

# Large Language Model

Kaiyang Zhou

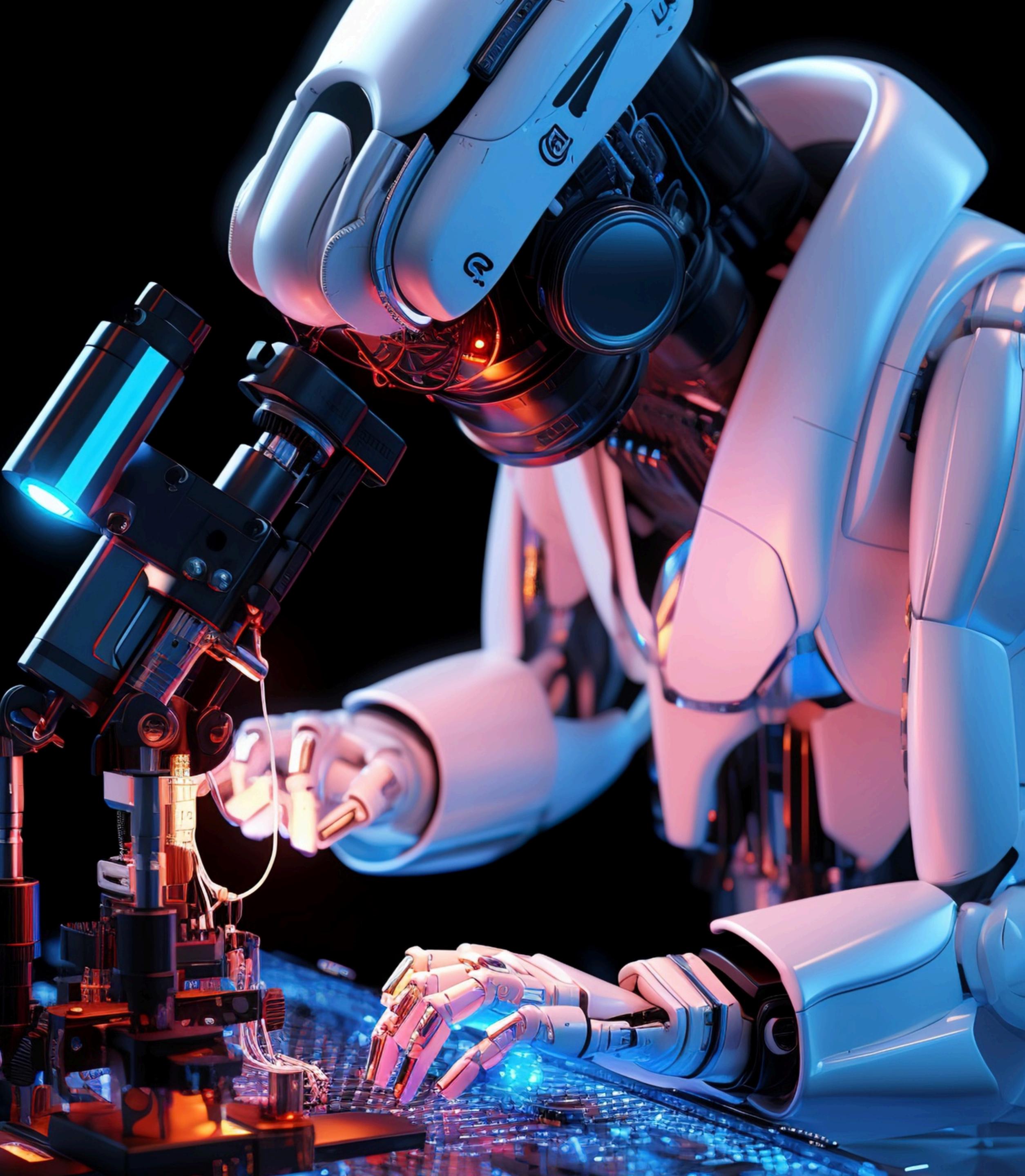
<https://kaiyangzhou.github.io/>



香港浸會大學  
HONG KONG BAPTIST UNIVERSITY

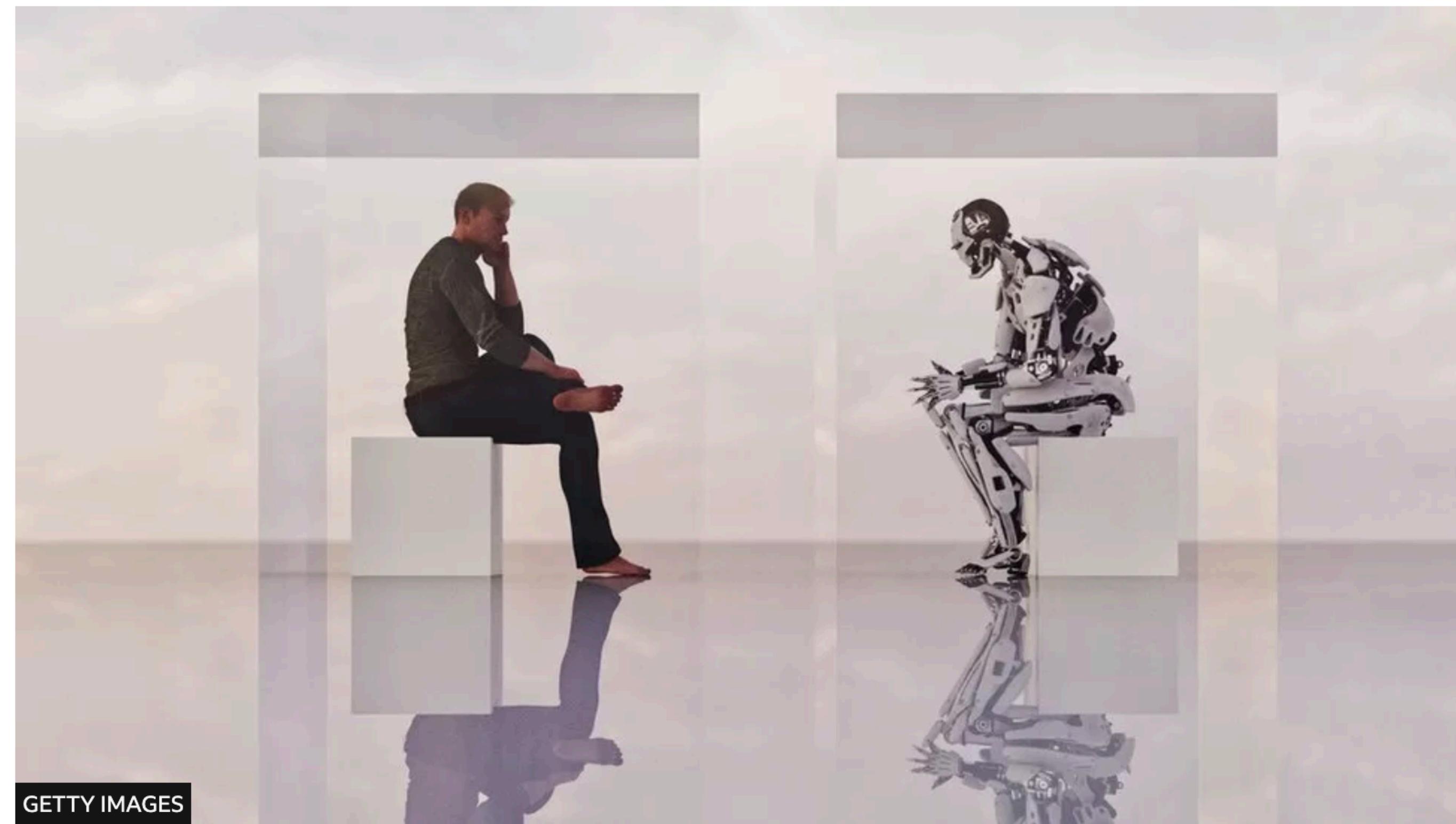


DEPARTMENT OF  
COMPUTER SCIENCE  
計算機科學系

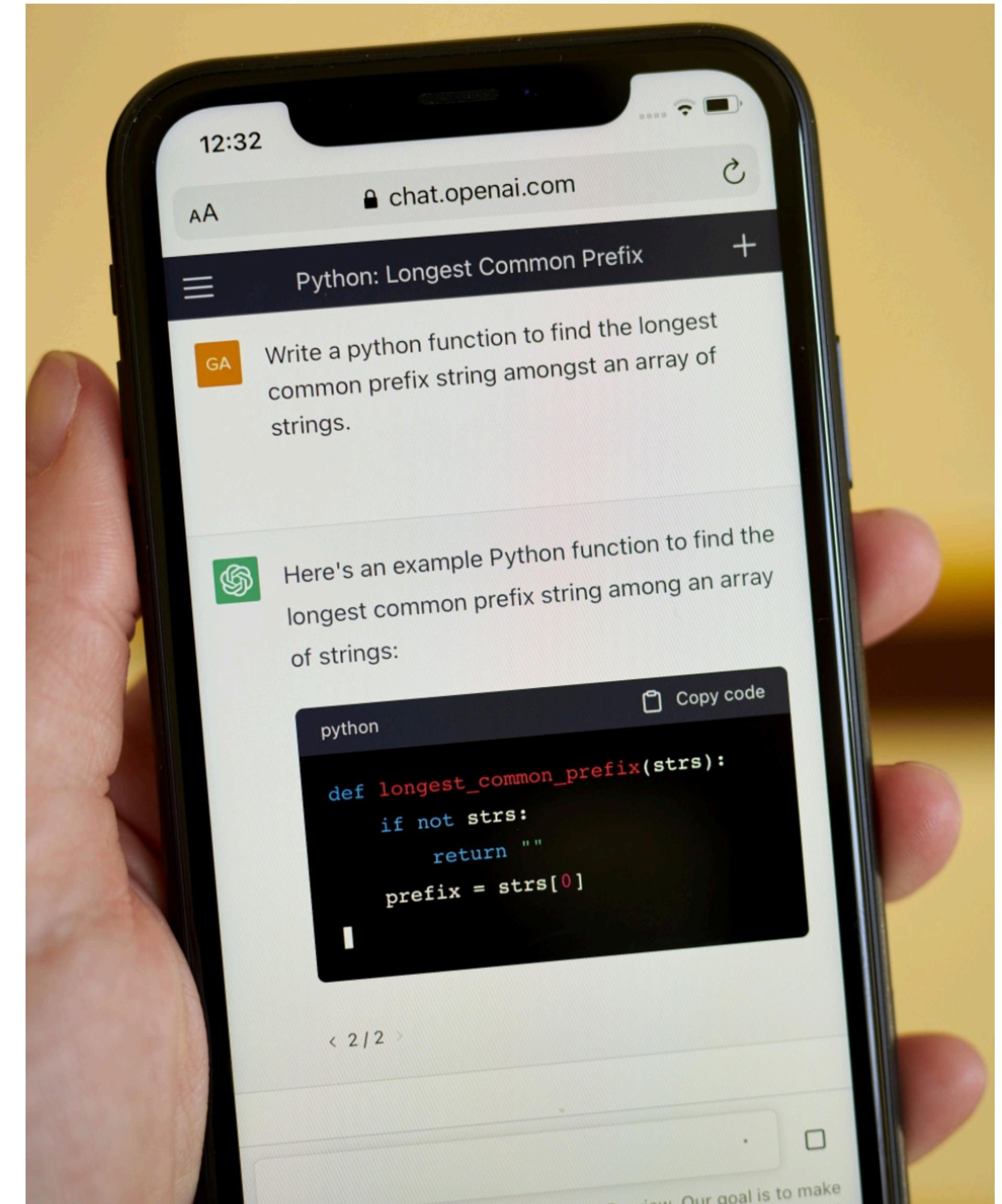


# ChatGPT: New AI chatbot has everyone talking to it

⌚ 7 December 2022



GETTY IMAGES





GITHUB COPILOT: CHAT

GitHub Copilot

Hi @monalisa, how can I help you?

