이화여자대학교
EWHA WOMANS UNIVERSITY

2026 - Winter

Transformer Circuits Thread

A Mathematical Framework for Transformer Circuits

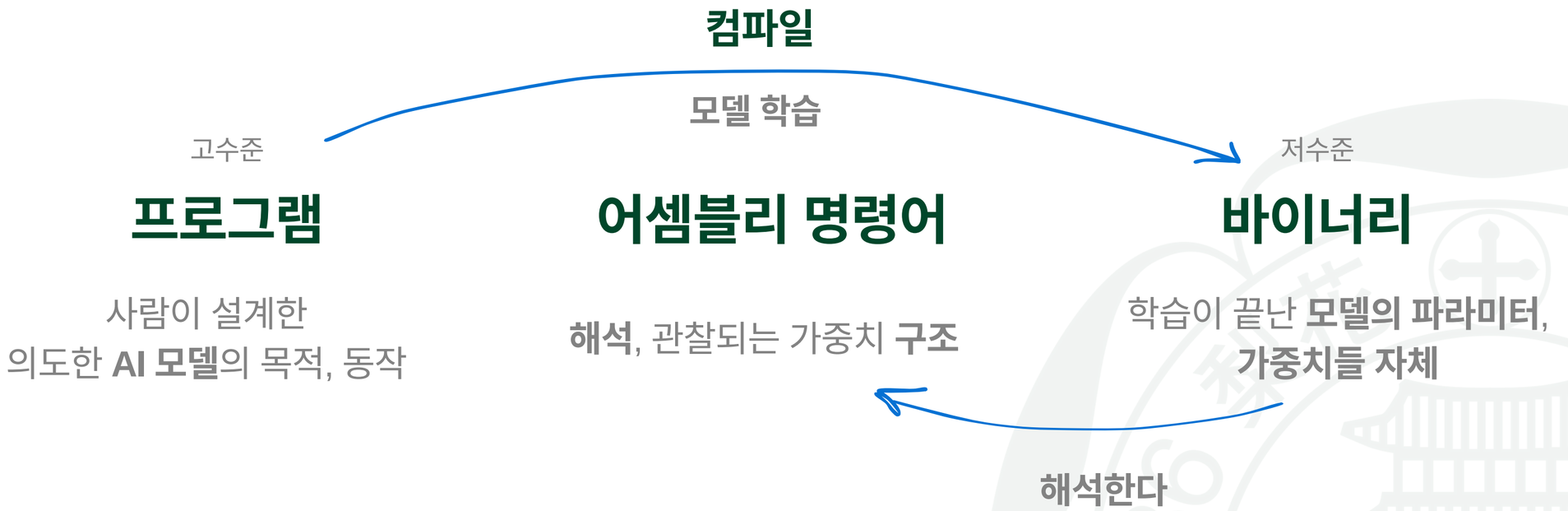
HAI Lab

2271064 한사랑
hangpfm0518@ewhain.net

Takeaway

- What is Mechanistic Interpretability, and why this paper matters
- What this talk will not focus on
- Goal of this seminar

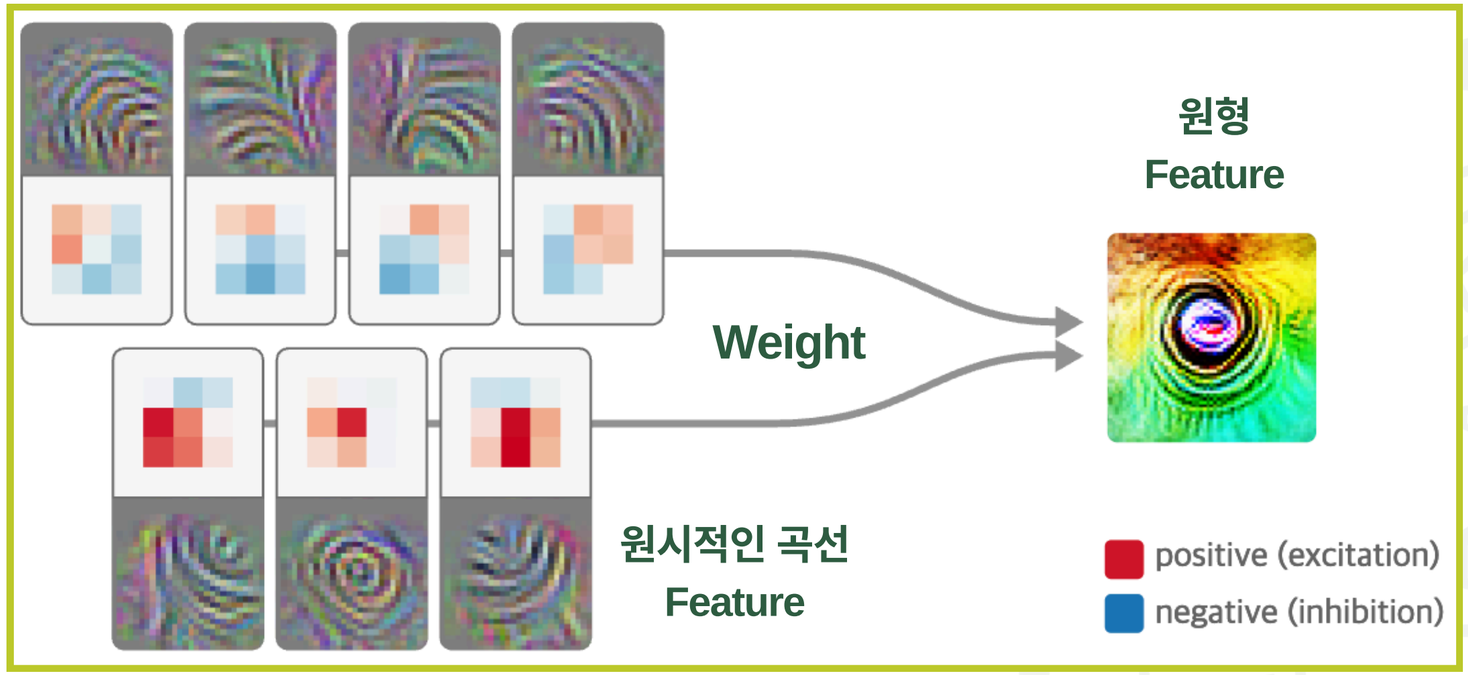
What is Mechanistic Interpretability?



Mechanistic Interpretability aims to *reverse-engineer* neural networks at the level of **features** and **circuits**, rather than treating them as black boxes.

What are ‘Features’ and ‘Circuits’

Circuit



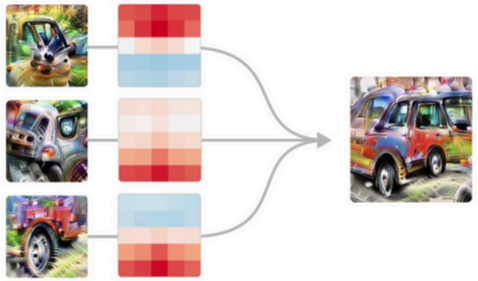
Why This Paper Matters

This paper is **an early, foundational attempt** to formalize Mechanistic Interpretability for **Transformer** (Language Model)

Thread: Circuits

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim

What can we learn if we invest heavily in reverse engineering a single neural network?



