

$$F = G \frac{m_1 m_2}{d^2}$$

# Deep Learning for Particle Physicists

Lewis Tunstall | AEC Graduate Seminar | June 3rd 2022

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

$$\frac{df}{dt} = \lim_{h \rightarrow 0} \frac{f(t+h) - f(t)}{h}$$

This week in ML

# Transformers (and a digression into NLP)

# Transformers in HEP

## Symmetries and self-supervision in particle physics

**Barry M. Dillon**

Institut für Theoretische Physik  
Universität Heidelberg  
dillon@thphys.uni-heidelberg.de

**Gregor Kasieczka**

Institut für Experimentalphysik  
Universität Hamburg  
gregor.kasieczka@cern.ch

**Hans Olschläger**

Institut für Theoretische Physik  
Universität Heidelberg

**Tilman Plehn**

Institut für Theoretische Physik  
Universität Heidelberg  
plehn@uni-heidelberg.de

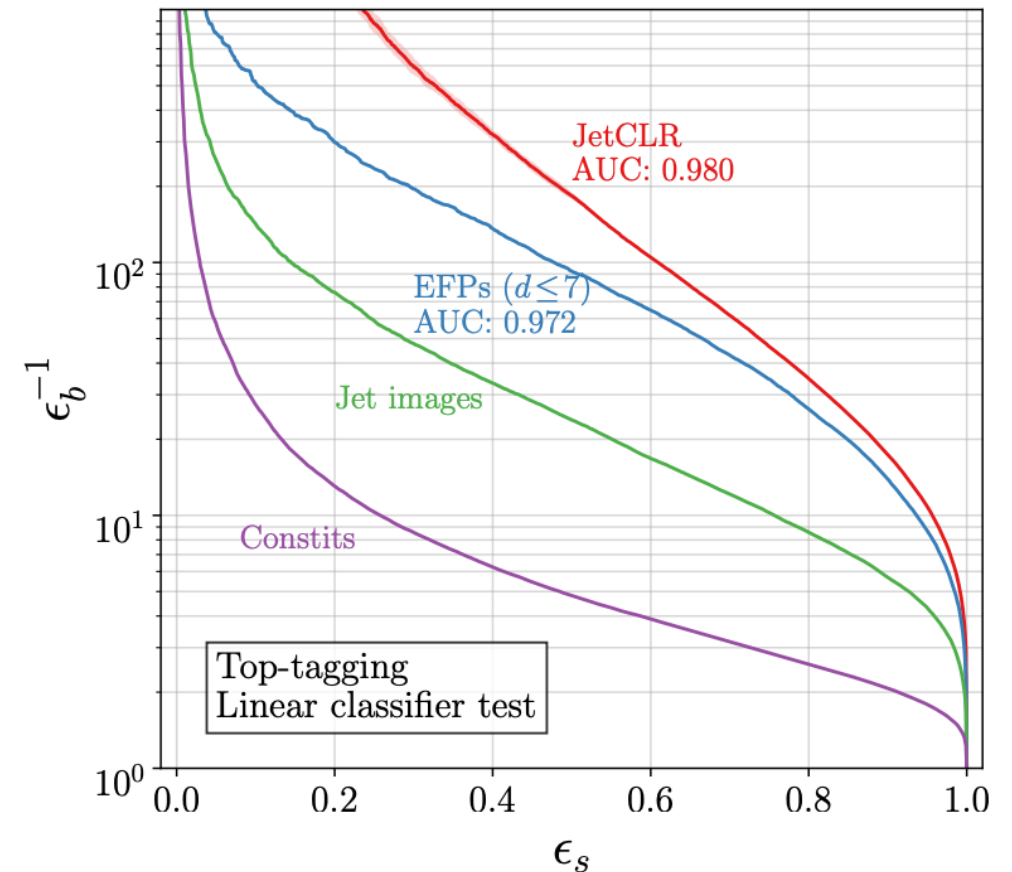
**Peter Sorrenson**

Institut für Theoretische Physik  
Heidelberg Collaboratory for Image Processing  
Universität Heidelberg  
peter.sorrenson@iwr.uni-heidelberg.de

**Lorenz Vogel**

Institut für Theoretische Physik  
Universität Heidelberg

[Paper link](#)



Still early days, pretraining & transfer learning less common than other fields

# Transformers in HEP

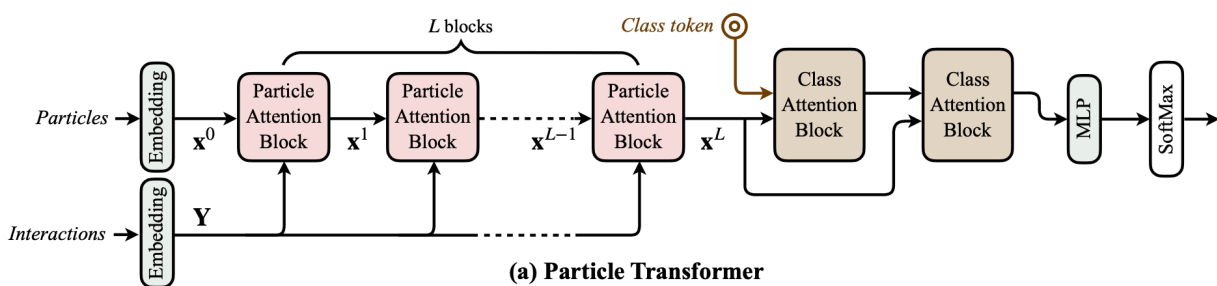
arXiv > hep-ph > arXiv:2202.03772

High Energy Physics – Phenomenology

[Submitted on 8 Feb 2022]

## Particle Transformer for Jet Tagging

Huilin Qu, Congqiao Li, Sitian Qian

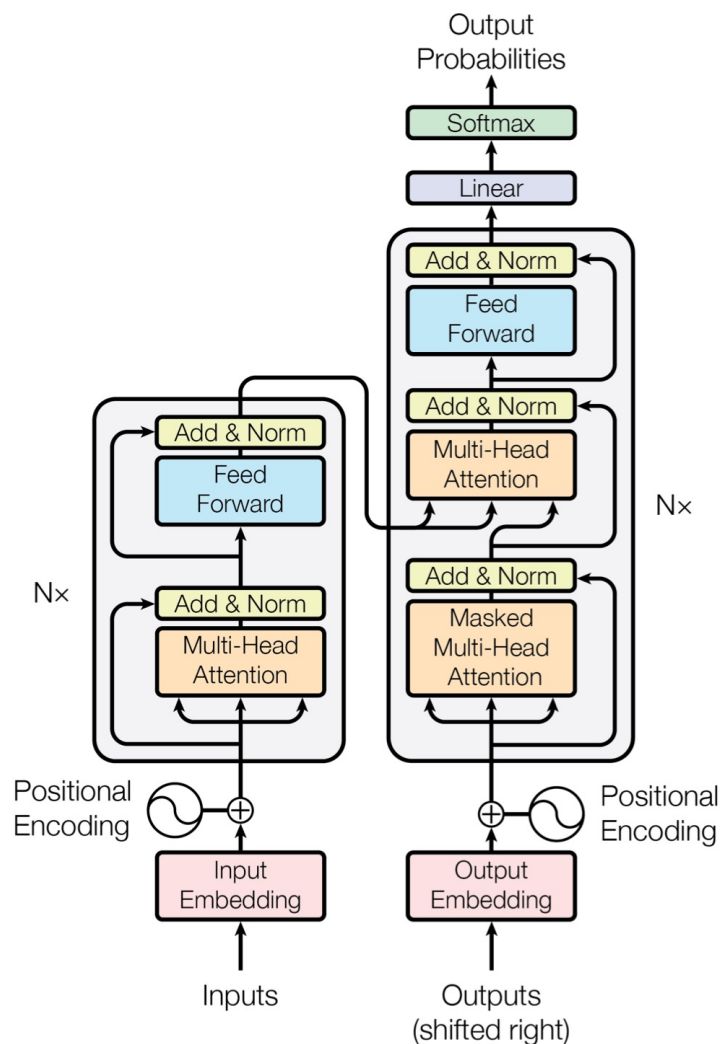


	Accuracy	AUC	Rej <sub>50%</sub>	Rej <sub>30%</sub>
P-CNN	0.930	0.9803	$201 \pm 4$	$759 \pm 24$
PFN	—	0.9819	$247 \pm 3$	$888 \pm 17$
ParticleNet	0.940	0.9858	$397 \pm 7$	$1615 \pm 93$
JEDI-net (w/ $\sum O$ )	0.930	0.9807	—	774.6
PCT	0.940	0.9855	$392 \pm 7$	$1533 \pm 101$
LGN	0.929	0.964	—	$435 \pm 95$
rPCN	—	0.9845	$364 \pm 9$	$1642 \pm 93$
ParT	0.940	0.9858	$413 \pm 16$	$1602 \pm 81$
<b>ParT-f.t.</b>	<b>0.944</b>	<b>0.9877</b>	<b><math>691 \pm 15</math></b>	<b><math>2766 \pm 130</math></b>

