

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{U}_M,\ \boldsymbol{S}' = \boldsymbol{U}'\boldsymbol{V}_M. \tag{46}$$

C.2 Deriving the Projection Kernel using orthonormal bases

For $\mathrm{PK}(\mathcal{S}, \mathcal{S}')$ in (8), using $\cos\theta_i = \sigma_i$ in (45), we can derive $\mathrm{PK}(\mathcal{S}, \mathcal{S}') = \|\boldsymbol{U}^\top \boldsymbol{U}'\|_{\mathrm{F}}^2$ in (9) as follows:

$$\begin{aligned} &\mathrm{PK}(\mathcal{S}, \mathcal{S}') \\ &= \sum_{i=1}^{m} \cos^2\theta_i = \sum_{i=1}^{m} \sigma_i^2 = \mathrm{tr}\left(\boldsymbol{\Sigma}_M^\top \boldsymbol{\Sigma}_M\right) \\ &= \mathrm{tr}\left(\boldsymbol{V}_M{}^\top \boldsymbol{V}_M \boldsymbol{\Sigma}_M^\top \boldsymbol{U}_M{}^\top \boldsymbol{U}_M \boldsymbol{\Sigma}_M\right) \\ &= \mathrm{tr}\left(\boldsymbol{V}_M \boldsymbol{\Sigma}_M^\top \boldsymbol{U}_M{}^\top \boldsymbol{U}_M \boldsymbol{\Sigma}_M \boldsymbol{V}_M{}^\top\right) \\ &= \mathrm{tr}\left(\left(\boldsymbol{U}_M \boldsymbol{\Sigma}_M \boldsymbol{V}_M{}^\top\right)^\top \boldsymbol{U}_M \boldsymbol{\Sigma}_M \boldsymbol{V}_M{}^\top\right) \\ &= \mathrm{tr}\left(\boldsymbol{M}^\top \boldsymbol{M}\right) = \|\boldsymbol{M}\|_{\mathrm{F}}^2 = \|\boldsymbol{U}^\top \boldsymbol{U}'\|_{\mathrm{F}}^2. \end{aligned} \tag{47}$$

C.3 Deriving the Projection Kernel using orthogonal projections

For $\mathrm{PK}(\mathcal{S}, \mathcal{S}')$ in (8), using $\mathrm{PK}(\mathcal{S}, \mathcal{S}') = \|\boldsymbol{U}^\top \boldsymbol{U}'\|_{\mathrm{F}}^2$ in (47), we can derive $\mathrm{PK}(\mathcal{S}, \mathcal{S}') = \mathrm{tr}\,(\boldsymbol{P}\boldsymbol{P}')$ in (9) as follows:

$$\begin{aligned}
&\mathrm{PK}(\mathcal{S}, \mathcal{S}') \\
&= \|\boldsymbol{U}^\top \boldsymbol{U}'\|_{\mathrm{F}}^2 = \|\boldsymbol{M}\|_{\mathrm{F}}^2 = \mathrm{tr}\left(\boldsymbol{M}^\top \boldsymbol{M}\right) \\
&= \mathrm{tr}\left((\boldsymbol{U}^\top \boldsymbol{U}')^\top \boldsymbol{U}^\top \boldsymbol{U}'\right) = \mathrm{tr}\left(\boldsymbol{U}'^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{U}'\right) \\
&= \mathrm{tr}\left(\boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{U}'\boldsymbol{U}'^\top\right) = \mathrm{tr}\left(\boldsymbol{P}\boldsymbol{P}'\right). \qquad (48)
\end{aligned}$$

C.4 Relationship between the Projection Kernel and the Composition Score

Based on the definition of CS in (6), we show (10).

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

$$\begin{aligned}
\|\boldsymbol{P}\|_{\mathrm{F}}^2 &= \mathrm{tr}\left(\boldsymbol{P}^\top \boldsymbol{P}\right) = \mathrm{tr}\left((\boldsymbol{U}\boldsymbol{U}^\top)^\top \boldsymbol{U}\boldsymbol{U}^\top\right) \\
&= \mathrm{tr}\left(\boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{U}^\top\right) = \mathrm{tr}\left(\boldsymbol{U}\boldsymbol{I}_m\boldsymbol{U}^\top\right) \\
&= \mathrm{tr}\left(\boldsymbol{U}^\top \boldsymbol{U}\right) = \mathrm{tr}(\boldsymbol{I}_m) = m. \qquad (49)
\end{aligned}$$

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

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

$$\begin{aligned}
\|\boldsymbol{P}'\boldsymbol{P}\|_{\mathrm{F}}^2 &= \mathrm{tr}((\boldsymbol{P}'\boldsymbol{P})^\top \boldsymbol{P}'\boldsymbol{P}) = \mathrm{tr}(\boldsymbol{P}^\top \boldsymbol{P}'^\top \boldsymbol{P}'\boldsymbol{P}) \\
&= \mathrm{tr}(\boldsymbol{P}\boldsymbol{P}'\boldsymbol{P}'\boldsymbol{P}) = \mathrm{tr}(\boldsymbol{P}\boldsymbol{P}\boldsymbol{P}'\boldsymbol{P}') \\
&= \mathrm{tr}(\boldsymbol{P}\boldsymbol{P}') = \mathrm{PK}(\mathcal{S}, \mathcal{S}') \quad (\because (48)). \qquad (50)
\end{aligned}$$

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

By (49) and (50),

$$\mathrm{CS}(\boldsymbol{P}', \boldsymbol{P})^2 = \frac{\|\boldsymbol{P}'\boldsymbol{P}\|_{\mathrm{F}}^2}{\|\boldsymbol{P}'\|_{\mathrm{F}}^2 \|\boldsymbol{P}\|_{\mathrm{F}}^2} = \frac{\mathrm{PK}(\mathcal{S}, \mathcal{S}')}{m^2}, \quad (51)$$

which proves (10).

C.5 Rotation invariance of the Projection Kernel

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

$$\begin{aligned}
\mathrm{PK}(\mathcal{S}_\Psi, \mathcal{S}'_\Psi) &= \mathrm{tr}\left(\boldsymbol{R}\boldsymbol{U}(\boldsymbol{R}\boldsymbol{U})^\top \boldsymbol{R}\boldsymbol{U}'(\boldsymbol{R}\boldsymbol{U}')^\top\right) \\
&= \mathrm{tr}\left(\boldsymbol{R}\boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{R}^\top \boldsymbol{R}\boldsymbol{U}'\boldsymbol{U}'^\top \boldsymbol{R}^\top\right) \\
&= \mathrm{tr}\left(\boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{U}'\boldsymbol{U}'^\top\right) = \mathrm{tr}(\boldsymbol{P}\boldsymbol{P}') \\
&= \mathrm{PK}(\mathcal{S}, \mathcal{S}') \quad (\because (48)). \qquad (52)
\end{aligned}$$

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

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

D Details of wiring diagrams for Projection Kernel and Composition Score

In Section 6, we examined the PK wiring diagram showing the top 20 edges for each of the OQ, OK, and OV pairings in Fig. 6. In this section, we show wiring diagrams for PK and CS while varying the number of top edges displayed.

Figures 10 and 11 show the PK wiring diagrams for OQ, OK, and OV, displaying the top 10 and top 30 edges per pairing, respectively. Although Fig. 10 is sparse, $\mathrm{PK}_{\mathrm{OK}}$ remains large between Previous Token Heads and Induction Heads. It also shows large $\mathrm{PK}_{\mathrm{OQ}}$ scores among Name-Mover-related heads, and confirms that L4H7 acts as a hub. In contrast, Fig. 11 is denser and contains more information, but interpretation becomes harder. Compared with Fig. 6, it also includes more heads that do not belong to any head class in the middle layers. For these reasons, in Section 6 we set the number of displayed edges to 20.

Figures 12, 13, and 14 show the CS wiring diagrams for OQ, OK, and OV, showing the top 10, top 20, and top 30 edges per pairing, respectively. Compared with the PK wiring diagrams, more heads in shallow layers appear, and many edges connect shallow and deep 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 are present under PK, do not appear under CS.

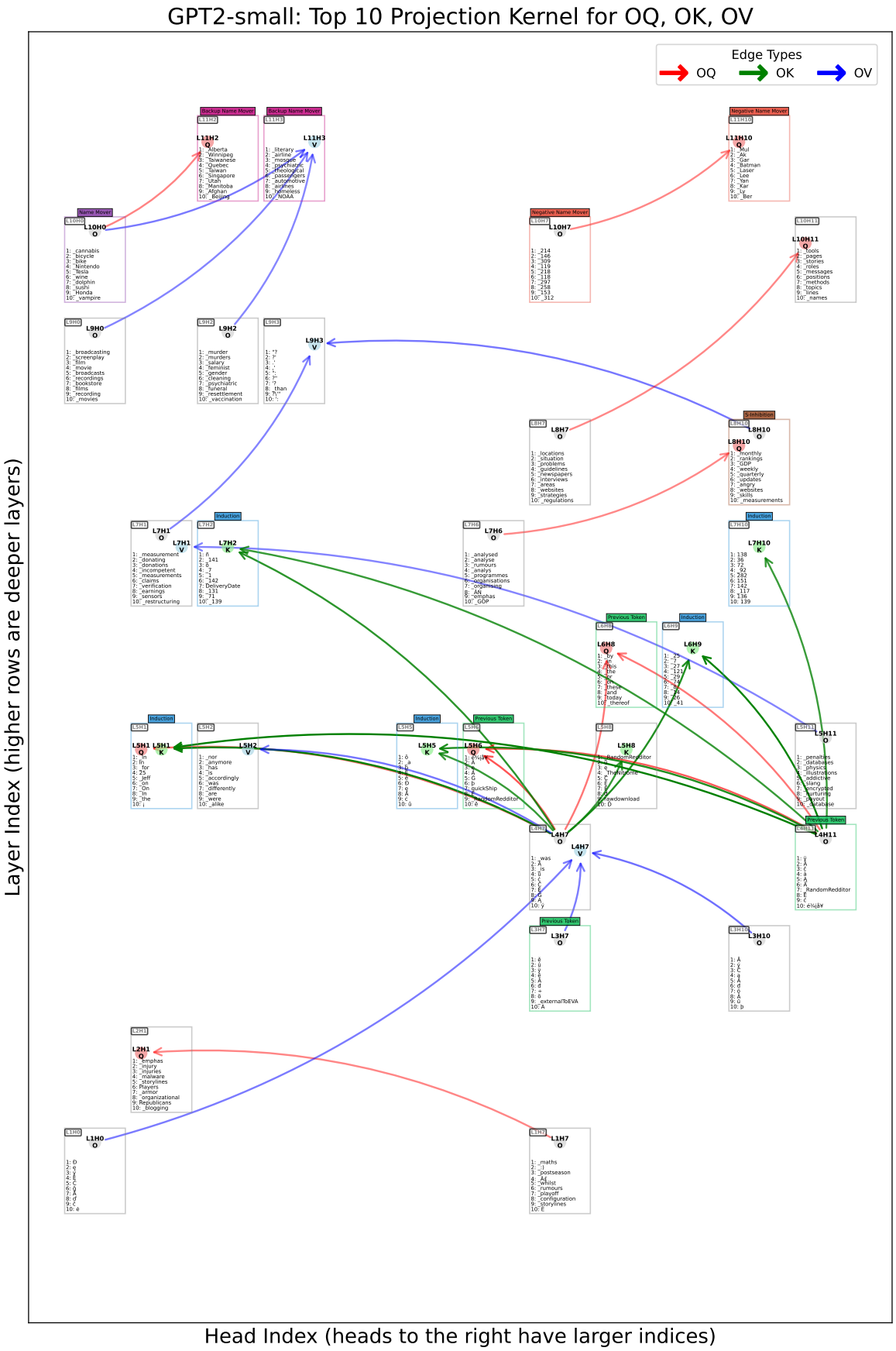


Figure 10: For GPT2-small, we visualize the top 10 scores of $\mathrm{PK}_{\mathrm{OQ}}$, $\mathrm{PK}_{\mathrm{OK}}$, and $\mathrm{PK}_{\mathrm{OV}}$. For each head h, we show the top 10 tokens that represent $\boldsymbol{W}_{\mathrm{O}}^{h}$ using the projection method in Section 4. For readability,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

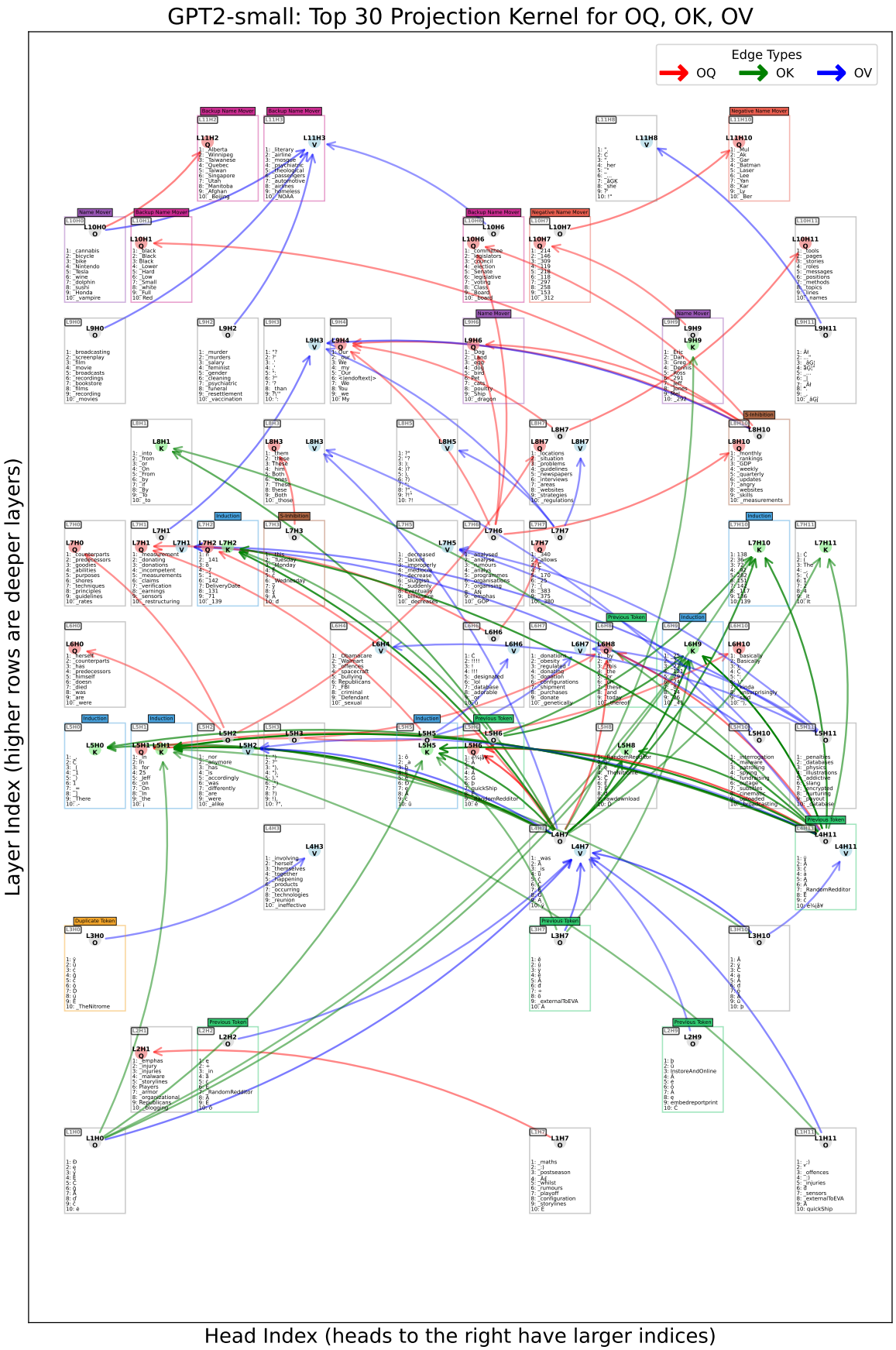


Figure 11: For GPT2-small, we visualize the top 30 scores of $\mathrm{PK}_{\mathrm{OQ}}$, $\mathrm{PK}_{\mathrm{OK}}$, and $\mathrm{PK}_{\mathrm{OV}}$. For each head h, we show the top 10 tokens that represent $\boldsymbol{W}_{\mathrm{O}}^{h}$ using the projection method in Section 4. For readability, we replace Ġ in tokens with _.