Vision Model
Thread: Circuits, 2020

A Mathematical Framework for Transformer Circuits

AUTHORS

Nelson Elhage*†, Neel Nanda*, Catherine Olsson*, Tom Henighan†, Nicholas Joseph†, Ben Mann†, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, Chris Olah‡

AFFILIATION

Anthropic

PUBLISHED

Dec 22, 2021

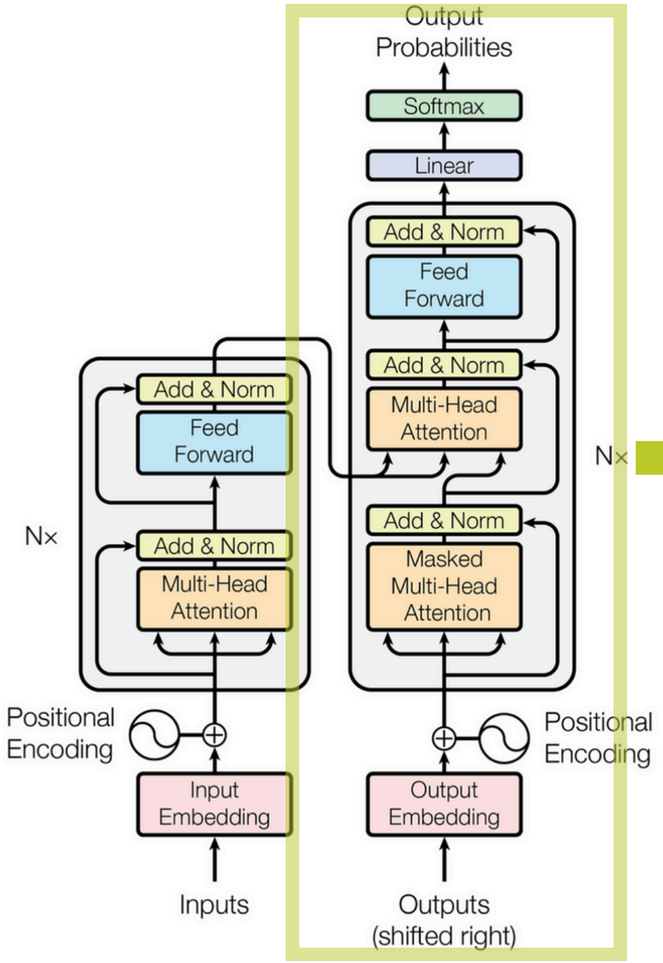
* Core Research Contributor; † Core Infrastructure Contributor; ‡ Correspondence to colah@anthropic.com; Author contributions statement below.

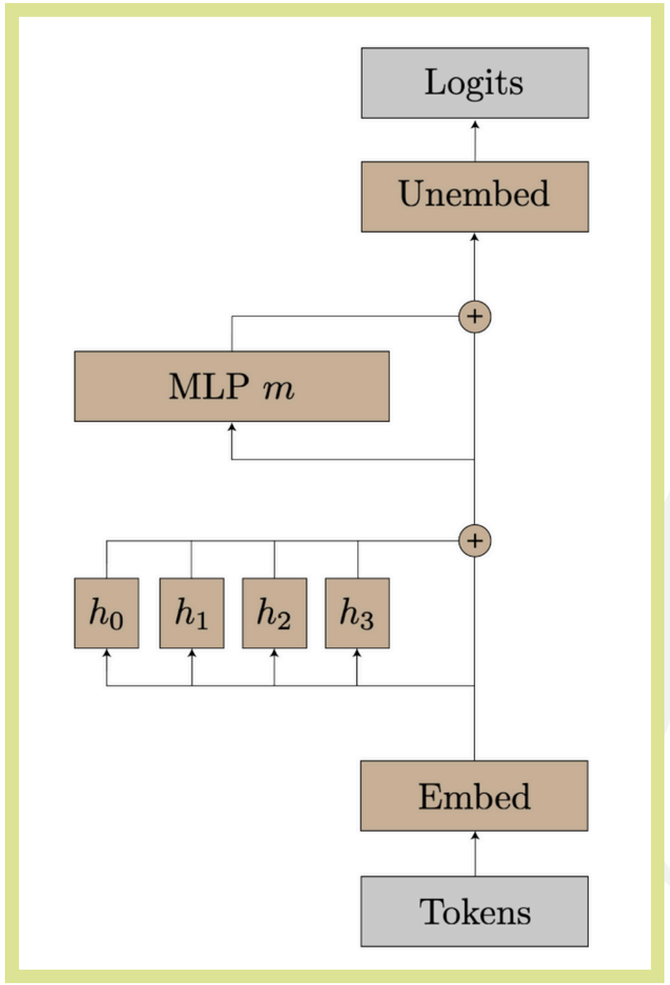
Language Model

"A **previous project** has attempted to reverse engineer vision models, but so far there hasn't been a comparable project for transformers or language models."

How does this paper approach?

High-Level Architecture of Transformers

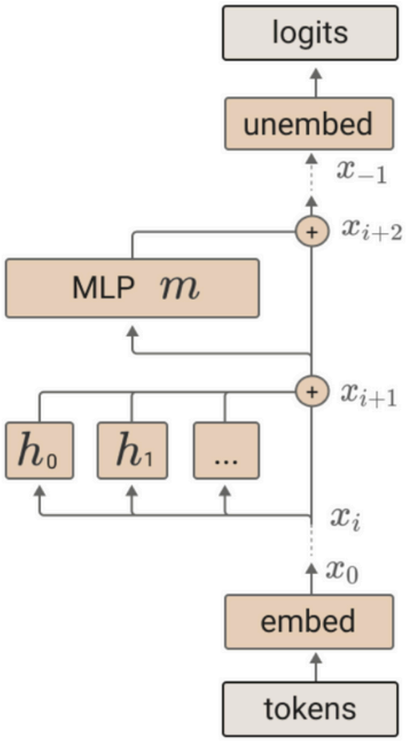




- **Masked Multi-Head Attention (self-attention)**
  - 중요한 메커니즘이므로 유지
- **Multi-Head Attention (cross-attention)**
  - 생략, 보통 encoder 구조는 LLM에 없으므로
- **Feed Forward (MLP)**
  - 유지 (ReLU같은 비선형 함수는 선형근사)
- **Add & Norm**
  - '+' 형태로 모델 수식에 포함됨
  - norm은 단순 scailing 효과로 보고 생략

How does this paper approach?

