Transformers, parallelism, and the role of depth

Daniel Hsu

Columbia University

University of Chicago Booth School of Business
October 31, 2024

Large Language Models (LLMs)

Shannon's *N*-gram model:
Distribution of next word is determined by last *N* words

`once upon a time and a very`

Small *N*:

Produces garbage; not predictive

Large *N*:

Too many parameters: $\exp(\Omega(N))$; likely overfitting to training data

Today's solution: neural language models

Zoo of neural architectures for language models!

Finding Structure in Time 1990

JEFFREY L. ELMAN

LONG SHORT-TERM MEMORY 1997

Neural Computation 9(8):1735–1780, 1997

Sepp Hochreiter   Jürgen Schmidhuber

Sequence to Sequence Learning with Neural Networks 2014

Ilya Sutskever, Google, ilyasu@google.com
Oriol Vinyals, Google, vinyals@google.com
Quoc V. Le, Google, qvl@google.com

ADAPTIVE LANGUAGE MODELING USING MINIMUM DISCRIMINANT ESTIMATION* 1992

S. Della Pietra, V. Della Pietra, R. L. Mercer, S. Roukos
Continuous Speech Recognition Group,
Thomas J. Watson Research Center
P. O. Box 704, Yorktown Heights, NY 10598

Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation 2014

Kyunghyun Cho
Bart van Merriënboer Caglar Gulcehre
Université de Montréal
Dzmitry Bahdanau
Jacobs University, Germany
...ersity.de
...R Senior Fellow

What sets one neural architecture apart from the others?

Réjean Ducharme DUCHARME@IRO.UMONTREAL.CA
Pascal Vincent VINCENTP@IRO.UMONTREAL.CA
Christian Jauvin JAUVINC@IRO.UMONTREAL.CA

NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE 2015

Dzmitry Bahdanau
Jacobs University Bremen, Germany
KyungHyun Cho Yoshua Bengio*
Université de Montréal

A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning 2008

Ronan Collobert COLLOBER@NEC-LABS.COM
Jason Weston JASONW@NEC-LABS.COM

Attention Is All You Need 2017

Ashish Vaswani*, Google Brain, avaswani@google.com
Noam Shazeer*, Google Brain, noam@google.com
Niki Parmar*, Google Research, nikip@google.com
Jakob Uszkoreit*, Google Research, usz@google.com
Llion Jones*, Google Research, llion@google.com
Aidan N. Gomez* †, University of Toronto, aidan@cs.toronto.edu
Łukasz Kaiser*, Google Brain, lukaszkaiser@google.com
Illia Polosukhin* ‡

Recurrent Neural Network based Language Modeling in Meeting Recognition 2011

Stefan Kombrink, Tomáš Mikolov, Martin Karafiát, Lukáš Burget

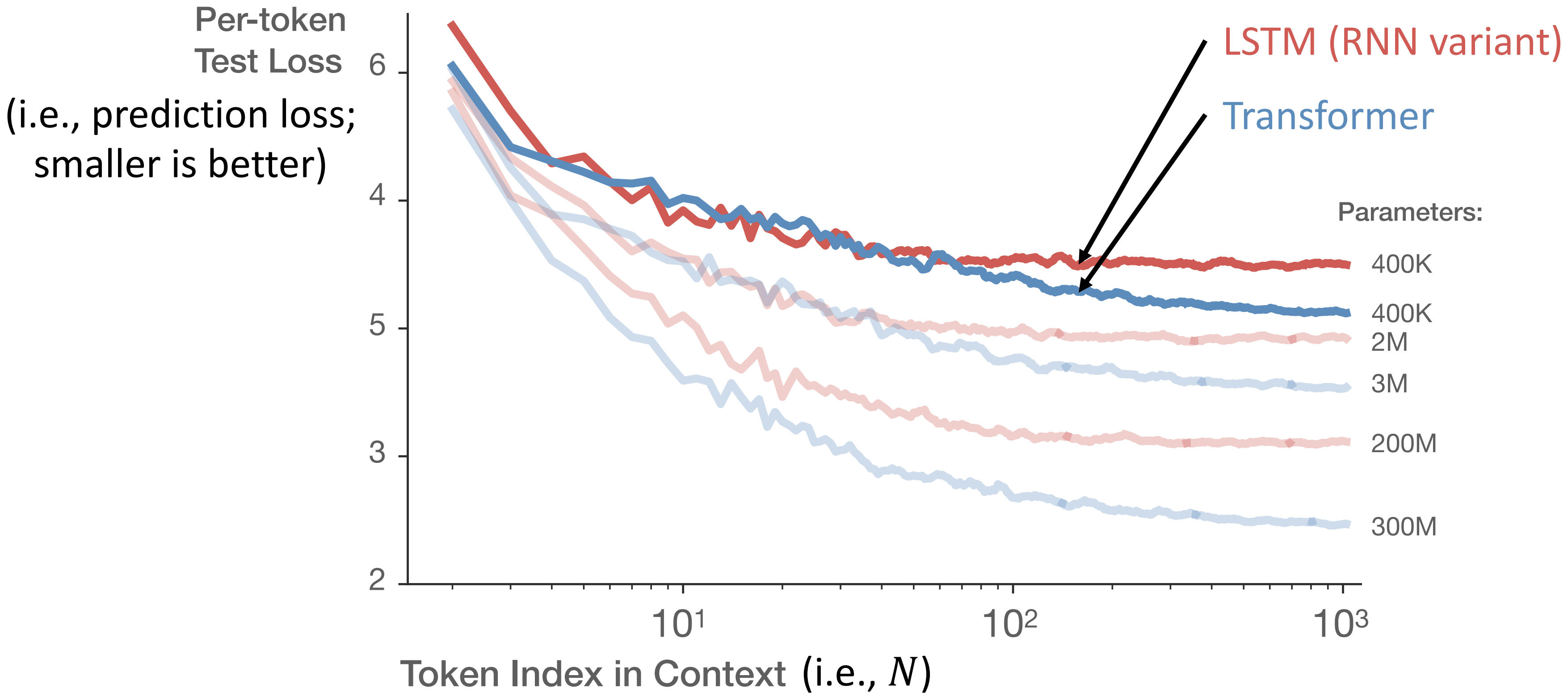
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding 2018

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova

Language Models are Unsupervised Multitask Learners 2019

Alec Radford *[1] Jeffrey Wu *[1] Rewon Child [1] David Luan [1] Dario Amodei **[1] Ilya Sutskever **[1]

RNN versus Transformer



[Figure from Kaplan et al, 2020]

The measure of a model

Many aspects may contribute to a neural architecture's success:

- Representational power
- Complexity of inference
- Learnability with SGD
- …

"Fair comparisons" of neural architectures are difficult:

- Parameter count?
- Inference time or cost?
- Data efficiency?
- …

Focus of this talk:
representational power
enabled by parallelism

Plan for the talk

1. Role of depth for in-context learning
2. Transformers & Massively Parallel Computation
3. Limitations of sequential neural architectures

Joint work with:
Clayton Sanford (Columbia → Google Research)
Matus Telgarsky (New York University)

[NeurIPS 2023, ICML 2024, arXiv:2408.14332]