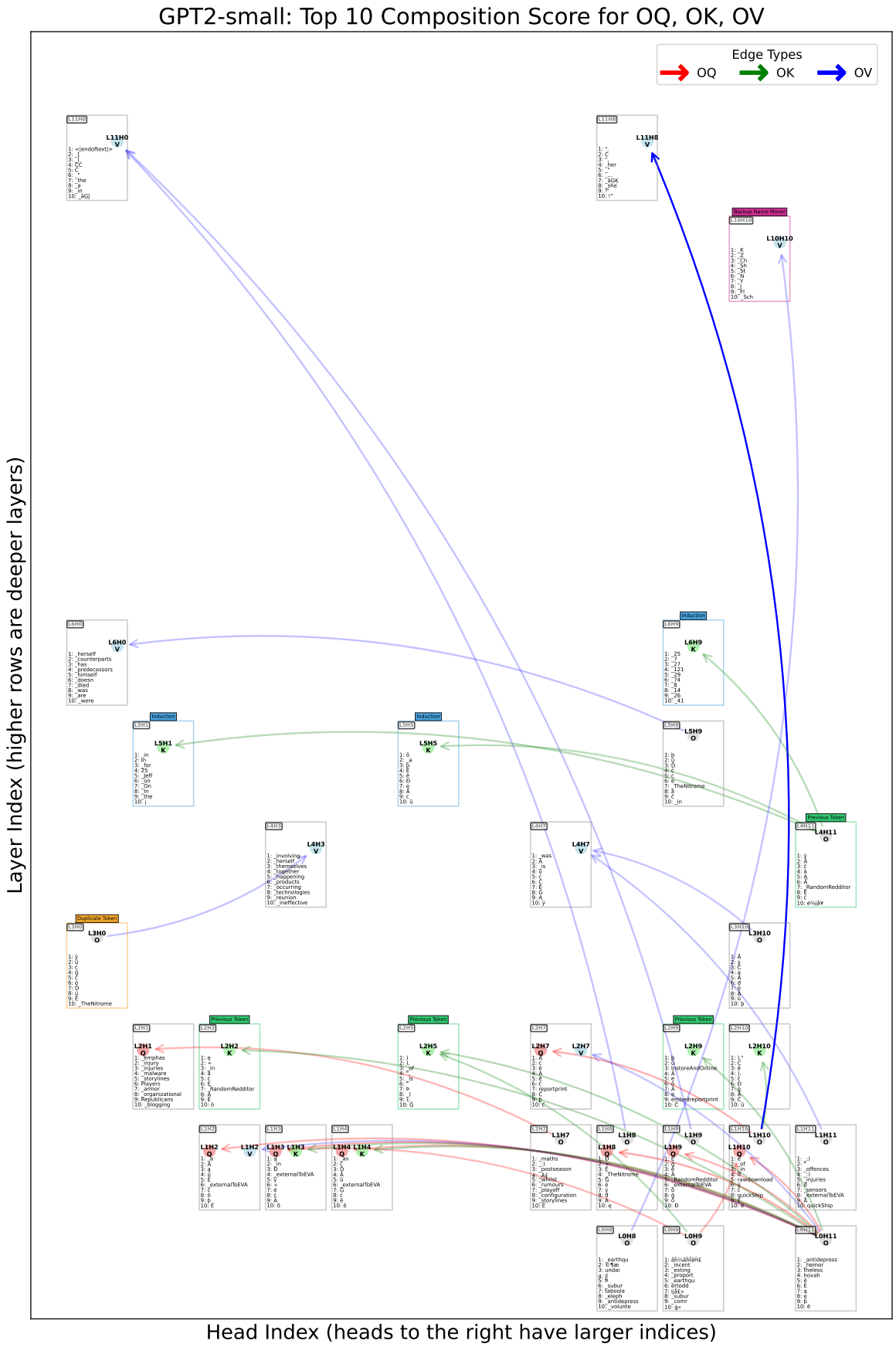


Figure 12: For GPT2-small, we visualize the top 10 scores of $\mathrm{CS}_{\mathrm{OQ}}$, $\mathrm{CS}_{\mathrm{OK}}$, and $\mathrm{CS}_{\mathrm{OV}}$. For each head h, we show the top 10 tokens that represent $\boldsymbol{W}_{\mathrm{O}}^{h}$ using the projection method in Section 4. For readability, we replace $\dot{\mathrm{G}}$ in tokens with _.

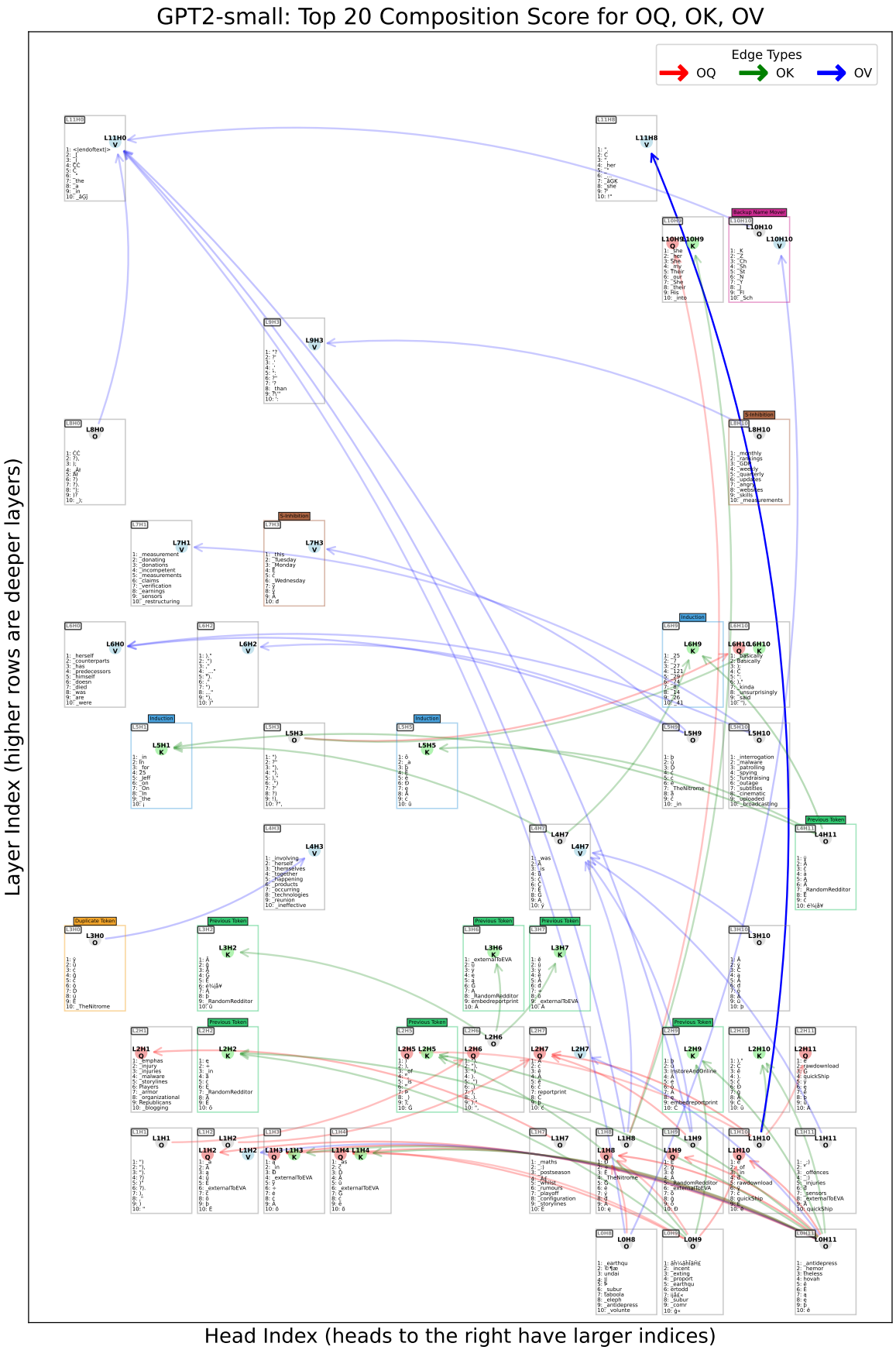


Figure 13: For GPT2-small, we visualize the top 20 scores of CS_{OQ}, CS_{OK}, and CS_{OV}. For each head h, we show the top 10 tokens that represent $\boldsymbol{W}_{\text{O}}^{h}$ using the projection method in Section 4. For readability, we replace Ġ in tokens with _.

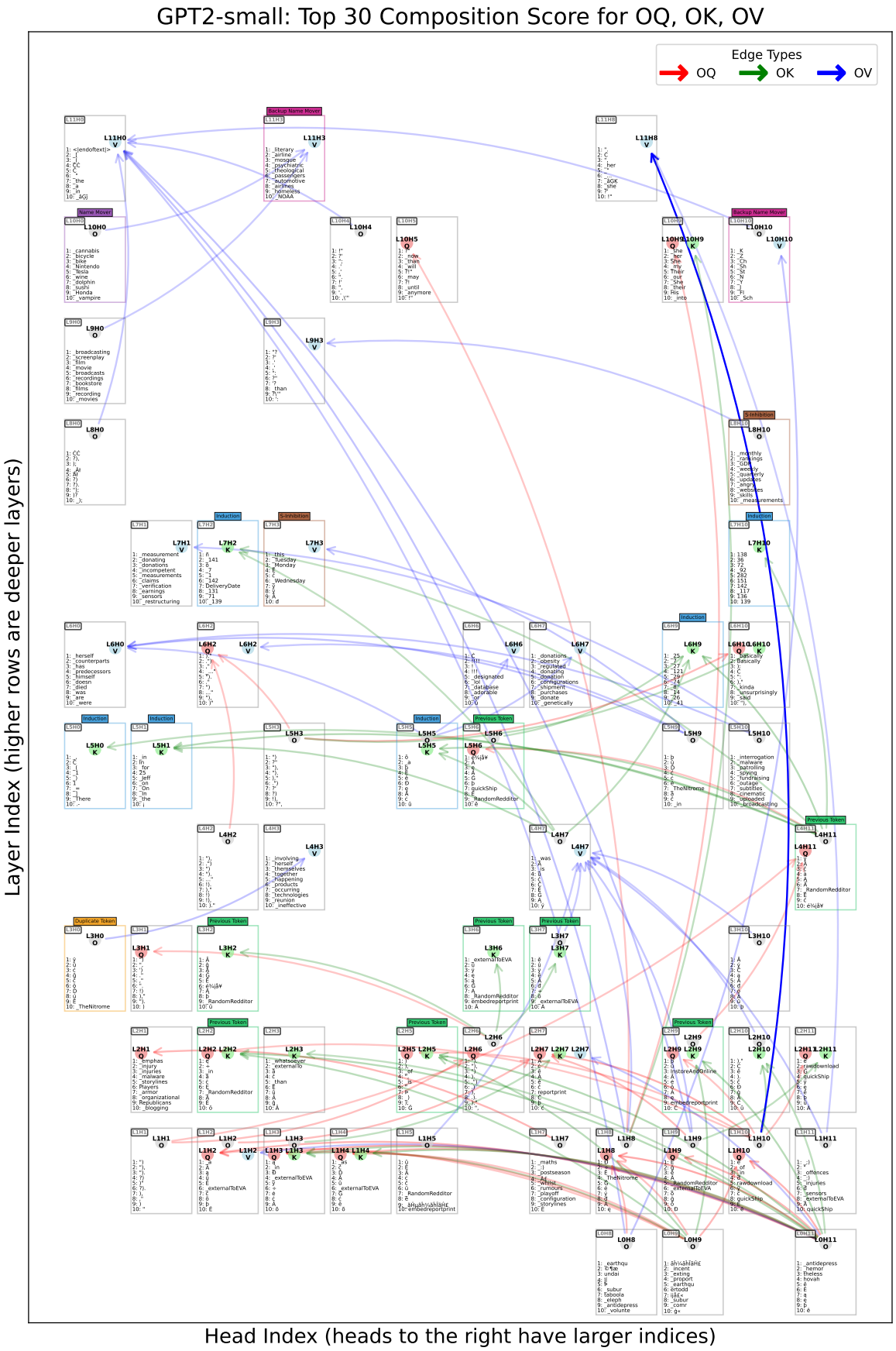


Figure 14: For GPT2-small, we visualize the top 30 scores of $\mathrm{CS}_{\mathrm{OQ}}$, $\mathrm{CS}_{\mathrm{OK}}$, and $\mathrm{CS}_{\mathrm{OV}}$. For each head h, we show the top 10 tokens that represent $\boldsymbol{W}_{\mathrm{O}}^{h}$ using the projection method in Section 4. For readability, we replace Ġ in tokens with _.

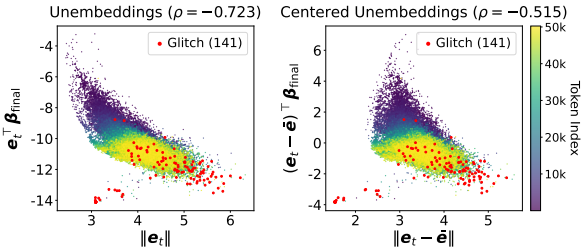


Figure 15: Scatter plots of the norm of unembedding vectors in GPT2-small and the inner product with the bias term β_{final} of the final Layer Normalization in (53), (left) without centering and (right) with centering. Since GPT2 uses byte pair encoding (BPE), the token ID can be viewed as a proxy for frequency, where larger token IDs tend to be less frequent (Hayase et al., 2024). 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.

