Spring25 CS598YP

19.2: Orca

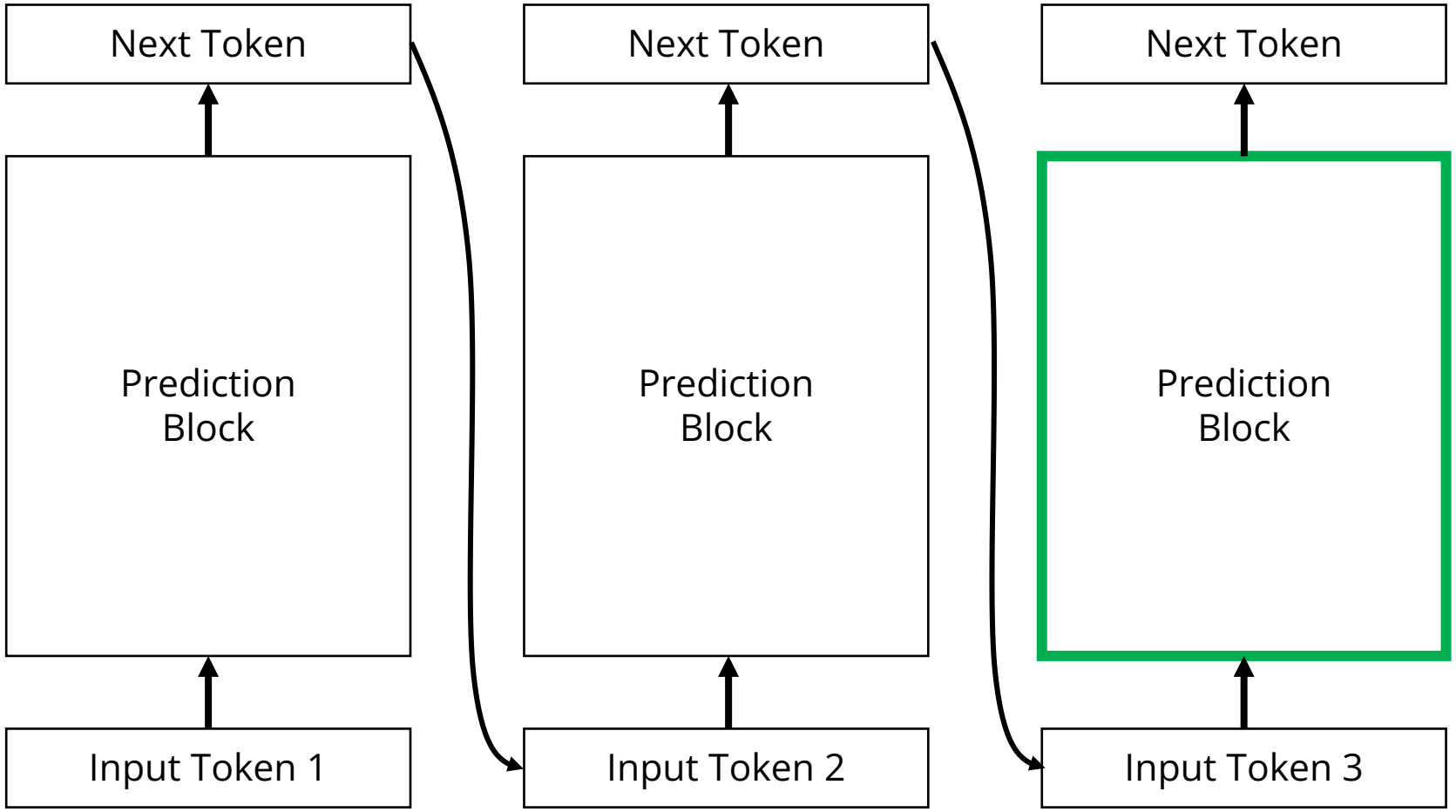
Yongjoo Park

University of Illinois Urbana-Champaign

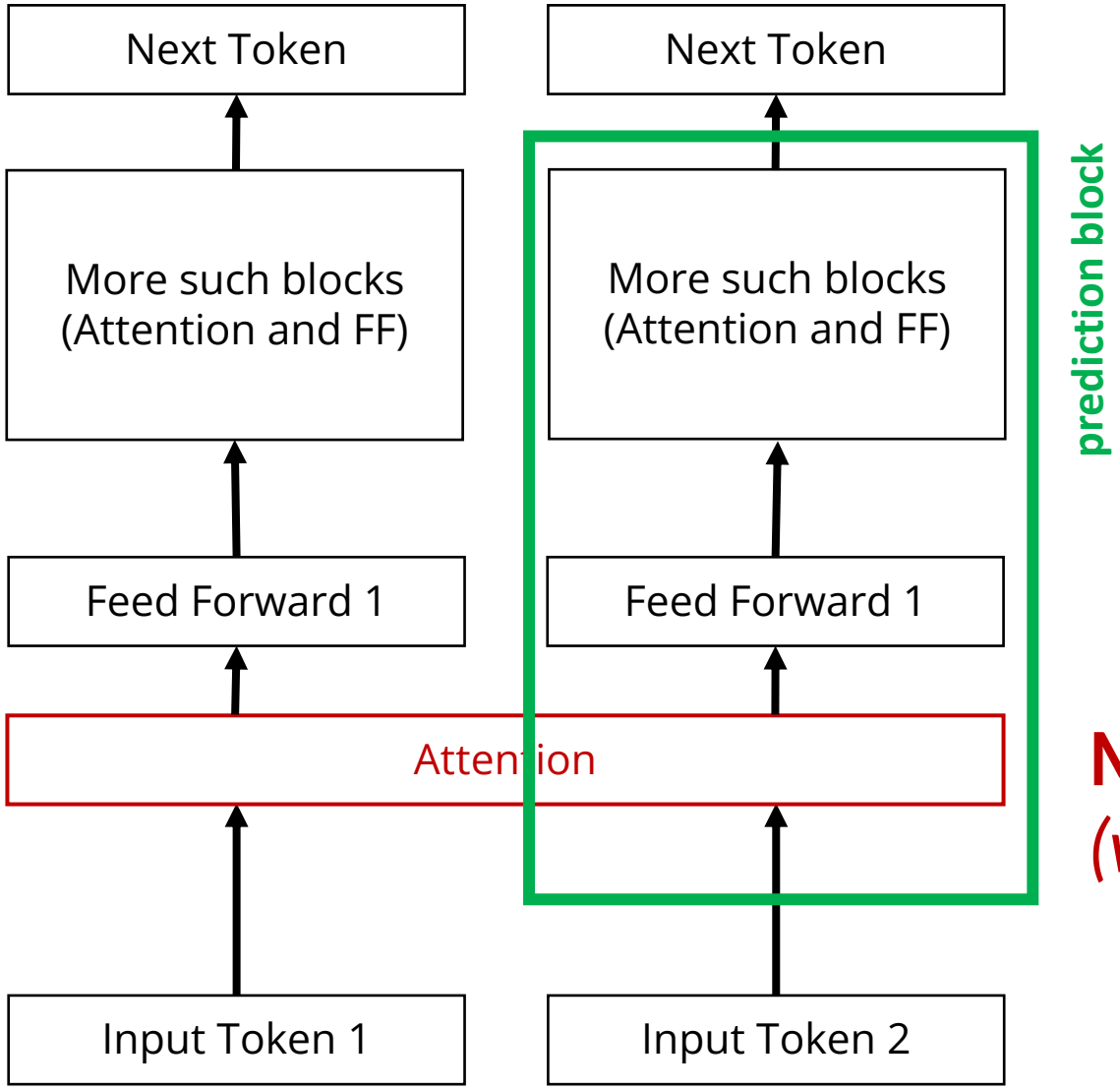
Outline

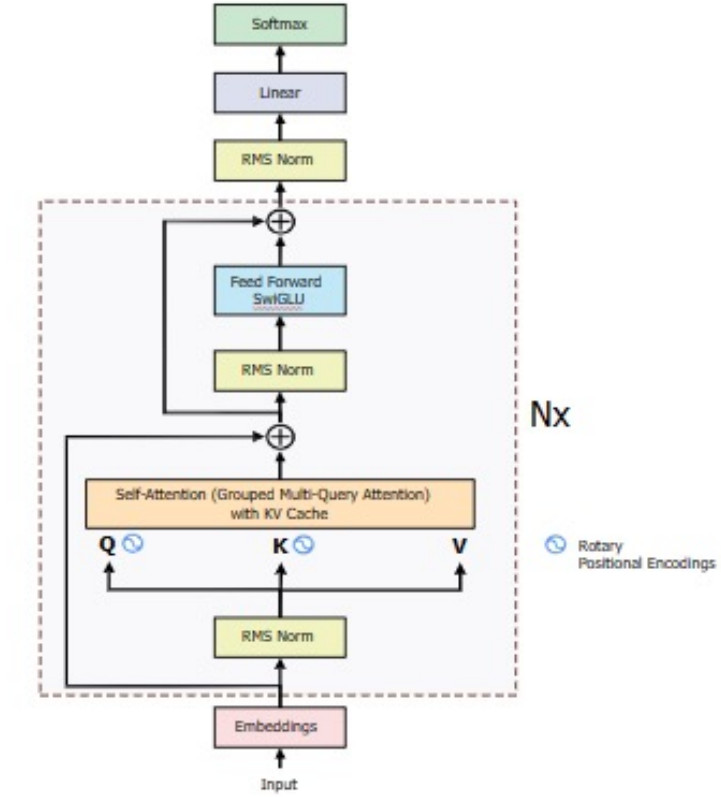
- ***Attention*** inside Transformer
- Static batching vs ***continuous batching***
- ***Selective batching*** for continuous batching