High-Level Architecture of Transformers



The final logits are produced by applying the unembedding.

$$T(t) = W_U x_{-1}$$

An MLP layer, m, is run and added to the residual stream.

$$x_{i+2} = x_{i+1} + m(x_{i+1})$$

Each attention head, h, is run and added to the residual stream.

$$x_{i+1} = x_i + \sum_{h \in H_i} h(x_i)$$

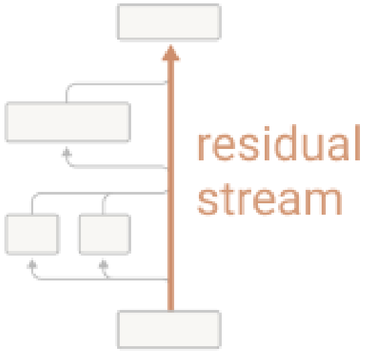
Token embedding.

$$x_0 = W_E t$$

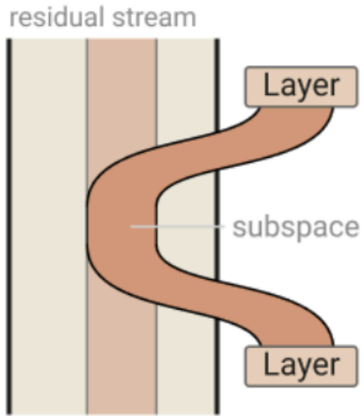
"An important first step is finding **the representation which makes it easiest to reason** about the model."

How does this paper approach?

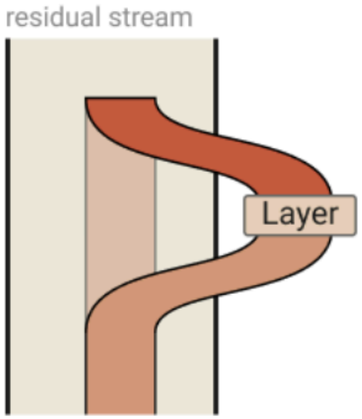
The **Residual Stream** as a Communication Channel



The residual stream is high dimensional, and can be divided into different subspaces.



Layers can interact by writing to and reading from the same or overlapping subspaces. If they write to and read from disjoint subspaces, they won't interact. Typically the spaces only partially overlap.

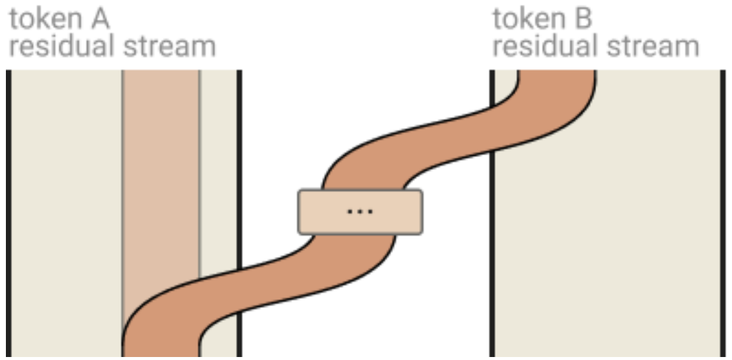


Layers can delete information from the residual stream by reading in a subspace and then writing the negative verison.

- **Residual stream**은 레이어들이 공유하는 통신 채널, 즉 **공용 메모리 공간**처럼 동작한다
- **Attention head**는 한 토큰의 residual stream에서 정보를 **읽고**, 결과를 다시 residual stream의 부분공간(subspace)에 **쓴다**.
- 같은/겹치는 공간에 쓰면 상호작용하고, 서로 다른 공간를 쓰면 거의 간섭 없이 독립적으로 동작할 수 있다.
  한 번 써진 정보는 다른 레이어가 덮어쓰거나 적극적으로 상쇄하지 않는 한 유지된다.

How does this paper approach?

Attention Heads as Information Movement



Attention heads copy information from the residual stream of one token to the residual stream of another. They typically write to a different subspace than they read from.

- The fundamental role of attention heads is **moving information.**
- They **read** information from the residual stream of one token, and **write** it to the residual stream of another token
- Choosing the source tokens is **independent** of what is read and how it is written.

How does this paper approach?

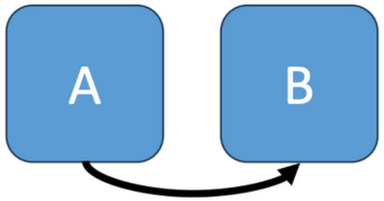
Model Simplification: Why Start with Toy Models?

"The aim is to discover **simple algorithmic patterns, motifs, or frameworks** that can subsequently be applied to larger and more complex models."

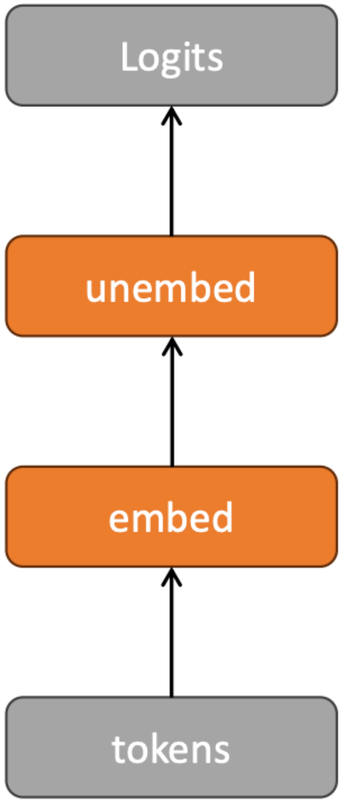
Summary of results

- Zero-Layer transformer: **Bigram** [A] → [B]
- One-Layer transformer: **Skip-trigram** [A] .. [B] → [A] copy
- Two-Layer transformer: "**Induction head**" (more complex algorithm) [A] [B] .. [A] → [B]

Zero-Layer Transformers



Bigram Model = $p(w_i|w_{i-1})$



Logit

$\text{Logits} = W_U x_0$

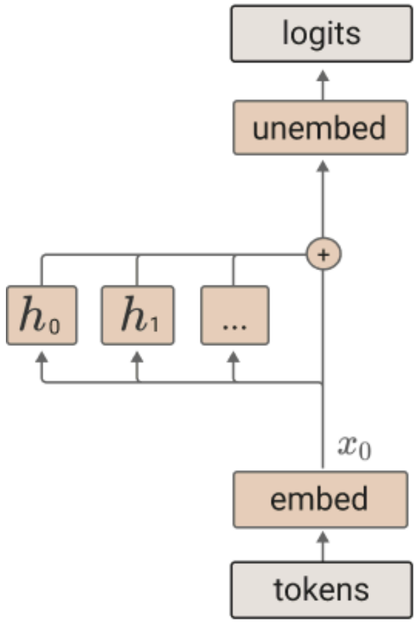
$x_0 = W_E t$

t

$\text{Logits} = W_U W_E t$

- 0-layer transformer는 현재 토큰을 그대로 embed→unembed 경로로 보내서 다음 토큰을 예측한다. (‘지금 단어 다음에 보통 뭐가 오지?’ 같은 bigram 방식으로 동작함)
- WuWe 항은 모든 트랜스포머에 남는 direct 경로이다.
- 큰 모델에서는 다른 복잡한 회로들이 규칙으로 설명하기 어려운 단순한 연속 패턴(예: Barack→Obama)을 보조적으로 메워주는 역할을 하기도 한다.

One-Layer Transformers (Attention-Only)



The final logits are produced by applying the unembedding.

