各式各樣的 Attention

Hung-yi Lee 李宏毅

Prerequisite

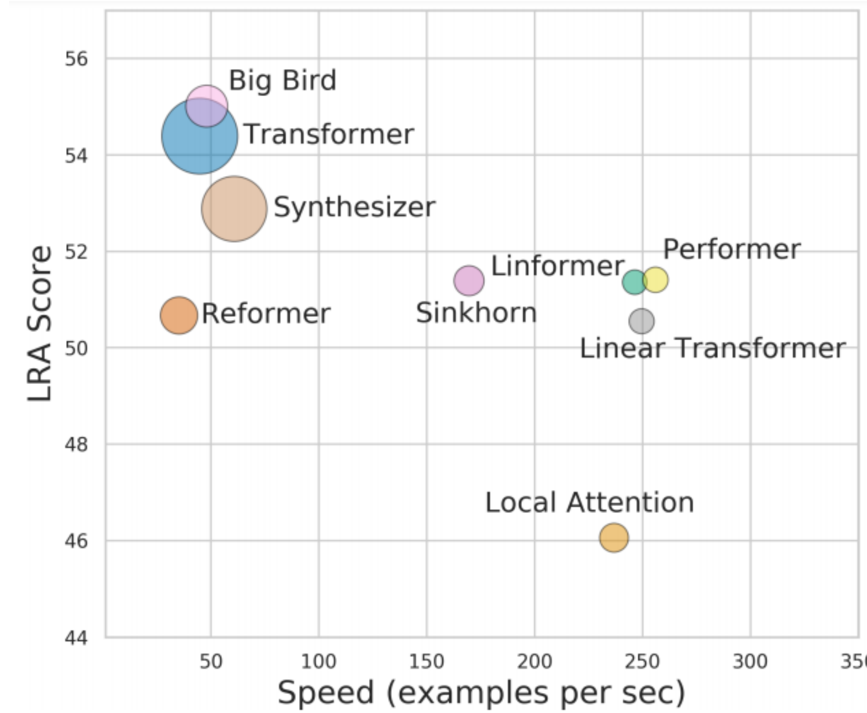
https://youtu.be/hYdO9CscNes

【機器學習2021】自注意力機制 (Self-attention) (上)

https://youtu.be/gmsMY5kc-zw

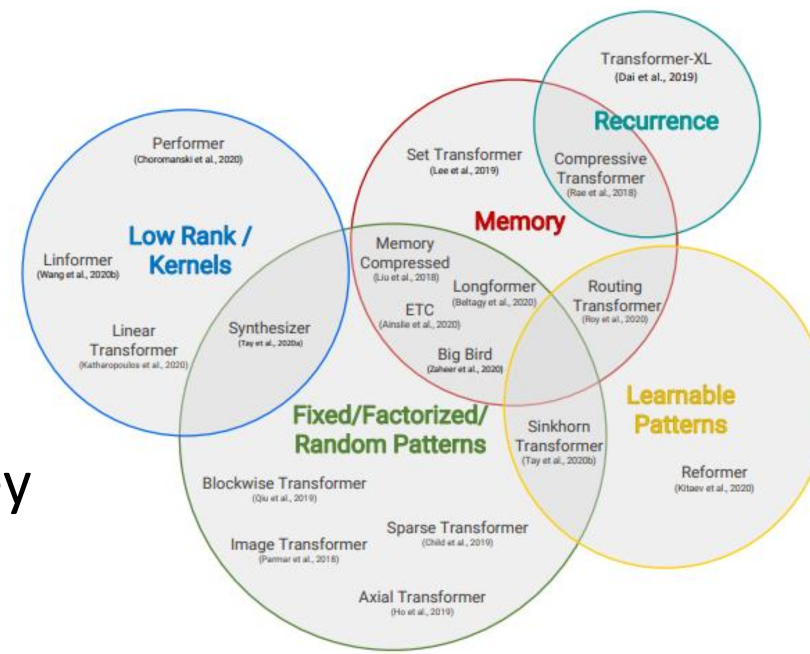
【機器學習2021】自注意力機制 (Self-attention) (下)

To Learn More …



Long Range Arena: A Benchmark for Efficient Transformers

https://arxiv.org/abs/2011.04006

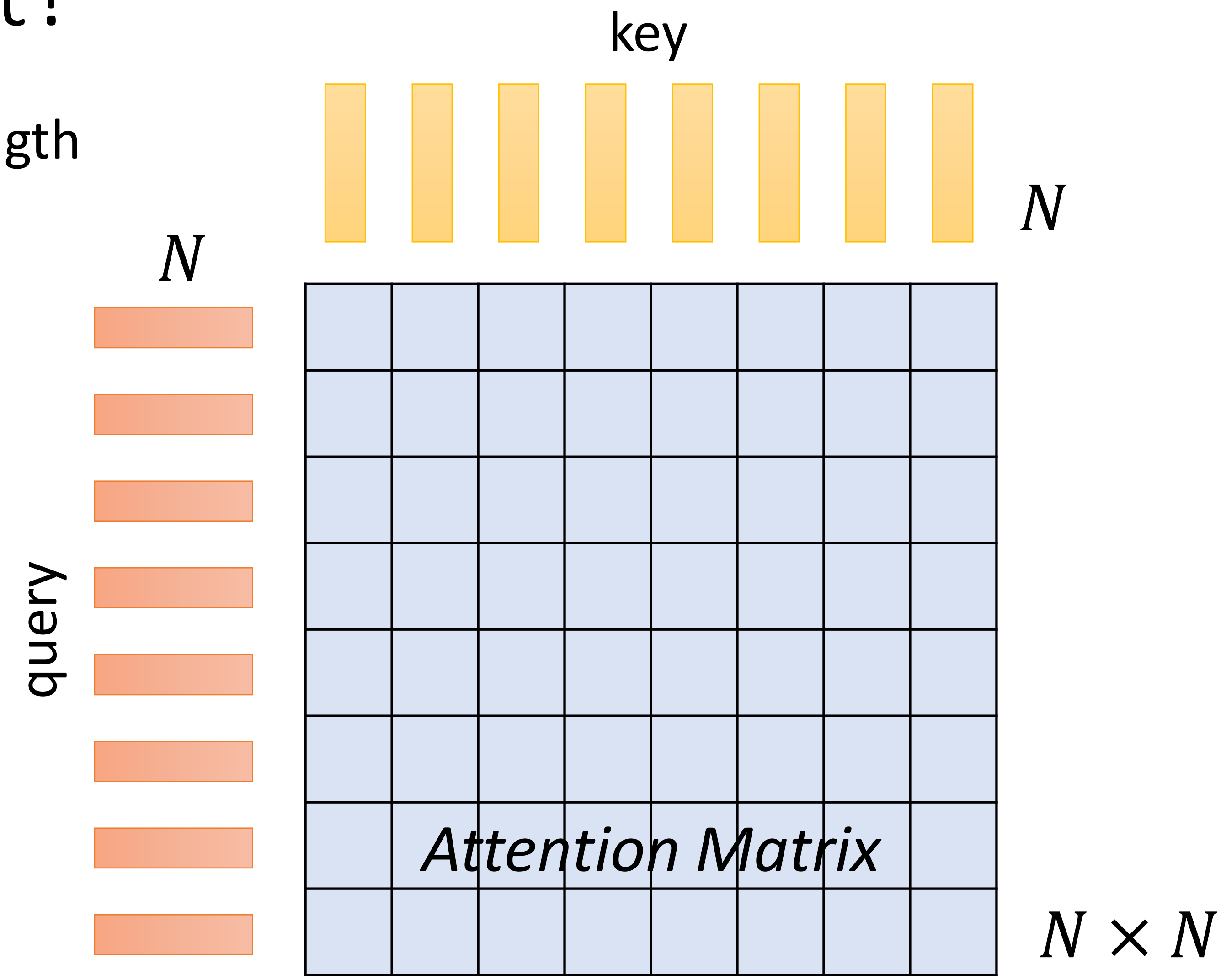


Efficient Transformers: A Survey

https://arxiv.org/abs/2009.06732

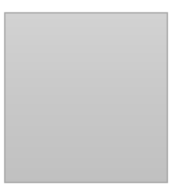
How to make self-attention efficient?

Sequence length $= N$



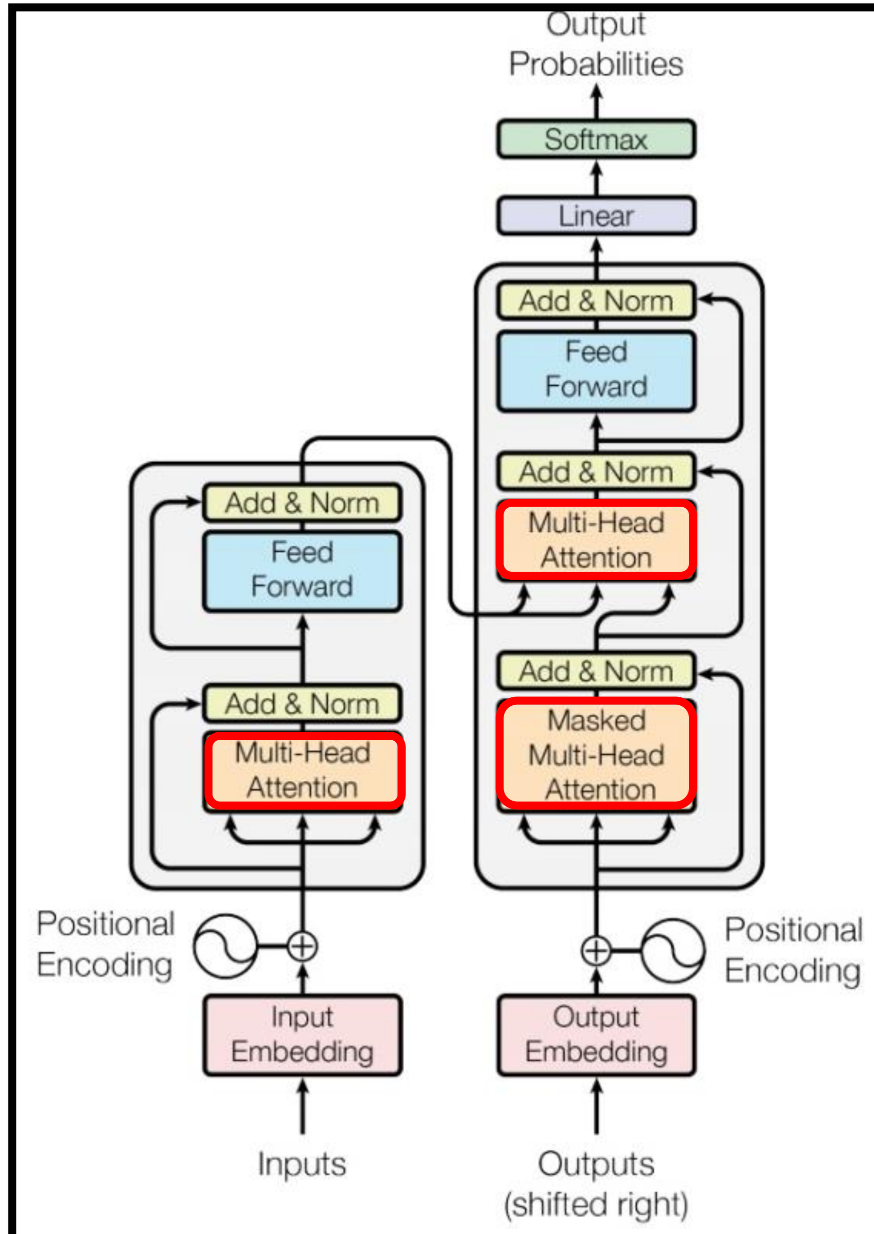
Notice

- Self-attention is only a module in a larger network.
- Self-attention dominates computation when N is large.
- Usually developed for image processing

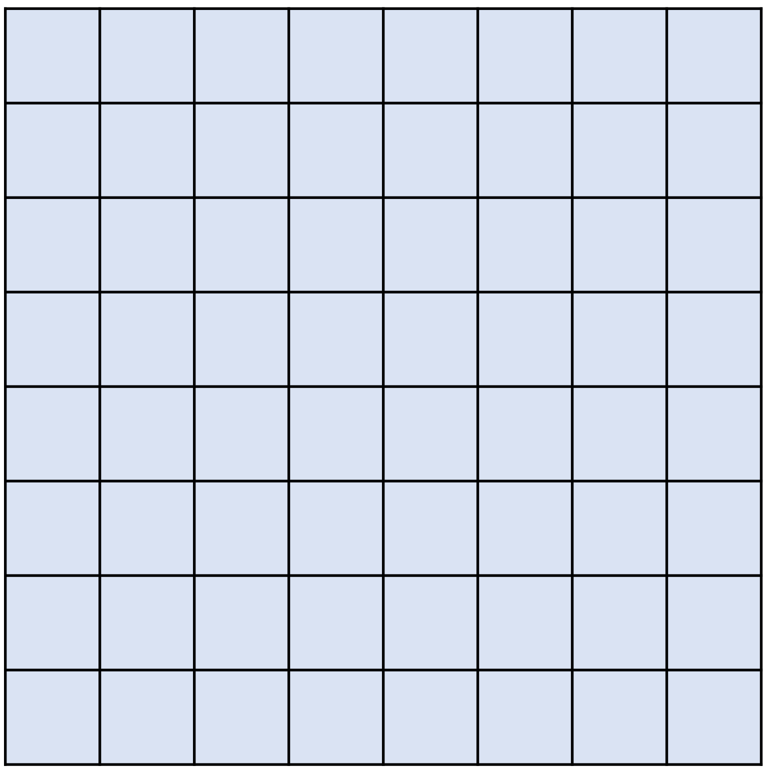
256 

256

$$N = 256 * 256$$

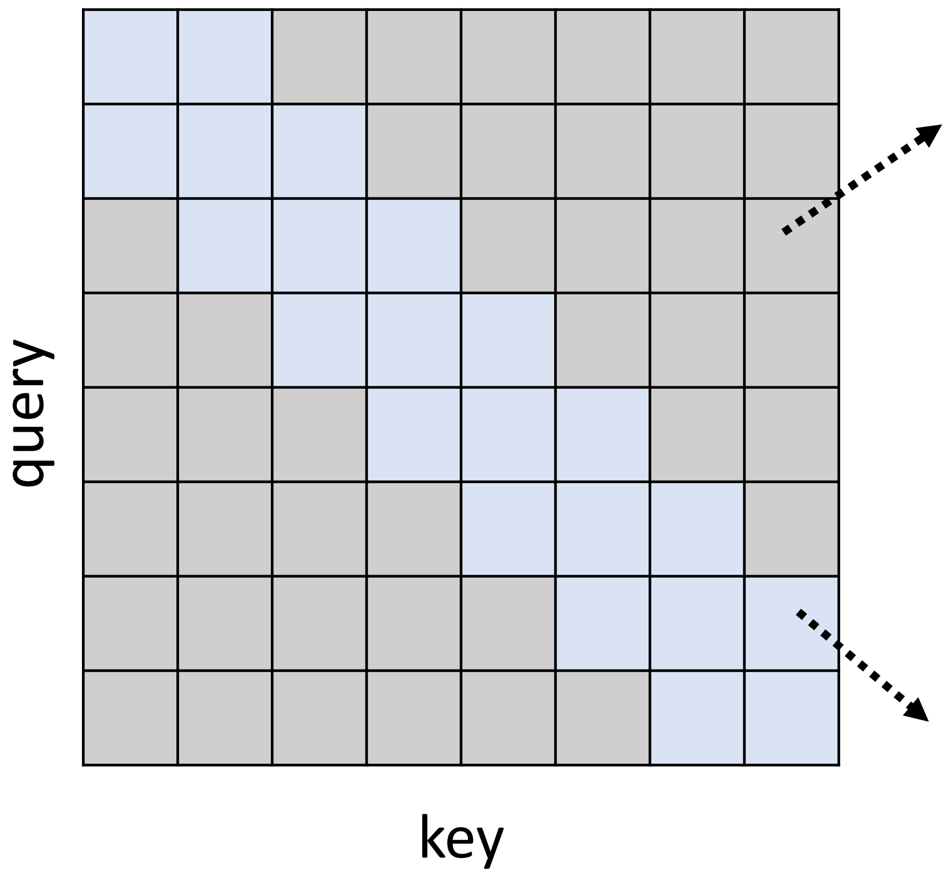


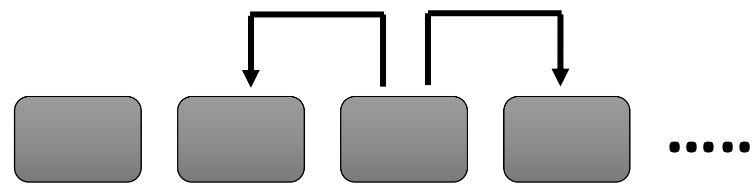
Skip Some Calculations with Human Knowledge



Can we fill in some values with human knowledge?

