Understanding LLMs from the Inside Out

An Introduction to Mechanistic Interpretability & How Attention Really Works

Gabriel Franco
Boston University

BOSTON UNIVERSITY
NSF
Coefficient Giving

About me

UFV (Undergrad + Master's)
2014 - 2021

Undergraduate Research Fellow with Rodrigo Batista from DEM

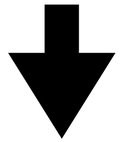
DOTA 2 research with Marcos and Giovanni

Learning from label proportions research with Giovanni

About me

UFV (Undergrad + Master's)
2014 - 2021

BU (PhD)
2021 - Present

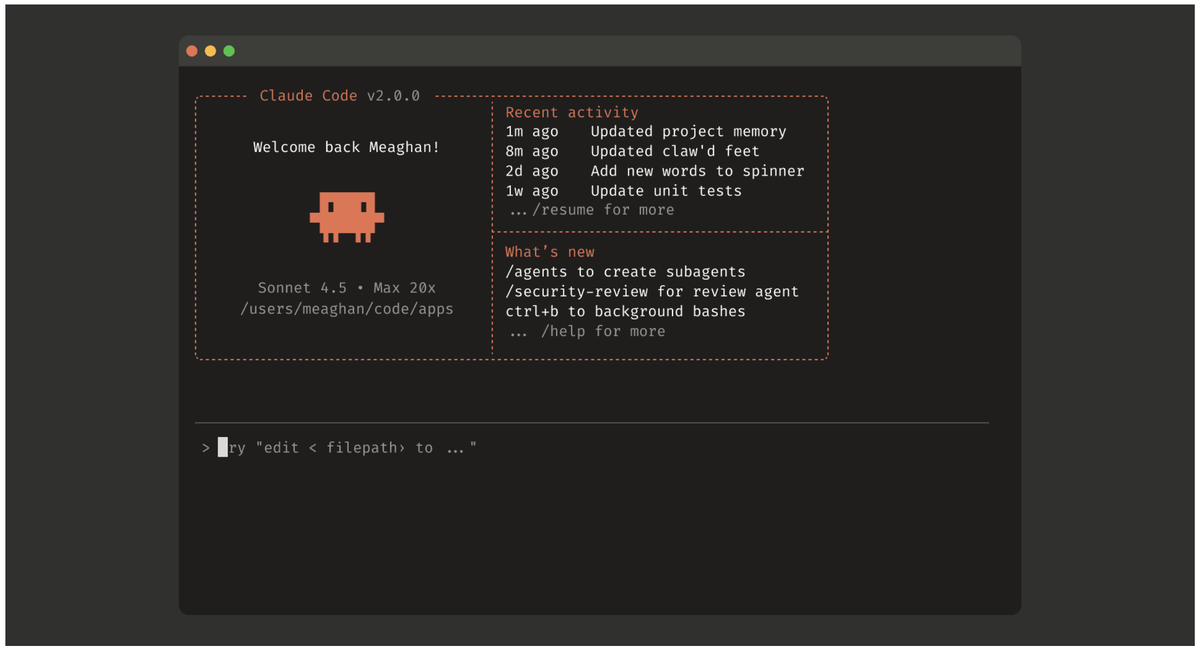


Learning from label proportions (2021-2024)

Mechanistic interpretability (2024 - now)

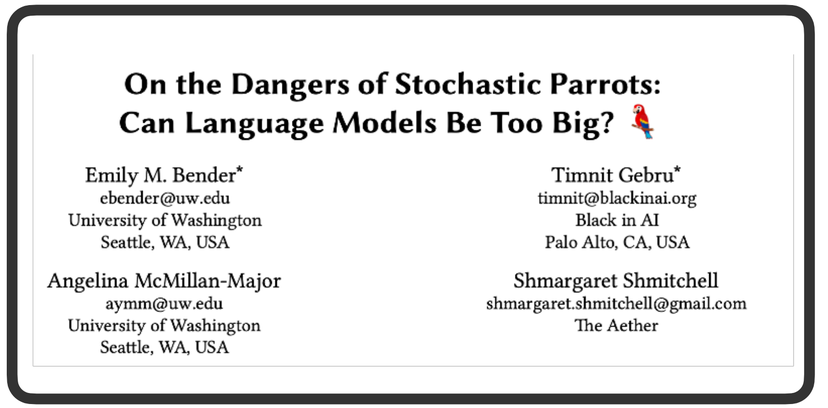
The moment we are in

- Large Language Models (LLMs) like ChatGPT and Claude are dramatically changing how we work and learn:
  - We proved a theorem in our paper with help with ChatGPT
- How will we evaluate and trust systems that are fundamentally inscrutable?
  - Billions of parameters
  - Unknown training data



How do they do it?

Current debate: Do language models operate on “statistical pattern recognition”, or “true comprehension”?



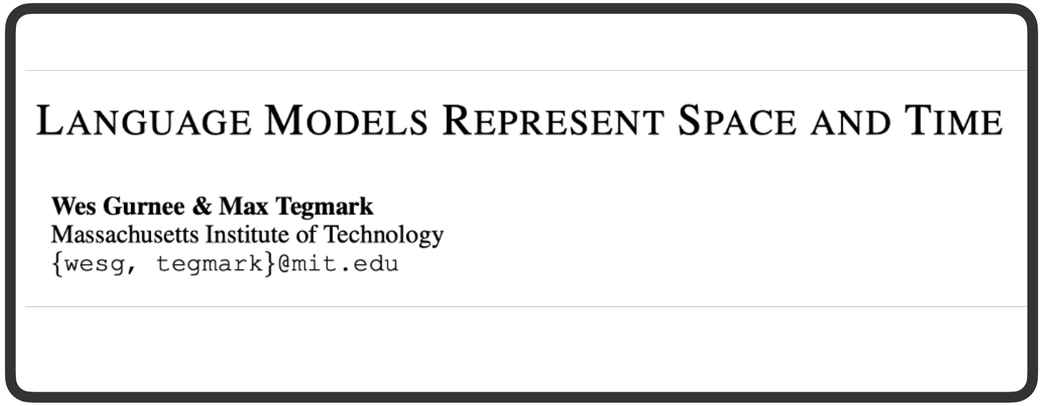
On the Dangers of Stochastic Parrots:
Can Language Models Be Too Big? 🦜

Emily M. Bender*
ebender@uw.edu
University of Washington
Seattle, WA, USA

Timnit Gebru*
timnit@blackinai.org
Black in AI
Palo Alto, CA, USA

Angelina McMillan-Major
aymm@uw.edu
University of Washington
Seattle, WA, USA

Shmargaret Shmitchell
shmargaret.shmitchell@gmail.com
The Aether



LANGUAGE MODELS REPRESENT SPACE AND TIME

Wes Gurnee & Max Tegmark
Massachusetts Institute of Technology
{wesg, tegmark}@mit.edu

Mechanistic interpretability is the field that seeks to answer this by reverse-engineering the internal mechanisms of neural networks

Mechanistic interpretability: understanding LLMs from the inside

We don't program neural networks, we don't make them, we grow them. The architecture is a scaffold that circuits grow on. The objective is a light it grows toward. But the thing we actually create — it's this almost biological entity that we're studying.

At the end of the day, we end up with this artifact that can do all these amazing things, and ***we have no idea*** *how it does it. That is a really deep and exciting scientific question.*

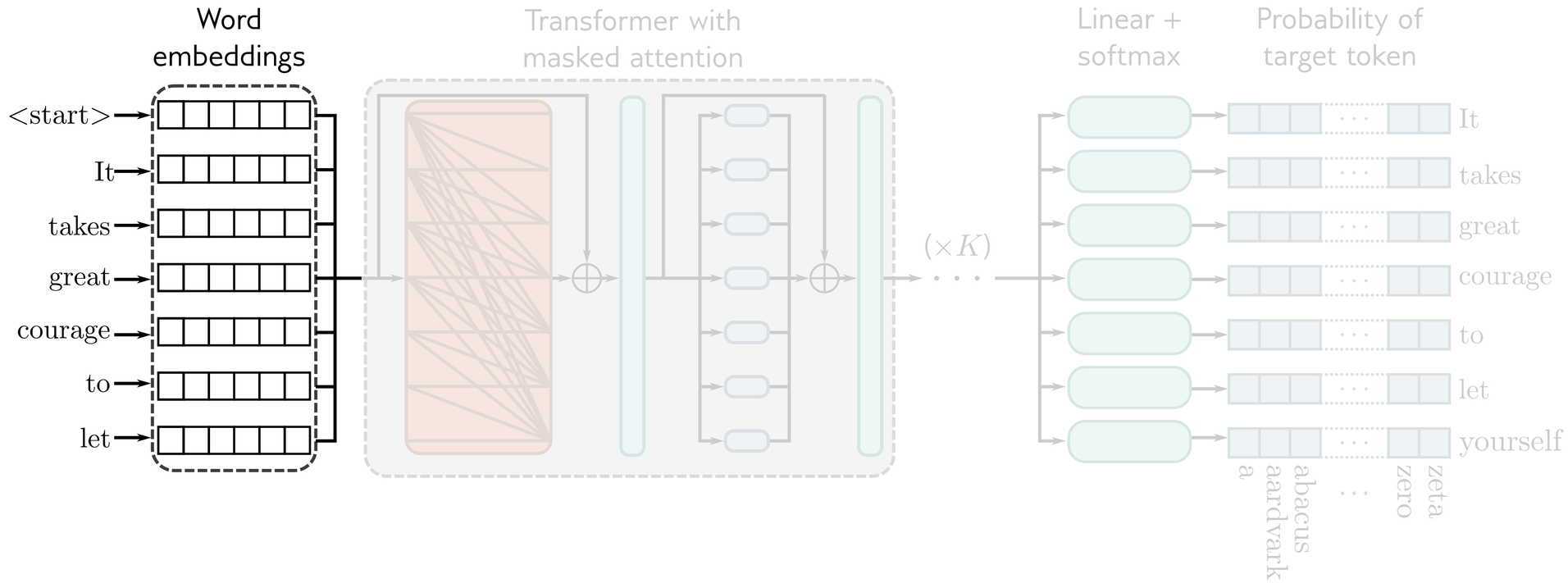
Chris Olah, Anthropic (adapted from Lex Fridman Podcast)

A Minimal Guide to Transformers

Just enough to follow the rest of the talk

How do language models work?

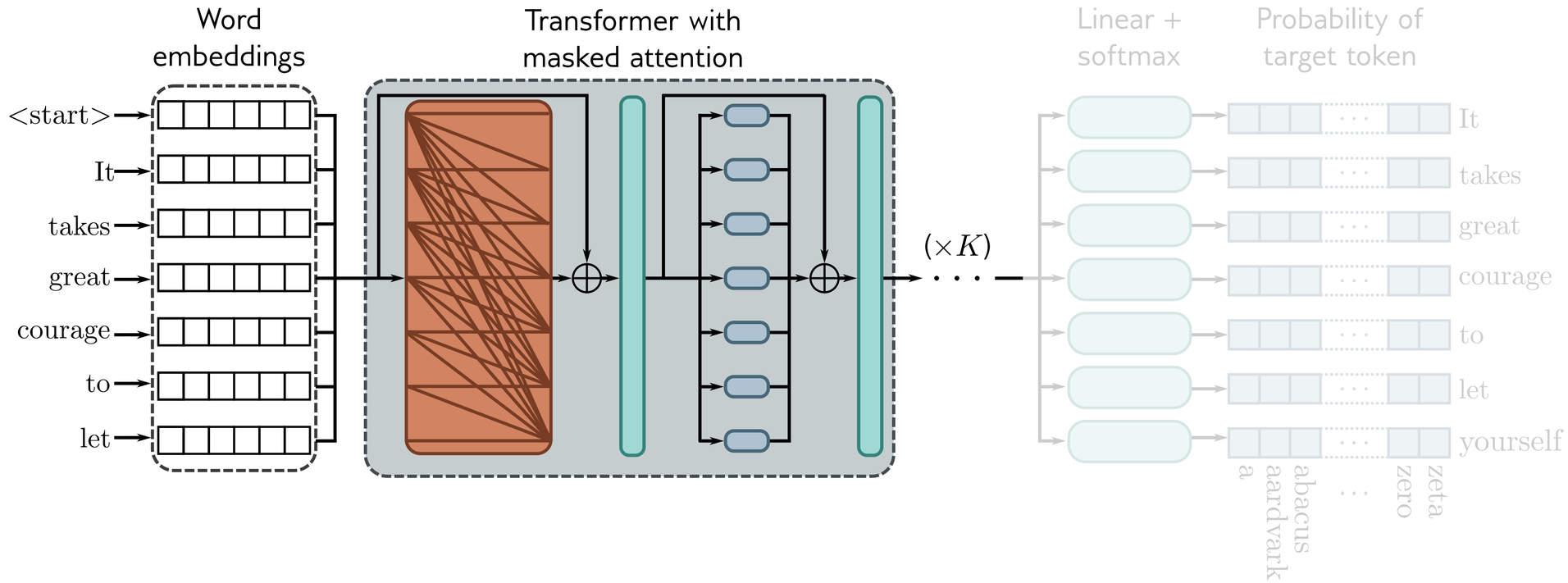
Each word (token) is mapped into a high-dimensional vectors



Transformer architecture. Adapted from Understanding Deep Learning (Prince, 2023)

How do language models work?