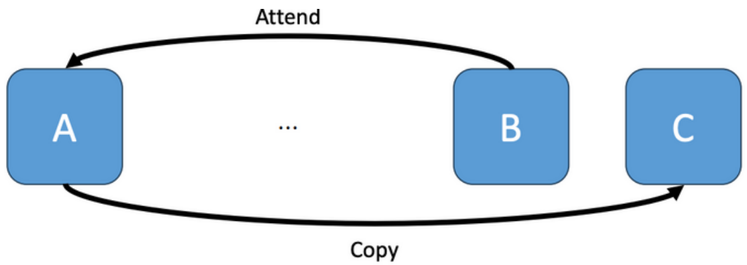
$$T(t) = W_U x_1$$

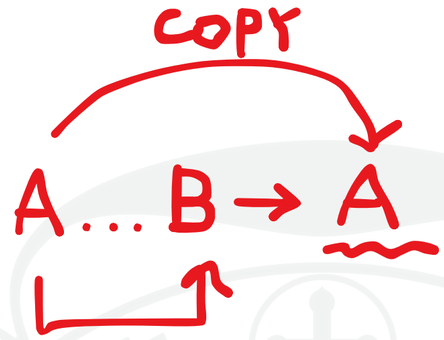
Each attention head, h, is run and added to the residual stream.

$$x_1 = x_0 + \sum_{h \in H} h(x_0)$$

Token embedding.

$$x_0 = W_E t$$





어텐션 헤드가 어디를 볼지를 고르고,
그 위치의 정보를 현재 토큰 위치로 Copy한다.

Skip-trigram처럼 작동한다는 것:
헤드가 어떤 레이어는 skip하면서 선택적으로
정보를 볼 수 있다

One-Layer Transformers (Attention-Only)

The path expansion **trick**

$$T = \underbrace{\mathrm{Id} \otimes W_U}_{\text{The token unembedding maps residual stream vectors to logits.}} \cdot \underbrace{\left(Id + \sum_{h \in H_1} A^h \otimes W_{OV}^h\right)}_{\text{The attention layer has multiple heads. The result of each is added into the residual stream.}} \cdot \underbrace{\mathrm{Id} \otimes W_E}_{\text{The token embedding maps tokens to residual stream vectors.}}$$

where $A^h = \underbrace{\mathrm{softmax}^*}_{\text{Softmax with autoregressive masking}}\left(t^T \cdot W_E^T W_{QK}^h W_E \cdot t\right)$

Attention pattern logits are produced by multiplying pairs of tokens through different sides of W_{QK}^h.

$$T = \mathrm{Id} \otimes W_U W_E + \sum_{h \in H} A^h \otimes (W_U W_{OV}^h W_E)$$

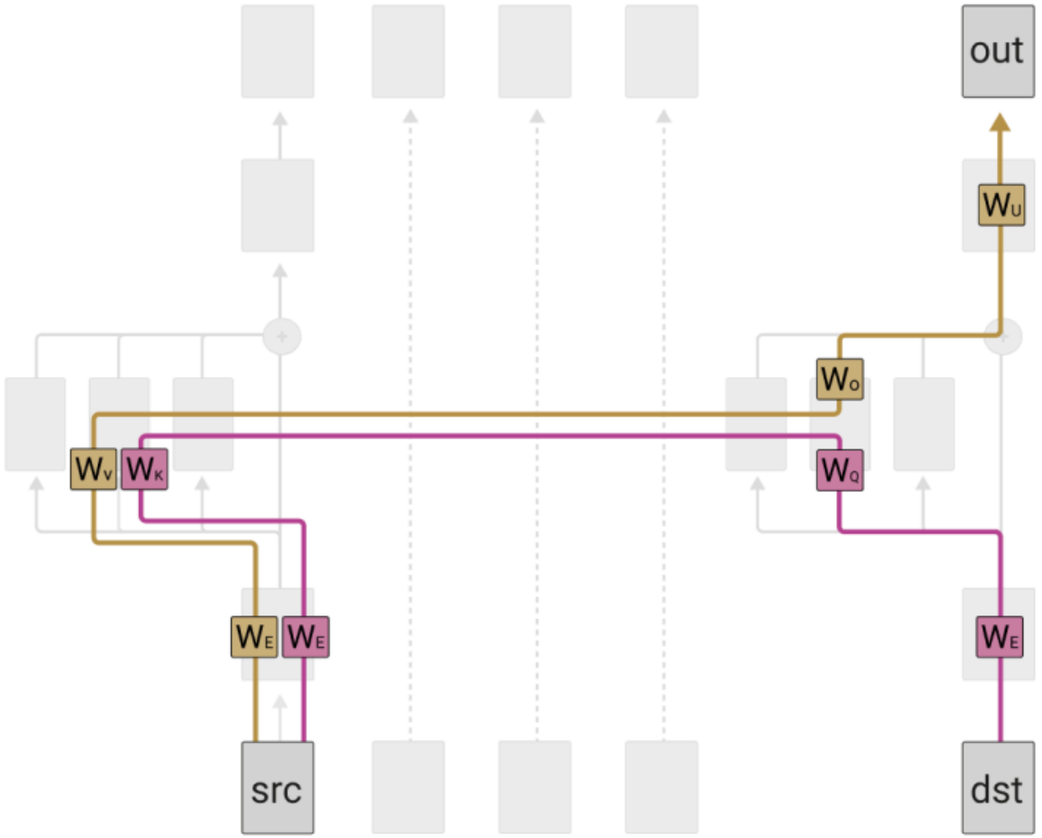
"Direct path" term contributes to bigram statistics.

The **attention head** terms describe the effects of attention heads in linking input tokens to logits. A^h describes which tokens are attended to while $W_U W_{OV}^h W_E$ describes how each token changes the logits if attended to.

결국 1-layer transformer 이해는
전개된 OV/QK 행렬식을 이해하는 문제

One-Layer Transformers (Attention-Only)

Splitting Attention Head terms into **QK** and **OV** Circuits ⭐



The **OV ("output-value") circuit** determines how attending to a given token affects the logits.

$$W_U W_O W_V W_E$$

The **QK ("query-key") circuit** controls which tokens the head prefers to attend to.

$$W_E^T W_Q^T W_K W_E$$

주요 발견

어텐션 헤드는 두개의 분리 가능한 연산으로 구성되어있다 (QK / OV)

Copy role

Match role

One-Layer Transformers (Attention-Only)

Interpretation as Skip-Trigrams

[source]... [destination] [out]

Some examples of large entries QK/OV circuit

Source Token	Destination Token	Out Token	Example Skip Tri-grams
" perfect"	" are", " looks", " is", " provides"	" perfect", " super", " absolute", " pure"	" perfect... are perfect", " perfect... looks super"
" large"	" contains", " using", " specify", " contain"	" large", " small", " very", " huge"	" large... using large", " large... contains small"
" two"	" One", "\n ", " has", "\r\n ", "One"	" two", " three", " four", " five", " one"	" two... One two", " two... has three"
"lambda"	" $\\", "}{\\", "+\\", "(\\", " ${\\"	"lambda", "sorted", " lambda", "operator"	"lambda... $\\lambda", "lambda... +\\lambda"
"nbsp"	"&", " \"&", "}&", ">&", "=&"	"nbsp", "01", "gt", "00012", "nbs", "quot"	"nbsp... ", "nbsp... > "
"Great"	"The", " The", " the", " contains", " /"	" Great", " great", " poor", " Every"	"Great... The Great", "Great... the great"

