# Transformers / Vision Transformers

EE/CS/CNS148 2022

Neehar Kondapaneni

# Transformers (2017)

**Attention Is All You Need**

**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* ‡** illia.polosukhin@gmail.com

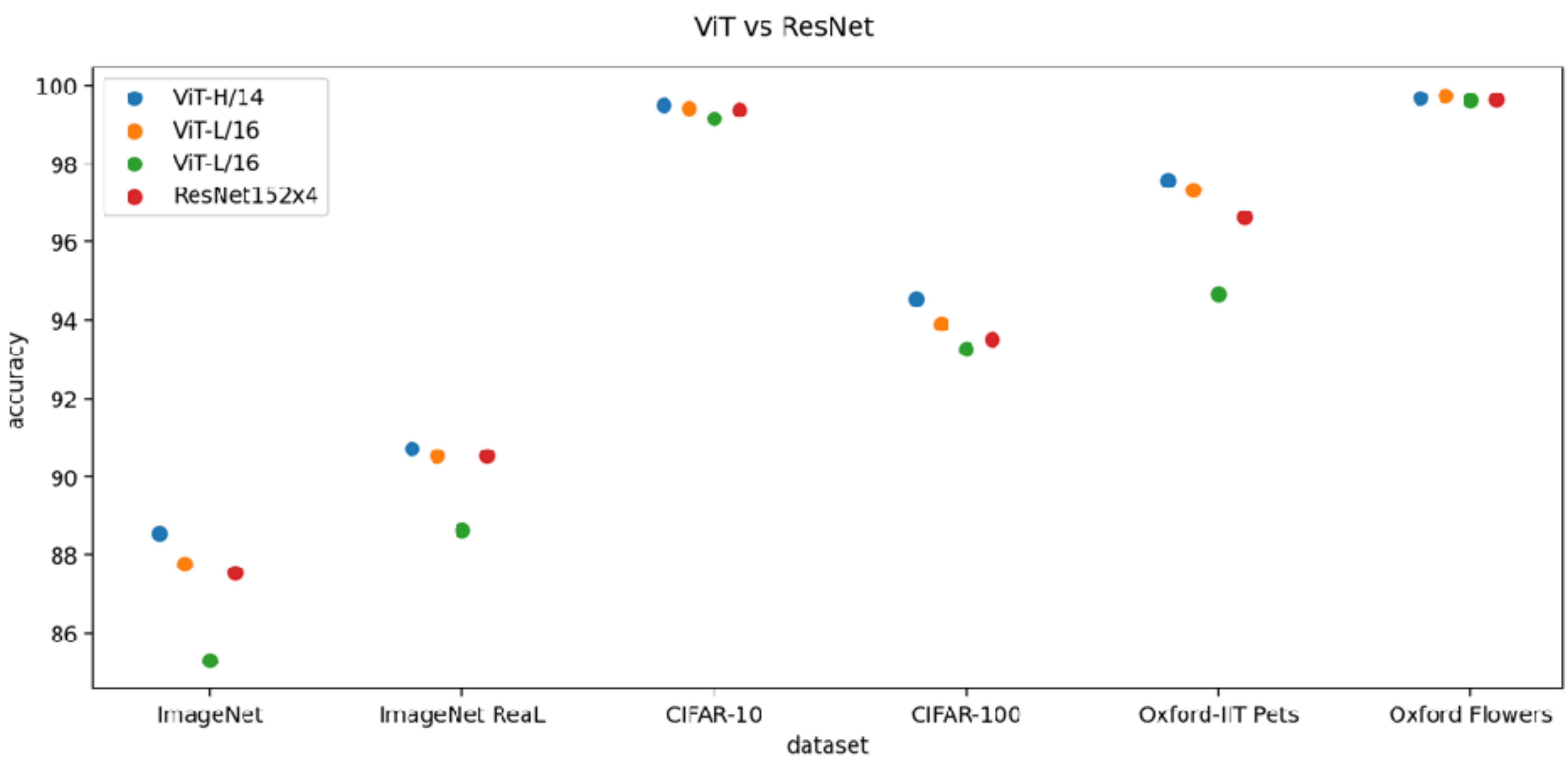
The quality of the text generated by GPT-3 is so high that it can be difficult to determine whether or not it was written by a human, which has both benefits and risks.[4]

Transformers were originally designed for natural language processing… here's a link to an interactive example.

# Vision Transformers (ViT) (2021)

AN IMAGE IS WORTH 16X16 WORDS:
TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy*,†, Lucas Beyer*, Alexander Kolesnikov*, Dirk Weissenborn*,
Xiaohua Zhai*, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby*,†
*equal technical contribution, †equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com




****Ongoing research on whether ResNets or Transformers are better…**

# Lecture Roadmap

1. **Motivation**
2. Word Embeddings
3. Attention
    - What is it, intuitively?
    - What is it, mathematically?
4. Scalar Dot-Product Attention
    - Why?
    - Queries, Keys, Values
    - Computing Attention
5. Multi-Headed Attention
    - MHA Intuition
6. Transformer Architecture
7. Vision Transformers
    - Moving from text to images
    - Comparing transformers and CNNs

# Motivation

Suppose we want to do language translation…

**Italian**: Io la sto mangiando.

**Direct Translation**: I it am eating.

**English**: I am eating it.

# Motivation

Suppose we want to do language translation…

**Italian**: Io la sto mangiando.

**Direct Translation**: I it am eating.

**English**: I am eating it.

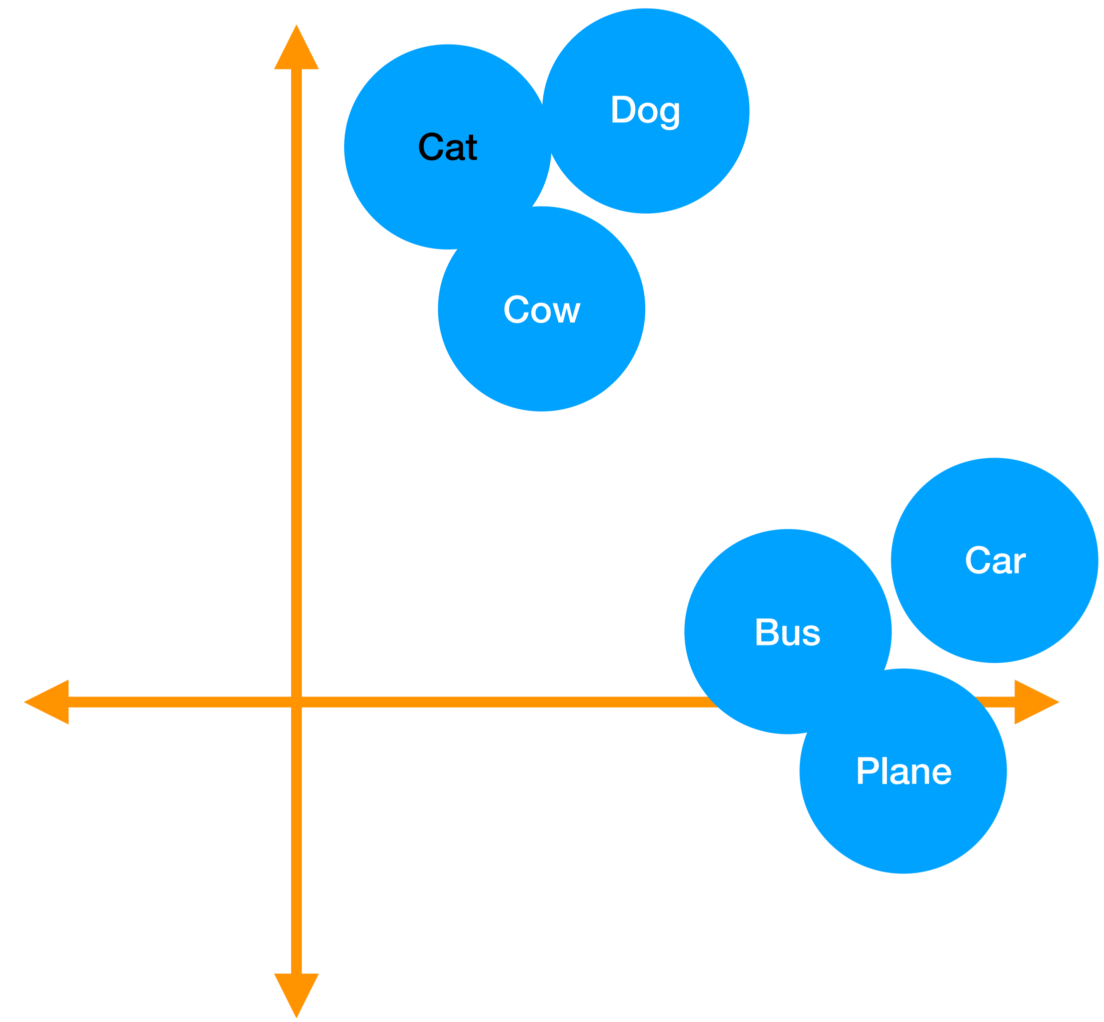
A translator would need to determine what parts of the **Italian** sentence to pay **attention** to, in order to translate it correctly.

# Lecture Roadmap

1. Motivation
2. **Word Embeddings**
3. Attention
    - What is it, intuitively?
    - What is it, mathematically?
4. Scalar Dot-Product Attention
    - Why?
    - Queries, Keys, Values
    - Computing Attention
5. Multi-Headed Attention
    - MHA Intuition
6. Transformer Architecture
7. Vision Transformers
    - Moving from text to images
    - Comparing transformers and CNNs

# Word Embeddings


**Word2Vec (2013)



# Lecture Roadmap

1. Motivation
2. Word Embeddings
3. **Attention**
    - What is it, intuitively?
    - What is it, mathematically?
4. Scalar Dot-Product Attention
    - Why?
    - Queries, Keys, Values
    - Computing Attention
5. Multi-Headed Attention
    - MHA Intuition
6. Transformer Architecture
7. Vision Transformers
    - Moving from text to images
    - Comparing transformers and CNNs

# Attention

**Italian**: Io la sto mangiando.

When reading a word in this sentence, what do I need to pay **attention** to.

# Attention

**Italian**: Io la sto mangiando.

When reading a word in this sentence, what do I need to pay **attention** to.

English is a **subject - verb - object** language

**English**: I am eating it.

# Attention

**Italian**: Io la sto mangiando.

When reading a word in this sentence, what do I need to pay **attention** to.

English is a **subject - verb - object** language

**English**: I am eating it.

**What is the subject doing?**

**Io** la sto mangiando.

# Attention

**Italian**: Io la sto mangiando.

When reading a word in this sentence, what do I need to pay **attention** to.

English is a **subject - verb - object** language

**English**: I am eating it.

**What is the subject doing?**

**Io** la sto mangiando.

**What is the verb operating on?**

Io la sto **mangiando**.

# Attention

How might we encode this mathematically using our word embeddings?

**Word embedding**

$$e_i$$

# Attention

How might we encode this mathematically using our word embeddings?

$t$

**Io** la sto mangiando.

**Word embedding**

$e_i$

# Attention

How might we encode this mathematically using our word embeddings?

$$\begin{array}{cccccc} & t & & & & \\ \alpha_{ti} = [ & 0.33 & 0.00 & 0.33 & 0.33 & ] \\ & \textbf{lo} & \text{la} & \text{sto} & \text{mangiando.} & \end{array}$$

**Attention weights**

**Word embedding**

$$\alpha_{ti} \quad e_i$$

# Attention

How might we encode this mathematically using our word embeddings?

$$
\begin{array}{ccccc}
 & t & & & \\
\alpha_{ti} = & [\; 0.33 & 0.00 & 0.33 & 0.33 \;] \\
 & \textbf{lo} & \text{la} & \text{sto} & \text{mangiando.}
\end{array}
$$

