

A Mathematical Framework for Transformer Circuits

COMS6998: Theoretical Foundations of LLMs

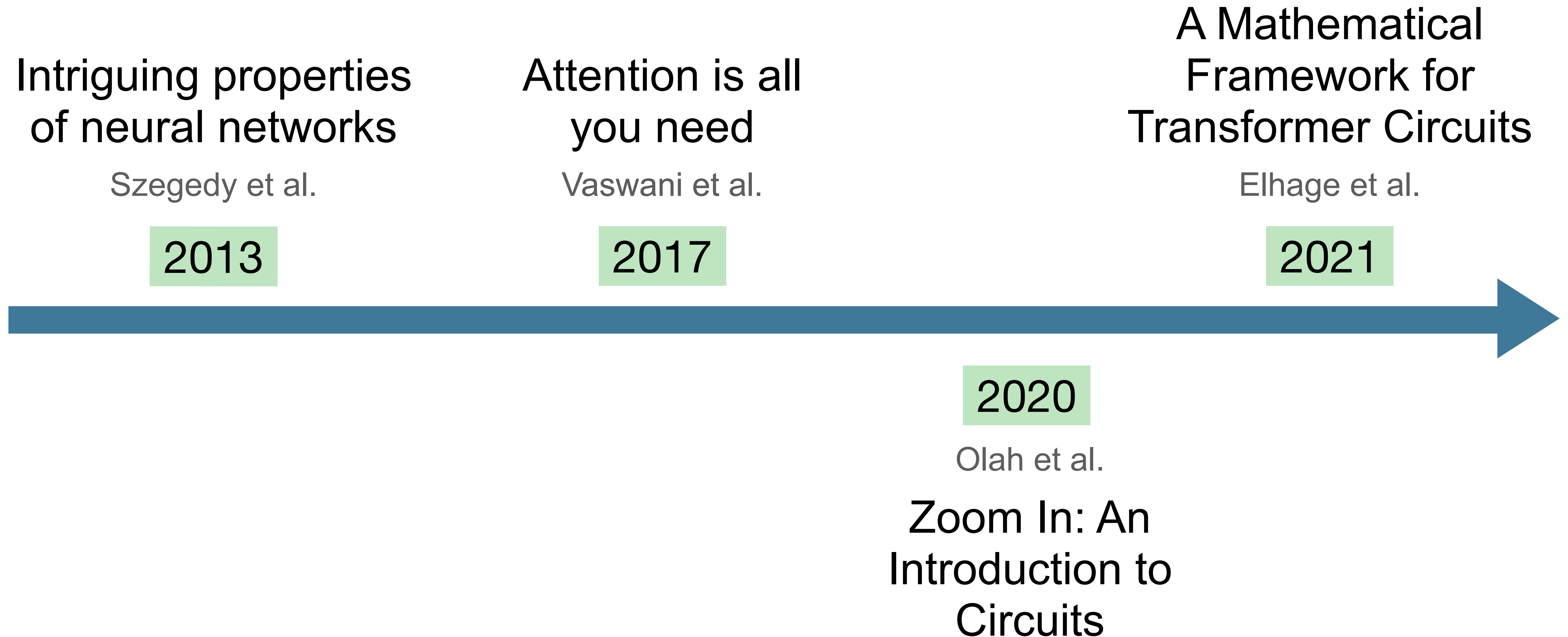
Presented by Nicolas Beltran and Sweta Karlekar

Overview

- Some historical background
- Reconceptualizing transformers
- Empirical results
- Current research directions

Historical Background

Historical Context



Intriguing properties of neural networks

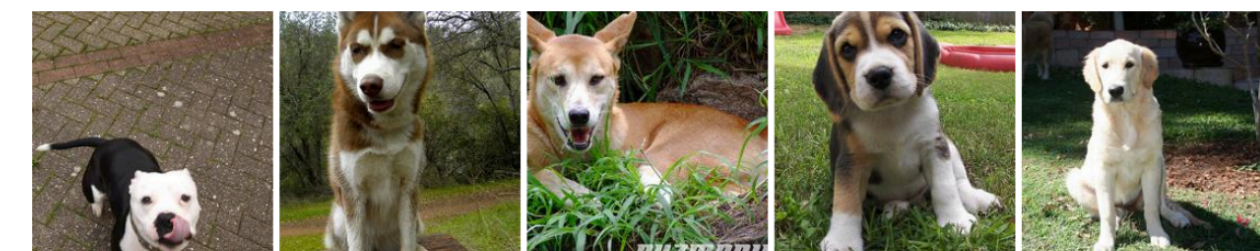
2013

Found that neurons and non-axis aligned directions **activated with specific semantic content.**

Highlighted the importance of investigating neural network internals to better understand their decision-making processes.

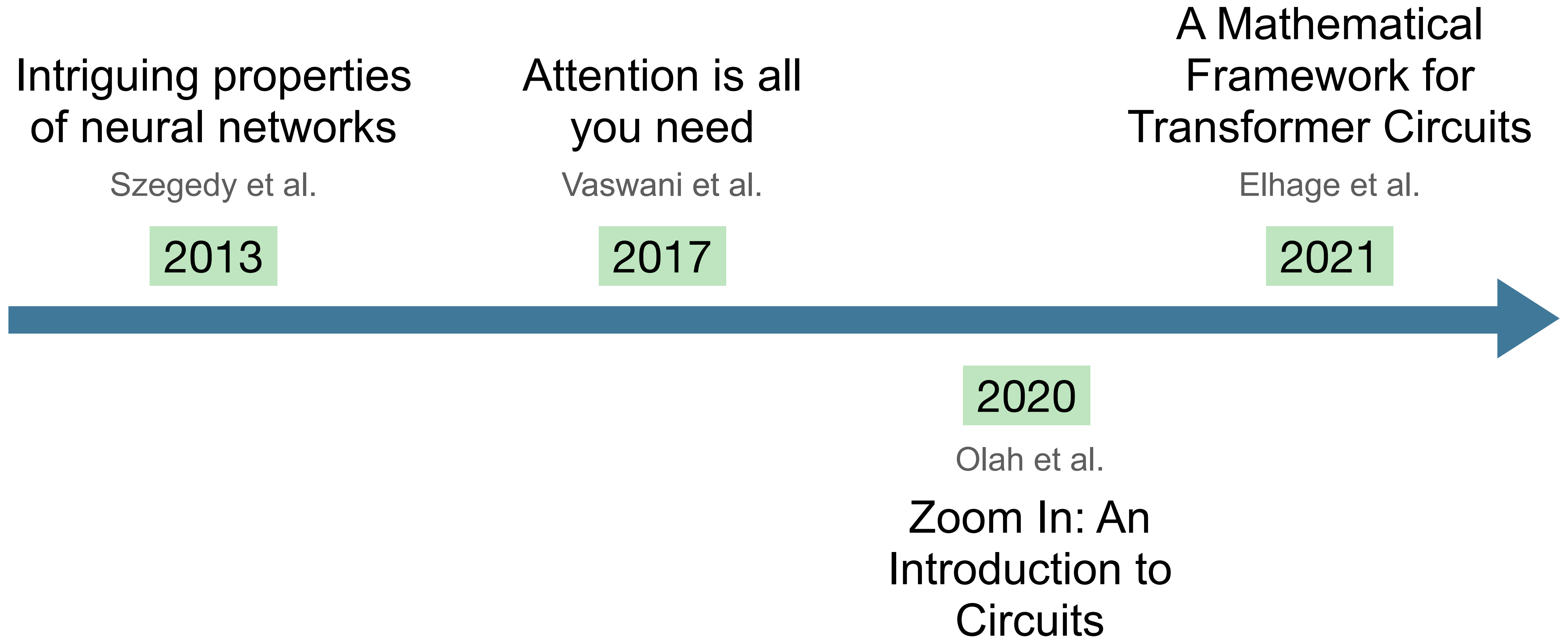


(a) Unit sensitive to white flowers.

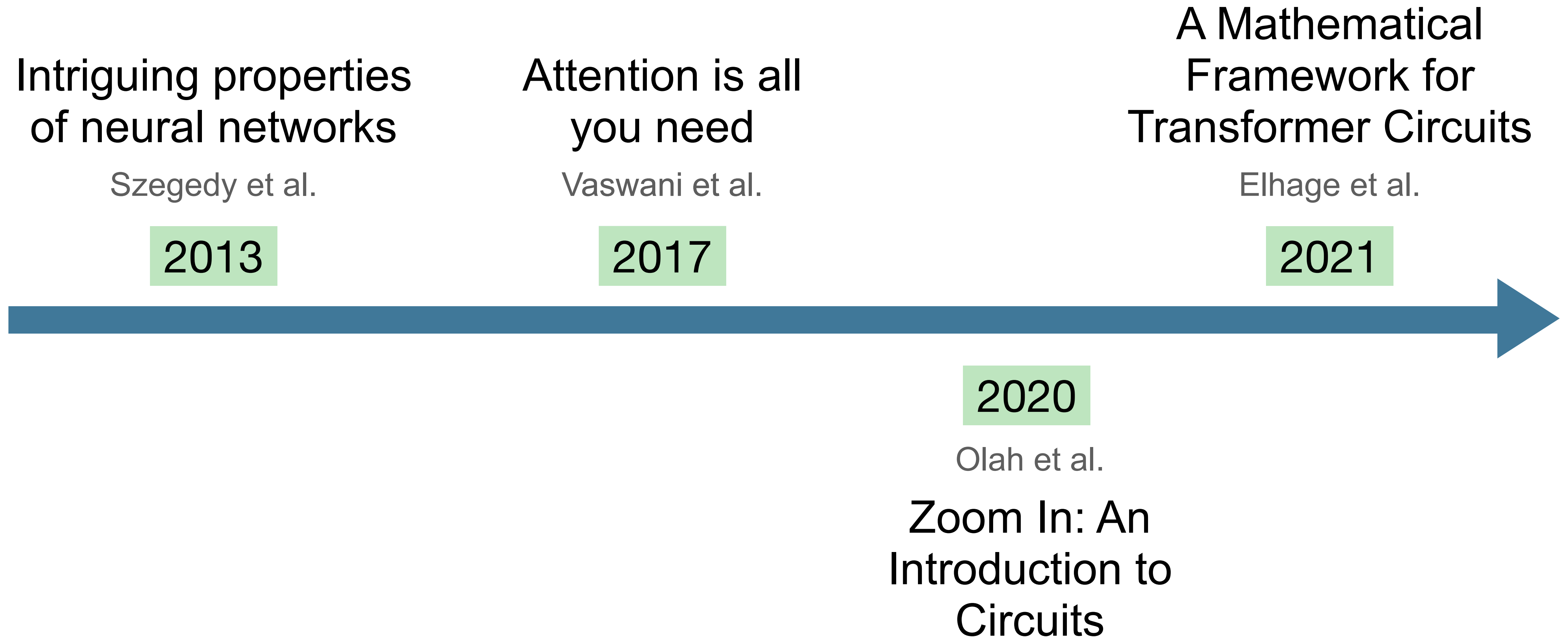


(b) Direction sensitive to white dogs.