Limited Expressivity Can Create Bugs which Seem Strange from the Outside

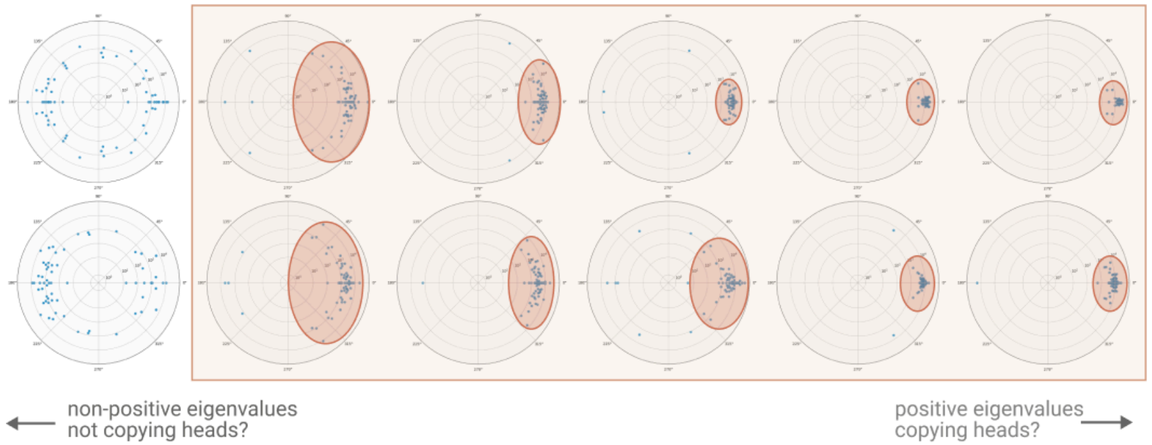
Source Token	Destination Token	Out Token	"Correct" Skip Tri-grams	"Bug" Skip Tri-grams
" Pixmap"	" P", " Q", "P", " p", " U"	"ixmap", "Canvas", "Embed", "grade"	" Pixmap... Pixmap", " Pixmap... QCanvas"	" Pixmap... PCanvas"
Source Token	**Destination Token**	**Out Token**	**"Correct" Skip Tri-grams**	**"Bug" Skip Tri-grams**
" Lloyd"	"L", " L", " P", "P", "R", " C"	"loyd", "alph", "\n ", "acman", ... "atherine"	" Lloyd... Lloyd", " Lloyd... Catherine"	" Lloyd... Cloyd", " Lloyd... Latherine"
Source Token	**Destination Token**	**Out Token**	**"Correct" Skip Tri-grams**	**"Bug" Skip Tri-grams**
" keep"	" in", " at", " out", " under", " off"	" bay", ... " mind", ... " wraps"	" keep... in mind", " keep... at bay", " keep... under wraps"	" keep... in bay", " keep... at wraps", " keep... under mind"

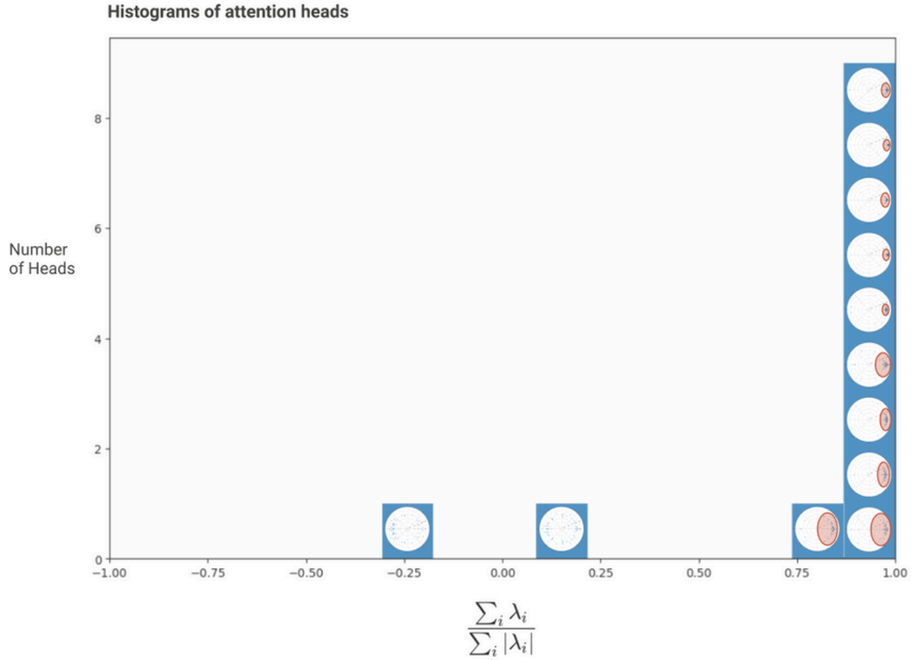
Primitive In-Context Learning Patterns

[b]...[a]→[b]	[b]...[a]→[b']	[ab]...[a]→[b]	[ab]...[a]→[b']
[two]...[One]→[two]	[two]...[has]→[three]	[Ralph]...[R]→[alph]	[Ralph]...[R]→[ALPH]
[perfect]...[are]→[perfect]	[perfect]...[looks]→[super]	[Pike]...[P]→[ike]	[Pike]...[P]→[ikes]
[nbsp]...[&]→[nbsp]	[nbsp]...[&]→[gt]	[Pixmap]...[P]→[ixmap]	
[lambda]...[$\\]→[lambda]	[lambda]...[$\\]→[operator]	[Lloyd]...[L]→[loyd]	

One-Layer Transformers (Attention-Only)

Summarizing OV/QK Matrices





Histograms of attention heads by the fraction of their eigenvalues that are positive can be a useful summary.

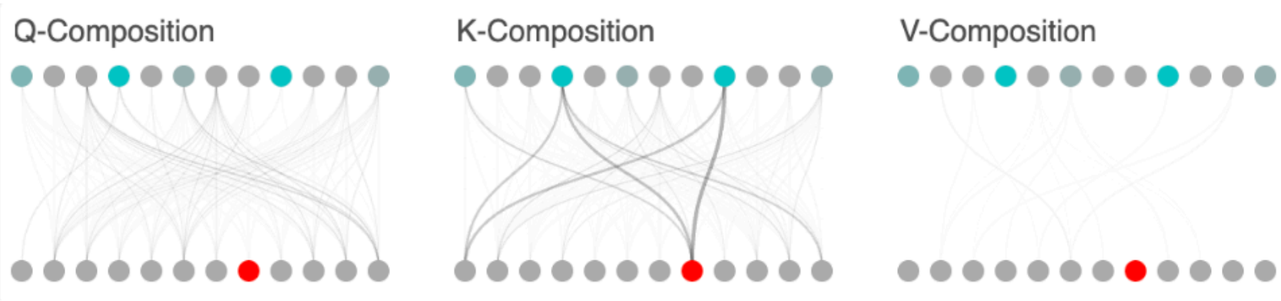
=> 실제로 1-layer에서 Head들이 ‘Copy’ 방식으로 작동하고 있다는 강한 증거