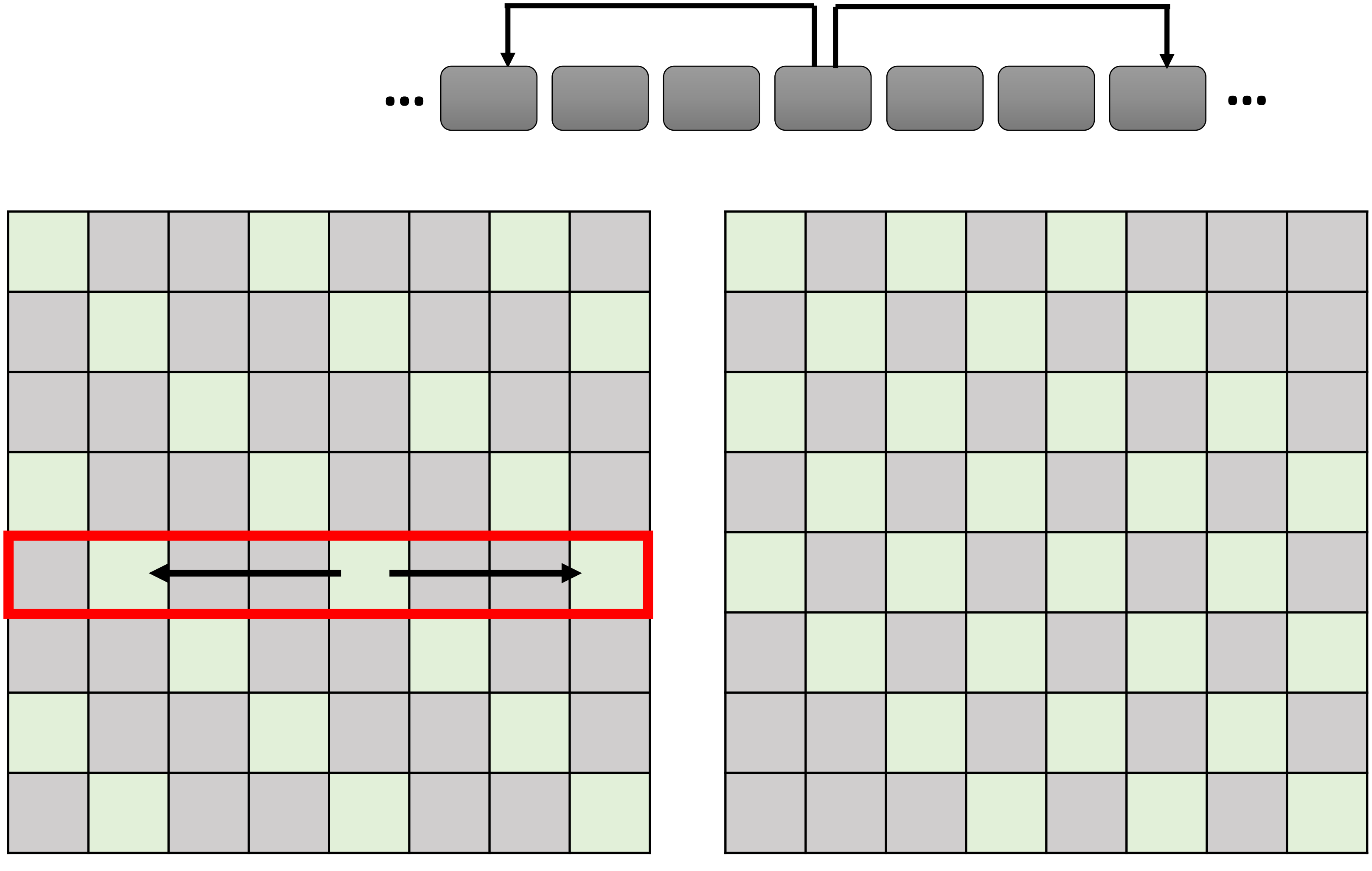
Local Attention / Truncated Attention





Similar with CNN

Stride Attention

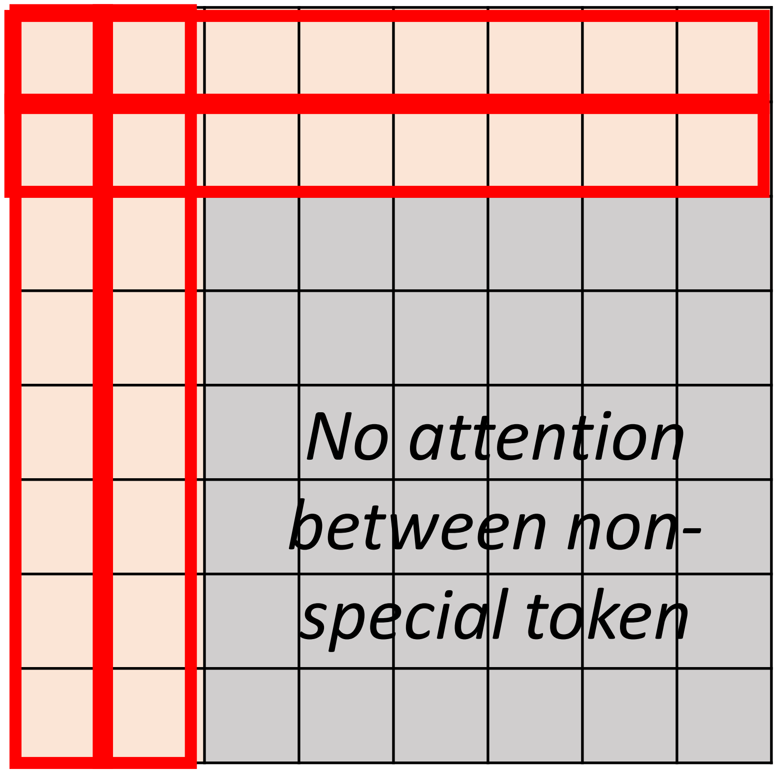


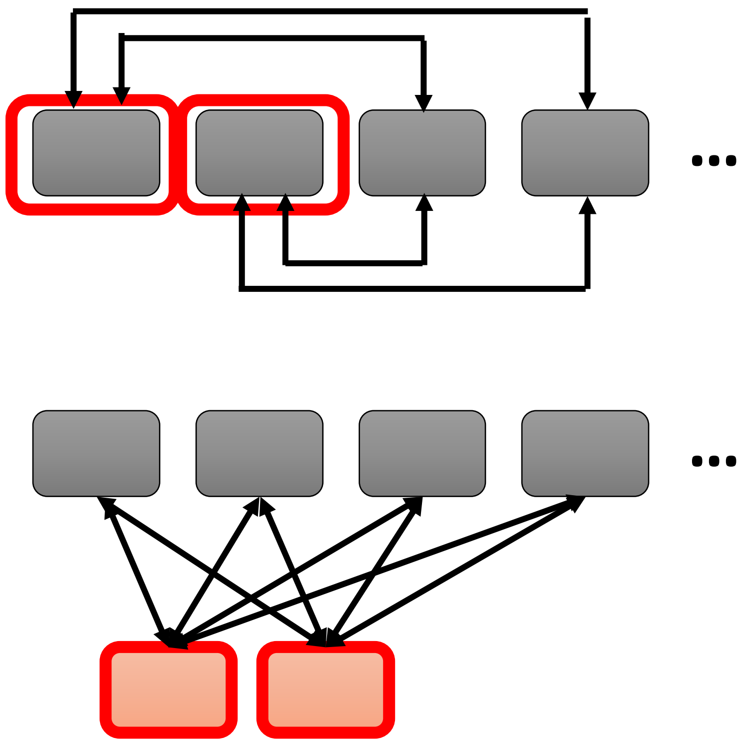
Global Attention

special token = “token中的里長伯“

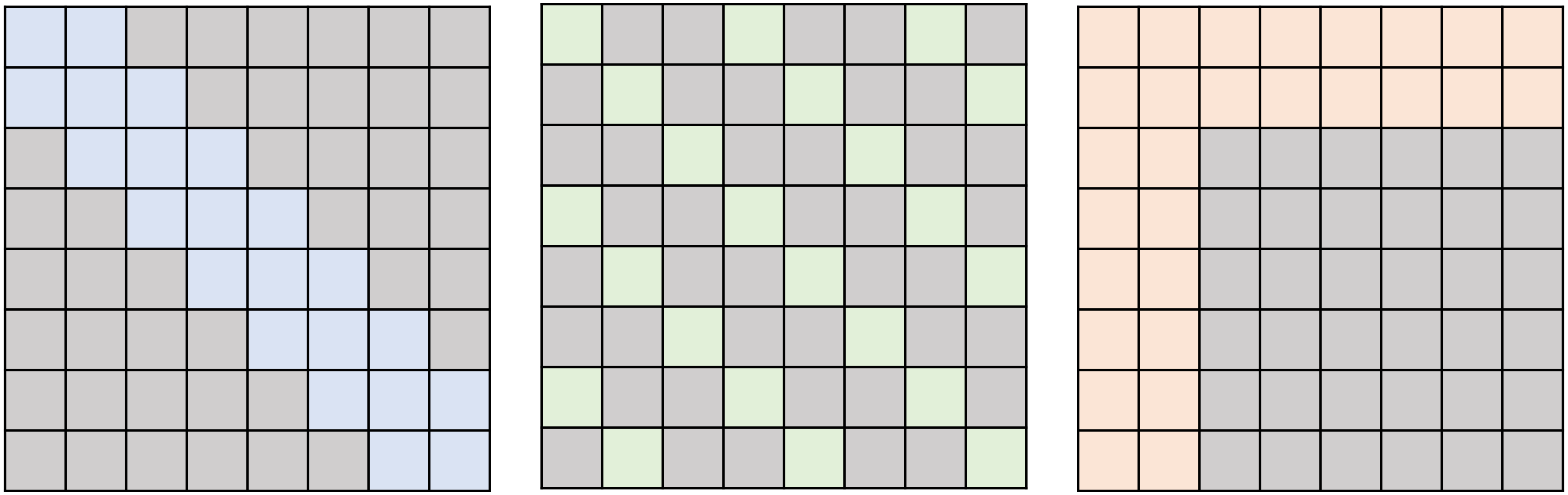
Add special token into original sequence

- Attend to every token → collect global information
- Attended by every token → it knows global information





Many Different Choices …



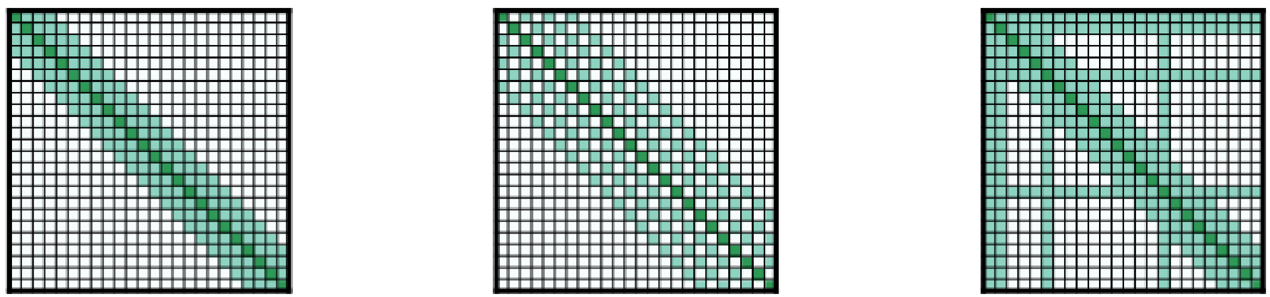
小孩子才做選擇．．．

Different heads use different patterns.