Historical Context



Historical Context

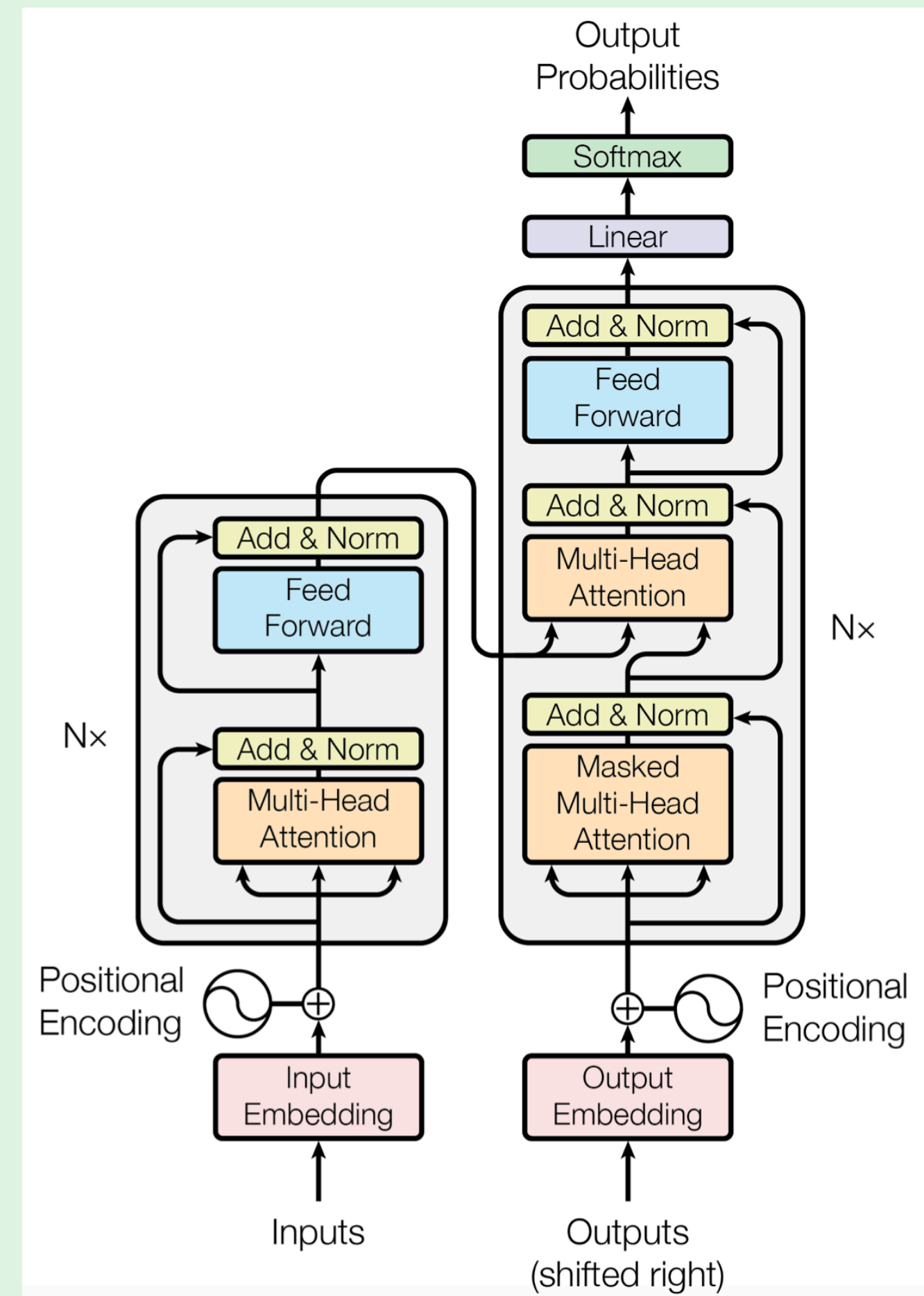


Attention is all you need

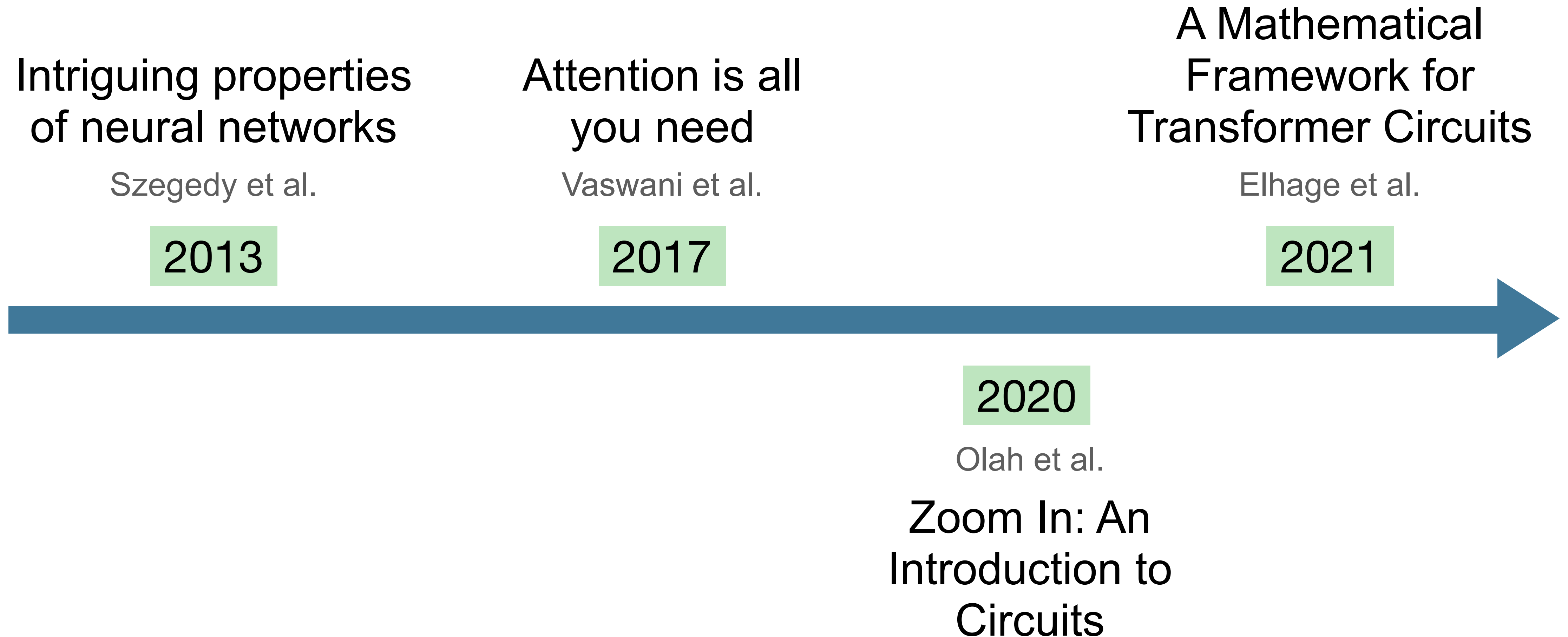
2017

Introduced the transformer architecture.

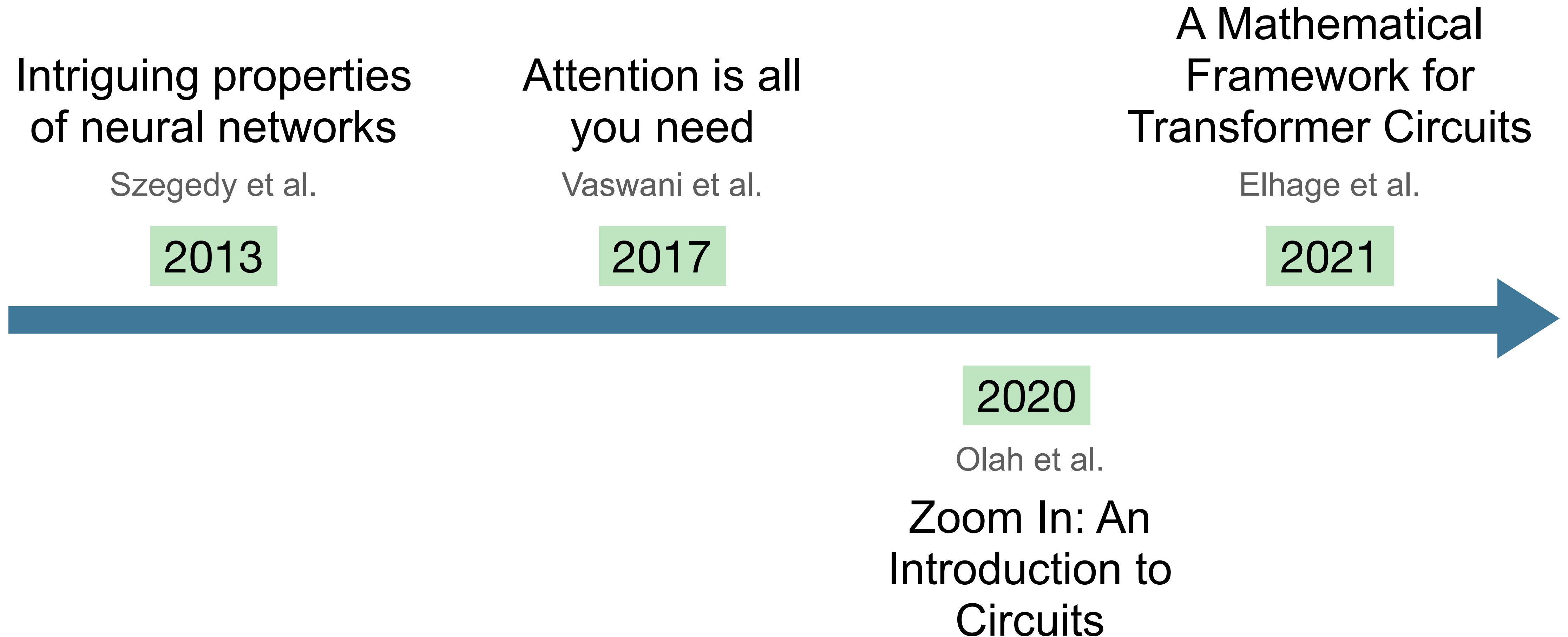
This became the basis for a number of advanced models like BERT, GPT, etc.



Historical Context



Historical Context



Zoom in: An introduction to circuits

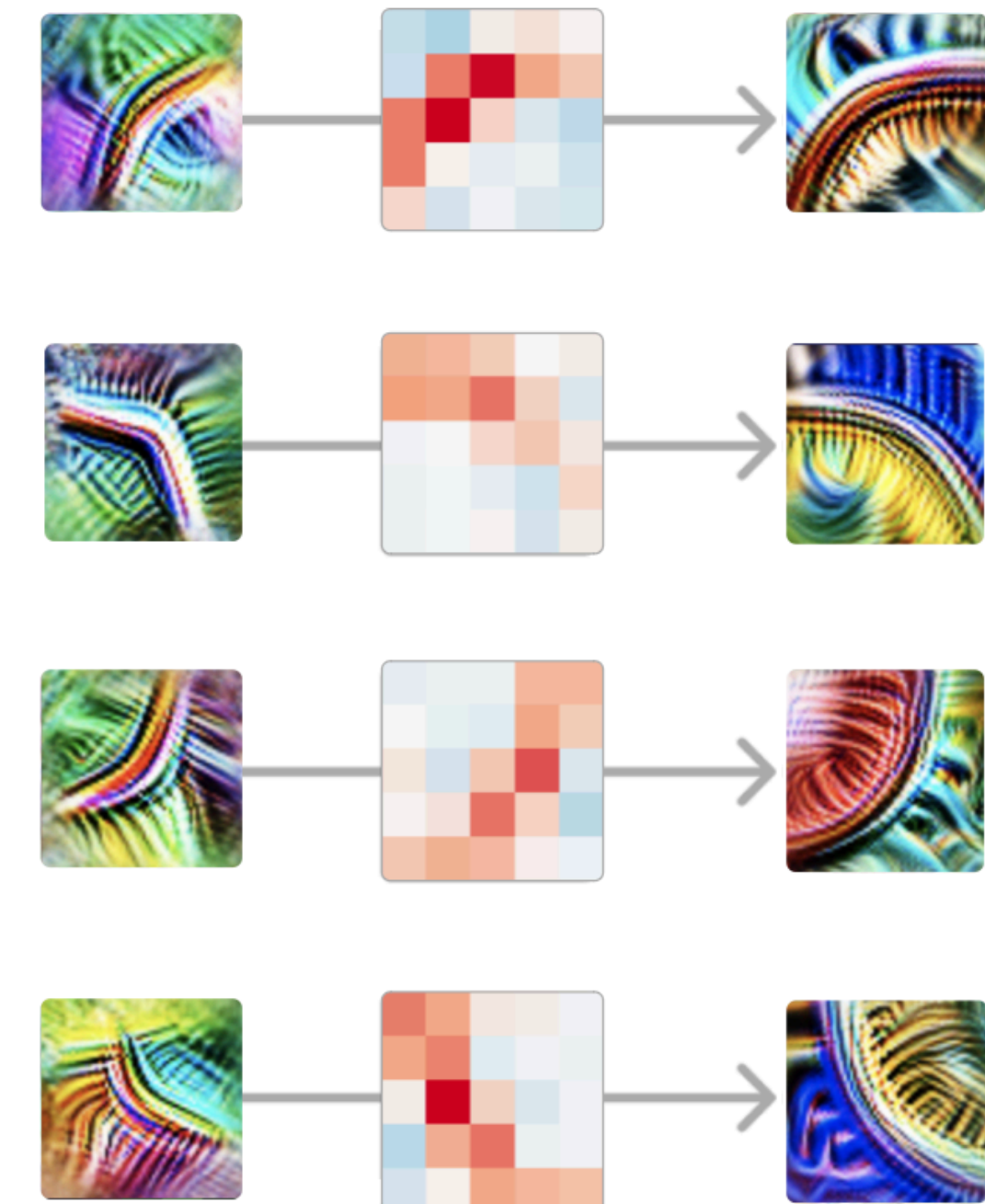
2020

Defined the circuit-based interpretability research program with an emphasis on convolutional architectures.

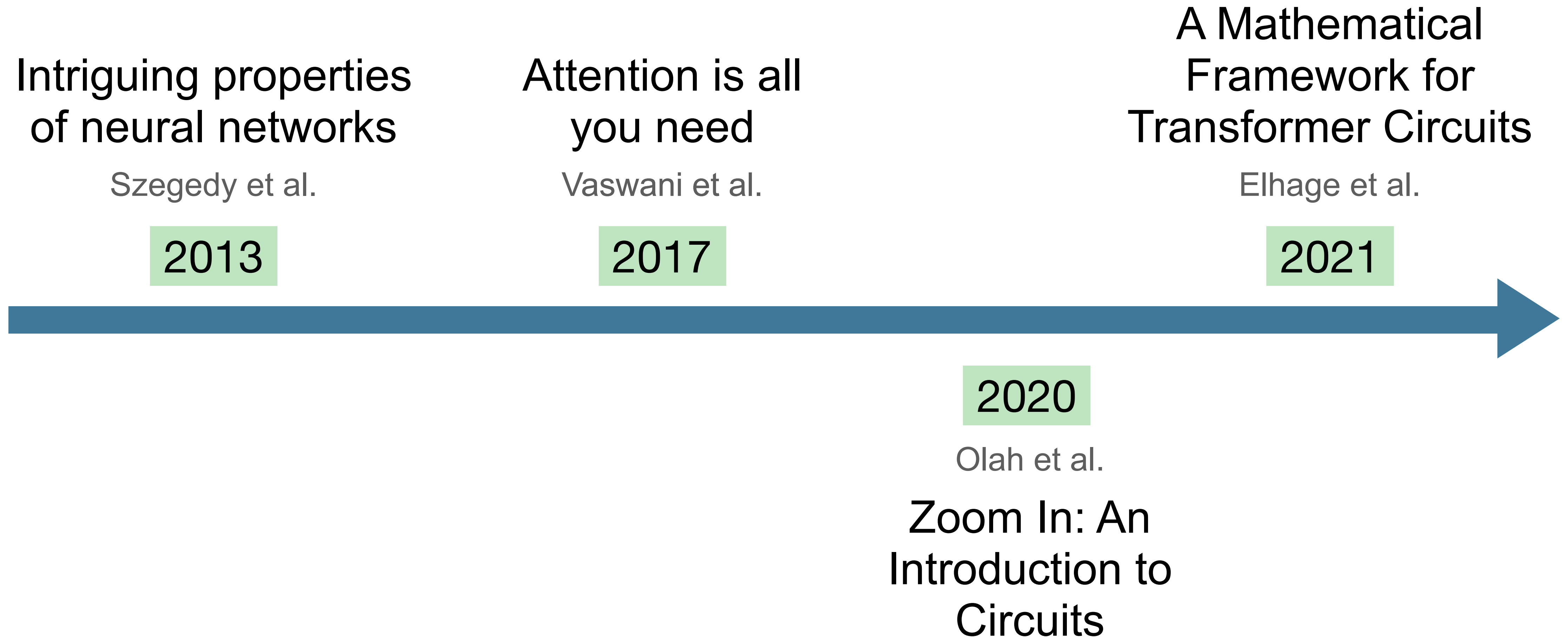
It hypothesizes that:

- a) **Features** are the fundamental units of analysis in NNs.
- b) Features are interconnected by weights forming **circuits**.
- c) Different models have similar features and circuits.

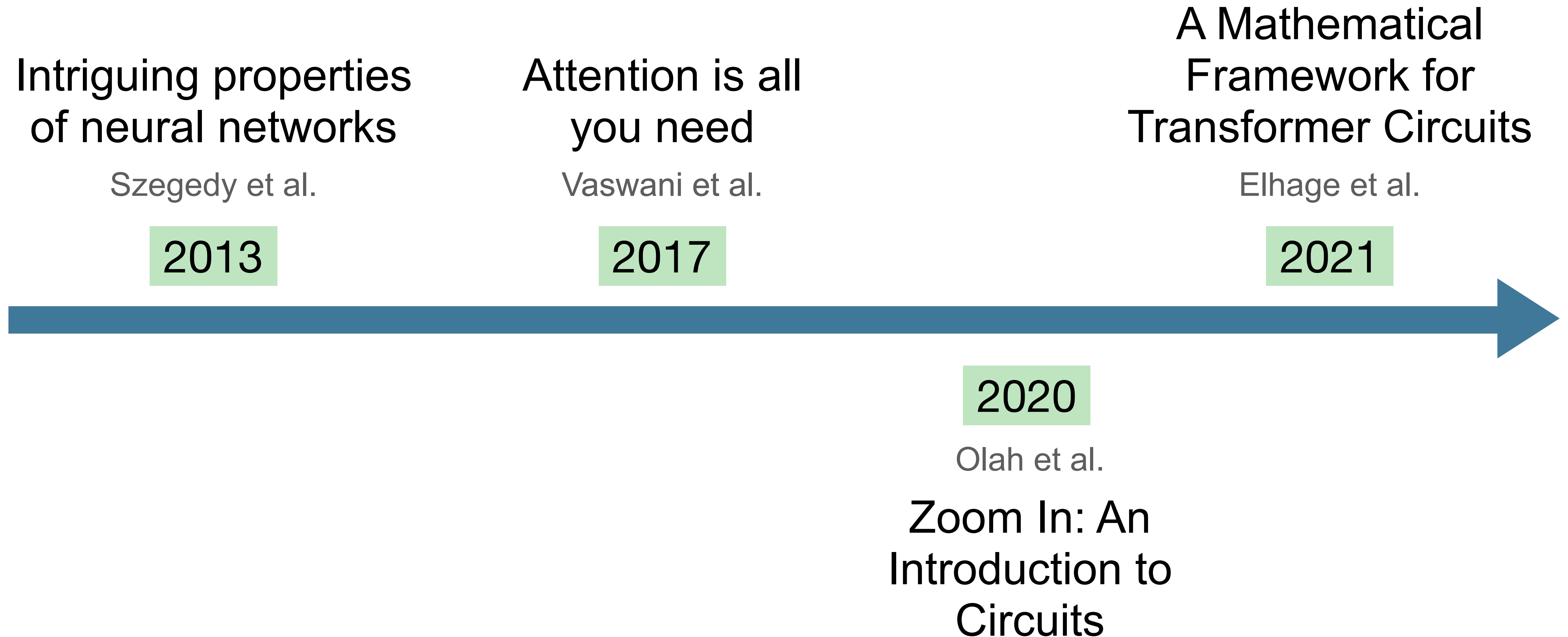
Curve detectors are **excited** by earlier detectors in **similar orientations**...



