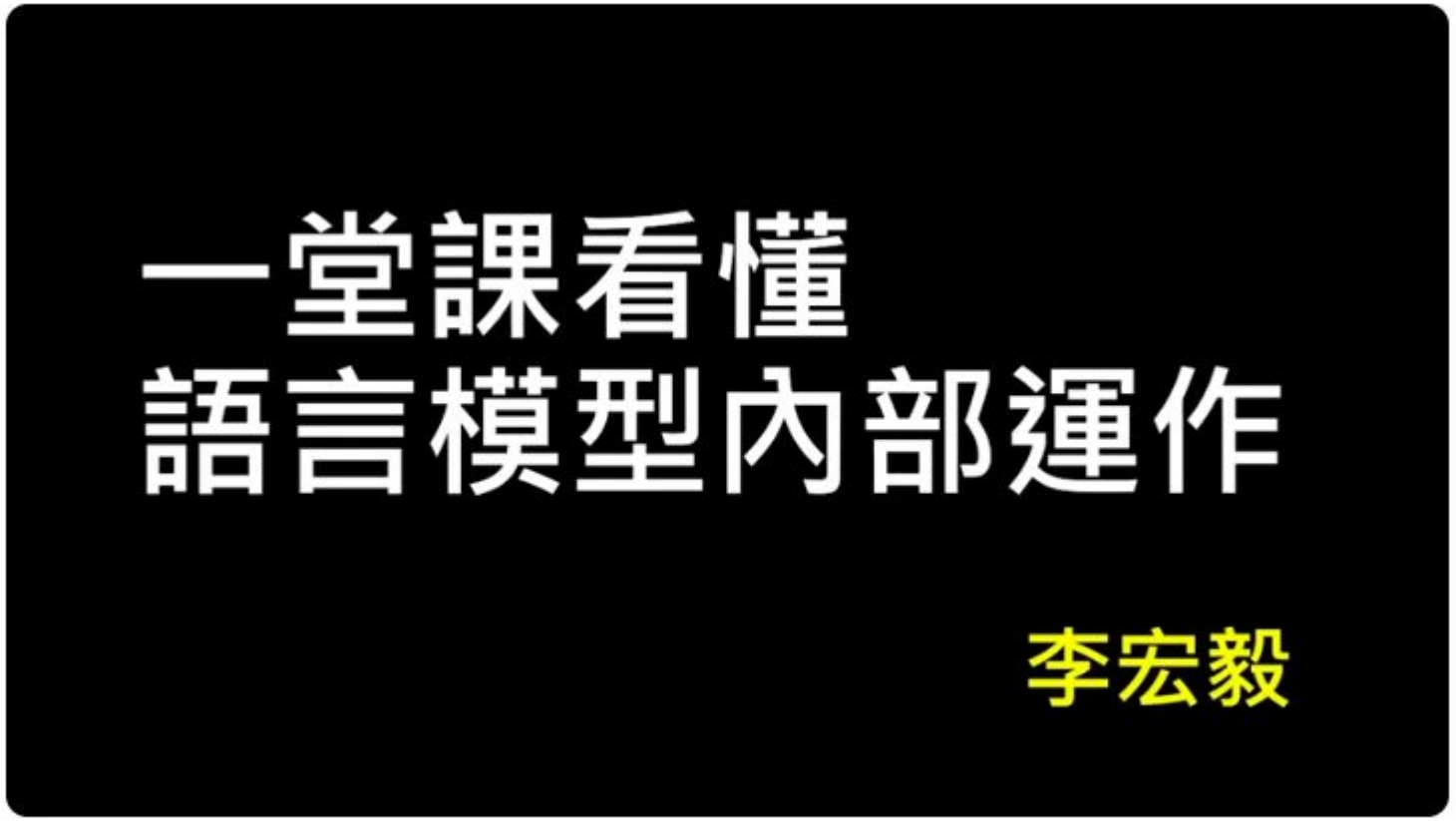
加快語言模型的生成速度

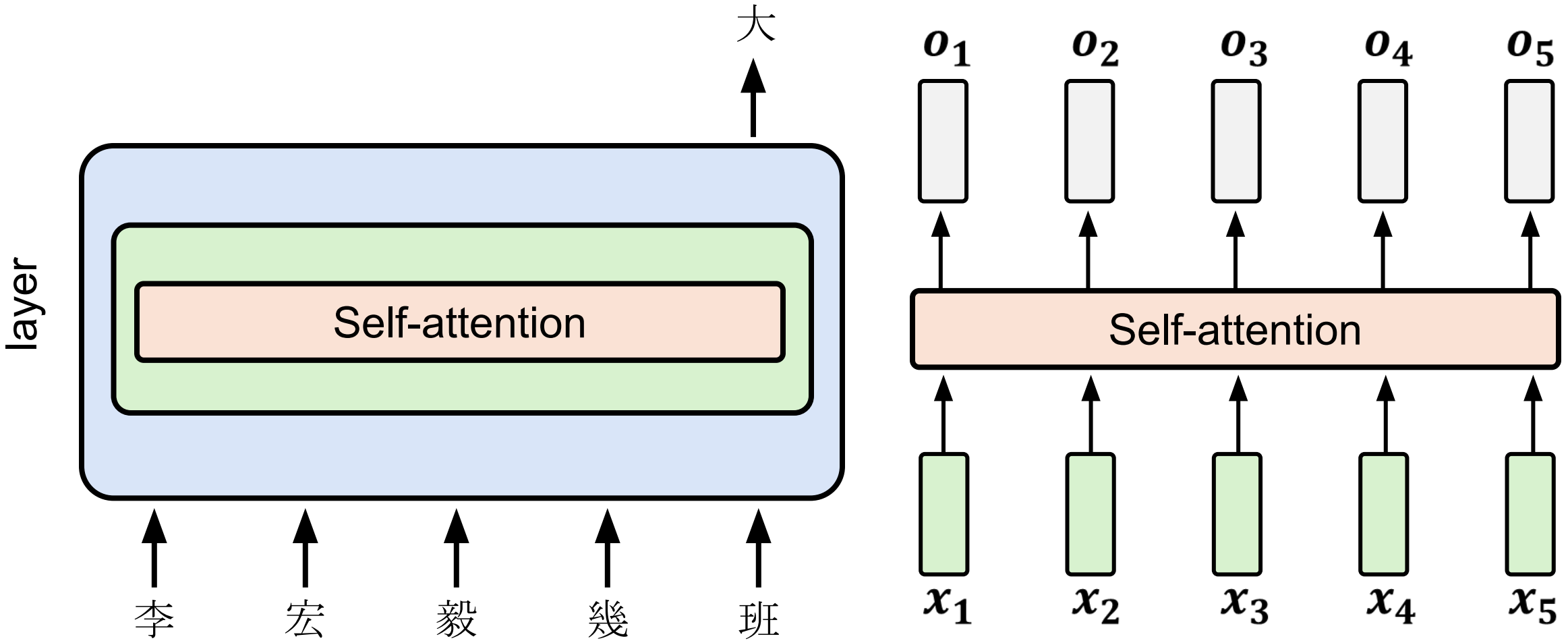
先備知識