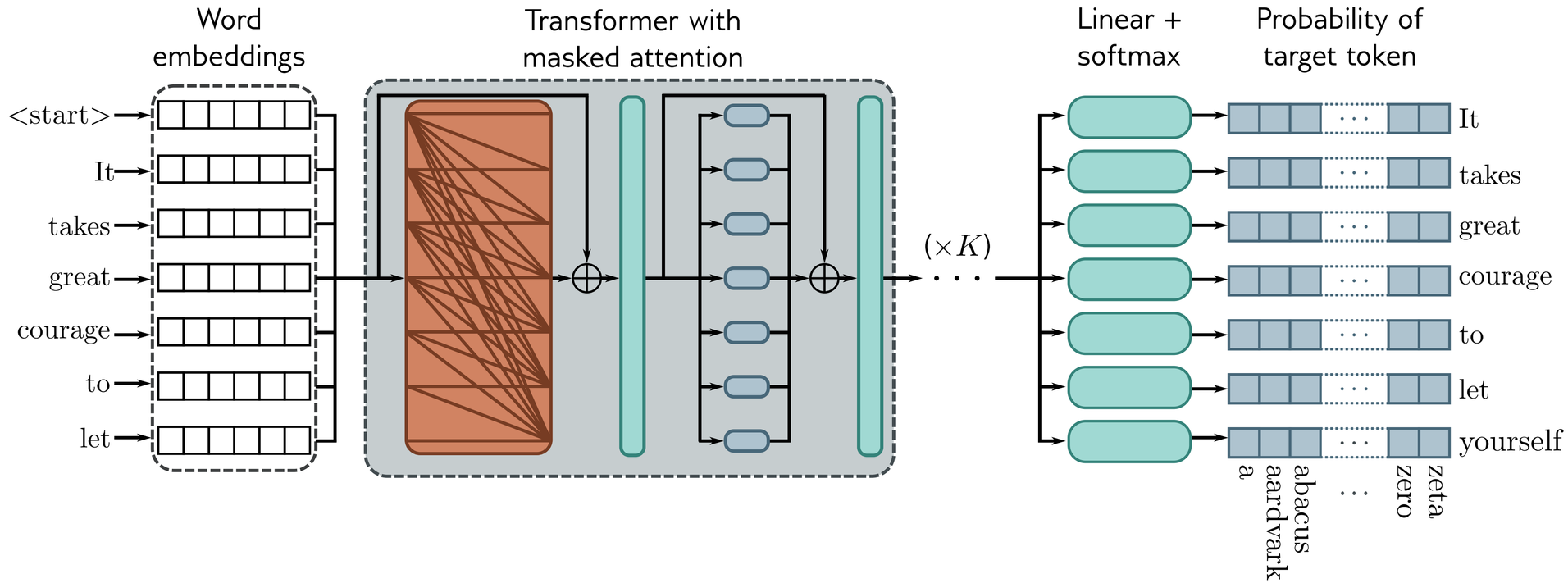
We have K layers of transformer blocks, consisting of masked attention and MLPs



Transformer architecture. Adapted from Understanding Deep Learning (Prince, 2023)

How do language models work?

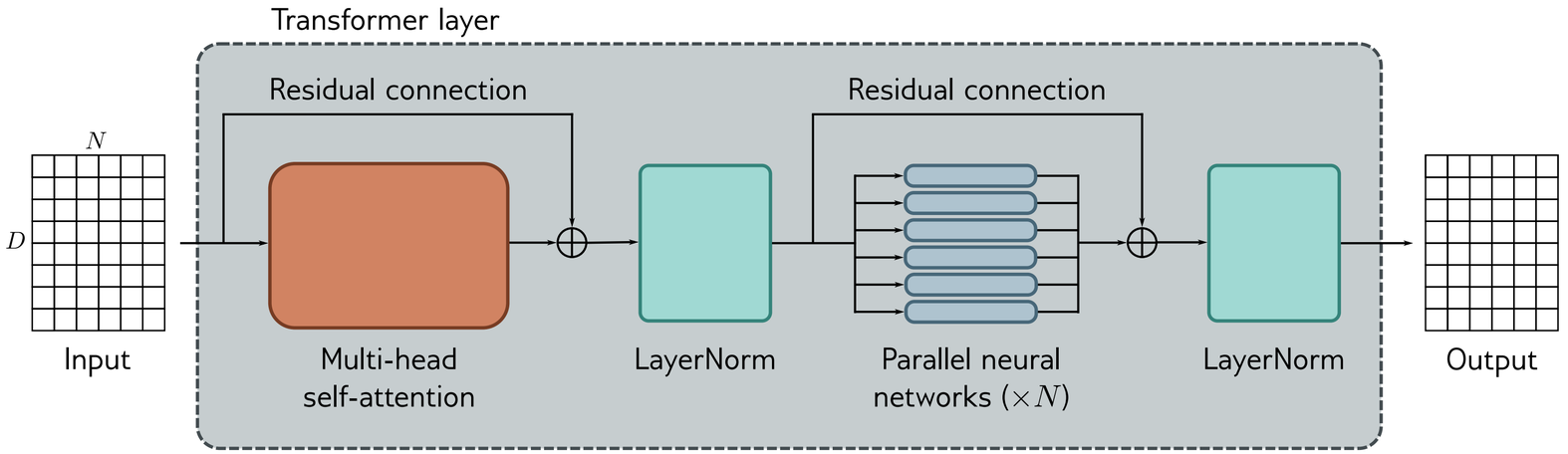
The vector is used to compute a probability distribution over the tokens (next token prediction)



Transformer architecture. Adapted from Understanding Deep Learning (Prince, 2023)

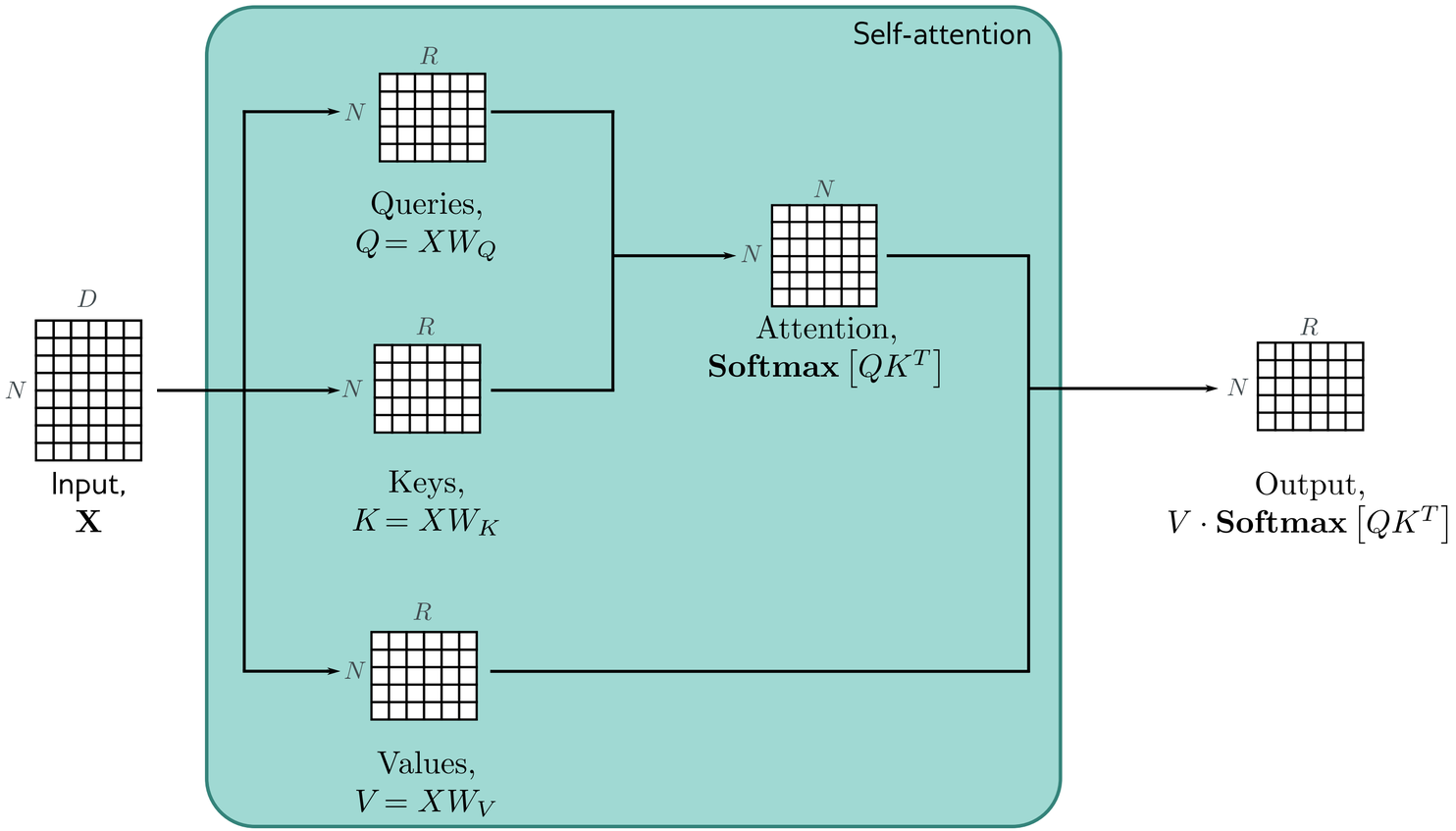
The transformer block

The vectors live in a shared space called the residual stream:
a "shared whiteboard" that every component can read from and write to



Transformer block. Adapted from Understanding Deep Learning (Prince, 2023)

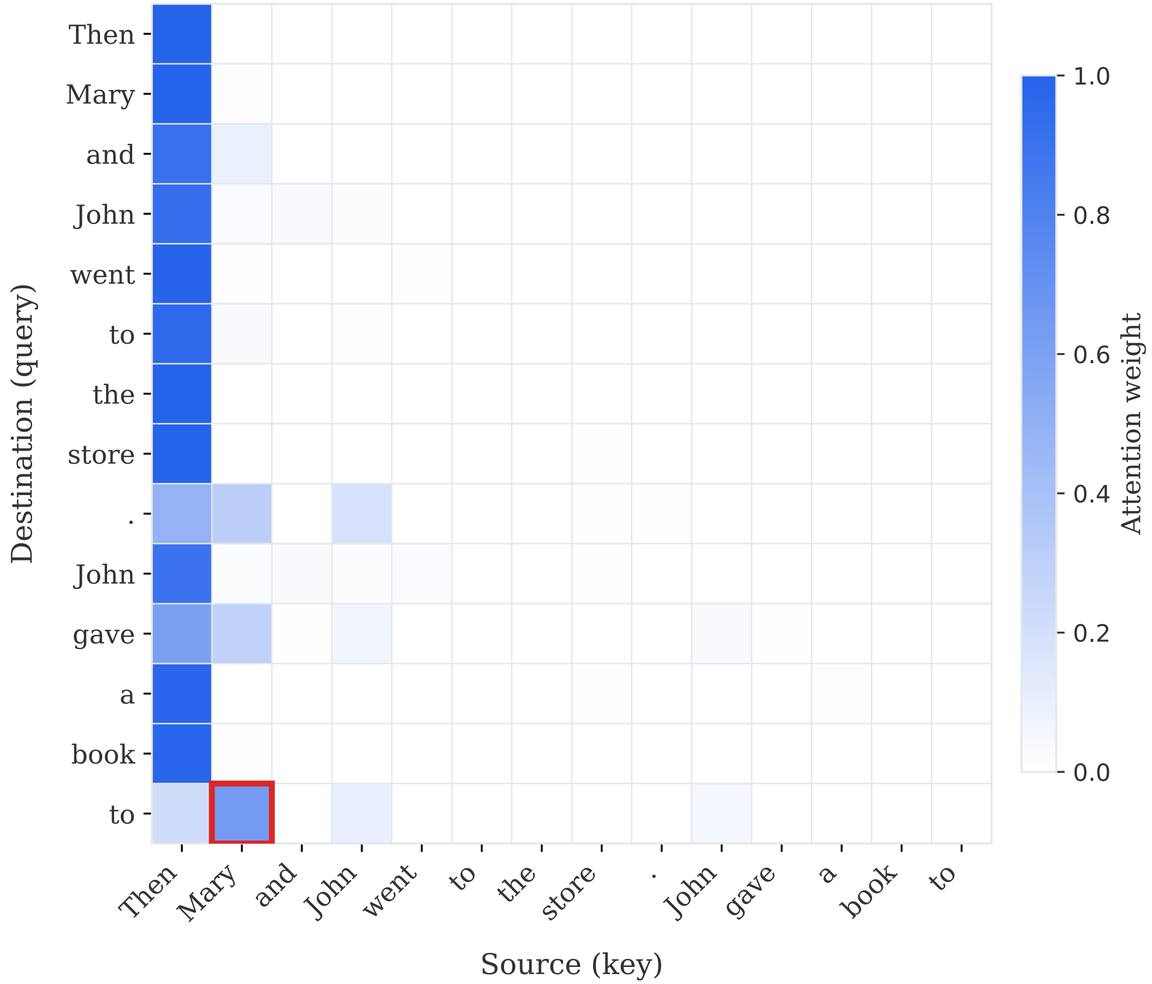
The attention mechanism



Adapted from Understanding Deep Learning (Prince, 2023)

