LECTURE 10 · TRANSFORMERS AND ACT · NOVEMBER 18

Transformers and ACT

From observations to a sequence of actions

How can a robot combine several observations and predict a coordinated movement?



Mathematics for Robotics · ROB-GY 6013 · NYU Tandon · Jing Zhang · Fall 2026

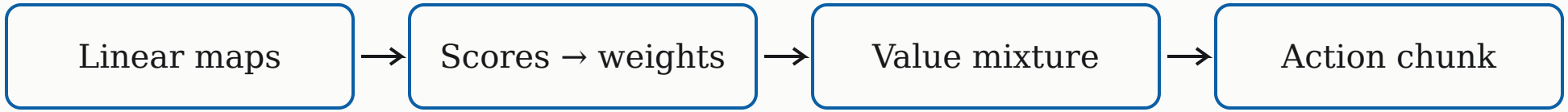
Zhao et al. (2023), §IV-B–C, Appendix C

CONCEPT MAP

Transformers and ACT

The mathematical route

Attention selects information. Position identifies order. A policy turns the resulting representation into actions.



Vaswani et al. (2017), §3.2.1, Eq. 1

DEFINITION

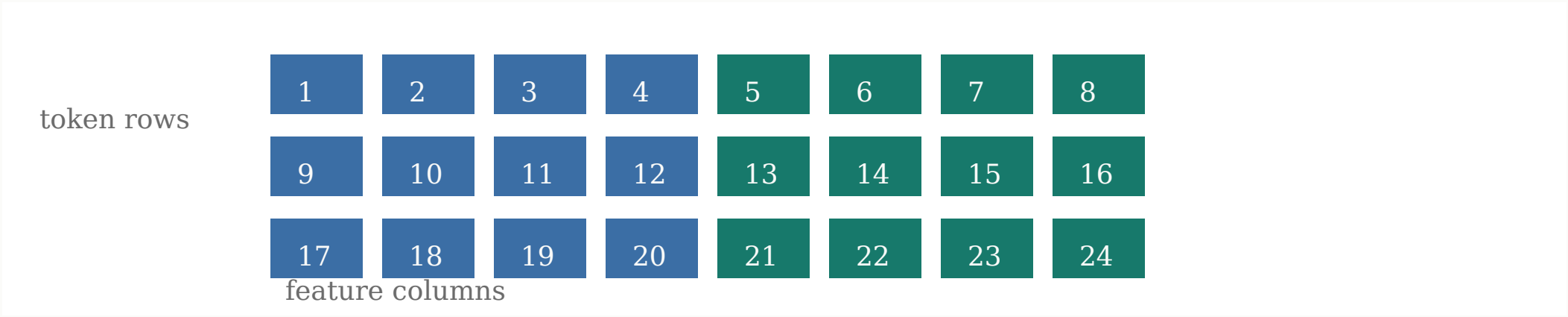
Tokens and projections

A token is a feature vector

$$X = [x_1 \ \cdots \ x_T]^T \in \mathbb{R}^{T \times D}$$

x_i column feature vector in $\mathbb{R}^D$

T, D tokens, features per token



Rows identify tokens; columns identify features.

Vaswani et al. (2017), §3.2.1, Eq. 1

DERIVATION

Tokens and projections

Column weights and row-stored tokens

$$q_i = W_Q x_i \quad \Rightarrow \quad Q = X W_Q^T$$

W_Q learned $d_k \times D$ map

Q $T \times d_k$ query matrix

Vaswani et al. (2017), §3.2.1, Eq. 1

DEFINITION

Tokens and projections

Three learned views of the same input

$$Q = XW_Q^T, \quad K = XW_K^T, \quad V = XW_V^T$$

Q, K T × d_k: queries and keys

V T × d_v: values

A query asks. A key describes what can be matched. A value supplies the information to combine.

Vaswani et al. (2017), §3.2.1, Eq. 1

WORKED EXAMPLE

Tokens and projections

A projection computed by hand

$$X = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 1 & 3 \end{bmatrix}, \quad W_Q = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & -1 \end{bmatrix}$$

X two tokens with three features

W_Q two output coordinates

$$Q = XW_Q^T = \begin{bmatrix} 0 & 5 \\ 3 & -1 \end{bmatrix}$$

Vaswani et al. (2017), §3.2.1, Eq. 1 · Course example

THINK, EXPLAIN, COMPARE

Tokens and projections

A new token, the same learned map

$x = (2, -1, 1)^T$, $W_Q = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & -1 \end{bmatrix}$. Find $q = W_Q x$.

A $(3, -3)^T$

B $(1, -1)^T$

C A three-coordinate vector

Vaswani et al. (2017), §3.2.1, Eq. 1 · Course example

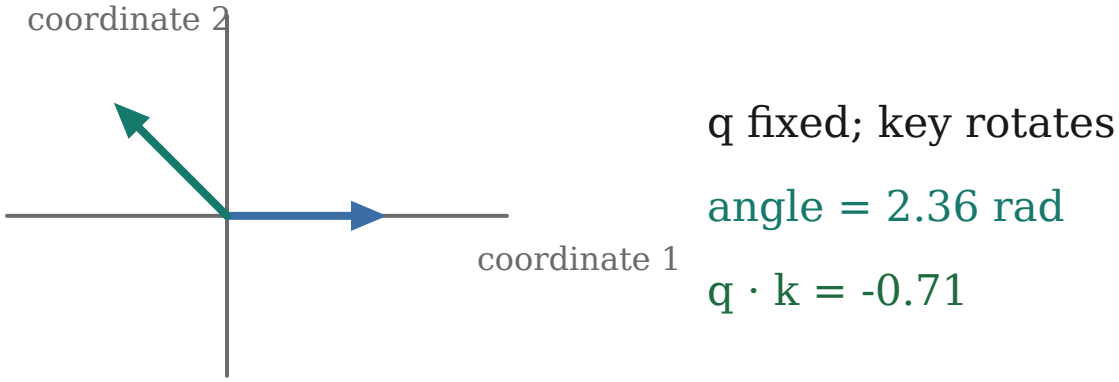
DEFINITION

Scaled dot-product attention

A score compares a query with a key

$$s_{ij} = \frac{q_i^T k_j}{\sqrt{d_k}}$$

- i, j query index, key index
- q_i, k_j d_k-dimensional columns
- s_{ij} scalar compatibility score



The key direction changes; the query stays fixed.

Vaswani et al. (2017), §3.2.1, Eq. 1

THINK, EXPLAIN, COMPARE

Scaled dot-product attention

Alignment and length both affect the score

$q = (1, 0)^T$, $k_1 = (1, 0)^T$, $k_2 = (2, 0)^T$, $d_k = 2$. Which key has the larger score?

A Key 1

B Key 2

C They must tie

Vaswani et al. (2017), §3.2.1, Eq. 1 · Course example

DERIVATION

Scaled dot-product attention

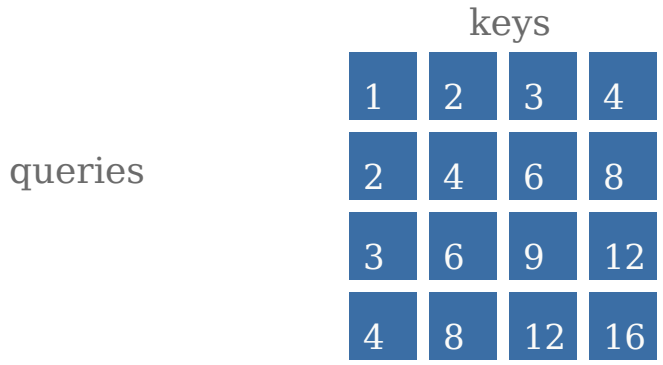
All query-key pairs in one matrix

$$S = \frac{QK^T}{\sqrt{d_k}} \in \mathbb{R}^{T_q \times T_k}$$

Q Tq × d_k query rows

K Tk × d_k key rows

S_{ij} query i versus key j



one scalar per query-key pair

Q = K = $(1,2,3,4)^T$; $d_k = 1$

Each row compares one query with all keys.

