Intro to Transformers

Eliot Kim

June 27th 2025

Machine Learning Journal Club

Intro to Transformers

Eliot Kim

June 27th 2025

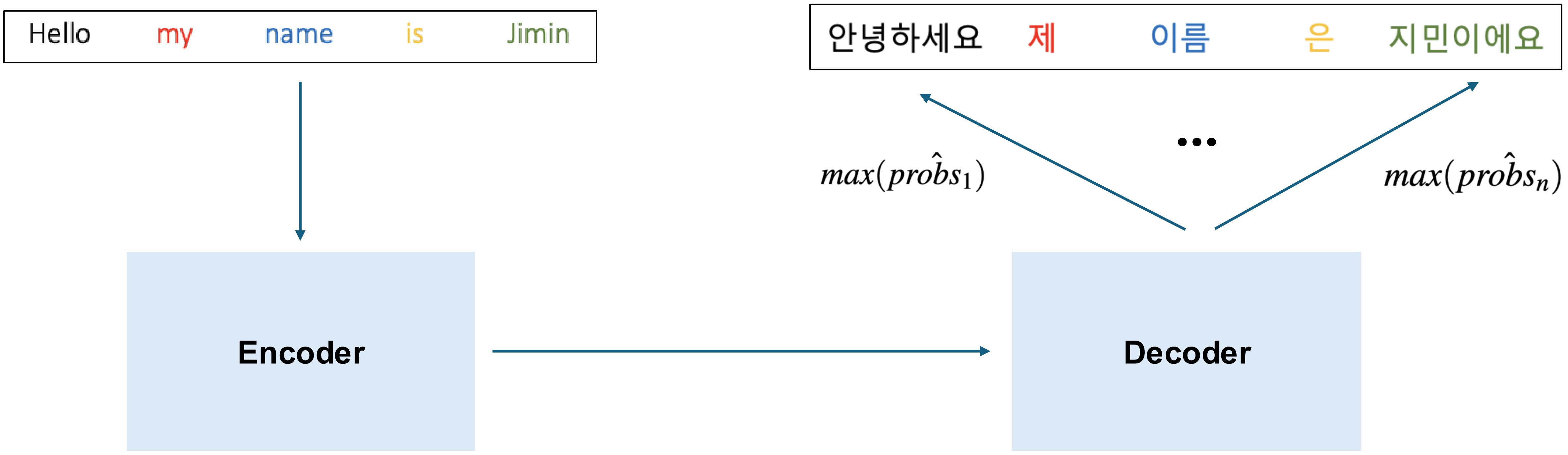
Machine Learning Journal Club

Q: How might strengths of the transformer architecture suit weather and climate modeling?

Motivation for transformers

Goal: Predict the most likely next term(s) given a sequence

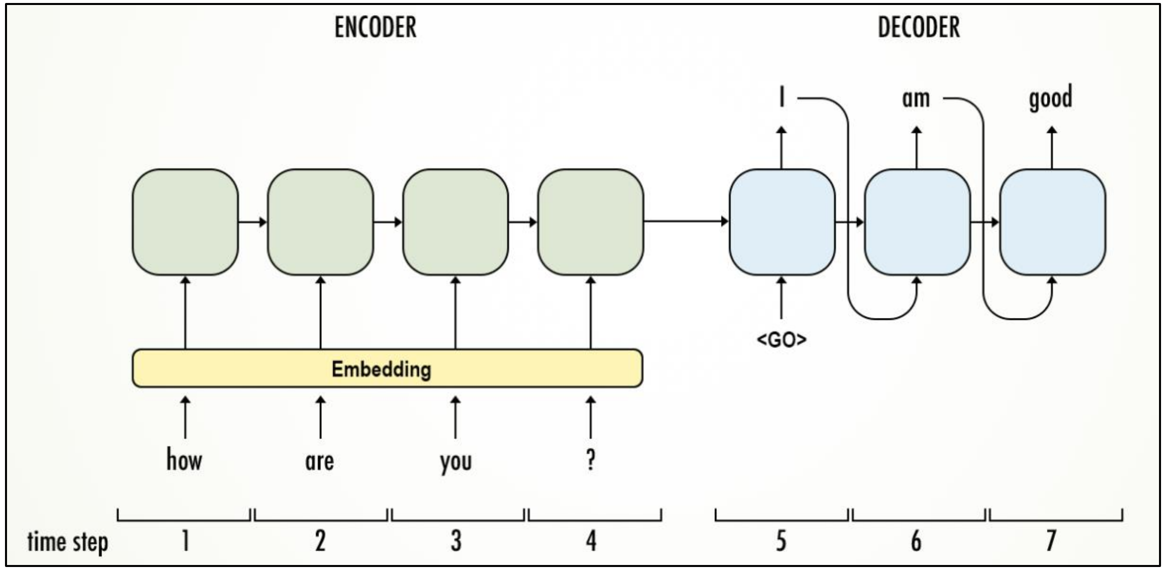
Ex: Translation



Motivation for transformers

Goal: Predict the most likely next term(s) given a sequence

Previously, **Recurrent Neural Networks** (RNNs):



– Not parallelizable
– Exploding gradients for long sequences
– Difficult to store long-term context (even with LSTMs)
– Sequence length and computational cost are directly related!

Advantages of transformers

Goal: Predict the next term(s) given a sequence

Then, in 2017: **"Attention is all you need"**

Attention is all you need A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, AN Gomez, ... Advances in neural information processing systems 30	186609	2017

Transformers, based on the **Attention** mechanism:

+ Look at whole sequence at once
+ Long-term context
+ Scalable
+ Parallelizable

Vaswani et al. 2017

Structure of transformers

Data Set-Up

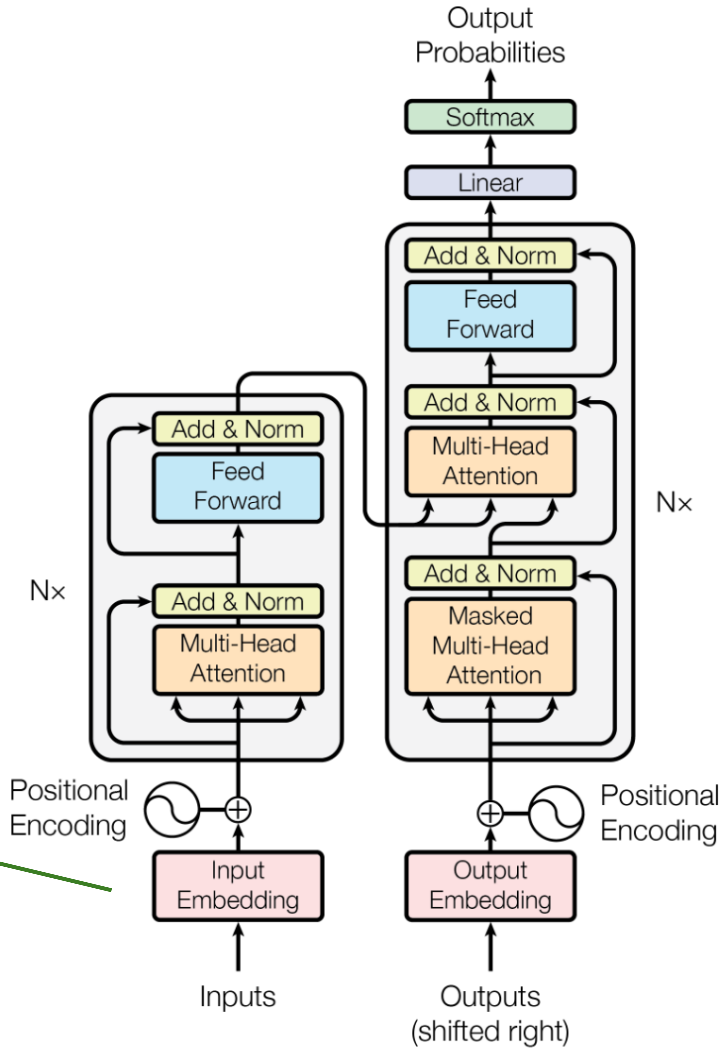
1) (<u>Tokenize</u> sequence)
2) <u>Embed</u> each token in high-dimensional space