I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and share feedback so that we can learn and improve.

monalisa

Write unit tests for this function

GitHub Copilot

```
import unittest
import datetime

class TestParseExpenses(unittest.TestCase):
    def test_
```

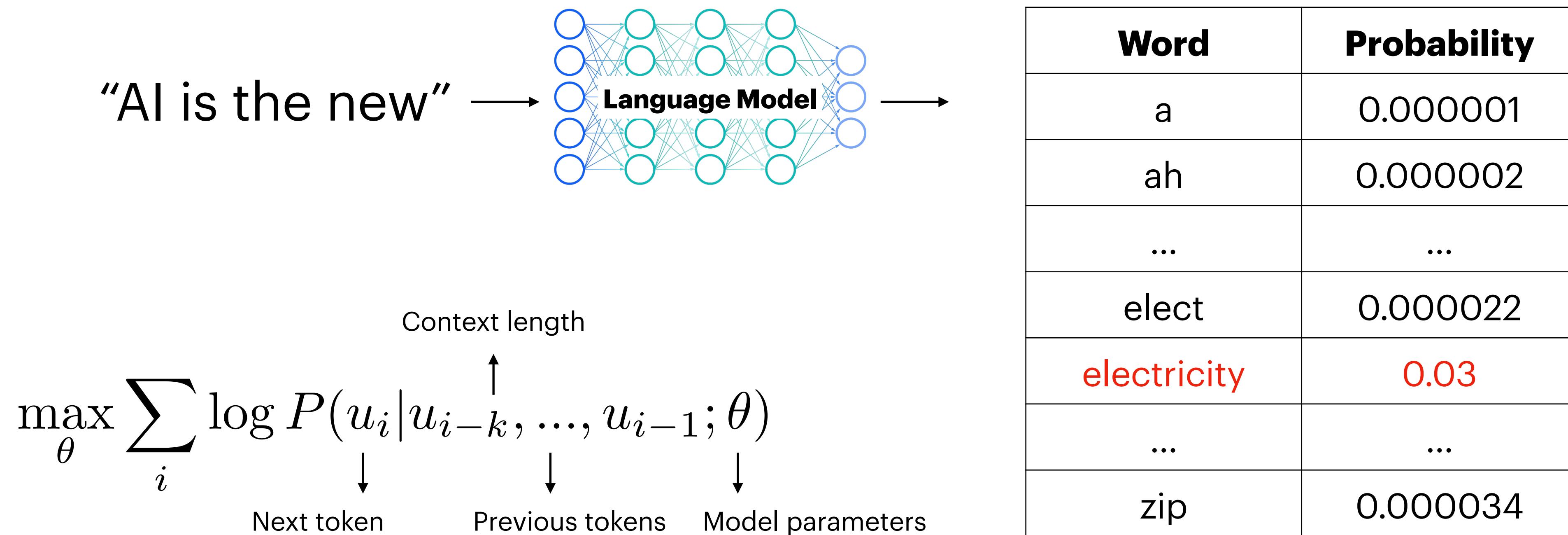
Ask a question or type '/' for commands ➤

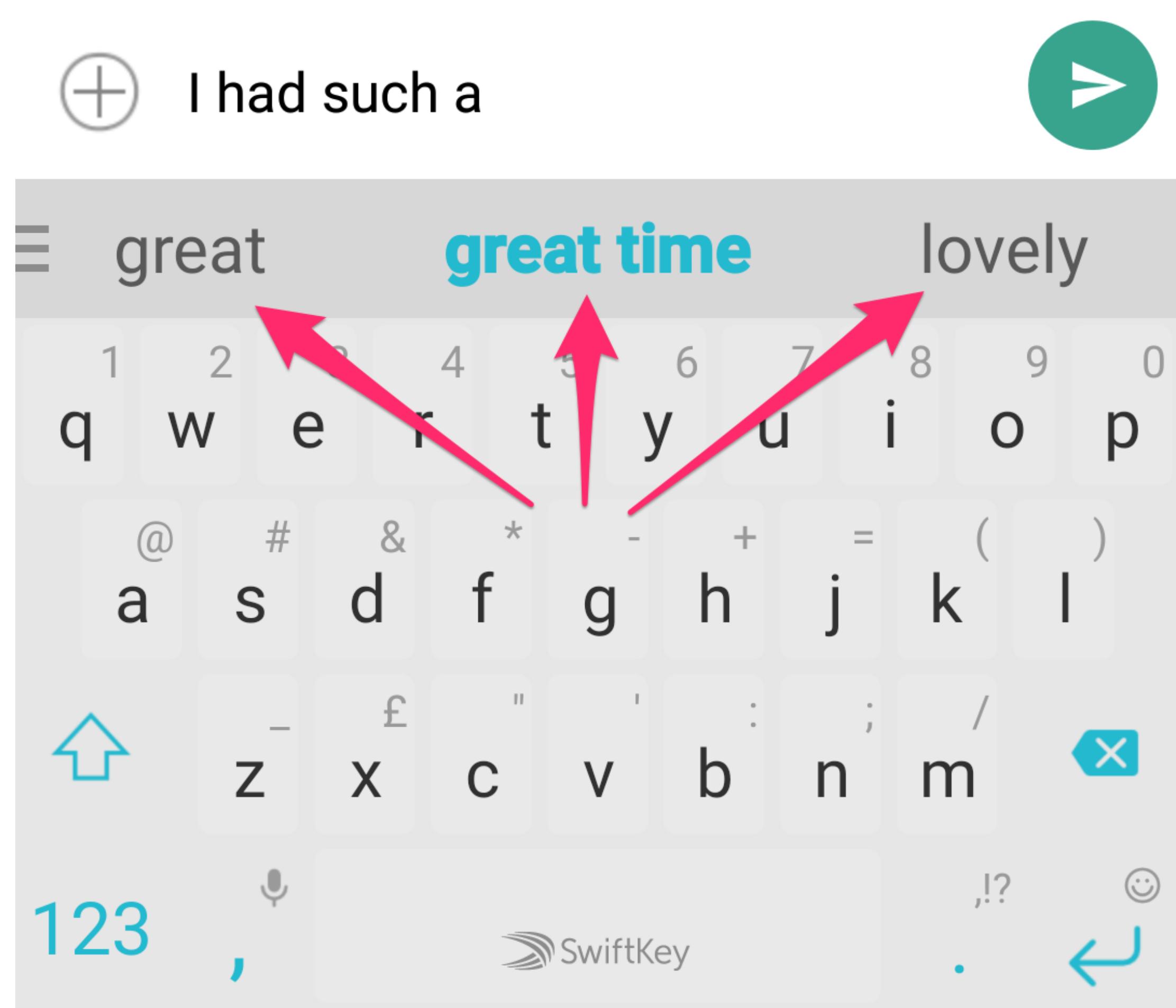
parse\_expenses.py X addresses.rb X sentiments.ts X

```
1 import datetime
2
3 def parse_expenses(expenses_string):
4     """Parse the list of expenses and return the list of triples (date, amount, currency)
5     Ignore lines starting with #.
6     Parse the date using datetime.
7     Example expenses_string:
8         2023-01-02 -34.01 USD
9         2023-01-03 2.59 DKK
10        2023-01-03 -2.72 EUR
11    """
12    expenses = []
13
14    for line in expenses_string.splitlines():
15        if line.startswith("#"):
16            continue
17        date, value, currency = line.split(" ")
18        expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
19                         float(value),
20                         currency)))
21    return expenses
22
23 expenses_data = '''2023-01-02 -34.01 USD
24                     2023-01-03 2.59 DKK
25                     2023-01-03 -2.72 EUR'''
```

# Pre-training a language model

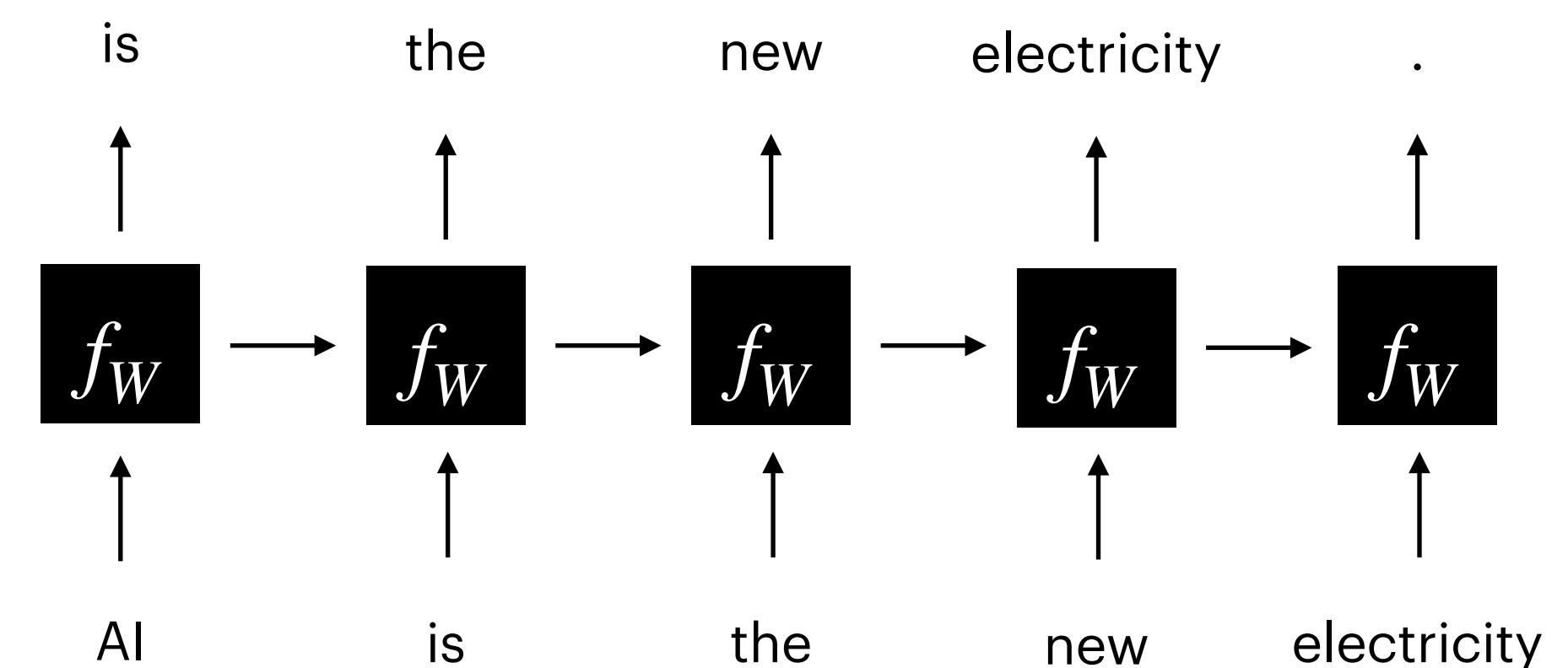
## Objective: next-token prediction





# Recurrent Neural Network

- (+) Can process any length.
- (+) Computation increases linearly with length.
- (+) Reuses weights.
- (-) Recurrent computation is slow.
- (-) May lose past information for long context.