Historical Context



Historical Context



Historical Context

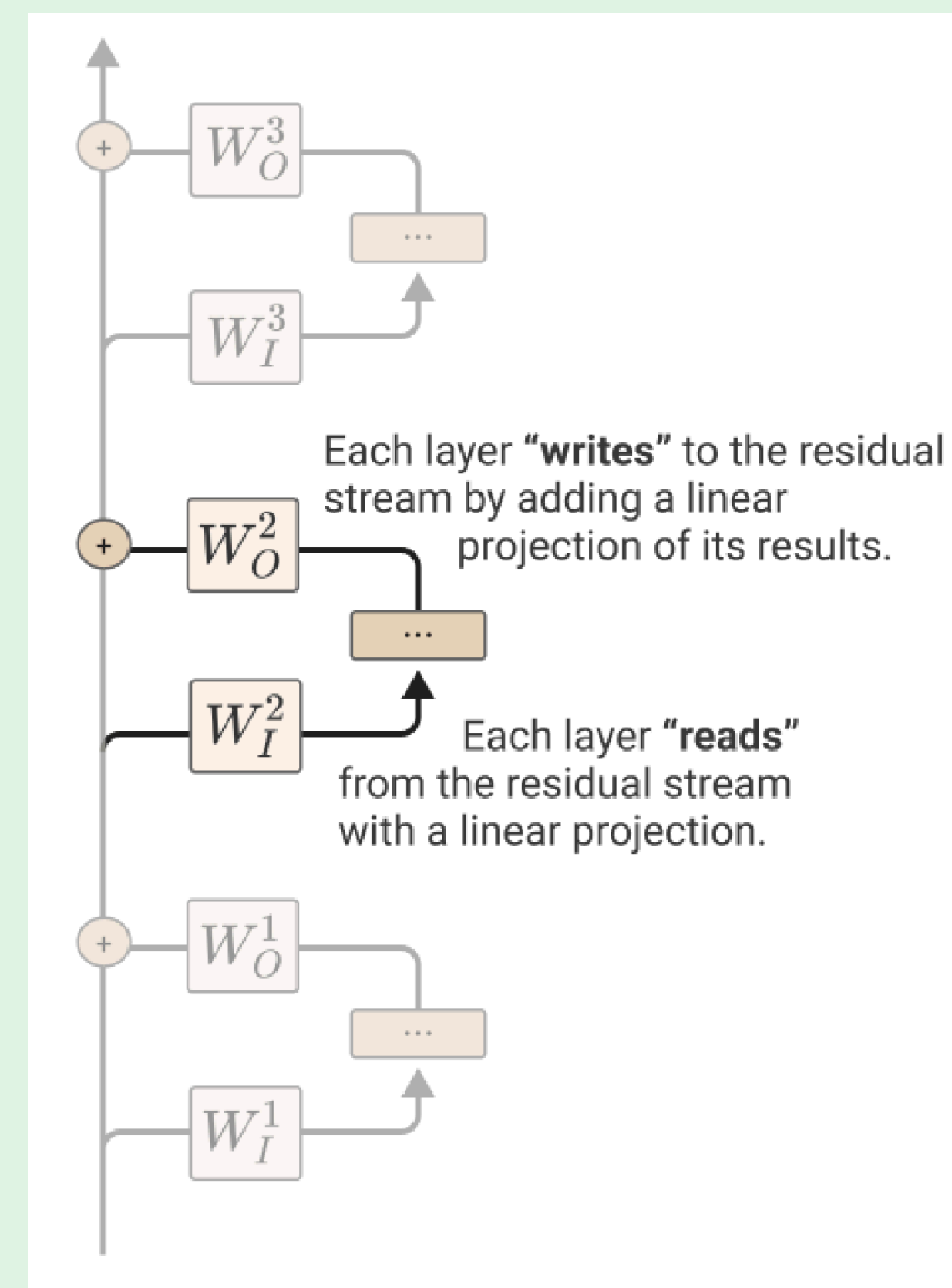
A mathematical framework for transformer circuits

2021

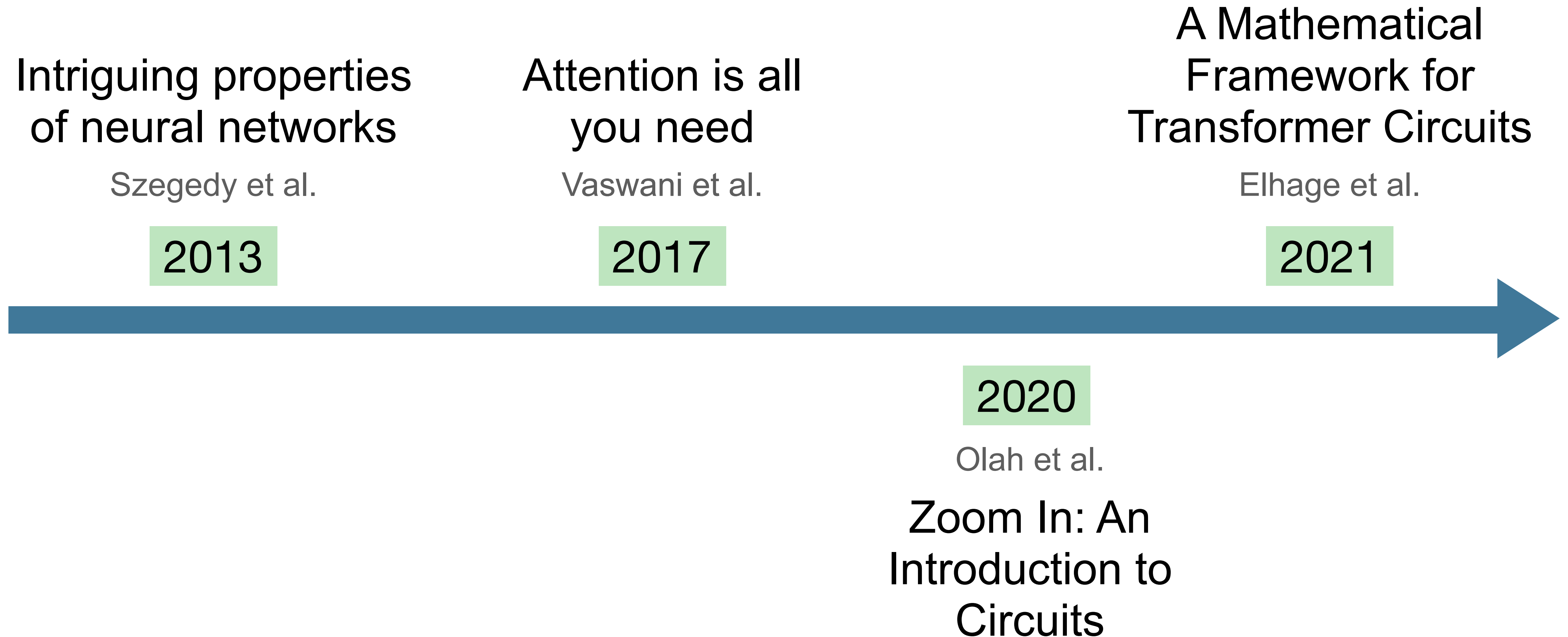
Aimed to find algorithmic motifs and a unified frameworks through which to analyze transformers.

Introduced a lot of the common vocabulary in mechanistic interpretability.

First found many famous and non-trivial behaviors such as induction heads, skip-trigrams, and copying.