Structure of transformers

Data Set-Up

1) (Tokenize sequence)
2) Embed each token in high-dimensional space

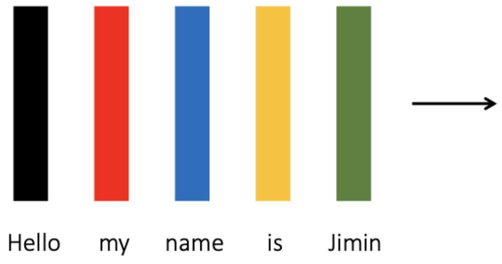
$$\mathbf{apple} = \begin{bmatrix} 0.42 \\ -1.37 \\ 3.14 \\ 2.72 \\ \vdots \\ 0.09 \end{bmatrix} \in R^{d_{\text{embed}}=512}$$



Structure of transformers

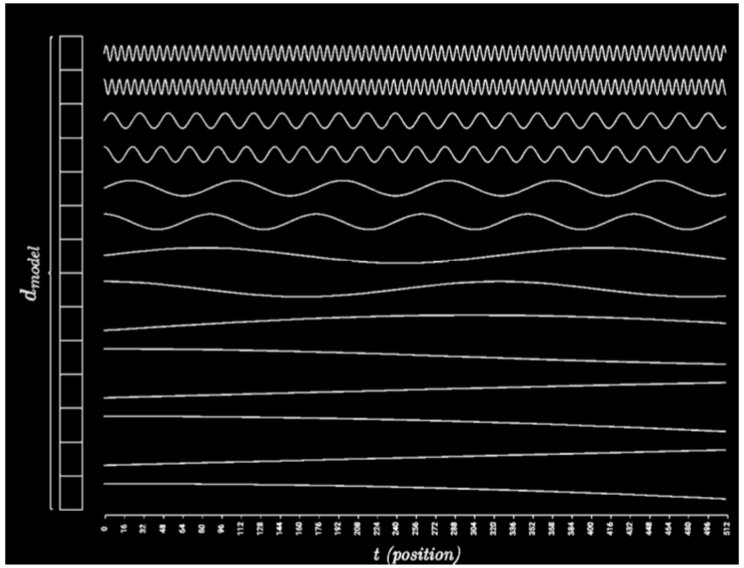
Data Set-Up

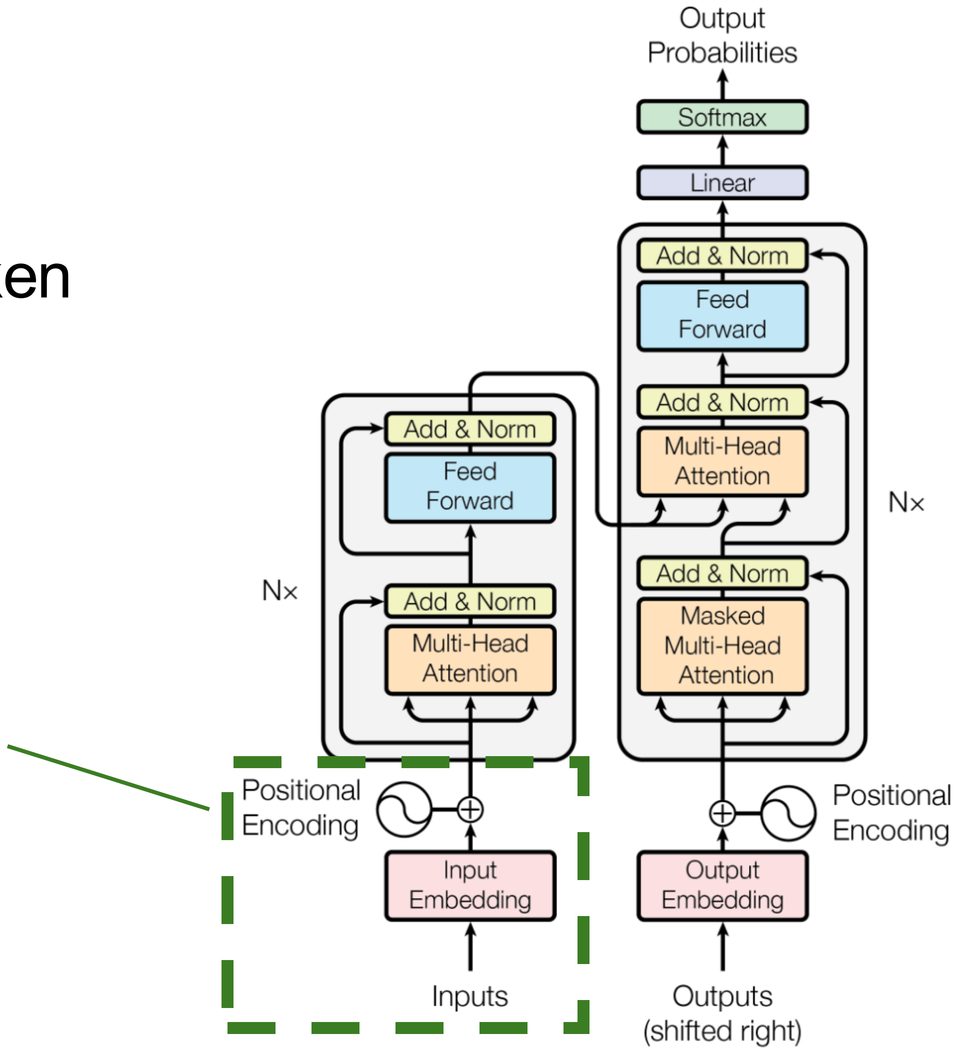
1) (<u>Tokenize</u> sequence)
2) <u>Embed</u> each token
3) Add <u>position encoding</u> to embedding of each token using sine and cosine functions



k *is even*: $\sin\left(\frac{t}{10000^{\frac{k}{d_{embedding}}}}\right)$

k *is odd*: $\cos\left(\frac{t}{10000^{\frac{k}{d_{embedding}}}}\right)$



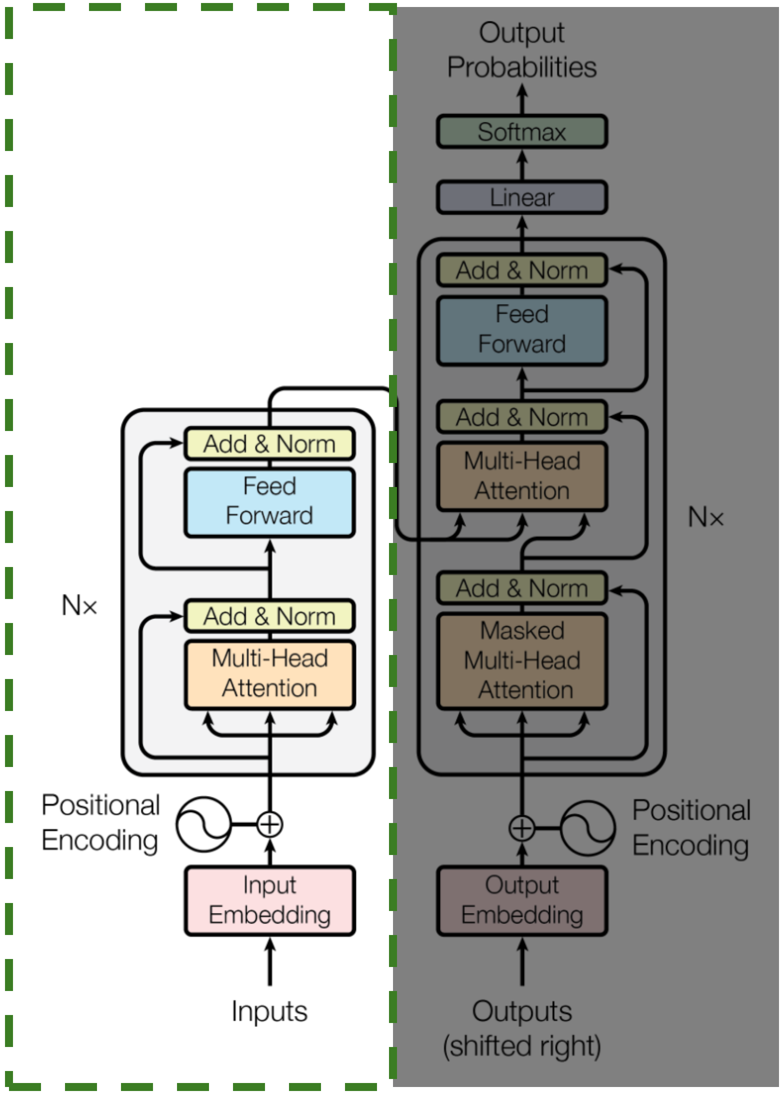


Structure of transformers

Data Set-Up

1) (Tokenize sequence)
2) Embed each token
3) Add position encoding for each token

