

https://introml.mit.edu

6.390 Intro to Machine Learning

Lecture 9: Transformers

Shen Shen

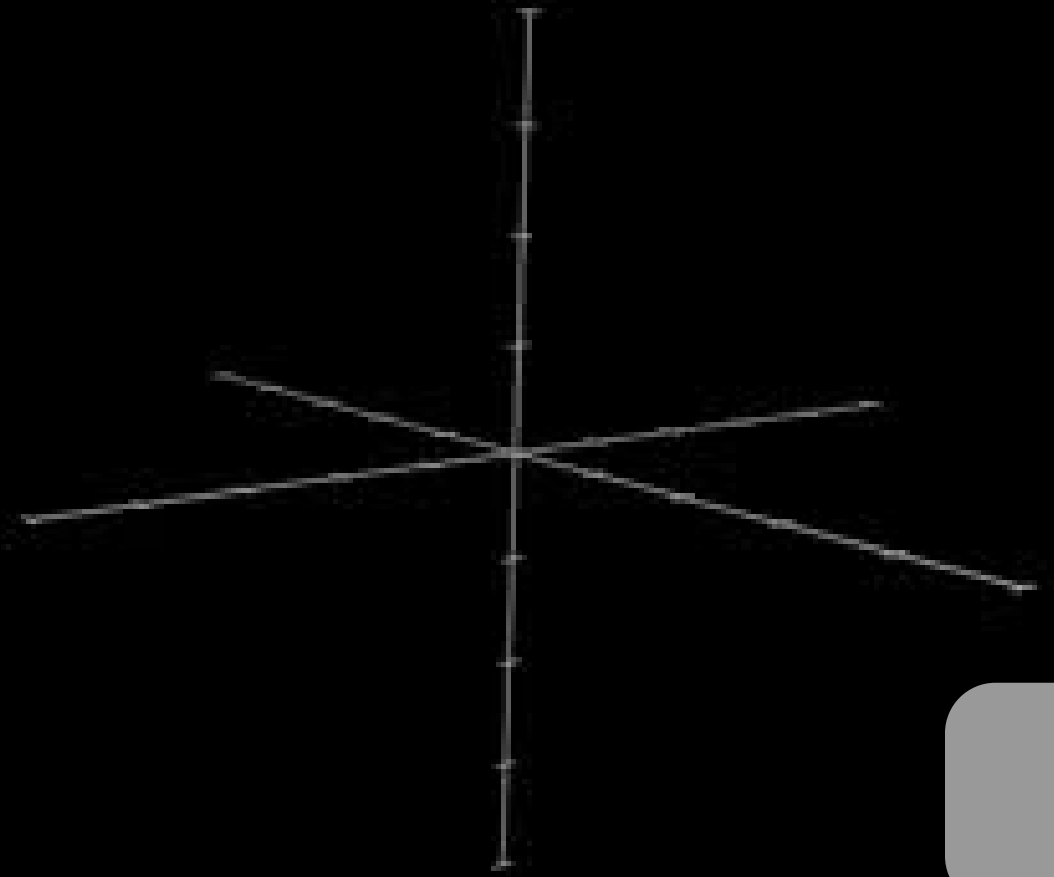
Nov 2, 2026

2:30pm, Room 10-250

Slides and Lecture Recording

Recap: **Word embedding**

Words → Vectors

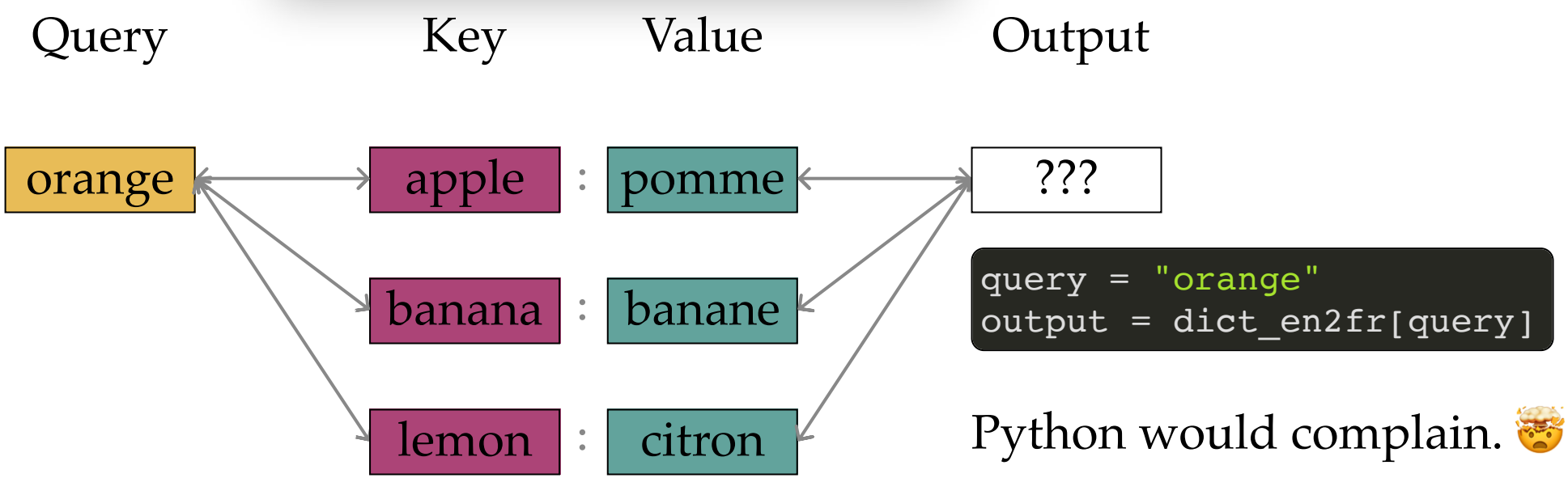


[video edited from 3b1b]

Good word-embeddings space is equipped with *semantically meaningful vector arithmetic*

this enables "soft" dictionary look-up:

```python
dict_en2fr = {
  "apple" : "pomme",
  "banana": "banane",
  "lemon" : "citron"}
```



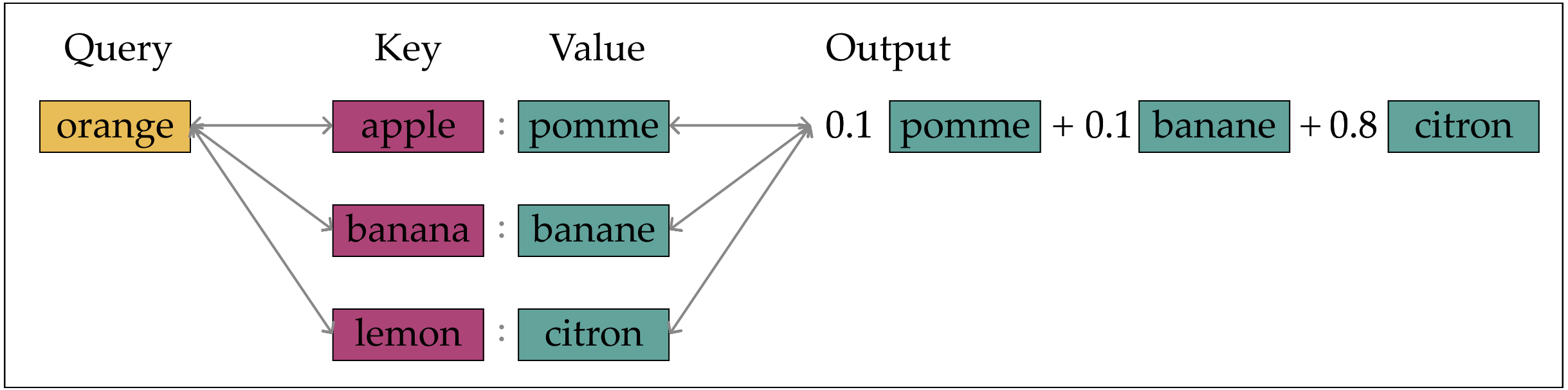

```python
query = "orange"
output = dict_en2fr[query]
```

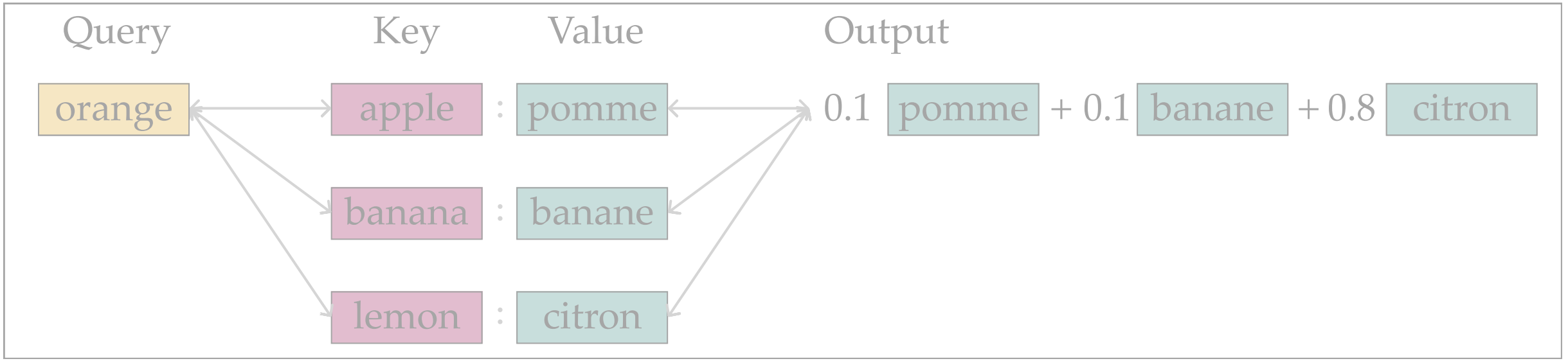
Python would complain. 🤯

```python
dict_en2fr = {
  "apple" : "pomme",
  "banana": "banane",
  "lemon" : "citron"}

query = "orange"
output = dict_en2fr[query]
```

But we can probably see the rationale behind something like this:





The attention mechanism in transformers mimics this:

- compare query and key (dot-product similarity) for merging percentages:

$$\text{softmax}\left(\begin{bmatrix}\text{orange}\\ \text{apple}\end{bmatrix}, \begin{bmatrix}\text{orange}\\ \text{banana}\end{bmatrix}, \begin{bmatrix}\text{orange}\\ \text{lemon}\end{bmatrix}\right) = \begin{bmatrix} 0.1 & 0.1 & 0.8 \end{bmatrix}$$

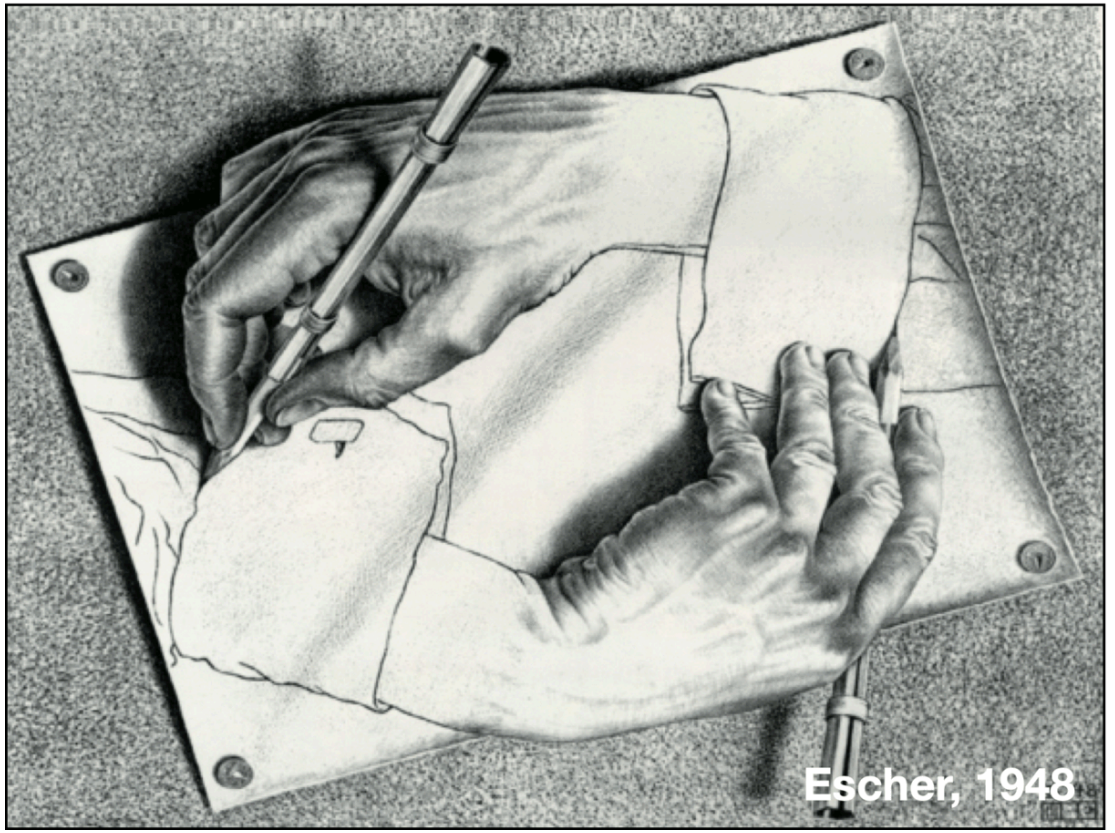
- output mixed values: $0.1\ \text{pomme} + 0.1\ \text{banane} + 0.8\ \text{citron}$

We picked these percentages by hand; let's see how a transformer comes up with them.

Outline

- Intuition and overview
- Attention mechanism
- Multi-head attention
- Attention beyond language

Large Language Models (LLMs) are trained in this self-supervised way:

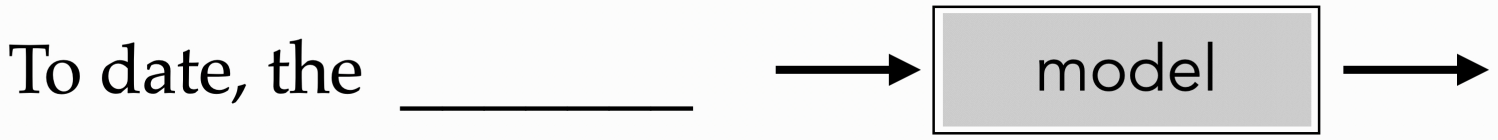


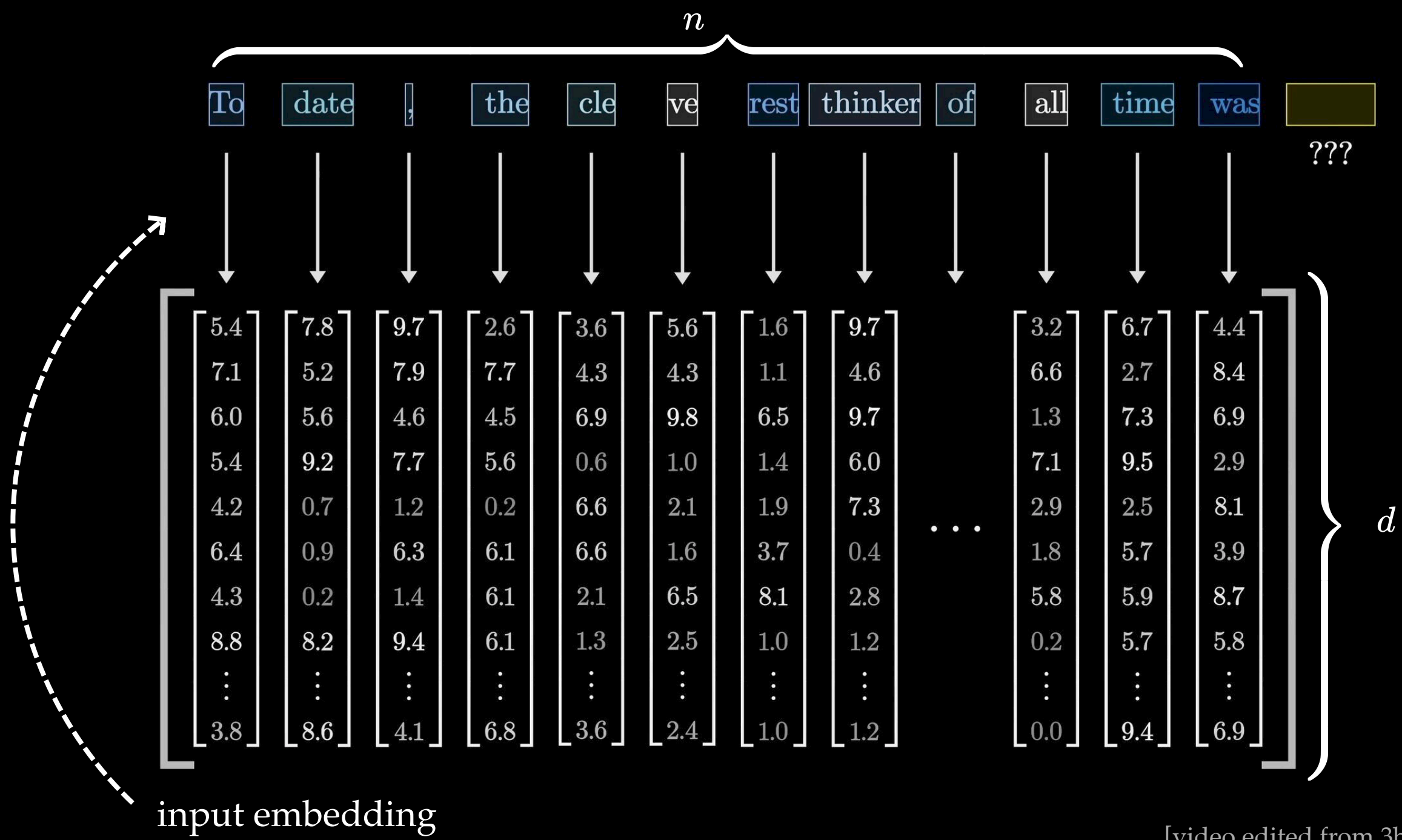
- Scrape the internet for plain texts.
- Cook up “labels” from these texts.
- Use the unhidden part to predict the hidden label.

"To date, the cleverest thinker of all time was Isaac."

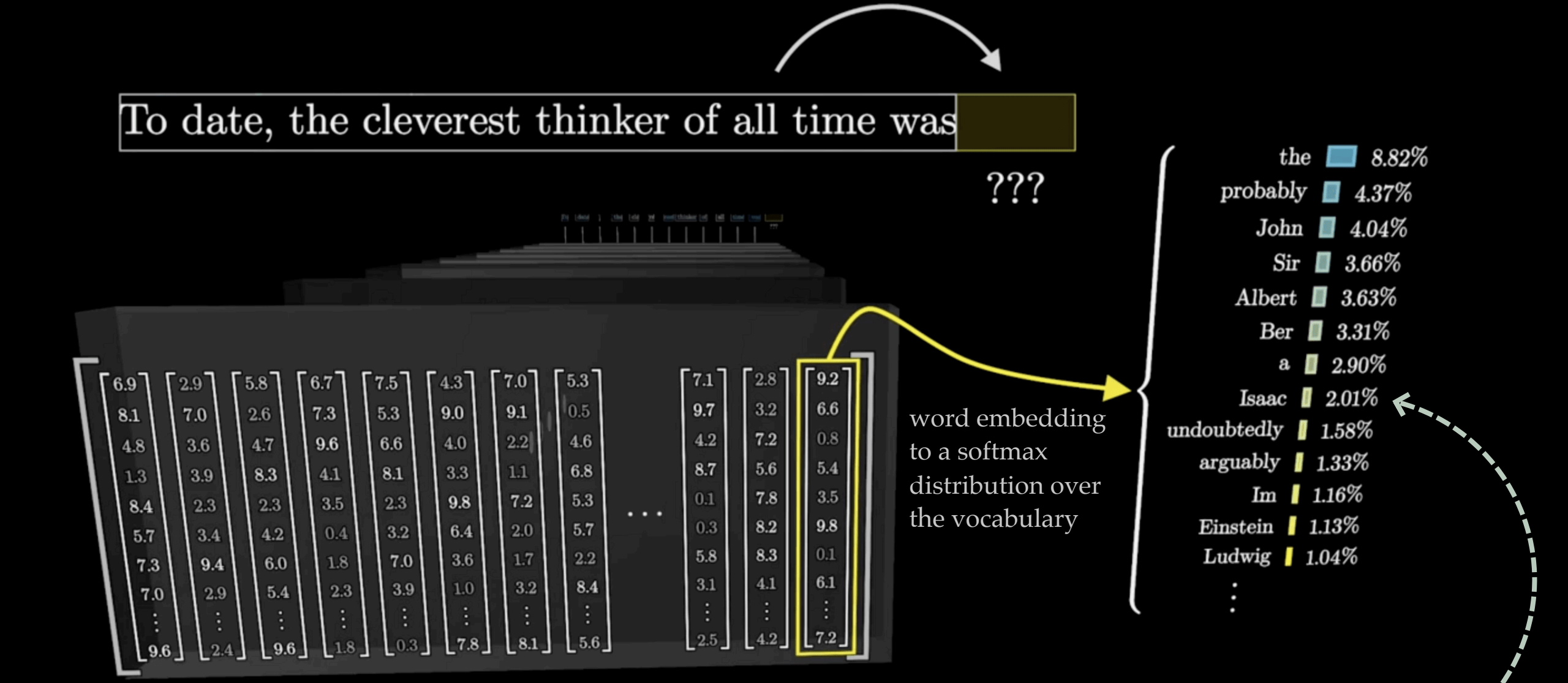
feature	label
To date, the	cleverest
To date, the cleverest	thinker
To date, the cleverest thinker	was
⋮	⋮
To date, the cleverest thinker of all time was	Isaac

auto-regressive prediction





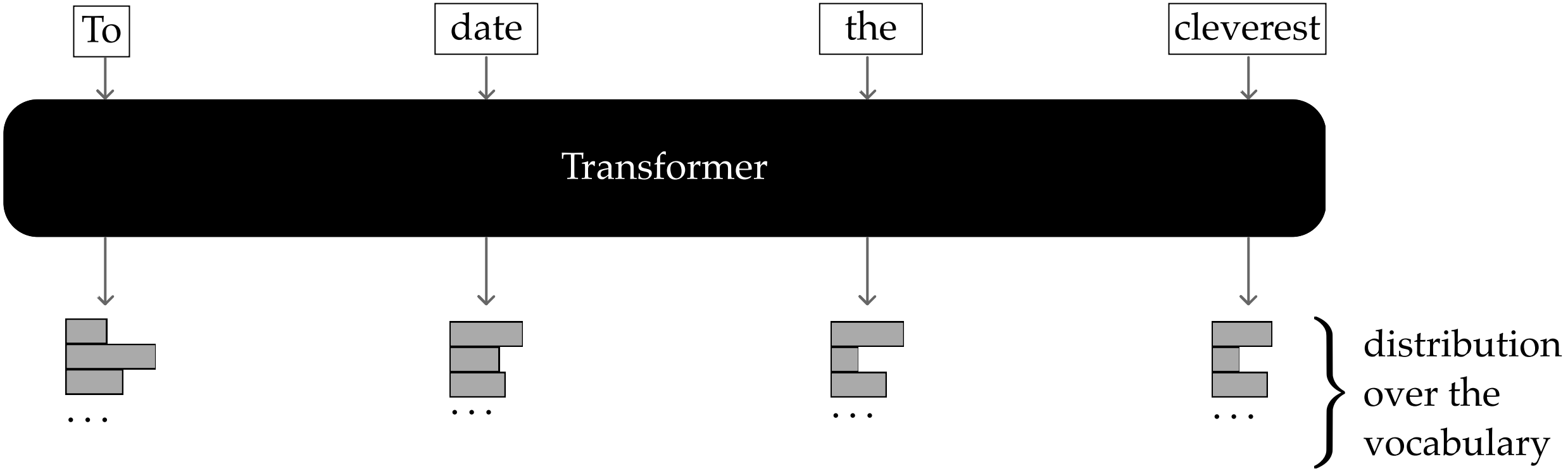
[video edited from 3b1b]



Minimizing cross-entropy loss drives the weights to assign higher probability to the correct next token

[video edited from 3b1b]

"To date, the cleverest [thinker] of all time was Isaac.

