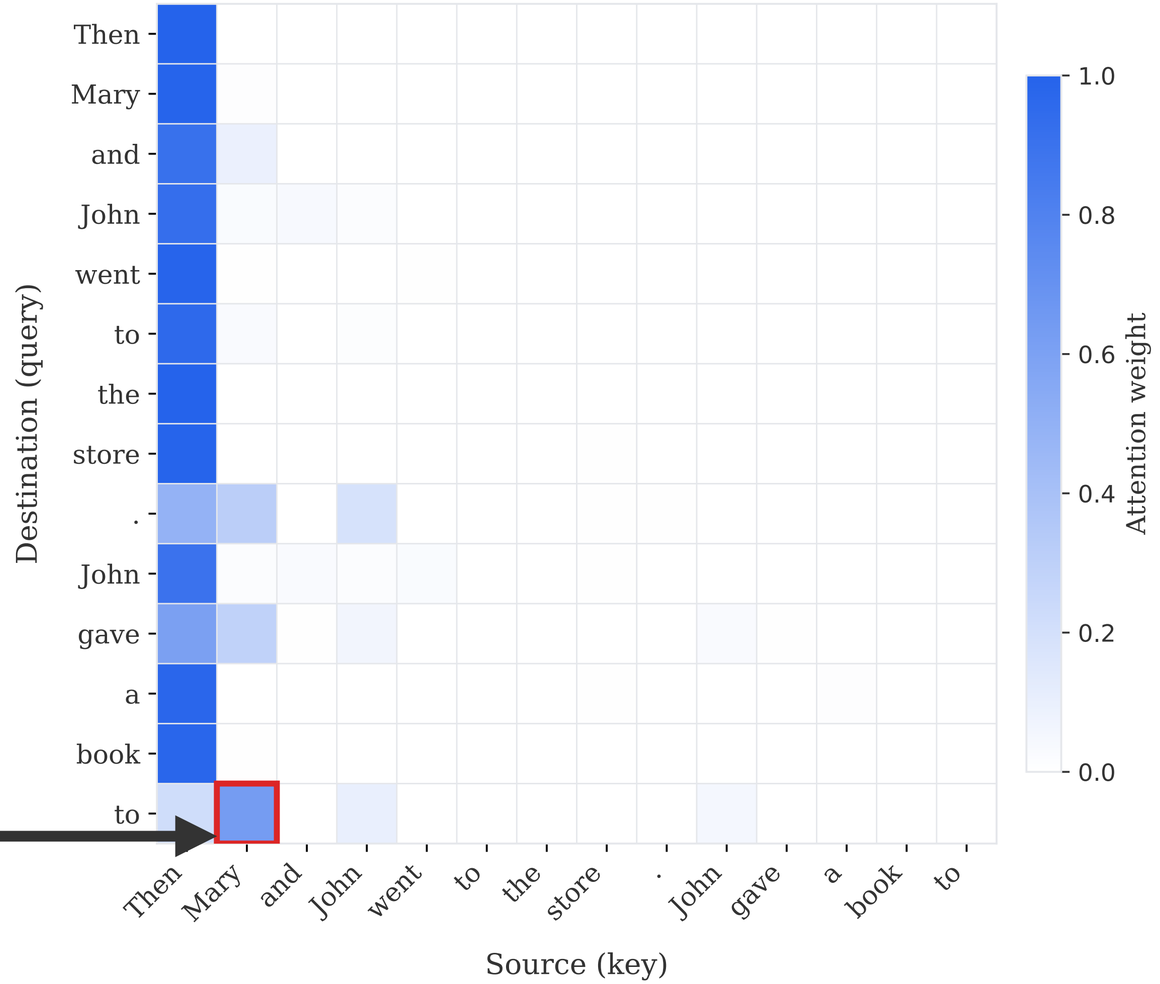
Understanding Attention from First Principles

From Singular Vectors to Interpretable Causal Circuits

Gabriel Franco
Boston University

Coefficient Giving

Why does this head attend here?



Strong attention to one token pair

Near-zero everywhere else

Why this token pair? We lack a mechanistic explanation

This talk: a unified framework to explain attention from first principles

What features in the residual stream does this head look for?

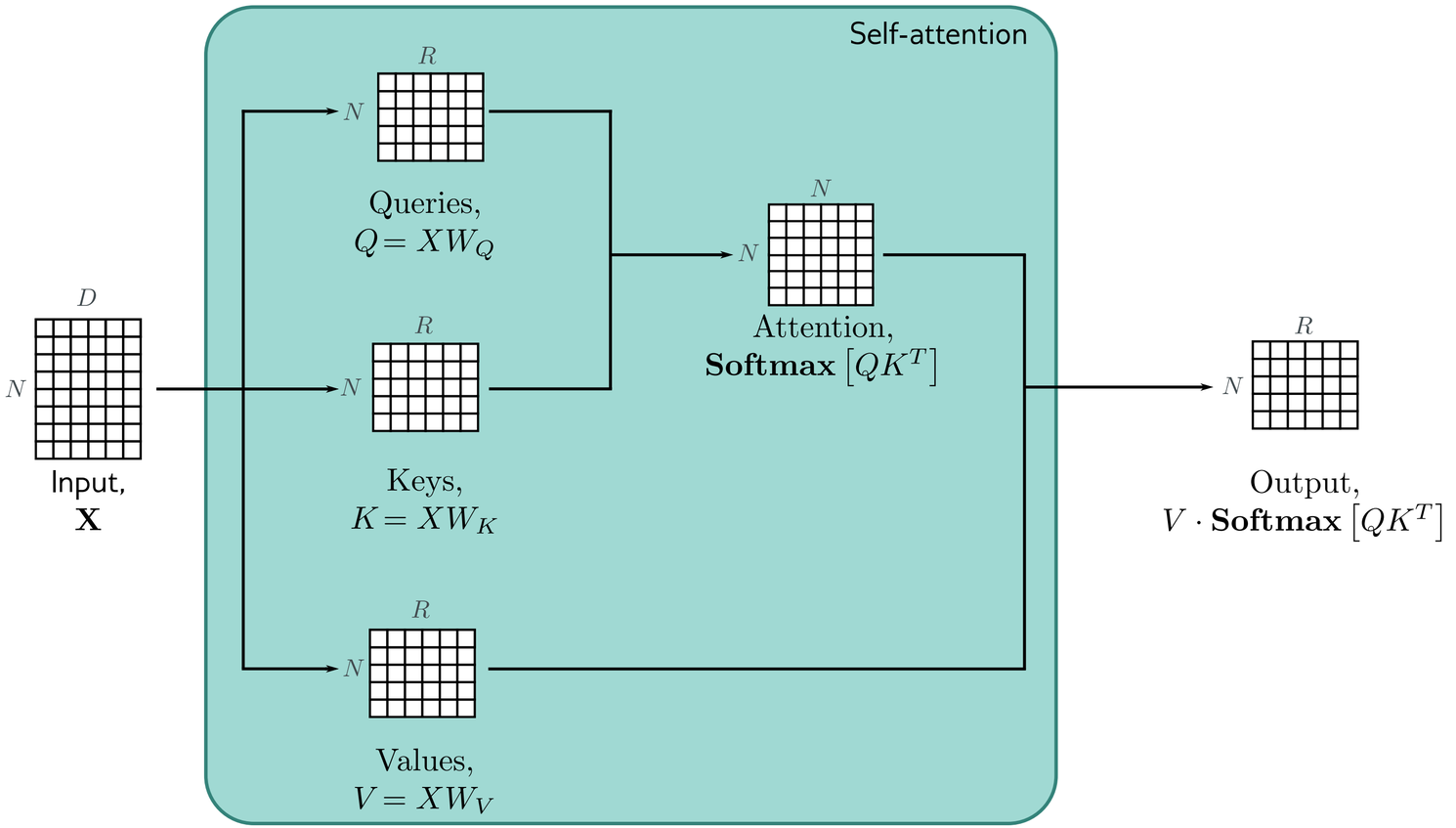
Which upstream components wrote those features? Are causally responsible?

How do these causes compose into an interpretable circuit?

What does this head look for?

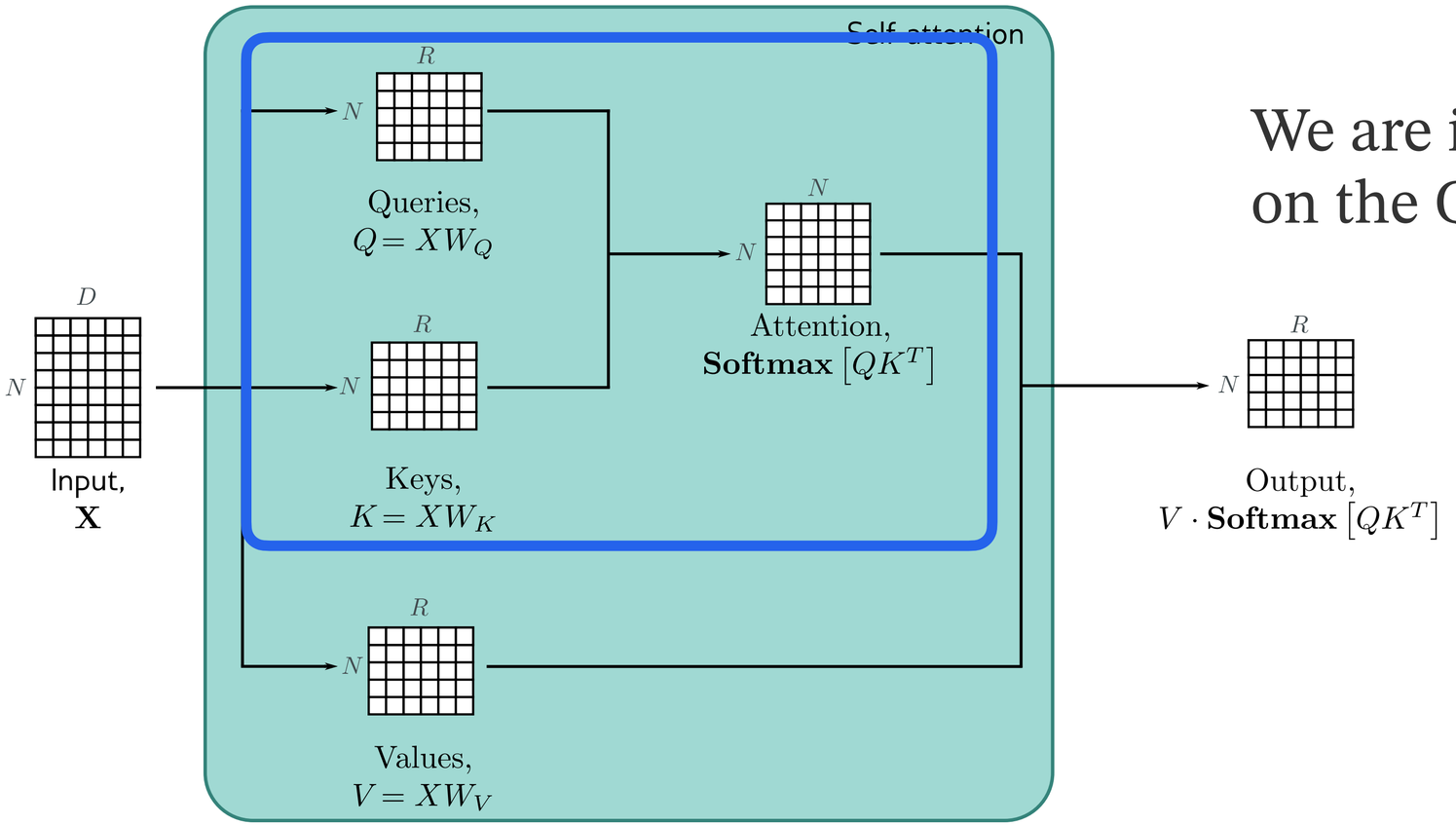
The geometry of attention and singular vector–feature alignment

Recap: the attention mechanism



Adapted from Understanding Deep Learning (Prince, 2023)

Recap: the attention mechanism



We are interested on the QK circuit

Adapted from Understanding Deep Learning (Prince, 2023)

Attention is a bilinear form over the residual stream

$$A'_{ds} = \mathbf{x}^{d\top} \Omega \mathbf{x}^{s}, \quad \Omega = W_Q W_K^{\top}$$

The score depends entirely on which directions in $\mathbf{x}^{d}$ and $\mathbf{x}^{s}$ align through Ω

Prior work decomposes Ω via SVD: why is this useful?

$$\Omega = \sum_k \mathbf{u}_k \sigma_k \mathbf{v}_k^\top$$

$$A'_{ds} = \sum_k \underbrace{(\mathbf{x}^{d\top} \mathbf{u}_k)}_{\text{query projection}} \cdot \sigma_k \cdot \underbrace{(\mathbf{v}_k^\top \mathbf{x}^s)}_{\text{key projection}}$$

Each term is a “communication channel”: active only if both tokens project onto the paired directions $\mathbf{u}_k$, $\mathbf{v}_k$

Several papers assume $\mathbf{u}_k$, $\mathbf{v}_k$ are meaningful features, but why should they be?

Merullo et al. 2024; Ahmad et al. 2025; Pan et al. 2024; Franco & Crovella 2024.

Prior work decomposes Ω via SVD: why is this useful?

$$\Omega = \sum_k \mathbf{u}_k \sigma_k \mathbf{v}_k^\top$$

$$A'_{ds} = \sum_k \underbrace{(\mathbf{x}^{d\top} \mathbf{u}_k)}_{\text{query projection}} \cdot \sigma_k \cdot \underbrace{(\mathbf{v}_k^\top \mathbf{x}^s)}_{\text{key projection}}$$

Each term is a “communication channel”: active only if both tokens project onto the paired directions $\mathbf{u}_k$, $\mathbf{v}_k$

Several papers assume $\mathbf{u}_k$, $\mathbf{v}_k$ are meaningful features, but why should they be?

Merullo et al. 2024; Ahmad et al. 2025; Pan et al. 2024; Franco & Crovella 2024.

The linear representation hypothesis: tokens as sums of features

$$\mathbf{x} = \sum_i f_i \mathbf{w}_i$$

f_i are the feature strength

$\mathbf{w}_i$ are the representation vectors/ feature directions

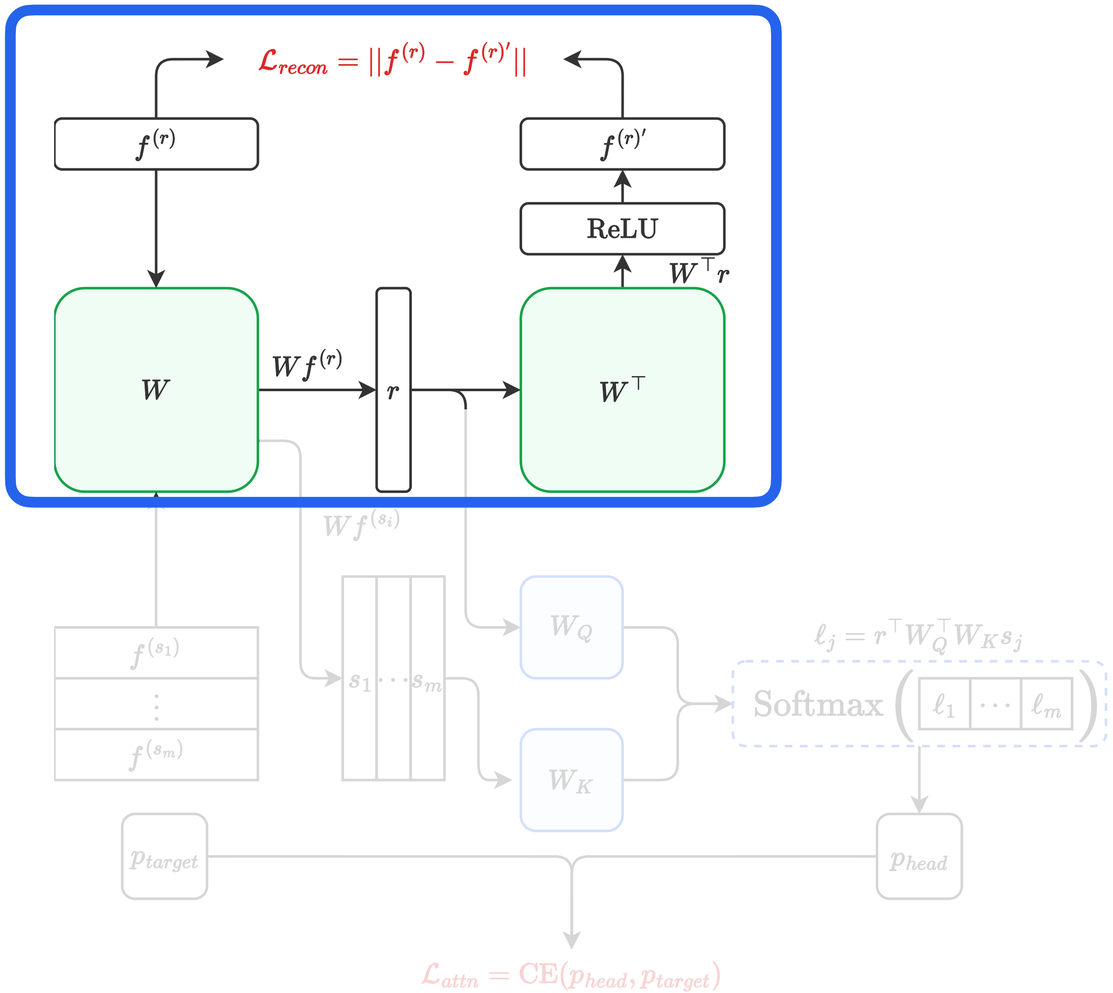
This is a core assumption underlying this work