Learning



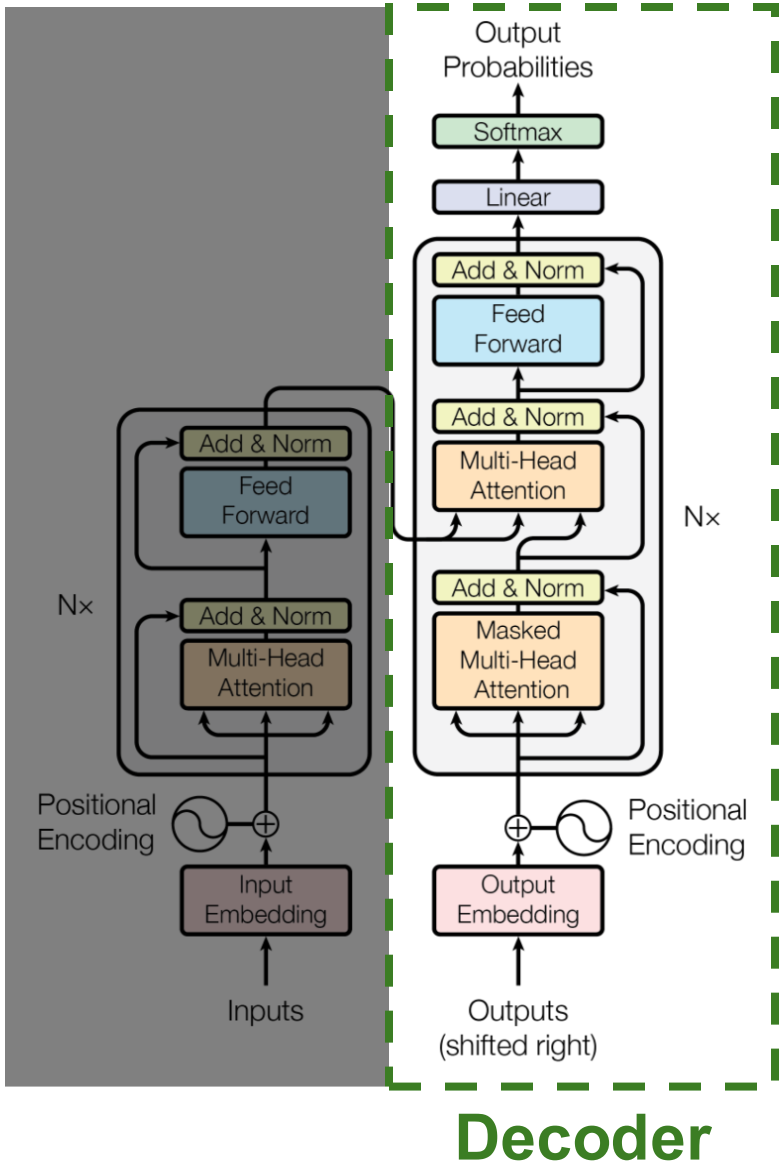
Encoder

Structure of transformers

Data Set-Up

1) (Tokenize sequence)
2) Embed each token
3) Add position encoding for each token

Learning



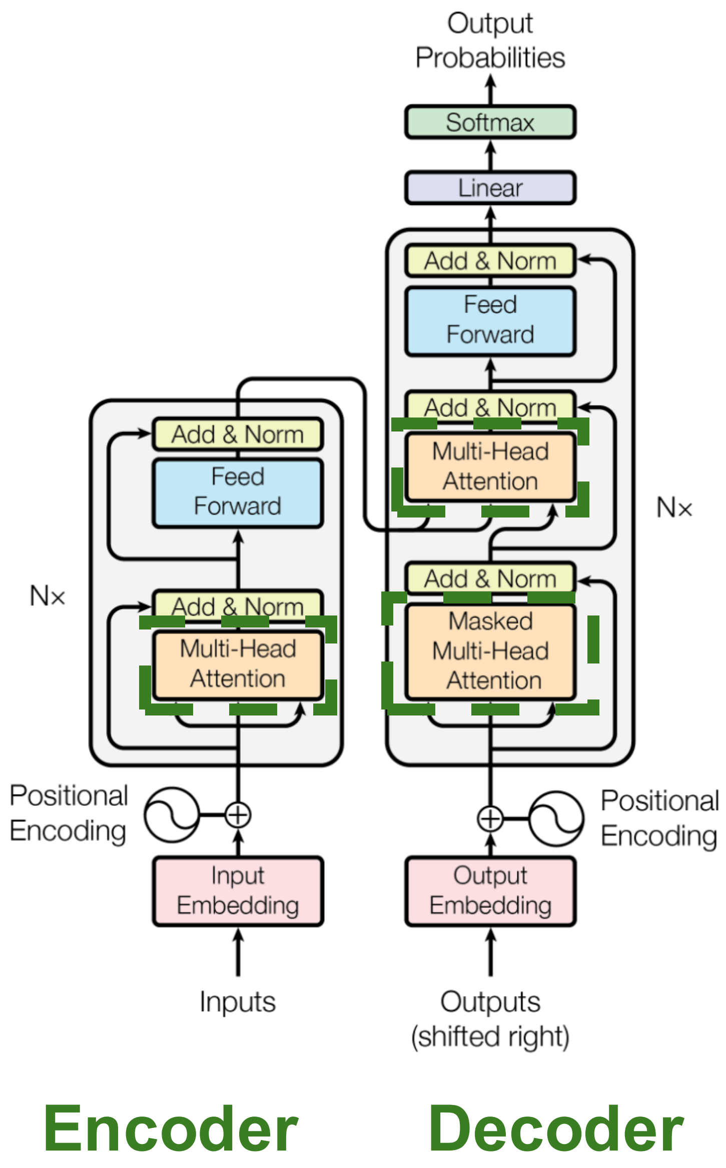
Structure of transformers

Data Set-Up

1) (Tokenize sequence)
2) Embed each token
3) Add position encoding for each token

Learning

1) Attention: Update each embedding with **relevant contextual information**



Structure of transformers

Data Set-Up

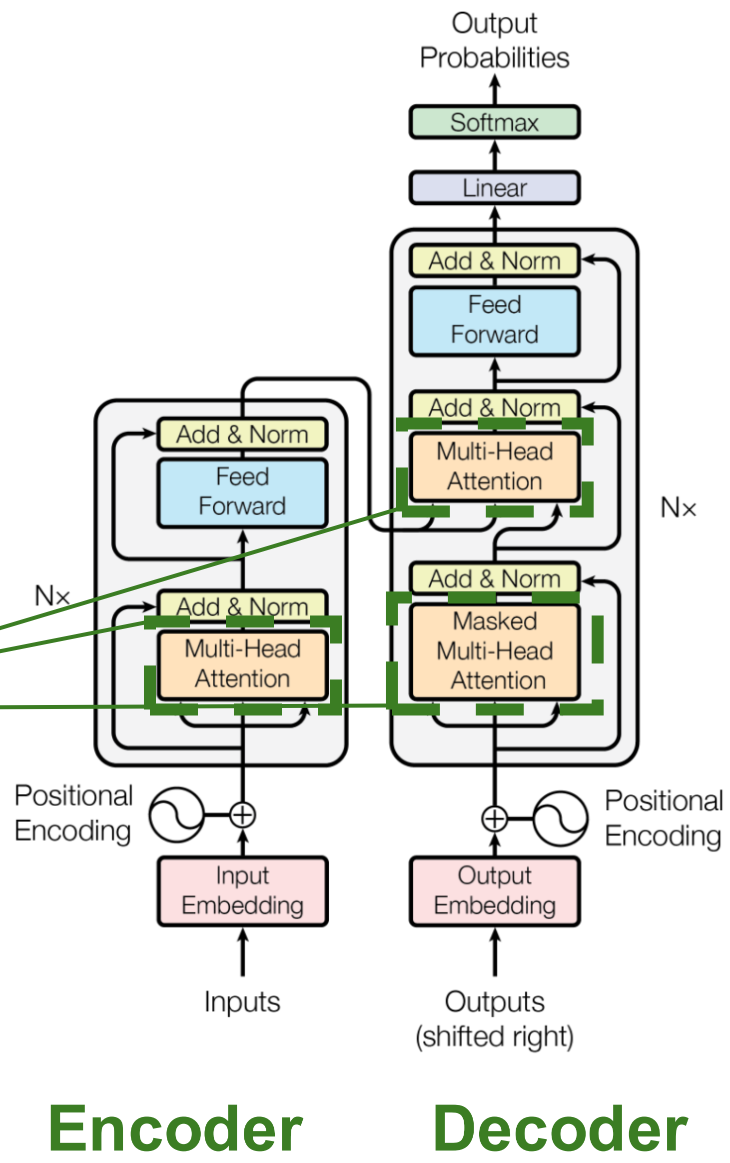
1) (Tokenize sequence)
2) Embed each token
3) Add position encoding for each token