E Details of Projections of the Unembedding Matrix in Section 4

In Section 4, to interpret heads in GPT2-small, we project preprocessed unembedding vectors onto the subspace $\widetilde{\mathcal{S}}^r$ spanned by the weights $\widetilde{W}^r$ after the final LN layer, 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.

E.1 Tokens with large unembedding norms tend to have low generation probabilities

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

$$\text{LN}_{\text{final}}(\boldsymbol{x}) = \gamma_{\text{final}} \odot \frac{\boldsymbol{x} - \text{Mean}(\boldsymbol{x})\mathbf{1}_d}{\text{Std}(\boldsymbol{x})} + \beta_{\text{final}}, \tag{53}$$

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 $\boldsymbol{e}_t$ and the bias term β_{final} in (53), namely $\boldsymbol{e}_t^\top \beta_{\text{final}}$, as a token specific statistic related to generation probabilities[18] (Nanda and Bloom, 2022). In fact, the logit for token t is computed as the inner product between the unembedding vector $\boldsymbol{e}_t$ and $\text{LN}_{\text{final}}(\boldsymbol{x})$ in (53),

$$\begin{aligned} &\boldsymbol{e}_t^\top \text{LN}_{\text{final}}(\boldsymbol{x}) \\ &= \boldsymbol{e}_t^\top \left(\gamma_{\text{final}} \odot \frac{\boldsymbol{x} - \text{Mean}(\boldsymbol{x})\mathbf{1}_d}{\text{Std}(\boldsymbol{x})} \right) + \boldsymbol{e}_t^\top \beta_{\text{final}} \end{aligned} \tag{54}$$

where the first term on the right-hand side of (54) depends on the final-layer output $\boldsymbol{x}$, whereas the second term $\boldsymbol{e}_t^\top \beta_{\text{final}}$ is an input-independent, token-specific statistic (i.e., a scalar bias term). In particular, tokens with more negative $\boldsymbol{e}_t^\top \beta_{\text{final}}$ tend to have smaller probabilities under the softmax.

To understand the behavior of $\boldsymbol{e}_t^\top \beta_{\text{final}}$, we examine its relationship with the unembedding norm. For comparison, we also consider centering the unembedding vectors using their mean $\bar{\boldsymbol{e}} \in \mathbb{R}^d$. 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 trained[19]. 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 tokens[20].

[18] See https://transformerlensorg.github.io/TransformerLens/generated/demos/Main_Demo.html for details.

[19] We use the list of 141 glitch tokens provided by Rumbelow and Watkins (2023).

[20] Note that some glitch tokens also have small norms.

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

Table 4: For the 144 heads in GPT2-small, the mean and standard deviation of Spearman's ρ between the unembedding norm and the norm of the vector projected onto $\widetilde{\mathcal{S}}^r$ after preprocessing f, reported separately for each weight type. We also report the results when ignoring weight types (All). Without normalization ($f(\boldsymbol{e}_t) = \boldsymbol{e}_t$ or $f(\boldsymbol{e}_t) = \boldsymbol{e}_t - \bar{\boldsymbol{e}}$), the pre and post projection norms show a strong positive correlation. With normalization ($f(\boldsymbol{e}_t) = \boldsymbol{e}_t/\lVert\boldsymbol{e}_t\rVert$ or $f(\boldsymbol{e}_t) = (\boldsymbol{e}_t - \bar{\boldsymbol{e}})/\lVert\boldsymbol{e}_t - \bar{\boldsymbol{e}}\rVert$), we instead observe a weak negative correlation.

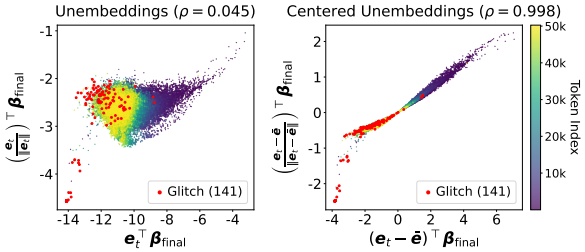


Figure 16: Scatter plots of the inner product with $\boldsymbol{\beta}_{\text{final}}$ before and after normalization, (left) without centering and (right) with centering. As in Fig. 15, we indicate token IDs as a proxy for frequency, where larger token IDs tend to be less frequent, and we label glitch 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.

E.2 Normalization makes projected norms smaller for less interpretable tokens

We consider projecting preprocessed unembedding vectors onto the subspace $\widetilde{\mathcal{S}}^r$ spanned by $\widetilde{\boldsymbol{W}}^r$, where $\widetilde{\boldsymbol{W}}^r$ is obtained by applying LN_{final} to $\boldsymbol{W}^r \in \mathcal{W}$. As candidates for preprocessing f, we consider the identity map, centering, normalization, and normalization after centering. Table 4 reports, for these four choices, the correlation between the norm of the (centered or uncentered) unembedding vector and the norm of its projection onto $\widetilde{\mathcal{S}}^r$. 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 4, we obtain the following. Without normalization, less interpretable tokens that have large unembedding norms and strongly negative inner products with $\boldsymbol{\beta}_{\text{final}}$ 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.

E.3 Why centering before normalization is necessary

Since the inner product between an unembedding vector and $\boldsymbol{\beta}_{\text{final}}$ can be regarded as a token specific statistic related to generation probabilities, regardless of whether we center[21], it is desirable that this inner product remains correlated even after normalizing the unembedding vectors. Figure 16 plots how the inner product with $\boldsymbol{\beta}_{\text{final}}$ 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. The mean cosine similarity between $\boldsymbol{e}_t$ and the mean vector $\bar{\boldsymbol{e}} \in \mathbb{R}^d$, namely $\cos(\boldsymbol{e}_t, \bar{\boldsymbol{e}})$, is 0.518. The mean cosine similarity with $\boldsymbol{\beta}_{\text{final}}$, namely $\cos(\boldsymbol{e}_t, \boldsymbol{\beta}_{\text{final}})$, is -0.225. These results[22] suggest that the unembedding vectors are anisotropic[23] toward the direction of $\bar{\boldsymbol{e}}$, and that this dominant direction is somewhat opposite to $\boldsymbol{\beta}_{\text{final}}$.

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 $\boldsymbol{\beta}_{\text{final}}$. Based on these results and discussion, when interpreting top tokens of attention heads, we use unembedding vectors that are normalized after centering.

[21] The inner product between the centered unembedding vector $\boldsymbol{e}_t - \bar{\boldsymbol{e}}$ and $\boldsymbol{\beta}_{\text{final}}$ equals $\boldsymbol{e}_t^\top \boldsymbol{\beta}_{\text{final}} - \bar{\boldsymbol{e}}^\top \boldsymbol{\beta}_{\text{final}}$, and $\bar{\boldsymbol{e}}^\top \boldsymbol{\beta}_{\text{final}}$ 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 K.1 for details.

[22] For reference, $\cos(\bar{\boldsymbol{e}}, \boldsymbol{\beta}_{\text{final}}) = -0.431$.

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

F Details of the similarity metric evaluation in Section 7

In this section, we provide details of the similarity metric evaluation in Section 7. We first describe the baseline methods other than PK and CS in Section 7.1. We then present additional details on classwise classification performance for QQ, KK, VV, and OO in Section 7.3.

F.1 Details of baseline methods

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

F.1.1 Simple-CS

We explain Simple-CS using the example of the weights associated with an output and a query. In $\mathrm{CS}_{\mathrm{OQ}}$ in (3), we compute $\mathrm{CS}({\boldsymbol{W}_{\mathrm{QK}}^{h'}}^\top, \boldsymbol{W}_{\mathrm{OV}}^h)$ using $\boldsymbol{W}_{\mathrm{OV}}^h, {\boldsymbol{W}_{\mathrm{QK}}^{h'}}^\top \in \mathbb{R}^{d\times d}$. Thus, even when measuring similarity between weights associated with an output and a query, CS uses the OV matrix $\boldsymbol{W}_{\mathrm{OV}}^h$ as the output side and the QK matrix ${\boldsymbol{W}_{\mathrm{QK}}^{h'}}^\top$ as the query side. As a result, $\boldsymbol{W}_{\mathrm{OV}}^h$ in (2) includes the value weight matrix $\boldsymbol{W}_{\mathrm{V}}^h$, and $\boldsymbol{W}_{\mathrm{QK}}^{h'}$ in (1) includes the key weight matrix $\boldsymbol{W}_{\mathrm{K}}^{h'}$.

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

$$\text{Simple-CS}(r, r') = \mathrm{CS}(\boldsymbol{W}^{r'}, \boldsymbol{W}^r). \tag{55}$$

F.1.2 Linear CKA

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

$$\boldsymbol{w}_{\mathrm{mean}}^r = \frac{1}{d_{\mathrm{head}}} \sum_{i=1}^{d_{\mathrm{head}}} \boldsymbol{w}_i^r \in \mathbb{R}^d. \tag{56}$$

We then define the centered matrix $\boldsymbol{W}_c^r$ as[24]

$$\boldsymbol{W}_c^r = \boldsymbol{W}^r - \boldsymbol{w}_{\mathrm{mean}}^r {\boldsymbol{1}_{d_{\mathrm{head}}}}^\top \in \mathbb{R}^{d\times d_{\mathrm{head}}}. \tag{57}$$

[24]Using the centering matrix $\boldsymbol{C}_{d_{\mathrm{head}}}$ in (110), we have $\boldsymbol{W}_c^r = \boldsymbol{W}^r \boldsymbol{C}_{d_{\mathrm{head}}}$.

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

$$\begin{aligned}&\text{Linear-CKA}(r, r') \\ &= \frac{\left\|\boldsymbol{W}_c^r {\boldsymbol{W}_c^{r'}}^\top\right\|_{\mathrm{F}}^2}{\left\|\boldsymbol{W}_c^r {\boldsymbol{W}_c^r}^\top\right\|_{\mathrm{F}} \left\|\boldsymbol{W}_c^{r'} {\boldsymbol{W}_c^{r'}}^\top\right\|_{\mathrm{F}}}.\end{aligned} \tag{58}$$

To verify that $0 \le \text{Linear-CKA}(r, r') \le 1$, we note the following inequality[25]

$$\begin{aligned}&\left\|\boldsymbol{W}_c^r {\boldsymbol{W}_c^{r'}}^\top\right\|_{\mathrm{F}}^2 \\ &= \mathrm{tr}\left(\left(\boldsymbol{W}_c^r {\boldsymbol{W}_c^{r'}}^\top\right)^\top \boldsymbol{W}_c^r {\boldsymbol{W}_c^{r'}}^\top\right) \\ &= \mathrm{tr}\left(\boldsymbol{W}_c^{r'} {\boldsymbol{W}_c^r}^\top \boldsymbol{W}_c^r {\boldsymbol{W}_c^{r'}}^\top\right) \\ &= \mathrm{tr}\left({\boldsymbol{W}_c^r}^\top \boldsymbol{W}_c^r {\boldsymbol{W}_c^{r'}}^\top \boldsymbol{W}_c^{r'}\right) \\ &= \mathrm{tr}\left(\left({\boldsymbol{W}_c^r}^\top \boldsymbol{W}_c^r\right)^\top {\boldsymbol{W}_c^{r'}}^\top \boldsymbol{W}_c^{r'}\right) \\ &= \langle {\boldsymbol{W}_c^r}^\top \boldsymbol{W}_c^r, {\boldsymbol{W}_c^{r'}}^\top \boldsymbol{W}_c^{r'}\rangle_{\mathrm{F}} \\ &\le \|{\boldsymbol{W}_c^r}^\top \boldsymbol{W}_c^r\|_{\mathrm{F}} \|{\boldsymbol{W}_c^{r'}}^\top \boldsymbol{W}_c^{r'}\|_{\mathrm{F}} \\ &= \|\boldsymbol{W}_c^r {\boldsymbol{W}_c^r}^\top\|_{\mathrm{F}} \|\boldsymbol{W}_c^{r'} {\boldsymbol{W}_c^{r'}}^\top\|_{\mathrm{F}},\end{aligned} \tag{59}$$

where the inequality follows from the Cauchy–Schwarz inequality for the Frobenius inner product, and equality holds if and only if ${\boldsymbol{W}_c^r}^\top \boldsymbol{W}_c^r = \lambda {\boldsymbol{W}_c^{r'}}^\top \boldsymbol{W}_c^{r'}$ for some $\lambda \ge 0$. Therefore, when $\|\boldsymbol{W}_c^r {\boldsymbol{W}_c^r}^\top\|_{\mathrm{F}} \|\boldsymbol{W}_c^{r'} {\boldsymbol{W}_c^{r'}}^\top\|_{\mathrm{F}} \ne 0$, we have

$$0 \le \text{Linear-CKA}(r, r') \le 1. \tag{60}$$

F.1.3 Procrustes Similarity

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

$$\underset{\substack{\boldsymbol{\Phi}\in\mathbb{R}^{d\times d}, \\ \boldsymbol{\Phi}^\top\boldsymbol{\Phi}=\boldsymbol{\Phi}\boldsymbol{\Phi}^\top=\boldsymbol{I}_d}}{\text{minimize}} \quad \|\boldsymbol{\Phi}\boldsymbol{W}^r - \boldsymbol{W}^{r'}\|_{\mathrm{F}}. \tag{61}$$

[25]We use

$$\begin{aligned}\|{\boldsymbol{W}_c^r}^\top \boldsymbol{W}_c^r\|_{\mathrm{F}} &= \sqrt{\mathrm{tr}\left(\left({\boldsymbol{W}_c^r}^\top \boldsymbol{W}_c^r\right)^\top {\boldsymbol{W}_c^r}^\top \boldsymbol{W}_c^r\right)} \\ &= \sqrt{\mathrm{tr}\left({\boldsymbol{W}_c^r}^\top \boldsymbol{W}_c^r {\boldsymbol{W}_c^r}^\top \boldsymbol{W}_c^r\right)} \\ &= \sqrt{\mathrm{tr}\left(\boldsymbol{W}_c^r {\boldsymbol{W}_c^r}^\top \boldsymbol{W}_c^r {\boldsymbol{W}_c^r}^\top\right)} \\ &= \sqrt{\mathrm{tr}\left(\left(\boldsymbol{W}_c^r {\boldsymbol{W}_c^r}^\top\right)^\top \boldsymbol{W}_c^r {\boldsymbol{W}_c^r}^\top\right)} \\ &= \|\boldsymbol{W}_c^r {\boldsymbol{W}_c^r}^\top\|_{\mathrm{F}}.\end{aligned}$$

Similarly, $\|{\boldsymbol{W}_c^{r'}}^\top \boldsymbol{W}_c^{r'}\|_{\mathrm{F}} = \|\boldsymbol{W}_c^{r'} {\boldsymbol{W}_c^{r'}}^\top\|_{\mathrm{F}}$.