Described by the "Toy Models of Superposition" (Elhage et al. 2022) and "How Many Features Can a Language Model Store Under the Linear Representation Hypothesis?" (Garg, Kleinberg, and Peng 2026)

A controlled setting where we can observe features directly

The input of the model is the feature strength vector:

$f^{(r)} = [0, 0.2, \cdots, 0]$



Adapted from Toy Models of Superposition (Elhage et al. 2022)

Tokens are sum of features (columns of W):

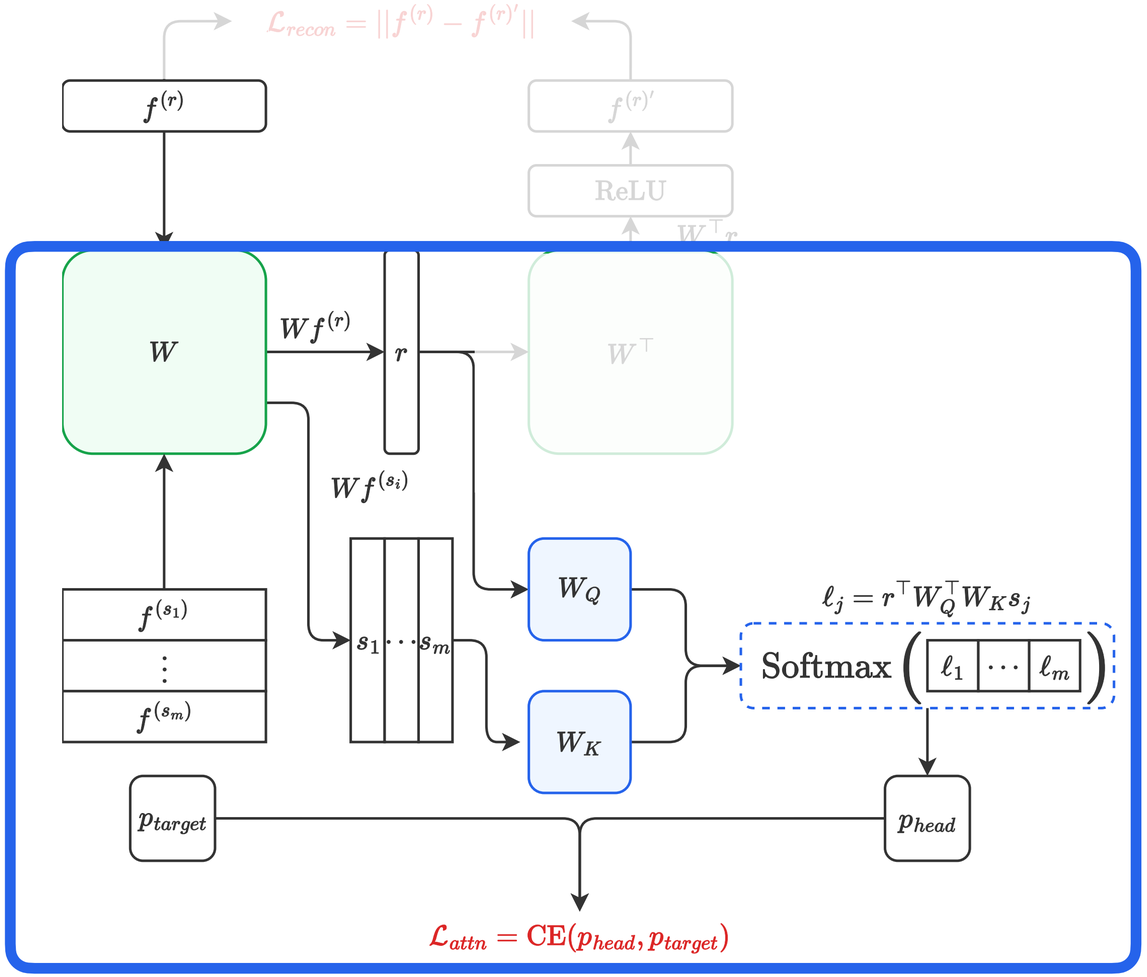
$$r = \sum_i f_i^{(r)} \mathbf{w}_i$$

The model need to learn W

A controlled setting where we can observe features directly

Now the input has extra feature strengths $f^{(s_1)}, \ldots, f^{(s_m)}$

and a target attention p_{target}



Adapted from Toy Models of Superposition (Elhage et al. 2022)

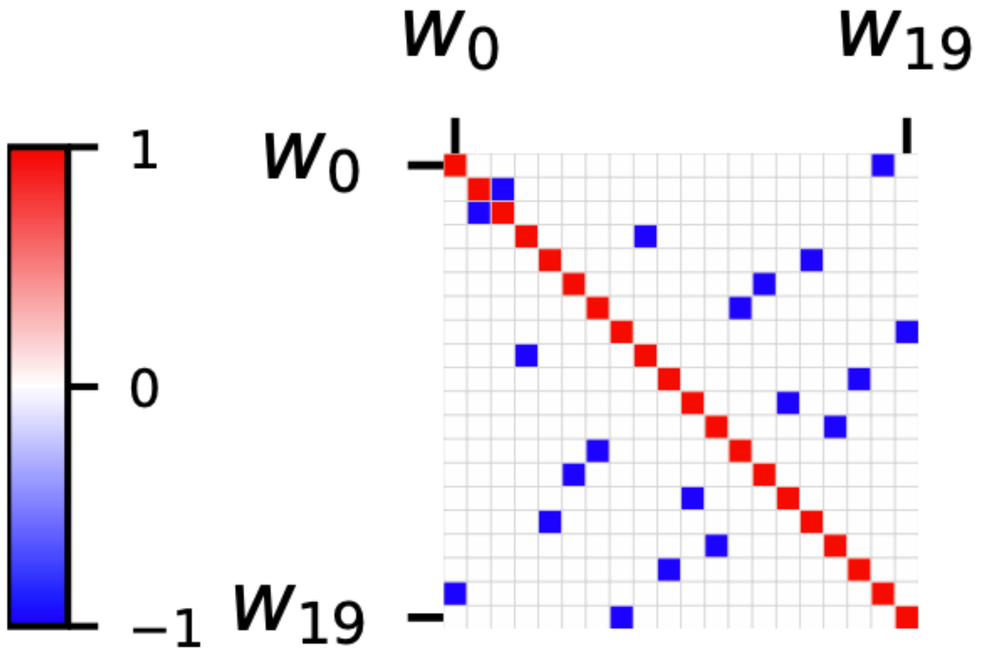
Attention is computed over a query token r and multiple source tokens $s_1, \cdots, s_m$

Combined loss:

$$\mathcal{L} = \mathcal{L}_{recon} + \lambda \mathcal{L}_{attn}$$

Baseline: without attention, features spread isotropically

Antipodal pairs minimize interference

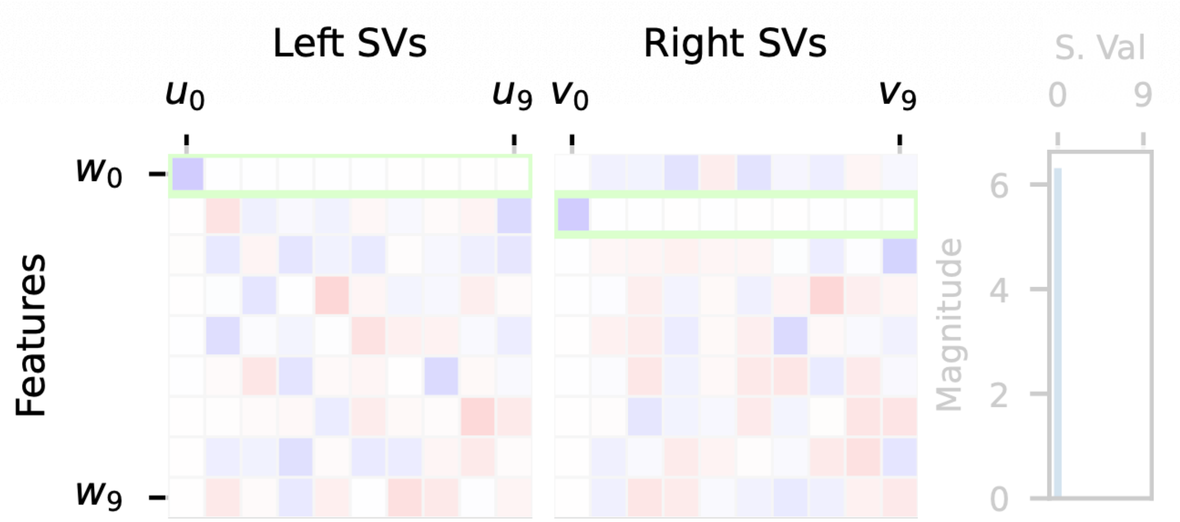


No preferred direction: features spread uniformly

Replicates Toy Models of Superposition (Elhage et al. 2022). $N = 20$ features, $D = 10$ dimensions

Adding attention: singular vectors align with features

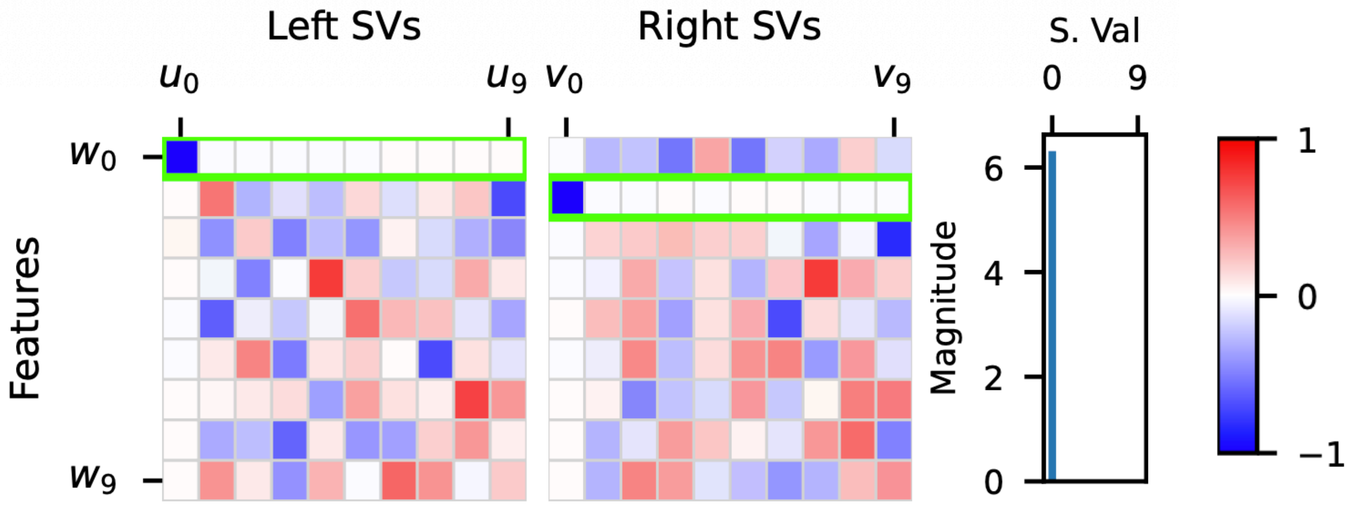
Simplest case:
$h^T \Omega s_i$ is high iff
h **contains** w_0
and
s_i **contains** w_1



Each cell = cosine similarity between a feature direction w_i and a singular vector

Adding attention: singular vectors align with features

$w_0 \leftrightarrow u_0$ *(query feature aligns with top left SV)*



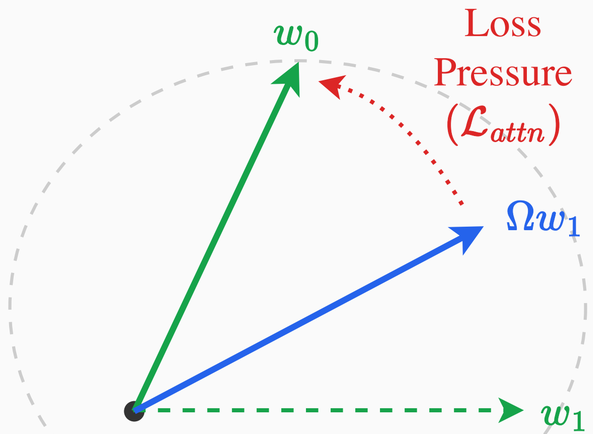
$w_1 \leftrightarrow v_0$ *(key feature aligns with top right SV)*

We refer to this as **Singular Vector Feature (SVF)** alignment

The head allocates one singular dimension entirely to the attended feature pair.
Spectrum shows a single dominant singular value

Intuition: the loss geometry pushes Ω to align with features

1. Vector Space Geometry



Attention loss pushes Ωw_1 toward w_0

2. Mathematical Consequence

To maximize the score, Ω must align the features:

$$\Omega w_1 \approx \alpha_0 w_0 \text{ and } \Omega^\top w_0 \approx \alpha_1 w_1$$

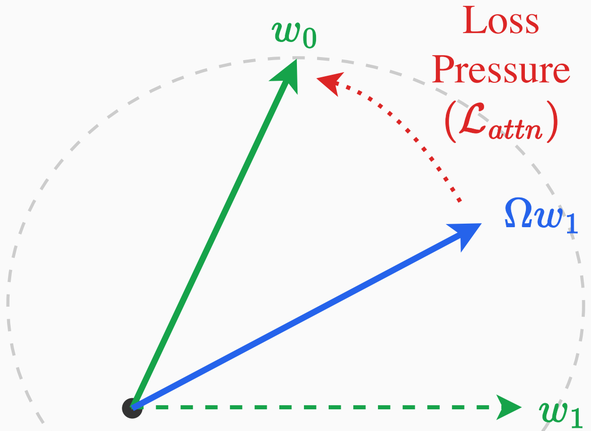
Chaining both equations gives:

$$\Omega^\top \Omega w_1 \approx \alpha w_1 \implies w_1 \approx v_0$$

This forces w_0 and w_1 to become the top singular vectors of Ω

Intuition: the loss geometry pushes Ω to align with features

1. Vector Space Geometry



Attention loss pushes Ωw_1 toward w_0

2. Mathematical Consequence

To maximize the score, Ω must align the features:

$$\Omega w_1 \approx \alpha_0 w_0 \text{ and } \Omega^\top w_0 \approx \alpha_1 w_1$$

Chaining both equations gives:

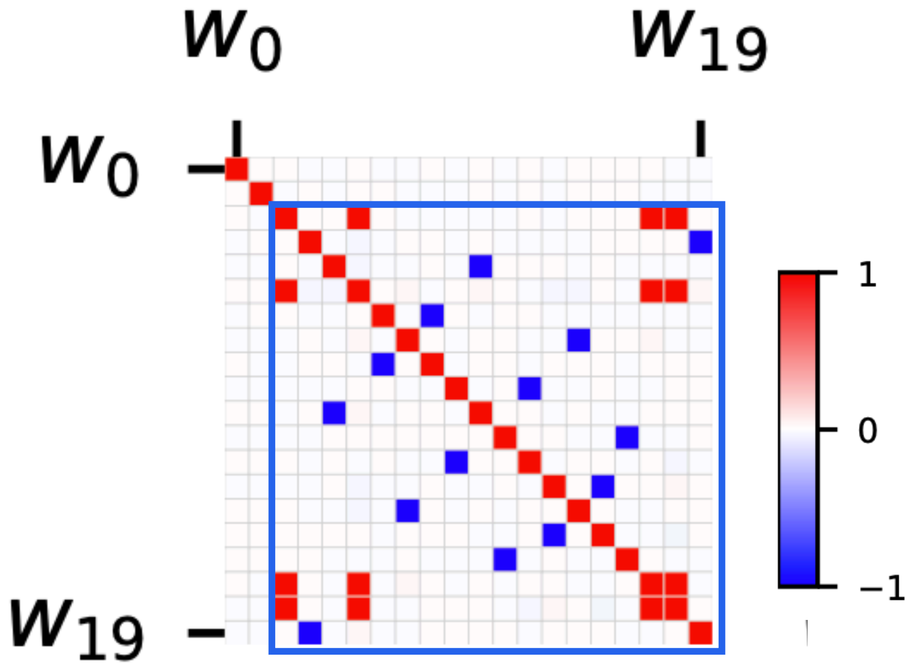
$$\Omega^\top \Omega w_1 \approx \alpha w_1 \implies w_1 \approx v_0$$

This forces w_0 and w_1 to become the top singular vectors of Ω

Proven formally in Theorems 2 & 3 (SVF paper). Holds exactly under isotropy; approximately under bounded interference