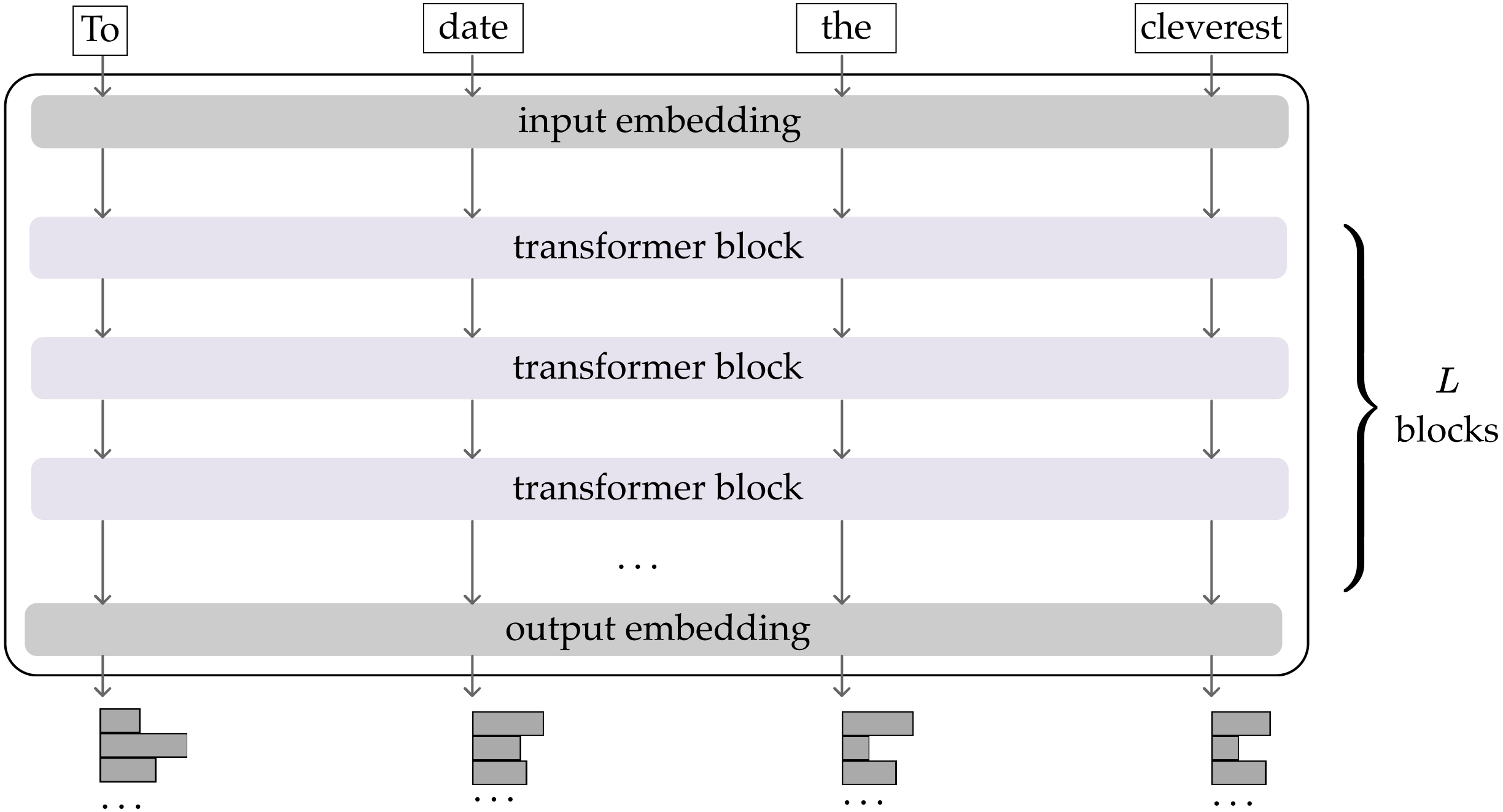


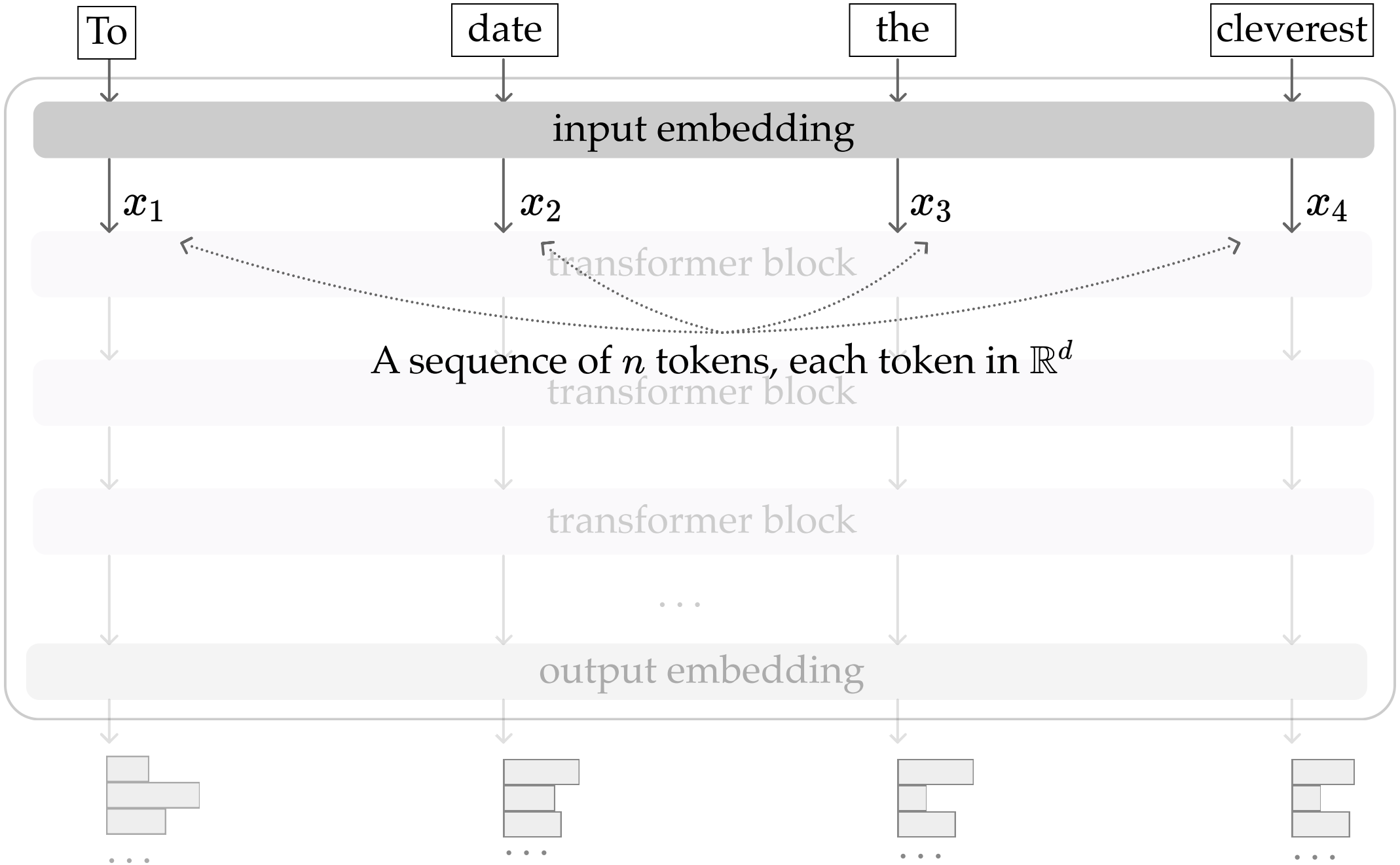
push for Prob("date") to be high

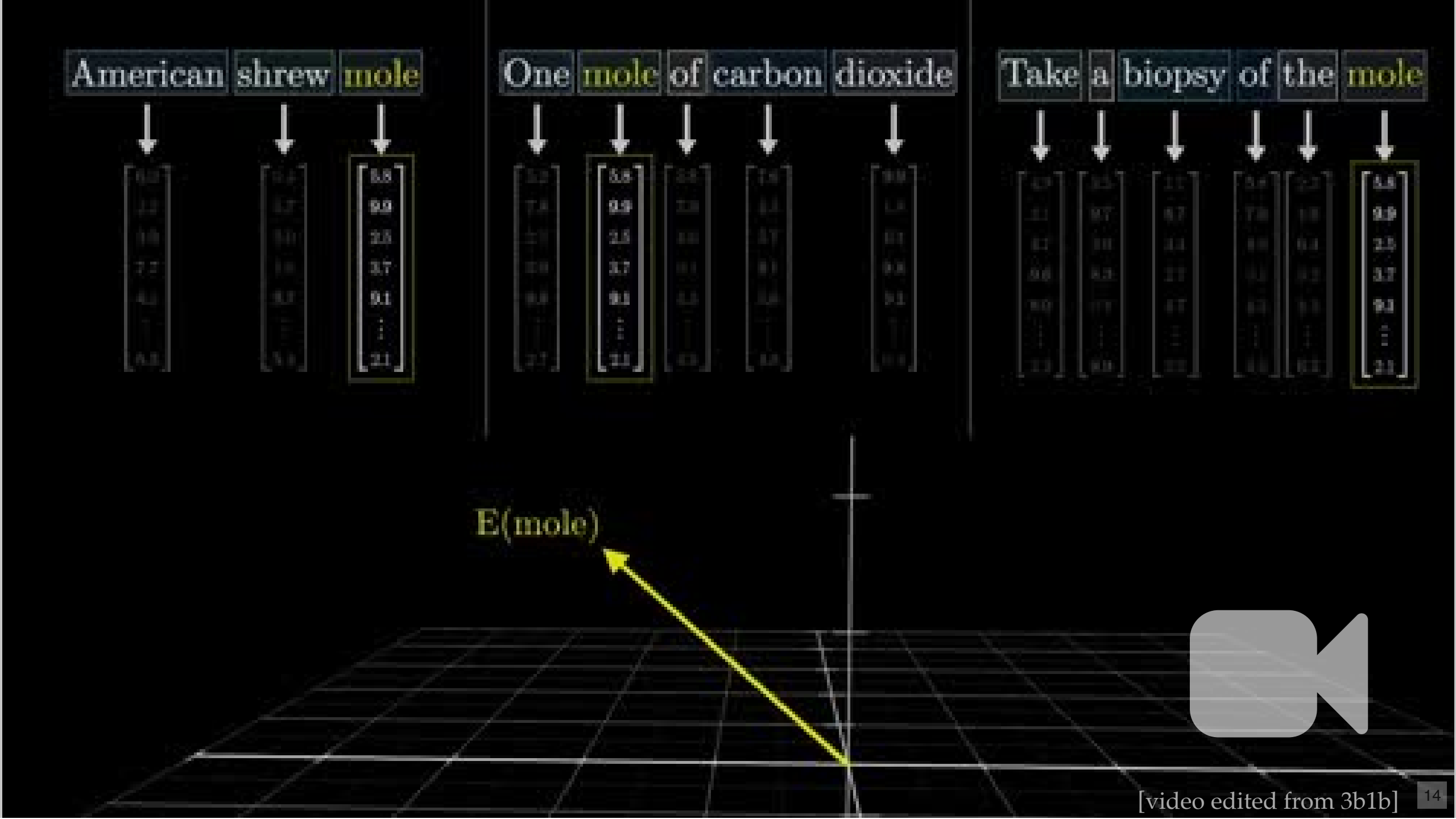
push for Prob("the") to be high

push for Prob("cleverest") to be high

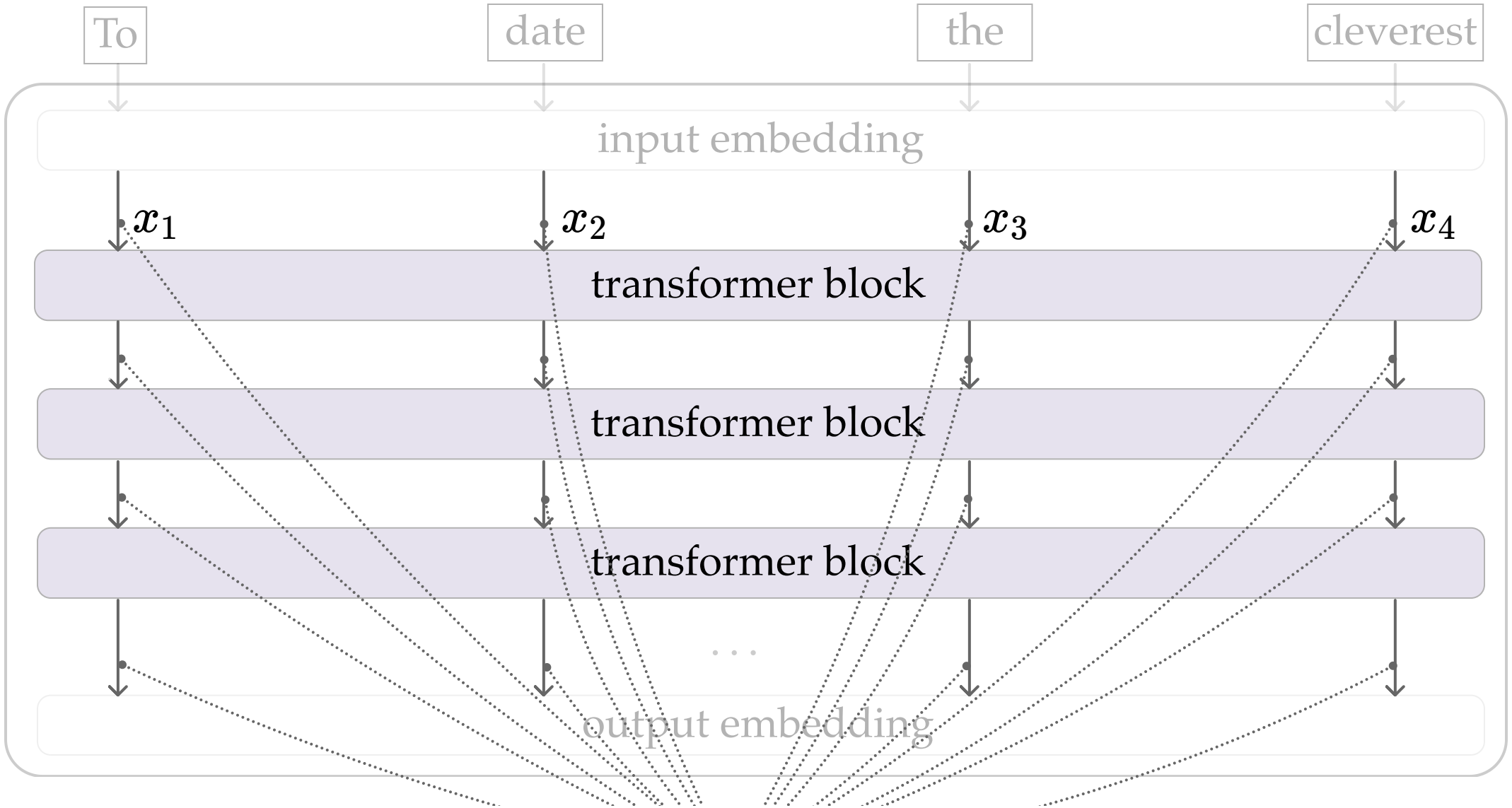
push for Prob("thinker") to be high


To
date
the
cleverest
input embedding
transformer block
transformer block
transformer block
. . .
output embedding
L
blocks
. . .
. . .
. . .
. . .


To
date
the
cleverest
input embedding
x_1
x_2
x_3
x_4
transformer block
A sequence of n tokens, each token in $\mathbb{R}^d$
transformer block
transformer block
. . .
output embedding

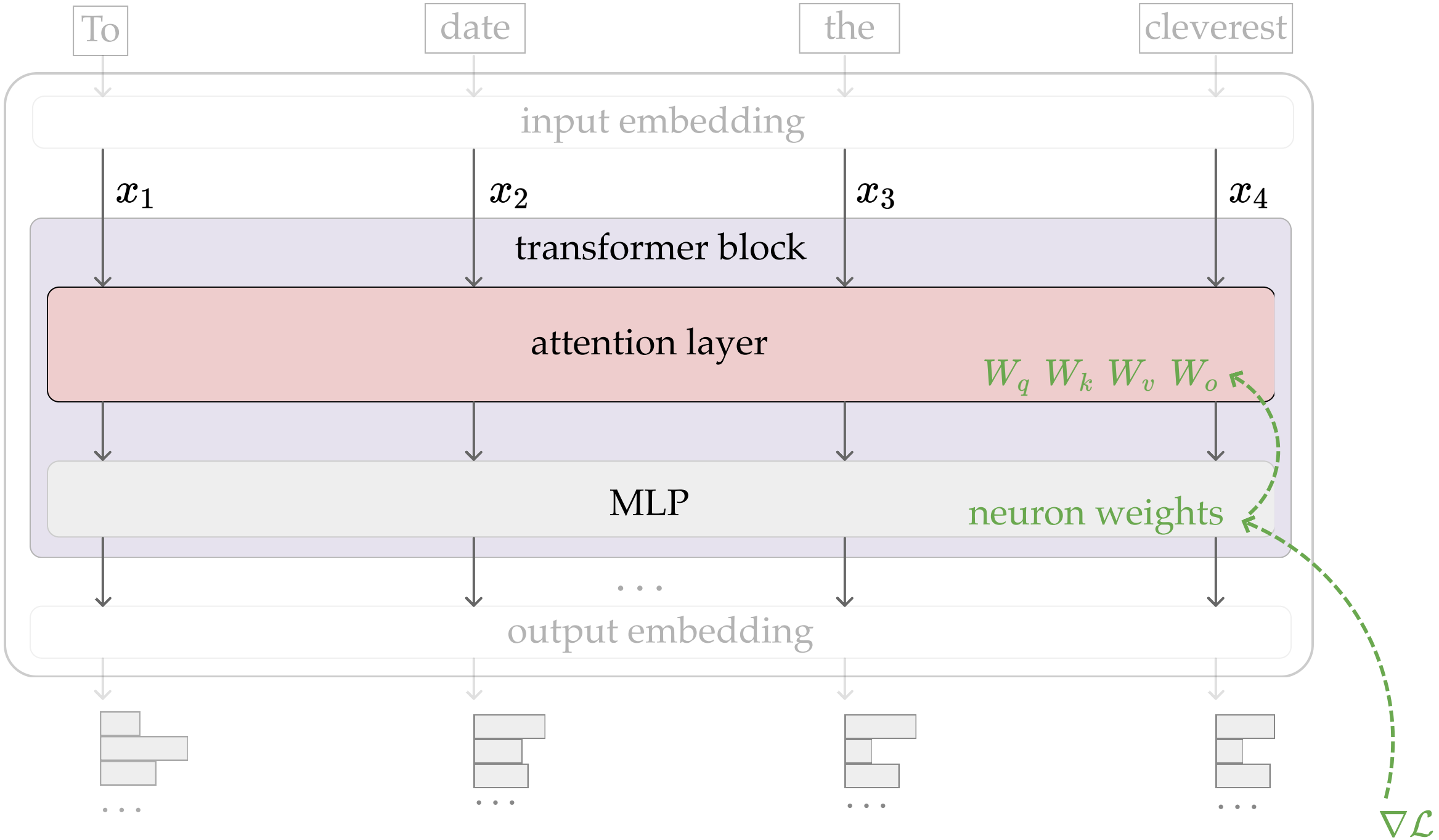


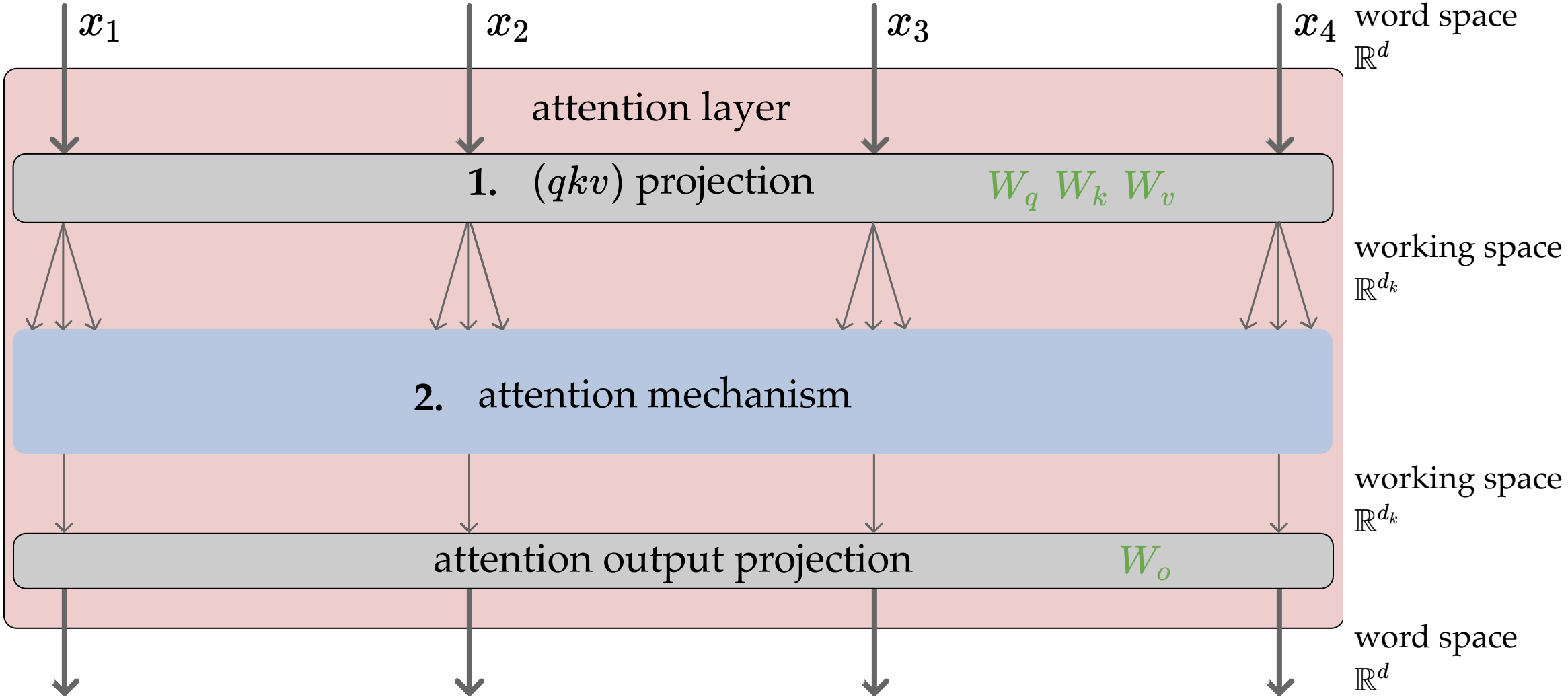
[video edited from 3b1b]



each of the n tokens transformed, block by block

within a shared d-dimensional word-embedding space.

