

W07 · H3 · 16:00–17:00 · 19 October 2026

From Attention to a Transformer

Attention mixes positions. A complete decoder must also represent order and predict the next token.

Machine Learning in Engineering Science
Fall 2026 · Week 07 · Hour 3
Chi-Hua Jonny Yu · NCKU / LAiMM



A complete causal decoder, not attention alone

This hour: assemble and test a decoder

Trace the complete computation, then test whether future tokens can affect earlier predictions.

Goal: explain a block and preserve a causal training experiment.

00-22 min

Positions + the complete block



22-36 min

Shifted targets + CPU notebook



36-50 min

Baseline + intervention checks



50-60 min

Generation + limits + summary

Mechanism diagram / illustrative teaching values

Position information supplies order

Unmasked attention alone is permutation-equivariant when positions are absent.

$$\text{Attention}(PX) = P \text{ Attention}(X)$$

A

Same tokens, different order

B

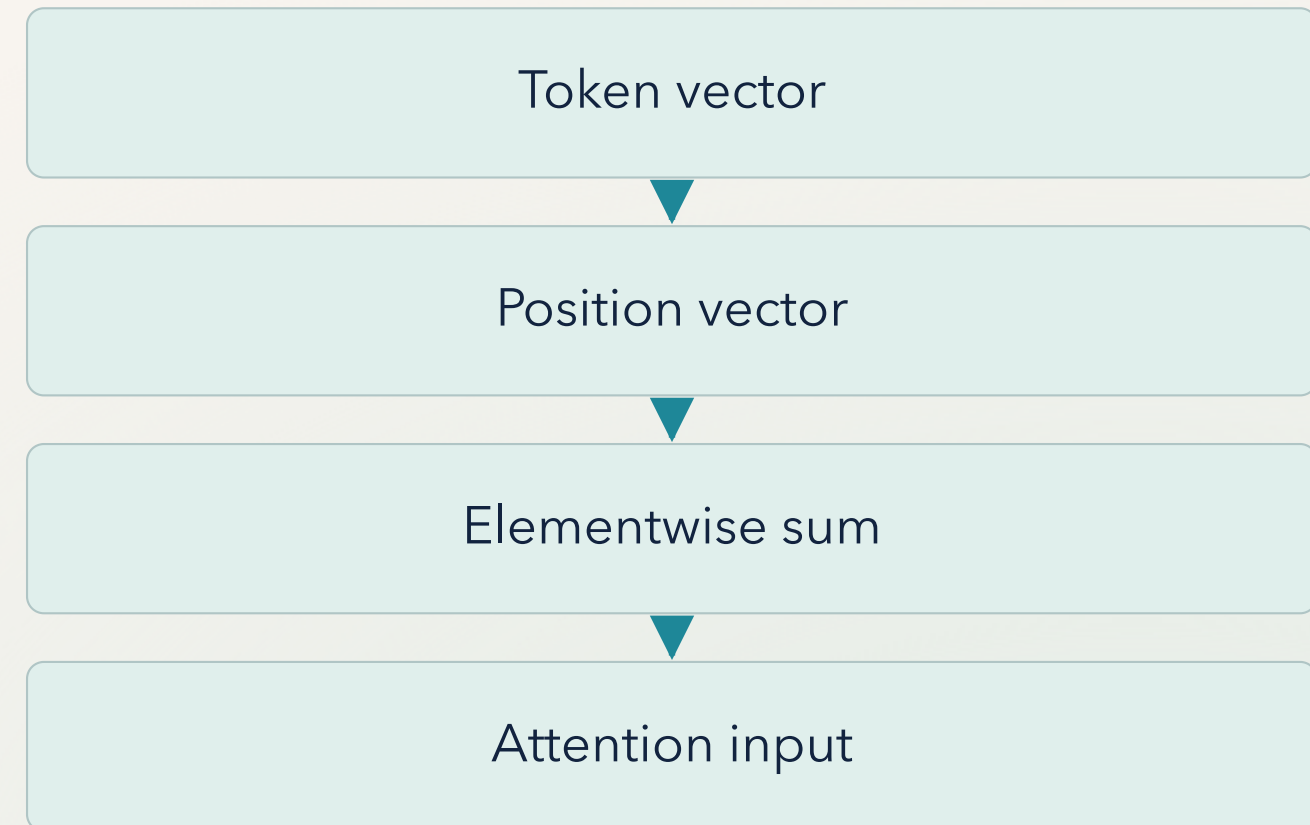
Represent order using positions

Mechanism diagram / illustrative teaching values

There are several position mechanisms

The teaching model adds learned position vectors; the original Transformer used sinusoidal encodings.

$$z_t = \text{token_embedding}(\text{id}_t) + \text{position_embedding}(t)$$

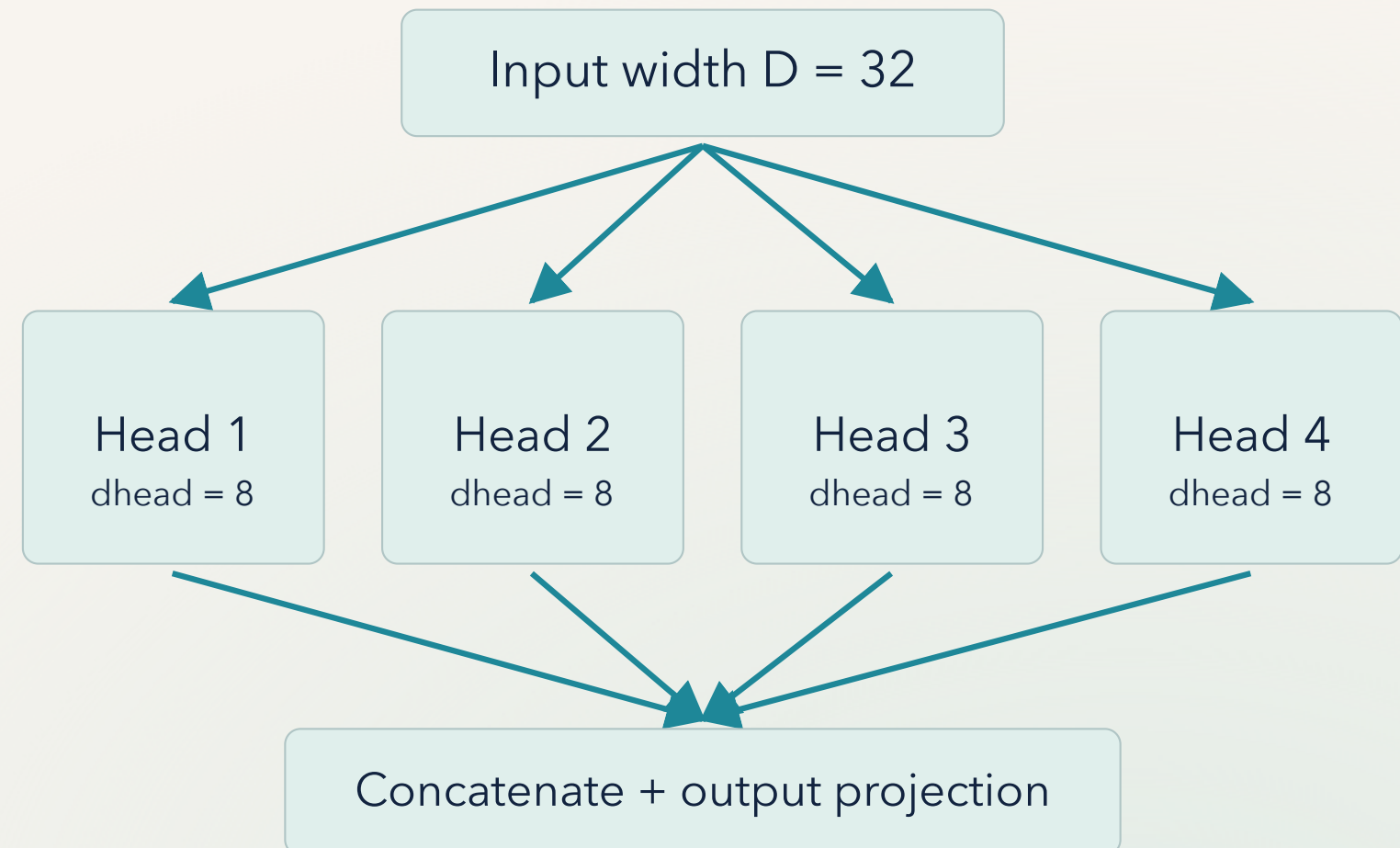


Mechanism diagram / illustrative teaching values

Multiple heads learn multiple projections

Each head routes information in its own learned query/key/value subspace.

