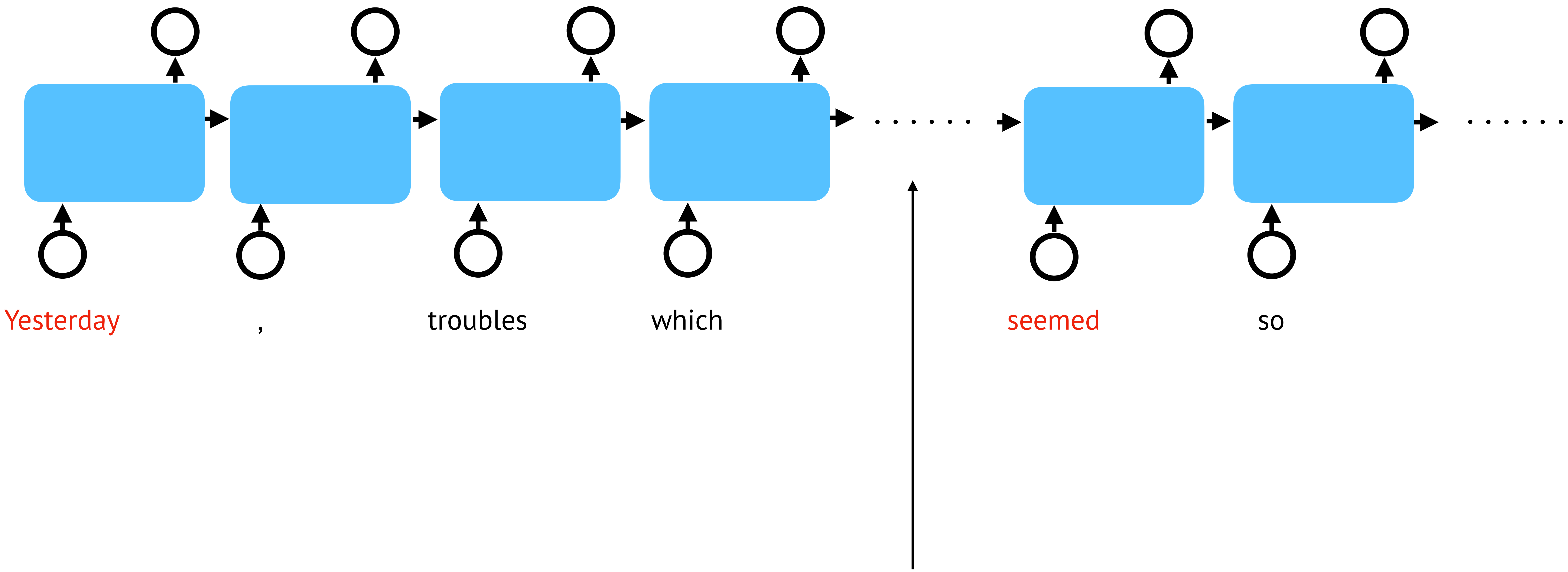
Transformers (1)

COMP3361 — Week 4

Lingpeng Kong

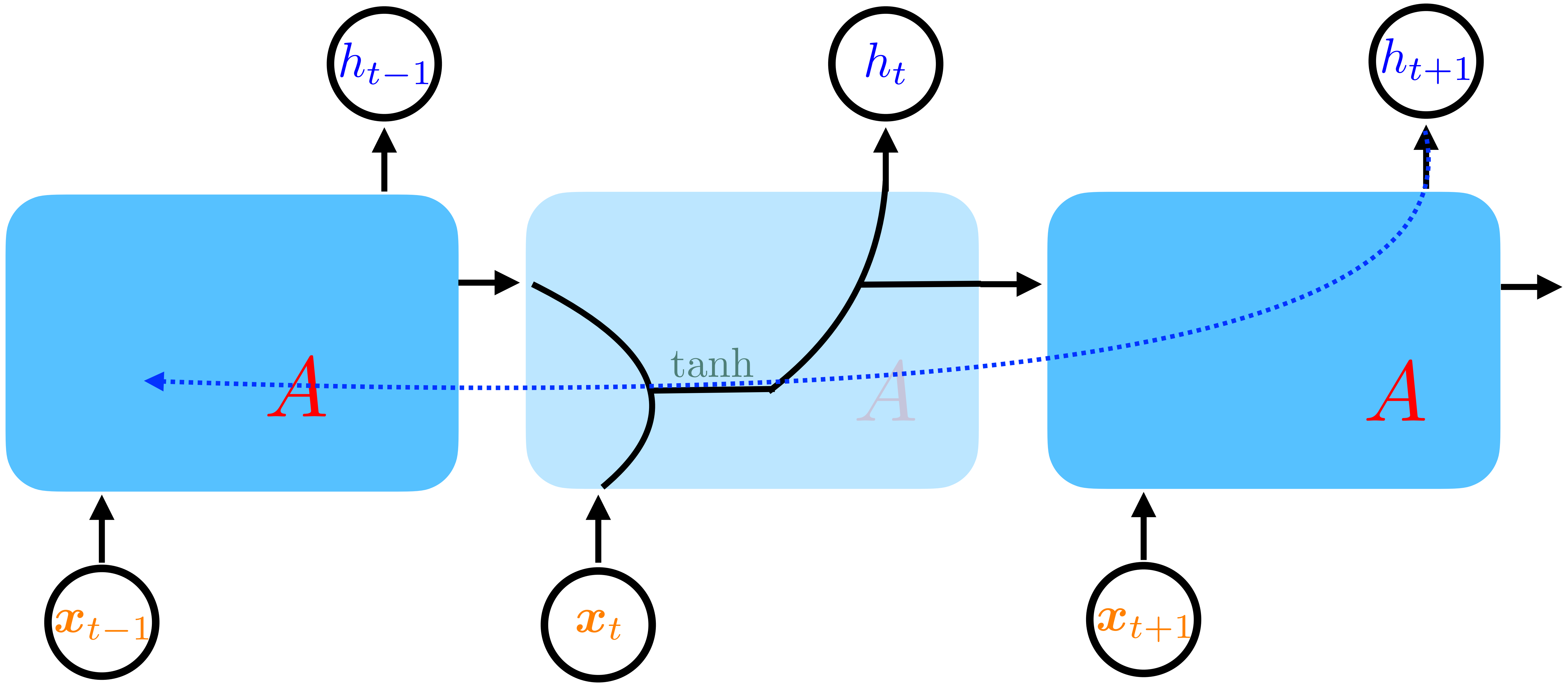
Department of Computer Science, The University of Hong Kong