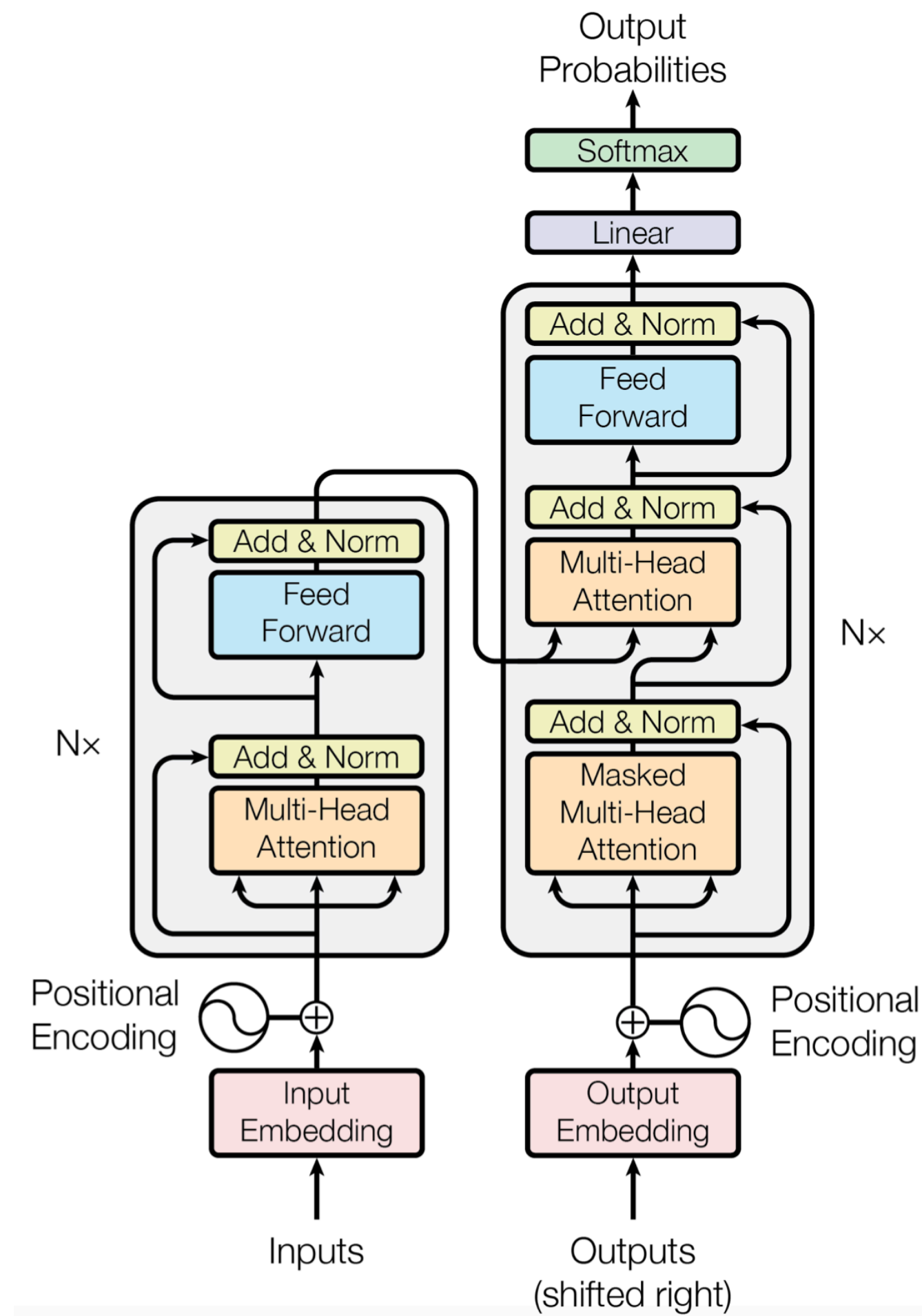


Historical Context

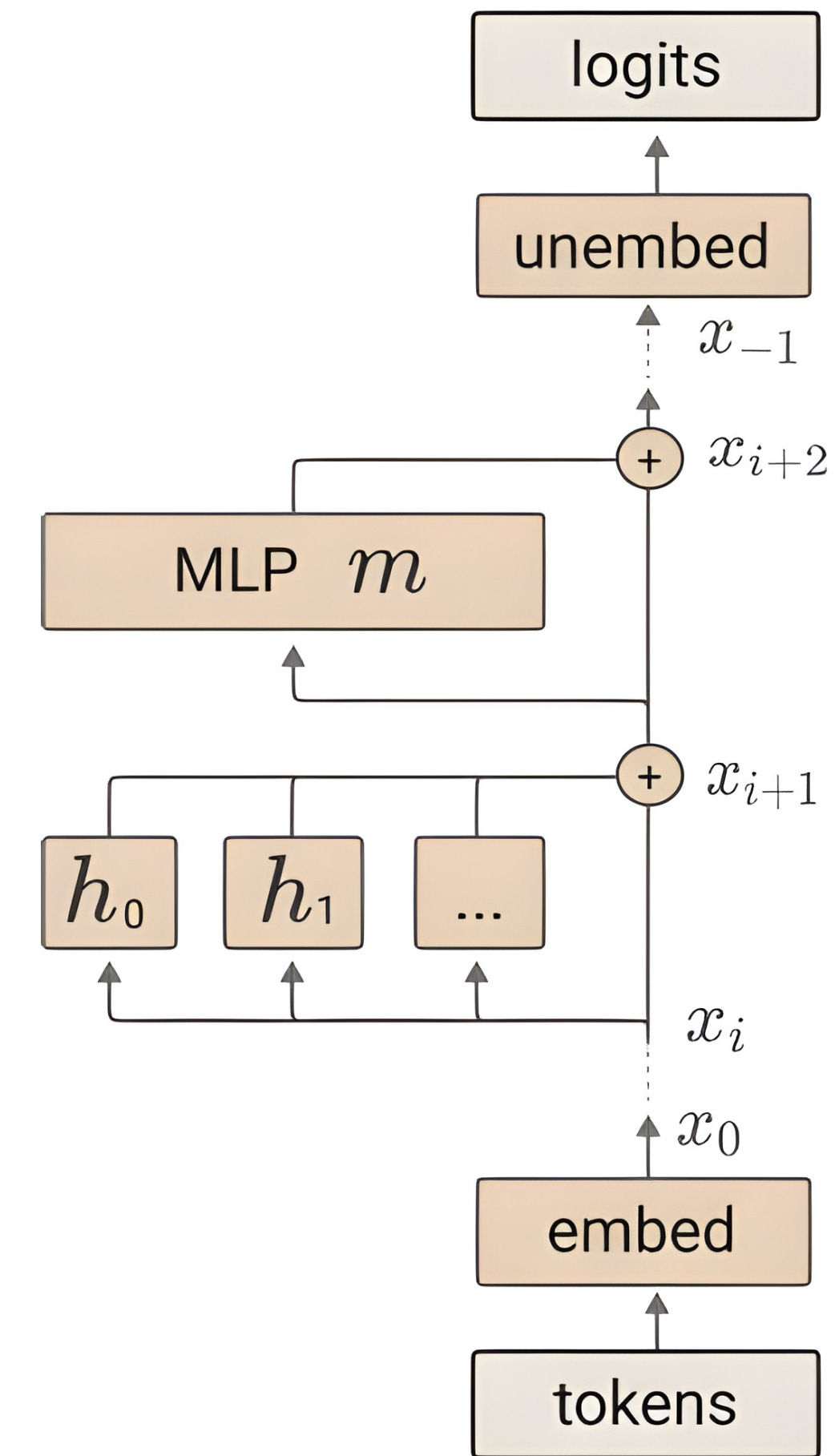


Reconceptualizing transformers

A new way of looking at transformers

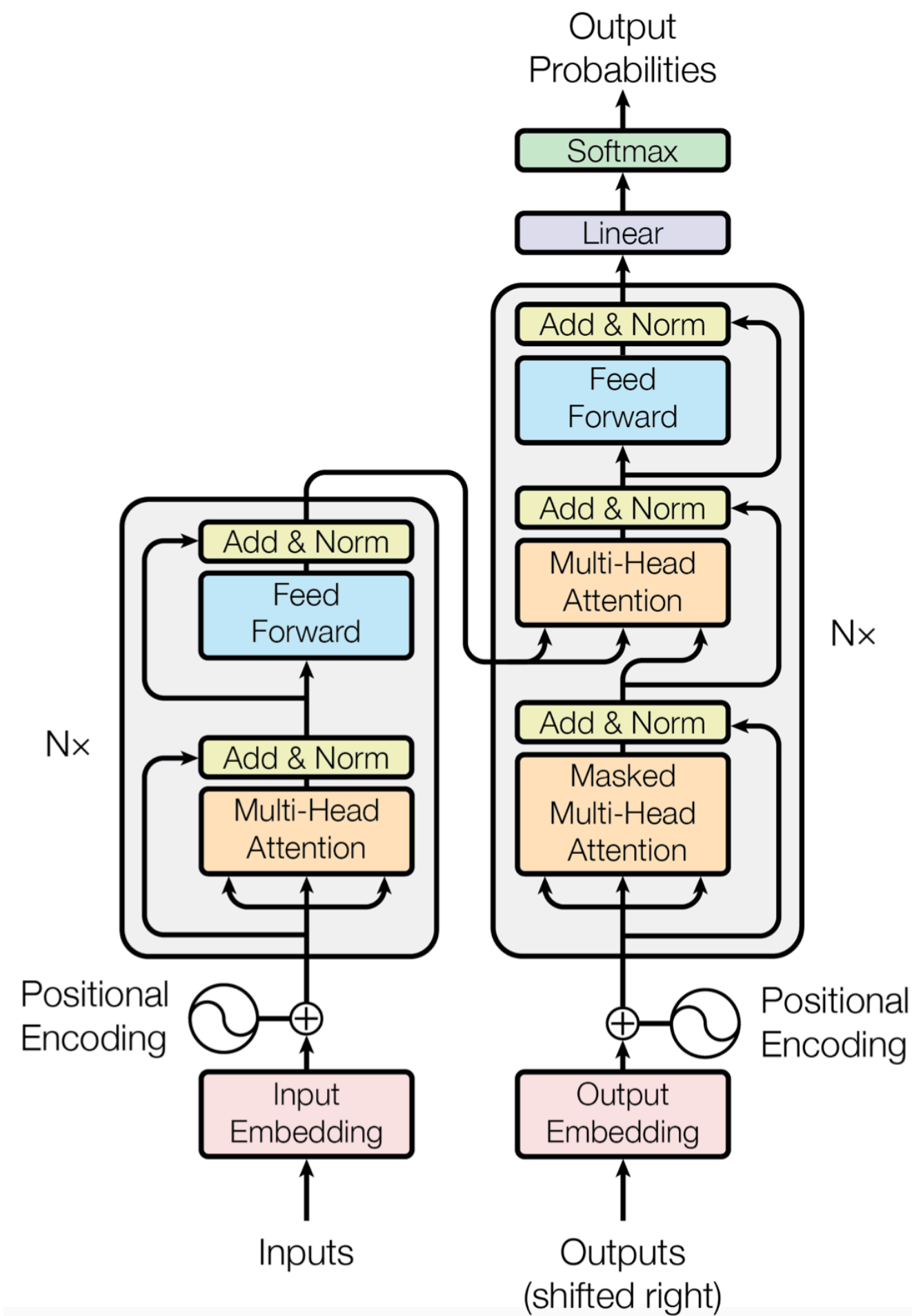


Vaswani et al.



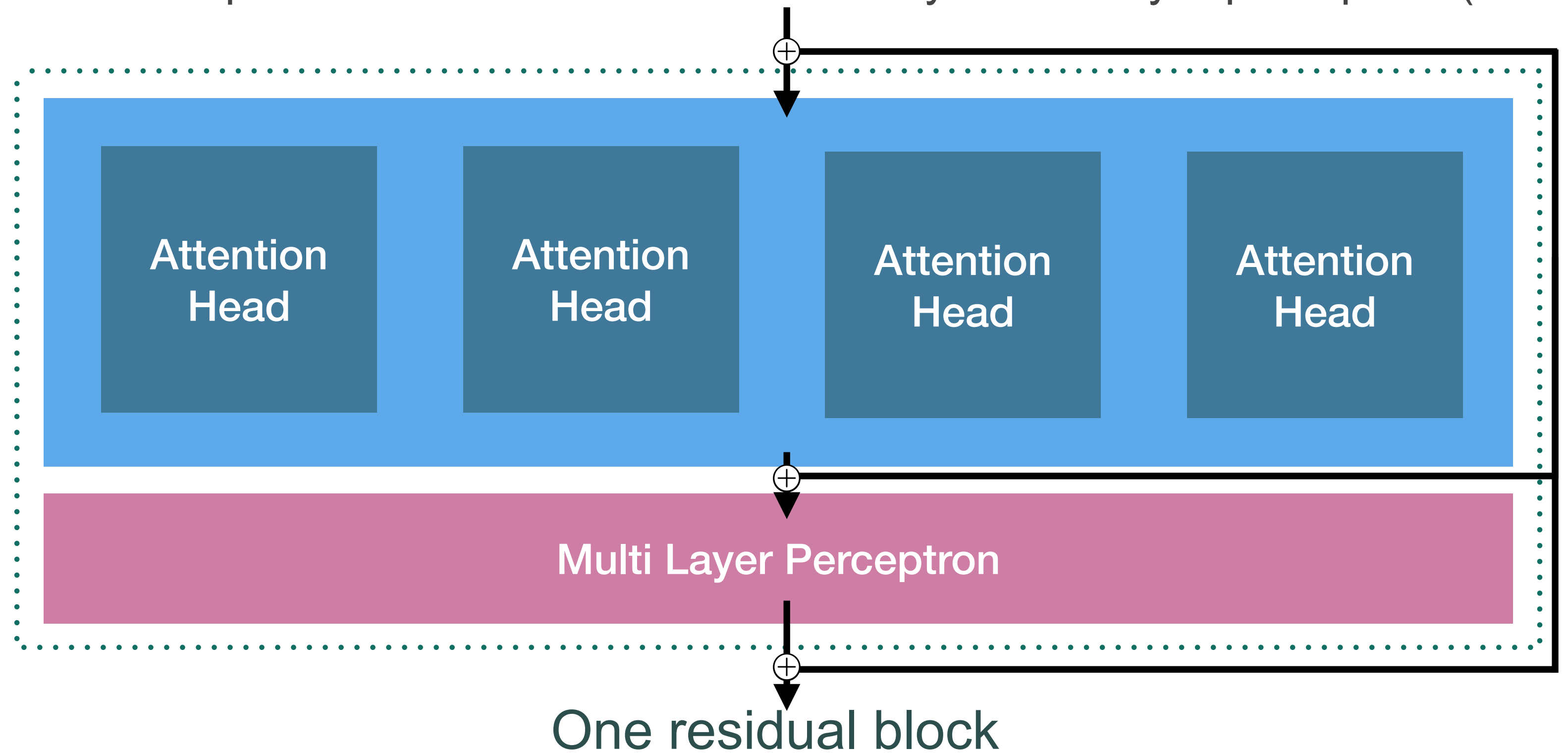
Elhage et al.

A new way of looking at transformers



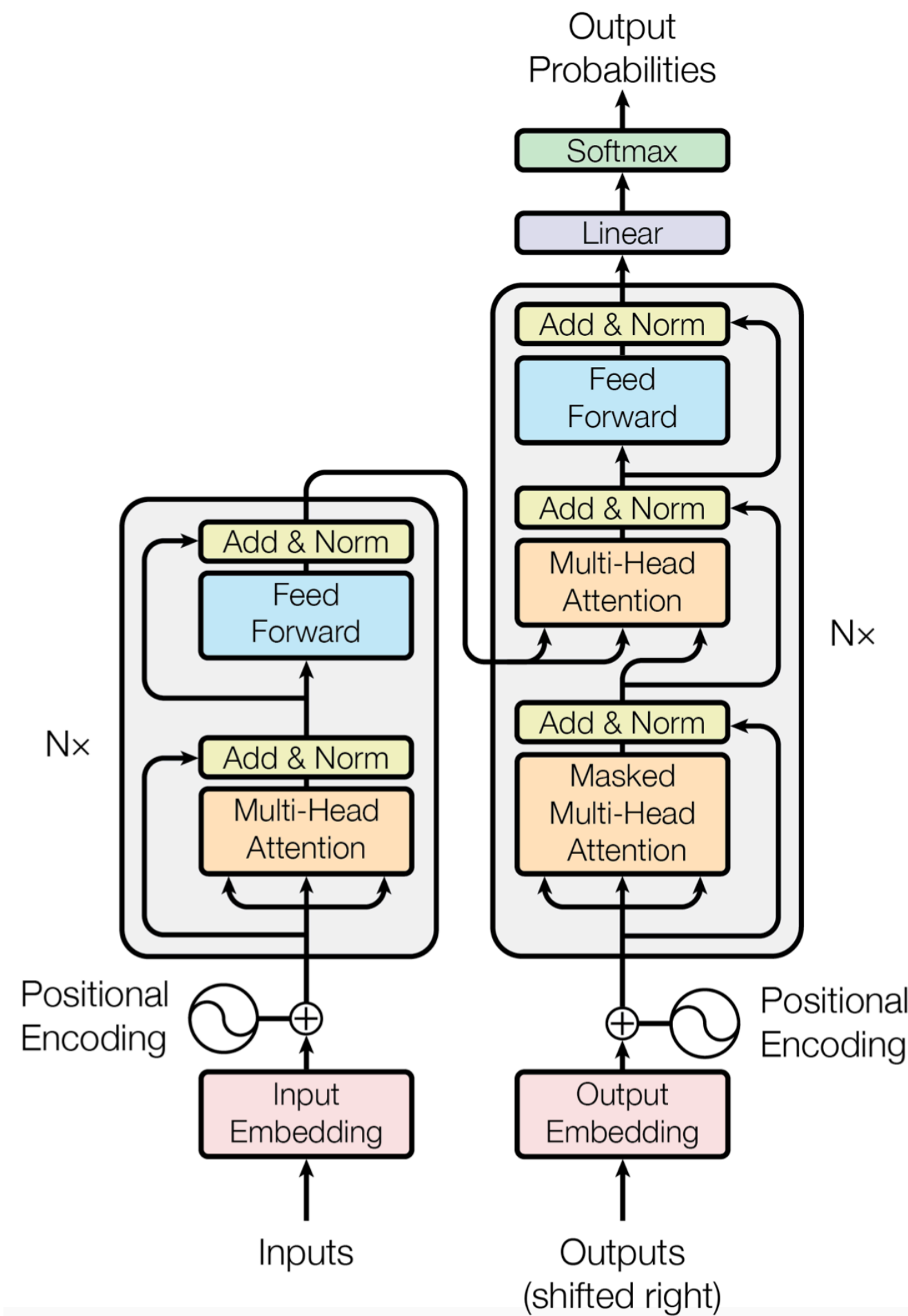
Vaswani et al.

Transformers are a sequence of **residual blocks**: these are made up of a series of parallel attention heads followed by a multi-layer perceptron (MLP).



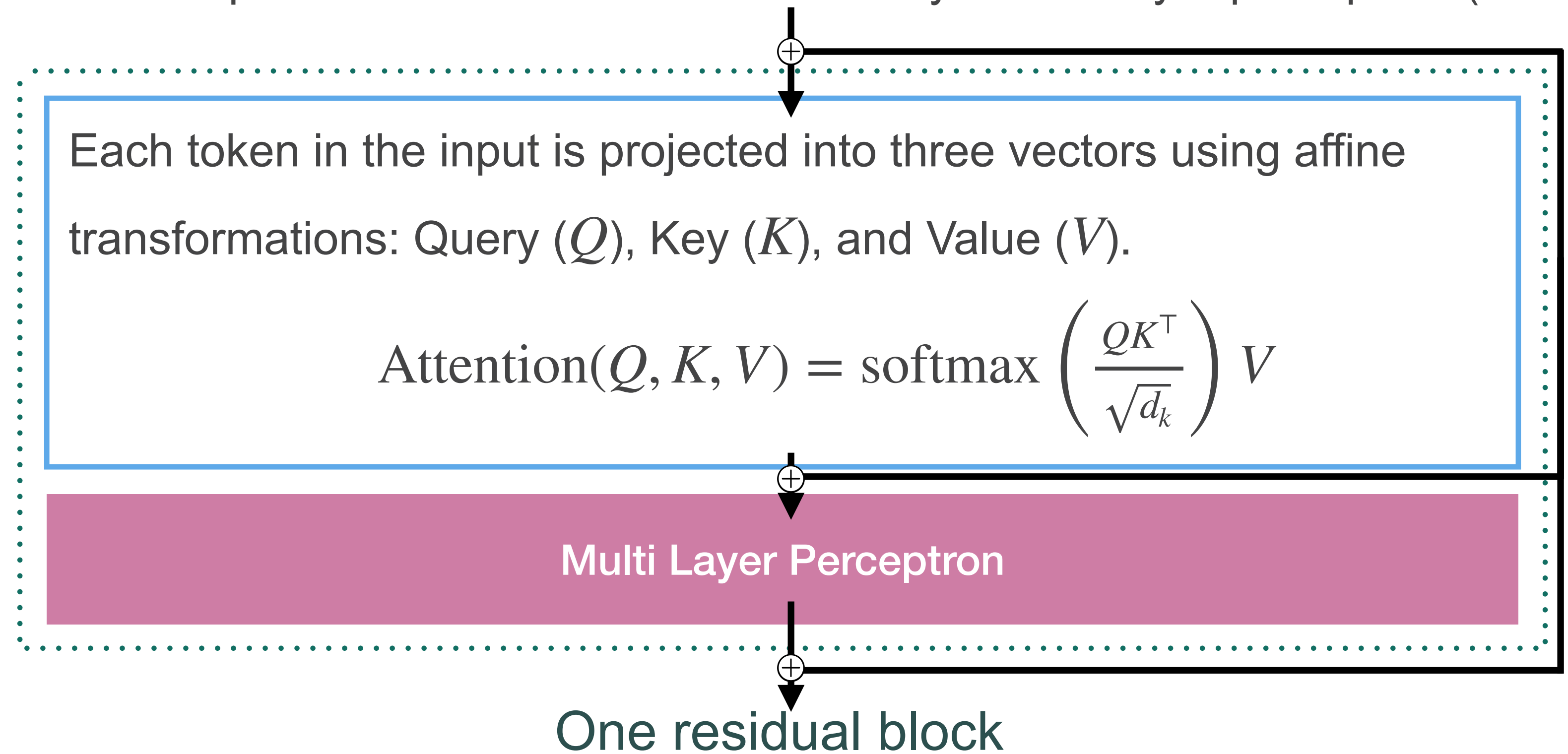
* This is an extremely over-simplified description.

