

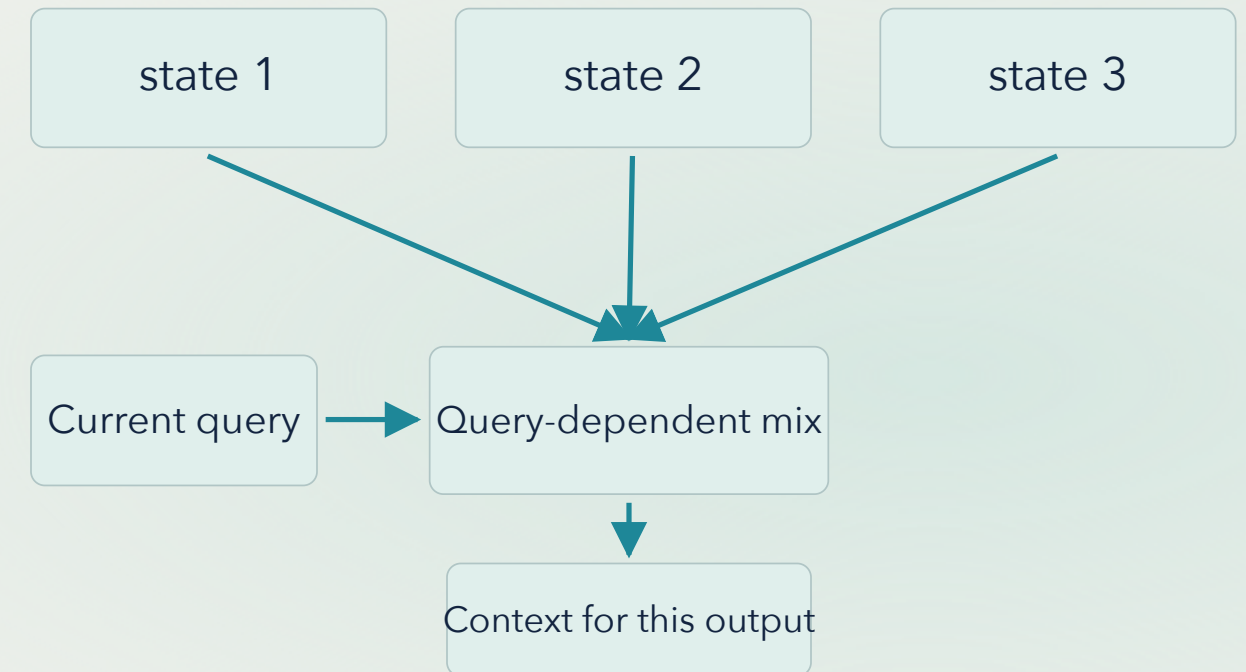
W07 · H2 · 15:00–16:00 · 19 October 2026

Attention: selective retrieval

Why force every output to use the same compressed memory?

Machine Learning in Engineering Science
Fall 2026 · Week 07 · Hour 2
Chi-Hua Jonny Yu · NCKU / LAiMM

ONE MEMORY, DIFFERENT QUESTIONS



This hour: learn to retrieve

Compare a query with keys, then use normalized weights to combine values.

Goal: compute a context vector and explain a causal mask.

00-14 min

Motivation + query / key / value



14-34 min

Scores $\rightarrow$ weights $\rightarrow$ context



34-48 min

Masks + self / cross attention



48-60 min

Notebook checks + summary

Mechanism diagram / illustrative teaching values

Alignment is output-dependent

A translation decoder may emphasize different source positions at different steps.

Rows: queries / receiving positions

0.70	0.20	0.10
0.10	0.70	0.20
0.15	0.15	0.70

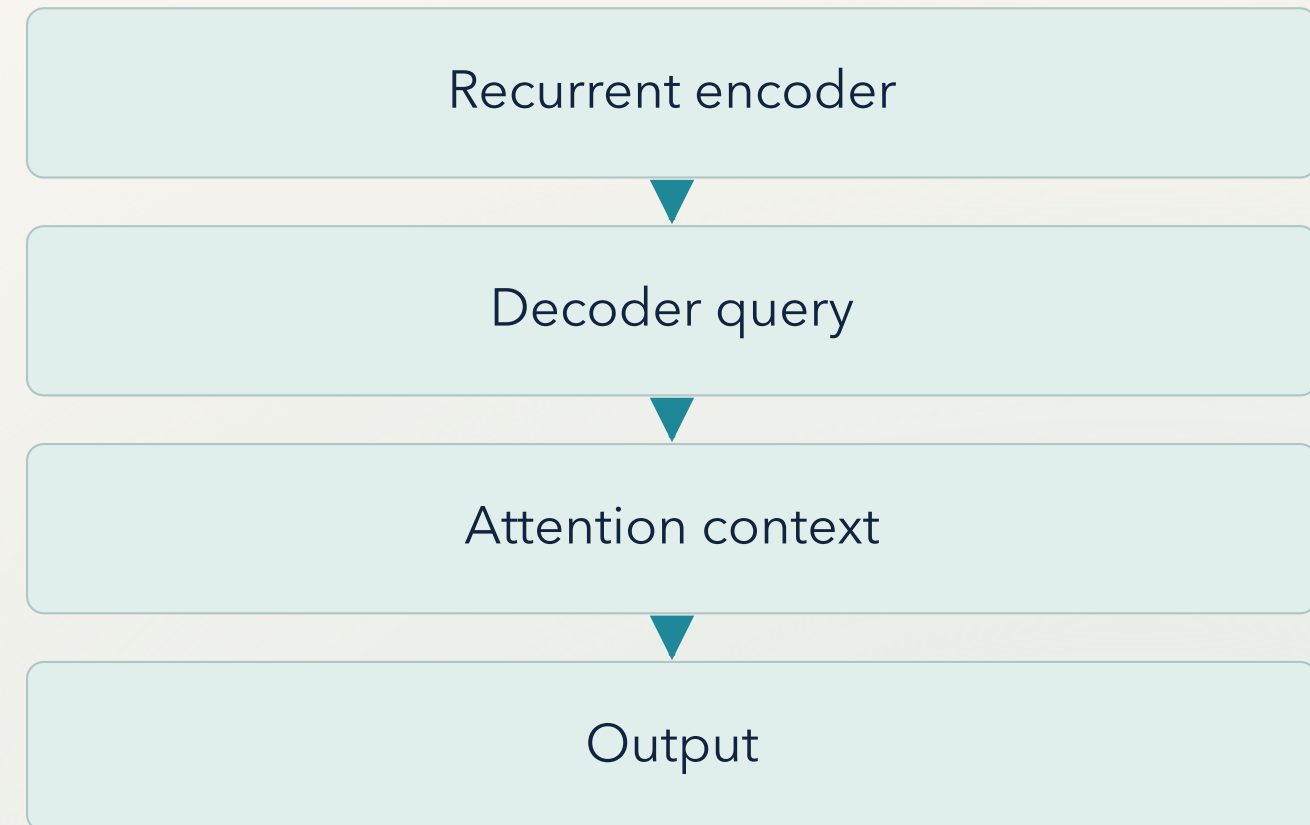
Key / source position →

Mechanism diagram / illustrative teaching values

Attention predates the Transformer

LSTM or GRU encoders can supply states to an attention mechanism.