# TRANSFORMER

---

## Attention Is All You Need

---

**Ashish Vaswani\***

Google Brain

[avaswani@google.com](mailto:avaswani@google.com)

**Noam Shazeer\***

Google Brain

[noam@google.com](mailto:noam@google.com)

**Niki Parmar\***

Google Research

[nikip@google.com](mailto:nikip@google.com)

**Jakob Uszkoreit\***

Google Research

[usz@google.com](mailto:usz@google.com)

**Llion Jones\***

Google Research

[llion@google.com](mailto:llion@google.com)

**Aidan N. Gomez\* †**

University of Toronto

[aidan@cs.toronto.edu](mailto:aidan@cs.toronto.edu)

**Lukasz Kaiser\***

Google Brain

[lukaszkaiser@google.com](mailto:lukaszkaiser@google.com)

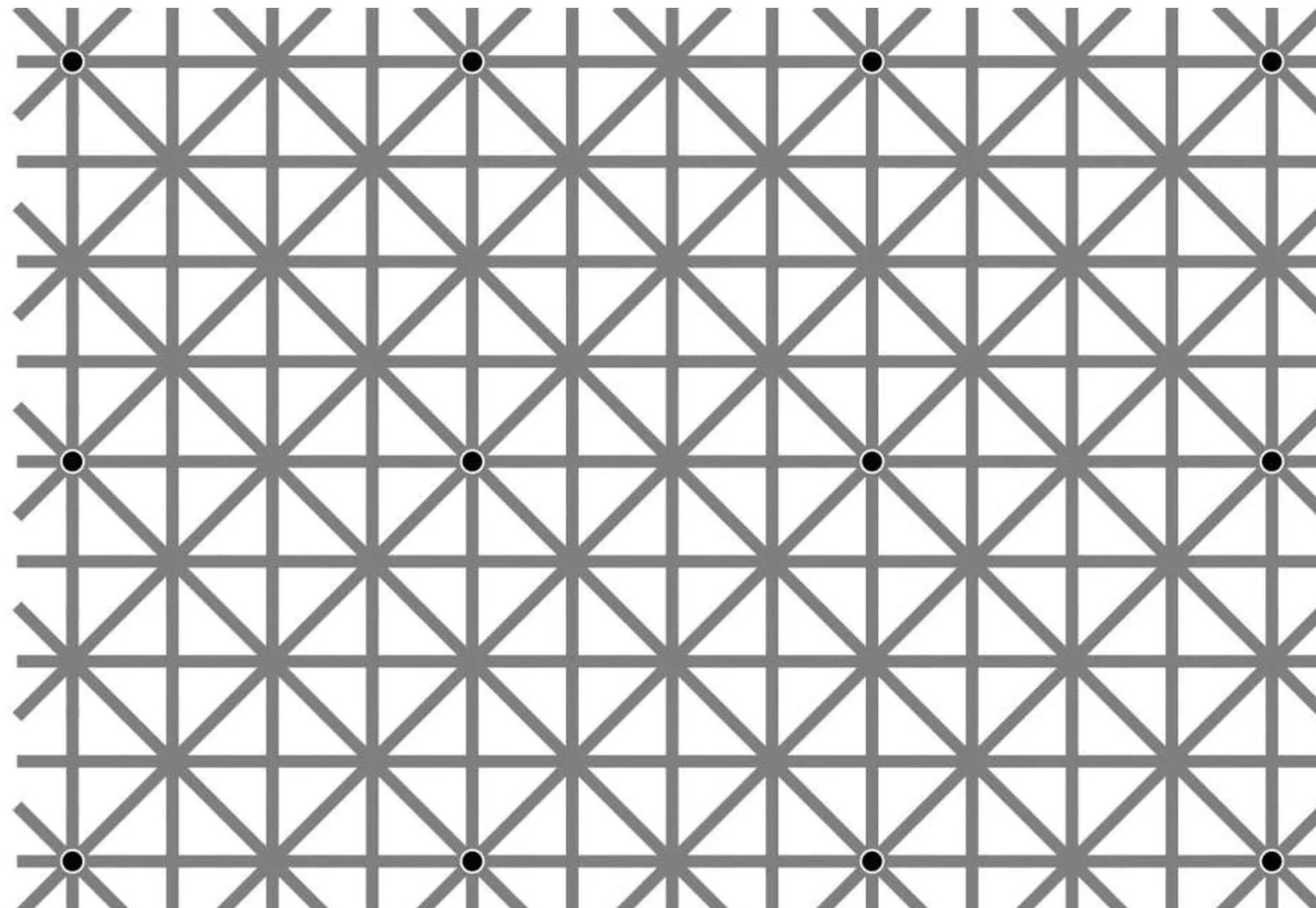
**Illia Polosukhin\* ‡**

[illia.polosukhin@gmail.com](mailto:illia.polosukhin@gmail.com)

# ATTENTION



# Can you see 12 black dots at the same time?



# Five factors that guide attention in visual search

[Jeremy M. Wolfe](#)  & [Todd S. Horowitz](#)

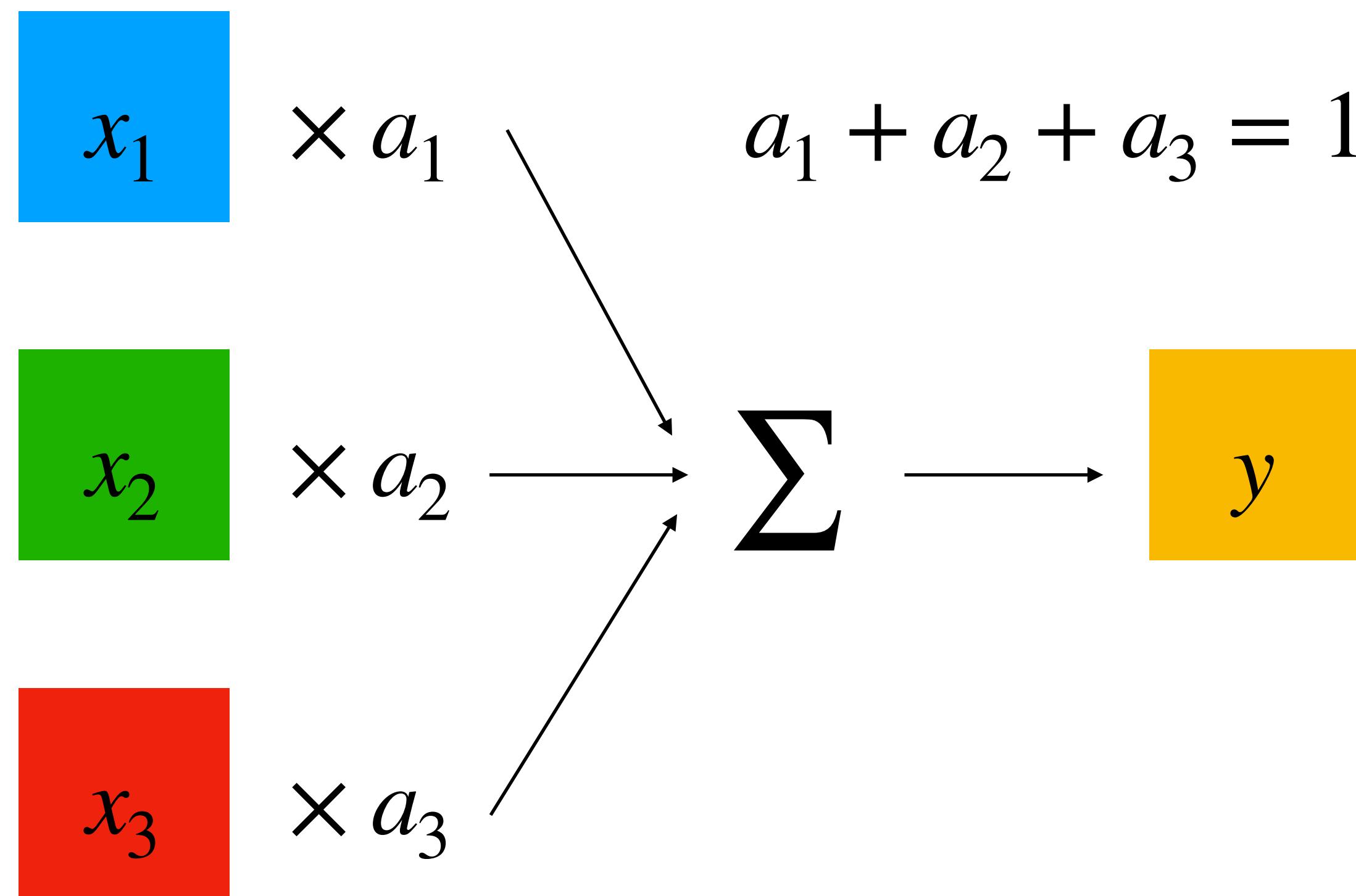
*Nature Human Behaviour* 1, Article number: 0058 (2017) | [Cite this article](#)

10k Accesses | 432 Citations | 49 Altmetric | [Metrics](#)

## Abstract

How do we find what we are looking for? Even when the desired target is in the current field of view, we need to search because fundamental limits on visual processing make it impossible to recognize everything at once. Searching involves directing attention to objects that might be the target. This deployment of attention is not random. It is guided to the most promising items and locations by five factors discussed here: bottom-up salience, top-down feature guidance, scene structure and meaning, the previous history of search over timescales ranging from milliseconds to years, and the relative value of the targets and distractors. Modern theories of visual search need to incorporate all five factors and specify how these factors combine to shape search behaviour. An understanding of the rules of guidance can be used to improve the accuracy and efficiency of socially important search tasks, from security screening to medical image perception.

# Attention is just weighted averaging



Vector/Matrix forms:

- $\text{Softmax}(\mathbf{a})\mathbf{X}, \mathbf{a} \in \mathbb{R}^{1 \times n}, \mathbf{X} \in \mathbb{R}^{n \times d}$
- $\text{Softmax}(\mathbf{A})\mathbf{X}, \mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{X} \in \mathbb{R}^{n \times d}$

# Seq2seq with attention

## Neural machine translation by jointly learning to align and translate

D Bahdanau, K Cho, Y Bengio - arXiv preprint arXiv:1409.0473, 2014 - arxiv.org

... In this section, we propose a novel architecture for **neural machine translation**. The new architecture consists of a bidirectional RNN as an encoder (Sec. 3.2) and a decoder that ...

☆ Save ⚡ Cite Cited by 33520 Related articles All 25 versions ➔

The context vector  $c_i$  depends on a sequence of *annotations*  $(h_1, \dots, h_{T_x})$  to which an encoder maps the input sentence. Each annotation  $h_i$  contains information about the whole input sequence with a strong focus on the parts surrounding the  $i$ -th word of the input sequence. We explain in detail how the annotations are computed in the next section.

