Attention mechanism

Alberto TONDA, Ph.D. (Senior permanent researcher, DR)

UMR 518 MIA-PS, INRAE, AgroParisTech, Université Paris-Saclay
UAR 3611, Institut des Systèmes Complexes de Paris Île-de-France

Outline

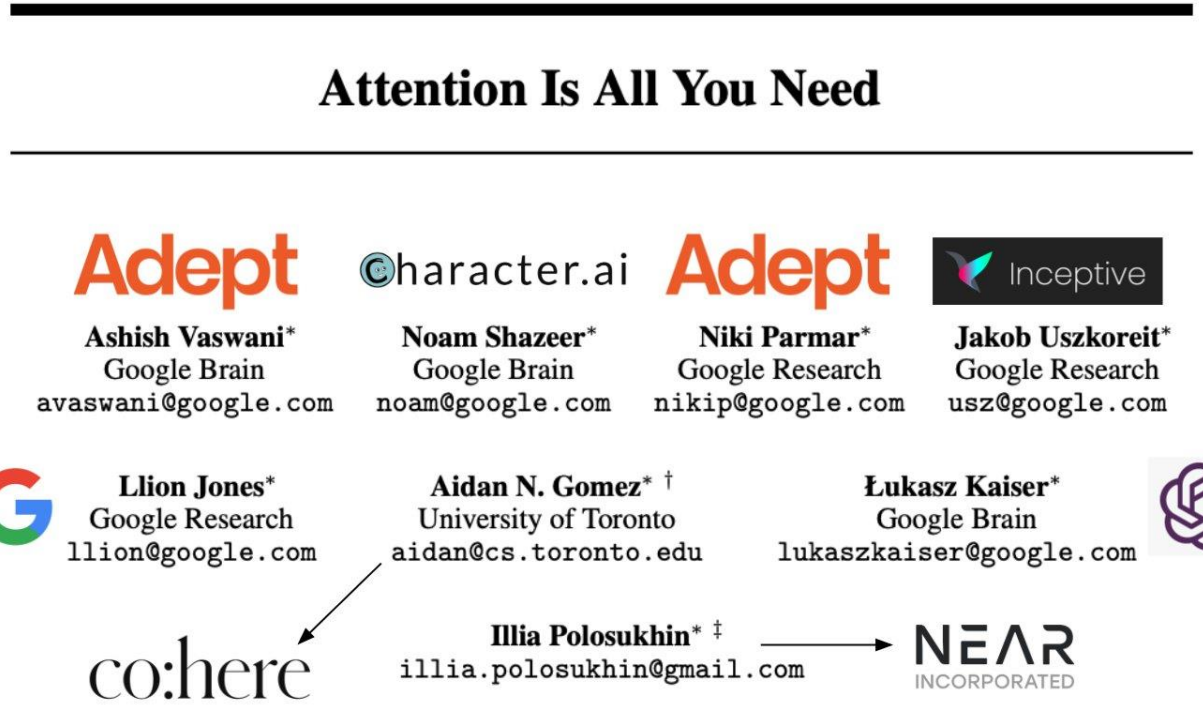
- What is Attention?
- Explaining Attention
- Attention head
- Attention-based architectures

What is Attention?

- When dealing with sequences
  - Elements away from current position might have **strong impact**!
  - E.g. "**Don't** go insane, find the real identity of, and kill one by one all the reviewers that have just rejected your awesome paper."
  - "Don't" here is extremely important for meaning of sentence

- Seq2seq architectures (even LSTMs) tend to *lose memory*
  - Start of sequence has *low impact* on long sequences
  - Old information tends to be lost with updates of hidden state

What is Attention?



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

What is Attention?

Attention is all you need [PDF] neurips.cc

A Vaswani, N Shazeer, N Parmar... - Advances in neural ..., 2017 - proceedings.neurips.cc

... to attend to **all** positions in the decoder up to and including that position. **We need** to prevent

... **We** implement this inside of scaled dot-product **attention** by masking out (setting to $-\infty$) ...

☆ Salva 99 Cita Citato da 206686 Articoli correlati Tutte e 70 le versioni »

Adept — **Ashish Vaswani*** Google Brain avaswani@google.com

character.ai — **Noam Shazeer*** Google Brain noam@google.com

Adept — **Niki Parmar*** Google Research nikip@google.com

Inceptive — **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

co:here — **Illia Polosukhin*** ‡ illia.polosukhin@gmail.com → NEAR INCORPORATED

Emulation is the sincerest form of flattery

Cognition is All You Need

The Next Layer of AI Above Large Language Models

Pre-Publication Position Paper Draft 1.1 March 4, 2024, For Comments

Nova Spivack[1], Sam Douglas[1], Michelle Crames[1], Tim Connors[1]

Pretraining on the Test Set Is All You Need

Rylan Schaeffer

September 19, 2023

Abstract

Inspired by recent work demonstrating the promise of smaller Transformer-based language models pretrained on carefully curated data, we supercharge such approaches by investing heavily in curating a novel, high quality, non-synthetic data mixture based solely on evaluation benchmarks. Using our novel dataset mixture consisting of less than 100 thousand tokens, we pretrain a 1 million parameter transformer-based LLM **phi-CTNL** (pronounced "fictional") that achieves perfect results across diverse academic benchmarks, strictly outperforming all known foundation models. **phi-CTNL** also beats power-law scaling and exhibits a never-before-seen grokking-like ability to accurately predict downstream evaluation benchmarks' canaries.

Theory Is All You Need: AI, Human Cognition, and Decision Making †

Teppo Felin
Utah State University
& Oxford University

Matthias Holweg
Oxford University

Money Is All You Need

Nick Debu
Tokyo Institute of Bamboo Steamer

Abstract

Transformer-based models routinely achieve state-of-the-art results on a number of tasks but training these models can be prohibitively costly, especially on long sequences. We introduce one technique to improve the performance of Transformers. We replace NVIDIA P100s by TPUs, changing its memory from hoge GB to piyo GB. The resulting model performs on par with Transformer-based models while being much more ""TSUYO TSUYO"".

(Computer scientists think they are funny)

- For example, here is the CV of the lead author of YOLO, the CNN for semantic segmentation

Obstacles in explaining Attention

- Architecture of an Attention module has two aims
  - Follow a certain **idea** of what it should do (“what”)
  - Maximize computational **speed** (“how”)
  - The practical implementation can be hard to follow
  - Computations of Attention can be *massively* parallelized

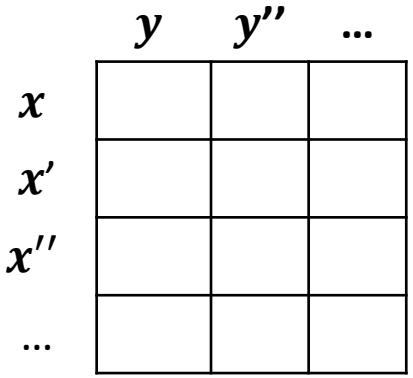
“When it does not make *sense*, it makes *speed*”

Similarity between vectors

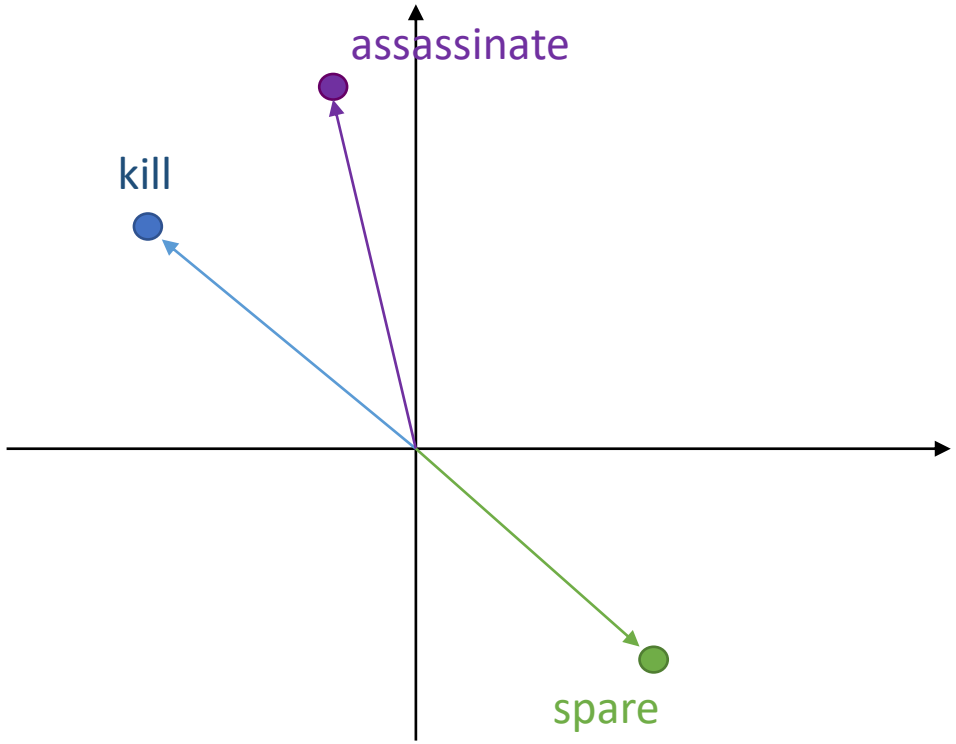
- Vectors == Points in high-dimensional space, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$
- How to evaluate similarity between vectors?
  - Euclidean distance is not bad, but *small* if *similar*
  - Dot product is **large** if **similar**, *small* if *not similar*
  - And can be computed with a matrix multiplication!

$$\boldsymbol{x} \cdot \boldsymbol{y} = \sum_{i=o}^{d} x_i \cdot y_i = \boldsymbol{x}\boldsymbol{y}^T$$

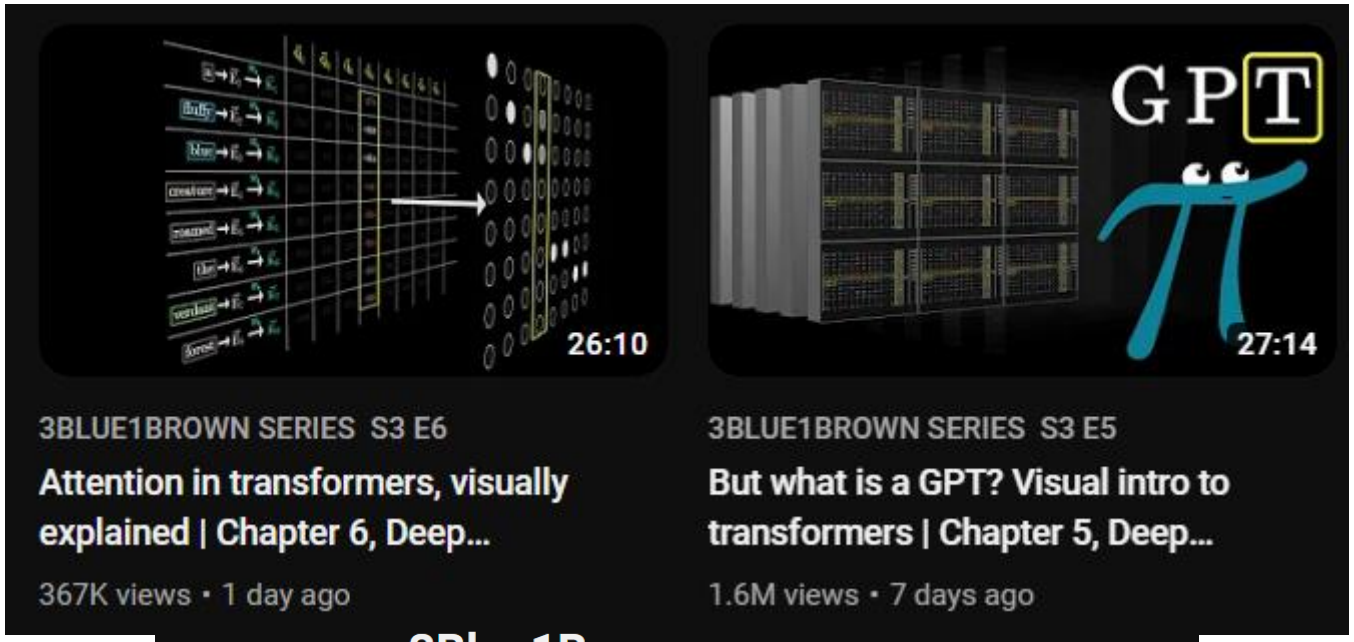
$$\boldsymbol{X} \cdot \boldsymbol{Y} = \boldsymbol{X}\boldsymbol{Y}^T$$

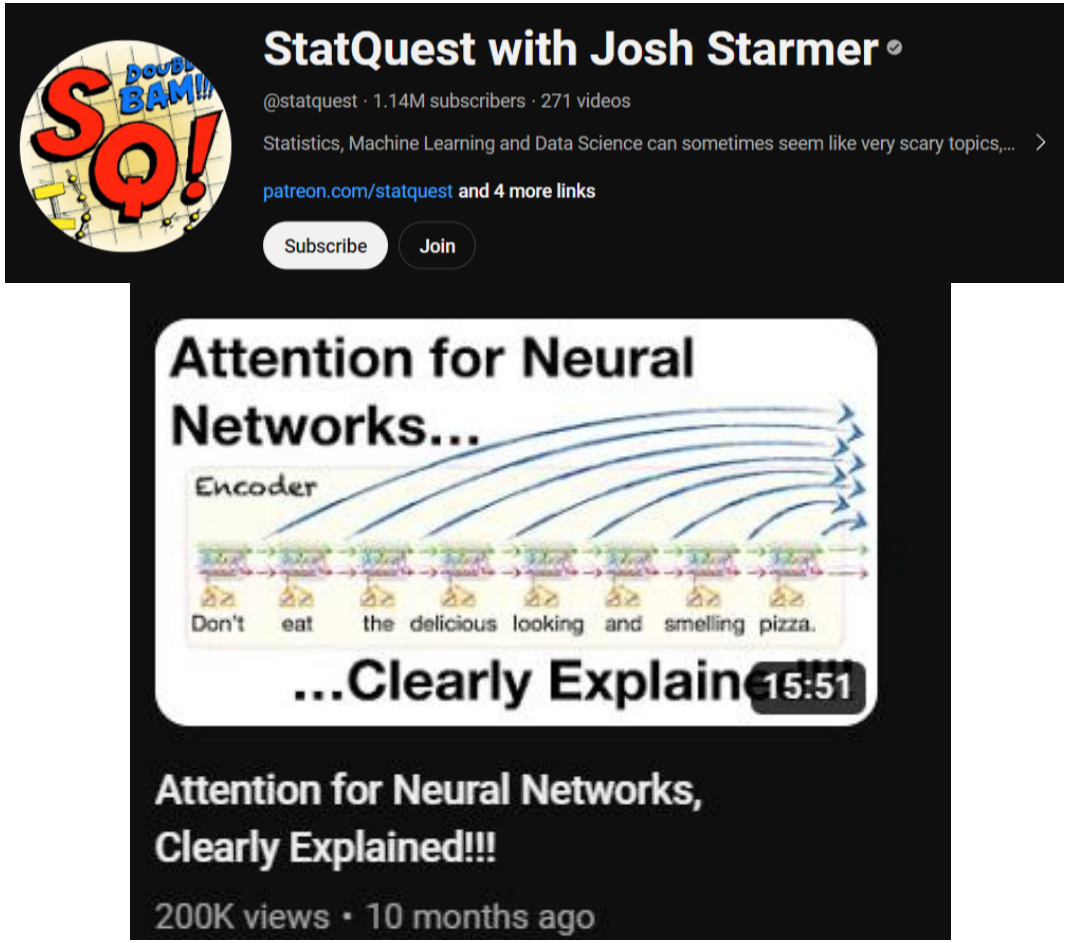


Similarity between vectors



Attention, explained better than here





Attention, explained better than here

Attention

- Attention is specifically designed to work on **vocabularies**
  - More specifically, *embeddings* of vocabularies
  - It can only work between two finite sets of discrete elements
  - Among elements of the same sequence (**self-attention**)
  - But sometimes, different sets/sequences (**cross-attention**)

