

Some Memory Mechanisms in Transformers and their Emergence

Alberto Bietti

Flatiron Institute, Simons Foundation

Memory and Vision workshop.
ICCV 2025.



Some Memory Mechanisms in Transformers and their Emergence

Alberto Bietti

Flatiron Institute, Simons Foundation

Memory and Vision workshop.
ICCV 2025.

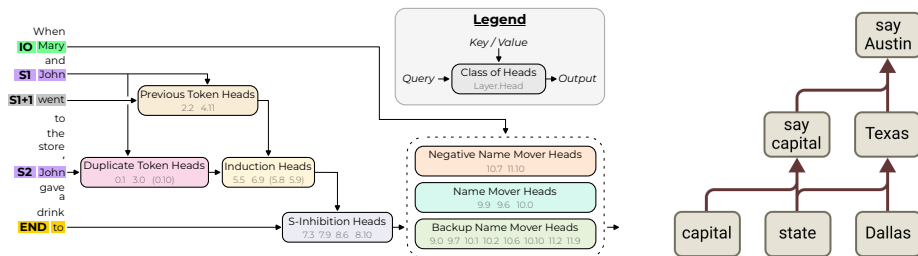
w/ V. Cabannes, E. Dohmatob, D. Bouchacourt, H. Jégou, L. Bottou (Meta AI),
E. Nichani, Z. Wang, J. Lee (Princeton), L. Chen, D. Wu, J. Bruna (NYU), D. Hsu (Columbia)



Mechanisms inside Transformer LLMs

Reasoning over context

- Circuits of attention heads (Elhage et al., 2021; Olsson et al., 2022; Wang et al., 2022)
- A “biology” of circuits found by mechanistic interpretability (e.g., Anthropic, 2025)



Mechanisms inside Transformer LLMs

Reasoning over context

- Circuits of attention heads (Elhage et al., 2021; Olsson et al., 2022; Wang et al., 2022)
- A “biology” of circuits found by mechanistic interpretability (e.g., Anthropic, 2025)

Knowledge storage

- Memorization, factual recall, parameter scaling
 - ▶ (Geva et al., 2020; Meng et al., 2022; Allen-Zhu and Li, 2024)
- Learn rules that help higher-level reasoning



Dan Hendrycks ✓ @DanHendrycks · Mar 14, 2023

It knows many esoteric facts (e.g., the meaning of obscure songs, knows what area a researcher works in, can contrast ML optimizers like Adam vs AdamW like in a PhD oral exam, and so on).

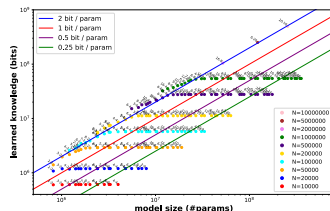
My rule-of-thumb is that
"if it's on the internet 5 or more times, GPT-4 remembers it."

1

28

184

25K



Mechanisms inside Transformer LLMs

Reasoning over context

- Circuits of attention heads (Elhage et al., 2021; Olsson et al., 2022; Wang et al., 2022)
- A “biology” of circuits found by mechanistic interpretability (e.g., Anthropic, 2025)

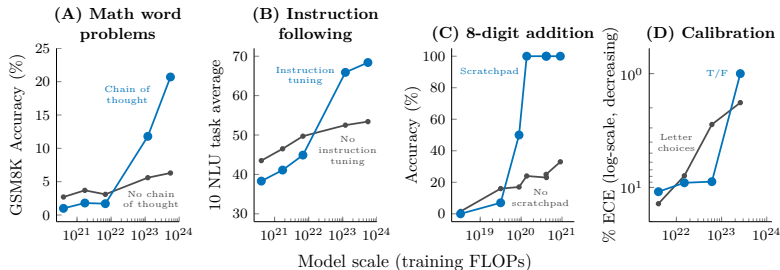
Knowledge storage

- Memorization, factual recall, parameter scaling
 - ▶ (Geva et al., 2020; Meng et al., 2022; Allen-Zhu and Li, 2024)
- Learn rules that help higher-level reasoning

Q: How do these arise during training?

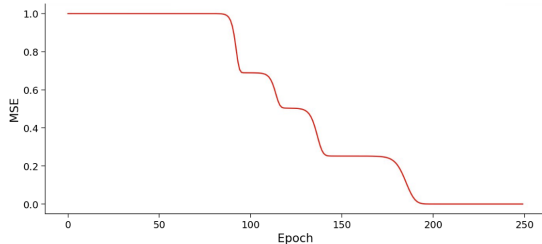
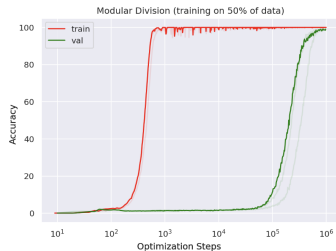
Emergence and training dynamics

Emergent capabilities and sequential learning (Wei et al. 2022; Power et al., 2022; Saxe et al. 2013)



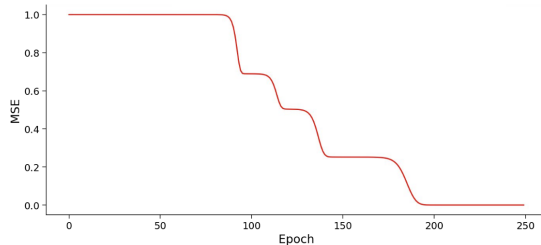
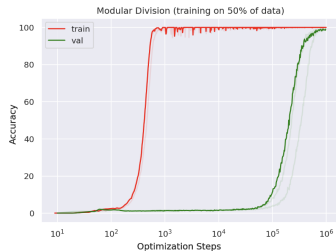
Emergence and training dynamics

Emergent capabilities and sequential learning (Wei et al. 2022; Power et al., 2022; Saxe et al. 2013)



Emergence and training dynamics

Emergent capabilities and sequential learning (Wei et al. 2022; Power et al., 2022; Saxe et al. 2013)



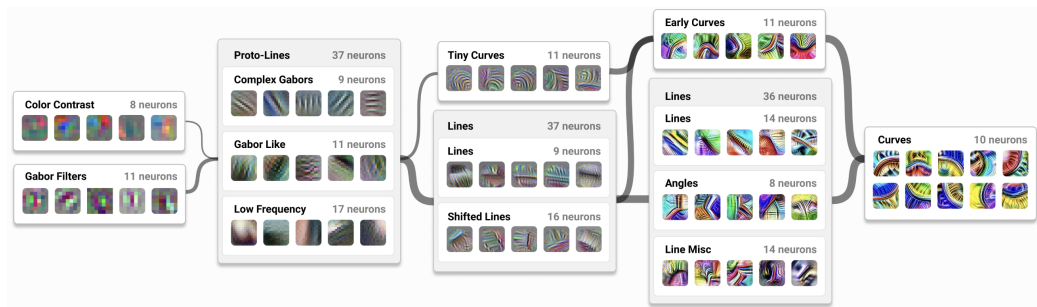
Why do we care about understanding?

- Highlight role of **data**, training **algorithm**, **architecture**
- $\Rightarrow$ improve training methodology

What about computer vision?

A mix of continuous and discrete (?)

- Low-level processing usually involves **continuous operations** (e.g., Gabor filters)
- Some evidence for learned **circuits**
- Models also involve **discrete semantic concepts** and **knowledge** (especially VLMs)



(Cammarata et al., 2021)

What about computer vision?

A mix of continuous and discrete (?)