https://youtu.be/8iFvM7WUUs8?si=gymWr9Vurpb8ri2Z

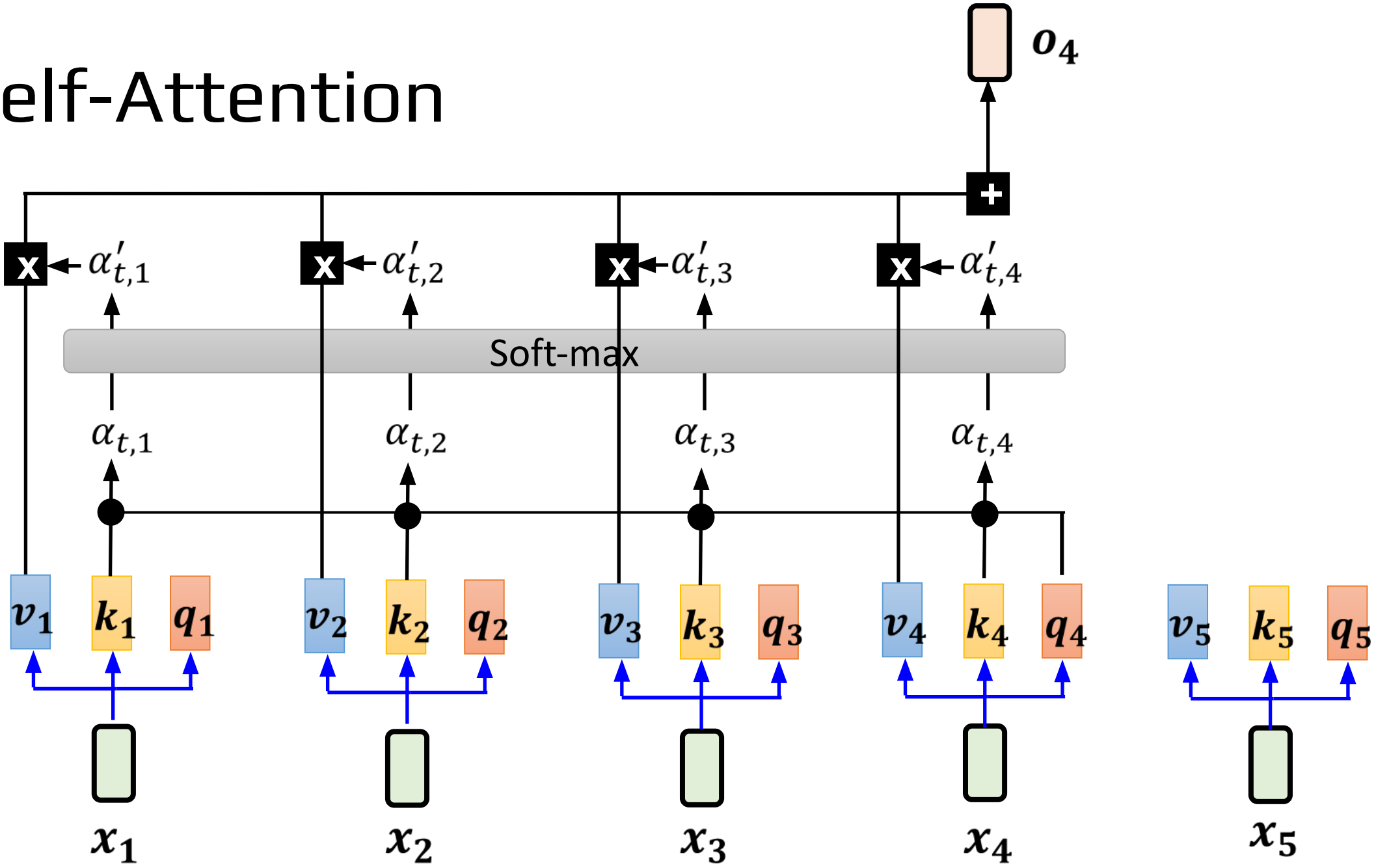