Decoding-only task



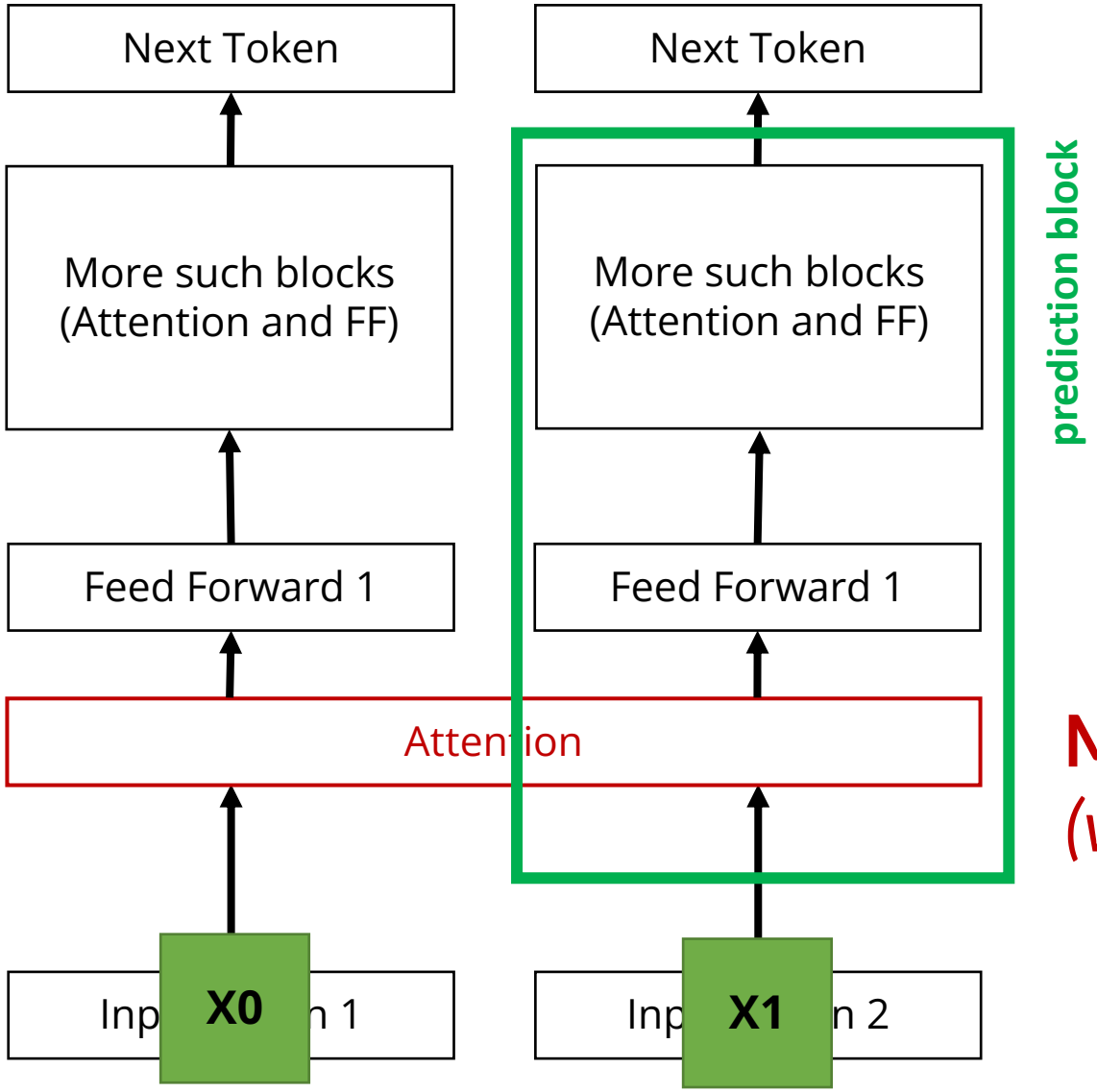
Attention: Captures dependency

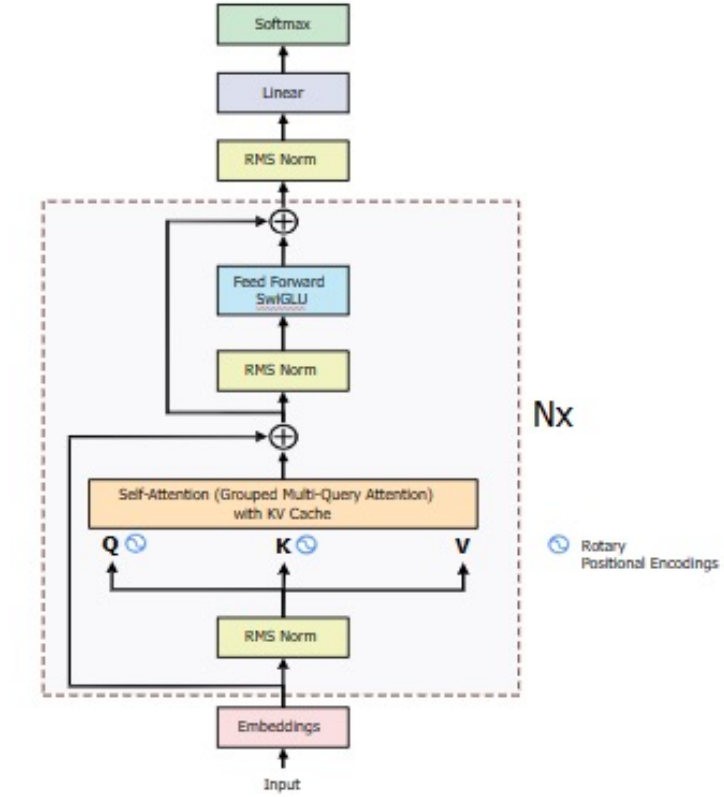




Mut-mul to capture relationship
(w/ key, query, value vectors)

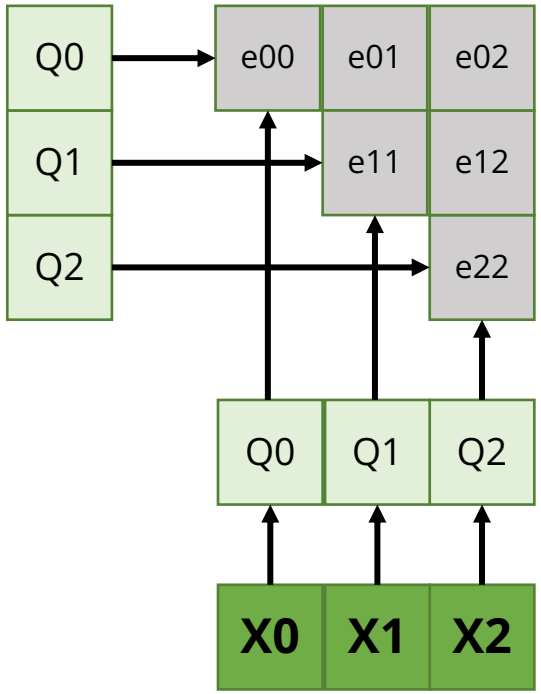
Attention: Captures dependency





Mut-mul to capture relationship
(w/ key, query, value vectors)

Naïve Approach 1: similarity vector

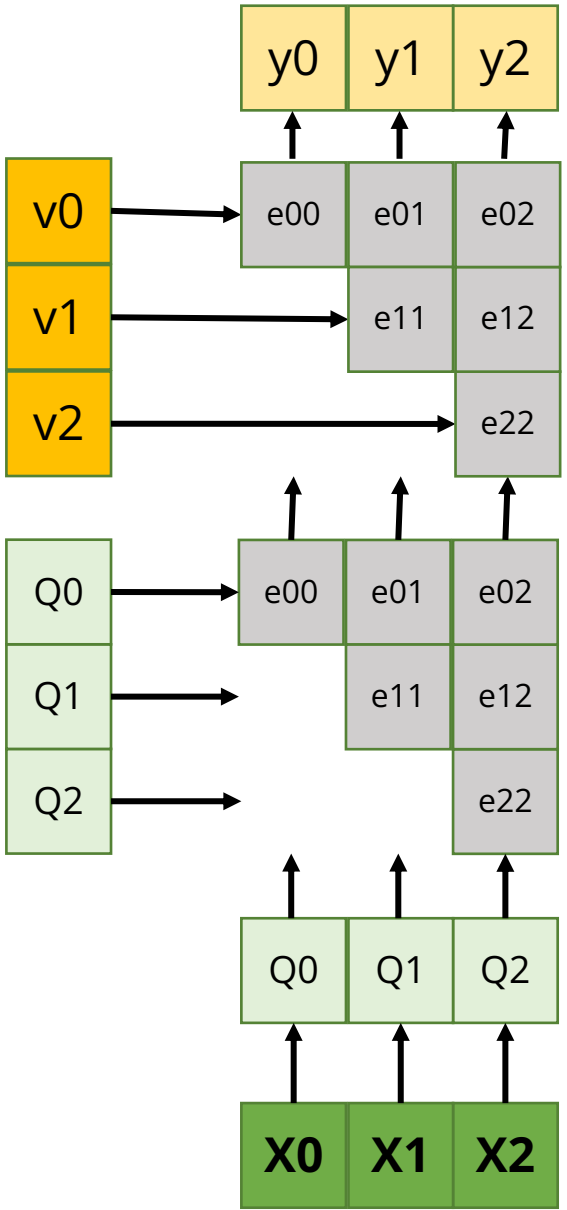


pairwise similarity calculations

We can calculate similarities in this way, but

How can we express the meaning of each token?

Naïve Approach 2: similarity vector + **value** vector

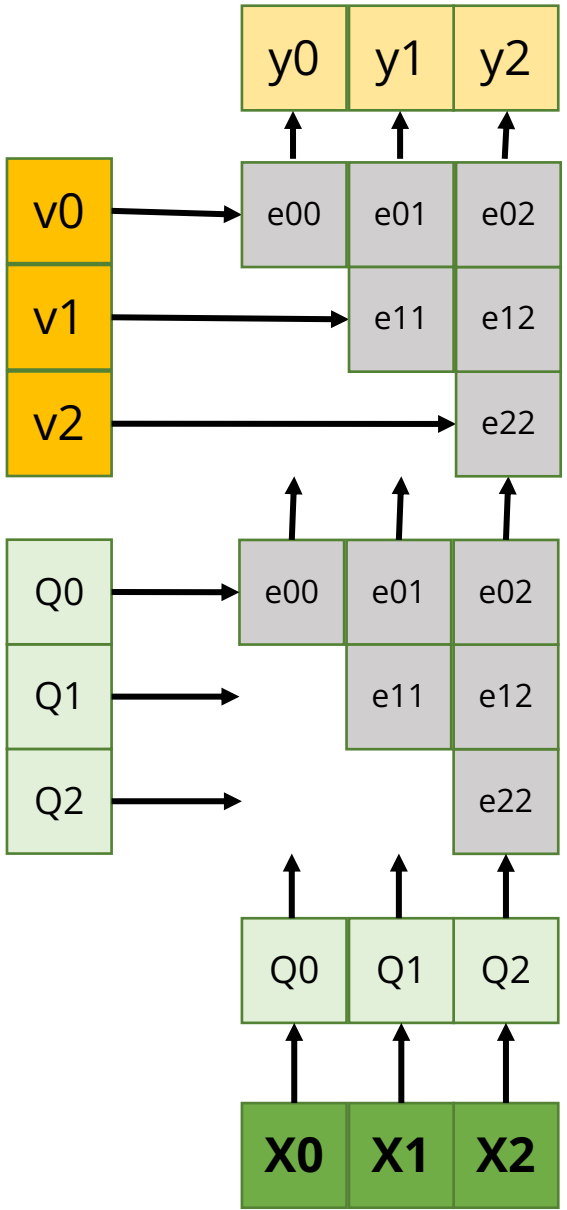


multiply respective values

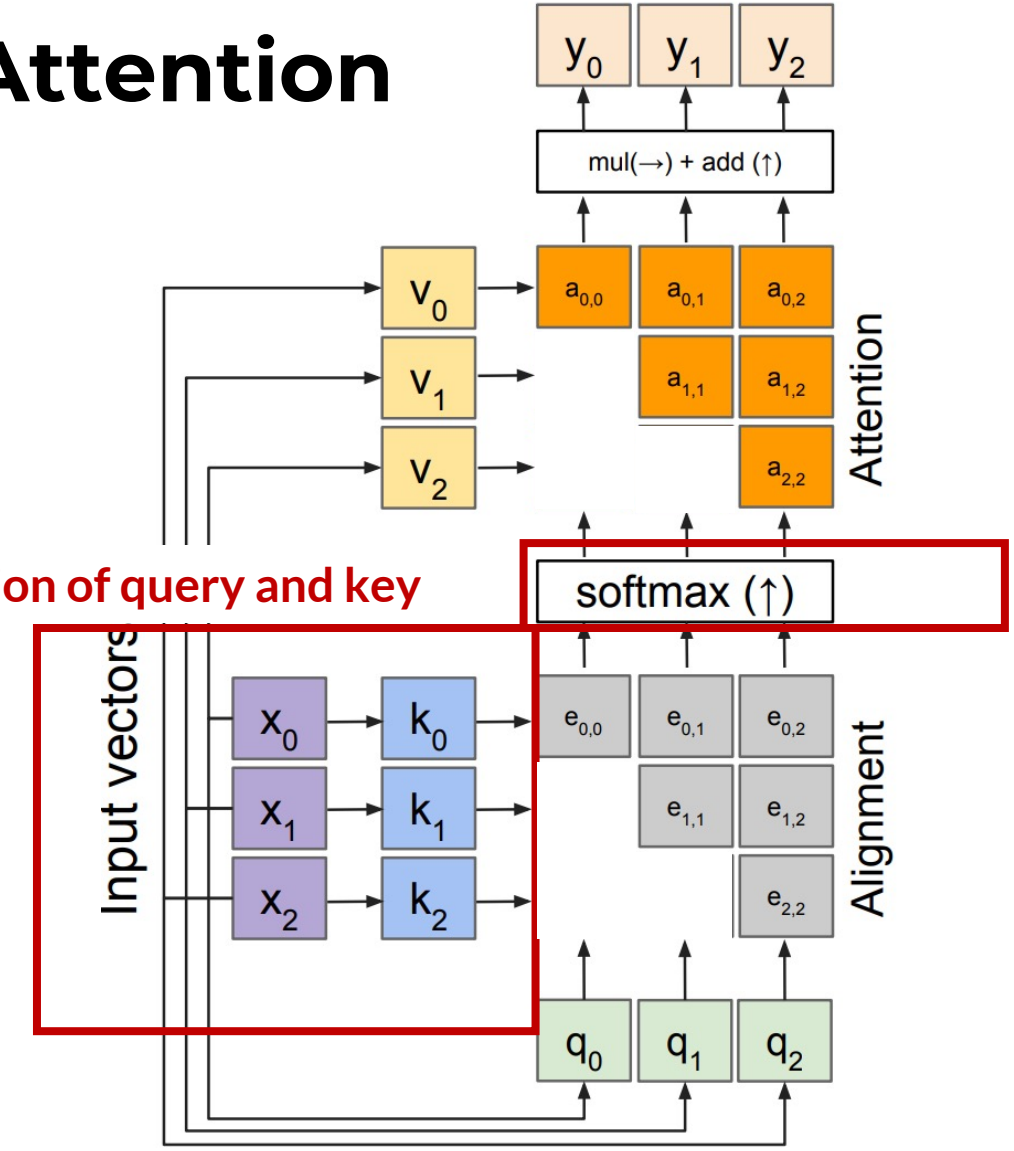
y0 = v0 * e00
y1 = v0 * e01 + v1 * e11
y2 = v0 * e02 + v1 * e12 + v2 * e22