Vaswani et al. (2017), §3.2.1, Eq. 1

DEFINITION

Scaled dot-product attention

Weights combine values

$$P_{ij} = \frac{e^{S_{ij}}}{\sum_{\ell} e^{S_{i\ell}}}, \qquad y_i = \sum_j P_{ij} v_j$$

P_{ij} weight of key j for query i

v_j, y_i d_v-dimensional value and output

Vaswani et al. (2017), §3.2.1, Eq. 1

WORKED EXAMPLE

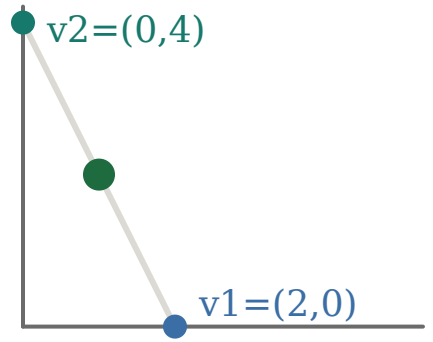
Scaled dot-product attention

Equal scores give an ordinary average

$$s = (0, 0), \quad v_1 = (2, 0)^T, \quad v_2 = (0, 4)^T$$

$s; v_1, v_2$ one query's scores; two value vectors

$p = (1/2, 1/2)$, so $y = (1, 2)^T$.



weight of v2 = 0.50

y = (1.00, 2.00)

value coordinate 1

Equal weights place the output halfway between the value vectors.

Vaswani et al. (2017), §3.2.1, Eq. 1 · Course example

THINK, EXPLAIN, COMPARE

Scaled dot-product attention

Unequal scores change the mixture

$s = (0, \log 3)$, $v_1 = (2, 0)^T$, $v_2 = (0, 4)^T$. Find the output.

A $(1, 2)^T$

B $(1/2, 3)^T$

C $(0, 4)^T$

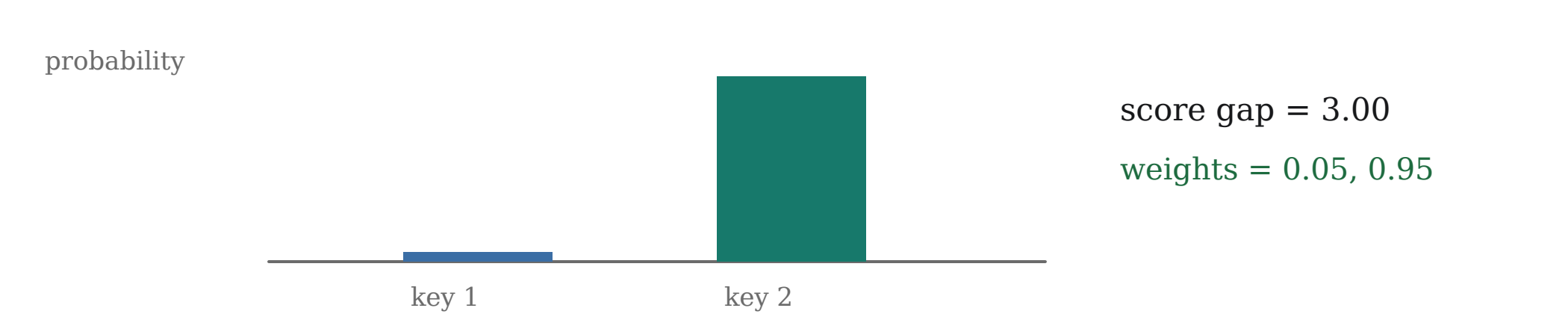
Vaswani et al. (2017), §3.2.1, Eq. 1 · Course example

MOTIVATION

Scaled dot-product attention

Why divide by the square root?

Wider query-key vectors can produce more variable dot products. Very unequal scores make softmax nearly one-hot.



Scaling a fixed score gap changes the softmax weights.

Vaswani et al. (2017), §3.2.1, Eq. 1

DERIVATION

Scaled dot-product attention

Variance of one product

$$\mathbb{E}[q_r k_r] = 0, \quad \mathrm{Var}(q_r k_r) = \mathbb{E}[q_r^2]\mathbb{E}[k_r^2] = 1$$

q_r, k_r independent, mean 0, variance 1

Vaswani et al. (2017), §3.2.1, Eq. 1

DERIVATION

Scaled dot-product attention

Variance of the sum

$$\mathrm{Var}(q^T k) = d_k, \qquad \mathrm{Var}\left(\frac{q^T k}{\sqrt{d_k}}\right) = 1$$

d_k number of independent coordinate products

Vaswani et al. (2017), §3.2.1, Eq. 1

THINK, EXPLAIN, COMPARE

Scaled dot-product attention

The scale is determined by the assumptions

Independent zero-mean, unit-variance coordinates; $d_k = 16$. What is the standard deviation of the unscaled dot product?

A 16

B 4

C 1

Vaswani et al. (2017), §3.2.1, Eq. 1 · Course example

DERIVATION

Stable attention and masks

The largest logit can be subtracted

$$\frac{e^{s_j - c}}{\sum_\ell e^{s_\ell - c}} = \frac{e^{-c} e^{s_j}}{e^{-c} \sum_\ell e^{s_\ell}} = p_j$$

- s_j logit: score before softmax
- c same shift for every entry in the row

Blanchard, Higham & Higham (2021), §4

WORKED EXAMPLE

Stable attention and masks

A large-logit example

$$s = (1000, 1000 + \log 3), \quad s - \max s = (-\log 3, 0)$$

s one score row

$e^{s-\max s} = (1/3, 1)$; normalize to $(1/4, 3/4)$.

Blanchard, Higham & Higham (2021), §4 · Course example

THINK, EXPLAIN, COMPARE

Stable attention and masks

A shift or a temperature?

A score row is $s = (0, \log 3)$. Which change preserves its softmax weights?

A. Subtract 5 from both entries

B. Multiply both entries by 2

C. Add 5 only to the second entry

Blanchard, Higham & Higham (2021), §4 · Course example

DEFINITION

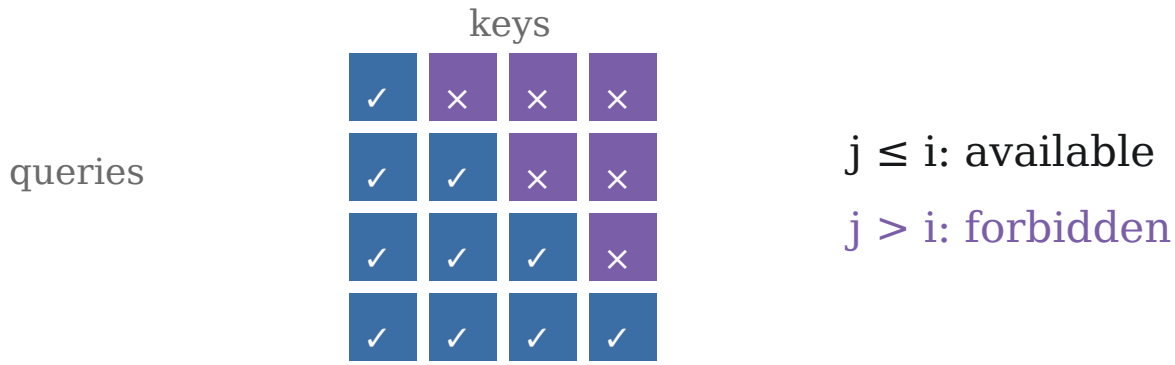
Stable attention and masks

A causal mask forbids future keys

$$M_{ij} = \begin{cases} 0 & j \le i \\ -\infty & j > i \end{cases}, \qquad P = \mathrm{softmax}_{\mathrm{row}}(S + M)$$

i, j query position, key position

M additive causal mask



Rows are queries; columns are keys. The upper triangle is forbidden.

Vaswani et al. (2017), §3.2.1, Eq. 1

DEFINITION

Stable attention and masks

Padding and causality answer different questions

$$S : [B, H, T_q, T_k], \quad V_k : [B, 1, 1, T_k], \quad C : [1, 1, T_q, T_k]$$

V_k, C Boolean key-valid and causal masks; True means allowed

Broadcast $V_k \wedge C$ across heads and queries. Query validity $V_q : [B, 1, T_q, 1]$ is checked separately.