This problem admits an analytic solution. Let $\boldsymbol{W}^{r'}\boldsymbol{W}^{r\top} = \boldsymbol{U}^{r,r'}\boldsymbol{\Sigma}^{r,r'}\boldsymbol{V}^{r,r'\top}$ be the SVD, where $\boldsymbol{U}^{r,r'}, \boldsymbol{V}^{r,r'} \in \mathbb{R}^{d\times d}$ are orthogonal and $\boldsymbol{\Sigma}^{r,r'} \in \mathbb{R}^{d\times d}$ is diagonal with nonnegative entries. Then (61) is minimized by $\boldsymbol{\Phi}^* = \boldsymbol{U}^{r,r'}\boldsymbol{V}^{r,r'\top} \in \mathbb{R}^{d\times d}$ (Schönemann, 1966). We refer to $\|\boldsymbol{\Phi}^*\boldsymbol{W}^r - \boldsymbol{W}^{r'}\|_{\mathrm{F}}$ as the Procrustes distance (Rohlf, 1999).

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

$$\begin{aligned}
&\|\boldsymbol{\Phi}^*\boldsymbol{W}^r - \boldsymbol{W}^{r'}\|_{\mathrm{F}}^2 \\
&= \mathrm{tr}\left(\left(\boldsymbol{\Phi}^*\boldsymbol{W}^r - \boldsymbol{W}^{r'}\right)^\top \left(\boldsymbol{\Phi}^*\boldsymbol{W}^r - \boldsymbol{W}^{r'}\right)\right) \\
&= \mathrm{tr}\left((\boldsymbol{\Phi}^*\boldsymbol{W}^r)^\top \boldsymbol{\Phi}^*\boldsymbol{W}^r\right) + \mathrm{tr}\left(\boldsymbol{W}^{r'\top}\boldsymbol{W}^{r'}\right) \\
&\quad - 2\mathrm{tr}\left((\boldsymbol{\Phi}^*\boldsymbol{W}^r)^\top \boldsymbol{W}^{r'}\right) \\
&= \|\boldsymbol{\Phi}^*\boldsymbol{W}^r\|_{\mathrm{F}}^2 + \|\boldsymbol{W}^{r'}\|_{\mathrm{F}}^2 \\
&\quad - 2\mathrm{tr}\left((\boldsymbol{\Phi}^*\boldsymbol{W}^r)^\top \boldsymbol{W}^{r'}\right) \\
&\le \|\boldsymbol{\Phi}^*\boldsymbol{W}^r\|_{\mathrm{F}}^2 + \|\boldsymbol{W}^{r'}\|_{\mathrm{F}}^2, \qquad (62)
\end{aligned}$$

where the inequality uses

$$\begin{aligned}
\mathrm{tr}\left(\boldsymbol{W}^{r'\top}\boldsymbol{\Phi}^*\boldsymbol{W}^r\right) &= \mathrm{tr}\left(\left(\boldsymbol{W}^{r'\top}\boldsymbol{\Phi}^*\boldsymbol{W}^r\right)^\top\right) \\
&= \mathrm{tr}\left((\boldsymbol{\Phi}^*\boldsymbol{W}^r)^\top \boldsymbol{W}^{r'}\right) \qquad (63)
\end{aligned}$$

and

$$\begin{aligned}
&\mathrm{tr}\left((\boldsymbol{\Phi}^*\boldsymbol{W}^r)^\top \boldsymbol{W}^{r'}\right) \\
&= \mathrm{tr}\left(\boldsymbol{W}^{r\top}\boldsymbol{\Phi}^{*\top}\boldsymbol{W}^{r'}\right) = \mathrm{tr}\left(\boldsymbol{W}^{r'}\boldsymbol{W}^{r\top}\boldsymbol{\Phi}^{*\top}\right) \\
&= \mathrm{tr}\left(\boldsymbol{U}^{r,r'}\boldsymbol{\Sigma}^{r,r'}\boldsymbol{V}^{r,r'\top}\left(\boldsymbol{U}^{r,r'}\boldsymbol{V}^{r,r'\top}\right)^\top\right) \\
&= \mathrm{tr}\left(\boldsymbol{U}^{r,r'}\boldsymbol{\Sigma}^{r,r'}\boldsymbol{V}^{r,r'\top}\boldsymbol{V}^{r,r'}\boldsymbol{U}^{r,r'\top}\right) \\
&= \mathrm{tr}\left(\boldsymbol{\Sigma}^{r,r'}\right) \ \ge\ 0, \qquad (64)
\end{aligned}$$

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

$$0 \le \frac{\|\boldsymbol{\Phi}^*\boldsymbol{W}^r - \boldsymbol{W}^{r'}\|_{\mathrm{F}}^2}{\|\boldsymbol{\Phi}^*\boldsymbol{W}^r\|_{\mathrm{F}}^2 + \|\boldsymbol{W}^{r'}\|_{\mathrm{F}}^2} \le 1. \qquad (65)$$

We then define Procrustes Similarity as

$$\begin{aligned}
&\text{Procrustes-Similarity}(r, r') \\
&= 1 - \frac{\|\boldsymbol{\Phi}^*\boldsymbol{W}^r - \boldsymbol{W}^{r'}\|_{\mathrm{F}}^2}{\|\boldsymbol{\Phi}^*\boldsymbol{W}^r\|_{\mathrm{F}}^2 + \|\boldsymbol{W}^{r'}\|_{\mathrm{F}}^2}. \qquad (66)
\end{aligned}$$

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

Table 5 reports classwise averages of PR-AUC and ROC-AUC used to compute Table 2. We find that PK achieves high classification performance for Duplicate Token, Induction, Negative Name Mover, Backup Name Mover, and S-Inhibition Heads. These results indicate that PK performs well for many head classes. In contrast, CS outperforms PK for classifying Previous Token Heads, and Procrustes Similarity outperforms PK for classifying Name Mover Heads.

In addition, Table 6 reports PR-AUC and ROC-AUC for each head class, computed separately for the weight pairing types QQ, KK, VV, and OO, which are used in Table 2 and Table 5. These results show that classwise classification performance depends strongly on the weight pairing type as well as the method. For Previous Token Heads, PK underperforms CS, which is the best method, for QQ, KK, and VV, whereas PK attains the best performance for OO. For Name Mover Heads, PK underperforms Procrustes Similarity, which is the best method, 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.

G Details of head scores and head class annotations

In this section, we describe how we compute head scores using TransformerLens[26] (Nanda and Bloom, 2022) following Wang et al. (2023), and provide details of our head class annotations.

G.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 sen-

[26] https://github.com/TransformerLensOrg/TransformerLens/blob/main/demos/Exploratory_Analysis_Demo.ipynb

	Method	Dup (3)	Prev (10)	Ind (6)	NM (3)	N-NM (2)	B-NM (8)	S-Inh (4)	Avg.
PR-AUC	PK	**0.003**	0.019	**0.229**	0.024	0.008	**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	0.027	0.048	0.008	0.002	0.021	0.003	0.016
	Linear CKA	0.000	0.004	0.011	0.117	0.003	0.014	0.001	0.021
	Proc. Sim.	0.000	0.007	0.016	**0.138**	**0.071**	0.029	0.002	0.038
ROC-AUC	PK	0.448	0.817	**0.879**	0.881	**0.982**	0.733	**0.919**	**0.809**
	CS	0.477	**0.887**	0.821	0.810	0.932	0.712	0.855	0.785
	Simple-CS	0.474	0.853	0.845	0.819	0.943	0.733	0.874	0.792
	Linear CKA	**0.538**	0.407	0.729	0.927	0.885	0.754	0.765	0.715
	Proc. Sim.	0.402	0.612	0.855	**0.974**	0.979	**0.892**	0.794	0.787

Table 5: Classwise averages of PR-AUC and ROC-AUC, averaged over the weight pairing types QQ, KK, VV, and OO, and used to compute Table 2. The maximum is shown in bold and the second largest is underlined. These values are computed over directed head pairs in GPT2-small whose source is in an earlier or the same layer and whose target is in the same or a later layer (10,296 pairs, excluding identical head pairs). See Table 6 for results computed separately for the weight pairing types QQ, KK, VV, and OO. 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.

tence[27], 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 7a lists the top 10 Identity Heads. Except for L4H7, which we discussed in Section 8, these heads mainly appear in shallow 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. 17. Duplicate Token Heads mainly appear in shallow layers, whereas Previous Token Heads mainly appear in middle layers, and Induction Heads span middle to deep 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 7b, 7c, and 7d 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.

[27]TransformerLens (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.

G.2 Head class annotations

We adopt the IOI task head class annotations from Wang et al. (2023, Fig. 17). Based on Appendix G.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 7b, we replace L0H10, which does not appear in the table, with L0H5, which attains the largest score[28].

As shown in Tables 7c and 7d, 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 8 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 8. 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 8.

[28]In 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.

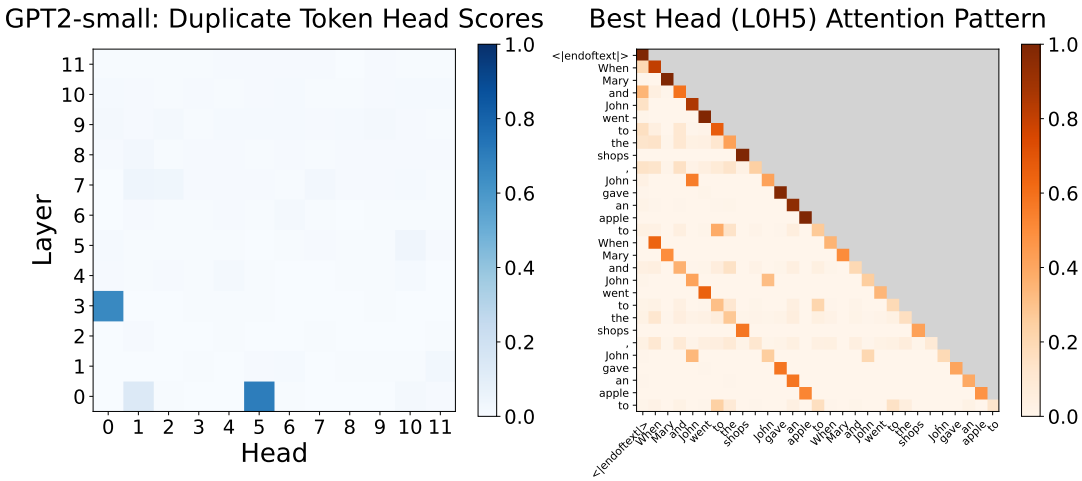
	Method	Dup (3)	Prev (10)	Ind (6)	NM (3)	N-NM (2)	B-NM (8)	S-Inh (4)	Avg
QQ	PK	**0.006**	0.012	**0.199**	**0.055**	0.001	**0.090**	**0.009**	**0.053**
	CS	0.001	**0.052**	0.023	0.005	0.000	0.020	0.003	0.015
	Simple-CS	0.002	0.021	0.045	0.016	0.001	0.041	0.005	0.019
	Linear CKA	0.001	0.003	0.036	0.003	0.001	0.006	0.001	0.007
	Procrustes	0.001	0.003	0.049	0.019	**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	0.002	0.055	0.137	0.011	0.002	0.031	0.004	0.034
	Linear CKA	0.001	0.003	0.002	0.001	0.008	0.003	0.003	0.003
	Procrustes	0.000	0.004	0.003	0.002	**0.250**	0.012	0.002	0.039
VV	PK	0.000	0.018	0.009	0.008	0.009	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	0.000	0.021	**0.010**	0.006	0.006	0.011	0.002	0.008
	Linear CKA	0.000	0.004	0.001	0.034	0.004	0.036	0.001	0.011
	Procrustes	0.000	0.007	0.003	**0.071**	**0.022**	**0.046**	0.002	**0.021**
OO	PK	**0.002**	**0.038**	**0.024**	0.002	**0.019**	0.002	**0.004**	0.013
	CS	0.000	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	0.430	0.000	0.010	0.001	0.064
	Procrustes	0.000	0.014	0.010	**0.461**	0.001	**0.018**	0.002	**0.072**

(a) PR-AUC

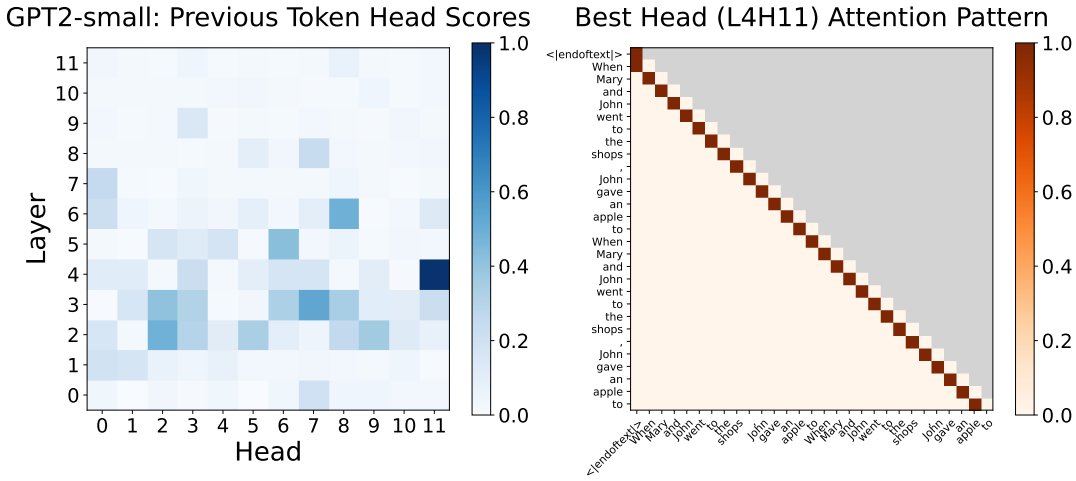
	Method	Dup (3)	Prev (10)	Ind (6)	NM (3)	N-NM (2)	B-NM (8)	S-Inh (4)	Avg
QQ	PK	0.453	0.748	0.967	**0.997**	0.952	0.922	**0.953**	0.856
	CS	0.416	**0.898**	0.898	0.973	0.902	0.895	0.911	0.842
	Simple-CS	0.454	0.842	0.943	0.992	0.921	0.921	0.931	**0.858**
	Linear CKA	**0.800**	0.265	0.971	0.923	0.945	0.779	0.730	0.773
	Procrustes	0.538	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	0.855
	Simple-CS	0.365	0.897	0.991	0.987	0.979	0.899	0.924	**0.863**
	Linear CKA	**0.739**	0.252	0.702	0.791	0.994	0.643	0.893	0.716
	Procrustes	0.394	0.468	0.806	0.908	**1.000**	0.895	0.730	0.743
VV	PK	0.418	0.874	0.709	0.821	0.994	0.753	0.850	0.774
	CS	**0.531**	**0.889**	0.725	0.777	0.981	0.725	0.829	0.780
	Simple-CS	0.520	0.885	0.734	0.792	0.992	0.749	0.847	**0.788**
	Linear CKA	0.287	0.458	0.454	0.996	0.989	**0.932**	0.745	0.694
	Procrustes	0.307	0.698	**0.746**	**0.998**	**0.998**	0.903	**0.856**	0.786
OO	PK	0.562	**0.914**	0.842	0.712	**0.997**	0.332	**0.910**	0.753
	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	0.999	0.611	0.660	0.693	0.676
	Procrustes	0.369	0.851	**0.893**	**1.000**	0.922	**0.802**	0.877	**0.816**