- Low-level processing usually involves **continuous operations** (e.g., Gabor filters)
- Some evidence for learned **circuits**
- Models also involve **discrete semantic concepts** and **knowledge** (especially VLMs)

Feature #34M/31164353 Golden Gate Bridge feature example

The feature activates strongly on English descriptions and associated concepts

in the Presidio at the end (that's the huge park right next to the Golden Gate bridge), perfect. But not all people

repainted, roughly every dozen years." "while across the country in San Francisco, the golden gate bridge was

it is a suspension bridge and has similar coloring, it is often compared to the Golden Gate Bridge in San Francisco, US

They also activate in multiple other languages on the same concepts

ゴールデン・ゲート・ブリッジ、金門橋は、アメリカ西海岸のサンフランシスコ湾と太平洋が接続するゴールデンゲート湾

골든게이트교 또는 금문교는 미국 캘리포니아주 골든게이트 해협에 위치한 현수교이다. 골든게이트교는 캘리포니아주 샌프란시스코

мост Золотые Ворота — висячий мост через пролив Золотые Ворота. Он соединяет город Сан-Фран

And on relevant images as well

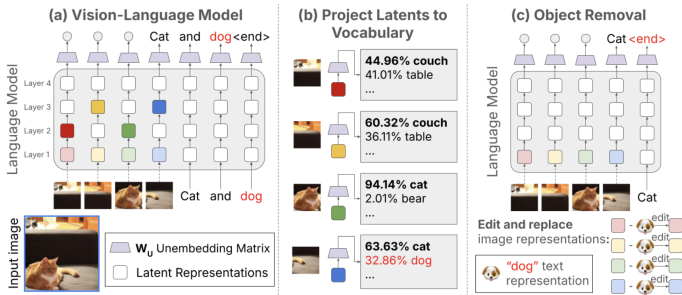


(Templeton et al., 2024)

What about computer vision?

A mix of continuous and discrete (?)

- Low-level processing usually involves **continuous operations** (e.g., Gabor filters)
- Some evidence for learned **circuits**
- Models also involve **discrete semantic concepts** and **knowledge** (especially VLMs)



(Jiang et al., 2025)

Transformer setup

Input: sequence of discrete tokens $(z_1, \dots, z_T) \in [N]^T$

Transformer setup

Input: sequence of discrete tokens $(z_1, \dots, z_T) \in [N]^T$

Embeddings

- input e_z , positional p_t , output u_y , in $\mathbb{R}^d$

Transformer setup

Input: sequence of discrete tokens $(z_1, \dots, z_T) \in [N]^T$

Embeddings

- input e_z , positional p_t , output u_y , in $\mathbb{R}^d$

Residual streams (Elhage et al., 2021)

- embed each token $z_t \in [N]$ as $x_t := e_{z_t} + p_t$
- (causal) self-attention $x_t := x_t + \text{MHSA}(x_t, x_{1:t})$



$$\text{MHSA}(x_t, x_{1:t}) = \sum_{h=1}^H \sum_{s=1}^t \beta_s^h W_O^{h\top} W_V^h x_s, \quad \text{with } \beta_s^h = \frac{\exp(x_s^\top W_K^{h\top} W_Q^h x_t)}{\sum_{s=1}^t \exp(x_s^\top W_K^{h\top} W_Q^h x_t)}$$

where $W_K, W_Q, W_V, W_O \in \mathbb{R}^{d_h \times d}$ (key/query/value/output matrices)

Transformer setup

Input: sequence of discrete tokens $(z_1, \dots, z_T) \in [N]^T$

Embeddings

- input e_z , positional p_t , output u_y , in $\mathbb{R}^d$

Residual streams (Elhage et al., 2021)

- embed each token $z_t \in [N]$ as $x_t := e_{z_t} + p_t$
- (causal) self-attention $x_t := x_t + \text{MHSA}(x_t, x_{1:t})$
- feed-forward $x_t := x_t + \text{MLP}(x_t)$



$$\text{MLP}(x_t) = V^T \sigma(Ux_t)$$

where $U, V \in \mathbb{R}^{m \times d}$, often $m = 4d$

Transformer setup

Input: sequence of discrete tokens $(z_1, \dots, z_T) \in [N]^T$

Embeddings

- input e_z , positional p_t , output u_y , in $\mathbb{R}^d$

Residual streams (Elhage et al., 2021)

- embed each token $z_t \in [N]$ as $x_t := e_{z_t} + p_t$
- (causal) self-attention $x_t := x_t + \text{MHSA}(x_t, x_{1:t})$
- feed-forward $x_t := x_t + \text{MLP}(x_t)$
- residual stream x_t is a sum of embeddings/“features”



Transformer setup

Input: sequence of discrete tokens $(z_1, \dots, z_T) \in [N]^T$

Embeddings

- input e_z , positional p_t , output u_y , in $\mathbb{R}^d$

Residual streams (Elhage et al., 2021)

- embed each token $z_t \in [N]$ as $x_t := e_{z_t} + p_t$
- (causal) self-attention $x_t := x_t + \text{MHSA}(x_t, x_{1:t})$
- feed-forward $x_t := x_t + \text{MLP}(x_t)$
- residual stream x_t is a sum of embeddings/“features”



Next-token prediction

- cross-entropy loss

$$\sum_{t < T} \ell(z_{t+1}; (u_j^\top x_t)_j)$$

Outline

1 Weights as memories

2 Context as memory

Weights as associative memories

- Consider sets of **nearly orthonormal embeddings** $\{e_z\}_{z \in \mathcal{Z}}$ and $\{u_y\}_{y \in \mathcal{Y}}$:

$$\|e_z\| \approx 1 \quad \text{and} \quad e_z^\top e_{z'} \approx 0$$

$$\|u_y\| \approx 1 \quad \text{and} \quad u_y^\top u_{y'} \approx 0$$

Weights as associative memories

- Consider sets of **nearly orthonormal embeddings** $\{e_z\}_{z \in \mathcal{Z}}$ and $\{u_y\}_{y \in \mathcal{Y}}$:

$$\|e_z\| \approx 1 \quad \text{and} \quad e_z^\top e_{z'} \approx 0$$

$$\|u_y\| \approx 1 \quad \text{and} \quad u_y^\top u_{y'} \approx 0$$

- For **pairwise associations** $z \mapsto f(z)$ with $z \in \mathcal{M}$, define:

$$W = \sum_{z \in \mathcal{M}} u_{f(z)} e_z^\top$$

Weights as associative memories

- Consider sets of **nearly orthonormal embeddings** $\{e_z\}_{z \in \mathcal{Z}}$ and $\{u_y\}_{y \in \mathcal{Y}}$:

$$\|e_z\| \approx 1 \quad \text{and} \quad e_z^\top e_{z'} \approx 0$$

$$\|u_y\| \approx 1 \quad \text{and} \quad u_y^\top u_{y'} \approx 0$$

- For **pairwise associations** $z \mapsto f(z)$ with $z \in \mathcal{M}$, define:

$$W = \sum_{z \in \mathcal{M}} u_{f(z)} e_z^\top \implies u_y^\top W e_z \approx \mathbb{1}\{y = f(z)\}$$

Weights as associative memories

- Consider sets of **nearly orthonormal embeddings** $\{e_z\}_{z \in \mathcal{Z}}$ and $\{u_y\}_{y \in \mathcal{Y}}$:

$$\|e_z\| \approx 1 \quad \text{and} \quad e_z^\top e_{z'} \approx 0$$

$$\|u_y\| \approx 1 \quad \text{and} \quad u_y^\top u_{y'} \approx 0$$

- For **pairwise associations** $z \mapsto f(z)$ with $z \in \mathcal{M}$, define:

$$W = \sum_{z \in \mathcal{M}} u_{f(z)} e_z^\top \implies u_y^\top W e_z \approx \mathbb{1}\{y = f(z)\}$$