$$D = H \times d_{\text{head}}$$



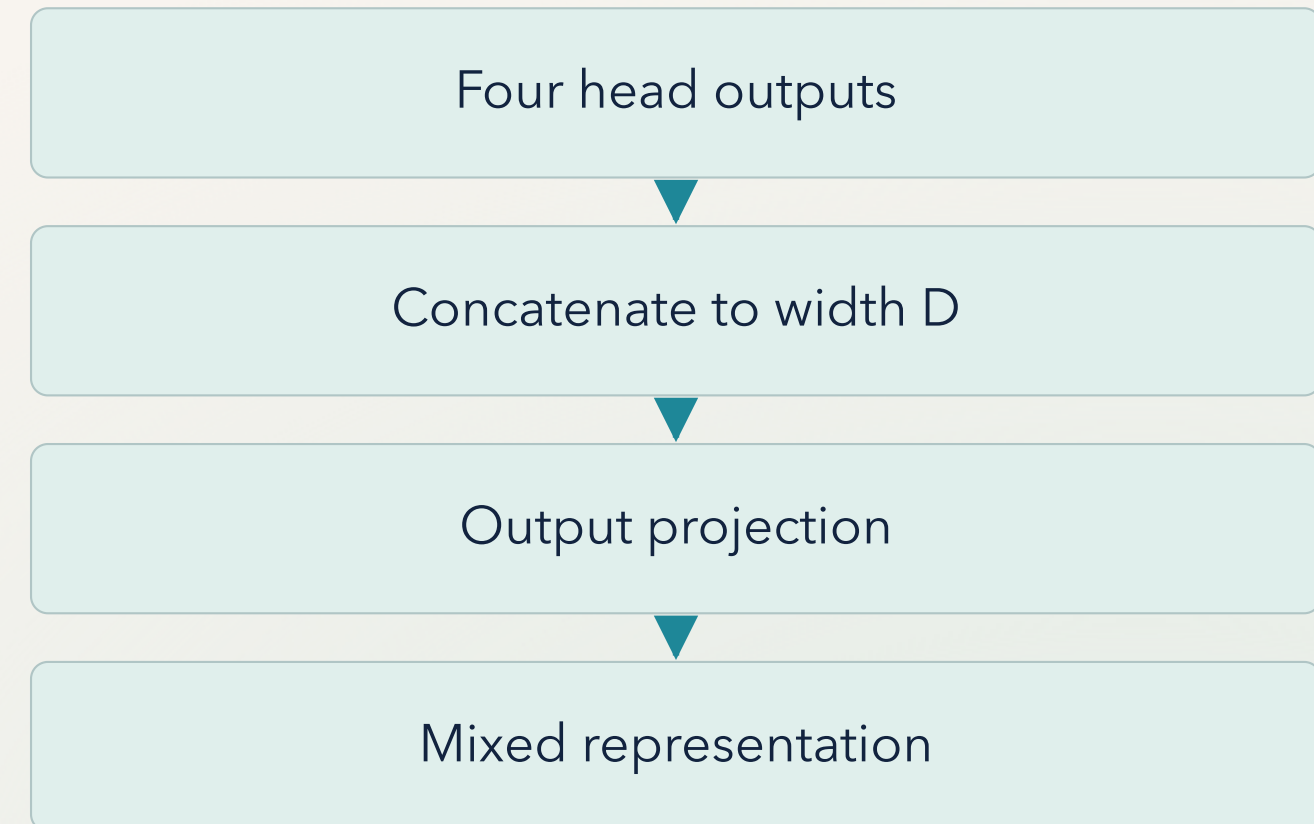
Mechanism diagram / illustrative teaching values

Concatenate heads, then project

Head outputs are joined and mixed by a learned output projection.

$\text{MultiHead} = \text{Concat}(\text{head}_1, \dots, \text{head}_H)$

WO

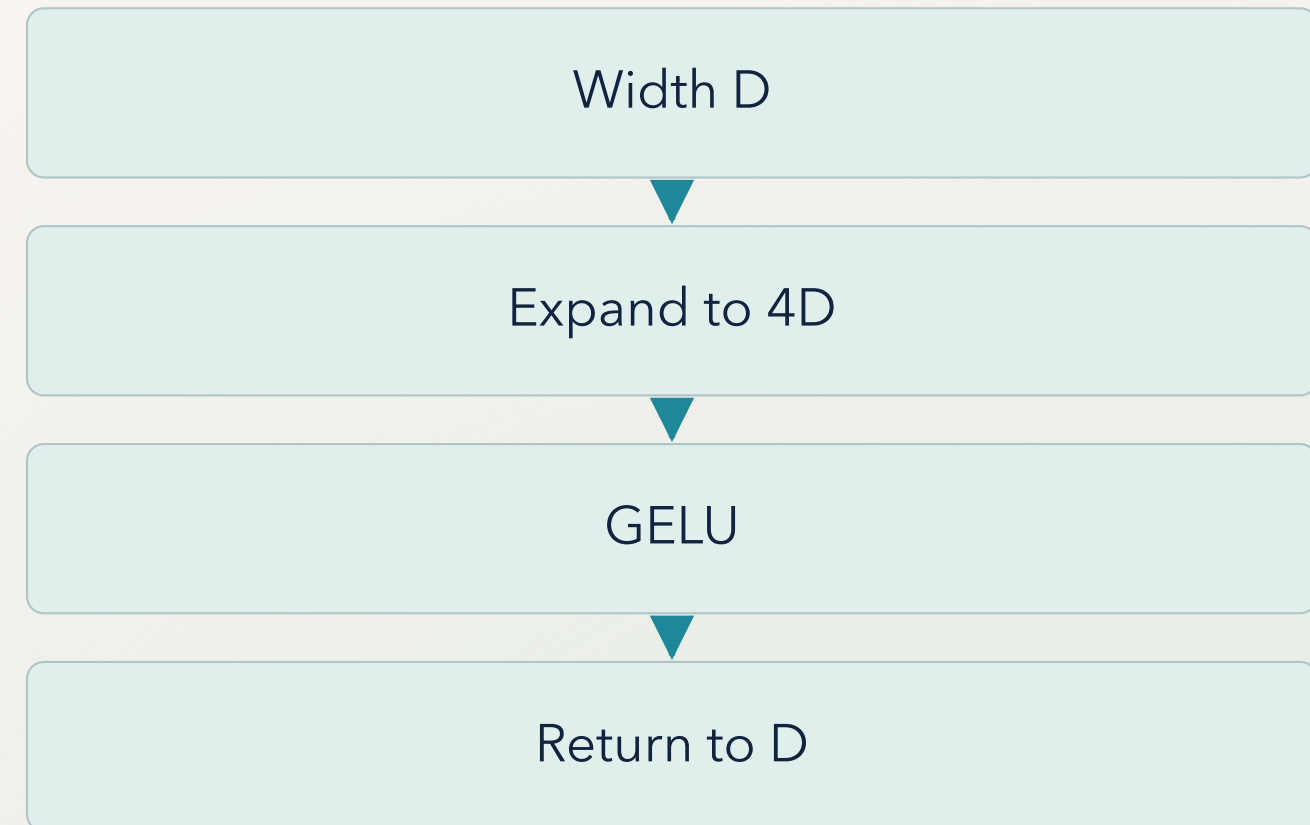


Mechanism diagram / illustrative teaching values

The feed-forward network transforms each token

The same nonlinear MLP is applied independently at each position.