The context vector  $c_i$  is, then, computed as a **weighted sum** of these annotations  $h_i$ :

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j. \quad (5)$$

The weight  $\alpha_{ij}$  of each annotation  $h_j$  is computed by

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

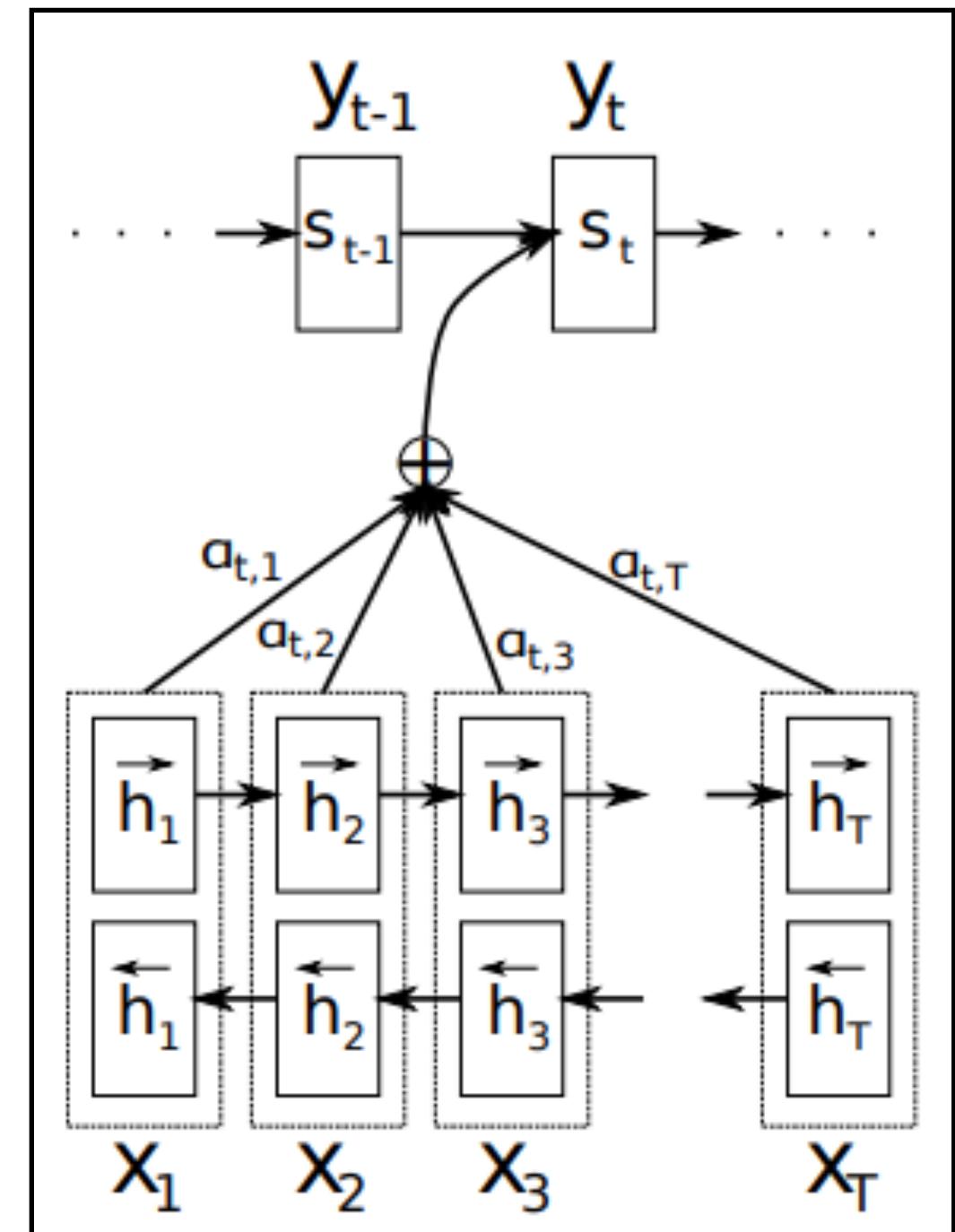
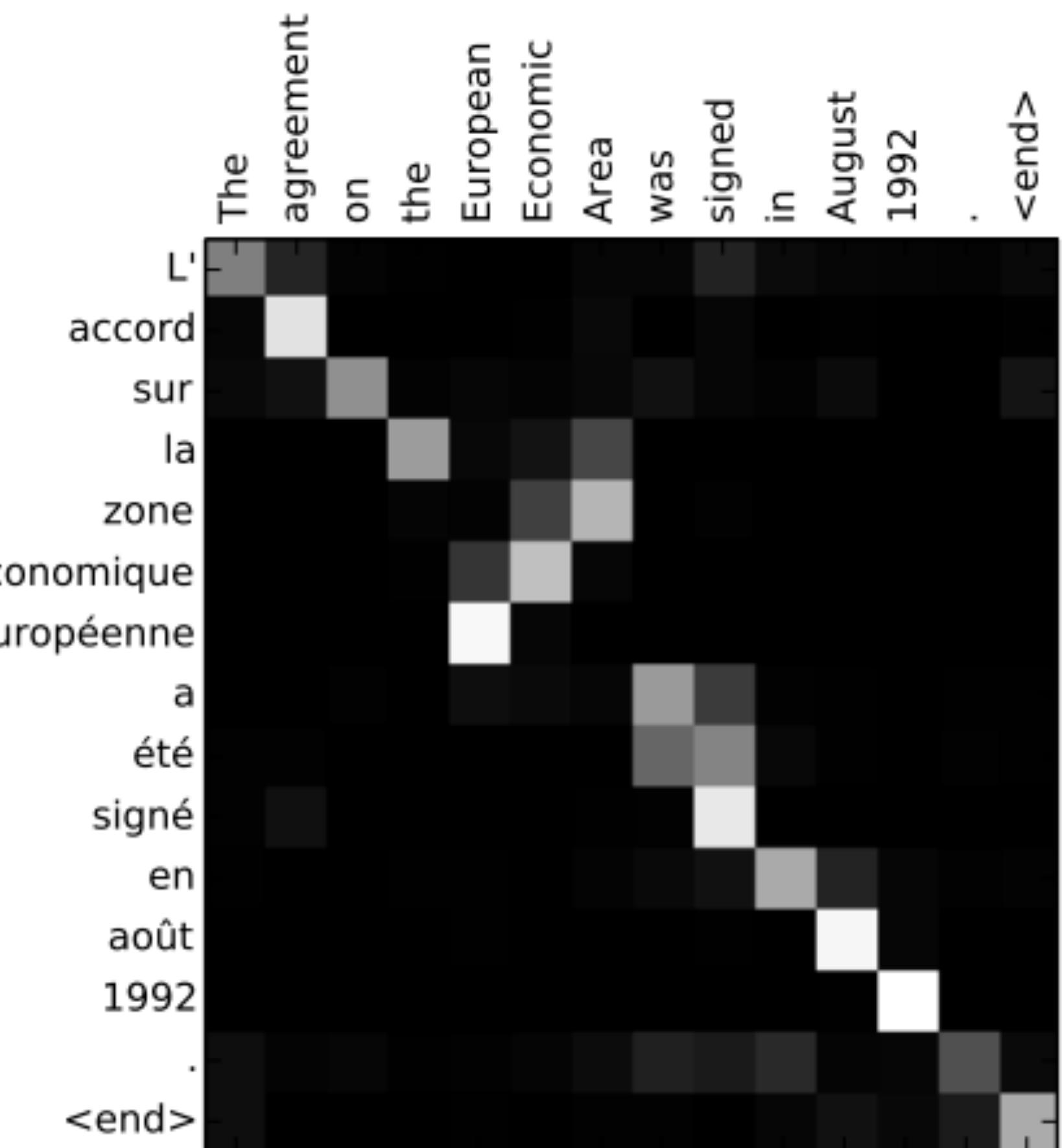
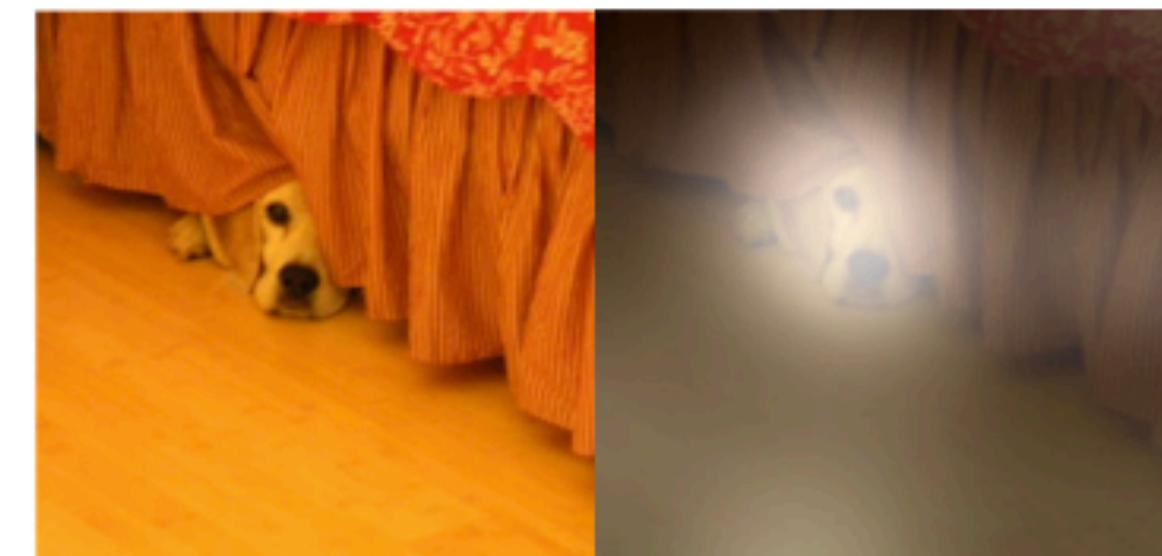
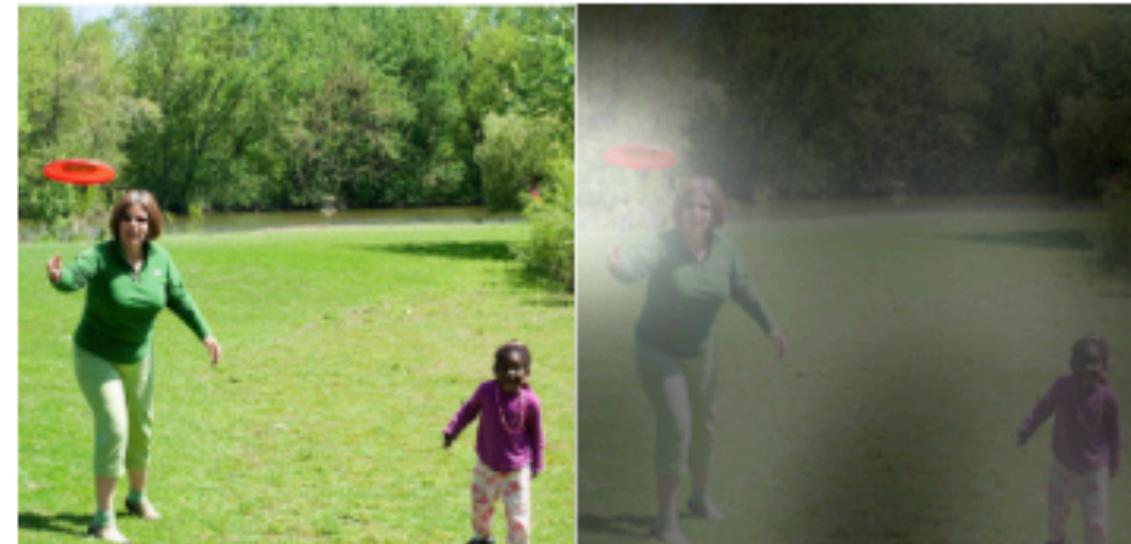
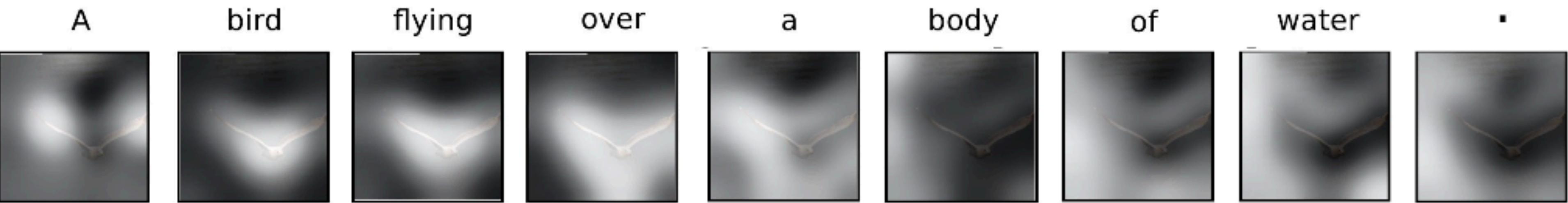


Figure 1: The graphical illustration of the proposed model trying to generate the  $t$ -th target word  $y_t$  given a source sentence  $(x_1, x_2, \dots, x_T)$ .