Many Different Choices …

- Longformer

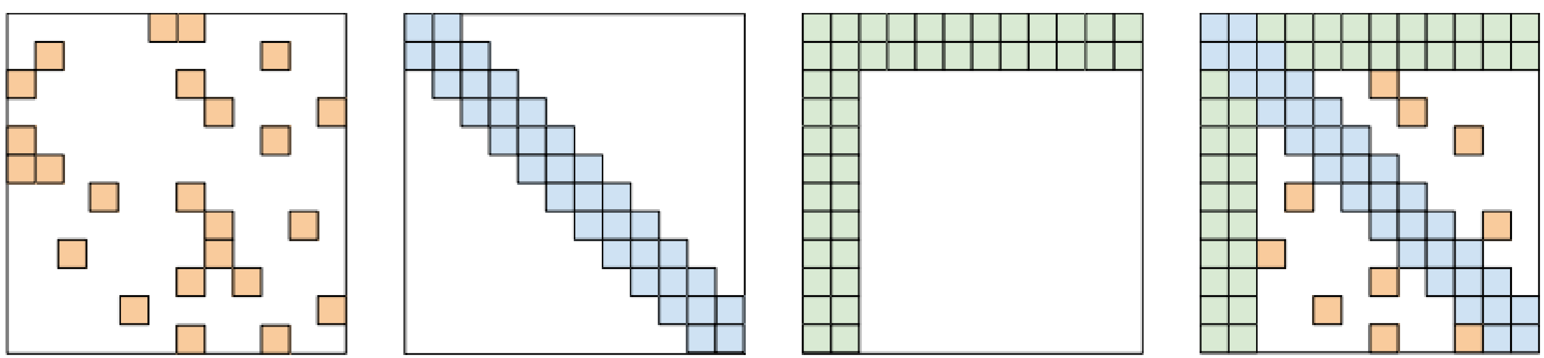
https://arxiv.org/abs/2004.05150



(b) Sliding window attention (c) Dilated sliding window (d) Global+sliding window

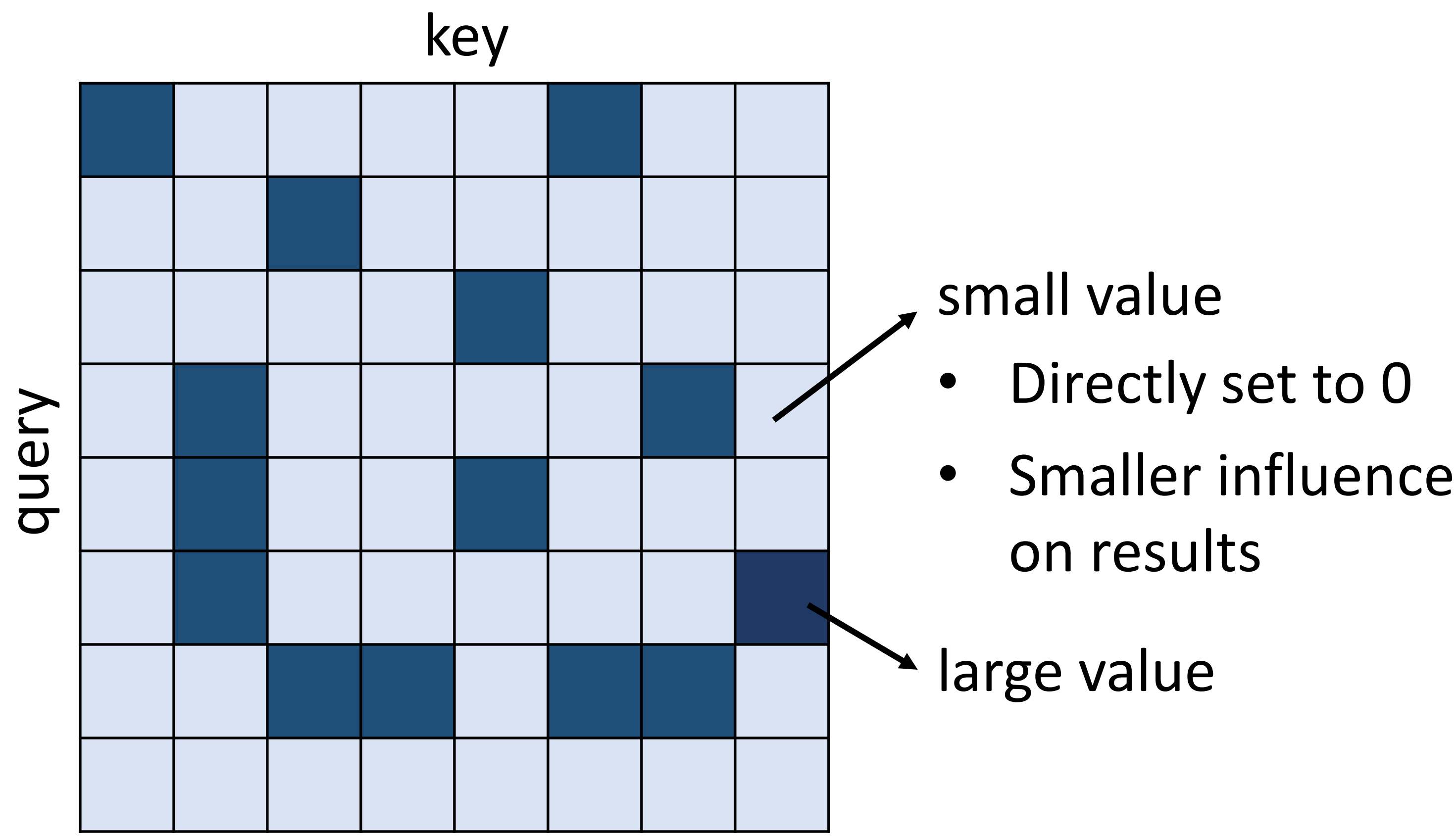
- Big Bird

https://arxiv.org/abs/2007.14062



(a) Random attention (b) Window attention (c) Global Attention (d) BIGBIRD

Can we only focus on Critical Parts?



How to quickly estimate the portion with small attention weights?

Clustering

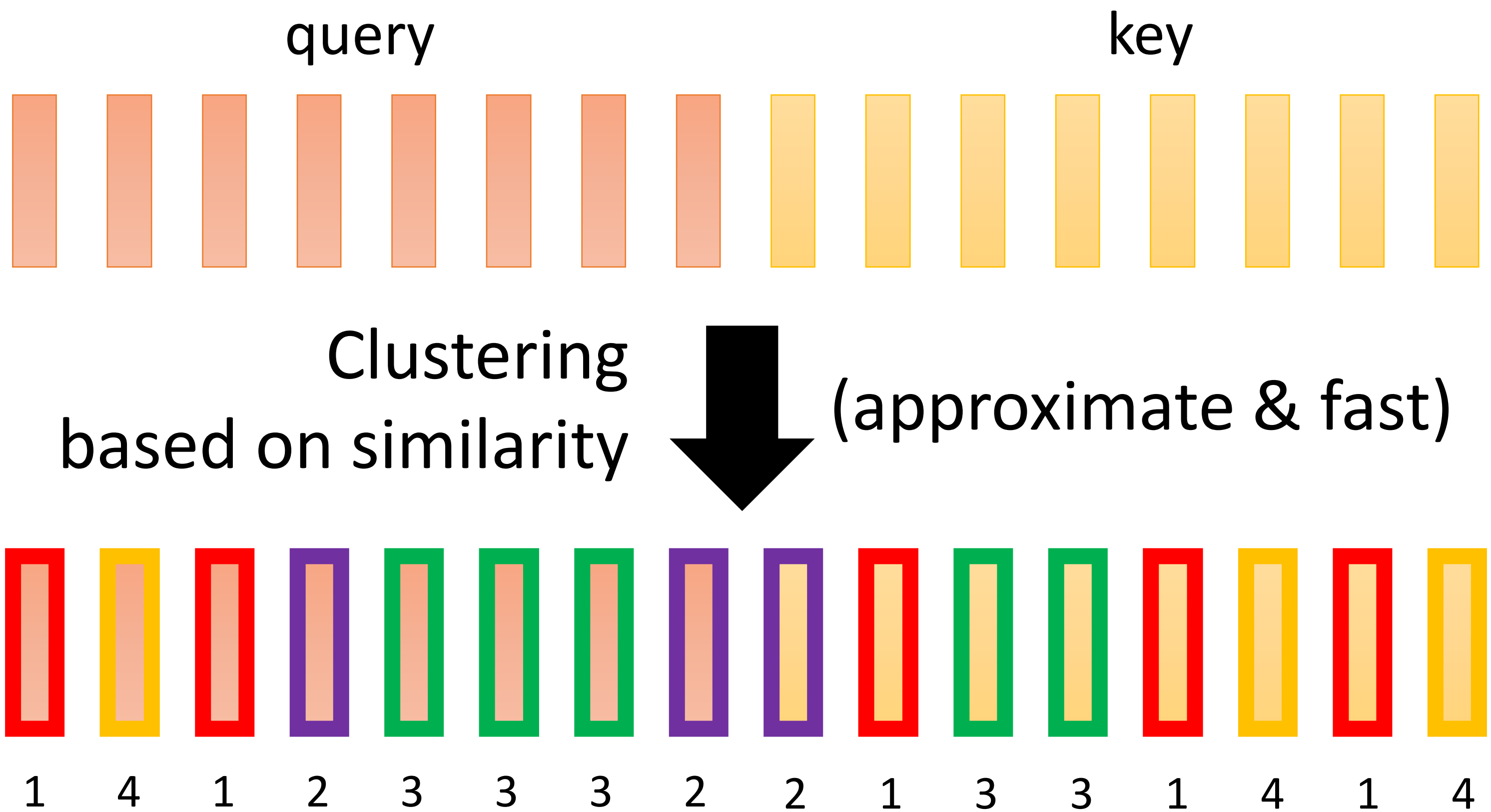
Reformer

https://openreview.net/forum?id=rkgNKkHtvB

Routing Transformer

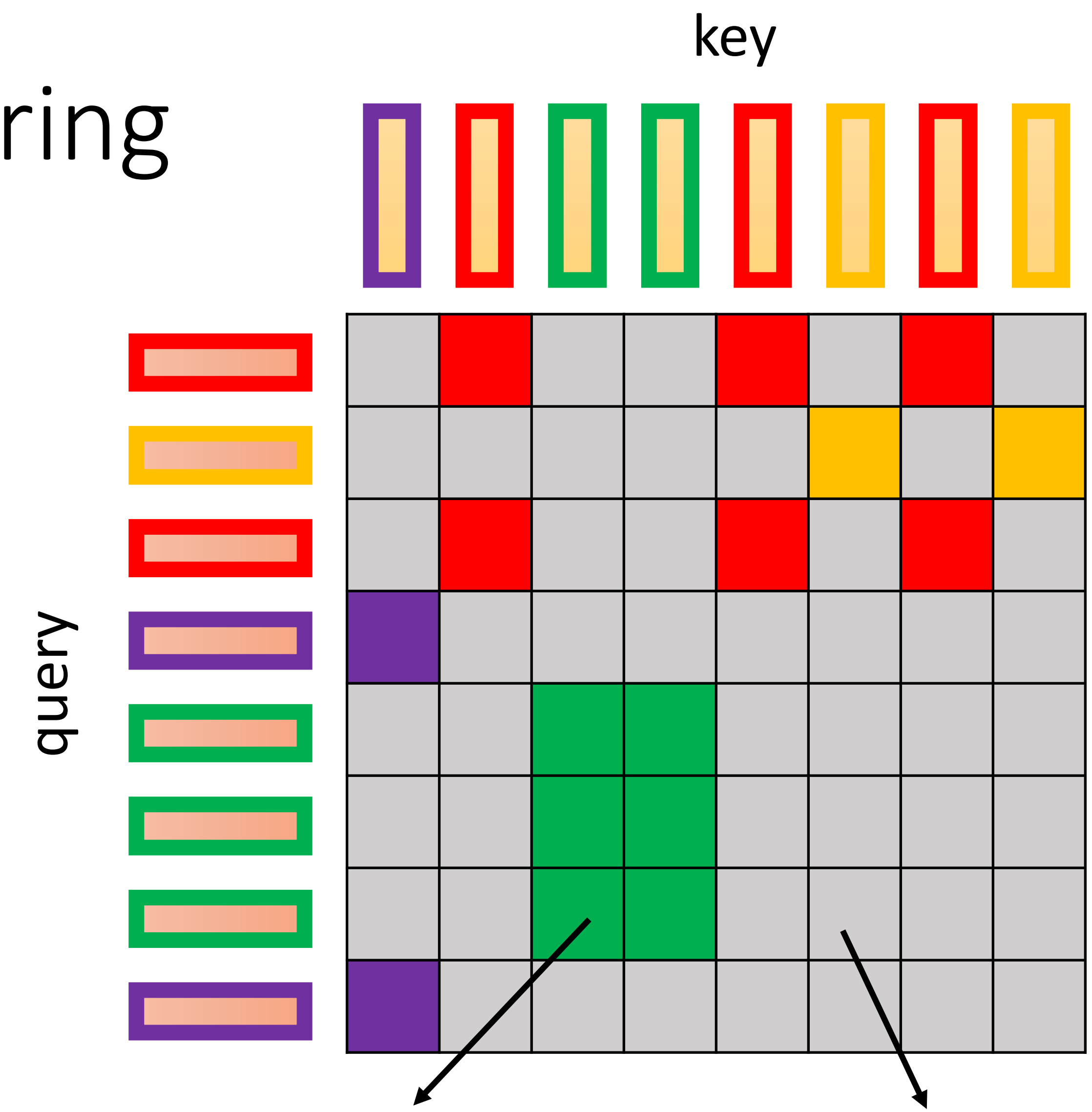
https://arxiv.org/abs/2003.05997

Step 1



Clustering

Step 2



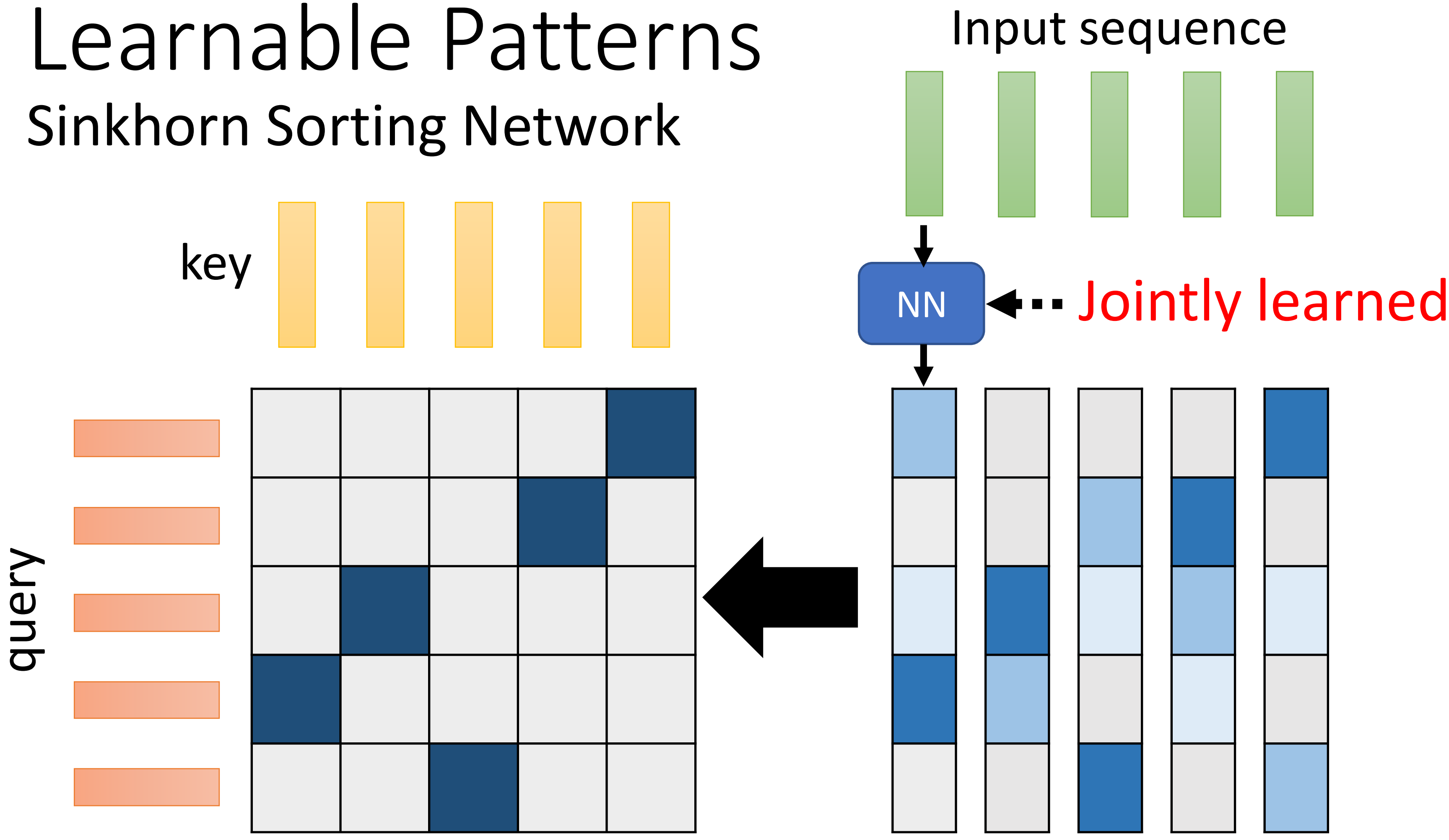
Belong to the same cluster, then calculate attention weight