$$\text{FFN}(x) = W_2 \text{ GELU}(W_1 x + b_1) + b_2$$

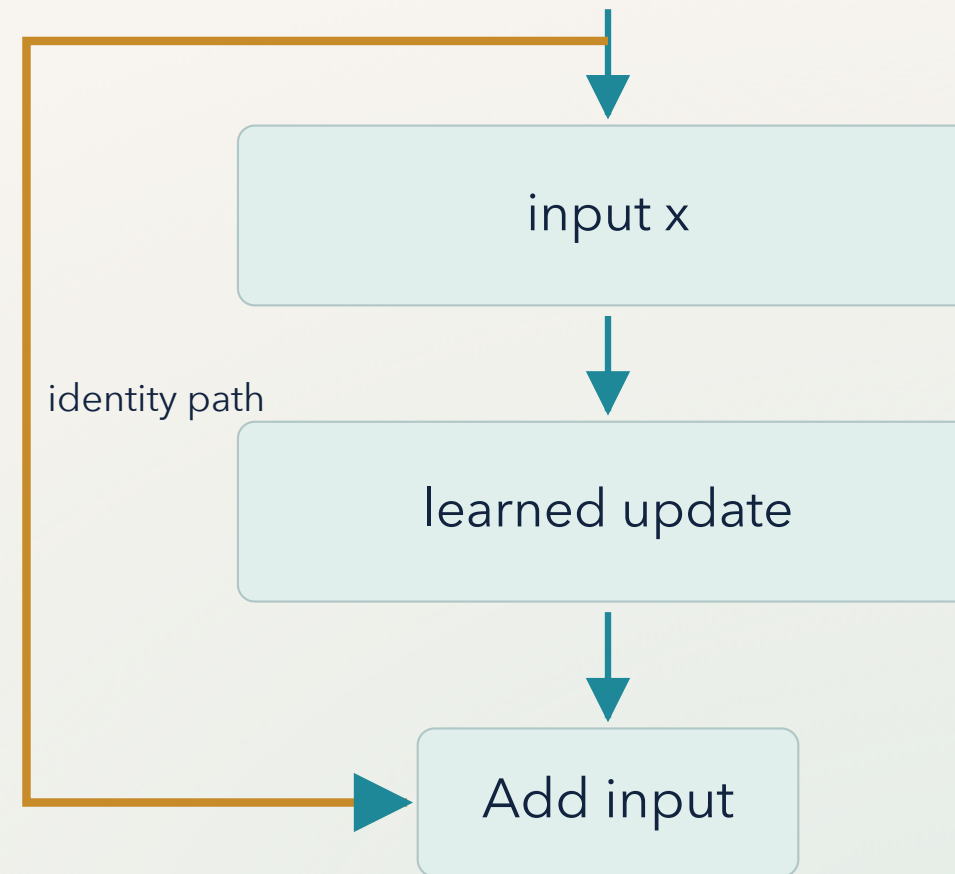


Mechanism diagram / illustrative teaching values

Residual paths preserve an identity route

A sublayer adds a learned update to its input representation.

$$y = x + \text{Sublayer}(x)$$

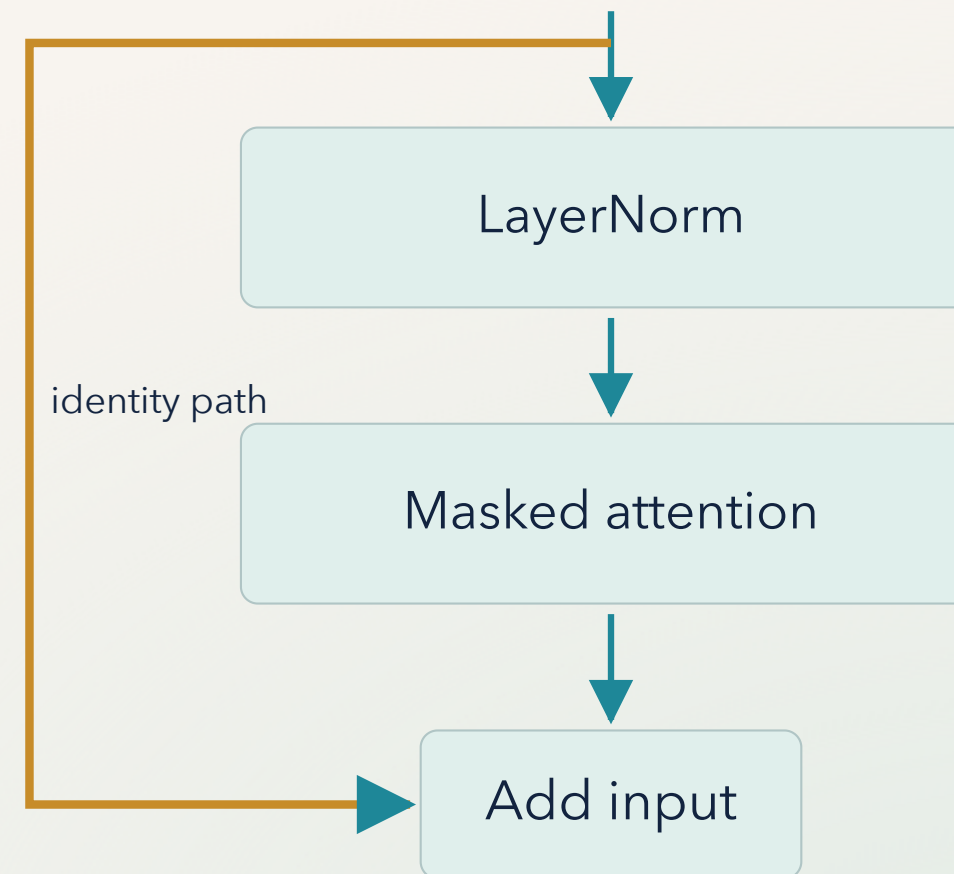


Mechanism diagram / illustrative teaching values

Normalization placement changes the block

Our teaching decoder normalizes before each sublayer.

$$x \leftarrow x + \text{Attention}(\text{LN}(x))$$



Then: normalized feed-forward update + residual

Mechanism diagram / illustrative teaching values

An encoder uses bidirectional context

Encoder self-attention can access all valid input positions.

Rows: queries / receiving positions

0.40	0.20	0.10	0.30
0.20	0.50	0.20	0.10
0.10	0.20	0.40	0.30
0.30	0.10	0.20	0.40

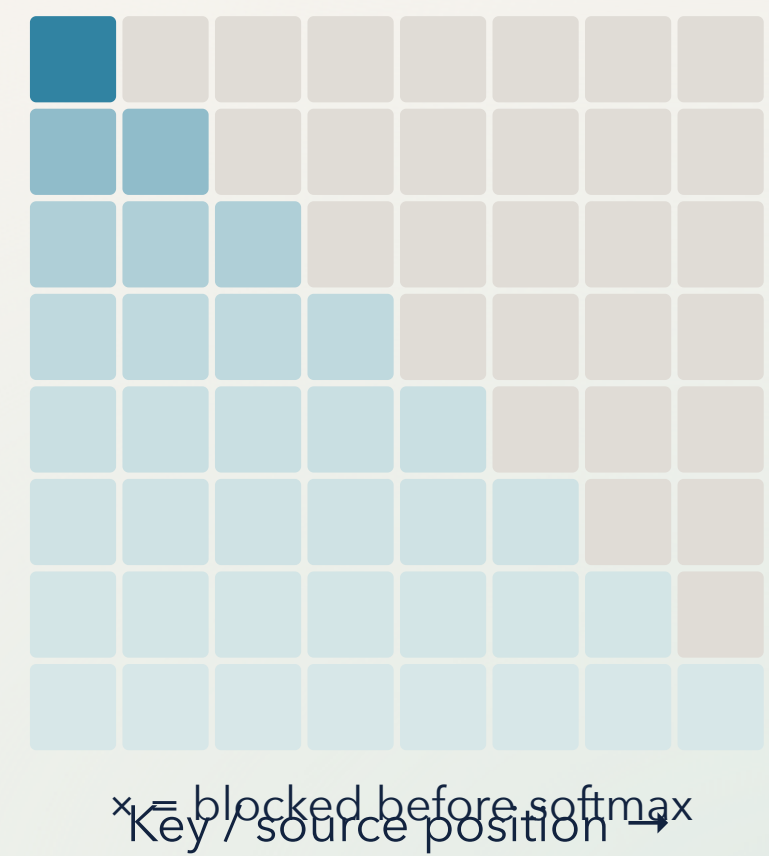
Key / source position →

Mechanism diagram / illustrative teaching values

A decoder-only model hides the future

Masked self-attention supports next-token prediction from a prefix.