Non-attended features get pushed orthogonal to keep the reconstruction loss low

Attended features occupy a 2D subspace of their own

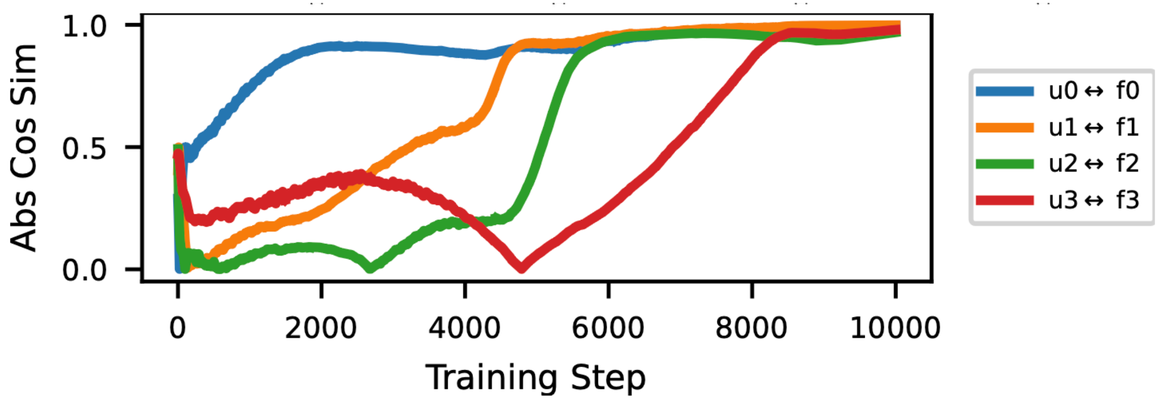


Remaining features live in the orthogonal complement, minimizing interference

Features orthogonalize without affecting attention loss because singular vectors shift to compensate

SVF alignment emerges one feature pair at a time, in order of importance

Higher-priority pairs (larger target logit) align first

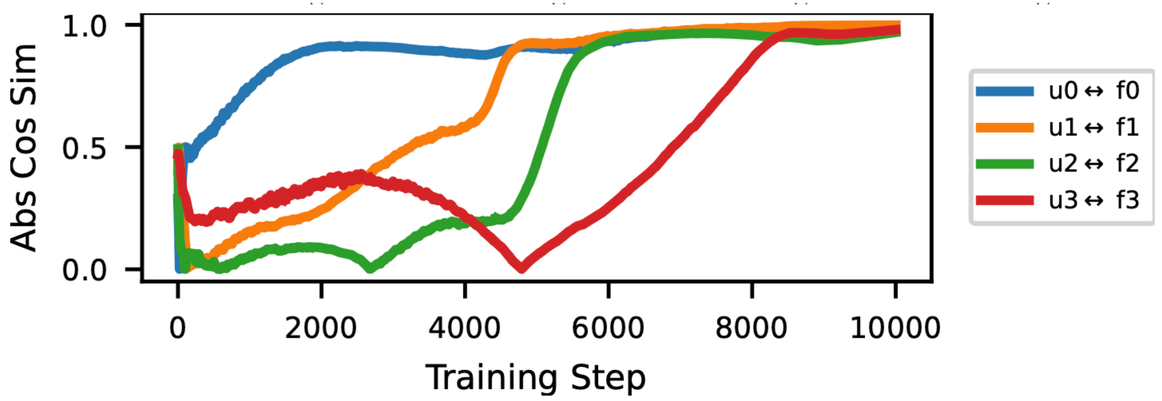


Each alignment event corresponds to a drop in attention loss

Singular vectors shift first; features then shift to follow.
Both evolve during training

SVF alignment emerges one feature pair at a time, in order of importance

Higher-priority pairs (larger target logit) align first



Each alignment event corresponds to a drop in attention loss

SVF alignment justifies using singular vectors to identify features.
We now leverage SVF to guide a search for the components responsible for writing them

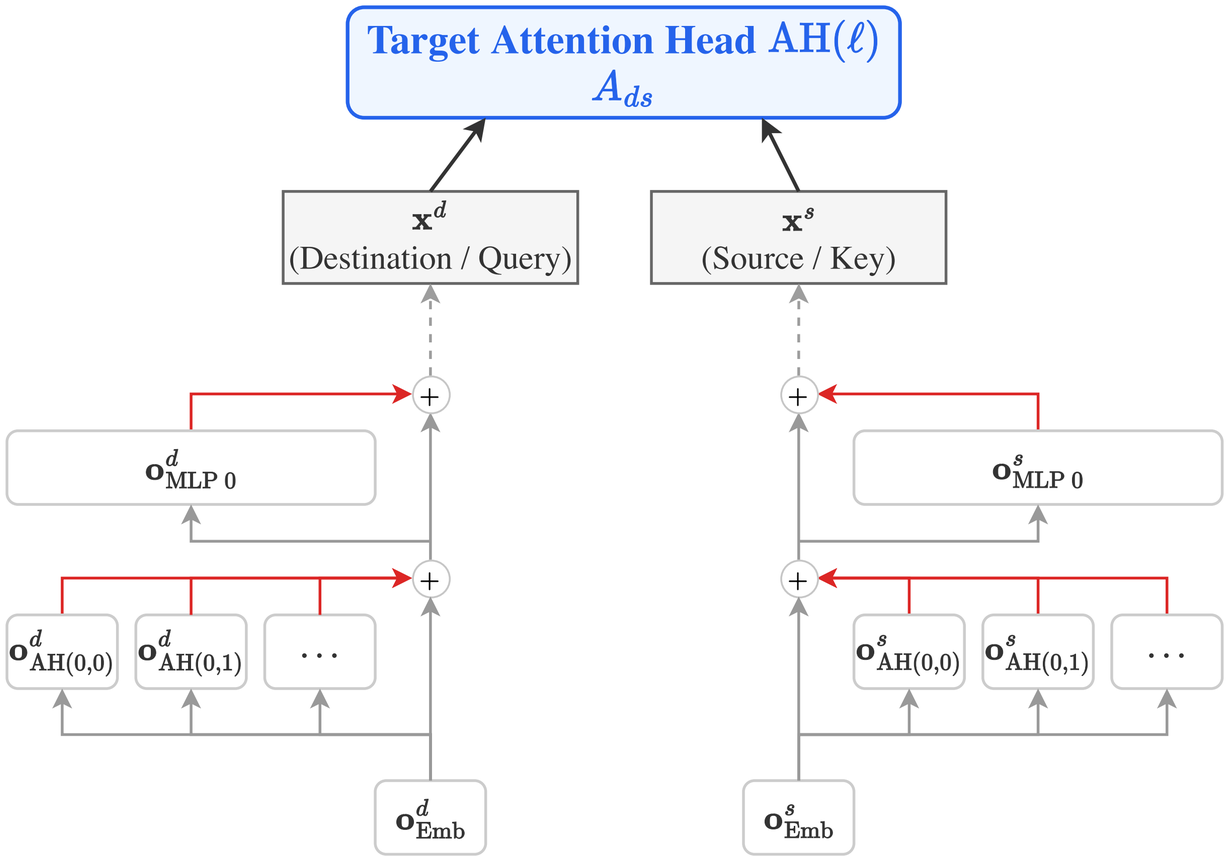
Singular vectors shift first; features then shift to follow.
Both evolve during training

Who wrote those features? Are they causally responsible?

From geometry to attribution

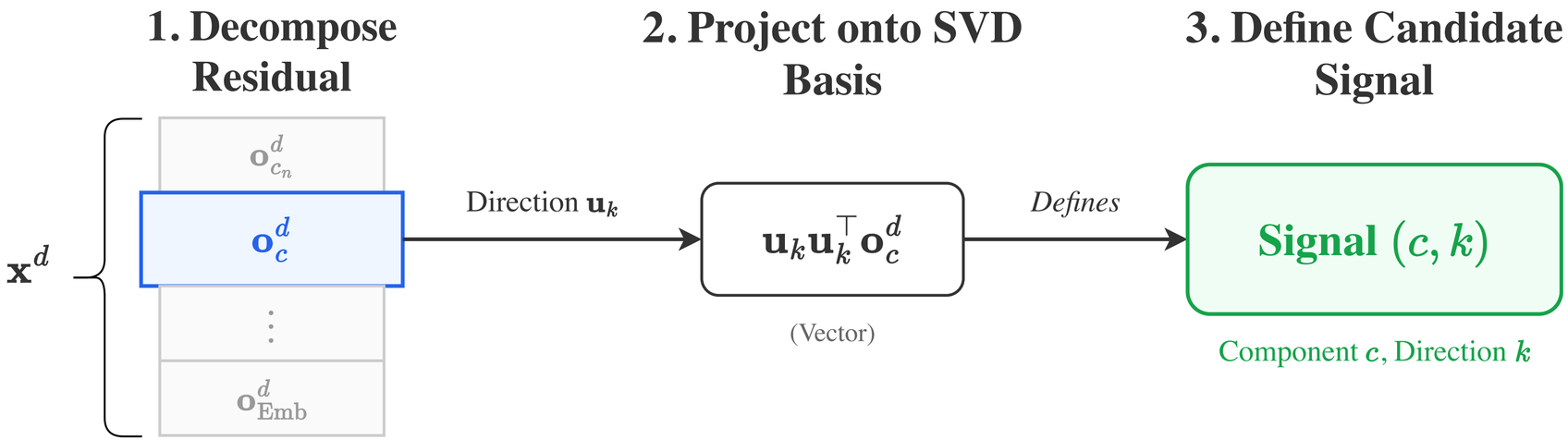
SVF tells us what features are present in residuals. But who put them there?

The residual stream $\mathbf{x}^d$ can be seen as a sum of outputs from all upstream components in the token d



Which ones wrote features that are **causally** responsible for this attention weight?

A signal is what an upstream component writes into an SVD channel



Destination Signals
What upstream components write into $\mathbf{u}_k$ directions (Query side)

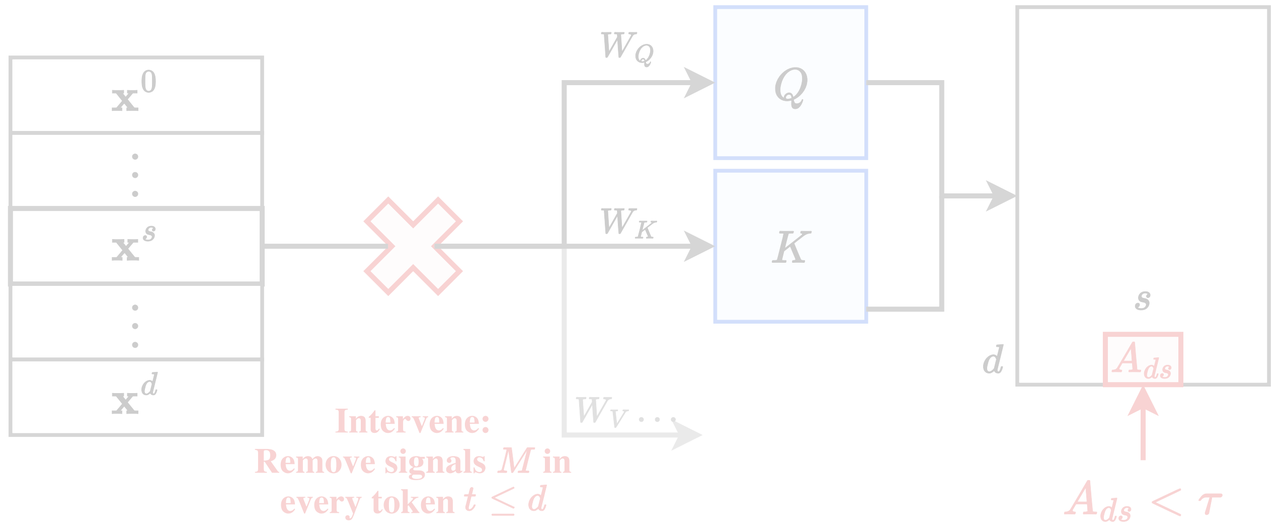
Source Signals
What upstream components write into $\mathbf{v}_k$ directions (Key side)

SVF alignment justifies using the SVD basis to define these signals.
Without SVF, there is no obvious basis to project onto

ACC++: a greedy counterfactual search over candidate signals

The Input

An attention head (ℓ, a) attending to a token pair (d, s) with a high attention weight A_{ds}.



The attention causal communication problem

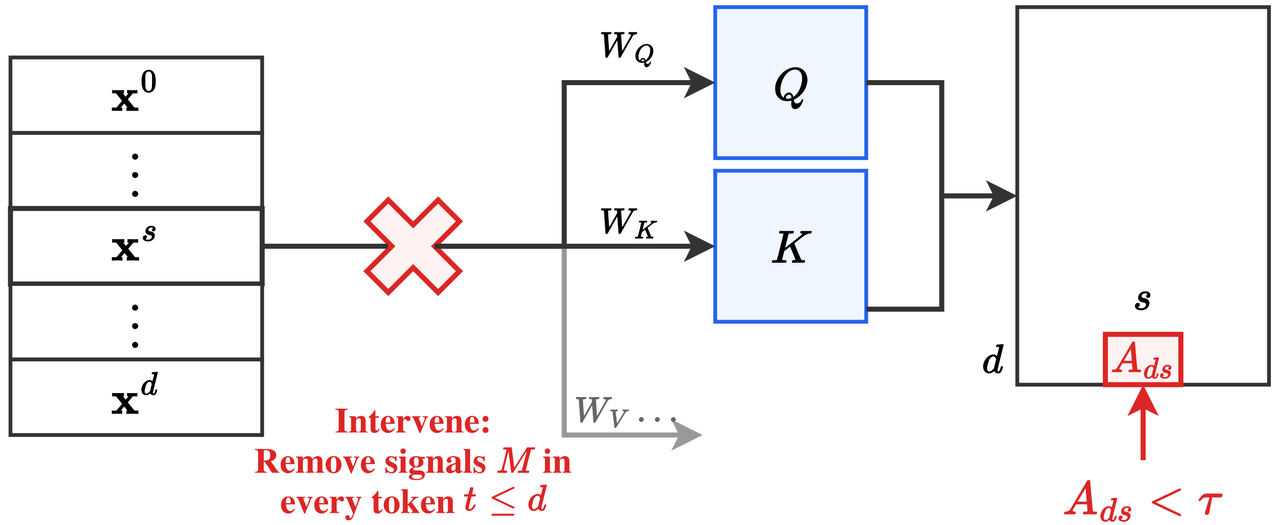
Find a minimal set M such that if we intervene by removing M, the attention A_{ds} drops below threshold τ.

Solved efficiently via greedy search guided by Integrated Gradients (IG) - **ACC++**

ACC++: a greedy counterfactual search over candidate signals

The Input

An attention head (ℓ, a) attending to a token pair (d, s) with a high attention weight A_{ds}.



The attention causal communication problem

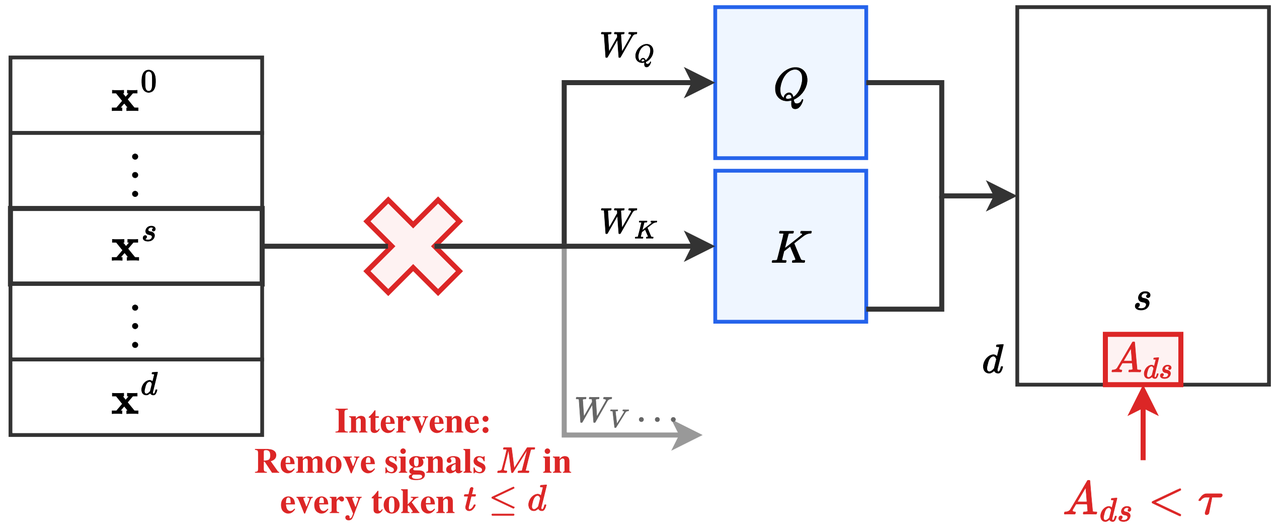
Find a minimal set M such that if we intervene by removing M, the attention A_{ds} drops below threshold τ.

Solved efficiently via greedy search guided by Integrated Gradients (IG) - **ACC++**

ACC++: a greedy counterfactual search over candidate signals

The Input

An attention head (ℓ, a) attending to a token pair (d, s) with a high attention weight A_{ds}.