**Attention weights**

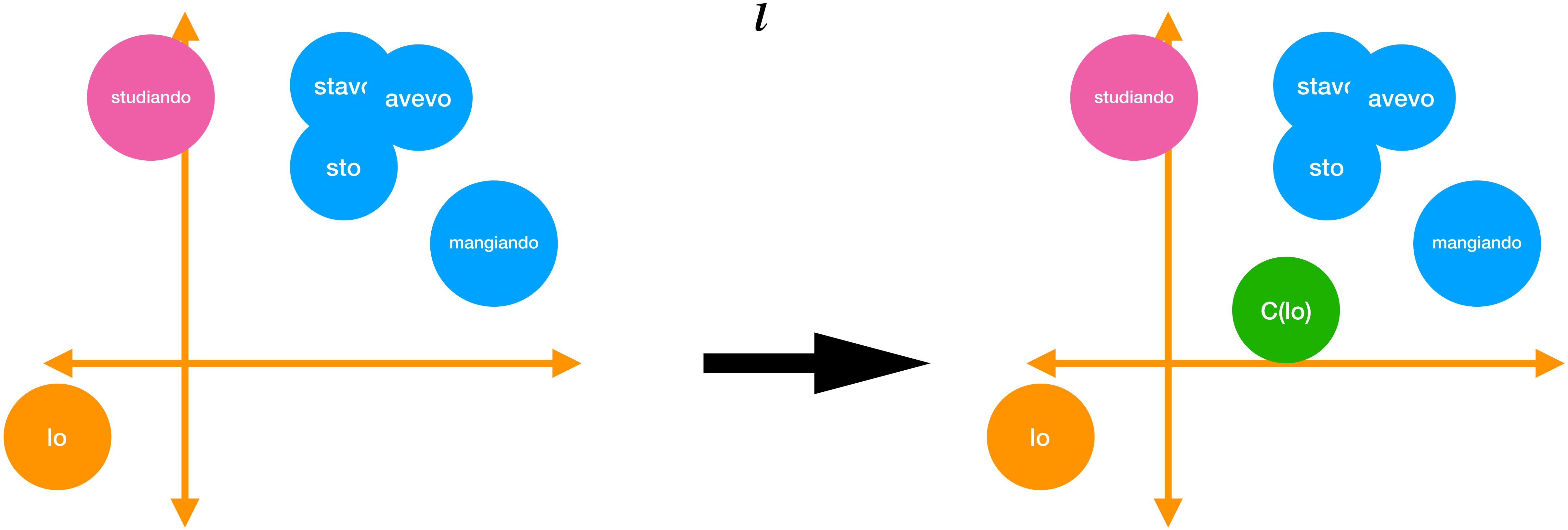
**Context vector**

**Word embedding**

$$c_t = \sum_{i}^{T} \alpha_{ti} \cdot e_i$$

# Attention

$$c_t = \sum_{i}^{T} \alpha_{ti} \cdot e_i$$




**Italian: Io** la sto mangiando.

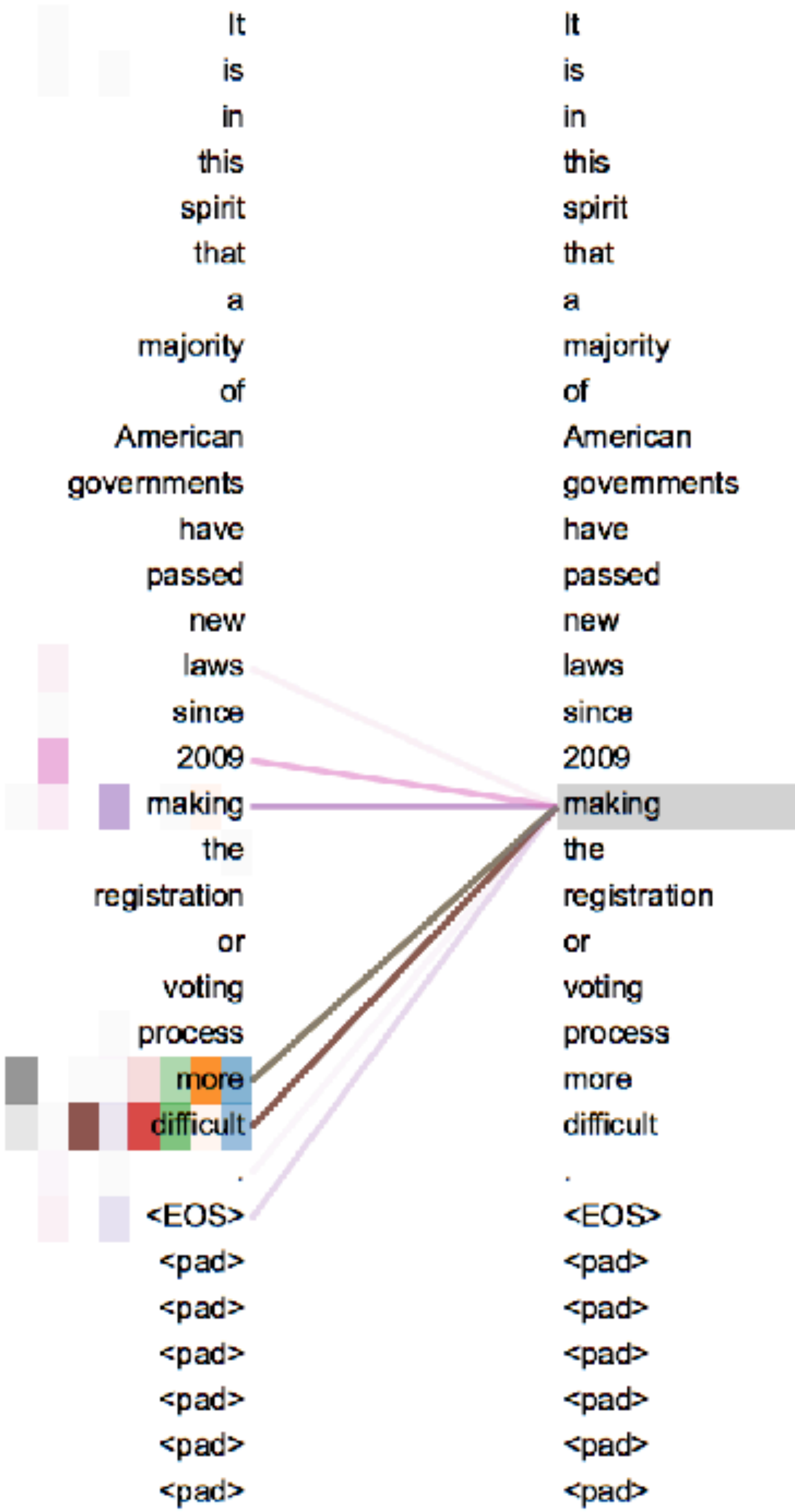
# Lecture Roadmap

1. Motivation
2. Word Embeddings
3. Attention
    - What is it, intuitively?
    - What is it, mathematically?
4. **Scalar Dot-Product Attention**
    - Why?
    - Queries, Keys, Values
    - Computing Attention
5. Multi-Headed Attention
    - MHA Intuition
6. Transformer Architecture
7. Vision Transformers
    - Moving from text to images
    - Comparing transformers and CNNs

# Why Scalar Dot-Product Attention?

**Let's look at a couple of motivating examples…**

From here on out, **'query'** will represent the word that we are encoding the context for.

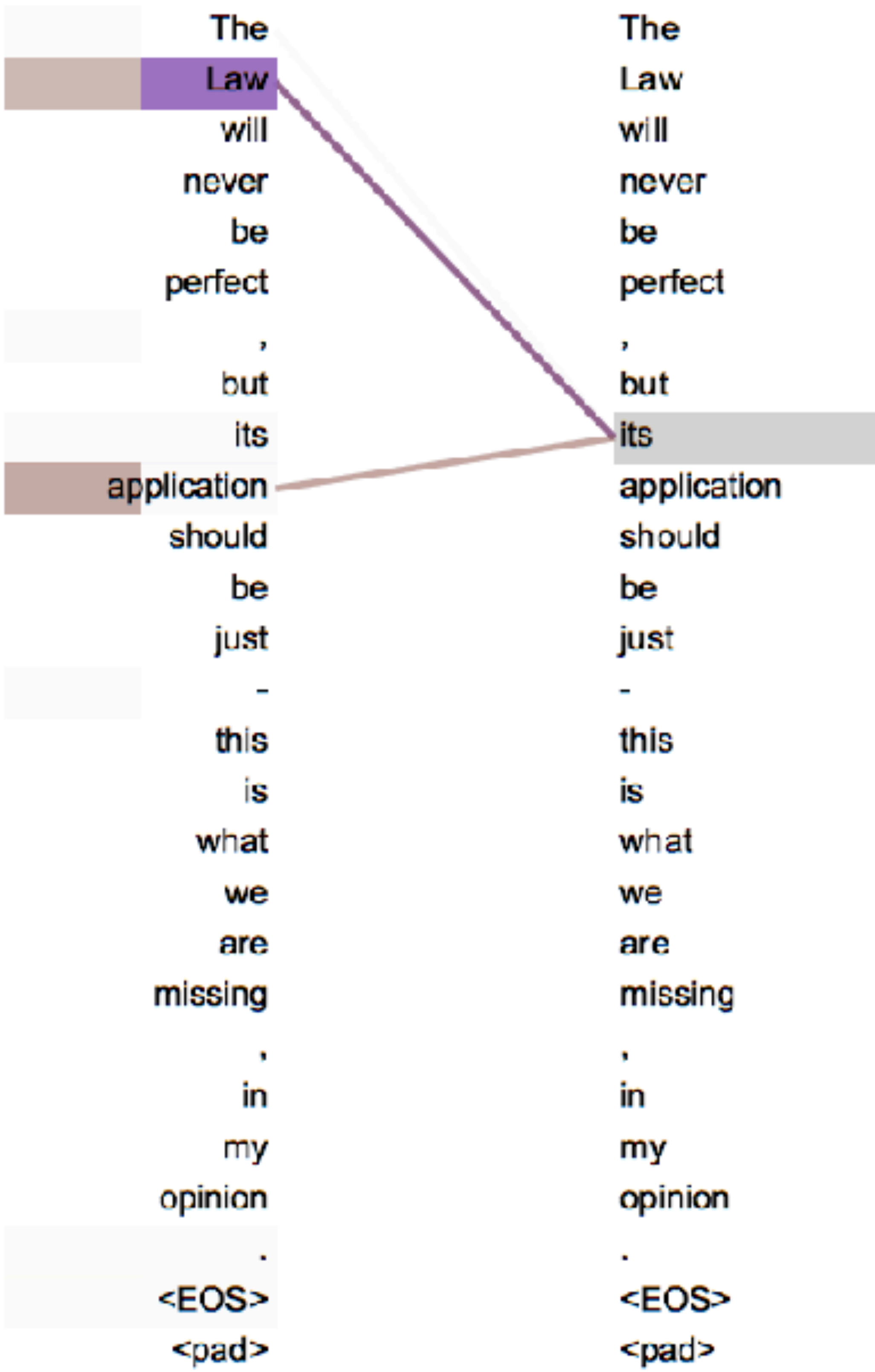



# Context for verbs

Here the **query** is *making*.

It puts most **attention** on itself and on the words *more difficult*, creating a **context** for the phrase *making [something] more difficult*.

# Anaphora Resolution




Anaphora Example:
Susan dropped the plate. **It** shattered loudly.

This layer resolves what words like “it” refer back to.

# Scalar Dot-Product Attention

**Each word in the sentence will have three representations.**

**Query, Key, Value.**

# Query, Key, Value Intuition

**Q, K, V are originally from retrieval systems (search).**

**In retrieval systems… like youtube search**

**Queries** - The sentences we type in to youtube to look for a video
**Keys** - The representations of the videos
**Values** - The videos of interest

# Query, Key, Value Intuition

**In sentences…**

**Queries** - Representations of the word of interest
**Keys** - Representations for all the words in the sentence
**Values** - The abstract, contextual representation of the words

# Query, Key, Value Intuition

**In sentences…**

**Queries** - Representations of the word of interest
**Keys** - Representations for all the words in the sentence
**Values** - The abstract semantic representation of the words

We want to determine how each **query**, relates to each **key** to compute an attention over the **values**.

# Query, Key, Value Intuition

**In sentences…**

**Queries** - Representations of the word of interest
**Keys** - Representations for all the words in the sentence
**Values** - The abstract semantic representation of the words

We want to determine how each **query**, relates to each **key** to compute an attention over the **values**.

**First, how do we compute these representations?**

# Query, Key, Value

**Start with the original word embedding.**

$$word_i \longrightarrow \boxed{\text{Embedding Function}} \longrightarrow z_i \in R^{d_z}$$

# Query, Key, Value

**Start with the original word embedding.**

$$word_i \longrightarrow \boxed{\text{Embedding Function}} \longrightarrow z_i \in R^{d_z}$$

**Linear projections of the original word embedding.**

$$z_i \cdot W^Q = q_i \qquad z_i \cdot W^K = k_i \qquad z_i \cdot W^V = v_i$$

**Learned**

# Query, Key, Value

$$W^Q \in R^{d_z \times d} \qquad W^K \in R^{d_z \times d} \qquad W^V \in R^{d_z \times d}$$

**W** usually **reduces** the dimensionality from the original embedding.
(Tall and skinny matrix)

# Query, Key, Value

**Collect all the linear projections into matrices.**

$$R^{N \times d}$$

$$Q = [q_1, q_2, \ldots, q_N]^T$$

$$K = [k_1, k_2, \ldots, k_N]^T$$

$$V = [v_1, v_2, \ldots, v_N]^T$$

# Scalar Dot Product Attention

Let's convert our intuitive ideas about queries, keys and values into an equation.

$$c_t = \sum_{i}^{T} \alpha_{ti} \cdot e_i$$

The original attention equation.

# Scalar Dot Product Attention

Let’s convert our intuitive ideas about queries, keys and values into an equation.

$$\boxed{V}$$

$$c_t = \sum_{i}^{T} \alpha_{ti} \cdot e_i$$

# Scalar Dot Product Attention

Let's convert our intuitive ideas about queries, keys and values into an equation.

$$\boxed{softmax(\frac{qK^T}{\sqrt{d}})}\boxed{V}$$

$$c_t = \sum_{i}^{T} \alpha_{ti} \cdot e_i$$

# Scalar Dot Product Attention

Let's convert our intuitive ideas about queries, keys and values into an equation.

**Note that K, V are not the same as** $W^K, W^V$

$$Attention(q, K, V) = \boxed{softmax(\frac{qK^T}{\sqrt{d}})}\boxed{V}$$

$$c_t = \sum_{i}^{T} \alpha_{ti} \cdot e_i$$

# Scalar Dot Product Attention

$$Attention(q, K, V) = softmax(\frac{qK^T}{\sqrt{d}})V$$

**Breaking it down…**

# Scalar Dot Product Attention

$$Attention(q, K, V) = softmax(\frac{qK^T}{\sqrt{d}})V$$

**Breaking it down…**

$$qK^T$$

Dot product - large when vectors are “similar”.
Encodes relevance of keys (other words) to a specific query (word).

so that $(1 \times d) \cdot (d \times N) = (1 \times N)$

# Scalar Dot Product Attention

$$Attention(q, K, V) = softmax(\frac{qK^T}{\sqrt{d}})V$$

$$qK^T$$

Dot product - large when vectors are “similar”.
Encodes relevance of keys (other words) to a specific query (word).

$$\frac{qK^T}{\sqrt{d}}$$

Scale the dot products down (to avoid vanishing gradient issues)

# Scalar Dot Product Attention

$$Attention(q, K, V) = softmax(\frac{qK^T}{\sqrt{d}})V$$

$$\frac{qK^T}{\sqrt{d}}$$

Scale the dot products down (to avoid vanishing gradient issues)

$$softmax(\frac{qK^T}{\sqrt{d}})$$

Transform the dot products into weights that sum to 1 (in each row of the output matrix)

# Scalar Dot Product Attention

$$Attention(q, K, V) = softmax(\frac{qK^T}{\sqrt{d}})V$$

$$softmax(\frac{qK^T}{\sqrt{d}})$$

Transform the dot products into weights that sum to 1 (in each row of the output matrix)

At this point, we have weights that we can apply to our Values representations. These weights dictate what Values we pay *attention* to.

# Scalar Dot Product Attention

$$Attention(q, K, V) = softmax(\frac{qK^T}{\sqrt{d}})V$$

$$softmax(\frac{qK^T}{\sqrt{d}})$$

Transform the dot products into weights that sum to 1 (in each row of the output matrix)

$$softmax(\frac{qK^T}{\sqrt{d}})V$$

We now have a representation that abstractly represents the ***context*** with which we read **each** query word.

$$(1 \times N) \times (N \times d) = (1 \times d)$$

# Scalar Dot Product Attention

We can compute a batch all at once by using a matrix of queries.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V$$

$$(N \times N) \times (N \times d) = (N \times d)$$

# Why three different representations?

**Main answer:** It worked better, other types of attentions have been explored, this worked the best.

# Why three different representations?

**Why can't we fold queries and keys into one representation?**

****These are opinions. StackOverflow has some discussion about this.**

# Why three different representations?

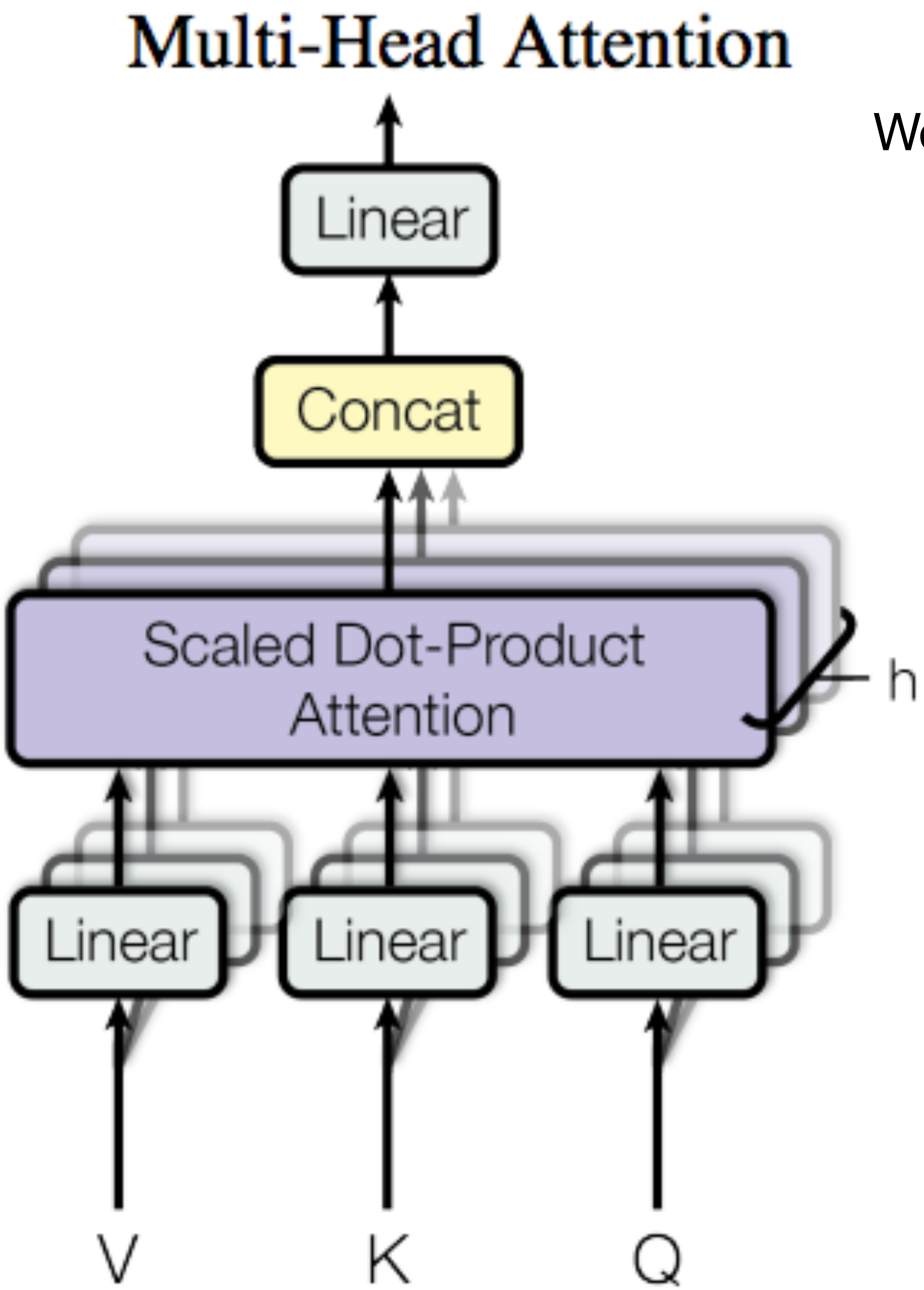
**Why can't we fold keys and values into one representation?**