Recurrent Neural Network

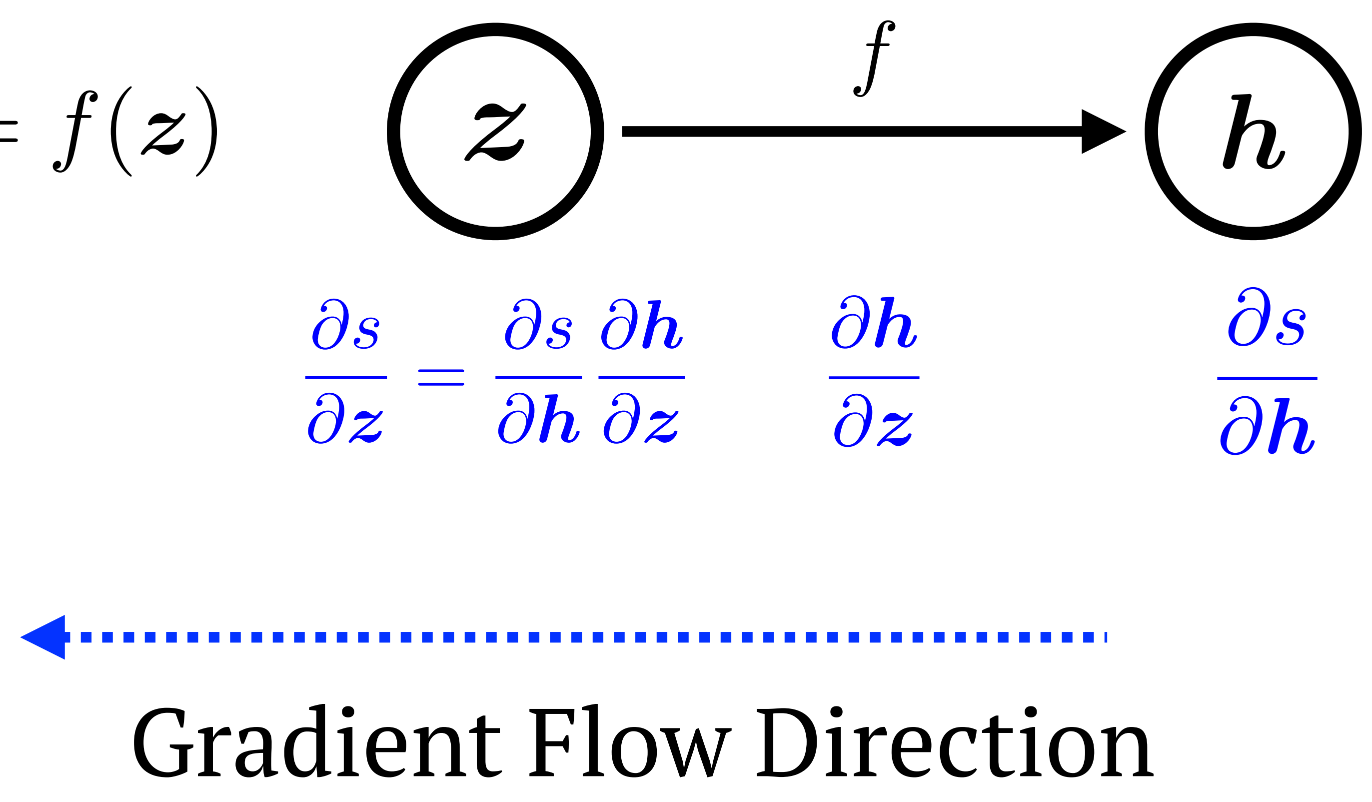


Possibly many steps [O(N)] steps before “yesterday” and “seemed” interact.