But promising results in jet tagging from large-scale datasets (100M events)



# What is a Transformer?



---

## 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

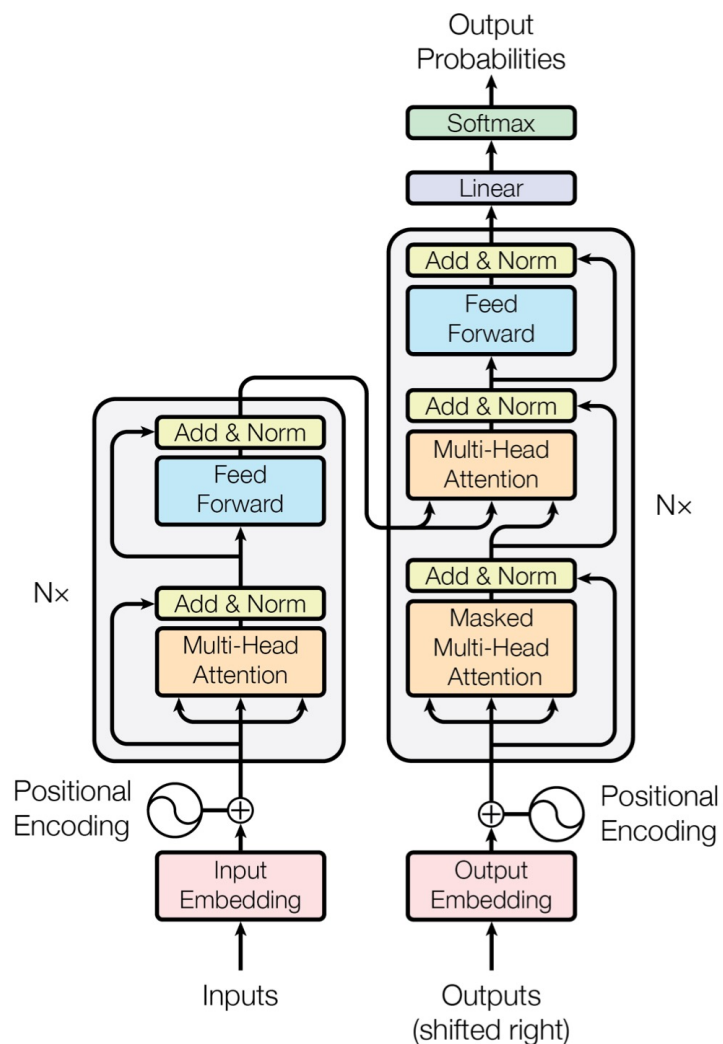
**Llion Jones\***  
Google Research  
llion@google.com

**Aidan N. Gomez\* †**  
University of Toronto  
aidan@cs.toronto.edu

**Łukasz Kaiser\***  
Google Brain  
lukaszkaizer@google.com

**Illia Polosukhin\* ‡**  
illia.polosukhin@gmail.com

# What is a Transformer?



---

## 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  
lukaszkaizer@google.com

**Illia Polosukhin\* ‡**  
illia.polosukhin@gmail.com

# Main ingredients



Attention  
mechanisms



Self-supervised learning  
(Pretraining)



Transfer learning  
(Fine-tuning)



# Main ingredients



Attention  
mechanisms



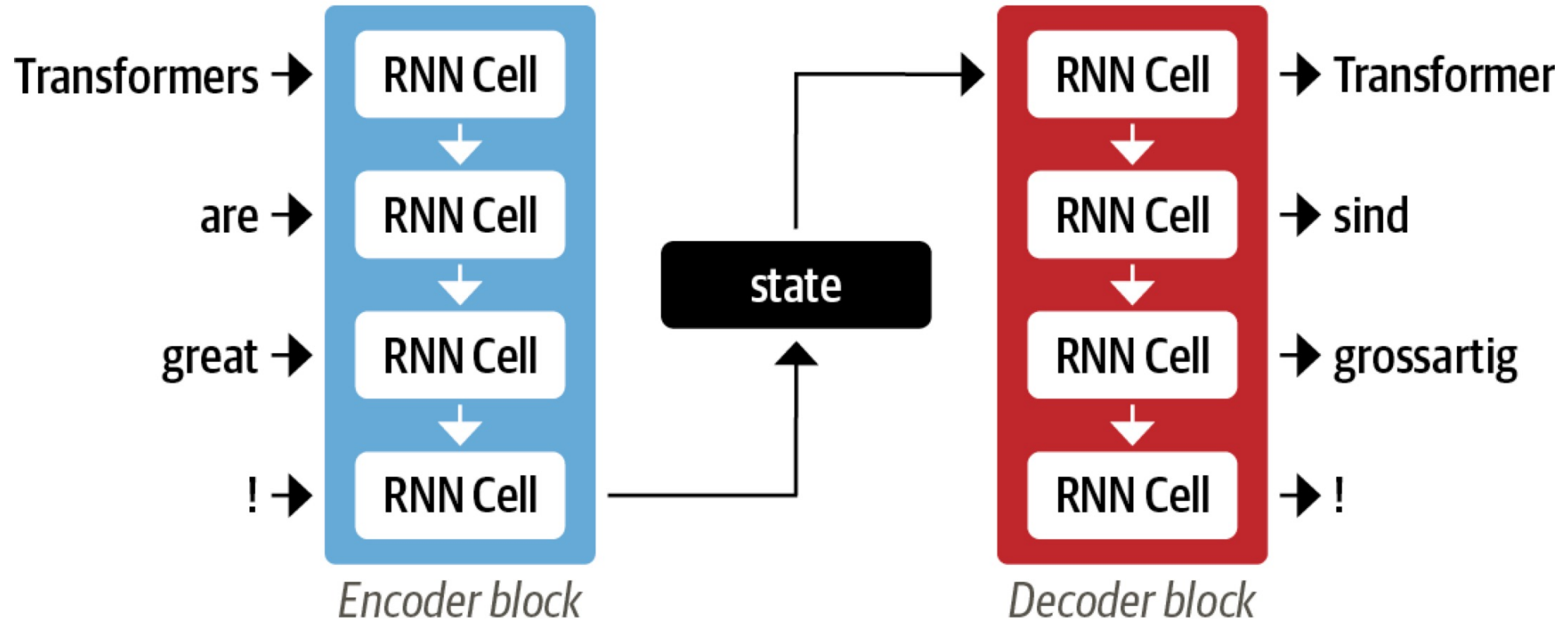
Self-supervised learning  
(Pretraining)



Transfer learning  
(Fine-tuning)