****StackOverflow has some discussion about this.**

# Lecture Roadmap

1. Motivation
2. Word Embeddings
3. Attention
    - What is it, intuitively?
    - What is it, mathematically?
4. Scalar Dot-Product Attention
    - Why?
    - Queries, Keys, Values
    - Computing Attention
5. **Multi-Headed Attention**
    - MHA Intuition
6. Transformer Architecture
7. Vision Transformers
    - Moving from text to images
    - Comparing transformers and CNNs

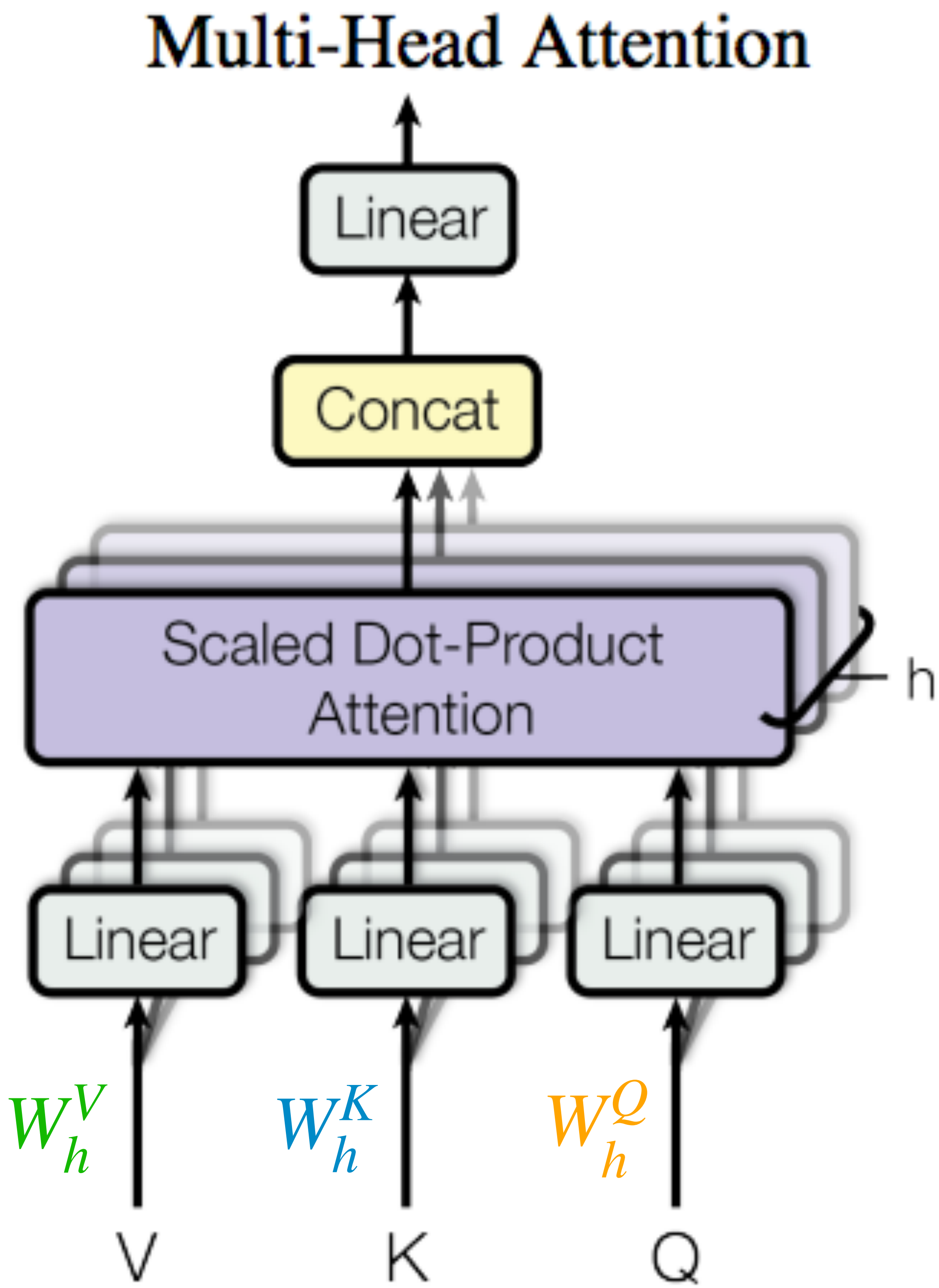
# Multi-Head Attention




We are learning a different transformation per head.

$W_h^V$ $W_h^K$ $W_h^Q$

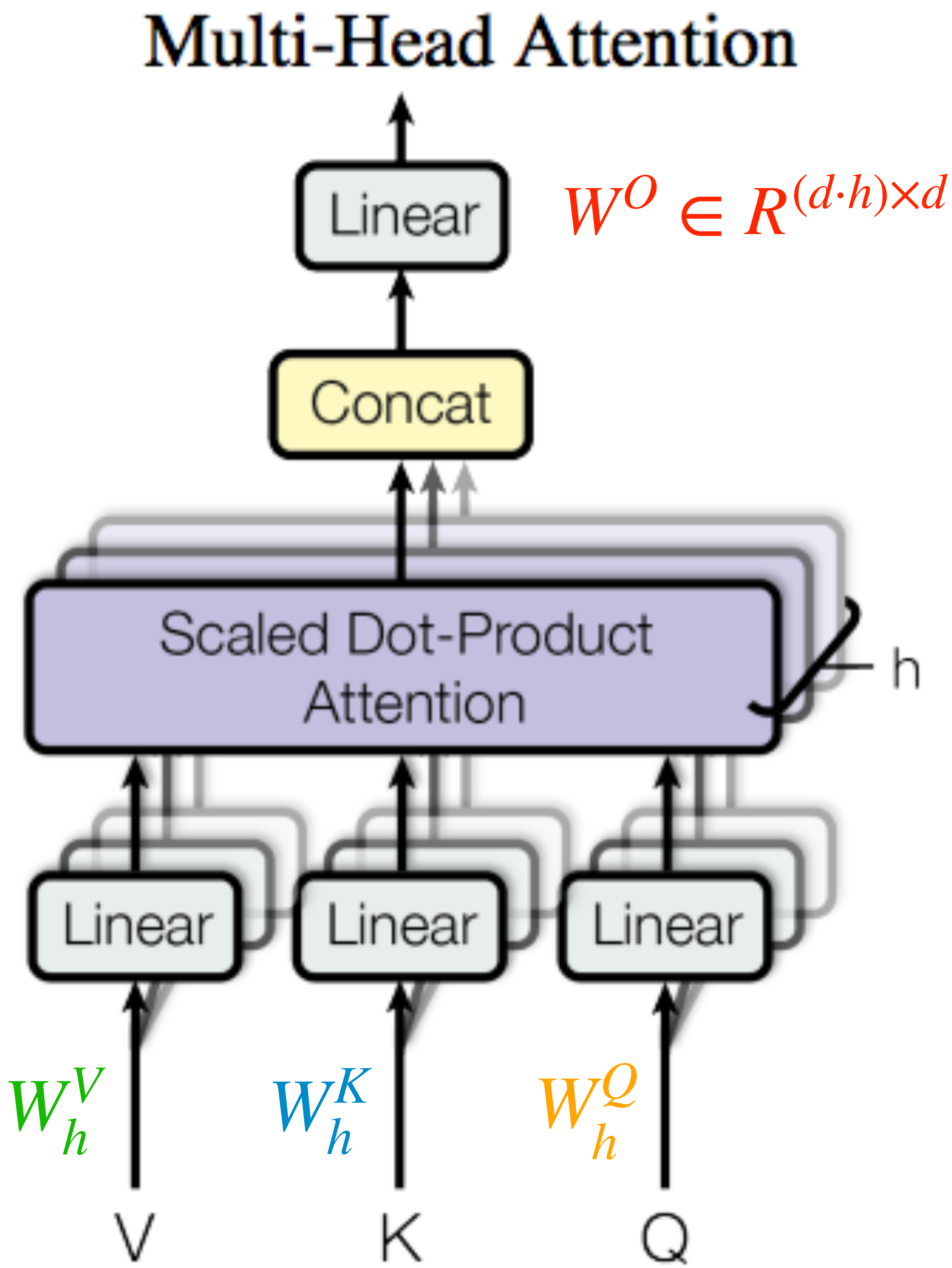
# Multi-Head Attention




$(N \times dh)$

Concatenate output of SDPA layers

# Multi-Head Attention




$$(N \times d) = (N \times dh) \cdot (dh \times d)$$

$$MHA = ConcatVec \cdot W^O$$

$$(N \times dh)$$

Concatenate output of SDPA layers

# MHA Intuition

**Why multiple heads?**

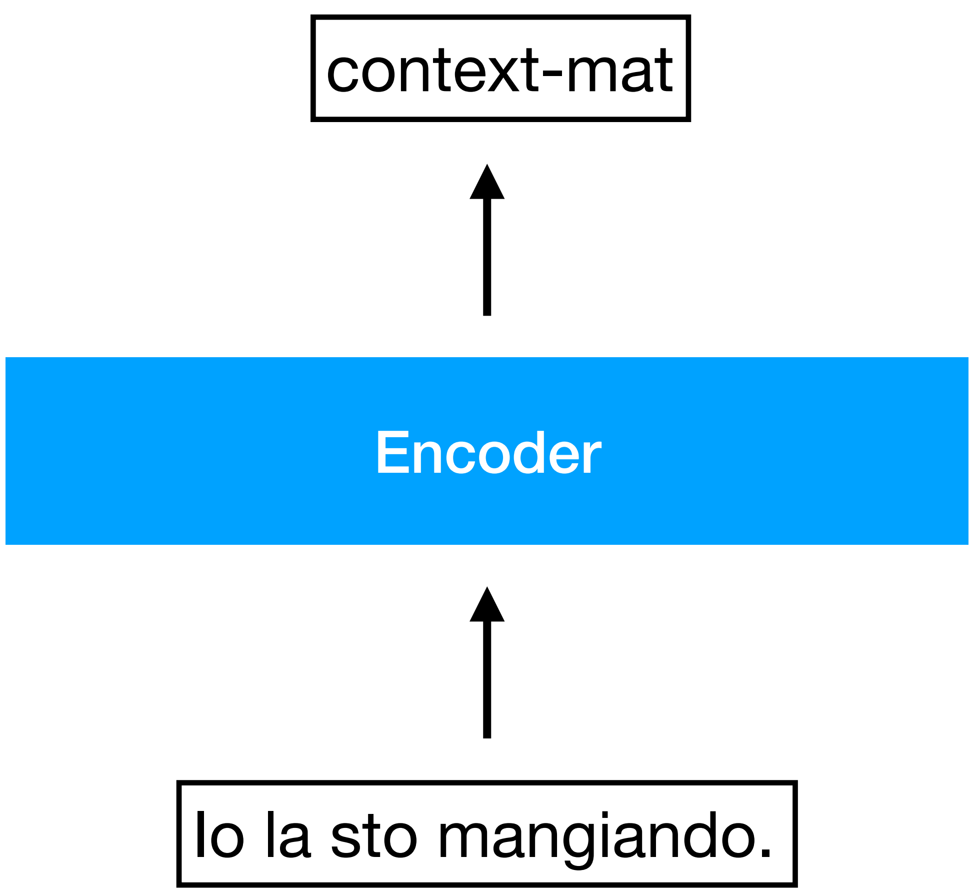
*It worked better in their paper.*

*The real answer is more nuanced and has to do with training stability.*
https://arxiv.org/pdf/2106.09650.pdf

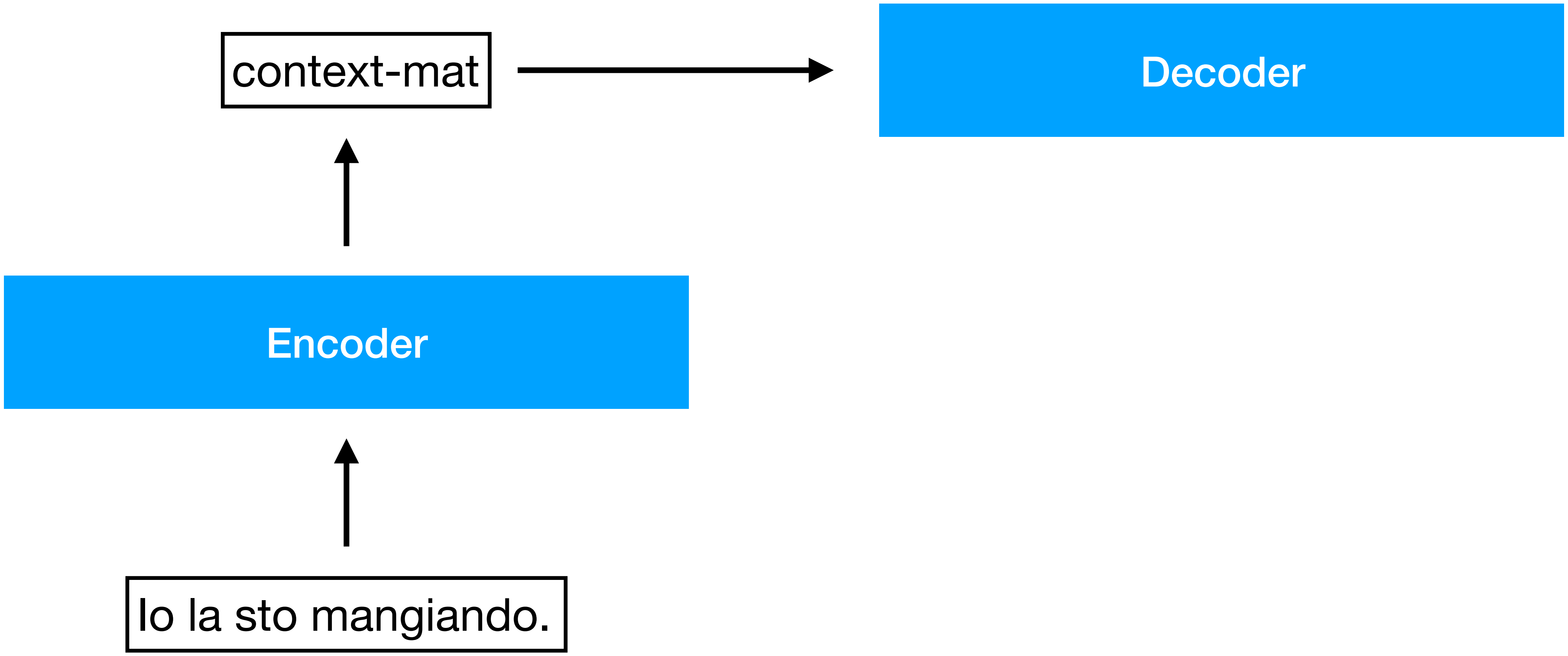
# Lecture Roadmap

1. Motivation
2. Word Embeddings
3. Attention
    - What is it, intuitively?
    - What is it, mathematically?
4. Scalar Dot-Product Attention
    - Why?
    - Queries, Keys, Values
    - Computing Attention
5. Multi-Headed Attention
    - MHA Intuition
6. **Transformer Architecture**
7. Vision Transformers
    - Moving from text to images
    - Comparing transformers and CNNs