Vanishing Gradient in RNNs

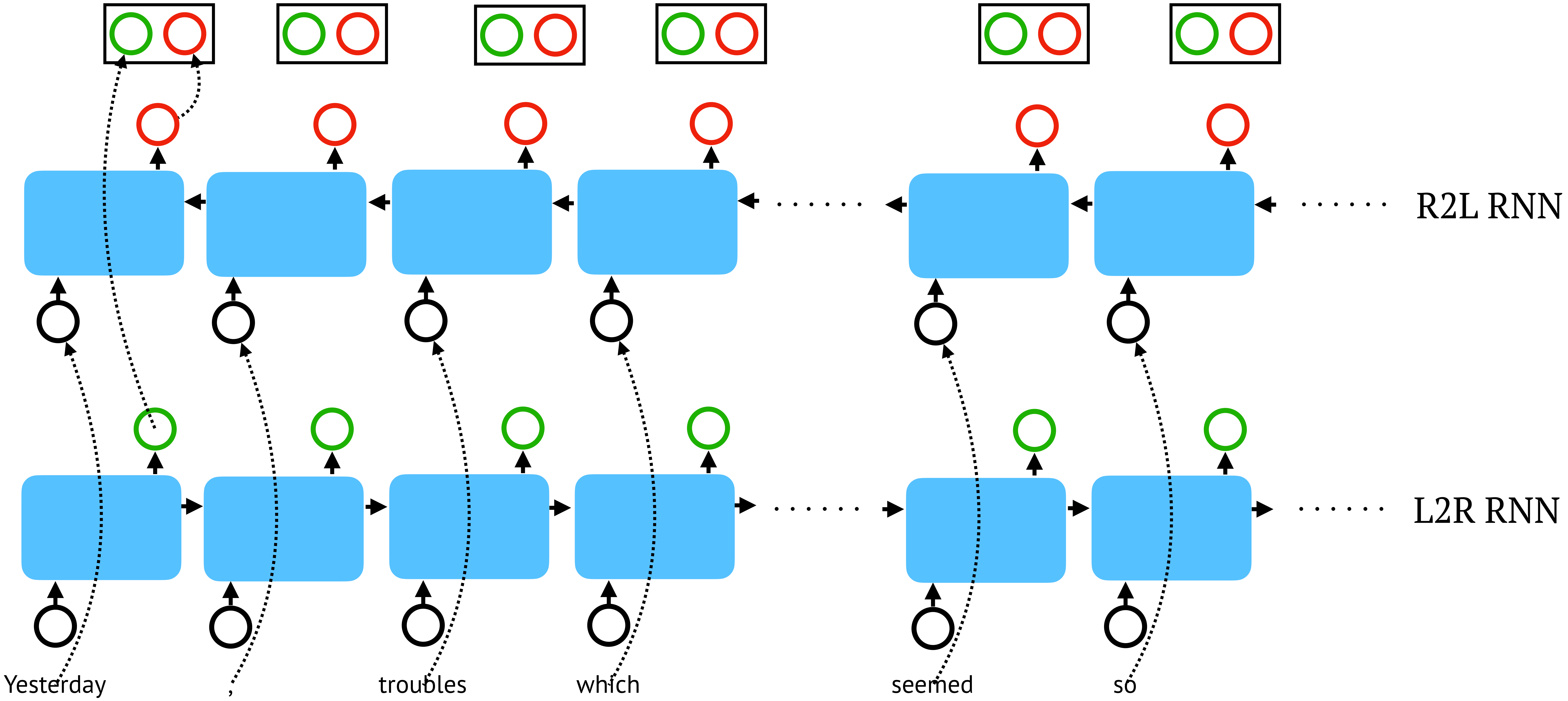


$\boldsymbol{h} = f(\boldsymbol{z})$

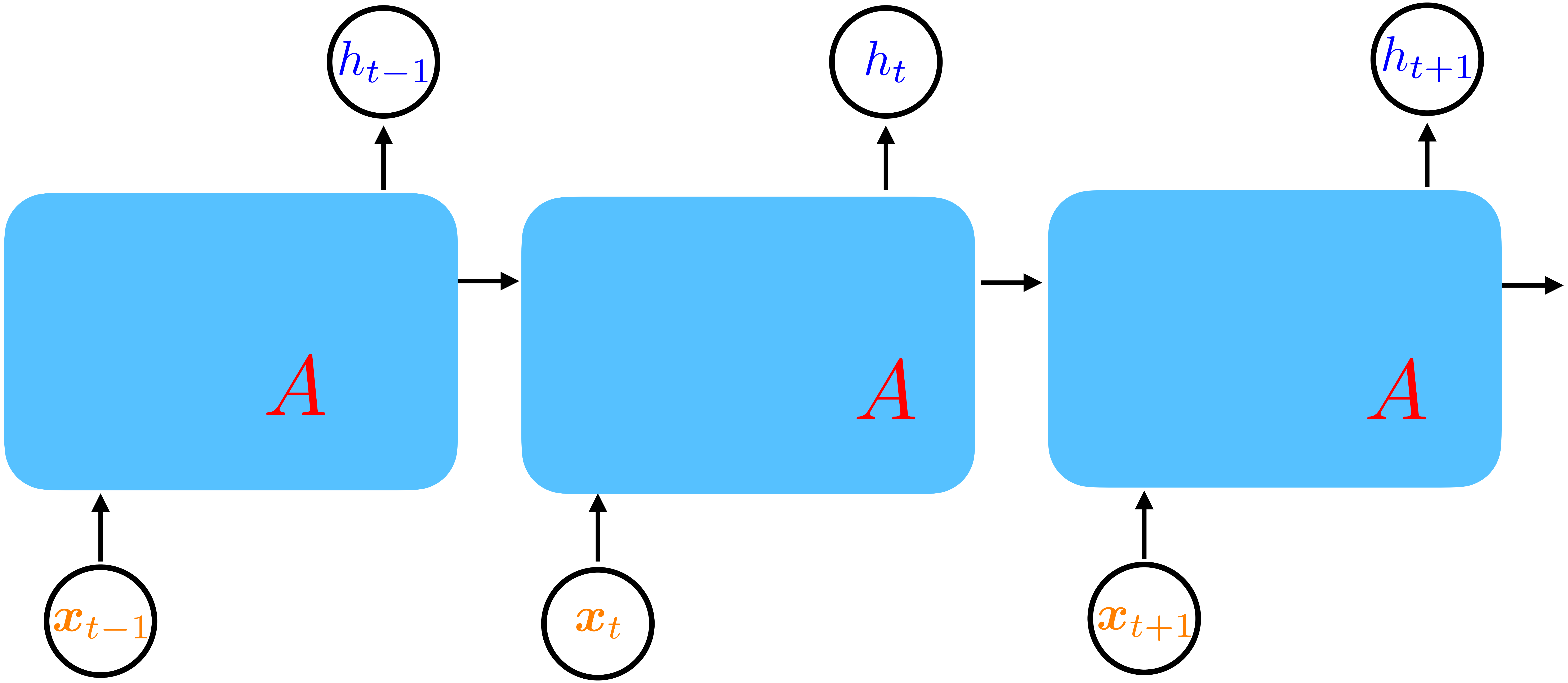


In general, the longer the path, the smaller the gradient signal.