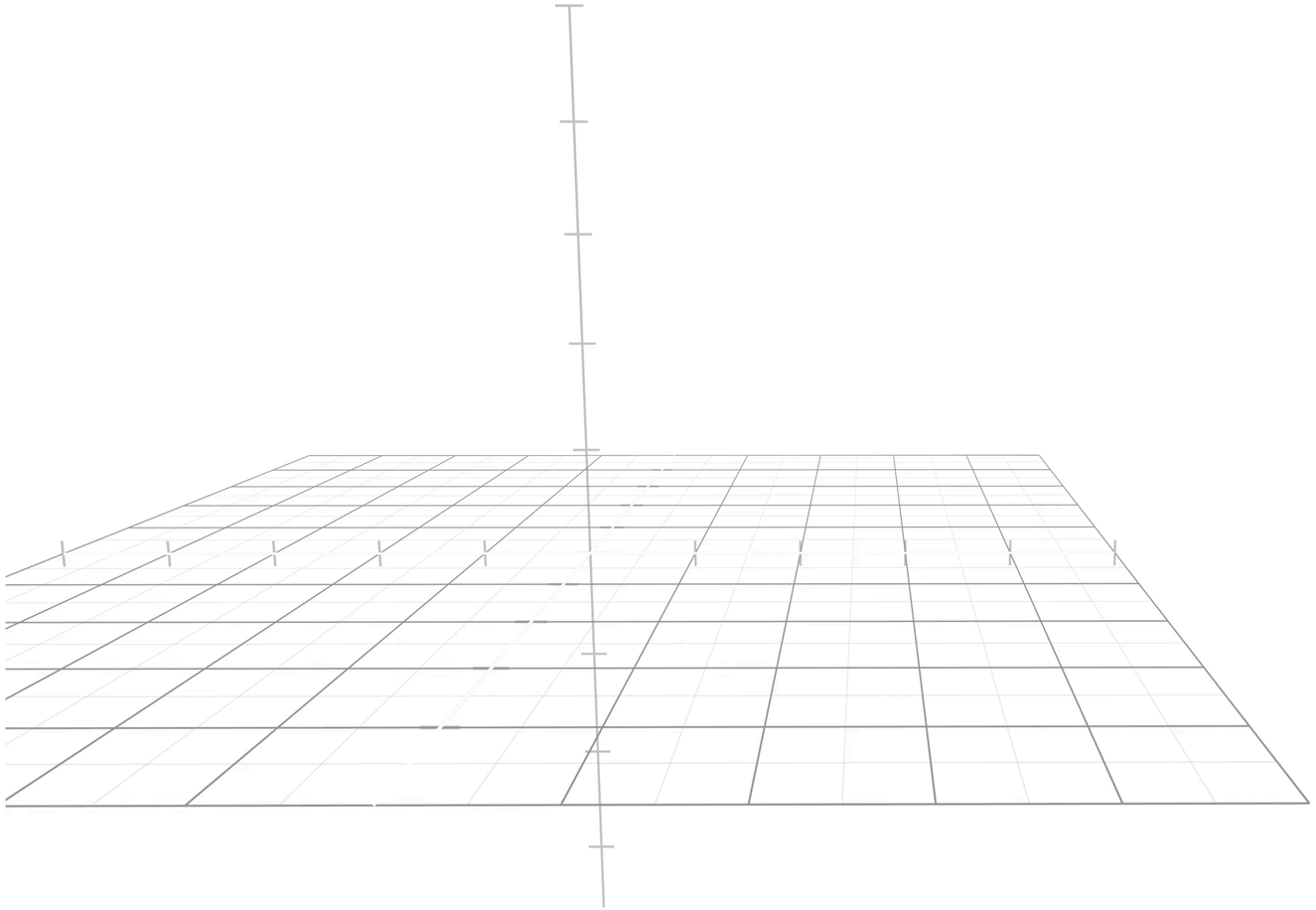
What is the attention matrix?

Every row of this matrix is a probability distribution: values between [0, 1], and they sum to 1



For every destination token, the attention head decides which tokens it needs to attend to (source tokens)

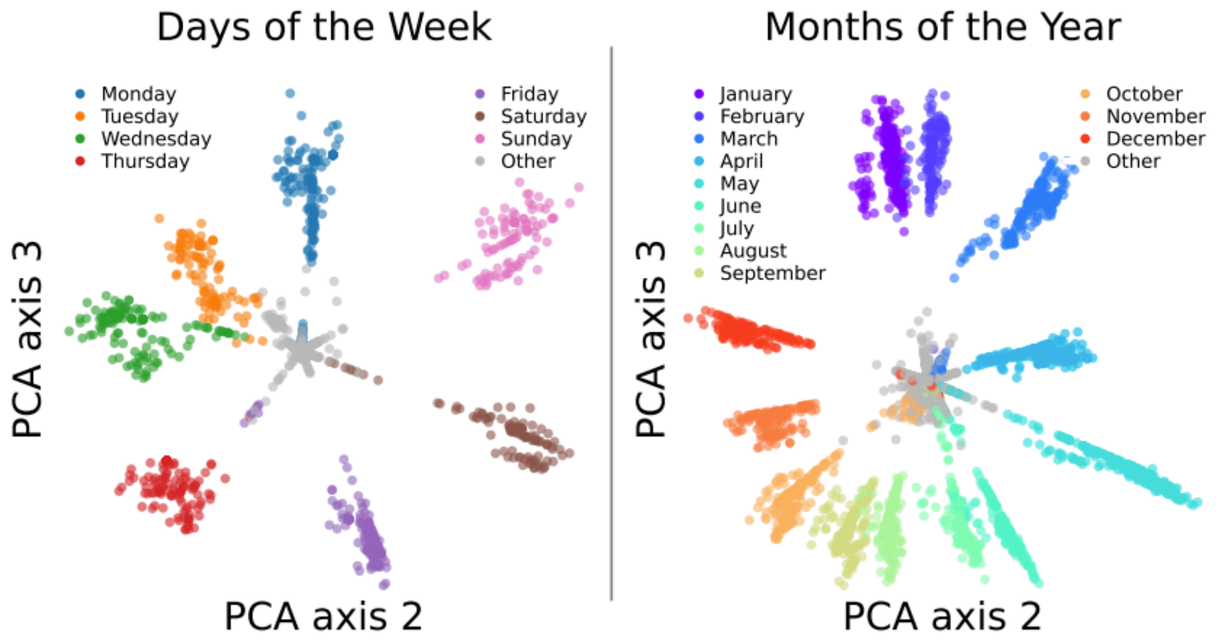
Features are directions in space




ALWAYS HAS BEEN.
WAIT, IT'S ALL LINEAR ALGEBRA?
imgflip.com

Representations are geometric

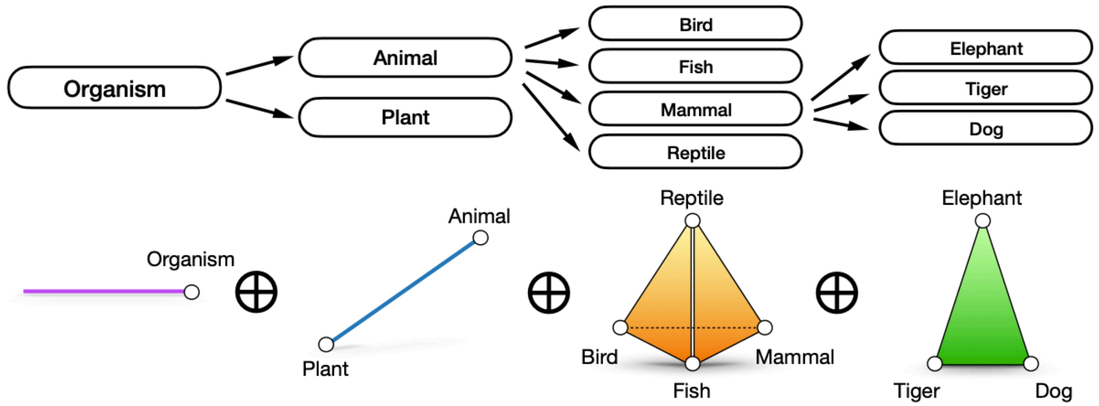
Days of week / months of year form circles in representation space

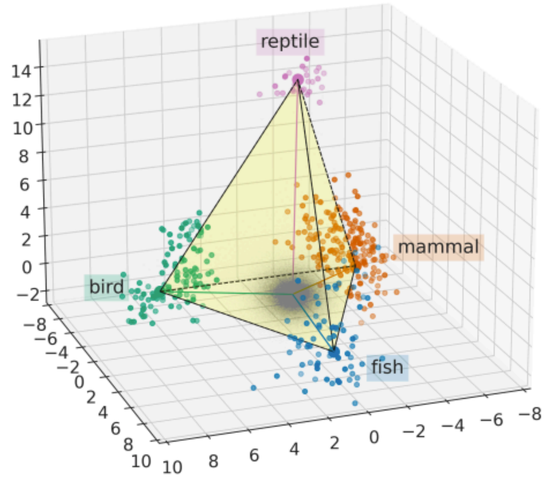


Not All Language Model Features Are One-Dimensionally Linear (Engels et al, 2024)

Representations are geometric

Concepts are encoded hierarchically in LLMs



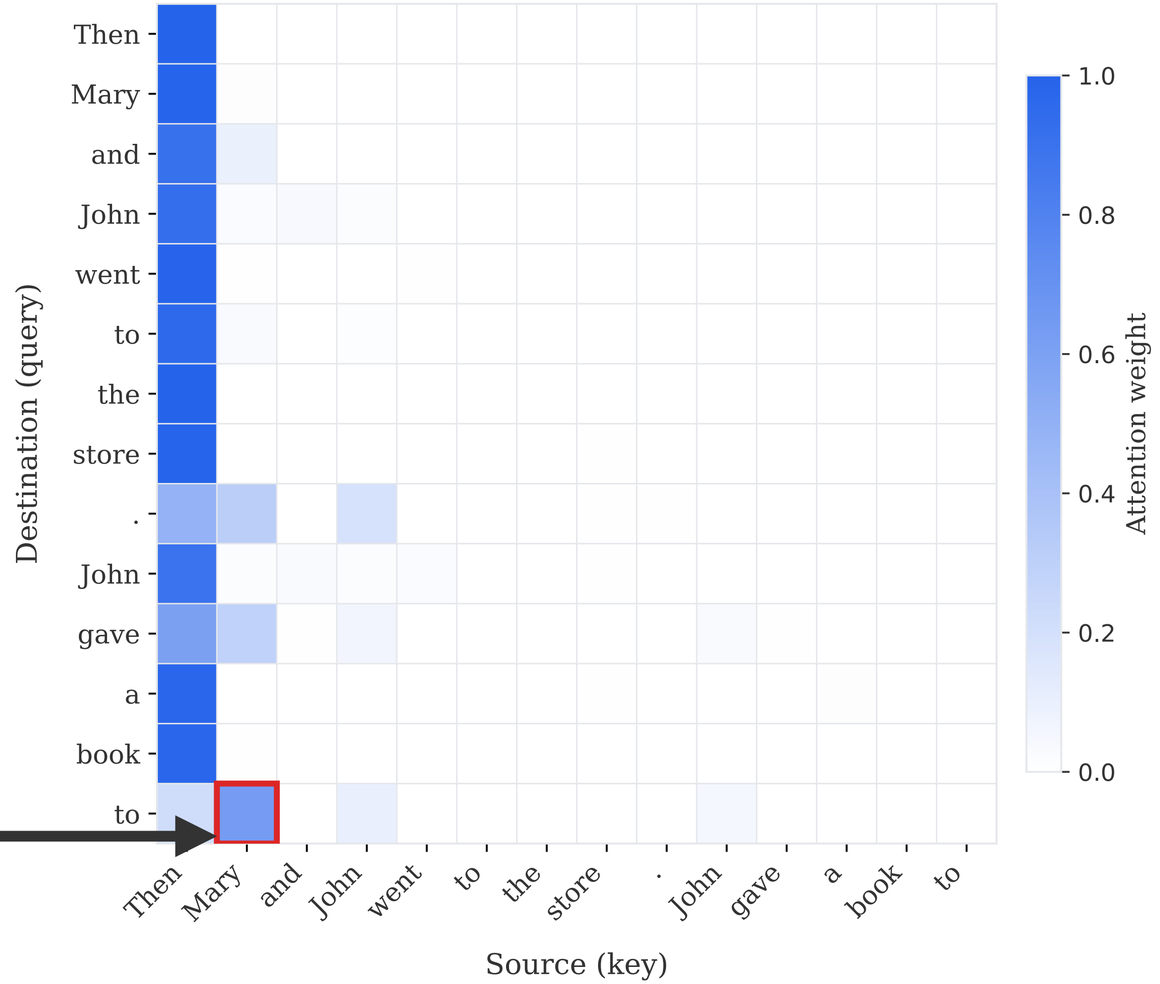


The Geometry of Categorical and Hierarchical Concepts in Large Language Models (Park et al, 2024)