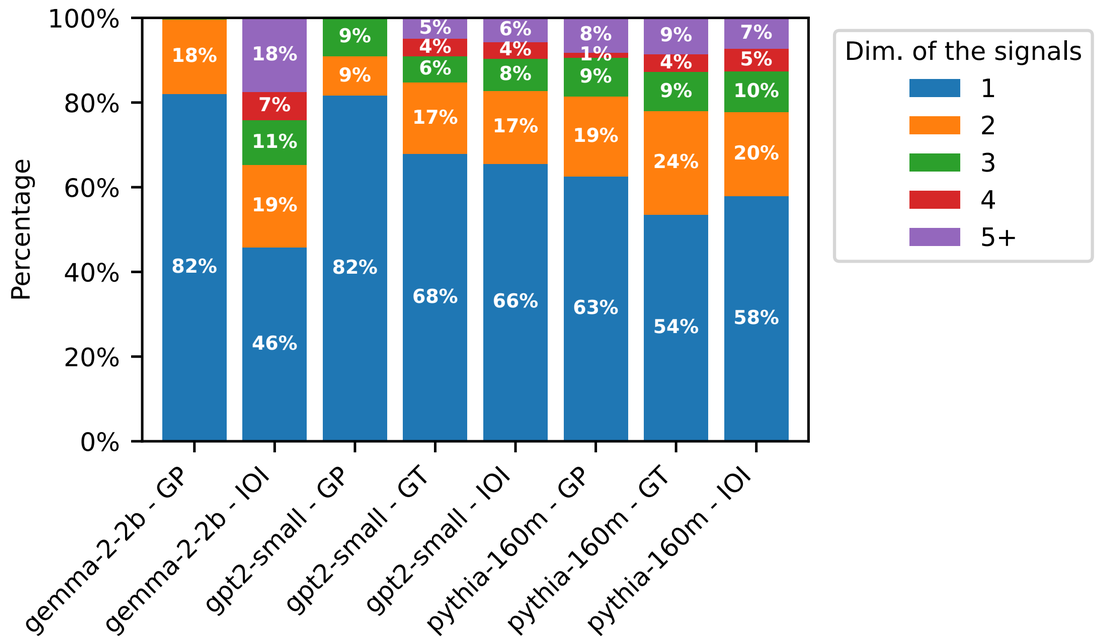
The attention causal communication problem

Find a minimal set M such that if we intervene by removing M, the attention A_{ds} drops below threshold τ.

Solved efficiently via greedy search guided by Integrated Gradients (IG) - **ACC++**

This algorithm ensures that the signals found are **causal** for attention A_{ds}

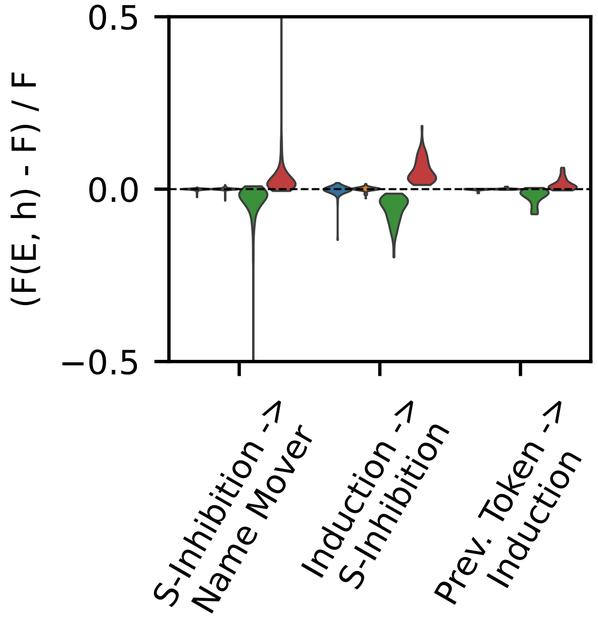
Signals are almost always rank-1

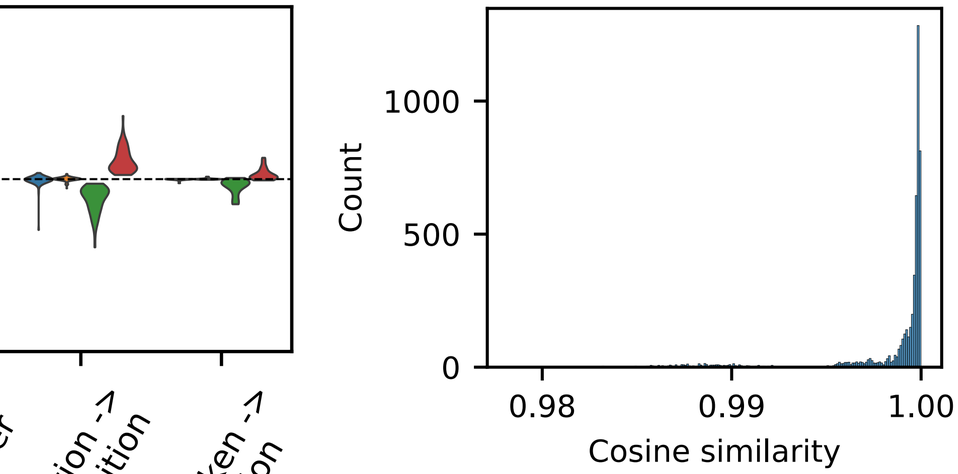


Median dimensionality = 1 across all models and tasks

Removing identified signals drops attention and changes model behavior

Behavior changes substantially: signals are also causal to the model performance





Intervention is precise: residual stream barely changes

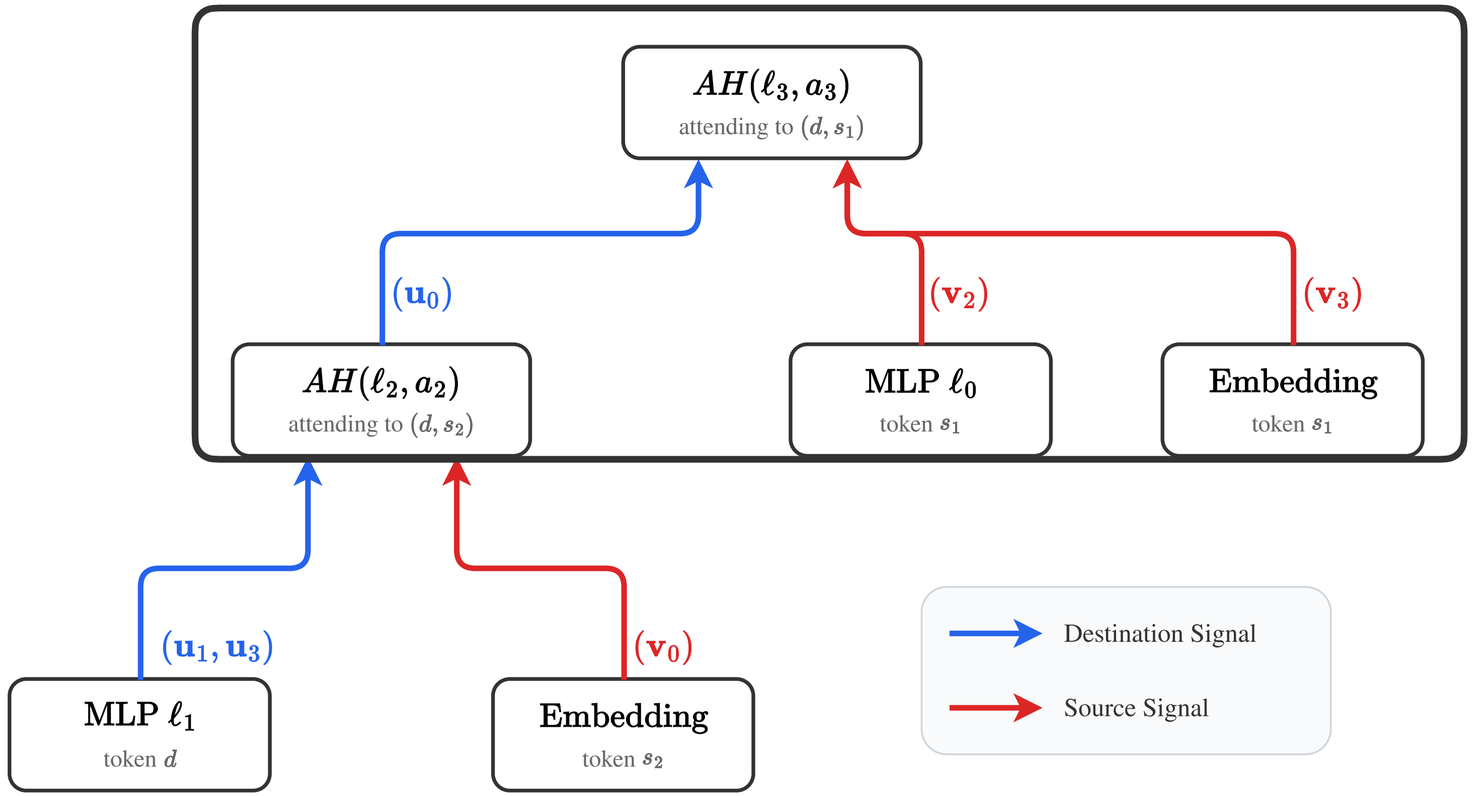
Low-rank, low-perturbation interventions have large behavioral effects. Random-signal controls do not

How do these causes compose into a circuit?

Per-prompt tracing and what it reveals

Tracing per-prompt circuits by applying ACC++ recursively

Each edge carries the specific signal (component, direction) used for communication

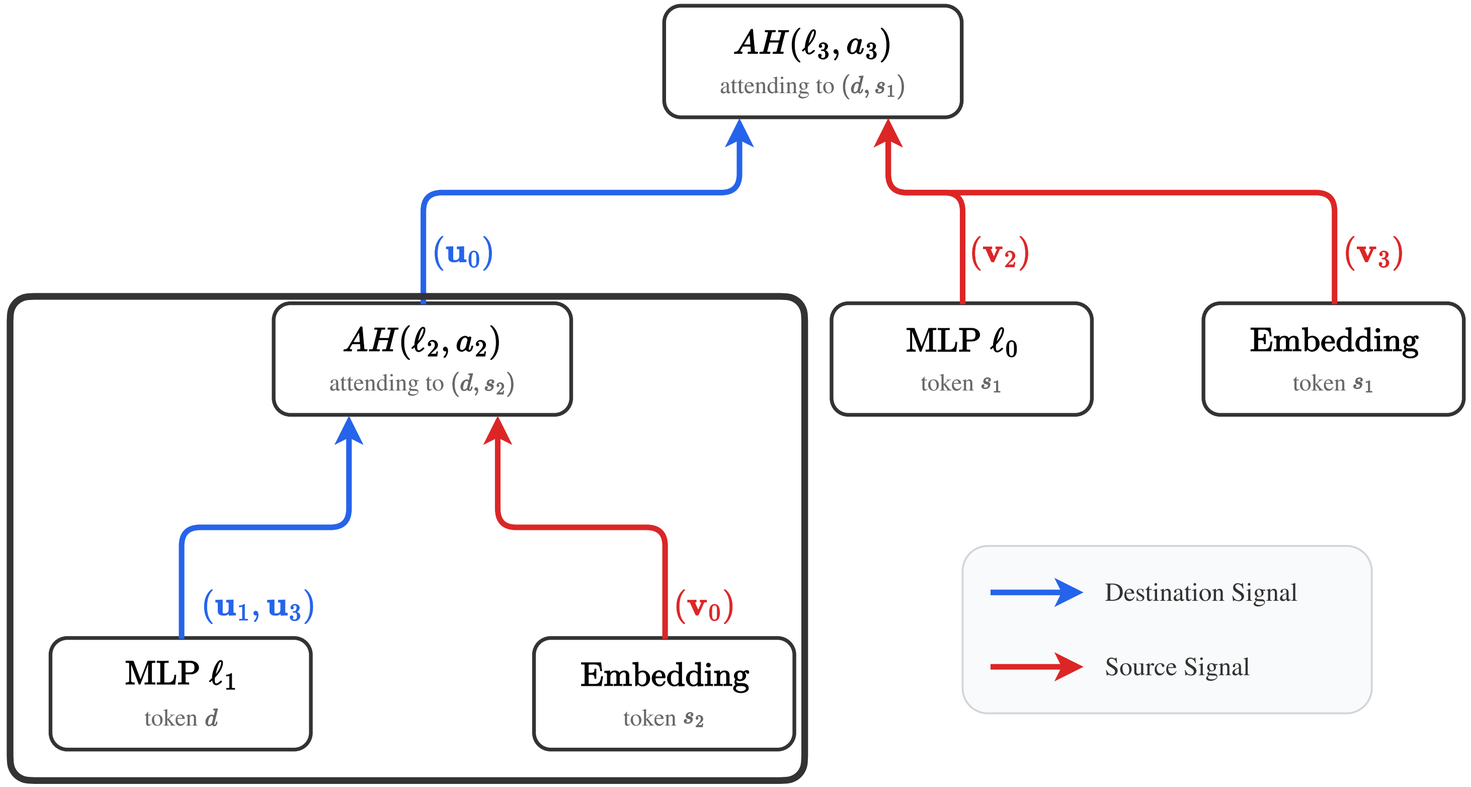


ACC++ solution

We do it using a single forward pass, with no counterfactual inputs or patching.
Every edge has a causal effect in attention downstream and carries a low-dimensional signal

Tracing per-prompt circuits by applying ACC++ recursively

Also an ACC++ solution

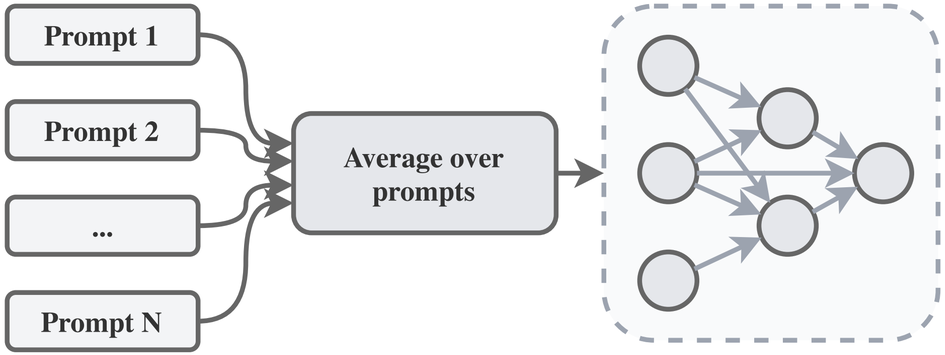


Terminates at embeddings or components with no upstream attention dependence

We do it using a single forward pass, with no counterfactual inputs or patching.
Every edge has a causal effect in attention downstream and carries a low-dimensional signal

Why per-prompt circuits? The problem with averaging

Single Task Circuit



Averaging assumes a single true mechanism, confounding distinct causal pathways into an unstable, dense structure

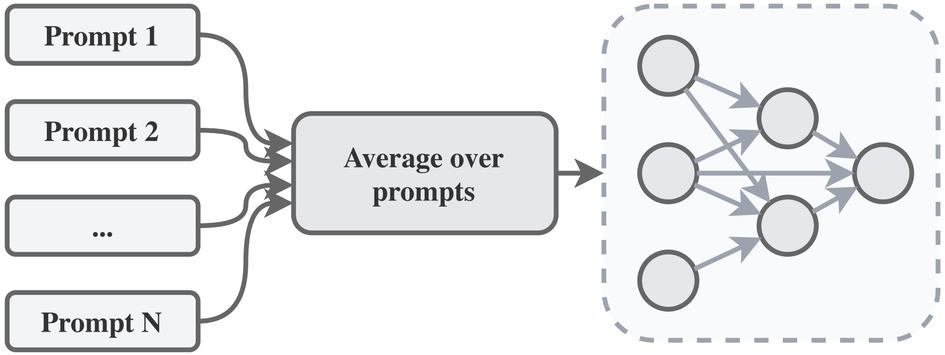
Prompt-Specific Circuits

Models learn multiple valid mechanisms for a task. Different prompts systematically route through entirely different subcircuits

Averaged task circuits exhibit massive structural variance (Méloux et al., 2025).
Our per-prompt ACC++ traces may explain why

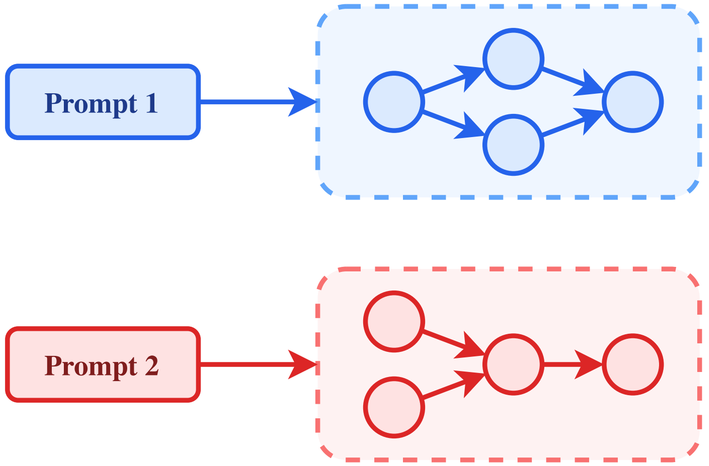
Why per-prompt circuits? The problem with averaging