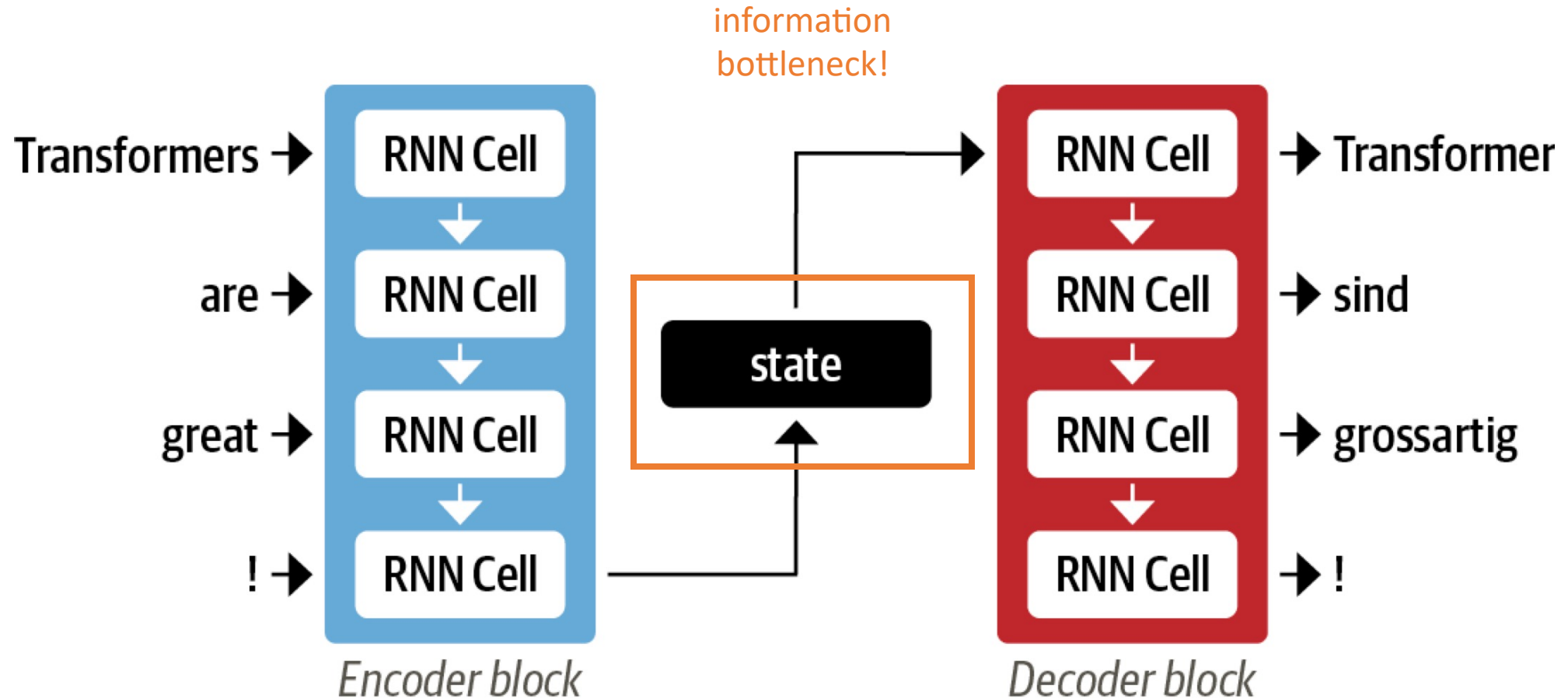
we first saw this in lecture 4 in the  
context of jet images and CNNs

# Attention mechanisms



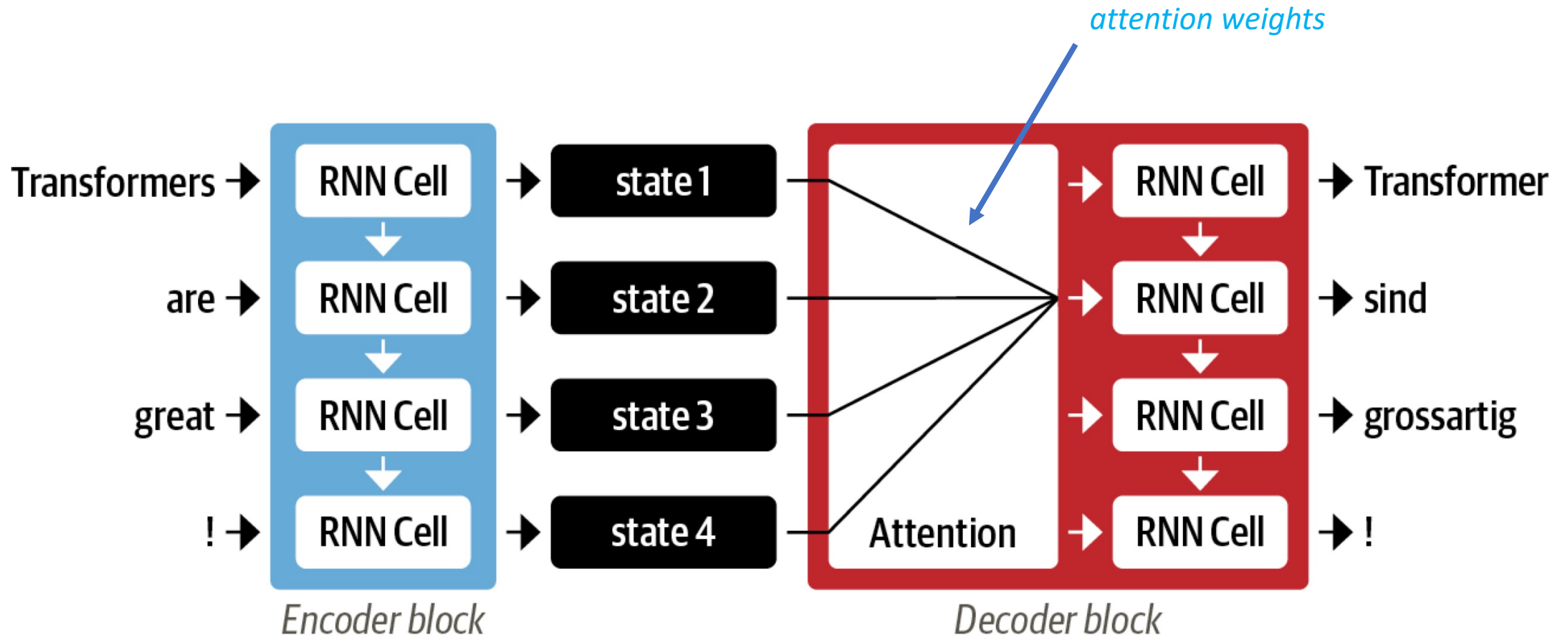
Originally developed for *recurrent neural networks*

# Attention mechanisms



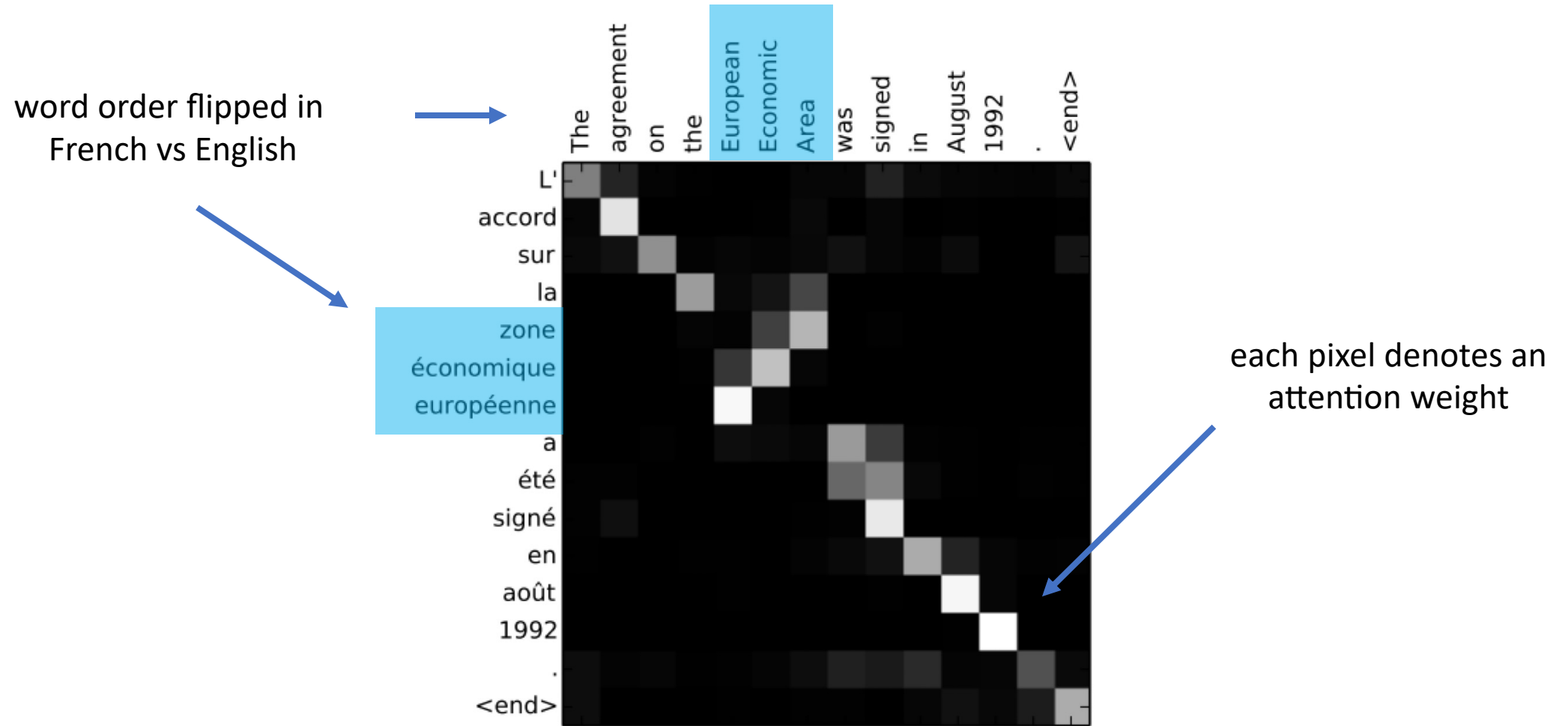
Originally developed for *recurrent neural networks*