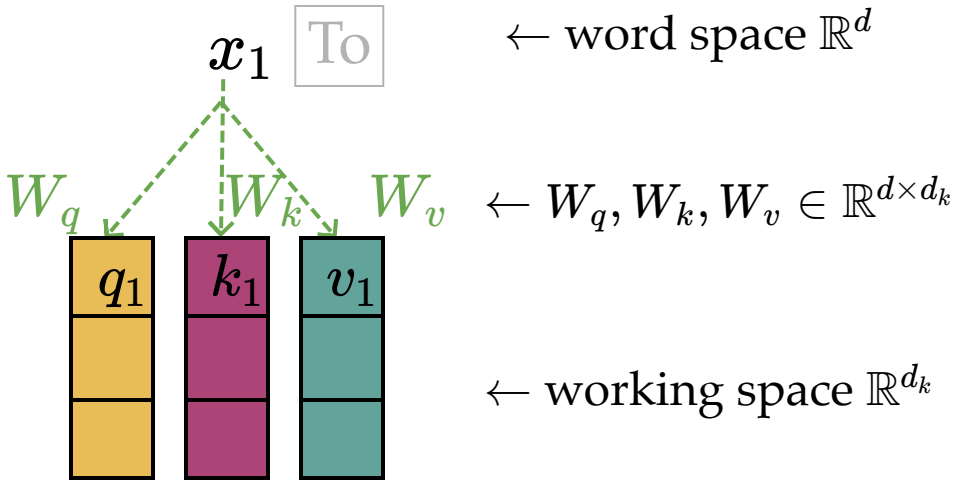

To
date
the
cleverest
input embedding
x_1
x_2
x_3
x_4
transformer block
attention layer
W_q W_k W_v W_o
MLP
neuron weights
output embedding
$\nabla \mathcal{L}$



working space $d_k \ll$ word space d

e.g., for GPT-2, $d = 768$, $d_k = 64$, both hyperparameters

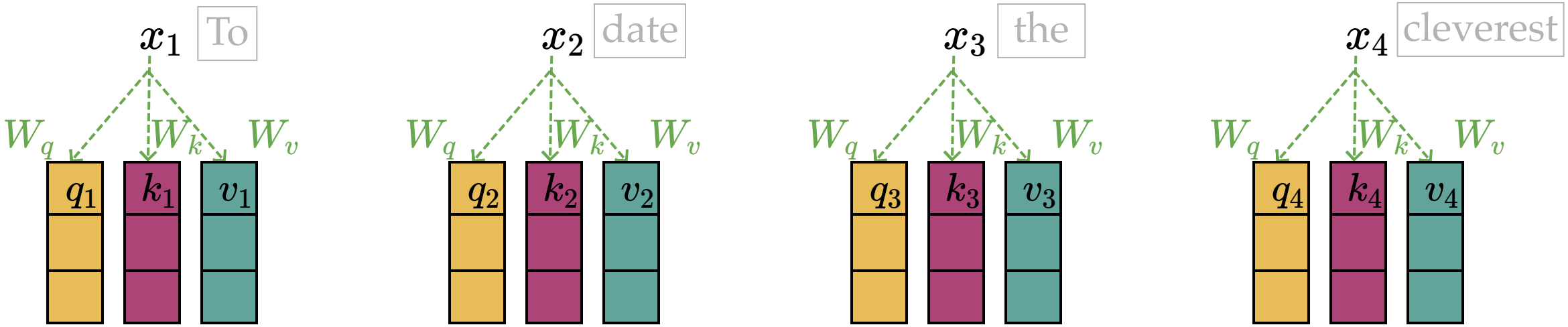
1. (query, key, value) projection — *why do we need anything beyond the word space?*



x has to play three roles:

- q: what it looks for (W_q learns *how* to ask)
- k: what it offers (W_k learns *how* to listen)
- v: what it hands over (W_v learns *how* to speak)

1. (query, key, value) projection — *why do we need anything beyond the word space?*

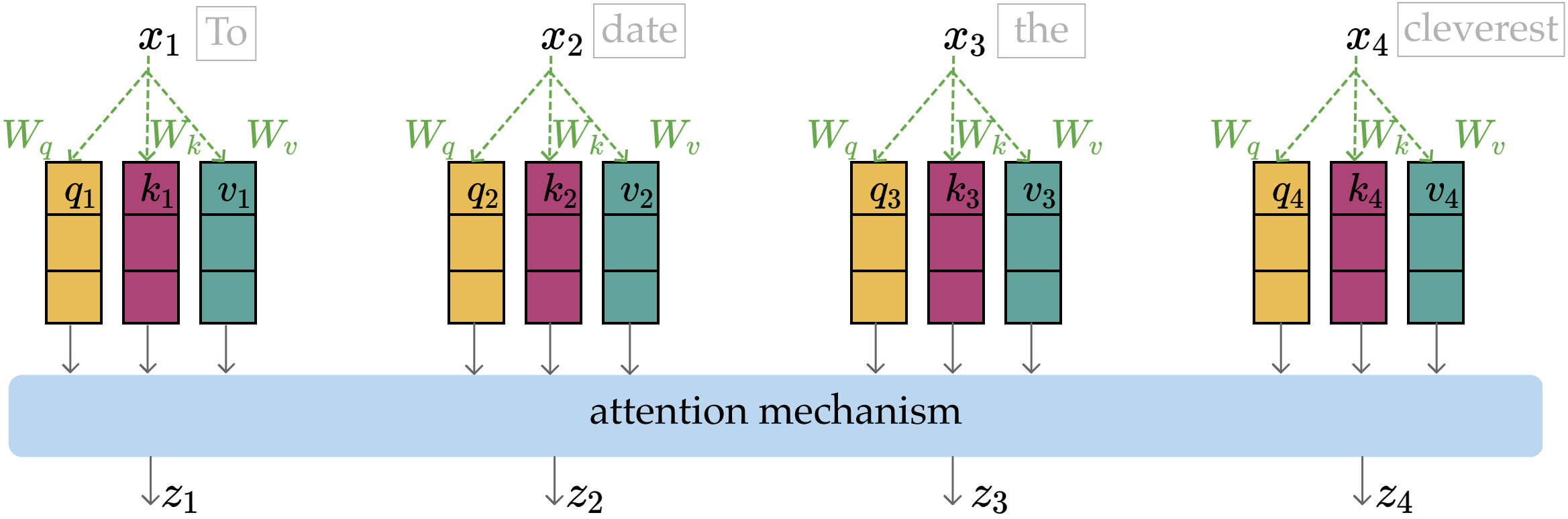


x has to play three roles:

- q: what it looks for (W_q learns *how* to ask)
- k: what it offers (W_k learns *how* to listen)
- v: what it hands over (W_v learns *how* to speak)

same W_q, W_k, W_v for every token: ***weight sharing,*** all in ***parallel***

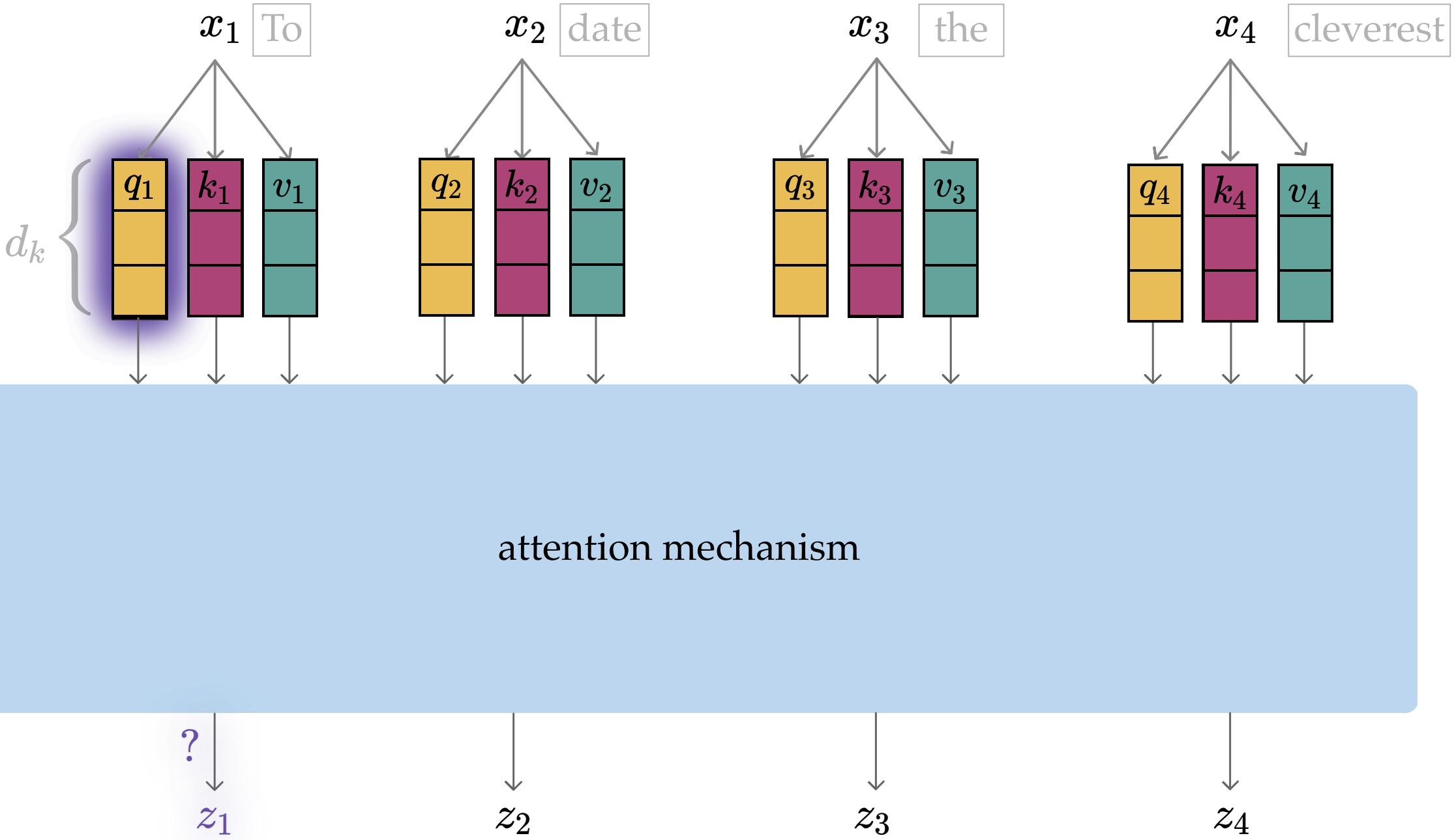
2. Attention mechanism

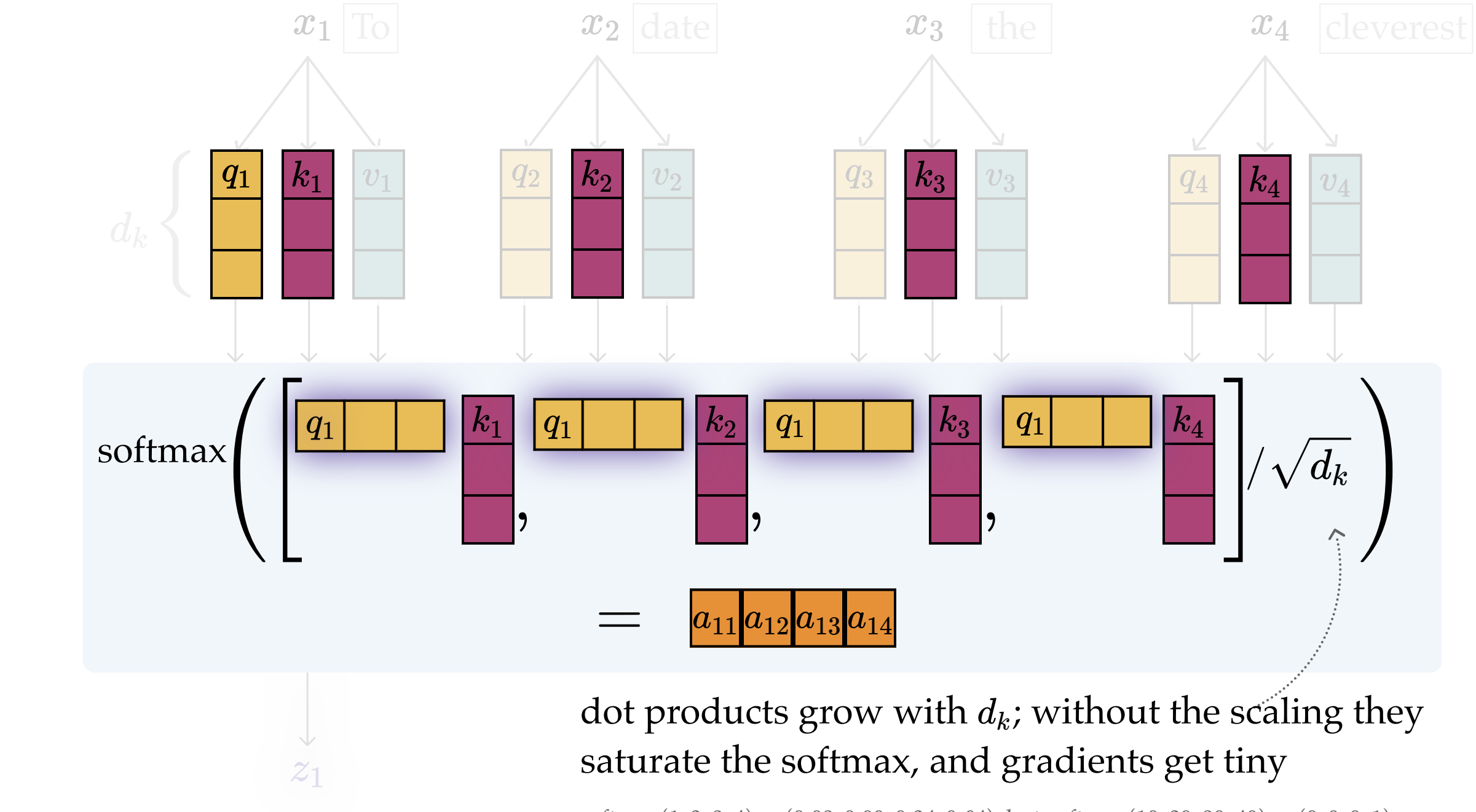


- Attention mechanism turns the projected (q, k, v) into z
- Each z is context-aware: a mixture of everyone's values, weighted by relevance

Outline

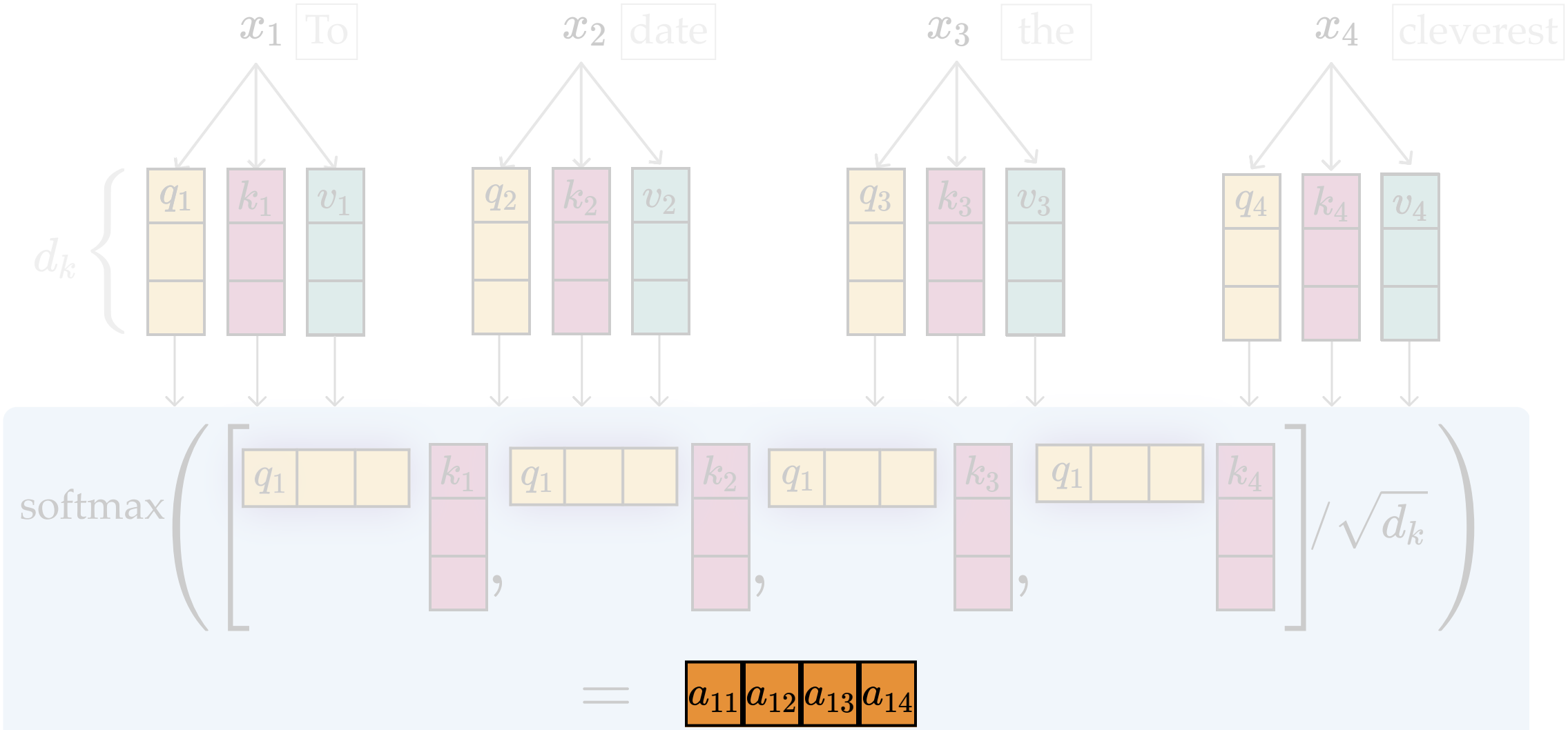
- Intuition and overview
- Attention mechanism
- Multi-head attention
- Attention beyond language


x_1 To
x_2 date
x_3 the
x_4 cleverest
q_1 k_1 v_1
q_2 k_2 v_2
q_3 k_3 v_3
q_4 k_4 v_4
d_k
attention mechanism
?
z_1
z_2
z_3
z_4



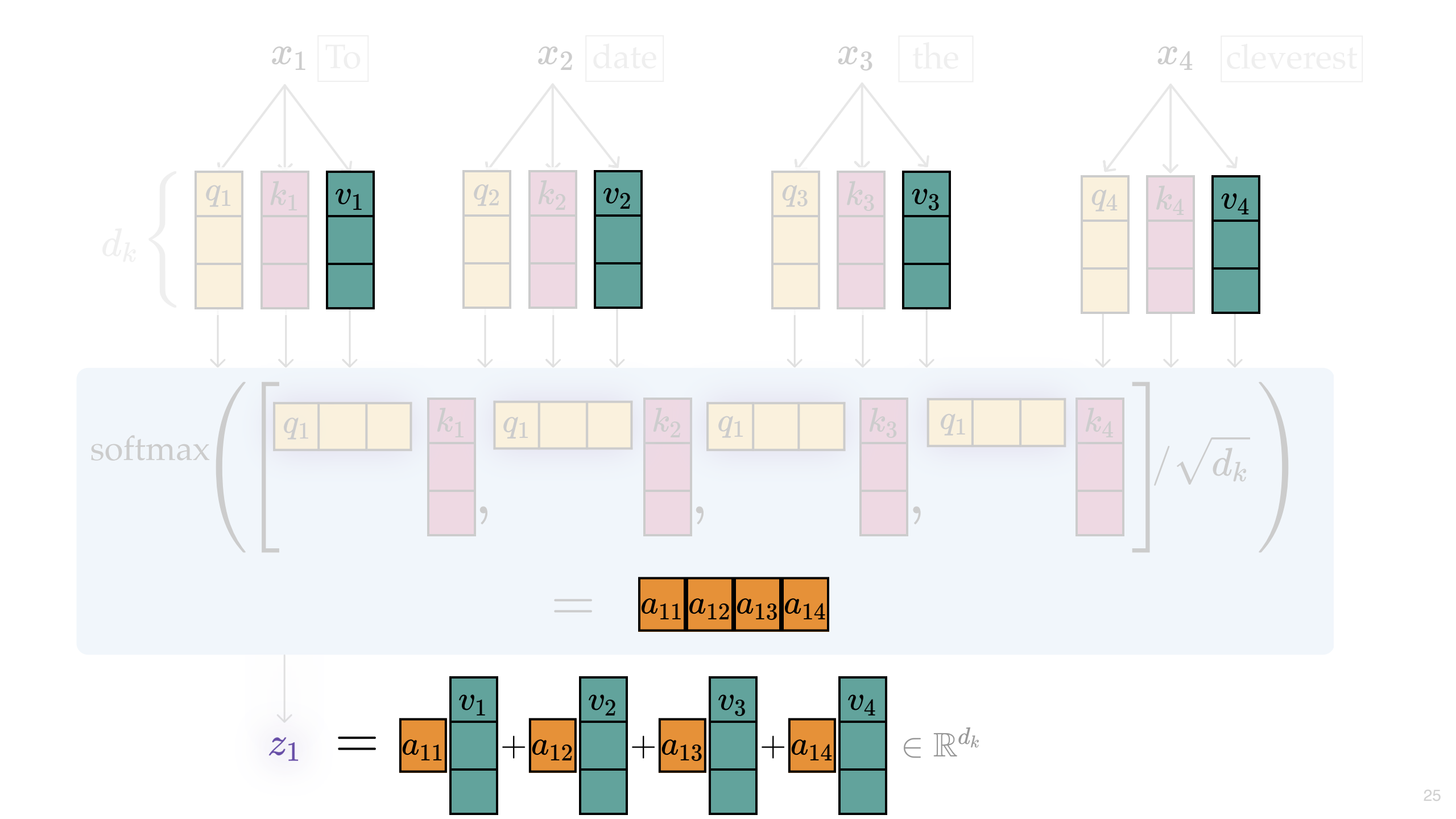
dot products grow with d_k; without the scaling they saturate the softmax, and gradients get tiny