2014/15: recurrent attention 2017:
Transformer



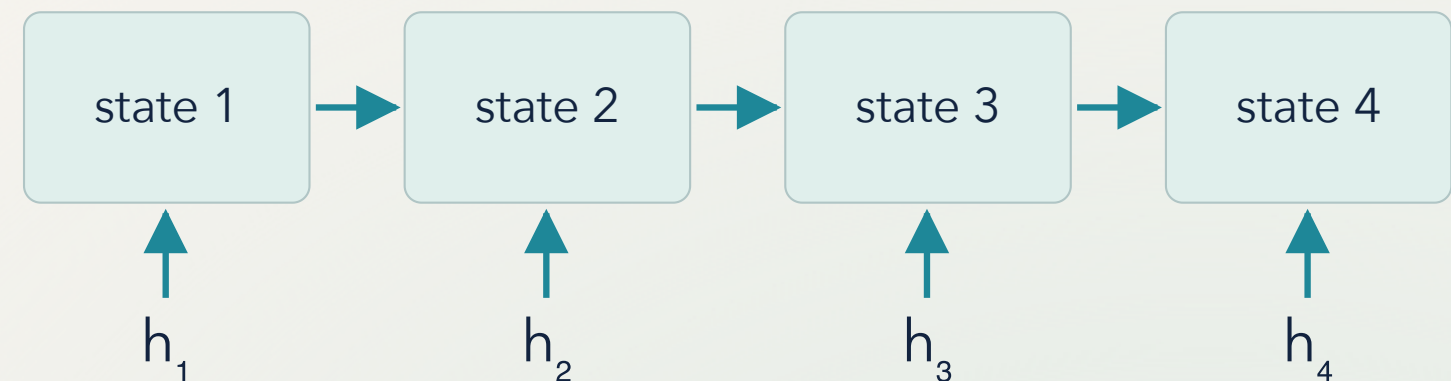
Mechanism diagram / illustrative teaching values

Keep the full set of encoder states

The decoder can retrieve from H instead of using only the final state.

$$H = [h_1; h_2; \dots; h_n]$$

Shared update parameters across time



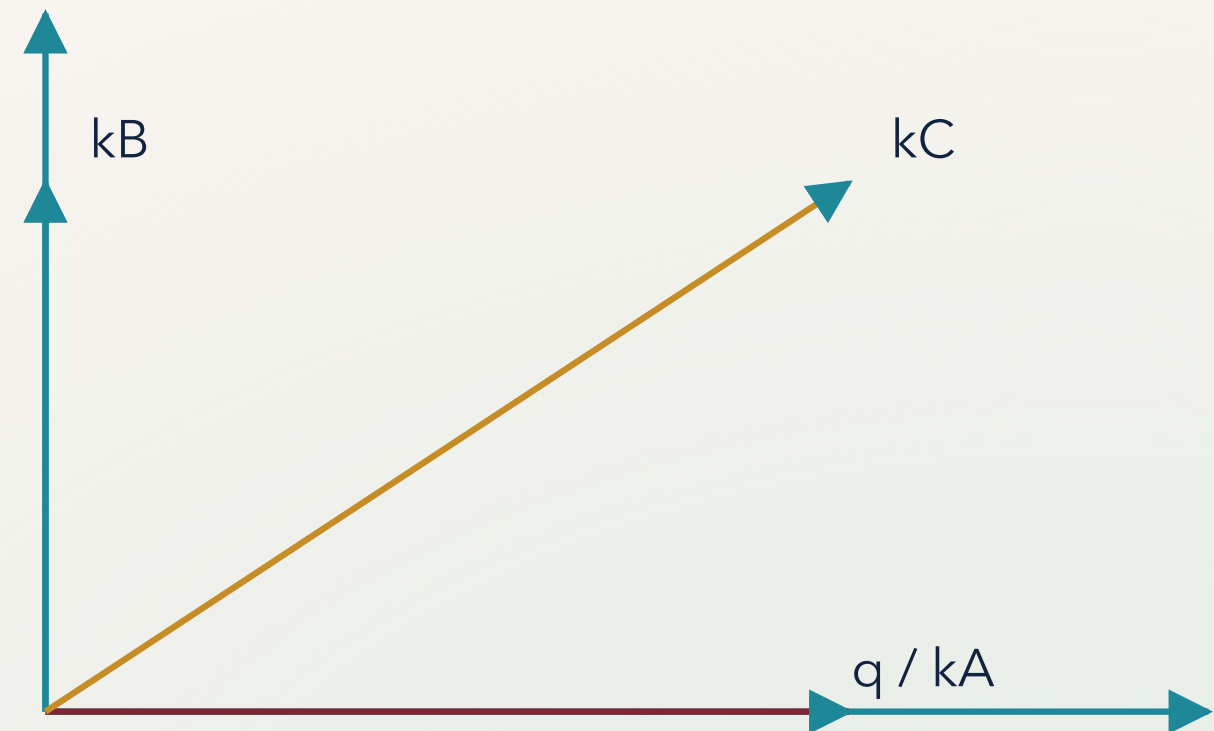
Past → running state → later state

Mechanism diagram / illustrative teaching values

A query specifies what to look for

The query is a vector used to compare the current need with memory keys.

$$q \in \mathbb{R}^{d_k}$$



Chosen vectors for an inspectable example

Mechanism diagram / illustrative teaching values

Keys address; values carry content

Matching uses keys; the retrieved combination uses values.

$$K \in \mathbb{R}^{N \times d_k} \quad V \in \mathbb{R}^{N \times d_v}$$

A: key [1,0] → value [10,0]

B: key [0,1] → value [0,10]

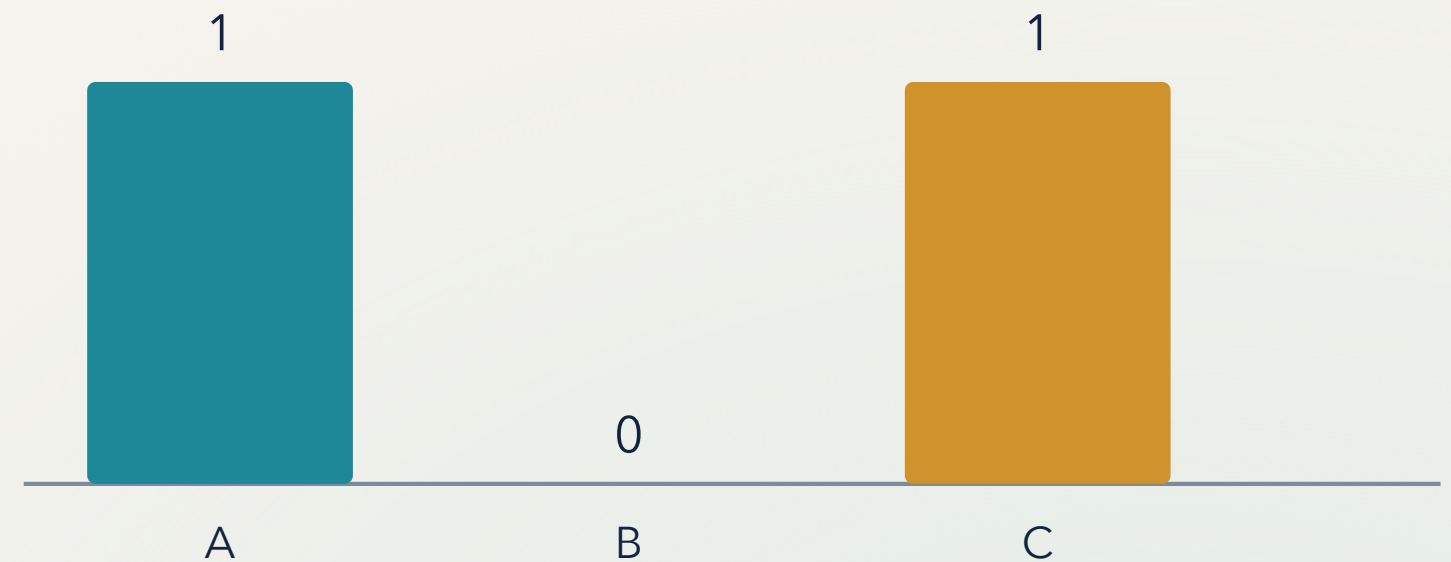
C: key [1,1] → value [10,10]