pairwise similarity calculations

Our naïve approach



Actual Attention



Separation of query and key

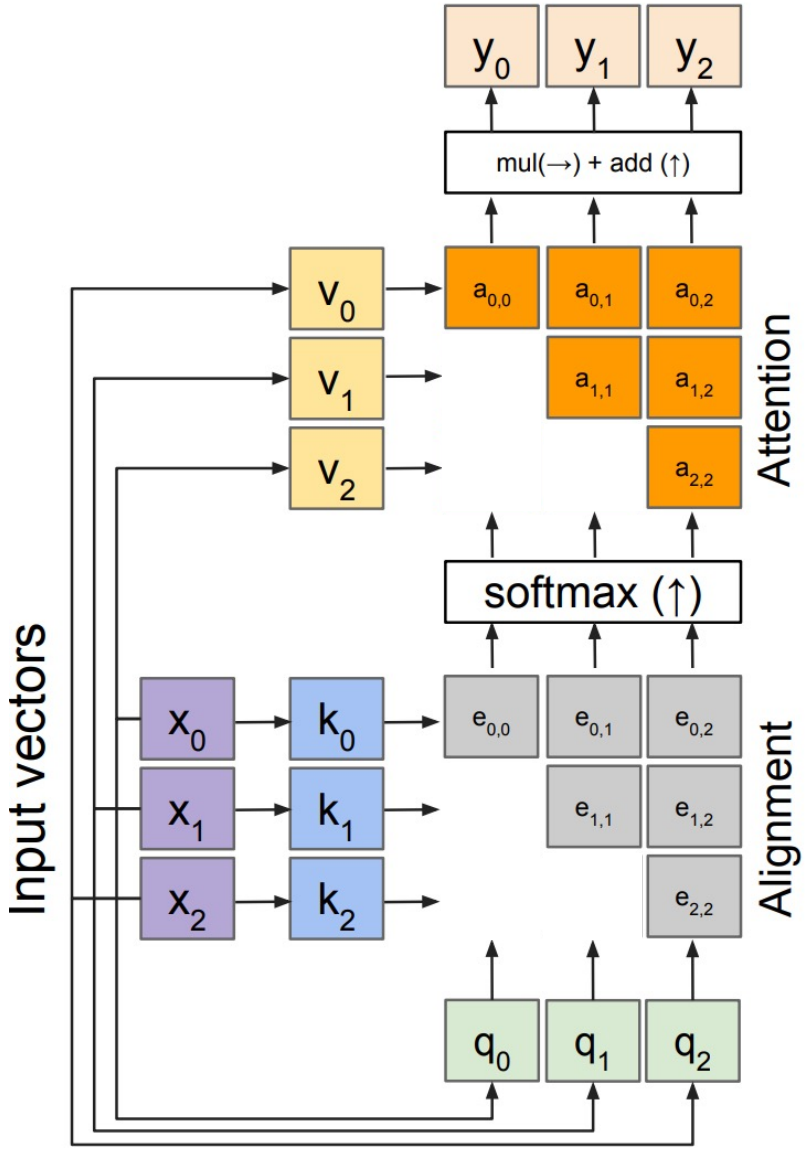
winners take it all

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

X0 X1 X2

What are the dimensions?

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



For y_0

Q = [1 H] K = [1 H] V = [1 H]

For y_1

Q = [2 H] K = [2 H] V = [2 H]

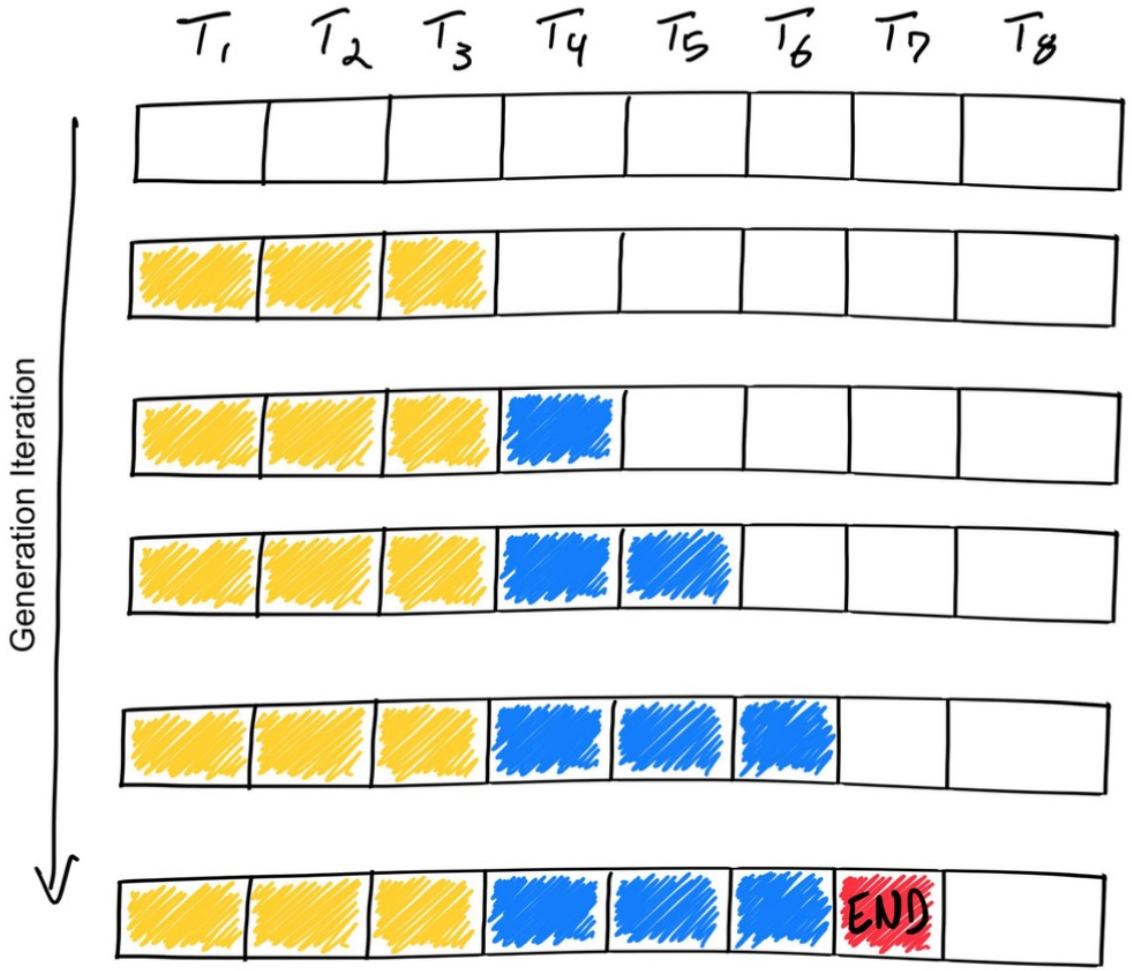
For y_N

Q = [N H] K = [N H] V = [N H]

Orca

slide credits:
https://cseweb.ucsd.edu/~yiying/cse291-winter24/reading/llm-serving.pdf
https://cs231n.stanford.edu/slides/2022/lecture_11_ruohan.pdf

LLM inference basics



How does text generation work?

Iterative: each forward pass generates a single token

Autoregressive: generation consumes prompt tokens + previously generated tokens

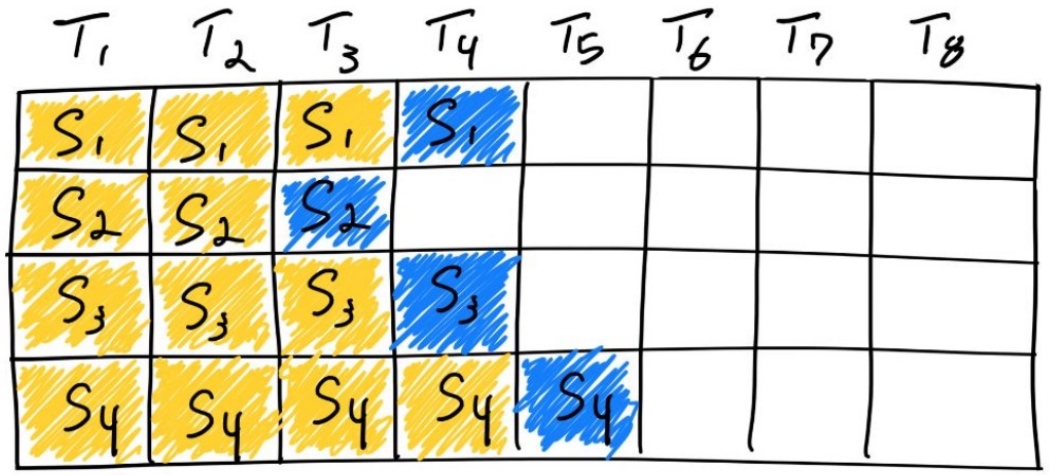
Completion potentially decided by model: A generated token can be the end-of-sequence token

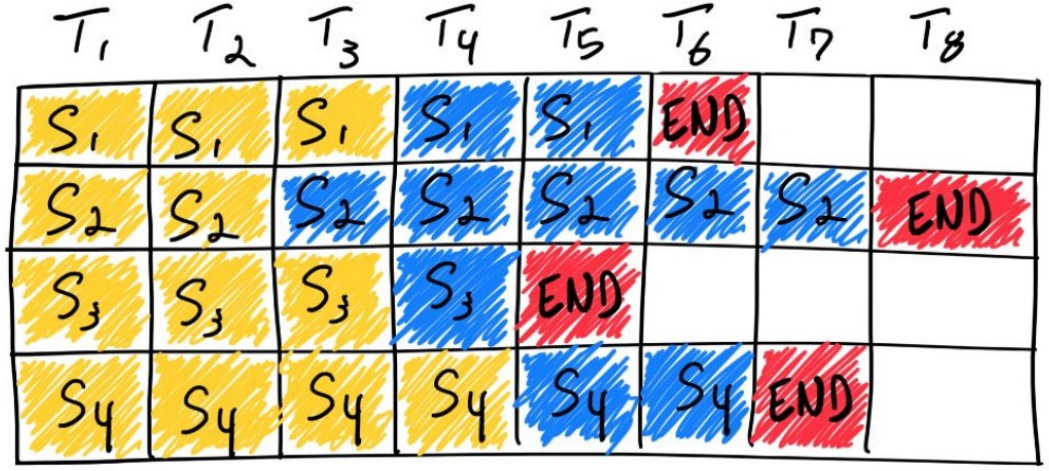
Legend:

- Yellow: prompt token
- Blue: generated token
- Red: end-of-sequence token

Static batching

- Batching multiple sequences on GPU, aka “static batching”
- Problem: GPU utilization drops as sequences complete