# Attention mechanisms



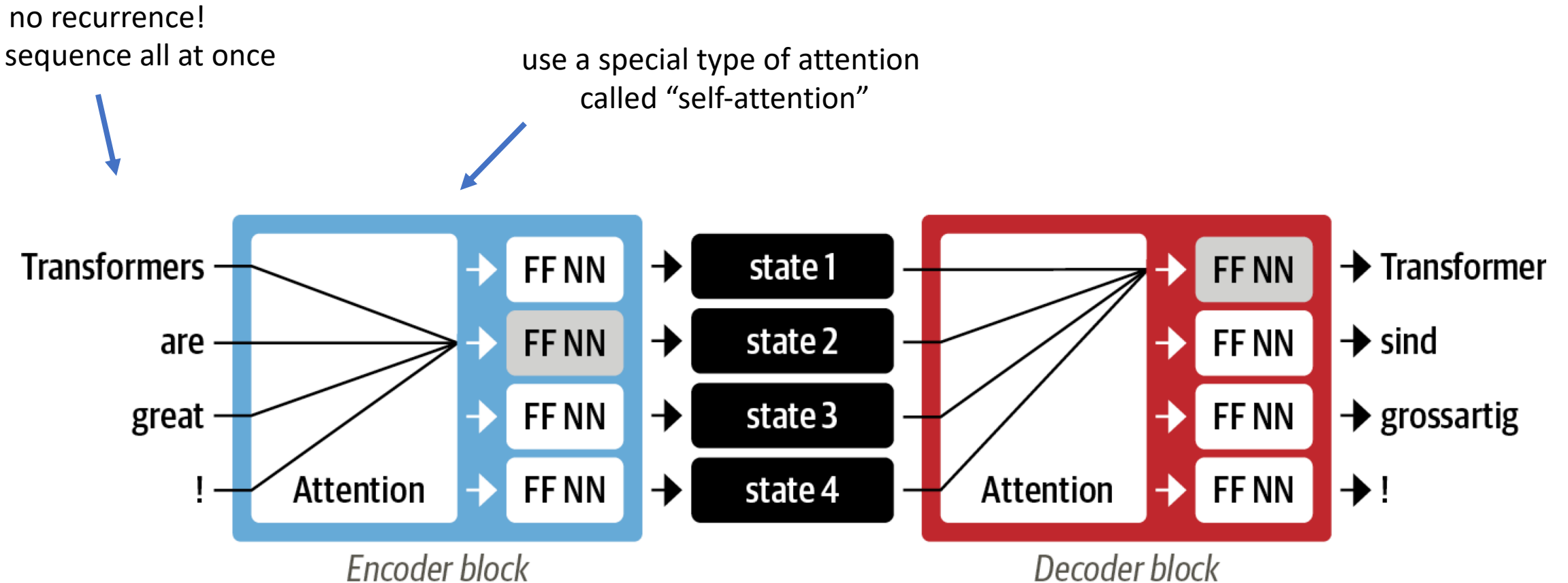
Use intermediate states but assign a *weight* or “pay attention”

# Attention mechanisms



Attention gives better modelling of word order

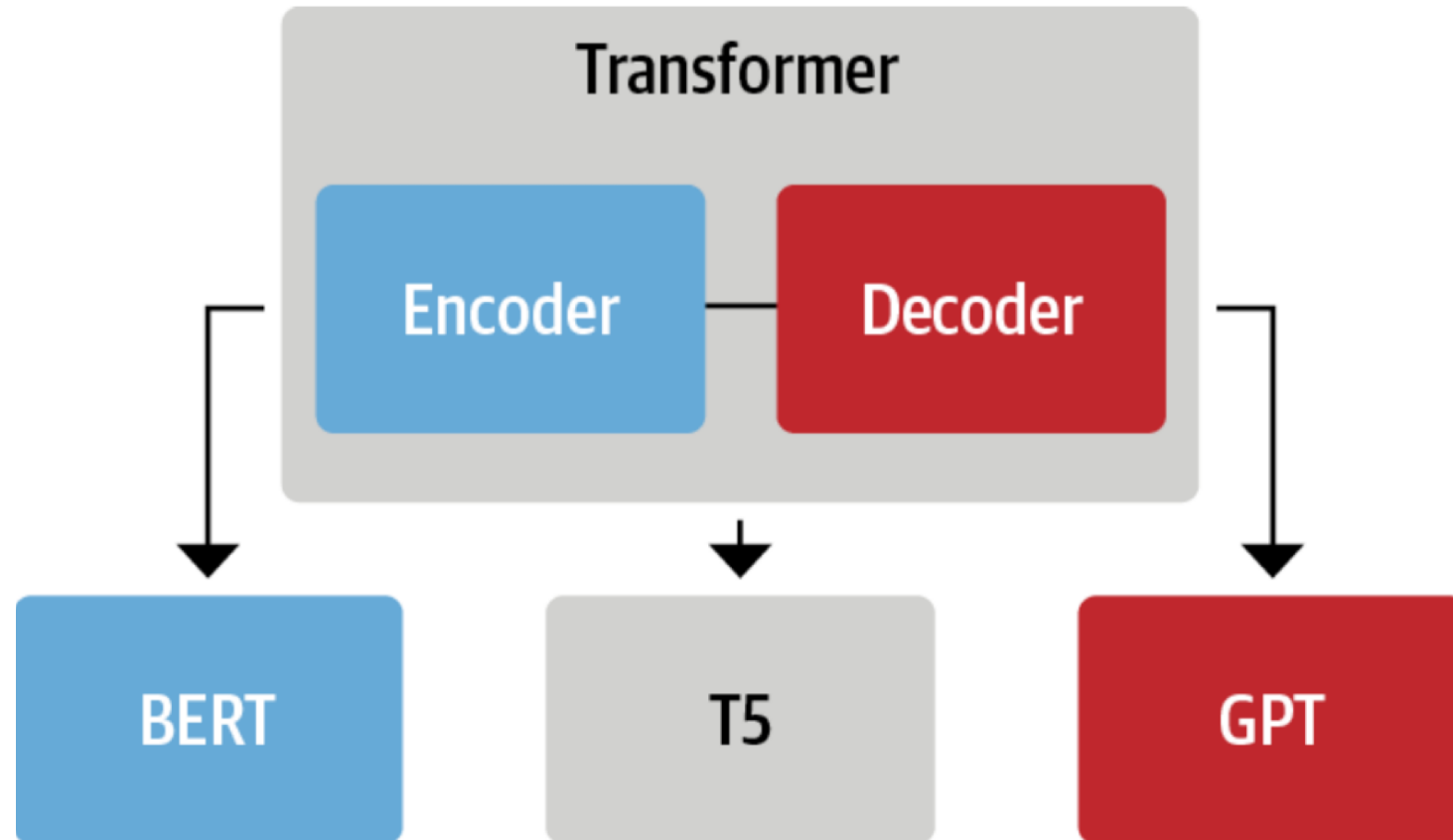
# Attention mechanisms



Transformers much easier to scale with compute & data



# Three types of architectures



Each architecture excels at specific types of tasks

# Transformer pretraining

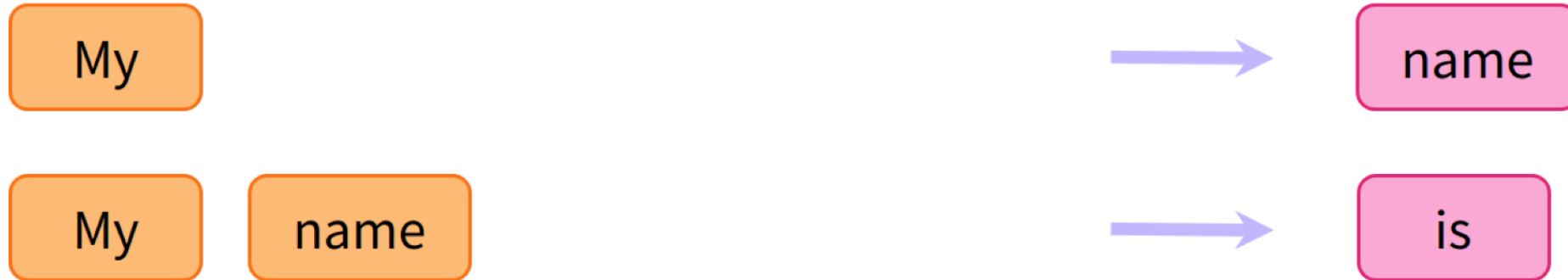
---



Trained on unlabeled data to predict the next token (GPT-like) ...