Not the same cluster, set to 0

https://arxiv.org/abs/2002.11296

Learnable Patterns

Sinkhorn Sorting Network



A grid should be skipped or not is decided by another learned module

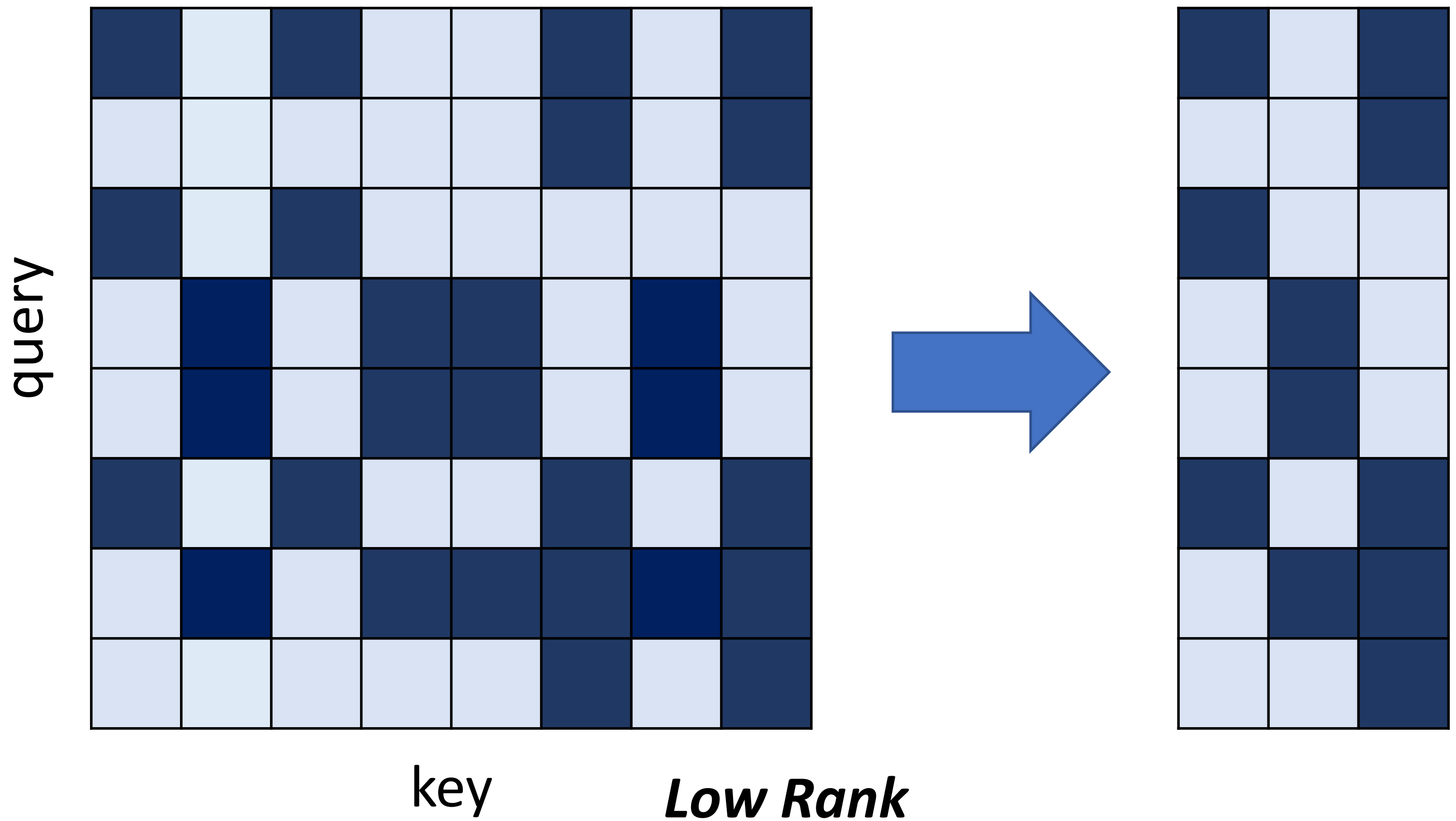
(simplified version)

Do we need full attention matrix?

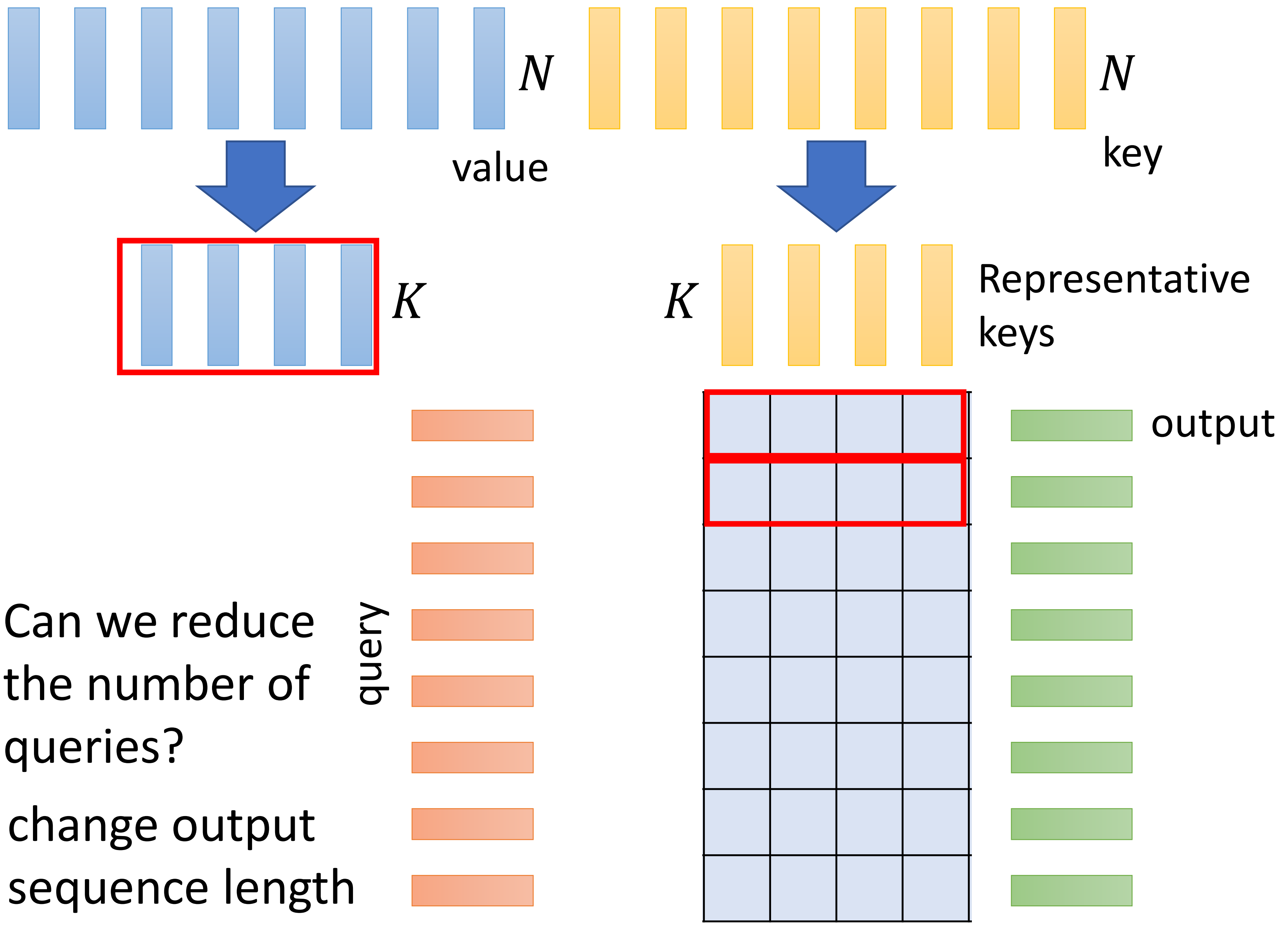
Linformer

https://arxiv.org/abs/2006.04768

Many redundant columns



Low Rank

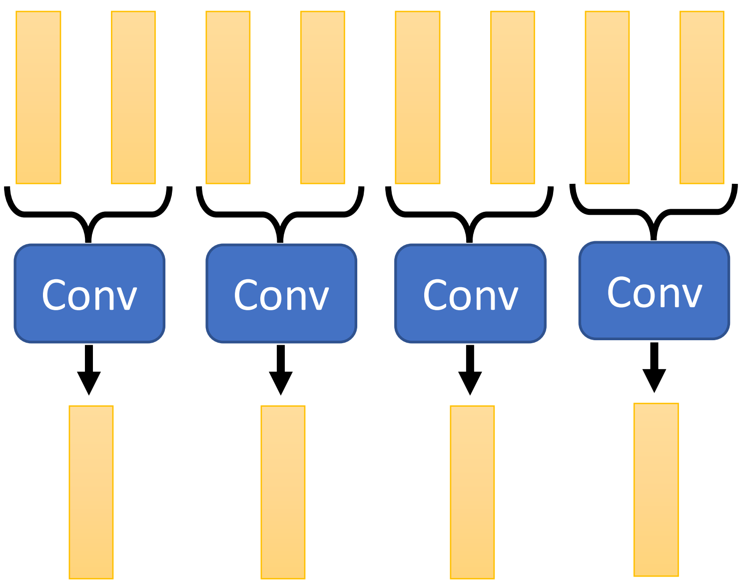


N
value
N
key
K
K
Representative keys
output
query
Can we reduce the number of queries?
change output sequence length