Mechanism diagram / illustrative teaching values

A score is a compatibility function

A score compares one query with one key before normalization.

dot score: $e_j = q \cdot k_j$



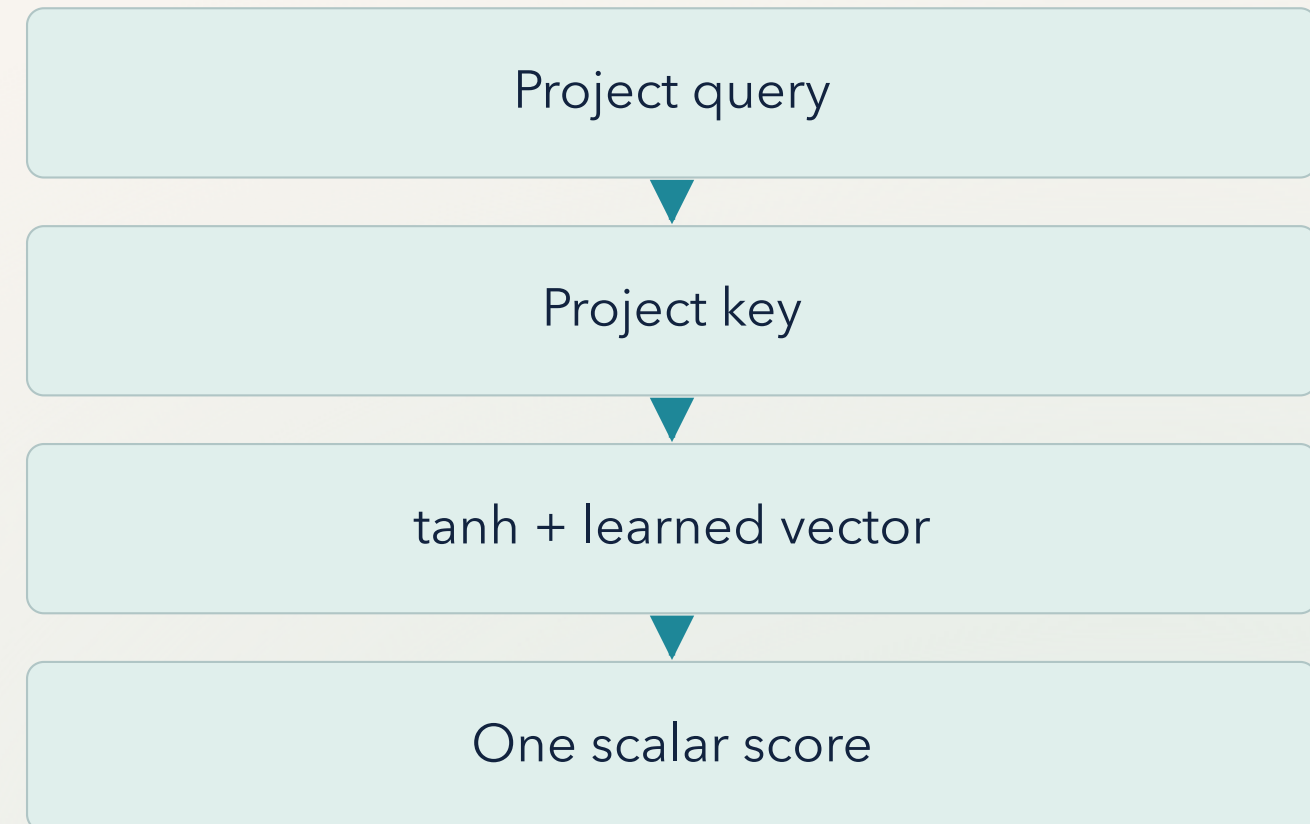
Illustrative values; see the stated calculation

Mechanism diagram / illustrative teaching values

Additive attention learns a scoring network

An MLP can compare a decoder query with each encoder key.

$$e_{ij} = v_a^T \tanh(W_q q_i + W_k k_j + b)$$

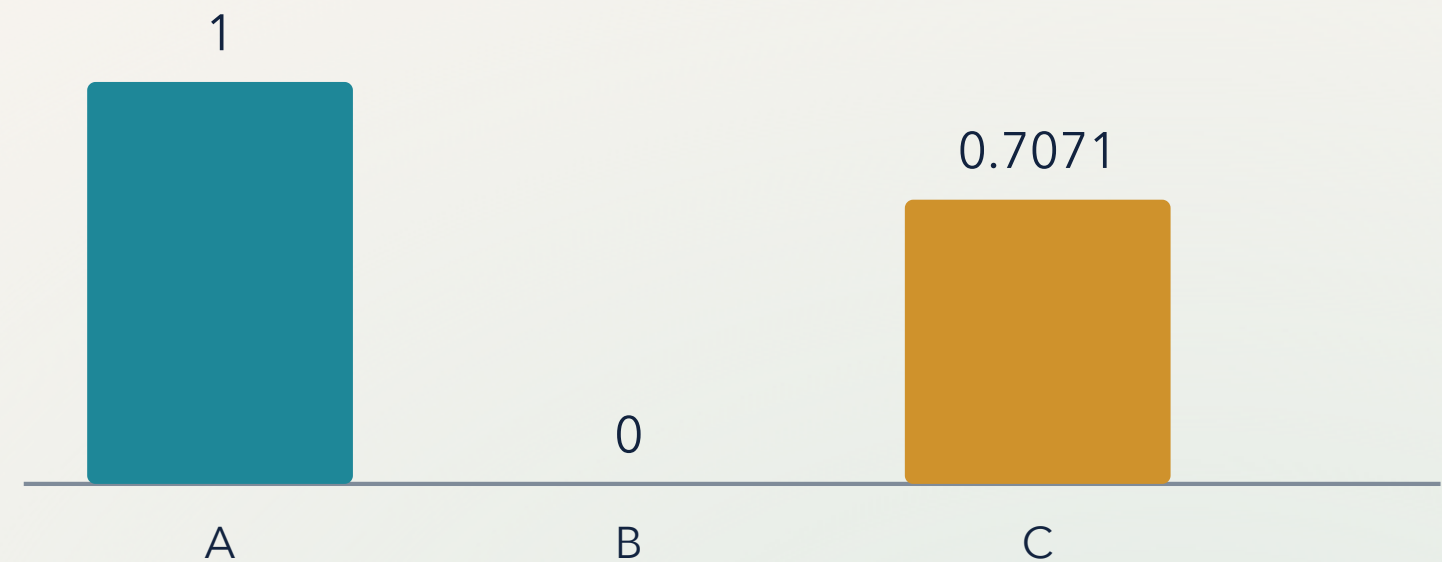


Mechanism diagram / illustrative teaching values

Scaled dot products control score magnitude

Divide by $\sqrt{d_k}$ so high-dimensional dot products are less likely to saturate softmax.