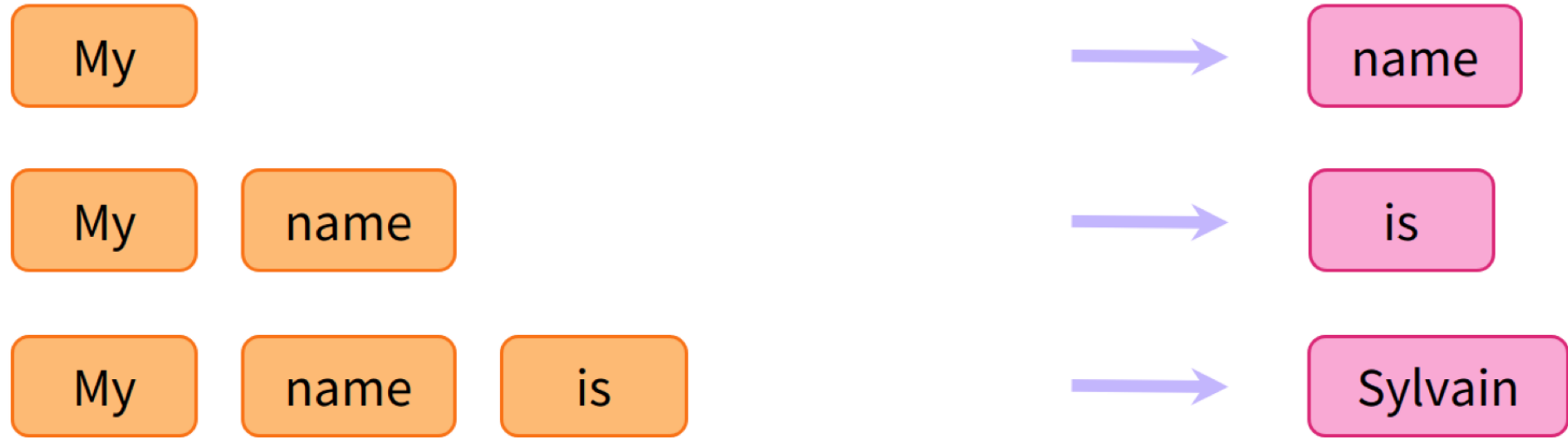
# Transformer pretraining

---



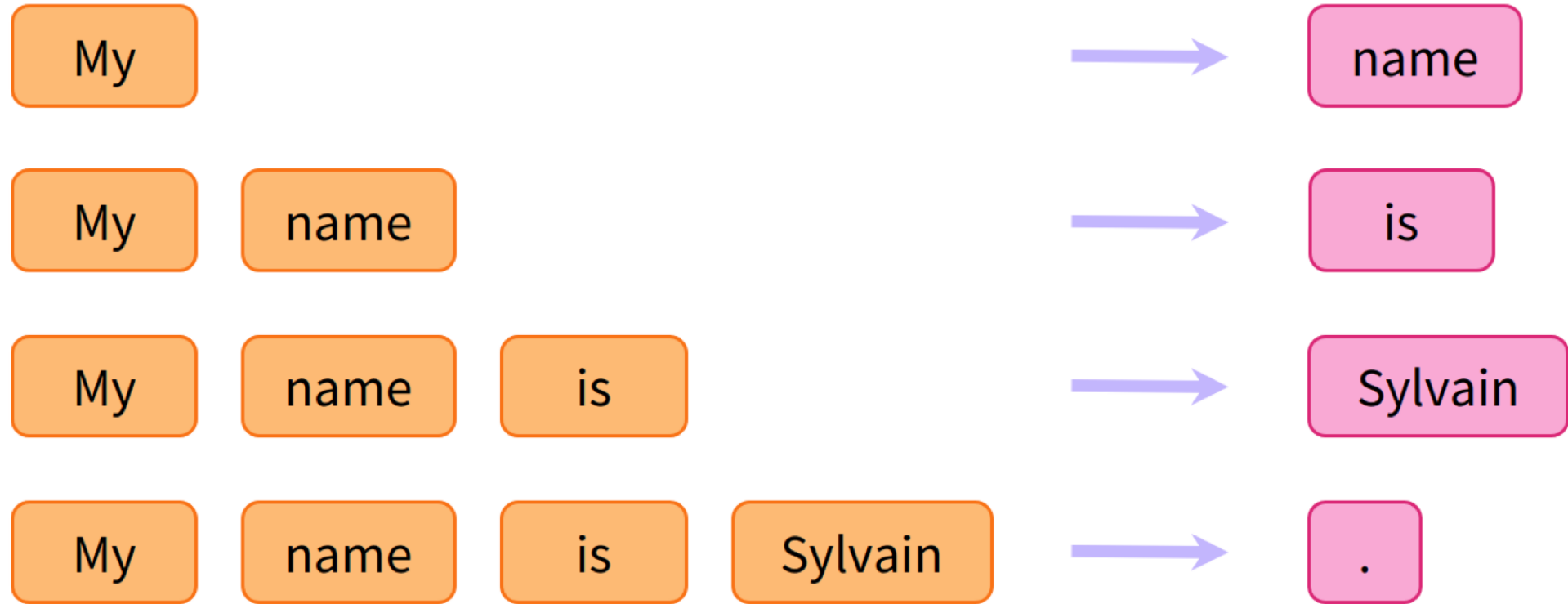
Trained on unlabeled data to predict the next token (GPT-like) ...

# Transformer pretraining



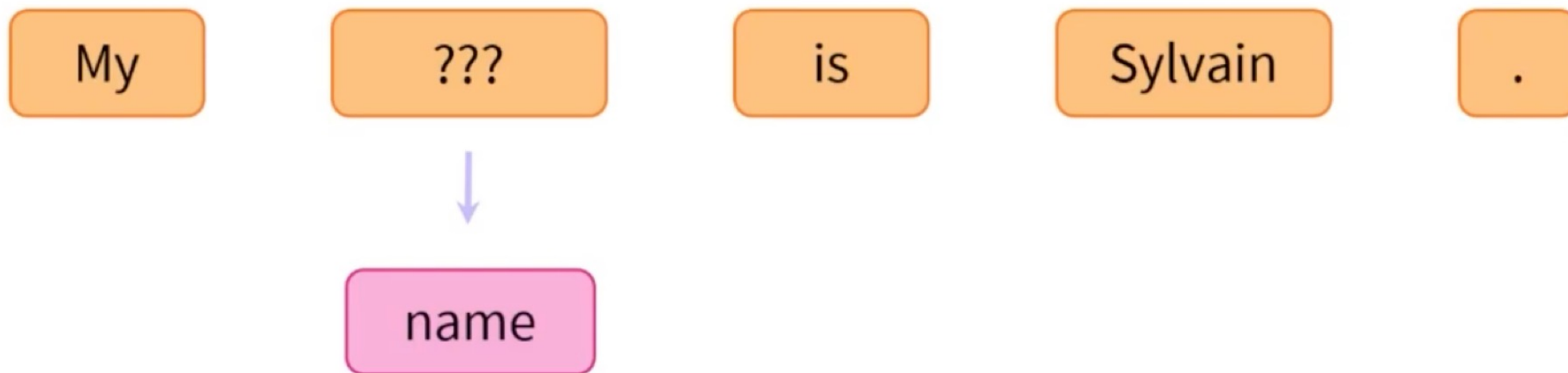
Trained on unlabeled data to predict the next token (GPT-like) ...

# Transformer pretraining



Trained on unlabeled data to predict the next token (GPT-like) ...