Reduce Number of Keys

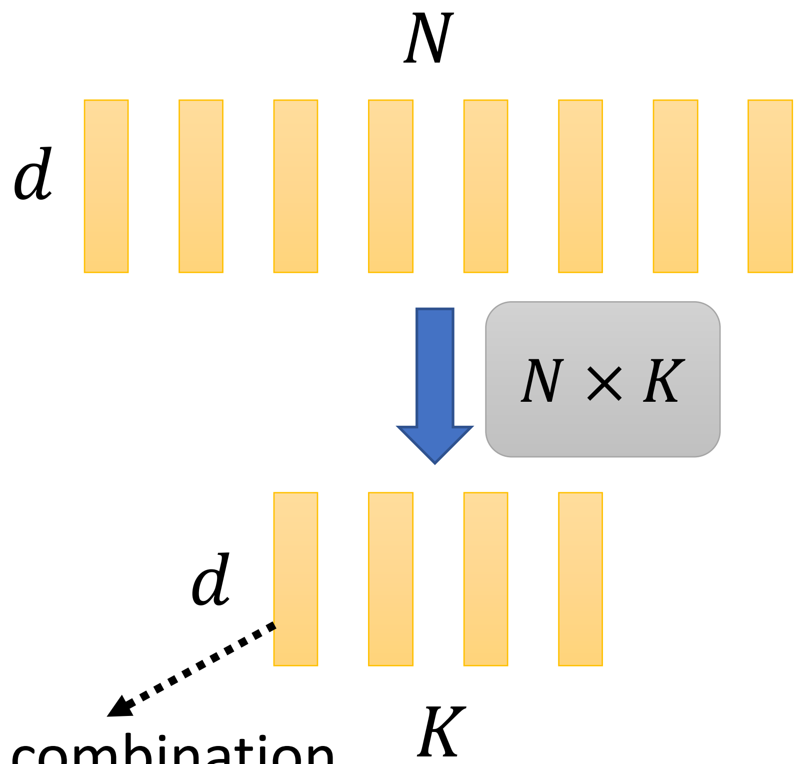
Compressed Attention

https://arxiv.org/abs/1801.10198

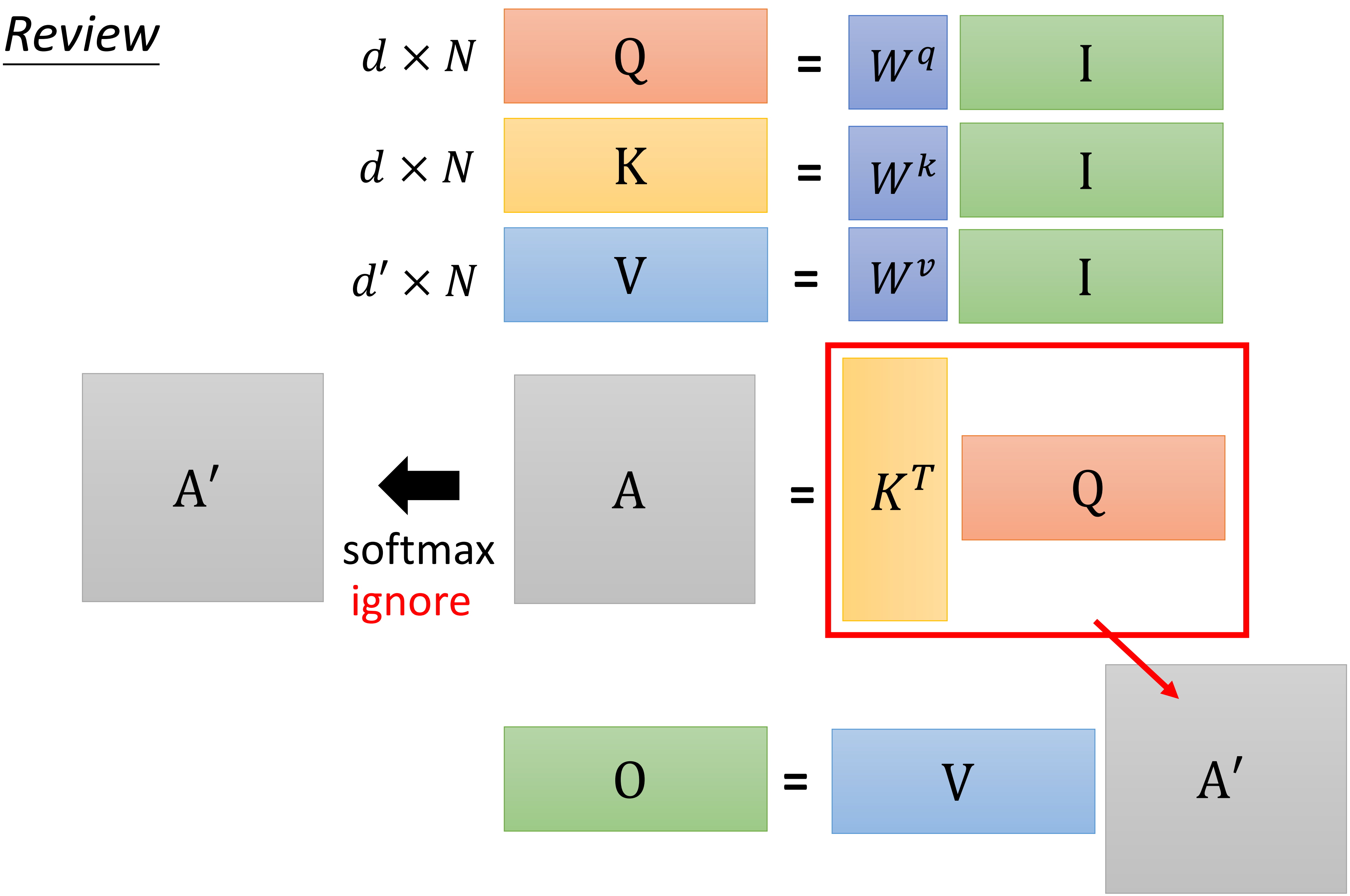


Linformer

https://arxiv.org/abs/2006.04768



Attention Mechanism is three-matrix Multiplication



Attention Mechanism is three-matrix Multiplication

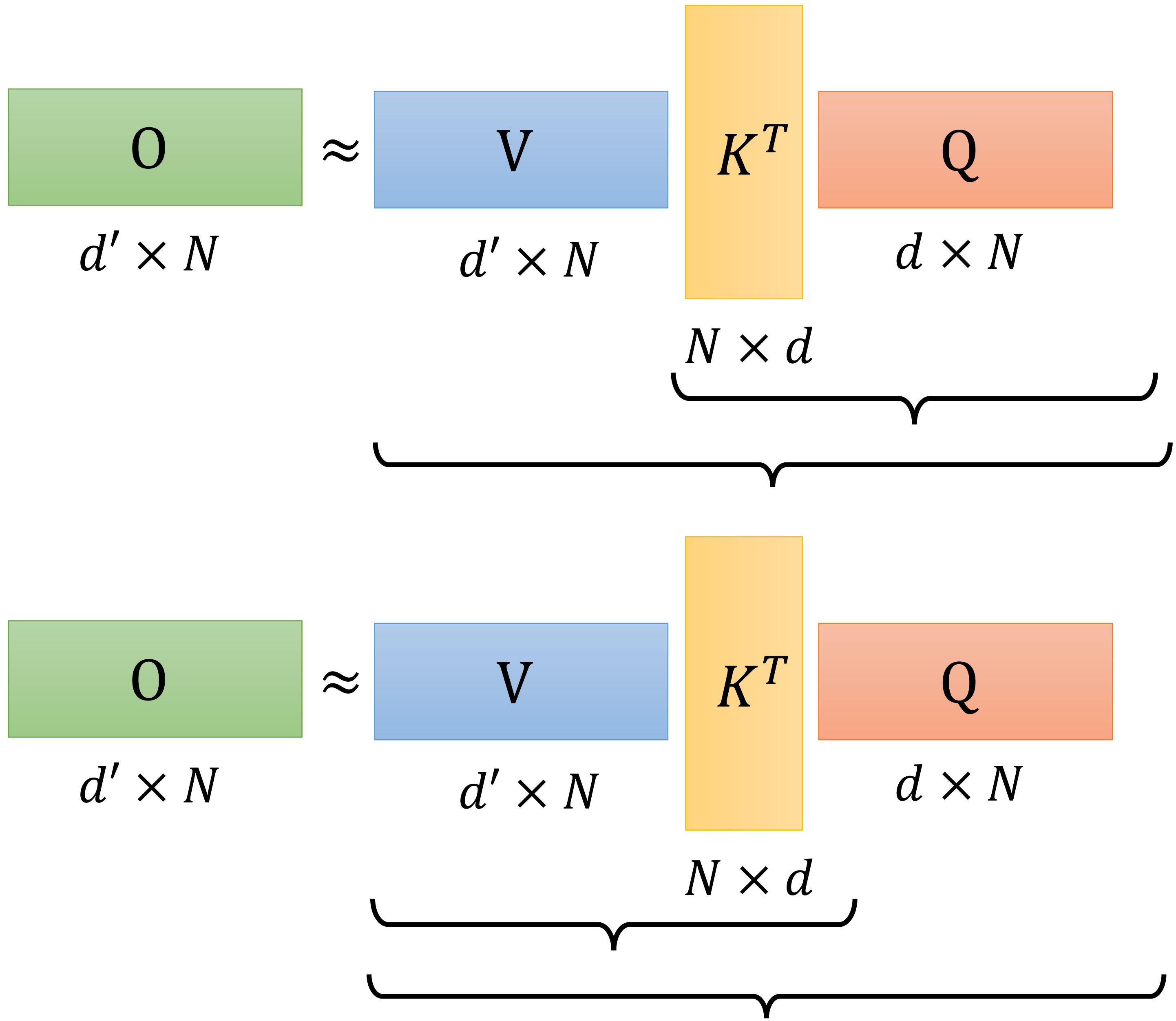
Review

$$\underset{d \times N}{Q} = W^q \, I$$

$$\underset{d \times N}{K} = W^k \, I$$

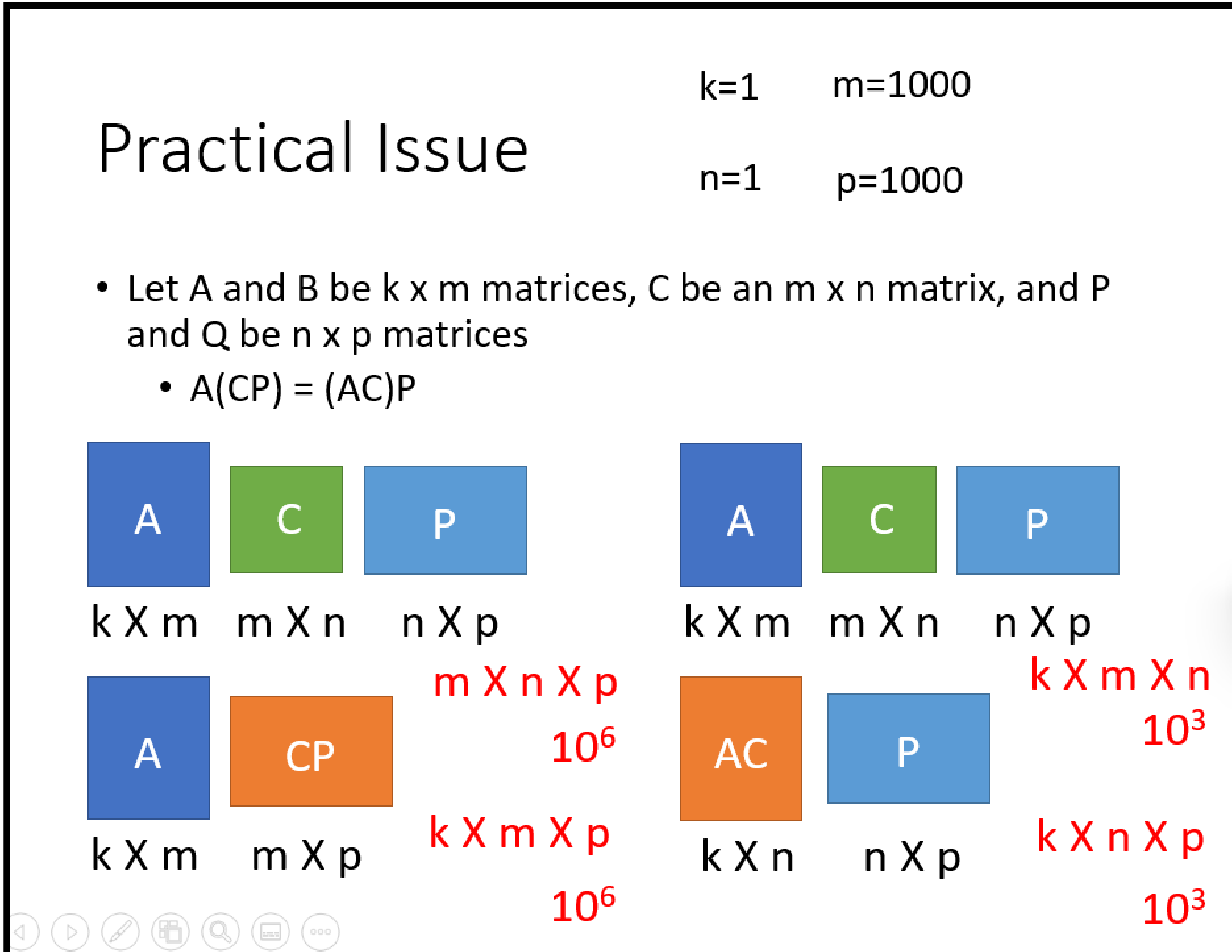
$$\underset{d' \times N}{V} = W^v \, I$$

$$\underset{d' \times N}{O} \approx \underbrace{\underset{d' \times N}{V} \; \underbrace{\underset{N \times d}{K^T} \; \underset{d \times N}{Q}}}$$

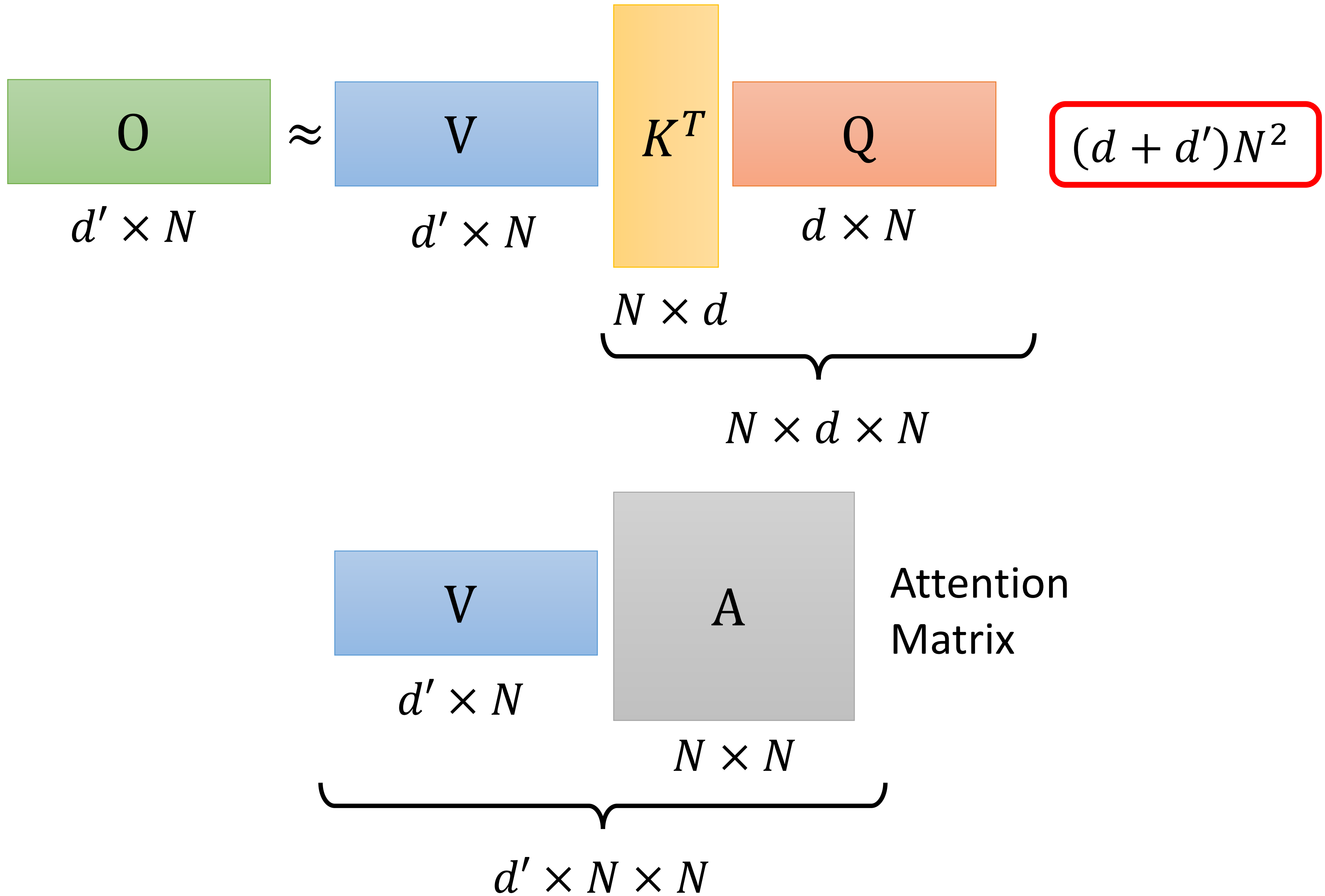


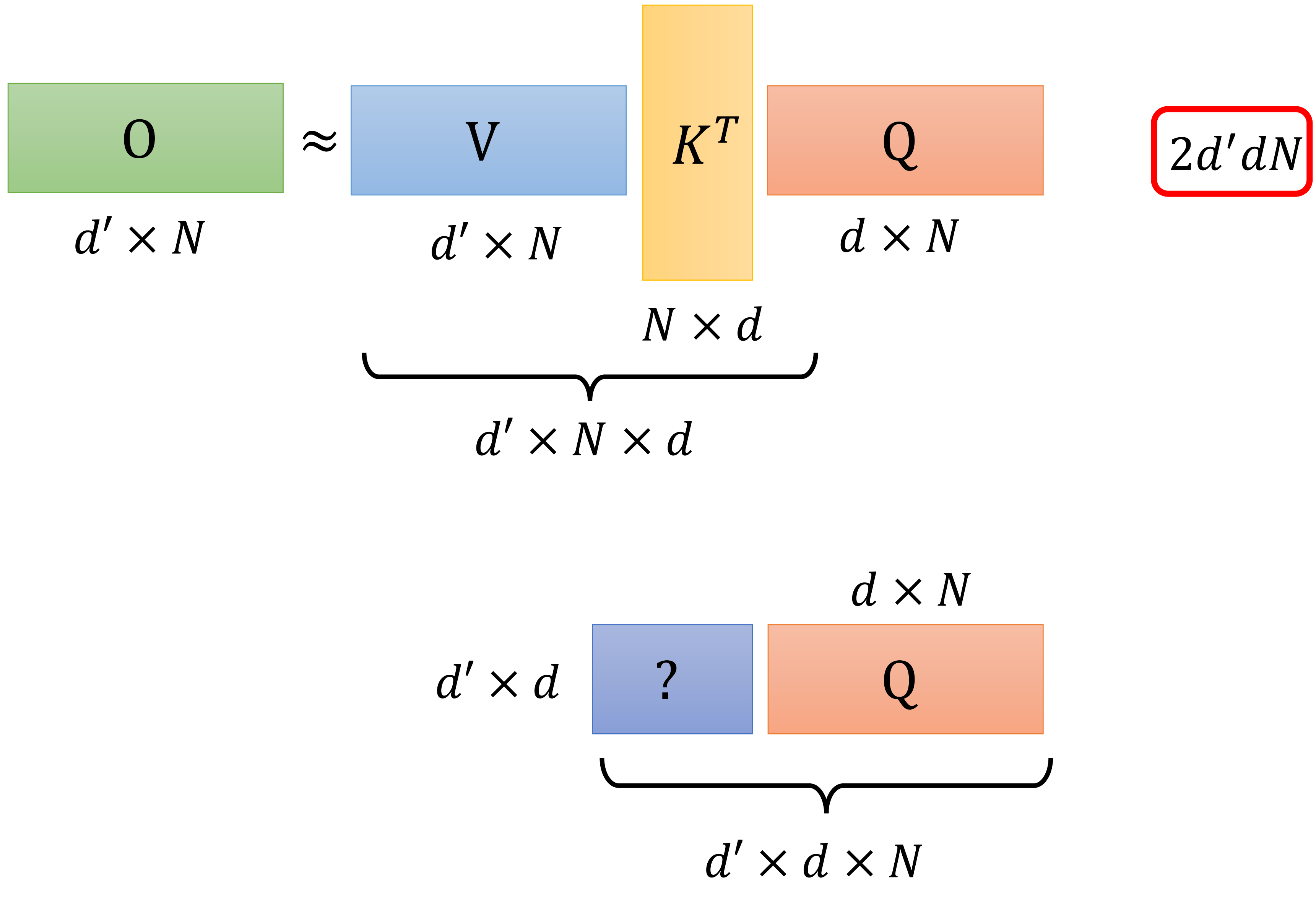
What is the difference?