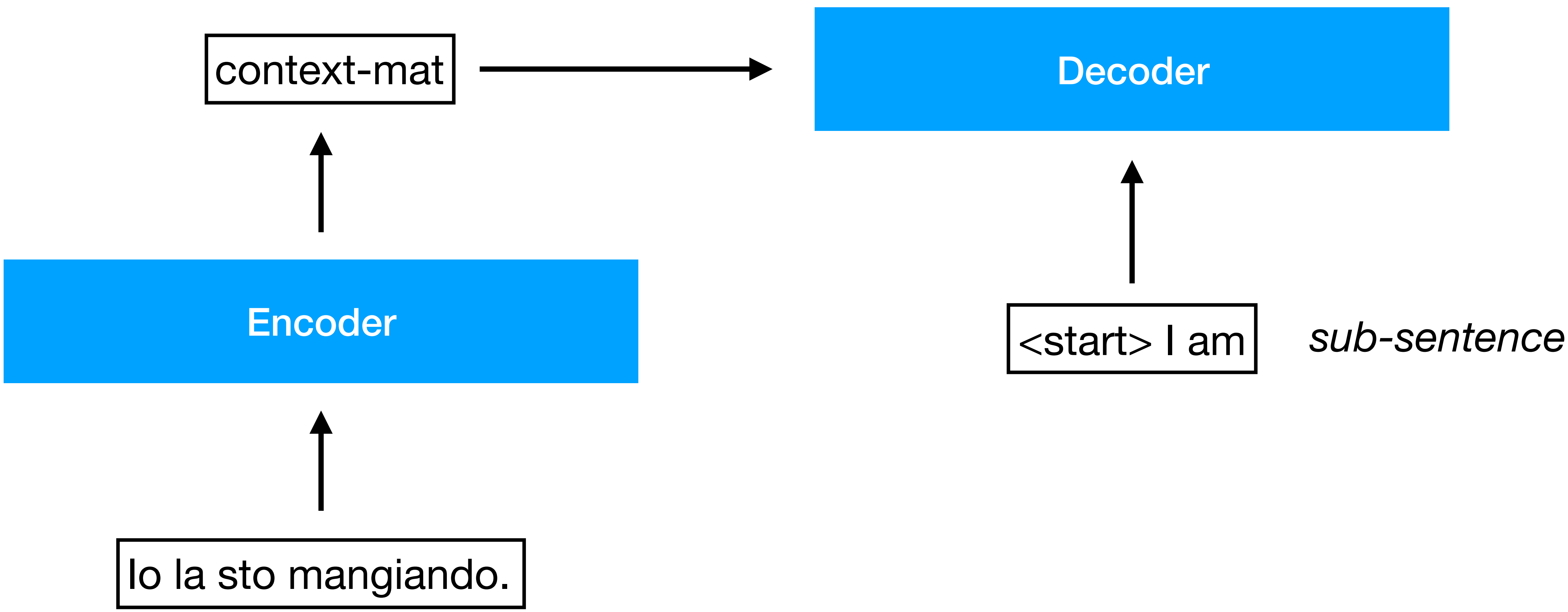
# Transformer Architecture



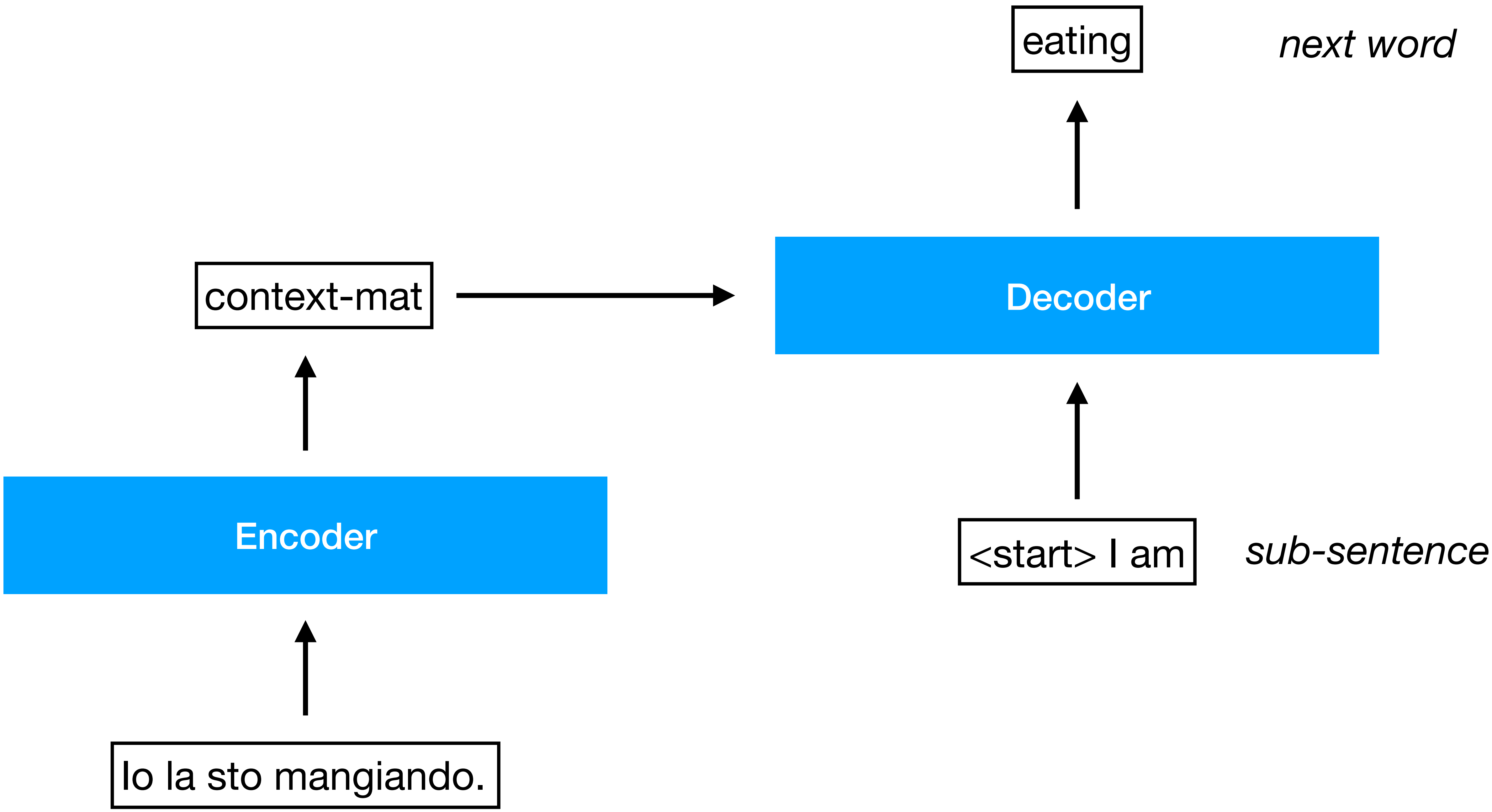
# Transformer Architecture



# Transformer Architecture

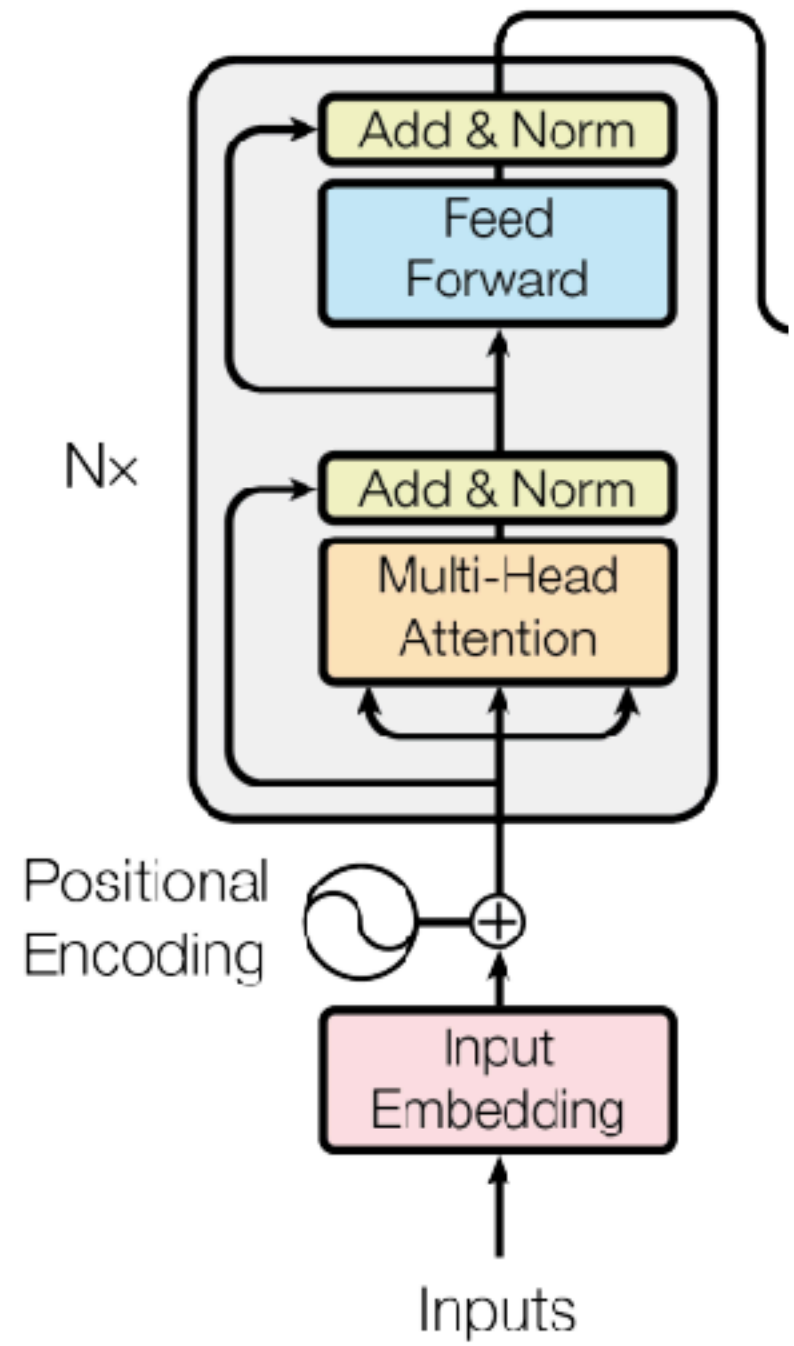


# Transformer Architecture



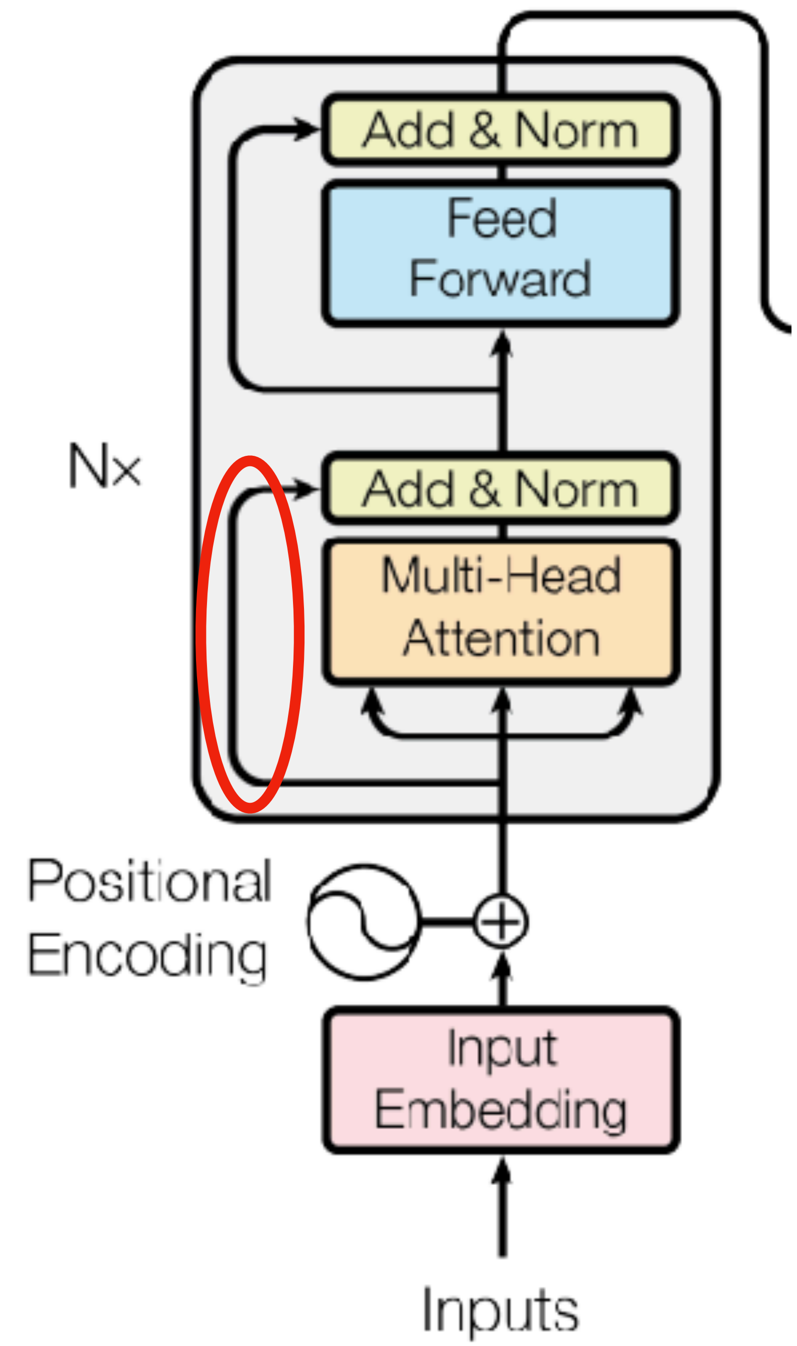
# Transformer Encoder

Encoder



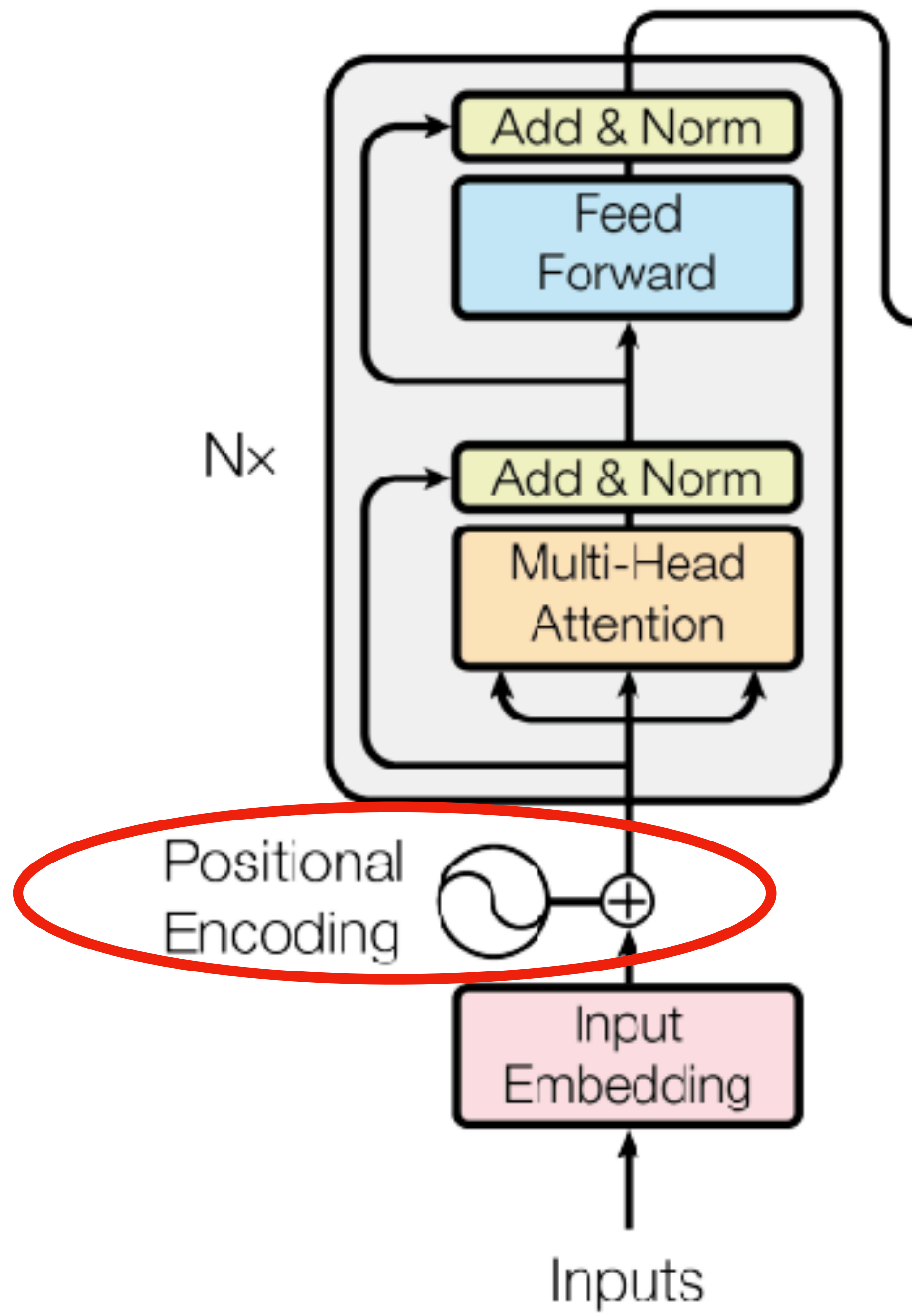
# Transformer Encoder

Encoder



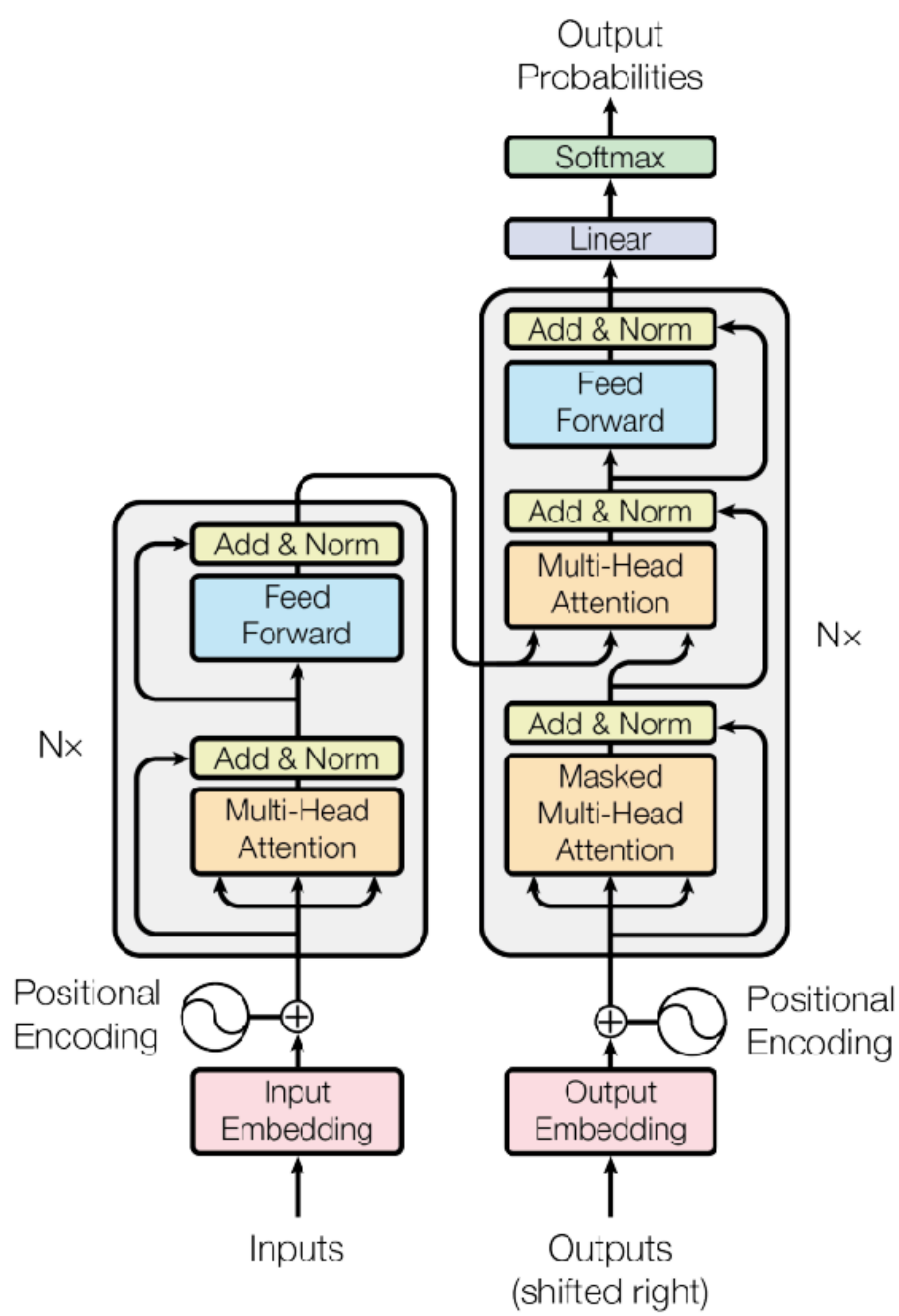
# Transformer Encoder

Encoder



# Transformer Architecture

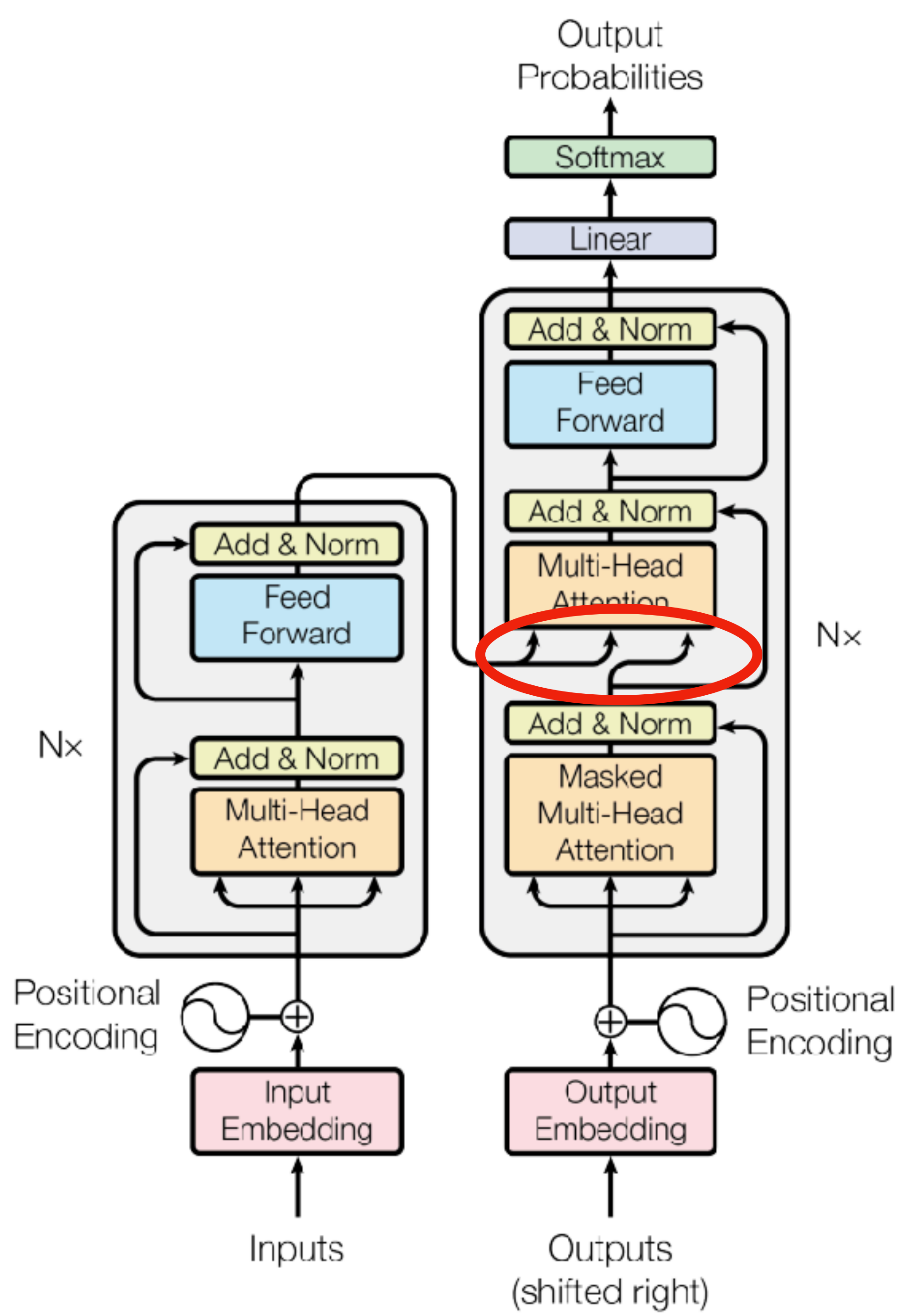
Encoder




Decoder

# Transformer Architecture

Encoder




Decoder

# Questions?

1. Motivation
2. Word Embeddings
3. Attention
    - What is it, intuitively?
    - What is it, mathematically?
4. Scalar Dot-Product Attention
    - Why?
    - Queries, Keys, Values
    - Computing Attention
5. Multi-Headed Attention
