Understanding Attention Heads*

Qijia Jiang

Abstract

We study ways to understand attention in this note. The aim is to identify reusable attention unit through a joint coupling between the relation QK subspace and the message OV subspace, with rank (r, c) respectively. Through this framework, each head in a layer can be decomposed into a shared set of $\{U_a, V_a\}_{a=1}^{A}$, the sum of which are trained to reconstruct the cached attention head outputs. The general methodology draws upon ideas from latent variable EM-type algorithm and spectral subspace clustering. The current experimental result suggests sensitivity to initialization and also calls for more systematic ways to conduct model selection/pick tuning parameters among the fitted decomposition families.

1 Motivation and Goal

Traditionally, when interpreting output of a MLP layer, we often resort to studying *token-level* general *concepts* that are captured by each neuron, and this has led to many successful methods such as SAE (and variants thereof). For attention layer, however, the correct level to study each head is on the *sequence-level*, and one needs to understand the general mechanisms (as opposed to concepts) captured by moving information across the sequence.

There are many ways to build mental models for a transformer-based attention architecture – one way is to view it as Graph Neural Network (a line graph, more specifically): the QK part builds edges and captures relationship strength between tokens in a sequence; while the OV part performs message passing (i.e., what's read and gets written over this attended edge/relationship). These 2 parts, *jointly*, tell us what is being learned over a particular sequence.

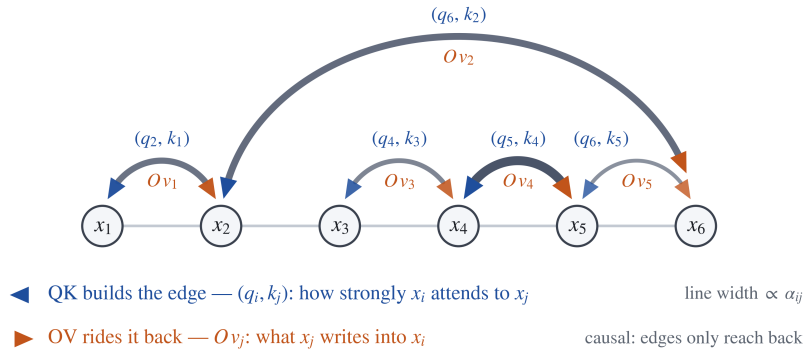

Figure 1: A pair of (q_i, k_j) and Ov_j, living in different spaces, jointly contain all the information in an edge

*This note is a work in progress, not a finished draft. The hope of sharing is simply to receive feedback in case of common interest, more than anything else.

Our goal is to decompose an attention head into *reusable mechanisms* that describe general ways in which both what relation information is captured and what is written back to the residual stream. The more ambitious goal is to apply this methodology to simultaneously decode all attention heads in a single layer (so the attention unit will be a layer-global shared one), but we will start simple with just single attention head here.

More formally, we define an attention unit to be the paired object

$$\Theta_a = (U_a, V_a), \qquad a \in \{1, \ldots, A\},$$

where U_a describes a relation operation and V_a describes its coupled message innovation – these are fixed across prompts for heads in the same layer. During discovery, one latent label jointly selects the two sides:

$$\big[(Q, K), OV\big] \quad \overset{\text{joint fit}}{\longleftrightarrow} \quad \text{latent } a \quad \longleftrightarrow \quad (U_a, V_a) \text{ as mechanism description.}$$

There is no cross-unit orthogonality constraint. In particular, $U_a \approx U_{a'}$ with $V_a \neq V_{a'}$ is permitted, but the split into different a's must earn its extra complexity through stable held-out reconstruction and coupling evidence.

1.1 Chosen head and natural corpus

The experiments are intentionally kept somewhat simple to surface successes and challenges. The 2 heads we consider here are L5H2 and L1H1 from Gemma-3-IB-IT.

The corpus contains 500 natural model-input contexts from the public HeadVis [2] Gemma sequence bank, retokenized and recomputed on the Gemma-3-1B-IT checkpoint. TRAIN and TEST each contain 250 contexts. We define an event as $g = (b, i)$, where b indexes a context, and i denotes the query token position within context b that is tested to capture the particular rule we are interested in. The complete causal row (i.e., tokens before position i within context b), including BOS and all source keys, is then retained for the event.

head	natural rule	TRAIN events	TEST events
L1H1	alphabetic word composition	1,975	1,949
	four-digit year composition	189	135
	other-number composition	476	436
	identifier composition	301	287
	newline structural routing	191	223
L5H2	exact induction	581	594
	tokenization-tolerant induction	131	131

Table 1: Event counts in the TRAIN/TEST split. No class-balancing is used so as to retain the natural family prevalence.

Let's give some concrete examples on what we suspect these 2 heads are doing. Arrows below point from attended source to query.

- Surface-level composition means that the final piece of a word, number, or identifier attends to earlier pieces of that type: for example, `mou` → `illa` in `mouilla`, earlier digits → the final piece of `2007`, or `hb`, `_`, `par` → `ni` in `hb_parni`.
- Structural newline routing means a newline query retrieves an earlier newline marking a paragraph, record, or dialogue boundary.

- Exact induction is the address rule $\ldots, x, y, \ldots, x$, where the second x attends to the earlier y.
- Tokenization-tolerant induction is the same prefix-to-continuation operation up to stripping surrounding whitespace, lowercasing, and nearby punctuation differences.

Our exploratory analysis suggests the following descriptive hypotheses for the 2 heads:

- L1H1 has semantic $A = 2$: one broad, high-rank surface-level-composition operation and one compact structural-newline operation.
- L5H2 has semantic $A = 1$: exact and tolerant induction are relation-side variants of one continuation-writing operation.