## English → French Translation



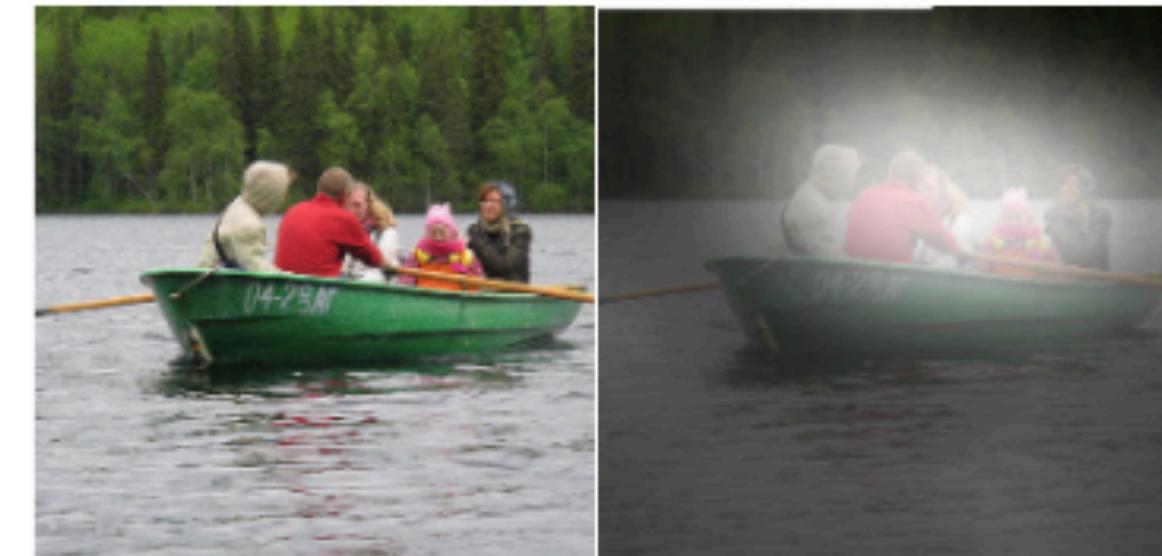
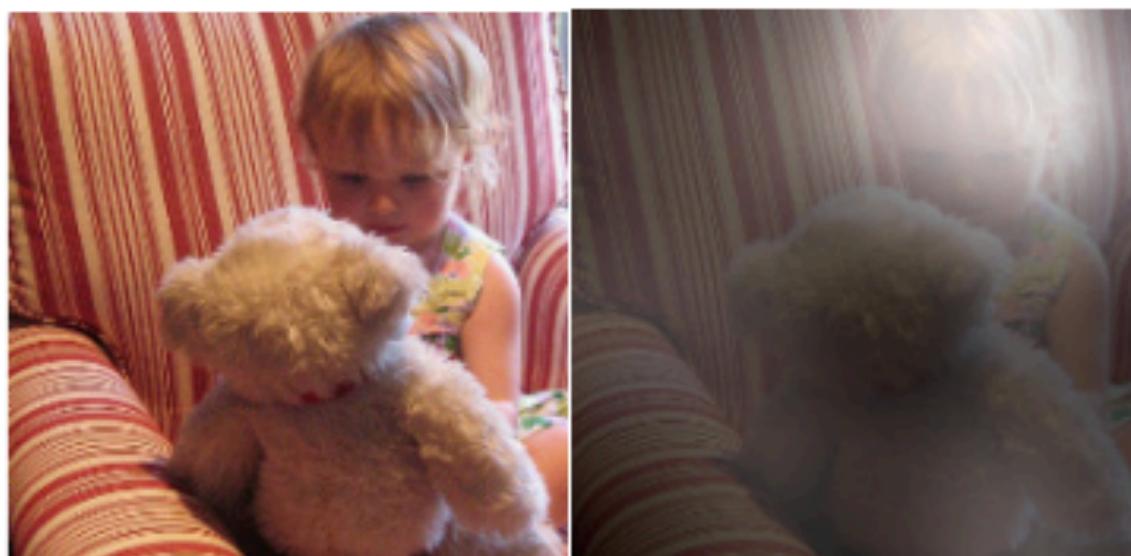
# Attention in image captioning



A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.



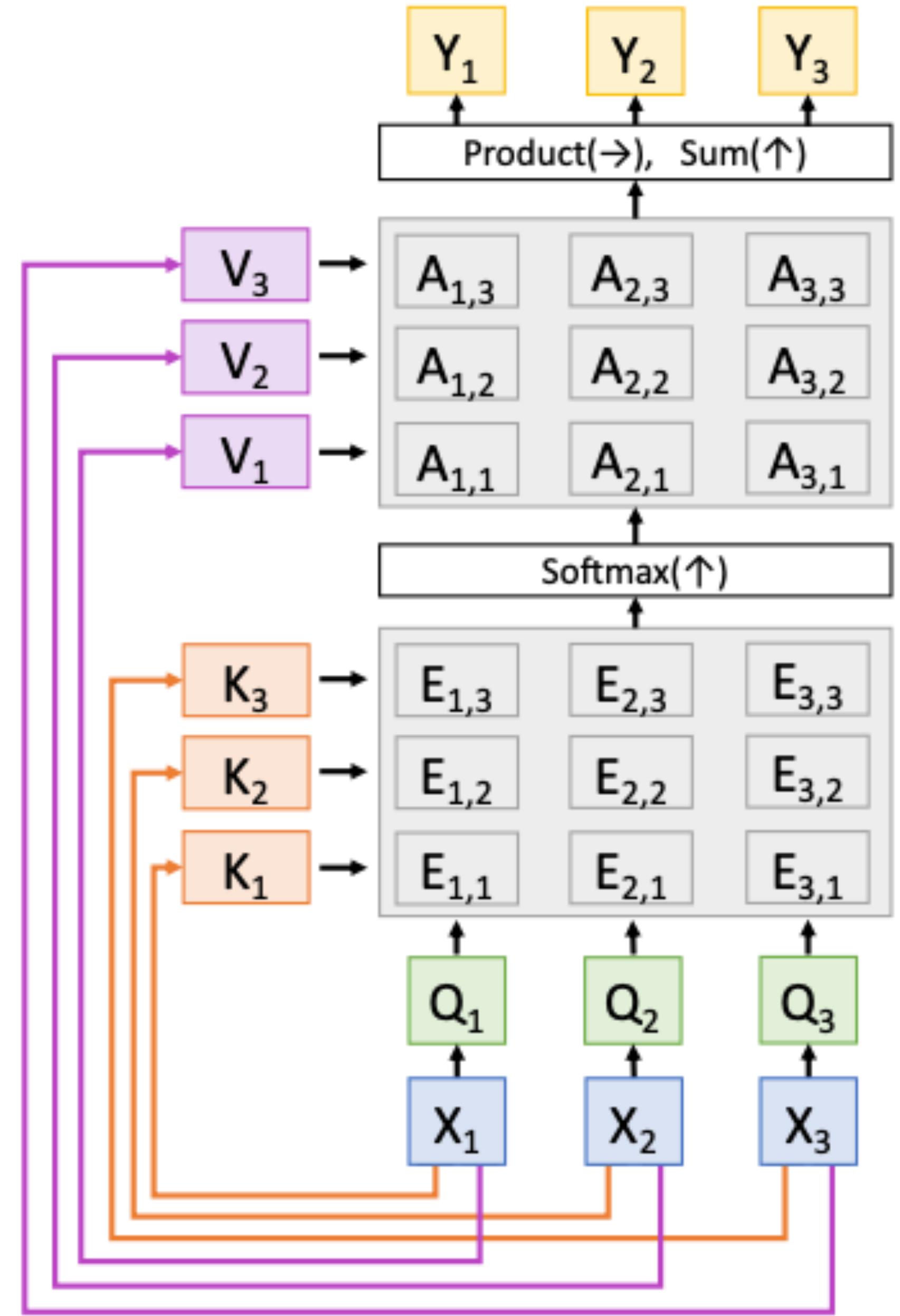
A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

# Self-attention

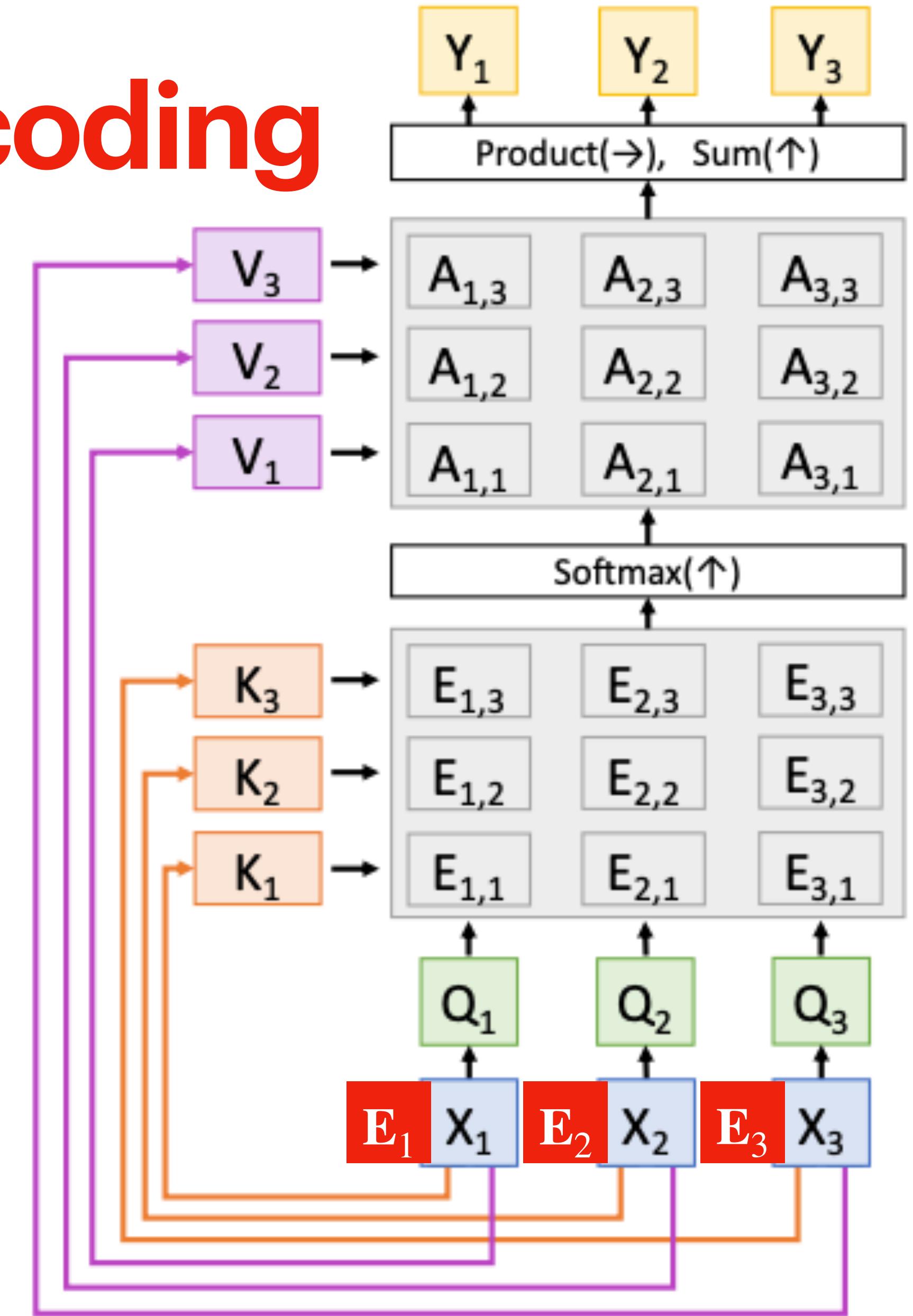
- All of the keys, queries, and values come from the same output of the previous layer,  $\mathbf{X} \in \mathbb{R}^{n \times d_X}$
- Key matrix:  $\mathbf{W}_K \in \mathbb{R}^{d_X \times d_Q}$
- Query matrix:  $\mathbf{W}_Q \in \mathbb{R}^{d_X \times d_Q}$
- Value matrix:  $\mathbf{W}_V \in \mathbb{R}^{d_X \times d_V}$
- Keys:  $\mathbf{K} = \mathbf{X}\mathbf{W}_K$ , Queries:  $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$ , Values:  $\mathbf{V} = \mathbf{X}\mathbf{W}_V$
- Similarities:  $\mathbf{S} = \mathbf{Q}\mathbf{K}^T / \sqrt{d_Q}$  (shape:  $n \times n$ )
- Attention:  $\mathbf{A} = \text{Softmax}(\mathbf{S})$  (applied to each row)
- Output:  $\mathbf{Y} = \mathbf{AV}$  (shape:  $n \times d_V$ )