Rows: queries / receiving positions



Mechanism diagram / illustrative teaching values

The original model had encoder and decoder

Decoder queries can retrieve from the separate encoder outputs.

A

Encoder: full self-attention + FFN

B

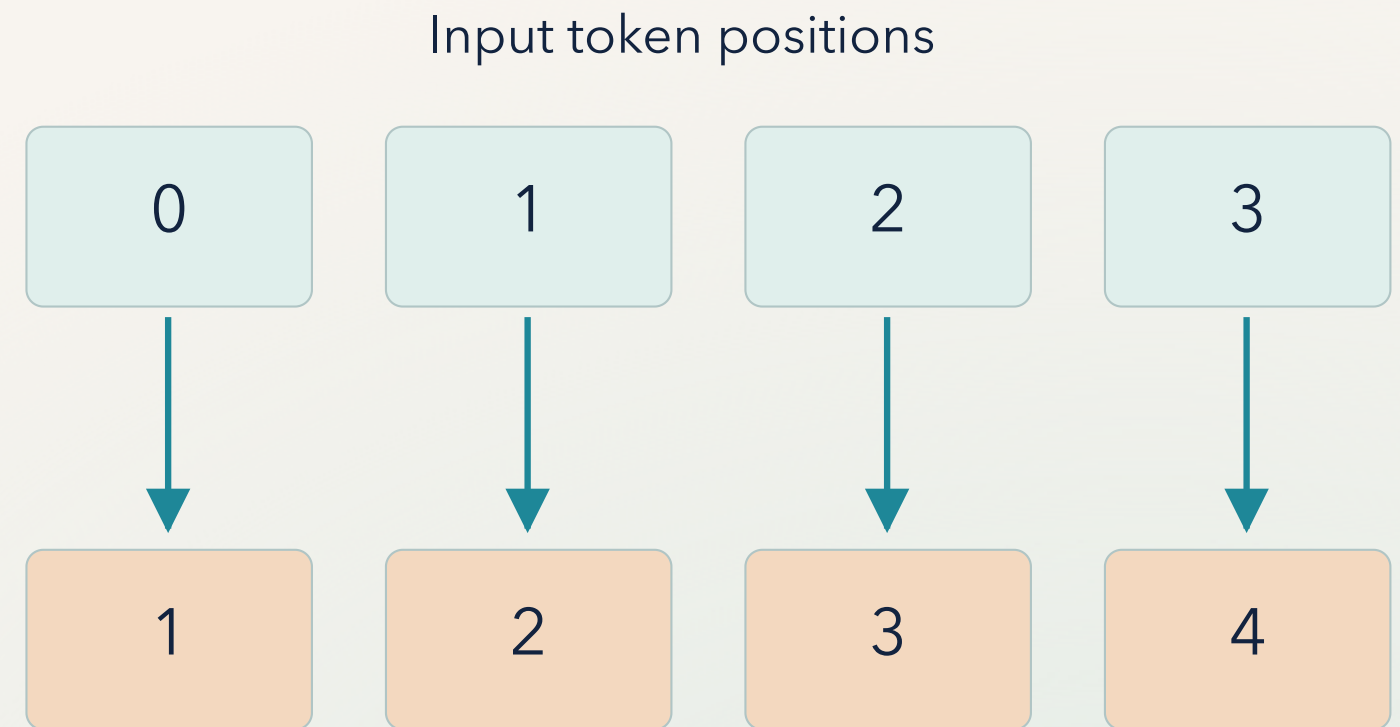
Decoder: causal self-attention + cross-attention + FFN

Mechanism diagram / illustrative teaching values

Predict the next token at every position

Shift targets one step ahead of inputs.

input: [0,1,2,3] target: [1,2,3,4]



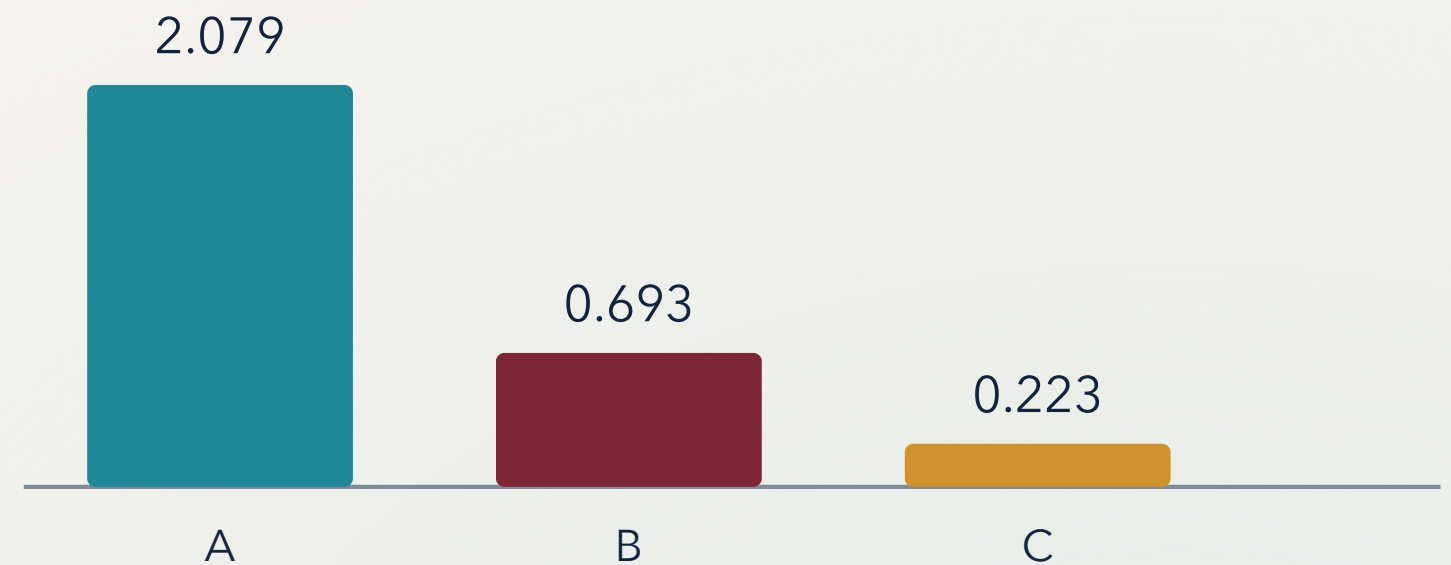
Targets are shifted one token ahead

Mechanism diagram / illustrative teaching values

Cross-entropy scores the true next token

The model should assign high probability to observed next-token targets.

$$L = -\text{mean } \log p(y_t \mid x_{\leq t})$$



Illustrative values; see the stated calculation

Mechanism diagram / illustrative teaching values

Training is parallel across known positions

The causal mask permits computing many training losses together.

A

Training: known sequence, masked positions

B

Generation: unknown next token, append and repeat

Mechanism diagram / illustrative teaching values

Open a tiny model you can inspect

One block, width 32, four heads, and eight symbols keep the experiment small.