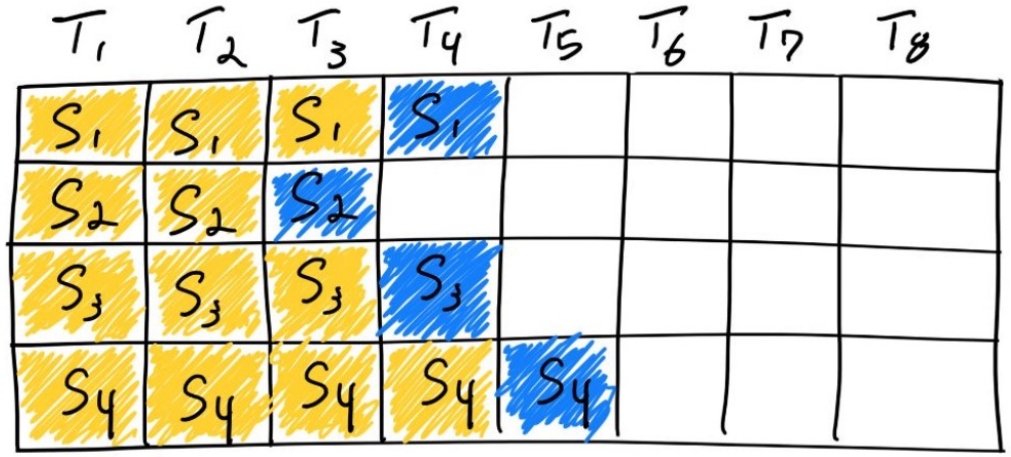
Legend:

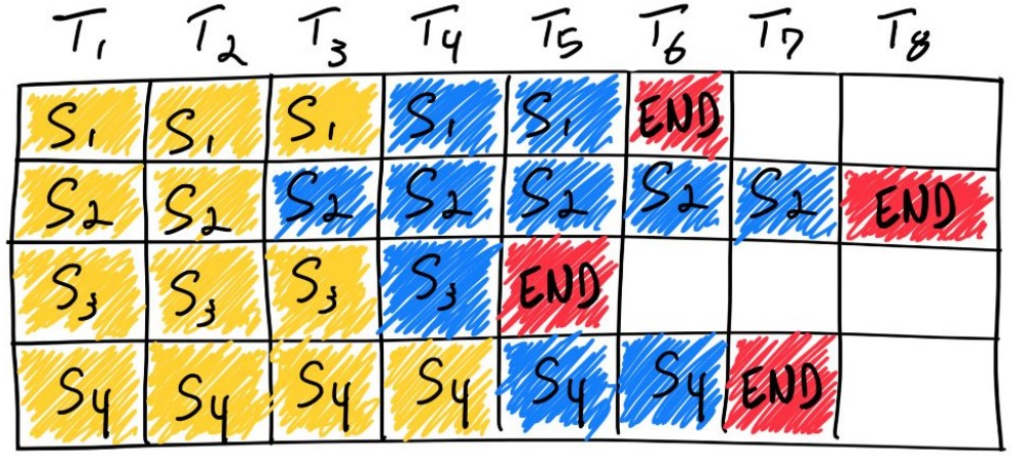
- Yellow: prompt token
- Blue: generated token
- Red: end-of-sequence token

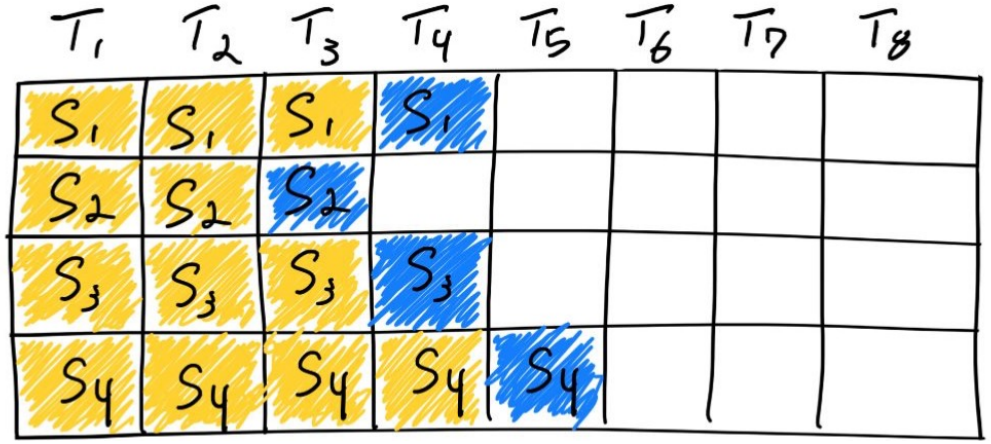
Continuous batching

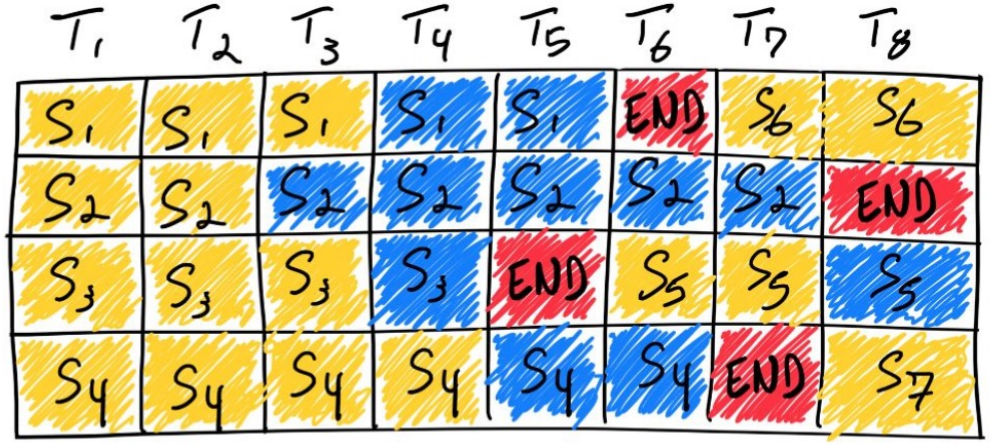
Top: static batching
Bottom: continuous batching

Legend:
- Yellow: prompt token
- Blue: generated token
- Red: end-of-sequence token