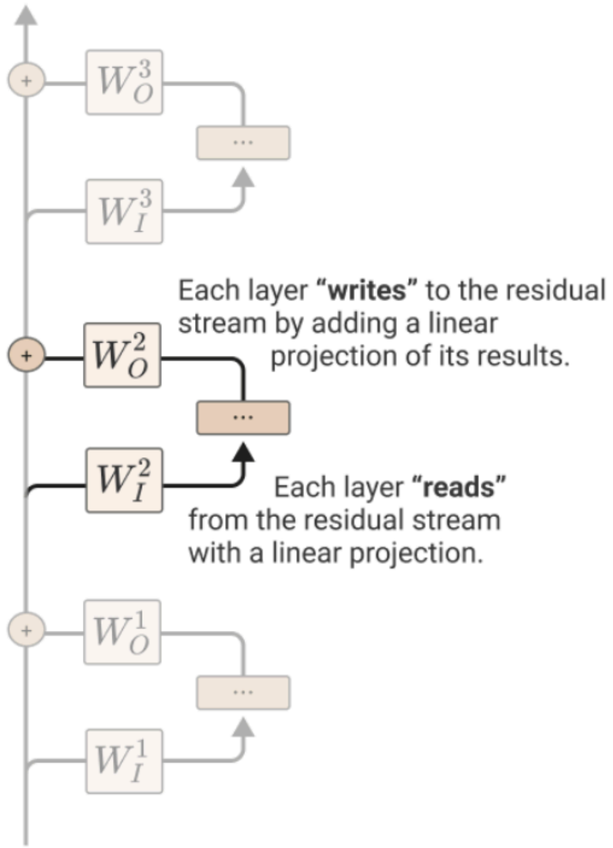
Two-Layer Transformers (Attention-Only)

Composition ⭐

When attention heads do compose, there are three options:

- Q-Composition: W_Q reads in a subspace affected by a previous head.
- K-Composition: W_K reads in a subspace affected by a previous head.
- V-Composition: W_V reads in a subspace affected by a previous head.





Attention head들은 residual stream이라는 공용 공간을 통해 서로 영향을 줄 수 있고,
이때 그 영향이 어디에서 발생하느냐에 따라 세 가지 composition으로 나뉜다.

Two-Layer Transformers (Attention-Only)

Path Expansion of Logits

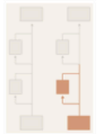
$$T = \underbrace{\mathrm{Id} \otimes W_U}_{\text{}} \cdot \underbrace{\left(Id + \sum_{h \in H_2} A^h \otimes W_{OV}^h\right)}_{\text{The second attention layer has multiple attention heads which add into the residual stream}} \cdot \underbrace{\left(Id + \sum_{h \in H_1} A^h \otimes W_{OV}^h\right)}_{\text{The first attention layer has multiple attention heads which add into the residual stream}} \cdot \underbrace{\mathrm{Id} \otimes W_E}_{\text{}}$$

$$= \underbrace{\mathrm{Id} \otimes W_U W_E}_{\text{“Direct path” term contributes to bigram statistics.}} + \underbrace{\sum_{h \in H_1 \cup H_2} A^h \otimes (W_U W_{OV}^h W_E)}_{\text{The individual attention head terms describe the effects of individual attention heads in linking input tokens to logits, similar to those we saw in the one layer model.}} + \underbrace{\sum_{h_2 \in H_2} \sum_{h_1 \in H_1} (A^{h_2} A^{h_1}) \otimes (W_U W_{OV}^{h_2} W_{OV}^{h_1} W_E)}_{\text{The virtual attention head terms correspond to V-composition of attention heads. They function a lot like individual attention heads, with their own attention patterns (the compositon of the heads patterns) and own OV matrix.}}$$

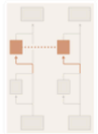
Two-Layer Transformers (Attention-Only)

Path Expansion of Attention Scores QK Circuit

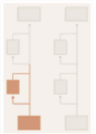
$$C_{QK}^{h\in H_2} = \left(\mathrm{Id}\otimes\mathrm{Id}\otimes W_E^T + \sum_{h_q\in H_1} A^{h_q}\otimes\mathrm{Id}\otimes(W_{OV}^{h_q}W_E)^T\right) \cdot \mathrm{Id}\otimes\mathrm{Id}\otimes W_{QK}^h \cdot \left(\mathrm{Id}\otimes\mathrm{Id}\otimes W_E + \sum_{h_k\in H_1}\mathrm{Id}\otimes A^{h_k}\otimes W_{OV}^{h_k}W_E\right)$$



The **"query side" residual stream** at the start of the second layer contains both the layer 1 direct path and layer 1 attention heads. All terms are of the form $\ldots\otimes\mathrm{Id}\otimes\ldots$ because they don't move key information.

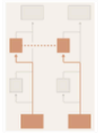


W_{QK} of the second layer head combines both sides into attention scores.

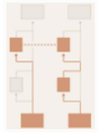


The **"key side" residual stream** at the start of the second layer contains both the layer 1 direct path and attention heads. All terms are of the form $\mathrm{Id}\otimes\ldots$ because they don't move query information.

$$= \mathrm{Id}\otimes\mathrm{Id}\otimes(W_E^T W_{QK}^h W_E) + \sum_{h_q\in H_1} A^{h_q}\otimes\mathrm{Id}\otimes\left(W_E^T {W_{OV}^{h_q}}^T W_{QK}^h W_E\right)$$

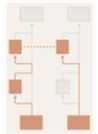


The no composition term. Both first layer follows the direct path on both the key and query side.

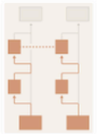


These terms correspond to pure **Q-composition**. A previous attention head is used to generate the query side, but the key side is the first layer direct path.

$$+ \sum_{h_k\in H_1}\mathrm{Id}\otimes A^{h_k}\otimes\left(W_E^T W_{QK}^h W_{OV}^{h_k} W_E\right) + \sum_{h_q\in H_1}\sum_{h_k\in H_1} A^{h_q}\otimes A^{h_k}\otimes\left(W_E^T {W_{OV}^{h_q}}^T W_{QK}^h W_{OV}^{h_k} W_E\right)$$



These terms correspond to pure **K-composition**. A previous attention head is used to generate part of the key, but the query side is the first layer direct path.



These terms are interactions between both **Q-composition** and **K-composition**. A previous attention head is used to generate the query and key sides.