Encoding the way in which the presence of nearby tokens modifies the meaning (embedding) of a target token, for all tokens in the context window

Attention

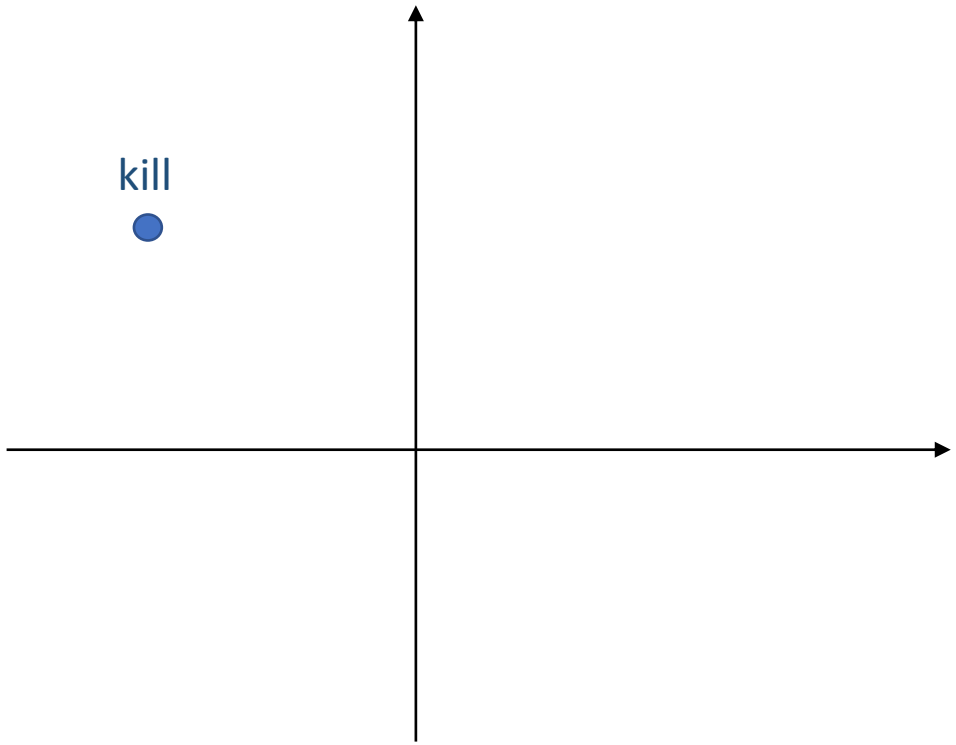
- Attention is specifically designed to work on **vocabularies**
  - More sp
  - It can on
  - Among e
  - But sometimes, di quences (**cross-attention**)

Meaning is encoded as a **position** in the embedding vector space; so, in practice, Attention computes a **displacement** in the embedding vector space that reflects the **shift in meaning** of the token, given the context.

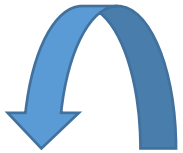
Encoding the way in which the presence of nearby tokens **modifies the meaning** (embedding) of a target token, for all tokens in the context window

Attention

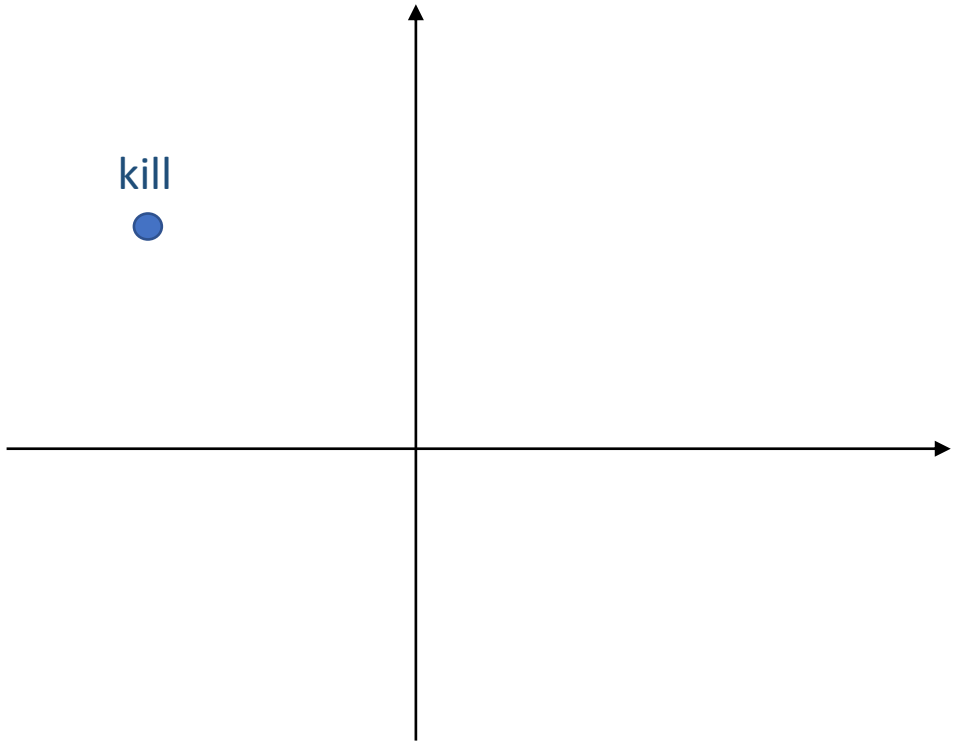
"Don't go insane, find the real identity of, and kill one by one all the reviewers that have just rejected your awesome paper."



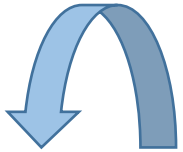
Attention



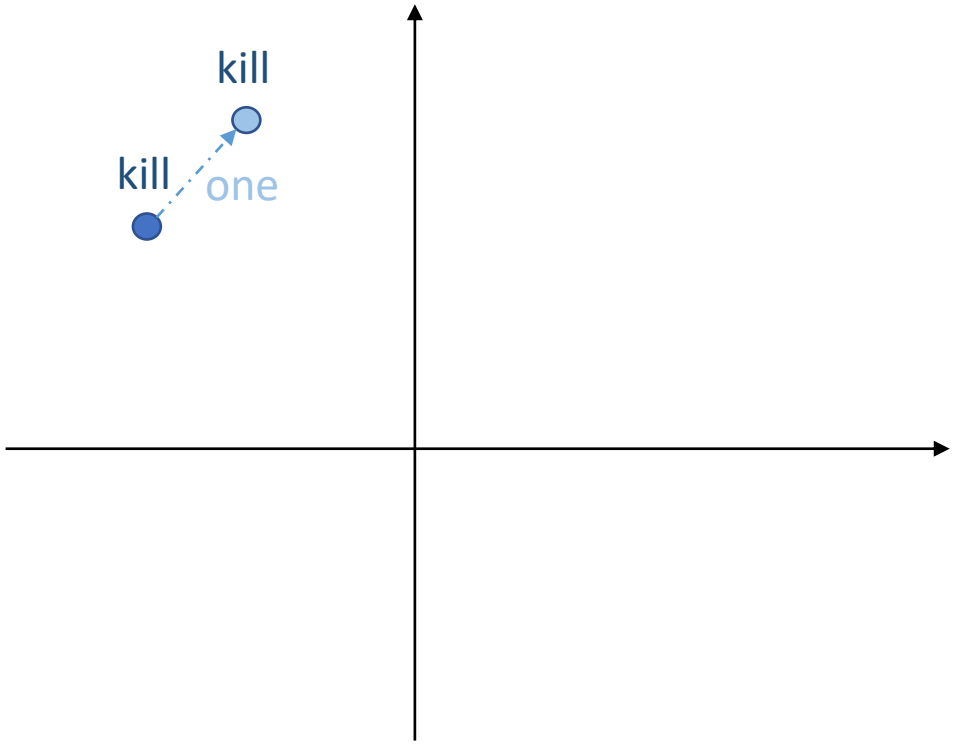
"Don't go insane, find the real identity of, and **kill** **one** by one all the reviewers that have just rejected your awesome paper."



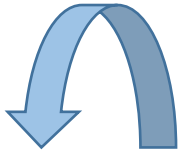
Attention



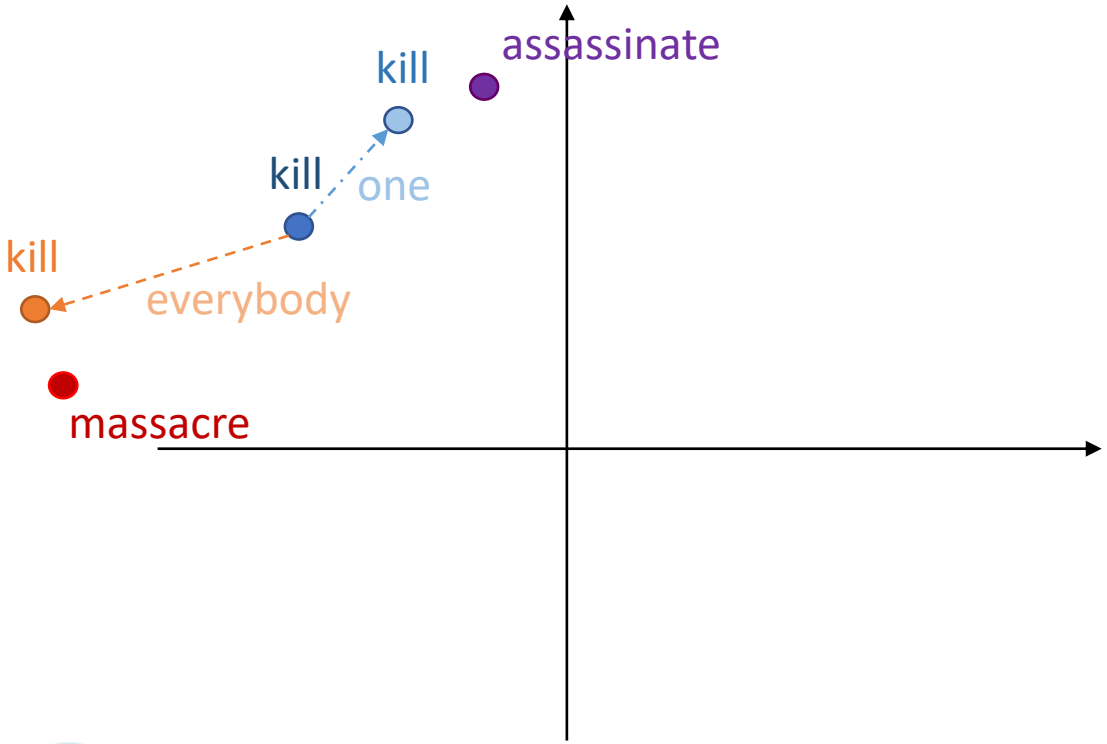
"Don't go insane, find the real identity of, and **kill** one by one all the reviewers that have just rejected your awesome paper."



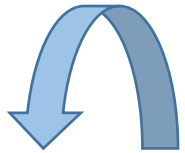
Attention



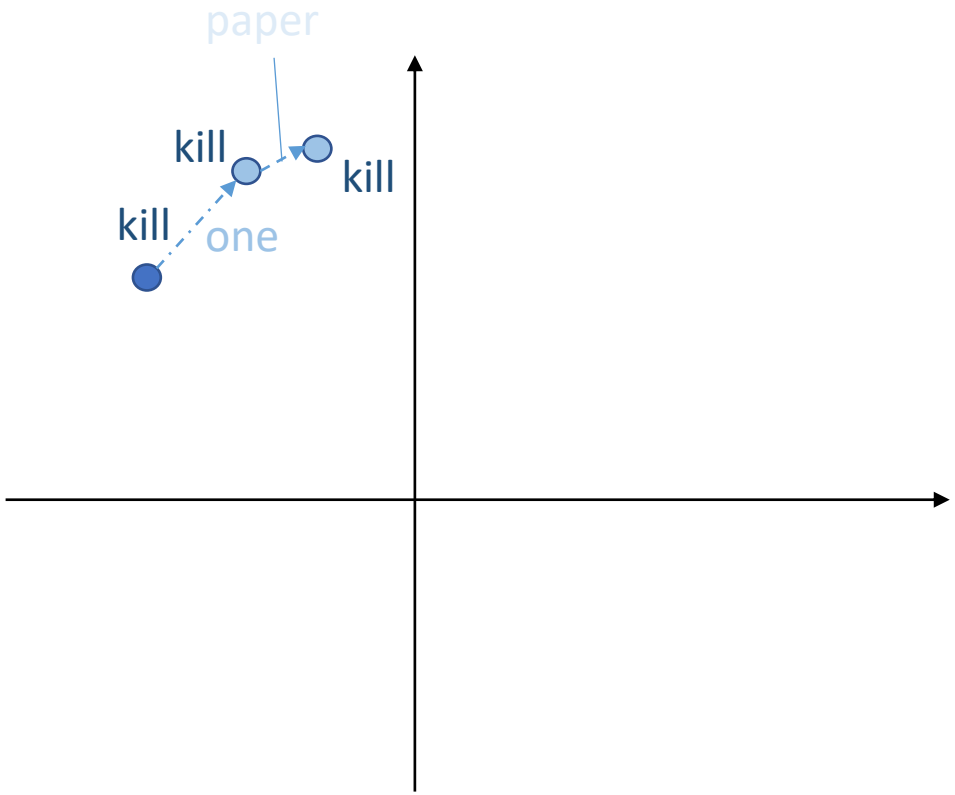
"Don't go insane, find the real identity of, and **kill** one by one all the reviewers that have just rejected your awesome paper."