My research

Understanding attention from first principles: From singular vectors to interpretable causal circuits

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

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

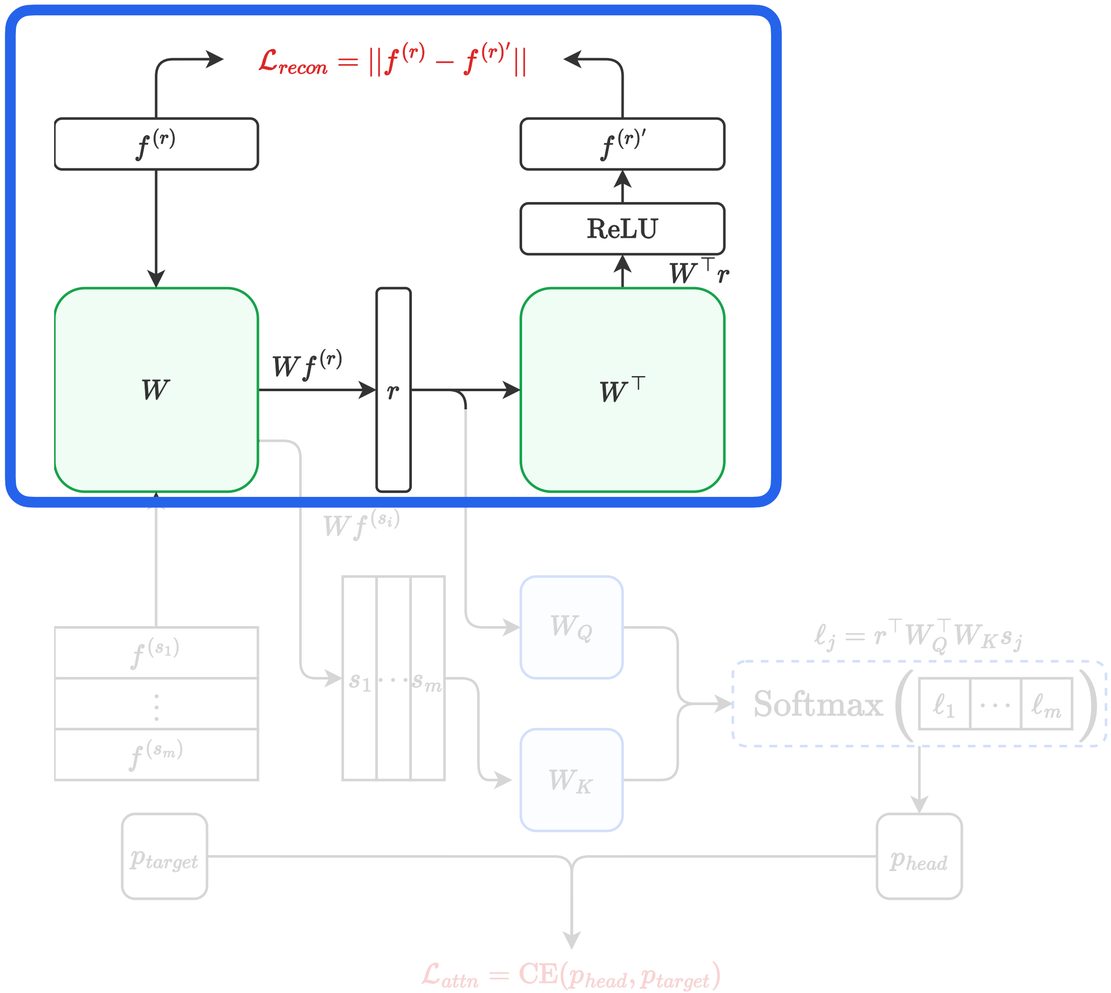
This is a core assumption underlying this work

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):

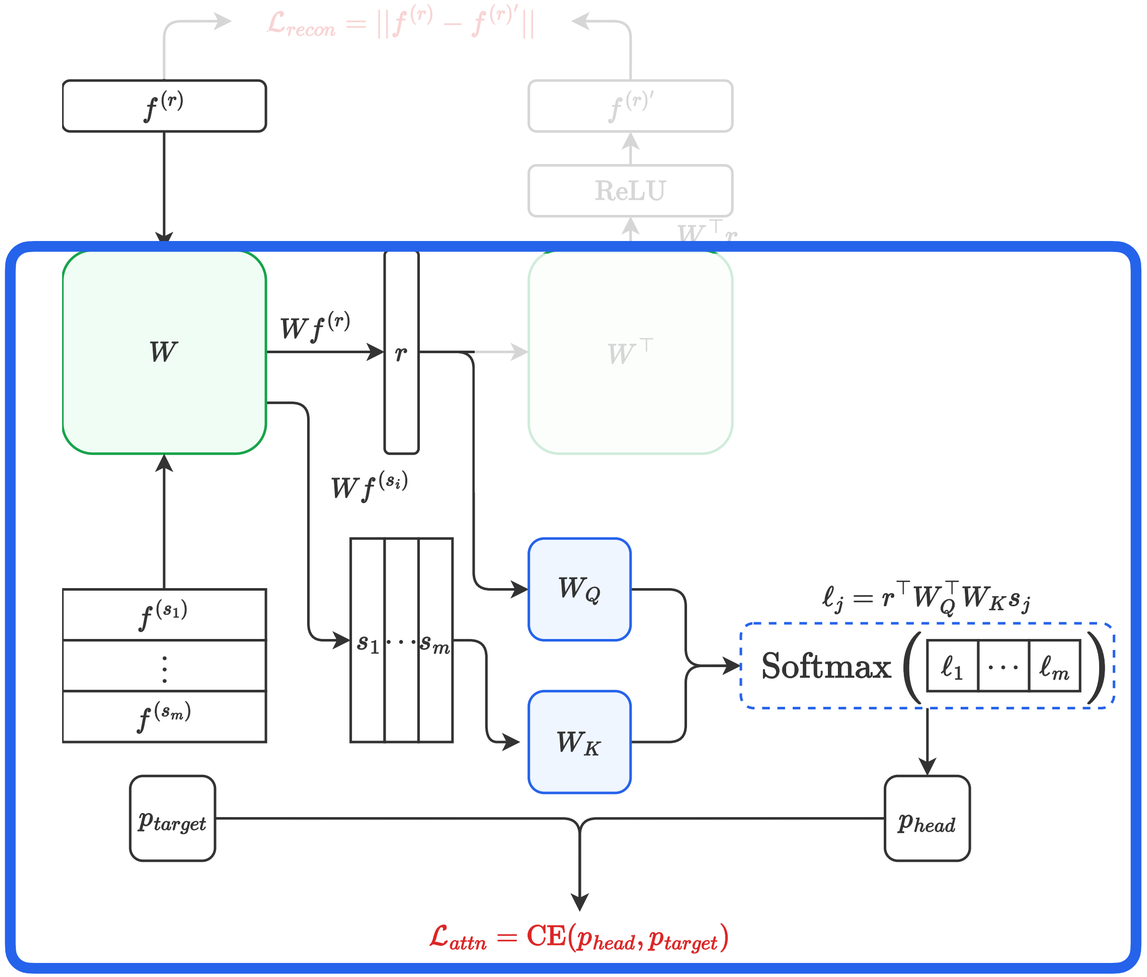
$$r = \sum_i f_i^{(r)} \mathbf{w}_i$$

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)

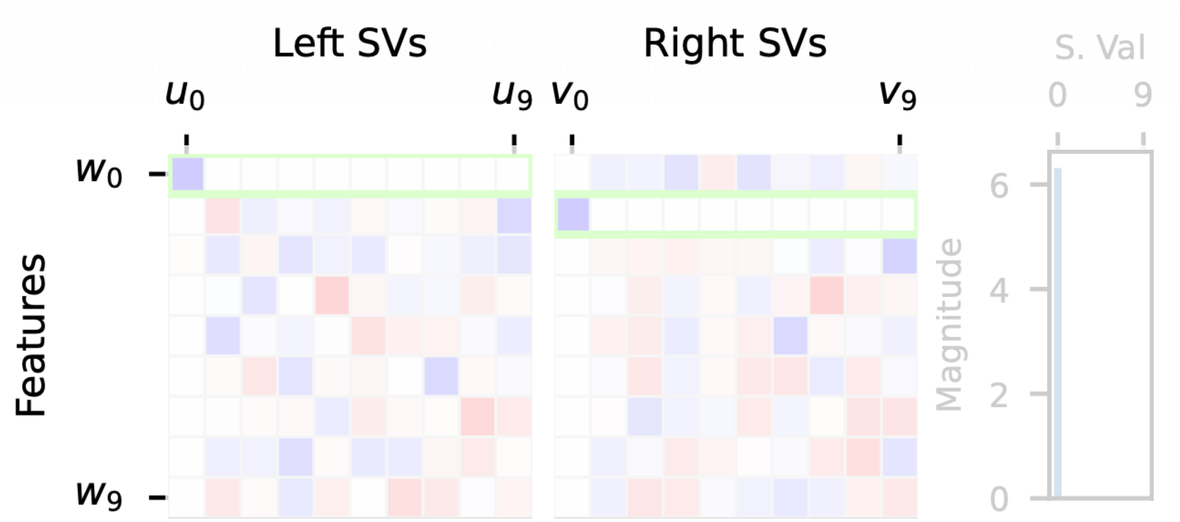
Attention is computed over a query token r and multiple source tokens $s_1, \cdots, s_m$

Combined loss:

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

Adding attention: singular vectors align with features

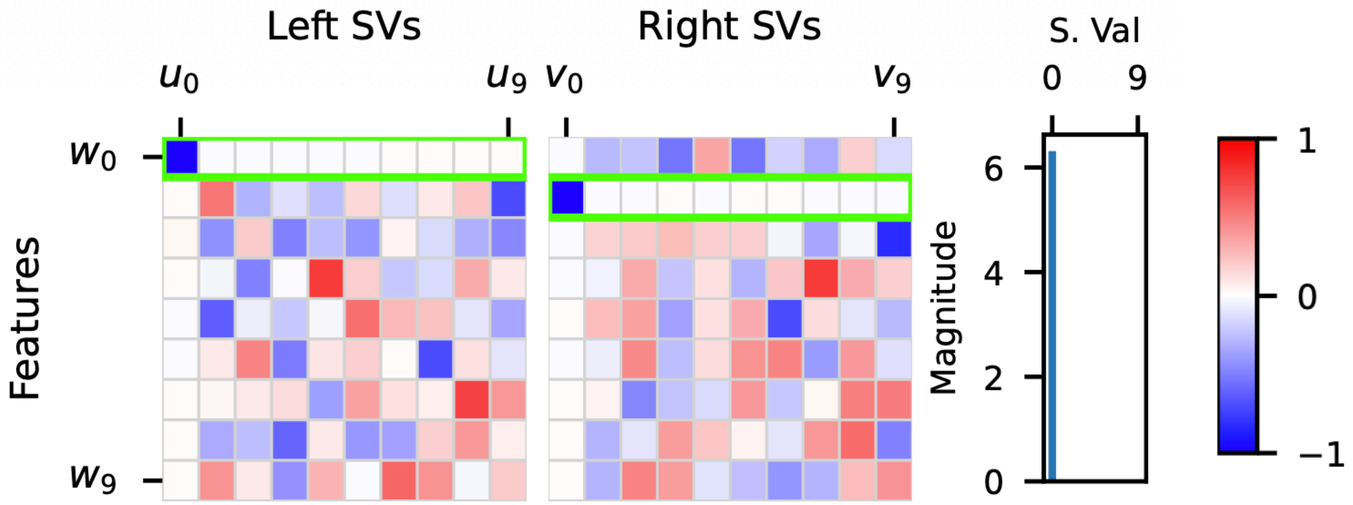
Simplest case:
$h^T \Omega s_i$ is high iff
h **contains** w_0
and
s_i **contains** w_1