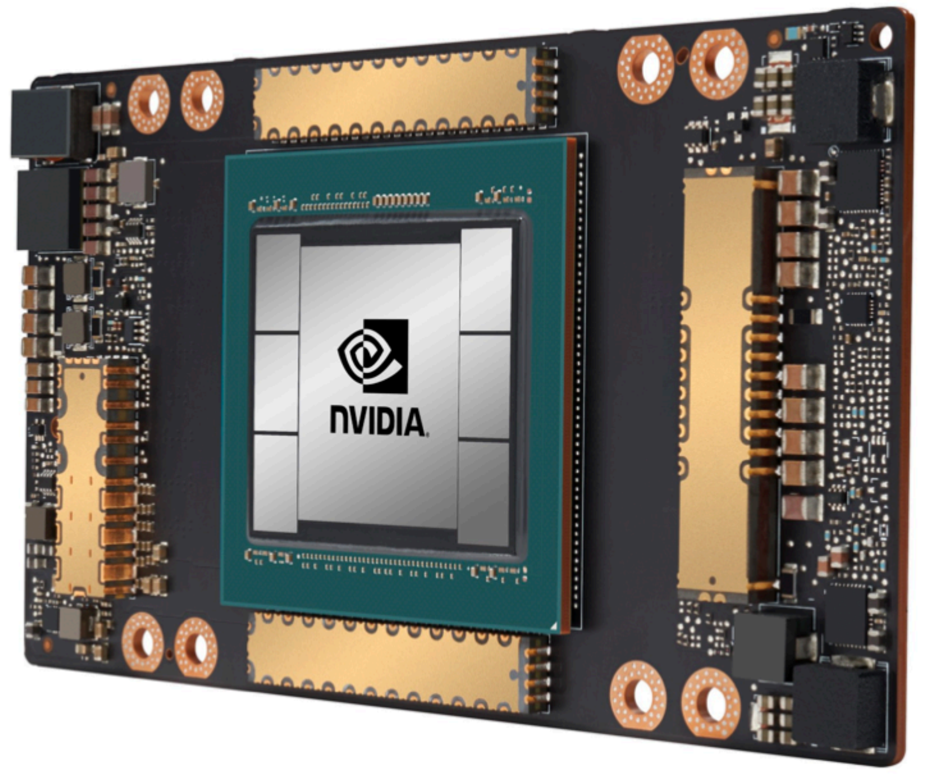
Bidirectional Recurrent Neural Network



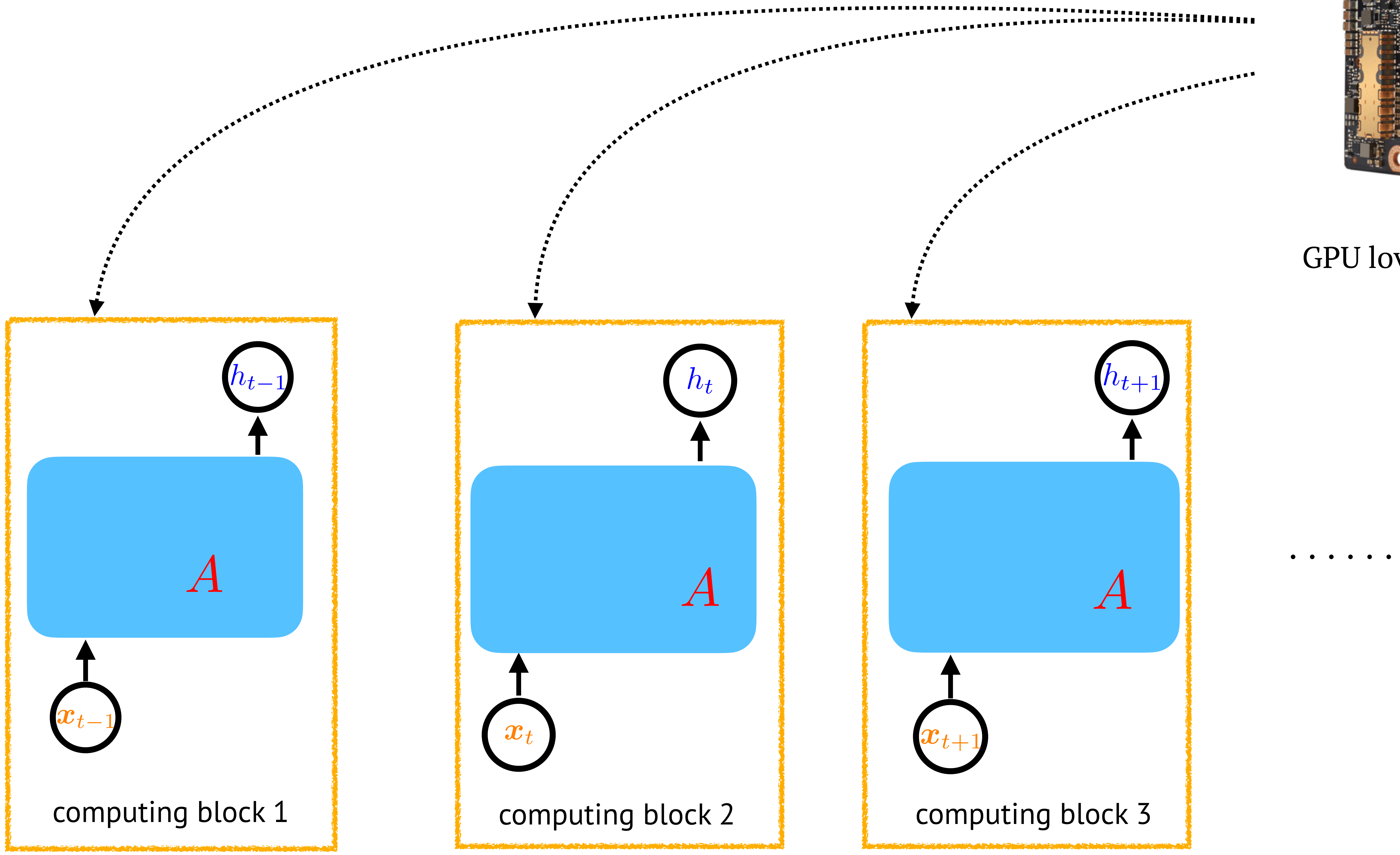
Sequential Computation



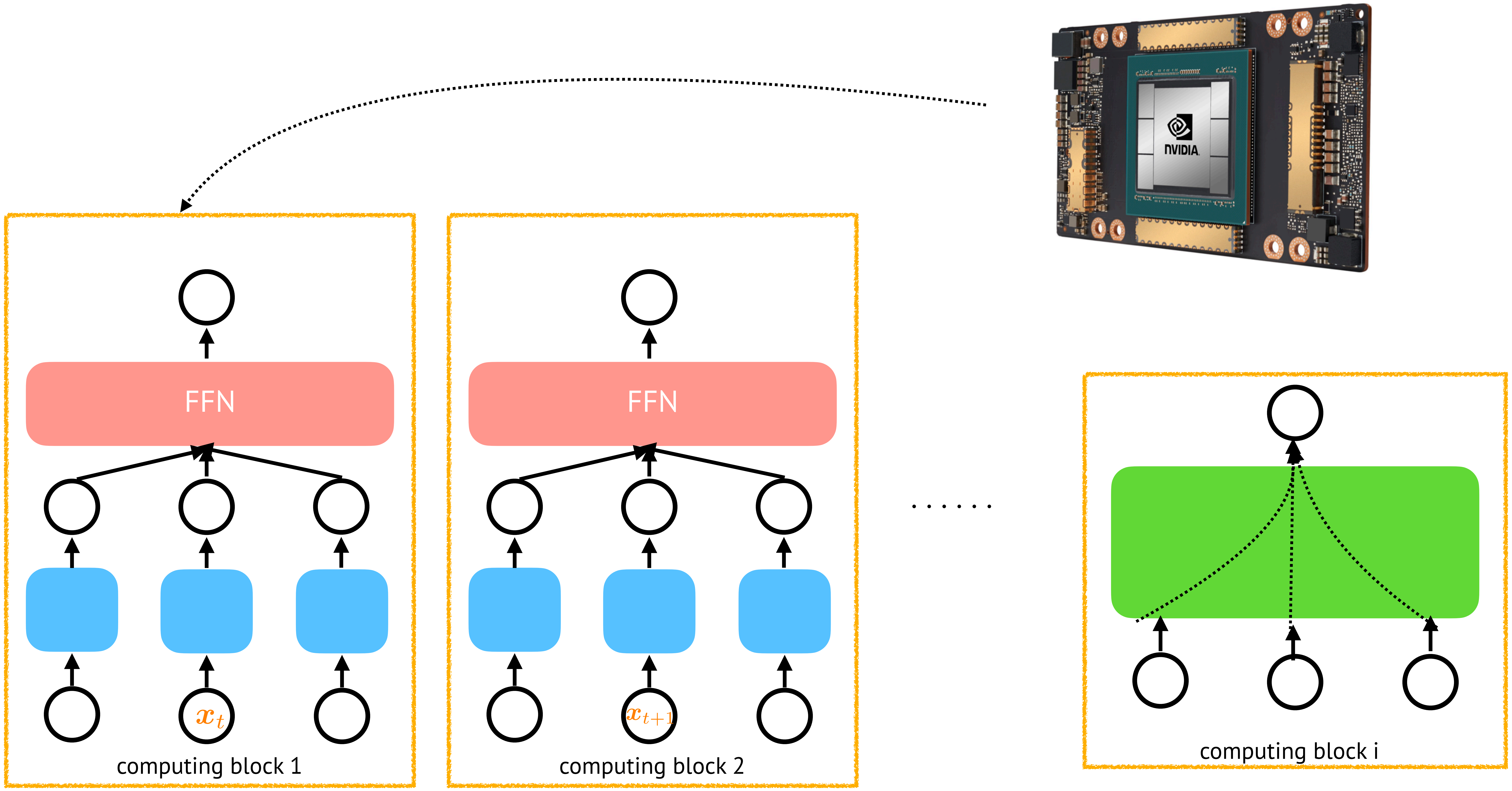
Parallel Computing?



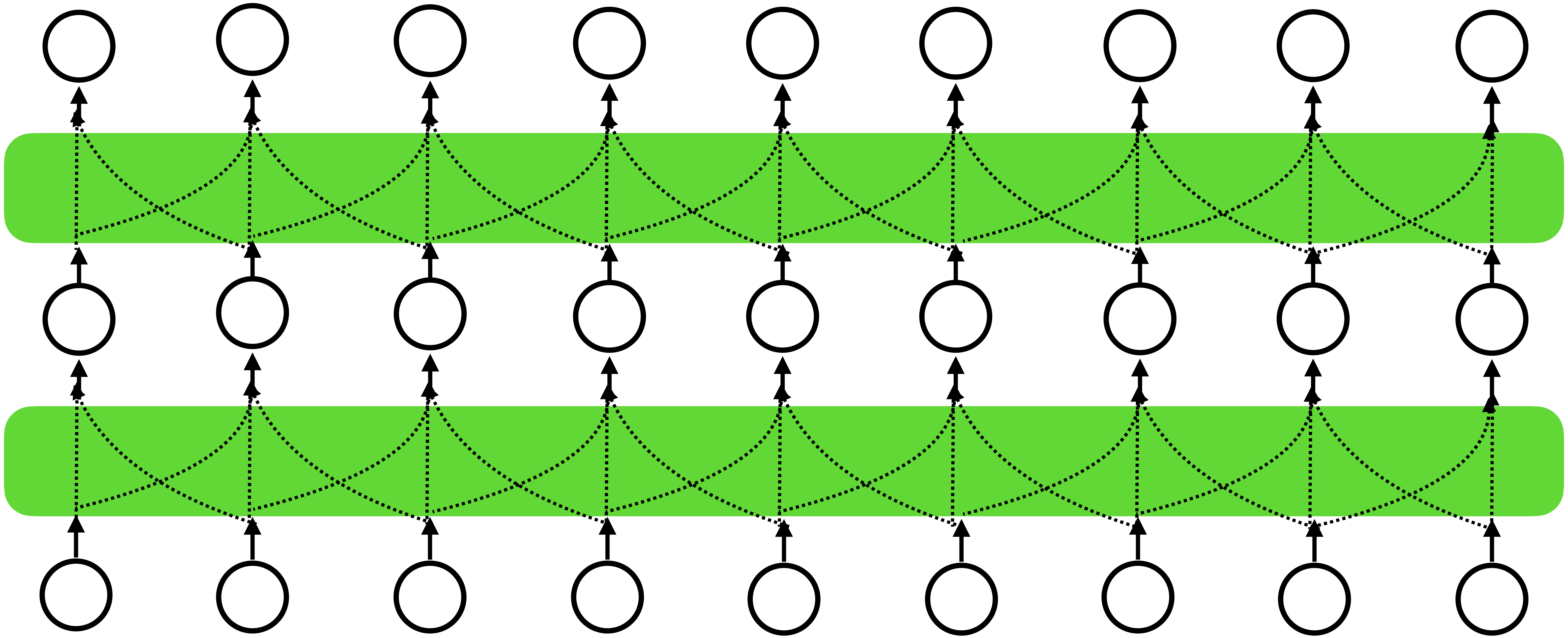
GPU loves parallel computing blocks!