Attention

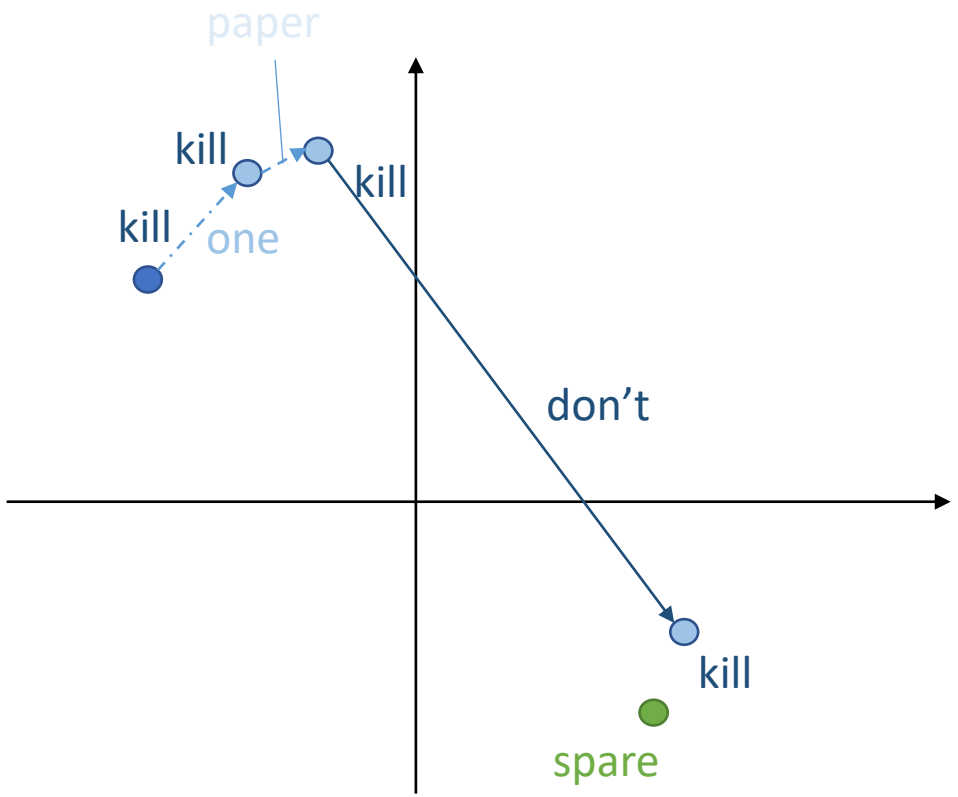


“Don’t go insane, find the real identity of, and kill one by one all the reviewers that have just rejected your awesome paper.”



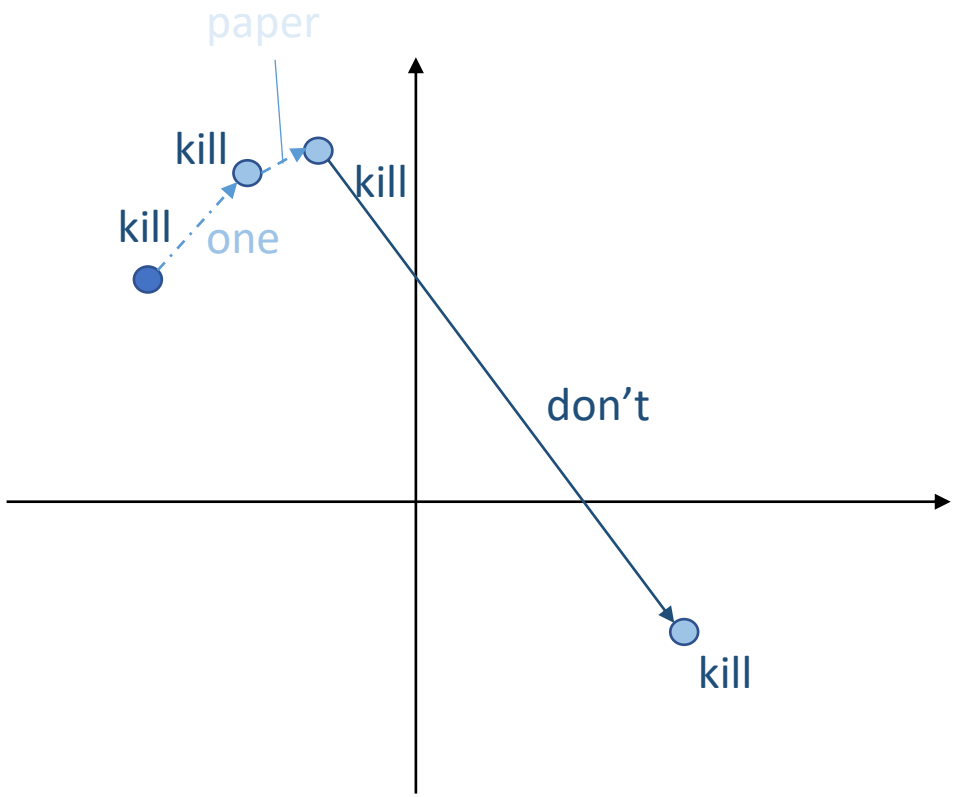
Attention

"**Don't** go insane, find the real identity of, and **kill** **one** by one all the reviewers that have just rejected your awesome **paper**."



Attention

"**Don't** go insane, find the real identity of, and **kill** **one** by one all the reviewers that have just rejected your awesome **paper**."



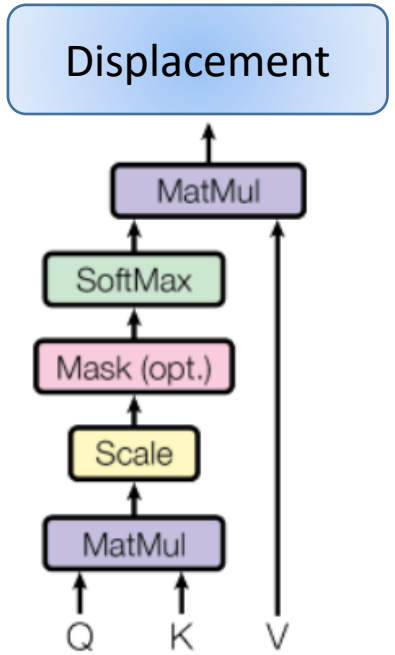
Attention module answers questions:

1. Which **other tokens** should modify target?
2. By **how much**?

Attention head at a glance

$$
\begin{aligned}
Q &= xW_Q, \\
K &= xW_K, \\
V &= xW_V,
\end{aligned}
$$

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

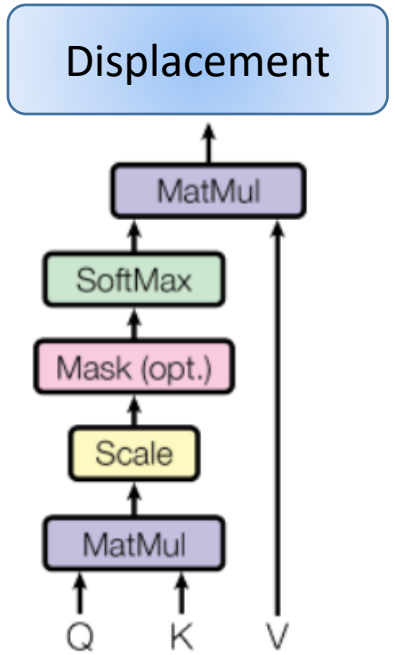


Attention head at a glance

$$Q = xW_Q,$$
$$K = xW_K,$$
$$V = xW_V,$$

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

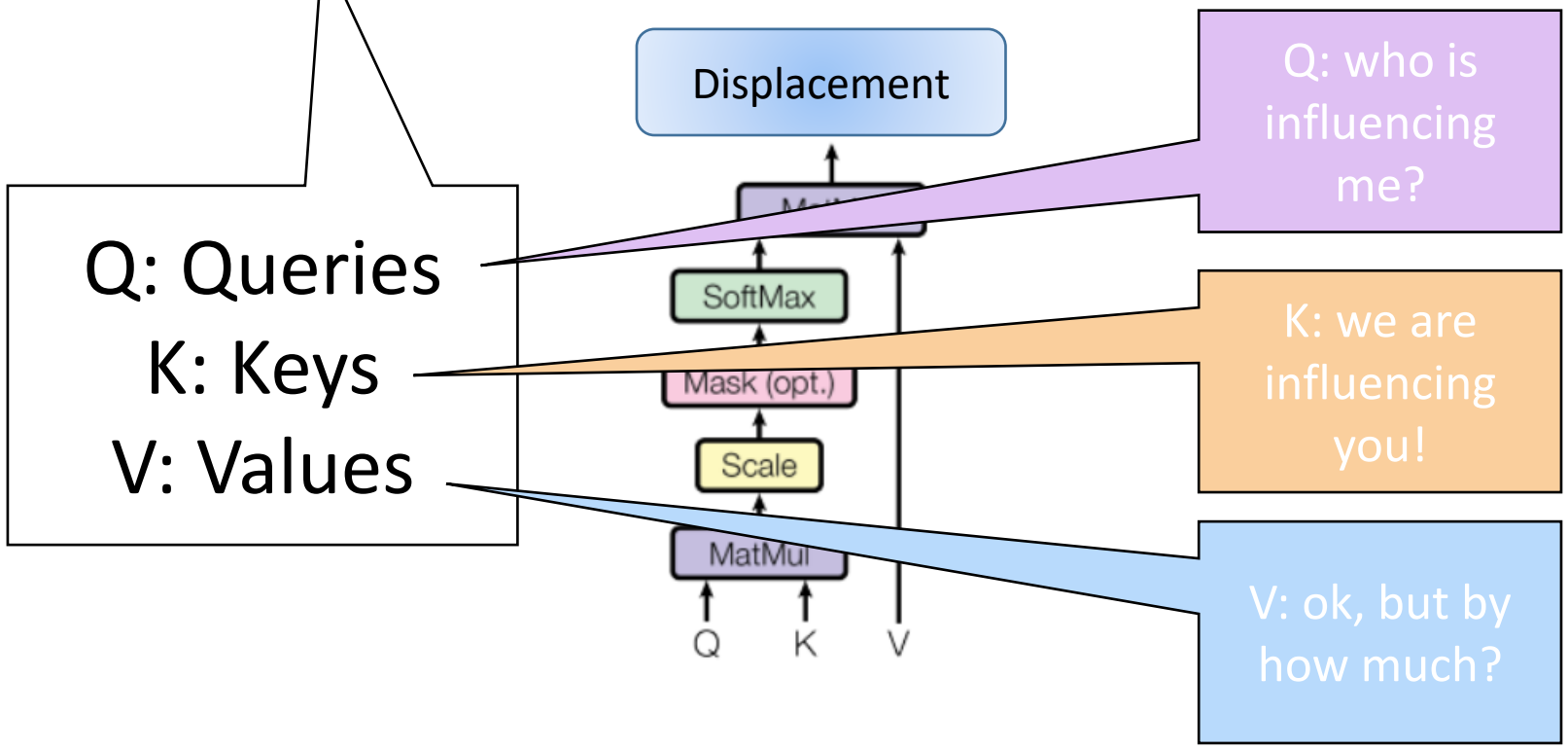
Q: Queries
K: Keys
V: Values



Attention head at a glance

$$Q = xW_Q,$$
$$K = xW_K,$$
$$V = xW_V,$$

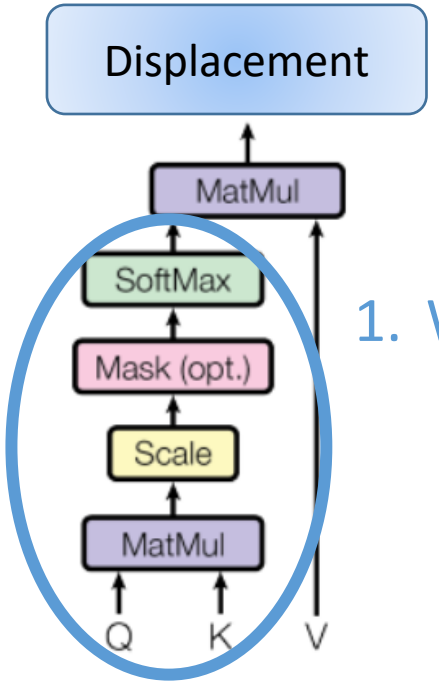
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Attention head at a glance