A new way of looking at transformers



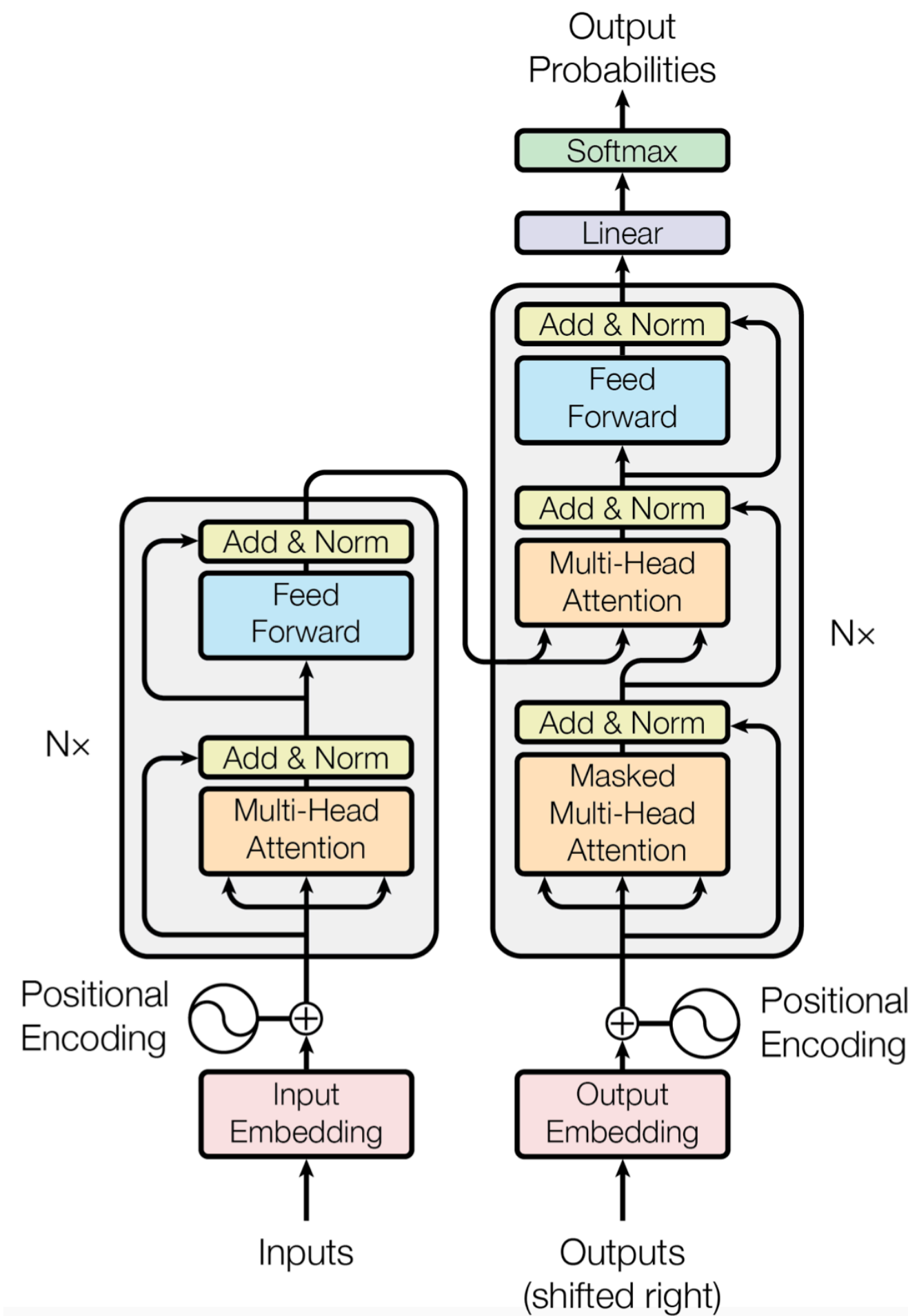
Vaswani et al.

Transformers are a sequence of **residual blocks**: these are made up of a series of parallel attention heads followed by a multi-layer perceptron (MLP).



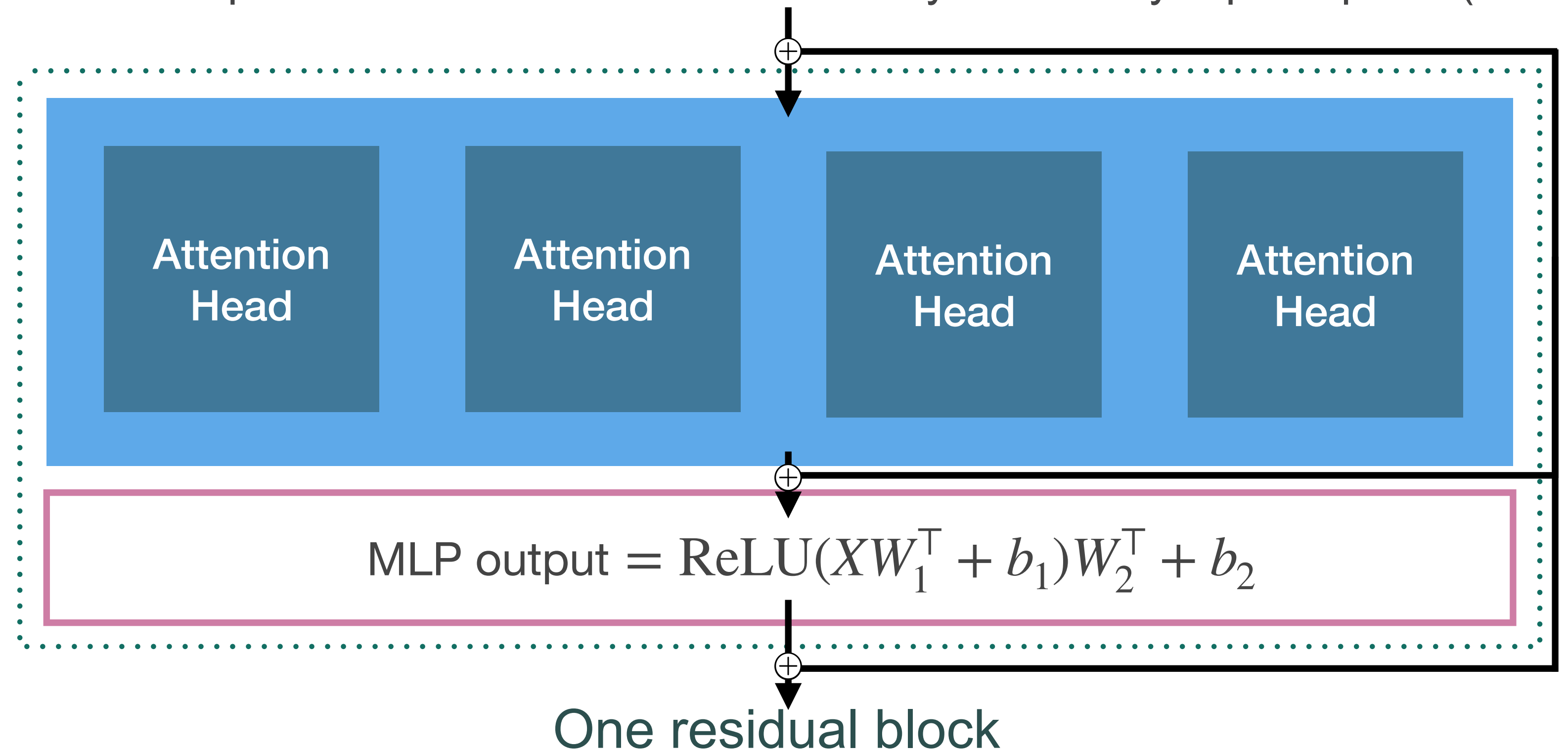
* This is an extremely over-simplified description.

A new way of looking at transformers



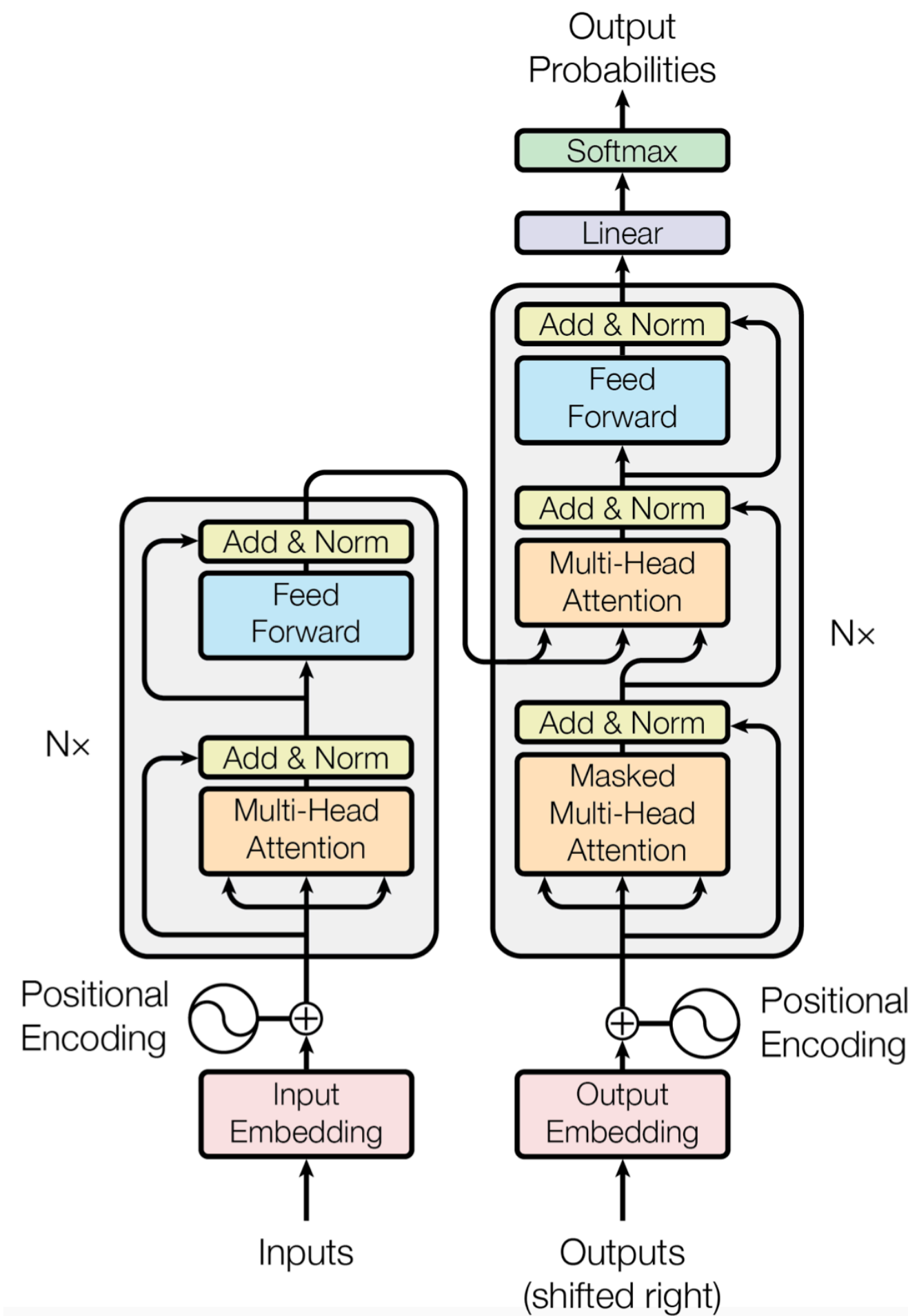
Vaswani et al.

Transformers are a sequence of **residual blocks**: these are made up of a series of parallel attention heads followed by a multi-layer perceptron (MLP).



* This is an extremely over-simplified description.

A new way of looking at transformers



Vaswani et al.

Transformers are a sequence of **residual blocks**: these are made up of a series of parallel attention heads followed by a multi-layer perceptron (MLP).

Each token in the input is projected into three vectors using affine transformations: Query (Q), Key (K), and Value (V).