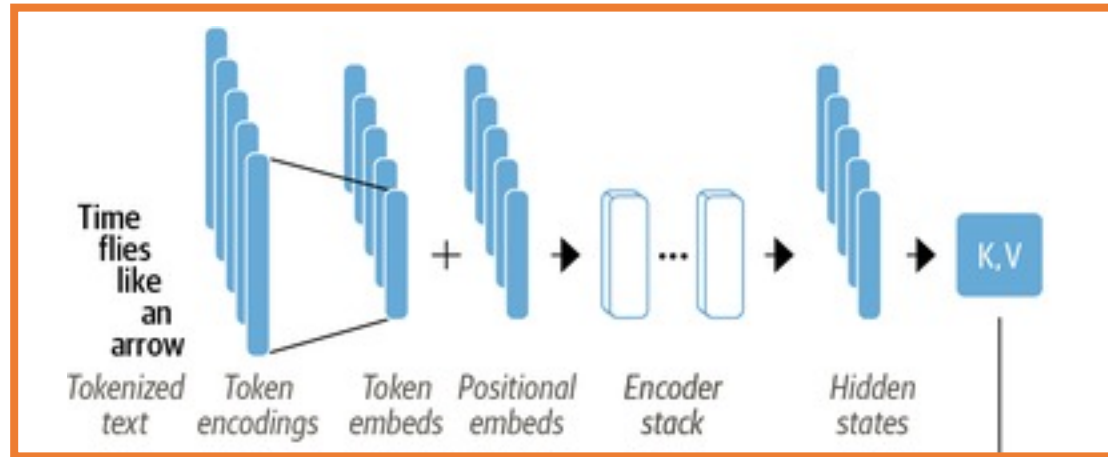
# Transformer pretraining



... or to predict the masked token (BERT-like)

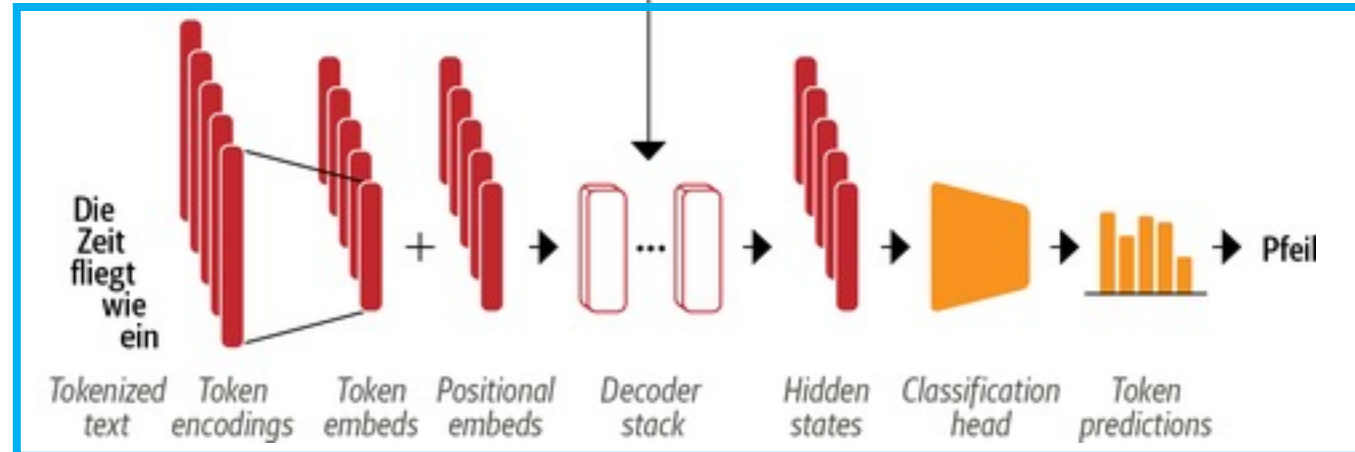


# The first Transformer architecture (2017)



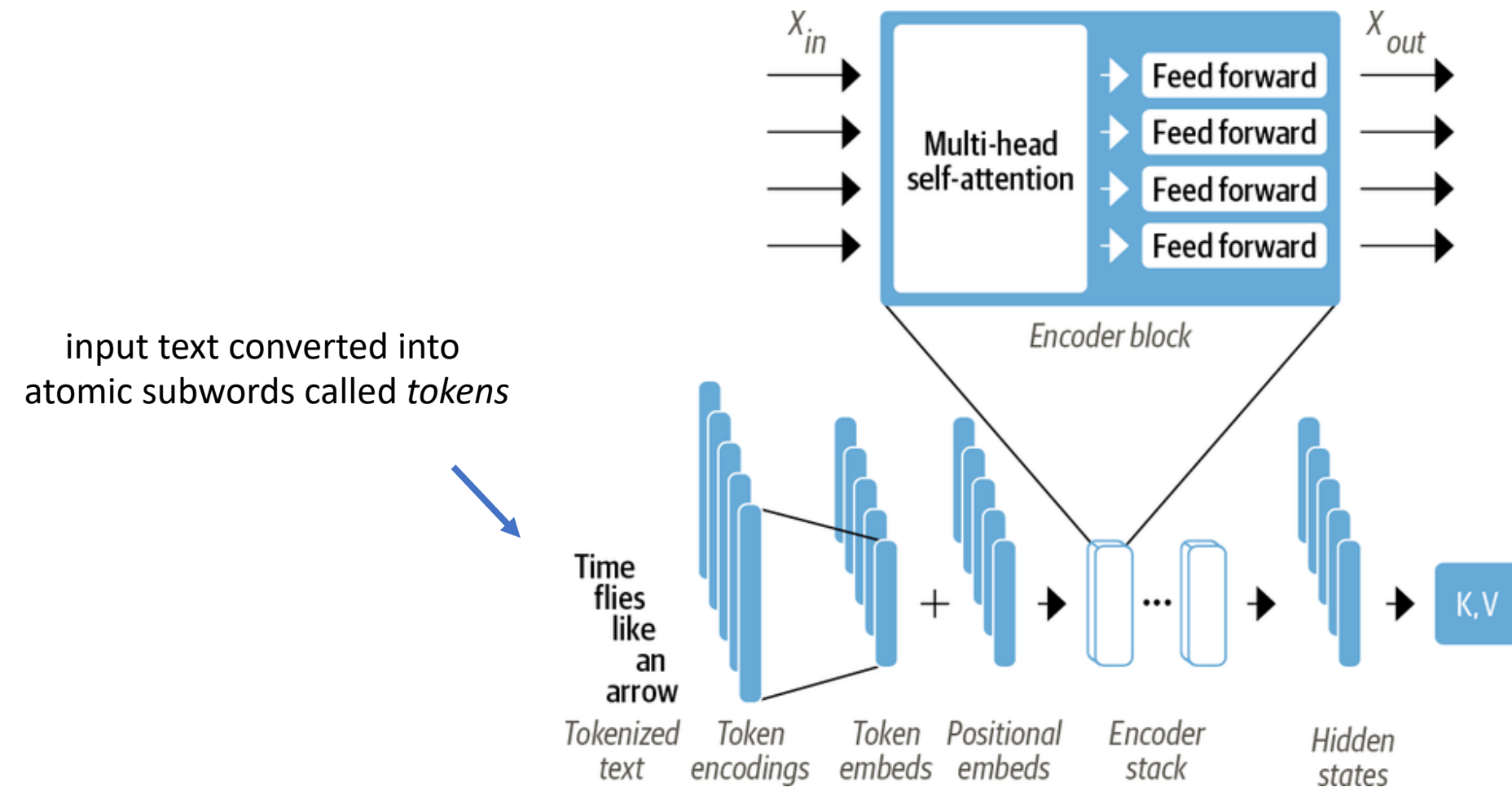
*Encoder* converts sequence of tokens to sequence of embedding vectors (context)