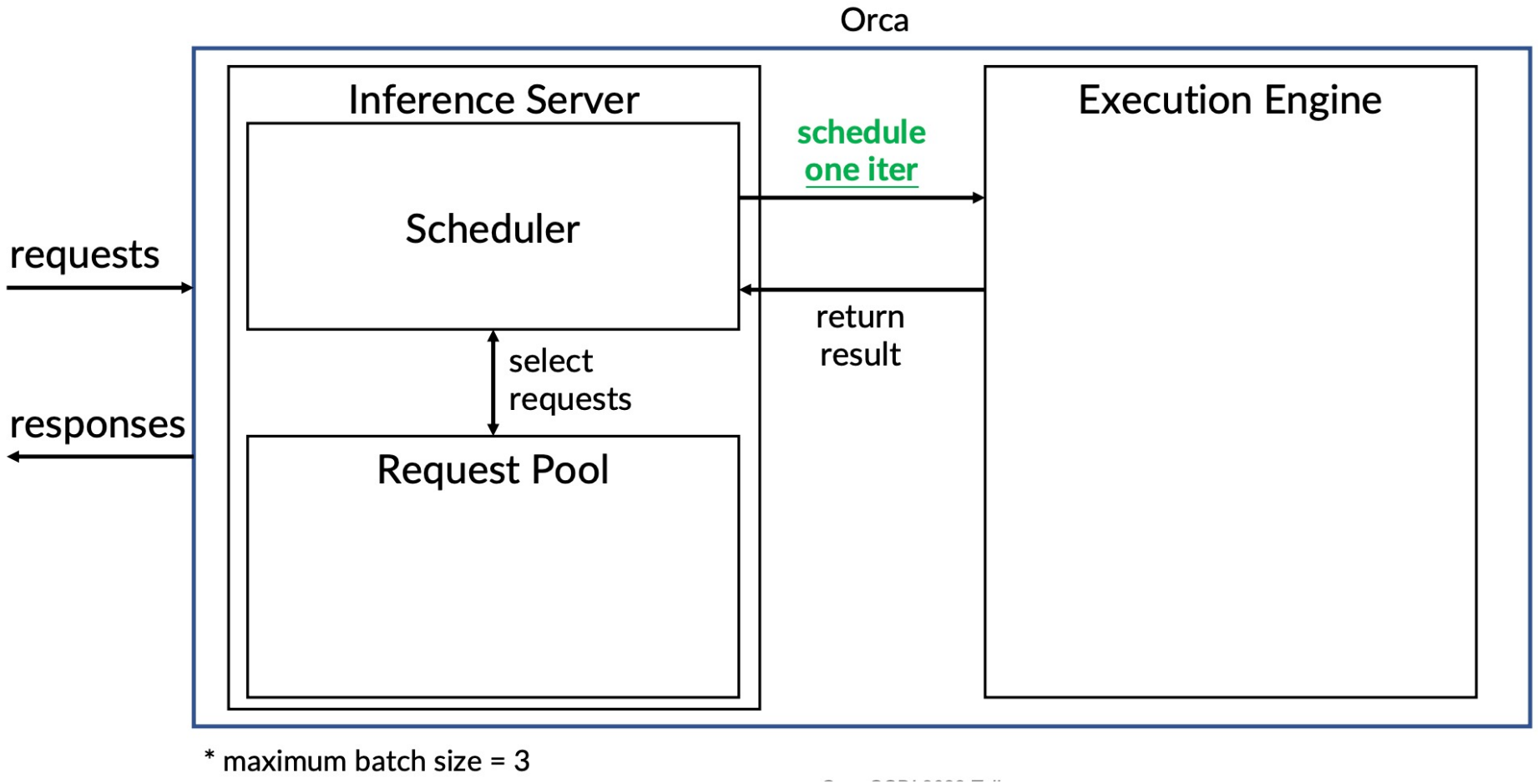




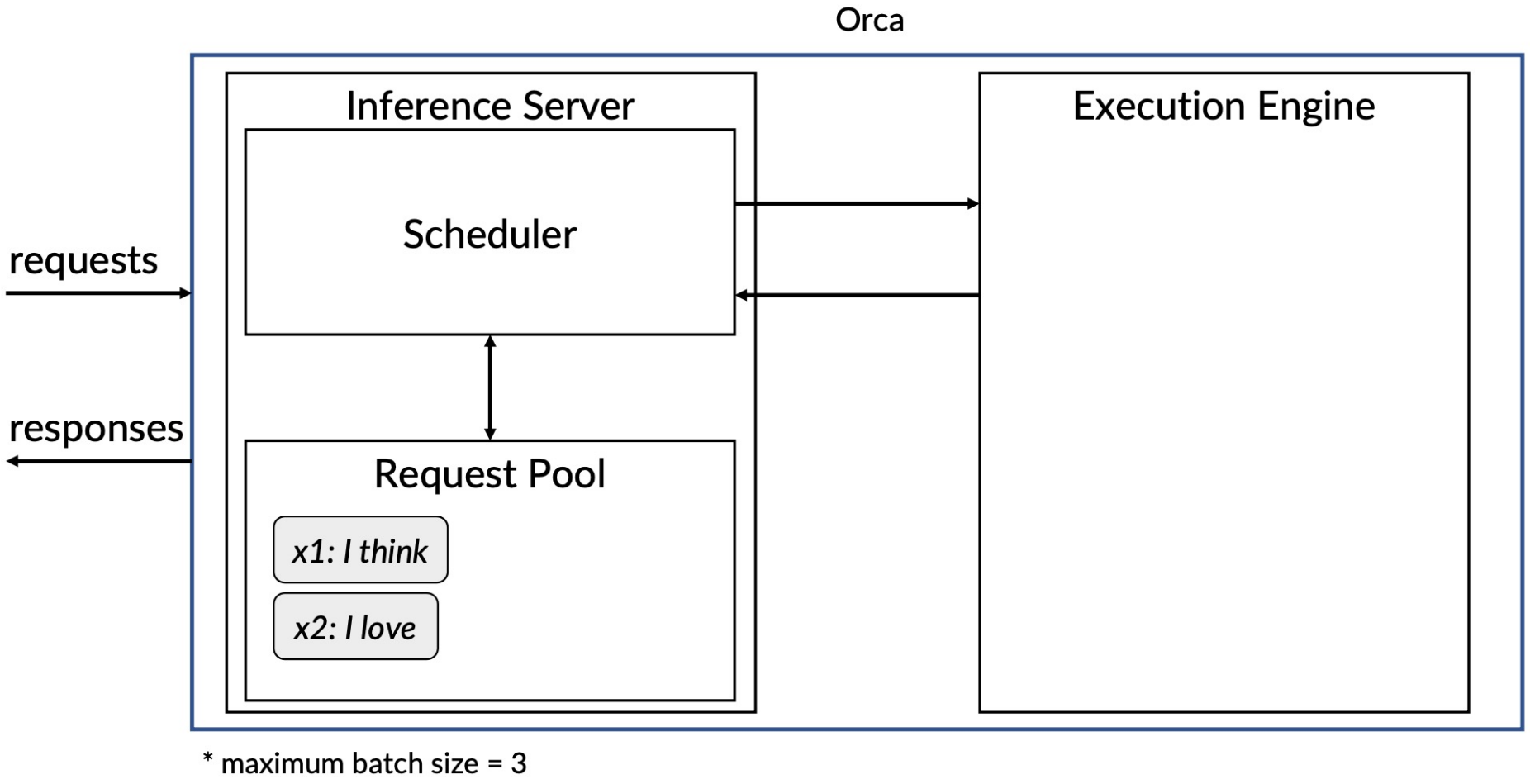




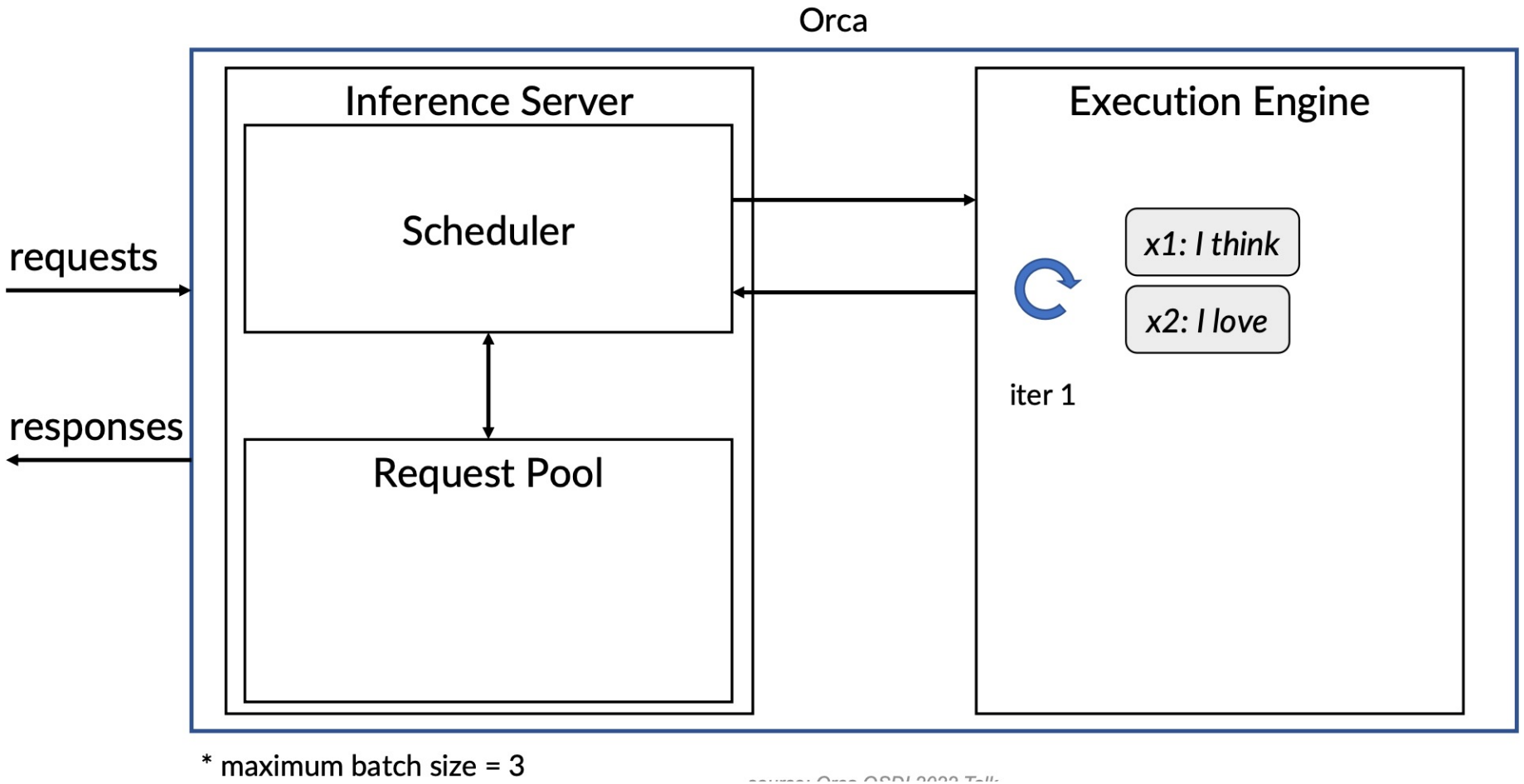
Iteration-level scheduling



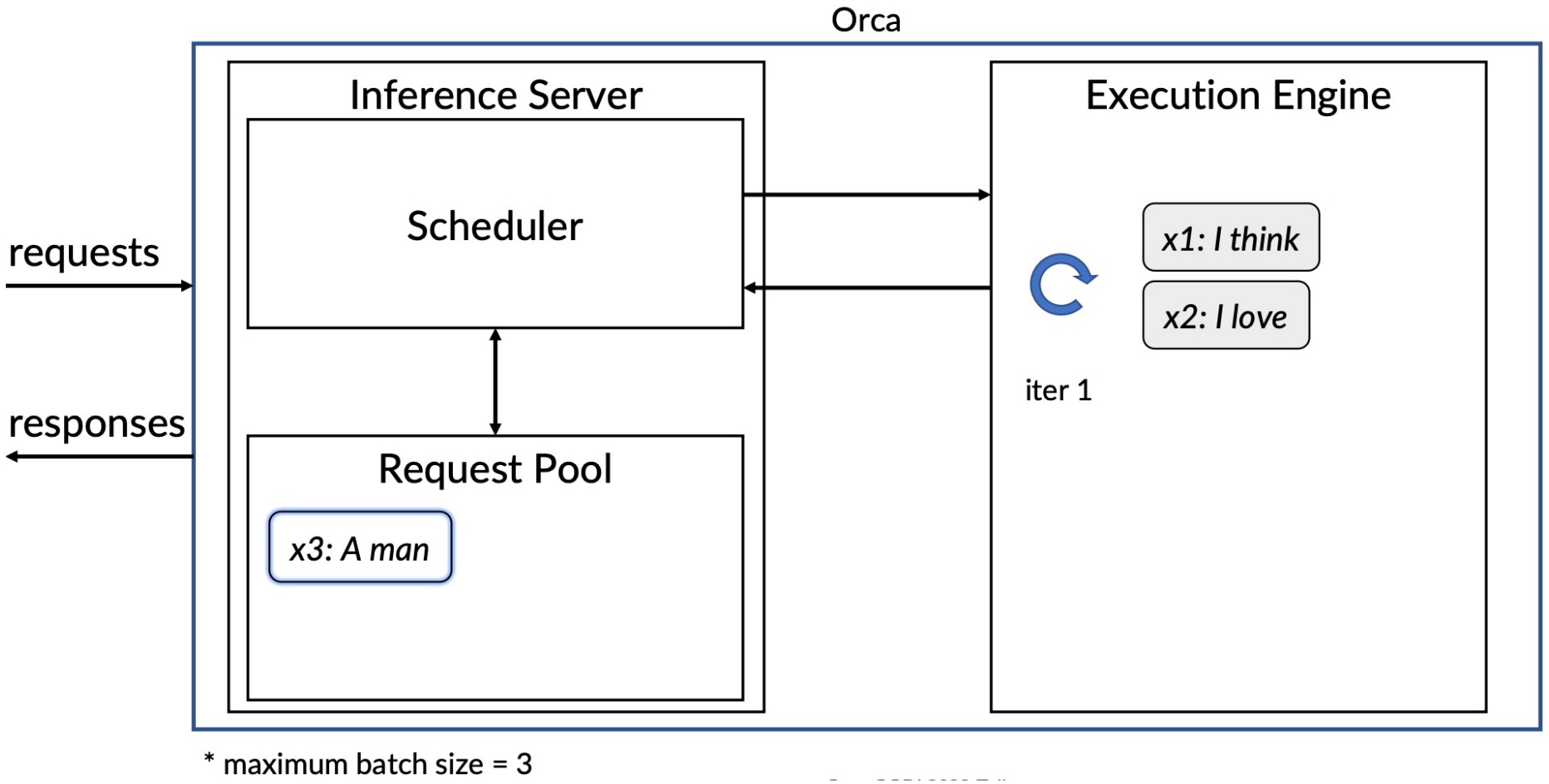
Iteration-level scheduling



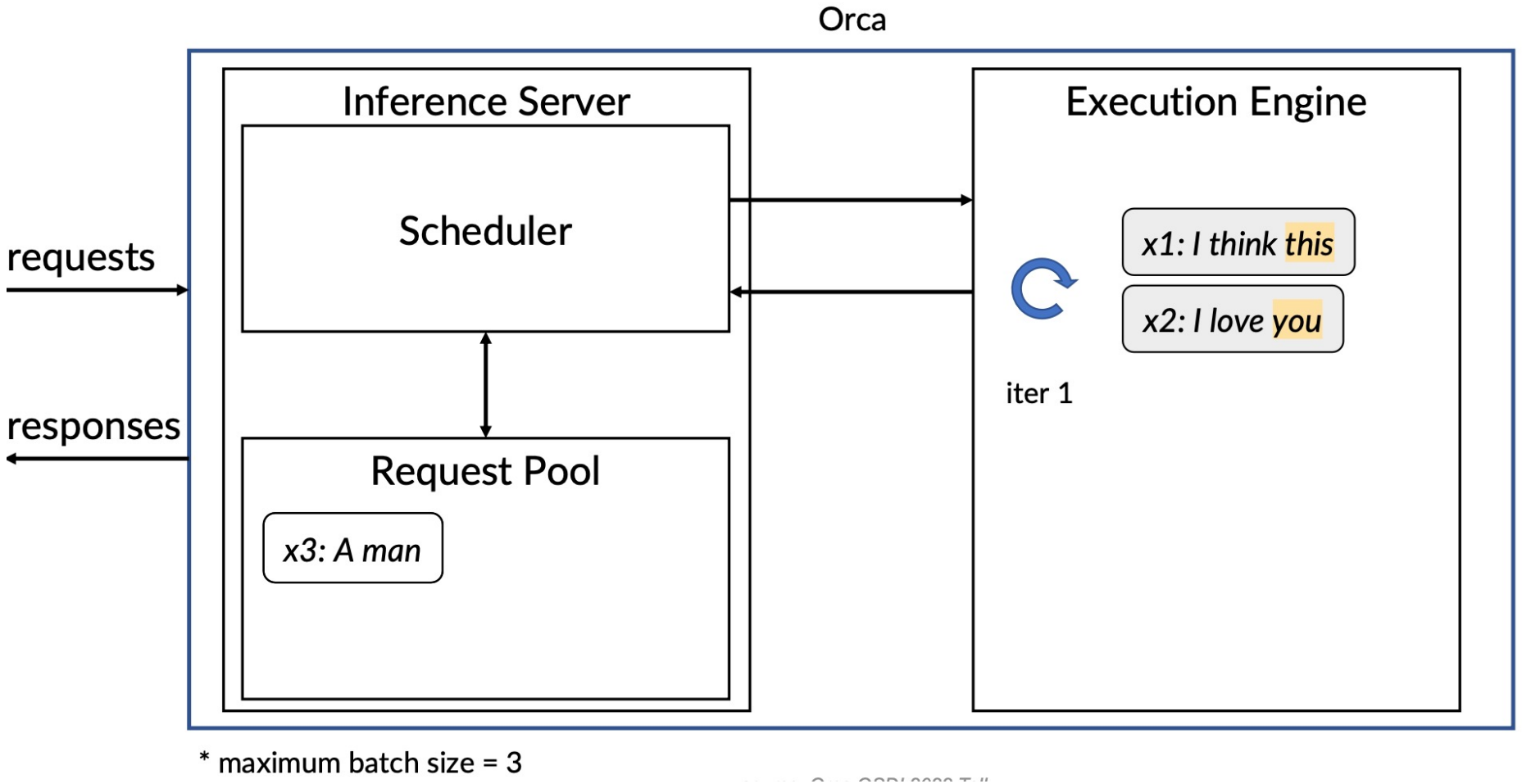
Iteration-level scheduling



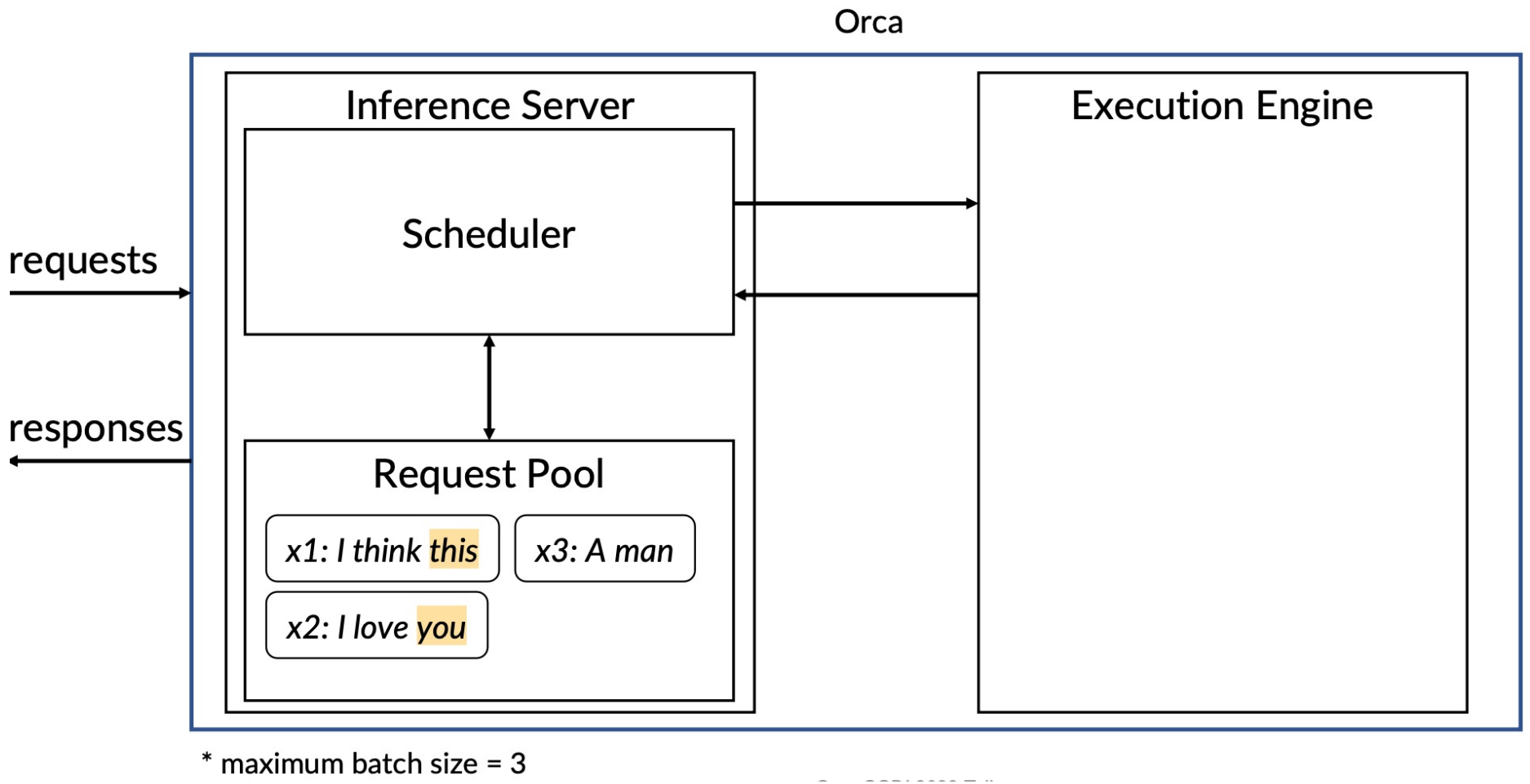
Iteration-level scheduling



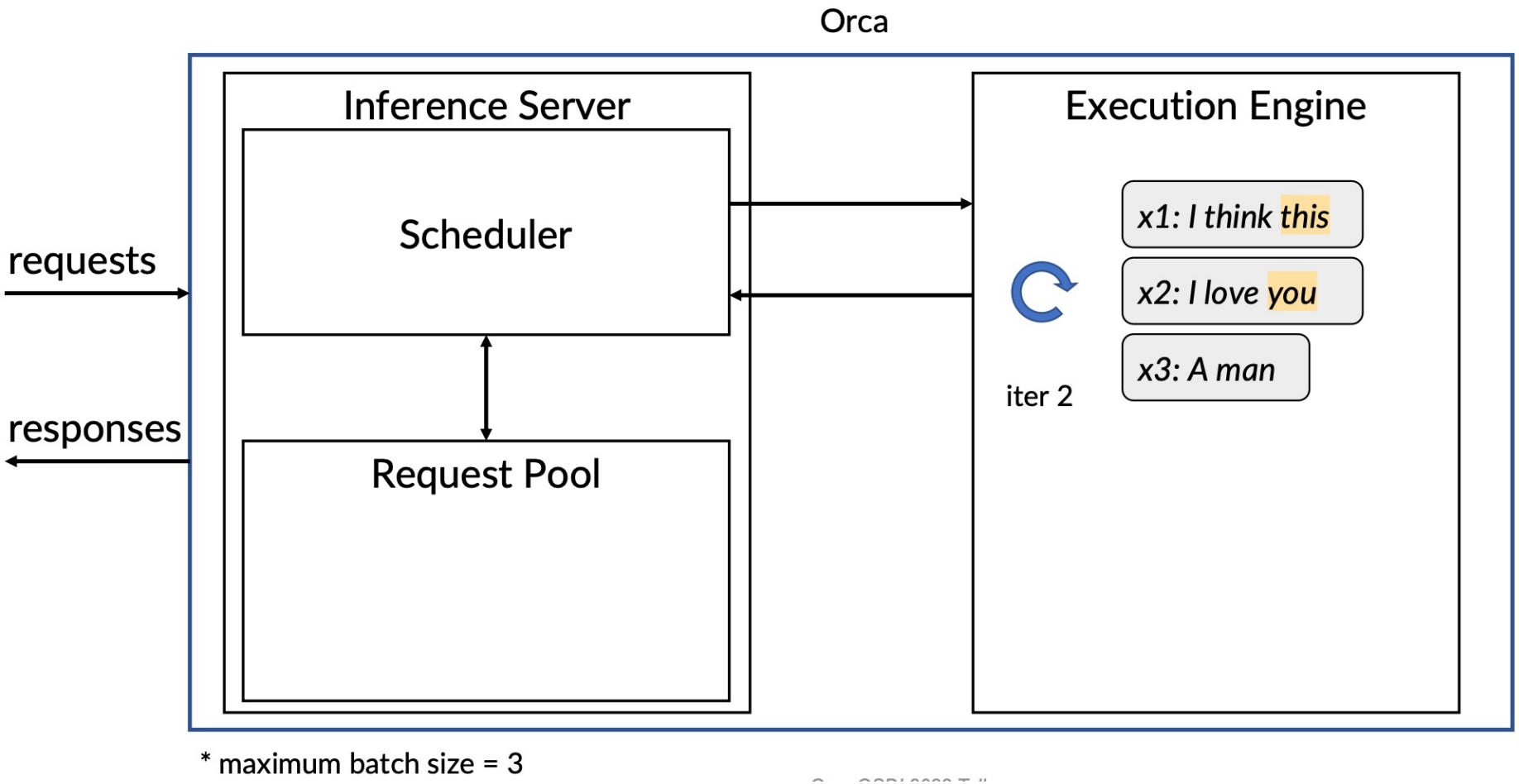
Iteration-level scheduling



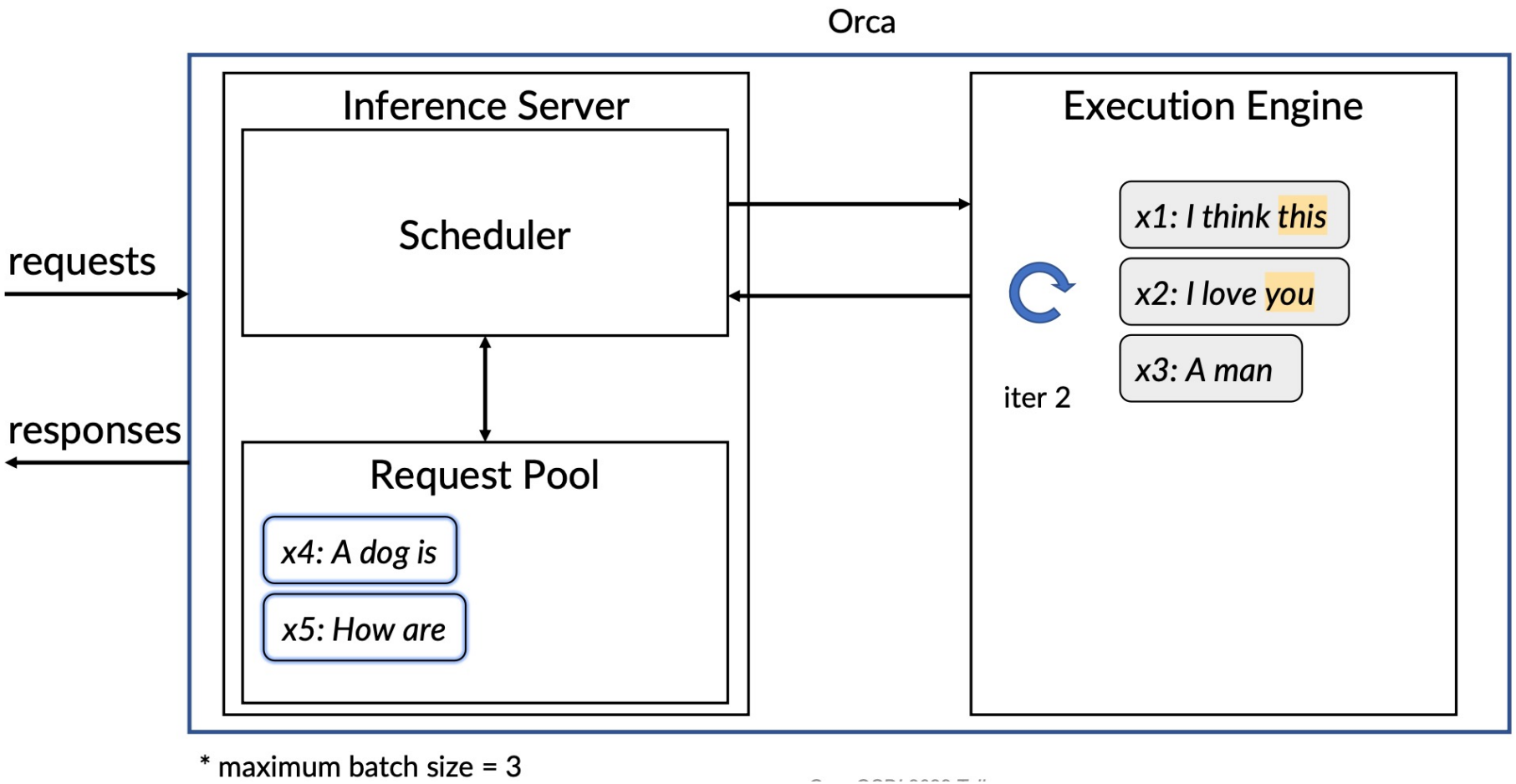
Iteration-level scheduling



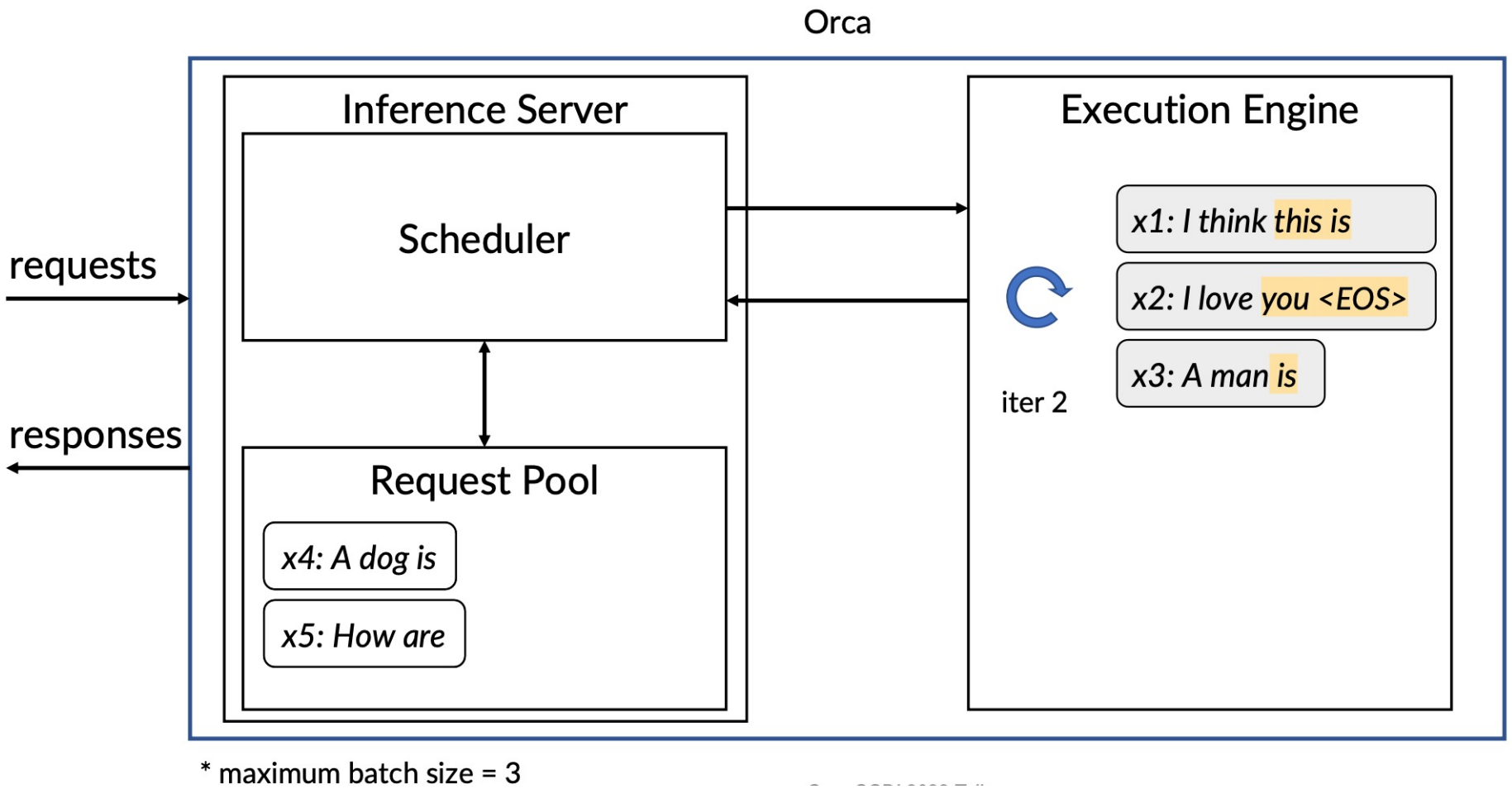