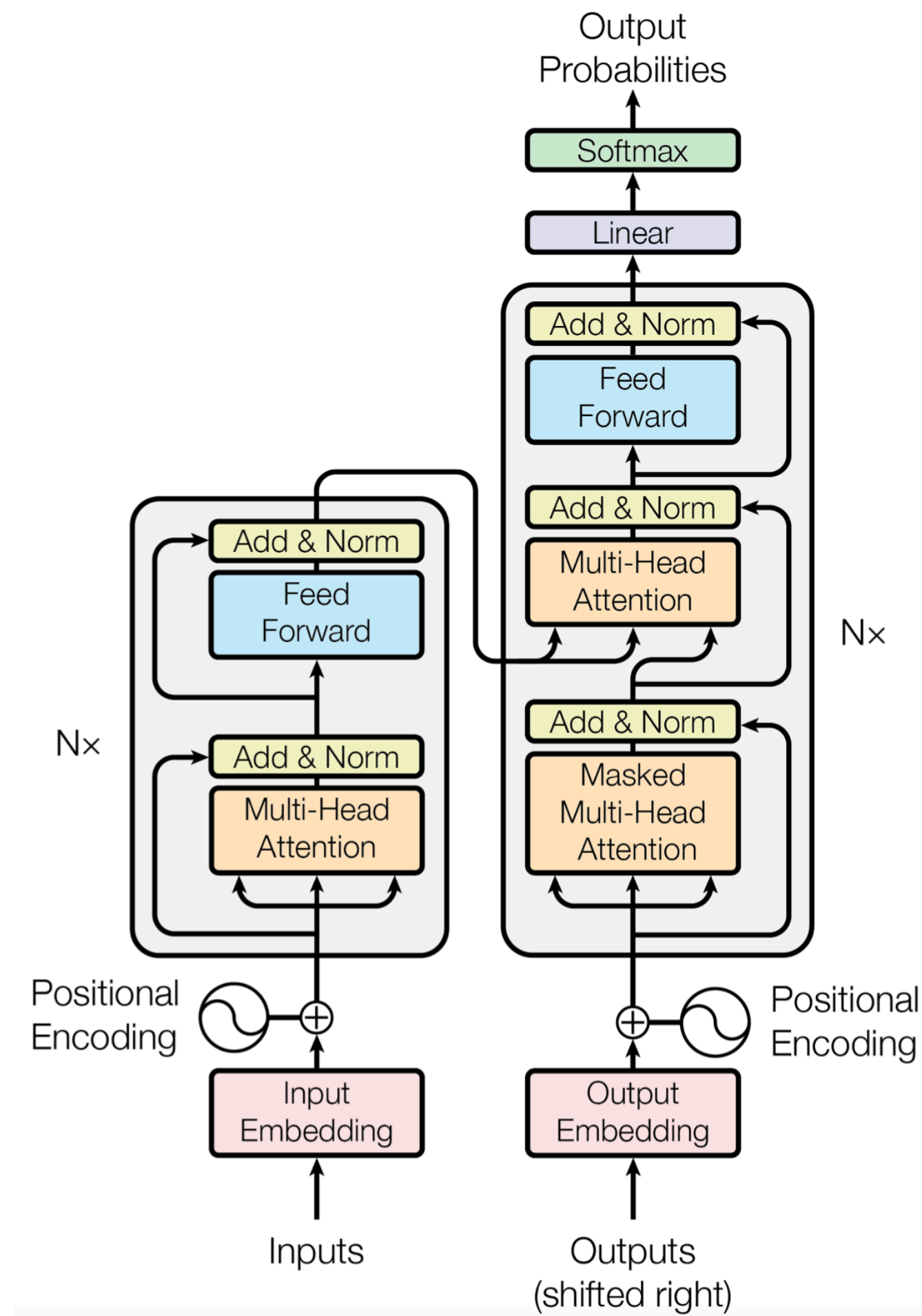
$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

$$\text{MLP output} = \text{ReLU}(XW_1^T + b_1)W_2^T + b_2$$

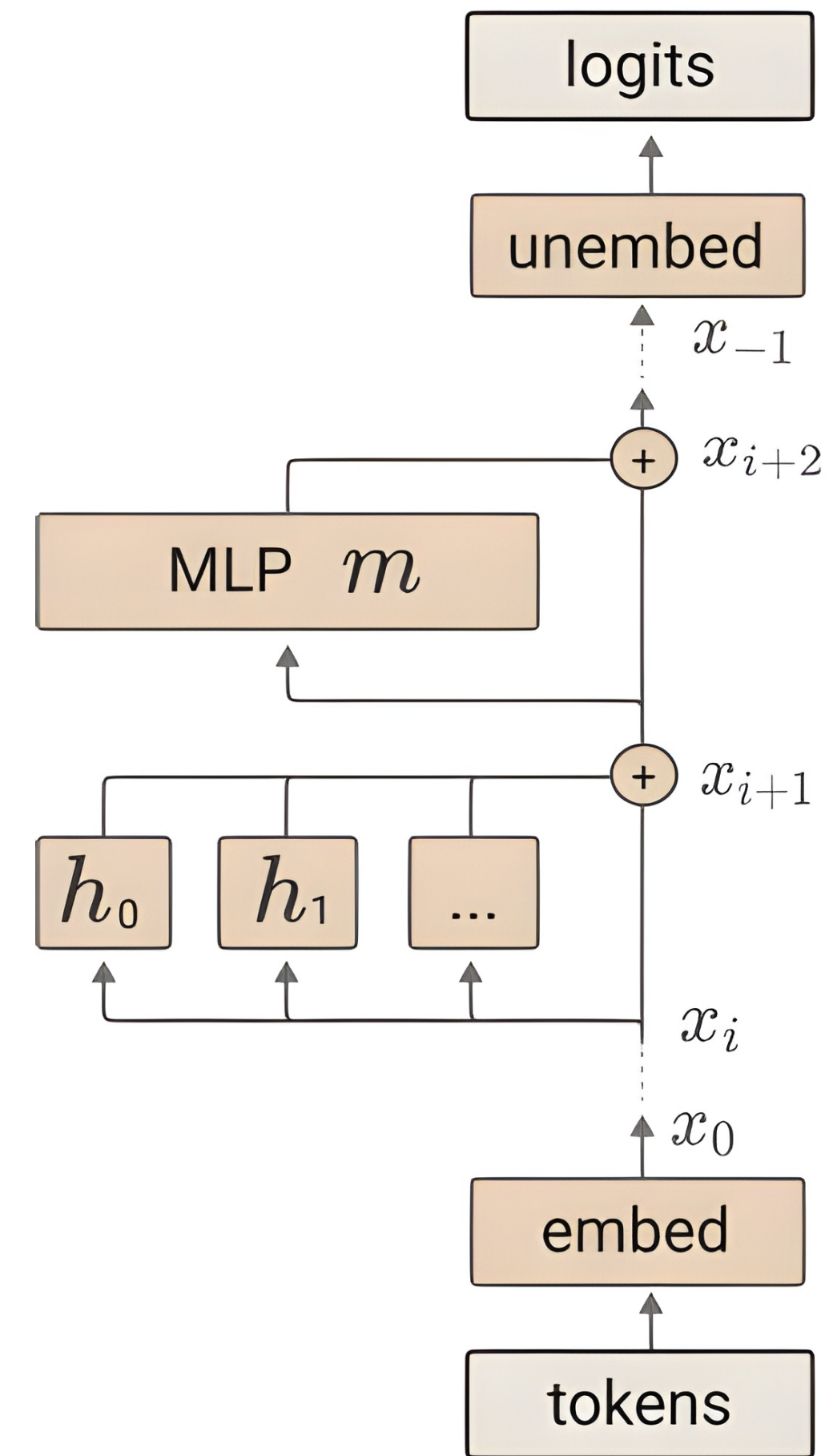
Importantly, we view the model as something that evolves embeddings at each layer, similar to traditional views of neural networks.

* This is an extremely over-simplified description.

A new way of looking at transformers



Vaswani et al.



Elhage et al.

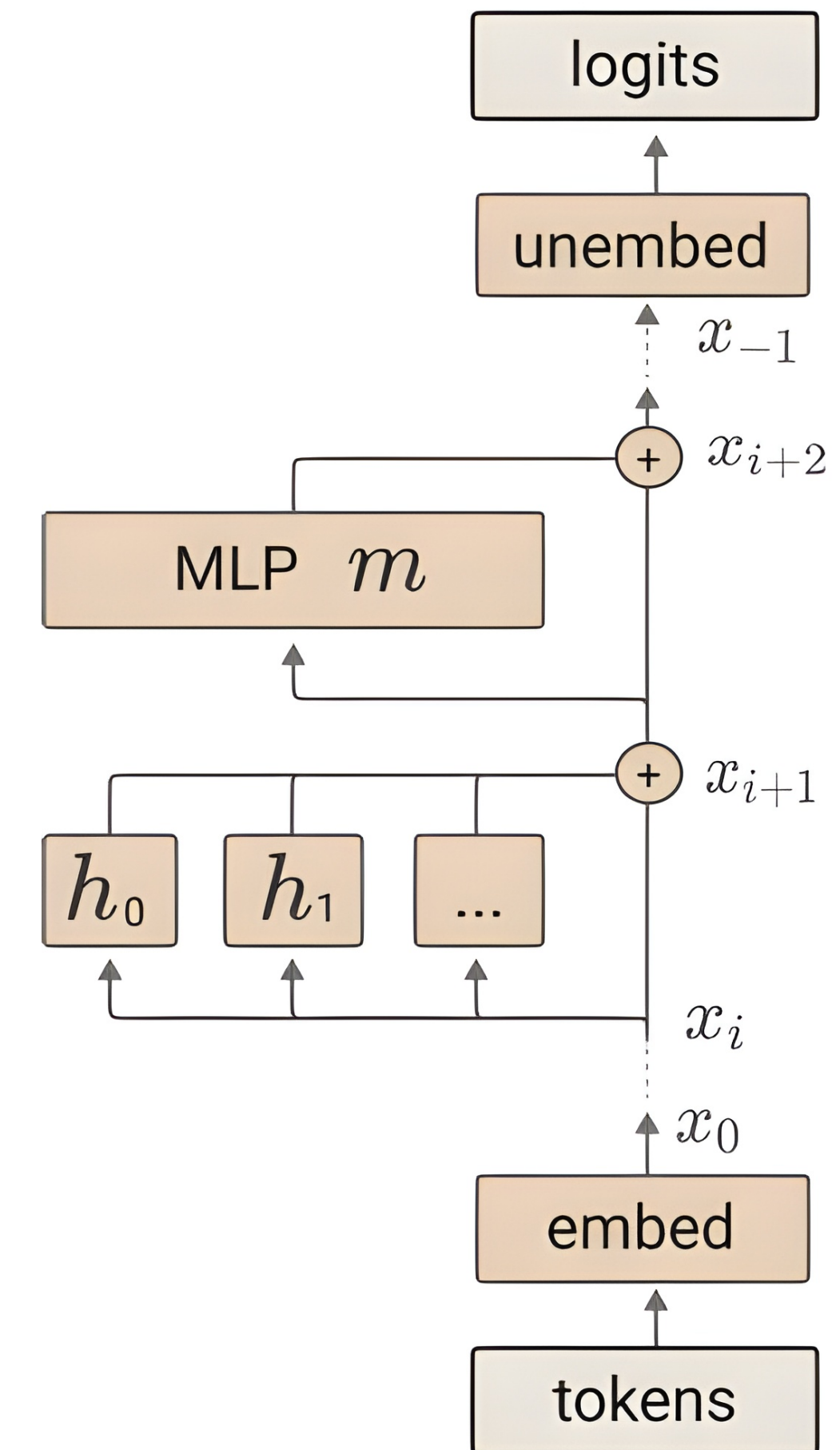
A new way of looking at transformers

Transformers are a sequence of (almost) linear transformations interspersed with nonlinear MLPs.

This means that the transformer can be seen as a collection of smaller, modular “**circuits**” that perform independent sub-computations.

These sub-computations get read to and written from a central “communication channel” (AKA the **residual stream**) of the model to be combined into the final output.

This view allows researchers to trace how information flows and transforms through the model.



Elhage et al.

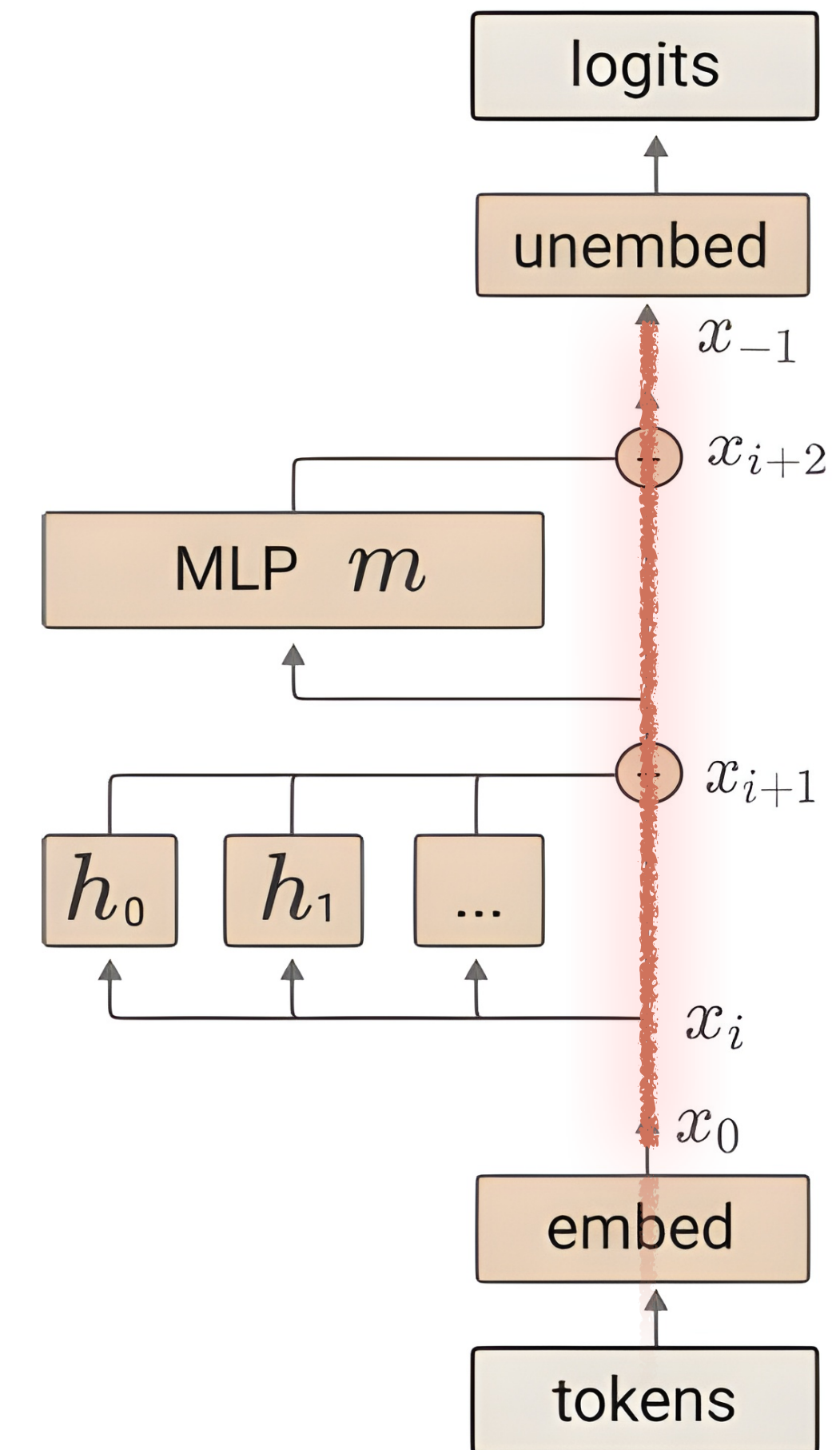
A new way of looking at transformers

Transformers are a sequence of (almost) linear transformations interspersed with nonlinear MLPs.

This means that the transformer can be seen as a collection of smaller, modular “**circuits**” that perform independent sub-computations.

These sub-computations get read to and written from a central “communication channel” (AKA the **residual stream**) of the model to be combined into the final output.

This view allows researchers to trace how information flows and transforms through the model.