0. Basics about transformers

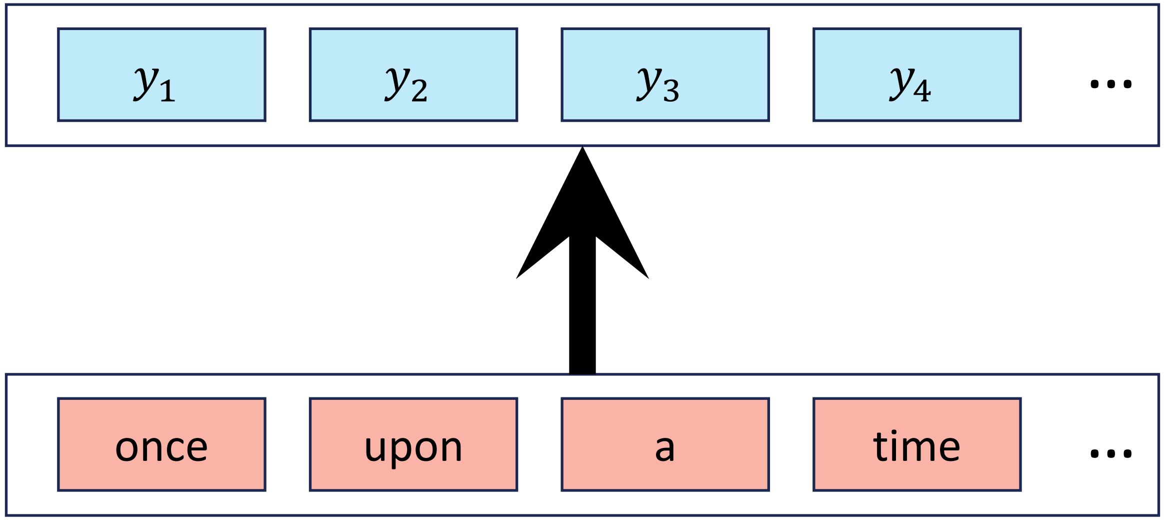
Transformers [Vaswani et al, 2017]

Transformer: a kind of sequence-to-sequence map, formed by compositions of self-attention heads

Ingredients:

1. Ways to embed tokens into vector space
2. Way to for embedded tokens to "interact" and produce new vectors

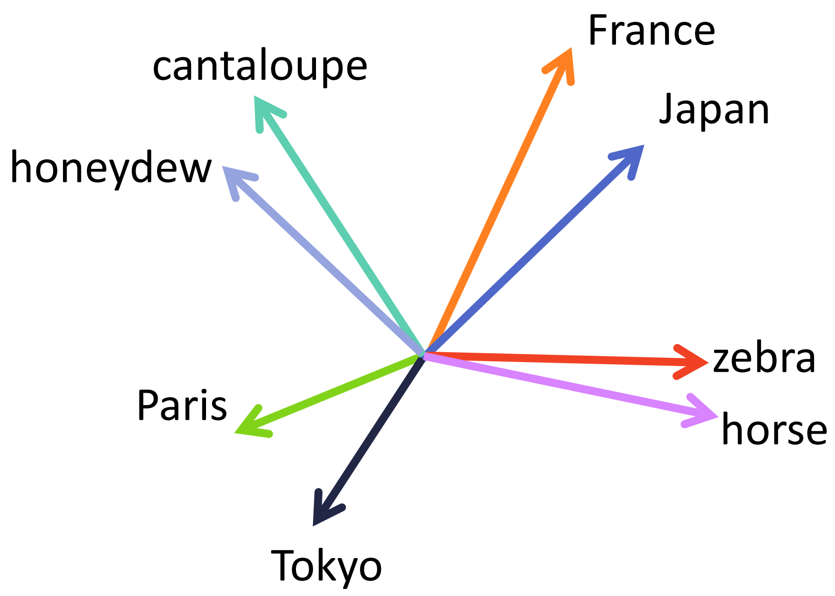
output sequence
(vectors)



input sequence
(tokens or vectors)

Word / token embeddings

Represent words with vectors [Deerwester et al, 1990; Mikolov et al, 2013; …]



__Example__:

Paris – France + Japan ≈ Tokyo

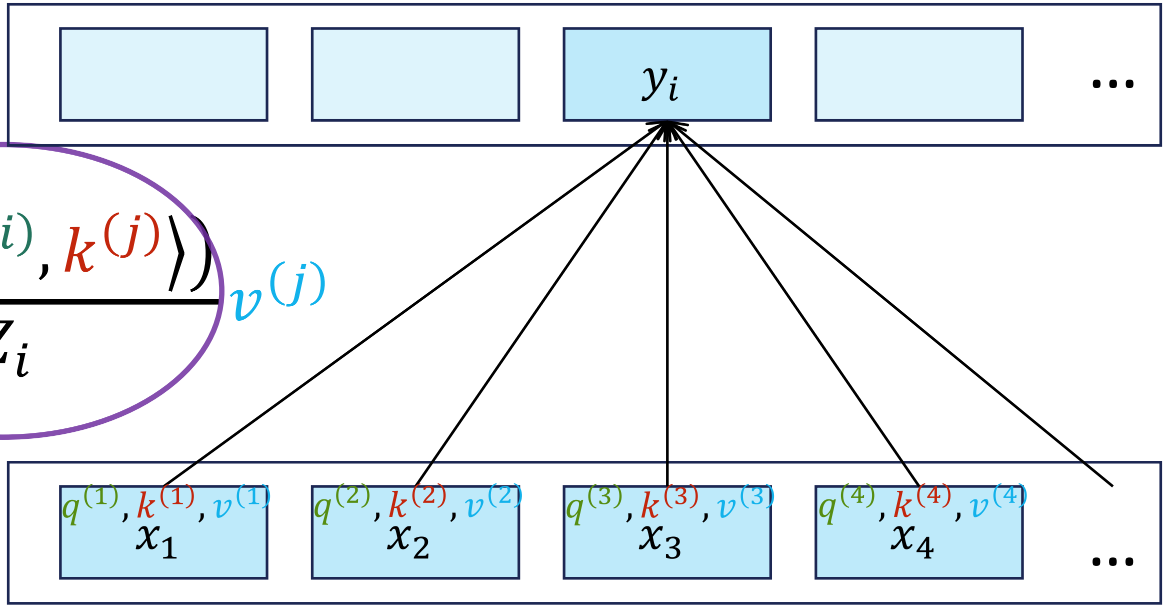
Data-driven "geometry" captures semantics

Self-attention head

Token embeddings created using "trained" multilayer Perceptrons (MLPs)

1. Independently create N query/key/value vectors from $x_1, \ldots, x_N$
2. For each $i \in [N]$: i^{th} output y_i = weighted average of all N values, where weights = "softmax" of $\langle i^{\text{th}}$ query, j^{th} key$\rangle$ for all $j \in [N]$

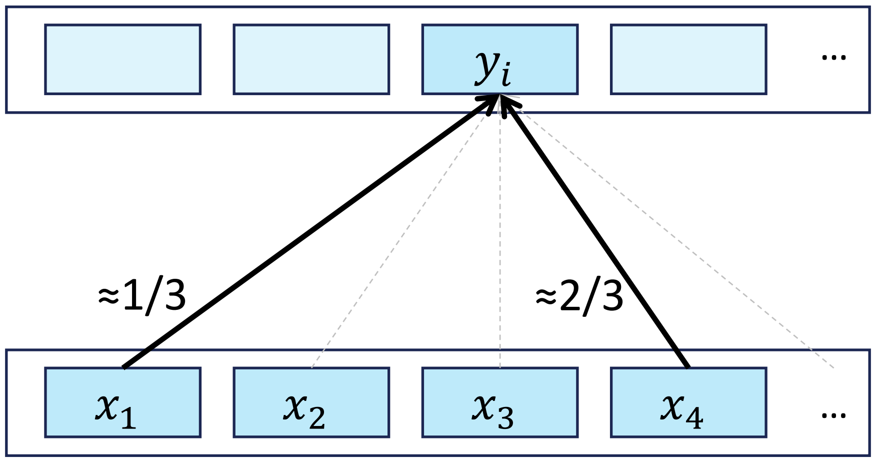
$$y_i = \sum_j \frac{\exp\left(\langle q^{(i)}, k^{(j)} \rangle\right)}{Z_i} v^{(j)}$$