For L1H1, the two message populations have visibly different complexity: surface-level-composition target writes need about 88 directions to retain 90% of their energy, whereas newline writes need only five or six. Both populations transfer across sources: their TRAIN spans retain about 88% and 86%, respectively, of TEST energy. The plot below also supports one broad surface unit and one compact newline unit for this head.

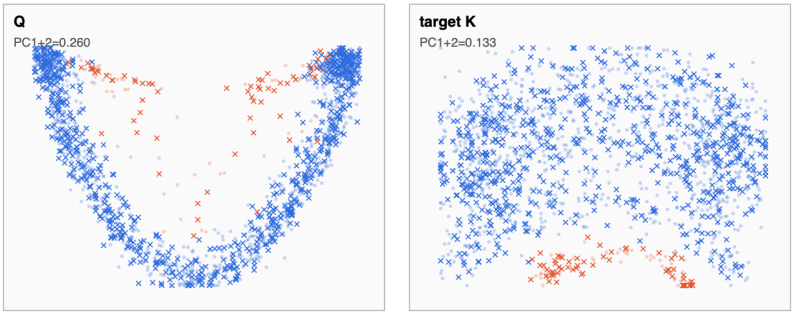

Figure 2: PCA is fit separately per panel on the training set. Here blue is surface-level composition and red is new-line structure. (circles: train; crosses: test, projected onto top 2 PC's)

L5H2 looks different. Pooling strict and tokenization-tolerant induction gives a stable, high-rank message population: its TRAIN/TEST 90%-energy ranks are 75/73, and the TRAIN span retains .844 of the TEST energy. The plot below also supports one continuation-writing operation with two address variants more strongly than two distinct message units.

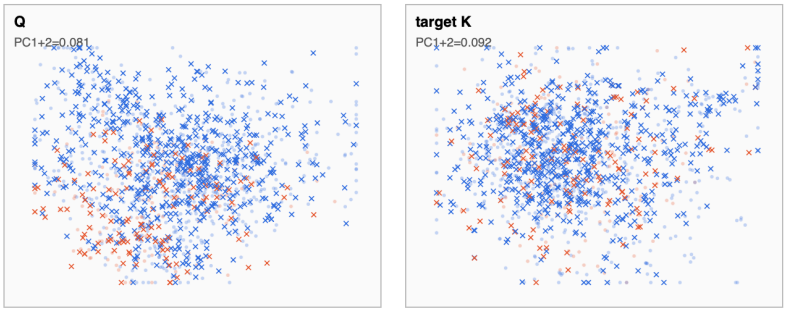

Figure 3: PCA is fit separately per panel on the training set. Here blue is strict induction and red is tokenization-tolerant induction. (circles: train; crosses: test, projected onto top 2 PC's)

These 2 heads – and their corresponding rule families – are inspired by the discovery in HeadVIs [2] of polysemanticity within attention head (e.g., the line width head) and high rank attentional feature (e.g., the answer selection head) on Haiku. In the following we test if our proposed methodology can identify the attention units without any label identities revealed.

2 Technical Ingredients

2.1 Cached head quantities

For query event $g = (b, i)$ and causal source $j \le i$, let $q_g, k_{gj} \in \mathbb{R}^d$ be the exact scaled post-RoPE query and key,

$$s_{gj} = q_g^\top k_{gj}, \qquad \alpha_{gj} = \mathrm{softmax}_{t \le i}(s_{gt})_j.$$

Let $m_{gj} = W_O^h v_{gj}$ be the message write from source j to the residual stream. Compute once an orthonormal basis B_M for $\mathrm{col}(W_O^h)$ and use the lossless coordinates [1]

$$\mu_{gj} = B_M^\top m_{gj} \in \mathbb{R}^{d_m}, \qquad \nu_g = B_M^\top \sum_{j \le i} \alpha_{gj} m_{gj} = \sum_{j \le i} \alpha_{gj} \mu_{gj}.$$

These quantities are saved as reconstruction targets and are specific to each head.

2.2 BOS and the frozen common message span

We observe in our exploratory analysis that the BOS source (with index = 0) is a genuine softmax competition but is not a semantic unit. We always keep its score and message exact:

$$\widehat{s}_{g0} = s_{g0}, \qquad \widehat{\mu}_{g0} = \mu_{g0}\,,$$

but only $j > 0$ is decomposed in our algorithm.

The same exploratory analysis also found transferable message structure shared by different families and background queries. To capture these common background message span (that doesn't contain much mechanism-revealing information), we sample 8 unannotated queries (i.e., does not belong to any active event) from every TRAIN context, form each exact non-BOS write

$$b_g^{\mathrm{bg}} = \sum_{j > 0} \alpha_{gj} \mu_{gj},$$

and let V_0 be the smallest leading uncentered-PCA span carrying 90% of their pooled TRAIN energy. Fit this once per head and freeze

$$P^0 = V_0 V_0^\top, \qquad \widetilde{\mu}_{gj} = (I - P^0)\mu_{gj}, \qquad \widetilde{\nu}_g = \sum_{j > 0} \alpha_{gj} \widetilde{\mu}_{gj}.$$

The constructed V_0 is label blind and fixed before any algorithm run – its role is to remove a shared output trunk before attempting to discover behavior-specific branches.

After these pre-processing, attention unit a owns orthonormal frames

$$U_a \in \mathbb{R}^{d \times r}, \qquad V_a \in \mathbb{R}^{d_m \times c}, \qquad V_0^\top V_a = 0,$$

with projectors $P_a^R = U_a U_a^\top$ and $P_a^M = V_a V_a^\top$. Different U_a and different V_a may overlap. The reconstructed message operator is $P^0 + P_a^M$ for unit a.