$$Q = xW_Q,$$
$$K = xW_K,$$
$$V = xW_V,$$

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

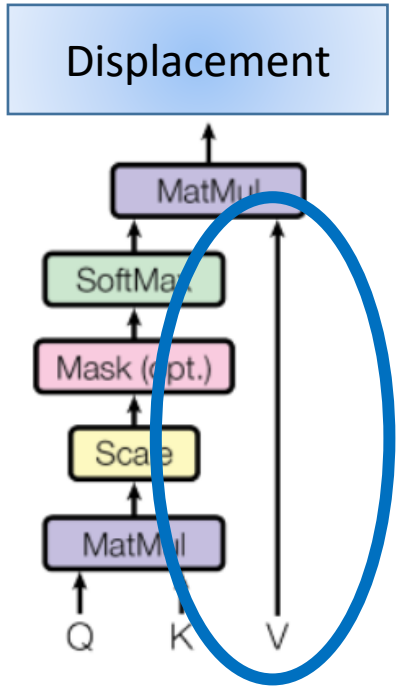


1. Which **other tokens** should modify target?

Attention head at a glance

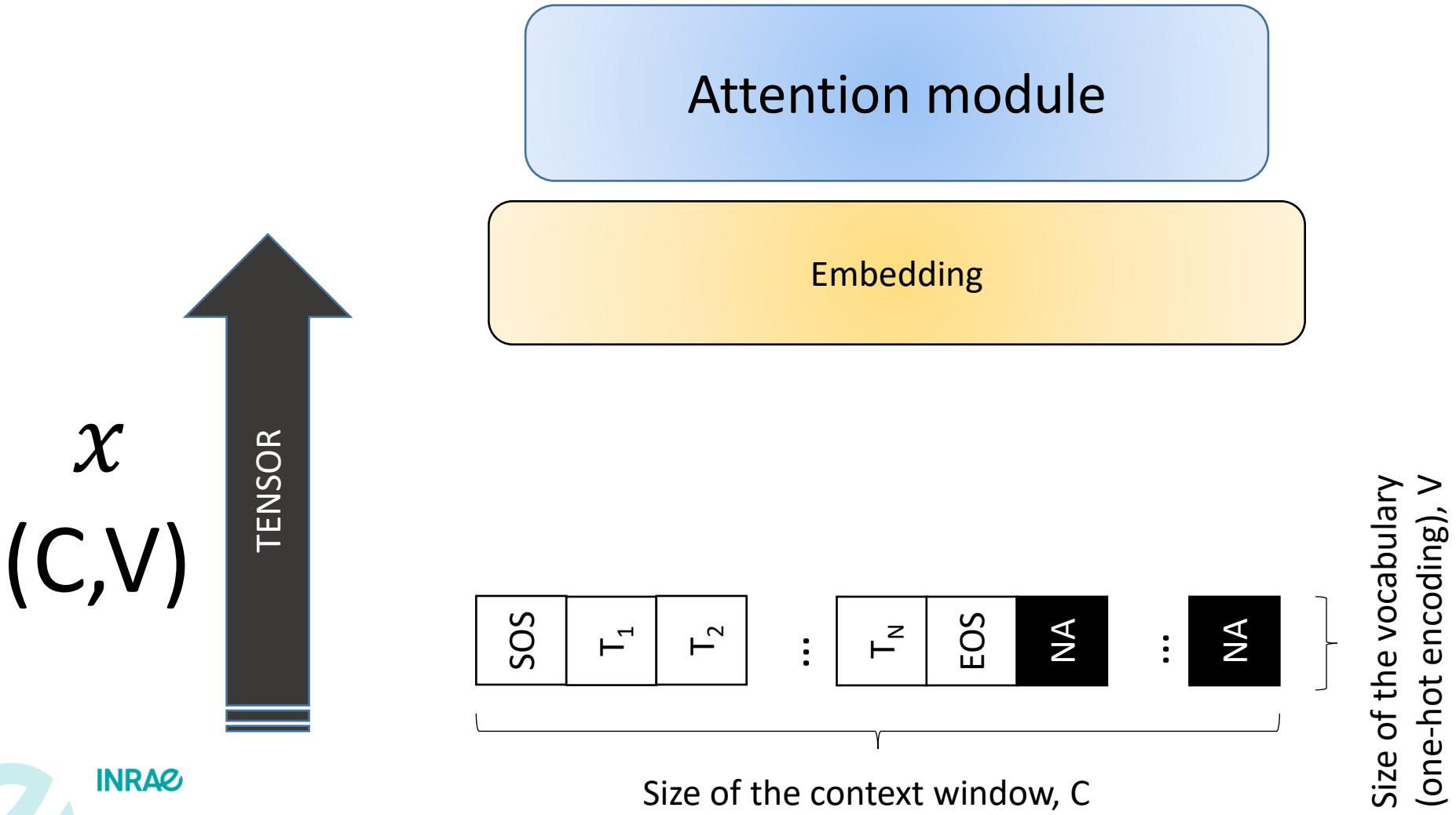
$$Q = xW_Q,$$
$$K = xW_K,$$
$$V = xW_V,$$