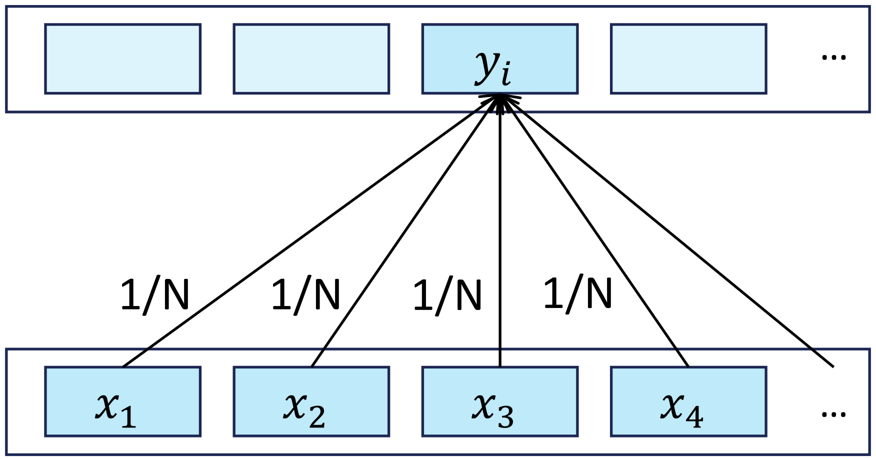
Outputs $y_1, \ldots, y_N$ can be produced in parallel

Prototypical attention patterns

Few keys well-align with query
("sparse attention")

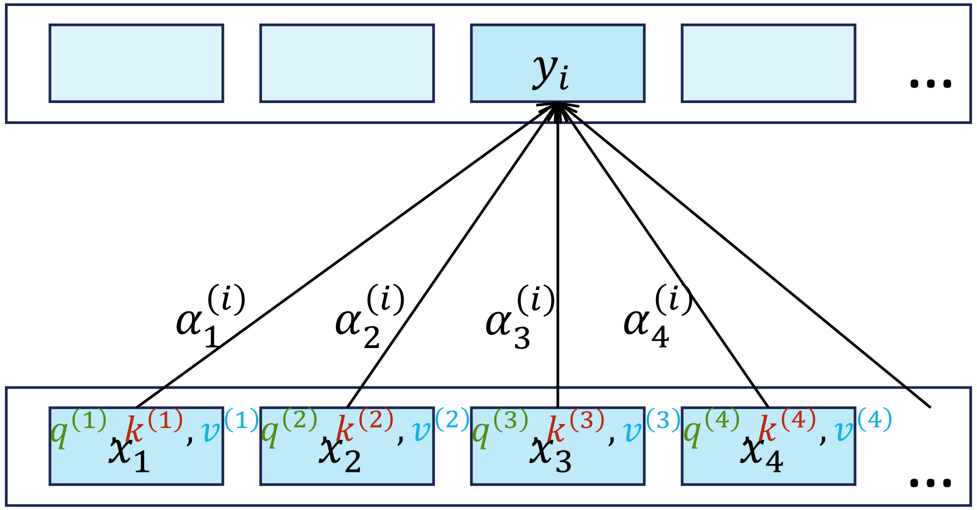


All keys equally aligned with query
("uniform averaging")



Attention pattern entirely determined by token embeddings (query/key vectors)
(… and tokens' positions via "positional embeddings")

Comparison to feedforward neural networks



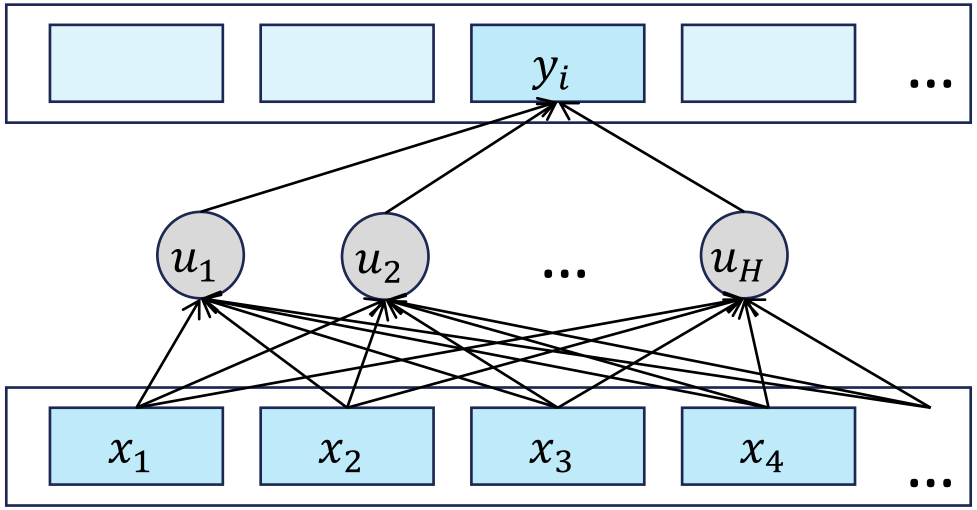
Self-attention head

Shared parameterized mapping

$$x_i \mapsto (q^{(i)}, k^{(i)}, v^{(i)})$$

Weights $\alpha_j^{(i)}$ determined via softmax

Universal approximation if embedding dimension $D \to \infty$



Feedforward neural network

Each "weight" is a separate parameter

$$y_i = \sum_{j=1}^{H} A_{i,j} \sigma\left(\sum_{k=1}^{N} W_{j,k} x_k\right)$$

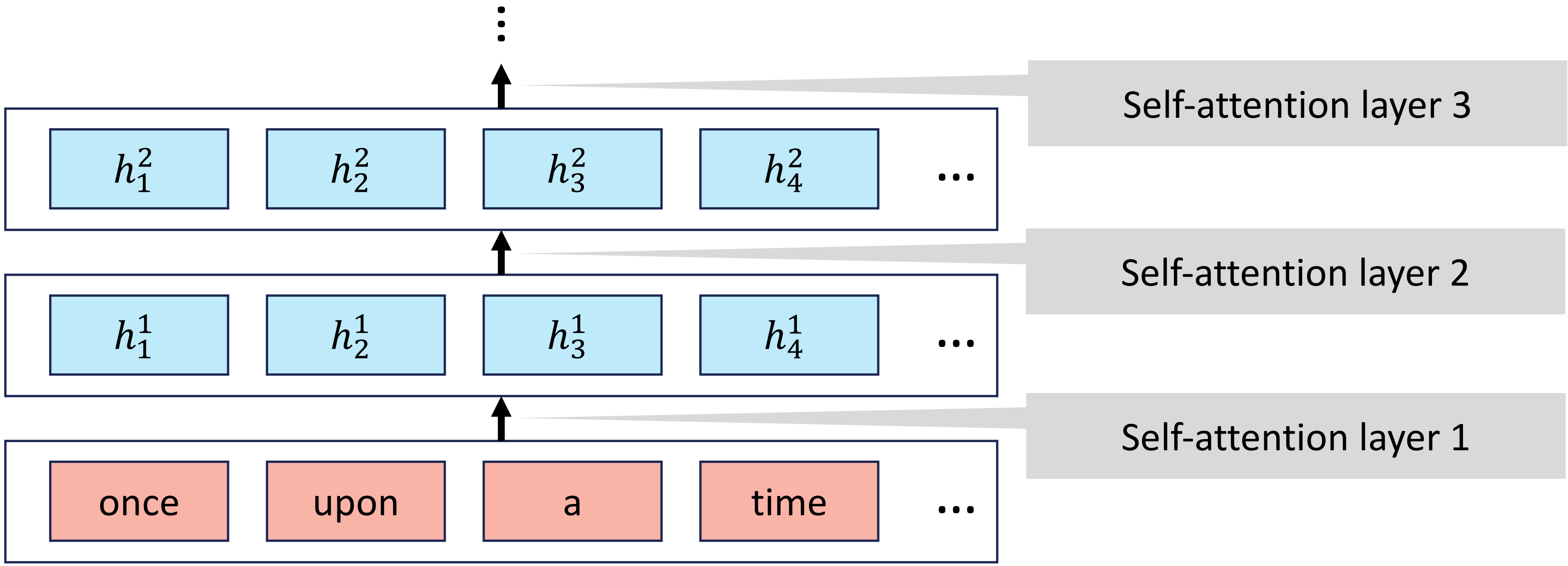
Universal Approximation Bounds for Superpositions of a Sigmoidal Function