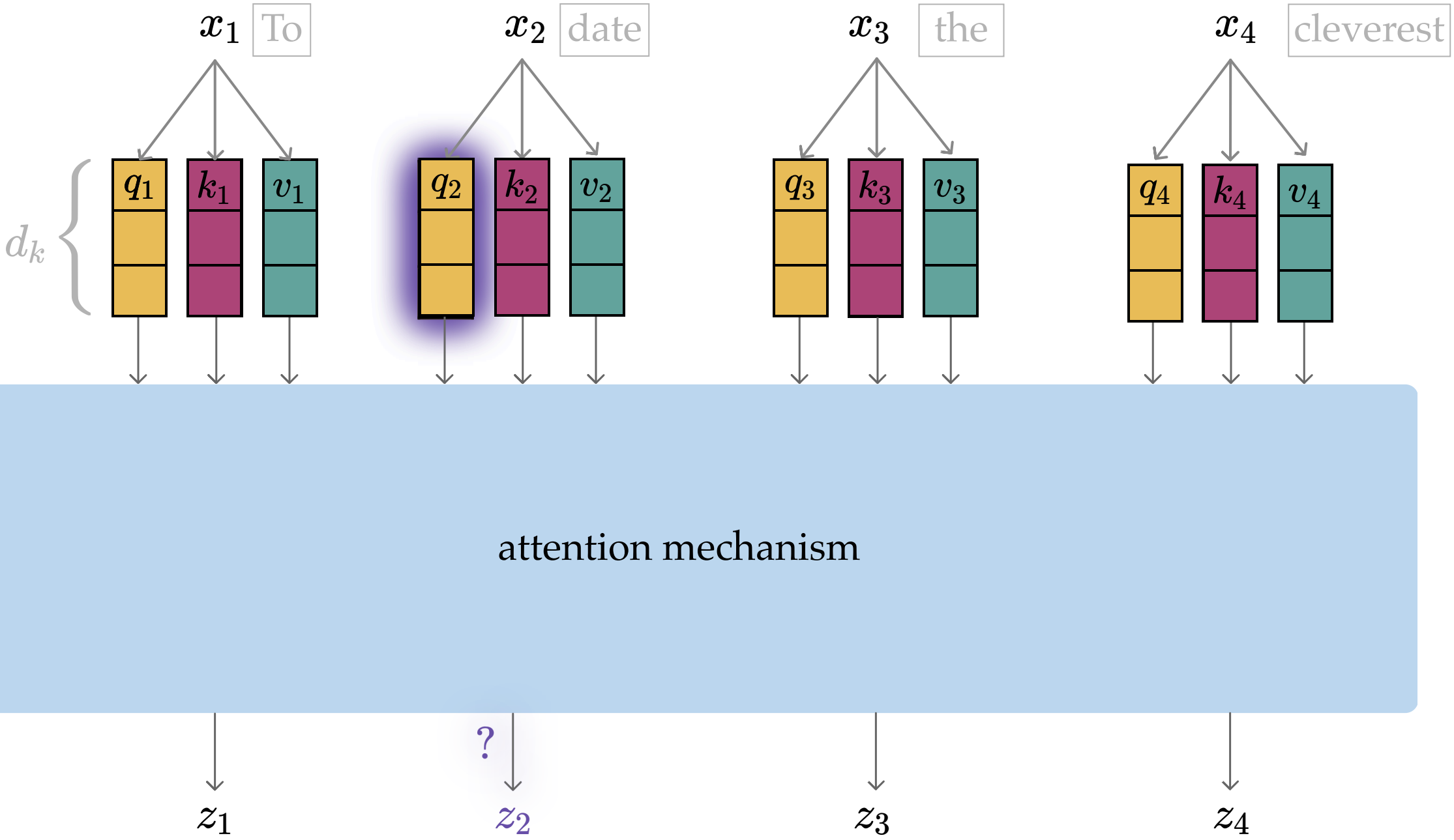
$\text{softmax}(1,2,3,4) \approx (0.03, 0.09, 0.24, 0.64)$, but $\text{softmax}(10,20,30,40) \approx (0,0,0,1)$

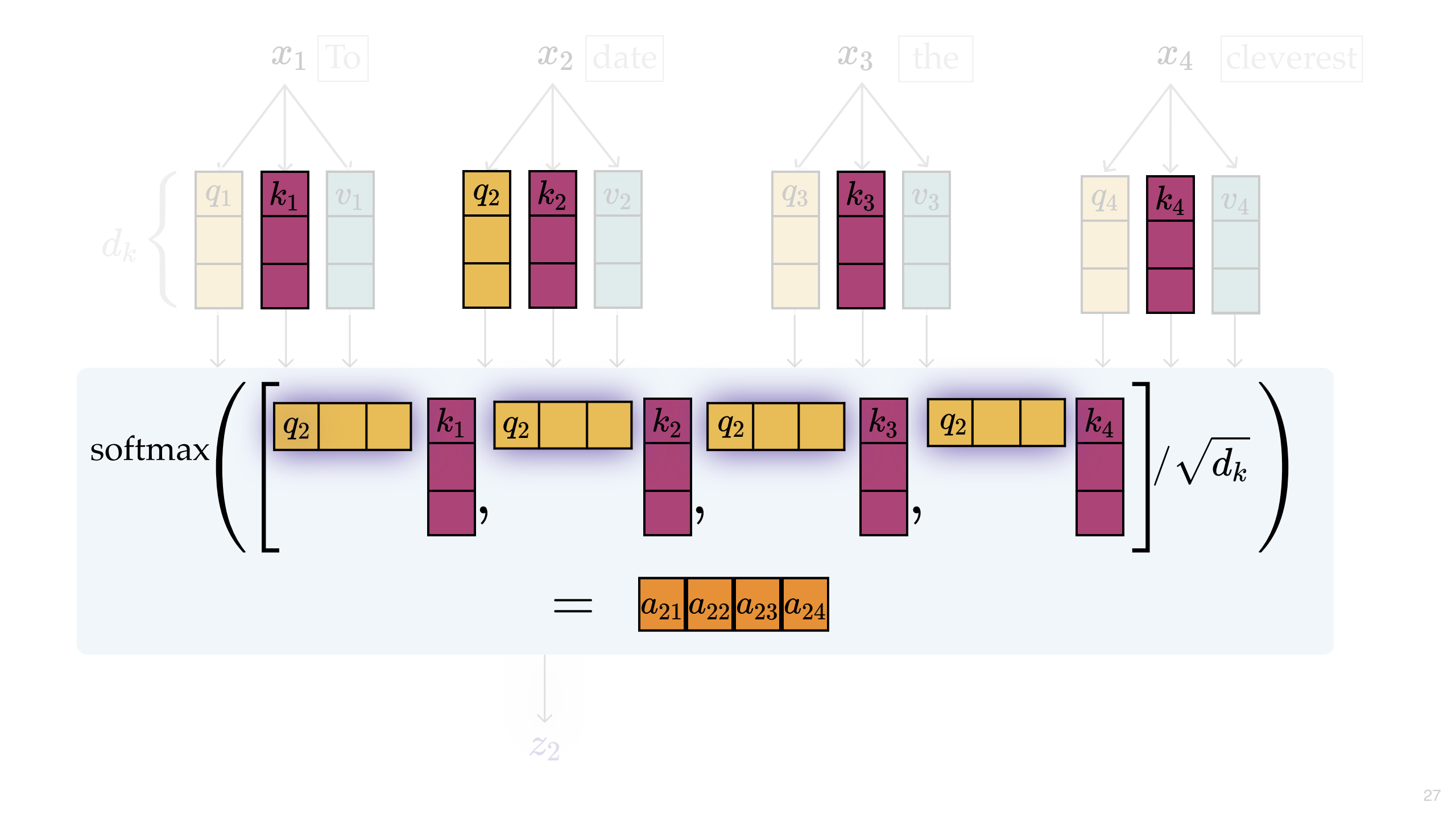


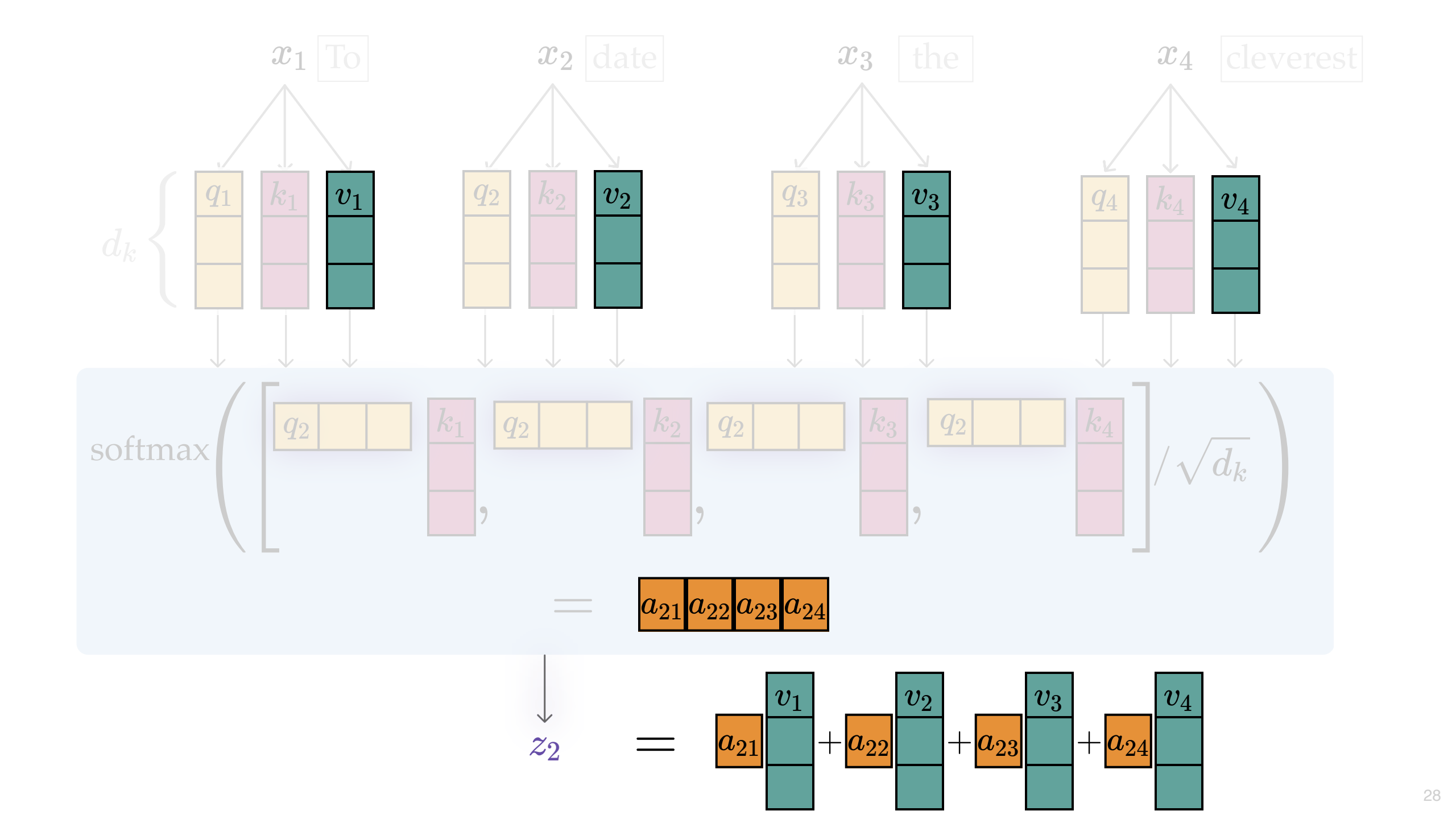
softmax'd attention scores: mixing percentages that sum to 1 (100%)

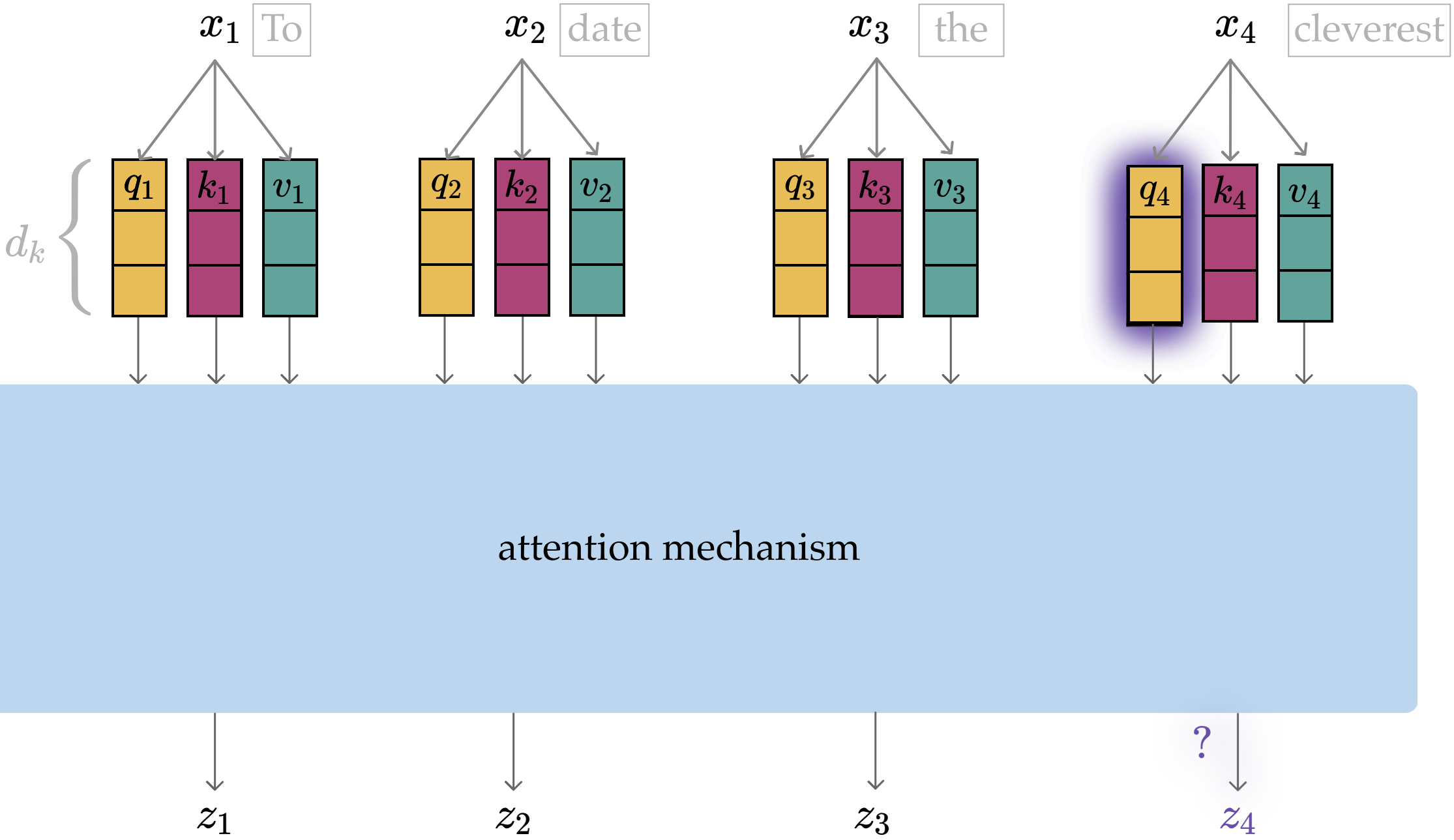
a_{12}: how much x_1 attends to x_2 when updating x_1's representation

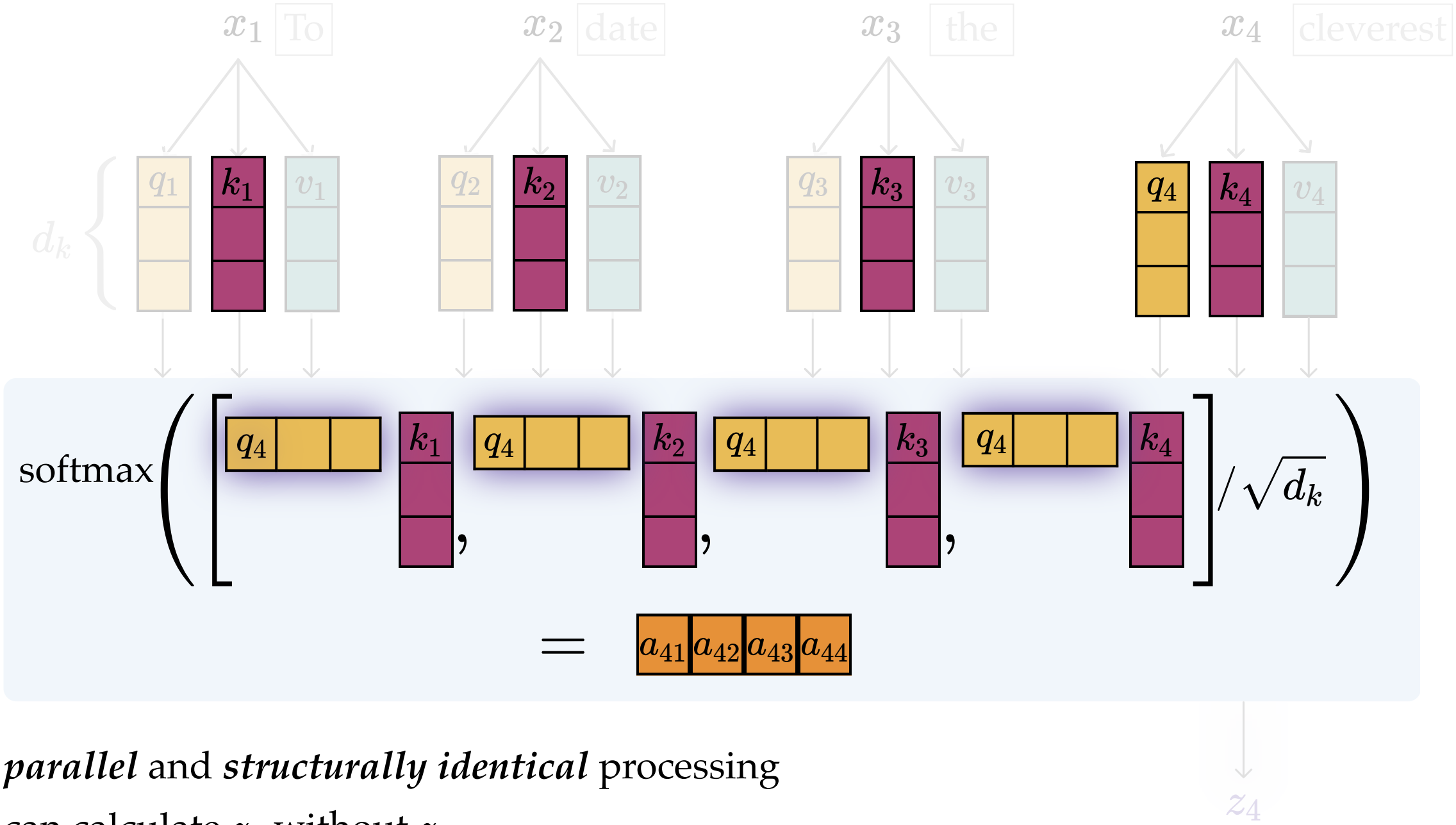

x_1 To
x_2 date
x_3 the
x_4 cleverest
q_1 k_1 v_1
q_2 k_2 v_2
q_3 k_3 v_3
q_4 k_4 v_4
d_k
$\text{softmax}\left(\left[q_1 k_1, q_1 k_2, q_1 k_3, q_1 k_4\right] / \sqrt{d_k}\right)$
$= a_{11} a_{12} a_{13} a_{14}$
$z_1 = a_{11} v_1 + a_{12} v_2 + a_{13} v_3 + a_{14} v_4 \in \mathbb{R}^{d_k}$


x_1 To
x_2 date
x_3 the
x_4 cleverest
d_k
q_1 k_1 v_1
q_2 k_2 v_2
q_3 k_3 v_3
q_4 k_4 v_4
attention mechanism
?
z_1
z_2
z_3
z_4


x_1 To
x_2 date
x_3 the
x_4 cleverest
q_1 k_1 v_1
q_2 k_2 v_2
q_3 k_3 v_3
q_4 k_4 v_4
d_k
softmax $\left(\left[q_2 k_1, q_2 k_2, q_2 k_3, q_2 k_4 \right] / \sqrt{d_k} \right)$
$= a_{21} a_{22} a_{23} a_{24}$
z_2

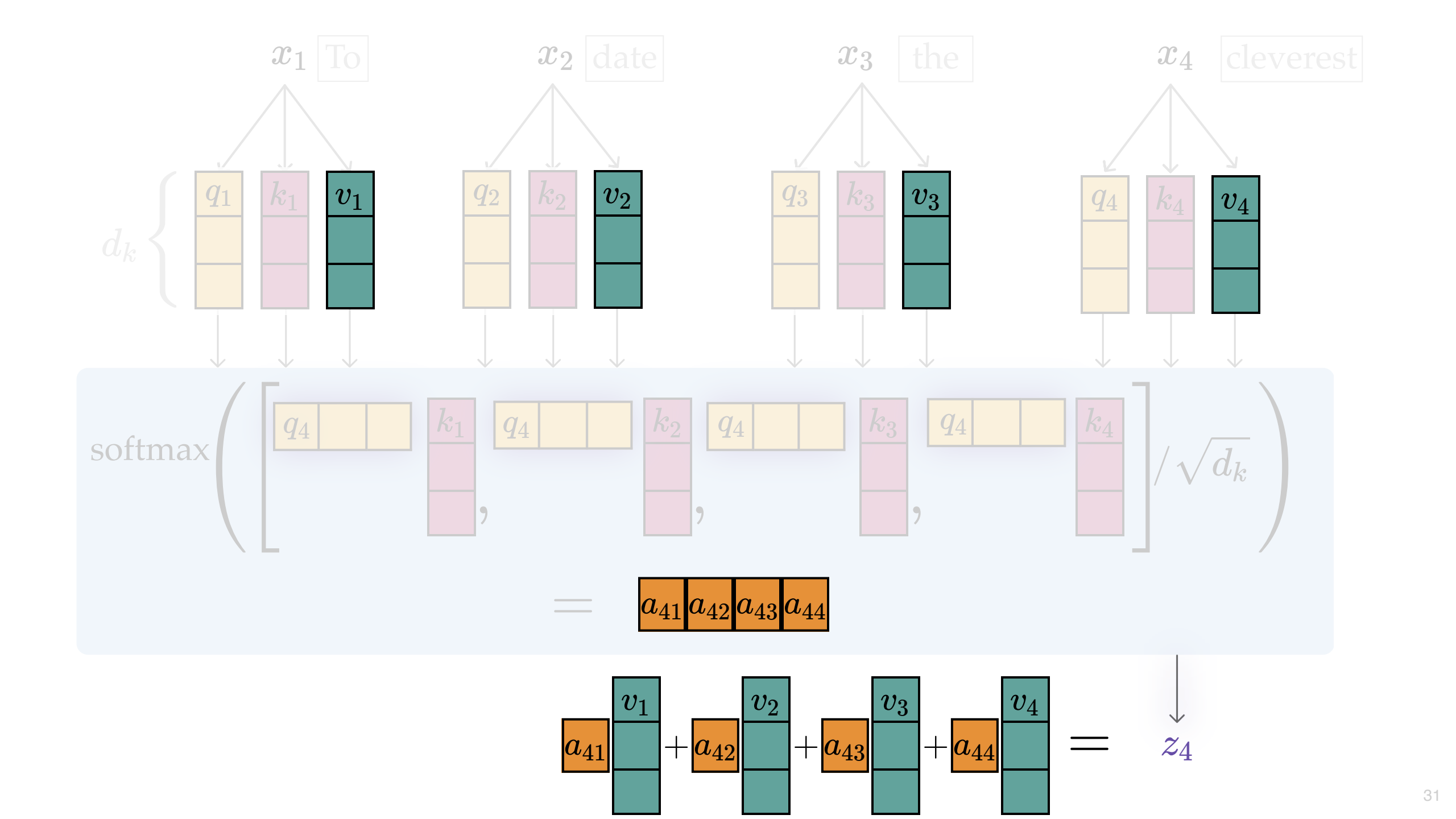

x_1 To
x_2 date
x_3 the
x_4 cleverest
q_1 k_1 v_1
q_2 k_2 v_2
q_3 k_3 v_3
q_4 k_4 v_4
d_k
softmax
$/\sqrt{d_k}$
$= a_{21}\ a_{22}\ a_{23}\ a_{24}$
z_2
$= a_{21} v_1 + a_{22} v_2 + a_{23} v_3 + a_{24} v_4$


x_1 To
x_2 date
x_3 the
x_4 cleverest
d_k
q_1 k_1 v_1
q_2 k_2 v_2
q_3 k_3 v_3
q_4 k_4 v_4
attention mechanism
?
z_1 z_2 z_3 z_4