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

SVF alignment justifies using singular vectors to identify features

We now leverage SVF to guide a search for the components responsible for writing them

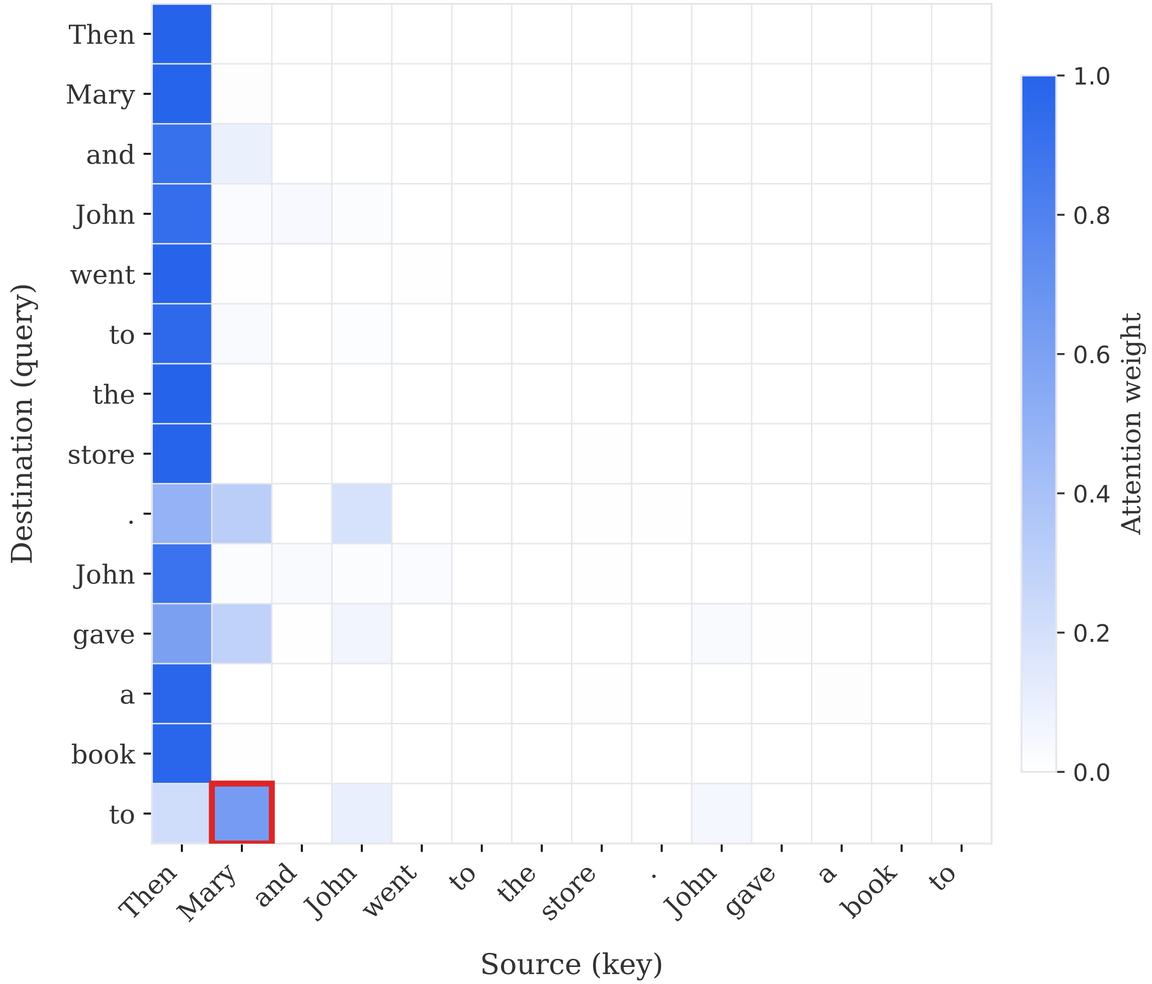
Who wrote those features? Are they causally responsible?

From geometry to attribution

Why does this head attend here?

Why the head decided that “to" has to attend to “Mary"?

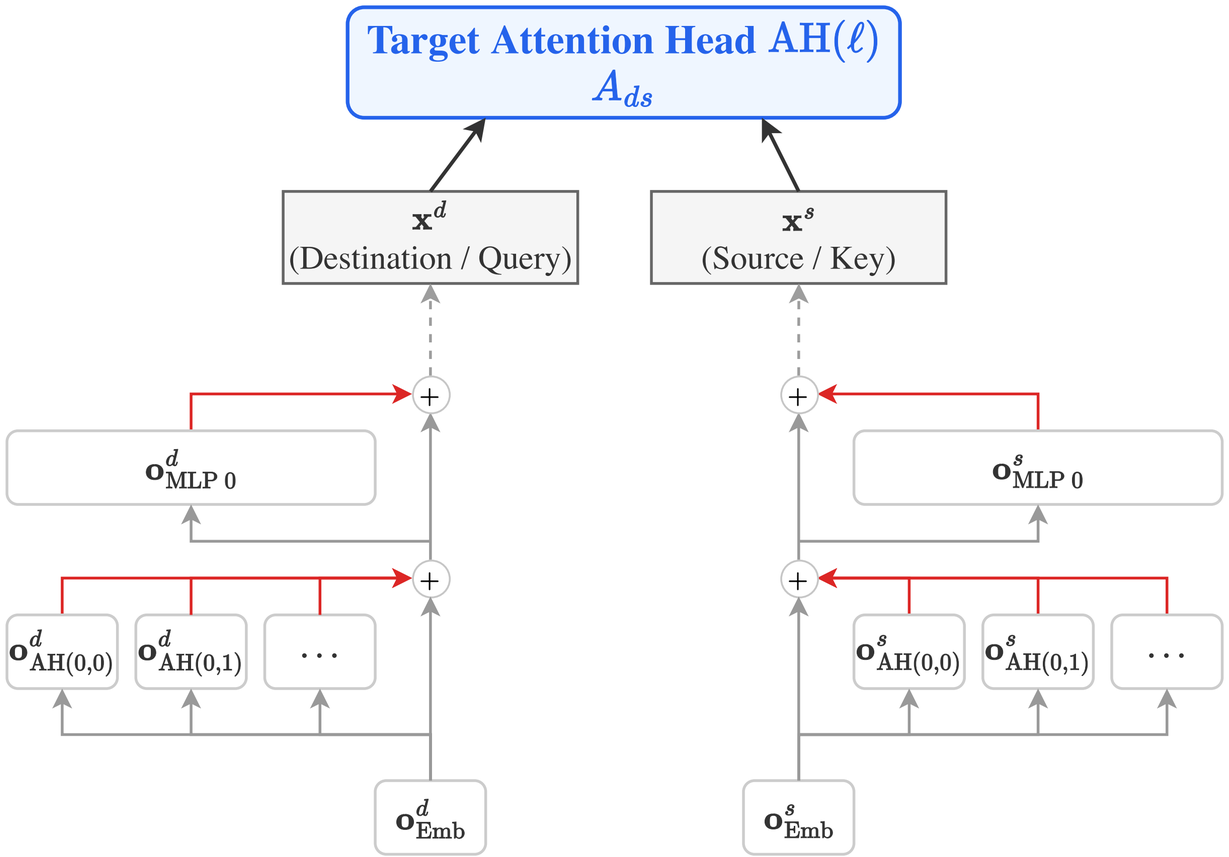
Let’s answer this question by solving two problems



1. **Which features are present?** SVF gives us that
2. **Which components wrote this feature in the residual?**

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

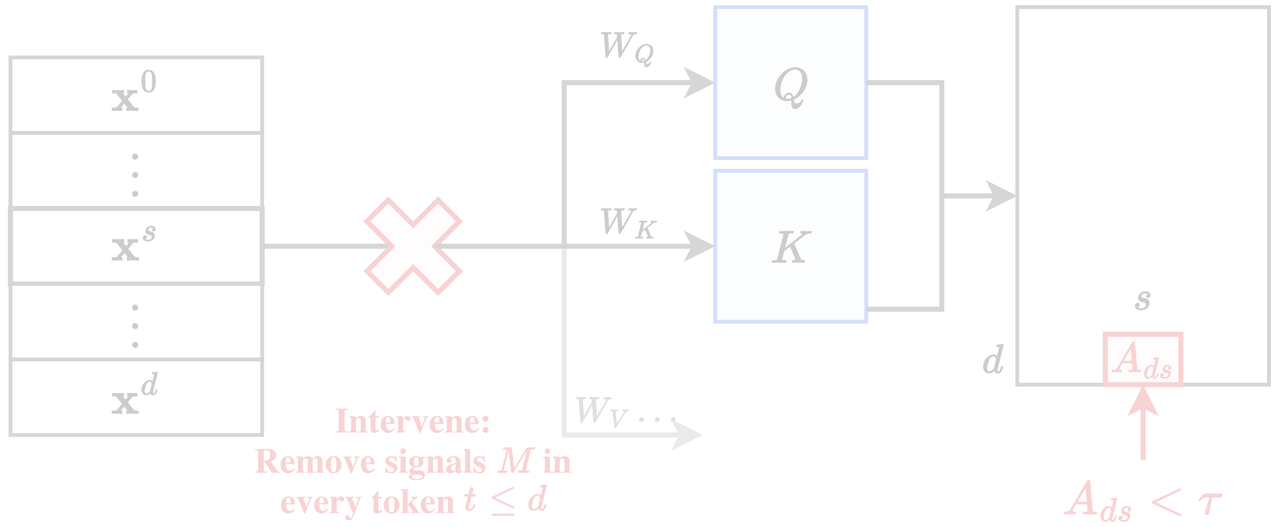


Which ones wrote features that are **causally** responsible for this attention weight?

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

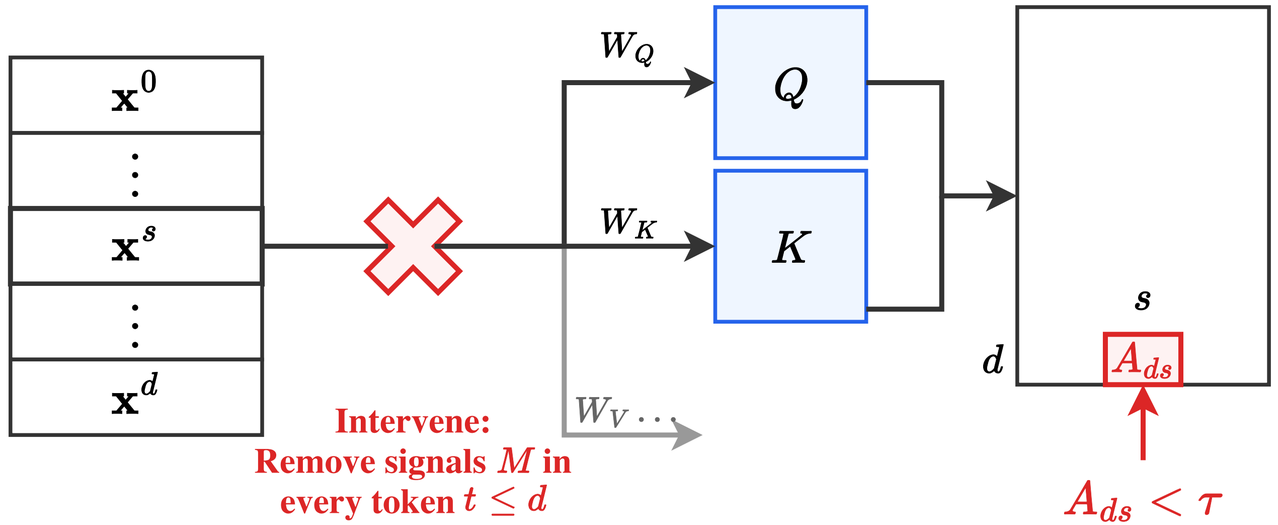
Find a minimal set M such that if we intervene by removing M, the attention A_{ds} drops below threshold τ.