- Examples in Transformers:

- ▶ Logits in attention heads: $x_k^\top W_{KQ} x_q$
- ▶ Logits in next-token prediction: $u_y^\top U \sigma(V x_t)$ or $u_y^\top W_{OV} x_k$

Weights as associative memories

- Consider sets of **nearly orthonormal embeddings** $\{e_z\}_{z \in \mathcal{Z}}$ and $\{u_y\}_{y \in \mathcal{Y}}$:

$$\|e_z\| \approx 1 \quad \text{and} \quad e_z^\top e_{z'} \approx 0$$

$$\|u_y\| \approx 1 \quad \text{and} \quad u_y^\top u_{y'} \approx 0$$

- For **pairwise associations** $z \mapsto f(z)$ with $z \in \mathcal{M}$, define:

$$W = \sum_{z \in \mathcal{M}} u_{f(z)} e_z^\top \implies u_y^\top W e_z \approx \mathbb{1}\{y = f(z)\}$$

- Examples in Transformers:
 - Logits in attention heads: $x_k^\top W_{KQ} x_q$
 - Logits in next-token prediction: $u_y^\top U \sigma(V x_t)$ or $u_y^\top W_{OV} x_k$
- For random embeddings, **capacity** $\approx$ **number of parameters**
 - See Cabannes et al. (2024); Nichani et al. (2025), extends to MLPs

Weights as associative memories

- Consider sets of **nearly orthonormal embeddings** $\{e_z\}_{z \in \mathcal{Z}}$ and $\{u_y\}_{y \in \mathcal{Y}}$:

$$\|e_z\| \approx 1 \quad \text{and} \quad e_z^\top e_{z'} \approx 0$$

$$\|u_y\| \approx 1 \quad \text{and} \quad u_y^\top u_{y'} \approx 0$$

- For **pairwise associations** $z \mapsto f(z)$ with $z \in \mathcal{M}$, define:

$$W = \sum_{z \in \mathcal{M}} u_{f(z)} e_z^\top \implies u_y^\top W e_z \approx \mathbb{1}\{y = f(z)\}$$

- Examples in Transformers:
 - Logits in attention heads: $x_k^\top W_{KQ} x_q$
 - Logits in next-token prediction: $u_y^\top U \sigma(V x_t)$ or $u_y^\top W_{OV} x_k$
- For random embeddings, **capacity** $\approx$ **number of parameters**
 - See Cabannes et al. (2024); Nichani et al. (2025), extends to MLPs
- Related to Hopfield (1982); Kohonen (1972); Willshaw et al. (1969); Iscen et al. (2017)

Gradients lead to associative memories

Lemma (Gradients as memories, B. et al., 2023)

Let p be a data distribution over $(z, y) \in [N]^2$, and consider the loss

$$L(W) = \mathbb{E}_{(z,y) \sim p}[\ell(y, F_W(z))], \quad F_W(z)_k = u_k^\top W e_z,$$

with ℓ the **cross-entropy loss** and e_z, u_k input/output embeddings.

Gradients lead to associative memories

Lemma (Gradients as memories, B. et al., 2023)

Let p be a data distribution over $(z, y) \in [N]^2$, and consider the loss

$$L(W) = \mathbb{E}_{(z,y) \sim p}[\ell(y, F_W(z))], \quad F_W(z)_k = u_k^\top W e_z,$$

with ℓ the **cross-entropy loss** and e_z , u_k input/output embeddings. Then,

$$\nabla L(W) = \sum_{k=1}^K \mathbb{E}_z[(\hat{p}_W(y = k|z) - p(y = k|z)) u_k e_z^\top]$$

Gradients lead to associative memories

Lemma (Gradients as memories, B. et al., 2023)

Let p be a data distribution over $(z, y) \in [N]^2$, and consider the loss

$$L(W) = \mathbb{E}_{(z,y) \sim p}[\ell(y, F_W(z))], \quad F_W(z)_k = u_k^\top W e_z,$$

with ℓ the **cross-entropy loss** and e_z , u_k input/output embeddings. Then,

$$\nabla L(W) = \sum_{k=1}^K \mathbb{E}_z[(\hat{p}_W(y = k|z) - p(y = k|z)) u_k e_z^\top]$$

- **Example:** $z \sim \text{Unif}([N])$, $y = f_*(z)$

Gradients lead to associative memories

Lemma (Gradients as memories, B. et al., 2023)

Let p be a data distribution over $(z, y) \in [N]^2$, and consider the loss

$$L(W) = \mathbb{E}_{(z,y) \sim p}[\ell(y, F_W(z))], \quad F_W(z)_k = \mathbf{u}_k^\top W \mathbf{e}_z,$$

with ℓ the **cross-entropy loss** and $\mathbf{e}_z$, $\mathbf{u}_k$ input/output embeddings. Then,

$$\nabla L(W) = \sum_{k=1}^K \mathbb{E}_z[(\hat{p}_W(y = k|z) - p(y = k|z)) \mathbf{u}_k \mathbf{e}_z^\top]$$

- **Example:** $z \sim \text{Unif}([N])$, $y = f_*(z)$

- ▶ After **one gradient step** on the population loss, assuming near-orthonormal embeddings

$$W_1 = \frac{\eta}{N} \sum_{z,k} \left(\mathbb{1}\{f_*(z) = k\} - \frac{1}{N} \right) \mathbf{u}_k \mathbf{e}_z^\top \quad \implies \quad \mathbf{u}_k^\top W_1 \mathbf{e}_z \approx \frac{\eta}{N} \left(\mathbb{1}\{f_*(z) = k\} - \frac{1}{N} \right)$$

Gradients lead to associative memories

Lemma (Gradients as memories, B. et al., 2023)

Let p be a data distribution over $(z, y) \in [N]^2$, and consider the loss

$$L(W) = \mathbb{E}_{(z,y) \sim p}[\ell(y, F_W(z))], \quad F_W(z)_k = \mathbf{u}_k^\top W \mathbf{e}_z,$$

with ℓ the **cross-entropy loss** and $\mathbf{e}_z$, $\mathbf{u}_k$ input/output embeddings. Then,

$$\nabla L(W) = \sum_{k=1}^K \mathbb{E}_z[(\hat{p}_W(y = k|z) - p(y = k|z)) \mathbf{u}_k \mathbf{e}_z^\top]$$

• **Example:** $z \sim \text{Unif}([N])$, $y = f_*(z)$

► After **one gradient step** on the population loss, assuming near-orthonormal embeddings

$$W_1 = \frac{\eta}{N} \sum_{z,k} \left(\mathbb{1}\{f_*(z) = k\} - \frac{1}{N} \right) \mathbf{u}_k \mathbf{e}_z^\top \implies \mathbf{u}_k^\top W_1 \mathbf{e}_z \approx \frac{\eta}{N} \left(\mathbb{1}\{f_*(z) = k\} - \frac{1}{N} \right)$$

► **Corollary:** $\hat{f}(z) = \arg \max_k \mathbf{u}_k^\top W_1 \mathbf{e}_z$ has near-perfect accuracy

Gradients lead to associative memories

Lemma (Gradients as memories, B. et al., 2023)