Context length: 16



Model width: 32 / heads: 4

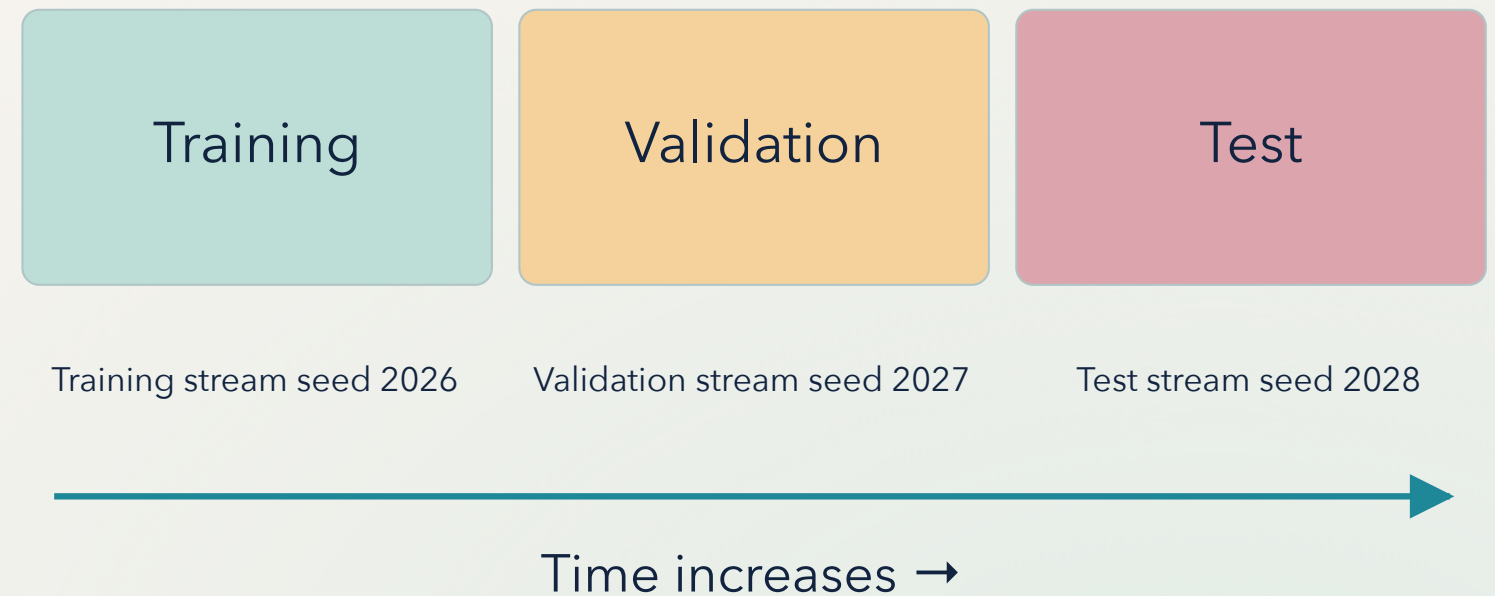


Vocabulary size: 8 / CPU only

Mechanism diagram / illustrative teaching values

Define what generalization means here

Separate streams use different noise seeds but share the same synthetic process.

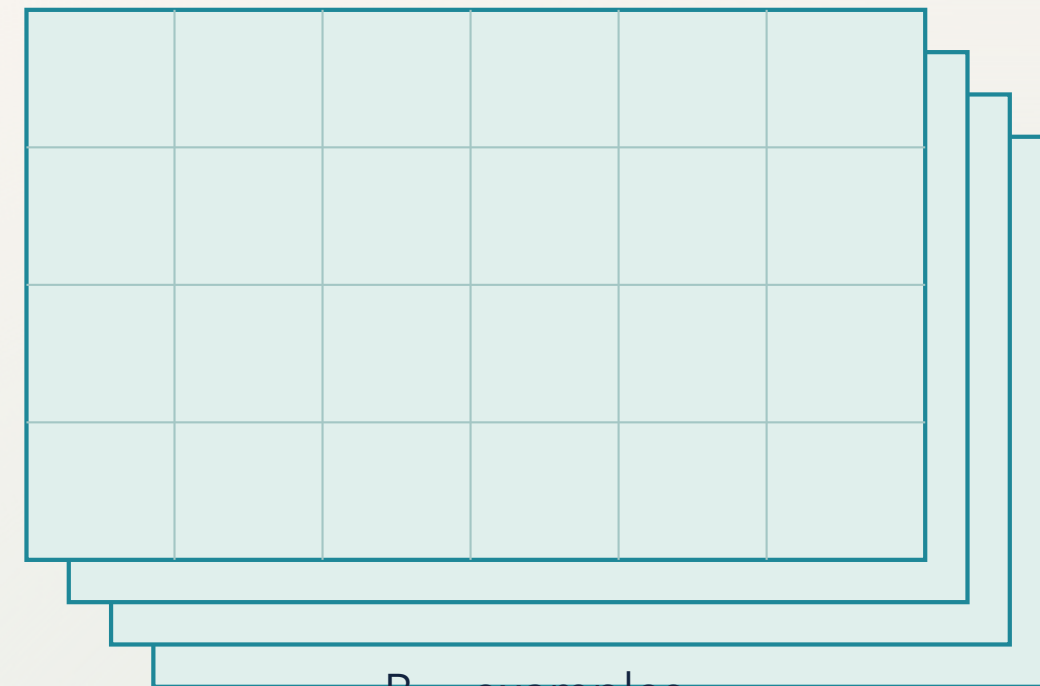


Mechanism diagram / illustrative teaching values

Read the four-dimensional score tensor

Every head produces a query-by-key grid for each example.

scores: (B, H, T, T)



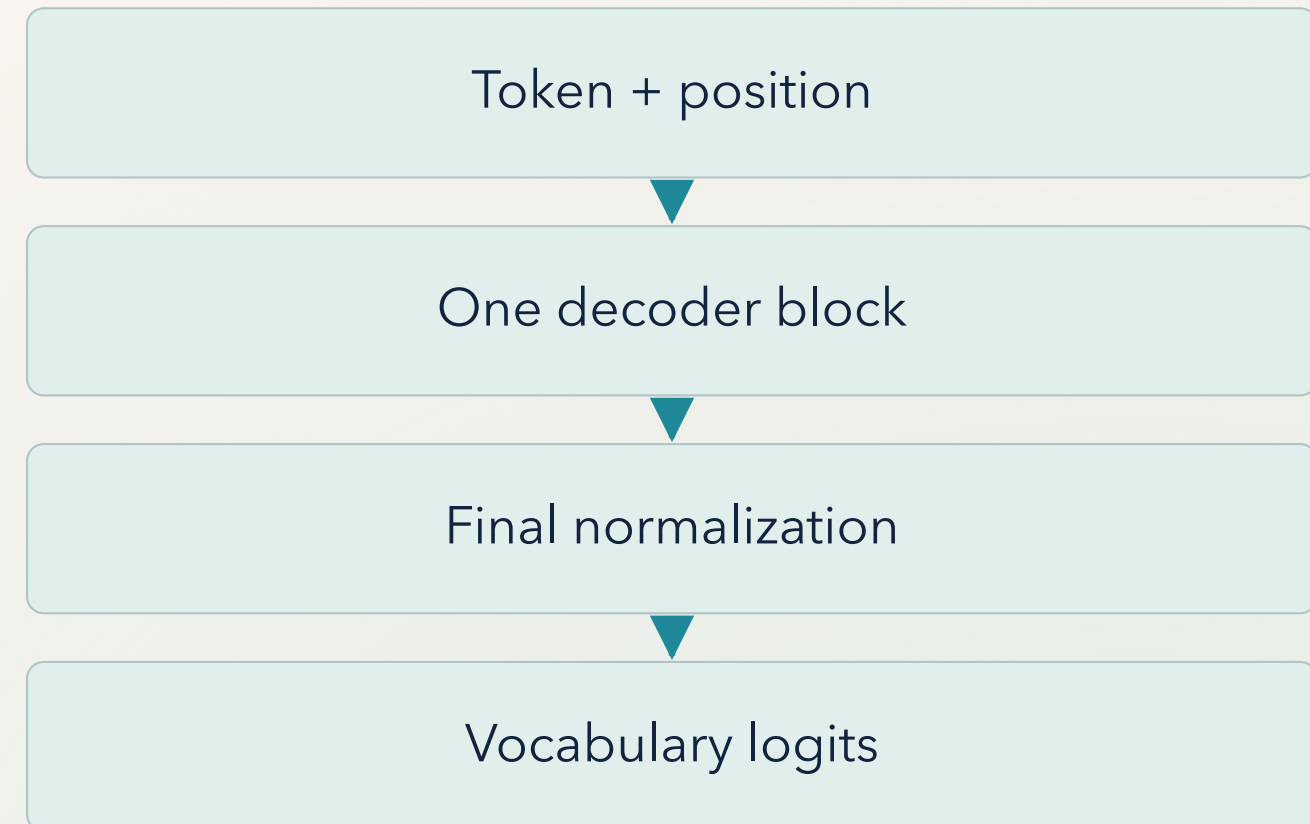
B = examples
H = attention heads
 $T \times T$ = query/key pairs

Mechanism diagram / illustrative teaching values

Trace the model before training

The output has one vocabulary-logit vector per input position.