Attn score - QK의 해부

2-layer에선 1-layer의 여러 head 출력이 residual에 섞인 결과로 QK가 만들어짐

=> 어떤 path가 있는지 전개해보자

1) layer 1의 head들이 query쪽에 영향을 주는 경우
2) key 쪽에 영향을 주는 경우
3) 둘 다에 영향을 주는 경우의 합

Induction head는 그중 key 쪽만 영향주는 항을 (K-Composition)을 핵심적으로 사용함

Two-Layer Transformers (Attention-Only)

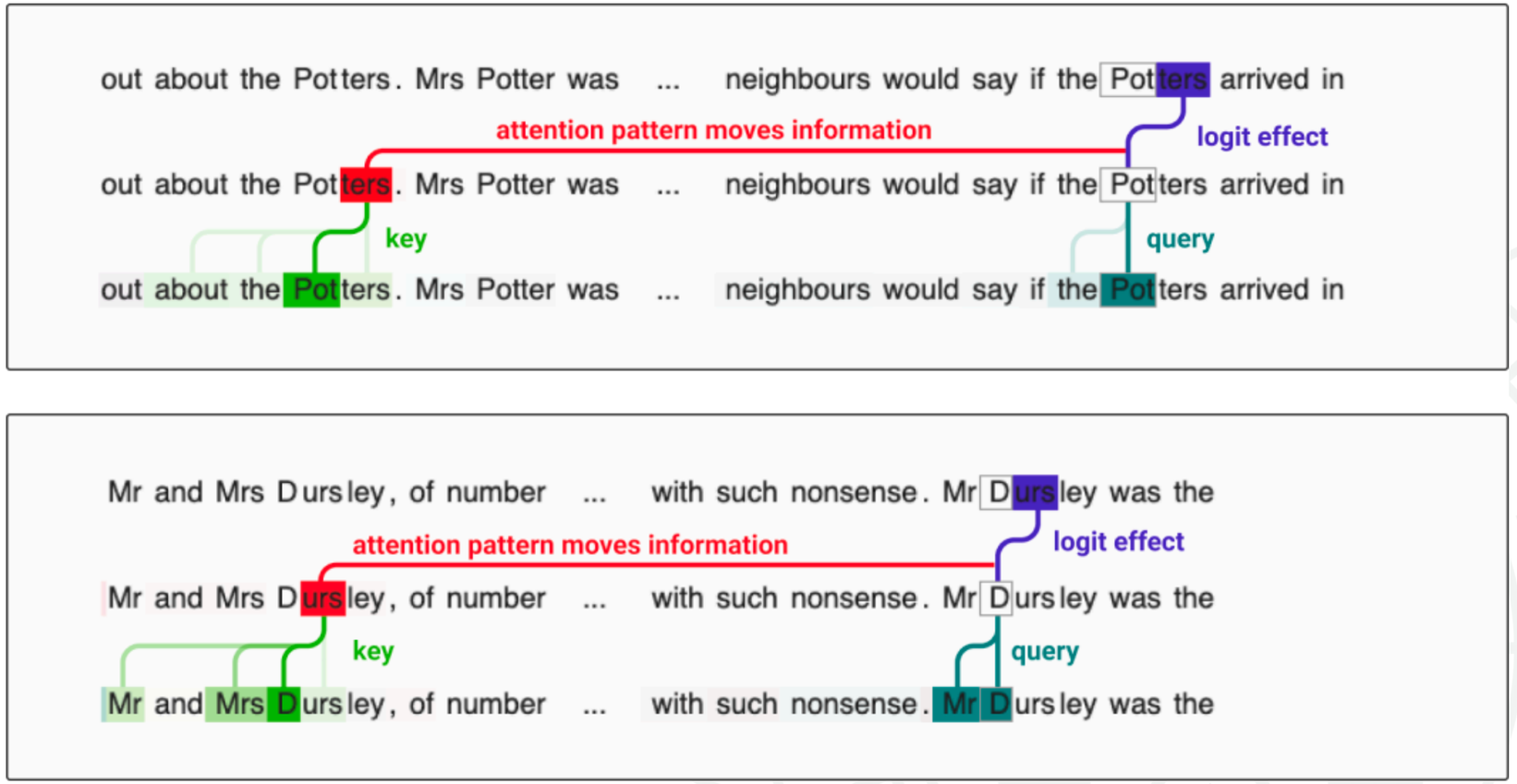
Induction head ⭐

[a] [b] … [a] → [b]

현재 토큰 = Pot
예측하려는 토큰 = ters

Pot을 쿼리로 검색하면
과거에 pot이 나왔던 다음 위치
(=ters 위치)로 attention

1-layer head:
at position 'ters'
writes: "my previous token was Pot"



2-layer 모델에서 Induction head가 다음 토큰을 가져오는 방법

Two-Layer Transformers (Attention-Only)

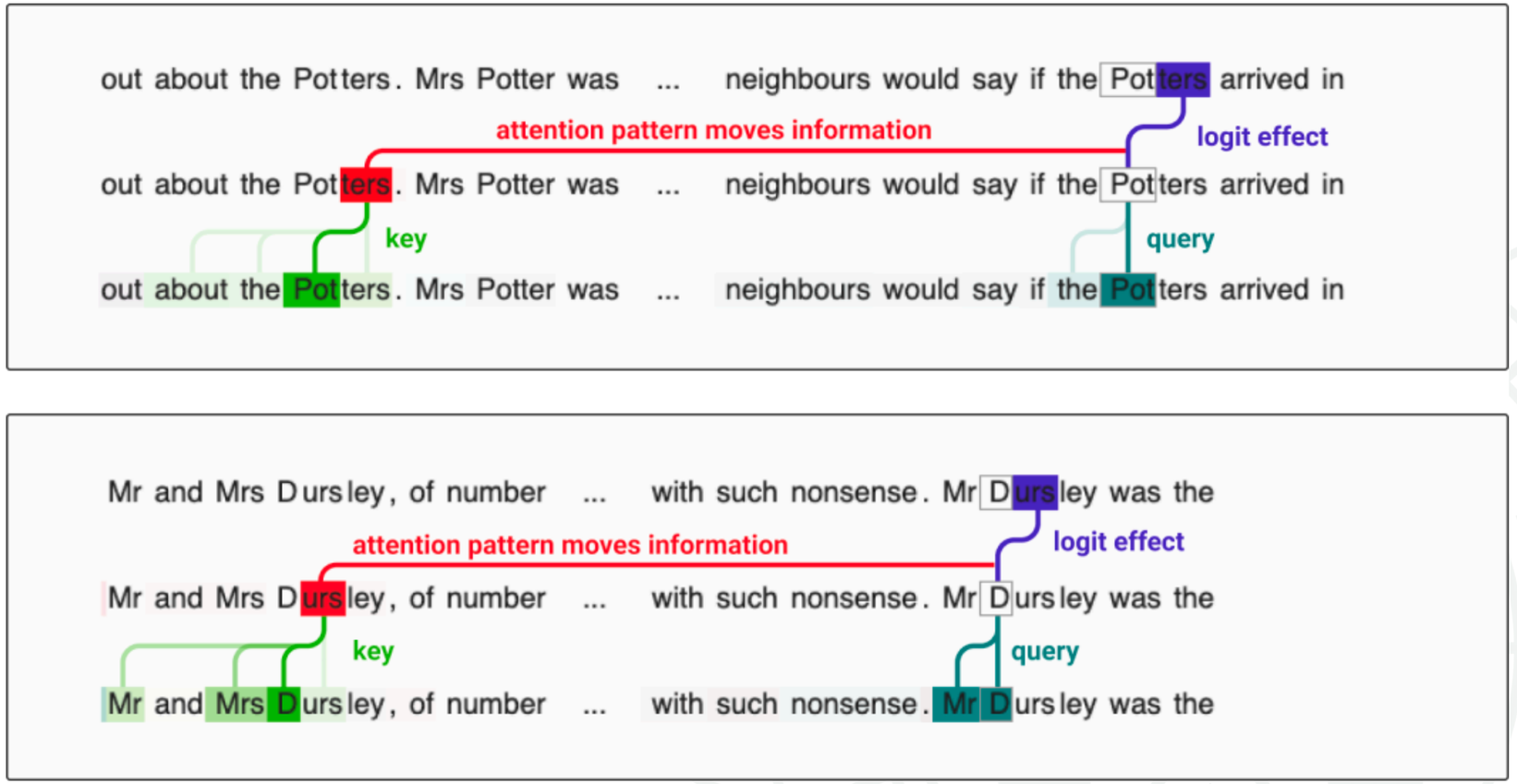
Induction head ⭐

만약 1-layer만 있었다면?