$$S = QK^T / \sqrt{d_k}$$



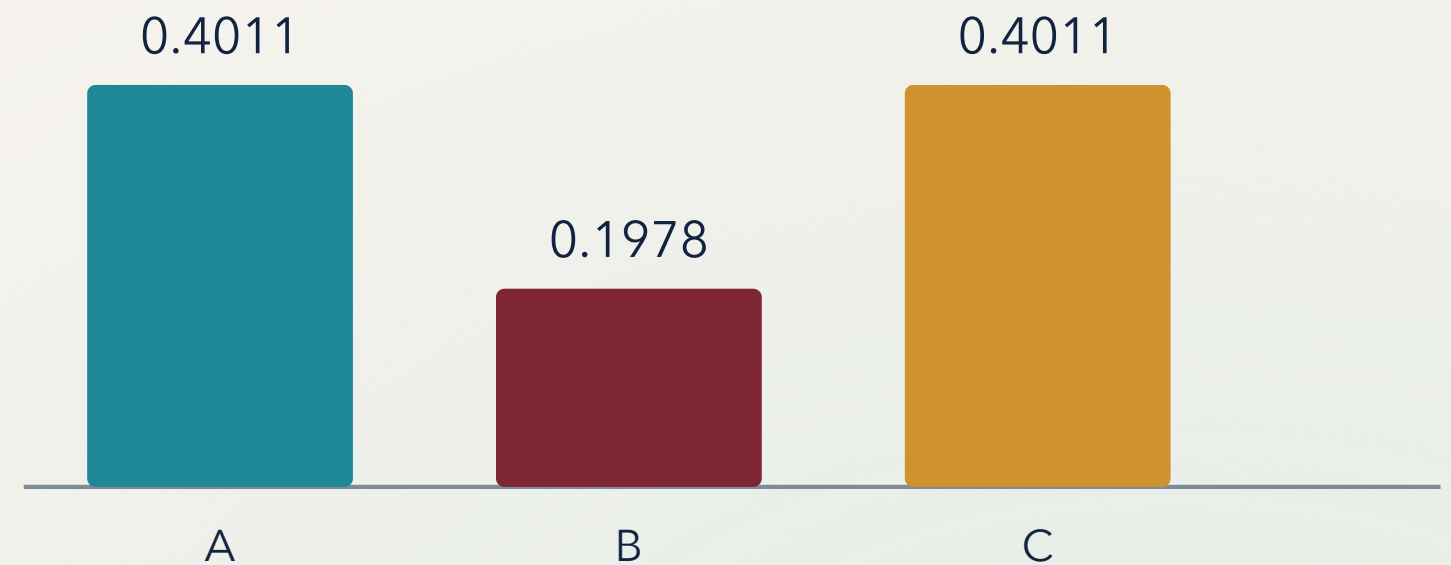
Illustrative values; see the stated calculation

Mechanism diagram / illustrative teaching values

Softmax allocates a unit total weight

Normalize separately for each query over its available keys.

$$a_{ij} = \exp(s_{ij}) / \sum_m \exp(s_{im})$$



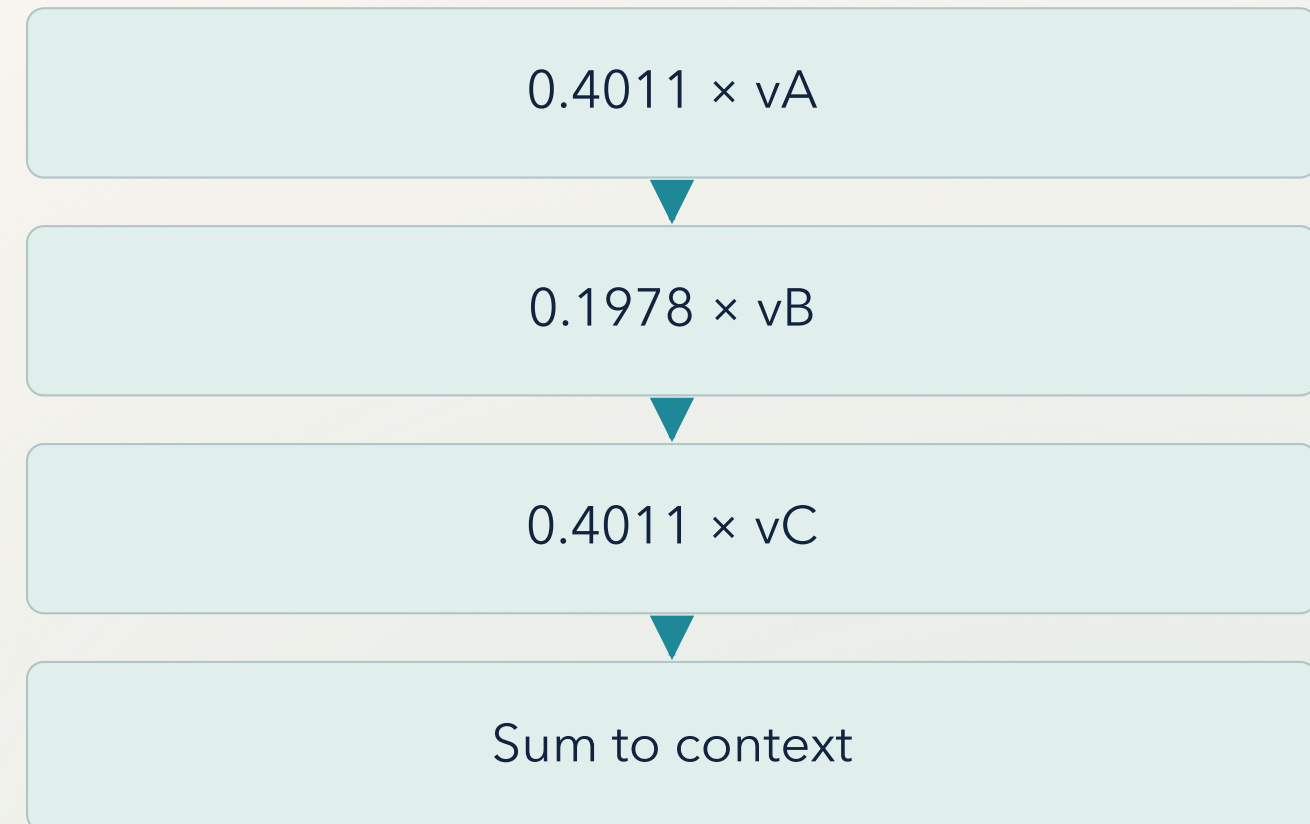
Illustrative values; see the stated calculation

Mechanism diagram / illustrative teaching values

Retrieval is a weighted sum

Each value contributes according to its attention weight.

$$c_i = \sum_j a_{ij} v_j$$



Mechanism diagram / illustrative teaching values

Read matrix axes before the formula

Rows are query destinations; columns are key sources.