    - MHA Intuition
6. Transformer Architecture

# Lecture Roadmap

1. Motivation
2. Word Embeddings
3. Attention
    - What is it, intuitively?
    - What is it, mathematically?
4. Scalar Dot-Product Attention
    - Why?
    - Queries, Keys, Values
    - Computing Attention
5. Multi-Headed Attention
    - MHA Intuition
6. Transformer Architecture
7. **Vision Transformers**
    - Moving from text to images
    - Comparing transformers and CNNs

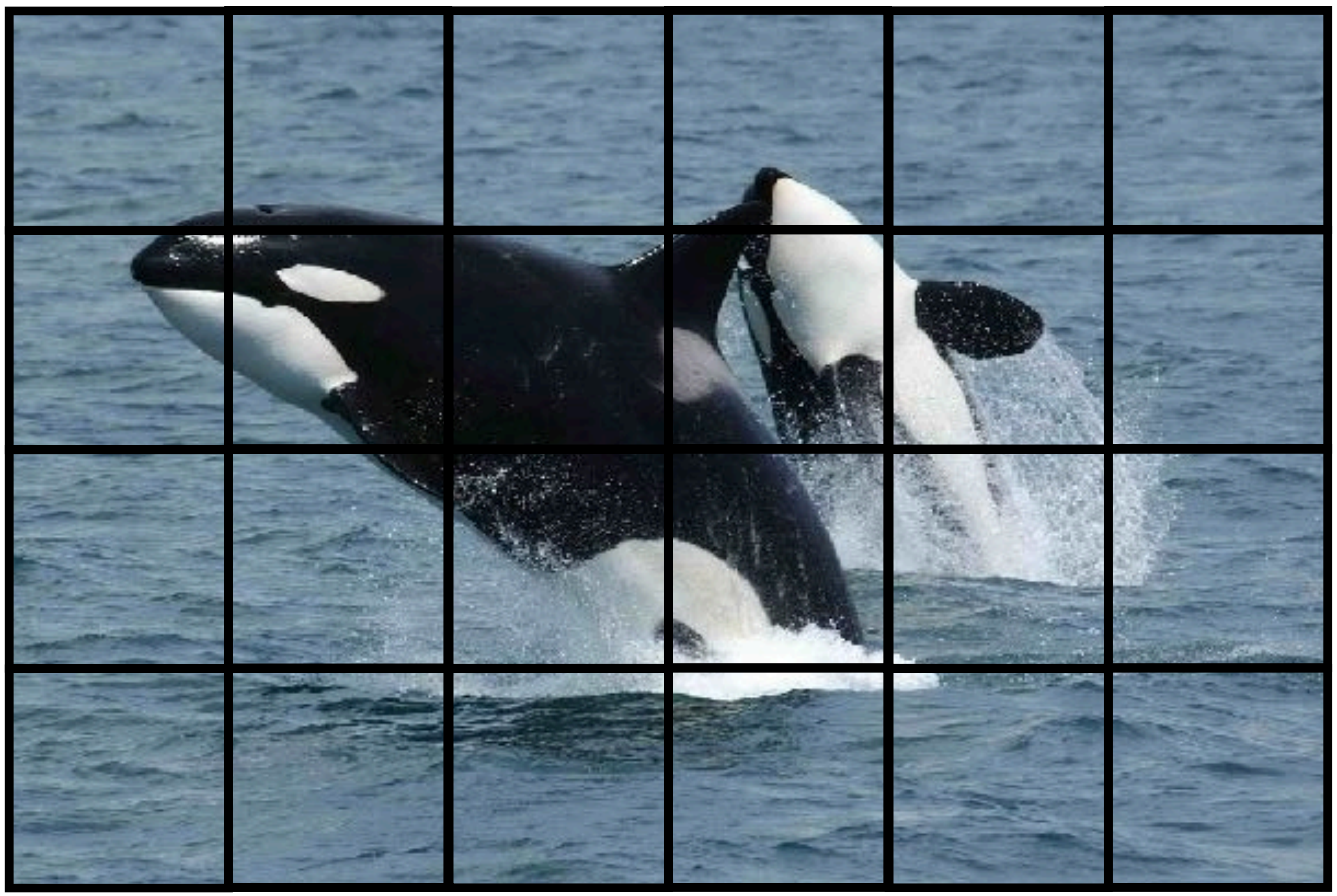
**What is the most naive way to generalize the transformer encoder for images?**

*What is a "word" in an image?*

**What is a more realistic, efficient way to generalize the transformer encoder for images?**

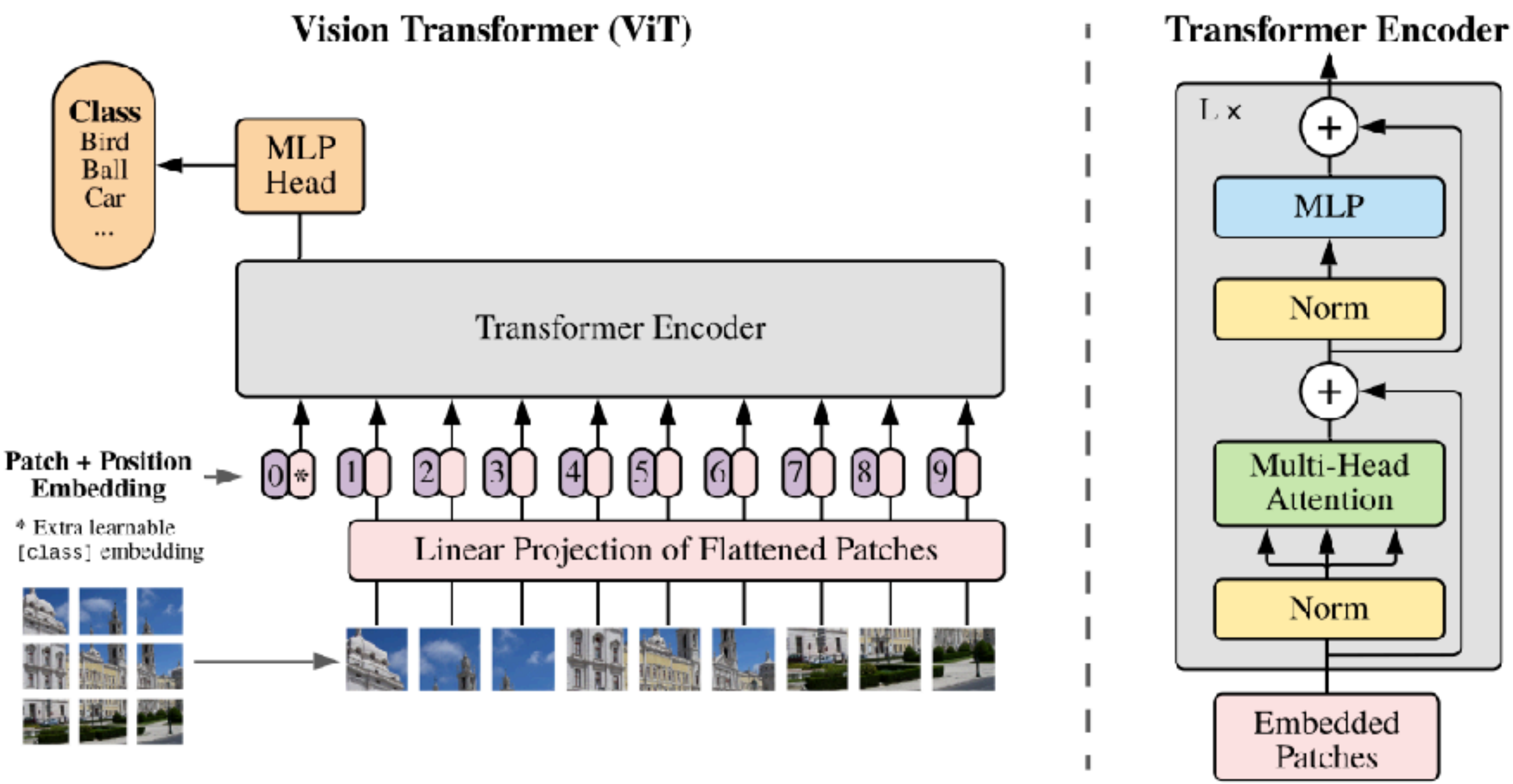
## What is a more realistic, efficient way to generalize the transformer encoder for images?