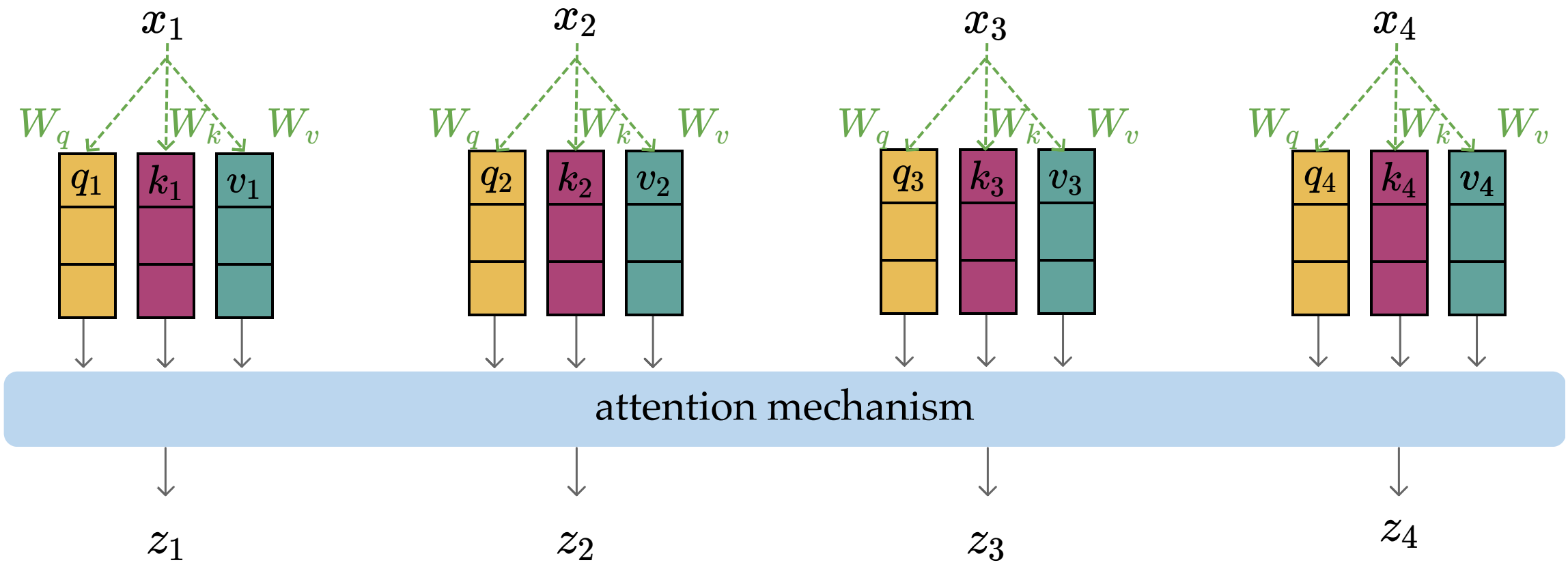


parallel and ***structurally identical*** processing

can calculate z_4 without z_3


x_1 To
x_2 date
x_3 the
x_4 cleverest
q_1 k_1 v_1
q_2 k_2 v_2
q_3 k_3 v_3
q_4 k_4 v_4
d_k
softmax $\left(\left[q_4 k_1, q_4 k_2, q_4 k_3, q_4 k_4 \right] / \sqrt{d_k} \right)$
$= a_{41} a_{42} a_{43} a_{44}$
$a_{41} v_1 + a_{42} v_2 + a_{43} v_3 + a_{44} v_4 = z_4$

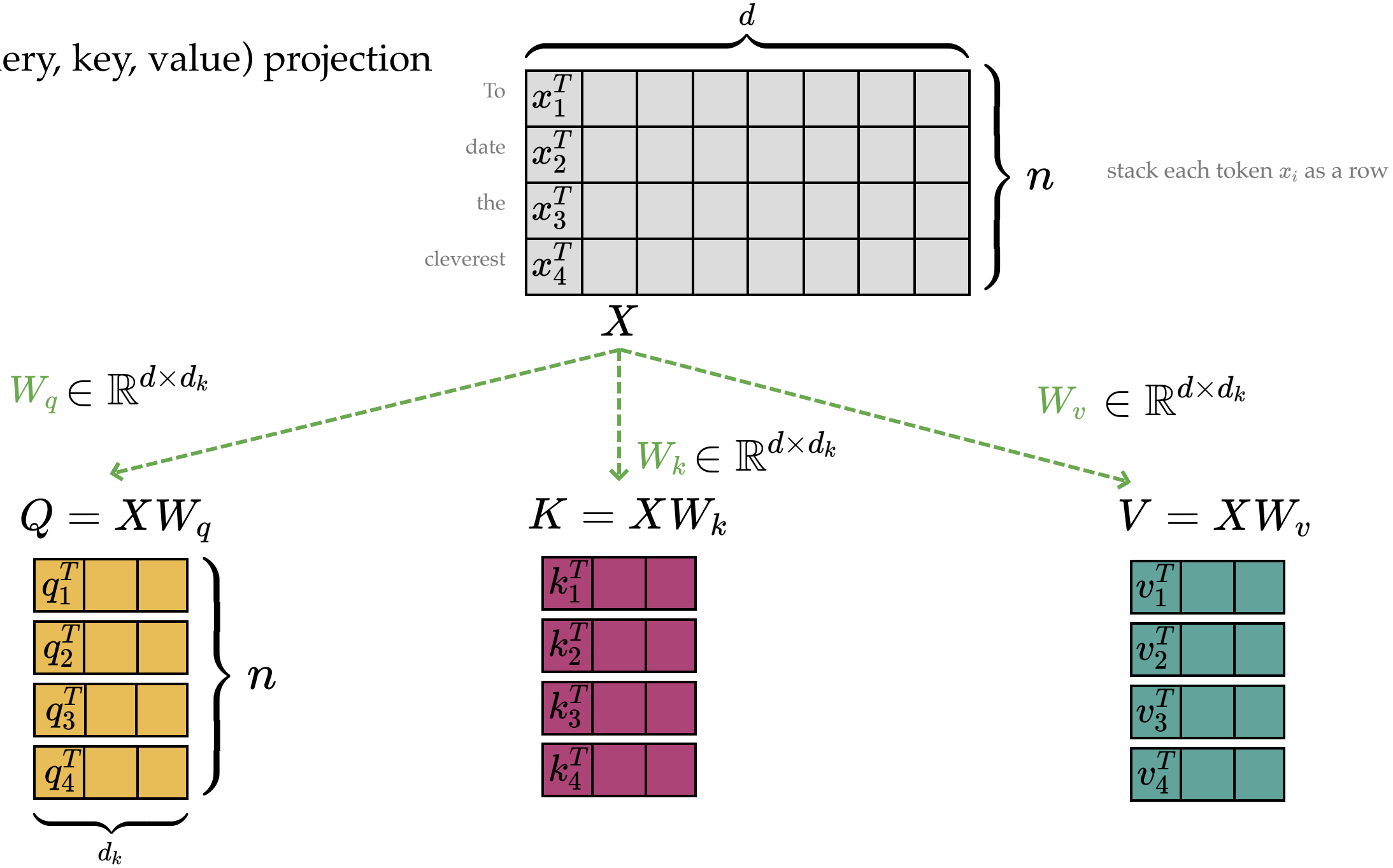
Attention head



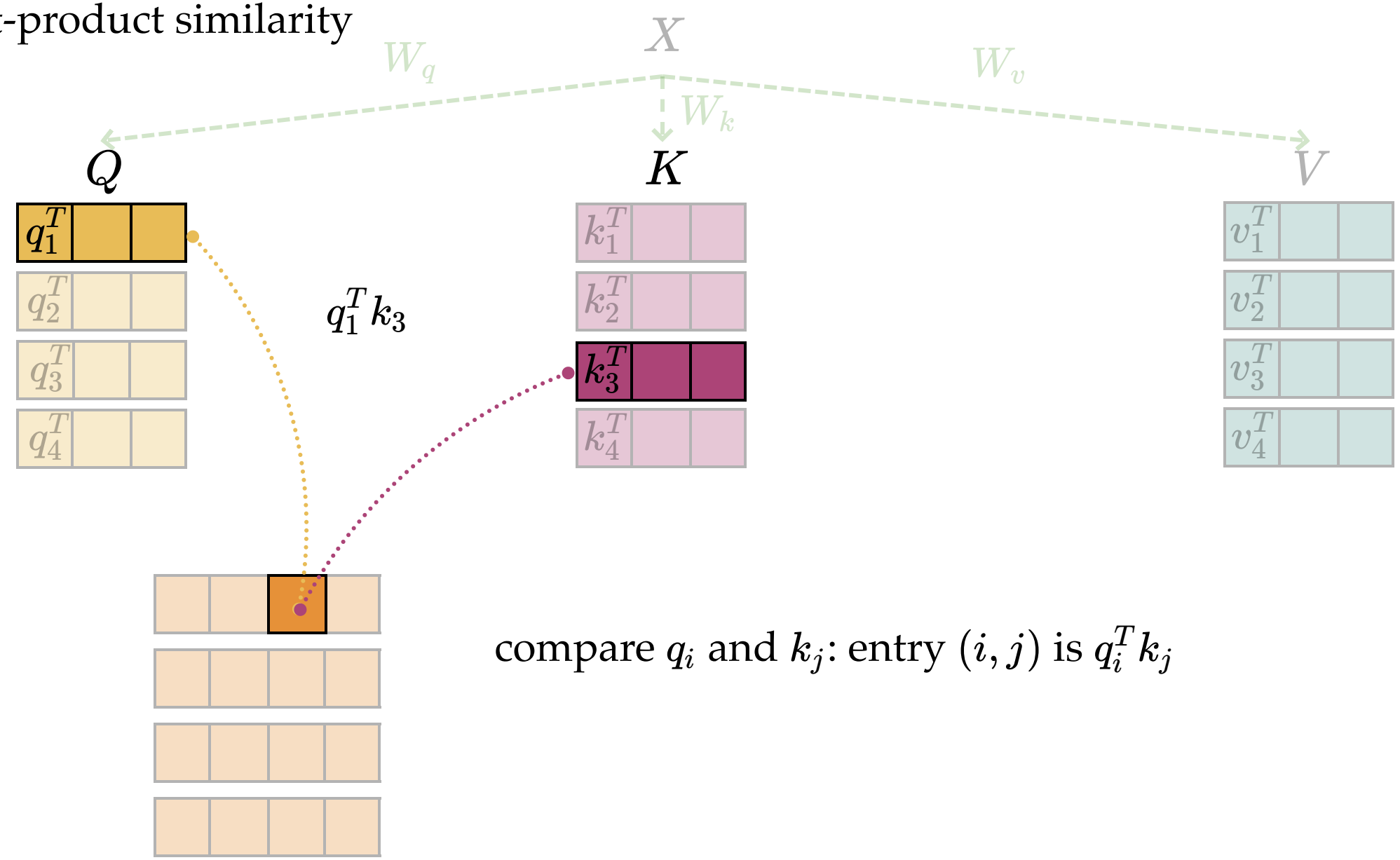
a head maps the sequence of x's to a sequence of context-aware z's, in two steps:

1. ***learned*** (query, key, value) projection
2. attention mechanism

1. (query, key, value) projection

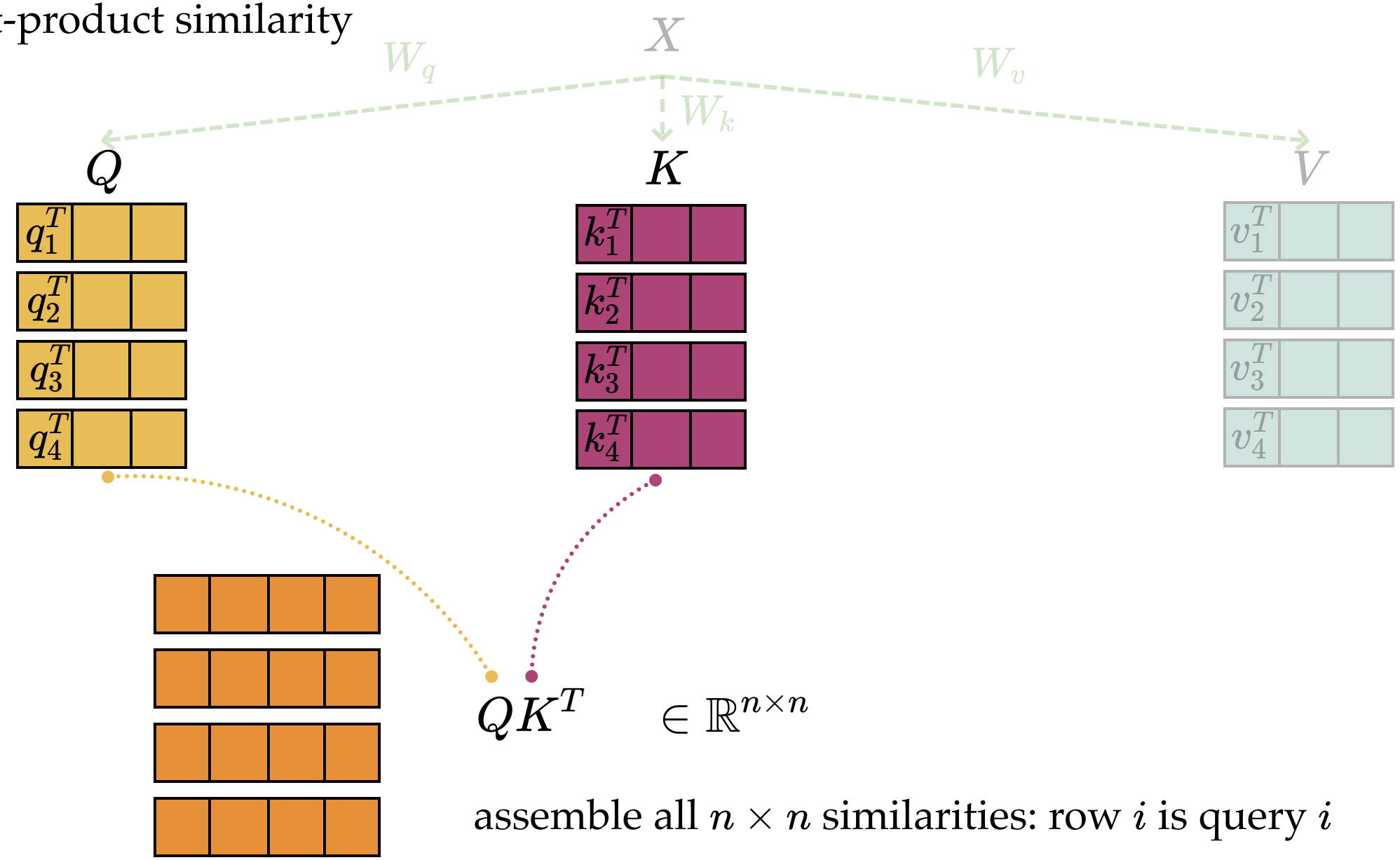


2a. dot-product similarity



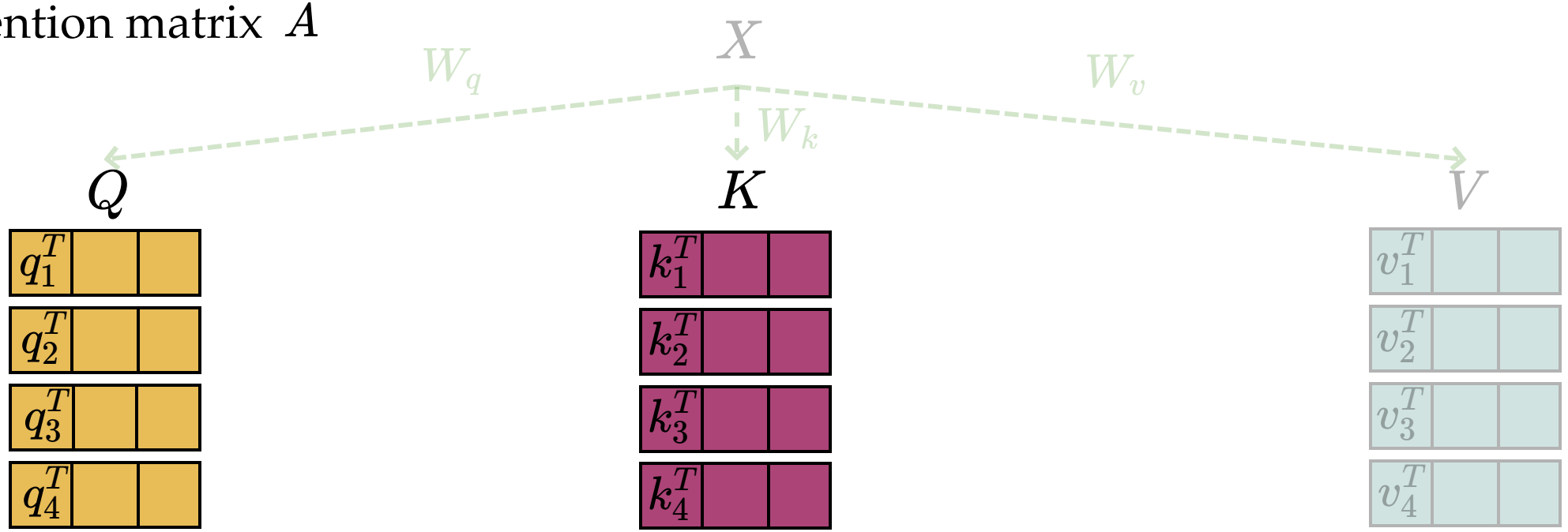
compare q_i and k_j: entry (i, j) is $q_i^T k_j$

2a. dot-product similarity



assemble all $n \times n$ similarities: row i is query i

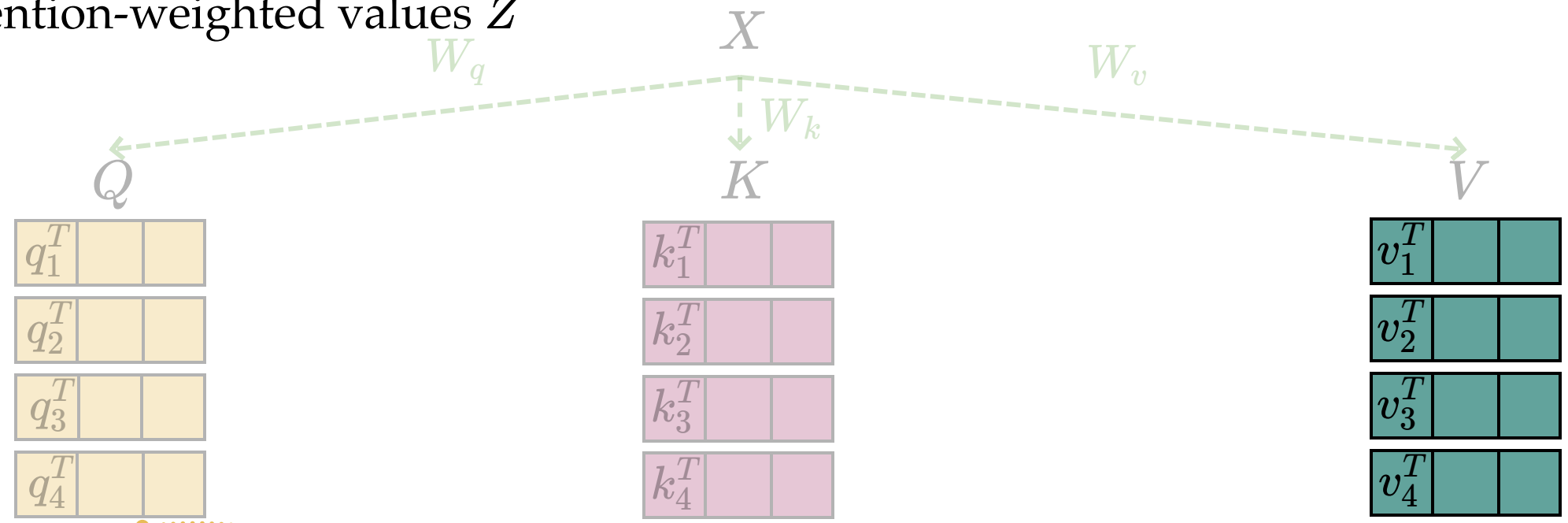
2b. attention matrix A



$$A = \begin{bmatrix} \text{softmax}(\square\square\square\square/\sqrt{d_k}) \\ \text{softmax}(\square\square\square\square/\sqrt{d_k}) \\ \text{softmax}(\square\square\square\square/\sqrt{d_k}) \\ \text{softmax}(\square\square\square\square/\sqrt{d_k}) \end{bmatrix} = \text{softmax}_{\text{row}}(QK^T/\sqrt{d_k}) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

each row sums up to 1

2c. attention-weighted values Z

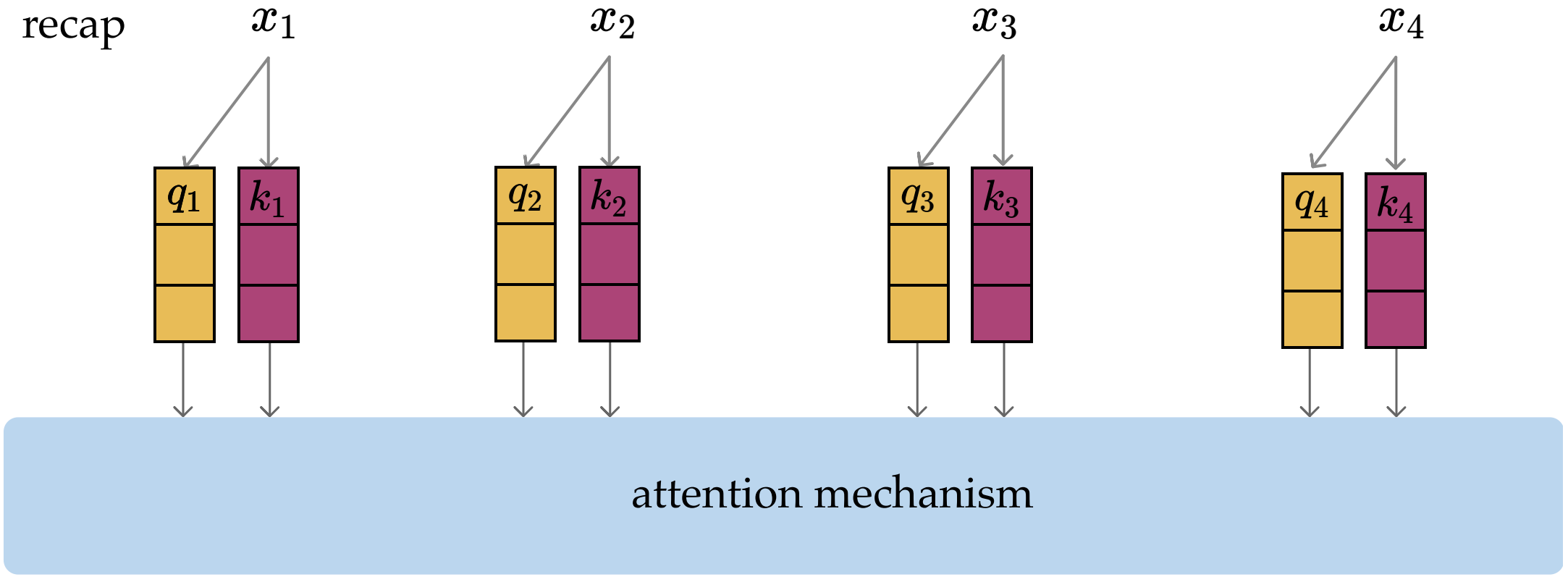


a_{11}	a_{12}	a_{13}	a_{14}
a_{21}	a_{22}	a_{23}	a_{24}
a_{31}	a_{32}	a_{33}	a_{34}
a_{41}	a_{42}	a_{43}	a_{44}