*Decoder* uses context to iteratively generate output sequence



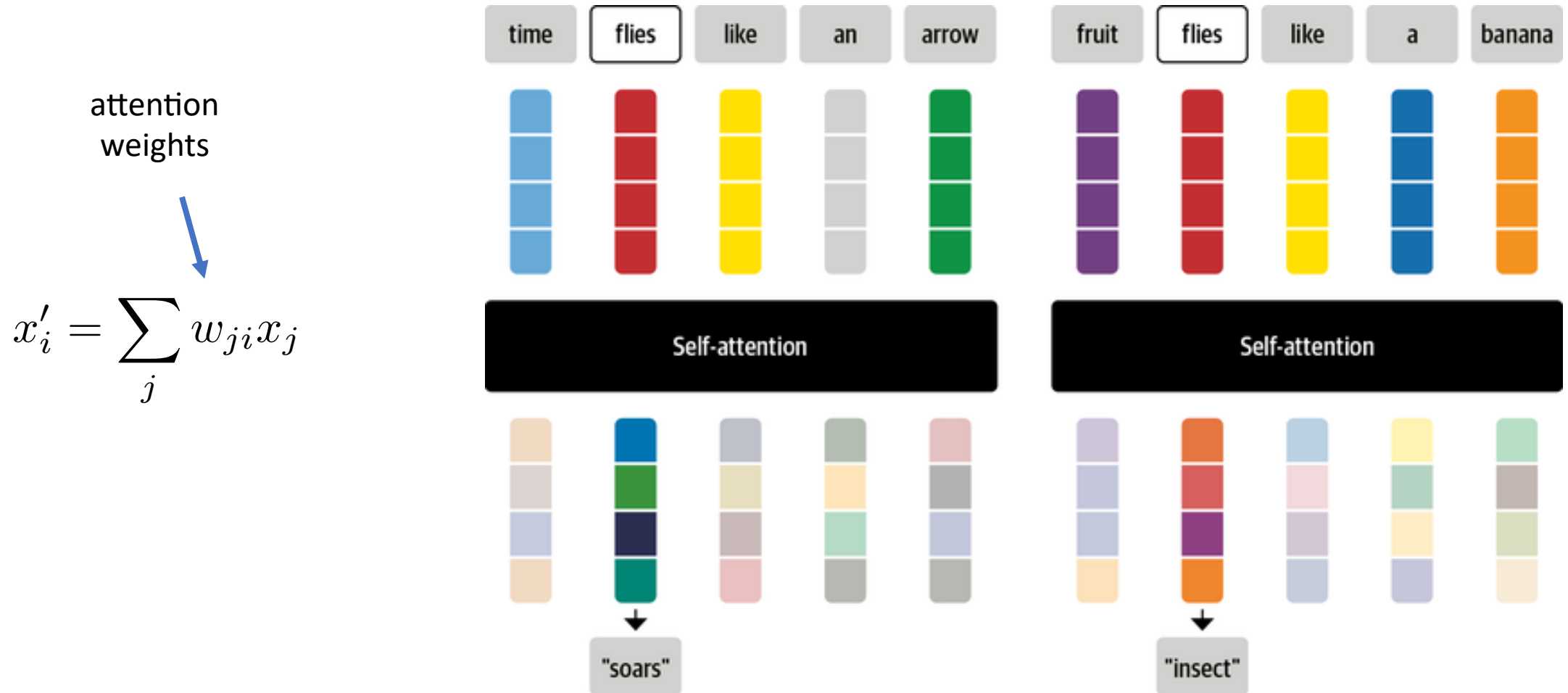
Consists of an *encoder* and *decoder*

# The encoder



Built from of a stack of *encoder layers* (similar to stacking convolutional layers)

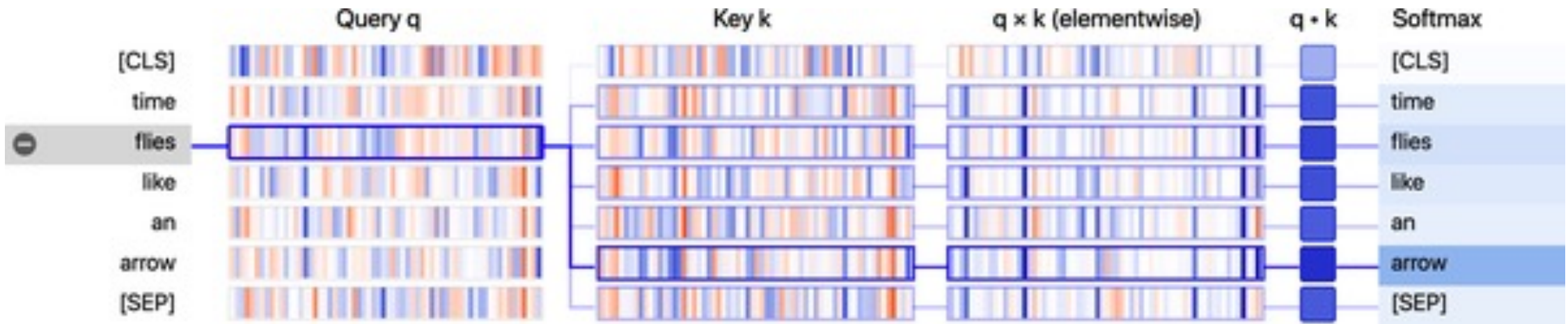
# Self-attention layers



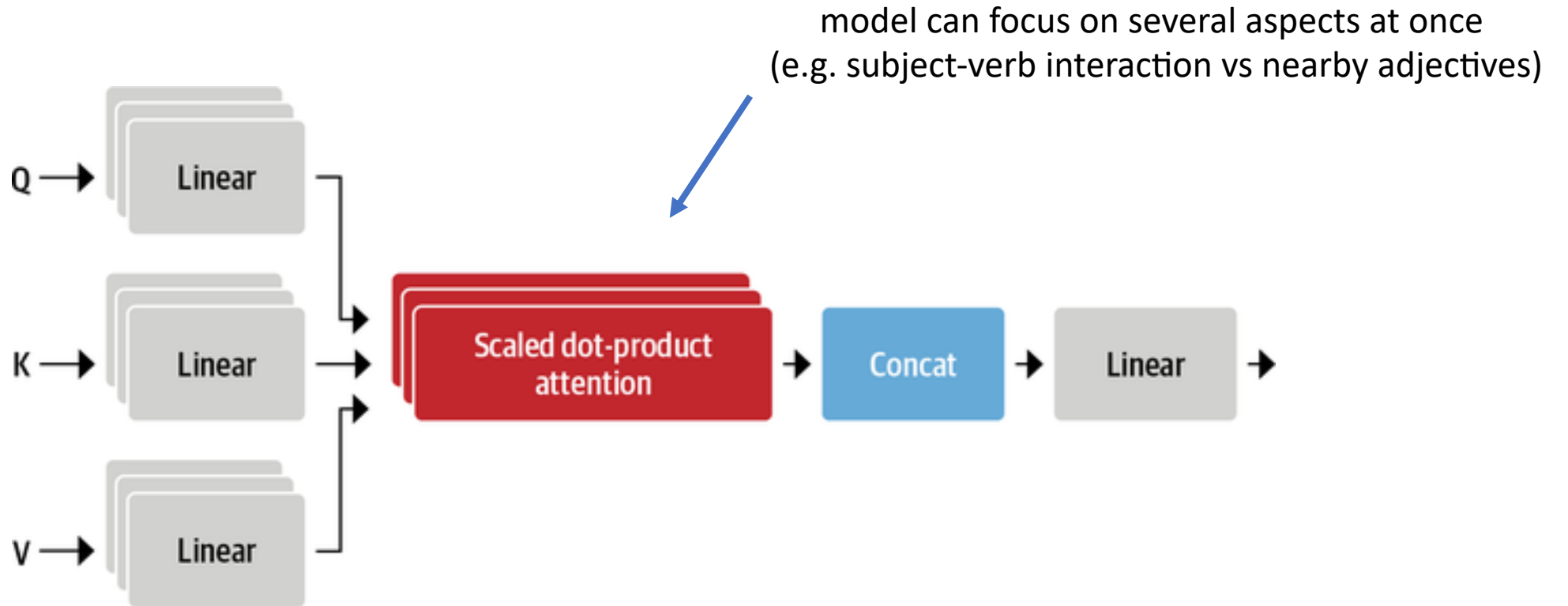
Self-attention updates *input* embeddings  $x$  into *contextualised* ones  $x'$

# Computing attention weights

1. Project each token embedding into 3 vectors called *query*, *key*, and *value*
2. Compute pairwise attention scores of *queries* and *keys* via similarity function (e.g. dot-product)
3. Compute attention weights  $w_{ji}$  (normalize with softmax)
4. Update token embeddings  $x'_i = \sum w_{ji}v_j$