## Patches

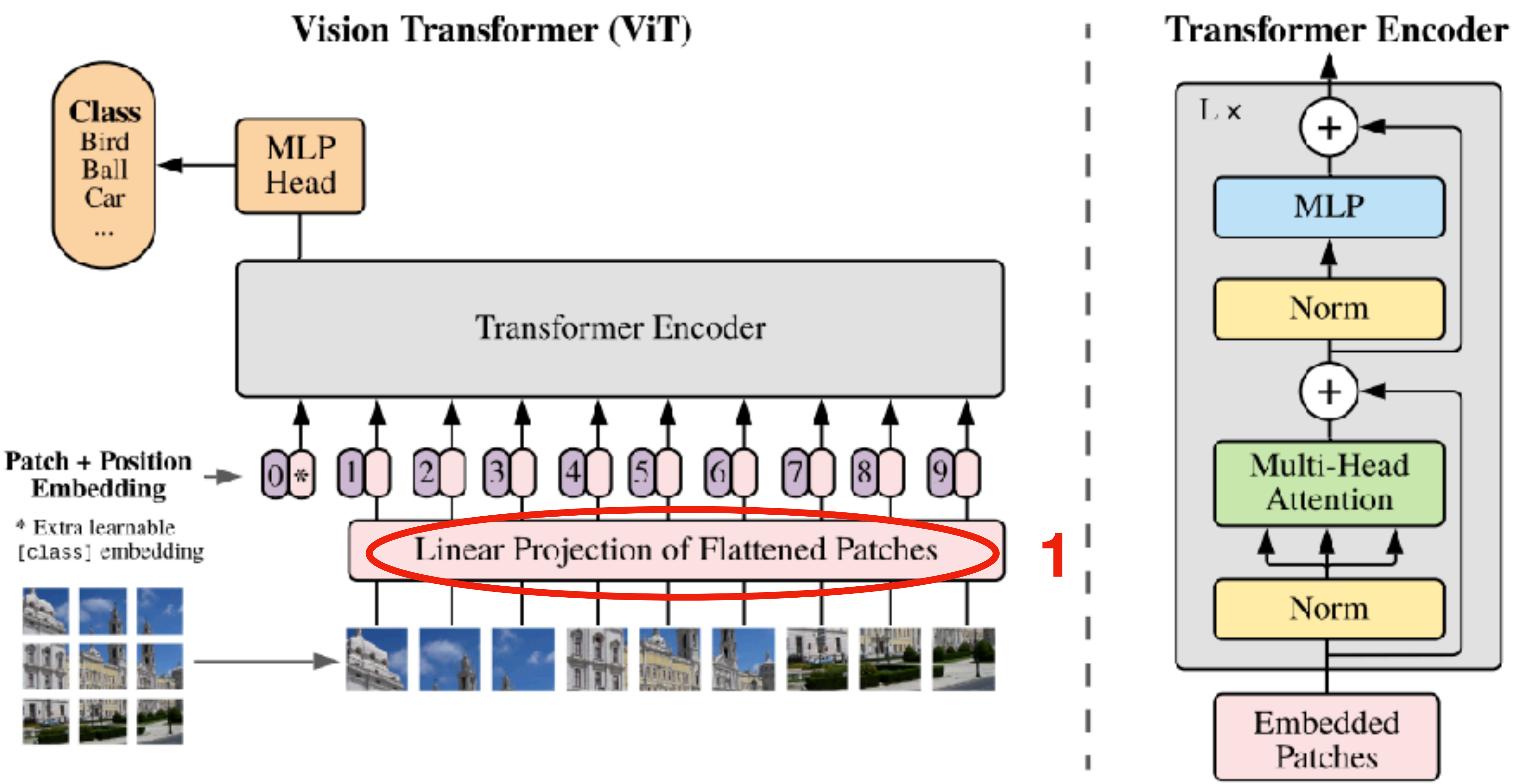


# Vision Transformers (ViT)

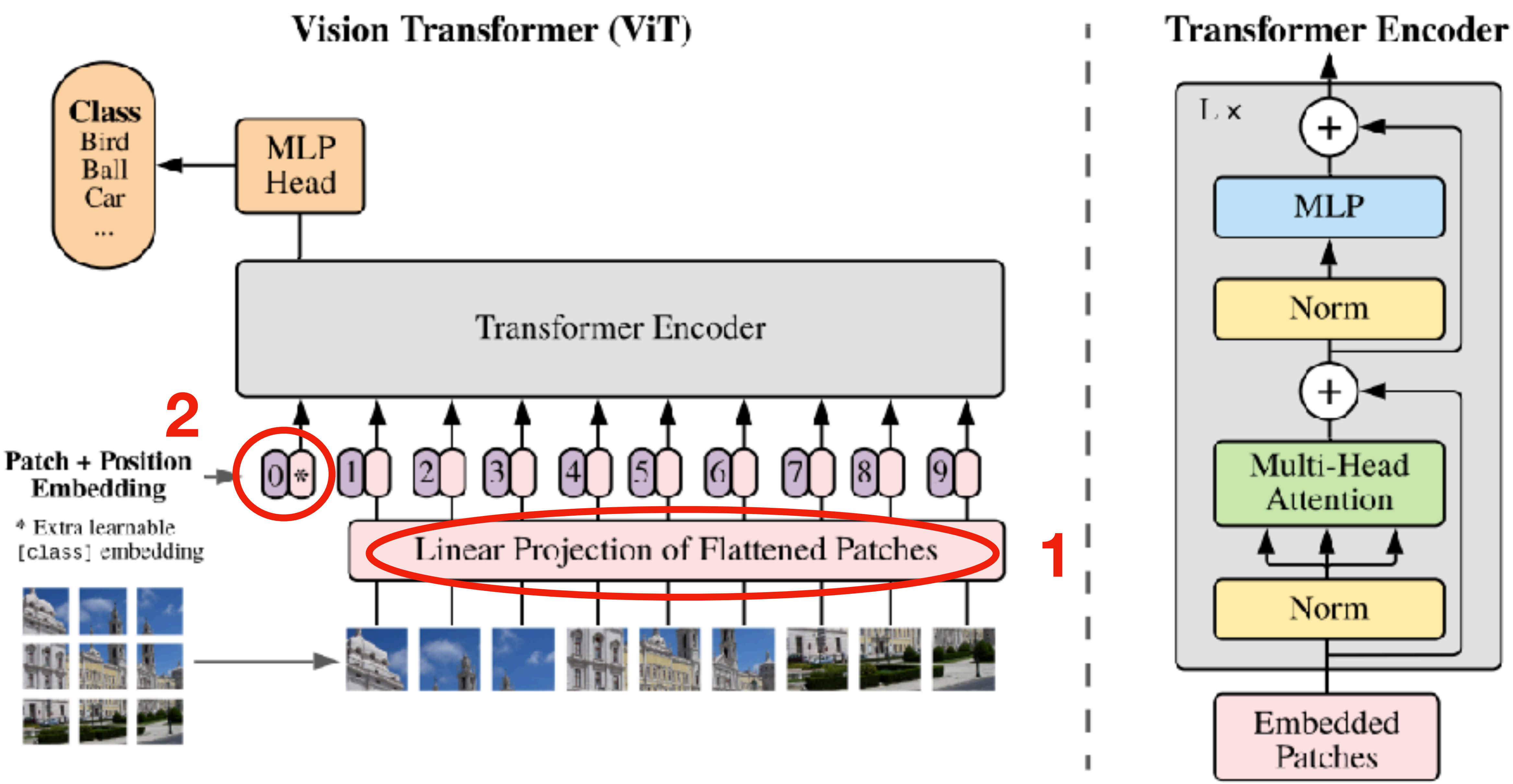
What's new?



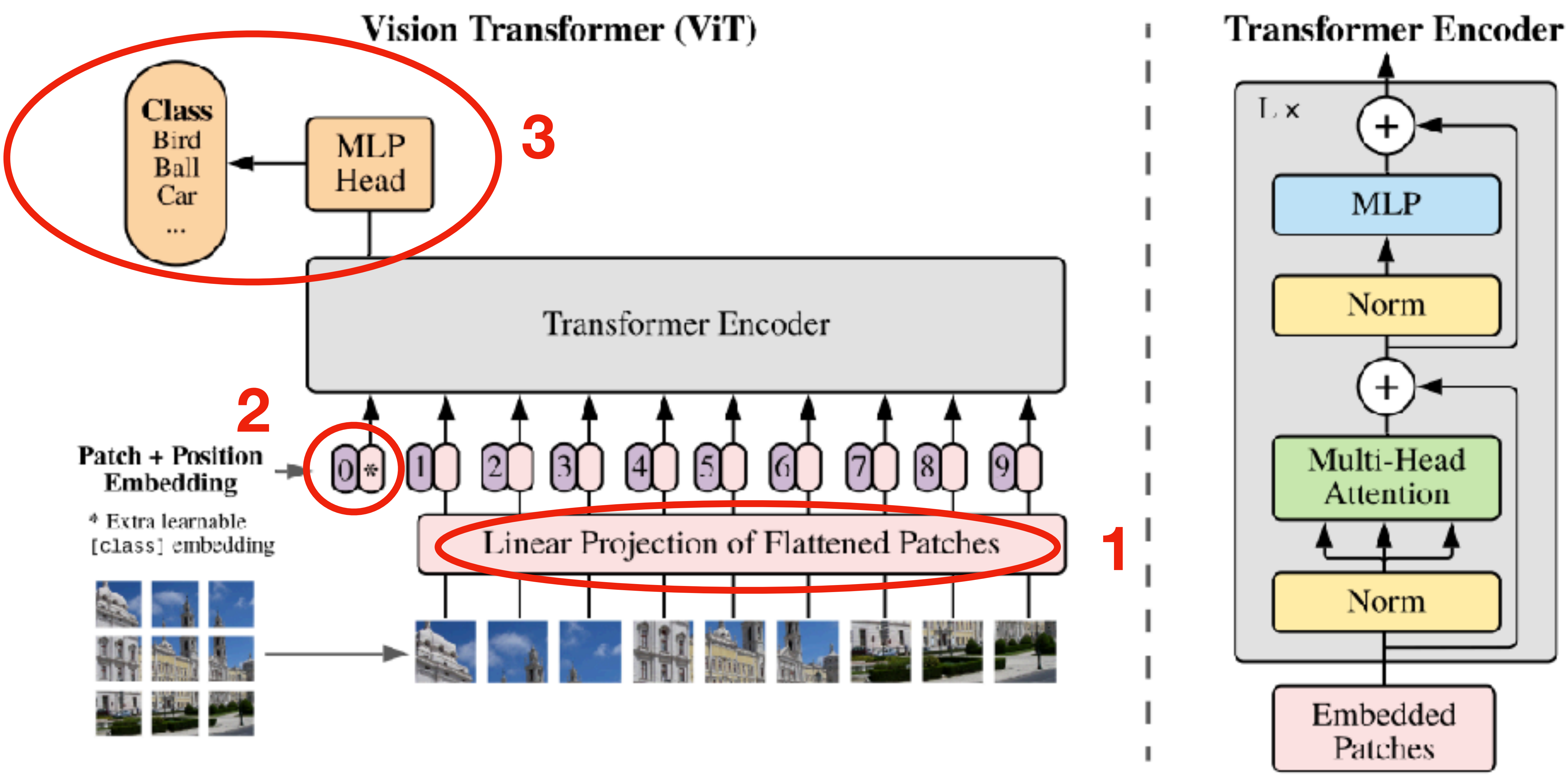
# Vision Transformers (ViT)