[1] This change of basis is only for computational efficiency – the operation does not lose any information

2.3 Losses & Notations

For $j > 0$ set $w_{gj} = \alpha_{gj}^2$. For exact-cached-Q/K values and labels $z(g) \in \{1, \cdots, A\}$, define

$$\ell_{gj}(z, U) = q_g^\top P_{z(g)}^R k_{gj} \, .$$

The executed attention is

$$\widehat{\alpha}_g = \mathrm{softmax}\big(s_{g0}, \{\ell_{gj}(z, U)\}_{j>0}\big).$$

The exact BOS logit anchors its mass against the projected non-BOS logits. We define the reconstruction error for the event g as:

$$W_g^{\mathrm{joint}}(z) = \frac{\left\| \widetilde{\nu}_g - \sum_{j>0} \widehat{\alpha}_{gj} P_{z(g)}^M \widetilde{\mu}_{gj} \right\|^2}{\|\widetilde{\nu}_g\|^2}.$$

Each active event matters equally with this definition; sources inside it are weighted by attention. The reason for assigning labels to active events $z(g)$ as opposed to pairs $z(g, j)$ follows from our exploratory analysis: roughly 1% of pairs instantiate the annotated behavior, yet they carry almost all squared pair-write energy. Therefore most of the information is carried at the aggregated event level.

W^{joint} couples U_a and V_a inside the actual re-softmaxed write and measures the executed joint operation. The complete reconstructed write for a model is

$$\widehat{\nu}_g = \widehat{\alpha}_{g0} \mu_{g0} + \sum_{j>0} \widehat{\alpha}_{gj} P^0 \mu_{gj} + \sum_{j>0} \widehat{\alpha}_{gj} P_{z(g)}^M \widetilde{\mu}_{gj}.$$

Lastly, for a symmetric matrix C and feasible space projector Q, write $\mathrm{TopEig}_r(C; Q)$ for the leading r-dimensional eigenspace of QCQ restricted to $\mathrm{im}(Q)$.

3 Methodological Approach

We are interested in identifying one mechanism acting at each active query event. In this section, we cast the problem as event-level bi-clustering with unknown latent variable a, which motivated an EM based algorithm – alternating between assigning cluster membership a, and fitting $\{U_a, V_a\}$. But we will see that picking the correct metric for identifying cluster and initialization will be key to its success.

3.1 Latent Variable Model for attentional features

Our model assigns one $z_g \in \{1, \cdots, A\}$ to the complete non-BOS row (corresponding to an event g) and evaluates its aggregate write performance. We denote the number of events as $|G|$.

To preserve Q/K vector geometry, compute the following scales on the TRAIN set and freeze them

$$\tau_q = |G|^{-1} \sum_g \|q_g\|^2, \qquad \tau_k = |G|^{-1} \sum_g \frac{\sum_{j>0} \alpha_{gj} \|k_{gj}\|^2}{\sum_{j>0} \alpha_{gj}},$$

then define

$$R_{ga} = \frac{1}{2} \left[\frac{\|(I - P_a^R) q_g\|^2}{\tau_q} + \frac{\sum_{j>0} \alpha_{gj} \|(I - P_a^R) k_{gj}\|^2}{\tau_k \sum_{j>0} \alpha_{gj}} \right].$$

Note that the second part of R_{ga} is aggregated over pairs in the event and attention-weighted. We fit the following joint event objective with EM:

$$L_{RW} = \frac{1}{|G|} \sum_g [R_{g,z_g} + W_g^{\text{joint}}(z_g)].$$

The E-step evaluates every event cost for $a \in \{1, \ldots, A\}$ and assigns it to the best latent group:

$$z_g = \arg\min_a \{R_{ga} + W_g^{\text{joint}}(a)\},$$

where in W_g^{joint} all non-BOS sources use candidate a, the full row is re-softmaxed with exact BOS against the candidate projected logits, and the innovation write is recomputed.

For fixed labels, the spectral minimizer for U_a of the R block is

$$U_a^{\text{spec}} = \text{TopEig}_r(C_a^R; I),$$

$$C_a^R = \sum_{g:z_g=a} \frac{1}{2|G|} \left[\frac{q_g q_g^\top}{\tau_q} + \frac{\sum_{j>0} \alpha_{gj} k_{gj} k_{gj}^\top}{\tau_k \sum_{j>0} \alpha_{gj}} \right].$$

We compare this with the previous U_a from the last iteration, retain the better of the two on L_{RW}, and refine with Grassmannian gradient update (c.f. Section 3.2) on L_{RW} to account for the W part of the loss (that depends on U_a through $\hat{\alpha}$). U_a^{spec} serves as a warm start for the M-step optimization.

The message block V_a has a simpler exact conditional update. With labels and U fixed, define the candidate attention innovation output

$$b_g = \sum_{j>0} \widehat{\alpha}_{gj}(U_{z_g}) \widetilde{\mu}_{gj}.$$

For a rank-c projector P,

$$\|\widetilde{\nu}_g - P b_g\|^2 = \|\widetilde{\nu}_g\|^2 - \left\langle P,\, 2\,\text{sym}(\widetilde{\nu}_g b_g^\top) - b_g b_g^\top \right\rangle.$$

Consequently the exact minimizer of message block is (with z, U fixed):

$$V_a^{\text{exact}} = \text{TopEig}_c(H_a^V; I - P^0), \qquad H_a^V = \sum_{g:z_g=a} \frac{2\,\text{sym}(\widetilde{\nu}_g b_g^\top) - b_g b_g^\top}{|G| \cdot \|\widetilde{\nu}_g\|^2}.$$

Here TopEig means the leading algebraic eigenspace; H_a^V need not be positive semidefinite. This update exactly minimizes W^{joint} over feasible rank-c message projectors for fixed (z, U), and R_{ga} is independent of V_a, so no further GD refinement is needed for V_a.