Vaswani et al. (2017), §3.2.1, Eq. 1

THINK, EXPLAIN, COMPARE

Stable attention and masks

Mask before normalizing

Scores are $(0, \log 3)$; the second key is forbidden. Which output weights are correct?

A $(1, 0)$

B $(1/4, 0)$

C $(1/4, 3/4)$

Vaswani et al. (2017), §3.2.1, Eq. 1 · Course example

BOUNDARY CASE

Stable attention and masks

Every valid query needs an allowed key

If every logit is $-\infty$, softmax has denominator zero. Subtracting the row maximum also produces $-\infty - (-\infty)$.

Vaswani et al. (2017), §3.2.1, Eq. 1

THINK, EXPLAIN, COMPARE

Stable attention and masks

Diagnose the NaN without changing the task

A padded query has no allowed key. Stable softmax returns NaN. Which repair preserves the declared information rule?

A. Use uniform weights over all keys

B. Mark the query invalid and output zero

C. Remove all masks for the batch

Vaswani et al. (2017), §3.2.1, Eq. 1 · Course example

INTEGRATED WORKED EXAMPLE

Stable attention and masks

One attention query, end to end

$$q = (\sqrt{2}, 0)^T, \quad K = \begin{bmatrix} 0 & 1 \\ \log 3 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$$

$d_k = 2$ two keys, both permitted

Vaswani et al. (2017), §3.2.1, Eq. 1 · Course example

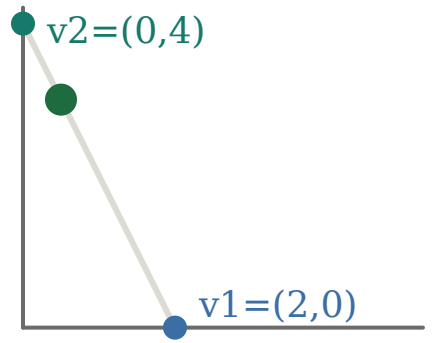
WORKED EXAMPLE

Stable attention and masks

The output follows the same data path

$$s = (0, \log 3) \Rightarrow p = (1/4, 3/4) \Rightarrow y = (1/2, 3)^T$$

- s scaled scores
- p row weights
- y mixture of $(2,0)^T$ and $(0,4)^T$



weight of v2 = 0.75

y = (0.50, 3.00)

Weights (1/4, 3/4) place the output at (1/2, 3).

Vaswani et al. (2017), §3.2.1, Eq. 1 · Course example

THINK, EXPLAIN, COMPARE

Stable attention and masks

Transfer: change the permitted information

$q = (\sqrt{2}, 0)^T$, $k_1 = (0, 1)^T$, $k_2 = (\log 3, 0)^T$; $v_1 = (2, 0)^T$, $v_2 = (0, 4)^T$. Only key 1 is allowed.

A $(1/2, 3)^T$

B $(2, 0)^T$

C $(0, 0)^T$

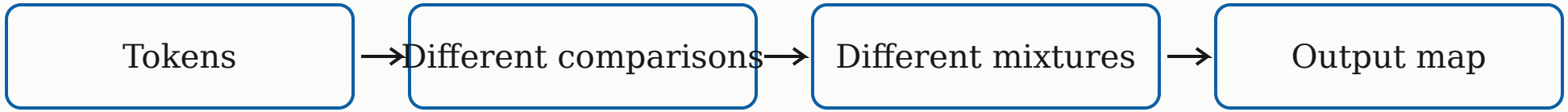
Vaswani et al. (2017), §3.2.1, Eq. 1 · Course example

MOTIVATION

Multi-head attention

One mixture can hide useful differences

Several heads learn different comparisons and preserve several value mixtures before combining them.



Vaswani et al. (2017), §3.2.2

DEFINITION

Multi-head attention

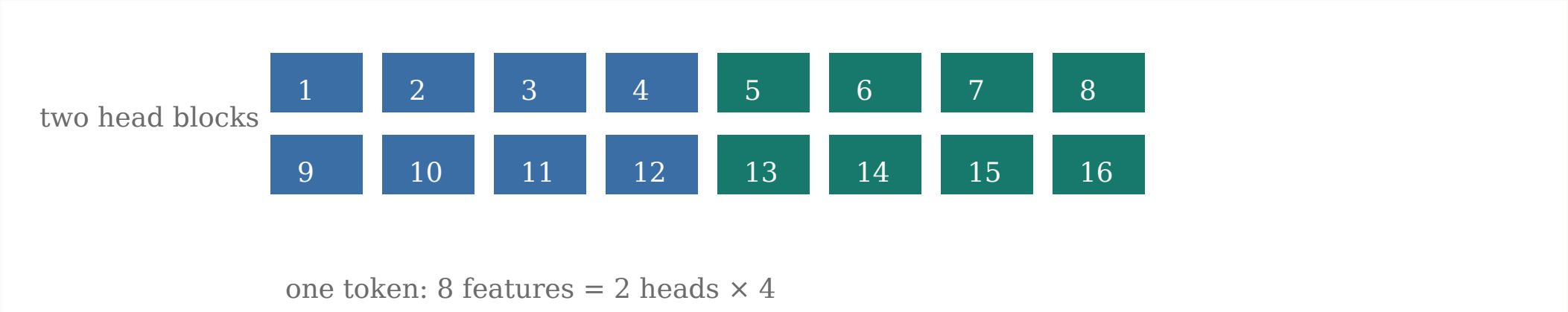
A feature axis can be split into heads

$$Q : [B, T, D] \to [B, T, H, d_h], \qquad D = H d_h$$

B, T batch size, tokens

H, d_h heads, features per head

```
Q = (X @ Wq.T).reshape(B,T,H,dh)
```



A packed feature row splits into two head blocks.

Vaswani et al. (2017), §3.2.2

DERIVATION

Multi-head attention

Transpose puts matrix axes last

$$[B, T, H, d_h] \to [B, H, T, d_h], \quad S : [B, H, T, T]$$

T, d_h last two axes used by matrix multiplication

Vaswani et al. (2017), §3.2.2

THINK, EXPLAIN, COMPARE

Multi-head attention

Trace a tensor through the heads

$B = 2$, $T = 3$, $D = 8$, $H = 2$. Find query and score shapes after splitting and transposing.

A Q [2,2,3,4]; S [2,2,3,3]

B Q [2,3,2,4]; S [2,3,2,2]

C Q [2,2,3,8]; S [2,2,8,8]

Vaswani et al. (2017), §3.2.2 · Course example

DEFINITION

Multi-head attention

Concatenate, then mix across heads

$$C = \mathrm{Concat}(Y^{(1)}, \ldots, Y^{(H)}), \quad Y = CW_O^T$$

C T × $\mathrm{Hd_v}$ concatenated outputs

W_O D × $\mathrm{Hd_v}$ learned output map

```
C = Y.transpose(1,2).reshape(B,T,H*dv)
out = C @ Wo.T
```

Vaswani et al. (2017), §3.2.2

WORKED EXAMPLE

Multi-head attention

Two heads, one explicit output map

$$c = (1, 2, 3, -1)^T, \quad W_O = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

c head 1: (1,2), head 2: (3,−1)

$W_O c = (1 + 3, 2 - 1)^T = (4, 1)^T.$

Vaswani et al. (2017), §3.2.2 · Course example

THINK, EXPLAIN, COMPARE

Multi-head attention

Transfer: a different output map

$c = (1, 2, 3, -1)^T$, $W_O = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}$. Find $W_O c$.

A $(3, 2)^T$

B $(4, 1)^T$

C $(1, 2, 3, -1)^T$

Vaswani et al. (2017), §3.2.2 · Course example

IMPLEMENTATION

Multi-head attention

The forward pass follows the equations

```
Q,K,V: [B,H,T,dₕ]
S = Q @ K.transpose(-2,-1) / sqrt(dₕ)
S = masked_fill(S, forbidden, -inf)
P = softmax(S, dim=-1)
Y = P @ V
```

Vaswani et al. (2017), §3.2.2

THINK, EXPLAIN, COMPARE

Multi-head attention

Correct shapes can hide a wrong normalization

Scores have shape [B,H,Tq,Tk]. A programmer uses `softmax(S,dim=-2)`. What is normalized?