Iteration-level scheduling



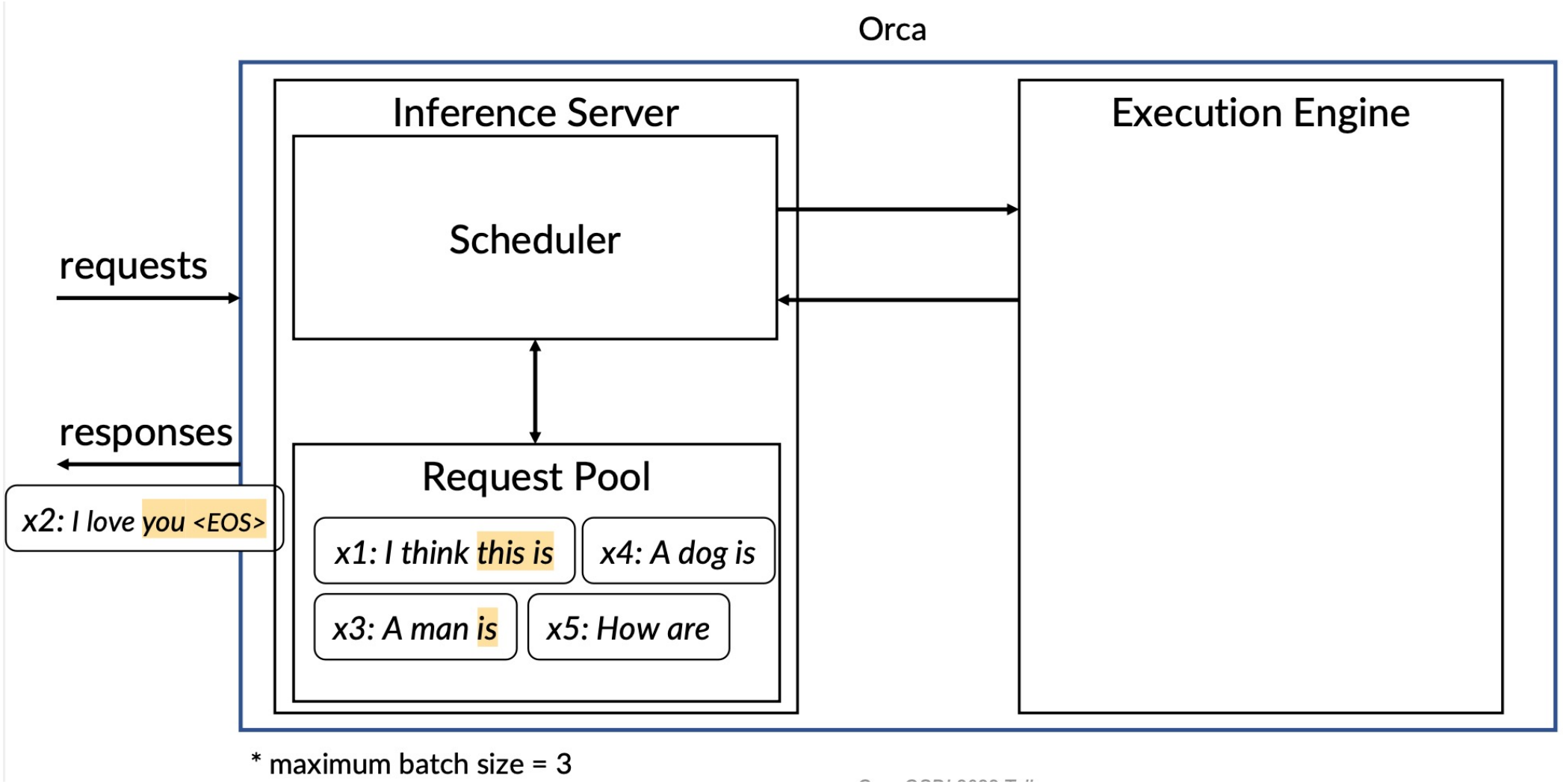
Iteration-level scheduling



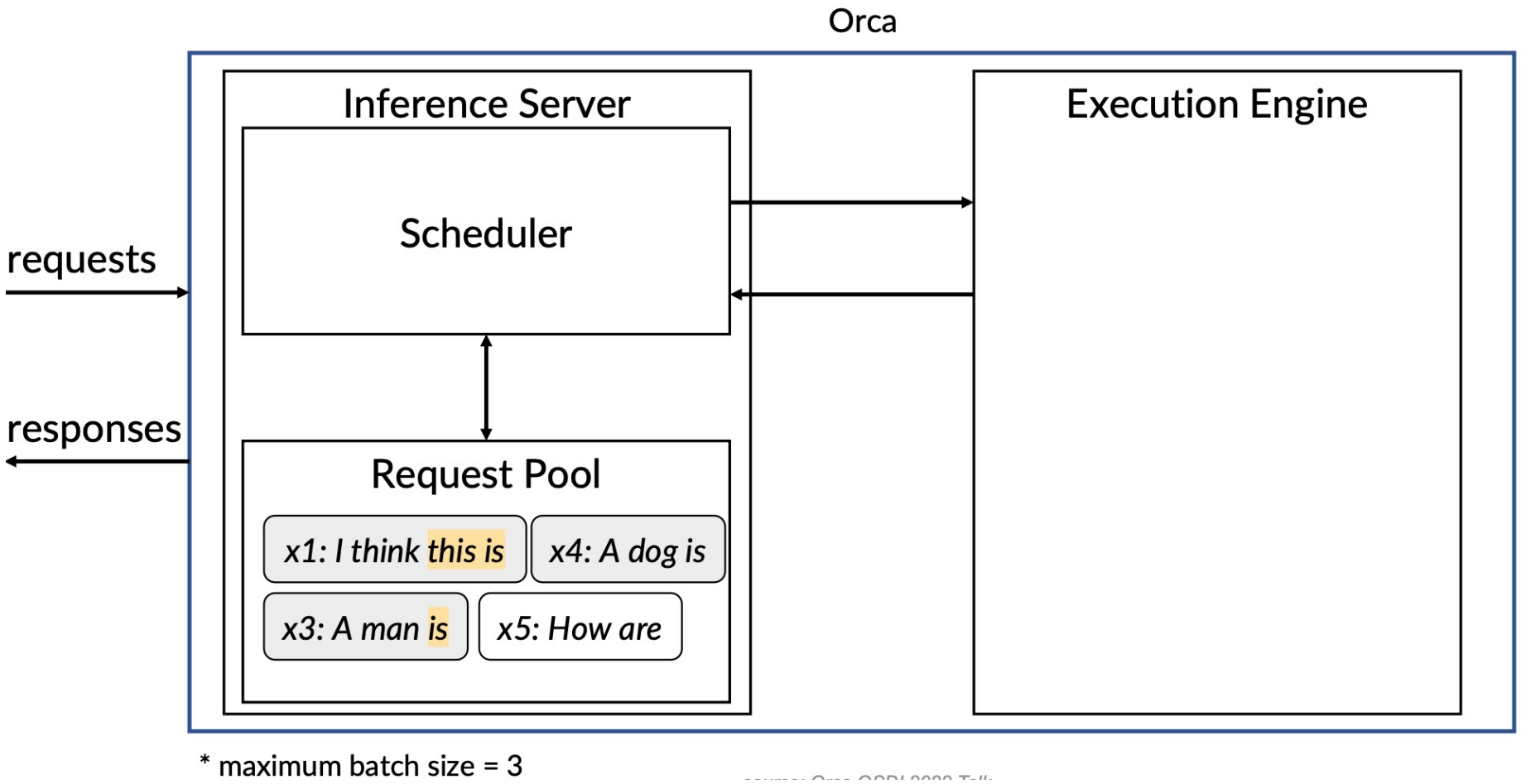
Iteration-level scheduling



Iteration-level scheduling

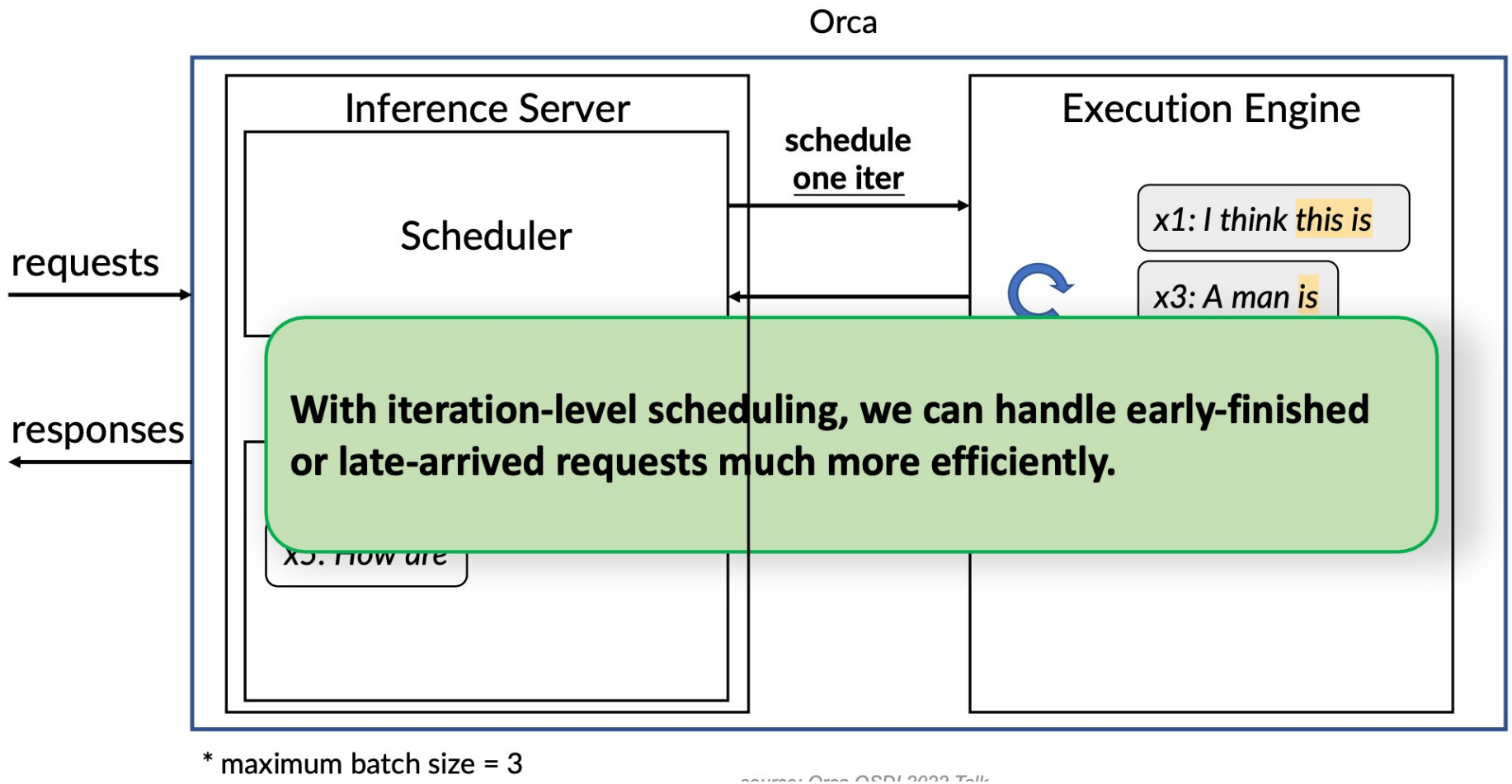


Iteration-level scheduling

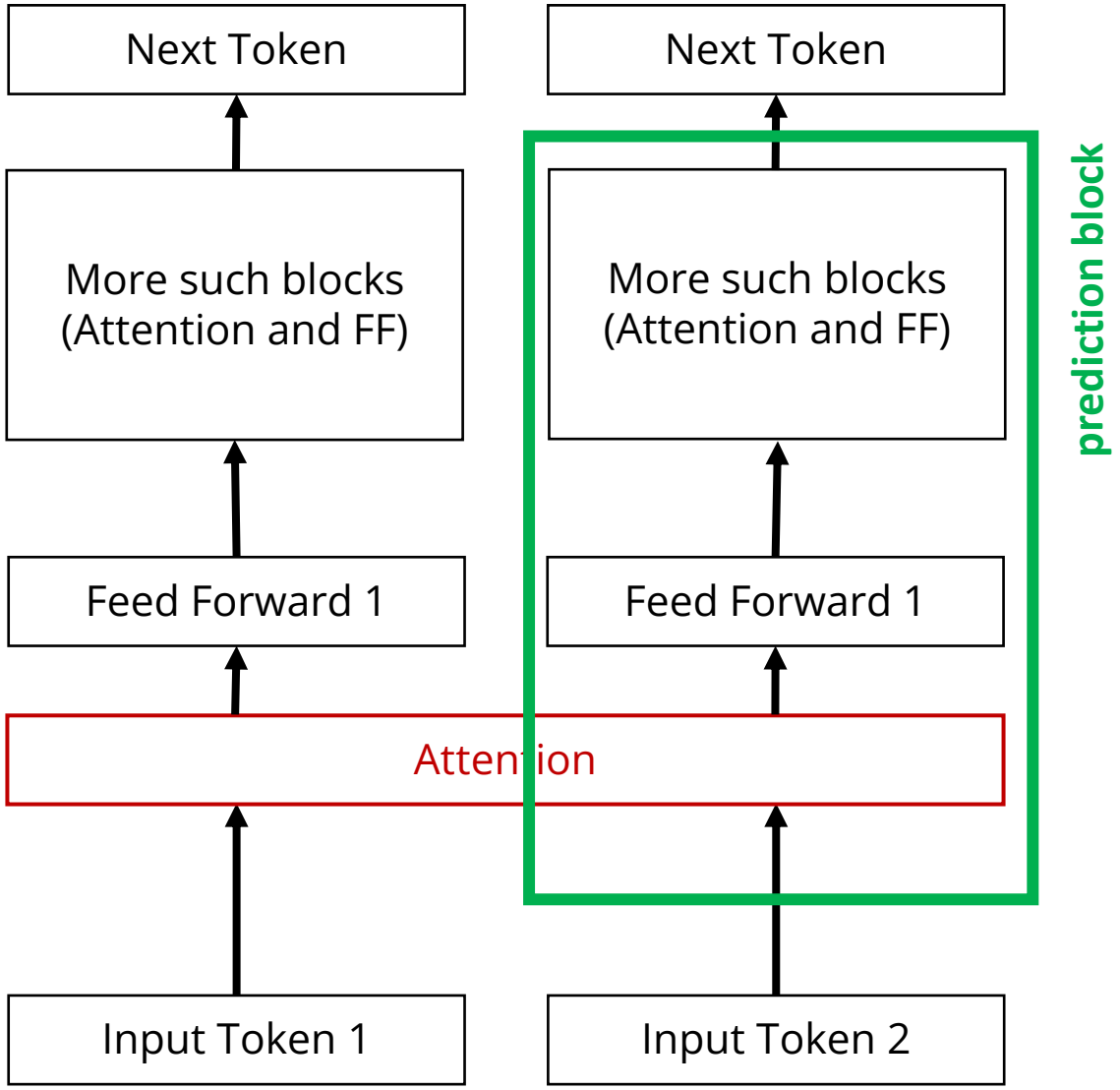


* maximum batch size = 3

Iteration-level scheduling



Feed-forward is independent, given attention



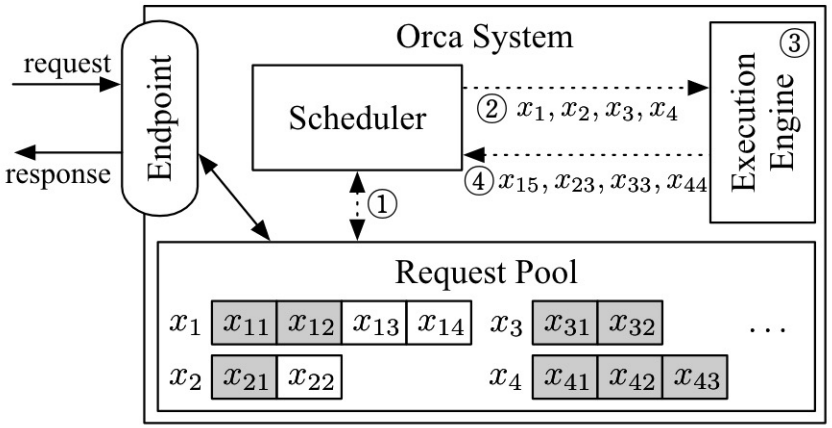
$$y_1 = f(Wx_1 + b), \quad y_2 = f(Wx_2 + b)$$