$$A = \text{softmax}_{\text{row}}\left(QK^T/\sqrt{d_k}\right)$$

$$Z = AV$$

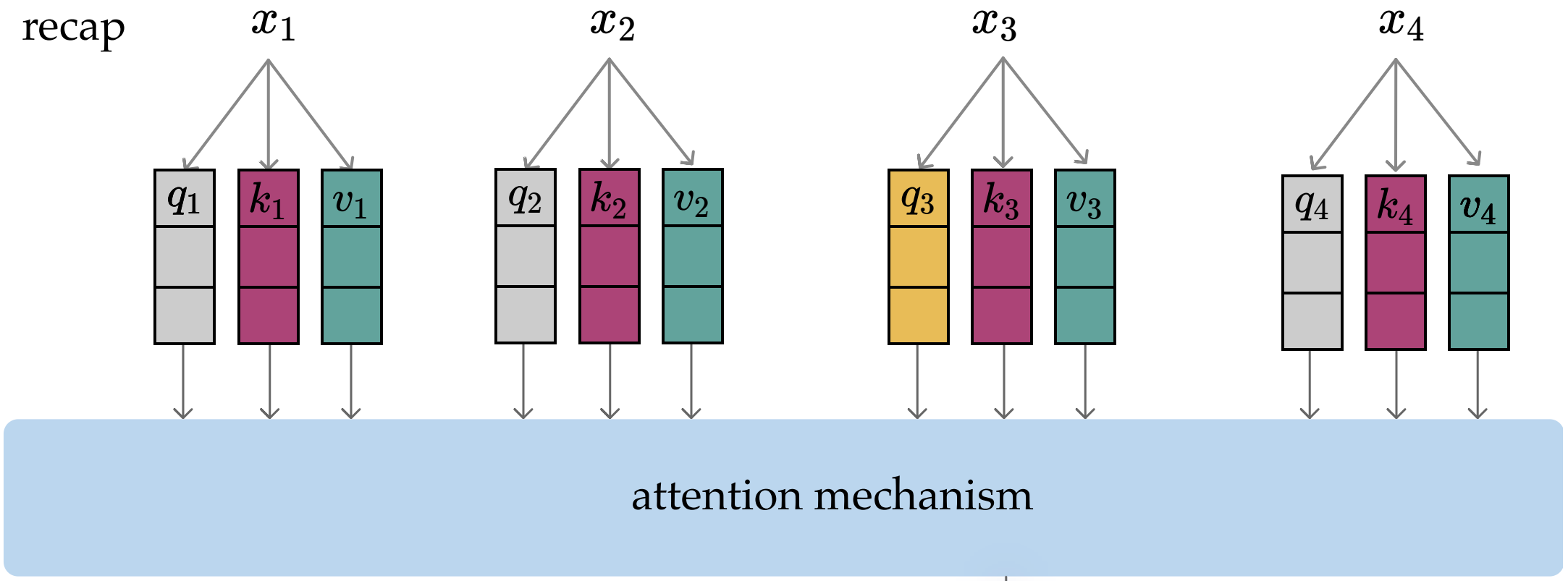
recap



a_{11}	a_{12}	a_{13}	a_{14}
a_{21}	a_{22}	a_{23}	a_{24}
a_{31}	a_{32}	a_{33}	a_{34}
a_{41}	a_{42}	a_{43}	a_{44}

attention scores depend on the (query, key) only

recap



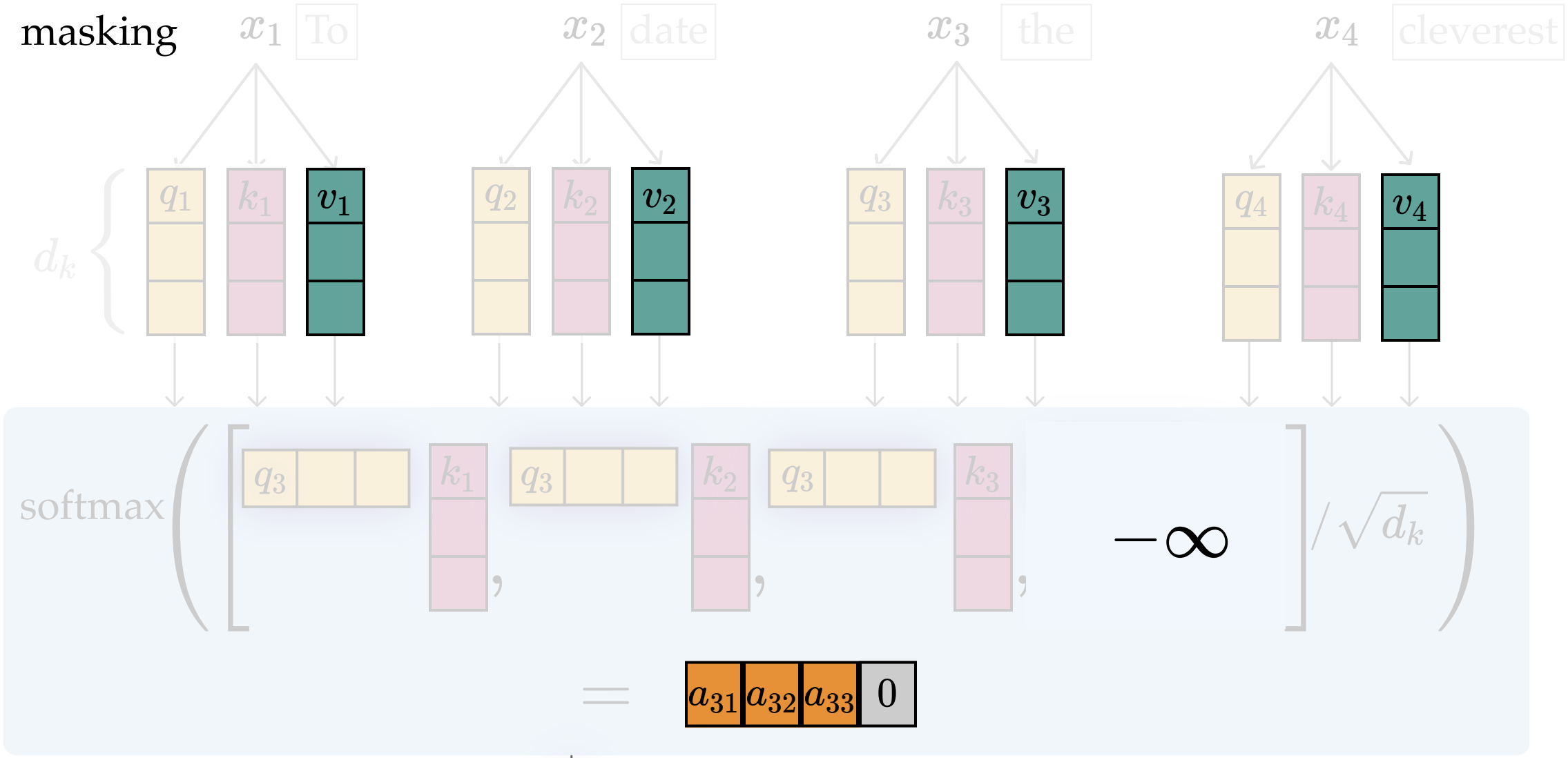
$$a_{31} v_1 + a_{32} v_2 + a_{33} v_3 + a_{34} v_4 = z_3 \in \mathbb{R}^{d_k}$$

a_{11}	a_{12}	a_{13}	a_{14}
a_{21}	a_{22}	a_{23}	a_{24}
a_{31}	a_{32}	a_{33}	a_{34}
a_{41}	a_{42}	a_{43}	a_{44}

z_3 is meant to help predict x_4

but z_3 already uses v_4: it peeks at the answer

masking



mask the future:

$q_3^T k_4 \to -\infty$, so $a_{34} = 0$

masking: no token can peek ahead

dot-product similarity scores

$$\frac{1}{\sqrt{d_k}}\begin{bmatrix} q_1^T k_1 & q_1^T k_2 & q_1^T k_3 & q_1^T k_4 \\ q_2^T k_1 & q_2^T k_2 & q_2^T k_3 & q_2^T k_4 \\ q_3^T k_1 & q_3^T k_2 & q_3^T k_3 & q_3^T k_4 \\ q_4^T k_1 & q_4^T k_2 & q_4^T k_3 & q_4^T k_4 \end{bmatrix} + \begin{bmatrix} 0 & -\infty & -\infty & -\infty \\ 0 & 0 & -\infty & -\infty \\ 0 & 0 & 0 & -\infty \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

causal mask M

$$= \frac{1}{\sqrt{d_k}}\begin{bmatrix} q_1^T k_1 & -\infty & -\infty & -\infty \\ q_2^T k_1 & q_2^T k_2 & -\infty & -\infty \\ q_3^T k_1 & q_3^T k_2 & q_3^T k_3 & -\infty \\ q_4^T k_1 & q_4^T k_2 & q_4^T k_3 & q_4^T k_4 \end{bmatrix}$$

masked scores

$$\xrightarrow{\text{softmax}_{\text{row}}} \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

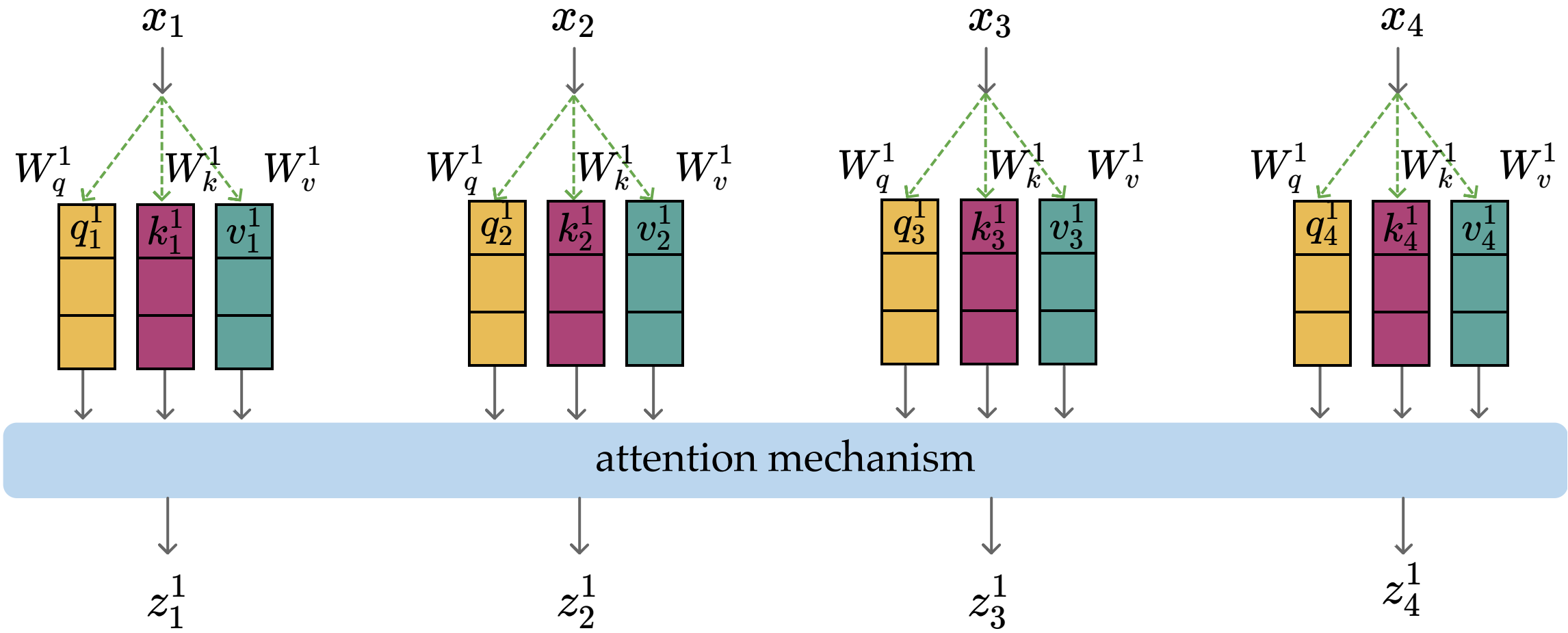
causal attention matrix A

$Z = AV$, same as before

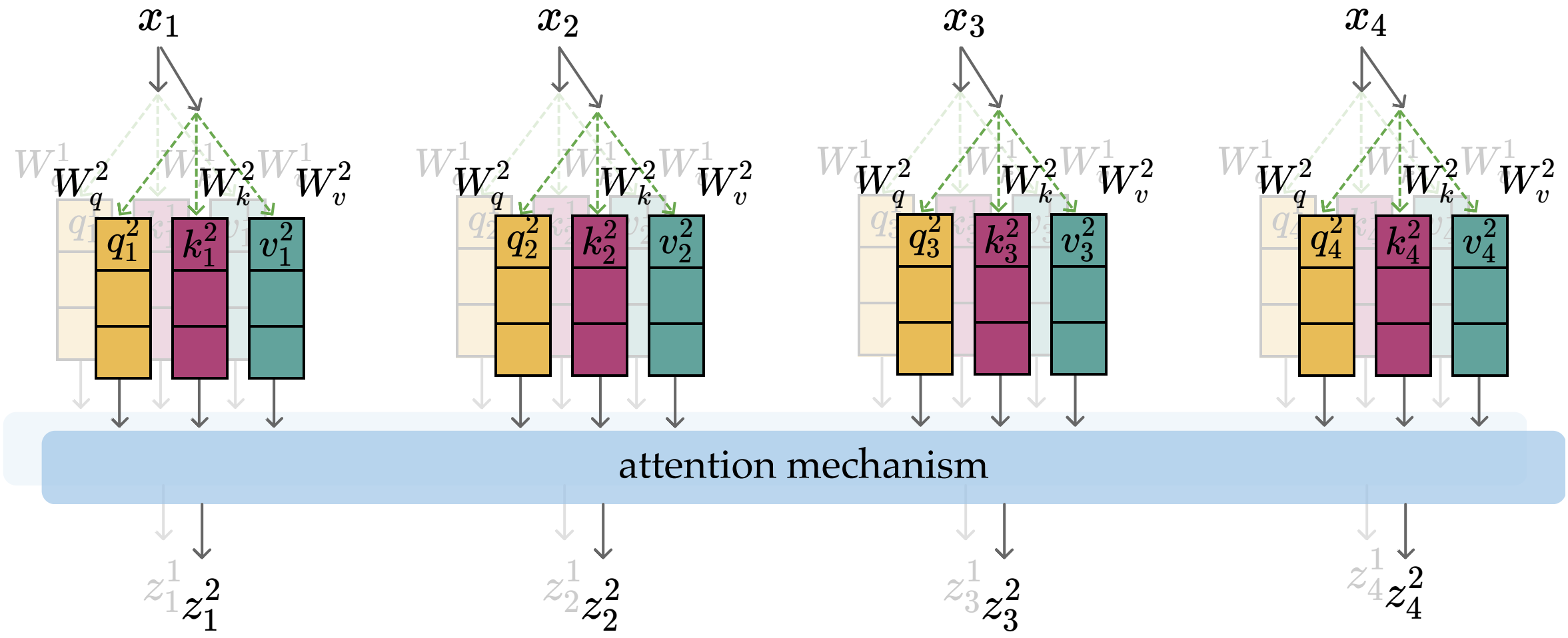
each z_i now mixes only $v_1, \ldots, v_i$

Outline

- Intuition and overview
- Attention mechanism
- Multi-head attention
- Attention beyond language

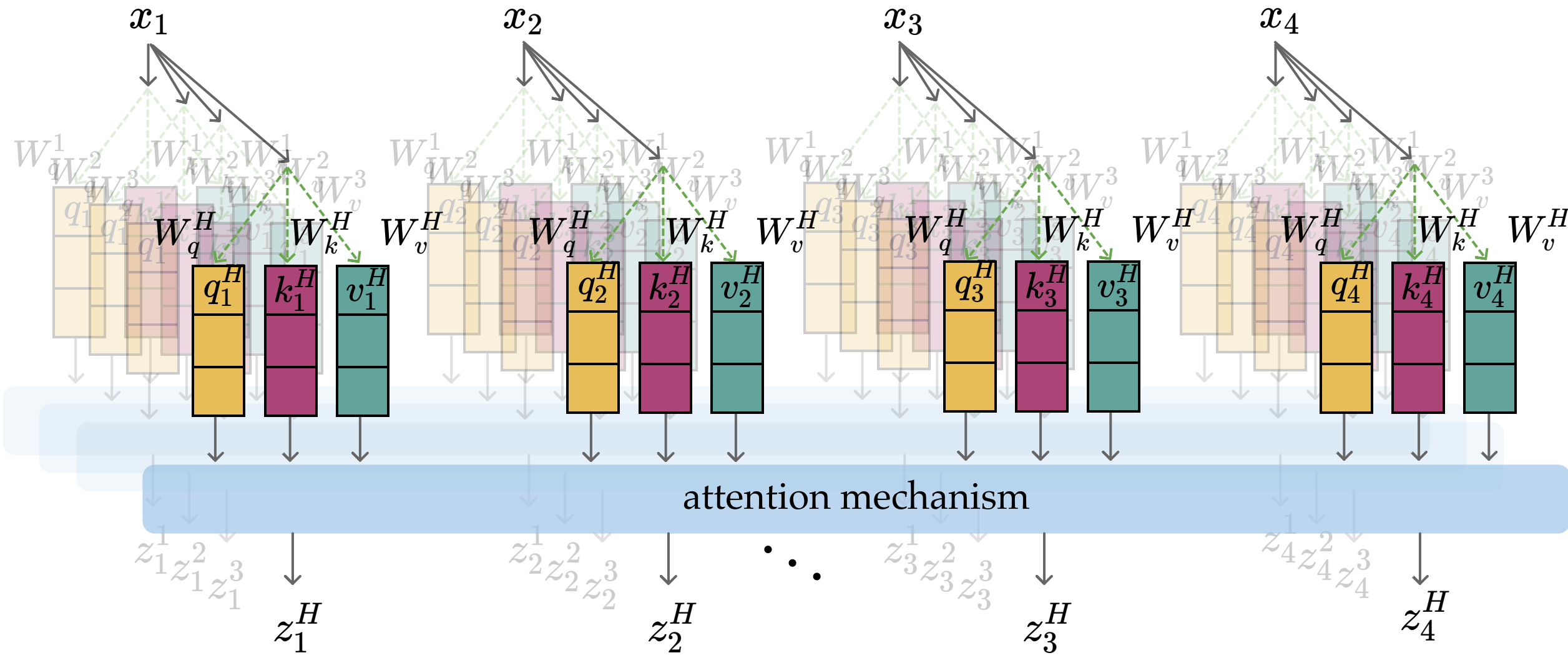


One attention head



head 2 learns its own W_q^2, W_k^2, W_v^2, so it can attend to different things

Two attention heads



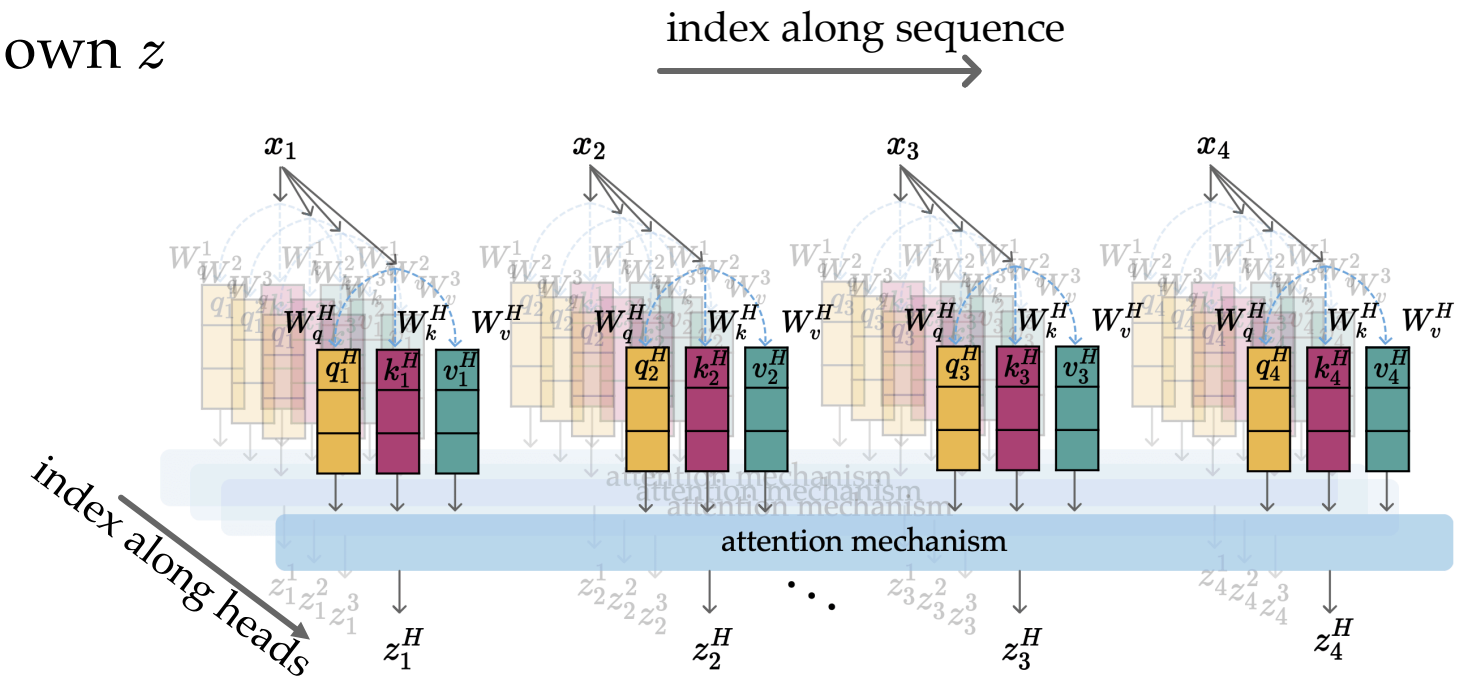
H attention heads

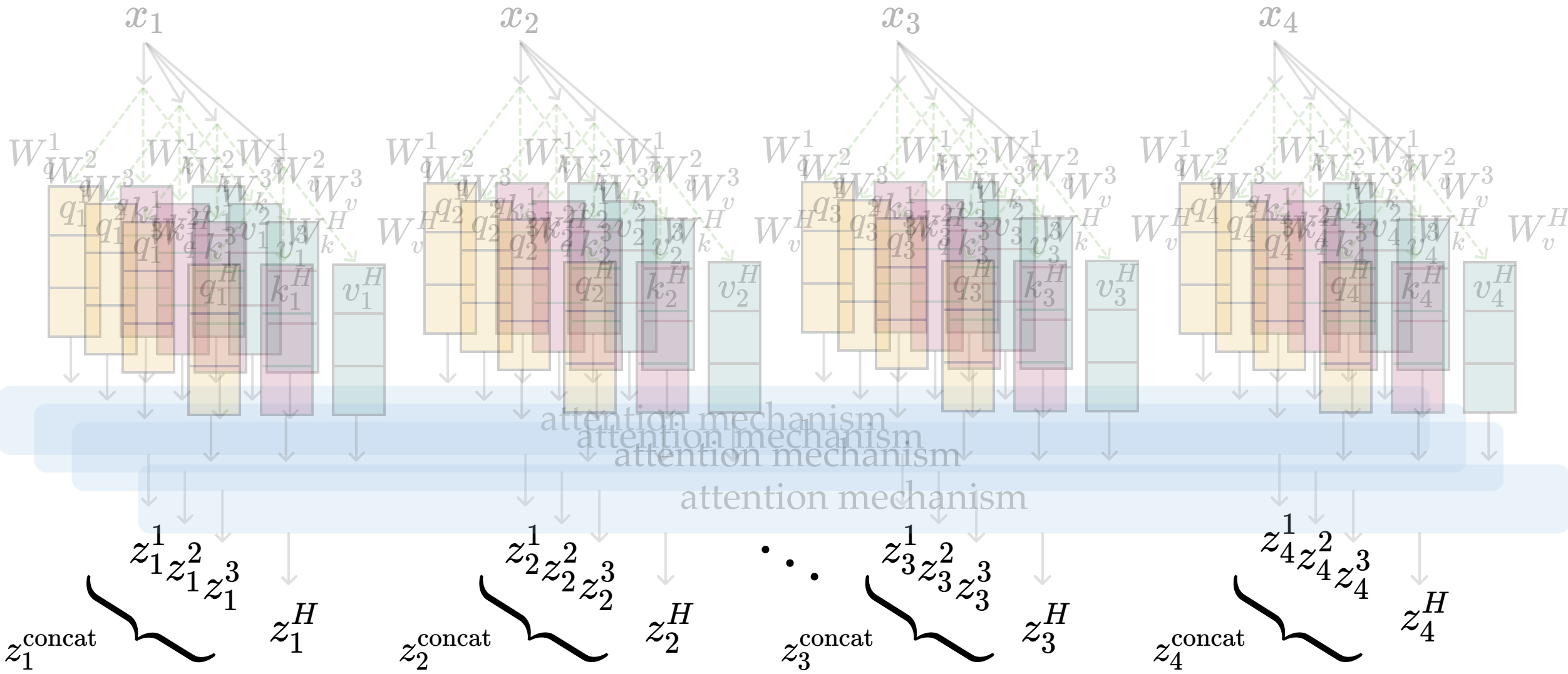
Multi-head Attention

Parallel, and ***structurally identical*** processing across all heads and tokens.