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

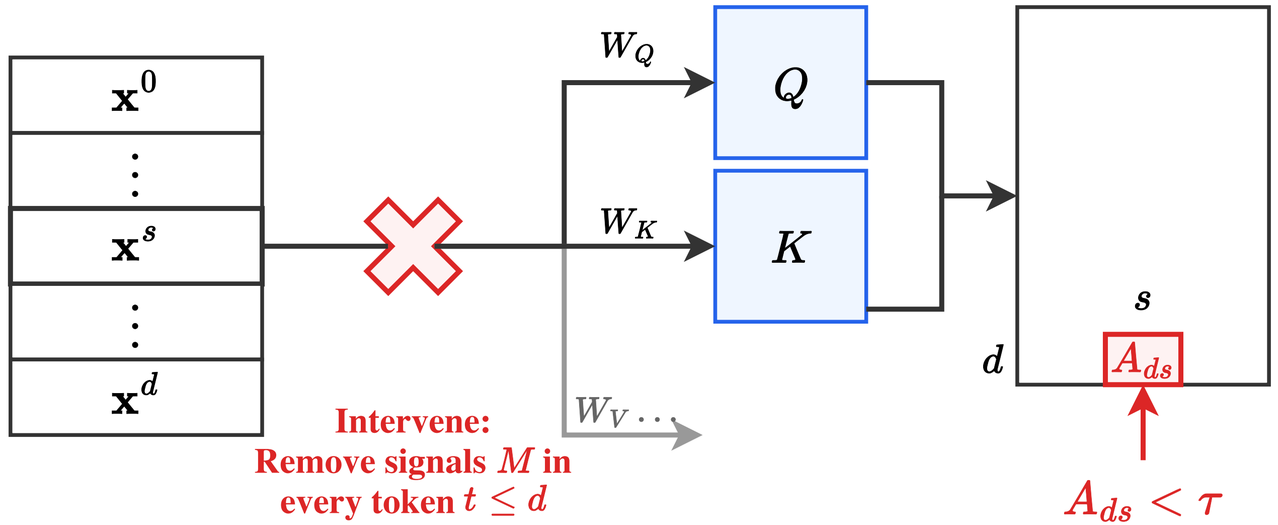
Find a minimal set M such that if we intervene by removing M, the attention A_{ds} drops below threshold τ.

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}.



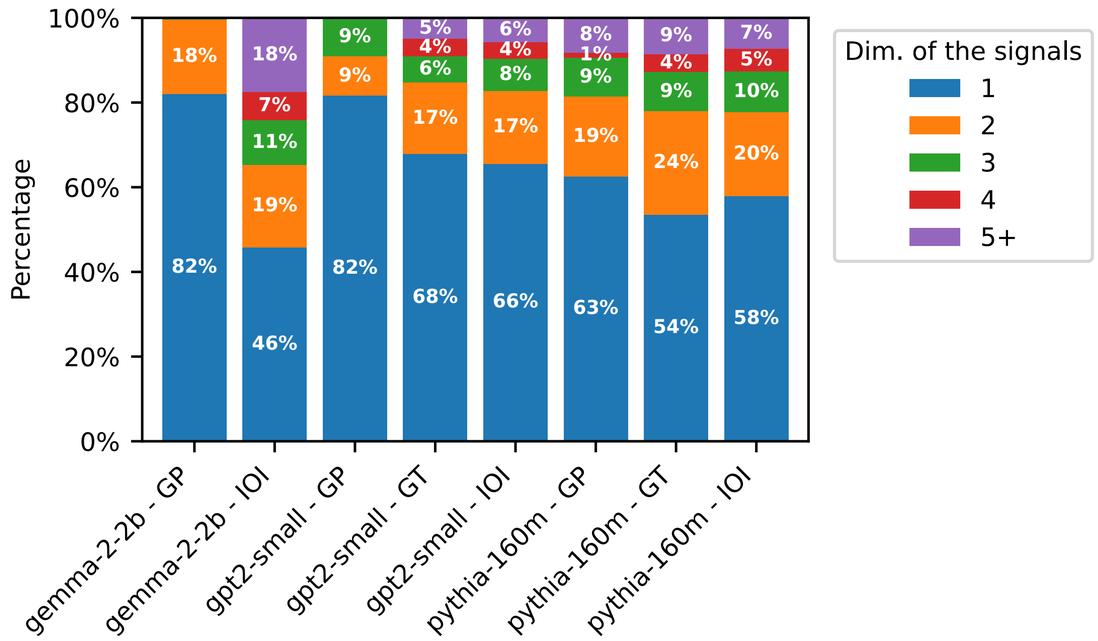
The attention causal communication problem

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 in a subspace with dim=1

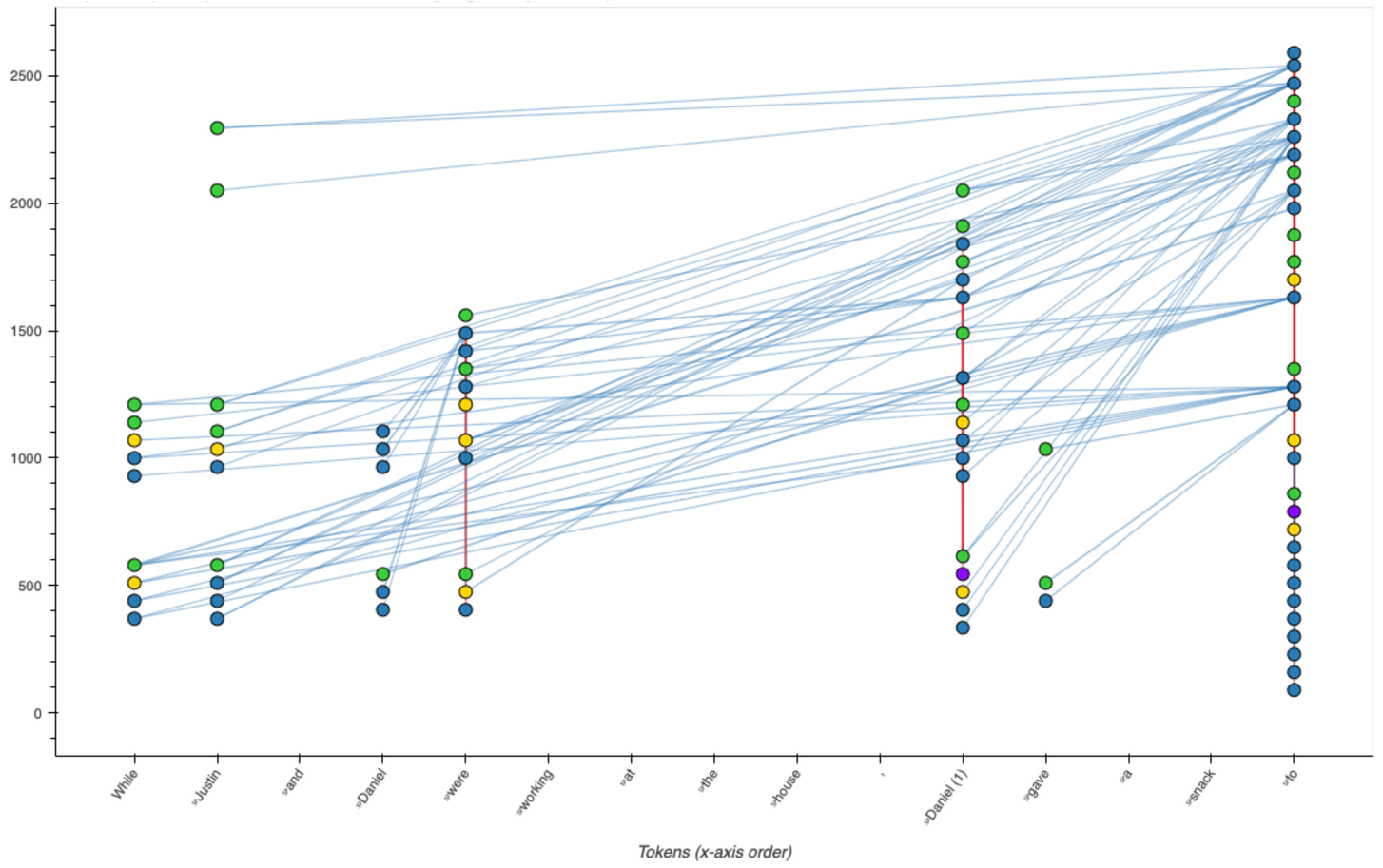


How do these causes compose into a circuit?

Per-prompt tracing and what it reveals

With signals, we can trace circuits

For a given prompt, a **circuit** is the subgraph of model components connected by **low-dimensional causal signals.** Each edge identifies which component wrote which feature that was read by a downstream attention head



We will study circuits in a very simple task, called Indirect Object Identification (IOI)

Testbed: Indirect Object Identification (IOI)

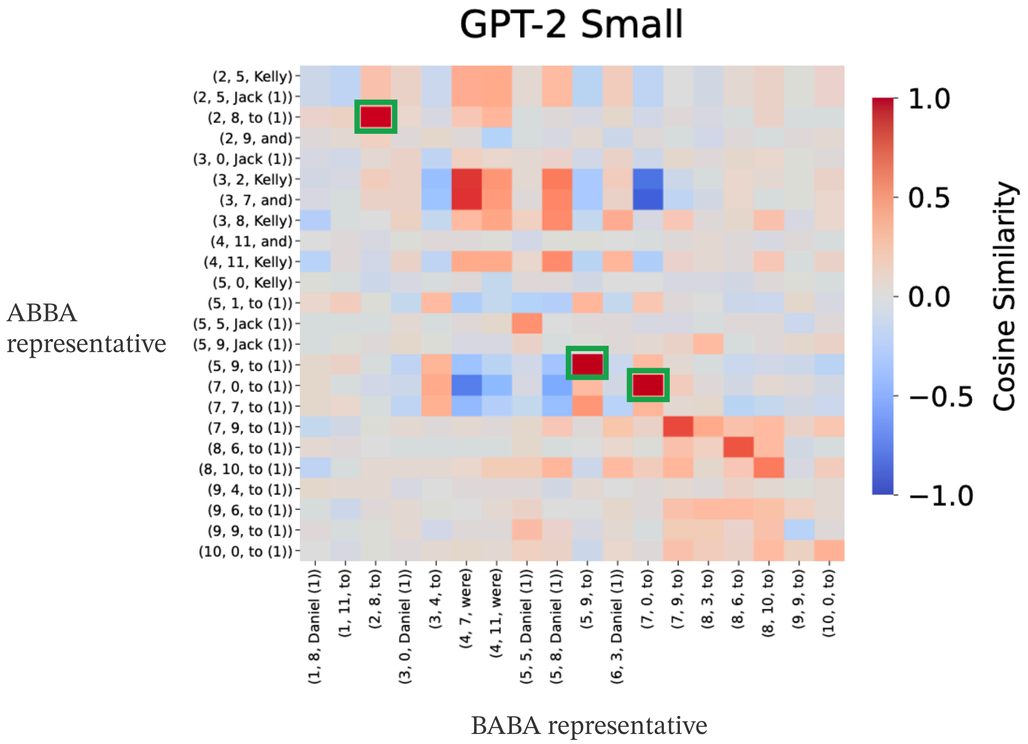
	High-level template	Low-level template
ABBA	Then, [A] and [B] went to the office. [B] gave a computer to [A]	Then, [A] and [B] went to the office. [B] gave a computer to [A]
	vs.	vs.
BABA	Then, [B] and [A] went to the office. [B] gave a computer to [A]	Friends [B] and [A] found a computer at the office.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

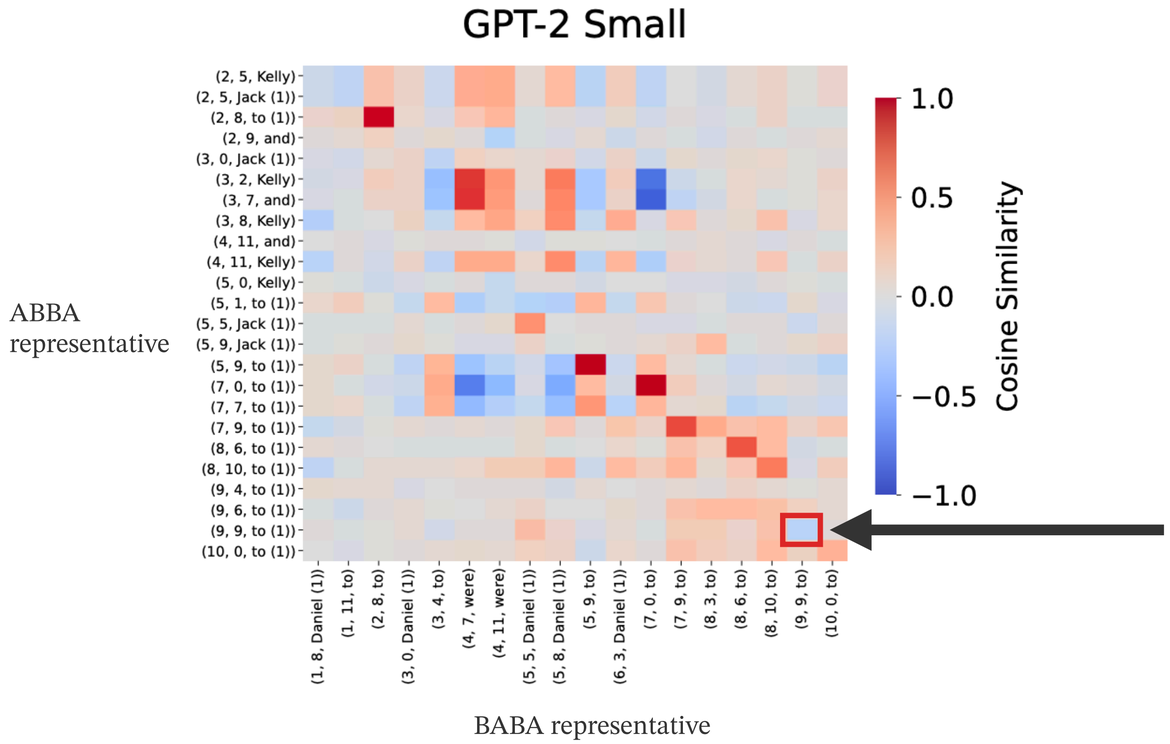
Signal-level comparison reveals identical components doing the same things