Let p be a data distribution over $(z, y) \in [N]^2$, and consider the loss

$$L(W) = \mathbb{E}_{(z,y) \sim p}[\ell(y, F_W(z))], \quad F_W(z)_k = \mathbf{u}_k^\top W \mathbf{e}_z,$$

with ℓ the **cross-entropy loss** and $\mathbf{e}_z, \mathbf{u}_k$ input/output embeddings. Then,

$$\nabla L(W) = \sum_{k=1}^K \mathbb{E}_z[(\hat{p}_W(y = k|z) - p(y = k|z)) \mathbf{u}_k \mathbf{e}_z^\top]$$

- **Example:** $z \sim \text{Unif}([N]), y = f_*(z)$

► After **one gradient step** on the population loss, assuming near-orthonormal embeddings

$$W_1 = \frac{\eta}{N} \sum_{z,k} \left(\mathbb{1}\{f_*(z) = k\} - \frac{1}{N} \right) \mathbf{u}_k \mathbf{e}_z^\top \implies \mathbf{u}_k^\top W_1 \mathbf{e}_z \approx \frac{\eta}{N} \left(\mathbb{1}\{f_*(z) = k\} - \frac{1}{N} \right)$$

► **Corollary:** $\hat{f}(z) = \arg \max_k \mathbf{u}_k^\top W_1 \mathbf{e}_z$ has near-perfect accuracy

- More generally, replace $\mathbf{u}_k$ by “backward” vector

Gradients lead to associative memories

Lemma (Gradients as memories, B. et al., 2023)

Let p be a data distribution over $(z, y) \in [N]^2$, and consider the loss

$$L(W) = \mathbb{E}_{(z,y) \sim p}[\ell(y, F_W(z))], \quad F_W(z)_k = \mathbf{u}_k^\top W \mathbf{e}_z,$$

with ℓ the **cross-entropy loss** and $\mathbf{e}_z$, $\mathbf{u}_k$ input/output embeddings. Then,

$$\nabla L(W) = \sum_{k=1}^K \mathbb{E}_z[(\hat{p}_W(y = k|z) - p(y = k|z)) \mathbf{u}_k \mathbf{e}_z^\top]$$

- **Example:** $z \sim \text{Unif}([N])$, $y = f_*(z)$

► After **one gradient step** on the population loss, assuming near-orthonormal embeddings

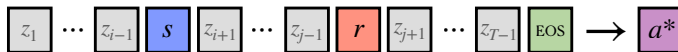
$$W_1 = \frac{\eta}{N} \sum_{z,k} \left(\mathbb{1}\{f_*(z) = k\} - \frac{1}{N} \right) \mathbf{u}_k \mathbf{e}_z^\top \implies \mathbf{u}_k^\top W_1 \mathbf{e}_z \approx \frac{\eta}{N} \left(\mathbb{1}\{f_*(z) = k\} - \frac{1}{N} \right)$$

► **Corollary:** $\hat{f}(z) = \arg \max_k \mathbf{u}_k^\top W_1 \mathbf{e}_z$ has near-perfect accuracy

- More generally, replace $\mathbf{u}_k$ by “backward” vector

Note: related to (Ba et al., 2022; Damian et al., 2022; Oymak et al., 2023; Yang and Hu, 2021)

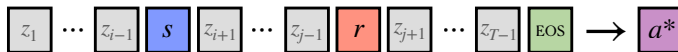
Application to factual recall: toy model



The capital of France is Paris

- $s \in \mathcal{S}$: subject token
- $r \in \mathcal{R}$: relation token
- $a^*(s, r) \in \mathcal{A}_r$: attribute/fact to be stored
- $z_i \in \mathcal{N}$: noise tokens

Application to factual recall: toy model



The capital of France is Paris

- $s \in \mathcal{S}$: subject token
- $r \in \mathcal{R}$: relation token
- $a^*(s, r) \in \mathcal{A}_r$: attribute/fact to be stored
- $z_i \in \mathcal{N}$: noise tokens

Q: How do Transformers solve this?

Two mechanisms

- One-layer Transformer, with or without MLP, random embeddings
- Embedding dimension d , head dimension d_h , MLP width m , H heads

Theorem (Nichani et al., 2025, informal)

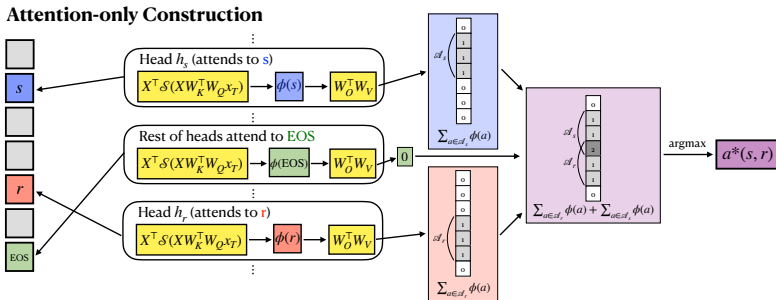
A one-layer Transformer with or without MLP can achieve perfect factual recall with $\tilde{O}(SR)$ parameters. (note: attention-only needs large enough d)

Two mechanisms

- One-layer Transformer, with or without MLP, random embeddings
- Embedding dimension d , head dimension d_h , MLP width m , H heads

Theorem (Nichani et al., 2025, informal)

A one-layer Transformer with or without MLP can achieve perfect factual recall with $\tilde{O}(SR)$ parameters. (note: attention-only needs large enough d)



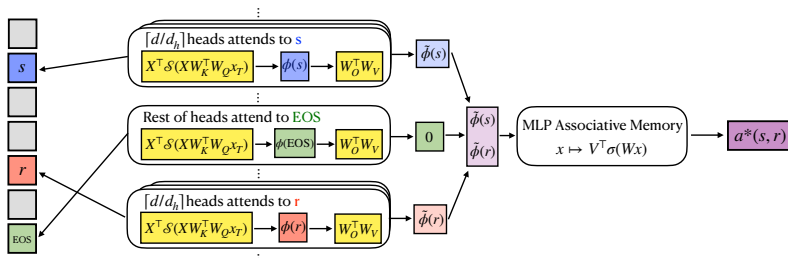
Two mechanisms

- One-layer Transformer, with or without MLP, random embeddings
- Embedding dimension d , head dimension d_h , MLP width m , H heads

Theorem (Nichani et al., 2025, informal)

A one-layer Transformer with or without MLP can achieve perfect factual recall with $\tilde{O}(SR)$ parameters. (note: attention-only needs large enough d)

Attention + MLP Construction

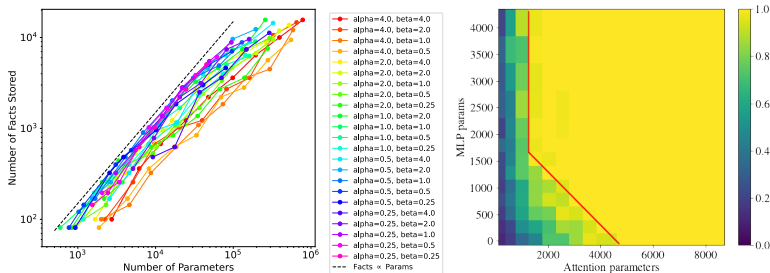


Two mechanisms

- One-layer Transformer, with or without MLP, random embeddings
- Embedding dimension d , head dimension d_h , MLP width m , H heads

Theorem (Nichani et al., 2025, informal)

A one-layer Transformer with or without MLP can achieve perfect factual recall with $\tilde{O}(SR)$ parameters. (note: attention-only needs large enough d)



Training dynamics

- One-layer Transformer with **linear attention** and **one-hot** embeddings
- Gradient flow with initialization $W_{OV}(a, z), w_{KQ}(z) \approx \alpha > 0$