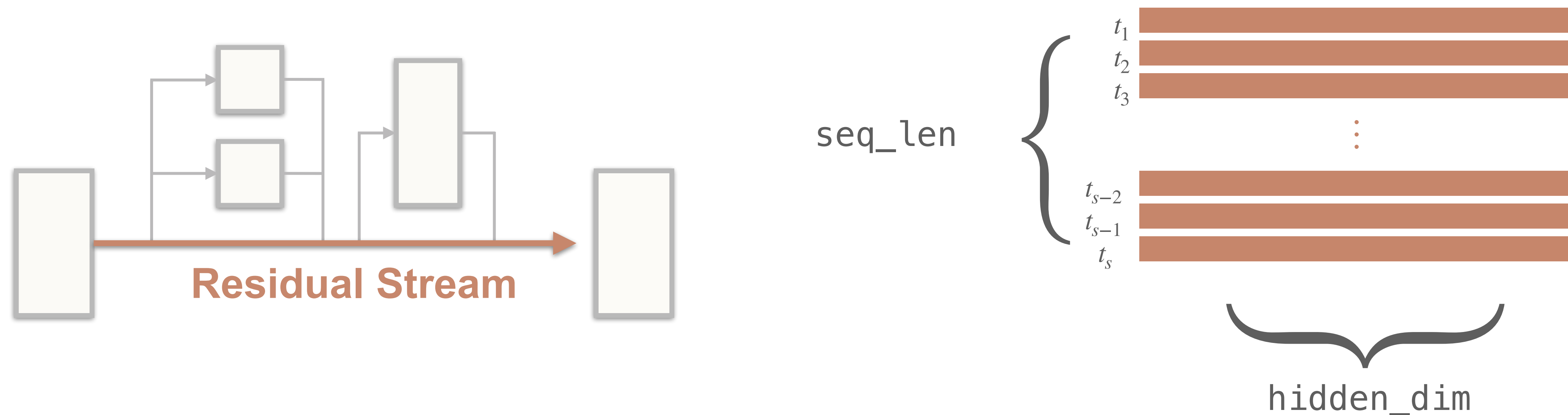


Elhage et al.

What is the residual stream?

Residual connections have the form $x_t = g(x_{t-1}) + x_{t-1}$, where g is some operation and x_t is of shape `[seq_len, hidden_dim]`

The sequence $(x_1, x_2, \dots, x_t, \dots)$ is called the residual stream.



What is the residual stream?

The residual stream has an inherently **linear structure** and the information is **persistent** across layers (a “write” at layer i might be read at layer say $i + 10$).

$$xW_V^T$$

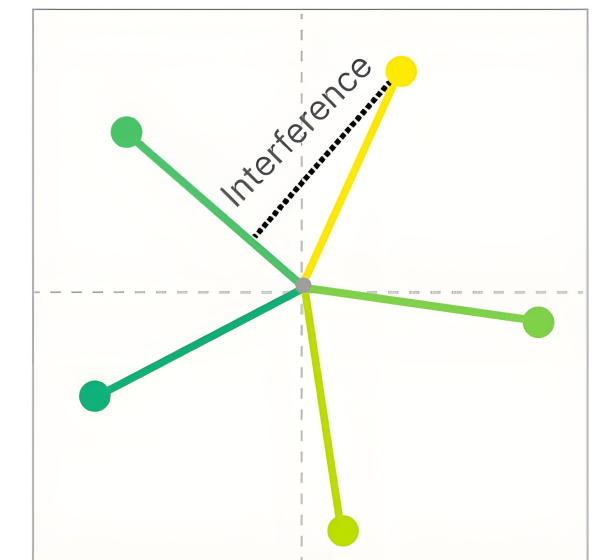
A **read** is a linear projection of residual stream.

$$x = x + w$$

A **write** is an addition to the residual stream

$$x = \sum_{i \in \text{writes}} w_i$$

The residual stream admits an **additive decomposition** into all of its writes.



Concepts are (roughly) **orthogonal directions** in the residual stream

x is [seq_len , hidden_dim]
 W is [head_dim, hidden_dim]

Attention heads and the residual stream

```
def head( $x_V$ ,  $x_K$ ,  $x_Q$ ):  
     $w = x_V(W_O W_V)^T$   
     $A = \text{softmax}(x_Q W_Q^T W_K x_K^T)$   
    return  $A w$ 
```

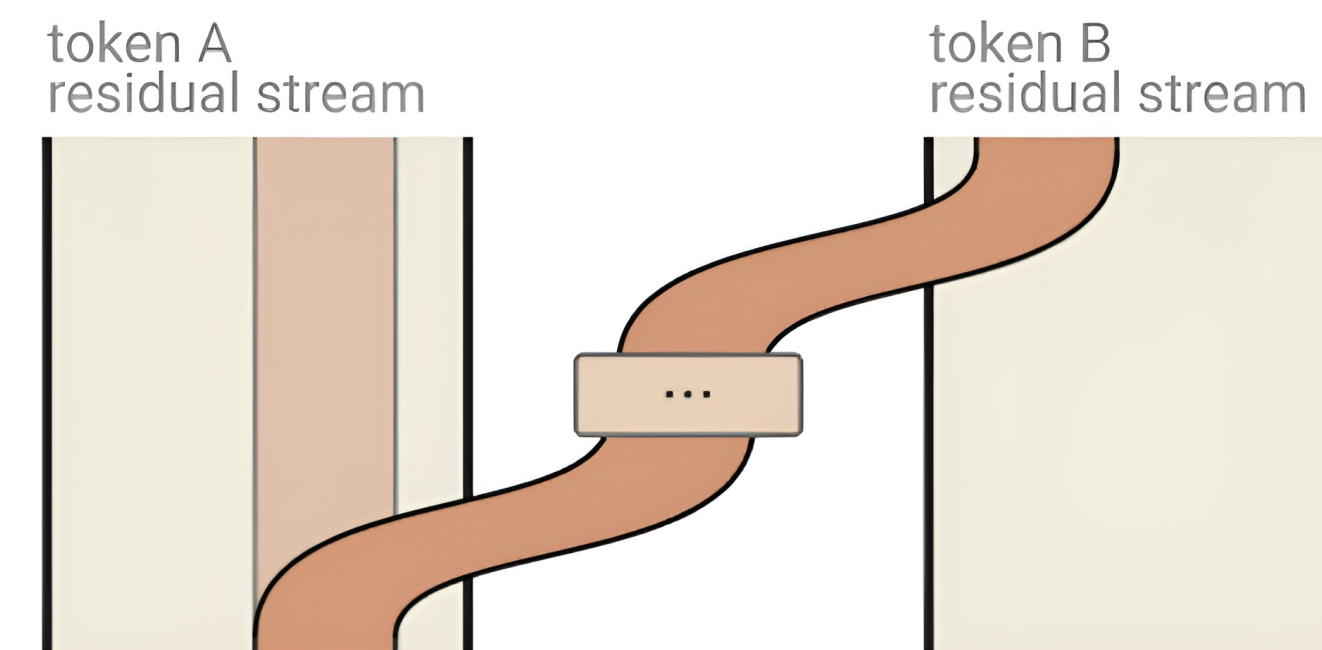
```
 $x = \text{head}(x, x, x) + x$ 
```

x_V, x_K, x_Q are [seq_len , hidden_dim]
 W_V, W_Q, W_K are [head_dim, hidden_dim]
 W_O is [hidden_dim, head_dim]

Two distinct operations.

OV-circuit (w): Decides **what to read and what to write** to the residual stream by linearly transforming each token position independently.

QK-circuit (A): Takes a weighted average over sequence dimension to **move information across tokens**.



Attention heads and the residual stream

```
def head( $x_V$ ,  $x_K$ ,  $x_Q$ ):  
     $w = x_V(W_O W_V)^T$   
     $A = \text{softmax}(x_Q W_Q^T W_K x_K^T)$   
    return  $A w$ 
```

```
 $x = \text{head}(x, x, x) + x$ 
```

x_V, x_K, x_Q are [seq_len , hidden_dim]
 W_V, W_Q, W_K are [head_dim, hidden_dim]
 W_O is [hidden_dim, head_dim]

Some observations:

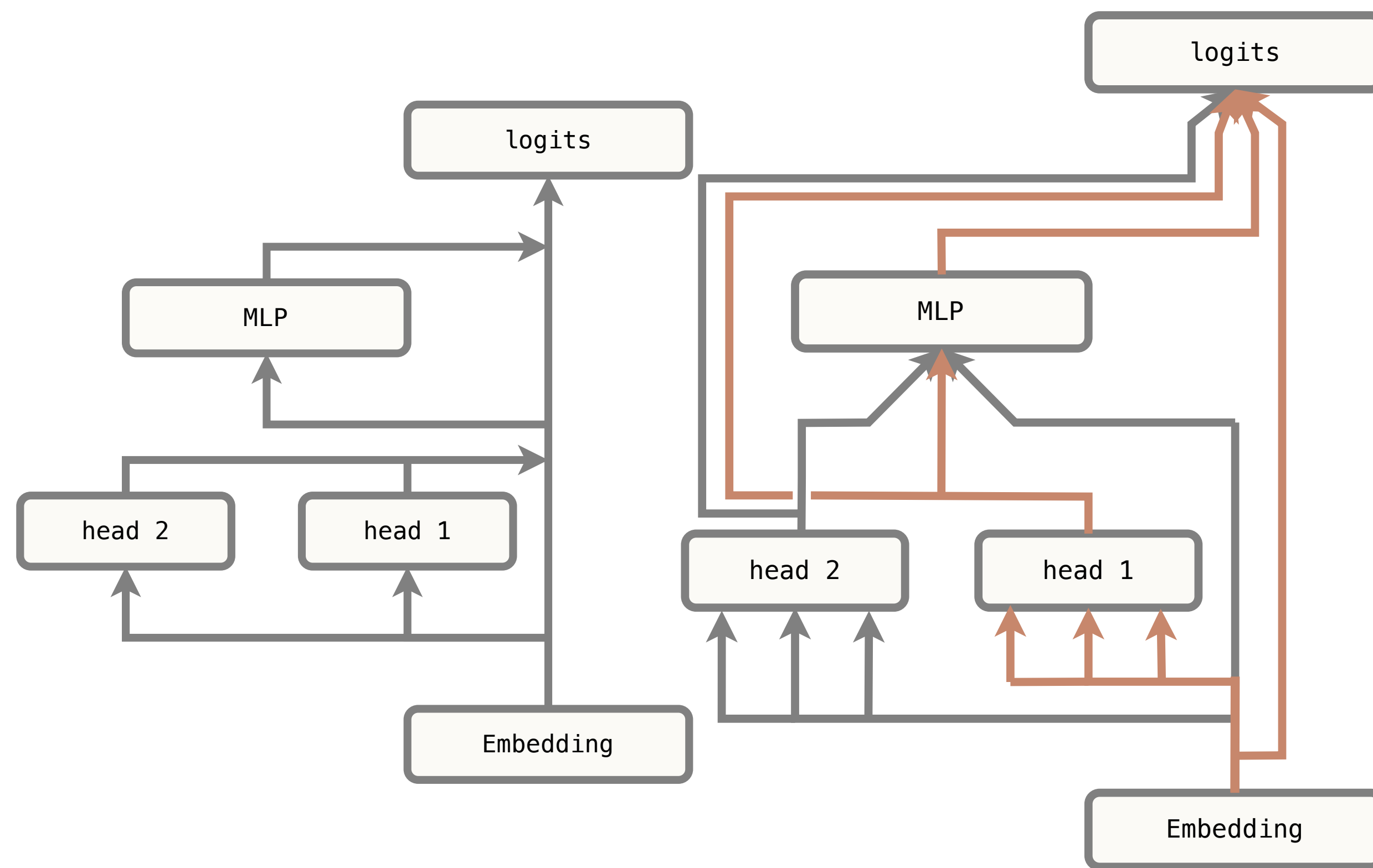
1. The residual stream enters into the operation in **three “independent ways”**.
2. The **head operation is linear** in x_V because we have

$$\text{head}(x_v, \dots) = A x_V (W_O W_V)^T$$

3. The **head** implicitly receives as **arguments all previous writes**. We can decide which.

$$\text{head}(x_V, x_K, x_Q) = \text{head}\left(\sum_i w_i, \sum_i w_i, \sum_i w_i\right)$$

The residual stream & path expansion



Current Model

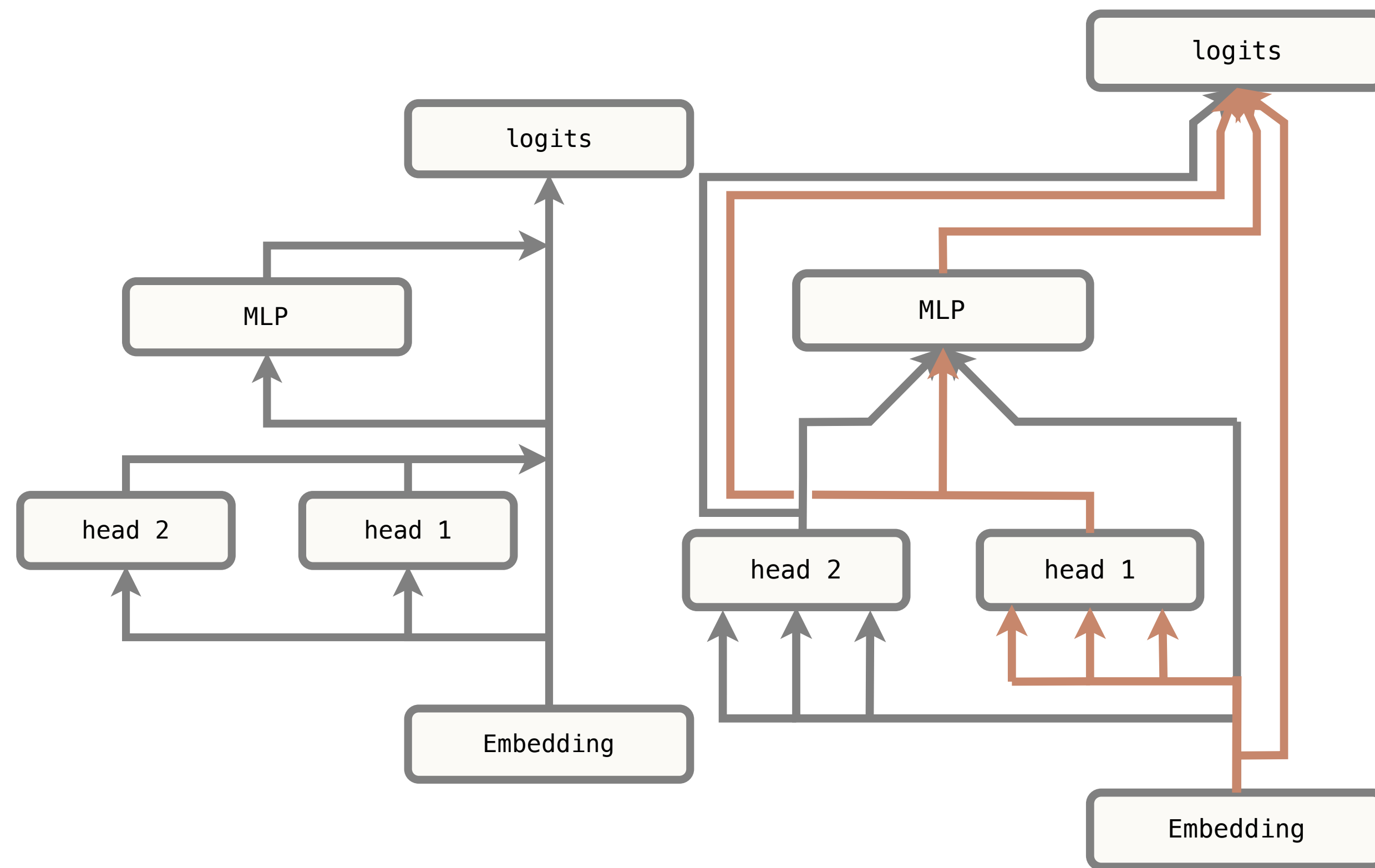
Expanded Graph
&
Example Circuit

Ideally, we want to study “paths” of computation in a transformer to understand how components interact and affect the output.