IDs: (B,T) → logits: (B,T,V)



Mechanism diagram / illustrative teaching values

Fit a training-only bigram baseline

Estimate the next state from the current state using transition counts.

$$P(\text{next} \mid \text{current}) = \frac{\text{smoothed count}}{\text{row total}}$$

Rows: queries / receiving positions

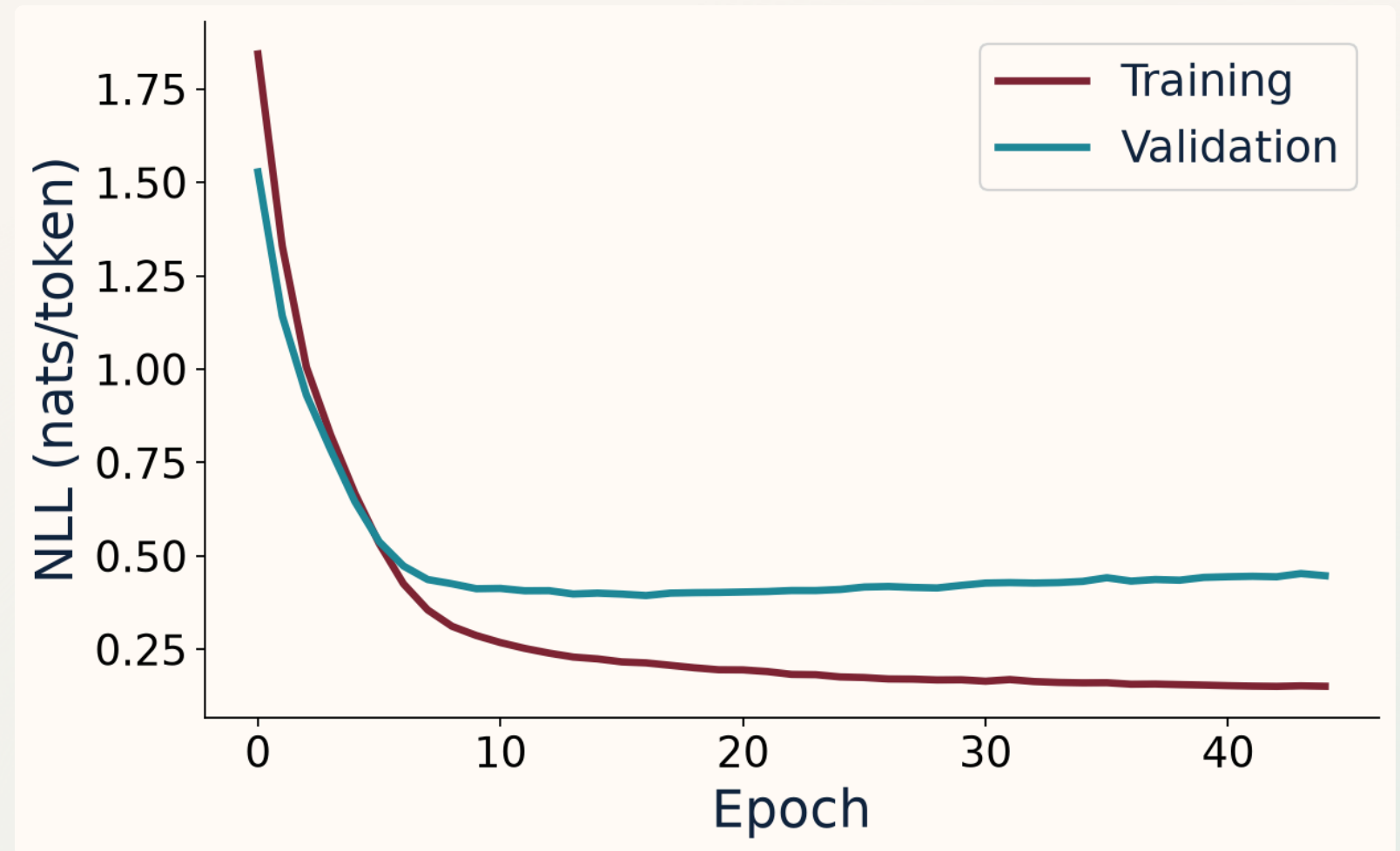
0.10	0.60	0.20	0.10
0.20	0.10	0.60	0.10
0.10	0.30	0.10	0.50
0.50	0.20	0.20	0.10

Key / source position →

Mechanism diagram / illustrative teaching values

Select by validation, evaluate test afterward

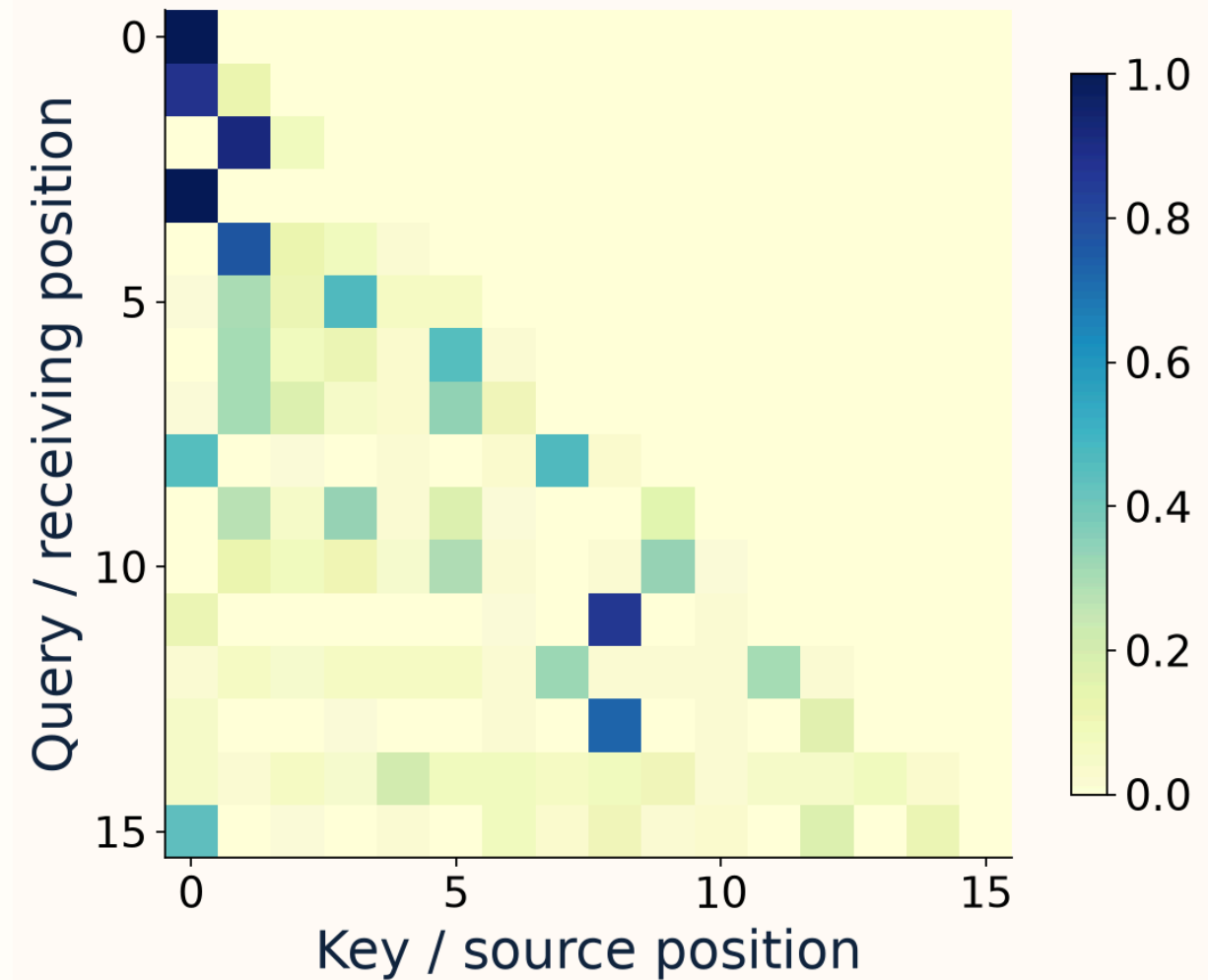
Keep the checkpoint with lowest validation negative log-likelihood.