Andrew R. Barron, *Member, IEEE* (if width $H \to \infty$)

Transformers as compositions

<u>Transformers</u>: compositions of self-attention layers

(layer = one self-attention head, or sum of several self-attention heads)

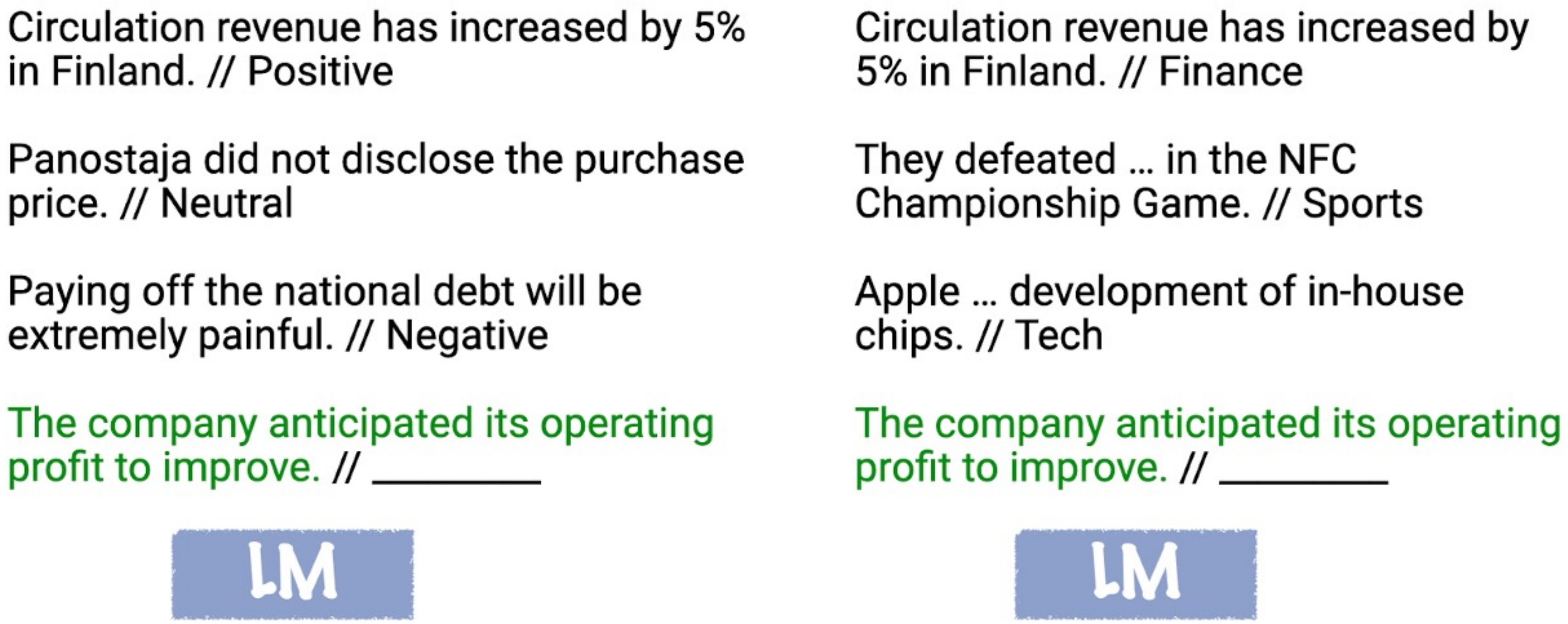


Why are multiple layers necessary?

1. Role of depth for in-context learning

In-context learning

[Brown et al, 2020]

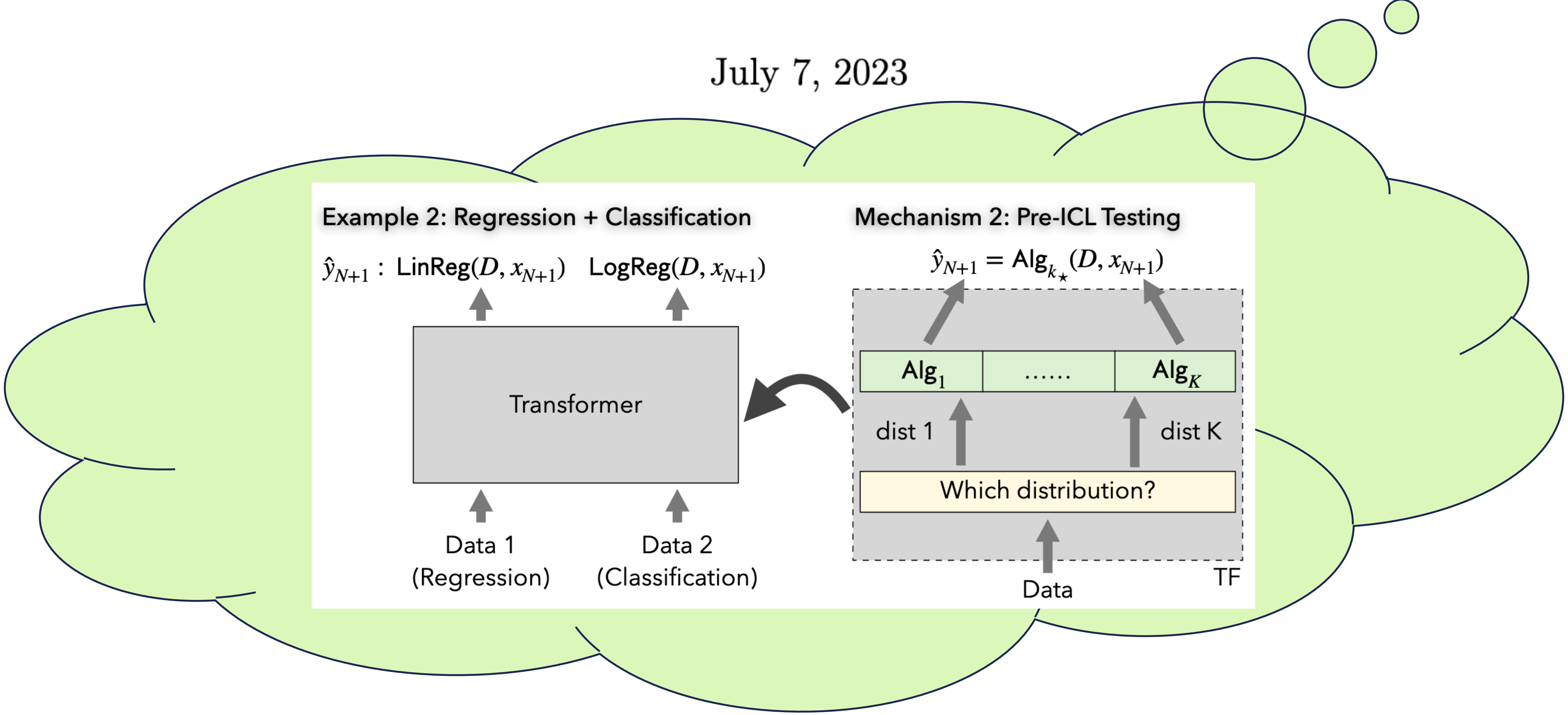


[Figure from Xie and Min, 2022]

Transformers as Statisticians: Provable In-Context Learning with In-Context Algorithm Selection

Yu Bai[*§] Fan Chen[†§] Huan Wang[*] Caiming Xiong[*] Song Mei[‡§]

July 7, 2023



Basic mechanism for in-context learning

[Anthropic: Elhage et al, 2021; Olsson et al, 2022]

Prompt (after tokenization):

[Mr] [and] [Mrs] [Durs] [ley] [,] [of] [number] [four] [,] [Pri] [vet] [Drive] [,] [were] [proud] [to] [say] [that] [they] [were] [perfectly] [normal] [,] [thank] [you] [very] [much] [.] [They] [were] [the] [last] [people] [you] ['d] [expect] [to] [be] [involved] [in] [anything] [strange] [or] [mysterious] [,] [because] [they] [just] [didn] ['t] [hold] [with] [such] [nonsense] [.] [Mr] [Durs]