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

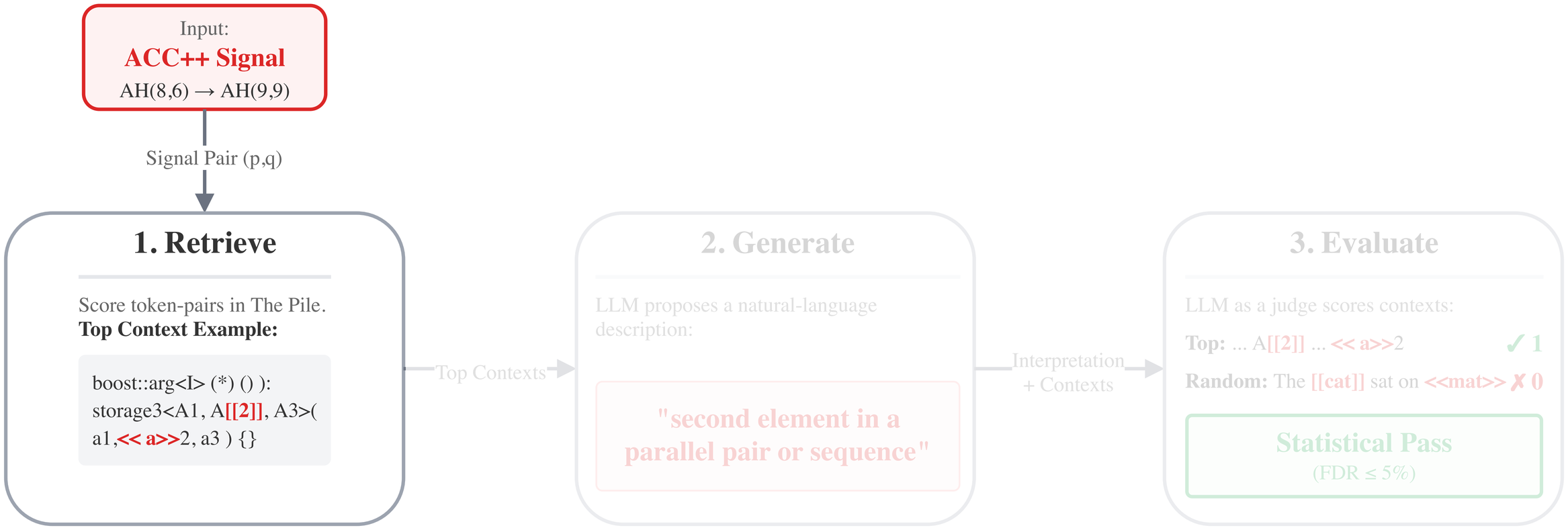
Signal-level comparison reveals identical components doing different things



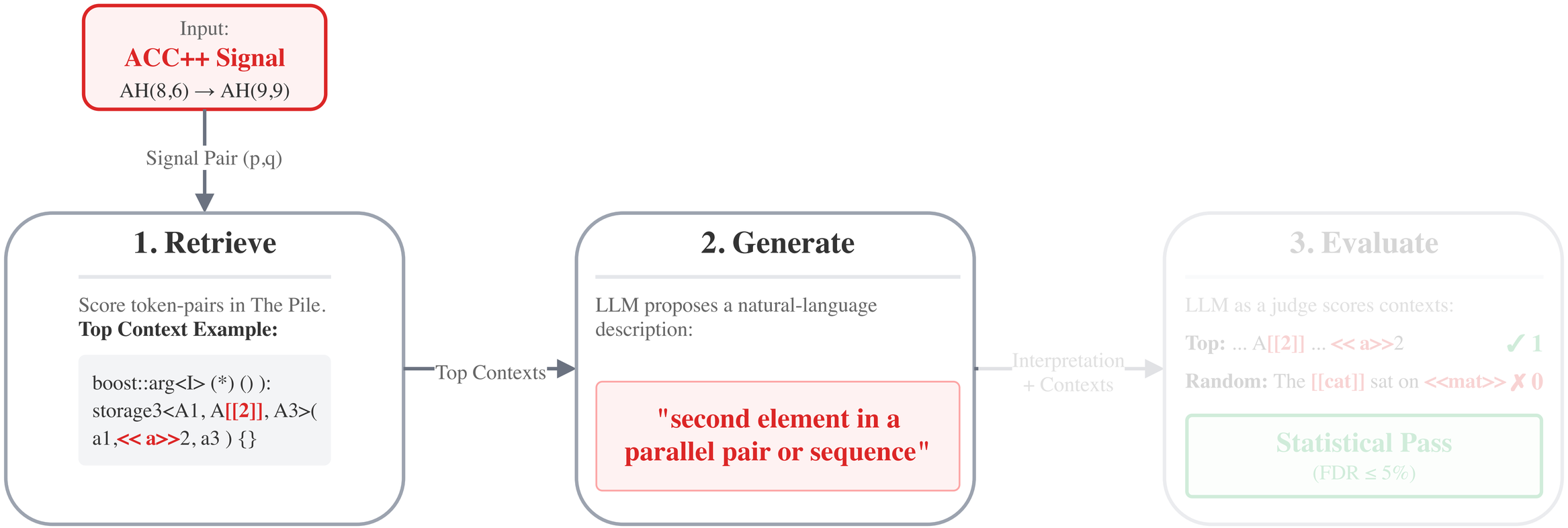
Changing mechanism: the canonical name-mover Head (9,9) shows *negative* cosine similarity between ABBA and BABA

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

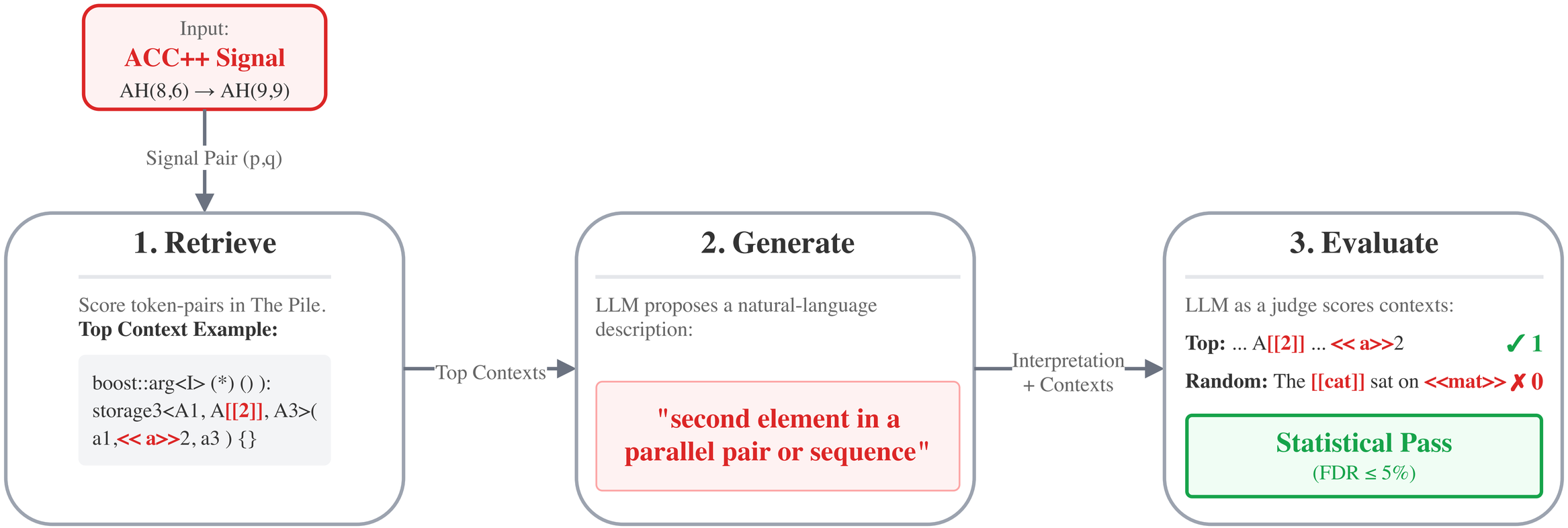
Autointerpretation pipeline for ACC++ signals



Autointerpretation pipeline for ACC++ signals



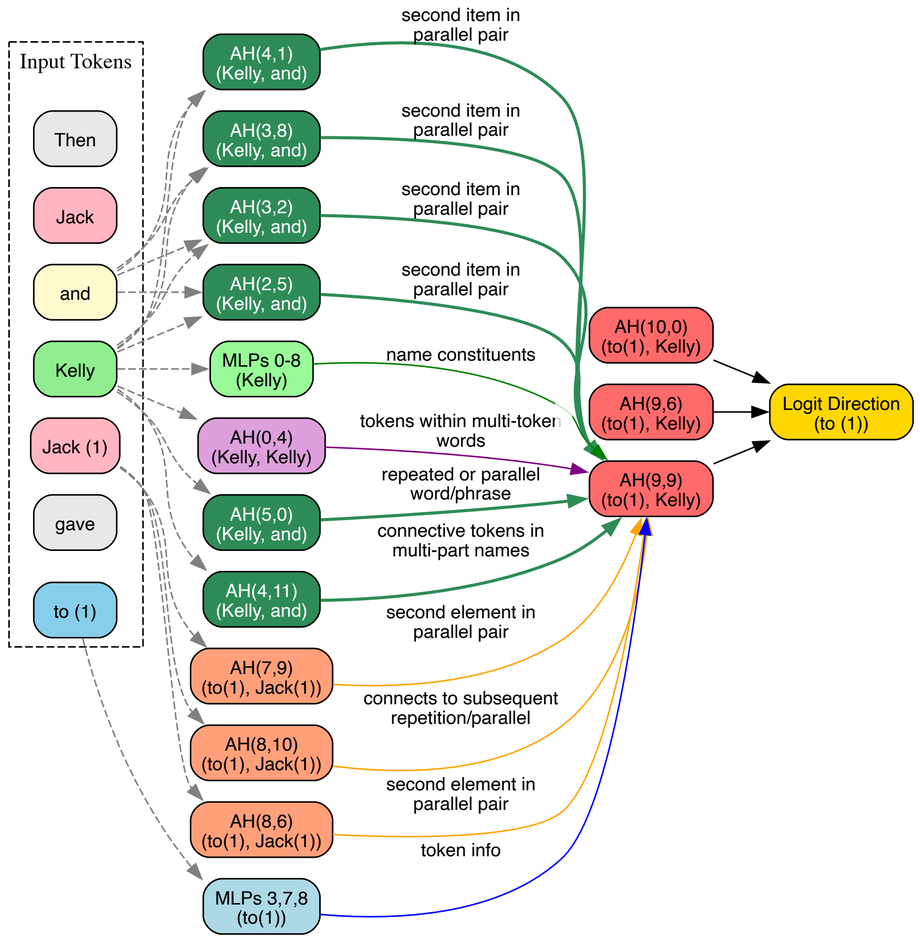
Autointerpretation pipeline for ACC++ signals