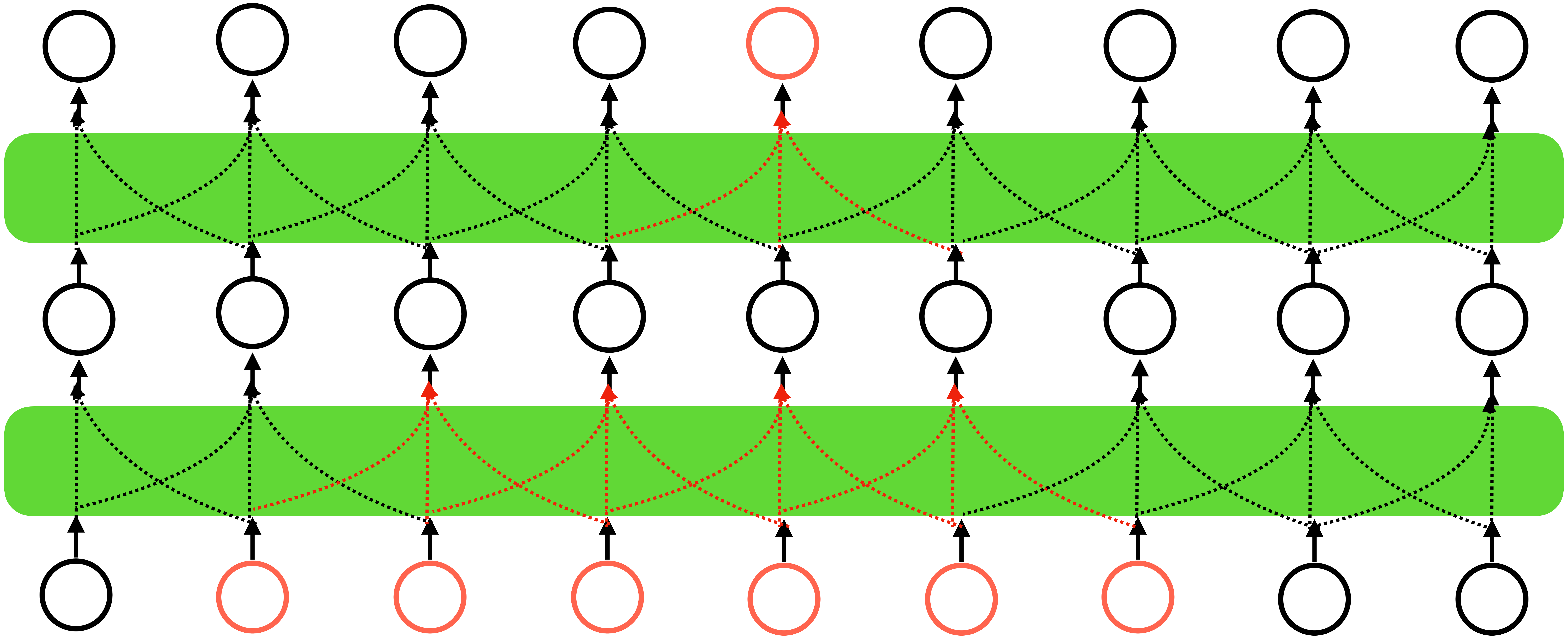
Parallel Computing?



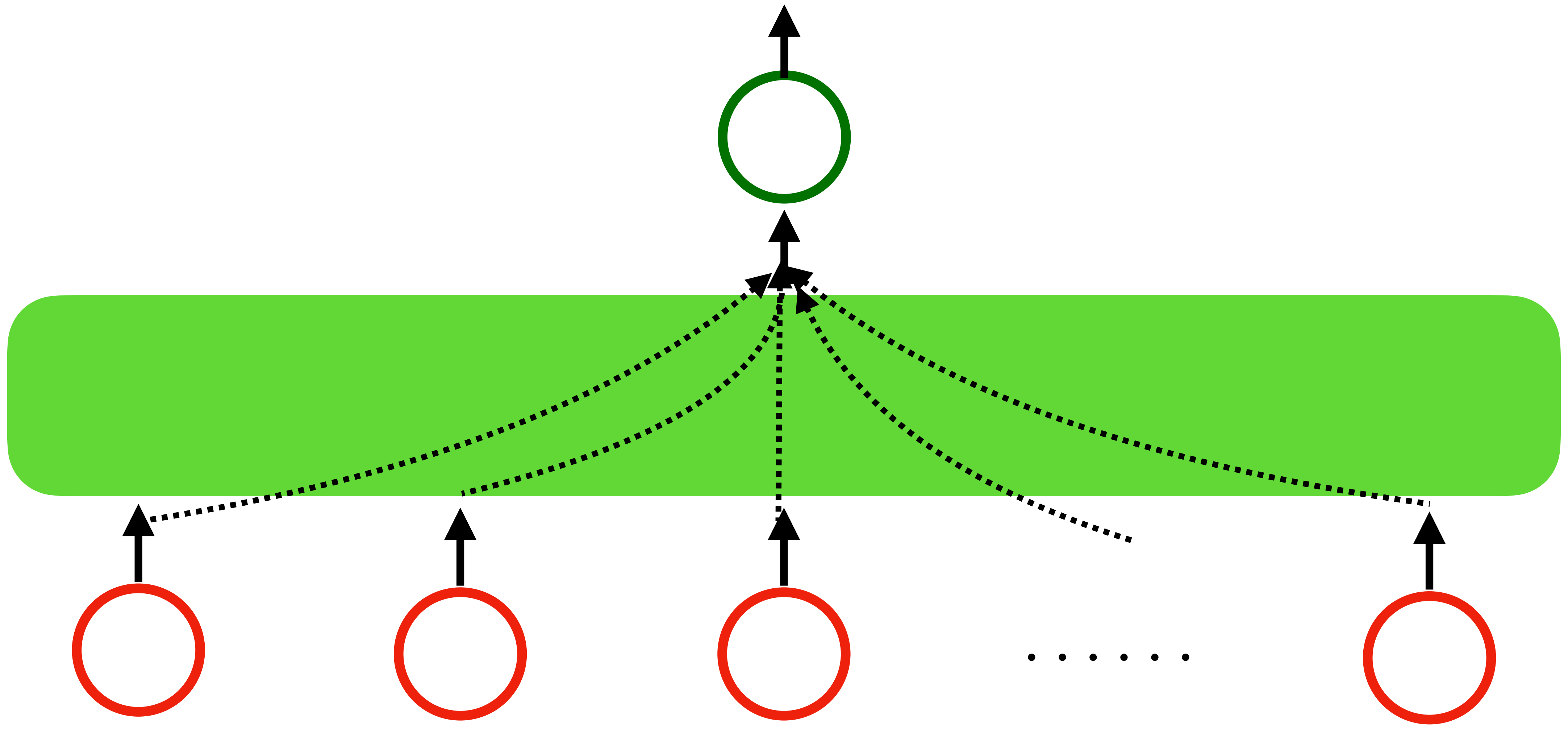
Convolution Style Models



Convolution Style Models

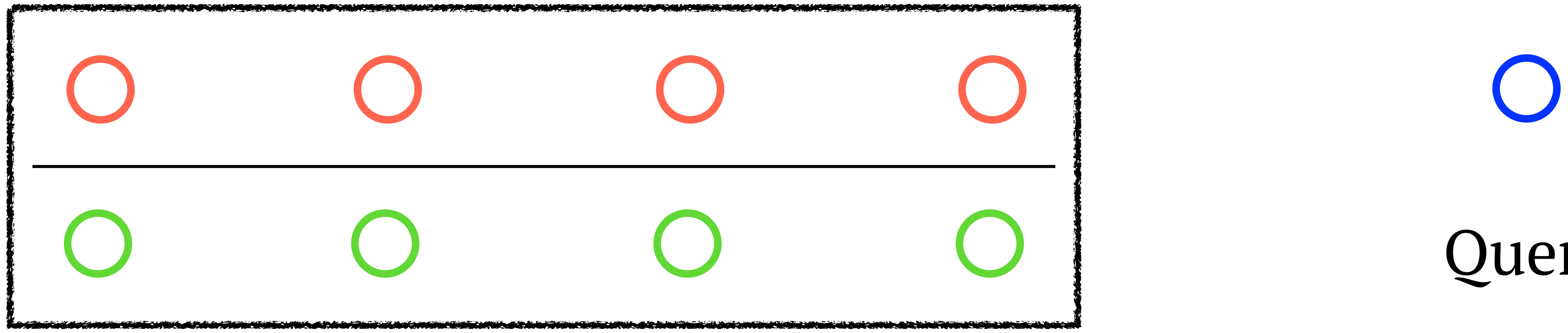


Considering the full sequence as context



How can we achieve this?