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

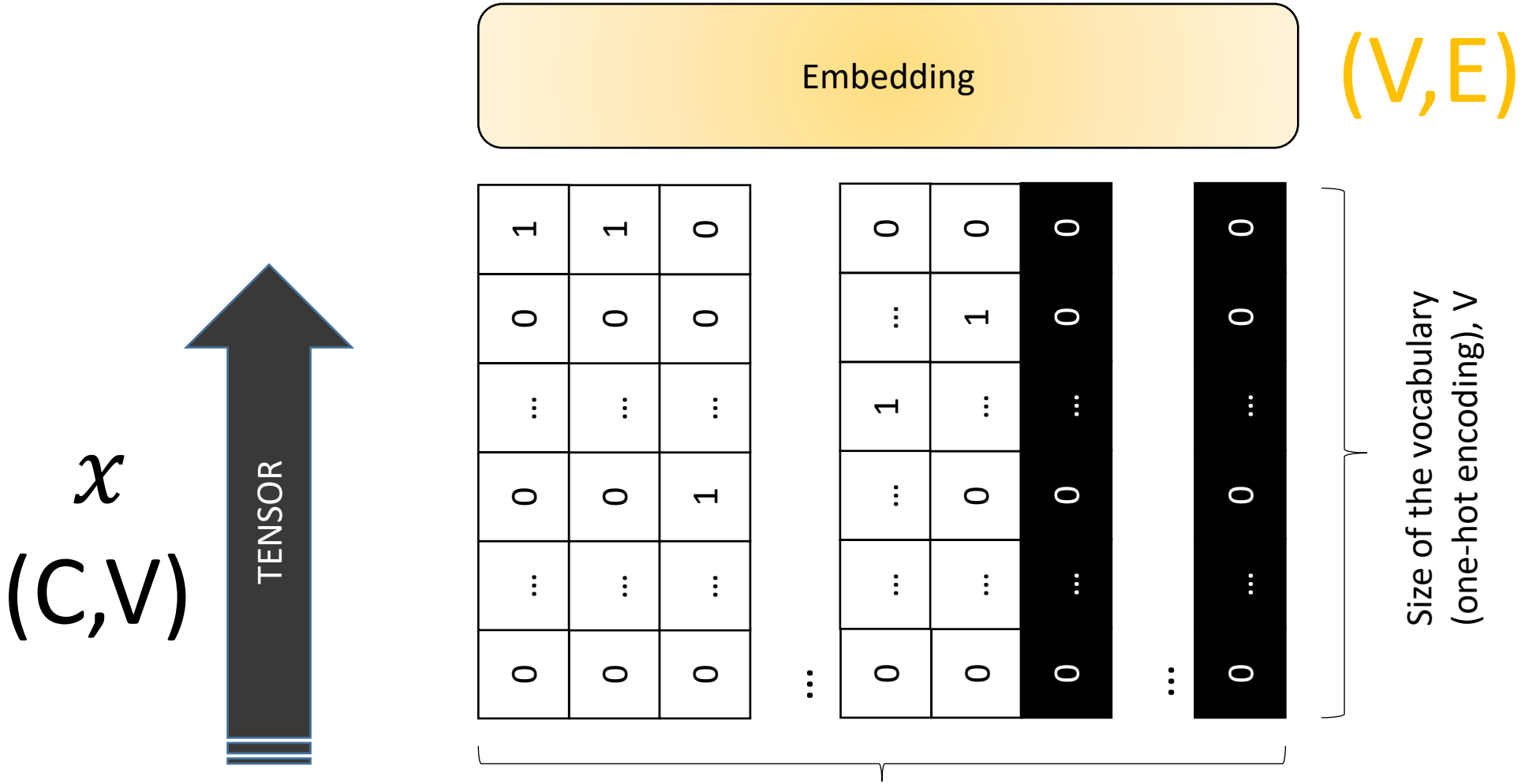


2. By how much?

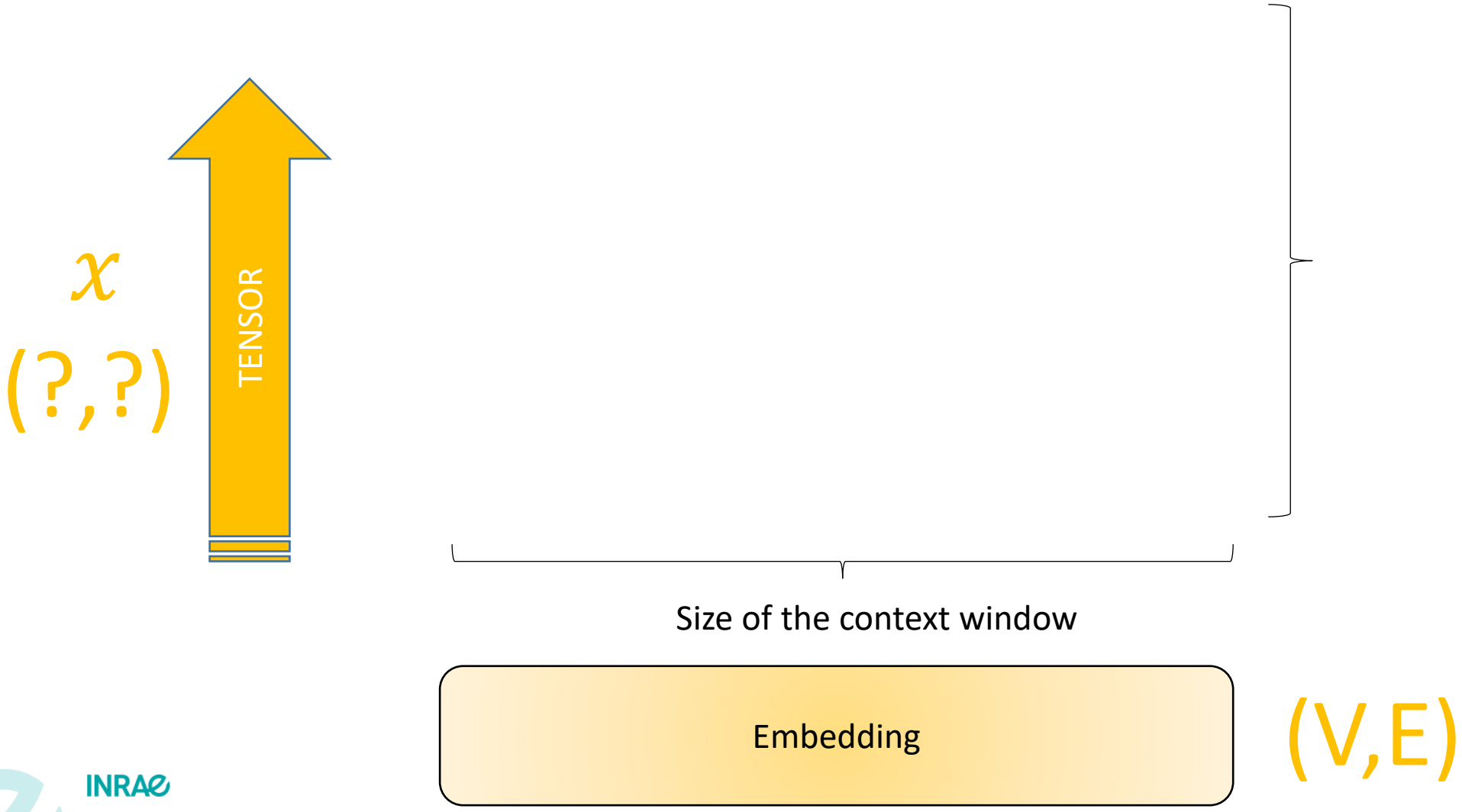
Attention head



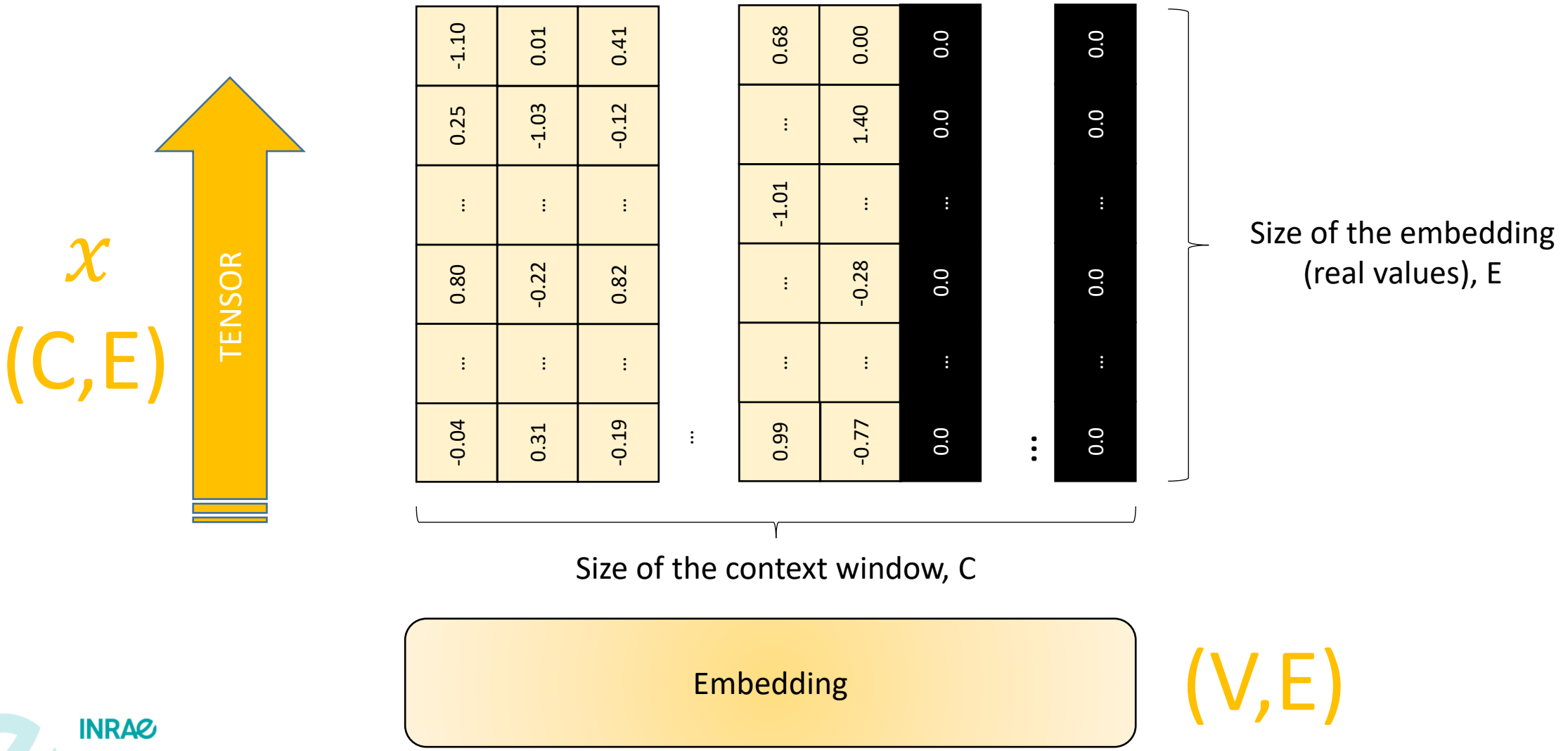
Attention head



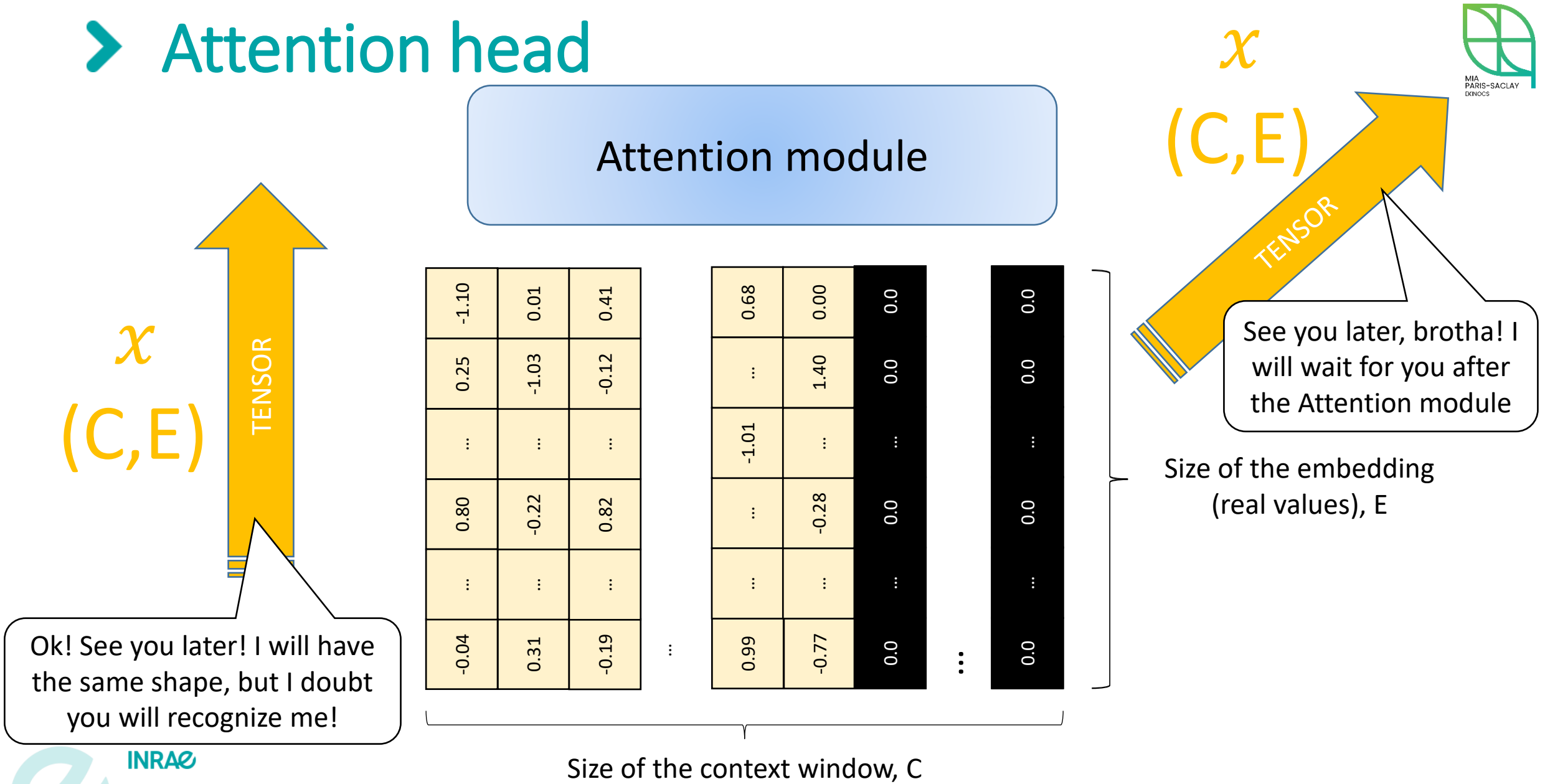
Attention head



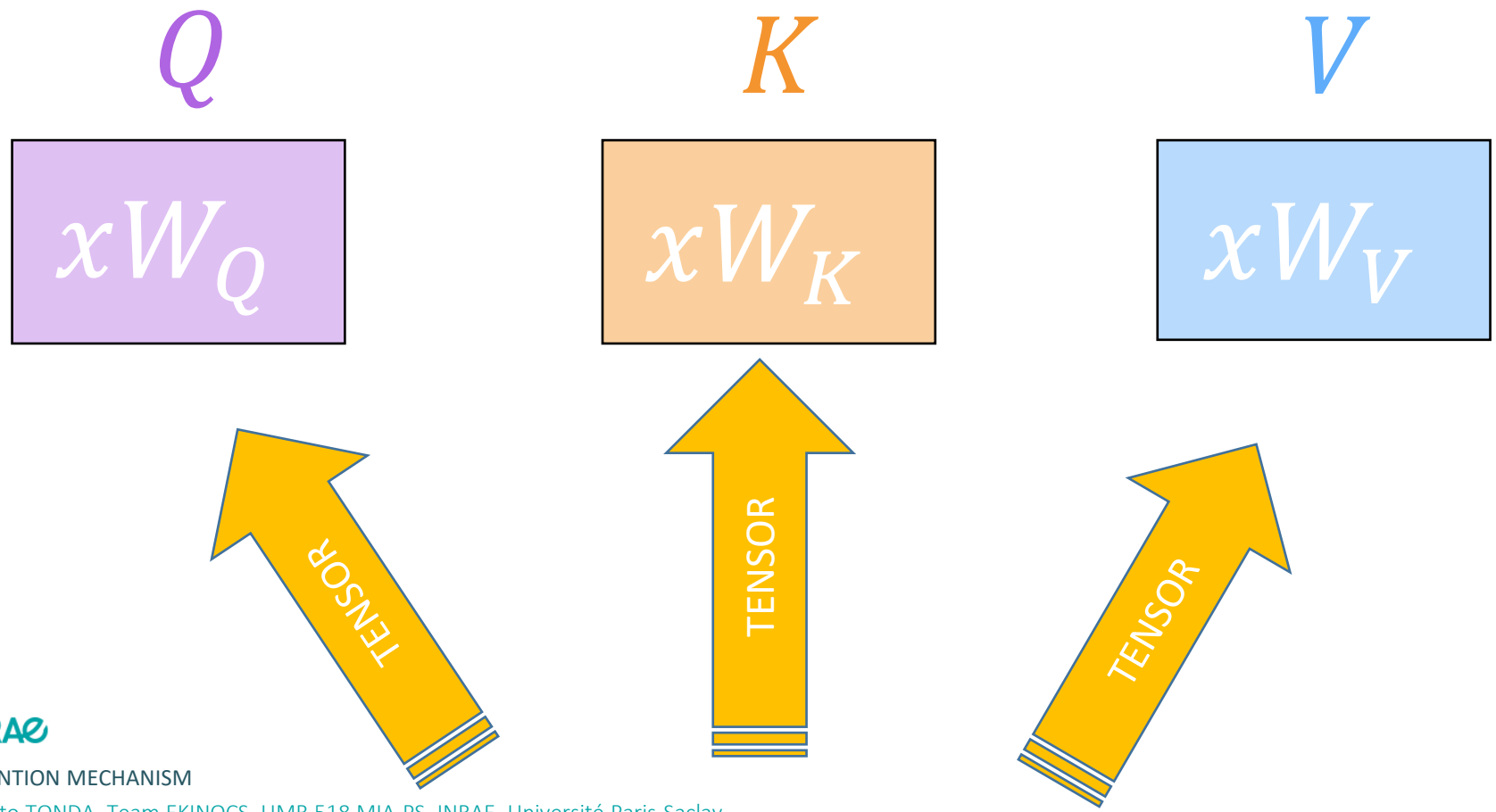
Attention head



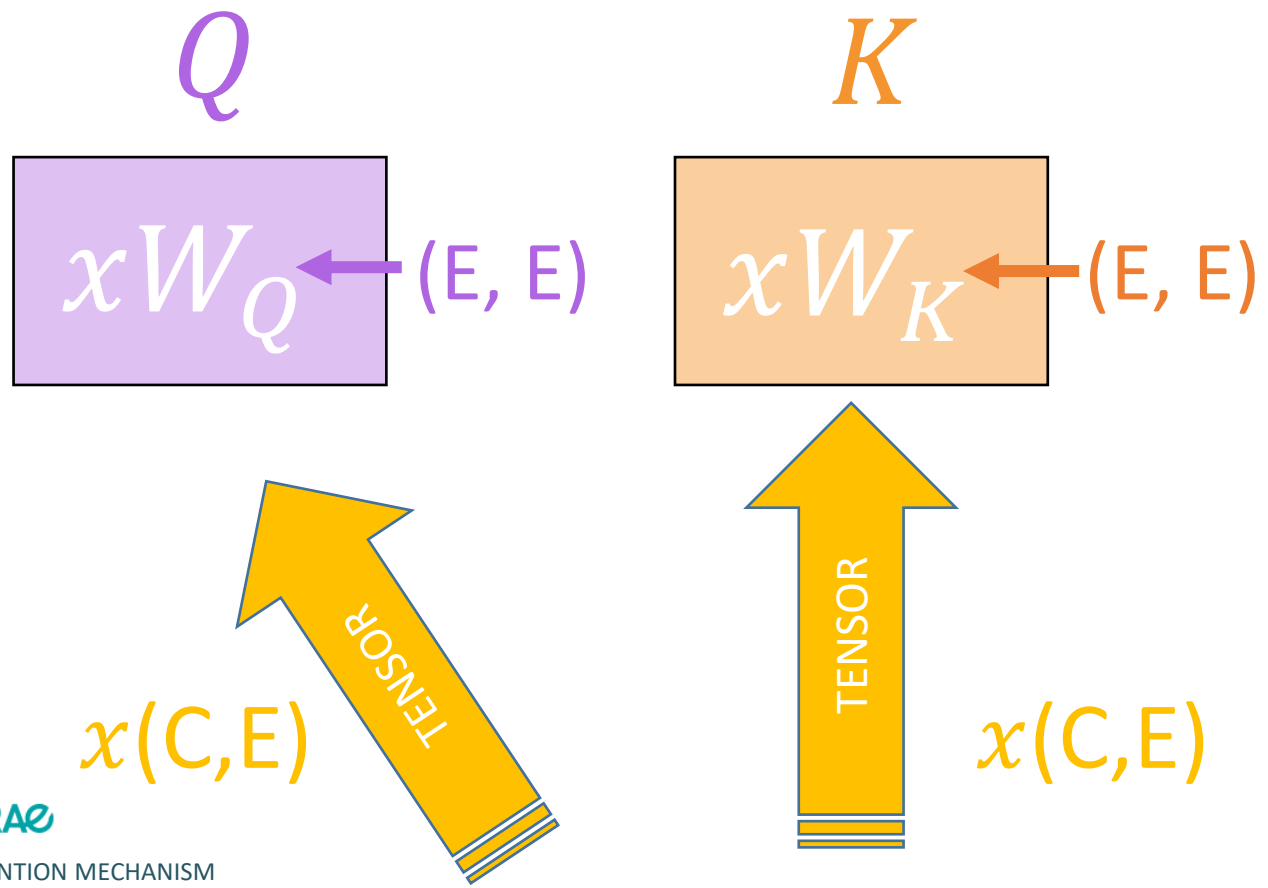
Attention head



Attention head



Attention head

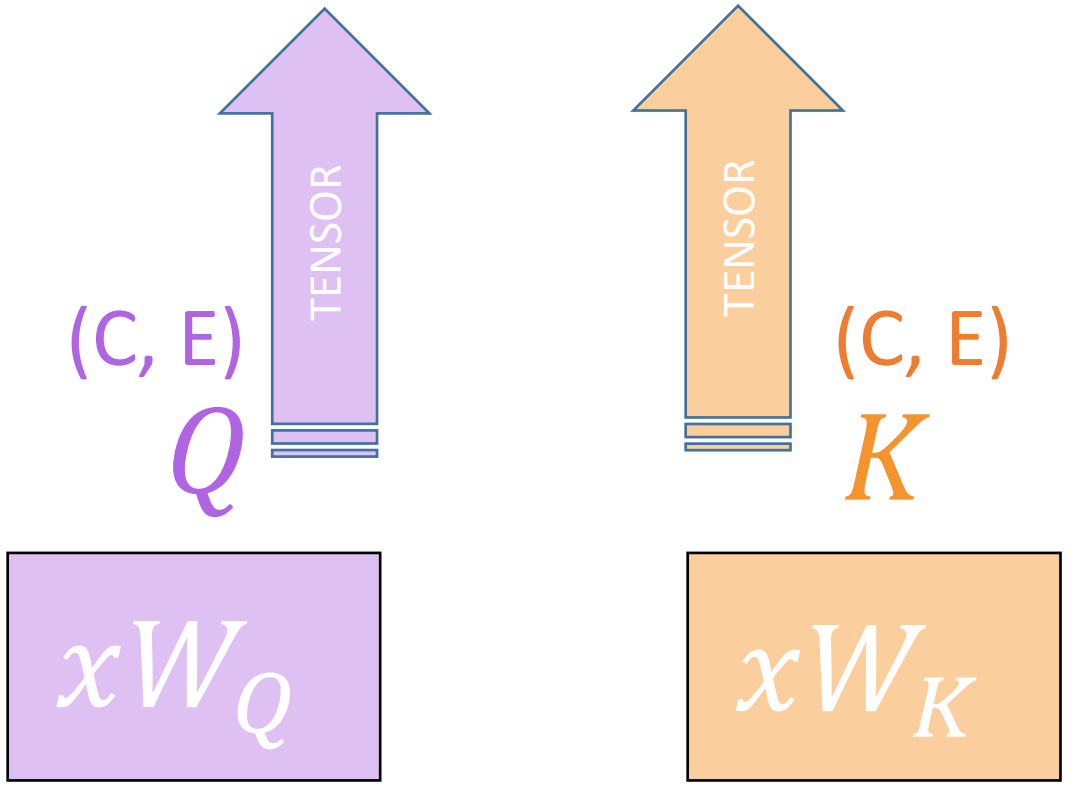


Project the original tensor x , that has size (C, E) into ANOTHER embedding of the same size.

The two weight matrices W_Q and W_K have the same size (E, E).

The idea is to **learn a different embedding** where it's easier to evaluate if a token is influenced by the other tokens in the same context window.

Attention head

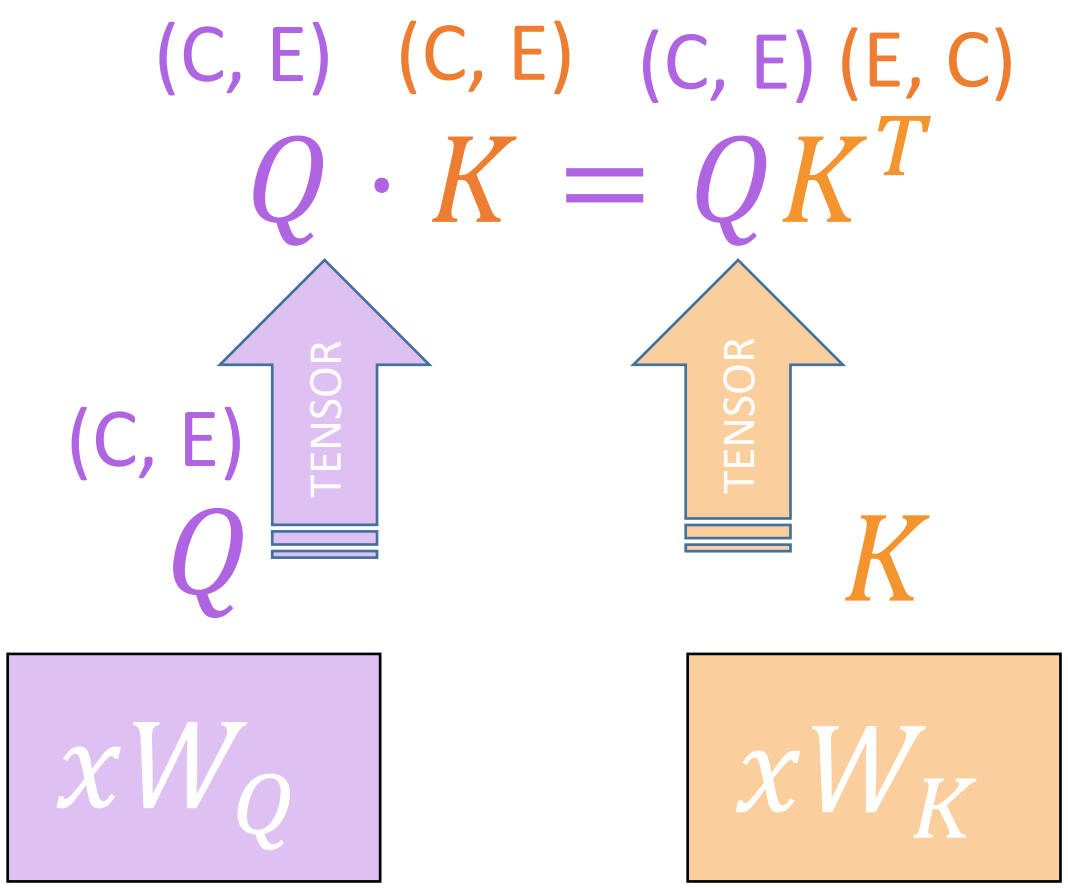


The two resulting tensors have size (C, E).