Single Task Circuit



Averaging assumes a single true mechanism, confounding distinct causal pathways into an unstable, dense structure

Prompt-Specific Circuits



Models learn multiple valid mechanisms for a task. Different prompts systematically route through entirely different subcircuits

Averaged task circuits exhibit massive structural variance (Méloux et al., 2025).
Our per-prompt ACC++ traces may explain why

Testbed: Indirect Object Identification (IOI)

	High-level template	Low-level template
ABBA	Then, [A] and [B] went to the office. [B] gave a computer to [A]	Then, [A] and [B] went to the office. [B] gave a computer to [A]
	vs.	vs.
BABA	Then, [B] and [A] went to the office. [B] gave a computer to [A]	Friends [B] and [A] found a computer at the office. [B] gave it to [A]

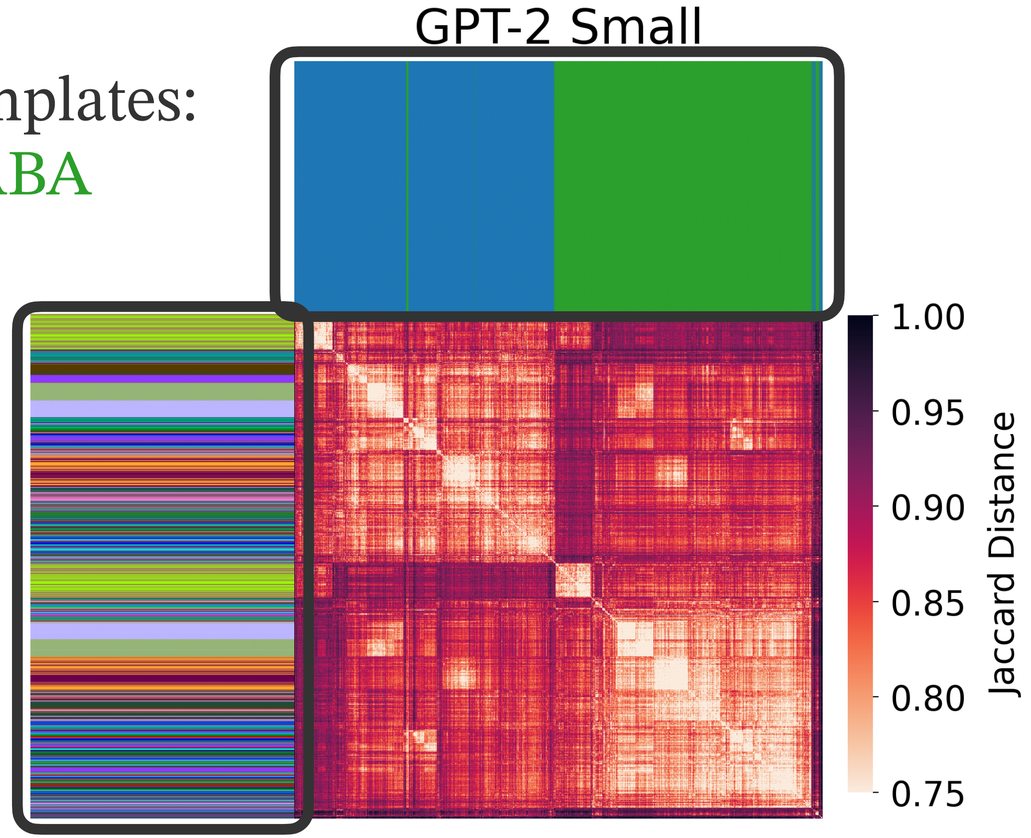
15 variations

15 low-level templates × 2 high-level templates × 100 examples = 3,000 prompts.
Models: GPT-2 Small, Pythia-160M, Gemma-2 2B

Prompts cluster into distinct circuit families

High-level templates:
ABBA and BABA

Low-level templates

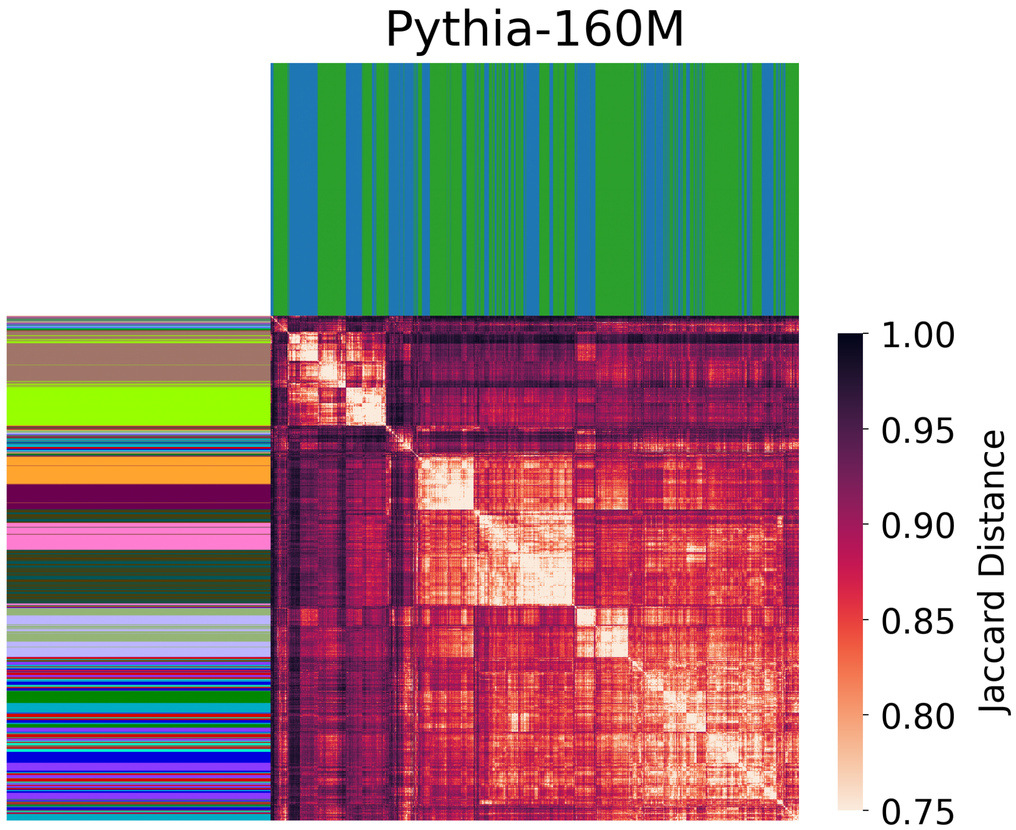


GPT-2 Small:
clean split by high-level template
(ABBA vs. BABA)

Circuits are represented as sets of edge–singular-vector pairs.
Average linkage clustering reveals that the "universal IOI circuit" is actually a collection of families

Prompts cluster into distinct circuit families

Pythia-160M: splits by low-level surface wording instead

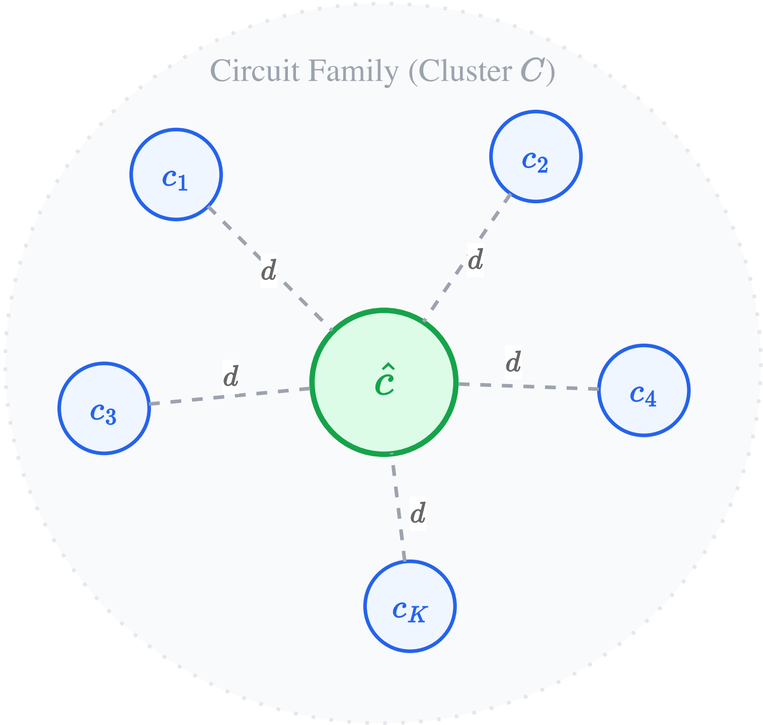


Circuits are represented as sets of edge–singular-vector pairs.
Average linkage clustering reveals that the "universal IOI circuit" is actually a collection of families

Summarizing circuit families: The representative circuit

We have 3,000 circuits per model/ task. We need one canonical example per family to inspect. The representative is that example

$$\hat{c} = \arg\min_{c \in C} \sum_{i=1}^{K} d(c_i, c)$$

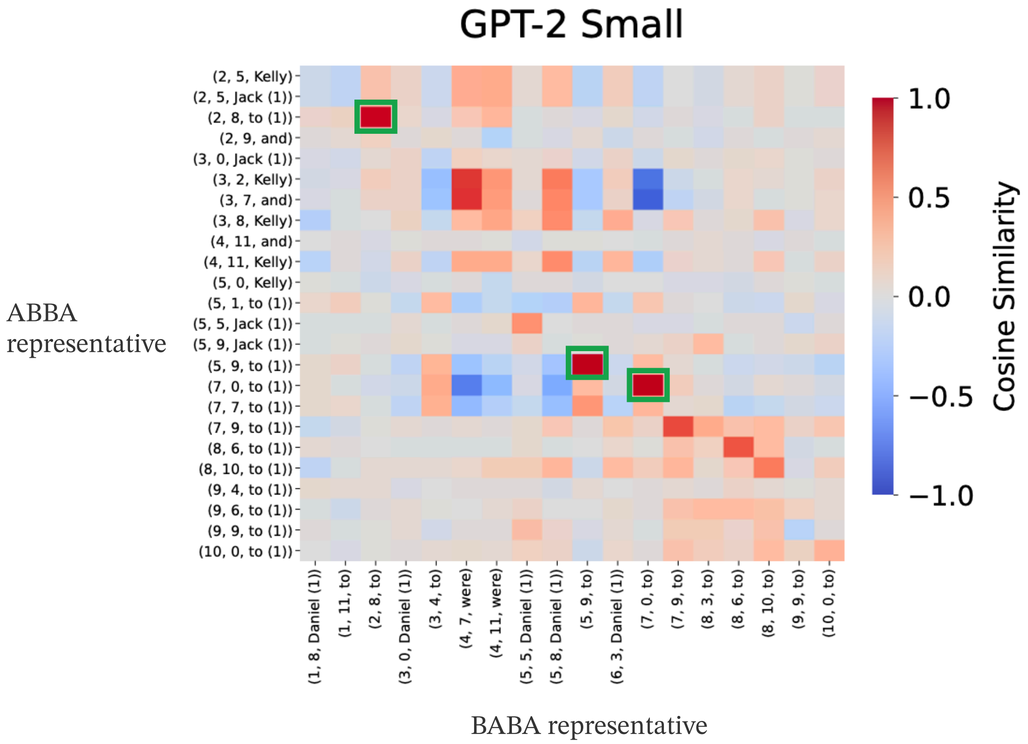


The representative minimizes average Jaccard distance to all other circuits within the same cluster

This gives us a principled unit of analysis for each prompt class.
Everything that follows is about comparing representatives

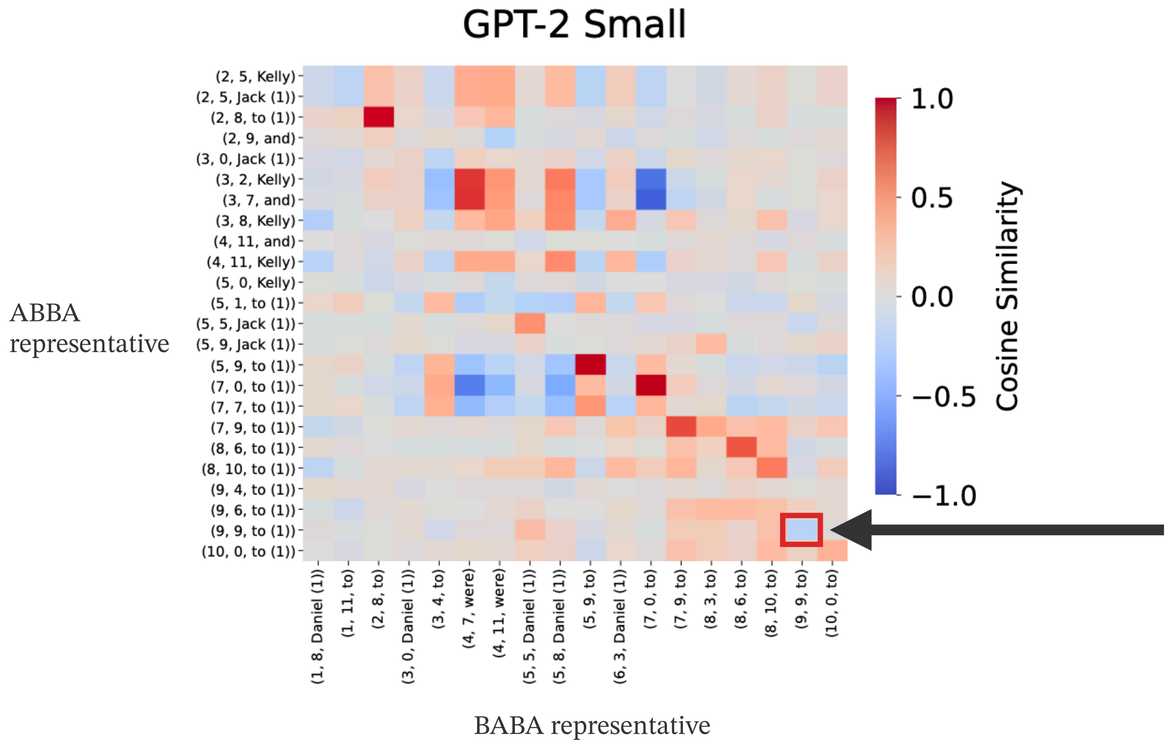
Signal-level comparison reveals identical components doing the same things

Same mechanism, same signal: Heads (2, 8), (5,9), and (7, 0) shows cosine similarity around 1.0 across both representatives



To compare representatives, we aggregate incoming ACC++ signals at each head and compute cosine similarity. This plot shows the aggregated destination signal

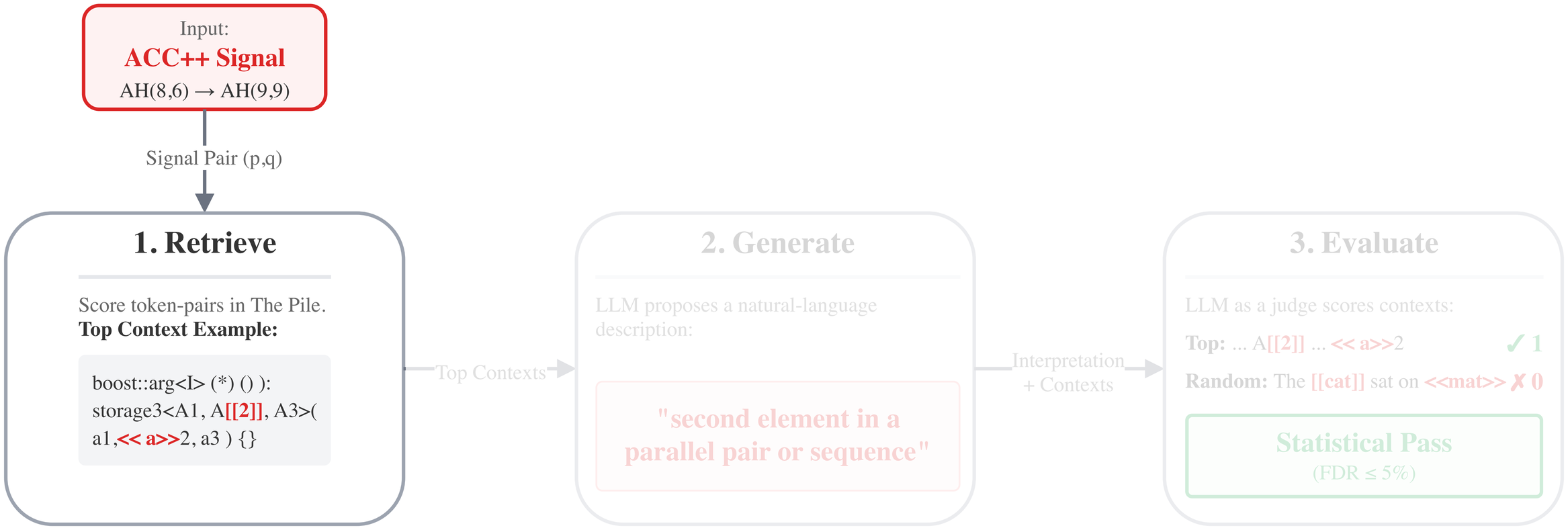
Signal-level comparison reveals identical components doing **different** things