Actual CPU notebook run · synthetic teaching data

Inspect actual learned attention

Rows are receiving positions; columns are source positions.



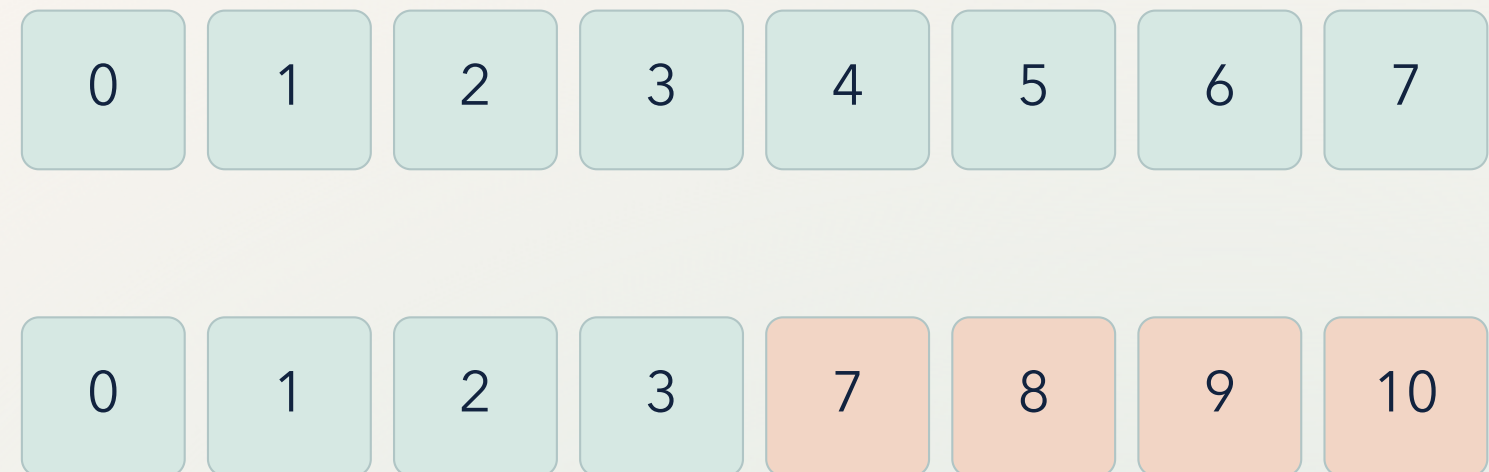
Actual CPU notebook run · synthetic teaching data

A future intervention must preserve earlier logits

Change tokens after position 7; logits at positions 0–7 should remain identical.

$$\Delta_{\text{earlier}} = \max |\text{old_logits} - \text{new_logits}|$$

Original prefix versus modified future



Compare earlier logits: expected $\Delta = 0$

Mechanism diagram / illustrative teaching values

Test the position claim precisely

Permute inputs to unmasked attention without positional information.

A

Permute input rows

B

Output rows permute correspondingly

Mechanism diagram / illustrative teaching values

Generate one token, then append it

Autoregressive decoding grows the prefix sequentially.



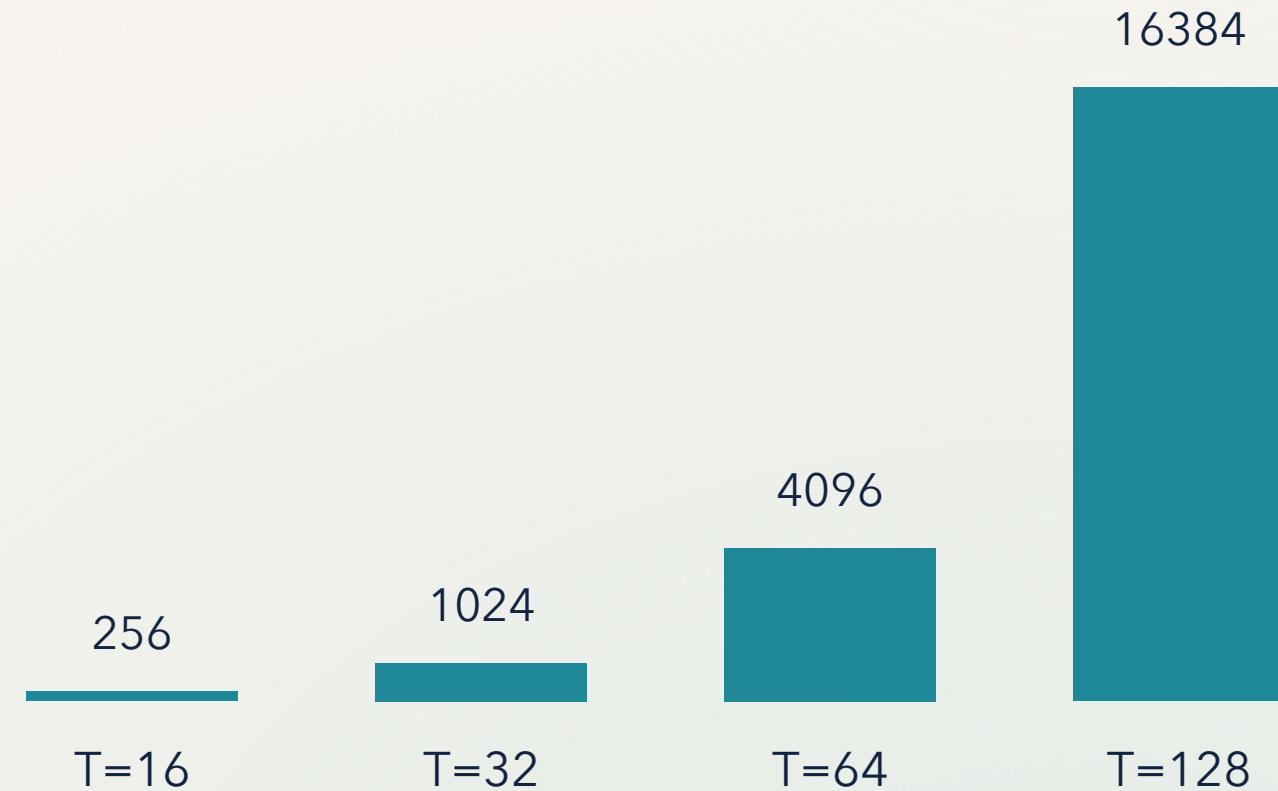
Mechanism diagram / illustrative teaching values

Long context has a measurable cost

Dense attention pair counts grow as T^2 at fixed head count.