Theorem (Nichani et al., 2025, informal)

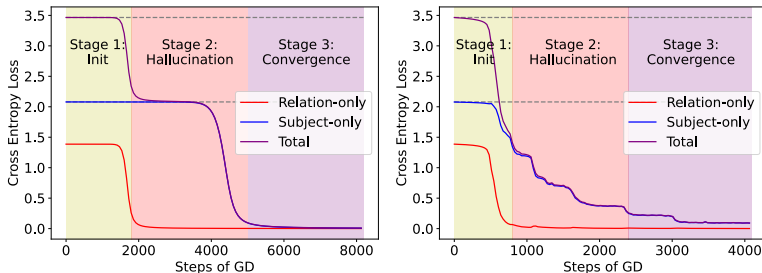
- *There is global convergence to zero loss*
- *There is an intermediate phase where the model predicts using $p(a|r)$ instead of $p(a|s, r)$*

Training dynamics

- One-layer Transformer with **linear attention** and **one-hot** embeddings
- Gradient flow with initialization $W_{OV}(a, z), w_{KQ}(z) \approx \alpha > 0$

Theorem (Nichani et al., 2025, informal)

- *There is global convergence to zero loss*
- *There is an intermediate phase where the model predicts using $p(a|r)$ instead of $p(a|s, r)$*
- Intermediate phase corresponds to **hallucination** (uniform over $\mathcal{A}_r$, ignoring s)

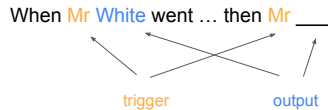


Outline

1 Weights as memories

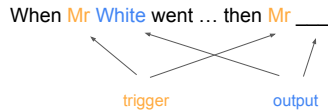
2 Context as memory

A toy data model for in-context retrieval



When Mr White went to the mall, it started raining, then Mr White witnessed an odd occurrence. While walking around the mall with his family, Mr White heard the sound of a helicopter landing in the parking lot. Curious, he made his way over to see what was going on.

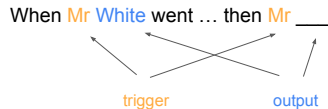
A toy data model for in-context retrieval



When Mr White went to the mall, it started raining, then Mr White witnessed an odd occurrence. While walking around the mall with his family, Mr White heard the sound of a helicopter landing in the parking lot. Curious, he made his way over to see what was going on.

Fix **trigger tokens**: $q_1, \dots, q_K$

A toy data model for in-context retrieval



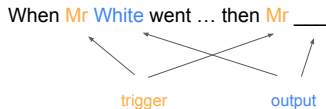
When Mr White went to the mall, it started raining, then Mr White witnessed an odd occurrence. While walking around the mall with his family, Mr White heard the sound of a helicopter landing in the parking lot. Curious, he made his way over to see what was going on.

Fix **trigger tokens**: $q_1, \dots, q_K$

Sample each sequence $z_{1:T} \in [N]^T$ as follows

- **Output tokens**: $o_k \sim \pi_o(\cdot | q_k)$ (random)

A toy data model for in-context retrieval



When Mr White went to the mall, it started raining, then Mr White witnessed an odd occurrence. While walking around the mall with his family, Mr White heard the sound of a helicopter landing in the parking lot. Curious, he made his way over to see what was going on.

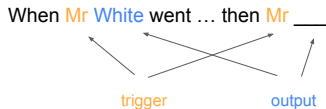
Fix **trigger tokens**: $q_1, \dots, q_K$

Sample each sequence $z_{1:T} \in [N]^T$ as follows

- **Output tokens**: $o_k \sim \pi_o(\cdot | q_k)$ (*random*)
- **Sequence-specific Markov model**: $z_1 \sim \pi_1$, $z_t | z_{t-1} \sim p(\cdot | z_{t-1})$ with

$$p(j|i) = \begin{cases} \mathbb{1}\{j = o_k\}, & \text{if } i = q_k, \quad k = 1, \dots, K \\ \pi_b(j|i), & \text{o/w.} \end{cases}$$

A toy data model for in-context retrieval



When Mr White went to the mall, it started raining, then Mr White witnessed an odd occurrence. While walking around the mall with his family, Mr White heard the sound of a helicopter landing in the parking lot. Curious, he made his way over to see what was going on.

Fix **trigger tokens**: $q_1, \dots, q_K$

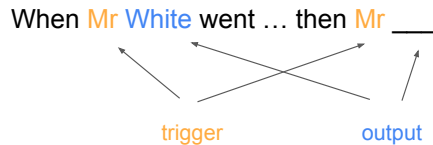
Sample each sequence $z_{1:T} \in [N]^T$ as follows

- **Output tokens**: $o_k \sim \pi_o(\cdot | q_k)$ (random)
- **Sequence-specific Markov model**: $z_1 \sim \pi_1$, $z_t | z_{t-1} \sim p(\cdot | z_{t-1})$ with

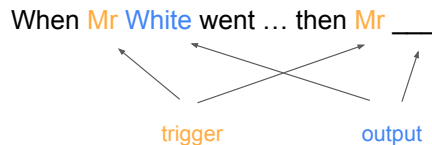
$$p(j|i) = \begin{cases} \mathbb{1}\{j = o_k\}, & \text{if } i = q_k, \quad k = 1, \dots, K \\ \pi_b(j|i), & \text{o/w.} \end{cases}$$

π_b : **global bigrams** model (estimated from Karpathy's character-level Shakespeare)

Transformers on the bigram task

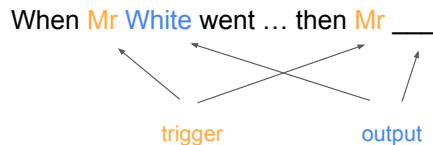


Transformers on the bigram task



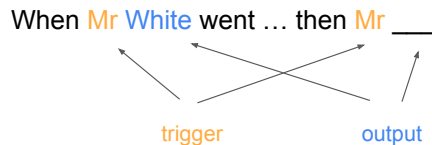
- **1-layer transformer fails:** $\sim 55\%$ accuracy on in-context output predictions

Transformers on the bigram task



- **1-layer transformer fails:** $\sim 55\%$ accuracy on in-context output predictions
- **2-layer transformer succeeds:** $\sim 99\%$ accuracy

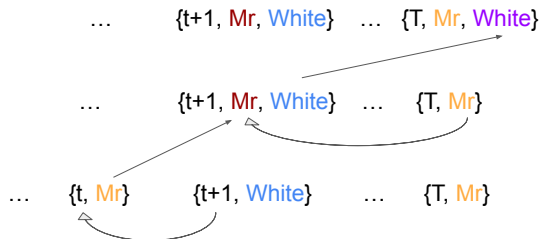
Transformers on the bigram task



- **1-layer transformer fails:** $\sim 55\%$ accuracy on in-context output predictions
- **2-layer transformer succeeds:** $\sim 99\%$ accuracy

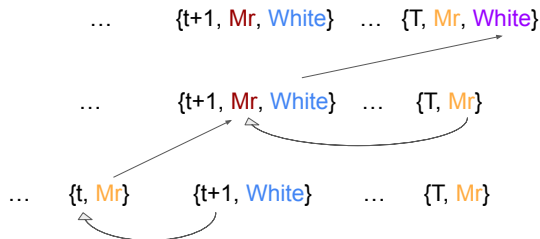
See (Sanford, Hsu, and Telgarsky, 2023, 2024b) for representational lower bounds

Induction head mechanism (Elhage et al., 2021; Olsson et al., 2022)