(b) ROC-AUC

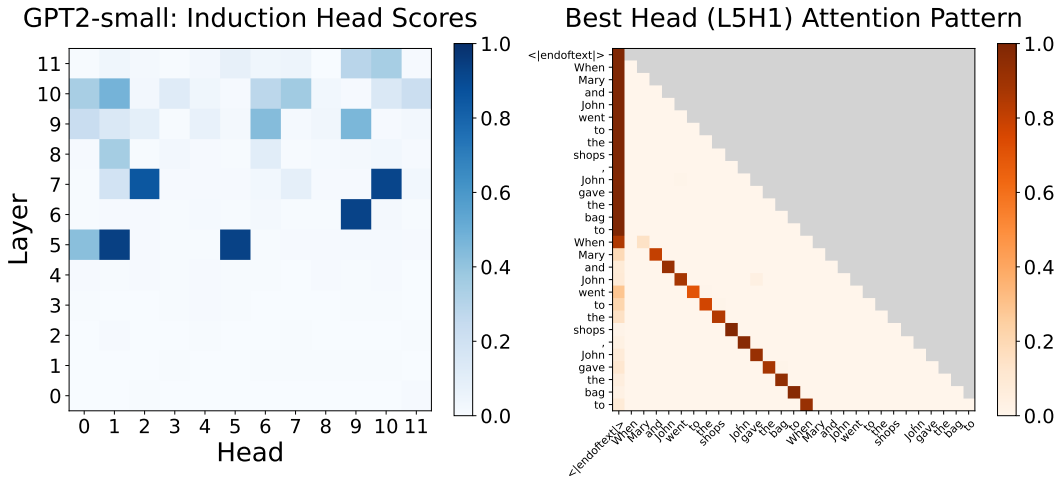
Table 6: PR-AUC and ROC-AUC for each head class, computed separately for the weight-pairing types QQ, KK, VV, and OO, and used to compute Table 2 and Table 5. The maximum is shown in bold and the second largest is underlined. These values are computed over directed head pairs in GPT2-small whose source is in an earlier or the same layer and whose target is in the same or a later layer ($10{,}296$ pairs, excluding identical head pairs). 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.



(a) Duplicate Token Heads



(b) Previous Token Heads



(c) Induction Heads

Figure 17: 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
L0H3	0.966
L0H1	0.918
L1H11	0.900
L0H4	0.826
L0H5	0.639
L4H7	0.358
L0H10	0.235
L3H2	0.197
L3H6	0.192
L3H0	0.163

(a) Identity

Head	Score
L0H5	0.703
L3H0	0.660
L0H1	0.133
L7H1	0.045
L5H10	0.042
L7H2	0.039
L1H11	0.030
L8H1	0.026
L7H7	0.025
L9H0	0.023

(b) Duplicate Token

Head	Score
L4H11	0.988
L3H7	0.533
L6H8	0.490
L2H2	0.485
L5H6	0.422
L3H2	0.408
L2H9	0.362
L3H8	0.344
L2H5	0.339
L3H6	0.330

(c) Previous Token

Head	Score
L5H1	0.936
L5H5	0.927
L6H9	0.925
L7H10	0.914
L7H2	0.849
L10H1	0.474
L9H9	0.458
L9H6	0.437
L5H0	0.425
L10H7	0.367

(d) Induction

Table 7: For Identity, Duplicate Token, Previous Token, and Induction Heads, we list the top 10 heads and their head scores. See Appendix G.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 8: 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. See Appendix G.2 for details.

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

In Fig. 4 in Section 3.5, using the `attn-only-2l` 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 it[29]. 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.

[29] Elhage 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.

I Details of inlet and outlet scores

Motivated by the wiring diagram based on PK in Fig. 6, where L4H7 acts as a hub, we aim to detect such hub heads. To this end, we decompose the hub role into inlet and outlet, and propose the inlet score and outlet score to identify heads that play each role[30]. This section describes the definitions and results.

First, for each of OQ, OK, and OV, we identify inlet heads by quantifying how strongly a target head receives the maximum PK from source heads in earlier layers. For a head h in a layer after layer 0, and for $\omega \in \{\text{"Q"}, \text{"K"}, \text{"V"}\}$, we define the PK

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

inlet score as

$$\text{PK-Inlet}_{\mathrm{O}\,\omega}(h) = \sum_{h' \in \bigcup_{\ell' < \ell_h} \mathcal{H}_{\ell'}} \mathrm{PK}((h', \mathrm{O}), (h, \omega)) \, \mathbb{I}\Bigg(\mathrm{PK}((h', \mathrm{O}), (h, \omega)) = \max_{h'' \in \bigcup_{\ell_{h'} < \ell''} \mathcal{H}_{\ell''}} \mathrm{PK}((h', \mathrm{O}), (h'', \omega)) \Bigg), \quad (67)$$

where

$$\mathbb{I}(\text{condition}) := \begin{cases} 1 & \text{if the condition holds,} \\ 0 & \text{otherwise.} \end{cases} \quad (68)$$

and ℓ_h denotes the layer index of head h, and $\mathcal{H}_\ell$ denotes the set of heads in layer ℓ. 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 $\mathrm{PK}((h', \mathrm{O}), (h'', \omega))$ for all target heads h'' in later layers $\ell_{h'} < \ell''$. If $\mathrm{PK}((h', \mathrm{O}), (h, \omega))$ 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 how strongly a source head attains the maximum PK to target heads in later layers. For a head h in a layer before the final layer, and for $\omega \in \{\text{“Q”}, \text{“K”}, \text{“V”}\}$, we define the PK outlet score as

$$\text{PK-Outlet}_{\mathrm{O}\,\omega}(h) = \sum_{h' \in \bigcup_{\ell_h < \ell'} \mathcal{H}_{\ell'}} \mathrm{PK}((h, \mathrm{O}), (h', \omega)) \, \mathbb{I}\Bigg(\mathrm{PK}((h, \mathrm{O}), (h', \omega)) = \max_{h'' \in \bigcup_{\ell'' < \ell_{h'}} \mathcal{H}_{\ell''}} \mathrm{PK}((h'', \mathrm{O}), (h', \omega)) \Bigg) \quad (69)$$

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 $\mathrm{PK}((h'', \mathrm{O}), (h', \omega))$ for all source heads h'' in earlier layers $\ell'' < \ell_{h'}$. If $\mathrm{PK}((h, \mathrm{O}), (h', \omega))$ 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 9 reports the top 10 heads and their scores for OQ, OK, and OV based on the inlet and outlet scores in (67) and (69). 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.

J Approximate distribution of the projection kernel 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 $\boldsymbol{U}_A, \boldsymbol{U}_B \in \mathcal{V}_m(\mathbb{R}^d)$ from the uniform distribution[31] on the Stiefel manifold of m orthonormal vectors in $\mathbb{R}^d$,

$$\mathcal{V}_m(\mathbb{R}^d) := \{\boldsymbol{U} \in \mathbb{R}^{d \times m} : \boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}_m\}, \quad (70)$$

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

$$\mathcal{O}(\mathbb{R}^d) = \{\boldsymbol{R} \in \mathbb{R}^{d \times d} : \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{I}_d\}. \quad (71)$$

J.1 Simplifying the setting using rotation invariance of the projection kernel

By the rotation invariance of PK (see Appendix C.5), letting $\boldsymbol{e}_i^{\mathrm{std}} \in \mathbb{R}^d$ denote the i-th stan-

[31] Following 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 $\boldsymbol{W} \in \mathbb{R}^{d \times m}$ with i.i.d. entries 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 $\boldsymbol{U}$ as a sample from the uniform distribution.

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 9: For OQ, OK, and OV, we report (a) the top 10 heads and their inlet scores in (67) and (b) the top 10 heads and their outlet scores in (69). When a head belongs to the head classes in Table 8 or to the top 10 heads by Identity Head score in Table 7a, 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.

dard basis vector, we may fix

$$\boldsymbol{U}_A = (\boldsymbol{e}_1^{\text{std}}, \ldots, \boldsymbol{e}_m^{\text{std}}) = \begin{pmatrix} \boldsymbol{I}_m \\ \boldsymbol{0} \end{pmatrix} \in \mathcal{V}_m(\mathbb{R}^d), \tag{72}$$

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

Next, we sample a random orthogonal matrix $\boldsymbol{R} \in \mathcal{O}(\mathbb{R}^d)$ from the uniform distribution on $\mathcal{O}(\mathbb{R}^d)$ and define[32]

$$\boldsymbol{U}_B = \boldsymbol{R}\boldsymbol{U}_A = \boldsymbol{R}_{[:,1:m]}, \tag{73}$$

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

Since $\boldsymbol{R}$ is random, the PK value between the subspaces $\mathcal{S}_A$ and $\mathcal{S}_B$ spanned by $\boldsymbol{U}_A$ and $\boldsymbol{U}_B$ is a random variable. Letting this random variable be Z, from (47), (72), and (73), we obtain

$$\begin{aligned} Z = \mathrm{PK}(\mathcal{S}_A, \mathcal{S}_B) &= \left\| \boldsymbol{U}_A^\top \boldsymbol{U}_B \right\|_{\mathrm{F}}^2 \\ &= \left\| \boldsymbol{U}_A^\top \boldsymbol{R} \boldsymbol{U}_A \right\|_{\mathrm{F}}^2 = \left\| \boldsymbol{R}_{[1:m],\,[1:m]} \right\|_{\mathrm{F}}^2 \\ &= \sum_{i=1}^{m} \sum_{j=1}^{m} R_{ij}^2, \end{aligned} \tag{74}$$

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

J.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 $\boldsymbol{R}$ sampled in Appendix J.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

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

2.6) showed that[33]

$$\sqrt{d}\, R_{ij} \xrightarrow{\mathcal{D}} \mathcal{N}(0,1) \quad (d \to \infty). \tag{75}$$

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

$$\mathcal{N}\left(0,\, \frac{1}{d}\right). \tag{76}$$

We therefore approximate R_{ij} by independent random variables

$$X_{ij} \sim \mathcal{N}\left(0,\, \frac{1}{d}\right). \tag{77}$$

For $d = 768$ and $m = 64$, which match the dimensions of attention weights in GPT2-small, sampling $\boldsymbol{U}_B = \boldsymbol{R}_{[:,1:m]} \in \mathcal{V}_m(\mathbb{R}^d)$ and overlaying the density of $\mathcal{N}(0, 1/d)$ on the histogram of R_{ij} shows a good fit (see Fig. 18a)[34].

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 distribution[35] with shape $k = 1/2$ and scale $\theta = 2/d$,

$$X_{ij}^2 \sim \text{Gamma}\left(\frac{1}{2}, \frac{2}{d}\right). \tag{78}$$

Similarly, for $d = 768$ and $m = 64$, sampling $\boldsymbol{U}_B = \boldsymbol{R}_{[:,1:m]} \in \mathcal{V}_m(\mathbb{R}^d)$ and overlaying the density of Gamma$(1/2,\, 2/d)$ on the histogram of R_{ij}^2 shows a good fit (see Fig. 18b). 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

$$\mathbb{E}[X_{ij}^2] = k\theta = \frac{1}{d}, \tag{79}$$

$$\text{Var}(X_{ij}^2) = k\theta^2 = \frac{2}{d^2}. \tag{80}$$

[33]Here, $\xrightarrow{\mathcal{D}}$ 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 $F_{X_d}(\tau) \to F_X(\tau)$ as $d \to \infty$ for every τ at which F_X is continuous, where $F_Y(\tau) := \mathbb{P}(Y \le \tau)$ (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 $\mathcal{L}$.

[34]This $\mathcal{N}(0, 1/d)$ 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).

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

Using the i.i.d. Gaussian variables X_{ij} introduced in (77), we define the $d = \infty$ version of the projection kernel by

$$Z^{(\infty)} := \sum_{i=1}^{m} \sum_{j=1}^{m} X_{ij}^2. \tag{81}$$

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,

$$\frac{Z^{(\infty)} - m^2 \mathbb{E}[X_{ij}^2]}{\sqrt{m^2\, \text{Var}(X_{ij}^2)}} \xrightarrow{\mathcal{D}} \mathcal{N}(0,1) \quad (m \to \infty). \tag{82}$$

Substituting (79) and (80) into (82), we obtain

$$\frac{Z^{(\infty)} - \frac{m^2}{d}}{\sqrt{\frac{2m^2}{d^2}}} \xrightarrow{\mathcal{D}} \mathcal{N}(0,1) \quad (m \to \infty). \tag{83}$$

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

$$\mathcal{N}\left(\frac{m^2}{d}, \frac{2m^2}{d^2}\right). \tag{84}$$

For $d = 768$ and $m = 64$, sampling[36] $\boldsymbol{U}_A, \boldsymbol{U}_B \in \mathcal{V}_m(\mathbb{R}^d)$ and overlaying the density of $\mathcal{N}\left(m^2/d,\, 2m^2/d^2\right)$ on the histogram of Z shows a rough match (Fig. 18c). However, due to both the large-d approximation in (76) and the large-m approximation in (84), the fit is weaker than for the approximations of R_{ij} and R_{ij}^2 (Figs. 18a and 18b).

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

In Appendix J.2, we derived an approximate distribution of Z from the approximate mean (79) and approximate variance (80) under the independence assumption in (77). 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.

[36]In the simulation, instead of using the identity-based $\boldsymbol{U}_A$ in (72), we generated $\boldsymbol{U}_A$ and $\boldsymbol{U}_B$ 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.

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

From (74), the mean of Z is

$$\mathbb{E}[Z] = \sum_{i=1}^{m}\sum_{j=1}^{m} \mathbb{E}\left[R_{ij}^2\right], \tag{85}$$

and the variance of Z is

$$\mathrm{Var}(Z) = \mathbb{E}[(Z - \mathbb{E}[Z])^2] = \mathbb{E}[Z^2] - \mathbb{E}[Z]^2. \tag{86}$$

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

Expanding Z^2 yields

$$Z^2 = \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{i'=1}^{m}\sum_{j'=1}^{m} R_{ij}^2 R_{i'j'}^2, \tag{87}$$

so

$$\mathbb{E}[Z^2] = \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{i'=1}^{m}\sum_{j'=1}^{m} \mathbb{E}[R_{ij}^2 R_{i'j'}^2]. \tag{88}$$

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

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

$$f_{R_{ij}}(x) = \frac{1}{B\left(\frac{1}{2}, \frac{d-1}{2}\right)}(1-x^2)^{\frac{d-3}{2}} \quad x \in [-1, 1], \tag{89}$$

$$\mathbb{E}[R_{ij}] = 0, \tag{90}$$

where B is the Beta function. Overlaying the density in (89) on the histogram of R_{ij} in Fig. 18a shows a good fit.