Source: Justin Johnson's EECS498/598 course

# Self-attention + positional encoding

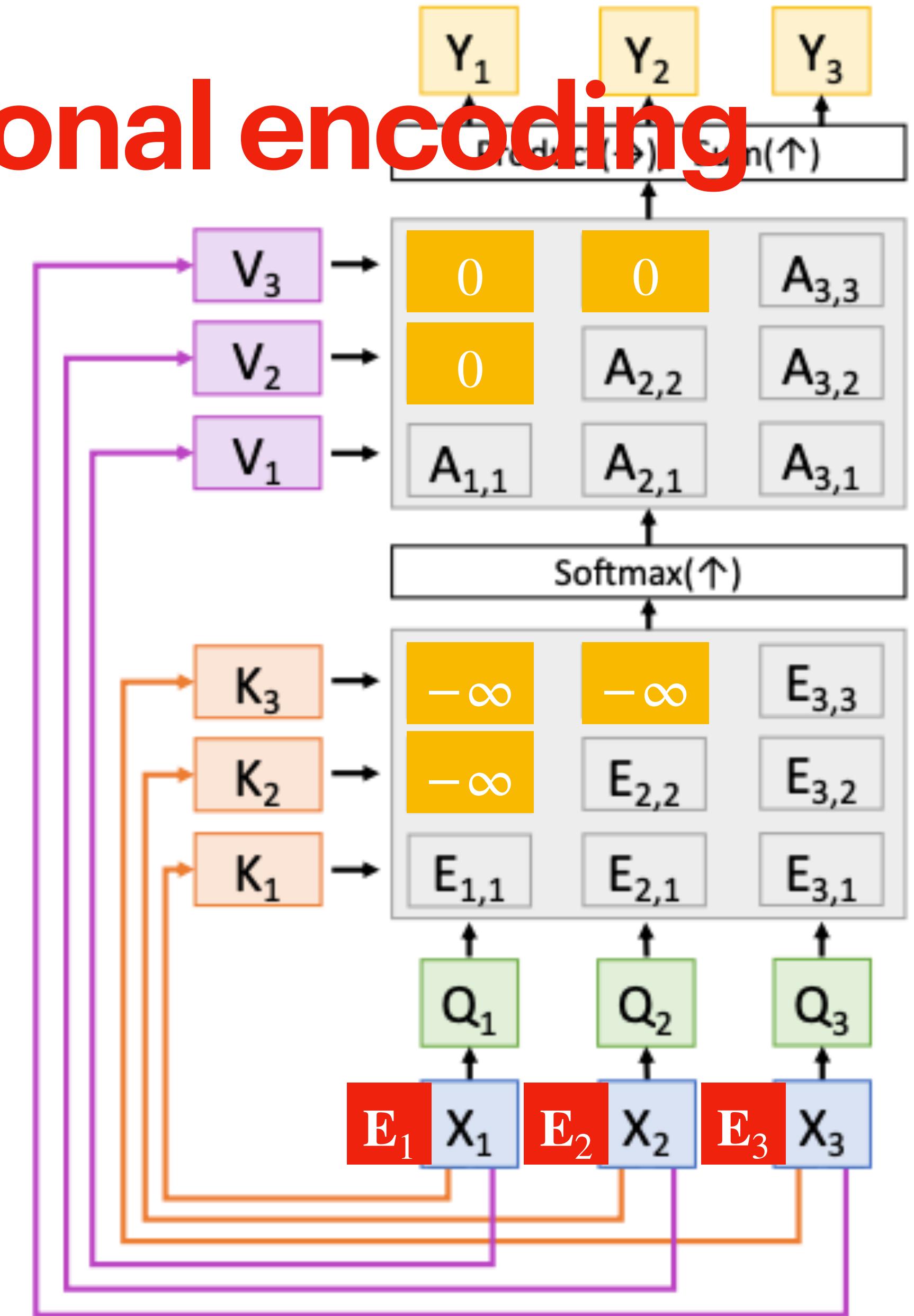
- All of the keys, queries, and values come from the same output of the previous layer,  $\mathbf{X} \in \mathbb{R}^{n \times d_X} + \mathbf{E}$  (fixed/learned)
- Key matrix:  $\mathbf{W}_K \in \mathbb{R}^{d_X \times d_Q}$
- Query matrix:  $\mathbf{W}_Q \in \mathbb{R}^{d_X \times d_Q}$
- Value matrix:  $\mathbf{W}_V \in \mathbb{R}^{d_X \times d_V}$
- Keys:  $\mathbf{K} = \mathbf{X}\mathbf{W}_K$ , Queries:  $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$ , Values:  $\mathbf{V} = \mathbf{X}\mathbf{W}_V$
- Similarities:  $\mathbf{S} = \mathbf{Q}\mathbf{K}^T / \sqrt{d_Q}$  (shape:  $n \times n$ )
- Attention:  $\mathbf{A} = \text{Softmax}(\mathbf{S})$  (applied to each row)
- Output:  $\mathbf{Y} = \mathbf{AV}$  (shape:  $n \times d_V$ )



Source: Justin Johnson's EECS498/598 course

# Masked self-attention + positional encoding

- All of the keys, queries, and values come from the same output of the previous layer,  $\mathbf{X} \in \mathbb{R}^{n \times d_X} + \mathbf{E}$  (fixed/learned)
- Key matrix:  $\mathbf{W}_K \in \mathbb{R}^{d_X \times d_Q}$
- Query matrix:  $\mathbf{W}_Q \in \mathbb{R}^{d_X \times d_Q}$
- Value matrix:  $\mathbf{W}_V \in \mathbb{R}^{d_X \times d_V}$
- Keys:  $\mathbf{K} = \mathbf{X}\mathbf{W}_K$ , Queries:  $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$ , Values:  $\mathbf{V} = \mathbf{X}\mathbf{W}_V$
- Similarities:  $\mathbf{S} = \mathbf{Q}\mathbf{K}^T / \sqrt{d_Q}$  (shape:  $n \times n$ )
- Attention:  $\mathbf{A} = \text{Softmax}(\mathbf{S})$  (applied to each row)
- Output:  $\mathbf{Y} = \mathbf{AV}$  (shape:  $n \times d_V$ )

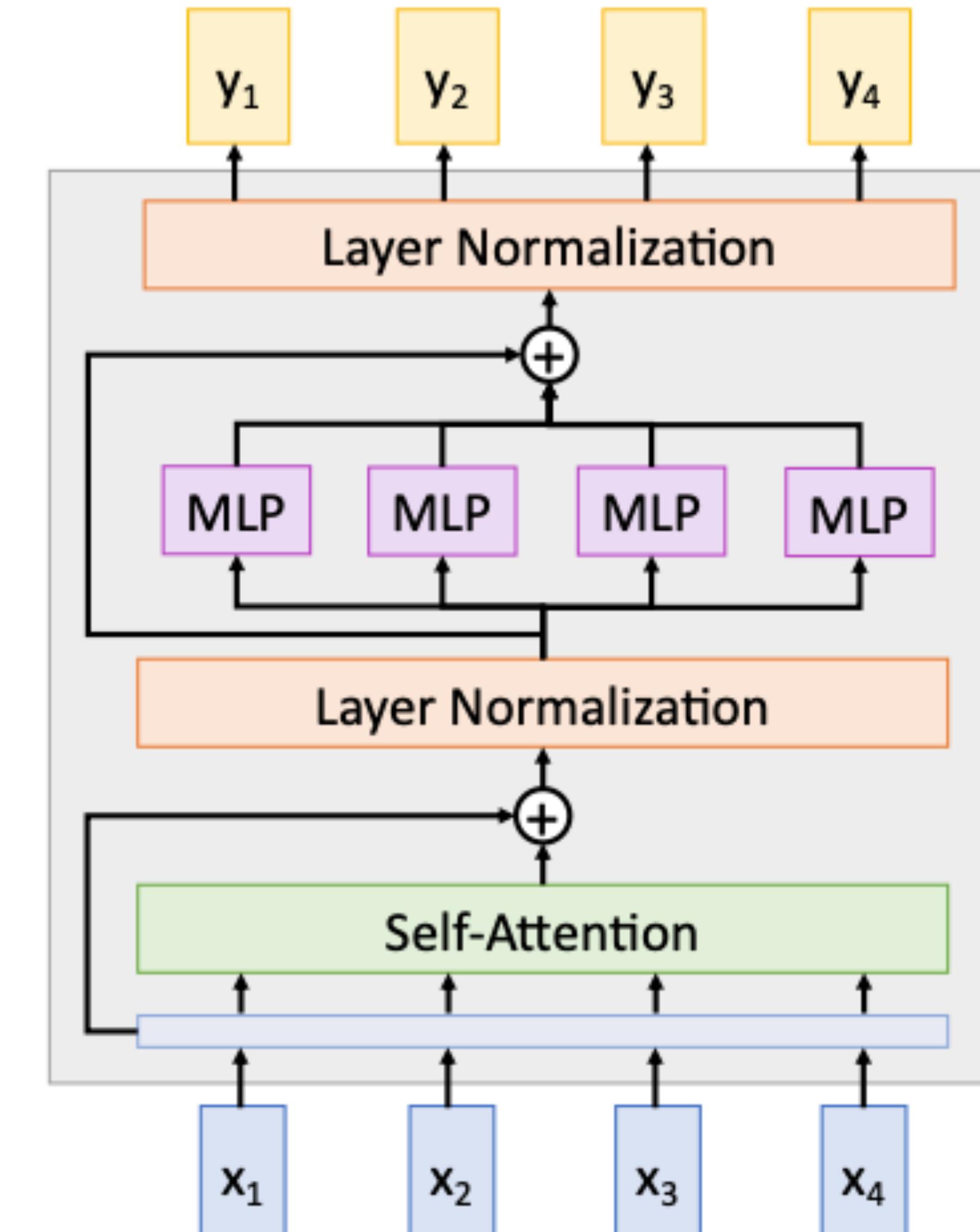
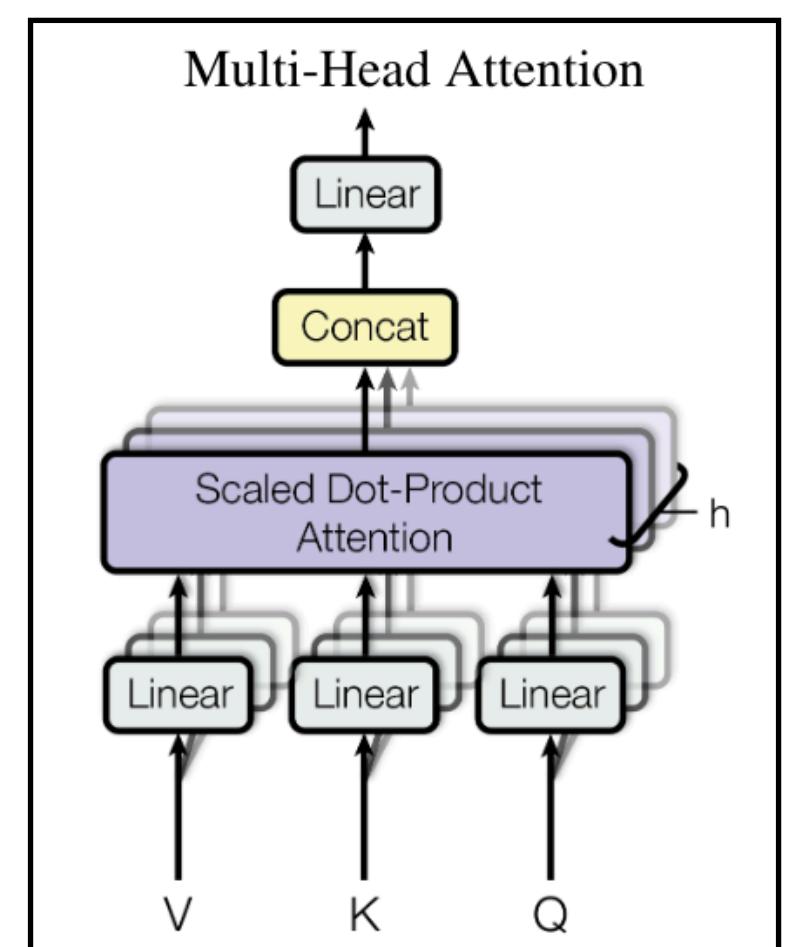
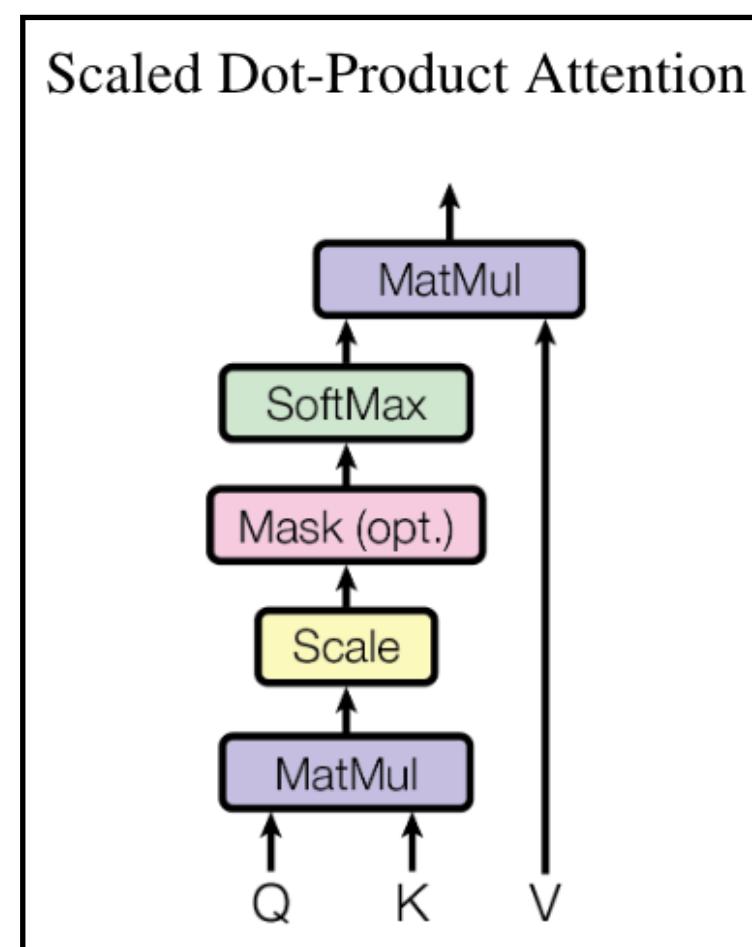