Learning

1) Attention: Update each embedding with **relevant contextual information**

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

$$\text{Layer Output} = \text{Norm}(\vec{E} + \Delta\vec{E})$$



Structure of transformers

Data Set-Up

1) (Tokenize sequence)
2) Embed each token
3) Add position encoding for each token

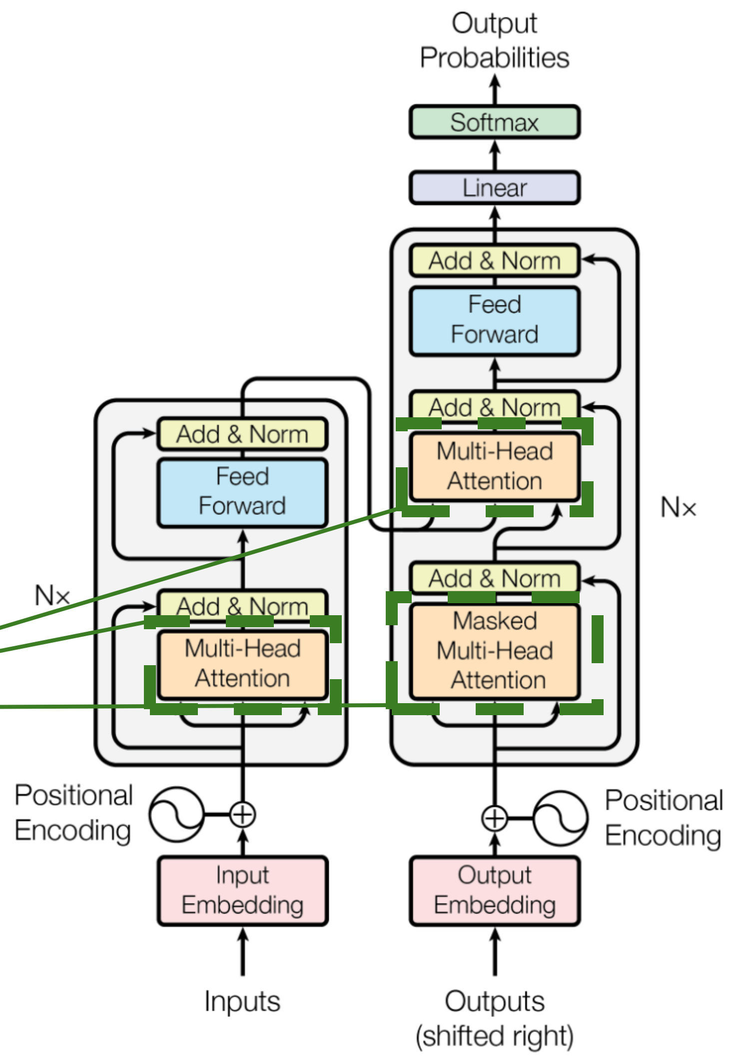
Learning

1) Attention: Update each embedding with **relevant contextual information**

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

Query Key Value

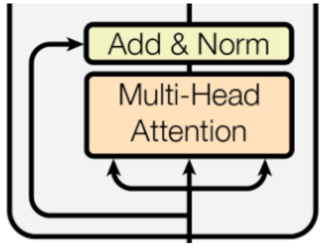
For numerical stability



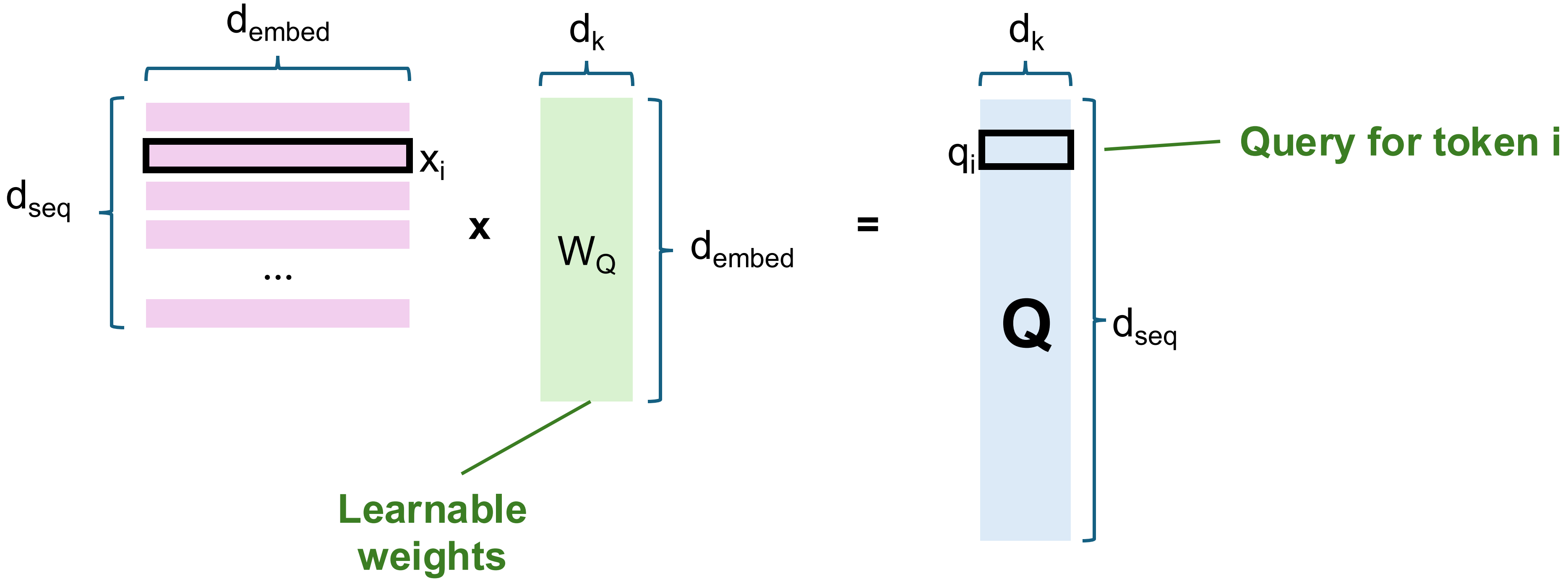
Encoder **Decoder**

Attention mechanism

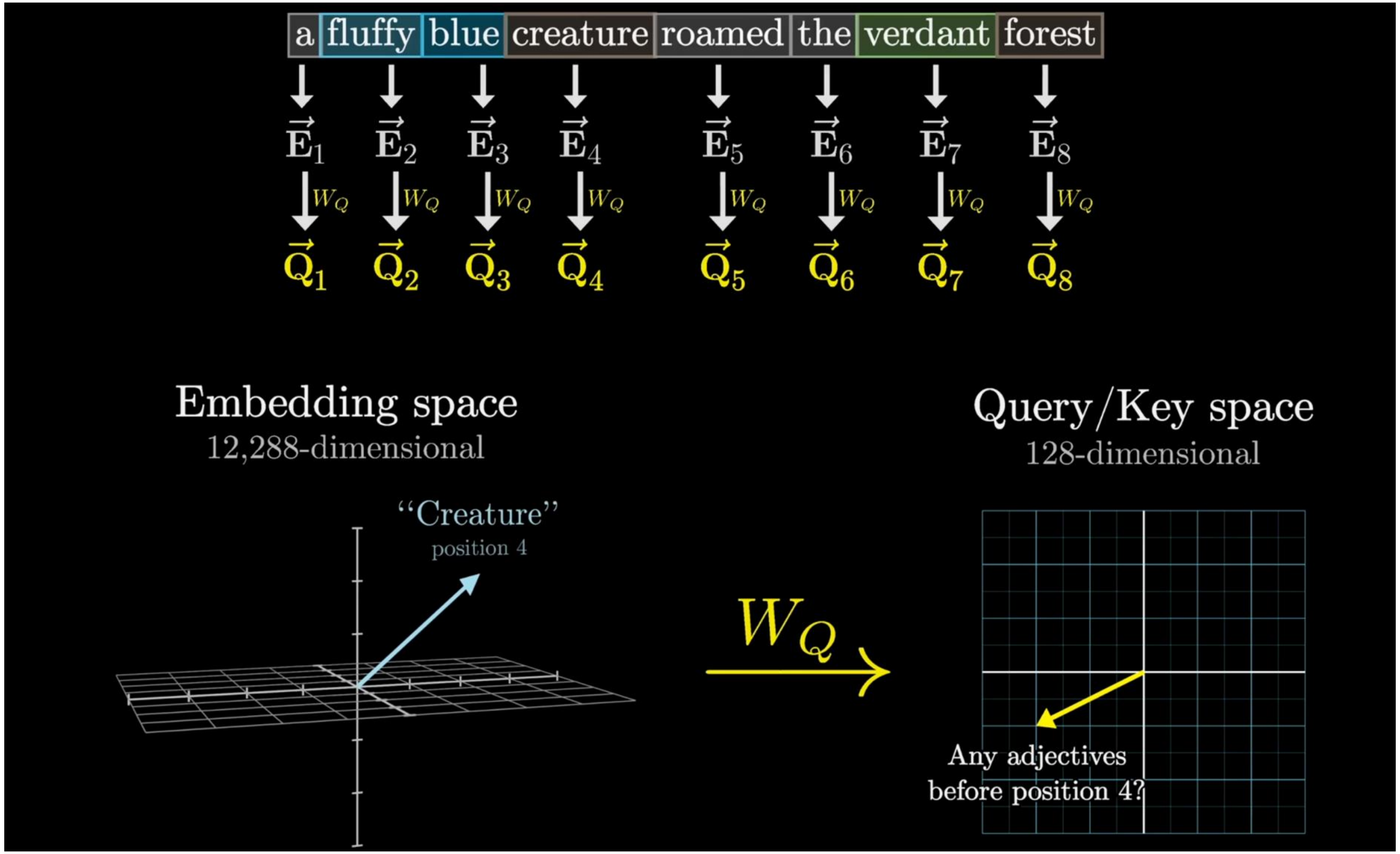
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Query