Algorithm (EM).

```
initialize event labels with either K-means or spectral clustering (Section 3.3)
initialize U from C^R; apply the exact conditional V update
repeat at most 100 outer iterations:
    M_U: compute spectral U; refine on R+W (Section 3.2)
    M_V: apply the exact top-eigenspace update above
    accept only finite, orthonormal, non-increasing blocks
    E: set z_new[g] = argmin_a {R[g,a]+W[g,a]} for every event g
    if z_new == z and relative outer decrease <= 1e-6: return
```

After training, our EM algorithm assigns each TEST event by $\arg\min_a R_{ga} + W_g^{\text{joint}}(a)$. No U_a, V_a projector is refit on TEST, but their performance is evaluated on the test set based on the label assignment.

3.2 Grassmann GD refinement

Our EM algorithm calls this refinement step only on the U_a blocks – each V_a block already has the exact update. For a relation-frame family U_a with k_a columns, obtain the Euclidean gradient G_a of the stated complete objective by automatic differentiation and form the tangent gradient

$$T_a = G_a - U_a(U_a^\top G_a).$$

Starting at $\delta = 10^{-2}$, update

$$U_a' = \operatorname{qr}\left[U_a - \delta\sqrt{k_a}\frac{T_a}{\max(\|T_a\|_F, 10^{-30})}\right]$$

with fixed-sign thin QR. We do backtracking line search on the stepsize δ at each iteration. The previous iterate is retained if no proposed U_a' lowers the complete objective.

3.3 Initialization of EM

3.3.1 PCA/K-means initialization

We started with a simple K-means intialization using PCA coordinates, where we independently center event query q_g and the pair key with the event summary

$$\bar{k}_g = \frac{\sum_{j>0}\alpha_{gj}k_{gj}}{\sum_{j>0}\alpha_{gj}},$$

retain the first three PCA coordinates for each, and concatenate the six coordinates.

With the constructed 3-q plus 3-$\bar{k}$ coordinates over the TRAIN events, run K-means for 20 iterations. A diagnostic that appended three PCA coordinates of the event innovation write tests whether OV augmentation improves this initializer – we did not observe significant changes in any fitted method.

3.3.2 Spectral subspace clustering initialization

The second initializer compares query events through a kernel rather than a centroid – this turns out to capture the "subspace" alignment much better than Euclidean distance. It uses 2 reductions for saving computational cost:

- **Select an anchor bank of TRAIN events:** For each TRAIN context b, rank its active query events by the label-blind non-BOS innovation-write energy

  $$E_g = \sum_{j>0}\alpha_{gj}^2\|\widetilde{\mu}_{gj}\|^2$$

  and retain at most 6 events. Only these anchors enter the TRAIN spectral kernel below, whose cost is quadratic in the number of events. This gives the anchor set – the remaining TRAIN events are assigned labels afterward by Nyström extension, as described below, and then participate normally in EM.

- **Compress the source row of each represented event:** Let J_g be the 8 non-BOS sources with largest exact attention for event g (or all of them if fewer than 8 exist). This is the only source-level truncation used to construct T_g and K_T below. Define

$$\pi_{gj} = \frac{\alpha_{gj}^2}{\sum_{t>0} \alpha_{gt}^2}, \qquad \dot{k}_{gj} = k_{gj} - \frac{\sum_{t>0} \alpha_{gt} k_{gt}}{\sum_{t>0} \alpha_{gt}}.$$

  where the centering mean in $\dot{k}_{gj}$ and weight denominator in π_{gj} still use the complete non-BOS row. Thus the retained weights need not sum exactly to one. Empirically though, $\sum_{j \in J_g} \pi_{gj} > .99$ in every head. The centering is just to standardize all the events – it's non-material in attention since a constant shift for all j doesn't change the result of the softmax operation.

Compare two events. We propose to summarize event g by the joint key–message second moment tensor

$$T_g = \sum_{j \in J_g} \pi_{gj} (\dot{k}_{gj} \dot{k}_{gj}^\top) \otimes (\widetilde{\mu}_{gj} \widetilde{\mu}_{gj}^\top).$$

Each retained source anchor in J_g contributes one descriptor that records its centered key direction together with its innovation-message direction. Then we define a kernel that computes the inner product between 2 events g and h as:

$$K_T(g,h) = \sum_{j \in J_g} \sum_{t \in J_h} \pi_{gj} \pi_{ht} (\dot{k}_{gj}^\top \dot{k}_{ht})^2 (\widetilde{\mu}_{gj}^\top \widetilde{\mu}_{ht})^2 = \langle T_g, T_h \rangle.$$

This intuitively says that one cross-event source pair contributes strongly only when both its centered keys and its innovation messages are similar. Requiring compatible queries as well gives

$$K_{\text{raw}}(g,h) = (q_g^\top q_h)^2 K_T(g,h), \qquad \text{and} \qquad K(g,h) = \frac{K_{\text{raw}}(g,h)}{\sqrt{K_{\text{raw}}(g,g) K_{\text{raw}}(h,h)}}$$

after normalization. Equivalently, the resulting kernel K is the normalized inner product of the tensor feature $(q_g q_g^\top) \otimes T_g$. We use this as a positive-semidefinite similarity over cached events.