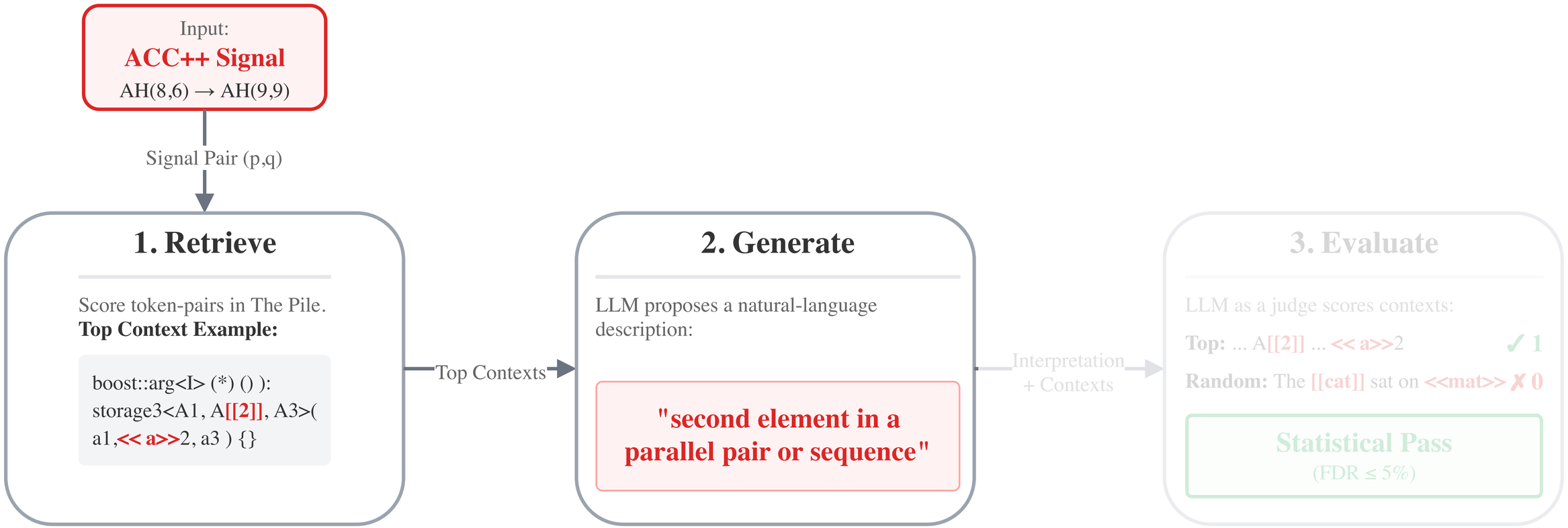
Changing mechanism: the canonical name-mover Head (9,9) shows *negative* cosine similarity between ABBA and BABA

To compare representatives, we aggregate incoming ACC++ signals at each head and compute cosine similarity. This plot shows the aggregated destination signal

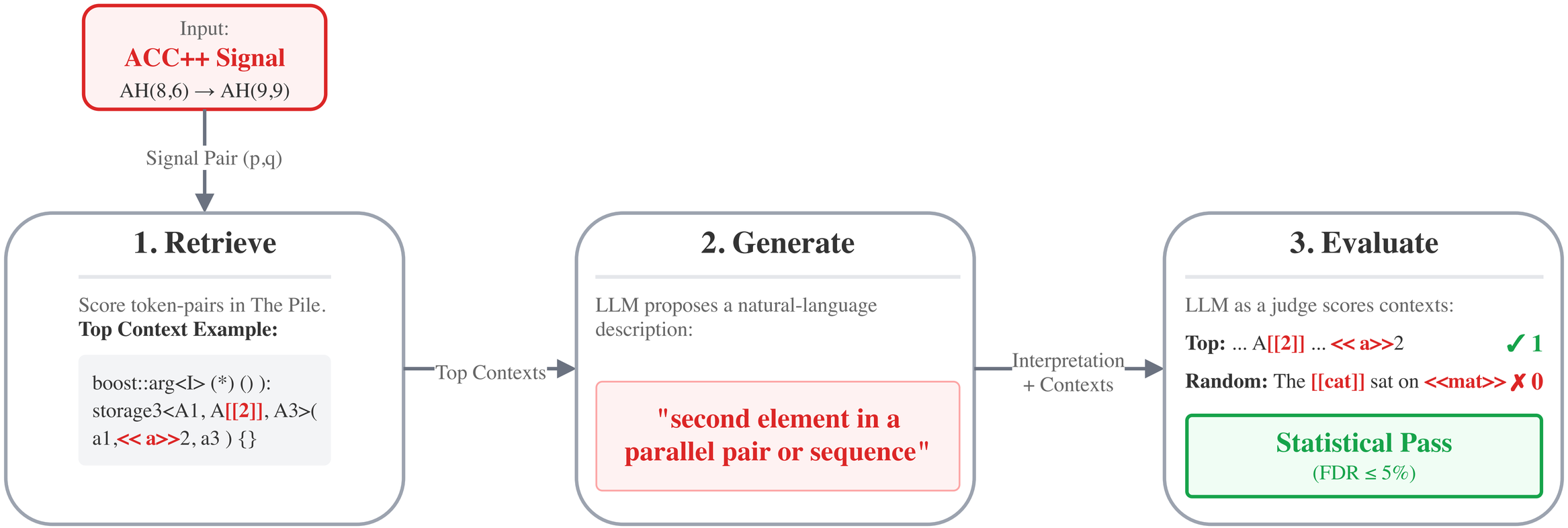
Autointerpretation pipeline for ACC++ signals



Autointerpretation pipeline for ACC++ signals



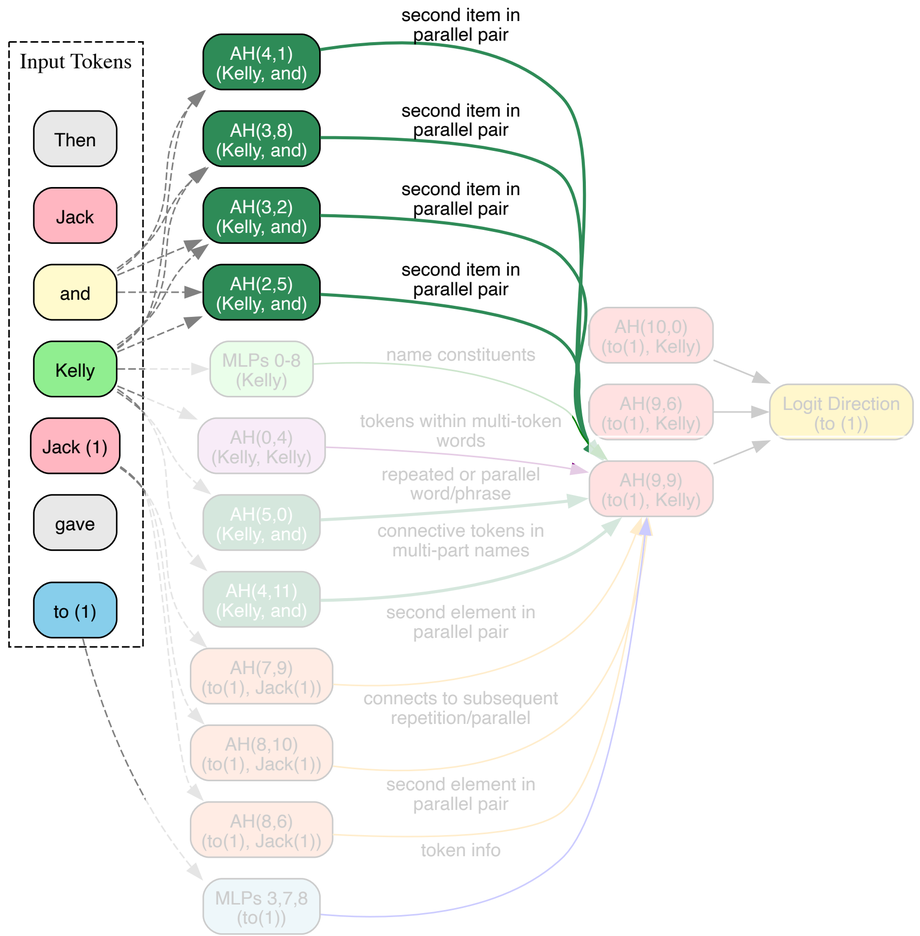
Autointerpretation pipeline for ACC++ signals



A significant portion of ACC++ signals pass this rigorous statistical evaluation (FDR ≤ 5%):
GPT-2 Small at **63%** (62–64%), Pythia-160M at **50%** (49–51%), and Gemma-2 2B at **31%** (30–32%)

Interpretable signals explain the algorithmic difference

We analyze the pruned and aggregated **BABA** representative graph with edge interpretations

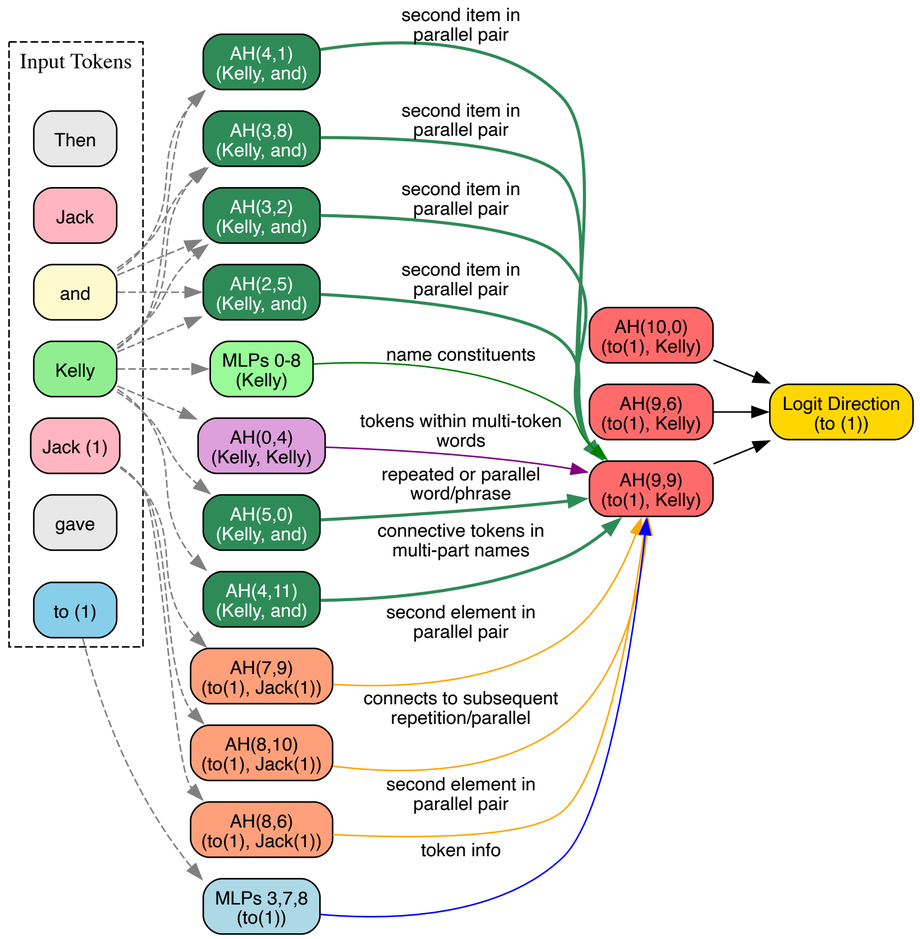


Multiple heads tag the IO token (“Kelly”) as “second item in a parallel pair”

BABA: "Then, Jack and Kelly went to the garden. Jack gave a basketball to"

Interpretable signals explain the algorithmic difference

S-inhibition heads provides suppression signals for the S token (“Jack")

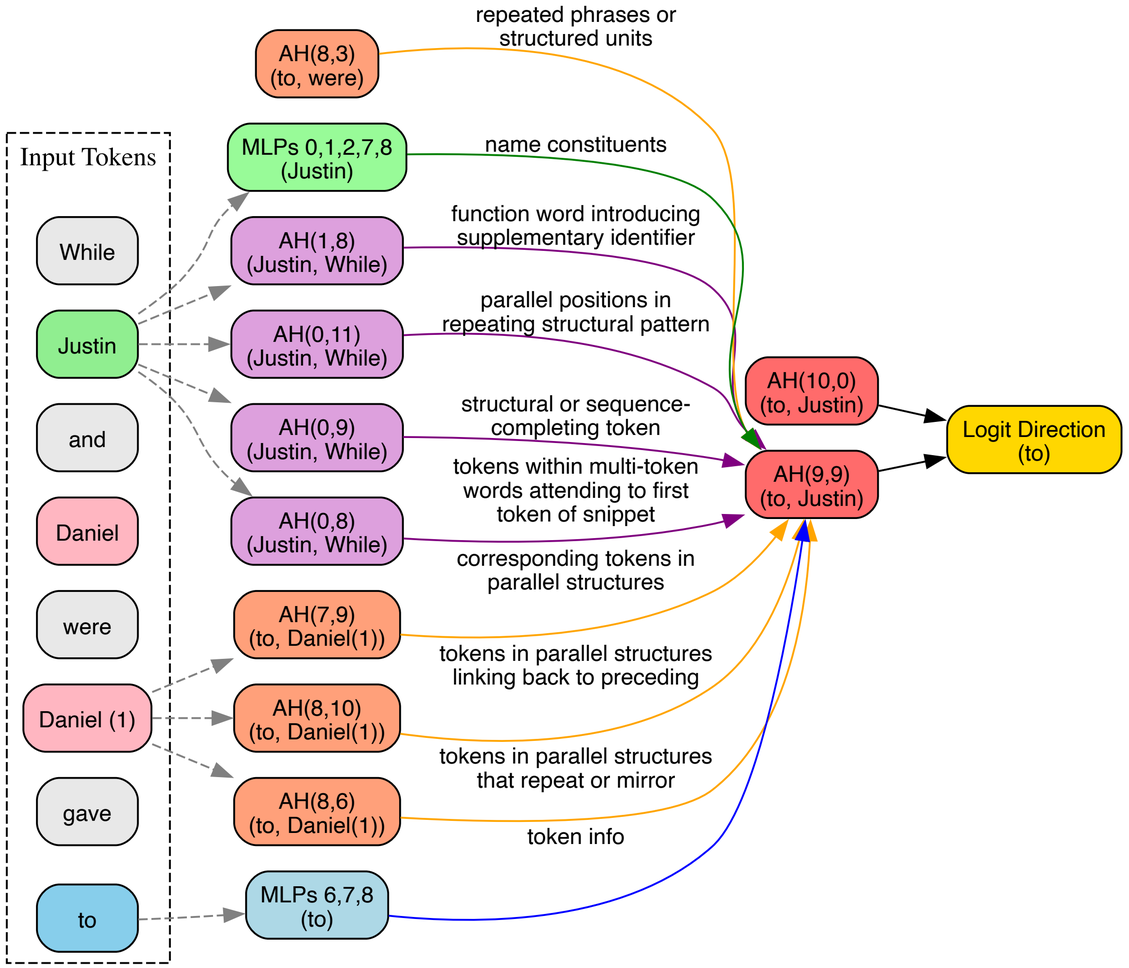


Name mover heads attends to (“to”, “Kelly”) and drives the prediction of “Kelly”

BABA: "Then, Jack and Kelly went to the garden. Jack gave a basketball to"

Interpretable signals explain the algorithmic difference

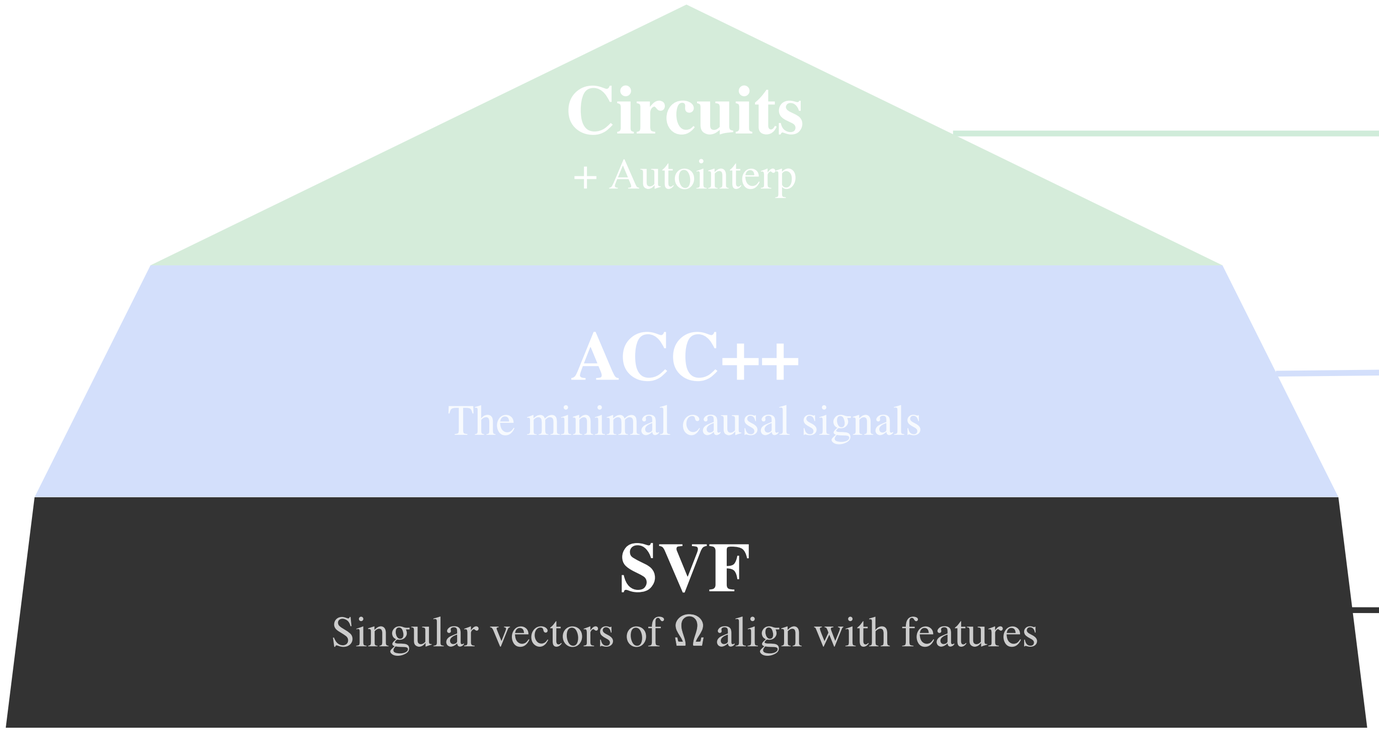
ABBA uses structural/ positional matching (not “second item”)