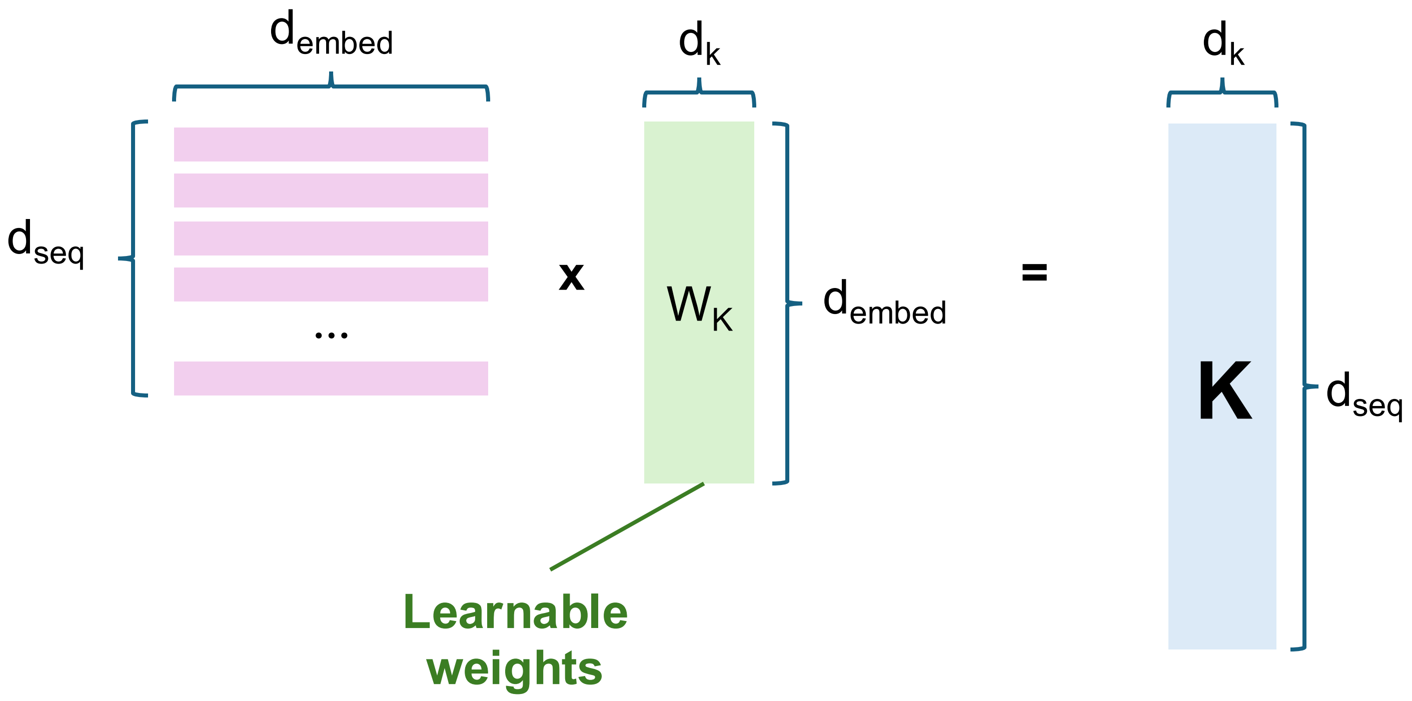
Attention mechanism



Attention mechanism

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

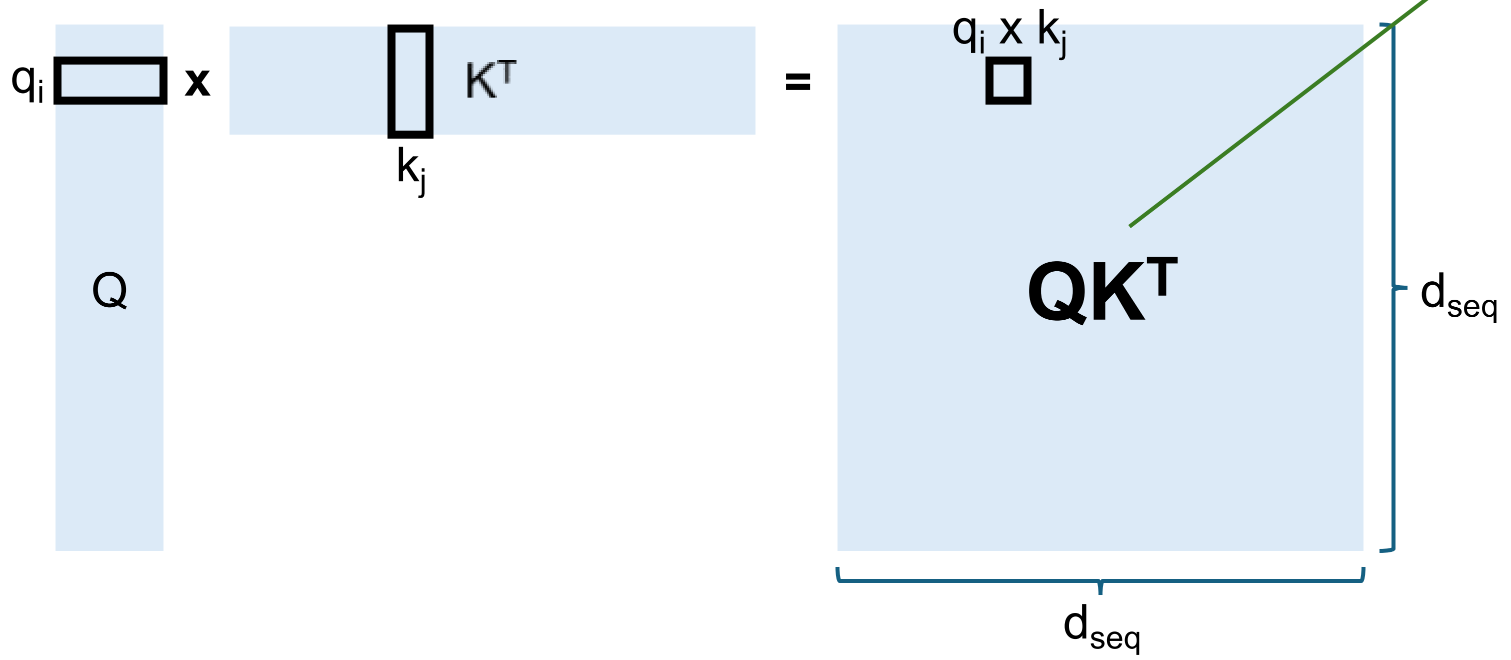
Key



Attention mechanism

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

Attention Map

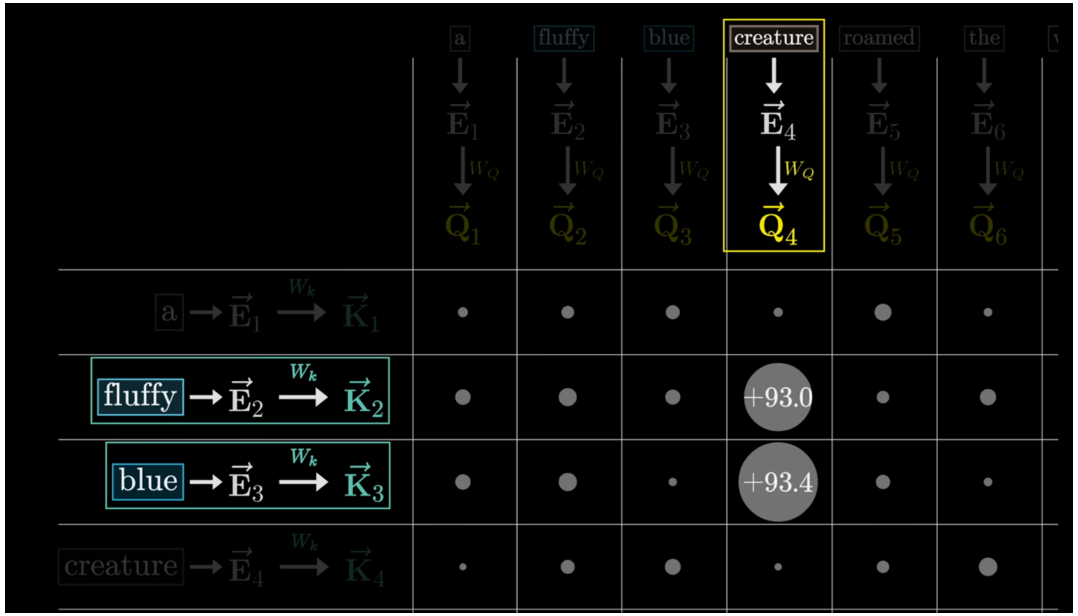