Spectral clustering on anchors[2], then initialize every TRAIN event. Let the N anchor events be $g_1, \ldots, g_N$ and write $K_{ih} = K(g_i, g_h)$. Regard these N nodes as a weighted graph: event g_i is a node and K_{ih} is the strength of its edge to event g_h. Set $K_{ii} = 0$ because an event's self-similarity does not reveal a cluster. Define the degree-normalized affinity matrix M as

$$d_i = \sum_h K_{ih}, \qquad D = \text{diag}(d_1, \ldots, d_N), \qquad M = D^{-1/2} K D^{-1/2}.$$

The normalization discounts nodes that are broadly similar to many events, so the embedding depends on each event's pattern of neighbors rather than only its total similarity. Let $(\rho_\ell, u_\ell)_{\ell=1}^A$ be the leading A eigenpairs of M. Anchor g_i receives the spectral coordinate

$$y_i = (u_1(i), \ldots, u_A(i)), \qquad x_i = y_i / \|y_i\|.$$

Events with similar graph neighborhoods have similar $x_i \in \mathbb{R}^A$. Run K-means on these N nodes in $\mathbb{R}^A$. Its centers $c_1, \ldots, c_A$ and anchor labels are frozen after this.

[2]The introduction of an anchor set is only to reduce computational cost.

For a non-anchor TRAIN event x, compute only the vector of similarities to the anchors,

$$k_i(x) = K(x, g_i), \qquad d_x = \sum_i k_i(x).$$

Nyström extension places x in the already-fitted spectral basis without recomputing:

$$y_\ell(x) = \frac{1}{\rho_\ell} \sum_{i=1}^{N} \frac{k_i(x)}{\sqrt{d_x d_i}} u_\ell(i), \qquad x_{\text{spec}} = \frac{(y_1(x), \ldots, y_A(x))}{\|(y_1(x), \ldots, y_A(x))\|}.$$

Assign the event to $\arg\min_a \|x_{\text{spec}} - c_a\|^2$ – this allows us to get labels for all TRAIN events.

Importantly, we also use this initialization step as a discovery tool for the choice of A: we compute the leading 5 eigenpairs of M, nominate $A \in \{1, \ldots, 4\}$ using the largest adjacent eigengap $\rho_A - \rho_{A+1}$, and construct the corresponding partitions from the nominated first A eigenvectors to evaluate the performance of this initializer.

4 Experimental Results

All implemented fits use the fixed capacity $A \in \{1, 2\}, r = 32, c = 64$ below.

4.1 Evaluation Metrics

The *pooled* model error used in result tables is

$$\overline{W}^{\text{joint}} = \frac{\sum_g \|\widetilde{\nu}_g - \sum_{j>0} \widehat{\alpha}_{gj} P^M_{z(g,j)} \widetilde{\mu}_{gj}\|^2}{\sum_g \|\widetilde{\nu}_g\|^2},$$

It differs from the equal-event, event-normalized objective term W_g^{joint} introduced earlier. Additionally, compared to $\overline{W}^{\text{joint}}$, full-write $\text{NMSE}_{\text{pool}}$ reported below restore BOS and P^0 projection (fit once on the TRAIN) before comparison.

Adjusted Rand index (ARI)[3] is permutation invariant and chance corrected for measuring accuracy of the predicted labels – an ARI of 1 means perfect agreement, where ARI ≈ 0 means it's no better than random grouping, and negative ARI implies the grouping is worse than chance. Permutation accuracy is the best-component-matching classification accuracy but can be misleading for L1H1 because newline is rare – this is corrected by

$$\text{balanced accuracy} = \frac{\text{surface-level composition recall} \;+\; \text{newline recall}}{2}$$

in our reporting for label diagnostics. L5H2 exact/tolerant ARI is only an exploratory diagnostic: it asks whether a forced $A = 2$ split happens to separate these subtypes. A low ARI does not count as failure under the $A = 1$ hypothesis.

If X_a and X_b are orthonormal bases for 2 equal-rank subspaces, their overlap is $\|X_a^\top X_b\|_F^2 / \operatorname{rank}(X)$, where zero and one mean orthogonal and identical subspaces.

We perform one coupling control: with labels and U_1, U_2 frozen, exchange V_1, V_2 and recompute TEST $\overline{W}^{\text{joint}}$. A positive swapped vs. matched gap shows that the observed relation partition is paired with different message operations.

[3] https://en.wikipedia.org/wiki/Rand_index

4.2 EM with K-means initialization enters a nonsemantic basin

With K-means initialization followed by EM, the $A = 2$ model gives lower TRAIN error for both heads, as extra capacity permits, but that gain does not transfer into an interpretable split, or TEST set evidence supporting it:

head	A	TEST $\bar{W}^{\text{joint}}$	full-write $\text{NMSE}_{\text{pool}}$	event ARI	interpretation
L1H1	1	.3855	.1200	–	pooled
L1H1	2	.3924	.1213	.0023	two word-like partitions
L5H2	1	.6032	.3722	–	pooled induction
L5H2	2	.6278	.4254	−.0364	unstable subtype split

Table 2: EM with $\text{Q}/\bar{K}$ PCA + K-means initialization.

L1H1's two learned subspaces overlap .662 on the relation side and .804 on the message side; both units' strongest examples are word-piece composition. At this initialization, the model supports L5H2 $A = 1$ but does not reveal L1H1's suspected second unit.

4.3 The rare L1H1 operation is present but not selected by K-means

To understand the reason behind this failure, we note that L1H1 newline comprises 6.10% of TRAIN events and only 1.05% of TRAIN exact pair-write energy; TEST values are 7.36% and 1.29%. These are just a reflection of natural corpus where one type occurs more often than the other. PCA/K-means therefore spends its second center on a broad variance direction, rather than capturing the rarer class accurately. The following 3D PCA plot makes the failure of the K-means initialization clearer:

Figure 4: Geometry used by K-means initializer on TRAIN set. Each event is represented by 3 TRAIN-fit query PCs and 3 TRAIN-fit PCs of its attention-weighted non-BOS key mean. The resulting 6 coordinates are projected onto 3 further PCs for display. Left: post-hoc annotated behavior. Right: $A = 2$ K-means labels and centers. Top: L1H1 newline events are localized, but K-means bisects the much larger surface cloud instead. Bottom: L5H2's two induction subtypes overlap significantly.

initializer/diagnostic	split	ARI	balanced accuracy
Q/K K-means	TRAIN	.0084	.5553
Q/K K-means (frozen centers from TRAIN)	TEST	.0006	.5039
known-behavior centroids in the same 6 coordinates	TRAIN	.3084	.9168

Table 3: L1H1 initialization geometry. The centroid diagnostic (last row) uses known labels to construct the centroid (in the same 6 coordinates) for assignment and is not a realistic initializer.