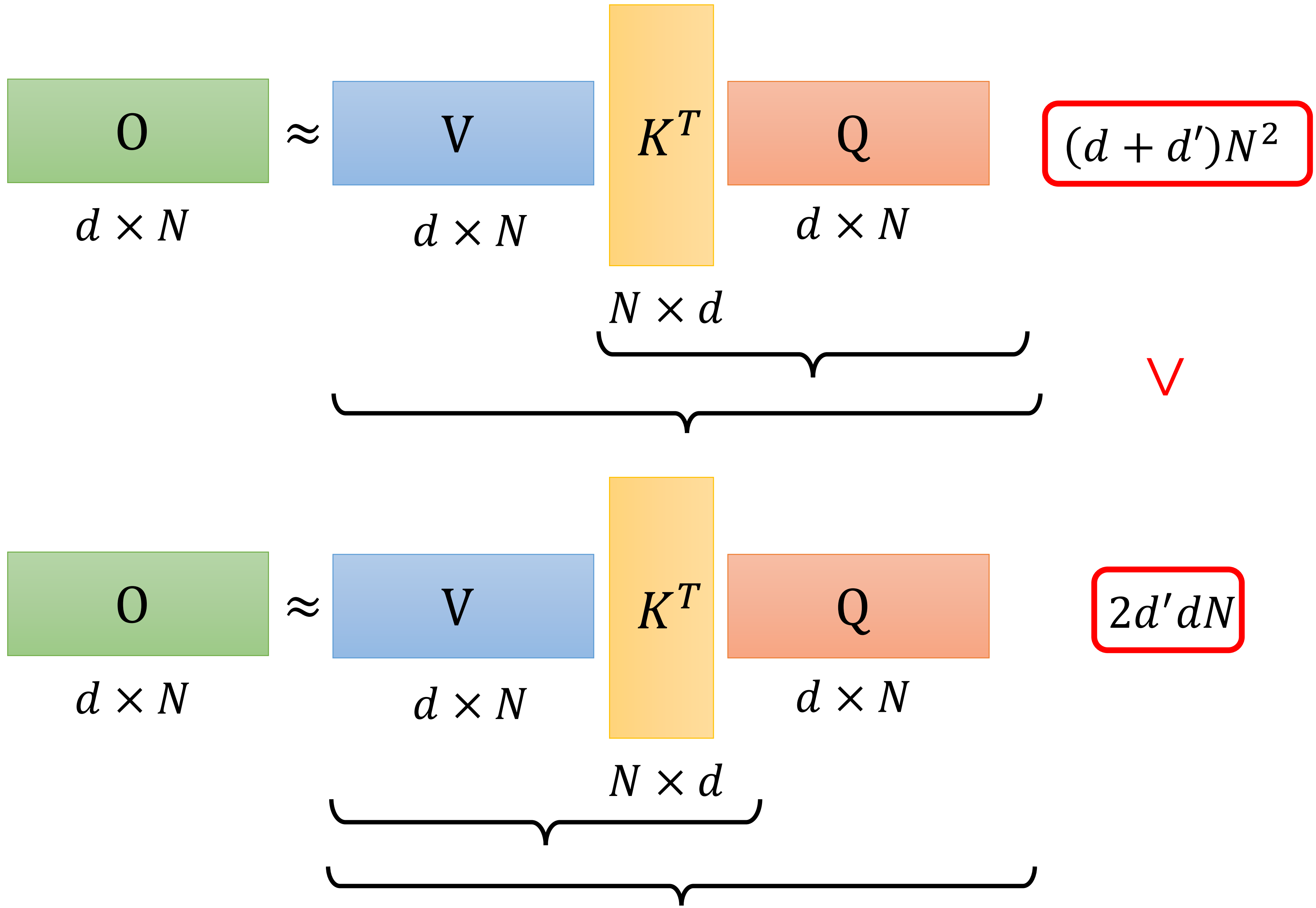
Review Linear Algebra



https://youtu.be/yO8lDzf4jMs


O
$\approx$
V
K^T
Q
$(d + d')N^2$
$d' \times N$
$d' \times N$
$d \times N$
$N \times d$
$N \times d \times N$
V
A
Attention Matrix
$d' \times N$
$N \times N$
$d' \times N \times N$

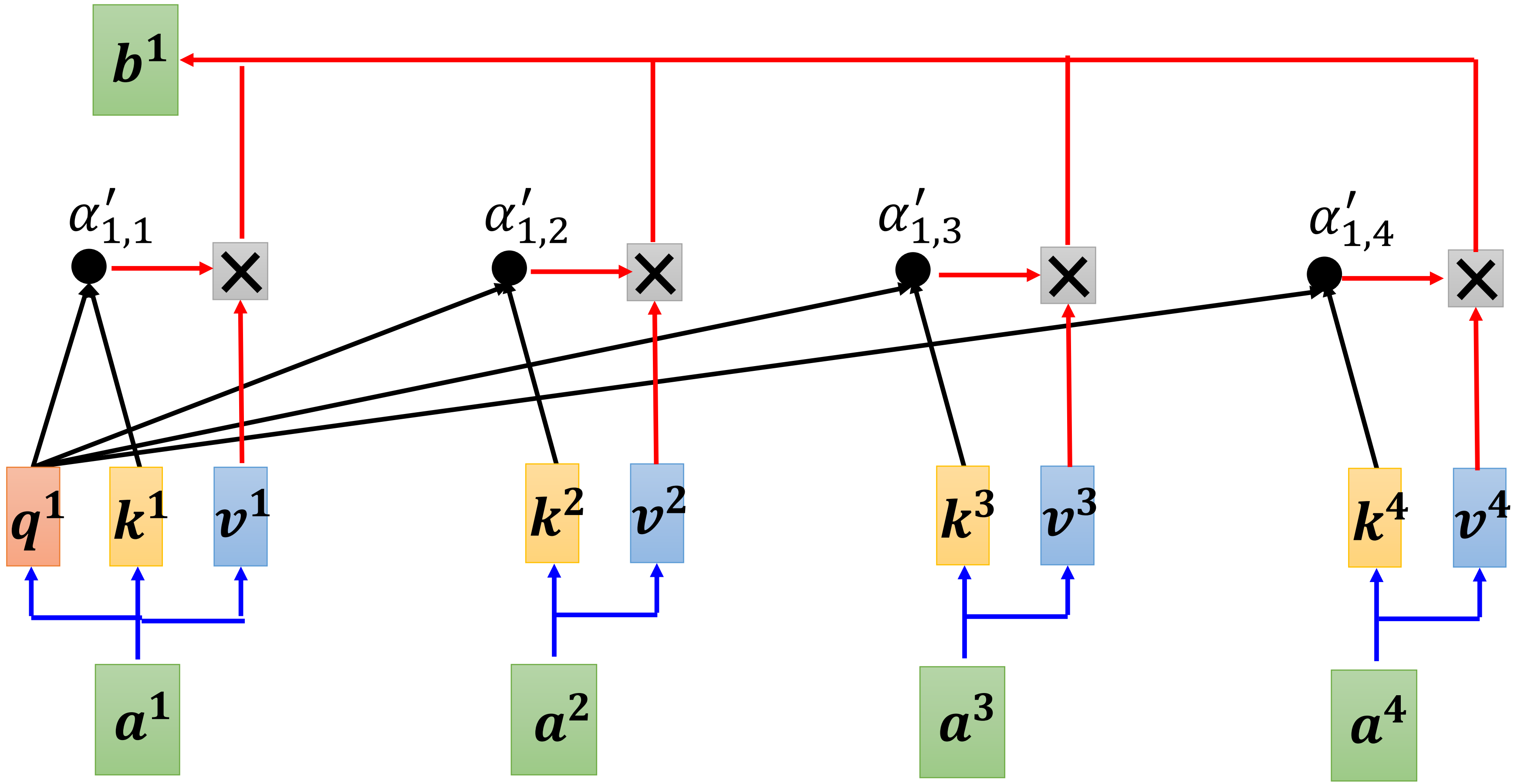

O
≈
V
K^T
Q
$2d'dN$
$d' \times N$
$d' \times N$
$N \times d$
$d \times N$
$d' \times N \times d$
$d \times N$
$d' \times d$
?
Q
$d' \times d \times N$


O ≈ V K^T Q
d × N
d × N
N × d
d × N
(d + d′)N^2
>
O ≈ V K^T Q
d × N
d × N
N × d
d × N
2d′dN

Let’s put softmax back …

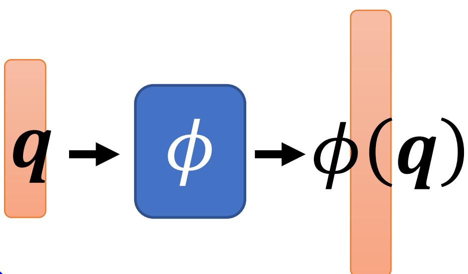
Warning of math

$$\boldsymbol{b}^{1} = \sum_{i=1}^{N} \boxed{\alpha'_{1,i}} \boldsymbol{v}^{i} = \sum_{i=1}^{N} \boxed{\frac{exp(\boldsymbol{q}^{1} \cdot \boldsymbol{k}^{i})}{\sum_{j=1}^{N} exp(\boldsymbol{q}^{1} \cdot \boldsymbol{k}^{j})}} \boldsymbol{v}^{i}$$



$$\boldsymbol{b}^{1} = \sum_{i=1}^{N} \alpha'_{1,i} \boldsymbol{v}^{i} = \sum_{i=1}^{N} \frac{exp(\boldsymbol{q}^{1} \cdot \boldsymbol{k}^{i})}{\sum_{j=1}^{N} exp(\boldsymbol{q}^{1} \cdot \boldsymbol{k}^{j})} \boldsymbol{v}^{i}$$

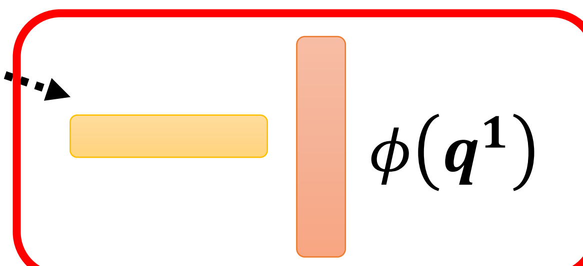
$$exp(\boldsymbol{q} \cdot \boldsymbol{k}) \approx \phi(\boldsymbol{q}) \cdot \phi(\boldsymbol{k})$$



$$= \sum_{i=1}^{N} \frac{\phi(\boldsymbol{q}^{1}) \cdot \phi(\boldsymbol{k}^{i})}{\sum_{j=1}^{N} \phi(\boldsymbol{q}^{1}) \cdot \phi(\boldsymbol{k}^{j})} \boldsymbol{v}^{i}$$

$$= \frac{\sum_{i=1}^{N} [\phi(\boldsymbol{q}^{1}) \cdot \phi(\boldsymbol{k}^{i})] \boldsymbol{v}^{i}}{\boxed{\sum_{j=1}^{N} \phi(\boldsymbol{q}^{1}) \cdot \phi(\boldsymbol{k}^{j})}}$$

$$\phi(\boldsymbol{q}^{1}) \cdot \boxed{\sum_{j=1}^{N} \phi(\boldsymbol{k}^{j})}$$



$$\boldsymbol{b}^{1} = \sum_{i=1}^{N} \alpha'_{1,i} \boldsymbol{v}^{i} = \frac{\boxed{\sum_{i=1}^{N} [\phi(\boldsymbol{q}^{1}) \cdot \phi(\boldsymbol{k}^{i})] \boldsymbol{v}^{i}}}{\phi(\boldsymbol{q}^{1}) \cdot \sum_{j=1}^{N} \phi(\boldsymbol{k}^{j})}$$

$$\boxed{\sum_{i=1}^{N} [\phi(\boldsymbol{q}^{1}) \cdot \phi(\boldsymbol{k}^{i})] \boldsymbol{v}^{i}} \qquad \phi(\boldsymbol{q}^{1}) = \begin{bmatrix} q_1^1 \\ q_2^1 \\ \vdots \end{bmatrix} \qquad \phi(\boldsymbol{k}^{1}) = \begin{bmatrix} k_1^1 \\ k_2^1 \\ \vdots \end{bmatrix}$$

$$= [\phi(\boldsymbol{q}^{1}) \cdot \phi(\boldsymbol{k}^{1})] \boldsymbol{v}^{1} + [\phi(\boldsymbol{q}^{1}) \cdot \phi(\boldsymbol{k}^{2})] \boldsymbol{v}^{2} + \cdots$$

$$= (q_1^1 k_1^1 + q_2^1 k_2^1 + \cdots) \boldsymbol{v}^{1} + (q_1^1 k_1^2 + q_2^1 k_2^2 + \cdots) \boldsymbol{v}^{2} + \cdots$$

$$= \underline{q_1^1 k_1^1 \boldsymbol{v}^{1}} + \underline{q_2^1 k_2^1 \boldsymbol{v}^{1}} + \cdots + \underline{q_1^1 k_1^2 \boldsymbol{v}^{2}} + \underline{q_2^1 k_2^2 \boldsymbol{v}^{2}} + \cdots + \cdots$$

$$= q_1^1 (k_1^1 \boldsymbol{v}^{1} + k_1^2 \boldsymbol{v}^{2} + \cdots) + q_2^1 (k_2^1 \boldsymbol{v}^{1} + k_2^2 \boldsymbol{v}^{2} + \cdots)$$

$$\boldsymbol{b}^{1} = \sum_{i=1}^{N} \alpha'_{1,i} \boldsymbol{v}^{i} = \frac{\boxed{\sum_{i=1}^{N} \left[\phi(\boldsymbol{q}^{1}) \cdot \phi(\boldsymbol{k}^{i})\right] \boldsymbol{v}^{i}}}{\phi(\boldsymbol{q}^{1}) \cdot \sum_{j=1}^{N} \phi(\boldsymbol{k}^{j})}$$