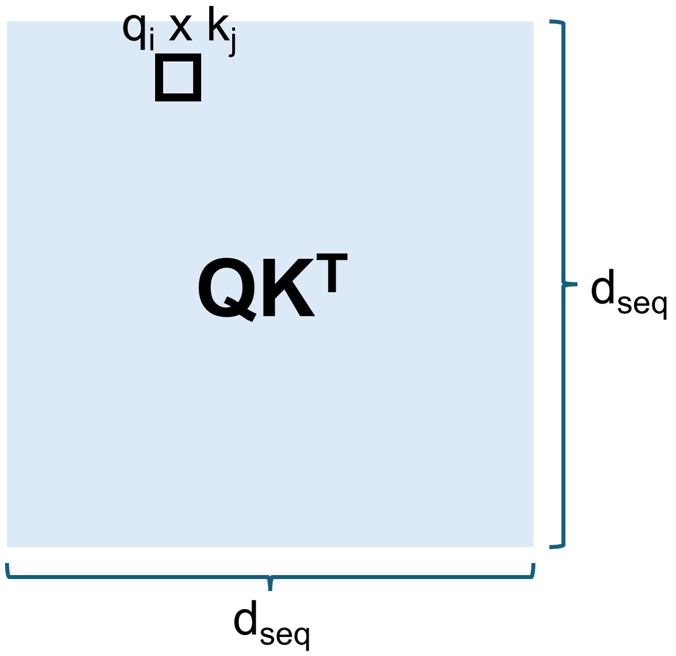
When enhancing the embedding of token <i>, how much attention should the model pay to token <j> ?

Attention mechanism

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

Attention Map



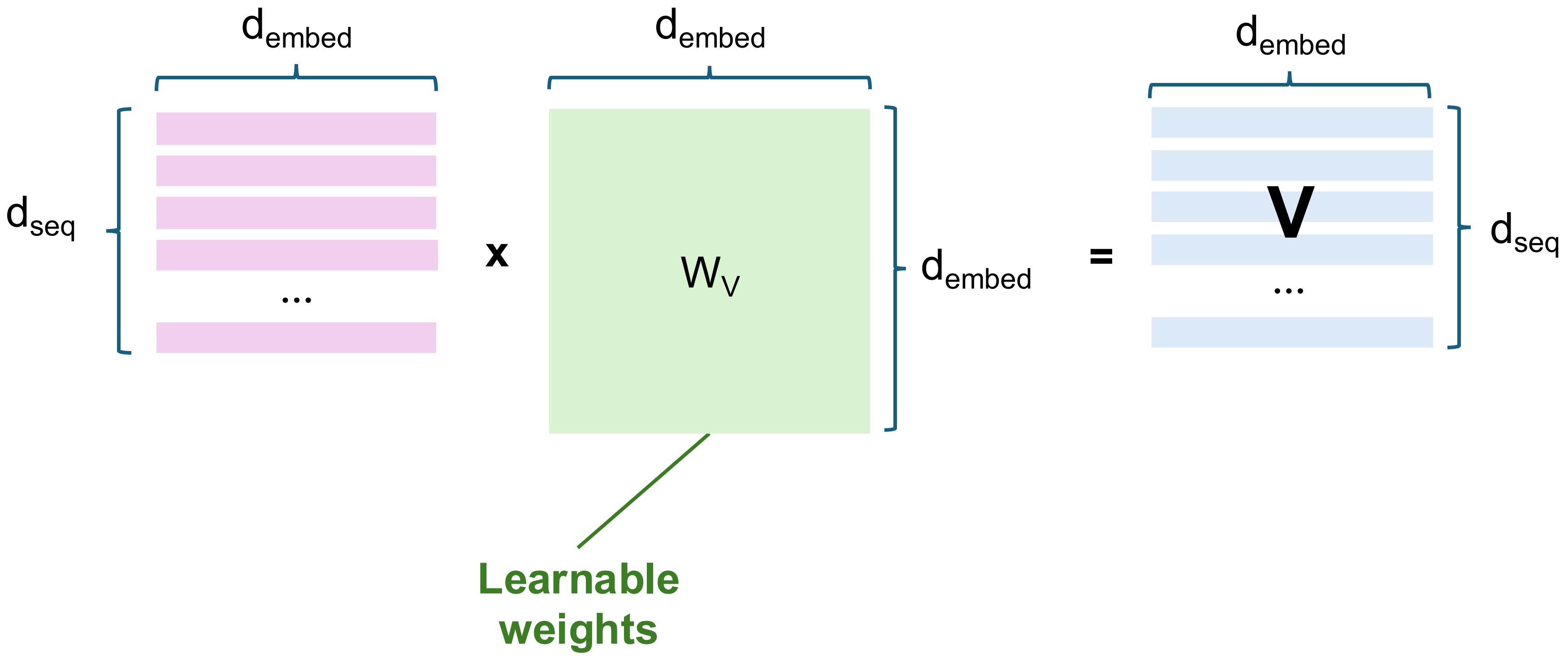


Does the embedding of token <j> "attend" to the embedding of token <i> ?