Why **two different projections** in same-sized space? Because a token could be **influenced by a lot of other tokens** (strong query), while **influence the others very little** (weak key). They could influence and be influenced differently.

If Q and K used the **same projection**, the meaning would be that a **token would influence and be influenced by the same other tokens**. But a name is strongly influenced by adjectives, and not vice-versa.

Attention head



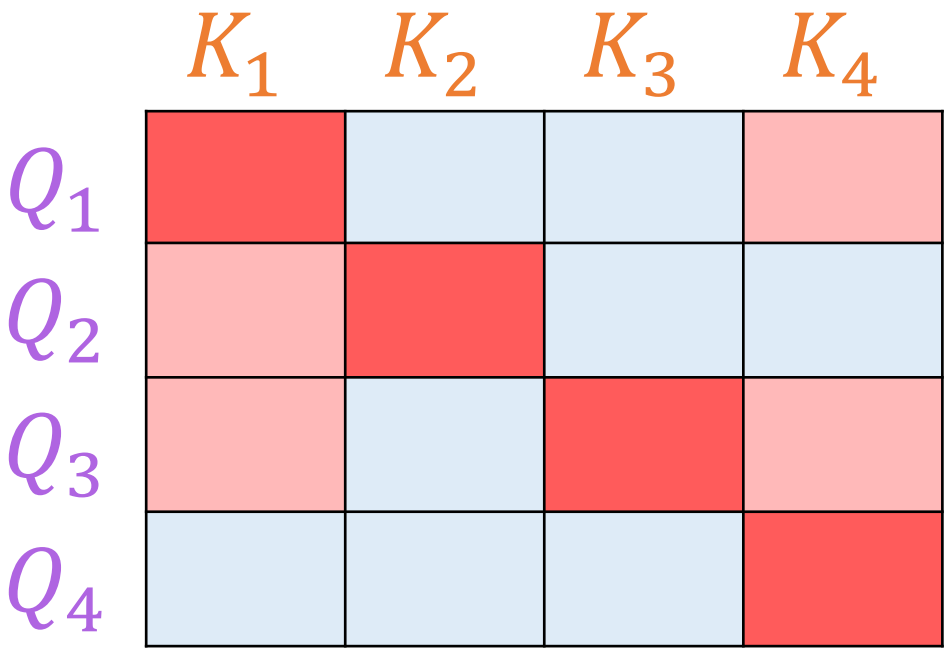
Dot product between the two tensors. What is the meaning?

Evaluation of similarity! High value, high similarity between vectors; low value, low similarity.

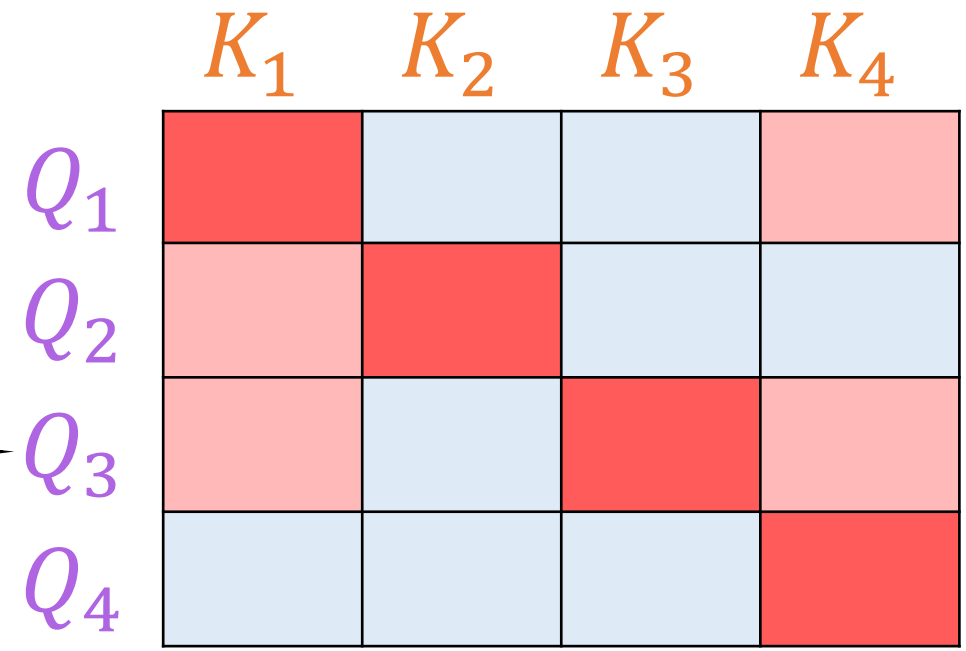
Here «similarity» means how likely it is that a token encoded in K will influence a token encoded in Q.

What will be the size of the tensor resulting from this operation?

Meaning of the $\boldsymbol{QK^T}$ matrix



Meaning of the QK^T matrix



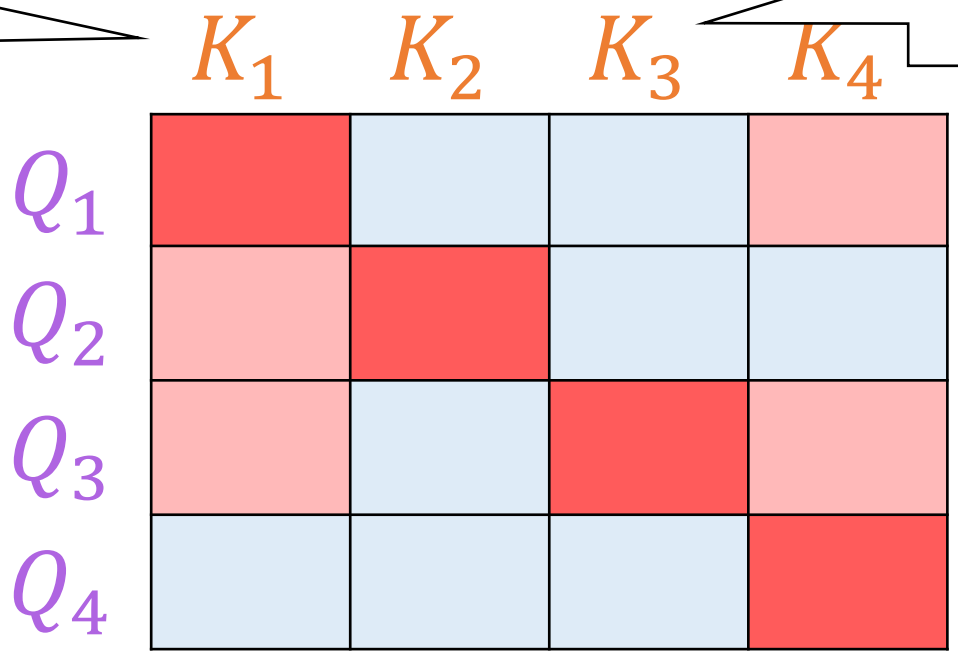
Token T_3 is strongly influenced by most other tokens