Rows: queries / receiving positions

0.60	0.30	0.10
0.20	0.20	0.60

Key / source position →

$$(N_q \times d_k) (d_k \times N_k) \rightarrow N_q \times N_k$$

Mechanism diagram / illustrative teaching values

Work one example: choose Q , K , and V

The values are deliberately small so all intermediates can be checked.

$q=[1,0]$ $K=[[1,0],[0,1],[1,1]]$

$Q: 1 \times 2$

$K: 3 \times 2$

$V: [[10,0],[0,10],[10,10]]$

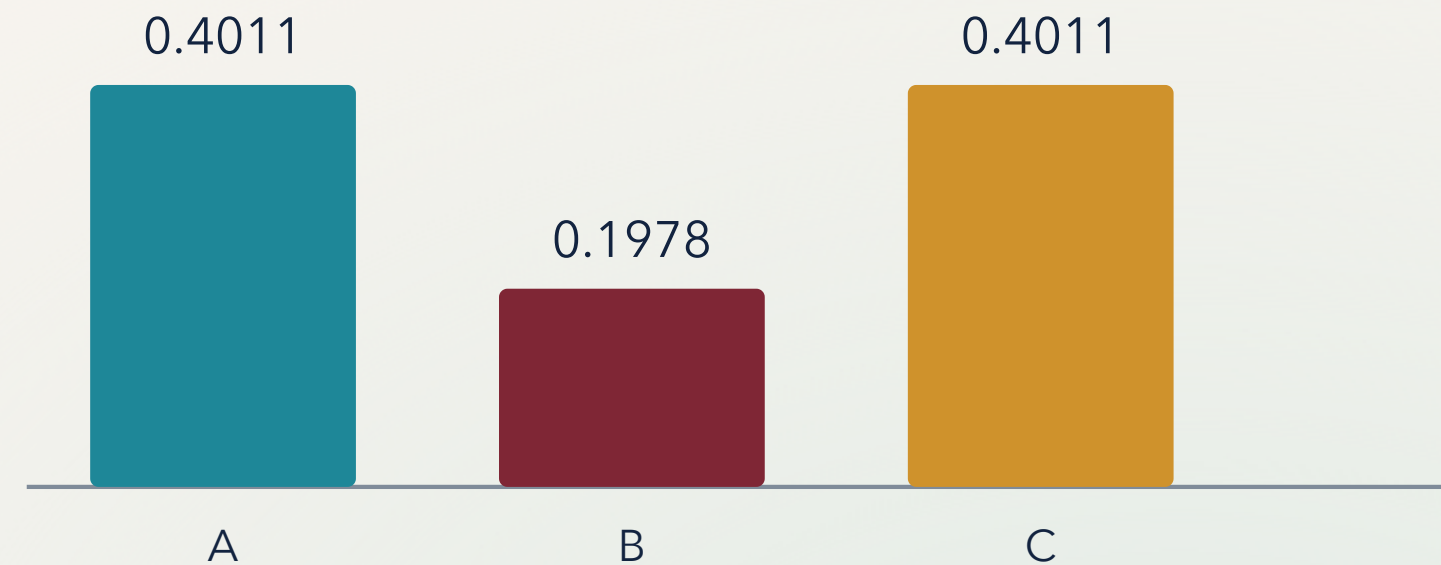
Mechanism diagram / illustrative teaching values

Work one example: scores become weights

A zero dot product still receives a positive normalized weight.

scores $\approx [0.7071, 0, 0.7071]$

What are the three attention weights?



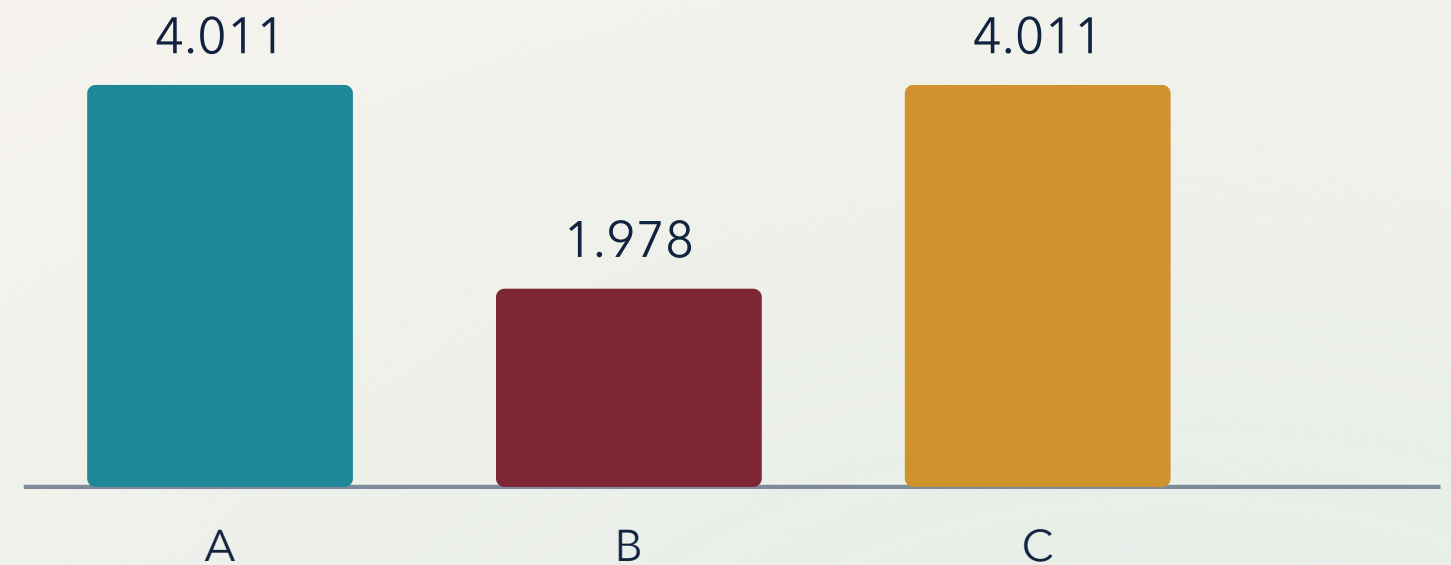
Illustrative values; see the stated calculation

Mechanism diagram / illustrative teaching values

Work one example: combine value coordinates

Use the same weights for every value coordinate.