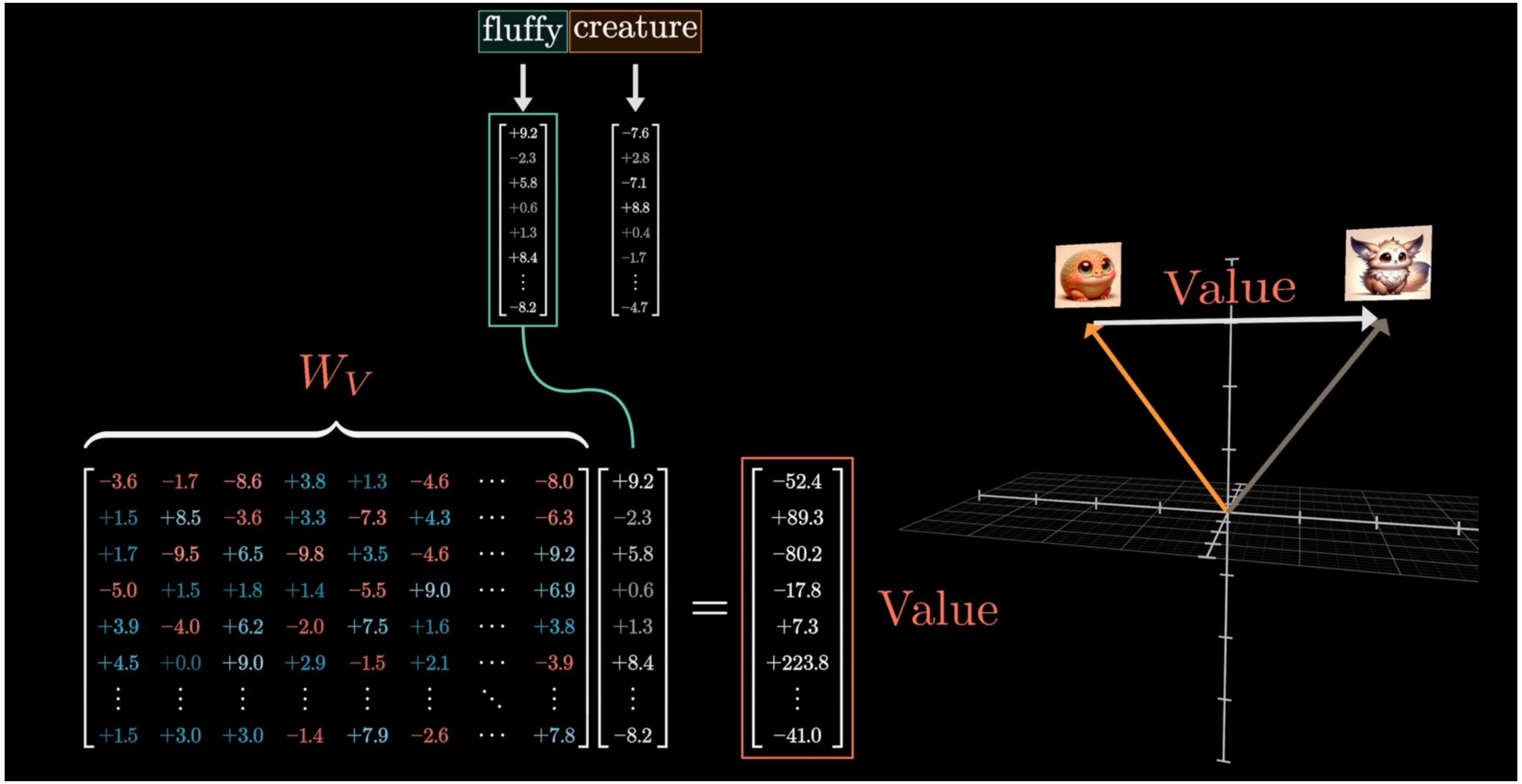
Attention mechanism

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

Value



Attention mechanism



"If this word is relevant for updating the meaning of something else, what should be added to the embedding of something else?"

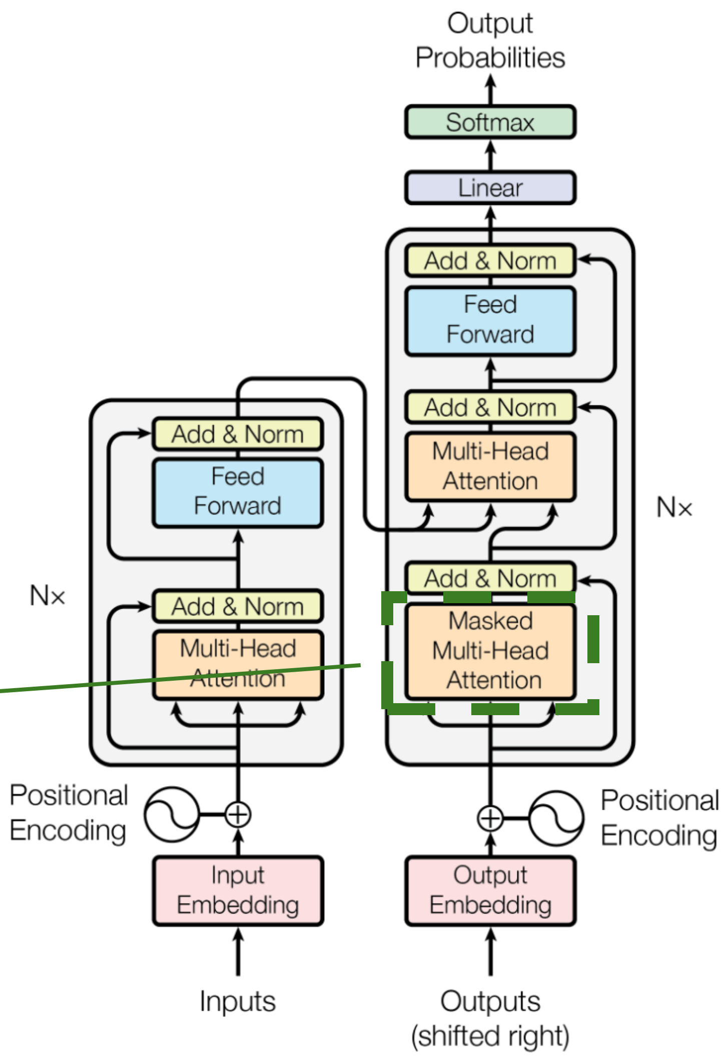
Structure of transformers

Data Set-Up

1) (Tokenize sequence)
2) Embed each token
3) Add position encoding for each token

Learning

4) Attention: Update each embedding with **relevant contextual information**
   a) Self-attention ✓
   b) Masking: *Prevent attention map from using information later in the sequence*
   c) Cross-attention
   d) Multi-headed attention



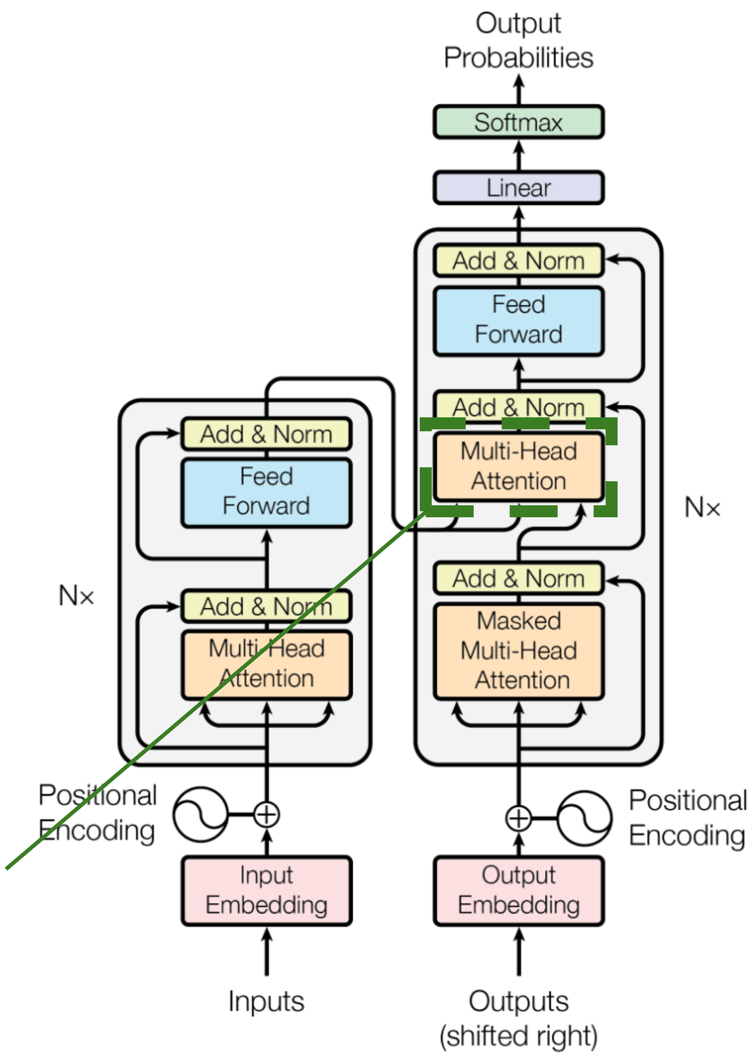
Structure of transformers

Data Set-Up

1) (Tokenize sequence)
2) Embed each token
3) Add position encoding for each token