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

$$R_{ij}^2 \sim \mathrm{Beta}\left(\frac{1}{2}, \frac{d-1}{2}\right), \tag{91}$$

$$\mathbb{E}[R_{ij}^2] = \frac{1}{d}. \tag{92}$$

Overlaying the density of $\mathrm{Beta}\,(1/2, (d-1)/2)$ on the histogram of R_{ij}^2 in Fig. 18b also shows a good fit. From (92), we have $\mathrm{Var}(R_{ij}) = \mathbb{E}[R_{ij}^2] - \mathbb{E}[R_{ij}]^2 = 1/d$.

Using (92) in (85), we obtain

$$\mathbb{E}[Z] = \sum_{i=1}^{m}\sum_{j=1}^{m} \mathbb{E}\left[R_{ij}^2\right] = m^2\mathbb{E}\left[R_{ij}^2\right] = \frac{m^2}{d}. \tag{93}$$

This matches the mean of the loose approximation in (84) for $Z^{(\infty)}$ in (81). Moreover, when $m = d$, we have $\boldsymbol{U}_A, \boldsymbol{U}_B \in \mathcal{V}_m(\mathbb{R}^d) = \mathcal{O}(\mathbb{R}^d)$, 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 $Z = \mathrm{PK}(\mathcal{S}_A, \mathcal{S}_B) = d$. Consistently, $m^2/d = d^2/d = d$.

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

Since $\boldsymbol{R} \in \mathcal{O}(\mathbb{R}^d)$ 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 (88), we consider the following four cases[37]:

(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

$$\begin{aligned}\mathbb{E}[Z^2] &= m^2\mathbb{E}[R_{ij}^4] + m^2(m-1)\mathbb{E}[R_{ij}^2 R_{ij'}^2] \\ &\quad + m^2(m-1)\mathbb{E}[R_{ij}^2 R_{i'j}^2] \\ &\quad + m^2(m-1)^2\mathbb{E}[R_{ij}^2 R_{i'j'}^2].\end{aligned} \tag{94}$$

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

$$\begin{aligned}&\mathbb{E}[R_{ij}R_{rs}R_{\alpha\beta}R_{\lambda\mu}] \\ &= -\frac{1}{(d-1)d(d+2)}(\delta_{ir}\delta_{\alpha\lambda}\delta_{j\beta}\delta_{s\mu} \\ &\quad + \delta_{ir}\delta_{\alpha\lambda}\delta_{j\mu}\delta_{s\beta} + \delta_{i\alpha}\delta_{r\lambda}\delta_{js}\delta_{\beta\mu} \\ &\quad + \delta_{i\alpha}\delta_{r\lambda}\delta_{j\mu}\delta_{\beta s} + \delta_{i\lambda}\delta_{r\alpha}\delta_{js}\delta_{\beta\mu} \\ &\quad + \delta_{i\lambda}\delta_{r\alpha}\delta_{j\beta}\delta_{s\mu}) \\ &\quad + \frac{d+1}{(d-1)d(d+2)}(\delta_{ir}\delta_{\alpha\lambda}\delta_{js}\delta_{\beta\mu} \\ &\quad + \delta_{i\alpha}\delta_{r\lambda}\delta_{j\beta}\delta_{s\mu} + \delta_{i\lambda}\delta_{r\alpha}\delta_{j\mu}\delta_{s\beta}).\end{aligned} \tag{95}$$

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

Setting $r = i$, $s = j$, $\alpha = \lambda = i'$, and $\beta = \mu = j'$ in (95) yields

$$
\begin{aligned}
&\mathbb{E}[R_{ij}^2 R_{i'j'}^2] \\
&= -\frac{1}{(d-1)d(d+2)}(\delta_{ii}\delta_{i'i'}\delta_{jj'}\delta_{jj'} \\
&\quad + \delta_{ii}\delta_{i'i'}\delta_{jj'}\delta_{jj'} + \delta_{ii'}\delta_{ii'}\delta_{jj}\delta_{j'j'} \\
&\quad + \delta_{ii'}\delta_{ii'}\delta_{jj'}\delta_{j'j} + \delta_{ii'}\delta_{ii'}\delta_{jj}\delta_{j'j'} \\
&\quad + \delta_{ii'}\delta_{ii'}\delta_{jj'}\delta_{jj'}) \\
&\quad + \frac{d+1}{(d-1)d(d+2)}(\delta_{ii}\delta_{i'i'}\delta_{jj}\delta_{j'j'} \\
&\quad + \delta_{ii'}\delta_{ii'}\delta_{jj'}\delta_{jj'} + \delta_{ii'}\delta_{ii'}\delta_{jj'}\delta_{jj'}) \\
&= -\frac{1}{(d-1)d(d+2)}(\delta_{jj'} + \delta_{jj'} + \delta_{ii'} \\
&\quad + \delta_{ii'}\delta_{jj'} + \delta_{ii'} + \delta_{ii'}\delta_{jj'}) \\
&\quad + \frac{d+1}{(d-1)d(d+2)}(1 + \delta_{ii'}\delta_{jj'} + \delta_{ii'}\delta_{jj'}) \\
&= -\frac{2}{(d-1)d(d+2)}(\delta_{jj'} + \delta_{ii'} + \delta_{ii'}\delta_{jj'}) \\
&\quad + \frac{d+1}{(d-1)d(d+2)}(1 + 2\delta_{ii'}\delta_{jj'}). \qquad (96)
\end{aligned}
$$

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

Setting $i' = i$ and $j' = j$ in (96), we obtain

$$
\begin{aligned}
&\mathbb{E}[R_{ij}^4] \\
&= -\frac{2}{(d-1)d(d+2)}(\delta_{jj} + \delta_{ii} + \delta_{ii}\delta_{jj}) \\
&\quad + \frac{d+1}{(d-1)d(d+2)}(1 + 2\delta_{ii}\delta_{jj}) \\
&= -\frac{6}{(d-1)d(d+2)} + \frac{3(d+1)}{(d-1)d(d+2)} \\
&= \frac{3(d-1)}{(d-1)d(d+2)} = \frac{3}{d(d+2)}. \qquad (97)
\end{aligned}
$$

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

$$
\begin{aligned}
&\mathbb{E}[R_{ij}^2 R_{ij'}^2] \\
&= -\frac{2}{(d-1)d(d+2)}(\delta_{jj'} + \delta_{ii} + \delta_{ii}\delta_{jj'}) \\
&\quad + \frac{d+1}{(d-1)d(d+2)}(1 + 2\delta_{ii}\delta_{jj'}) \\
&= -\frac{2}{(d-1)d(d+2)} + \frac{d+1}{(d-1)d(d+2)} \\
&= \frac{d-1}{(d-1)d(d+2)} = \frac{1}{d(d+2)}. \qquad (98)
\end{aligned}
$$

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

$$
\begin{aligned}
&\mathbb{E}[R_{ij}^2 R_{i'j}^2] \\
&= -\frac{2}{(d-1)d(d+2)}(\delta_{jj} + \delta_{ii'} + \delta_{ii'}\delta_{jj}) \\
&\quad + \frac{d+1}{(d-1)d(d+2)}(1 + 2\delta_{ii'}\delta_{jj}) \\
&= -\frac{2}{(d-1)d(d+2)} + \frac{d+1}{(d-1)d(d+2)} \\
&= \frac{d-1}{(d-1)d(d+2)} = \frac{1}{d(d+2)}. \qquad (99)
\end{aligned}
$$

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

$$
\begin{aligned}
&\mathbb{E}[R_{ij}^2 R_{i'j'}^2] \\
&= -\frac{2}{(d-1)d(d+2)}(\delta_{jj'} + \delta_{ii'} + \delta_{ii'}\delta_{jj'}) \\
&\quad + \frac{d+1}{(d-1)d(d+2)}(1 + 2\delta_{ii'}\delta_{jj'}) \\
&= \frac{d+1}{(d-1)d(d+2)}. \qquad (100)
\end{aligned}
$$

Note that (98) and (99) imply[38] $\mathbb{E}[R_{ij}^2 R_{ij'}^2] = \mathbb{E}[R_{ij}^2 R_{i'j}^2]$.

Substituting (97), (98), (99), and (100) into (94), we obtain

$$
\begin{aligned}
&\mathbb{E}[Z^2] \\
&= m^2\mathbb{E}[R_{ij}^4] + m^2(m-1)\mathbb{E}[R_{ij}^2 R_{ij'}^2] \\
&\quad + m^2(m-1)\mathbb{E}[R_{ij}^2 R_{i'j}^2] \\
&\quad + m^2(m-1)^2\mathbb{E}[R_{ij}^2 R_{i'j'}^2] \\
&= m^2\mathbb{E}[R_{ij}^4] + 2m^2(m-1)\mathbb{E}[R_{ij}^2 R_{ij'}^2] \\
&\quad + m^2(m-1)^2\mathbb{E}[R_{ij}^2 R_{i'j'}^2] \\
&= m^2\frac{3}{d(d+2)} + 2m^2(m-1)\frac{1}{d(d+2)} \\
&\quad + m^2(m-1)^2\frac{d+1}{(d-1)d(d+2)} \\
&= \frac{m^2}{(d-1)d(d+2)}\big\{3(d-1) \\
&\quad + 2(m-1)(d-1) + (m-1)^2(d+1)\big\} \\
&= \frac{m^2}{(d-1)d(d+2)}\Big[\\
&\quad \big\{(m^2 - 2m + 1) + 2(m-1) + 3\big\}d \\
&\quad + (m-1)^2 - 2(m-1) - 3\Big] \\
&= \frac{m^2}{(d-1)d(d+2)}\left\{(m^2+2)d + m(m-4)\right\}. \qquad (101)
\end{aligned}
$$

[38] This also follows from symmetry.

Using (85) and (101), we compute $\mathrm{Var}(Z)$ in (86) as

$$\begin{aligned}
&\mathrm{Var}(Z) = \mathbb{E}[Z^2] - (\mathbb{E}[Z])^2 \\
&= \frac{m^2}{d(d-1)(d+2)} \left\{(m^2+2)d + m(m-4)\right\} \\
&\quad - \frac{m^4}{d^2} \\
&= \frac{m^2}{d^2(d-1)(d+2)} \{(m^2+2)d^2 \\
&\quad + (m^2-4m)d - m^2(d^2+d-2)\} \\
&= \frac{m^2}{d^2(d-1)(d+2)} (2d^2 - 4md + 2m^2) \\
&= \frac{2m^2(d-m)^2}{d^2(d-1)(d+2)} \qquad (102)
\end{aligned}$$

In particular, when $m \ll d$, we have $(d-m)^2 \approx d^2 \approx (d-1)(d+2)$ and thus $\mathrm{Var}(Z) \approx 2m^2/d^2$, which matches the variance of the loose approximation in (84) for $Z^{(\infty)}$ in (81). Moreover, when $m = d$, we have seen that $Z = \mathrm{PK}(\mathcal{S}_A, \mathcal{S}_B) = d$ for any $\boldsymbol{U}_A, \boldsymbol{U}_B \in \mathcal{V}_m(\mathbb{R}^d) = \mathcal{O}(\mathbb{R}^d)$, and hence $\mathrm{Var}(Z) = 0$. This also follows from (102), since substituting $m = d$ yields $\mathrm{Var}(Z) = 0$.

J.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 J.2, we cannot directly treat Z as a sum of independent variables and apply the central limit theorem as is.

To assess the strength of these dependencies, we compare the variance and covariances of R_{ij}^2. From (92) and (97), the variance is

$$\begin{aligned}
&\mathrm{Var}(R_{ij}^2) = \mathbb{E}[R_{ij}^4] - \mathbb{E}[R_{ij}^2]^2 \\
&= \frac{3}{d(d+2)} - \frac{1}{d^2} = \frac{3d-(d+2)}{d^2(d+2)} \\
&= \frac{2(d-1)}{d^2(d+2)} = O\left(\frac{1}{d^2}\right) \qquad (103)
\end{aligned}$$

For $j \neq j'$, from (92) and (98), the covariance is

$$\begin{aligned}
&\mathrm{Cov}(R_{ij}^2, R_{ij'}^2) = \mathbb{E}[R_{ij}^2 R_{ij'}^2] - \mathbb{E}[R_{ij}^2]\mathbb{E}[R_{ij'}^2] \\
&= \frac{1}{d(d+2)} - \frac{1}{d^2} = \frac{d-(d+2)}{d^2(d+2)} \\
&= -\frac{2}{d^2(d+2)} = O\left(\frac{1}{d^3}\right) \qquad (104)
\end{aligned}$$

The same order holds for $\mathrm{Cov}(R_{ij}^2, R_{i'j}^2)$ when $i \neq i'$. For $i \neq i'$ and $j \neq j'$, from (92) and (100), we have

$$\begin{aligned}
&\mathrm{Cov}(R_{ij}^2, R_{i'j'}^2) \\
&= \mathbb{E}[R_{ij}^2 R_{i'j'}^2] - \mathbb{E}[R_{ij}^2]\mathbb{E}[R_{i'j'}^2] \\
&= \frac{d+1}{(d-1)d(d+2)} - \frac{1}{d^2} \\
&= \frac{d(d+1)-(d-1)(d+2)}{d^2(d-1)(d+2)} \\
&= \frac{d^2+d-(d^2+d-2)}{d^2(d-1)(d+2)} \\
&= \frac{2}{d^2(d-1)(d+2)} = O\left(\frac{1}{d^4}\right) \qquad (105)
\end{aligned}$$

Thus, the covariances in (104) and (105) are at least a factor of $1/d$ smaller than the variance in (103).

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 distribution.[39] Using the mean in (93) and the variance in (102), we adopt

$$\begin{aligned}
&\mathcal{N}\left(\mathbb{E}[Z],\ \mathrm{Var}(Z)\right) \\
&= \mathcal{N}\left(\frac{m^2}{d}, \frac{2m^2(d-m)^2}{d^2(d-1)(d+2)}\right) \qquad (106)
\end{aligned}$$

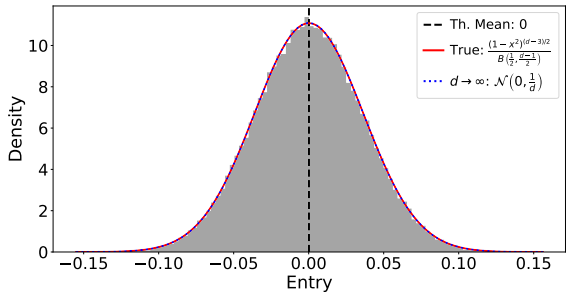
as a tight approximate distribution of Z. Overlaying the density of (106) on the histogram of Z in Fig. 18c shows a better fit than the loose approximation in Appendix J.2.