We can formalize such paths as follows:

1. We construct an **expanded graph** such that components C_1 and C_2 share an edge if a write w by C_1 is read by C_2 .
2. A **path** is any path (in the graph sense) from the embeddings to the logits.
3. A **circuit** is a collection of such paths (or more generally any sub-graph.)

The residual stream & path expansion



Current Model

Expanded Graph & Example Circuit

To run a circuit:

1. Run the transformer and, as you go along, cache the writes w_C for every component C .
2. When running a component C , feed as input only the writes corresponding to edges in the circuit.

$$\sum_{(C',C) \in G} w'_C$$

To note:

- a) Additive effect for logits (OV circuit preserves linearity)
- b) This generalizes **path expansion** in the paper.
- c) There are tricks but, in general, it is expensive.

Empirical Results

Analyzing the transformer

In order to use these concepts to reverse engineer results, Elhage et al. looked at some toy transformer models. Specifically, their setting for simplifying the models was the following:

- Attention-only (no MLP layers)
- Decoder-only (like GPT-3)
- No layer-norm or biases
- Reduced the number of layers; looked at 0L, 1L, and 2L transformers

OL: Bi-gram(ish) statistics

The simplest possible model applies the embedding and unembedding matrices. The best we can do with this model is to approximate the bi-gram statistics with the “distributed representations”.

$$\text{softmax}(t(W_U W_E)^T)$$

W_U is [n_vocab , hidden_dim]

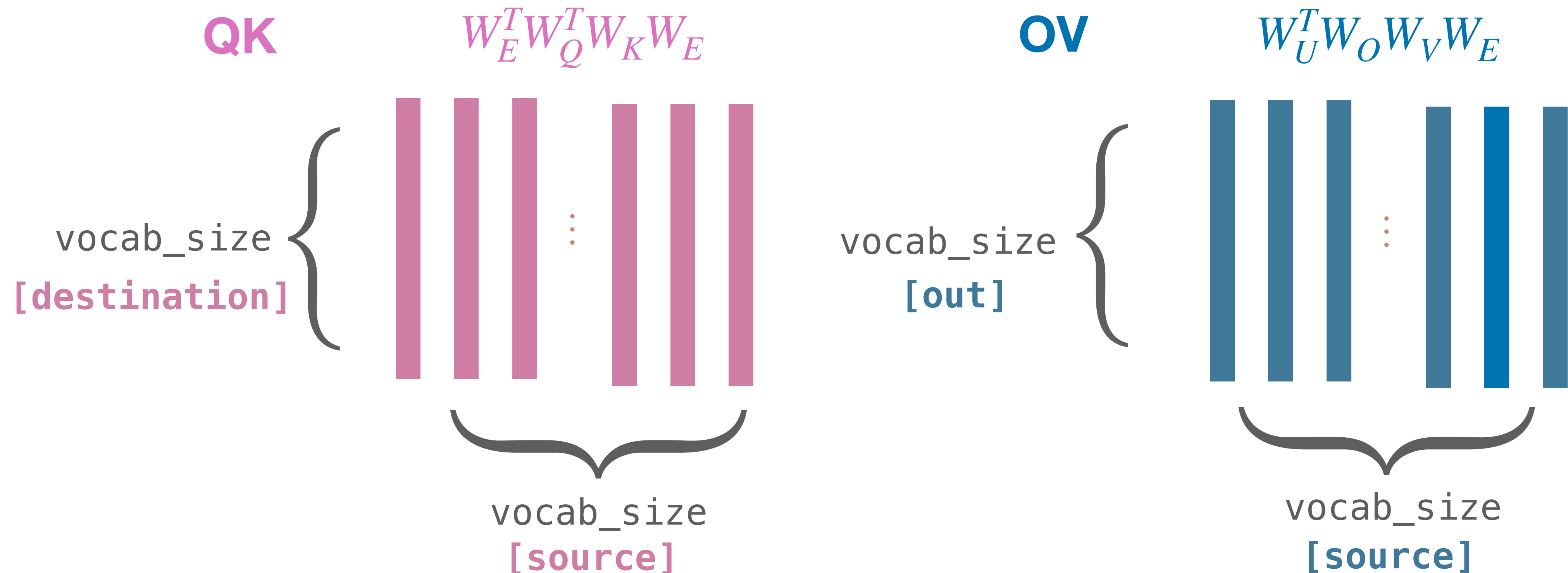
W_E is [hidden_dim, n_vocab]

t is [seq_len, n_vocab]

1L: Skip tri-grams

Reading off the **QK** and **OV** circuits gives us a “**source**” token, “**destination**” token, and “**out**” predictions for the next token. Together, these three tokens form a “**skip-trigram**” of the form:

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



1L: Skip tri-grams

Reading off the **QK** and **OV** circuits gives us a “**source**” token, “**destination**” token, and “**out**” predictions for the next token. Together, these three tokens form a “**skip-trigram**” of the form:

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

More examples of large entries QK/OV circuit

Source Token	Destination Token	Out Token	Example Skip Tri-grams
"indy"	" C", "C", " V", "V", " R", " c"	"indy", "obby", "INDY", "loyd"	"indy... Cindy", "indy... CINDY"
" Pike"	" P", "P", "V", "Sp", " V", "R"	"ike", "ikes", "ishing", "owler"	" Pike... Pike", " Pike... Spikes"
" Ralph"	"R", " R", " P", "P", "V", " r"	"alph", "ALPH", "obby", "erald"	" Ralph... Ralph", " Ralph... RALPH"
" Lloyd"	"L", " L", " P", "P", "R", " C"	"loyd", "alph", "\n ", "acman", ... "atherine"	" Lloyd... Lloyd", " Lloyd... Catherine"
" Pixmap"	" P", " Q", "P", " p", " U"	"ixmap", "Canvas", "Embed", "grade"	" Pixmap... Pixmap", " Pixmap... QCanvas"

1L: Skip tri-grams

Reading off the **QK** and **OV** circuits gives us a “**source**” token, “**destination**” token, and “**out**” predictions for the next token. Together, these three tokens form a “**skip-trigram**” of the form:

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

Summarization of observed 1L copying behavior:

Primitive In-Context Learning Patterns

[b]...[a] → [b]

[two]...[One] → [two]

[perfect]...[are] → [perfect]

[nbsp]...[&] → [nbsp]

[lambda]...[\$\\] → [lambda]

[b]...[a] → [b']

[two]...[has] → [three]

[perfect]...[looks] → [super]

[nbsp]...[&] → [gt]

[lambda]...[\$\\] → [operator]

[ab]...[a] → [b]

[Ralph]...[R] → [alph]

[Pike]...[P] → [ike]

[Pixmap]...[P] → [ixmap]

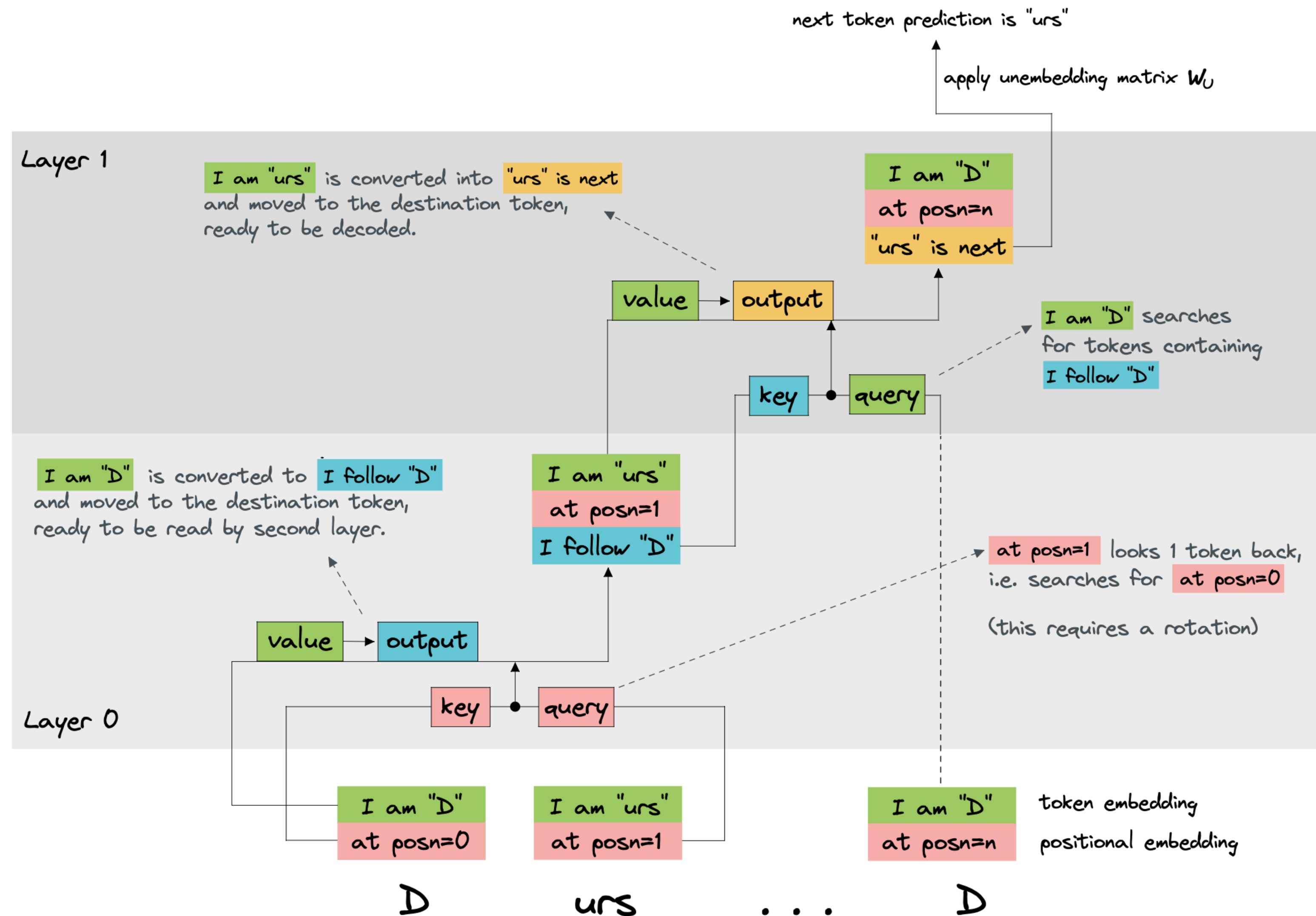
[Lloyd]...[L] → [loyd]

[ab]...[a] → [b']

[Ralph]...[R] → [ALPH]

[Pike]...[P] → [ikes]

2L: Induction heads



2L model induction head:

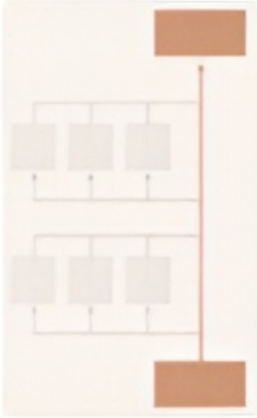
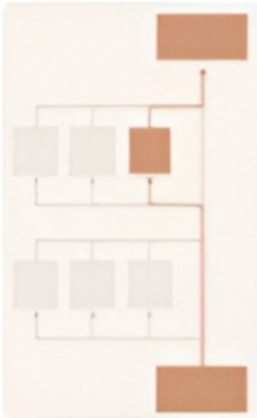
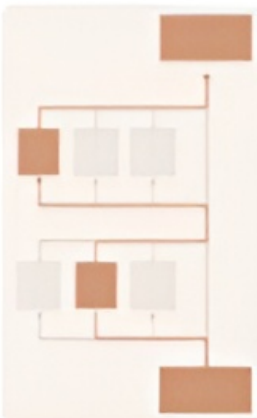
[a] [b] ... [a] → [b]

1L model copying head:

[b] ... [a] → [b]

Circuit ablation studies

We are able to determine the marginal information gain for all independent paths in the model.

Type	Example	Equation	Marginal Loss Reduction
direct path order 0		$W_U W_E$	- 1.8 nats relative to uniform predictions -1.8 nats/term (- 1.8 nats / 1 term)
individual attention head order 1		$A^h \otimes (W_U W_{OV}^h W_E)$	- 5.2 nats relative to only using direct path -0.2 nats/term (5.2 nats / 24 terms)
virtual attention head order 2		$(A^{h_2} A^{h_1}) \otimes (W_U W_{OV}^{h_2} W_{OV}^{h_1} W_E)$	- 0.3 nats relative to only using above -0.002 nats/term (0.3 nats / 144 terms)

Current Research Directions

INTERPRETABILITY IN THE WILD: A CIRCUIT FOR INDIRECT OBJECT IDENTIFICATION IN GPT-2 SMALL

Kevin Wang¹, Alexandre Variengien¹, Arthur Conmy¹, Buck Shlegeris¹ & Jacob Steinhardt^{1,2}

¹Redwood Research

²UC Berkeley

kevin@rdwrs.com, alexandre@rdwrs.com,
arthur@rdwrs.com, buck@rdwrs.com, jsteinhardt@berkeley.edu

ABSTRACT

Research in mechanistic interpretability seeks to explain behaviors of machine learning (ML) models in terms of their internal components. However, most previous work either focuses on simple behaviors in small models or describes complicated behaviors in larger models with broad strokes. In this work, we bridge this gap by presenting an explanation for how GPT-2 small performs a natural language task called indirect object identification (IOI). Our analysis

What behaviors can
we find from
circuits?

Towards Automated Circuit Discovery for Mechanistic Interpretability

Arthur Conmy*
Independent

Augustine N. Mavor-Parker*
UCL

Aengus Lynch*
UCL

Stefan Heimersheim
University of Cambridge

Adrià Garriga-Alonso*
FAIR AI

Abstract

Through considerable effort and intuition, several recent works have reverse-engineered nontrivial behaviors of transformer models. This paper systematizes the mechanistic interpretability process they followed. First, researchers choose a metric and dataset that elicit the desired model behavior. Then, they apply ac-

Can we automate
the discovery of
interesting
circuits?

Towards Monosemanticity: Decomposing Language Models With Dictionary Learning

Using a sparse autoencoder, we extract a large number of interpretable features from a one-layer transformer.

[Browse A/1 Features →](#)

[Browse All Features →](#)

Can we understand
the features in the
residual stream?

Questions?