A Keys for each query

B Queries for each key

C Features of each token

Vaswani et al. (2017), §3.2.2 · Course example

THINK, EXPLAIN, COMPARE

Multi-head attention

Transfer: broadcast a key-validity mask

Scores have shape [2,3,4,5]: batch, heads, queries, keys. Key validity has shape [2,5]. Which reshape shares the key rule across every head and query?

A [2,1,1,5]

B [2,5,1,1]

C [1,1,2,5]

Vaswani et al. (2017), §3.2.2 · Course example

DERIVATION

Position and order

Content-only attention follows a permutation

$$\mathrm{Attn}(\Pi X) = \Pi\,\mathrm{Attn}(X)$$

- Π token permutation; no positional terms or fixed mask

Vaswani et al. (2017), §3.5

THINK, EXPLAIN, COMPARE

Position and order

What does content alone fail to identify?

The same two feature vectors are presented in opposite orders to unmasked attention with no position features.

A. Dot products recover their original order

B. Outputs move with the input permutation

C. Every output becomes zero

Vaswani et al. (2017), §3.5 · Course example

DEFINITION

Position and order

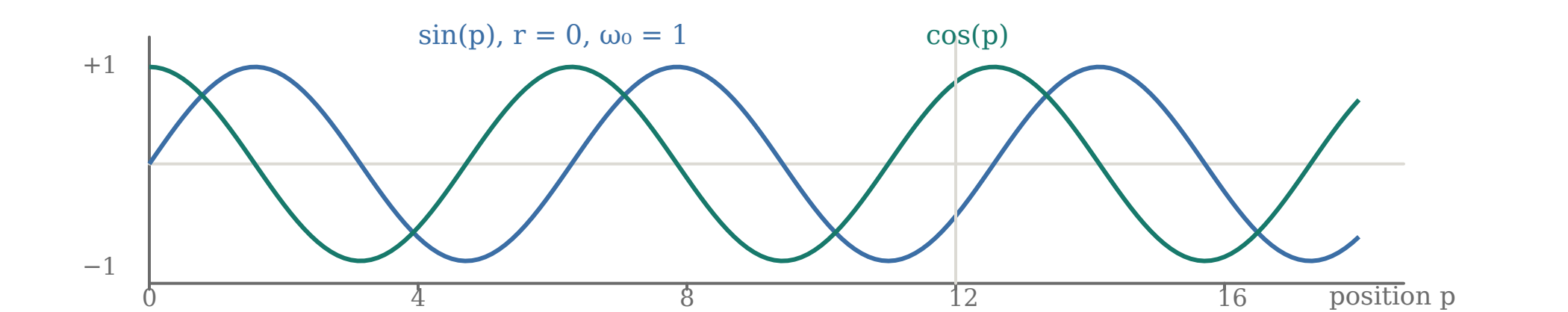
Sinusoids give each position a pattern

$$\mathrm{PE}_{p,2r} = \sin(p\omega_r), \quad \mathrm{PE}_{p,2r+1} = \cos(p\omega_r)$$

p token position

$\omega_r = 10000^{-2r/D}$ frequency of coordinate pair r

Add the pattern to the token features before projection.



One sine/cosine pair across token positions.

Vaswani et al. (2017), §3.5

DERIVATION

Position and order

A shift follows angle-addition identities

$$\begin{bmatrix}\sin((p+\Delta)\omega)\\ \cos((p+\Delta)\omega)\end{bmatrix} = \begin{bmatrix}\cos(\Delta\omega) & \sin(\Delta\omega)\\ -\sin(\Delta\omega) & \cos(\Delta\omega)\end{bmatrix}\begin{bmatrix}\sin(p\omega)\\ \cos(p\omega)\end{bmatrix}$$

p, Δ, ω position, offset, one frequency

Vaswani et al. (2017), §3.5

THINK, EXPLAIN, COMPARE

Position and order

Transfer: one sinusoidal pair at a new position

One coordinate pair is $(\sin(p\omega), \cos(p\omega))$. Take $\omega = \pi/2$ and move from p=0 to p=1.

A $(1,0)$

B $(0,1)$

C $(-1,0)$

Vaswani et al. (2017), §3.5 · Course example

DEFINITION

Position and order

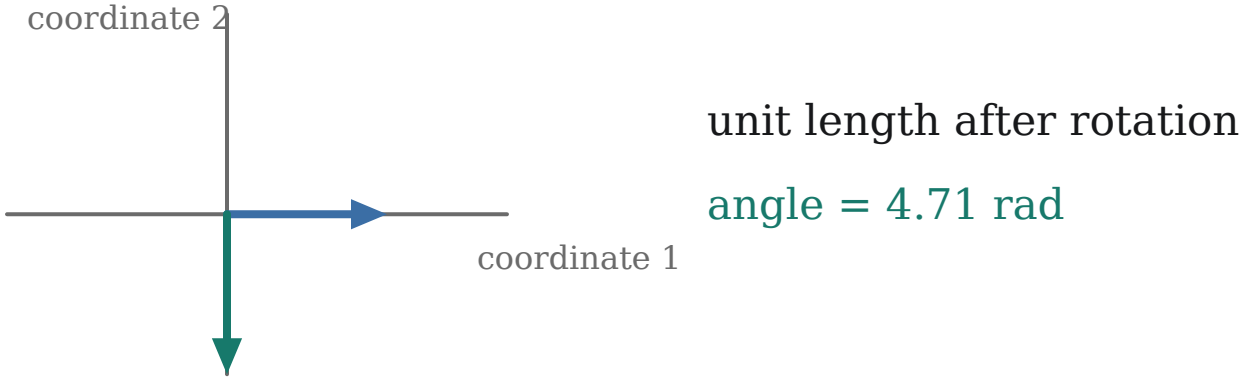
Rotary embeddings act inside the comparison

$$\tilde{q}_p = R(p\omega)q_p, \quad \widetilde{k}_s = R(s\omega)k_s$$

$R(\theta)$ 2D rotation by θ

p, s query and key positions

Rotate each coordinate pair at its own frequency. Values need not rotate.



A query pair rotates with position while keeping its length.

Su et al., RoFormer (2021), §3.2

DERIVATION

Position and order

Relative positions appear in the dot product

$$\tilde{q}_p^T \tilde{k}_s = q_p^T R(p\omega)^T R(s\omega) k_s = q_p^T R((s-p)\omega) k_s$$

$s - p$ key position relative to the query

Su et al., RoFormer (2021), §3.2

WORKED EXAMPLE

Position and order

A quarter-turn example

$$q = k = (1, 0)^T, \quad p = 0, \quad s = 1, \quad \omega = \pi/2$$

p, s positions

ω teaching example's angle per position

$\tilde{q} = (1, 0)^T$, $\widetilde{k} = (0, 1)^T$, so their dot product is 0.

Su et al., RoFormer (2021), §3.2 · Course example

THINK, EXPLAIN, COMPARE

Position and order

Transfer: shift both positions

$q = k = (1, 0)^T$, $\omega = \pi/2$. Change (p, s) from $(0, 1)$ to $(2, 3)$. Find the raw dot product.

A 0

B 1

C -1

Su et al., RoFormer (2021), §3.2 · Course example

DEFINITION

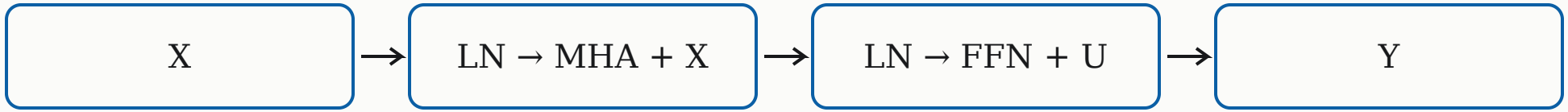
Transformer blocks

The pre-LN block has two residual paths

$$U = X + \mathrm{MHA}(\mathrm{LN}(X)), \quad Y = U + \mathrm{FFN}(\mathrm{LN}(U))$$

X, U, Y T × D representations

LN LayerNorm within each token



Xiong et al. (2020), §2, Fig. 1

RECALL

Transformer blocks

LayerNorm uses the token feature axis

$$\hat{x}_r = \frac{x_r - \mu}{\sqrt{v + \epsilon}}, \quad \mu = \frac{1}{D}\sum_r x_r, \quad v = \frac{1}{D}\sum_r (x_r - \mu)^2$$

x one token's D features

$\epsilon > 0$ stability constant

Output: $\gamma \odot \hat{x} + \beta$, with learned D-coordinate scale and shift.