K Comparing Projection Kernel and Composition Score on preprocessed weights

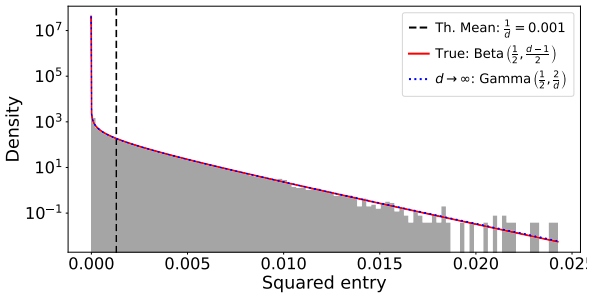
Pretrained weights can be simplified without changing the output of a language model (Elhage et al., 2021; Nanda and Bloom, 2022). Following Gurnee et al. (2024), we call this procedure *weight preprocessing*, and we call the resulting weights *preprocessed weights*.

In TransformerLens (Nanda and Bloom, 2022), which is widely used in mechanistic interpretability (Olah, 2022), 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

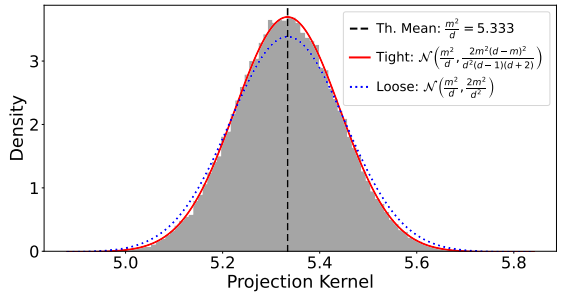
[39] We do not verify conditions required for a rigorous application of the central limit theorem, such as independence or specific dependence structures, and instead adopt a Gaussian approximation as a practical model for the distribution of Z.



(a) Entry R_{ij}



(b) Squared entry R_{ij}^2



(c) Projection kernel Z

Figure 18: 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 GPT-2 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 $d \to \infty$. For (c), we plot two approximations to the PK distribution: the tight approximation in Appendix J.3 and the loose approximation in Appendix J.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.

the weights may differ from the original ones. In this section, we first describe the weight preprocessing procedure based on Nanda and Bloom (2022). We then compare PK and CS computed on the original weights with those computed on the preprocessed weights.

K.1 Background: Weight preprocessing

Based on Nanda and Bloom (2022), we describe four types of weight preprocessing[40]. Table 10 summarizes these procedures and compares the original weights and the preprocessed weights.

K.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 $\boldsymbol{\gamma} \in \mathbb{R}^d$ and bias $\boldsymbol{\beta} \in \mathbb{R}^d$, LN for an input $\boldsymbol{x} = (x_1, \ldots, x_d)^\top \in \mathbb{R}^d$ is

$$\mathrm{LN}(\boldsymbol{x}) = \boldsymbol{\gamma} \odot \frac{\boldsymbol{x} - \mathrm{Mean}(\boldsymbol{x})\mathbf{1}_d}{\mathrm{Std}(\boldsymbol{x})} + \boldsymbol{\beta}, \quad (107)$$

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

$$\mathrm{Mean}(\boldsymbol{x}) = \frac{1}{d}\mathbf{1}_d{}^\top \boldsymbol{x}, \quad (108)$$

$$\mathrm{Std}(\boldsymbol{x}) = \sqrt{\frac{1}{d}\sum_{i=1}^{d}(x_i - \mathrm{Mean}(\boldsymbol{x}))^2}. \quad (109)$$

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

$$\boldsymbol{C}_d = \boldsymbol{I}_d - \frac{1}{d}\mathbf{1}_d\mathbf{1}_d{}^\top \in \mathbb{R}^{d\times d}, \quad (110)$$

where $\boldsymbol{I}_d \in \mathbb{R}^{d\times d}$ is the identity matrix. From the definition of $\boldsymbol{C}_d$ in (110) and $\mathrm{Mean}(\cdot)$ in (108), we have

$$\boldsymbol{C}_d\boldsymbol{x} = \boldsymbol{x} - \frac{1}{d}\mathbf{1}_d\mathbf{1}_d{}^\top\boldsymbol{x} = \boldsymbol{x} - \mathrm{Mean}(\boldsymbol{x})\mathbf{1}_d. \quad (111)$$

The matrix $\boldsymbol{C}_d$ is an orthogonal projection matrix since it satisfies

$$\boldsymbol{C}_d\boldsymbol{C}_d = \boldsymbol{C}_d, \quad (112)$$

$$\boldsymbol{C}_d{}^\top = \boldsymbol{C}_d. \quad (113)$$

[40] https://github.com/TransformerLensOrg/TransformerLens/blob/main/further_comments.md

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

Table 10: 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, $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^d$ are the LN scaling weight and bias, and $\boldsymbol{D}_\gamma \in \mathbb{R}^{d\times d}$ is the diagonal matrix with diagonal entries $\boldsymbol{\gamma}$. The matrices $\boldsymbol{C}_d \in \mathbb{R}^{d\times d}$ and $\boldsymbol{C}_T \in \mathbb{R}^{T\times T}$ are the centering matrices in (110) for dimensions d and T, where T is the vocabulary size. Note that only $\mathrm{LN}(\boldsymbol{x})$ is a function, whereas the other entries are parameters. See Appendix K.1 for more details.

Decomposing LN into four steps. LN can be decomposed into the following four steps[41]:

$$\begin{aligned}\boldsymbol{x}^{(1)} &= \boldsymbol{x} - \mathrm{Mean}(\boldsymbol{x})\mathbf{1}_d \\ &= \boldsymbol{C}_d\boldsymbol{x}, \quad (\because (111)) \end{aligned} \tag{114}$$

$$\begin{aligned}\boldsymbol{x}^{(2)} &= \frac{\boldsymbol{x}^{(1)}}{\sqrt{\frac{1}{d}\sum_{i=1}^d (x_i - \mathrm{Mean}(\boldsymbol{x}))^2}} \\ &= \sqrt{d}\,\frac{\boldsymbol{x}^{(1)}}{\sqrt{\sum_{i=1}^d \left(x_i^{(1)}\right)^2}} = \sqrt{d}\,\frac{\boldsymbol{x}^{(1)}}{\|\boldsymbol{x}^{(1)}\|},\end{aligned} \tag{115}$$

$$\boldsymbol{x}^{(3)} = \boldsymbol{\gamma} \odot \boldsymbol{x}^{(2)} = \boldsymbol{D}_\gamma \boldsymbol{x}^{(2)}, \tag{116}$$

$$\boldsymbol{x}^{(4)} = \boldsymbol{x}^{(3)} + \boldsymbol{\beta}. \tag{117}$$

Here, $x_i^{(1)}$ is the i-th component of $\boldsymbol{x}^{(1)}$ in (114), and $\boldsymbol{D}_\gamma \in \mathbb{R}^{d\times d}$ is the diagonal matrix whose diagonal entries are $\boldsymbol{\gamma}$. By definition, $\mathrm{LN}(\boldsymbol{x}) = \boldsymbol{x}^{(4)}$.

Folding into the weight and bias after LN. Based on the four-step decomposition, consider the affine map immediately after LN that takes $\mathrm{LN}(\boldsymbol{x}) = \boldsymbol{x}^{(4)}$ as input. Let $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{d_{\text{mid}}\times d}$ and $\boldsymbol{b}_{\text{in}} \in \mathbb{R}^{d_{\text{mid}}}$ be its weight and bias[42]. Then

$$\begin{aligned}&\boldsymbol{W}_{\text{in}}\boldsymbol{x}^{(4)} + \boldsymbol{b}_{\text{in}} \\ &= \boldsymbol{W}_{\text{in}}(\boldsymbol{x}^{(3)} + \boldsymbol{\beta}) + \boldsymbol{b}_{\text{in}} \quad (\because (117)) \\ &= \boldsymbol{W}_{\text{in}}\boldsymbol{D}_\gamma\boldsymbol{x}^{(2)} + \boldsymbol{W}_{\text{in}}\boldsymbol{\beta} + \boldsymbol{b}_{\text{in}} \quad (\because (116)) \\ &= \boldsymbol{W}_{\text{in}}\boldsymbol{D}_\gamma\frac{\sqrt{d}\boldsymbol{x}^{(1)}}{\|\boldsymbol{x}^{(1)}\|} + \boldsymbol{W}_{\text{in}}\boldsymbol{\beta} + \boldsymbol{b}_{\text{in}} \quad (\because (115)) \\ &= \boldsymbol{W}_{\text{in}}\boldsymbol{D}_\gamma\boldsymbol{C}_d\frac{\sqrt{d}\boldsymbol{x}}{\|\boldsymbol{x}^{(1)}\|} + \boldsymbol{W}_{\text{in}}\boldsymbol{\beta} + \boldsymbol{b}_{\text{in}} \quad (\because (114)) \\ &= \boldsymbol{W}_{\text{in}}\boldsymbol{D}_\gamma\boldsymbol{C}_d\boldsymbol{C}_d\frac{\sqrt{d}\boldsymbol{x}}{\|\boldsymbol{x}^{(1)}\|} \\ &\qquad + \boldsymbol{W}_{\text{in}}\boldsymbol{\beta} + \boldsymbol{b}_{\text{in}} \quad (\because (112)) \\ &= \boldsymbol{W}_{\text{in}}\boldsymbol{D}_\gamma\boldsymbol{C}_d\frac{\sqrt{d}\boldsymbol{x}^{(1)}}{\|\boldsymbol{x}^{(1)}\|} + \boldsymbol{W}_{\text{in}}\boldsymbol{\beta} + \boldsymbol{b}_{\text{in}} \quad (\because (114)) \\ &= \boldsymbol{W}_{\text{in}}\boldsymbol{D}_\gamma\boldsymbol{C}_d\boldsymbol{x}^{(2)} + \boldsymbol{W}_{\text{in}}\boldsymbol{\beta} + \boldsymbol{b}_{\text{in}} \quad (\because (115)).\end{aligned} \tag{118}$$

Define

$$\mathrm{LN}^{(2)}(\boldsymbol{x}) = \boldsymbol{x}^{(2)} = \frac{\boldsymbol{x} - \mathrm{Mean}(\boldsymbol{x})\mathbf{1}_d}{\mathrm{Std}(\boldsymbol{x})}, \tag{119}$$

$$\boldsymbol{W}'_{\text{in}} = \boldsymbol{W}_{\text{in}}\boldsymbol{D}_\gamma\boldsymbol{C}_d,\ \boldsymbol{b}'_{\text{in}} = \boldsymbol{W}_{\text{in}}\boldsymbol{\beta} + \boldsymbol{b}_{\text{in}}. \tag{120}$$

Then, from $\mathrm{LN}(\boldsymbol{x}) = \boldsymbol{x}^{(4)}$ and (118), we obtain

$$\boldsymbol{W}_{\text{in}}\mathrm{LN}(\boldsymbol{x}) + \boldsymbol{b}_{\text{in}} = \boldsymbol{W}'_{\text{in}}\mathrm{LN}^{(2)}(\boldsymbol{x}) + \boldsymbol{b}'_{\text{in}}. \tag{121}$$

Equation (121) shows that replacing LN in (107) with $\mathrm{LN}^{(2)}(\boldsymbol{x})$ in (119), and replacing $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{b}_{\text{in}}$ with $\boldsymbol{W}'_{\text{in}}$ and $\boldsymbol{b}'_{\text{in}}$, does not change the output of the affine map. In particular, $\mathrm{LN}^{(2)}(\boldsymbol{x})$ in (119) performs only centering and standard-deviation normalization, and is simpler than the original LN

[41] Since $\mathbf{1}_d{}^\top\boldsymbol{C}_d = \mathbf{1}_d{}^\top\left(\boldsymbol{I}_d - \mathbf{1}_d\mathbf{1}_d{}^\top/d\right) = \mathbf{0}^\top$, we have $\mathbf{1}_d{}^\top\boldsymbol{C}_d\boldsymbol{x} = 0$ for any $\boldsymbol{x} \in \mathbb{R}^d$. Thus, $\boldsymbol{C}_d$ is the orthogonal projection onto the orthogonal complement of the one-dimensional subspace spanned by $\mathbf{1}_d$. In particular, for (114), we have $\boldsymbol{x}^{(1)} = \boldsymbol{C}_d\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}^\top(\mathbf{1}_d/\sqrt{d})\mathbf{1}_d/\sqrt{d}$, so Nanda and Bloom (2022) view centering as removing the component of $\boldsymbol{x}$ in the $\mathbf{1}_d$ direction using the unit vector $\mathbf{1}_d/\sqrt{d}$. For the resulting $\boldsymbol{x}^{(1)}$, they obtain $\boldsymbol{x}^{(2)}$ in (115) by projecting it onto the sphere of radius $\sqrt{d}$ while preserving its direction.

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

in (107). For attention heads, this folding can be applied to the reading weights $\boldsymbol{W}_{\mathrm{Q}}^h$, $\boldsymbol{W}_{\mathrm{K}}^h$, and $\boldsymbol{W}_{\mathrm{V}}^h$.

K.1.2 Centering weights that write to the residual stream

Let $\boldsymbol{\zeta} \in \mathbb{R}^{d_{\text{mid}}}$ be an intermediate representation, and suppose the affine output $\boldsymbol{W}_{\text{out}}\boldsymbol{\zeta} + \boldsymbol{b}_{\text{out}} \in \mathbb{R}^d$ is written to the residual stream, where $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{d \times d_{\text{mid}}}$ and $\boldsymbol{b}_{\text{out}} \in \mathbb{R}^d$. Consider applying LN to $\boldsymbol{W}_{\text{out}}\boldsymbol{\zeta} + \boldsymbol{b}_{\text{out}}$. As seen in (114), LN performs centering in the first step. Therefore, centering gives

$$\begin{aligned} &\boldsymbol{C}_d(\boldsymbol{W}_{\text{out}}\boldsymbol{\zeta} + \boldsymbol{b}_{\text{out}}) \\ =&\boldsymbol{C}_d\boldsymbol{C}_d(\boldsymbol{W}_{\text{out}}\boldsymbol{\zeta} + \boldsymbol{b}_{\text{out}}) \quad (\because (112)) \\ =&\boldsymbol{C}_d(\boldsymbol{C}_d\boldsymbol{W}_{\text{out}}\boldsymbol{\zeta} + \boldsymbol{C}_d\boldsymbol{b}_{\text{out}}). \end{aligned} \tag{122}$$

Thus, defining

$$\boldsymbol{W}'_{\text{out}} = \boldsymbol{C}_d\boldsymbol{W}_{\text{out}},\ \boldsymbol{b}'_{\text{out}} = \boldsymbol{C}_d\boldsymbol{b}_{\text{out}}, \tag{123}$$

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

K.1.3 Other preprocessing steps

Under the default settings in TransformerLens (Nanda and Bloom, 2022) when loading pretrained model weights, the unembedding matrix is also centered, and the bias term in the output computation of attention is also folded. We briefly describe these preprocessing steps. These steps are unrelated to PK and CS.