Basic mechanism for in-context learning

[Anthropic: Elhage et al, 2021; Olsson et al, 2022]

<u>Prompt (after tokenization)</u>:

[Mr] [and] [Mrs] [Durs] [ley] [,] [of] [number] [four] [,] [Pri] [vet] [Drive] [,] [were] [proud] [to] [say] [that] [they] [were] [perfectly] [normal] [,] [thank] [you] [very] [much] [.] [They] [were] [the] [last] [people] [you] ['d] [expect] [to] [be] [involved] [in] [anything] [strange] [or] [mysterious] [,] [because] [they] [just] [didn] ['t] [hold] [with] [such] [nonsense] [.] [Mr] [Durs]

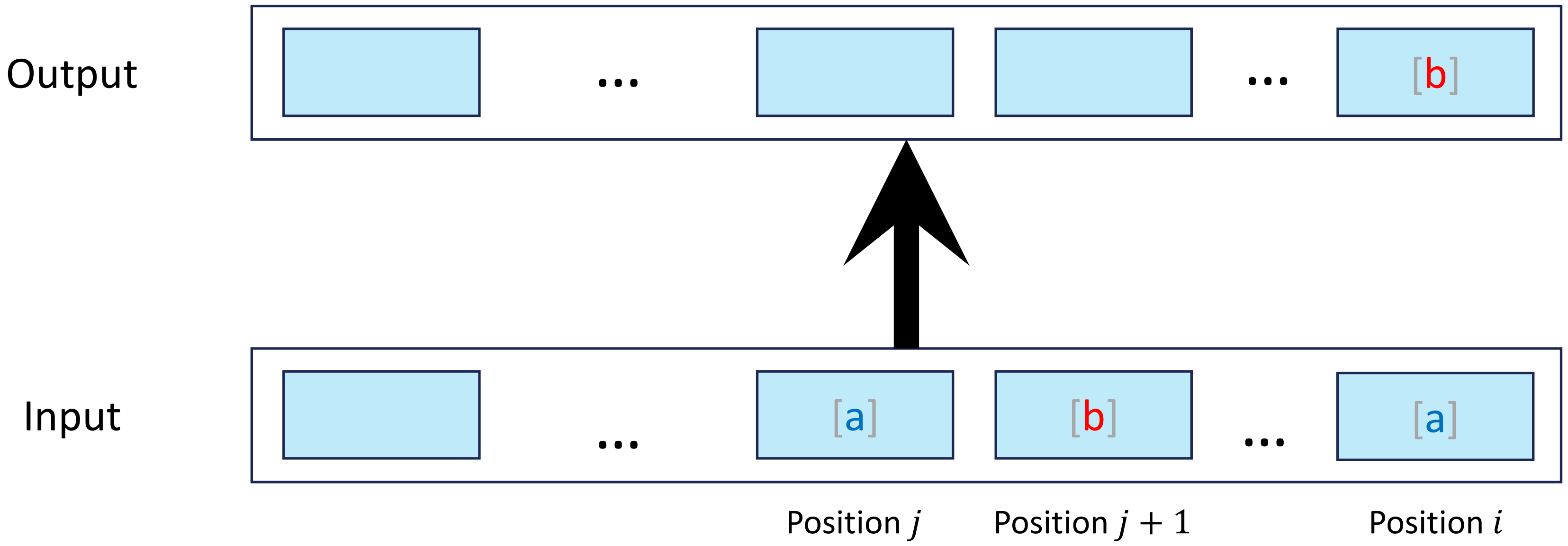
b a c b … c a b d b a

Induction heads abstraction

[Anthropic: Elhage et al, 2021; Olsson et al, 2022]

Induction head: abstraction of a salient sub-circuit found in LLMs

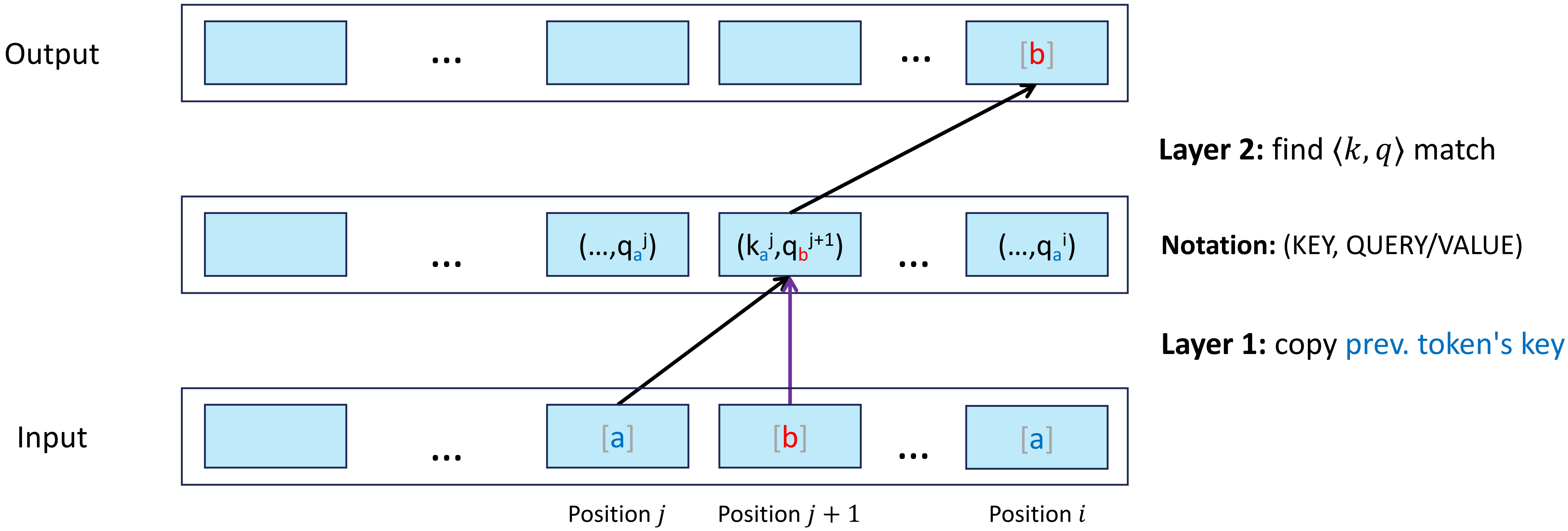
- i^{th} output: Find last position $j < i$ where x_i occurs, output x_{j+1}



Induction heads implementation

Composition of two "small" self-attention heads [e.g., Bietti et al, 2023]

Token embedding dimension
$O(\log N)$ suffices



Necessity of two layers

Theorem [SHT'24b]:
Single self-attention head* (one layer) with embedding dimension D cannot implement induction head for length N sequences unless

$$D \geq \tilde{\Omega}(N)$$

Exponentially larger than what's sufficient with *two* layers

Corroborates prior empirical findings
[Elhage et al, 2021; Olsson et al, 2022; Bietti et al, 2023]

*Using polylog N bits of numerical precision, even for $O(1)$-size input alphabet, allowing arbitrary size MLPs

Rudimentary in-context learning

Prompt: whale 1 dog 1 frog 0 shark 0 bat 1 owl 0 wolf

"Nearest neighbor"-like in-context learning:
Word embeddings + induction head

(Layers before induction head: help with prompt formatting, perhaps?)

Circulation revenue has increased by 5% in Finland. // Positive

Panostaja did not disclose the purchase price. // Neutral

Paying off the national debt will be extremely painful. // Negative

The company anticipated its operating profit to improve. // ________

Beyond two layers?