Dot-Product-Softmax Attention



Query

Memory (key-value pairs)

$\mathbf{q} \cdot \mathbf{k}_1$

$\mathbf{q} \cdot \mathbf{k}_2$

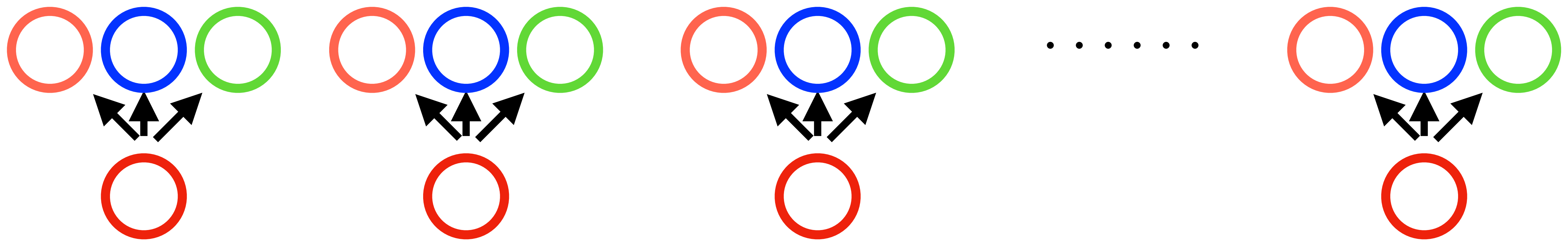
$\mathbf{q} \cdot \mathbf{k}_3$

$\mathbf{q} \cdot \mathbf{k}_4$

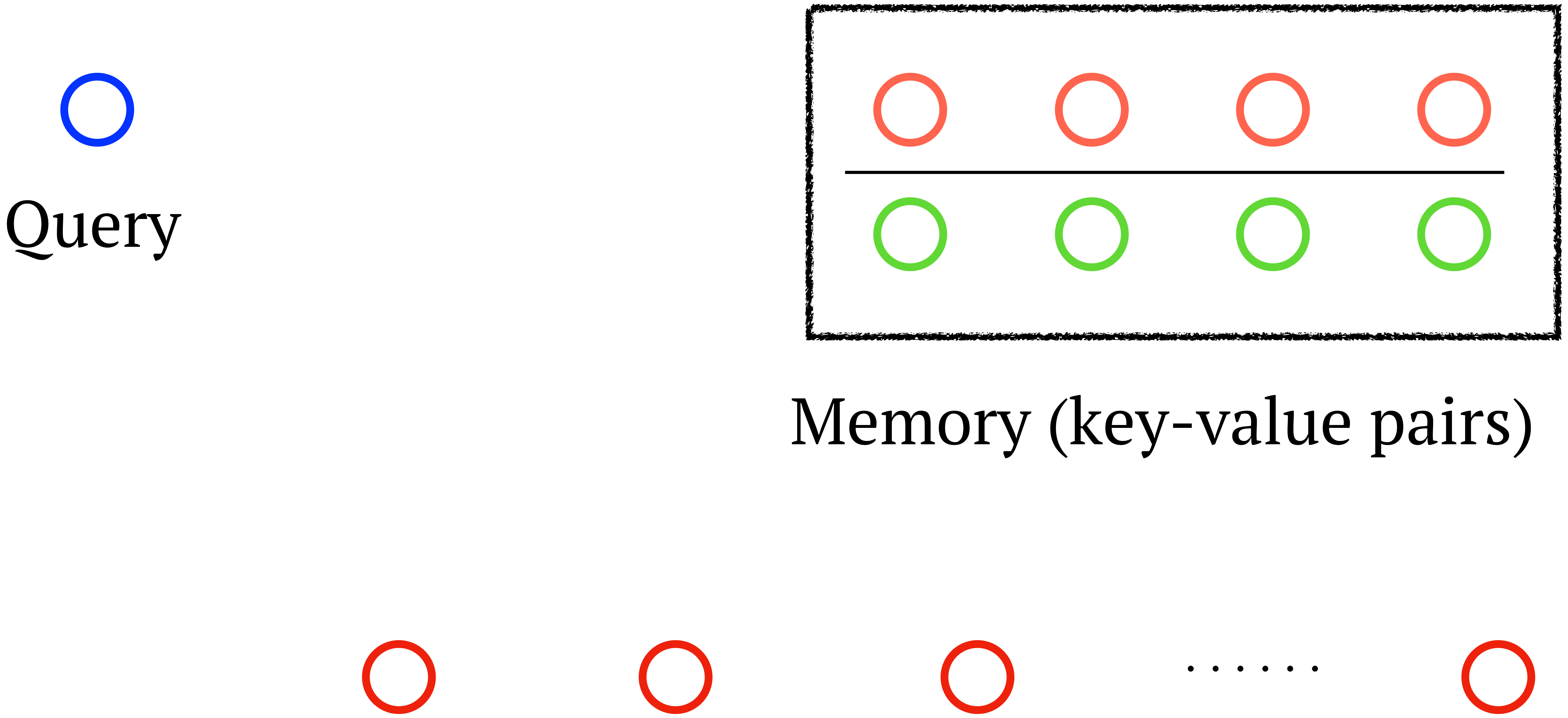
$$\text{softmax}\left(\begin{array}{c} \mathbf{q} \cdot \mathbf{k}_1 \\ \mathbf{q} \cdot \mathbf{k}_2 \\ \mathbf{q} \cdot \mathbf{k}_3 \\ \mathbf{q} \cdot \mathbf{k}_4 \end{array}\right) \rightarrow \begin{bmatrix} 0.6 \\ 0.1 \\ 0.2 \\ 0.1 \end{bmatrix} \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix} \rightarrow 0.6 \; + 0.1 \; + 0.2 \; + 0.1$$

$=$ context vector $\mathbf{c}$

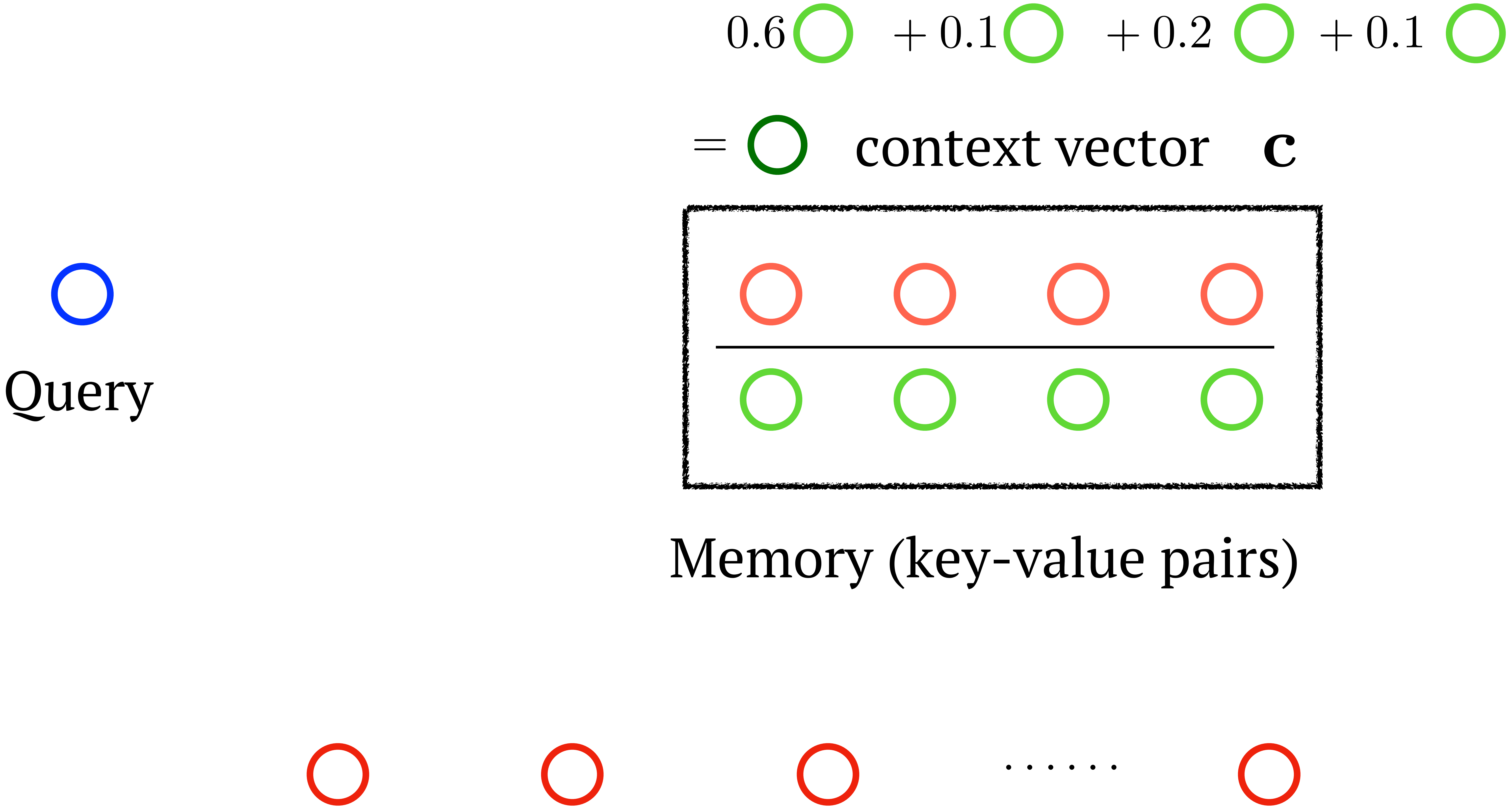
Considering the full sequence as context