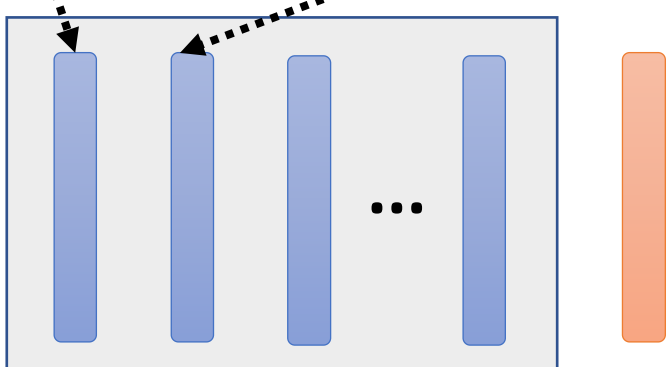
$$\boxed{\sum_{i=1}^{N} \left[\phi(\boldsymbol{q}^{1}) \cdot \phi(\boldsymbol{k}^{i})\right] \boldsymbol{v}^{i}}$$

$$\phi(\boldsymbol{q}^{1}) = \begin{bmatrix} q_1^1 \\ q_2^1 \\ \vdots \end{bmatrix} \qquad \phi(\boldsymbol{k}^{1}) = \begin{bmatrix} k_1^1 \\ k_2^1 \\ \vdots \end{bmatrix}$$

M dim

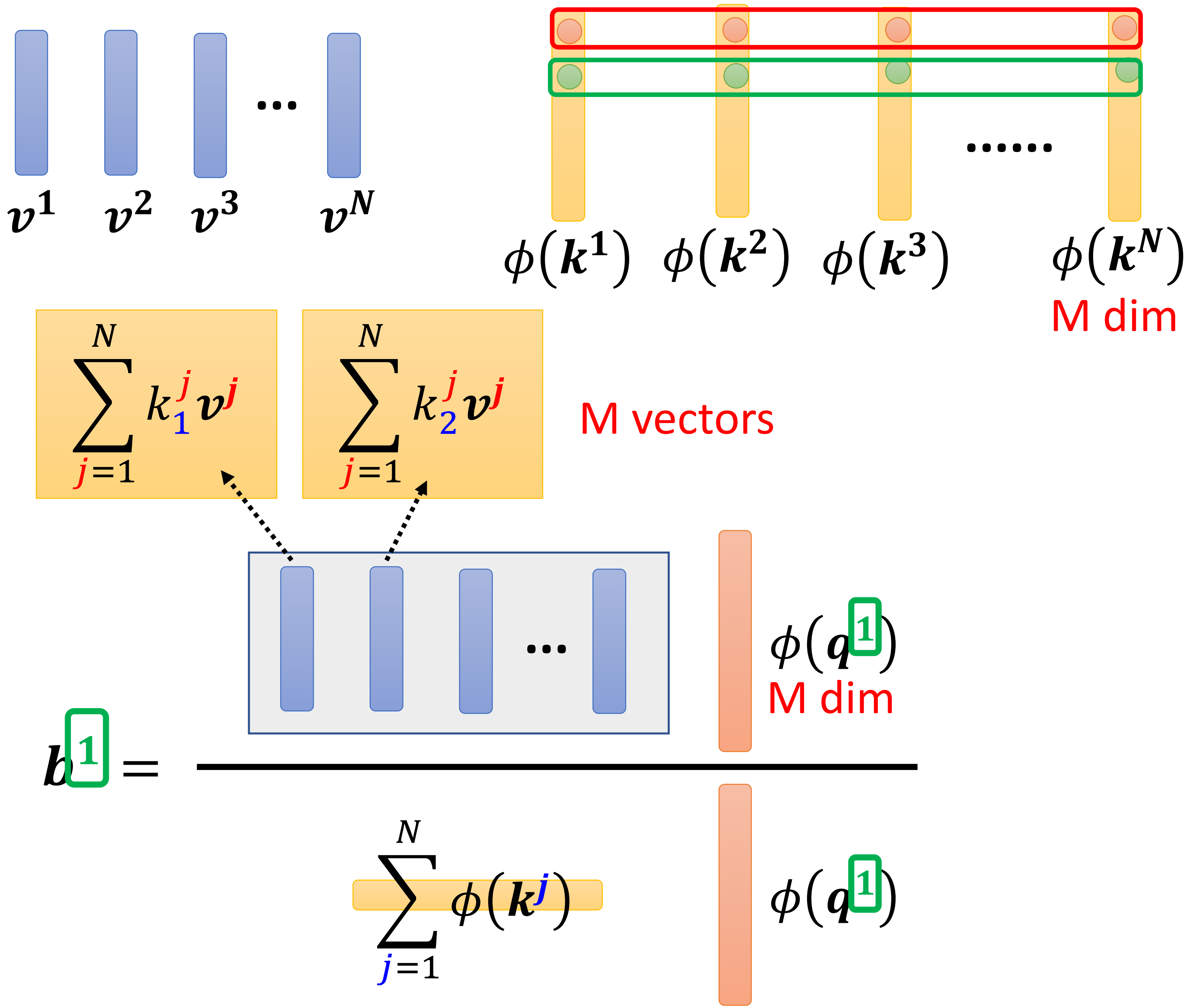
$$= \boxed{q_1^1}\left(k_1^1 \boldsymbol{v}^{1} + k_1^2 \boldsymbol{v}^{2} + \cdots\right) + \boxed{q_2^1}\left(k_2^1 \boldsymbol{v}^{1} + k_2^2 \boldsymbol{v}^{2} + \cdots\right)$$

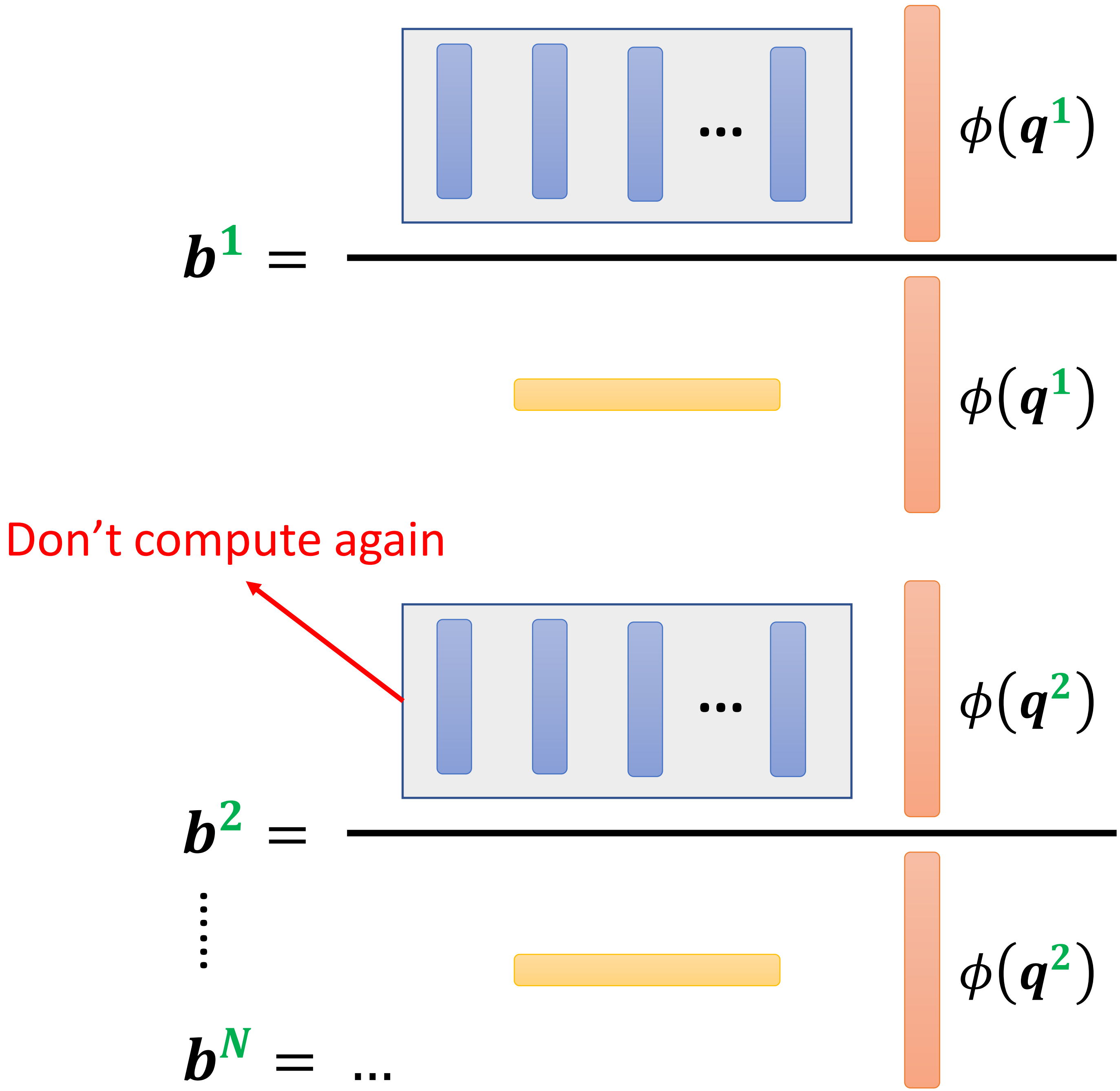
$$\sum_{j=1}^{N} k_1^j \boldsymbol{v}^{j} \qquad \sum_{j=1}^{N} k_2^j \boldsymbol{v}^{j}$$



$\phi(\boldsymbol{q}^{1})$

M vectors

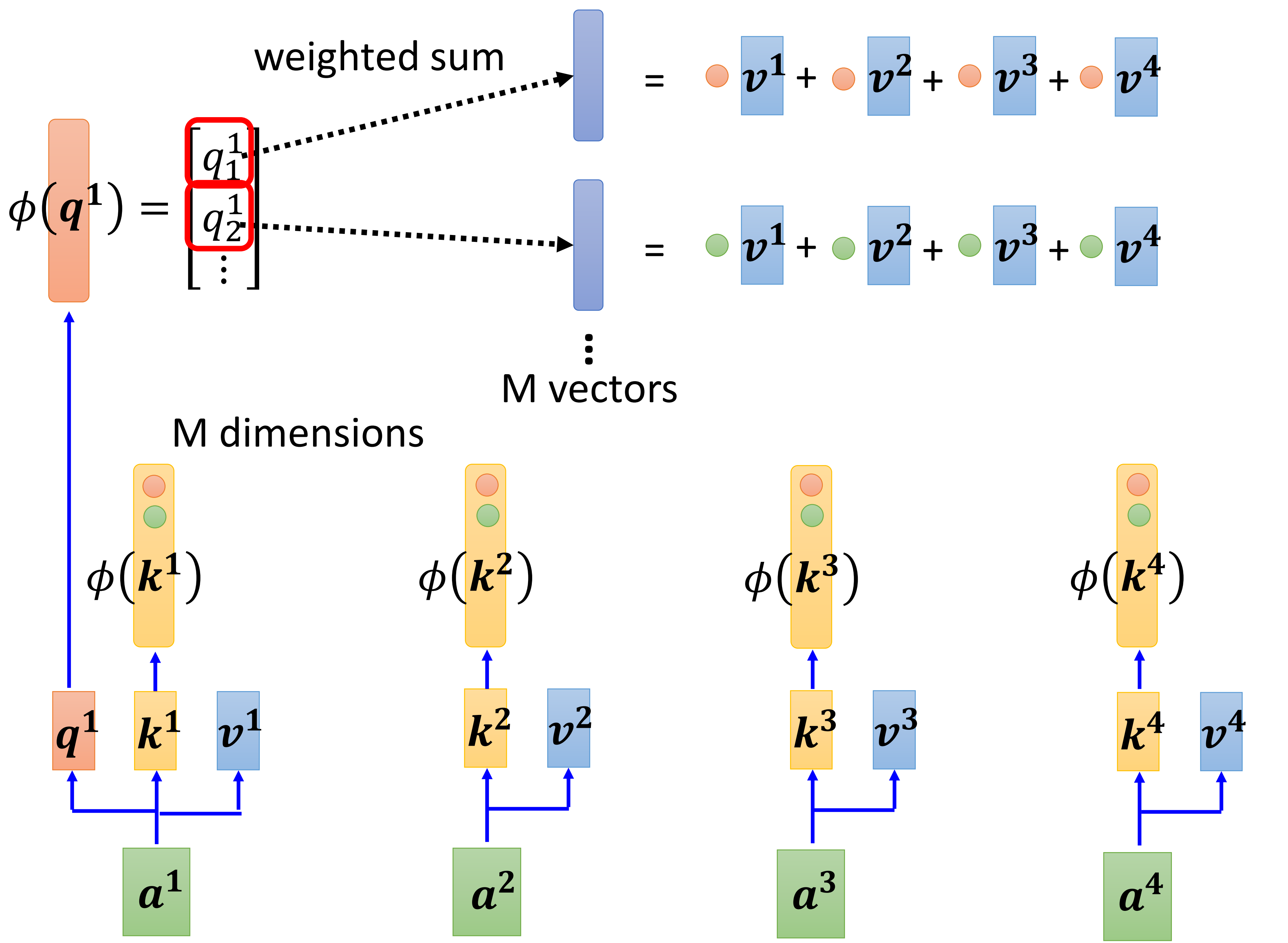

v^1 v^2 v^3 ... v^N
$\phi(k^1)$ $\phi(k^2)$ $\phi(k^3)$ ...... $\phi(k^N)$
M dim
$\sum_{j=1}^{N} k_1^j v^j$
$\sum_{j=1}^{N} k_2^j v^j$
M vectors
$\phi(q^1)$
M dim
$b^1 =$
$\sum_{j=1}^{N} \phi(k^j)$
$\phi(q^1)$