Multi-step reasoning problem [Peng, Narayanan, Papadimitriou, 2024]:

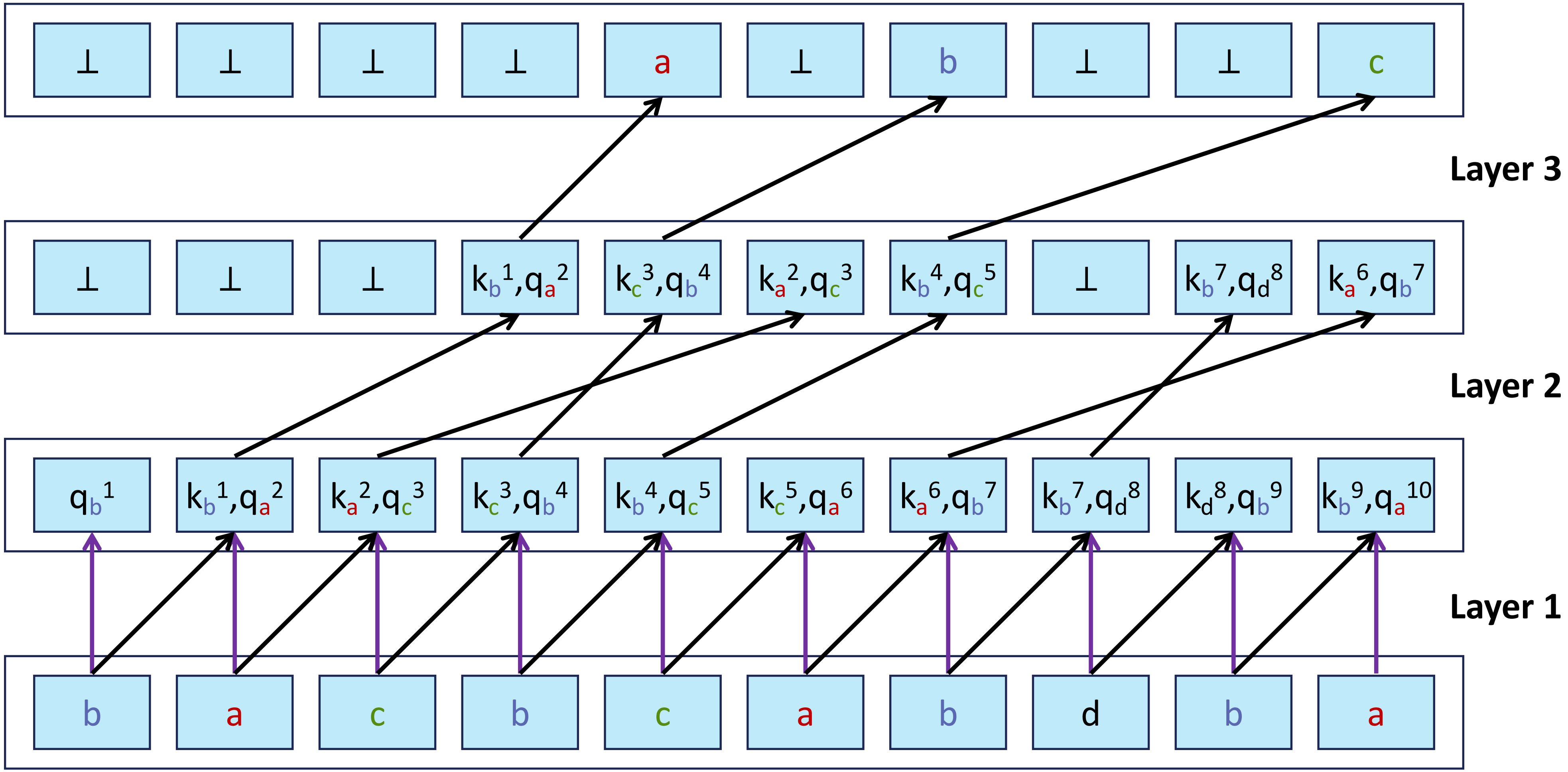
Prompt: "Jane is a teacher. Helen is a doctor. [...] The mother of John is Helen. The mother of Charlotte is Eve. [...] What's the profession of John's mother?"

Answer: doctor

b a c b c a b d b a

2-hop induction head

2-hop induction head



Key idea: Layers 1 & 2 solve 1-hop *for all positions in parallel*

k-hop induction head

Theorem [SHT'24a]:
There is a $2 + \lceil \log_2 k \rceil$ layer transformer* that implements k-hop …

Main idea: Each additional layer *doubles* the "reach"

Empirical surprise: SGD finds this $\Theta(\log k)$-layer solution!

… & under plausible conjecture about *massively parallel computation*, $\Omega(\log k)$ layers are necessary (under similar size constraints)

*Using one self-attention head per layer, $\log N$ dimensional embeddings, $\log N$ bits of numerical precision, assuming $\mathrm{poly}(N)$-size input alphabet

2. Transformers & Massively Parallel Computation

Massively Parallel Computation (MPC)

MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

Google, Inc.

hadoop

A Model of Computation for MapReduce

Howard Karloff* Siddharth Suri† Sergei Vassilvitskii‡

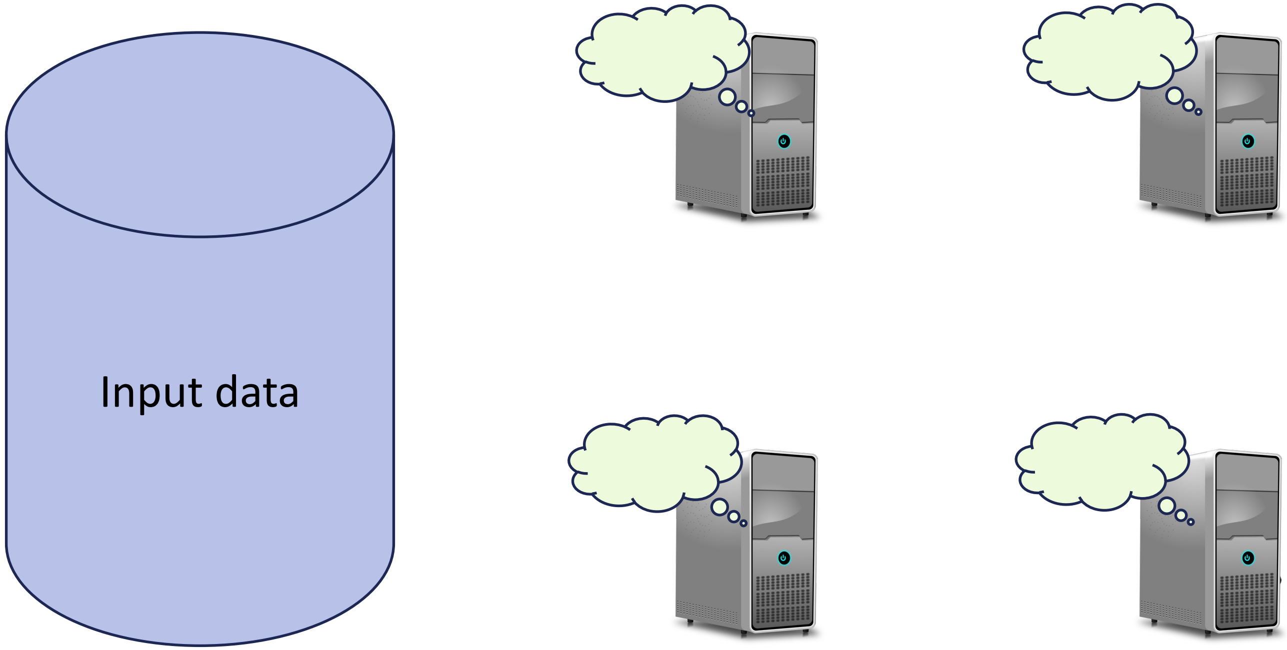
[Karloff et al, 2010; Goodrich et al, 2011; Beame et al, 2013; Andoni et al, 2014]

MPC model of computation

Input data size: N words $[N \leq M \times S]$

Number of machines: M

Memory size per machine: S words $[S = \Theta(N^{\delta})$ for small $\delta \in (0,1)]$