Xiong et al. (2020), §2, Fig. 1

THINK, EXPLAIN, COMPARE

Transformer blocks

Transfer: normalize one token

One token is $x = (1, 3)$. Use $\gamma = (1, 1)$, $\beta = (0, 0)$ and $\epsilon = 0$ for this nonconstant numerical example.

A $(-1, 1)$

B $(-1/\sqrt{2},\ 1/\sqrt{2})$

C $(1, 3)$

Xiong et al. (2020), §2, Fig. 1 · Course example

DEFINITION

Transformer blocks

The feed-forward branch transforms each token

$$f(x) = W_2\phi(W_1x + b_1) + b_2$$

W_1, W_2 F × D and D × F maps

ϕ coordinatewise nonlinearity

b_1, b_2 F- and D-coordinate biases

Vaswani et al. (2017), §3.3

THINK, EXPLAIN, COMPARE

Transformer blocks

A residual connection constrains the shape

X has shape [2,5,12], while the attention branch returns [2,5,8]. Can they be directly added?

A. Yes, ordinary broadcasting repairs it

B. No, the branch must return width 12

C. Yes, because both have five tokens

Xiong et al. (2020), §2, Fig. 1 · Course example

THINK, EXPLAIN, COMPARE

Transformer blocks

Transfer: a missing path changes the architecture

$Y = X + FFN(MHA(X))$ is described as the pre-LN block. What is missing?

A Only position information

B Both normalizations and the intermediate residual

C A final softmax only

Xiong et al. (2020), §2, Fig. 1 · Course example

DERIVATION

Cost and the KV cache

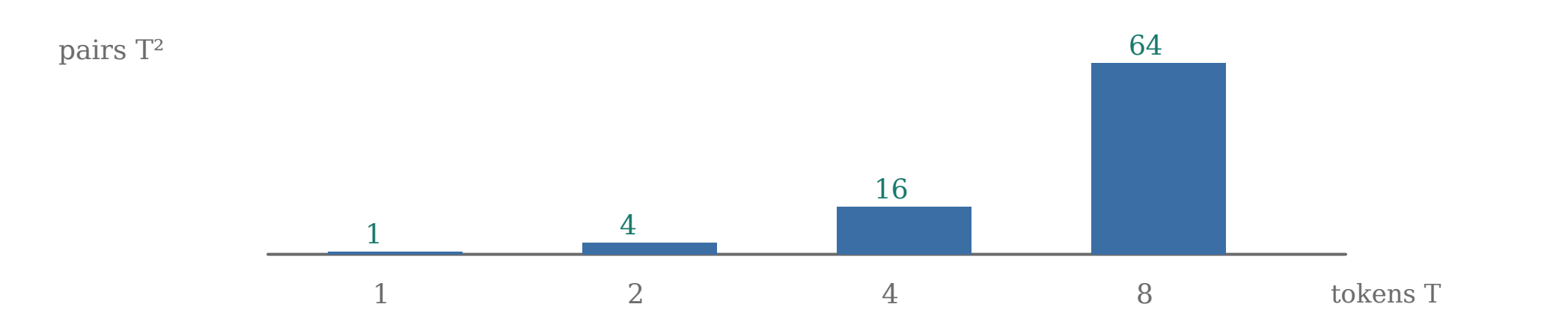
Pairwise comparison creates quadratic work

$$QK^T : O(T^2 d_h), \quad PV : O(T^2 d_h), \quad H d_h = D$$

T, D tokens, total feature width

H heads

Mixing: $O(T^2 D)$. Projections also cost $O(T D^2)$.



Query-key pair counts grow quadratically with token length.

Vaswani et al. (2017), §3.2.2

WORKED EXAMPLE

Cost and the KV cache

Memory follows the score shape

$$B = 1,\ H = 8,\ T = 1024: \quad BHT^2 = 8{,}388{,}608$$

B, H, T batch, heads, tokens

At 2 bytes per entry: 16 MiB for one materialized score tensor.

Vaswani et al. (2017), §3.2.2 · Course example

THINK, EXPLAIN, COMPARE

Cost and the KV cache

Transfer: a longer sequence

Keep B, H and datatype fixed. Double T. How does the materialized score-tensor memory change?

A ×2

B ×4

C Unchanged

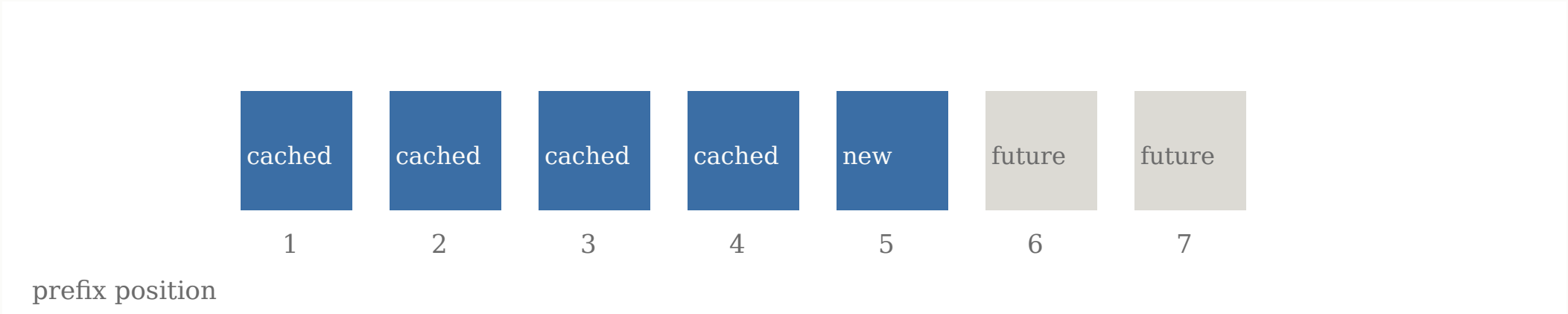
Vaswani et al. (2017), §3.2.2 · Course example

DEFINITION

Cost and the KV cache

Unchanged causal prefixes permit reuse

Earlier causal hidden states cannot read a newly appended token. Their keys and values can be cached per layer.



An unchanged prefix stays cached while one new row is appended.

Shazeer (2019), §2, incremental attention

DERIVATION

Cost and the KV cache

One new query reads all cached keys

$$q_t^T K_{1:t}^T : \ [1, d_h][d_h, t] = [1, t]$$

q_t new query column

$K_{1:t}$ t cached key rows

Per-step mixing: $O(tD)$. Cache storage: $O(tD)$ per layer and batch item.

```
K = concat([Kold, knew], axis=-2)
V = concat([Vold, vnew], axis=-2)
```

Shazeer (2019), §2, incremental attention

THINK, EXPLAIN, COMPARE

Cost and the KV cache

Transfer: can this memory be reused?

Camera tokens are replaced by a new frame. Can their old cross-attention keys and values serve as exact representations of the new frame?

A Yes, when shapes match

B No, recompute the changed memory

C Yes, if action horizon is fixed

Shazeer (2019), §2, incremental attention · Course example

WORKED EXAMPLE

Cost and the KV cache

Cache validity and cumulative decoding cost

$$\sum_{t=1}^{T} t = \frac{T(T+1)}{2}, \quad \sum_{t=1}^{T} t^2 = \frac{T(T+1)(2T+1)}{6}$$

t, T current and final prefix length

Cached mixing $O(T^2 D)$; naïve repeated full-prefix mixing $O(T^3 D)$.