IO is found via **generic structure/ position features** (sequence-completing / structural cues)

ABBA: "While Justin and Daniel were working at the house, Daniel gave a snack to"

Why does this head attend here? Now we have an answer

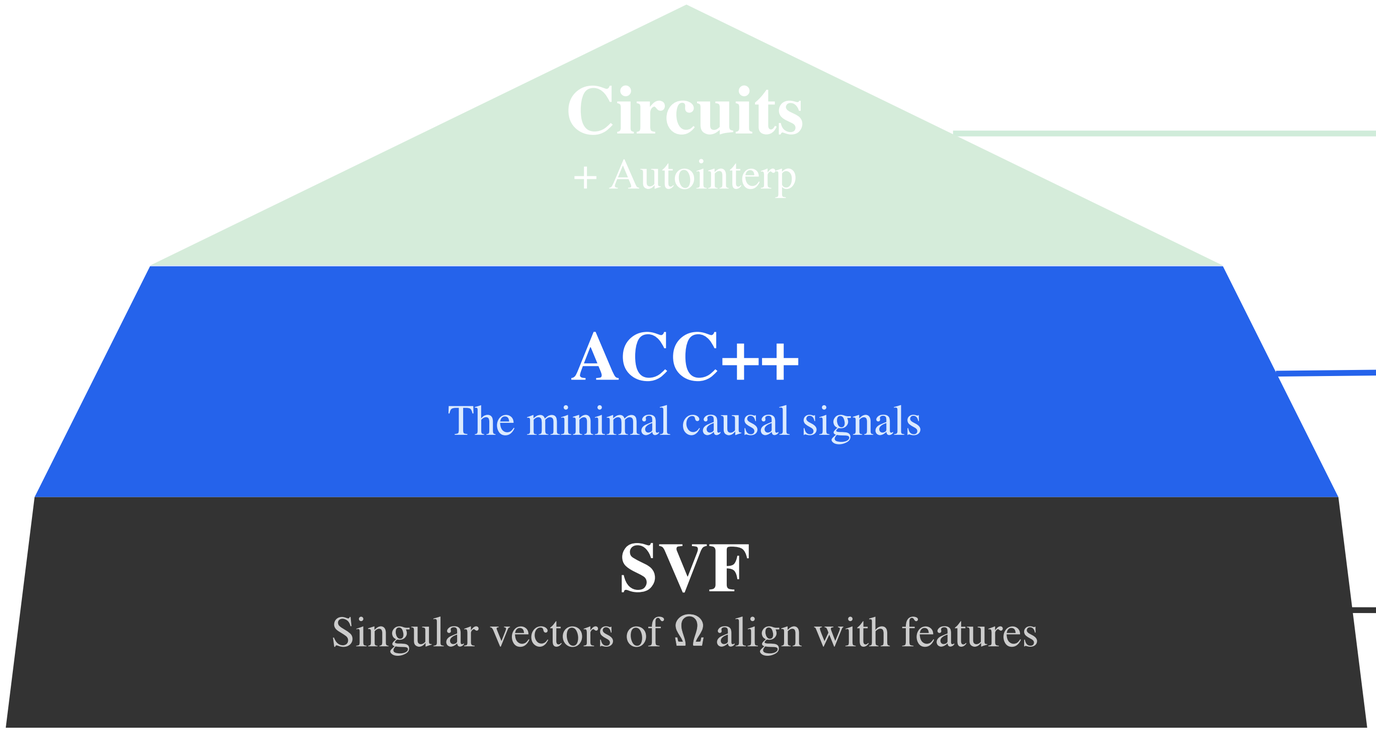


Causal and interpretable edges
Prompt-specific
Single forward pass

Causal signals to attention are identified

Proven theoretically
Confirmed empirically

Why does this head attend here?
Now we have an answer



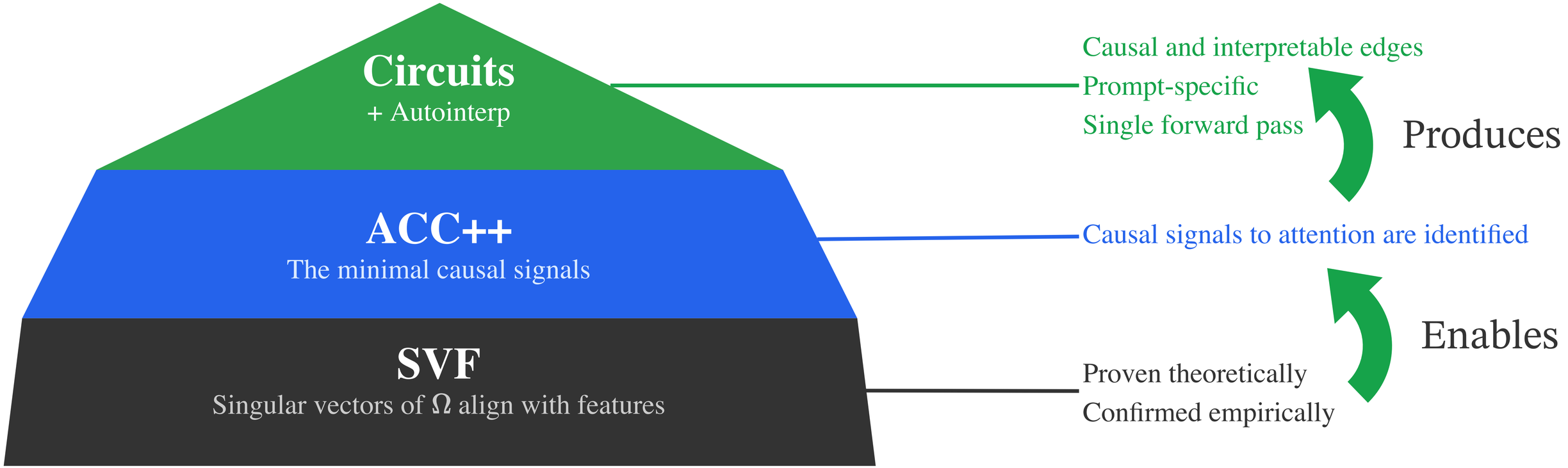
Causal and interpretable edges
Prompt-specific
Single forward pass

Causal signals to attention are identified

Enables

Proven theoretically
Confirmed empirically

Why does this head attend here?
Now we have an answer



Thanks to my collaborators

Mark Crovella

Lucas Tassis

Carson Loughridge

Azalea Rohr

Q&A

Pinpointing Attention-Causal Communication in Language Models

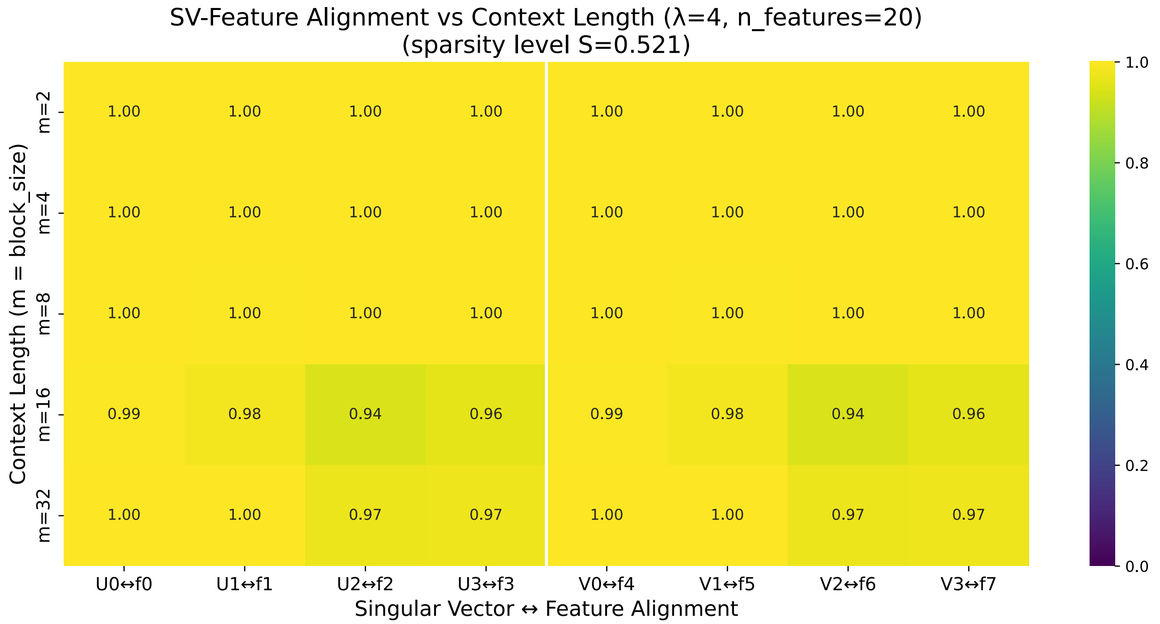
Singular Vectors of Attention Heads Align with Features

Finding Highly Interpretable Prompt-Specific Circuits in Language Models

Appendix

SVF alignment is robust across model hyperparameters

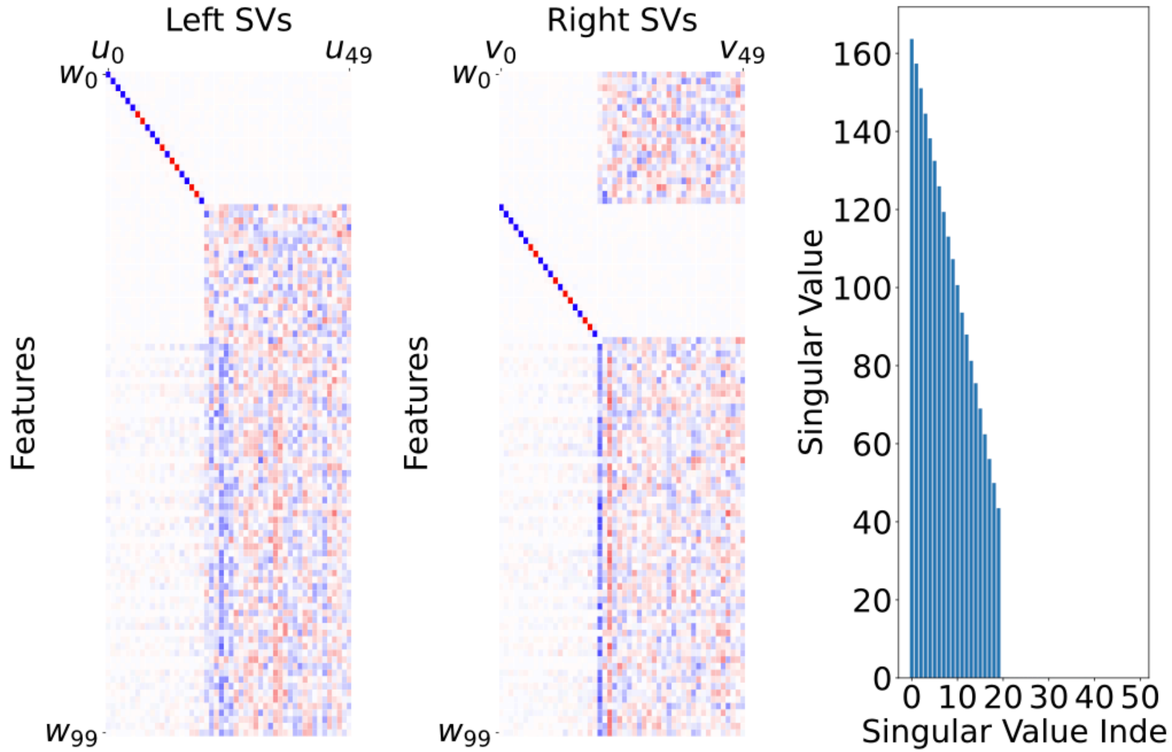
Robust to feature probability, loss weight, number of features, head dimension, context length



Consistent across 5 random seeds (mean ≈ 0.999, std ≈ 0.001)

SVF alignment holds with multiple features

Pairs of features (w_i, w_{i+20}), $0 \leq i \leq 19$, needs to be attended by the head with target logit $T_{i,i+20} = 26 - i$ (zero o.w.)



SVF alignment happens because the singular vectors align with features used by attention and the other features are orthogonalized

N=100 features, D=50 dimensions, H=50 heads dimensions

Sparse attention decomposition

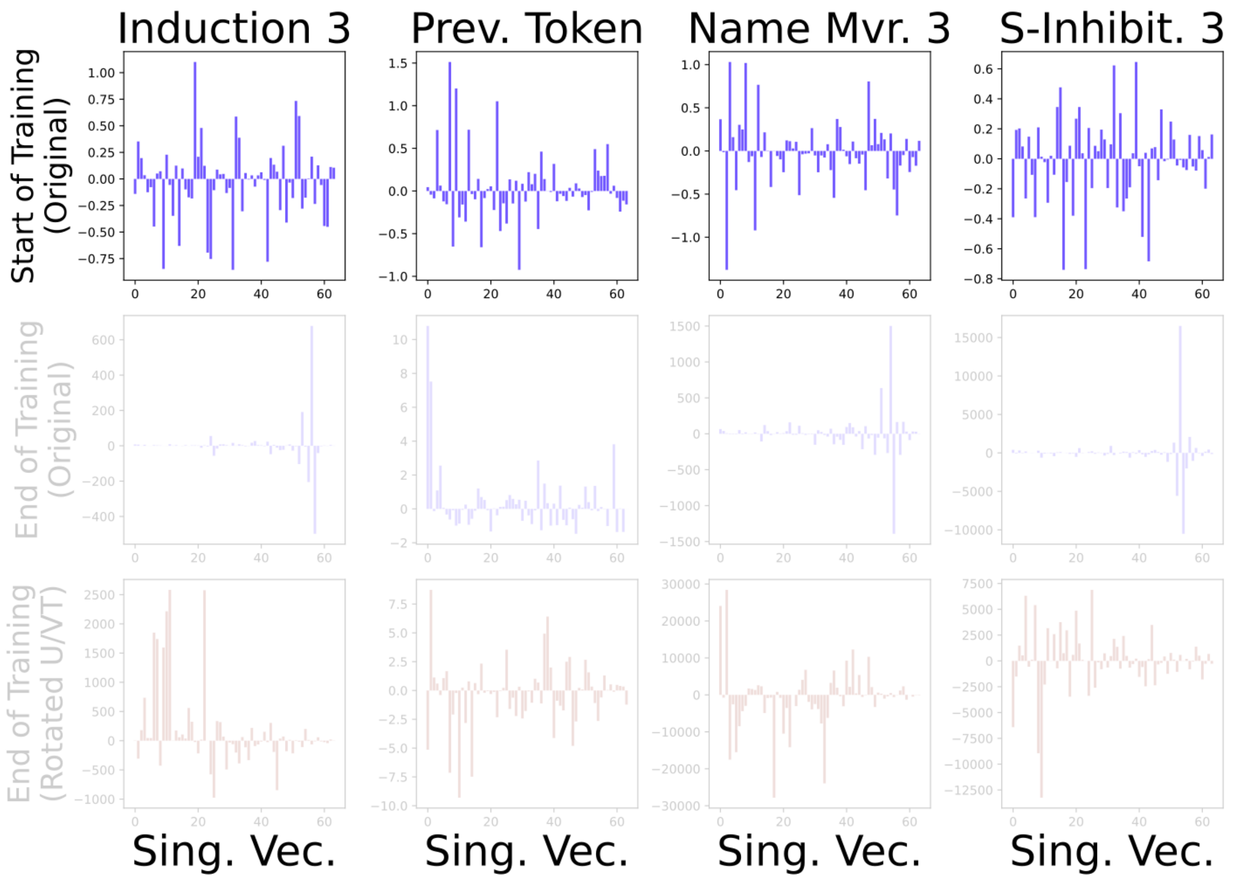
- We do not have the true features in real transformers
- We use the notion of relative attention to analyze when the token pair (d, s) is attended in a context with m tokens:

$$\mathrm{RA}(d, s) = A'_{ds} - \frac{1}{d-1} \sum_{j \leq d, j \neq s} A'_{dj}$$