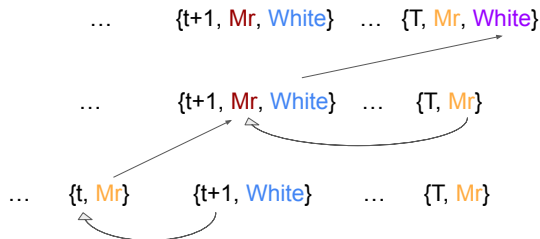
- 1st layer: **previous-token head**
 - ▶ attends to previous token and copies it to residual stream

Induction head mechanism (Elhage et al., 2021; Olsson et al., 2022)

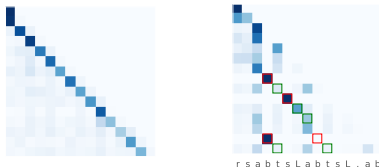


- 1st layer: **previous-token head**
 - ▶ attends to previous token and copies it to residual stream
- 2nd layer: **induction head**
 - ▶ attends to output of previous token head, copies attended token

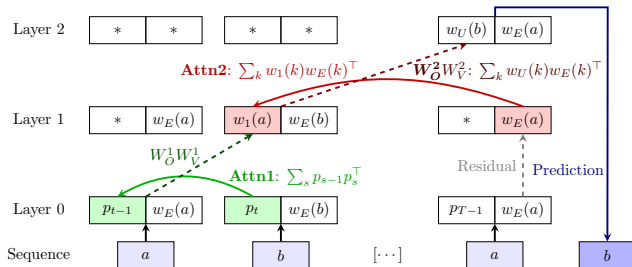
Induction head mechanism (Elhage et al., 2021; Olsson et al., 2022)



- 1st layer: **previous-token head**
 - ▶ attends to previous token and copies it to residual stream
- 2nd layer: **induction head**
 - ▶ attends to output of previous token head, copies attended token
- Matches observed attention scores:



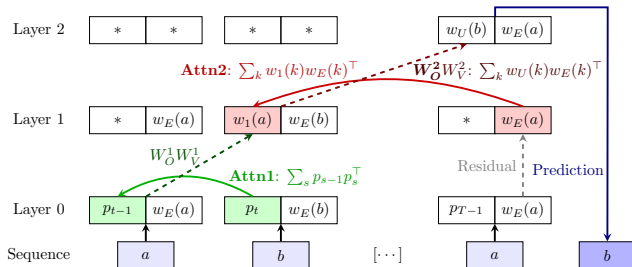
Induction head with associative memory weights



$$W_{KQ}^1 = \sum_{t=2}^T p_t p_{t-1}^\top, \quad W_{KQ}^2 = \sum_{k \in Q} e_k \tilde{e}_k^\top, \quad W_{OV}^2 = \sum_{k=1}^N u_k e_k^\top,$$

- Random embeddings e_k , u_k , random matrix W_{OV}^1 (frozen at init)
- **Remapped** previous tokens: $\tilde{e}_k := W_{OV}^1 e_k$

Induction head with associative memory weights



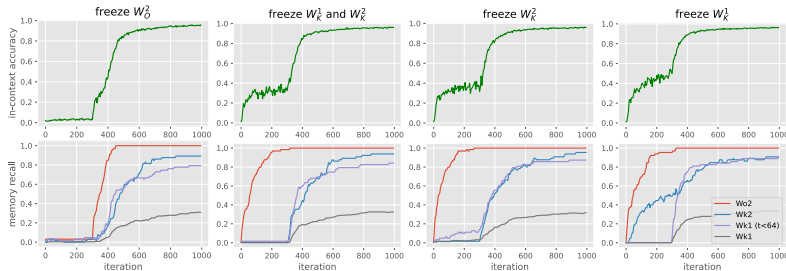
$$W_{KQ}^1 = \sum_{t=2}^T p_t p_{t-1}^\top, \quad W_{KQ}^2 = \sum_{k \in Q} e_k \tilde{e}_k^\top, \quad W_{OV}^2 = \sum_{k=1}^N u_k e_k^\top,$$

- Random embeddings e_k , u_k , random matrix W_{OV}^1 (frozen at init)
- **Remapped** previous tokens: $\tilde{e}_k := W_{OV}^1 e_k$

Q: Does this match practice?

Empirically probing the dynamics

Train only W_{KQ}^1 , W_{KQ}^2 , W_{OV}^2 , loss on predictable tokens after trigger

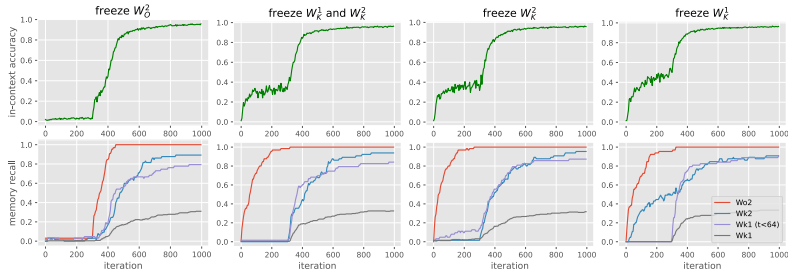


- “Memory recall **probes**”: for target memory $W_* = \sum_{i=1}^M u_i e_i^\top$, compute

$$R(\hat{W}, W_*) = \frac{1}{M} \sum_{i=1}^M \mathbb{1}\{i = \arg \max_j u_j^\top \hat{W} e_i\}$$

Empirically probing the dynamics

Train only W_{KQ}^1 , W_{KQ}^2 , W_{OV}^2 , loss on predictable tokens after trigger



- “Memory recall **probes**”: for target memory $W_* = \sum_{i=1}^M u_i e_i^\top$, compute

$$R(\hat{W}, W_*) = \frac{1}{M} \sum_{i=1}^M \mathbb{1}\{i = \arg \max_j u_j^\top \hat{W} e_i\}$$

- Natural learning “**order**”: W_{OV}^2 first, W_{KQ}^2 next, W_{KQ}^1 last
- Joint learning is faster

Gradient steps for the bigram task

Theorem (B. et al., 2023, informal)

*In a simplified setup, we can recover the desired associative memories with **3 sequential gradient steps** on the population loss: first on W_{OV}^2 , then W_{KQ}^2 , then W_{KQ}^1 .*

Gradient steps for the bigram task

Theorem (B. et al., 2023, informal)

*In a simplified setup, we can recover the desired associative memories with **3 sequential gradient steps** on the population loss: first on W_{OV}^2 , then W_{KQ}^2 , then W_{KQ}^1 .*

Key ideas

- Attention is uniform at initialization $\implies$ inputs are sums of embeddings
- W_{OV}^2 : correct output appears w.p. 1, while other tokens are noisy and cond. indep. of z_T
- $W_{KQ}^{1/2}$: correct associations lead to more focused attention

Gradient steps for the bigram task

Theorem (B. et al., 2023, informal)

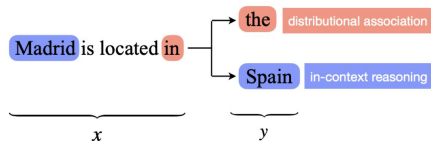
*In a simplified setup, we can recover the desired associative memories with **3 sequential gradient steps** on the population loss: first on W_{OV}^2 , then W_{KQ}^2 , then W_{KQ}^1 .*

Key ideas

- Attention is uniform at initialization $\implies$ inputs are sums of embeddings
- W_{OV}^2 : correct output appears w.p. 1, while other tokens are noisy and cond. indep. of z_T
- $W_{KQ}^{1/2}$: correct associations lead to more focused attention