Token T_4 is not very influenced by other tokens in the context window

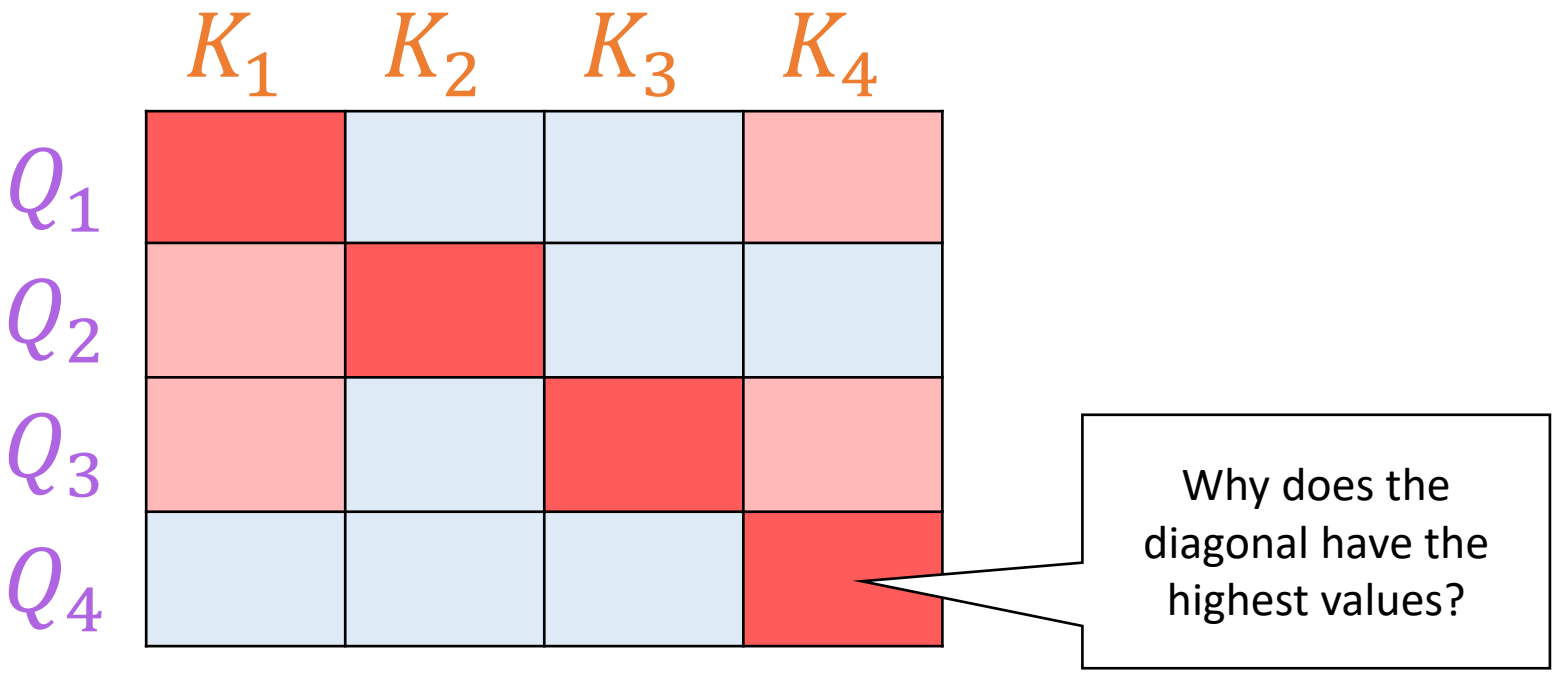
Meaning of the QK^T matrix

Tokens T_1 and T_4 have a strong influence on other tokens

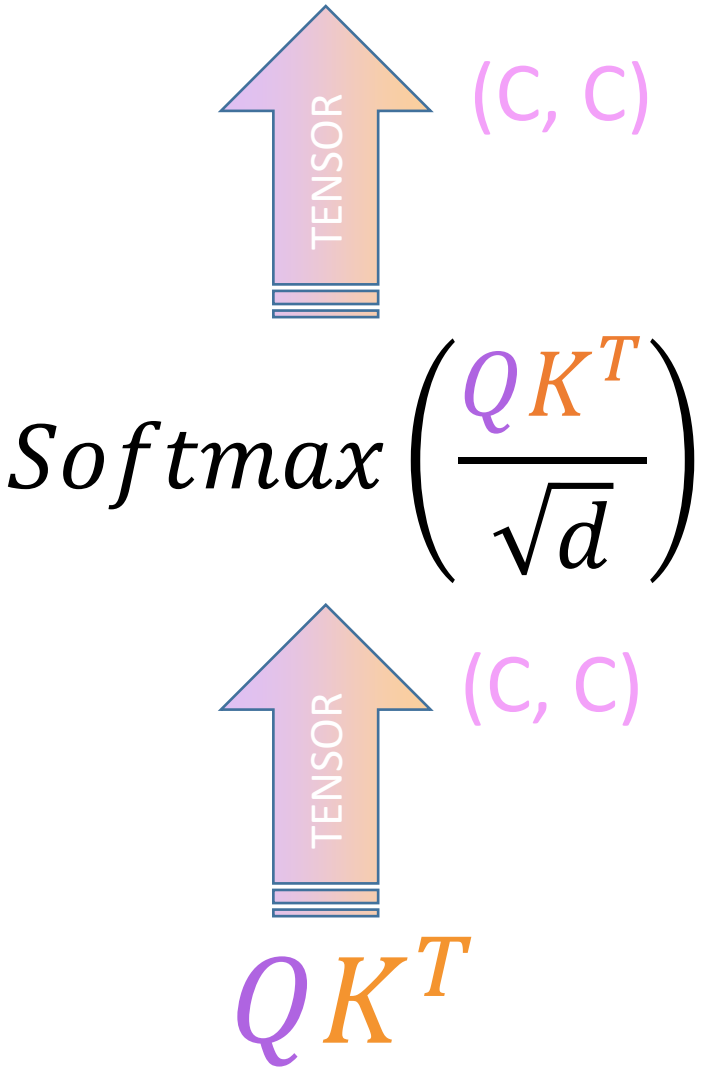
Token T_3 only influences itself



Meaning of the $\boldsymbol{QK^T}$ matrix



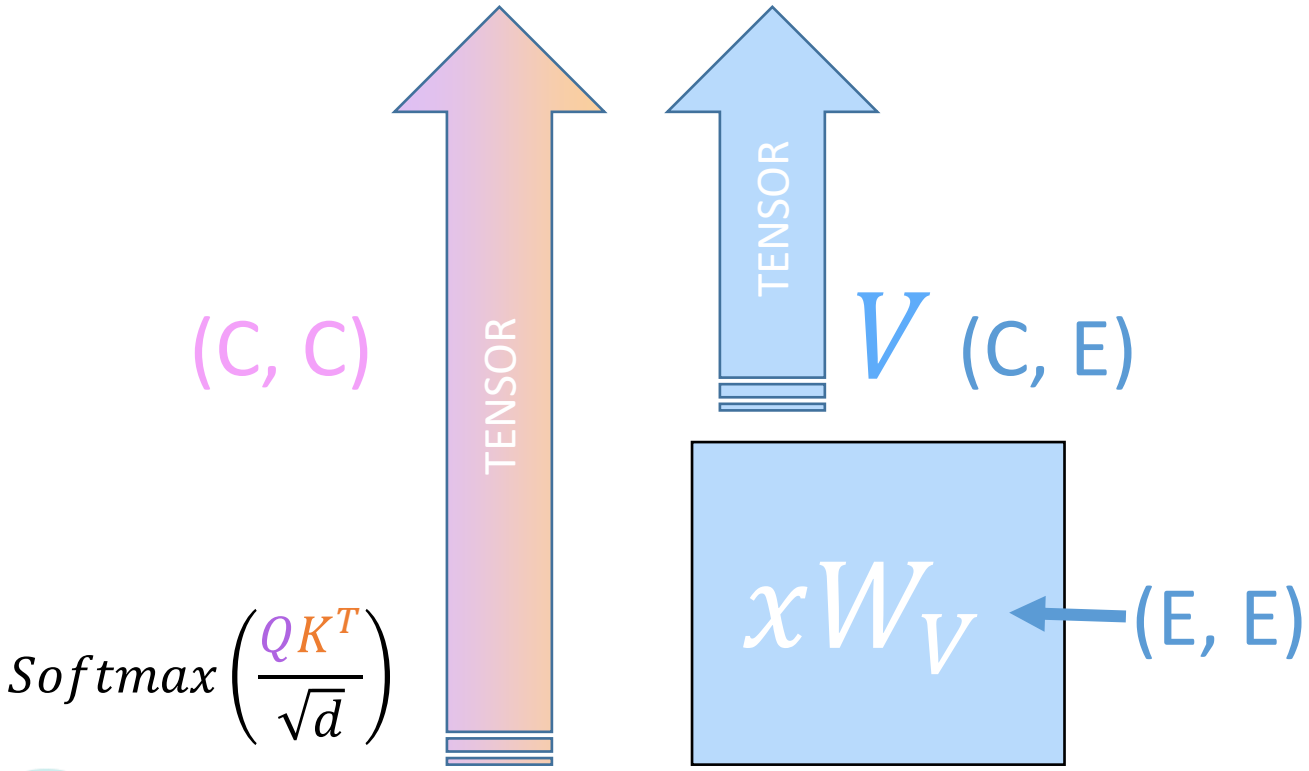
Attention head



The result has size (C, C). The meaning is how strongly each token in the context window has influence on each other token.

Now, the numbers here can be all over the place, so there is a normalization and a Softmax to have nice values in (0,1) that also add up to 1, for each column of the resulting matrix.

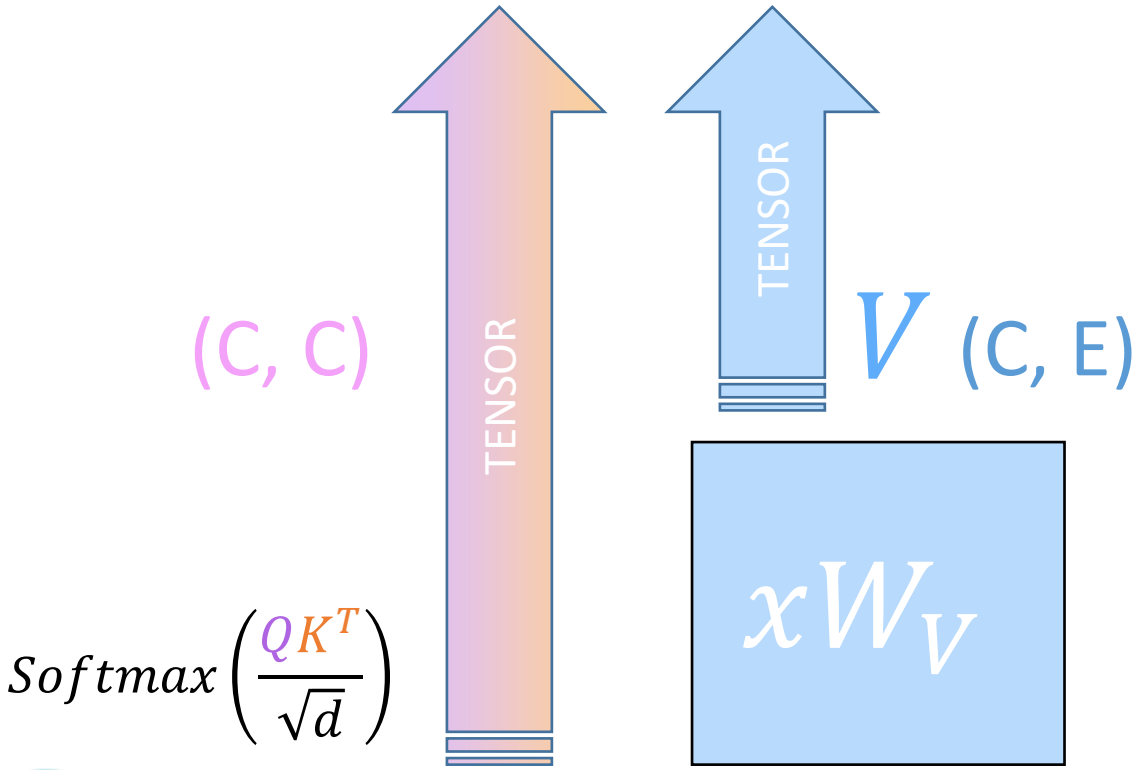
Attention head



We momentarily left the other part alone. The tensor $V = xW_V$ is computed, with weight matrix W_V of size (E, E). Since x had size (C, E), we find again that V has the same size as x.

Attention head

$$Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V$$

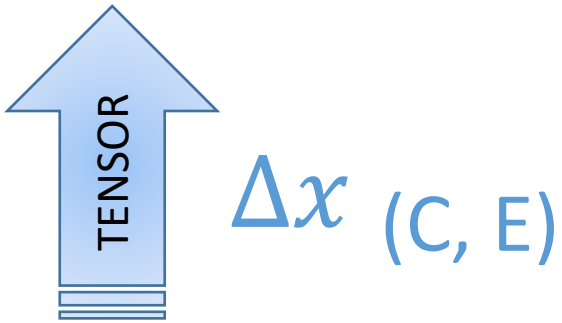


The last part is to multiply the results we obtained in the previous matrix multiplication by the tensor V.

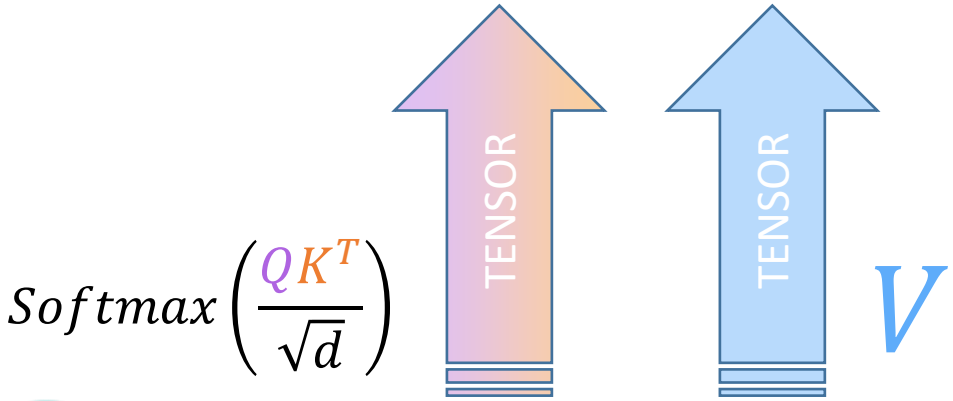
The meaning is to compute the final displacement for each token in the original tensor of embeddings, that will be later added to the original tensor.

What will be the final shape of the tensor resulting from this operation?

Attention head



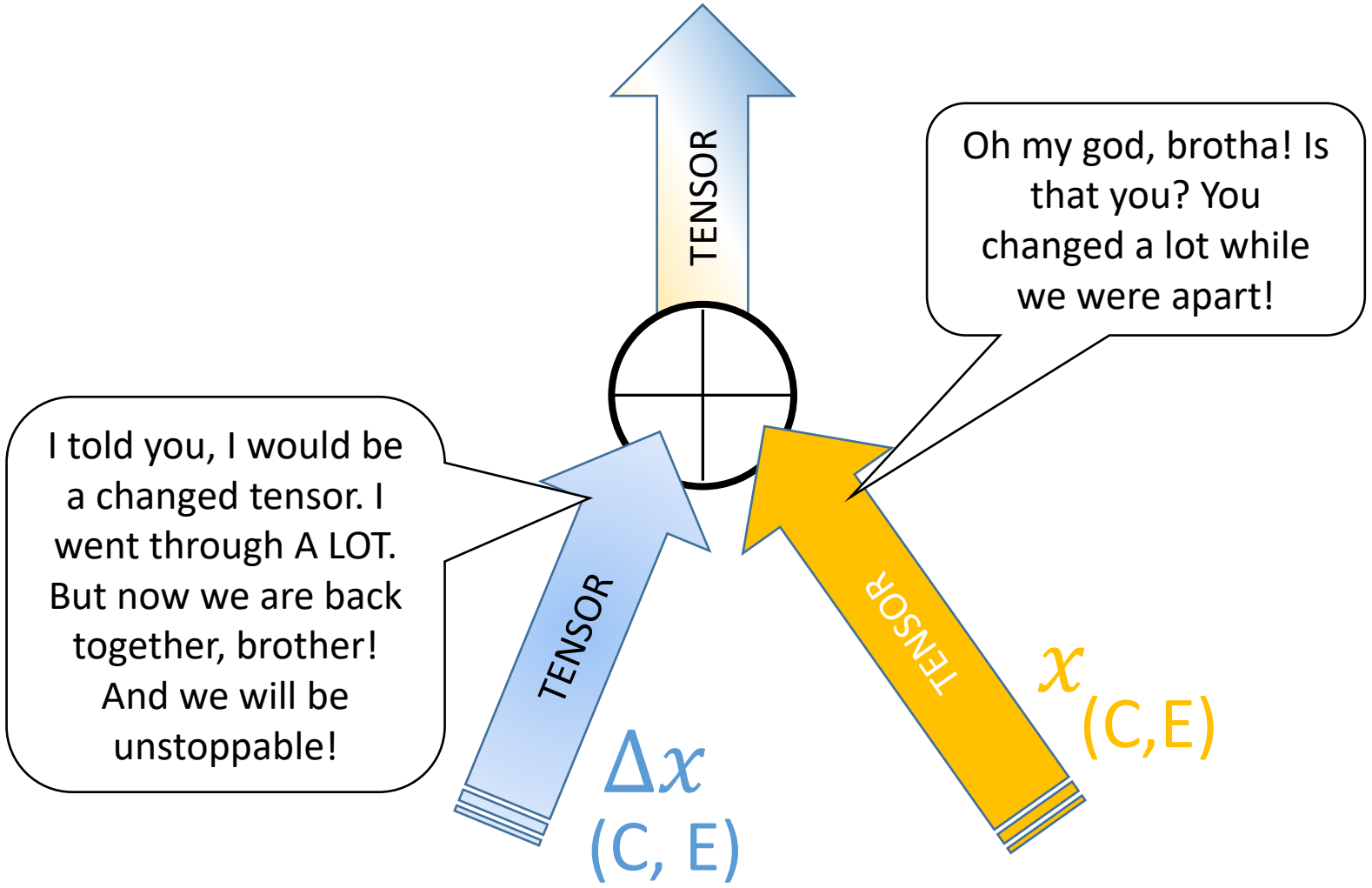
$$Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V$$



The computed displacement Δx has the **same shape** of the **original tensor** (C, E).

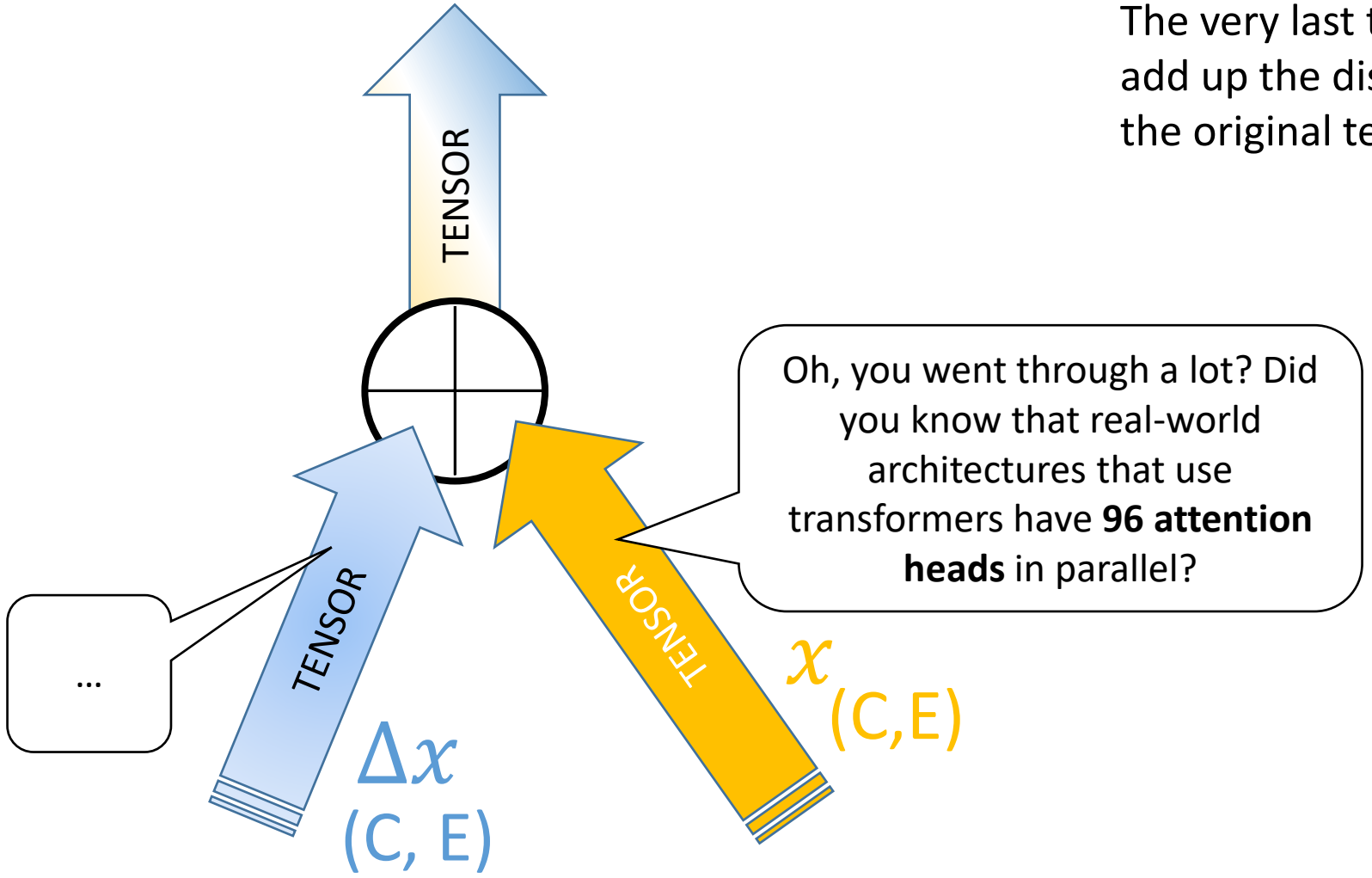
The work of the Attention head is finished!