# Vision Transformers (ViT)



# Vision Transformers (ViT)



# 1 Linear Projections of Flattened Patches

$P$ 

$P$

# 1 Linear Projections of Flattened Patches

$P$

$P$

Flatten

Vectorized Image Patch

# 1 Linear Projections of Flattened Patches

$P$

$P$

Flatten

Vectorized Image Patch

$$x_p \in R^{P^2}$$

# 1 Linear Projections of Flattened Patches

$P$

$P$

Flatten

Vectorized Image Patch

$x_p \in R^{P^2}$

$$x_p \cdot E = \hat{x}_p$$

**The matrix $E$ is learned**

# 1 Linear Projections of Flattened Patches

$P$

$P$

Flatten

Vectorized Image Patch

$x_p \in R^{P^2}$

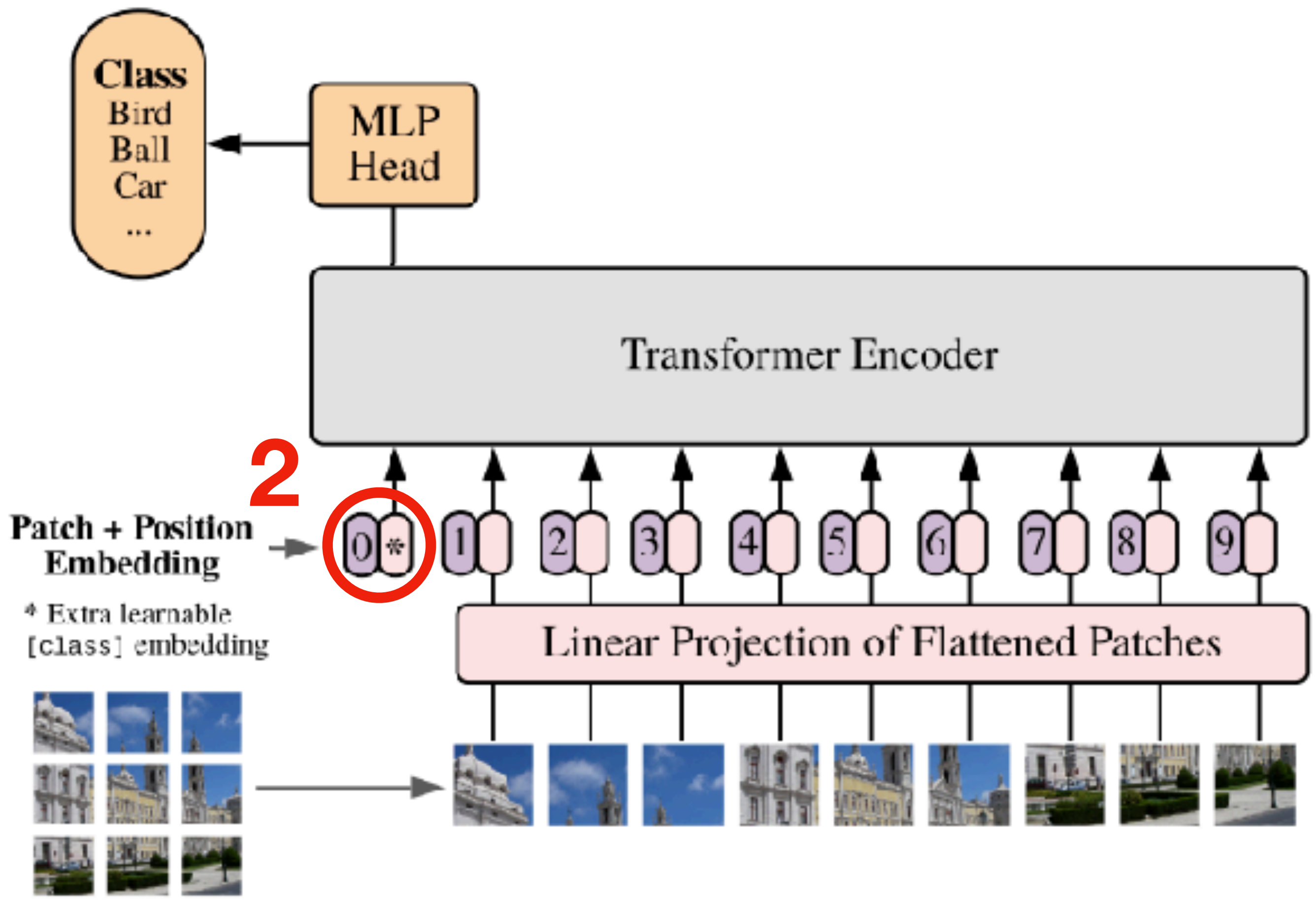
$$x_p \cdot E = \hat{x}_p$$

$$E \in R^{P^2 \times D}, \hat{x}_p \in R^D$$

**The matrix $E$ is learned**

# Extra learnable class embedding

Recall from multi-head attention, the output is $(N \times d)$

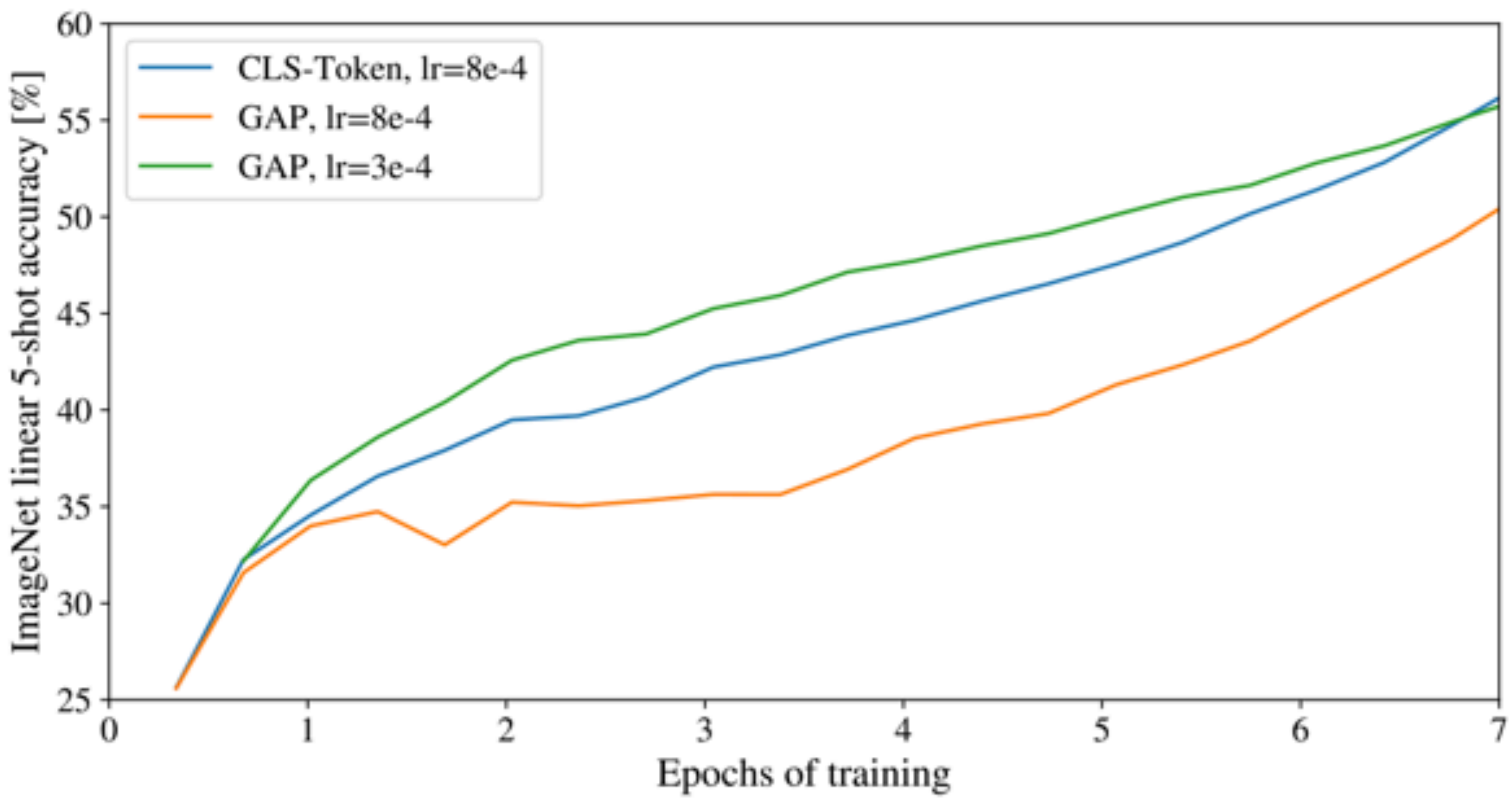



The extra learnable class embedding is the ***query*** that stores the ***context*** representation that is classified by the MLP (3).

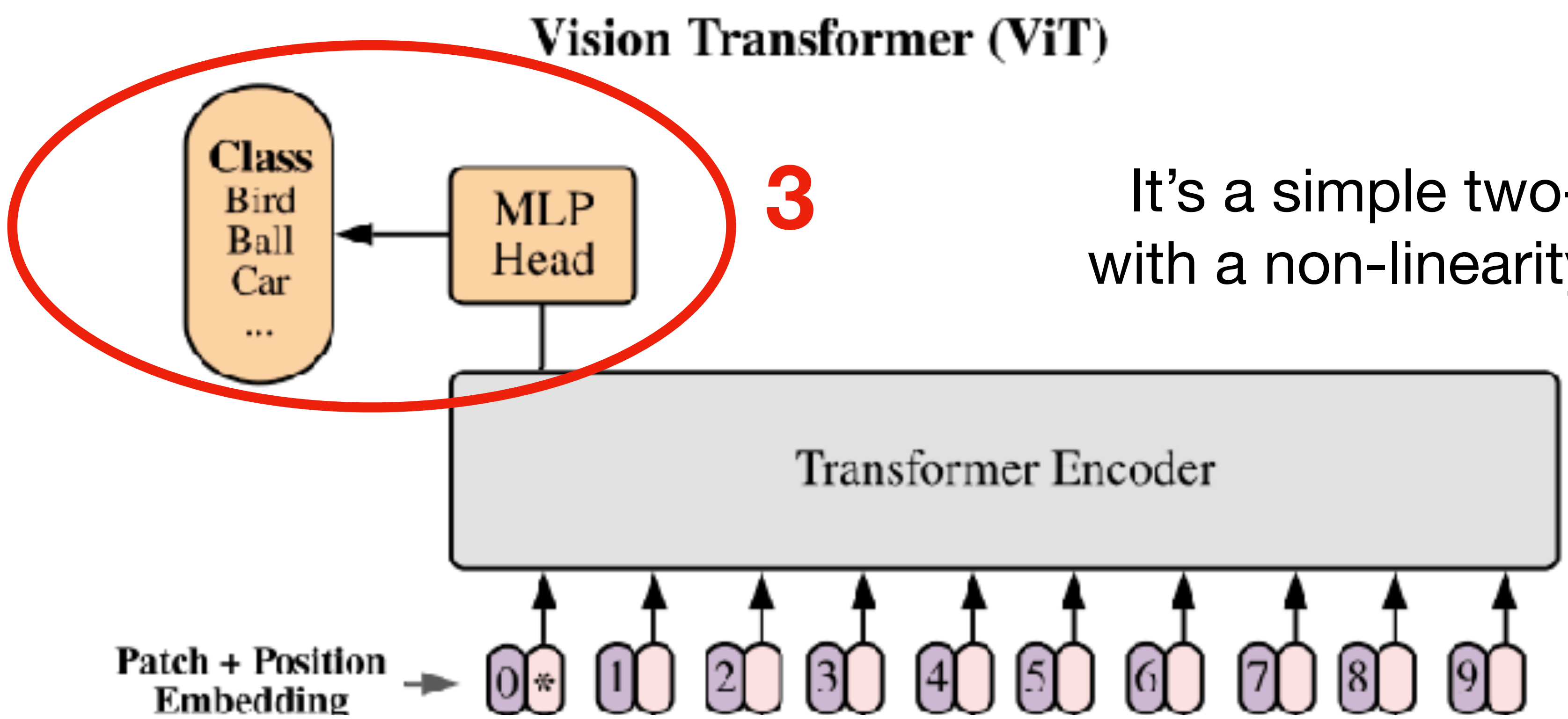
# Extra learnable class embedding

**Do we need this?**

GAP - global average pooling only
CLS-Token - include a class token in input



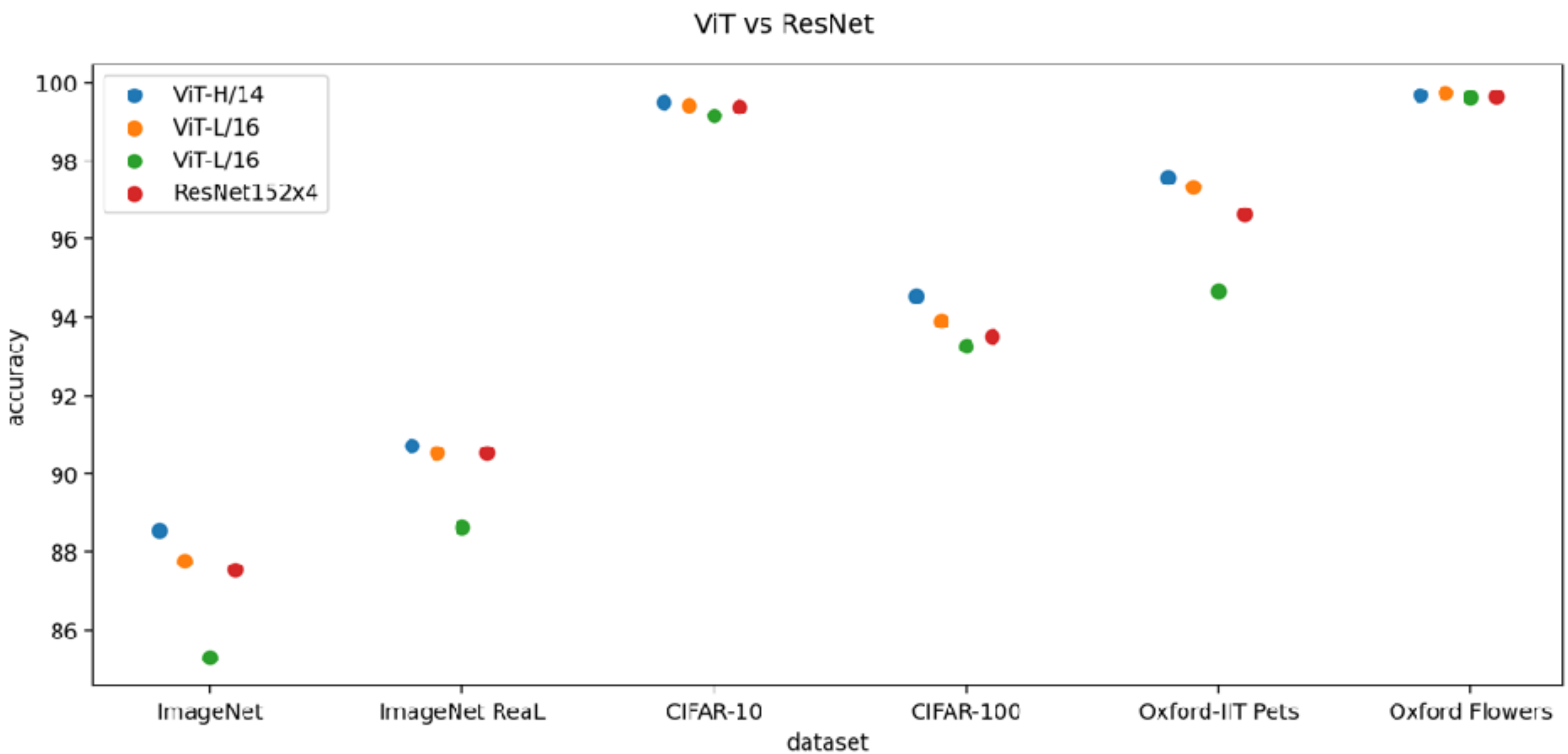
# Multi-layer Perceptron (MLP) head




It's a simple two-layer network with a non-linearity on the output.

Takes output of the encoder at the position of the class-token and predicts a class for the image.

# ViT (2021) Performance




**Slightly outperforms the ResNet152 based model.**

# Some interesting questions…

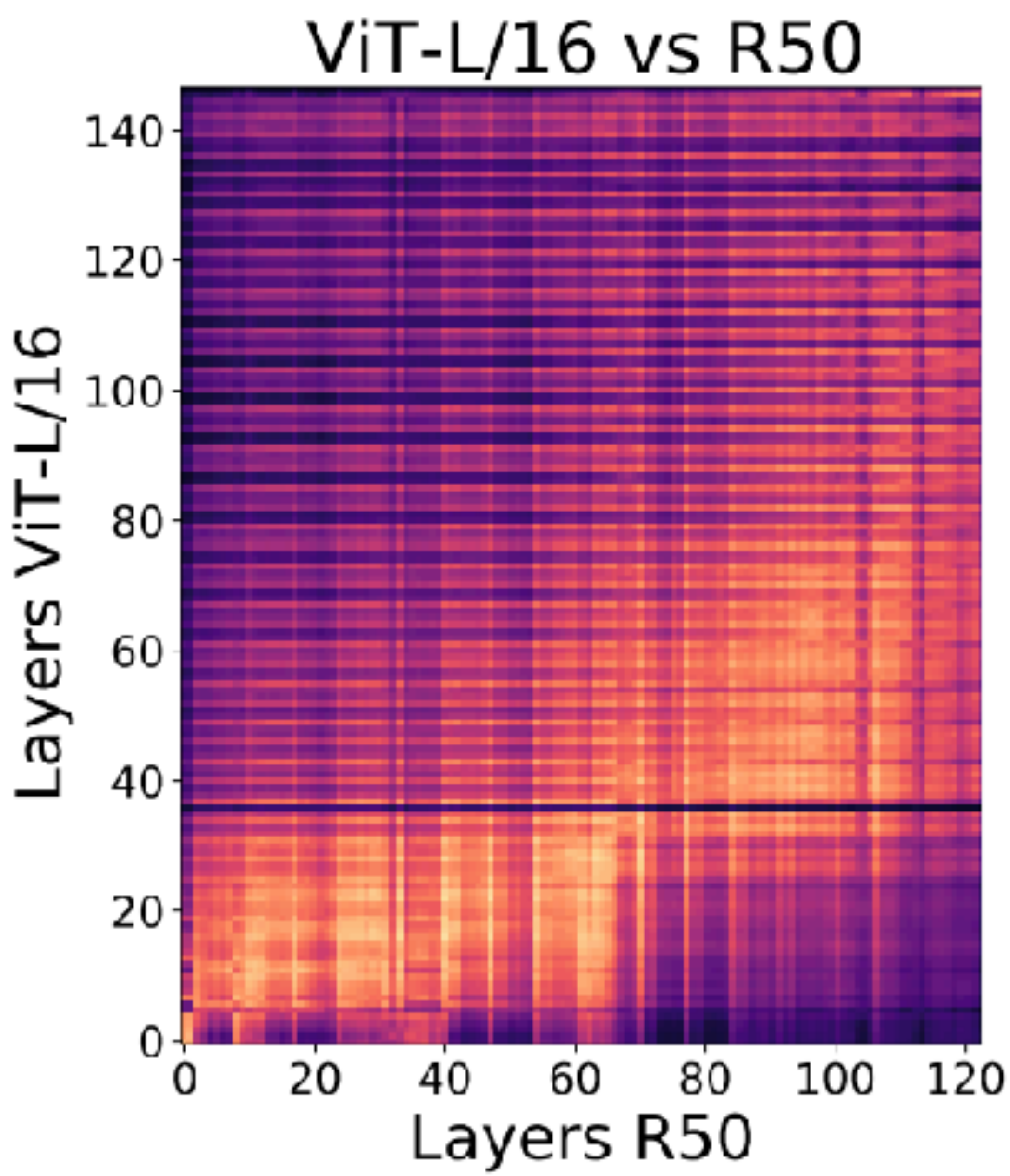
# What do the learned attention maps look like?