Shazeer (2019), §2, incremental attention · Course example

THINK, EXPLAIN, COMPARE

Cost and the KV cache

Transfer: count incremental comparisons

Generate four tokens from an empty prefix. Count query-key comparisons for one head, including each current token. Compare cached decoding with full-prefix recomputation.

A Cached 10; recomputed 30

B Cached 4; recomputed 16

C Cached 16; recomputed 16

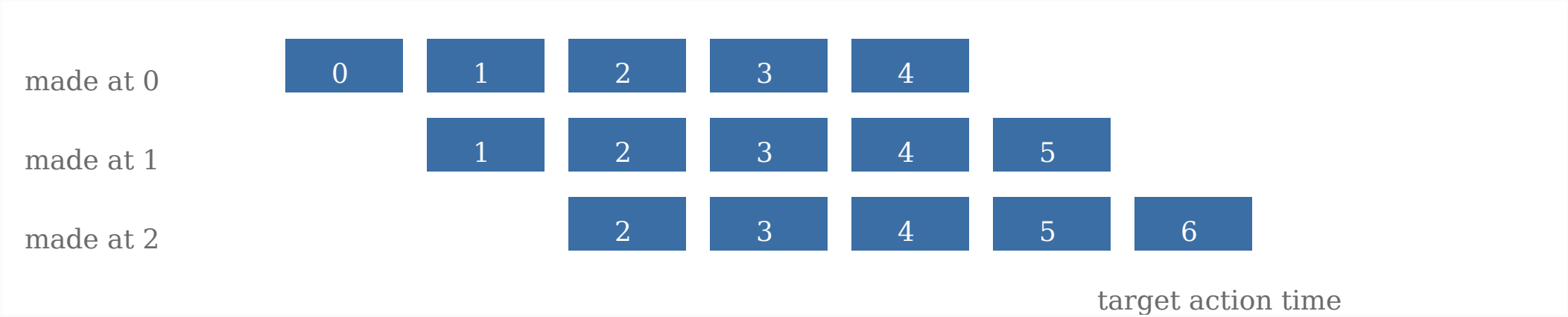
Shazeer (2019), §2, incremental attention · Course example

MOTIVATION

Action chunking

A coordinated movement spans several decisions

A policy can predict a short future action sequence from the current observation, then gather new observations as execution continues.



Each row is a chunk prediction; columns are target action times.

Zhao et al. (2023), §IV-A

DEFINITION

Action chunking

Prediction time and target time are different indices

$$\widehat{A}_t \in \mathbb{R}^{K \times d_a}, \quad \widehat{A}_t[j, :] = \hat{a}^T_{t+j|t}, \quad 0 \le j < K$$

t observation/prediction time

K, d_a chunk length, action dimension

Zhao et al. (2023), §IV-A

THINK, EXPLAIN, COMPARE

Action chunking

Chunk length is not execution length

A policy predicts K=6 future actions and executes E=2 before reading a new observation. Which targets from the current chunk are executed?

A. Times t through t+5

B. Times t and t+1

C. Only time t+5

Zhao et al. (2023), §IV-A · Course example

DEFINITION

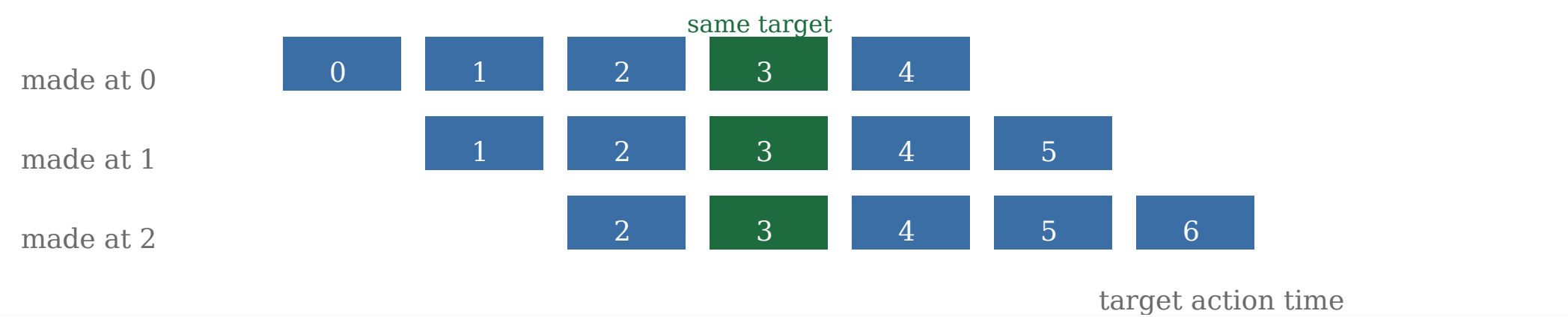
Temporal ensembling

Overlapping chunks predict the same target

$$\bar{a}_s = \frac{\sum_{j=0}^{J-1} w_j \hat{a}_{s|r_j}}{\sum_{j=0}^{J-1} w_j}, \qquad w_j = e^{-mj}$$

$r_0 < \cdots < r_{J-1}$ prediction times, oldest first

s, m common target time, nonnegative weight parameter



Combine one target-time column across overlapping chunks.

Zhao et al. (2023), §IV-A

WORKED EXAMPLE

Temporal ensembling

One target, two predictions

$$\hat{a}_{s|r_0} = 2, \quad \hat{a}_{s|r_1} = 4, \quad m = \log 2$$

$r_0 < r_1$ oldest then newest prediction

s same target time

$w = (1, 1/2)$, so $\bar{a}_s = (2 + 2)/(3/2) = 8/3$.

Zhao et al. (2023), §IV-A · Course example

THINK, EXPLAIN, COMPARE

Temporal ensembling

Transfer: three estimates for one target

Oldest-to-newest predictions for target s: 1, 3, 5. Let $m = \log 2$. Find the ensemble.

A 3

B 15/7

C 27/7

Zhao et al. (2023), §IV-A · Course example

INTERPRETATION

Temporal ensembling

Averages need compatible action representations

Average predictions for the same target time in the same coordinate system.

Smoothing alone does not enforce collision, dynamics or joint-limit constraints.

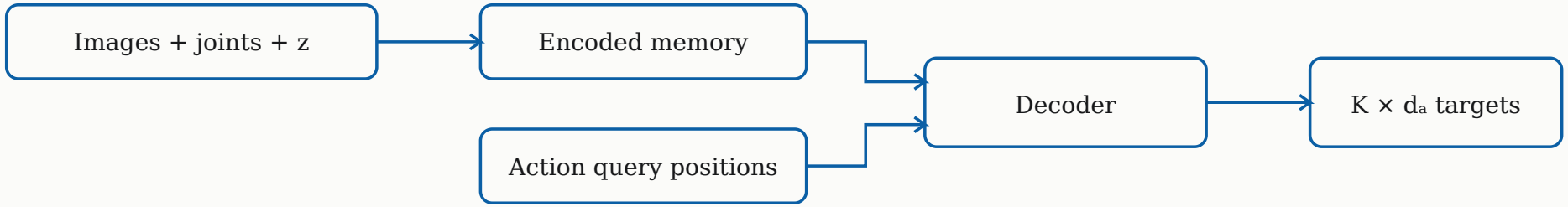
Zhao et al. (2023), §IV-A

DEFINITION

ACT: observations to actions

ACT combines observation tokens with action queries

Action Chunking with Transformers (ACT): encode image features, joint state and a latent variable; decode K ordered action queries into K motor targets.



Zhao et al. (2023), §IV-B–C, Appendix C

WORKED EXAMPLE

ACT: observations to actions

A concrete observation-token count

$$4(15 \cdot 20) + 2 = 1202$$

4	camera views in the ACT paper
15×20	feature grid per view
$+2$	joint-state and latent tokens

Each token has 512 features in the reported model. Spatial positions accompany image features.