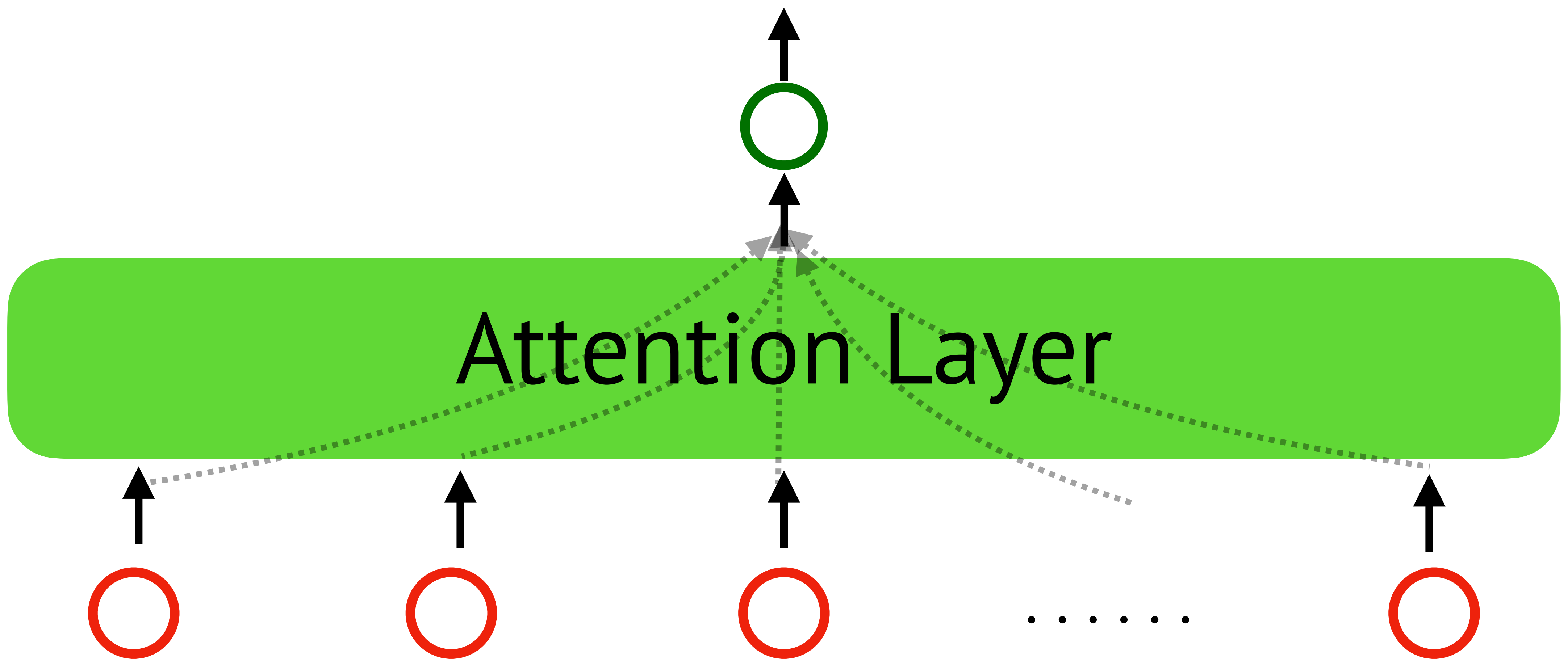
Attention Mechanism



Attention Mechanism

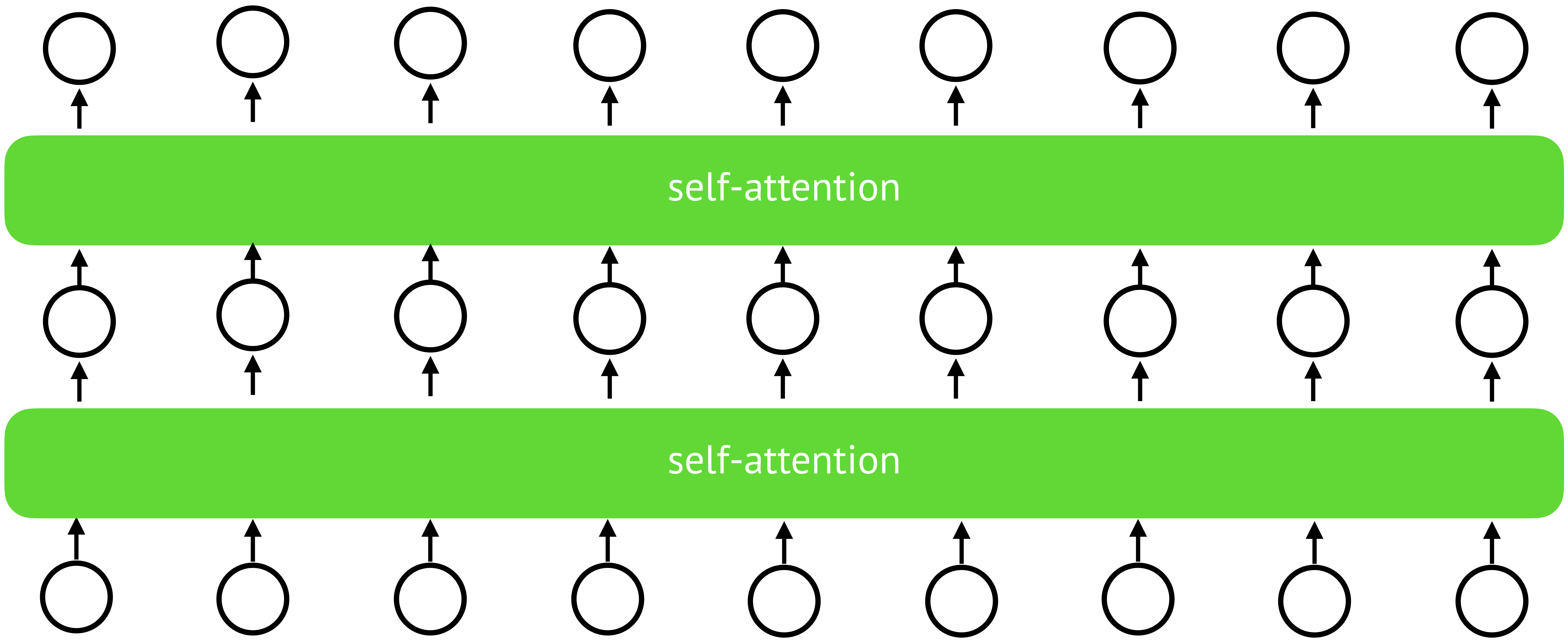


Self-attention

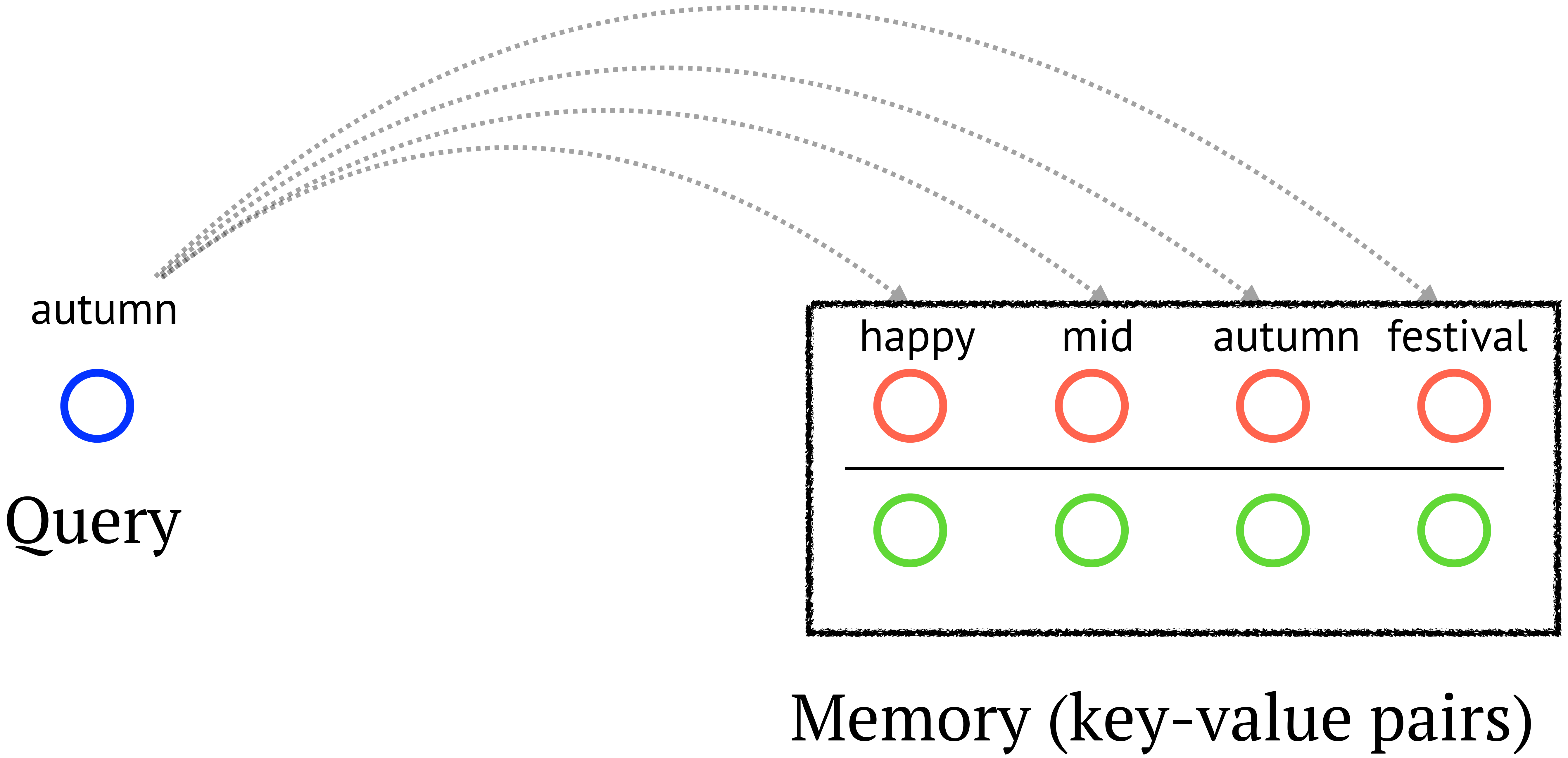