가능한 행동 (1-layer)

- pot이 보이면 → pot을 복사한다
- “pot이 보이면 → Potter, Potts, Potato 같은 연관 토큰을 밀어준다
- pot이 보이면 → 대문자/소문자 버전, 유사 철자 토큰을 밀어준다

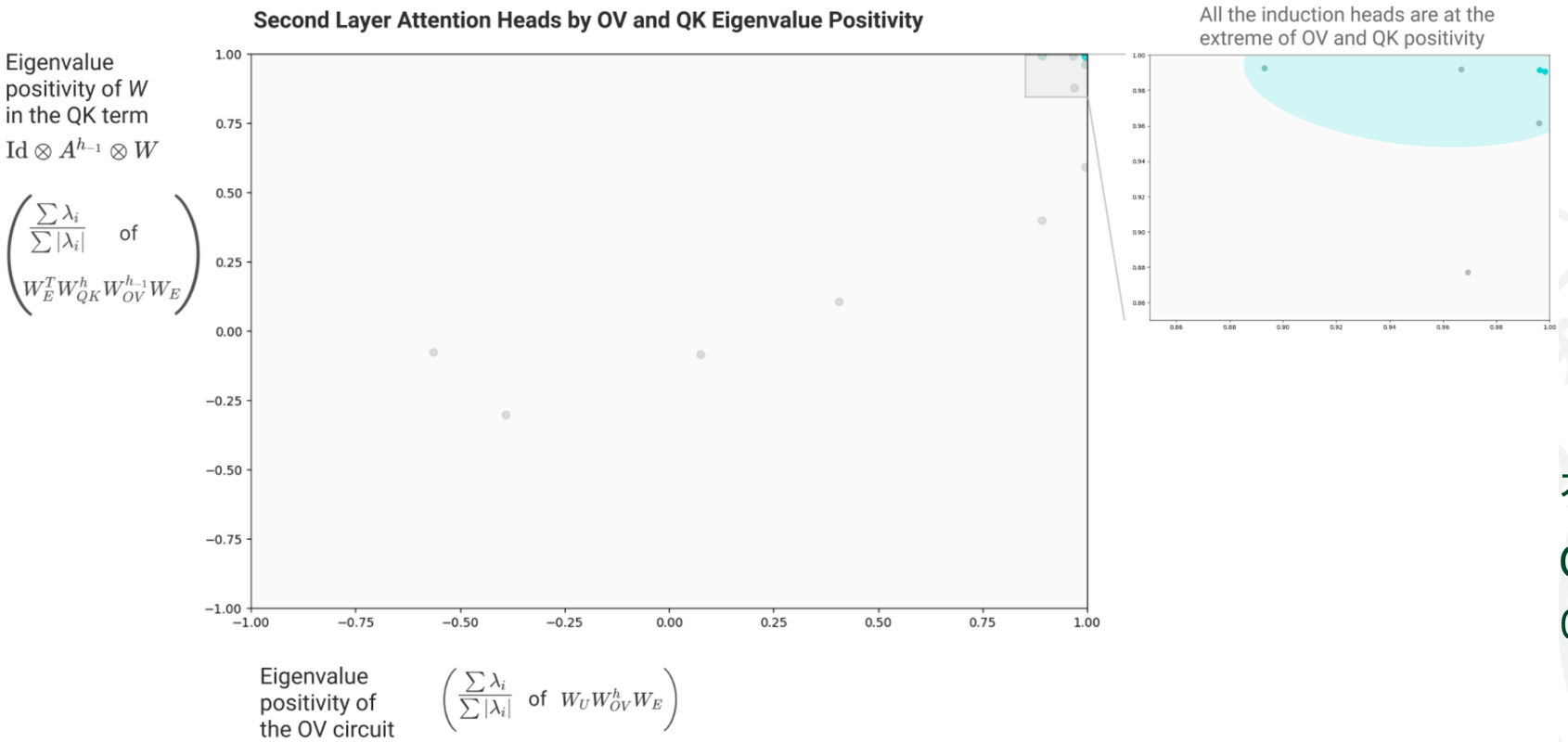
반면 2-layer는 ‘이 토큰이 예전에 어떻게 쓰였지'를 할 수 있음



2-layer 모델에서 Induction head가 다음 토큰을 가져오는 방법

Two-Layer Transformers (Attention-Only)

Checking the Mechanistic Theory



정말로 OV는 Copy
QK는 Match 역할을 하는가?
에 대한 증거

Where Does This Leave Us?

일반화 가능성

- 토이 모델에서 분석한 어텐션 회로(특히 K-composition 기반 induction head)는 대형 모델에서도 관찰되며, in-context learning의 핵심 메커니즘으로 보임

연구의 한계

- 어텐션 중심 회로는 잘 설명하지만, 모델 전체를 이해하기에는 한계가 있음
- MLP는 전체 파라미터의 약 2/3를 차지하며, 어텐션과 강하게 얽혀 있어 현재 분석 범위를 크게 제한함. (토이 모델에서는 MLP 부분을 배제함)

다음 과제

- 완전한 이해를 위해서는 Superposition으로 얽힌 MLP 내부에서 해석 가능한 피처를 분리하는 방법이 필요함.
- 본 연구는 출발점이고, 향후 핵심 과제는 MLP와 피처 해석 (Superposition 문제)임

Discussion

- **왜 논문은 ICL(In-Context Learning)을 계속 언급하는가?**
  - 시대적 맥락: GPT3
  - ICL: '모델이 학습을 안해도 문맥에서 규칙을 배운다'
  - 적어도 작은 회로(induction head)에서 ICL과 닮은 현상이 보여진다 → 처음으로 ICL를 구조적으로 설명 시도

- **왜 'feature'라는 말을 하지 못했을까**
  - 분명 기능적으로 분리된 계산 단위를 발견했음
  - 그런데 feature가 아닌 path, circuits, OV/QK 행렬 등으로 우회
  - "Residual stream'에는 'previleged basis'가 없다"
  - 실제 2025 revisit에서도 '우리가 행렬곱/경로 등으로 우회했었다 → 지금은 feature를 말할 수 있다'라고 언급

이화여자대학교
EWHA WOMANS UNIVERSITY

2026 - Winter

Thank you

HAI Lab

2271064 한사랑