context $\approx [8.0222, 5.9889]$

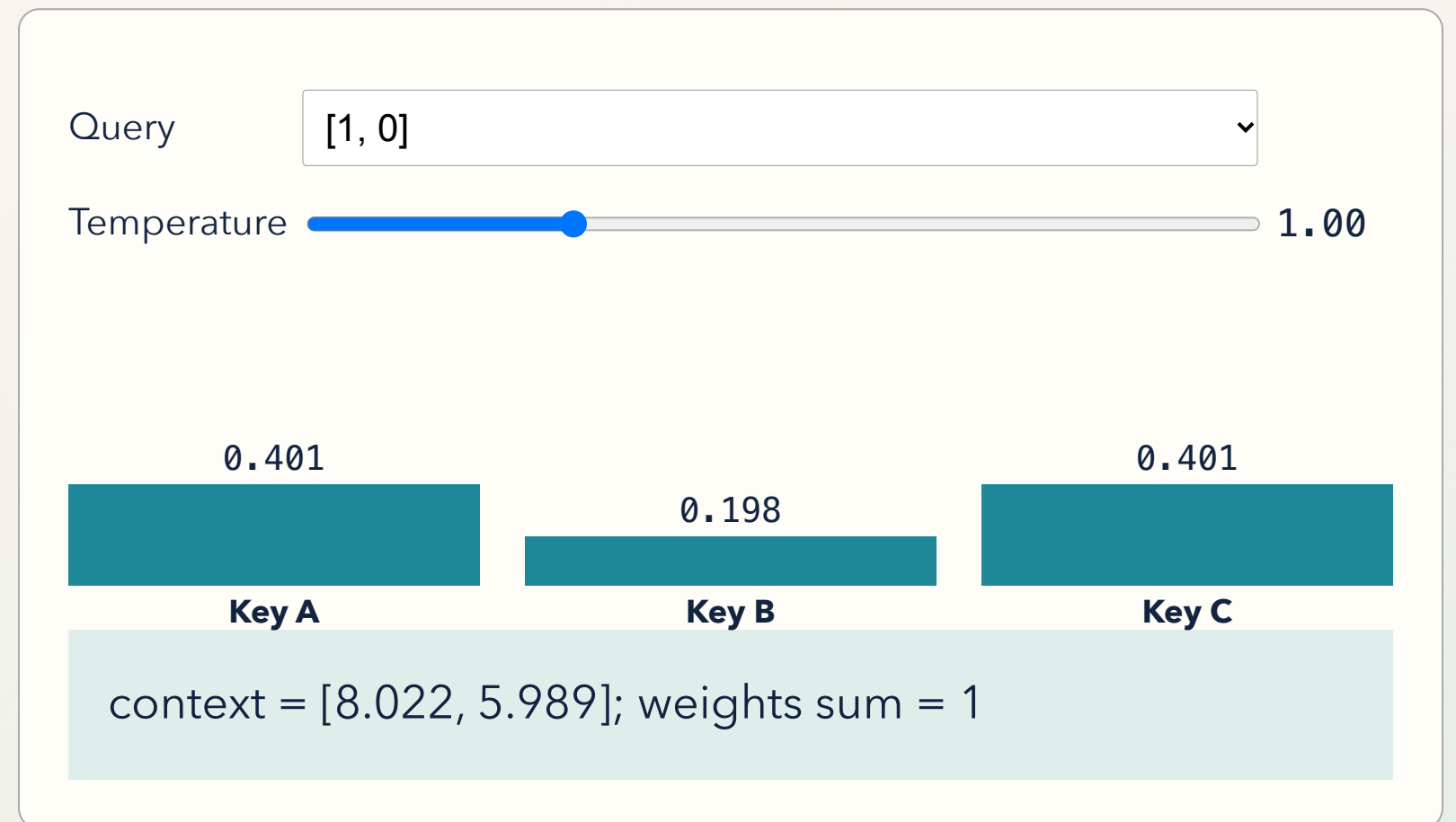


Illustrative values; see the stated calculation

Mechanism diagram / illustrative teaching values

Change the query; keep memory fixed

Attention can route the same memory differently for each request.

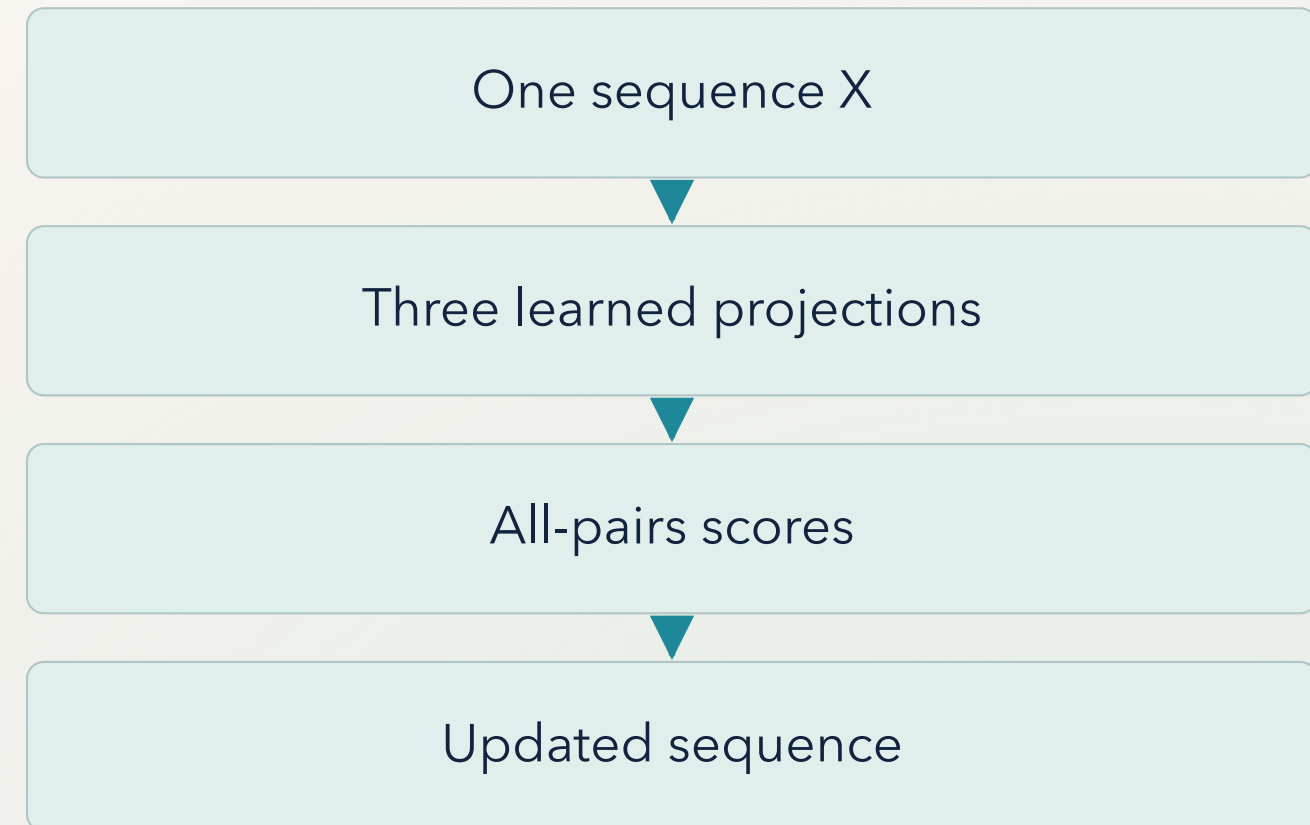


Interactive chosen values – change one variable

Self-attention uses one source sequence

Q, K, and V come from learned projections of the same representations.