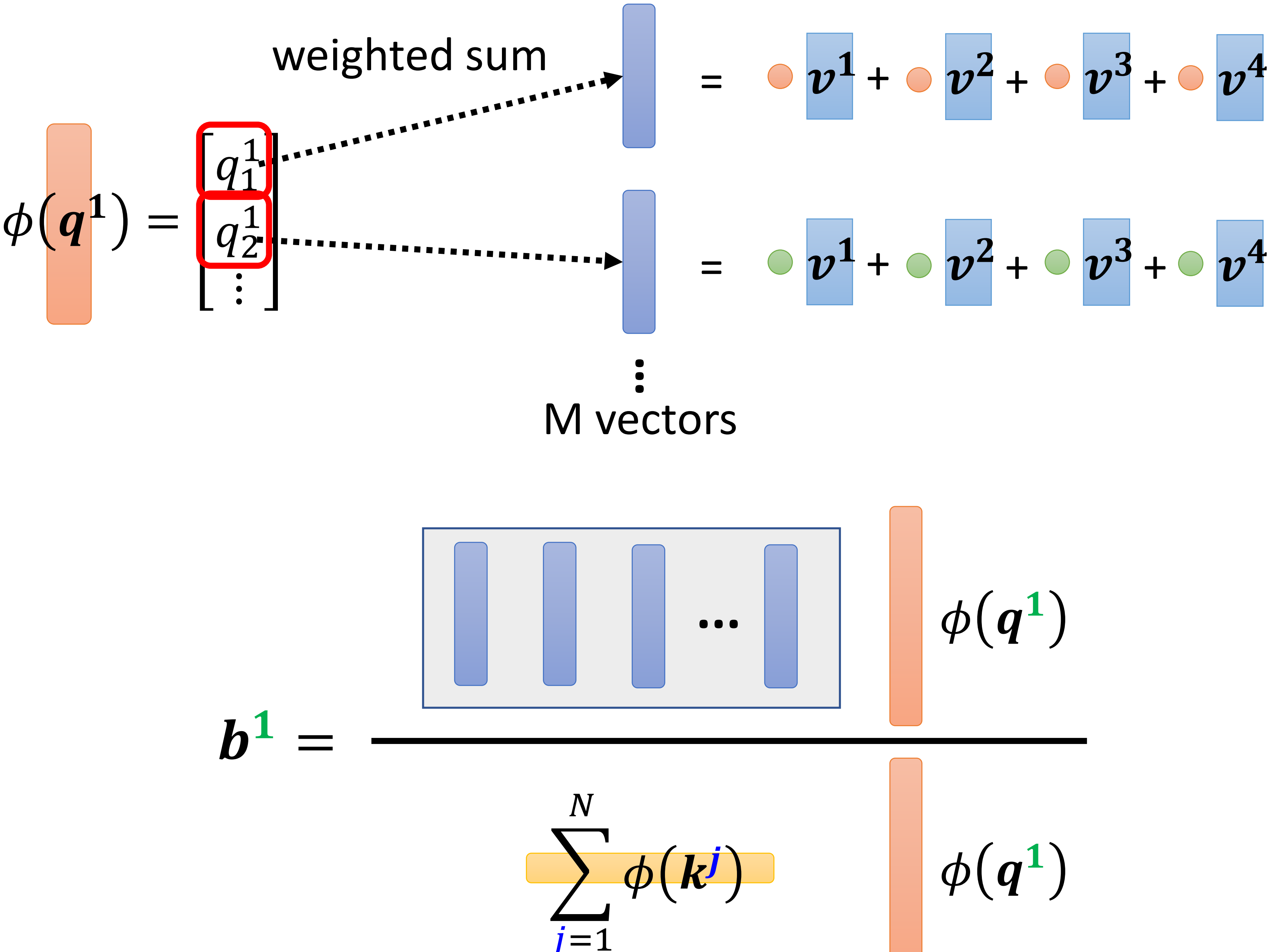


Don’t compute again
$\boldsymbol{b}^1 =$
$\phi(\boldsymbol{q}^1)$
$\phi(\boldsymbol{q}^1)$
$\boldsymbol{b}^2 =$
$\phi(\boldsymbol{q}^2)$
$\phi(\boldsymbol{q}^2)$
$\boldsymbol{b}^N = \ldots$

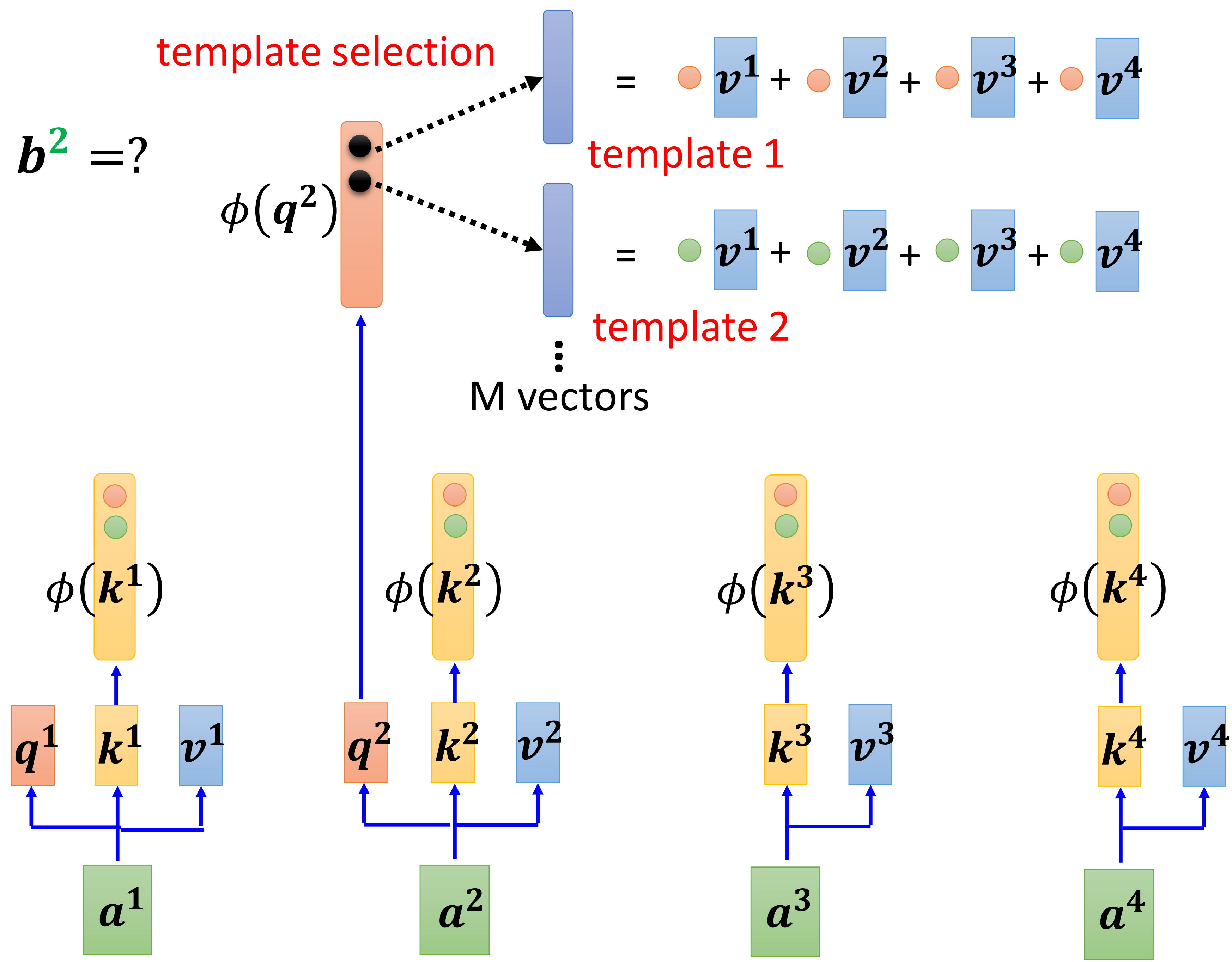
Let’s put softmax back …

End of warning


weighted sum
M vectors
M dimensions
$\phi(\boldsymbol{q}^1) = \begin{bmatrix} q_1^1 \\ q_2^1 \\ \vdots \end{bmatrix}$
$= \boldsymbol{v}^1 + \boldsymbol{v}^2 + \boldsymbol{v}^3 + \boldsymbol{v}^4$
$= \boldsymbol{v}^1 + \boldsymbol{v}^2 + \boldsymbol{v}^3 + \boldsymbol{v}^4$
$\phi(\boldsymbol{k}^1)$
$\phi(\boldsymbol{k}^2)$
$\phi(\boldsymbol{k}^3)$
$\phi(\boldsymbol{k}^4)$
$\boldsymbol{q}^1$
$\boldsymbol{k}^1$
$\boldsymbol{v}^1$
$\boldsymbol{k}^2$
$\boldsymbol{v}^2$
$\boldsymbol{k}^3$
$\boldsymbol{v}^3$
$\boldsymbol{k}^4$
$\boldsymbol{v}^4$
$\boldsymbol{a}^1$
$\boldsymbol{a}^2$
$\boldsymbol{a}^3$
$\boldsymbol{a}^4$



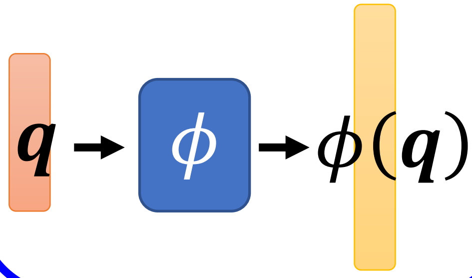
weighted sum
$\phi(\boldsymbol{q}^1) =$
q_1^1
q_2^1
$= \boldsymbol{v}^1 + \boldsymbol{v}^2 + \boldsymbol{v}^3 + \boldsymbol{v}^4$
$= \boldsymbol{v}^1 + \boldsymbol{v}^2 + \boldsymbol{v}^3 + \boldsymbol{v}^4$
M vectors
…
$\boldsymbol{b}^1 =$
$\phi(\boldsymbol{q}^1)$
$\sum_{j=1}^{N} \phi(\boldsymbol{k}^j)$
$\phi(\boldsymbol{q}^1)$



template selection
$b^2 = ?$
$\phi(q^2)$
$= v^1 + v^2 + v^3 + v^4$
template 1
$= v^1 + v^2 + v^3 + v^4$
template 2
M vectors
$\phi(k^1)$
$\phi(k^2)$
$\phi(k^3)$
$\phi(k^4)$
q^1 k^1 v^1
q^2 k^2 v^2
k^3 v^3
k^4 v^4
a^1
a^2
a^3
a^4

Realization

$$exp(\boldsymbol{q} \cdot \boldsymbol{k}) \approx \phi(\boldsymbol{q}) \cdot \phi(\boldsymbol{k})$$



- Efficient attention

  https://arxiv.org/pdf/1812.01243.pdf

- Linear Transformer

  https://linear-transformers.com/

- Random Feature Attention

  https://arxiv.org/pdf/2103.02143.pdf

- Performer

  https://arxiv.org/pdf/2009.14794.pdf

https://arxiv.org/abs/2005.00743

Do we need q and k to compute attention? Synthesizer!

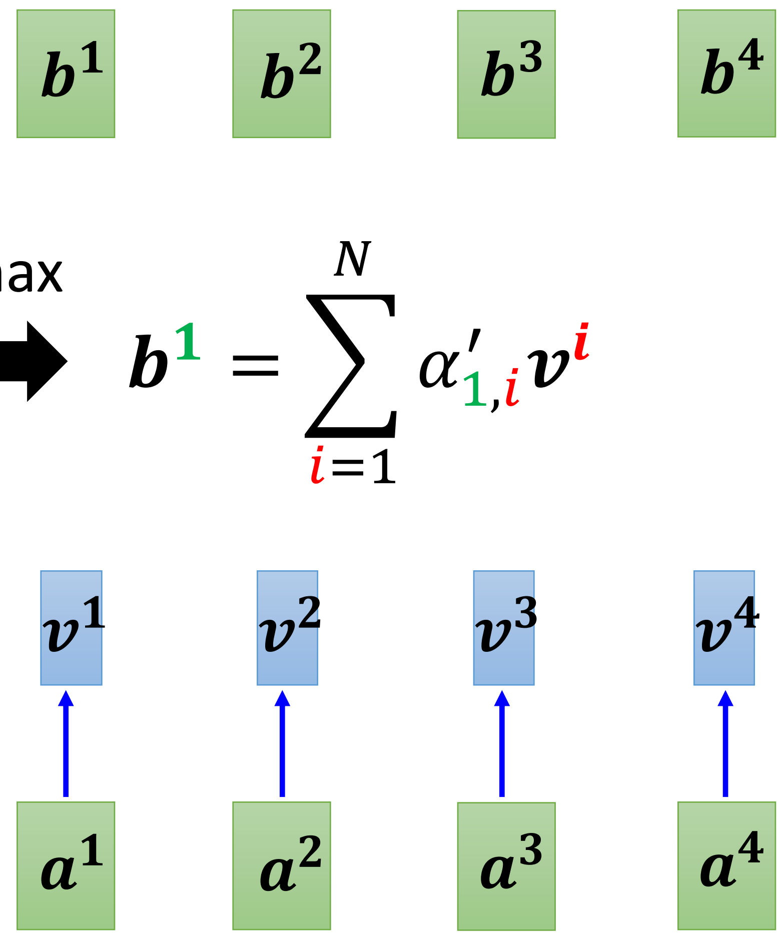
$\alpha_{1,1}$	$\alpha_{1,2}$	$\alpha_{1,3}$	$\alpha_{1,4}$
$\alpha_{1,2}$	$\alpha_{2,2}$	$\alpha_{2,3}$	$\alpha_{2,4}$
$\alpha_{1,3}$	$\alpha_{2,3}$	$\alpha_{3,3}$	$\alpha_{3,4}$
$\alpha_{1,4}$	$\alpha_{2,4}$	$\alpha_{3,4}$	$\alpha_{4,4}$

~~From q and k?~~

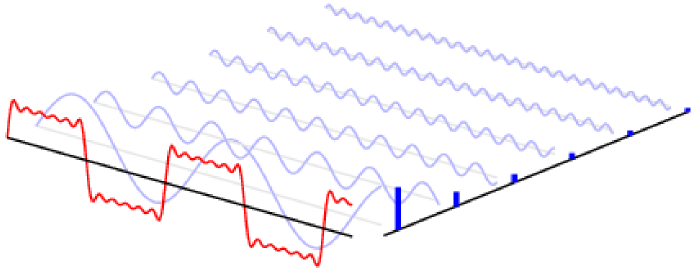
They are network parameters!

softmax

$$\boldsymbol{b}^{1} = \sum_{i=1}^{N} \alpha'_{1,i} \boldsymbol{v}^{i}$$



Attention-free?



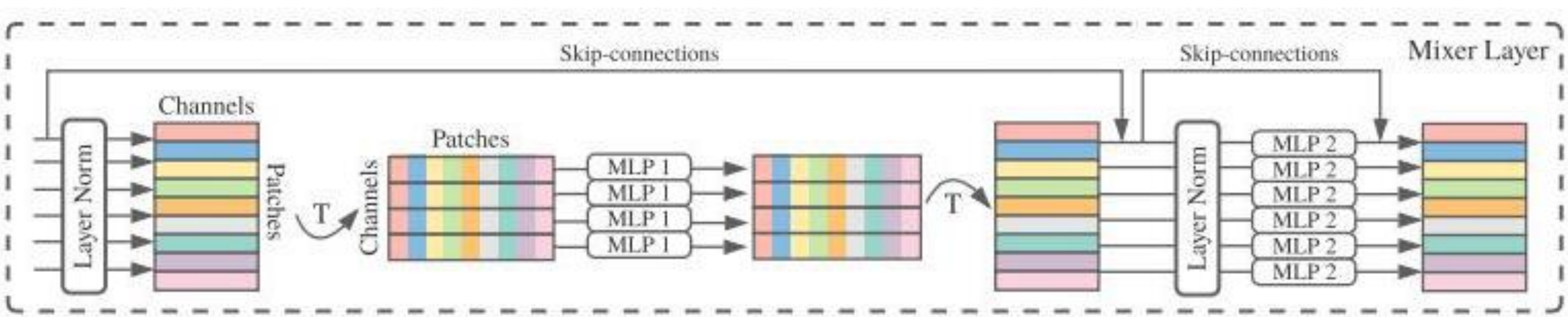
- Fnet: Mixing tokens with fourier transforms

https://arxiv.org/abs/2105.03824

- Pay Attention to MLPs https://arxiv.org/abs/2105.08050

- MLP-Mixer: An all-MLP Architecture for Vision

https://arxiv.org/abs/2105.01601



Summary

- Human knowledge
  - Local Attention, Big Bird
- Clustering
  - Reformer
- Learnable Pattern
  - Sinkforn
- Representative key
  - Linformer
- k,q first → v,k first
  - Linear Transformer, Performer
- New framework
  - Synthesizer