In particular, each head:

- learns its own W_q, W_k, W_v
- creates its own q, k, v
- computes its own z



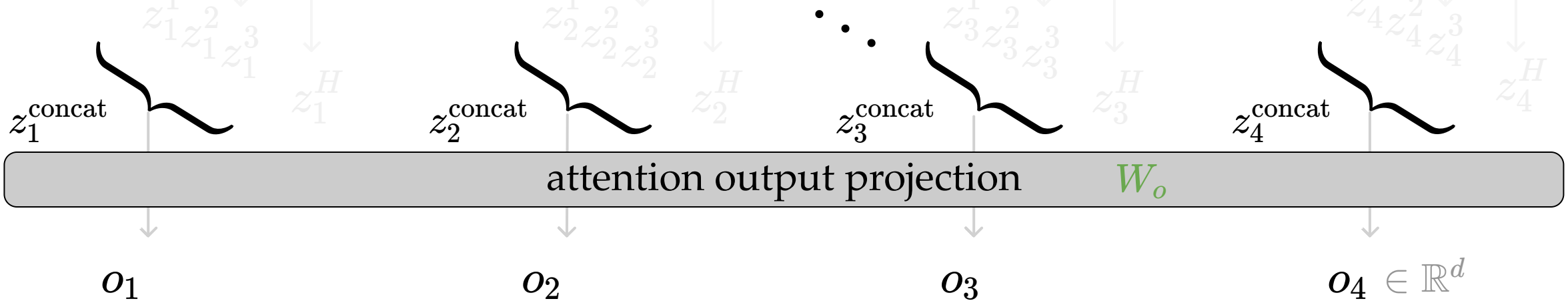


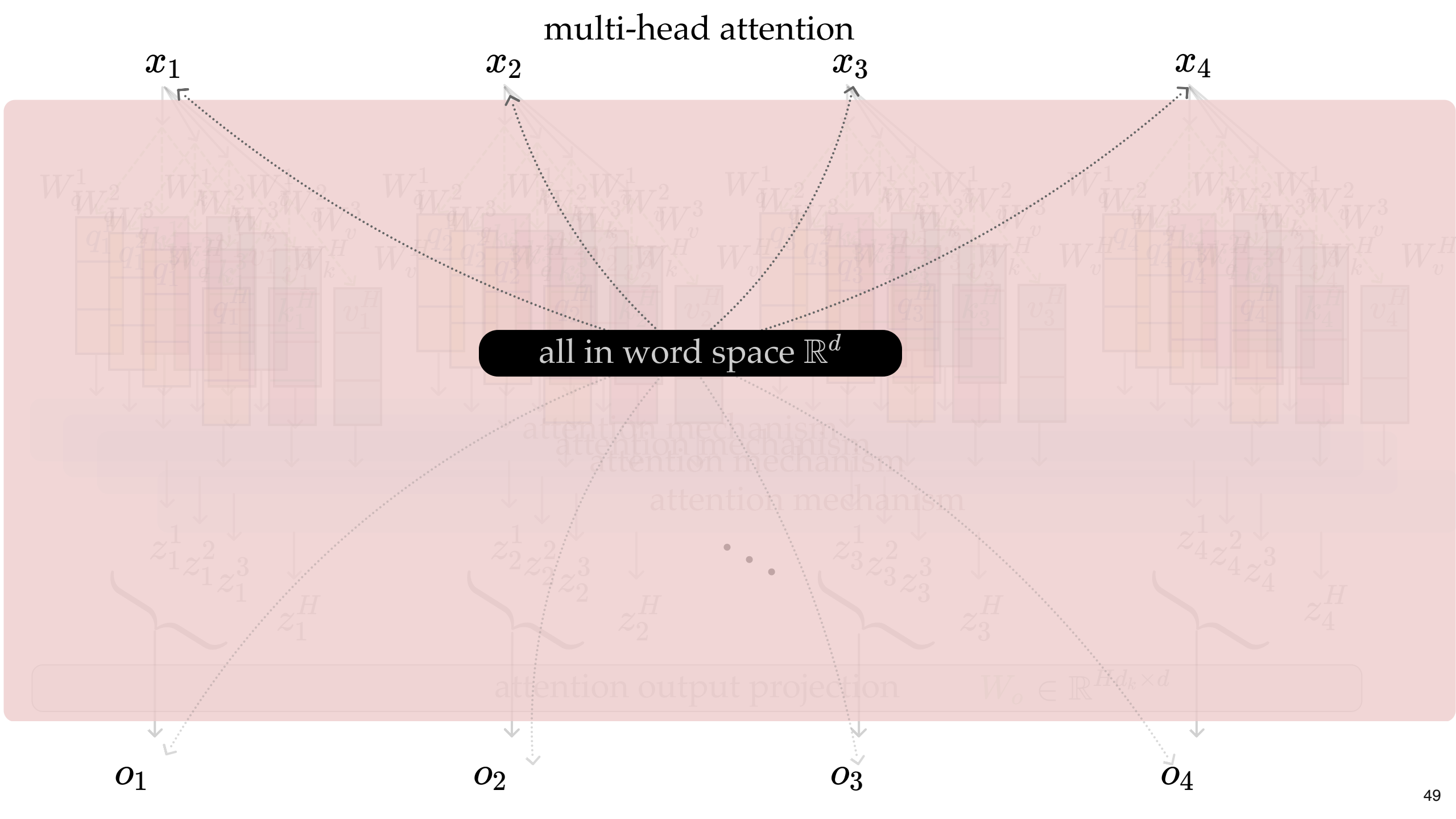
concatenate each token's H head outputs $z_i^h \in \mathbb{R}^{d_k}$: $z_i^{\text{concat}} \in \mathbb{R}^{Hd_k}$

W_o: the learned mixer, blending what the heads found back into the word space $\mathbb{R}^d$

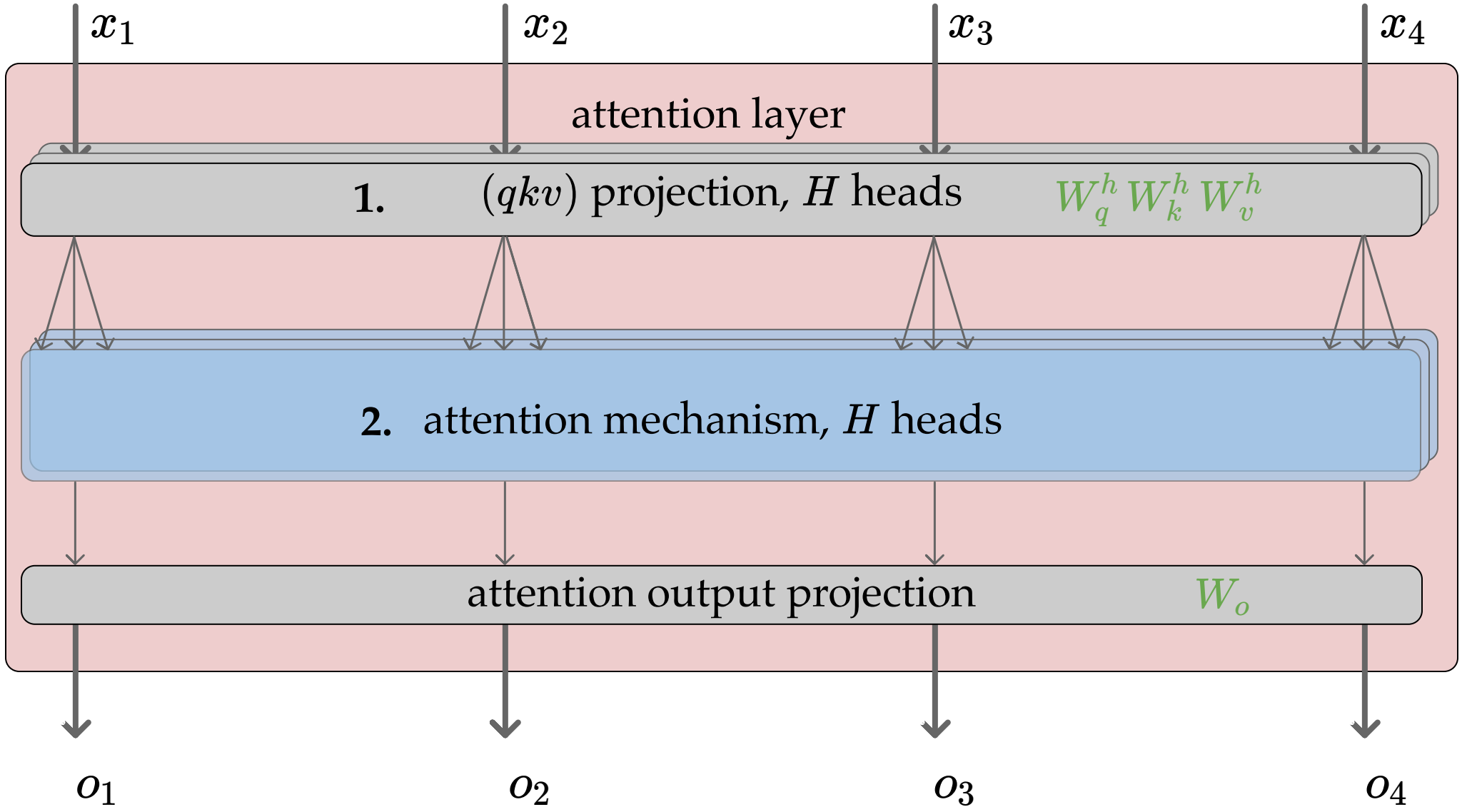
$$\underbrace{Z^{\text{concat}}}_{n \times Hd_k} \underbrace{W_o}_{Hd_k \times d} = \underbrace{O}_{n \times d}$$

typically $H \cdot d_k = d$ (GPT-2: $12 \times 64 = 768$)




multi-head attention
x_1
x_2
x_3
x_4
all in word space $\mathbb{R}^d$
attention mechanism
attention mechanism
attention mechanism
attention output projection
$W_o \in \mathbb{R}^{Hd_k \times d}$
o_1
o_2
o_3
o_4

multi-head attention layer, zoomed out



single-head version: slide 17
zoomed out further, inside a transformer block: slide 16

Shape Example:

n	num tokens	5
d	word-space dim	6
d_k	working-space dim	3
H	num heads	2

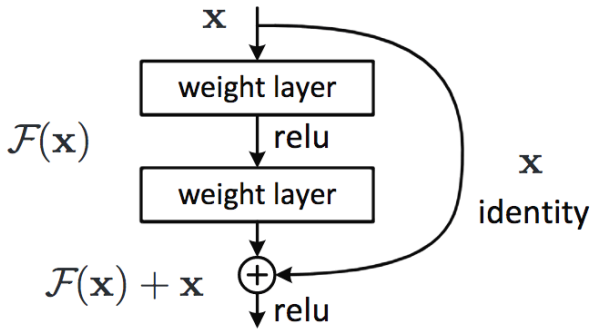
for a single attention head

X	**input**	$n \times d$	5×6
W_q^h	**query proj**	$d \times d_k$	6×3
W_k^h	**key proj**	$d \times d_k$	6×3
W_v^h	**value proj**	$d \times d_k$	6×3
Q^h	**query**	$n \times d_k$	5×3
K^h	**key**	$n \times d_k$	5×3
V^h	**value**	$n \times d_k$	5×3
A^h	**attn matrix**	$n \times n$	5×5
Z^h	**attn head out**	$n \times d_k$	5×3
Z^{concat}	**concatenated heads**	$n \times Hd_k$	5×6
W_o	**output proj**	$Hd_k \times d$	6×6
O	**attn layer out**	$n \times d$	5×6

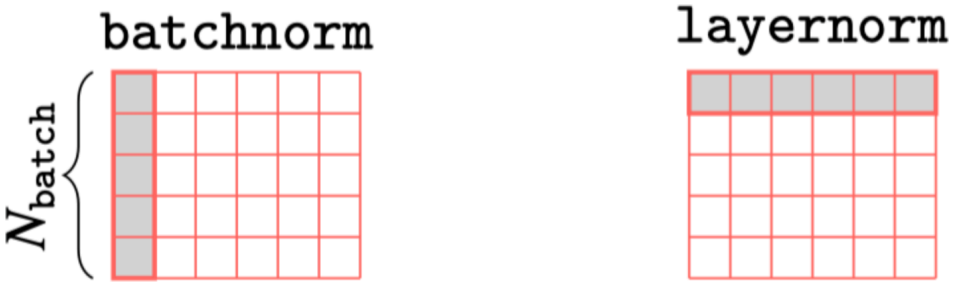
Ws are the learned weights

Practical techniques

Residual connection



Layer normalization



Positional encoding

attention alone ignores token order; positional encoding adds each token's position into x_i

we'll work through these in the lab
Explore interactively: Transformer Explainer

Outline

- Intuition and overview
- Attention mechanism
- Multi-head attention
- Attention beyond language

Generative Boba



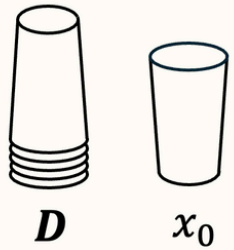
By Boyuan Chen
boyuanc@mit.edu

Intro:

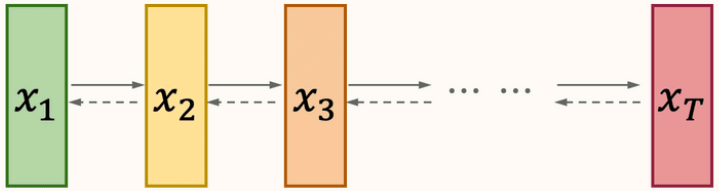
Boba tea is an essential component of Asian students' well-being and productivity.
Prior studies [1] have suggested the lack of accessible boba shop hurts MIT's reputation as a top CS school.
To address this problem, we open a novel boba shop at MIT - Generative Boba.
By providing PhD students with free boba every afternoon, Generative Boba boosts the research productivity of the floor by TBD%.
Generative Boba is also looking for Chief Boba Scientists to scale up! Contact me to be a co-author.

[1] Chen, B. (2022, June 20). *BEST COMPUTER SCIENCE SCHOOLS RANKED BY BOBA*. Boyuan's Blog. Retrieved March 27, 2024,.

Step 1:
Sample a cup from cup set



Step 2:
Gradually add t steps of ingredients



Step 3:
Graduate Student Descent





Fig 1: We propose boba diffusion, a novel generative model that boosts PhD student productivity

Today's special:

$\varepsilon_1 = topping$



20%

open boba pot for topping
close pot lid to keep warm

$\varepsilon_2 = milk$



30-60%

we serve lactose-free milk,
sometimes coconut milk

$\varepsilon_3 = tea$



20-50%

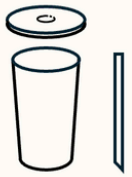
use 20% for matcha
use 50% for black tea

$\varepsilon_4 = syrup$



any

hold black sugar bottle
upside down for 5-10s

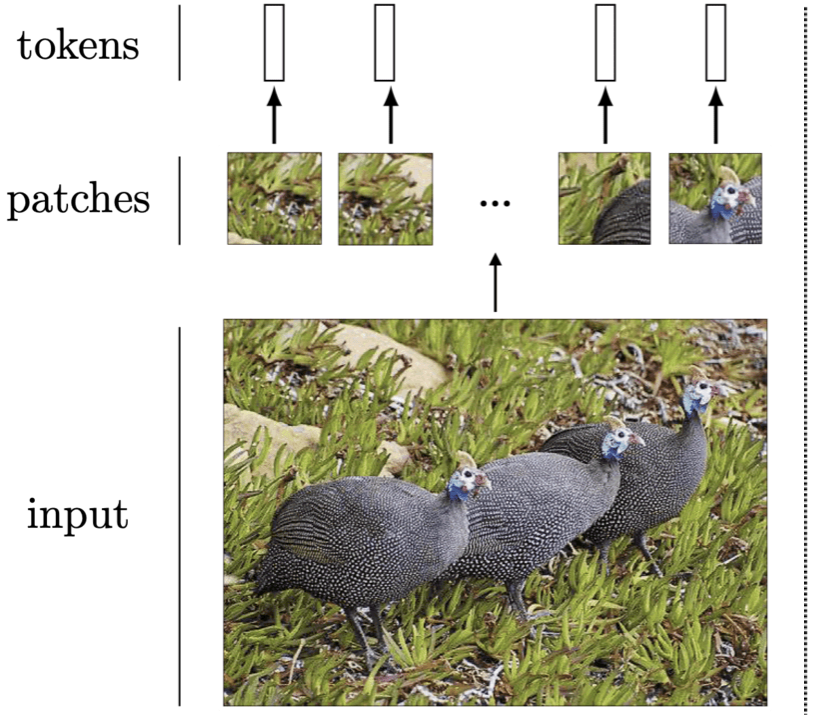


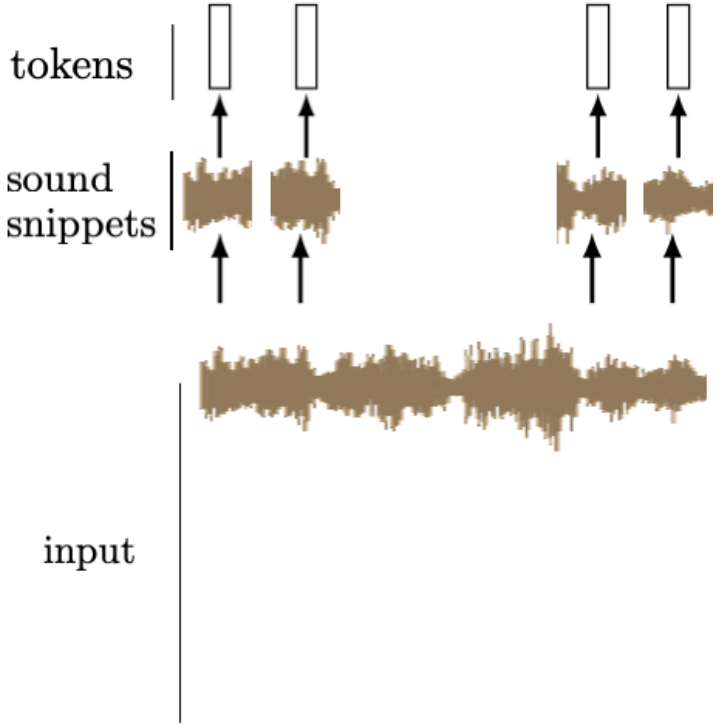
add a lid and straw
our straws are compostable

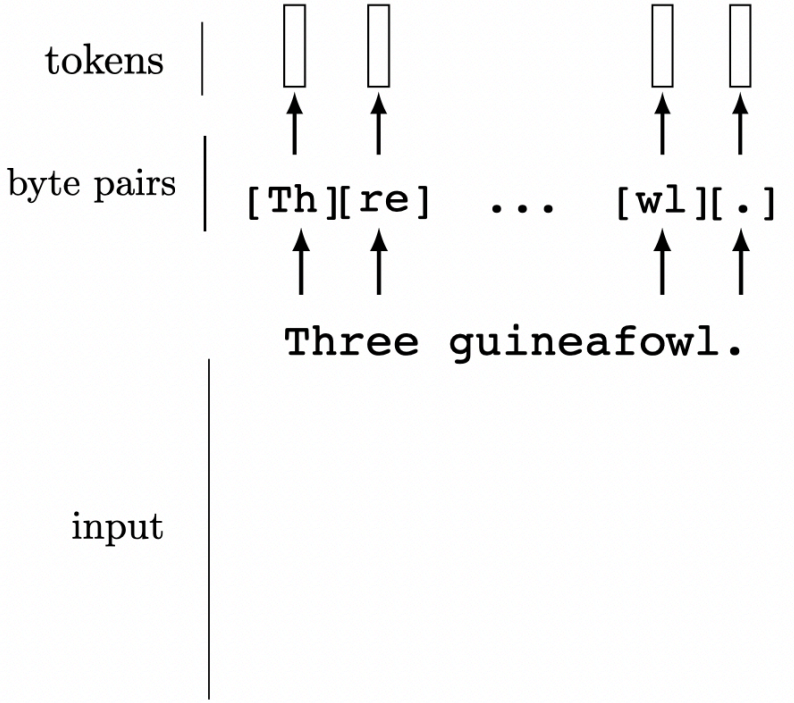
Fig 2: Implementation details of boba diffusion. We follow an efficient ingredient schedule while "today's special" provides special recipes from time to time.

We can tokenize anything.