$$Q=XW_Q \quad K=XW_K \quad V=XW_V$$



Mechanism diagram / illustrative teaching values

Cross-attention connects two streams

Queries come from one sequence; keys and values come from another.

$$Q=YWQ \quad K=XWK \quad V=XWV$$

A

Decoder Y supplies queries

B

Encoder X supplies keys and values

Mechanism diagram / illustrative teaching values

Causal masking hides future keys

Before softmax, set scores for key positions $j > i$ to $-\infty$.

$$\text{allowed}(i, j) \Leftrightarrow j \leq i$$

Rows: queries / receiving positions

1.00	x	x	x	x	x
0.50	0.50	x	x	x	x
0.33	0.33	0.33	x	x	x
0.25	0.25	0.25	0.25	x	x
0.20	0.20	0.20	0.20	0.20	x
0.17	0.17	0.17	0.17	0.17	0.17

x ← blocked before softmax
Key / source position →

Mechanism diagram / illustrative teaching values

Padding masks hide nonexistent tokens

Padding and causality enforce different constraints.

Rows: queries / receiving positions

1.00	×	×	×	×	×
0.50	0.50	×	×	×	×
0.33	0.33	0.33	×	×	×
0.25	0.25	0.25	0.25	×	×
0.20	0.20	0.20	0.20	×	×
0.17	0.17	0.17	0.17	×	×

× = blocked before softmax
Key / source position →

Mechanism diagram / illustrative teaching values

An all-masked row is undefined

Softmax requires at least one permitted source per query.

all scores = $-\infty$ $\rightarrow$ no valid normalization

A

Valid: at least one visible key

B

Invalid: every key is masked

Mechanism diagram / illustrative teaching values

Attention exposes a short information path

A query can access a distant key within one dense attention layer.

A

RNN: pass through many state updates

B

Attention: direct query-to-key edge

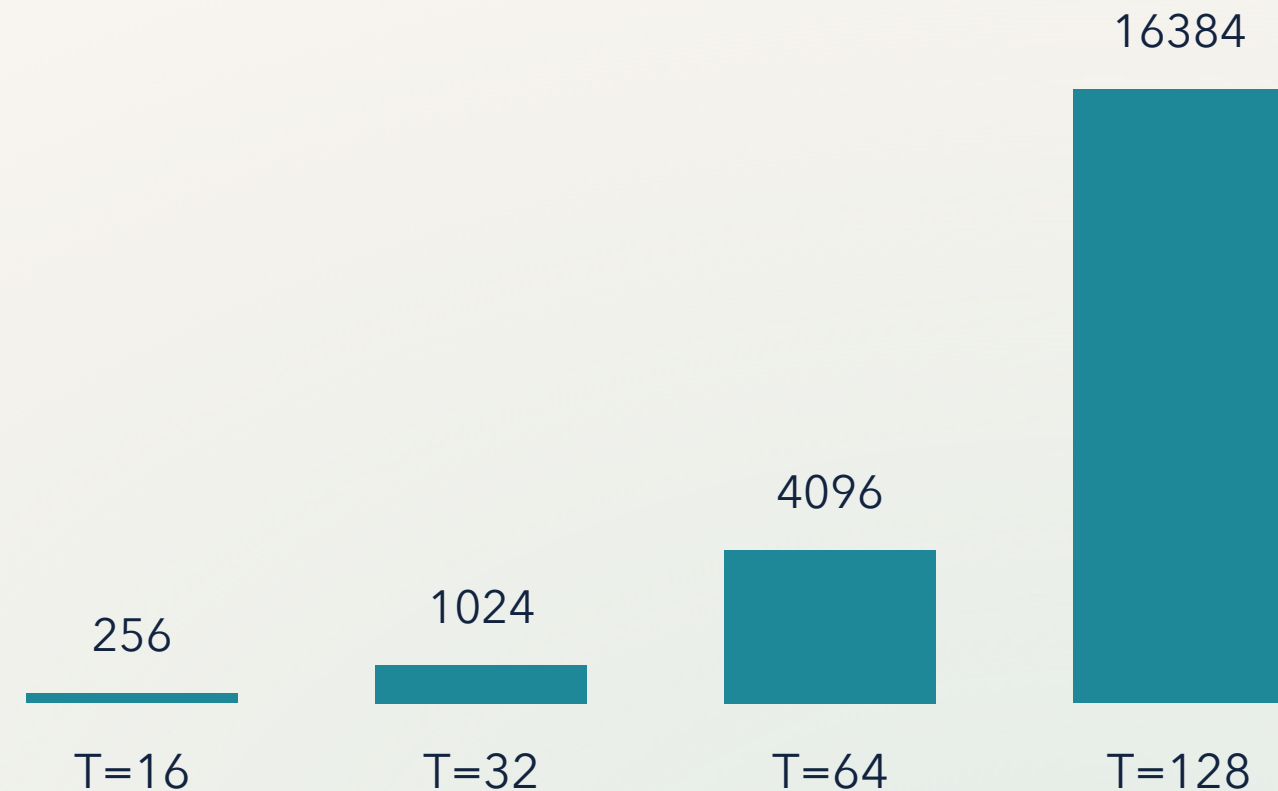
Mechanism diagram / illustrative teaching values

Dense attention has a quadratic score grid

Doubling context length quadruples score entries at fixed head count.