Learning

4) Attention: Update each embedding with **relevant contextual information**
   a) Self-attention ✓
   b) Masking: *Prevent attention map from using information later in the sequence*
   c) Cross-attention: *Create attention map between two different sequences (i.e. translation)*
   d) Multi-headed attention



Structure of transformers

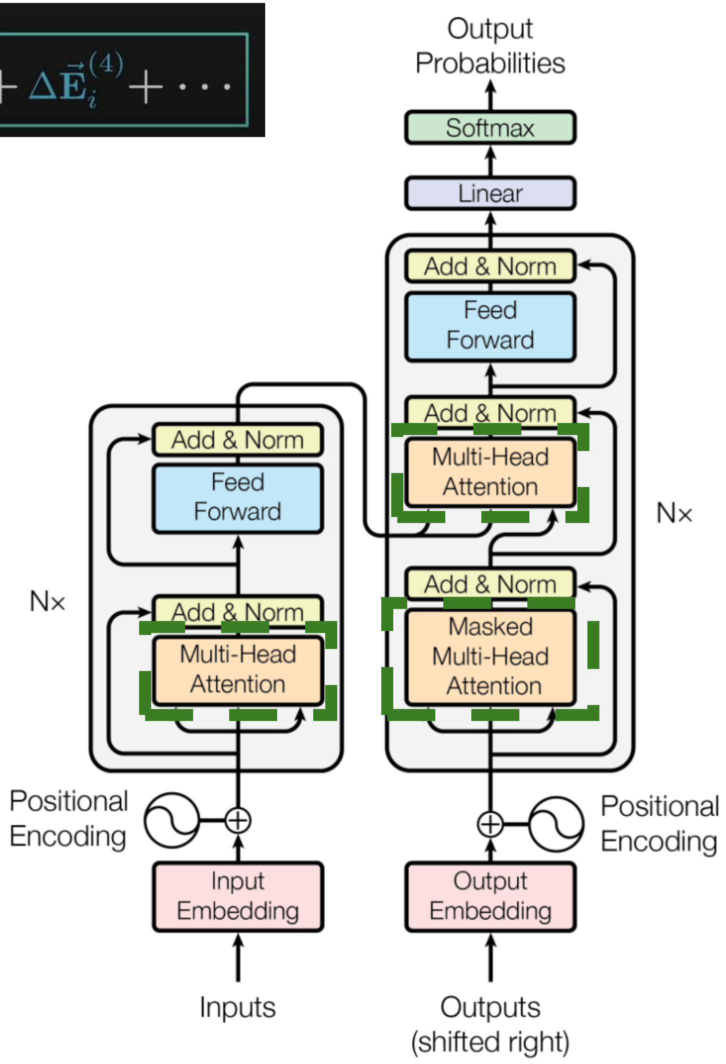
Data Set-Up

1) (Tokenize sequence)
2) Embed each token
3) Add position encoding for each token

New embedding $\vec{\mathbf{E}}_i + \Delta\vec{\mathbf{E}}_i^{(1)} + \Delta\vec{\mathbf{E}}_i^{(2)} + \Delta\vec{\mathbf{E}}_i^{(3)} + \Delta\vec{\mathbf{E}}_i^{(4)} + \cdots$

Learning

4) Attention: Update each embedding with **relevant contextual information**
   a) Self-attention ✓
   b) Masking: *Prevent attention map from using information later in the sequence*
   c) Cross-attention: *Create attention map between two different sequences (i.e. translation)*
   d) Multi-headed attention: *Each head learns a different context-based update to the embedding*



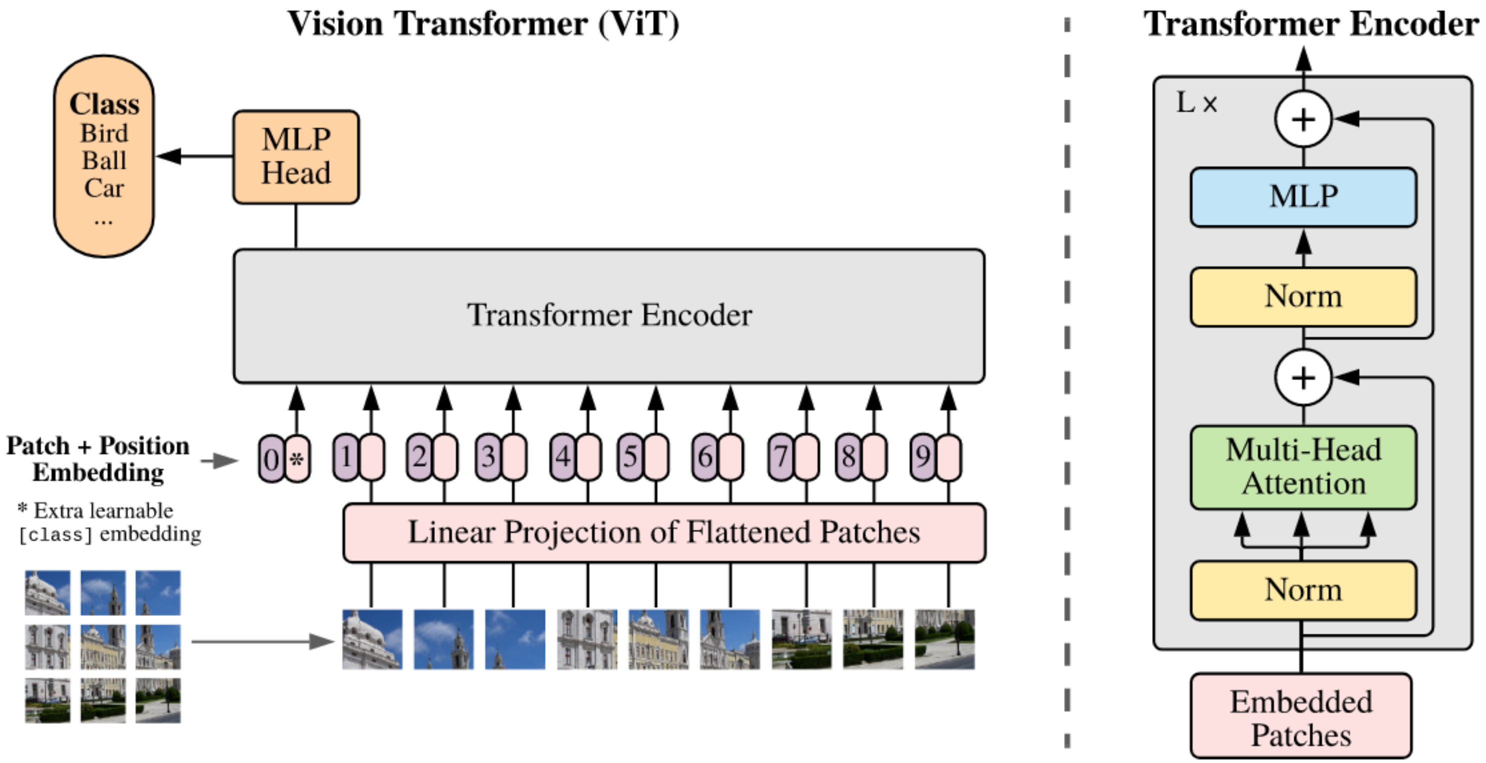
Vision Transformers

Self-attention becomes very expensive for pixel-to-pixel, so… patching!

Dosovitskiy et al. 2021

Vision Transformers

Self-attention becomes very expensive for pixel-to-pixel, so… patching!



- With sufficient dataset size, beats conventional CNN-based models

Dosovitskiy et al. 2021

Topics I didn't have time to dig into

- Training process
  - Pre-training
  - Fine-tuning
- Vision Transformer architectures
  - Shifted Window (Swin) → used in Aurora
- Disadvantages of transformers and open research areas
- Transformers for weather and climate modeling
  - ClimaX (Nguyen et al. 2023)
  - Stormer (Nguyen et al. 2024)
  - Aurora (Bodnar et al. 2025, Lehmann et al. 2025)

Sources

Attention Intro: https://www.youtube.com/watch?v=eMlx5fFNoYc

How LLMs store facts: https://www.youtube.com/watch?v=9-Jl0dxWQs8

Attention is All You Need Video: https://www.youtube.com/watch?v=iDulhoQ2pro

UW ECE 596 Course Notes

UW-Madison ML Prof’s Blog: https://sebastianraschka.com/blog/2023/self-attention-from-scratch.html