General strategy: chop the input up into chunks, *project* each chunk to an **embedding**



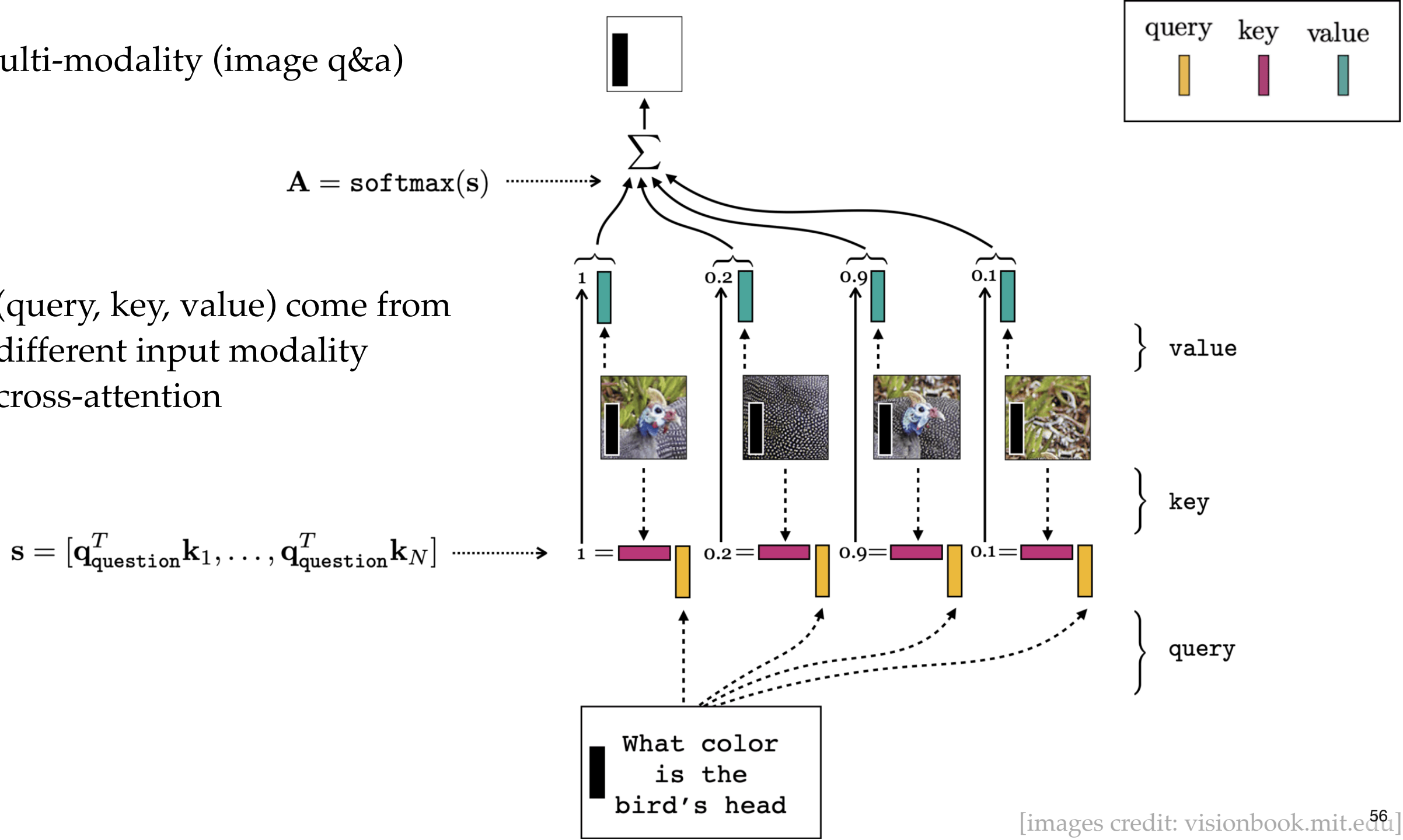




[images credit: visionbook.mit.edu]

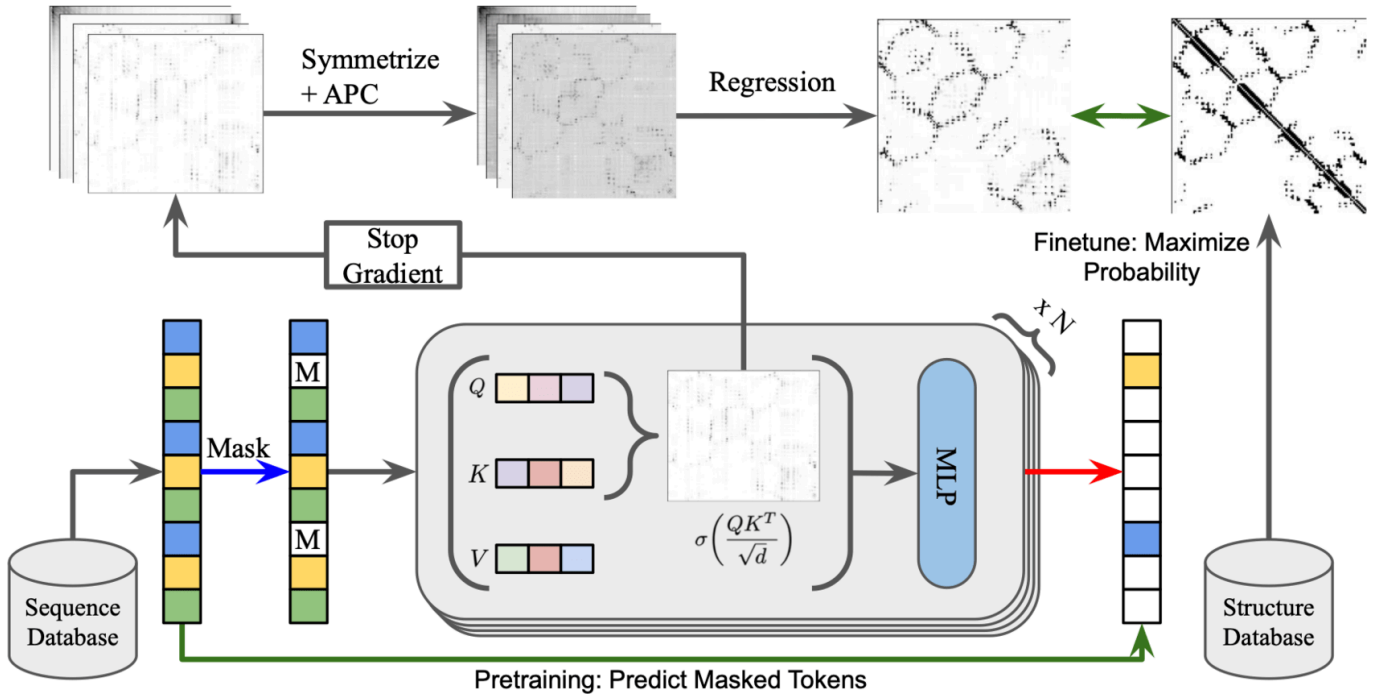
Multi-modality (image q&a)

- (query, key, value) come from different input modality
- cross-attention



[images credit: visionbook.mit.edu]

Biology: attention learns protein structure



- large $a_{ij} \approx$ amino acids i and j touch once the protein folds; no one taught it folding
- why it matters: a protein's 3D shape decides what it does (e.g., which drugs bind); sequences are cheap to read, shapes take months to measure in a lab
- ESMFold and AlphaFold predict the full 3D shape from sequence

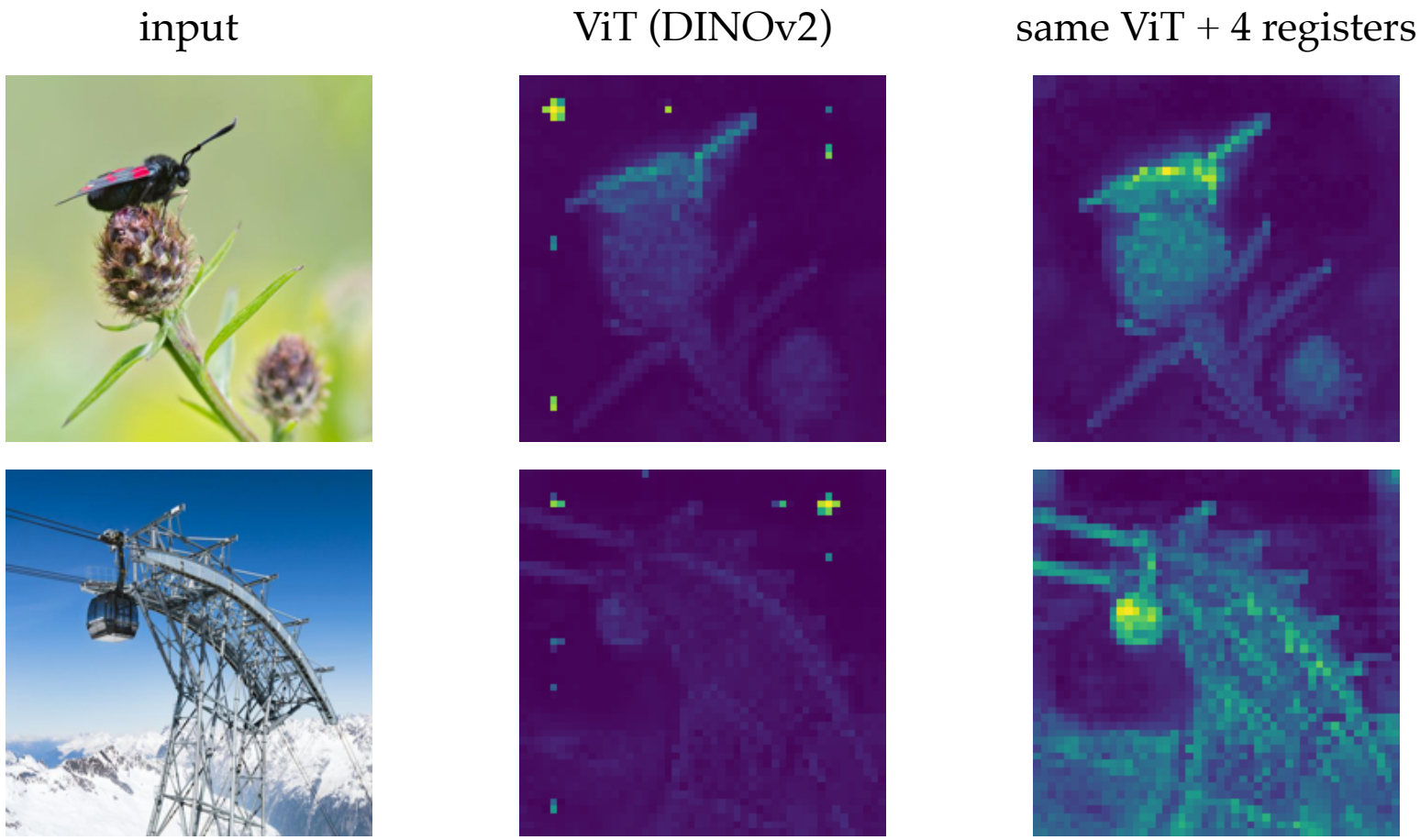
[Rao et al., ICLR 2021 · Fig. 1] [Lin et al., Science 2023]

Attention map highlights objects (DINO)



[Caron et al., Emerging Properties in Self-Supervised Vision Transformers, ICCV 2021]

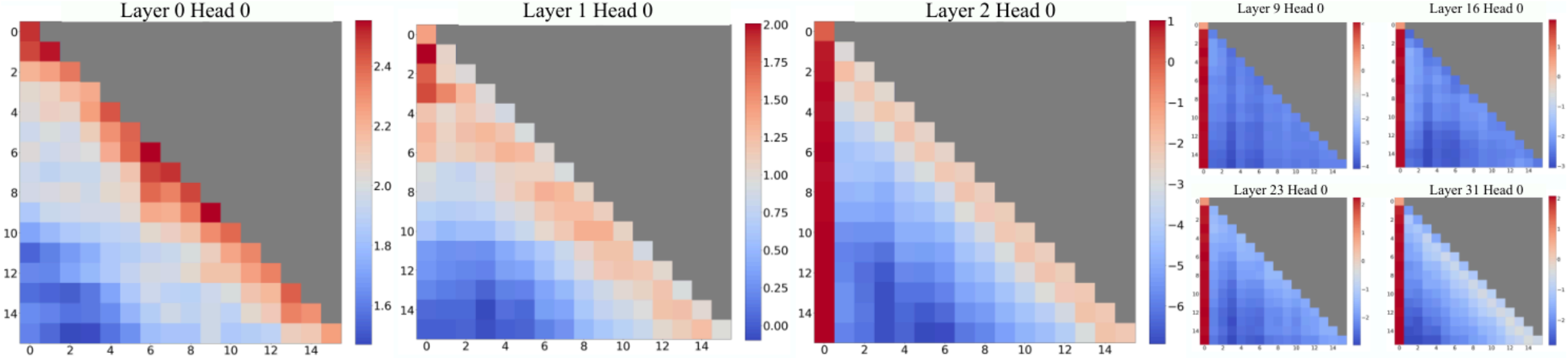
Attention maps can have artifacts



a few background patches get hijacked as scratch memory (bright dots, middle column)
blank "register" tokens give the model scratch space instead, and the maps clean up (right column)

[Darcet et al., Vision Transformers Need Registers, ICLR 2024]

Attention sinks: a big attention score isn't always meaningful

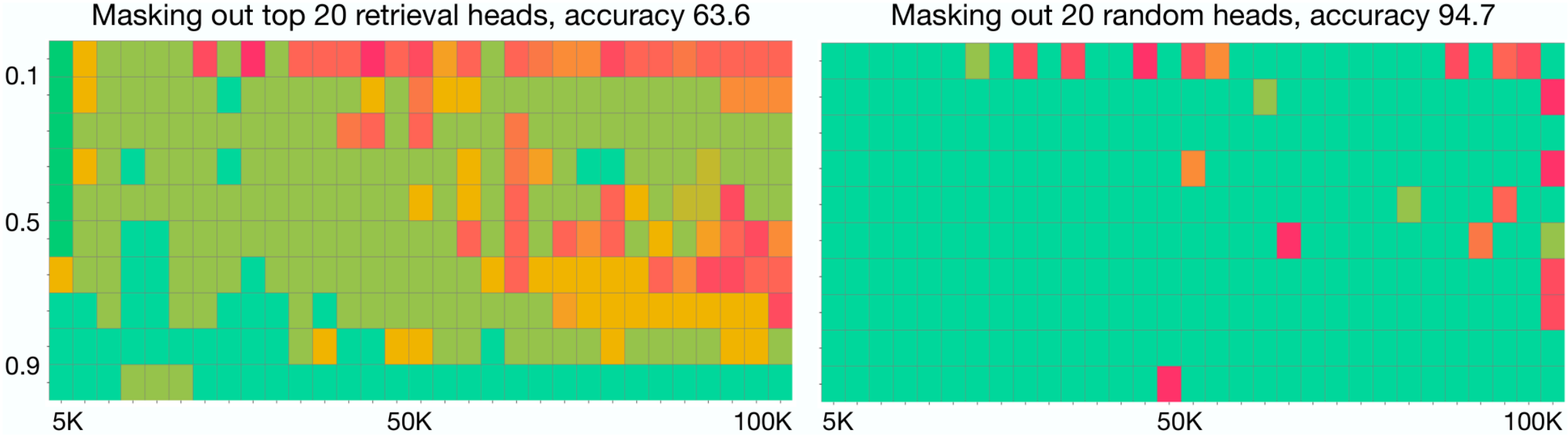


- average attention logits in Llama-2 (grey: masked future)
- each row must hand out 100%; every row can see the first token, so heads park leftover attention there
- so a big a_{i1} is a parking spot, not importance (like the registers)
- emerged on its own; later models add a dedicated sink token on purpose

[Xiao et al., Attention Sinks, ICLR 2024] [Jain & Wallace, Attention is not Explanation, NAACL 2019]

Does a head matter? Disable it and check

task: retrieve one fact hidden in a long document

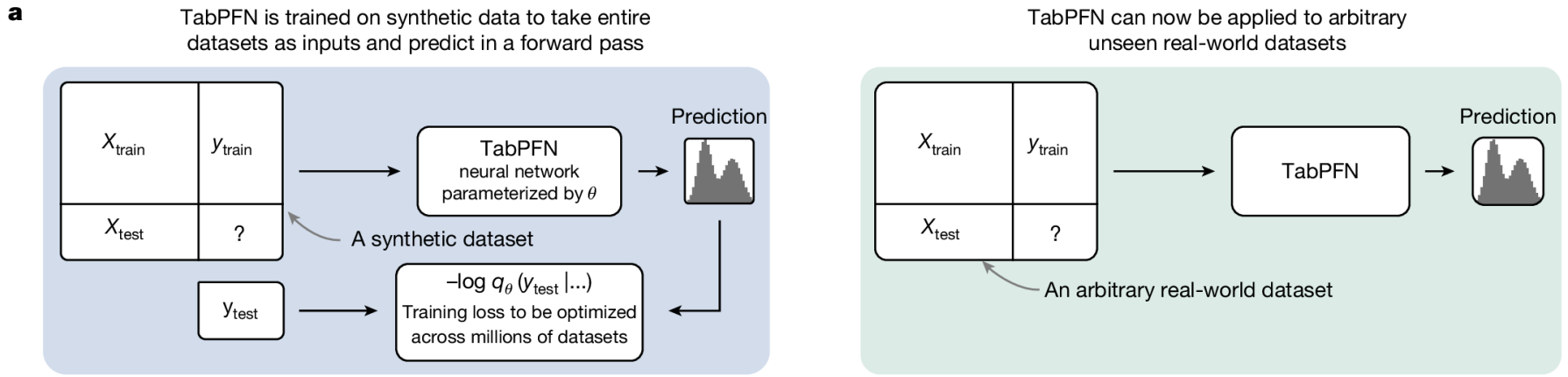


x: document length (5K–100K tokens) · y: where the fact sits (0.1 = start, 0.9 = end) · green = retrieved correctly

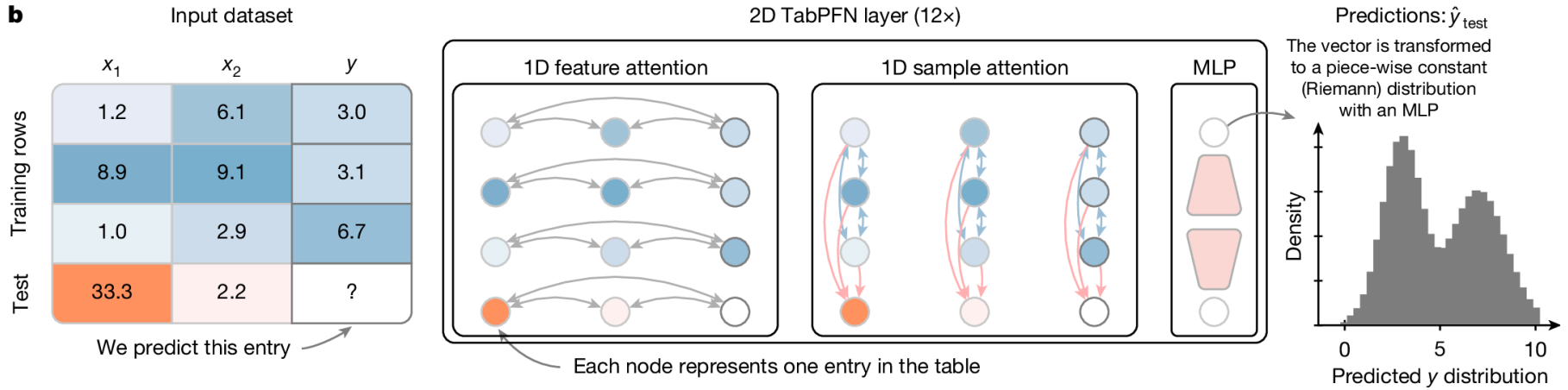
Removing retrieval heads hurts more than removing random heads.

[Wu et al., Retrieval Head Mechanistically Explains Long-Context Factuality, ICLR 2025 · Fig. 1]

TabPFN: no training on your table; it attends to it at prediction time



trained once on millions of synthetic tables; on a new table: no training, just one forward pass



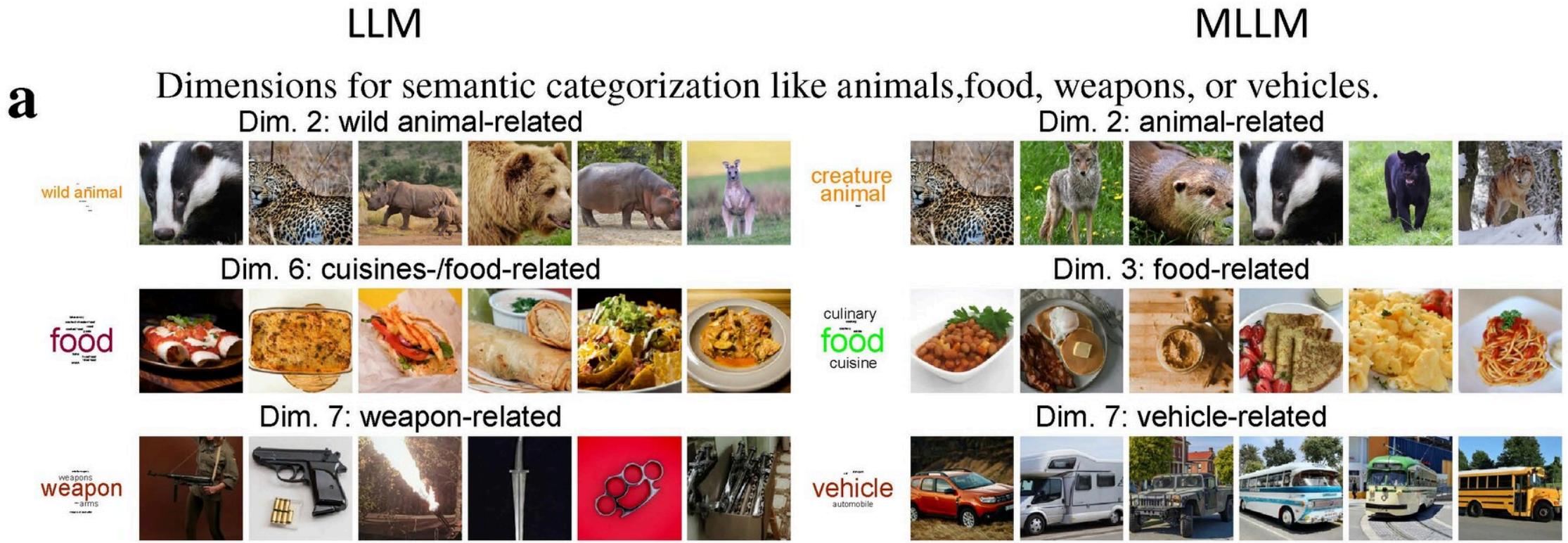
test row = query, training rows = keys, their labels = values: the soft dictionary lookup from slide 3

[Hollmann et al., Accurate predictions on small data with a tabular foundation model, Nature 2025 · Fig. 1]

Summary

- Attention creates context-aware representations: learned W_q, W_k, W_v project each token into a query, key, and value; queries and keys set the mixing percentages, values supply what gets combined.
- Multiple heads each learn their own W_q, W_k, W_v and working space $\mathbb{R}^{d_k}$; W_o maps the concatenated outputs back to word space $\mathbb{R}^d$.
- Training processes tokens in parallel: causal masking prevents access to future tokens.
- Attention extends beyond language: the same mechanism supports vision, proteins, and tabular data.

Human-like concepts emerge in LLMs



- from 4.7M odd-one-out judgments, each model gets 66 interpretable concept dimensions: animal, food, vehicle, ...
- these line up with human judgments, and with brain activity in category-selective regions

[Du et al., Human-like object concept representations emerge naturally in multimodal LLMs, Nature Machine Intelligence 2025]