Communication constraints per "shuffle" round:

Each machine sends $\leq S$ words

Each machine receives $\leq S$ words

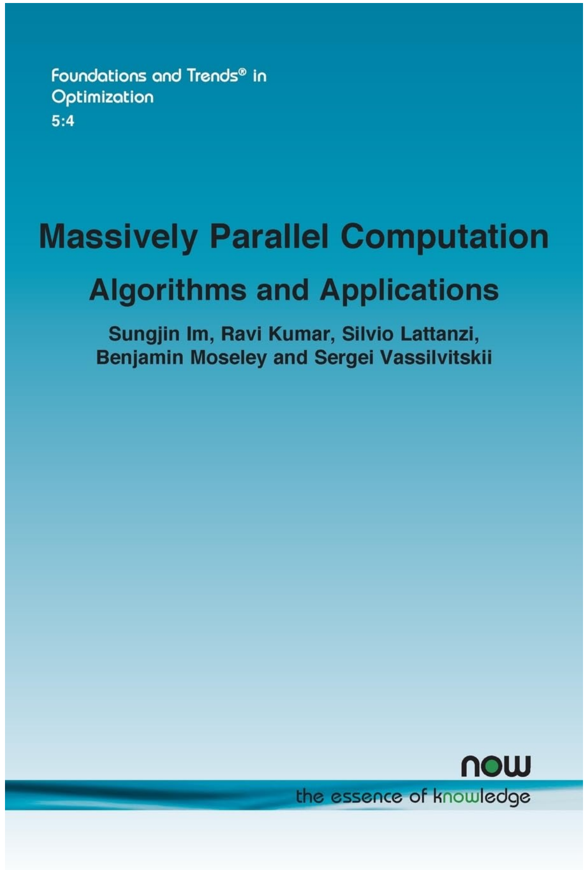
Between "shuffle" rounds:

Each machine performs arbitrary computation on local memory

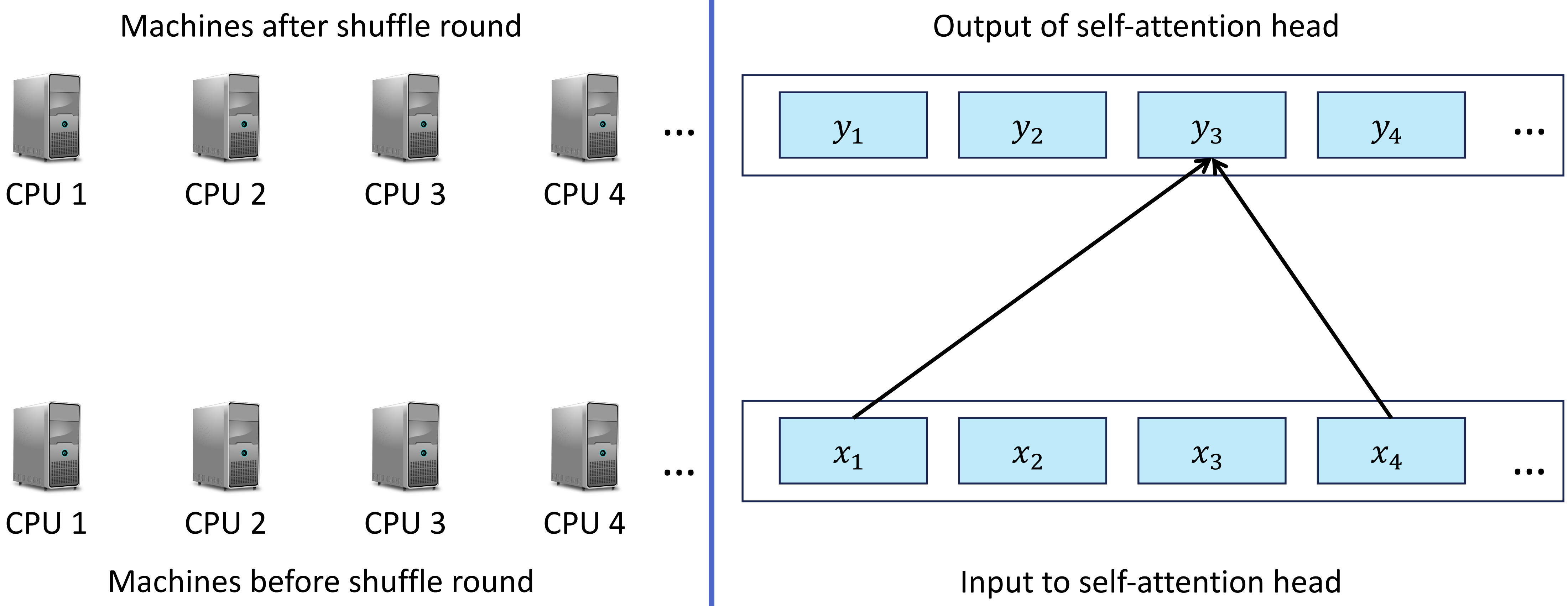
<u>Main question</u>: How many rounds R are needed?

MPC algorithms for many tasks

- Broadcast $R = O(1)$
- Sorting $R = O(1)$
- Prefix sum $R = O(1)$
- Problems on sparse graphs
  [Andoni et al, 2018, Behnezhad et al, 2019, …]
  - Connected components $R = \log(\text{diameter})$
  - Minimum spanning forest $R = \log(\text{diameter})$
  - …
- …
- Open question: $R = o(\log N)$ round algorithm for connectivity?



Simulating MPC shuffle round with self-attention



MPC algorithms ⇄ transformers

Theorem [SHT'24a; Sanford et al, 2024] (informal version):

can be simulated by

$\Theta(R)$-round MPC algorithm with local memory $\approx \Theta(N^{\delta})$

$\Theta(R)$-layer transformer with embedding dimension $\approx \Theta(N^{\delta})$

can be simulated by*

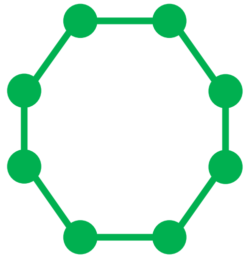
Easy for MPC ⇒ Easy for transformer

Hard for MPC ⇒ Hard for transformer

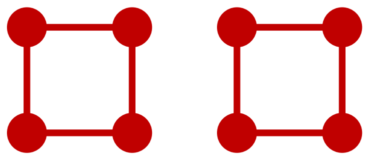
*With additional $\Theta(N^2)$ machines

What is hard for MPC?

1-vs-2 cycle problem: Given graph G that is promised to be either cycle on N vertices or union of two cycles on $N/2$ vertices each,



versus



decide if G is connected.

1-vs-2 cycle hypothesis (informal version):

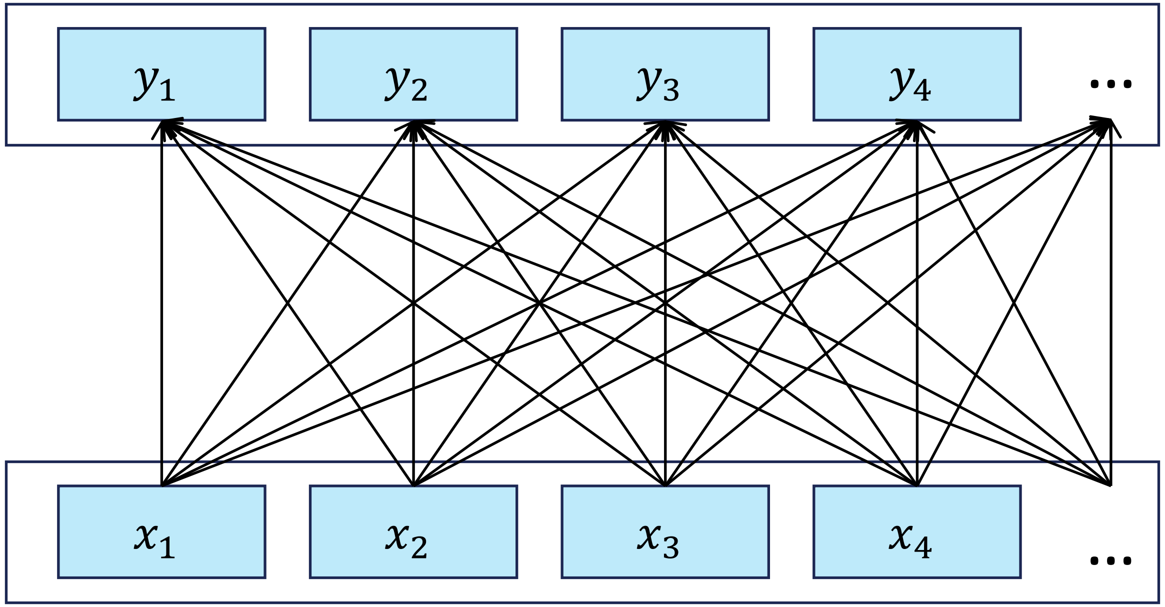
Every "efficient" MPC algorithm must use $R = \Omega(\log N)$ rounds