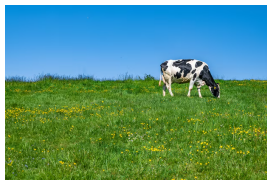
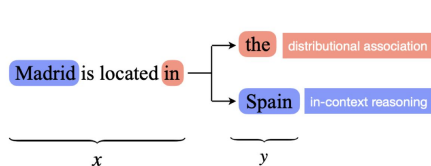
see also (Snell et al., 2021; Oymak et al., 2023)

Distributional vs in-context associations



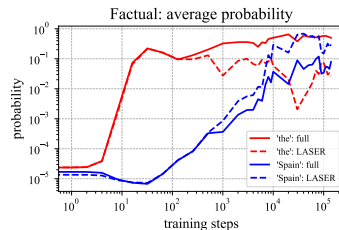
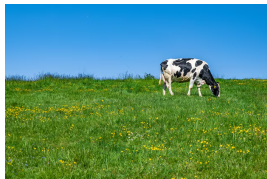
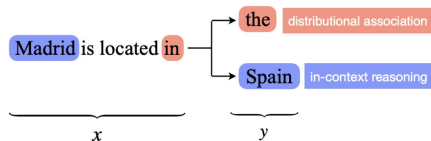
- **Distributional associations** (e.g., common bigrams like “in the”) are learned much faster than in-context reasoning, tend to be stored in late MLPs:
 - ▶ “Madrid is located in” → {the, Spain} on Pythia-1B

Distributional vs in-context associations



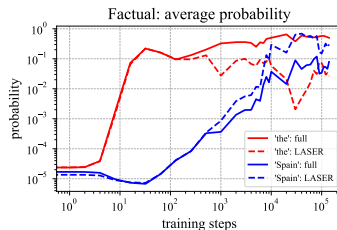
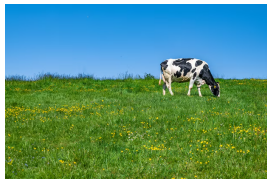
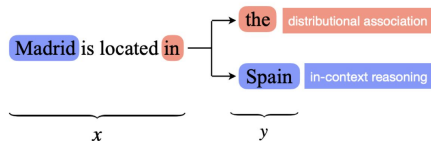
- **Distributional associations** (e.g., common bigrams like “in the”) are learned much faster than in-context reasoning, tend to be stored in late MLPs:
 - ▶ “Madrid is located in” → {the, Spain} on Pythia-1B

Distributional vs in-context associations



- **Distributional associations** (e.g., common bigrams like “in the”) are learned much faster than in-context reasoning, tend to be stored in late MLPs:
 - ▶ “Madrid is located in” $\rightarrow$ {the, Spain} on Pythia-1B
 - ▶ Ablating late-layer MLPs (Sharma et al., 2023) changes prediction from global to in-context

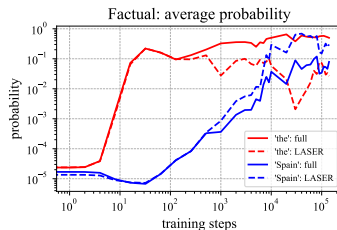
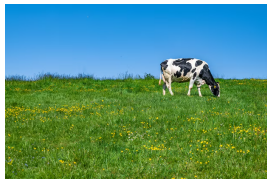
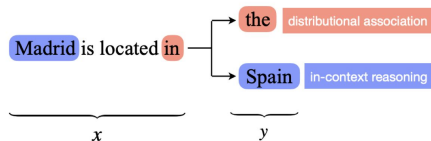
Distributional vs in-context associations



- **Distributional associations** (e.g., common bigrams like “in the”) are learned much faster than in-context reasoning, tend to be stored in late MLPs:
 - ▶ “Madrid is located in” $\rightarrow$ {the, Spain} on Pythia-1B
 - ▶ Ablating late-layer MLPs (Sharma et al., 2023) changes prediction from global to in-context
- We study this on simple induction head task + noisy bigram token z (“the”):

... **q** **a** d k i e ... **q** **a** ... **q** **z** ... **q** **z** ... **q** **z** ... **q** **a** ...

Distributional vs in-context associations



- **Distributional associations** (e.g., common bigrams like “in the”) are learned much faster than in-context reasoning, tend to be stored in late MLPs:
 - ▶ “Madrid is located in” $\rightarrow$ {the, Spain} on Pythia-1B
 - ▶ Ablating late-layer MLPs (Sharma et al., 2023) changes prediction from global to in-context
- We study this on simple induction head task + noisy bigram token z (“the”):

... **q** **a** d k i e ... **q** **a** ... **q** **z** ... **q** **z** ... **q** **z** ... **q** **a** ...

Theorem (Chen, Bruna, and B., 2024, informal)

In toy model above, feed-forward layer learns global bigram after $O(1)$ samples, attention after $O(N)$ samples due to noise.

Multi-step reasoning (Wang et al., 2025)

- Composing **in-context** and **in-weights** knowledge:

John is in the *kitchen*. *Mary* is in the *room*. *He* holds the *apple*.
She holds the *ball*. Where is the *fruit*?

Multi-step reasoning (Wang et al., 2025)

- Composing **in-context** and **in-weights** knowledge:

John is in the kitchen. Mary is in the room. He holds the apple.

She holds the ball. Where is the fruit?

- ▶ **In-context** associations: (apple → He) and (John → kitchen)
- ▶ **In-weights** associations: (apple → fruit) and (John → He)
- ▶ Composition of 2 hops, harder than the (1-hop) induction head task: (Mr → White)

Multi-step reasoning (Wang et al., 2025)

- Composing **in-context** and **in-weights** knowledge:

*John is in the kitchen. Mary is in the room. He holds the apple.
She holds the ball. Where is the fruit?*

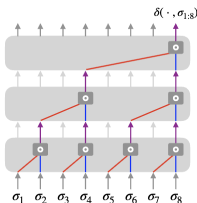
- ▶ **In-context** associations: (apple → He) and (John → kitchen)
- ▶ **In-weights** associations: (apple → fruit) and (John → He)
- ▶ Composition of 2 hops, harder than the (1-hop) induction head task: (Mr → White)
- We consider **k-hop reasoning**: compose k in-context and k in-weights functions

Multi-step reasoning (Wang et al., 2025)

- Composing **in-context** and **in-weights** knowledge:

*John is in the kitchen. Mary is in the room. He holds the apple.
She holds the ball. Where is the fruit?*

- In-context** associations: (**apple** $\rightarrow$ **He**) and (**John** $\rightarrow$ **kitchen**)
- In-weights** associations: (**apple** $\rightarrow$ **fruit**) and (**John** $\rightarrow$ **He**)
- Composition of 2 hops, harder than the (1-hop) induction head task: (**Mr** $\rightarrow$ **White**)
- We consider **k -hop reasoning**: compose k in-context and k in-weights functions
- Can be solved with just $\log k$ transformer layers (see also Liu et al., 2023; Sanford et al., 2024a)

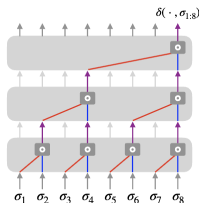


Multi-step reasoning (Wang et al., 2025)

- Composing **in-context** and **in-weights** knowledge:

*John is in the kitchen. Mary is in the room. He holds the apple.
She holds the ball. Where is the fruit?*

- In-context** associations: (apple $\rightarrow$ He) and (John $\rightarrow$ kitchen)
- In-weights** associations: (apple $\rightarrow$ fruit) and (John $\rightarrow$ He)
- Composition of 2 hops, harder than the (1-hop) induction head task: (Mr $\rightarrow$ White)
- We consider **k -hop reasoning**: compose k in-context and k in-weights functions
- Can be solved with just $\log k$ transformer layers (see also Liu et al., 2023; Sanford et al., 2024a)



Q: what about training dynamics?