This is almost transformer — except a few things.

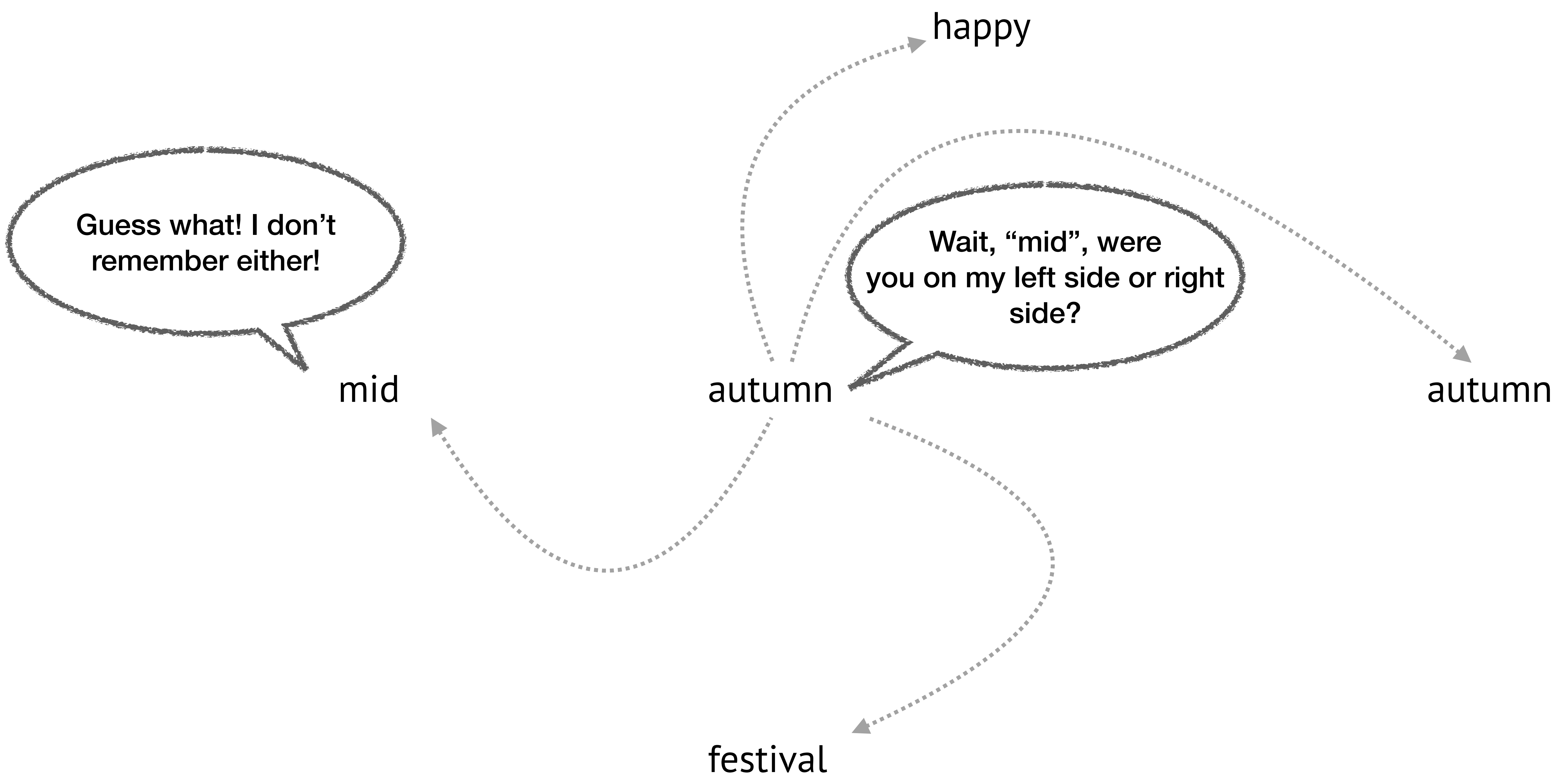
Transformer (almost)



Self-attention in Transformer



Self-attention in Transformer



Positional Embeddings