Theorem [SHT'24a]: 1-vs-2 cycle hypothesis implies necessity of $\Omega(\log k)$ layers in transformers for k-hop

3. Limitations of sequential neural architectures

Computational cost of transformers

For self-attention, quadratic computation appears to be inherent



Are there sub-quadratic alternatives to self-attention?

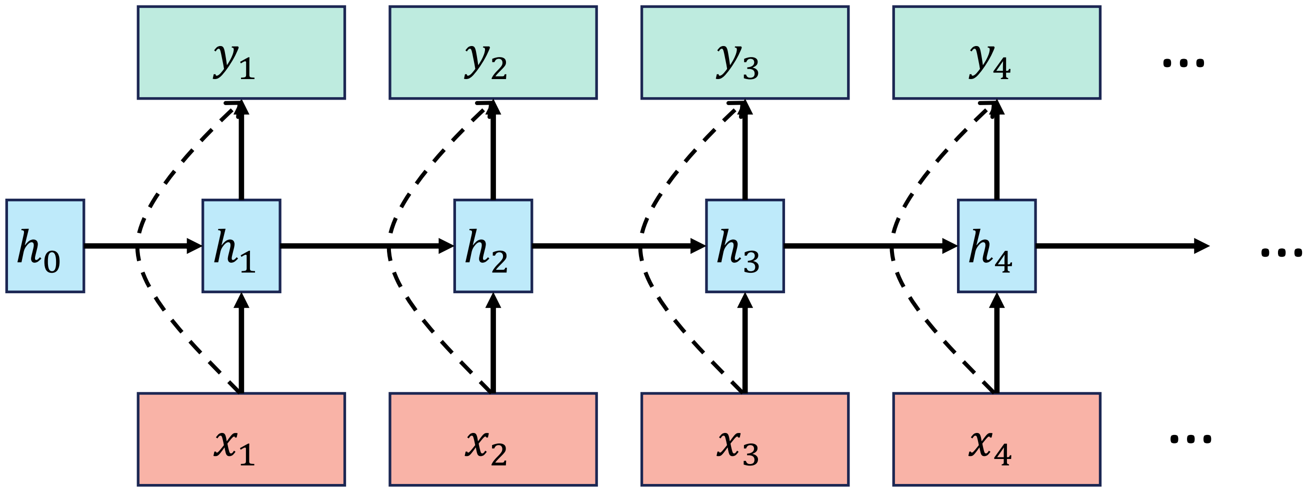
Sequential neural architectures

<u>Recurrent neural network (RNN)</u>:

Initialize "hidden state" h_0

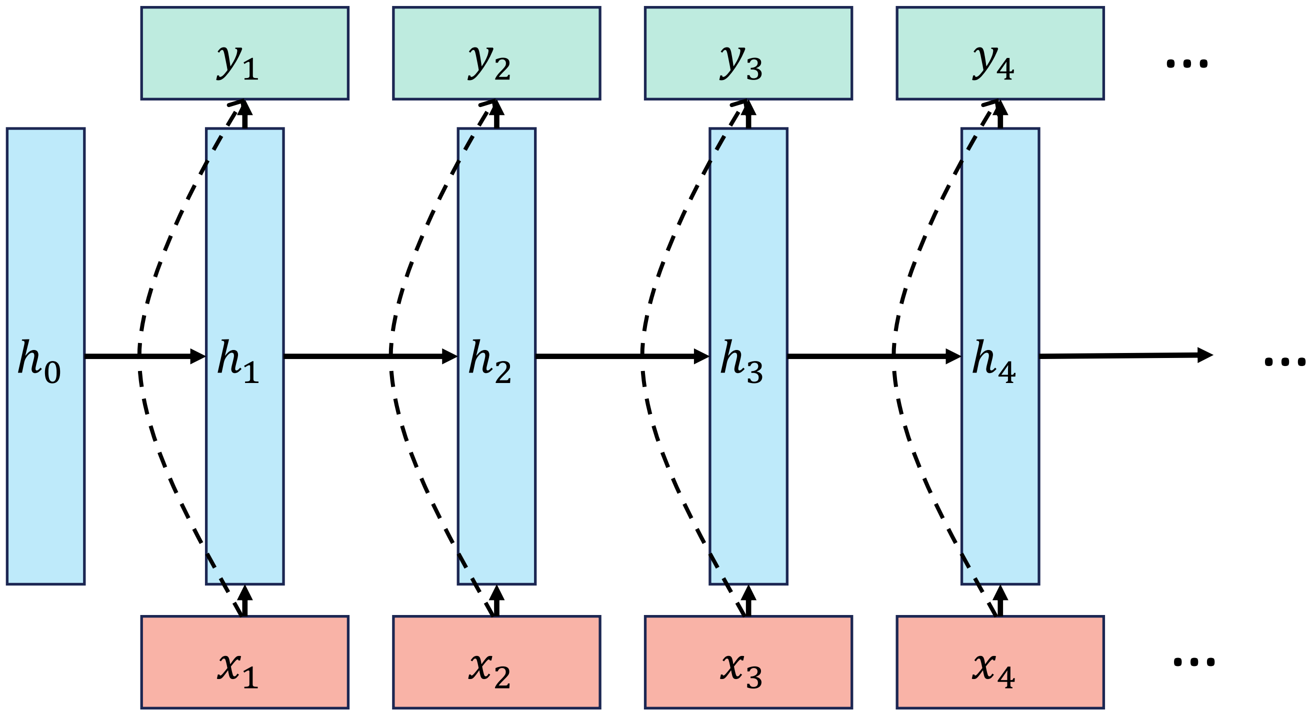
For $t = 1,2, \dots, N$:

$$h_t = \mathrm{update}_t(h_{t-1}, x_t)$$
$$y_t = \mathrm{output}_t(h_t, x_t)$$



Memory bottlenecks in RNNs

Theorem [SHT'23]:
Any RNN that computes N^{th} output of (1-hop) induction head must use a $\Omega(N)$-bit hidden state



Further limitations for sequential architectures

Consequences of (Assadi and N, 2021) [SHT'24a] (informal version):

For k-hop induction head, "sequential architectures" require

$$\text{"depth"} \geq k \quad \text{or} \quad \text{"size"} = \Omega(N/k^6)$$

(Applies to multi-layer RNNs, shallow TF with "chain-of-thought", …)

(Recall: For standard transformer, depth $= O(\log k)$, size $= O(\log N)$)

Aftermath and open problems

1. <u>Role of depth for in-context learning</u>
   - At least two layers are necessary for primitive underlying in-context learning
   - For k-fold compositions, $\log k$ layers sufficient (and probably necessary)
   - What are important function compositions in LLMs?
2. <u>Transformers & MPC</u>
   - Coarse reductions between transformers and MPC
   - How to characterize power of transformer "shuffle" operation?
3. <u>Limitations of sequential neural architectures</u>
   - How do we get around these limitations?

Thank you!