Centering the unembedding matrix. Let $\boldsymbol{E}_{\text{out}} \in \mathbb{R}^{d \times T}$ be the unembedding matrix and let $\widetilde{\boldsymbol{x}} \in \mathbb{R}^d$ be the vector after the final LN in (53), where $T = |\mathcal{T}|$. Logits are computed as ${\boldsymbol{E}_{\text{out}}}^\top \widetilde{\boldsymbol{x}} \in \mathbb{R}^T$, and token probabilities are computed by applying softmax. By definition of softmax, adding the same constant to all components of ${\boldsymbol{E}_{\text{out}}}^\top \widetilde{\boldsymbol{x}}$ does not change the output. Let $\boldsymbol{C}_T \in \mathbb{R}^{T \times T}$ be the T-dimensional centering matrix, analogous to (110). Then centering logits using $\boldsymbol{C}_T$ does not affect the softmax output, so

$$\begin{aligned} &\mathrm{softmax}({\boldsymbol{E}_{\text{out}}}^\top \widetilde{\boldsymbol{x}}) = \mathrm{softmax}(\boldsymbol{C}_T{\boldsymbol{E}_{\text{out}}}^\top \widetilde{\boldsymbol{x}}) \\ =&\,\mathrm{softmax}((\boldsymbol{E}_{\text{out}}{\boldsymbol{C}_T}^\top)^\top \widetilde{\boldsymbol{x}}) \\ =&\mathrm{softmax}((\boldsymbol{E}_{\text{out}}\boldsymbol{C}_T)^\top \widetilde{\boldsymbol{x}}) \in [0,1]^T \quad (\because (113)) \end{aligned} \tag{124}$$

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

Folding bias terms in the attention output computation. In $o^h(\boldsymbol{X})$ in (15), we omitted bias terms. Including the bias terms $\boldsymbol{b}_{\mathrm{V}}^h \in \mathbb{R}^{d_{\text{head}}}$ and $\boldsymbol{b}_{\mathrm{O}}^h \in \mathbb{R}^d$ in addition to $\boldsymbol{W}_{\mathrm{V}}^h$ and $\boldsymbol{W}_{\mathrm{O}}^h$ yields

$$\begin{aligned} &o^h(\boldsymbol{X}) \\ =&\boldsymbol{W}_{\mathrm{O}}^h(\boldsymbol{W}_{\mathrm{V}}^h\boldsymbol{X}\boldsymbol{A}^h + \boldsymbol{b}_{\mathrm{V}}^h) + \boldsymbol{b}_{\mathrm{O}}^h \\ =&\boldsymbol{W}_{\mathrm{O}}^h\boldsymbol{W}_{\mathrm{V}}^h\boldsymbol{X}\boldsymbol{A}^h + \boldsymbol{W}_{\mathrm{O}}^h\boldsymbol{b}_{\mathrm{V}}^h + \boldsymbol{b}_{\mathrm{O}}^h \\ =&\boldsymbol{W}_{\mathrm{O}}^h(\boldsymbol{W}_{\mathrm{V}}^h\boldsymbol{X}\boldsymbol{A}^h + \boldsymbol{0}_{d_{\text{head}}}) + \boldsymbol{W}_{\mathrm{O}}^h\boldsymbol{b}_{\mathrm{V}}^h + \boldsymbol{b}_{\mathrm{O}}^h \end{aligned} \tag{125}$$

Thus, defining $\boldsymbol{b}_{\mathrm{V}}^{h\prime} = \boldsymbol{0}_{d_{\text{head}}}$ and $\boldsymbol{b}_{\mathrm{O}}^{h\prime} = \boldsymbol{W}_{\mathrm{O}}^h\boldsymbol{b}_{\mathrm{V}}^h + \boldsymbol{b}_{\mathrm{O}}^h$, we can replace $\boldsymbol{b}_{\mathrm{V}}^h$ and $\boldsymbol{b}_{\mathrm{O}}^h$ with $\boldsymbol{b}_{\mathrm{V}}^{h\prime}$ and $\boldsymbol{b}_{\mathrm{O}}^{h\prime}$ without changing the attention output.

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

We compare wiring diagrams constructed from original GPT2-small weights (PK: Fig. 6, CS: Fig. 13) with those constructed from the preprocessed weights described in Section K.1, namely LN folding in (120) and centering writing weights in (123).

Figure 20 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)[43]. This result indicates that PK is robust to weight changes induced by weight preprocessing.

Figure 21 shows the top 20 CS edges computed from preprocessed weights for each of the OQ, OK, and OV pairings. This wiring diagram changes substantially compared with the original one in Fig. 13. In fact, among the 60 edges, only 18 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, many head pairs have edges in both OQ and OK, so using CS on preprocessed weights may reduce the difference between query weights and key weights. Moreover, in the original wiring diagram, edges between distant layers appear only for OV, whereas after weight preprocessing, such edges also appear for OQ and OK.

[43] Edges 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.

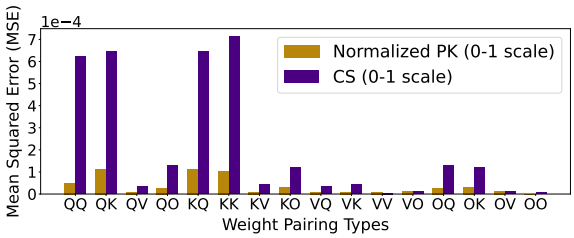


Figure 19: For GPT2-small, we report the MSE between similarities from the original and preprocessed weights for PK and CS, by weight-pairing type. Normalized PK denotes PK normalized by d_{head}, which maps it to the same $[0, 1]$ scale as CS.

K.3 MSE between similarities computed from original and preprocessed weights for PK and CS

In Appendix K.2, we observed that weight preprocessing leaves the PK wiring diagrams almost unchanged, whereas it substantially alters the CS wiring diagrams. In this subsection, we compare the mean squared error (MSE) between similarities computed from the original weights and those computed from the preprocessed weights, separately for each weight pairing type, for both PK and CS. 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 B.4 for details), and we refer to this quantity as Normalized PK.

Figure 19 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, consistent with the findings in Appendix K.2.

We also find a common trend in both Normalized PK and CS. Pairings that involve only query 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). Notably, although query, key, and value weights all undergo Layer Normalization folding, CS differs markedly between QQ or KK and VV. This observation aligns with the view that value weights play a role that is distinct from those of query and key weights.

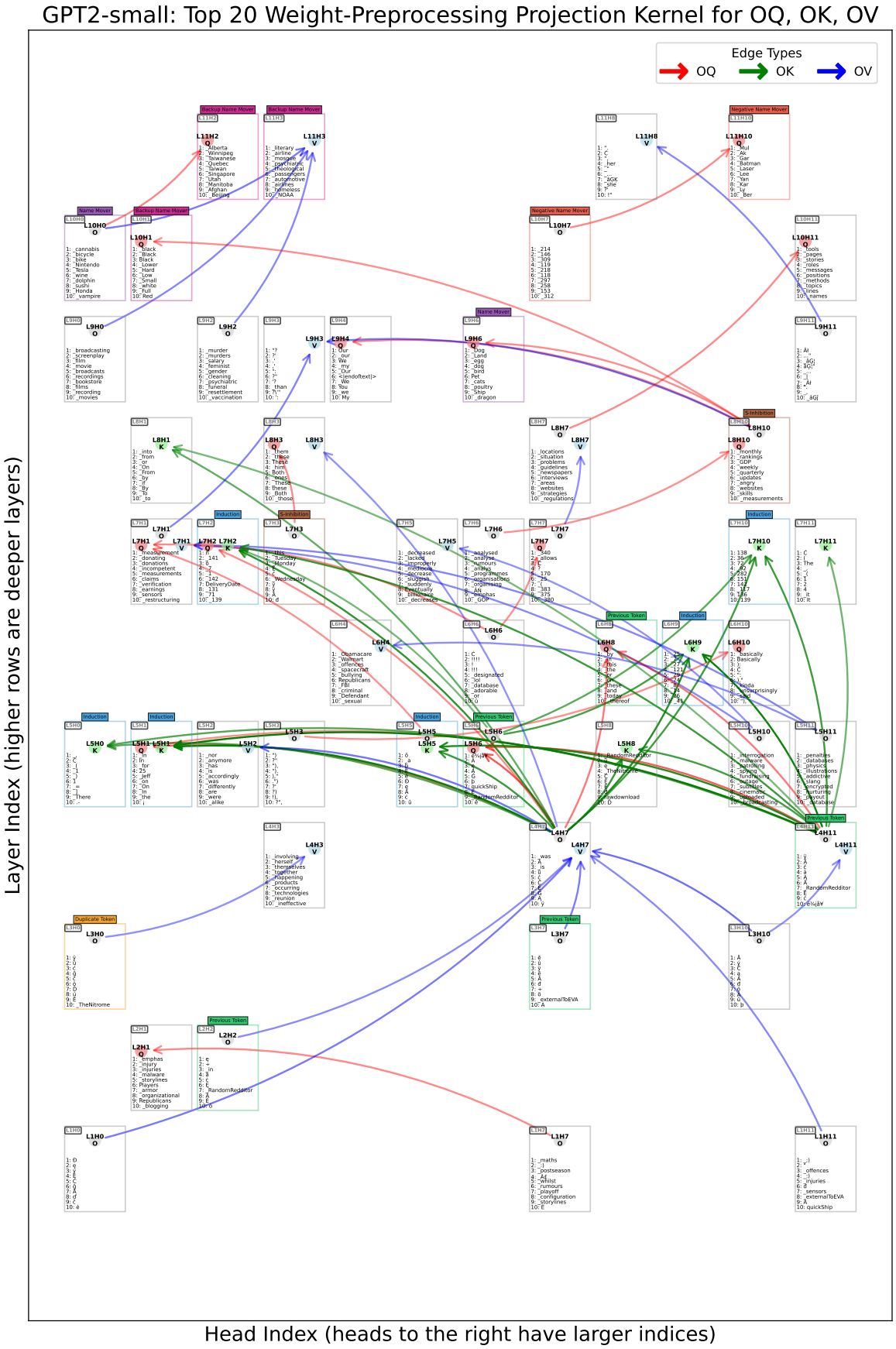


Figure 20: For GPT2-small, to compare with the wiring diagram constructed from PK on the original weights in Fig. 6, we visualize the top 20 scores of $\mathrm{PK}_{\mathrm{OQ}}$, $\mathrm{PK}_{\mathrm{OK}}$, and $\mathrm{PK}_{\mathrm{OV}}$ computed on the preprocessed weights.

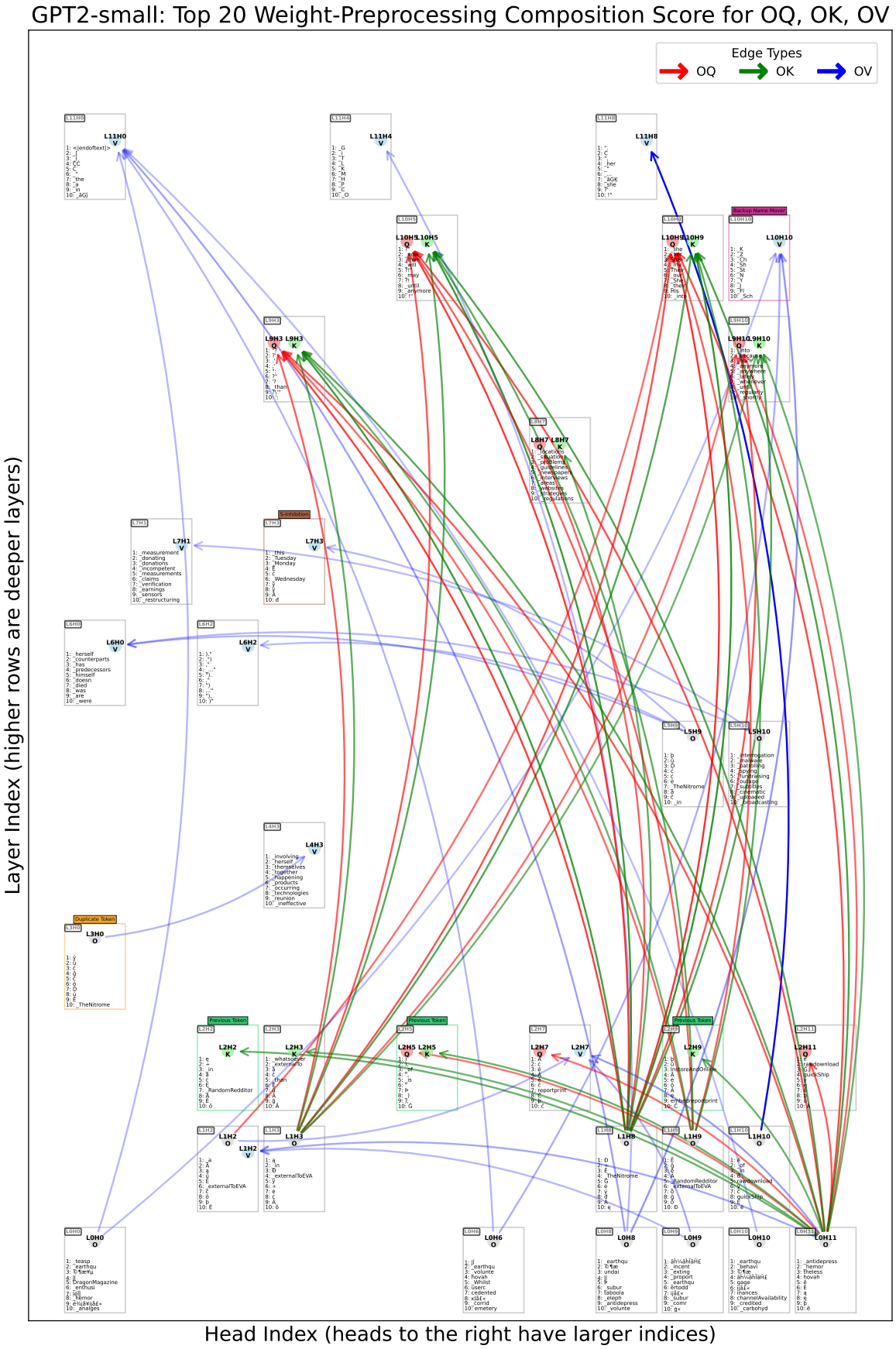


Figure 21: For GPT2-small, to compare with the wiring diagram constructed from CS on the original weights in Fig. 13, we visualize the top 20 scores of $\mathrm{CS}_{\mathrm{OQ}}$, $\mathrm{CS}_{\mathrm{OK}}$, and $\mathrm{CS}_{\mathrm{OV}}$ computed on the preprocessed weights.