【生成式人工智慧與機器學習導論2025】第3講：解剖大型語言模型

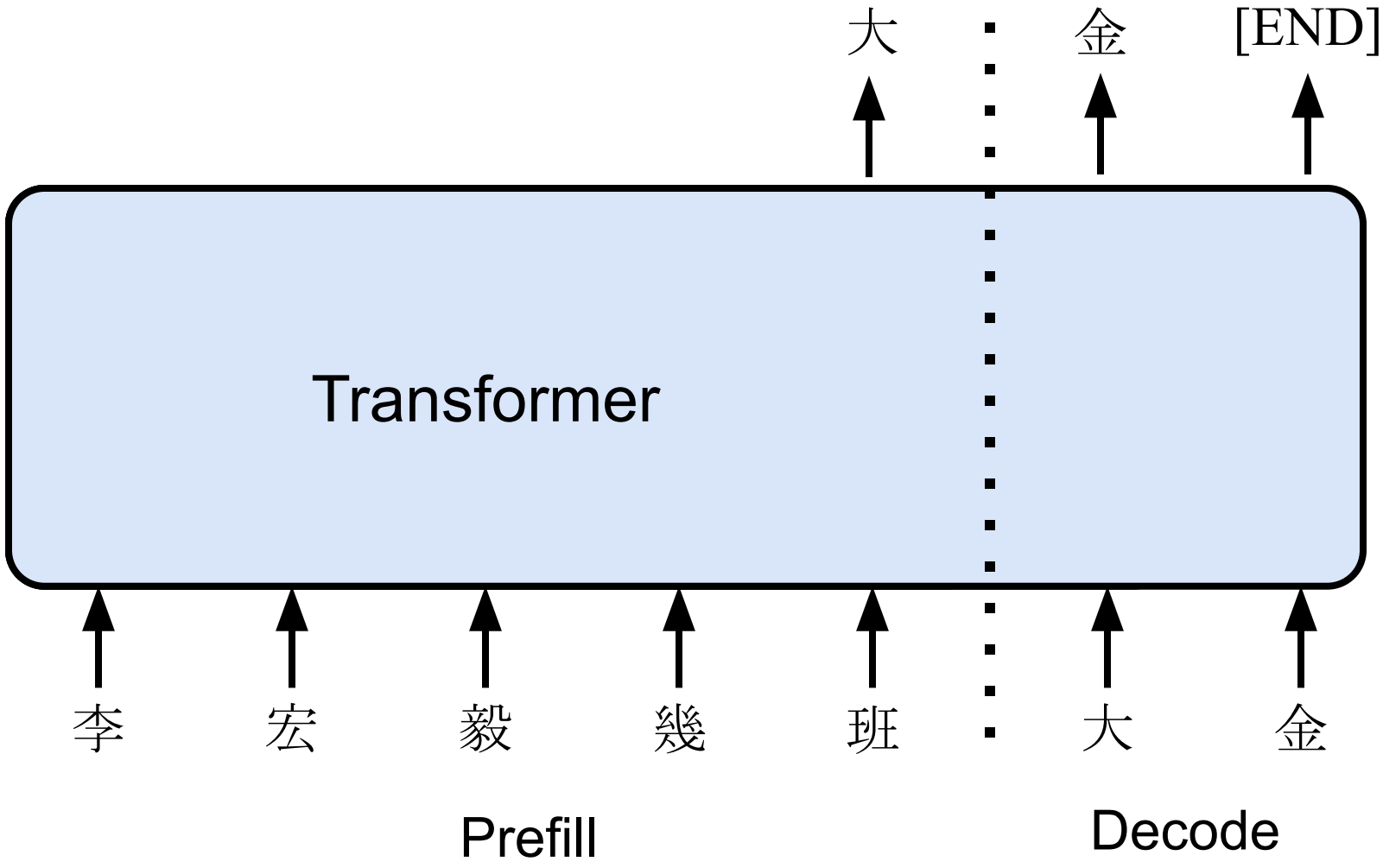
語言模型如何生成 (推論, Inference)