The geometry therefore contains separable surface-leve- composition/newline information, but unsupervised K-means does not choose it. For L5H2, the plot does not support an unsupervised strict/tolerant split, which is consistent with the one-unit hypothesis. These experiments suggest initialization is a bottleneck for this nonconvex objective that EM is targeting.

4.4 Label-blind spectral discovery

After the spectral clustering initialization step (with the product kernel), we observe the following result that largely succeed in identifying the latent groups in this geometry: the TRAIN ARI is .966 for L1H1 with $A = 2$, whereas forcing $A = 2$ on L5H2 merely isolates 2 TRAIN outlier events (with TRAIN ARI $-.004$), and the second component received no TEST event when we assign the TEST events via Nyström extension in the kernel geometry.

For an experiment testing the optimal A to use with this geometry, let $\rho_1 \geq \rho_2 \geq \cdots$ denote the eigenvalues of the normalized affinity matrix and define

$$\Delta_A = \rho_A - \rho_{A+1},$$

Our exploratory rule chooses the largest Δ_A for $A \in \{1, 2, 3, 4\}$:

head	Δ_1	Δ_2	Δ_3	Δ_4	nominated A
L1H1	.00395	**.03122**	.00565	.00192	2
L5H2	**.02999**	.02453	.00497	.00538	1

Table 4: Product-kernel eigengaps. This rule nominates a choice on A based on spectral info.

This supports our earlier hypothesis and offers support for using the product-kernel induced geometry for initialization.

4.5 Spectral initialization for EM

We observe that using spectral labels as the start on the TRAIN set, followed by regular EM, materially changes the result. On TEST set we report 2 assignments: the *joint* assignment minimizes $R_{ga} + W_g^{\text{joint}}(a)$ and measures how well the fitted components can reconstruct an event; the *Q/K-only* assignment minimize R_{ga} – it is a message-blind router that uses only the relation-side for label assignment.

head	A	TRAIN initializer	joint TEST $\bar{W}^{\text{joint}}$	joint full-write $\text{NMSE}_{\text{pool}}$	joint ARI	Q/K ARI
L1H1	1	single unit	.3855	.1200	–	–
L1H1	2	Q/K K-means	.3924	.1213	.0023	–
L1H1	2	spectral	**.3720**	**.1140**	.9258	.9972
L5H2	1	single unit	**.6032**	**.3722**	–	–
L5H2	2	spectral	.6040	.3756	−.0184	.0120

Table 5: EM TEST results with different initializers.

L1H1's spectral-start fit improves TEST reconstruction over $A = 1$, with relation/message overlaps .247/.438 instead of .662/.804 in the K-means basin (which loses the surface-leve composition/newline distinction). The spectral start followed by EM recovers the two behaviors under both joint and Q/K-only routing, and gives the lowest TEST errors. Initialization is therefore decisive here: the same EM objective contains an interpretable $A = 2$ solution but K-means does not reach it.

For L5H2, spectral initialization does not create useful evidence for a second unit. The extra unit receives only 2.1% of TEST events, does not improve either reconstruction error, and does not align with the two induction subtypes. These results favor $A = 1$ over the tested $A = 2$ split. However, the performance gap is mild in this experiment – we suspect it has to do with the choice of (r, c) here and the current projector model reconstructs L5H2 less faithfully compared to L1H1.

We conducted one fixed-label control checking that L1H1's relation and message components are genuinely paired. Keeping the labels and U_a fixed but swapping the two V_a raises TEST $\overline{W}^{\text{joint}}$ from .373 to .654 (a .281 penalty). This supports distinct surface-level composition and newline relation–message couplings.

4.6 Interpreting the discovered unit

4.6.1 Activated ranks

Our algorithm fixes, rather than selects, the frame dimensions: every U_a has rank 32 and every innovation V_a has rank 64. Below we report the smallest usage rank $r_{90}(X)$ carrying 90% of the squared singular-value energy of X.

Relation usage is measured on $q_g^\top U_a$ and on stacked rows $\sqrt{\alpha_{gj}/\sum_{t>0}\alpha_{gt}}\, k_{gj}^\top U_a$. Message usage is measured on the α_{gj}^2-weighted source coordinates $\widetilde{\mu}_{gj}^\top V_a$.

split	head/unit	events	$r_{90}(qU)$	$r_{90}(kU)$	$r_{90}(\widetilde{\mu}V)$
TRAIN	L1H1 surface composition	2,938	17	8	51
TEST	L1H1 surface composition	2,806	17	8	50
TRAIN	L1H1 newline	194	11	2	8
TEST	L1H1 newline	224	8	2	7
TRAIN	L5H2 induction	712	18	19	51
TEST	L5H2 induction	725	16	18	48

Table 6: Activated ranks of the spectral-start EM fits. Stored projector ranks are exactly $r = 32, c = 64$ in every row. TEST labels use R_{ga} only for assignment.

We see that TRAIN/TEST usage ranks agree closely. The newline operation uses few innovation directions, while surface-level composition and induction require many more innovation directions at the 90%-energy threshold, which is intuitive.