Zhao et al. (2023), §IV-B–C, Appendix C · Course example

DEFINITION

ACT: observations to actions

Action queries use cross-attention

$$Q_a \in \mathbb{R}^{K \times d_k}, \quad K_o \in \mathbb{R}^{T_o \times d_k}, \quad S_a \in \mathbb{R}^{K \times T_o}$$

K, T_o action queries, observation tokens

Zhao et al. (2023), §IV-B–C, Appendix C

THINK, EXPLAIN, COMPARE

ACT: observations to actions

Transfer: encoder tokens and action slots

A model has 20 observation tokens and predicts a chunk of 6 actions. Each action query reads every observation token. What is one head’s cross-attention score shape?

A. 20×20

B. 6×20

C. 6×6

Zhao et al. (2023), §IV-B–C, Appendix C · Course example

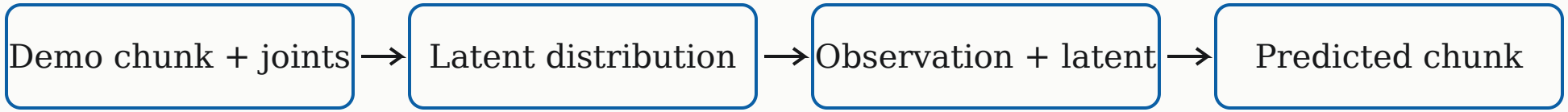
MOTIVATION

ACT: latent-variable training

Demonstrations can differ at the same observation

A latent variable z can represent variation among demonstrated action chunks.

Training infers z using the demonstration; deployment follows a separate rule.



Zhao et al. (2023), §IV-B–C, Appendix C

DEFINITION

ACT: latent-variable training

The encoder predicts a diagonal Gaussian

$$q_\phi(z \mid A_t, r_t) = \mathcal{N}(\mu, \mathrm{diag}(\sigma^2)), \quad p(z) = \mathcal{N}(0, I)$$

- μ, σ latent mean and positive standard deviation
- r_t joint observation; A_t is the demonstration

Kingma & Welling (2014), §2, Appendix B

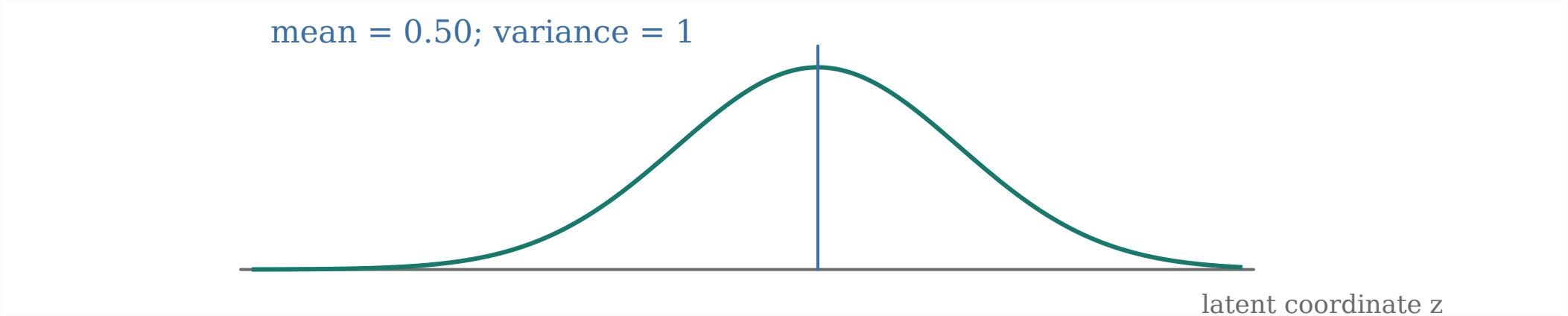
DERIVATION

ACT: latent-variable training

Reparameterization exposes a differentiable path

$$\epsilon \sim \mathcal{N}(0, I), \qquad z = \mu + \sigma \odot \epsilon$$

- ϵ parameter-independent random draw
- μ, σ encoder outputs
- z latent sample



Changing the mean shifts a Gaussian without changing its width.

Kingma & Welling (2014), §2, Appendix B

THINK, EXPLAIN, COMPARE

ACT: latent-variable training

Log-variance is not the standard deviation

One latent coordinate has $\mu = 1$, $\ell = \log \sigma^2 = \log 4$, and $\epsilon = -1$. Find z.

A -3

B -1

C 1

Kingma & Welling (2014), §2, Appendix B · Course example

DERIVATION

ACT: latent-variable training

A bound makes the latent objective tractable

$$\log p(A, z \mid o) = \log p(A \mid o) + \log p(z \mid A, o)$$

$$\mathbb{E}_q \log \frac{p(A, z \mid o)}{q(z)} = \log p(A \mid o) - D_{KL}(q \| p(z \mid A, o))$$

p, q model and encoder densities; θ suppressed for space

Rules: Bayes' factorization, expectation, then the KL definition.

Kingma & Welling (2014), §2, Appendix B

DERIVATION

ACT: latent-variable training

Nonnegative KL gives the evidence lower bound

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_q \log \frac{p(A \mid o, z)p(z)}{q(z)} \\ &= \mathbb{E}_q \log p(A \mid o, z) - \mathbb{E}_q \log \frac{q(z)}{p(z)} \\ &= \mathbb{E}_q \log p(A \mid o, z) - D_{KL}(q \| p)\end{aligned}$$

$\mathcal{L}$ ELBO; p(z) is the prior

Nonnegative posterior-gap KL gives $\log p(A \mid o) \geq \mathcal{L}$.

Kingma & Welling (2014), §2, Appendix B

THINK, EXPLAIN, COMPARE

ACT: latent-variable training

Transfer: which divergence creates the gap?

$\log p_\theta(A \mid o) = \mathcal{L}_{ELBO} + G$. Which G makes this an exact identity?

A $D_{KL}(q \| p_\theta(z \mid A, o))$

B $-D_{KL}(q \| p(z))$

C The action prediction error alone

Kingma & Welling (2014), §2, Appendix B · Course example

DEFINITION

ACT: latent-variable training

ACT balances reconstruction and latent regularization

$$\mathcal{L} = \frac{1}{K d_a} \sum_{j,r} |\hat{A}_{jr} - A_{jr}| + \beta D_{KL}(q_\phi \| \mathcal{N}(0, I))$$

- $A, \hat{A}$ target and predicted action chunks
- β chosen regularization weight

Zhao et al. (2023), §IV-B–C, Appendix C

DERIVATION

ACT: latent-variable training

Gaussian KL follows from a log-density ratio

$$\log \frac{q_j(z_j)}{p_j(z_j)} = -\log \sigma_j - \frac{(z_j - \mu_j)^2}{2\sigma_j^2} + \frac{z_j^2}{2}$$

q_j, p_j $\mathrm{N}(\mu_j, \sigma_j^2)$ and $\mathrm{N}(0,1)$ densities

Rule: cancel Gaussian log-density constants. Then $\mathbb{E}_q[(z_j - \mu_j)^2] = \sigma_j^2$ and $\mathbb{E}_q[z_j^2] = \mu_j^2 + \sigma_j^2$.

Kingma & Welling (2014), §2, Appendix B

DERIVATION

ACT: latent-variable training

The diagonal-Gaussian KL is explicit

$$\mathbb{E}_q \log(q_j/p_j) = -\log\sigma_j - \frac{1}{2} + \frac{1}{2}(\mu_j^2 + \sigma_j^2)$$

$$D_{KL}(q\|p) = \frac{1}{2}\sum_j (\mu_j^2 + \sigma_j^2 - 1 - \log\sigma_j^2)$$

j independent Gaussian latent coordinate

Rules: substitute the two moments; add coordinate log ratios.

Kingma & Welling (2014), §2, Appendix B

WORKED EXAMPLE

ACT: latent-variable training

A latent penalty calculated by hand

$$\mu = (1, 0)^T, \quad \sigma^2 = (1, 4)^T$$

μ, σ^2 two encoder-coordinate means and variances

$D_{KL} = \frac{1}{2}[1 + (4 - 1 - \log 4)] = 2 - \log 2 \approx 1.3069.$

Kingma & Welling (2014), §2, Appendix B · Course example

THINK, EXPLAIN, COMPARE

ACT: latent-variable training

Transfer: combine the two loss terms

Four action errors are $1, -1, 0, 2$. A one-dimensional encoder has $\mu = 1$, $\sigma^2 = 1$. Use mean L1 and $\beta = 2$. Find total loss.