per head: $T \times T$ scores

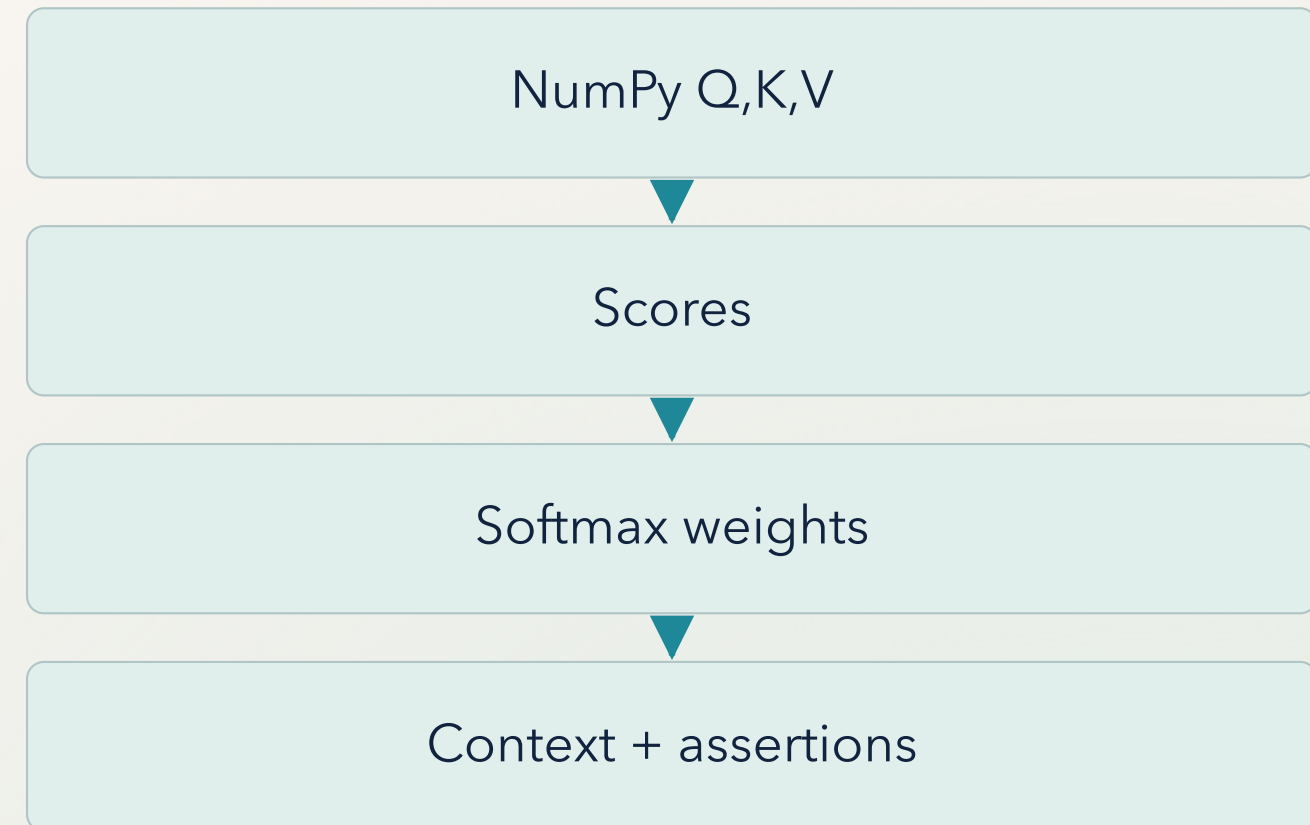
Dense score entries per head



Mechanism diagram / illustrative teaching values

Open the attention notebook

Verify your hand calculation before exploring masks.



Mechanism diagram / illustrative teaching values

Test causality and normalization

Every permitted row sums to one; every future weight is zero.

$$\text{row_sum}(A)=1 \quad A[i,j>i]=0$$

Rows: queries / receiving positions

1.00	×	×
0.30	0.70	×
0.20	0.30	0.50

Key / source position →

Mechanism diagram / illustrative teaching values

Compare two implementations of one operation

NumPy and PyTorch should agree for the same inputs, scale, and mask.

A

Explicit NumPy operation

B

PyTorch scaled-dot-product API

Mechanism diagram / illustrative teaching values

Routing weights are not causal explanations

Changing values can alter the output while attention weights remain fixed.

A

Same $Q, K \rightarrow$ same attention weights

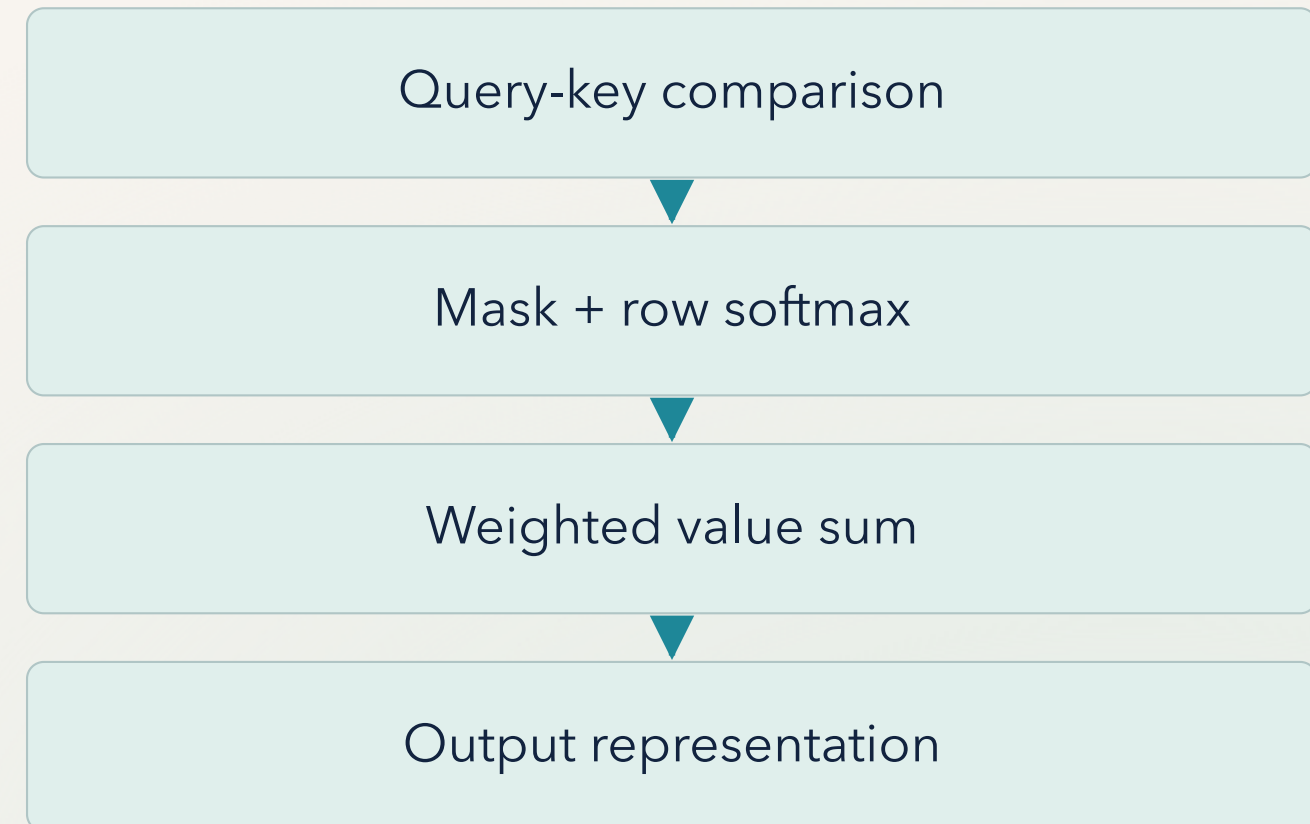
B

Different $V \rightarrow$ different retrieved context

Mechanism diagram / illustrative teaching values

Explain the mechanism before the architecture

State the sequence: compare,
normalize, retrieve.



Mechanism diagram / illustrative teaching values

Summary: compare, normalize, retrieve

Attention retrieves a query-dependent mixture of values. Masks determine which information is available.

$$A = \text{softmax}((QK^T / \sqrt{d_k}) + \text{mask})$$
$$\text{Context} = AV$$

Compare

Query-key scores measure compatibility

Normalize

Mask first; softmax across keys

Retrieve

Use the weights to combine values

NEXT: add positions, MLPs, and residual paths

Mechanism diagram / illustrative teaching values