- We decompose $\mathrm{RA}(d, s)$ in terms the singular vectors of Ω

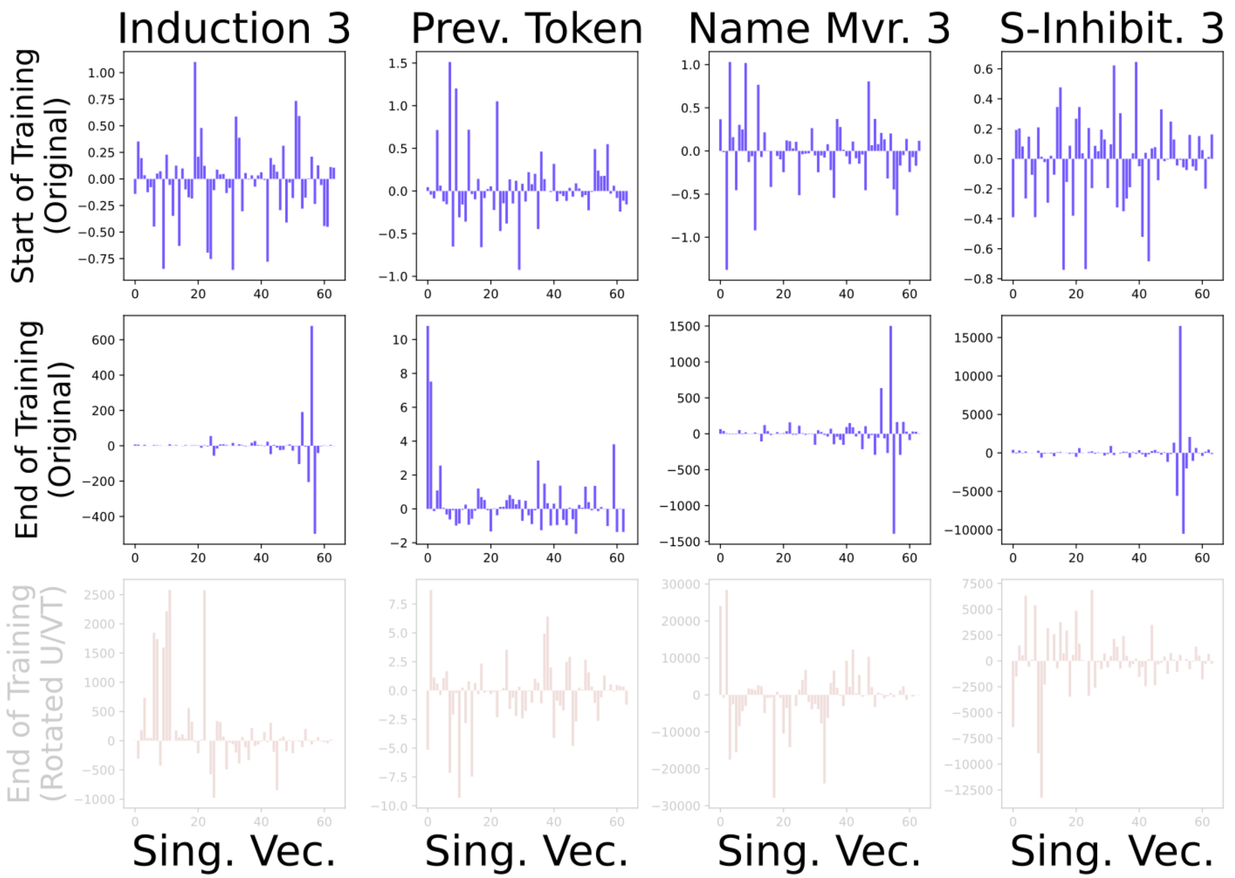
SVF alignment holds in real models: attention decomposes sparsely

No sparsity in the start of the training



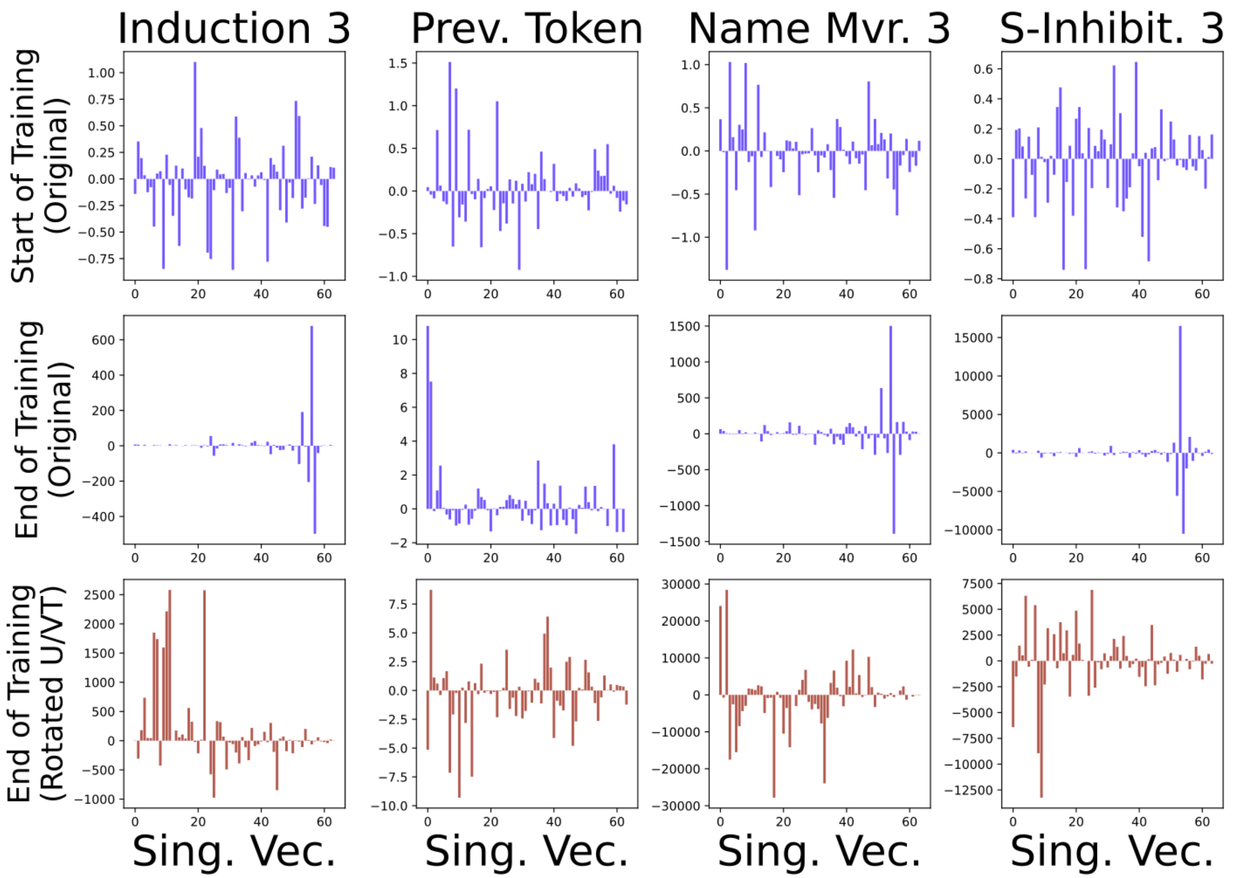
SVF alignment holds in real models: attention decomposes sparsely

A few singular vectors dominate in the end of training



SVF alignment holds in real models: attention decomposes sparsely

A few singular vectors dominate in the end of training



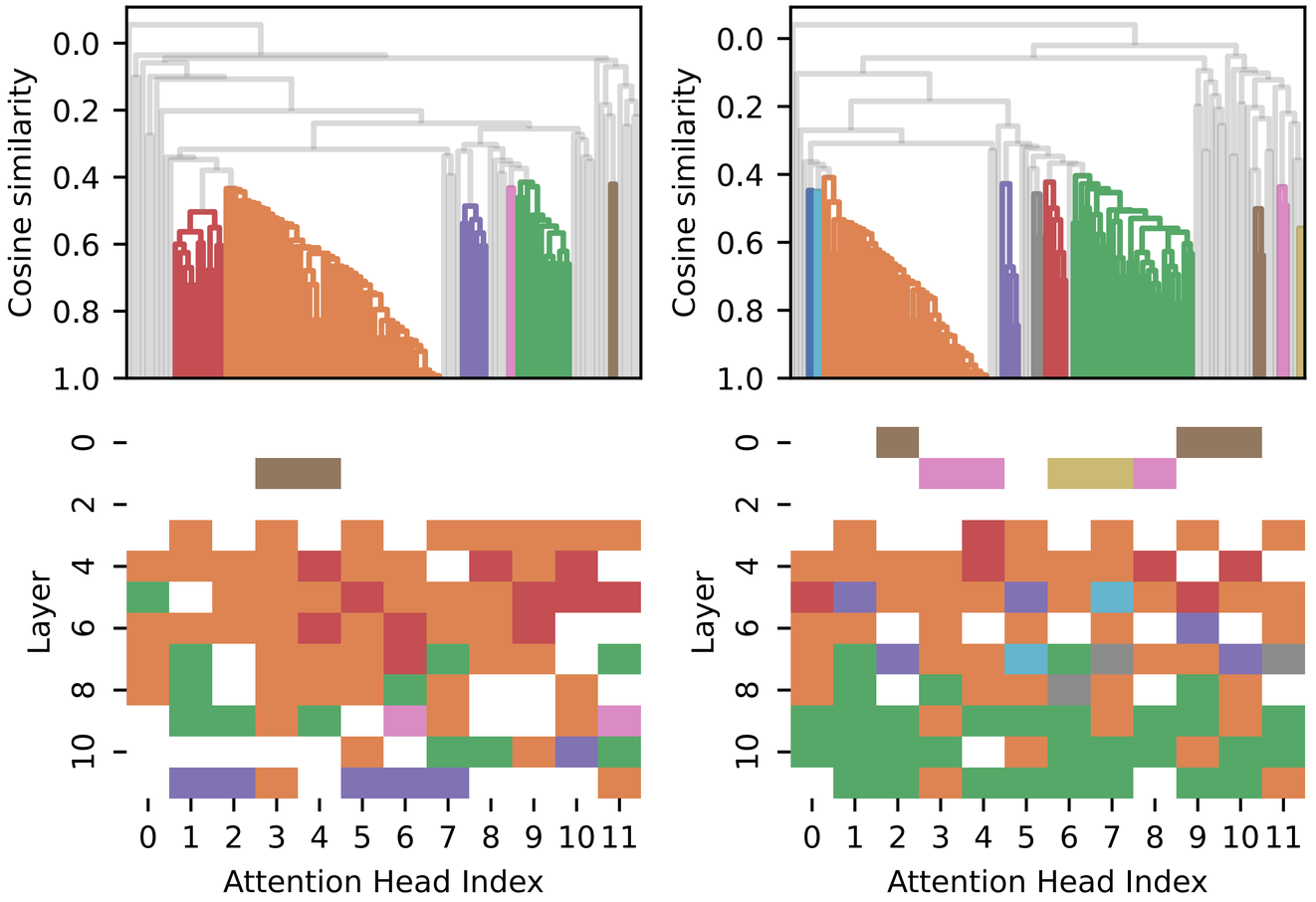
Rotating the U, V^T destroys sparsity: it is not a low-rank artifact

Pythia-160M, IOI circuit heads.

Sparse attention decomposition is a testable, falsifiable prediction of SVF alignment, and it holds

The same framework explains attention sinks

Attention sinks are caused by a small set of global control signals



Multiple heads across layers share the same signal — a reused global mechanism

Destination signals (left), source signals (right), GPT-2 small

ACC++ in more details

1. Decompose the QK matrix using SVD

$$\Omega = \sum_{k=1}^{R} \mathbf{u}^k \sigma^k \mathbf{v}^{k\top}$$

2. Decompose the residuals as a linear sum of upstream components outputs

$$\mathbf{x}^{\ell d} = \sum_{c \in C(\ell)} \mathbf{o}_c^d$$

ACC++ in more details

3. Projecting each component's output onto the SVD basis

$$\mathbf{x}^d = \sum_k \sum_c \underbrace{P_U^k \mathbf{o}_c^d}_{\text{signal candidate}}$$

4. Applying the decomposition of the residual stream in the attention scores, we have:

$$A'_{ds} = \mathbf{x}^{d\top} \Omega \mathbf{x}^s = \sum_k \sum_c \left(P_U^k \mathbf{o}_c^d\right)^\top \Omega \mathbf{x}^s$$

ACC++ in more details

5. Let N be the number of signal candidates. We build a contribution matrix $C \in \mathbb{R}^{N \times d}$ with the terms from the previous sum for every $A'_{dj}, \forall j \leq d$

$$C = \underbrace{\begin{bmatrix} \vdots & \cdots & \vdots \\ \vdots & \ddots & \vdots \\ \vdots & \cdots & \vdots \end{bmatrix}}_{\text{source tokens}} \Big\} N$$

6. C decomposes the attention logit vector for the destination token d

$$A'_d = \sum_{i=1}^{N} C_i, \quad C_i \in \mathbb{R}^d$$

ACC++ in more details

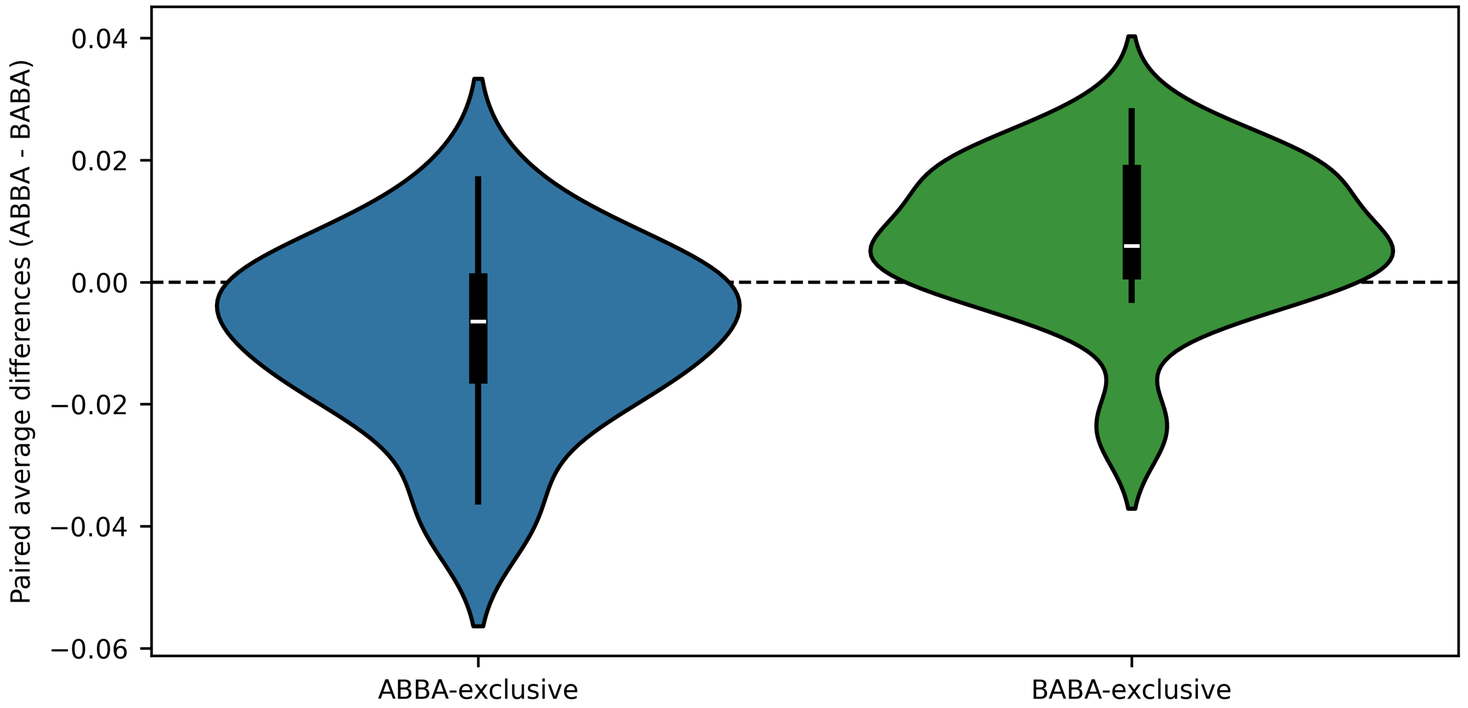
7. We know that the attention weights are given by:

$$A_d = \text{softmax}(A'_d), \quad A_{ds} = \frac{e^{A'_{ds}}}{\sum_j e^{A'_{dj}}}$$

8. How much did each C_i contribute to A_{ds}?

We use **integrated gradients** (IG) to perform the attribution.

Representatives are more causal to their own cluster



Intervening on ABBA exclusive edges has higher effect in ABBA prompts than BABA prompts (and vice-versa)