$$\hat{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^{2n}$$

$$\hat{W} = \begin{bmatrix} W & W \end{bmatrix} \in \mathbb{R}^{m \times 2n}$$

$$\hat{y} = \hat{W}\hat{x} + \hat{b} = Wx_1 + Wx_2 + 2b = y_1 + y_2$$

Incompatible scheduling: Attention sizes are different



- Two prefills of different lengths (x3 and x4)
- Two decoding at different indexes (x1 and x2)
- Prefill and decoding (x1 and x3)

Mismatch in attention sizes

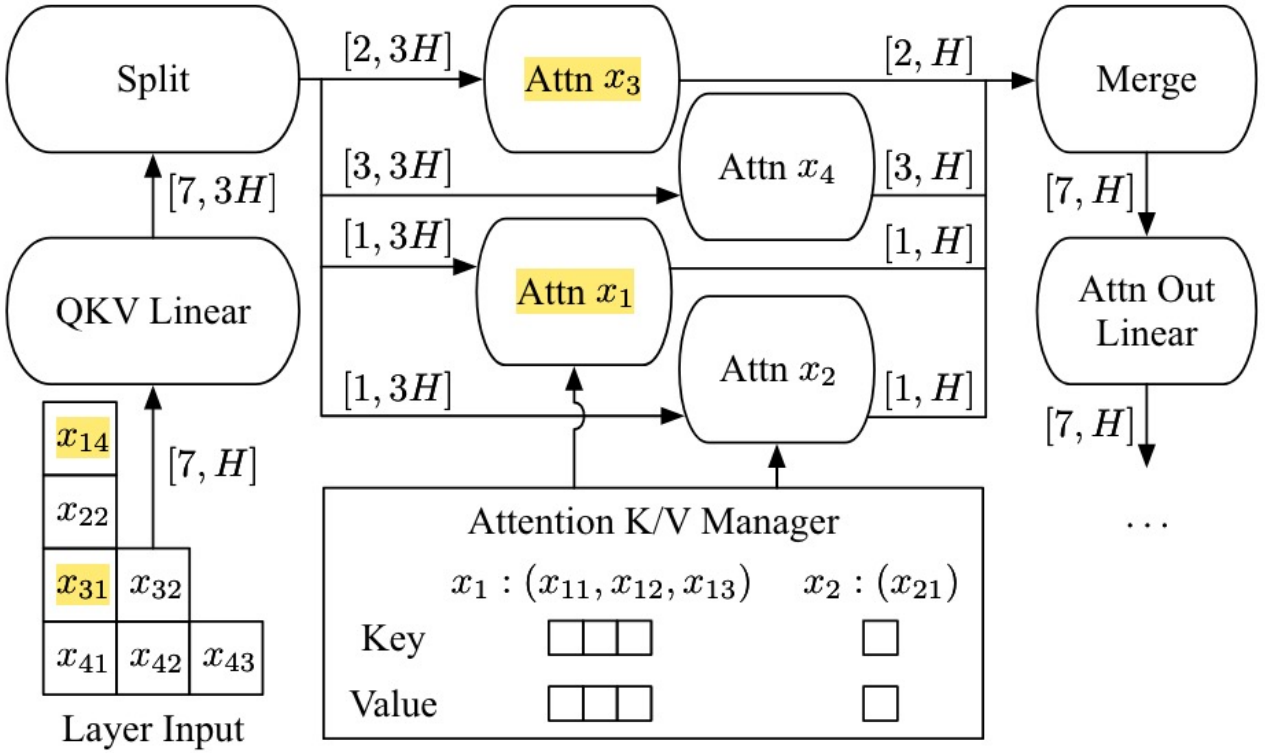
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Non-trivial to merge in matrices

prompt 1: $Q = \begin{pmatrix} n1 * d \end{pmatrix}$ $K^T = \begin{pmatrix} d * n1 \end{pmatrix}$

prompt 2: $Q = \begin{pmatrix} n2 * d \end{pmatrix}$ $K^T = \begin{pmatrix} d * n2 \end{pmatrix}$

Selective batching: *separate* Attention computing



Summary

- ***Attention*** inside Transformer: is a fundamental mechanism for capturing semantic relatedness
- ***Continuous batching****: allows efficient GPU resource utilization*
- ***Selective batching:*** *merged feed-forward + separate attention computations*

Questions?