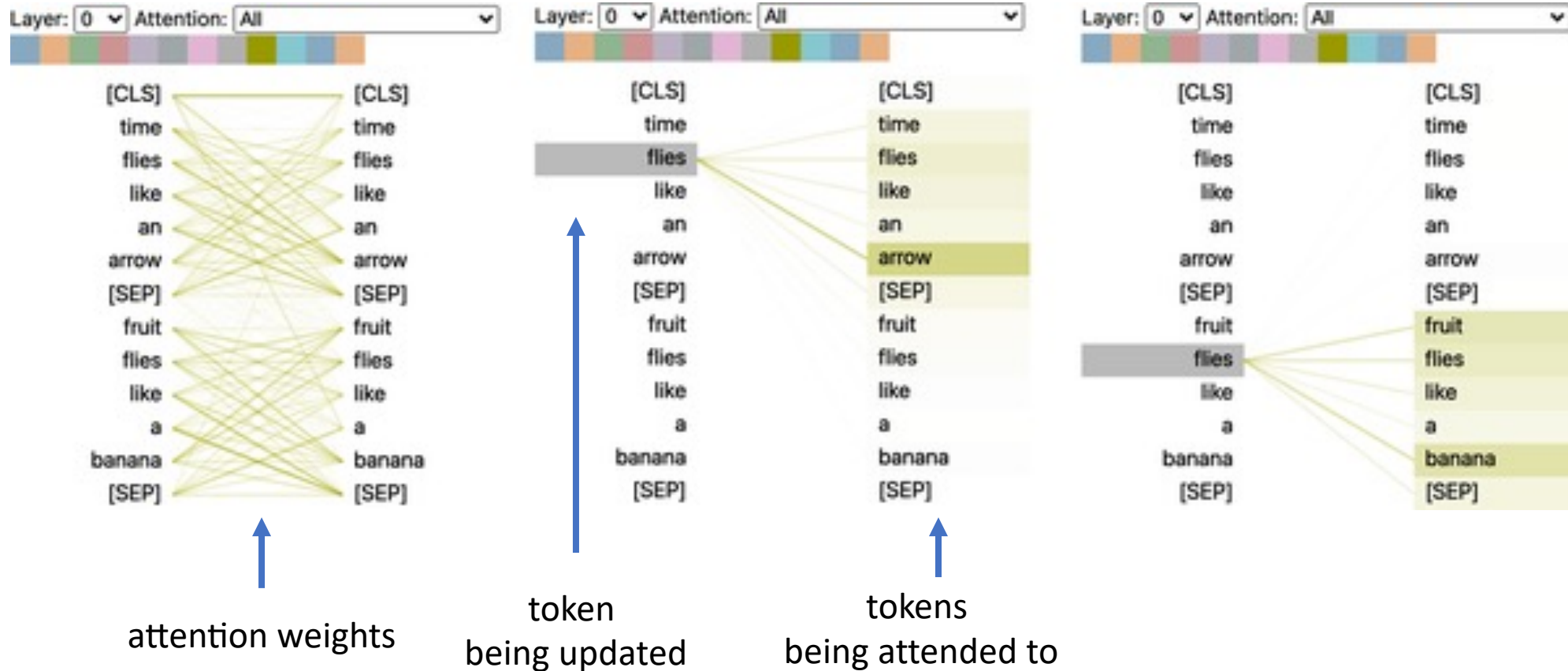


# Multi-headed attention



Beneficial to have multiple attention layers or “heads”

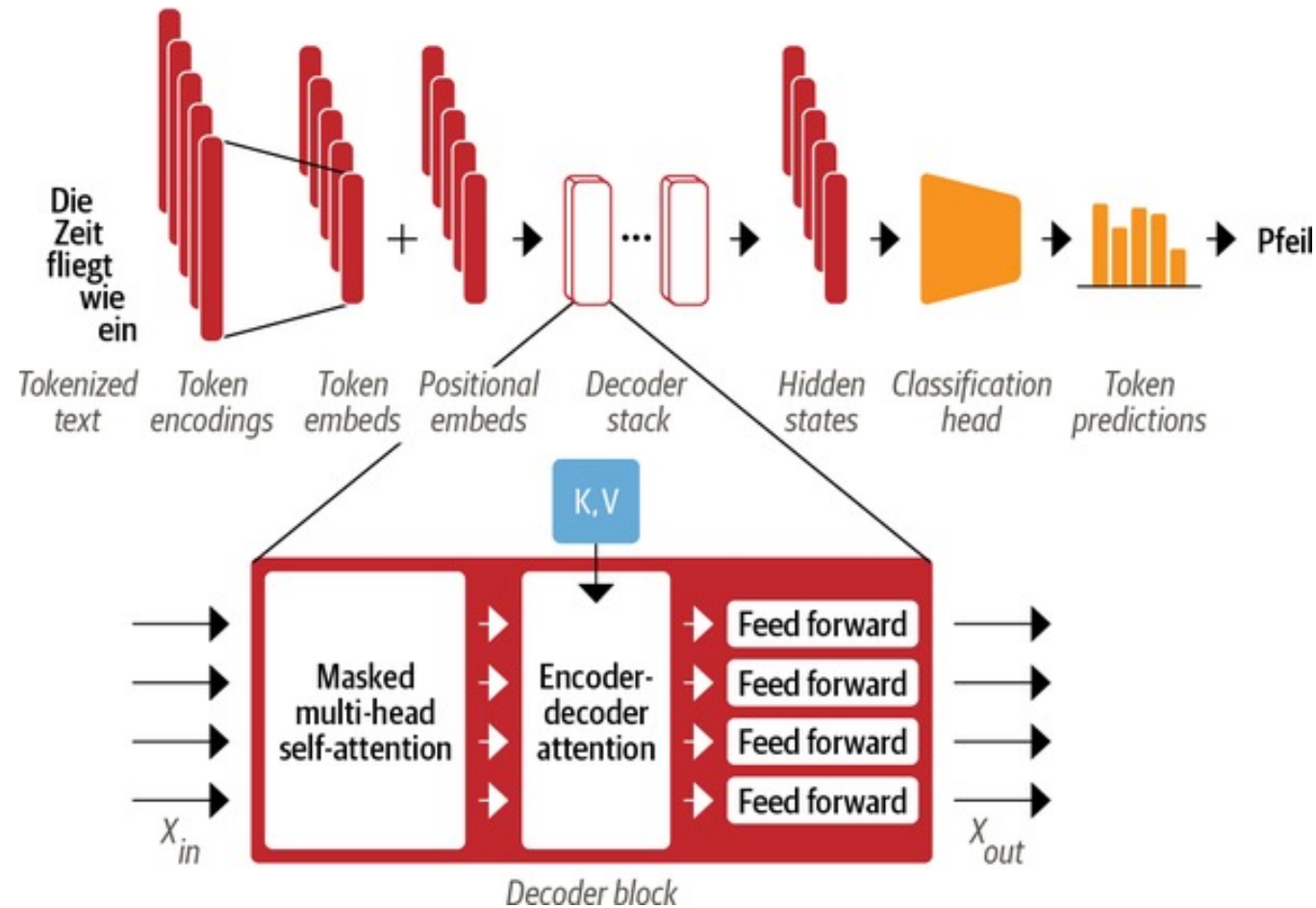
# Multi-headed attention



Self-attention updates *input* embeddings into *contextualised* ones

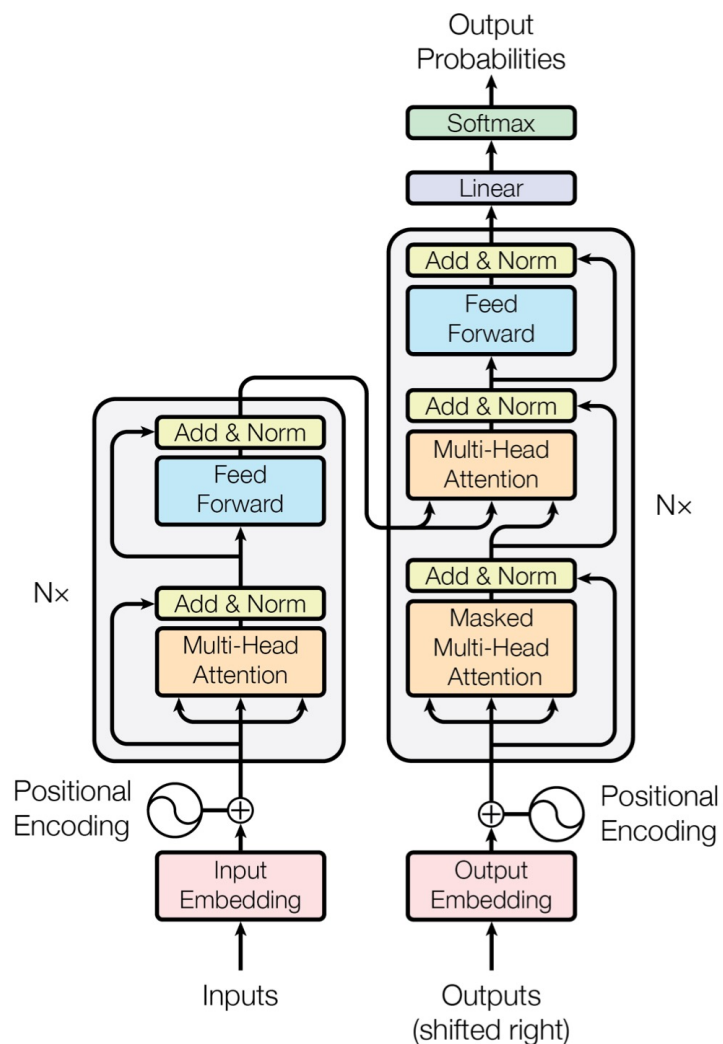


# The decoder



Similar to encoder but built from of a stack of *decoder layers*

# What is a Transformer?



---

## 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  
lukaszkaizer@google.com

**Illia Polosukhin\* ‡**  
illia.polosukhin@gmail.com