4.6.2 Context-specific causal interpretation of V_a

The fitted V_a are early-layer message-operation subspaces, not token labels or directly readable vocabulary directions. Its interpretable object in event g is the projected innovation write. To interpret them causally, at a selected TEST event choose the unit without messages:

$$a_g^{QK} = \arg\min_a R_{ga},$$

and compare the exact and projected innovation coordinates

$$h_g^{\text{exact}} = \widetilde{\nu}_g, \qquad h_g^{\text{msg}} = P^M_{a_g^{QK}} h_g^{\text{exact}} = \sum_{j>0} \alpha_{gj} P^M_{a_g^{QK}} \widetilde{\mu}_{gj}.$$

These h's are coordinates in the lossless message basis; $B_M h$ is the corresponding residual-stream vector. Let y_g^{exact} be the unedited summed post-O, pre-post-attention-RMSNorm output, and let $\Pi_g(y)$ be the next-token probability vector obtained by continuing the unchanged model from hook value y (i.e., pass through all subsequent layers till the final one). Define

$$\begin{aligned}
y_g^{\text{ablated}} &= y_g^{\text{exact}} - B_M h_g^{\text{exact}}, \\
p_g^{\text{ablated}} &= \Pi_g(y_g^{\text{ablated}}), \\
p_g^{\text{exact}} &= \Pi_g(y_g^{\text{ablated}} + B_M h_g^{\text{exact}}) = \Pi_g(y_g^{\text{exact}}), \\
p_g^{\text{msg}} &= \Pi_g(y_g^{\text{ablated}} + B_M h_g^{\text{msg}}).
\end{aligned}$$

Thus ablation removes only this head's exact innovation, whereas message-only replaces it by the fitted V_a projection. Exact attention, BOS, the frozen common span, other heads, and every later

layer are unchanged. At the same query's next-token readout, we record

$$\Delta p_g^{\text{exact}} = p_g^{\text{exact}} - p_g^{\text{ablated}},$$

which is the downstream effect of the actual innovation, and

$$\Delta p_g^{\text{msg}} = p_g^{\text{msg}} - p_g^{\text{ablated}},$$

which is the effect preserved by the chosen V_a approximation. For vocabulary token t, the sign of $\Delta p_{g,t}^{\text{msg}}$ says whether the projected write promotes or suppresses t relative to ablation. This is a context-dependent downstream effect, not a direct unembedding or a fixed token meaning for V_a. Cosine similarity/NMSE between Δp_g^{msg} and $\Delta p_g^{\text{exact}}$ measures how well the projected write reproduces the original innovation's full-vocabulary effect in that event.

4.6.3 Representative examples

The following TEST examples illustrate what the discovered U_a, V_a (fit on the TRAIN set) mean in complete local contexts. The attention masses below are reported as exact/fitted $\alpha_{g,s}/\widehat{\alpha}_{g,s}$ for the key s of interest, and final-model next-token effects are obtained by the intervention described above (which use exact attention so it isolates the V_a part).

L1H1 surface-level composition unit

- **alphabetic words:** In "we investigate the subcortical connections of the PPHG ... to characterize thalamic and striatal connections," the query piece `ic` in `thalamic` attends to `thalam` with mass .975/.985; both the exact and fitted innovations suppress the immediate `and` continuation and favor a hyphen-like alternative. The cosine/NMSE between Δp_g^{msg} and $\Delta p_g^{\text{exact}}$ is .982/.035.
- **four-digit year-like:** In "Patent Laid-Open No. 2010-282816, paragraphs 0017 and 0018," the last piece of `0017` attends mainly to its earlier zero and both writes favor `to` over a bare hyphen. The cosine/NMSE between Δp_g^{msg} and $\Delta p_g^{\text{exact}}$ is 1.000/.067.
- **other numbers:** In the French financial sentence ending "tel qu'exposé à la note 1.2.1 aux Etats financiers," the last `1` retrieves an earlier numeric piece, and both writes favor number punctuation while suppressing nearby French function-word continuations. The cosine/NMSE between Δp_g^{msg} and $\Delta p_g^{\text{exact}}$ is .954/.090.
- **identifiers:** In C++ binding code containing "hb_parni(2); obj->scrollTo(*par1, ...)," `ni` attends to `par` with mass .990/.958; the exact write favors `();` and an opening parenthesis, while the fitted write preserves much of the function-call punctuation. The cosine/NMSE between Δp_g^{msg} and $\Delta p_g^{\text{exact}}$ is 0.984/.052.

L1H1 newline unit. In "How do I learn to do the butterfly stroke? [blank line] Assistant: I'd be happy to assist you," the blank line attends to the initial dialogue boundary with mass .864/.869 and the two writes again agree nearly exactly, this time favoring `Okay`, `The`, and `This` while suppressing a bold-list opener. The low-rank unit carries a reusable boundary/formatting state; it does not promote one fixed token independent of the surrounding discourse.

L5H2 induction unit: exact matches. In the Dutch sentence "... onzen penningmeester per postwissel worden voldaan," a repeated query `w` attends to the earlier continuation `issel` with mass .967/.508, and both exact and fitted write directly promote `issel`. The cosine/NMSE between Δp_g^{msg} and $\Delta p_g^{\text{exact}}$ is .991/.188.

L5H2 induction unit: tokenization-tolerant matches. In a hardware discussion containing both `Nvidia NForce` and the misspelled `nfroce 6100`, the query ` n` attends to the earlier capitalized `Fro` with mass .947/.848; both writes favor the capitalized continuation, although they disagree on the cached lower-case piece. The cosine/NMSE between Δp_g^{msg} and $\Delta p_g^{\text{exact}}$ is .999/.037.