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

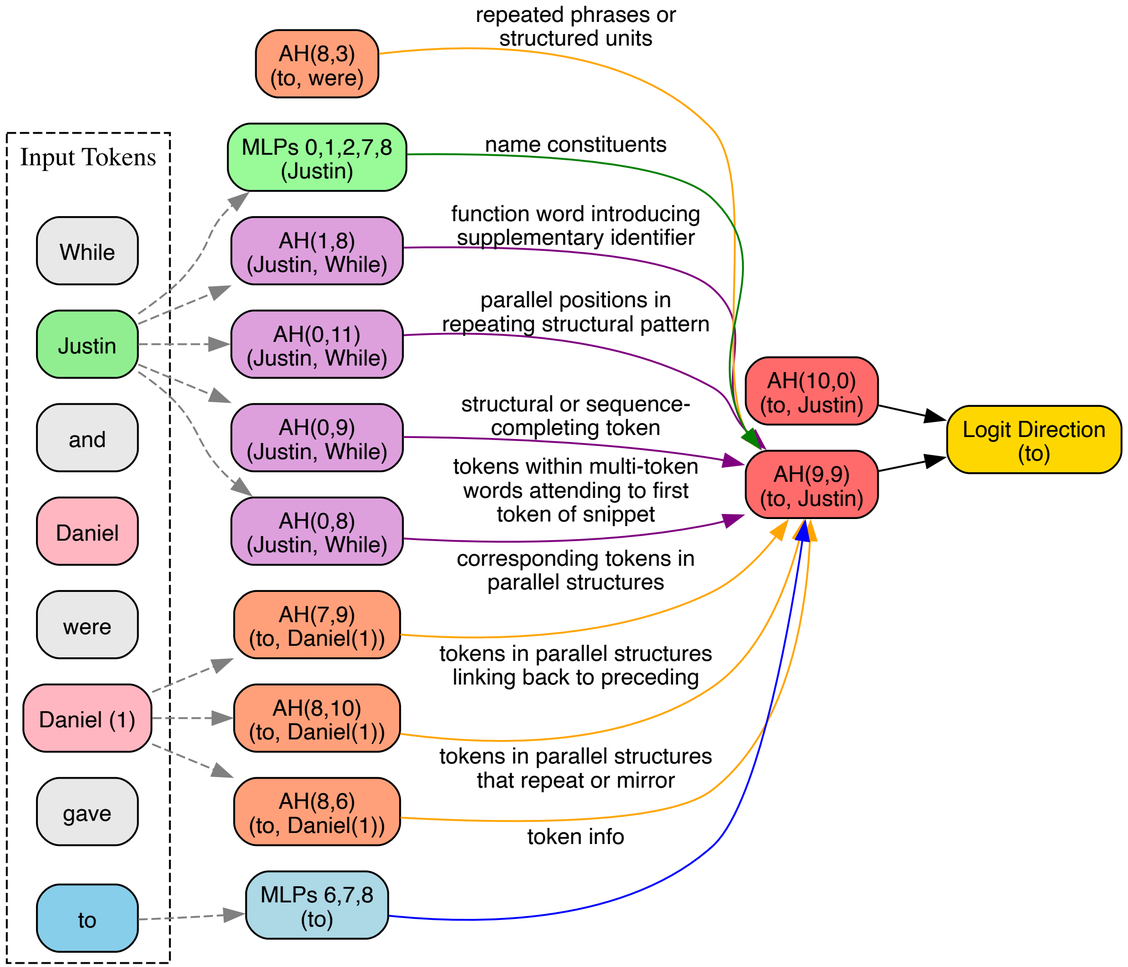


Multiple heads tag the IO token (“Kelly”) as “second item in a parallel pair”

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

Interpretable signals explain the algorithmic difference

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"

Same task, different algorithms

BABA

IO found via **semantic role**: “second item in a parallel pair”

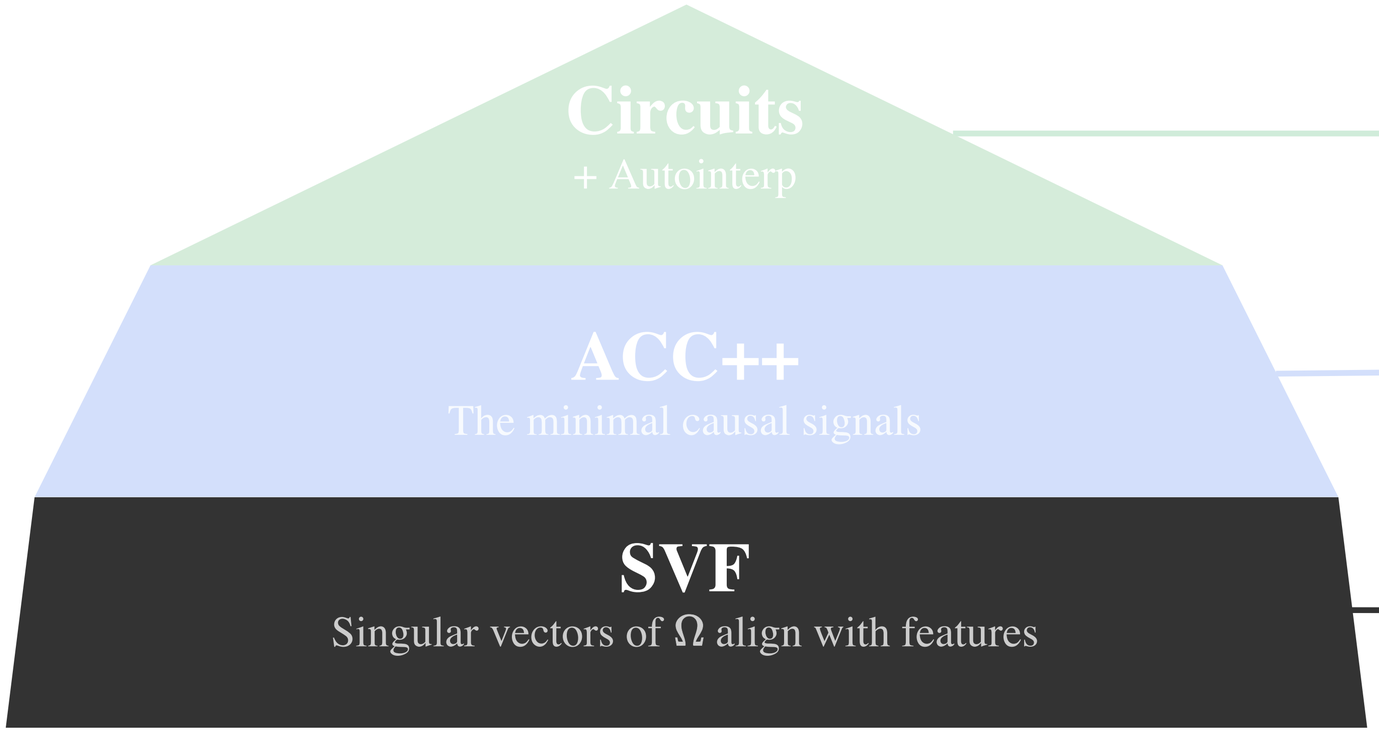
ABBA

IO found via **structural position**: "parallel positions in repeating pattern”

There is no single “IOI algorithm”

The model has **multiple strategies** and selects based on input structure

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

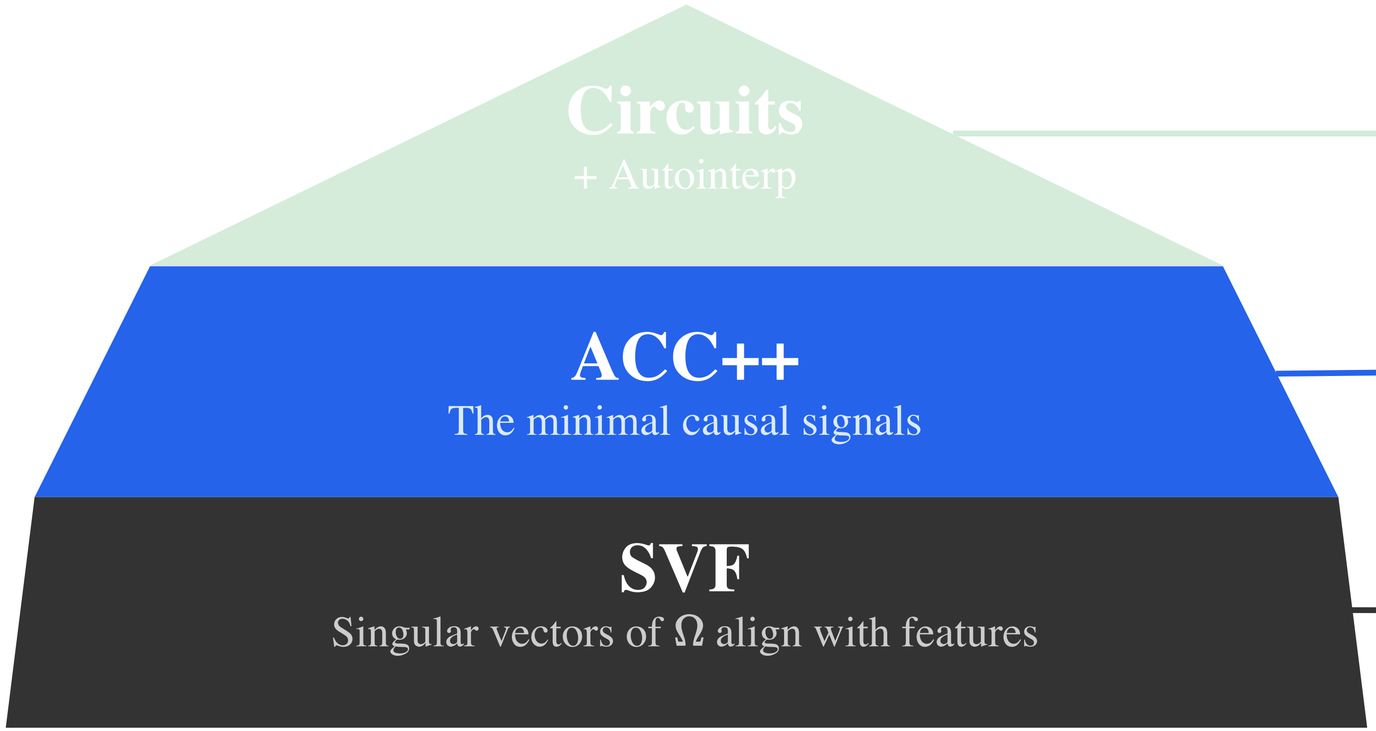


Causal and interpretable edges
Prompt-specific
Single forward pass

Causal signals to attention are identified

Proven theoretically
Confirmed empirically

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



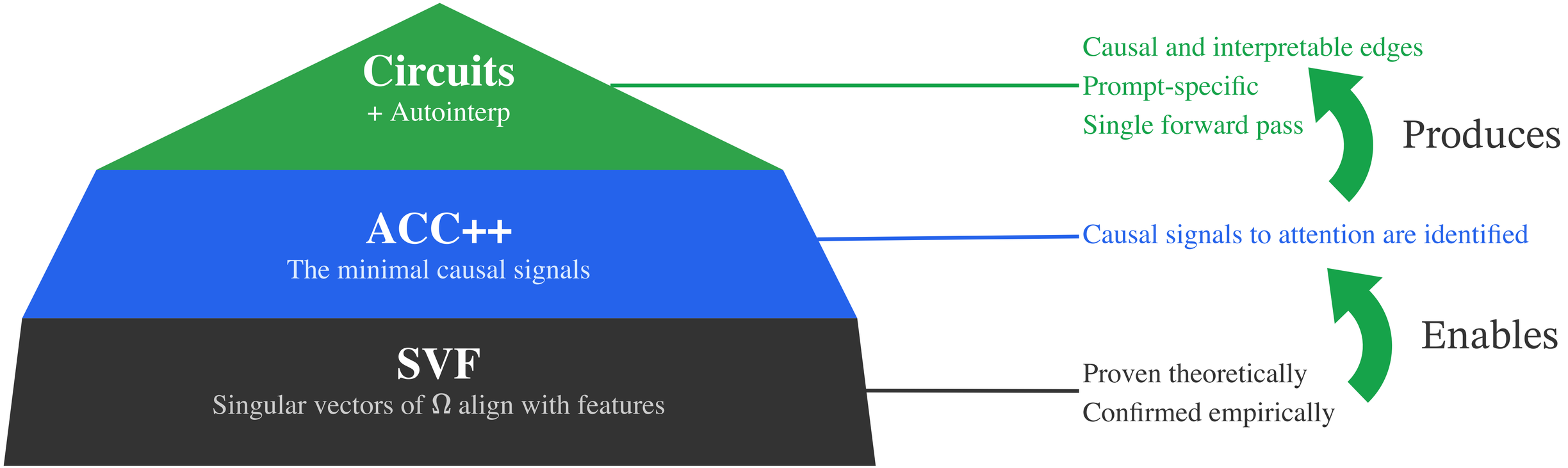
Causal and interpretable edges
Prompt-specific
Single forward pass

Causal signals to attention are identified

Enables

Proven theoretically
Confirmed empirically

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



The bigger picture

Models learn **bags of heuristics**, not single algorithms

LLMs are hard to understand: we need linear algebra, calculus, and statistics to start understanding AI systems from the inside

There is **a lot** to be done. This is an exciting and open area, with direct implications for how we test, audit, and trust AI systems

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