Current token can only see previous tokens

Source: Justin Johnson's EECS498/598 course

# Transformer

- Built entirely based on attention.
- LayerNorm and MLP work for each vector.
- Multi-head attention: concatenation of the outputs of multiple self-attention layers.
- Highly scalable & parallelizable.
- LMs only use decoder-only transformer (i.e., w/ masked attention).



Attention is all you need

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

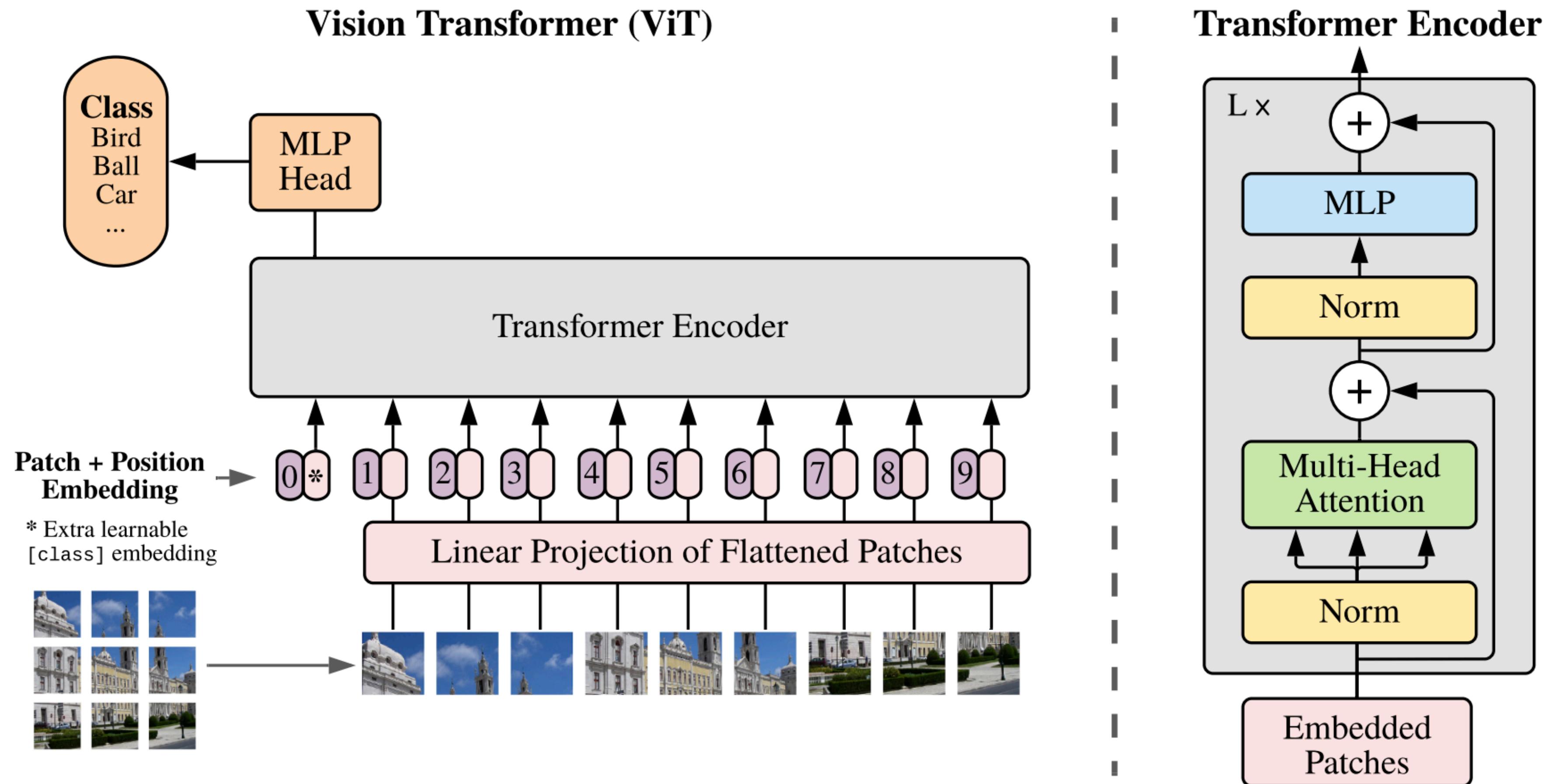
... to attend to **all** positions in the decoder up to and including that position. **We need** to prevent

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

☆ Save ⚡ Cite Cited by 113407 Related articles All 87 versions ➔

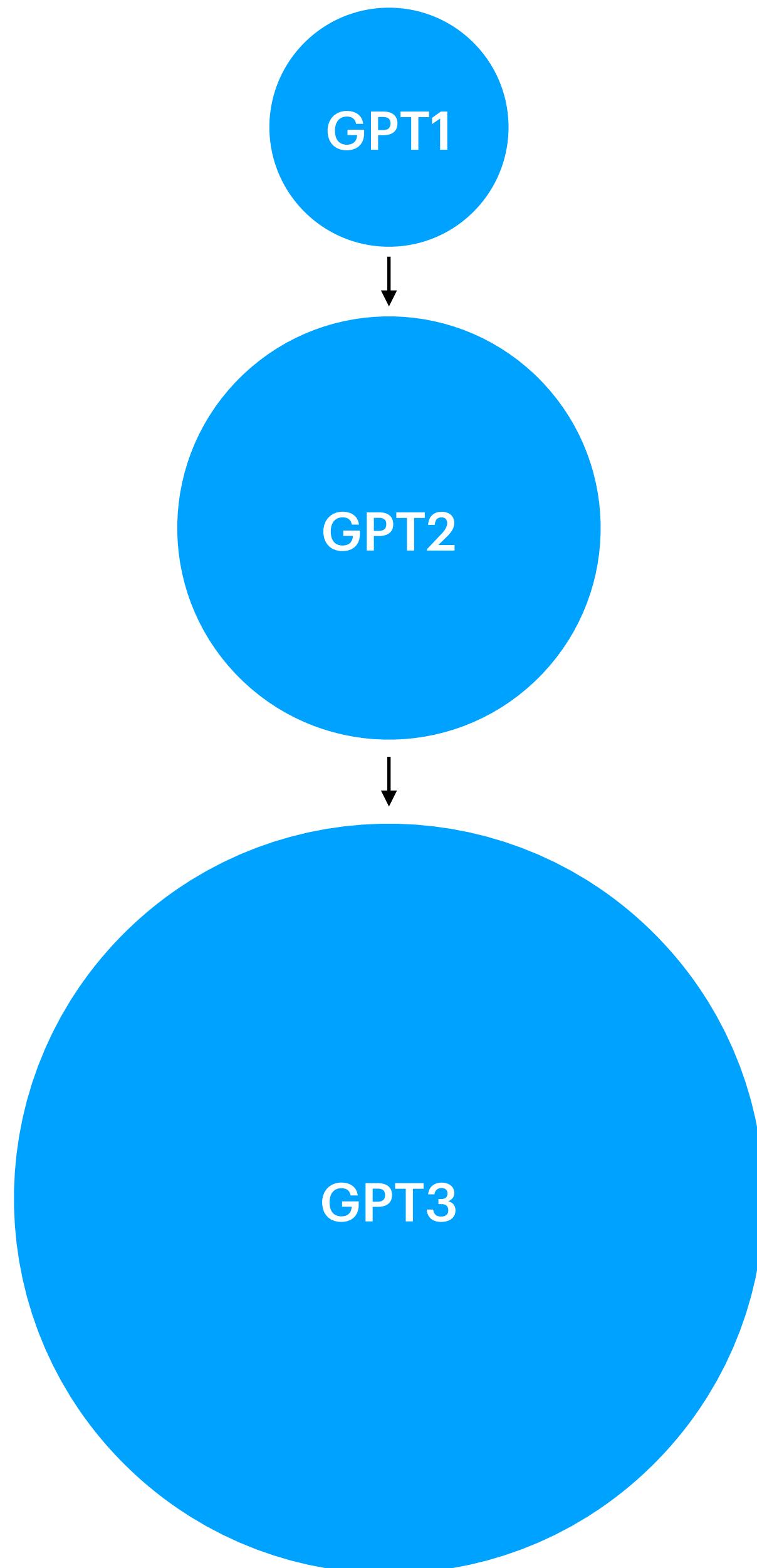
Source: Justin Johnson's EECS498/598 course