Therefore we see that the fitted components align with and reproduce known relation patterns on the hold-out set: L1H1's U_{surf} routes later pieces of words, numbers, and identifiers to their earlier pieces; U_{nl} routes a newline query to a preceding paragraph, dialogue, or record boundary. L5H2's single U approximates induction: after a prefix recurs, the query attends to the continuation following its earlier occurrence. Strict and tokenization-tolerant matches differ in recognizing the prefix, not in the subsequent continuation write.

5 Parting Thoughts

5.1 Summary and next steps

The current evidence supports the following conclusion: at fixed $(r, c) = (32, 64)$, a label-blind spectral initialization lets the latent event model recover a L1H1 surface/newline decomposition that transfers to source-disjoint TEST on this corpus; the same procedure leaves L5H2 effectively one-unit. Ordinary K-means misses the rare L1H1 basin. This is evidence for a useful candidate relation split that supports reusable relation–message decomposition.

As for next steps and things to improve on:

- Although replying on spectral gap to select A seems reasonably stable, we need better automatic ways of tuning r, c – there is evidence suggesting the current choice isn't optimal.
- Ultimately the mechanisms we can discover depend on the prompts that we test them on (i.e., we can't discover unknown unknowns).
- Generalize the same methodology to layer-global shared U_a, V_a so one can understand distributed reusable mechanisms across heads in a layer.
- Inductive bias on U_a, V_a may make the geometry better for optimization. We've tried several low-rankness type of regularization but getting things to reliably work seems tricky.
- There are considerable degrees of freedom in the losses we can run the EM on (and we've tried out a few in addition to the ones reported here). But importantly, we are handcrafting metrics/objectives for clustering – one could imagine learning a mapping directly from the cached quantities (or transformations thereof) to latents/mechanisms, if we have some labeled data of such type.

5.2 Earlier Attempt: learned rank-one pair co-clustering

We embarked on this journey with a slightly different angle – while we haven't got this approach to work, it might be worth recording here to possibly revisit later.

Motivated by randomized-kernel feature map as proposed in [1], one can "linearize" the nonlinear softmax kernel as:

$$\exp(q_i^\top k_j) \approx \phi(q_i)^\top \phi(k_j)\,.$$

Generalizing the idea, this trick allows us to record each relationship-message pair/edge in Figure 1 as a rank-1 matrix, and as we'll see, the problem in some sense becomes bi-clustering on these matrices when aggregated over all the pairs in a batch of contexts. The elegance of this approach is that we turned a bilinear operation into a simpler matrix-valued object (without losing much information) and we originally hoped that simple matrix factorization techniques can help us identify reusable coupled relationships from them.

To allow flexibility, instead of using fixed feature map ϕ as advocated in [1], we learn encoders that produce nonnegative query and key features. For each target head, train one nonnegative Q/K map tied between Q and K:

$$f_\theta(x) = \mathrm{TopK}_{64}\{\mathrm{softplus}(W_\theta x + b_\theta)\} \in \mathbb{R}^{512}_{\geq 0}.$$

On that head's annotated active TRAIN event, for a non-BOS pair $n = (g, j) = (b, i, j)$, define

$$\alpha^\theta_{bij} = \frac{f_\theta(q_g)^\top f_\theta(k_{gj})}{\sum_{t\leq i}[f_\theta(q_g)^\top f_\theta(k_{gt})]}$$

as the learned attention between query i and key j and minimize mean row-wise $\mathrm{KL}(\alpha_{bi,:} \| \alpha^\theta_{bi,:})$, after which we freeze f_θ. This allows us to rewrite the softmax attention as an inner product of learned relation-side features f_θ (up to a normalization). Let

$$p_{gj} = \frac{f_\theta(q_g) \odot f_\theta(k_{gj})}{\sum_{t\leq i}[f_\theta(q_g)^\top f_\theta(k_{gt})]},$$

be the normalized elementwise-product vector and construct the rank-1 matrix that summarize all the information in this edge (QK & OV jointly):

$$X_{gj} = p_{gj}\widetilde{\mu}_{gj}^\top.$$

In matrix X_{gj}, high value at index (m_1, m_2) implies that the head implements an algorithm that searches for matching on the m_1-th relation feature which writes to the m_2-th message atom. Feature-side frame $U^p_a \in \mathbb{R}^{512\times r}$ and message-side frame V_a reconstruct X_{gj} as $P^p_a X_{gj} P^M_a$, where $P^p_a = U^p_a U^{p\top}_a$. We can proceed to fit

$$J_{\mathrm{pair}} = \sum_{g,j>0} \|X_{gj} - P^p_{z_{gj}} X_{gj} P^M_{z_{gj}}\|^2_F$$

with iterative EM steps on the z's and $\{U_a, V_a\}$.

Note that the coupled output approximation is obtained directly from the reconstructed rank-one matrices:

$$\widehat{\nu}^{\mathrm{pair}}_g = \sum_{j>0} \left(\mathbf{1}^\top P^p_{z_{gj}} p_{gj}\right) P^M_{z_{gj}} \widetilde{\mu}_{gj}.$$

In our experiments, when fitting the f_θ's attention KL on the TRAIN set, we observe the same learned features struggle to transfer to hold-out set for reconstructing the attention scores. Variance of the estimators seem to be large, but we haven't thoroughly tested more powerful feature encoders.

References

[1] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking Attention with Performers. In *International Conference on Learning Representations*, 2021.

[2] R. Luger, Harish Kamath, Doug Finkbeiner, Purvi Goel, Adam Jermyn, Sam Zimmerman, Joshua Batson, and Tom Conerly. HeadVis: An Interactive Tool For Investigating Attention Heads. *Anthropic Transformer Circuits Thread*, 2026.