Vaswani et al. (2017)

# How does the way vision transformers “see” differ from CNNs?

## Representational Similarity



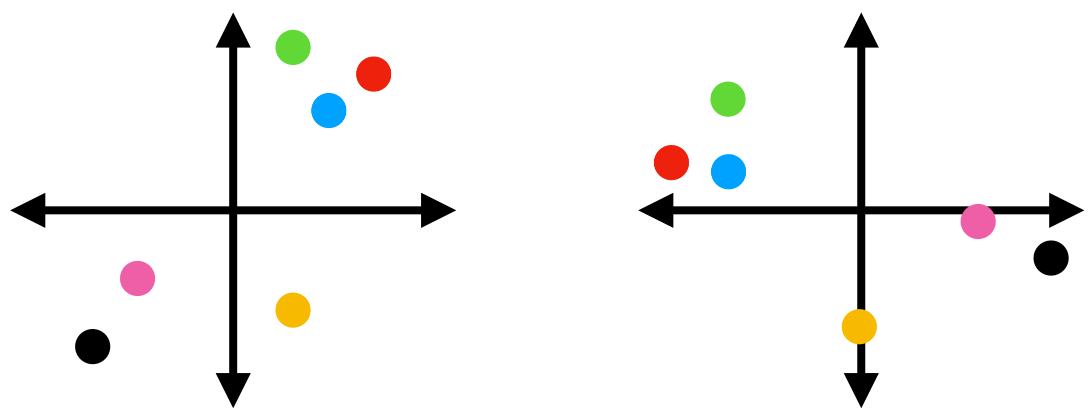

ViT uses a different method to compute low-level representations.

**Maithra et al. (2017)**

# Sub-question: how to compute representational similarity?

## Centered Kernel Alignment (CKA)

High level: Compute a **similarity** between the **similarities** in two different layers.



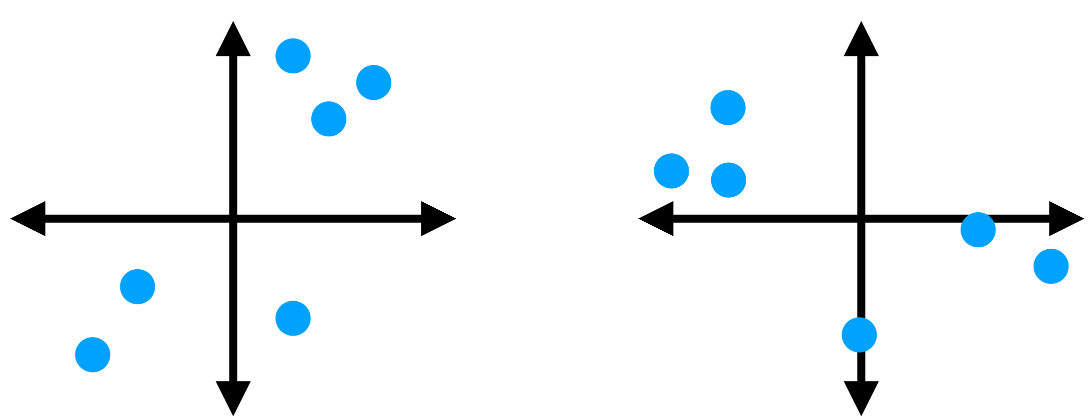
**These are effectively the same representation, but directly comparing them won't work.**

# Sub-question: how to compute representational similarity?

## Centered Kernel Alignment (CKA)

**High level: Compute a similarity between the similarities in two different layers.**

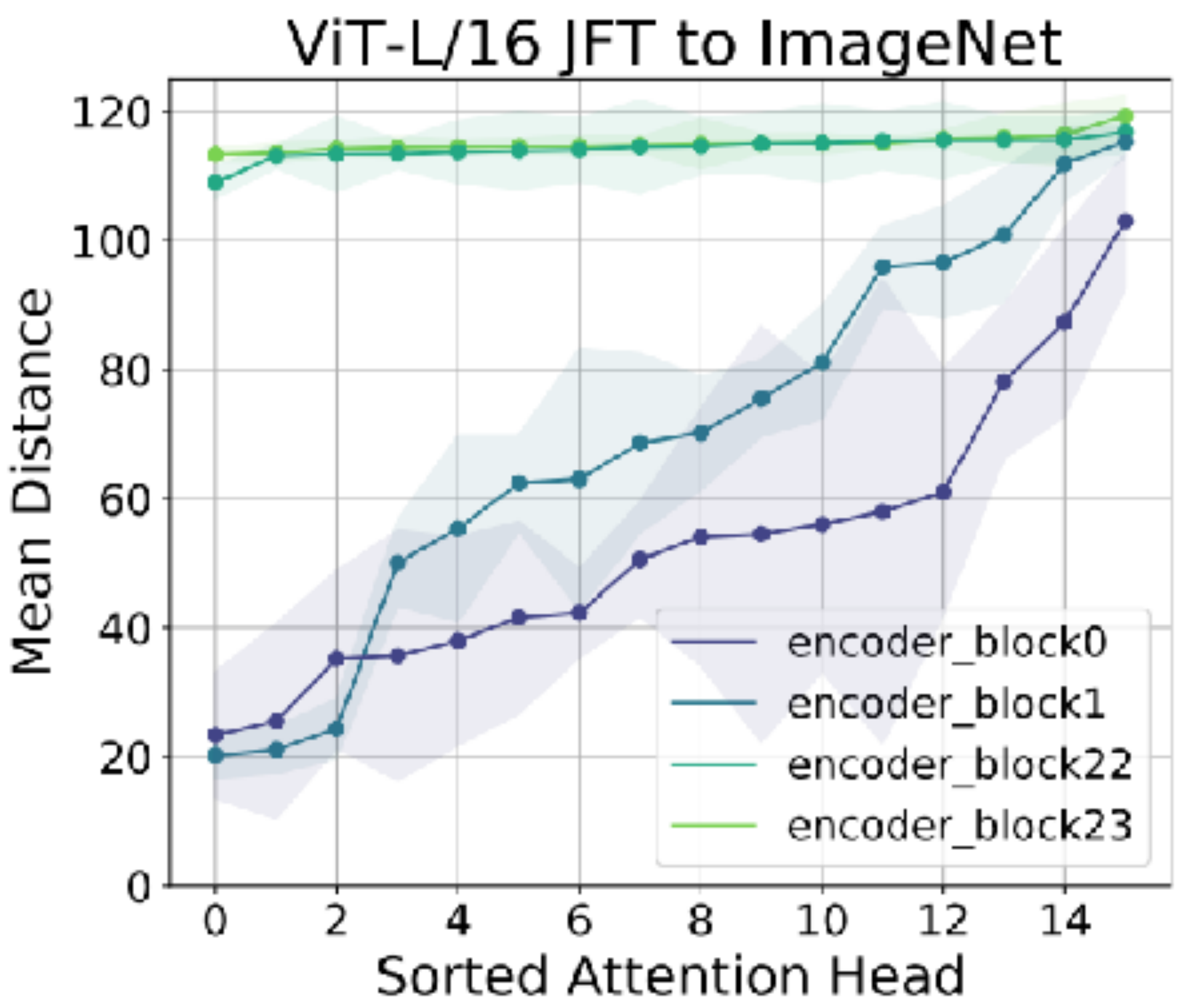
1. Generate the embeddings at two specific layers for the same set of data-points.
2. Compute the Gram matrices (measure the similarity between points in the same representation)
3. Compute a similarity between the Gram matrices



**These are effectively the same representation, but directly comparing them won't work.**

# How does the way vision transformers "see" differ from CNNs?

## Local and Global Information




ViT early layers (block0, block1) encode both local and global relationships.

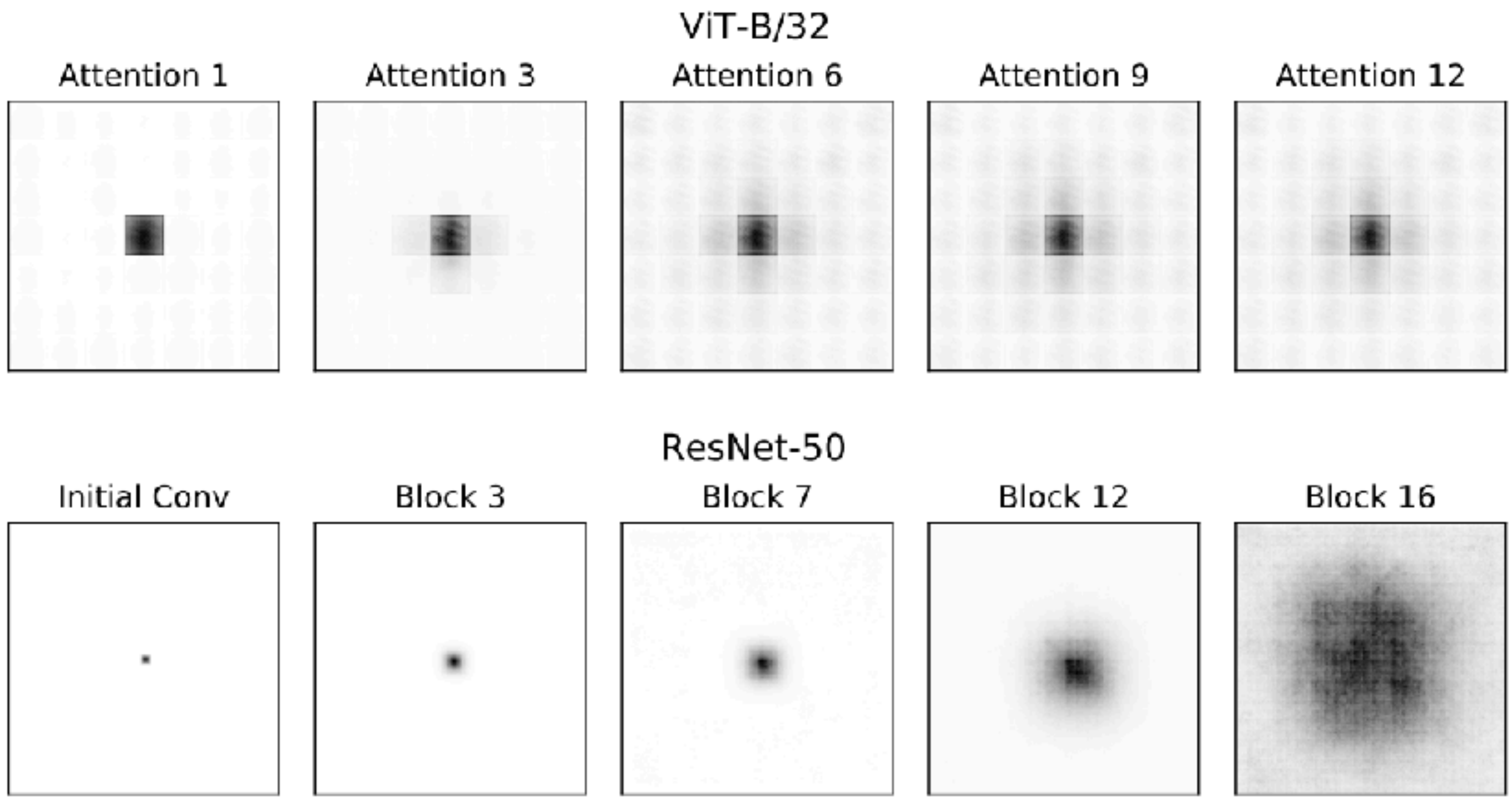
By structure CNNs only encode local information early on.



**Maithra et al. (2017)**

# How does the way vision transformers “see” differ from CNNs?

## Receptive Fields




**ViT starts incorporating global information much earlier.**

**Maithra et al. (2017)**

# Are transformers actually better than CNNs (ResNet)?

**These models are still undergoing training procedure improvements.**

ResNet strikes back: An improved training procedure in timm

Ross Wightman[°] Hugo Touvron[*,†] Hervé Jégou[*]

[°]Independent researcher [*]Facebook AI [†]Sorbonne University

**(Oct 2021)** - Propose a new, state-of-the-art training procedure for ResNets that beats the best ViT under some conditions.

DeiT III: Revenge of the ViT

Hugo Touvron[*,†] Matthieu Cord[†] Hervé Jégou[*]

[*]Meta AI [†]Sorbonne University

**(Apr 2022)** - Propose a new, state-of-the-art supervised training procedure for ViT, which beats ResNet under some conditions.

[ResNet strikes back: An improved training procedure in timm]()
[DeiT III: Revenge of the ViT]()

# References


1. Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, et al. “An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale.” arXiv, June 3, 2021. http://arxiv.org/abs/2010.11929.

2. Raghu, Maithra, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. “Do Vision Transformers See Like Convolutional Neural Networks?” arXiv, March 3, 2022. http://arxiv.org/abs/2108.08810.

3. Touvron, Hugo, Matthieu Cord, and Hervé Jégou. “DeiT III: Revenge of the ViT.” arXiv, April 14, 2022. http://arxiv.org/abs/2204.07118.

4. Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. “Attention Is All You Need.” arXiv, December 5, 2017. http://arxiv.org/abs/1706.03762.

5. Wightman, Ross, Hugo Touvron, and Hervé Jégou. “ResNet Strikes Back: An Improved Training Procedure in Timm.” arXiv, October 1, 2021. http://arxiv.org/abs/2110.00476.

# Useful demos, blogs, stack-overflow posts

1. https://demo.allennlp.org/next-token-lm
2. https://medium.com/deeper-learning/glossary-of-deep-learning-word-embedding-f90c3cec34ca
3. https://jalammar.github.io/illustrated-transformer/
4. https://towardsdatascience.com/illustrated-guide-to-transformers-step-by-step-explanation-f74876522bc0
5. https://stats.stackexchange.com/questions/498955/in-the-attention-mechanism-why-are-there-separate-weight-matrices-for-the-queri
6. https://stats.stackexchange.com/questions/421935/what-exactly-are-keys-queries-and-values-in-attention-mechanisms?rq=1
7. https://stats.stackexchange.com/questions/515477/when-calculating-self-attention-for-transformer-ml-architectures-why-do-we-need
8. https://stats.stackexchange.com/questions/430812/why-k-and-v-are-not-the-same-in-transformer-attention?rq=1