Training dynamics for k -hop reasoning

Failure of gradient descent (“statistical query” model)

Theorem (Wang, Nichani, et al., 2025+, informal)

*Gradient descent requires **exponential** in k samples or compute to solve the k -hop task.*

Training dynamics for k -hop reasoning

Failure of gradient descent (“statistical query” model)

Theorem (Wang, Nichani, et al., 2025+, informal)

*Gradient descent requires **exponential** in k samples or compute to solve the k -hop task.*

Easy-to-hard data to the rescue

Theorem (Wang, Nichani, et al., 2025+, informal)

*When using easy-to-hard **curriculum** learning, or training on a **mixture** of hops, gradient descent solves k -hop with $O(k)$ samples/compute.*

Training dynamics for k -hop reasoning

Failure of gradient descent (“statistical query” model)

Theorem (Wang, Nichani, et al., 2025+, informal)

*Gradient descent requires **exponential** in k samples or compute to solve the k -hop task.*

Easy-to-hard data to the rescue

Theorem (Wang, Nichani, et al., 2025+, informal)

*When using easy-to-hard **curriculum** learning, or training on a **mixture** of hops, gradient descent solves k -hop with $O(k)$ samples/compute.*

- Hops 1, 2, 4, etc. are learned incrementally (layer-wise)

Training dynamics for k -hop reasoning

Failure of gradient descent (“statistical query” model)

Theorem (Wang, Nichani, et al., 2025+, informal)

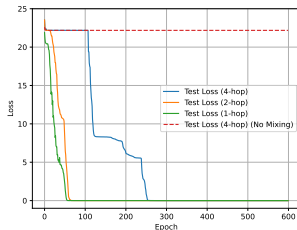
*Gradient descent requires **exponential** in k samples or compute to solve the k -hop task.*

Easy-to-hard data to the rescue

Theorem (Wang, Nichani, et al., 2025+, informal)

*When using easy-to-hard **curriculum** learning, or training on a **mixture** of hops, gradient descent solves k -hop with $O(k)$ samples/compute.*

- Hops 1, 2, 4, etc. are learned incrementally (layer-wise)



Concluding remarks

Mechanisms in Transformers

- Weights as associative memories
- (Multi-hop) reasoning with attention circuits
- Distributional associations vs reasoning in MLPs vs attention

Concluding remarks

Mechanisms in Transformers

- Weights as associative memories
- (Multi-hop) reasoning with attention circuits
- Distributional associations vs reasoning in MLPs vs attention

Training dynamics elucidate phenomena

- Hallucinations as an intermediate training phase
- Multi-hop reasoning can be solved only with curriculum
- MLPs learn faster than attention

Concluding remarks

Mechanisms in Transformers

- Weights as associative memories
- (Multi-hop) reasoning with attention circuits
- Distributional associations vs reasoning in MLPs vs attention

Training dynamics elucidate phenomena

- Hallucinations as an intermediate training phase
- Multi-hop reasoning can be solved only with curriculum
- MLPs learn faster than attention

Future directions

- Reasoning vs knowledge in computer vision and physical sciences
- Learning embeddings
- External / agentic memory (Zhu et al., 2025)

Concluding remarks

Mechanisms in Transformers

- Weights as associative memories
- (Multi-hop) reasoning with attention circuits
- Distributional associations vs reasoning in MLPs vs attention

Training dynamics elucidate phenomena

- Hallucinations as an intermediate training phase
- Multi-hop reasoning can be solved only with curriculum
- MLPs learn faster than attention

Future directions

- Reasoning vs knowledge in computer vision and physical sciences
- Learning embeddings
- External / agentic memory (Zhu et al., 2025)

Thank you!

References I

- Z. Allen-Zhu and Y. Li. Physics of language models: Part 3.3, knowledge capacity scaling laws. *arXiv preprint arXiv:2404.05405*, 2024.
- A. B., V. Cabannes, D. Bouchacourt, H. Jegou, and L. Bottou. Birth of a transformer: A memory viewpoint. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
- J. Ba, M. A. Erdogdu, T. Suzuki, Z. Wang, D. Wu, and G. Yang. High-dimensional asymptotics of feature learning: How one gradient step improves the representation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- V. Cabannes, E. Dohmatob, and A. B. Scaling laws for associative memories. In *International Conference on Learning Representations (ICLR)*, 2024.
- L. Chen, J. Bruna, and A. B. How truncating weights improves reasoning in language models. *arXiv preprint arXiv:2406.03068*, 2024.
- A. Damian, J. Lee, and M. Soltanolkotabi. Neural networks can learn representations with gradient descent. In *Conference on Learning Theory (COLT)*, 2022.
- N. Elhage, N. Nanda, C. Olsson, T. Henighan, N. Joseph, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, N. DasSarma, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021.

References II

- M. Geva, R. Schuster, J. Berant, and O. Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.
- J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.
- A. Iscen, T. Furon, V. Gripon, M. Rabbat, and H. Jégou. Memory vectors for similarity search in high-dimensional spaces. *IEEE transactions on big data*, 4(1):65–77, 2017.
- T. Kohonen. Correlation matrix memories. *IEEE Transactions on Computers*, 1972.
- B. Liu, J. T. Ash, S. Goel, A. Krishnamurthy, and C. Zhang. Transformers learn shortcuts to automata. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.
- K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- E. Nichani, J. D. Lee, and A. B. Understanding factual recall in transformers via associative memories. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.
- C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, S. Johnston, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

References III

- S. Oymak, A. S. Rawat, M. Soltanolkotabi, and C. Thrampoulidis. On the role of attention in prompt-tuning. In *International Conference on Machine Learning*, 2023.
- C. Sanford, D. Hsu, and M. Telgarsky. Representational strengths and limitations of transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
- C. Sanford, B. Fatemi, E. Hall, A. Tsitsulin, M. Kazemi, J. Halcrow, B. Perozzi, and V. Mirrokni. Understanding transformer reasoning capabilities via graph algorithms. *arXiv preprint arXiv:2405.18512*, 2024a.
- C. Sanford, D. Hsu, and M. Telgarsky. One-layer transformers fail to solve the induction heads task. *arXiv preprint arXiv:2408.14332*, 2024b.
- P. Sharma, J. T. Ash, and D. Misra. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. *arXiv preprint arXiv:2312.13558*, 2023.
- C. Snell, R. Zhong, D. Klein, and J. Steinhardt. Approximating how single head attention learns. *arXiv preprint arXiv:2103.07601*, 2021.
- K. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.
- Z. Wang, E. Nichani, A. B., A. Damian, D. Hsu, J. D. Lee, and D. Wu. Learning compositional functions with transformers from easy-to-hard data. In *Conference on Learning Theory (COLT)*, 2025.

References IV

- D. J. Willshaw, O. P. Buneman, and H. C. Longuet-Higgins. Non-holographic associative memory. *Nature*, 222(5197):960–962, 1969.
- G. Yang and E. J. Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.
- H. Zhu, T. Guo, S. Mei, S. Russell, N. Ghosh, A. B., and J. Jiao. Gsm-agent: Understanding agentic reasoning using controllable environments. *arXiv preprint arXiv:2509.21998*, 2025.