A 1

B 2

C 5

Zhao et al. (2023), §IV-B–C, Appendix C · Course example

DEFINITION

ACT: deployment

The deployed ACT policy fixes the latent to zero

$$\hat{A}_t = f_\theta(o_t, z = 0)$$

o_t current image and joint observations

$z = 0$ mean of the standard Gaussian prior

The demonstration-conditioned encoder is discarded at inference.

Zhao et al. (2023), §IV-B–C, Appendix C

THINK, EXPLAIN, COMPARE

ACT: deployment

Training and inference use different information

Which input route matches the reported ACT inference procedure?

A Encode the unknown future target chunk

B Current observations plus z=0

C Draw z from the demonstration posterior

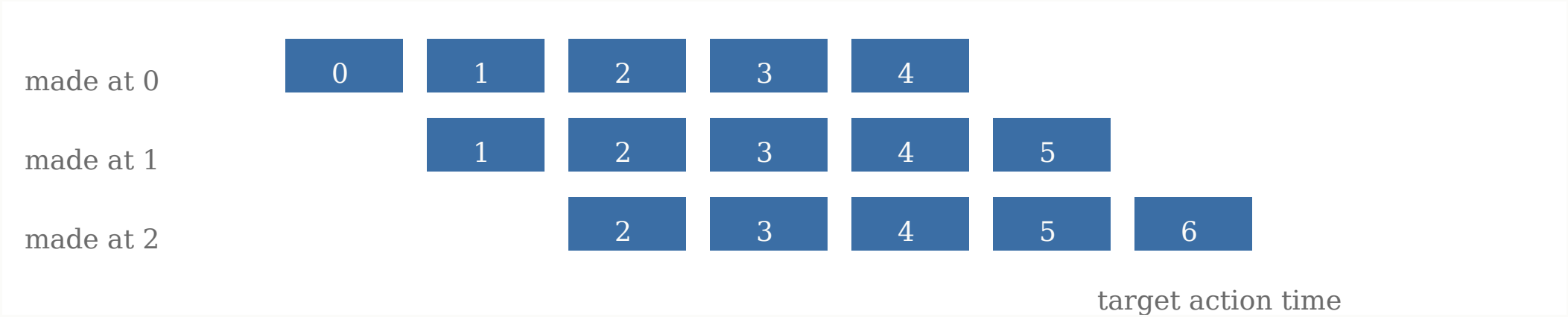
Zhao et al. (2023), §IV-B–C, Appendix C · Course example

INTEGRATED SYNTHESIS

ACT: deployment

One complete robot-policy calculation

Observe → encode → attend → predict a chunk → select same-time predictions → execute → observe again.



Replanning updates future chunks as fresh observations arrive.

Zhao et al. (2023), §IV-B–C, Appendix C

THINK, EXPLAIN, COMPARE

ACT: deployment

Transfer: trace the live policy loop

At target time s, two chunks predict actions 2 and 4, oldest first. Use $m = \log 2$. A fresh image arrives after execution. What is the executed target and the next computation?

A. Execute 8/3; encode the fresh observation

B. Execute 3; change β

C. Execute 4; reuse all old image features

Zhao et al. (2023), §IV-B–C, Appendix C · Course example

THINK, EXPLAIN, COMPARE

ACT: deployment

Transfer: a changed scene after prediction

A chunk predicts six actions. After one executed action, the object moves. Which response uses the new observation?

A. Keep executing all old targets without recomputing

B. Encode the new observation and predict a new chunk

C. Change the KL coefficient at inference

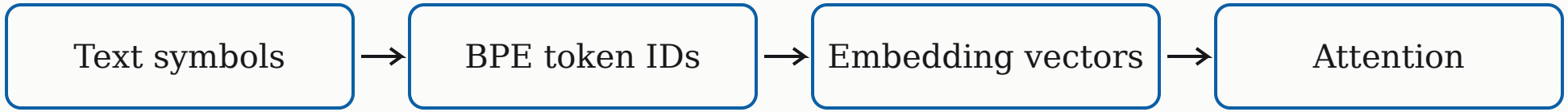
Zhao et al. (2023), §IV-B–C, Appendix C · Course example

BRIDGE

Discrete tokens and continuous actions

Byte-pair encoding is a text representation choice

Byte-pair encoding (BPE) builds a subword vocabulary by repeatedly merging frequent adjacent symbol pairs. A tokenizer maps text to discrete token IDs.



Sennrich, Haddow & Birch (2016), §3.2

DEFINITION

Discrete tokens and continuous actions

Top-p restricts a categorical sampling set

$$\sum_{j=1}^{k-1} p_{(j)} < \rho \leq \sum_{j=1}^{k} p_{(j)}$$

$p_{(1)} \geq p_{(2)} \geq \cdots$ sorted category probabilities

$\rho \in (0, 1]$ top-p threshold

Keep the smallest prefix reaching ρ, renormalize, then sample one category.

Holtzman et al. (2020), §3.1

THINK, EXPLAIN, COMPARE

Discrete tokens and continuous actions

Transfer: the retained set depends on mass

Sorted token probabilities are $(0.5, 0.3, 0.2)$ and top-p threshold $\rho = 0.7$. Which distribution is sampled?

A $(0.5, 0.3, 0.2)$

B $(0.625, 0.375, 0)$

C $(1, 0, 0)$

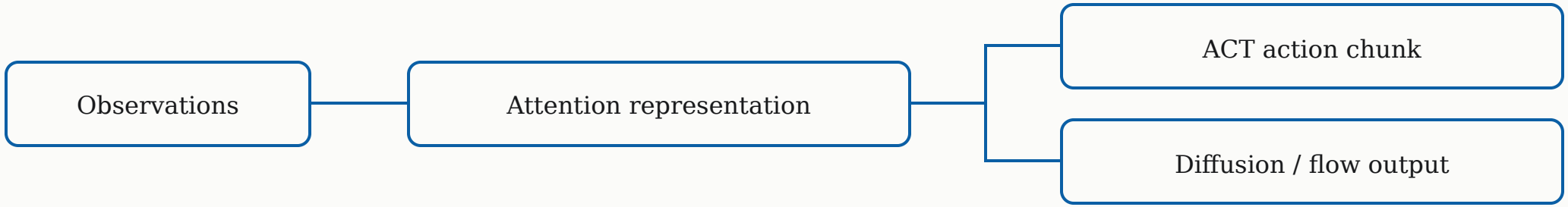
Holtzman et al. (2020), §3.1 · Course example

SYNTHESIS

Transformers and ACT

One representation, several output models

Attention builds a conditional representation. ACT maps it to an action chunk. Diffusion and flow matching will use conditioning to guide iterative generation.



Zhao et al. (2023), §IV-B–C, Appendix C

References

Primary sources and reading 1

1. Vaswani et al. (2017). Attention Is All You Need. §3.2–3.5, Eq. 1; NeurIPS.
2. Blanchard, Higham & Higham (2021). Accurately Computing the Log-Sum-Exp and Softmax Functions. §4.

References

Primary sources and reading 2

1. Su et al. (2021). RoFormer: Enhanced Transformer with Rotary Position Embedding. §3.2.
2. Xiong et al. (2020). On Layer Normalization in the Transformer Architecture. §2, Fig. 1; ICML.
3. Shazeer (2019). Fast Transformer Decoding: One Write-Head is All You Need. §2.

References

Primary sources and reading 3

1. Zhao et al. (2023). Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. §IV and Appendix C; RSS.
2. Kingma & Welling (2014). Auto-Encoding Variational Bayes. §2, Appendix B; ICLR.

References

Primary sources and reading 4

1. Sennrich, Haddow & Birch (2016). Neural Machine Translation of Rare Words with Subword Units. §3.2; ACL.
2. Holtzman et al. (2020). The Curious Case of Neural Text Degeneration. §3.1; ICLR.