T=16: 256 pairs T=32: 1024 pairs

Dense score entries per head



Mechanism diagram / illustrative teaching values

Choose models using task evidence

Streaming state, context length, data size, and compute budget all matter.

A

LSTM: recurrent state and sequential updates

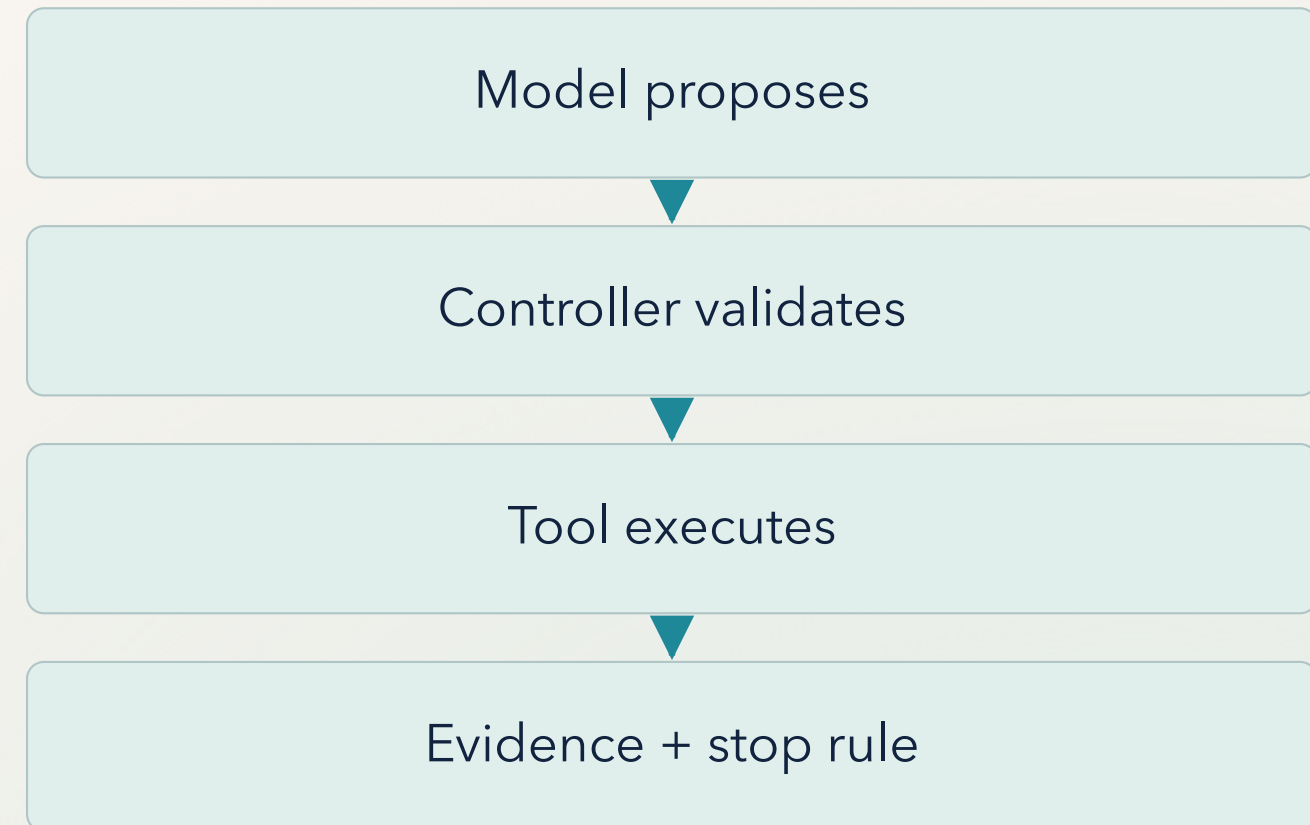
B

Transformer: attention context and parallel training

Mechanism diagram / illustrative teaching values

A token generator is one agent component

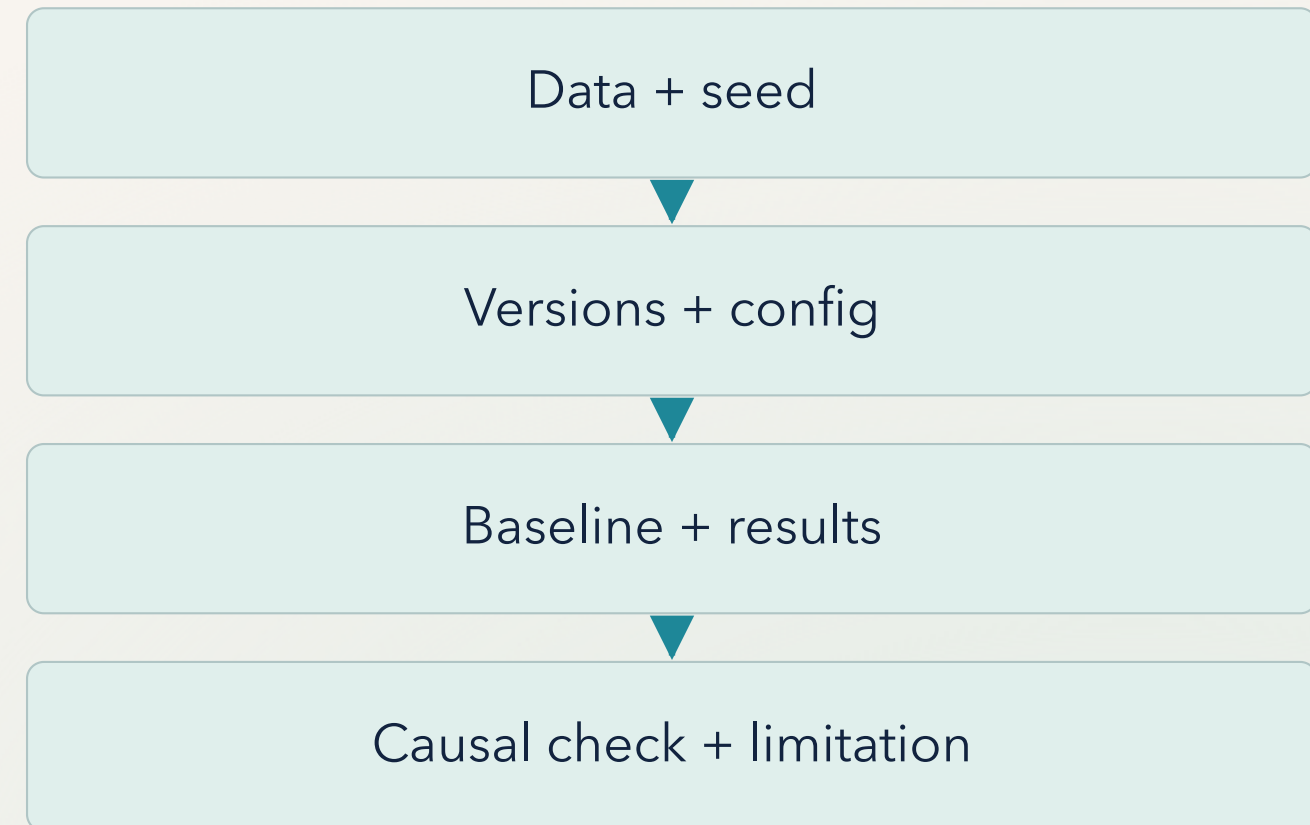
An agent also needs tools, state management, permissions, and a stopping rule.



Mechanism diagram / illustrative teaching values

Preserve the experiment, not just a score

Record data identity, configuration, baseline, checkpoint, and observed tests.



Mechanism diagram / illustrative teaching values

Summary: architecture plus evidence

A causal decoder combines attention with position and feature transformations. Its predictions still need evidence.

prefix → masked decoder → next-token logits
Check: baseline error + future-token intervention

Represent

Token vectors + position information

Transform

Masked attention + tokenwise MLP

Check

Baseline comparison + causal test

TAKE HOME: a token generator is not a complete agent

Mechanism diagram / illustrative teaching values