# Vision Transformer



# Pre-training: GPT

- Learning objective: next-token prediction.
- GPT1:
  - 12 decoder-only transformer layers, 117M params
  - 512 context tokens, 768-d hidden size
  - trained on BooksCorpus (>7,000 books)
- GPT2:
  - 48 layers, 1.5B params
  - 1024 context tokens, 1600-d hidden size
  - trained on larger datasets
- GPT3:
  - 96 layers, 175B params
  - 2048 context tokens, 12288-d hidden size
  - trained on even larger datasets (300B tokens)



# ...increasingly convincing generations (by GPT2)

**Context (human-written):** In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

**GPT-2:** The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

# ...increasingly convincing generations (by GPT3)

---

Context →

The City  
BY C. P. CAVAFY  
TRANSLATED BY EDMUND KEELEY  
[Poem text omitted]

SOME TREES  
John Ashbery  
[Poem text omitted]

Shadows on the Way  
Wallace Stevens

---

----- Generated Poem 1 -----

I must have shadows on the way  
If I am to walk I must have  
Each step taken slowly and alone  
To have it ready made

And I must think in lines of grey  
To have dim thoughts to be my guide  
Must look on blue and green  
And never let my eye forget  
That color is my friend  
And purple must surround me too

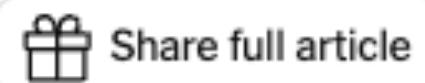
----- Generated Poem 3 -----

The sun was all we had. Now, in the shade  
All is changed. The mind must dwell on those  
White fields, that to its eyes were always old;  
Those ancient gleams, convoluting  
The way they lay among the huge roots,  
The great dark tomes of reverie,  
The plumed labyrinth of the sea.

When our own breath gonne faint and fade  
Up the long stairway of the dead  
We shall remember well

# ***The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work***

Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.

 Share full article



 1.3K



In one example of how A.I. systems use The Times's material, the suit showed that Browse With Bing, a Microsoft search feature powered by ChatGPT, reproduced almost verbatim results from Wirecutter, The Times's product review site. The text results from Bing, however, did not link to the Wirecutter article, and they stripped away the referral links in the text that Wirecutter uses to generate commissions from sales based on its recommendations.

“...is not what it is designed to do and not what it does...this isn't a document retrieval system. It is a large language model...” by OpenAI’s lawyer Joseph Gratz

# LM ≠ Assistant

Prompt *Explain the moon landing to a 6 year old in a few sentences.*

Completion GPT-3

*Explain the theory of gravity to a 6 year old.*

*Explain the theory of relativity to a 6 year old in a few sentences.*

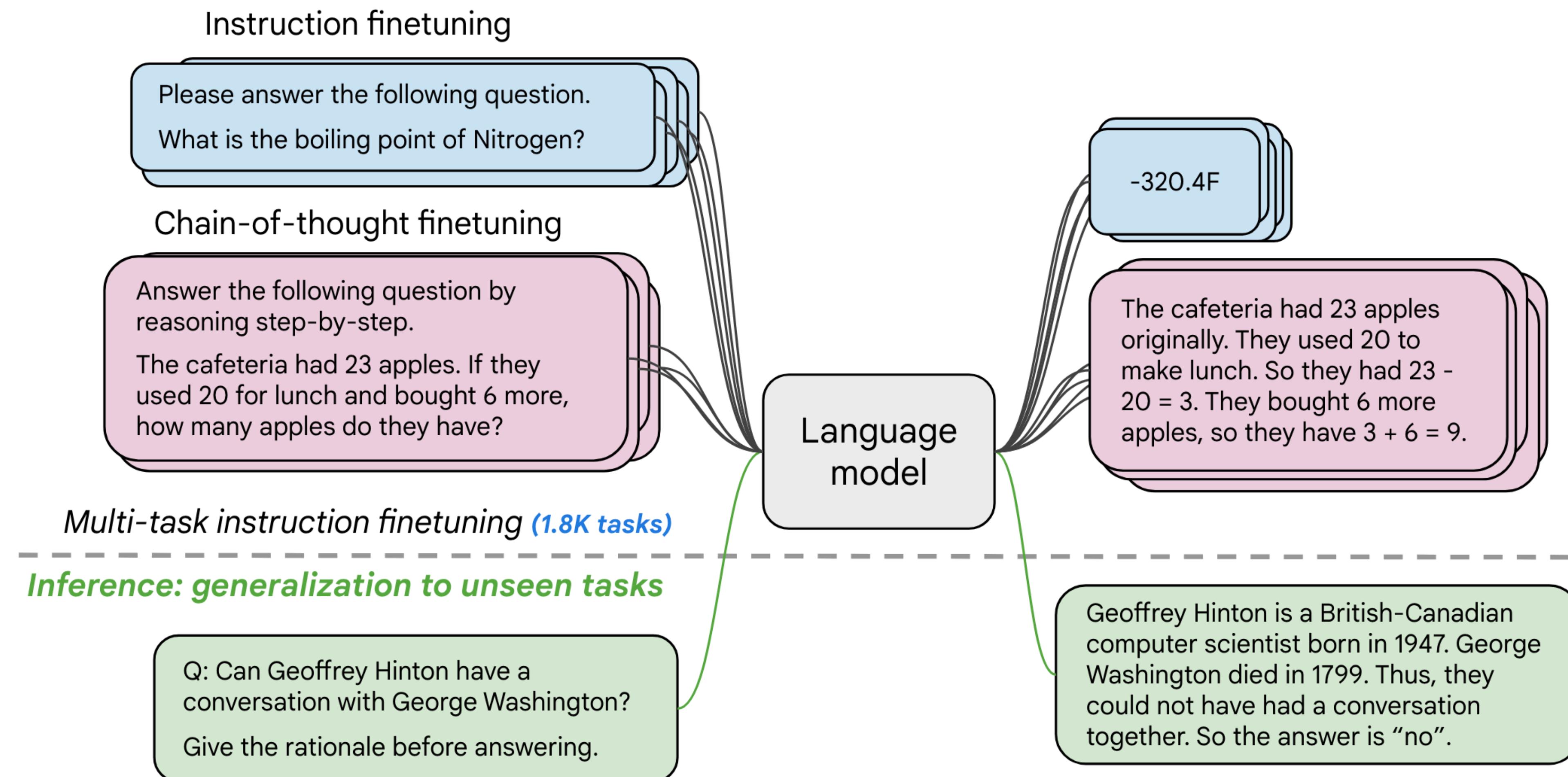
*Explain the big bang theory to a 6 year old.*

*Explain evolution to a 6 year old.*

InstructGPT

*People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.*

# Instruction tuning

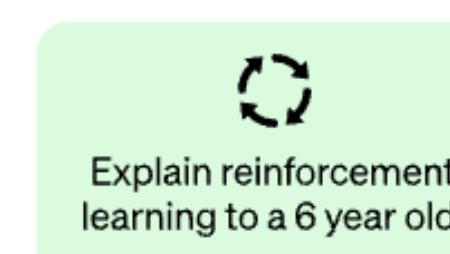


# Reinforcement Learning from Human Feedback (RLHF)

Step 1

Collect demonstration data and train a supervised policy.

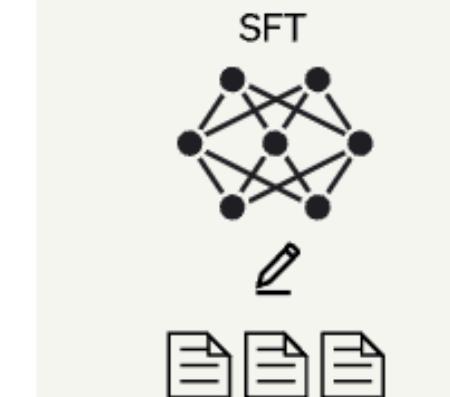
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



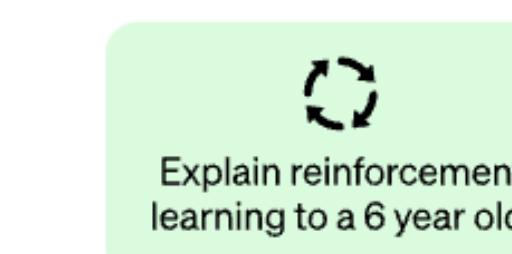
This data is used to fine-tune GPT-3.5 with supervised learning.



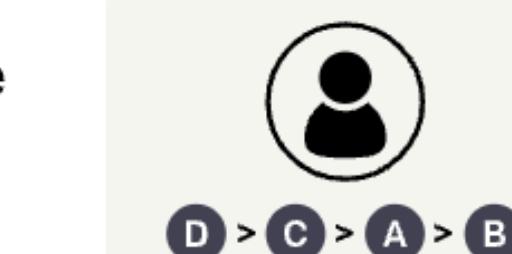
Step 2

Collect comparison data and train a reward model.

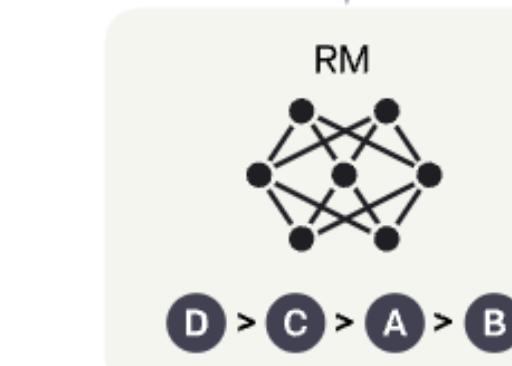
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



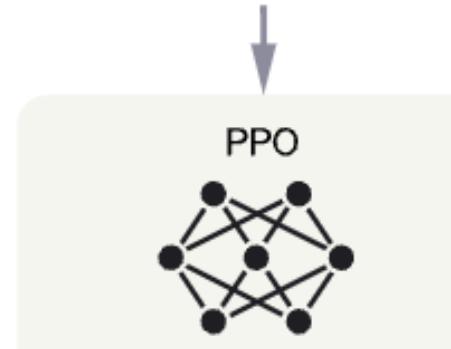
Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

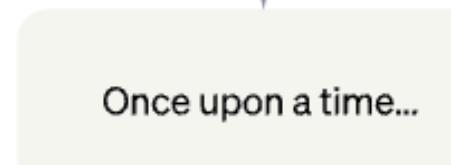
A new prompt is sampled from the dataset.



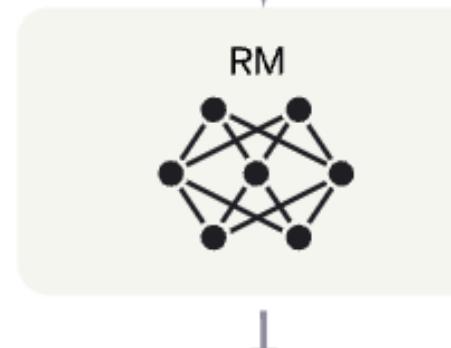
The PPO model is initialized from the supervised policy.



The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



# Further reading

- A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.
- <https://openai.com/index/chatgpt/>
- Liu et al. "Visual instruction tuning." Advances in neural information processing systems 36 (2024).

# Happy Coding