Attention head



The very last thing left to do is to add up the displacement Δx to the original tensor x.

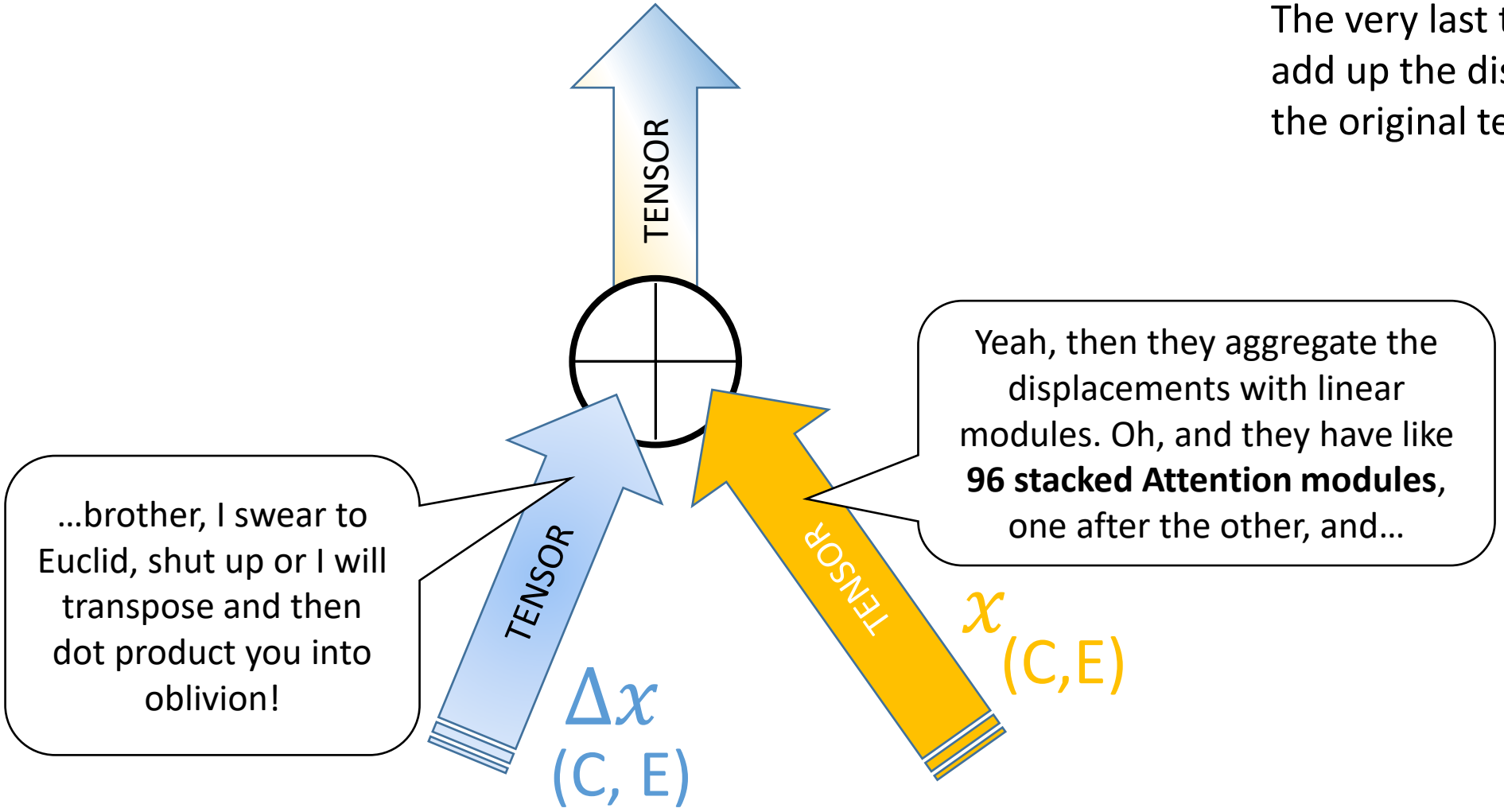
Attention head



The very last thing left to do is to add up the displacement Δx to the original tensor x.

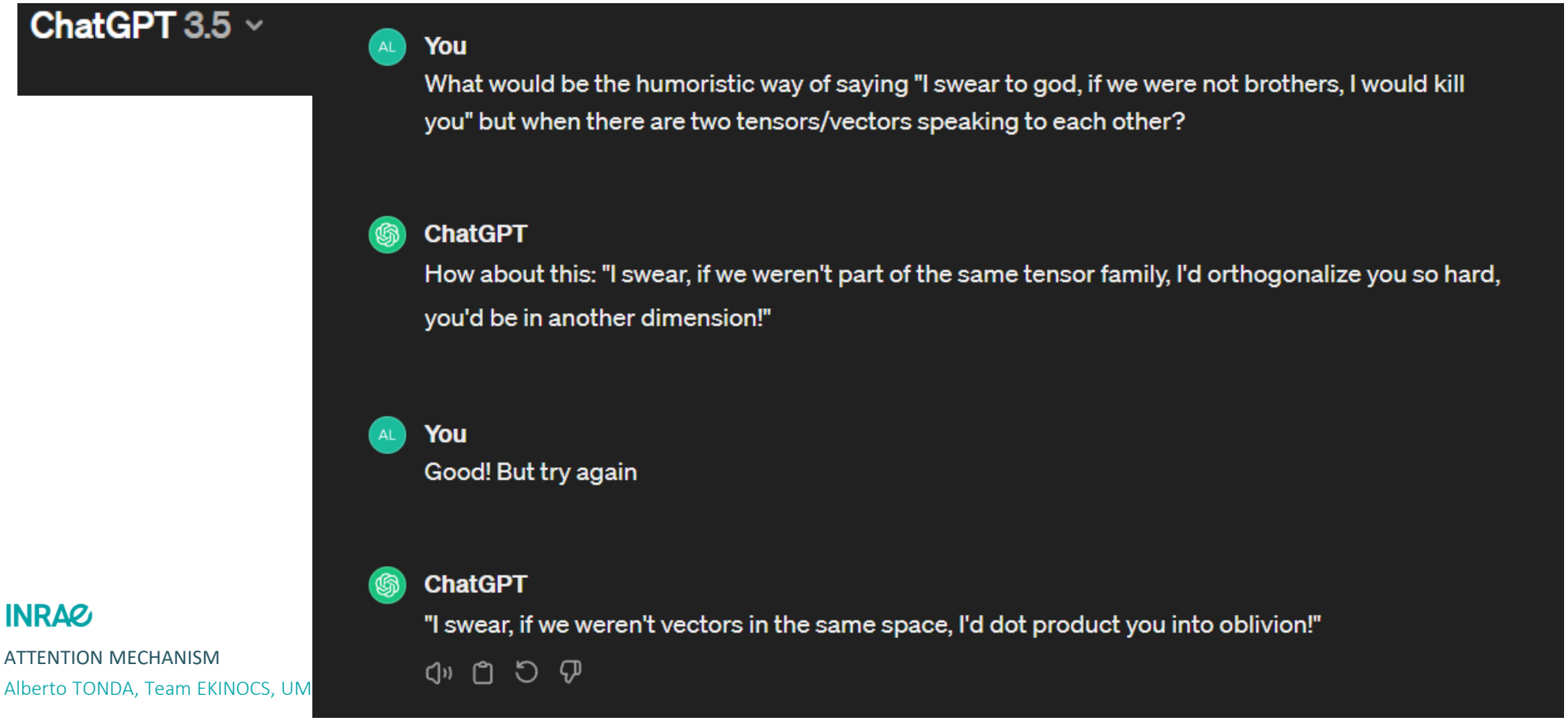
Attention head

The very last thing left to do is to add up the displacement Δx to the original tensor x.



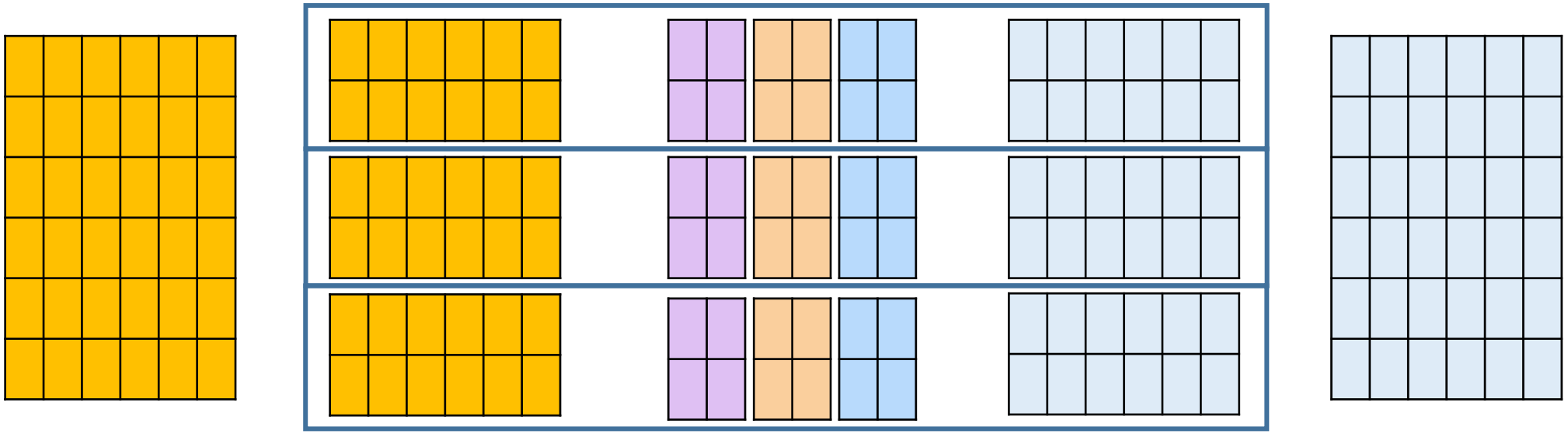
It works!

- This type of architecture makes it possible to embed complex semantic meaning in vectors



But the implementation…?

- The implementation is using **multi-head attention**



Attention to positions…?

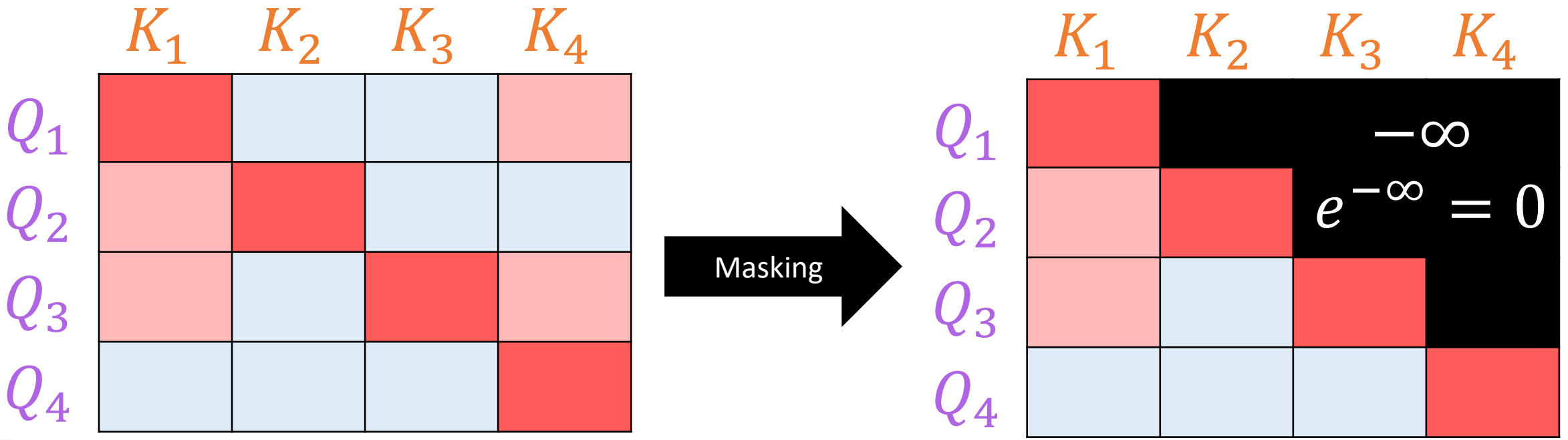
- How does an Attention head take into account **relative position** of tokens in the sequence?

Attention to position

- *It does not*. The result of Attention is **permutation invariant**
- However, for sequences, order is important!
- Several ways of solving the issue

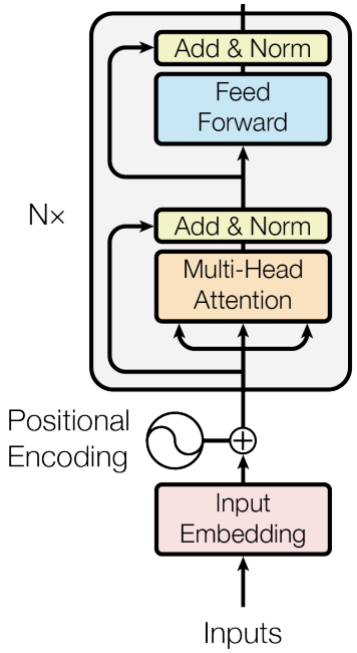
Masking

- Force a part of the QK^T matrix to zero (–inf for Softmax)
  - Prevent “future” tokens in sequence from influencing previous
  - Used in next-token prediction



Positional encoding

- Encode the relative position of the token in its *embedding*
  - Small displacement added to the embedding of each token
  - Displacement is the same per same relative position in window



Outputs of a Transformer architecture

Questions?

Bibliography

- Vaswani, A. et al. (2017). *Attention is all you need*. Advances in Neural Information Processing Systems.
- Starmer, J. (2023). *Attention for Neural Networks, Clearly Explained!!!*, https://www.youtube.com/watch?v=PSs6nxngL6k
- 3blue1brown (2024). *Transformers (how LLMs work) explained visually | DL5*, https://www.youtube.com/watch?v=wjZofJX0v4M
- 3blue1brown (2024). *Attention in transformers, visually explained | DL6*, https://www.youtube.com/watch?v=eMlx5fFNoYc