Self-Attention



語言模型如何生成 (推論, Inference)



加快推論速度的經典方法

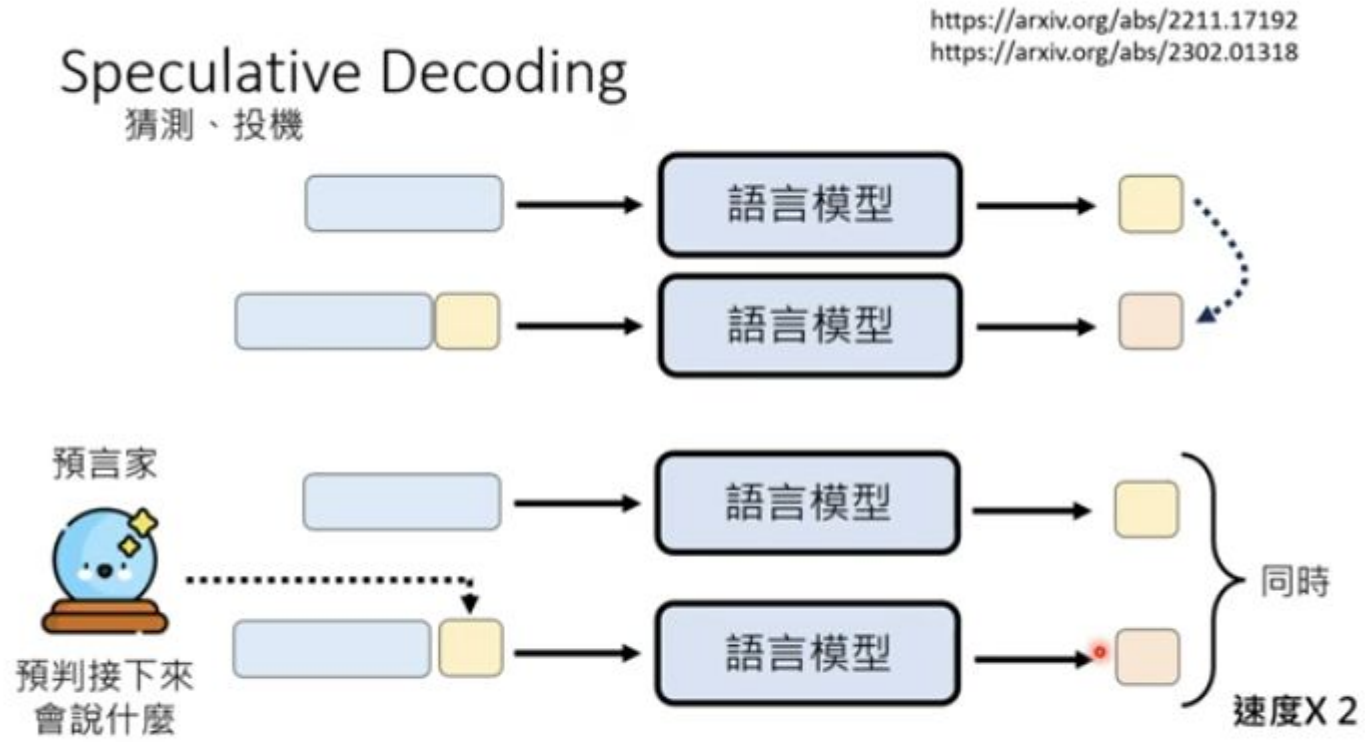
Today

Flash Attention

KV Cache

Speculative Decoding

Speculative Decoding



【生成式AI導論 2024】第16講：可以加速所有語言模型生成速度的神奇外掛 — Speculative Decoding

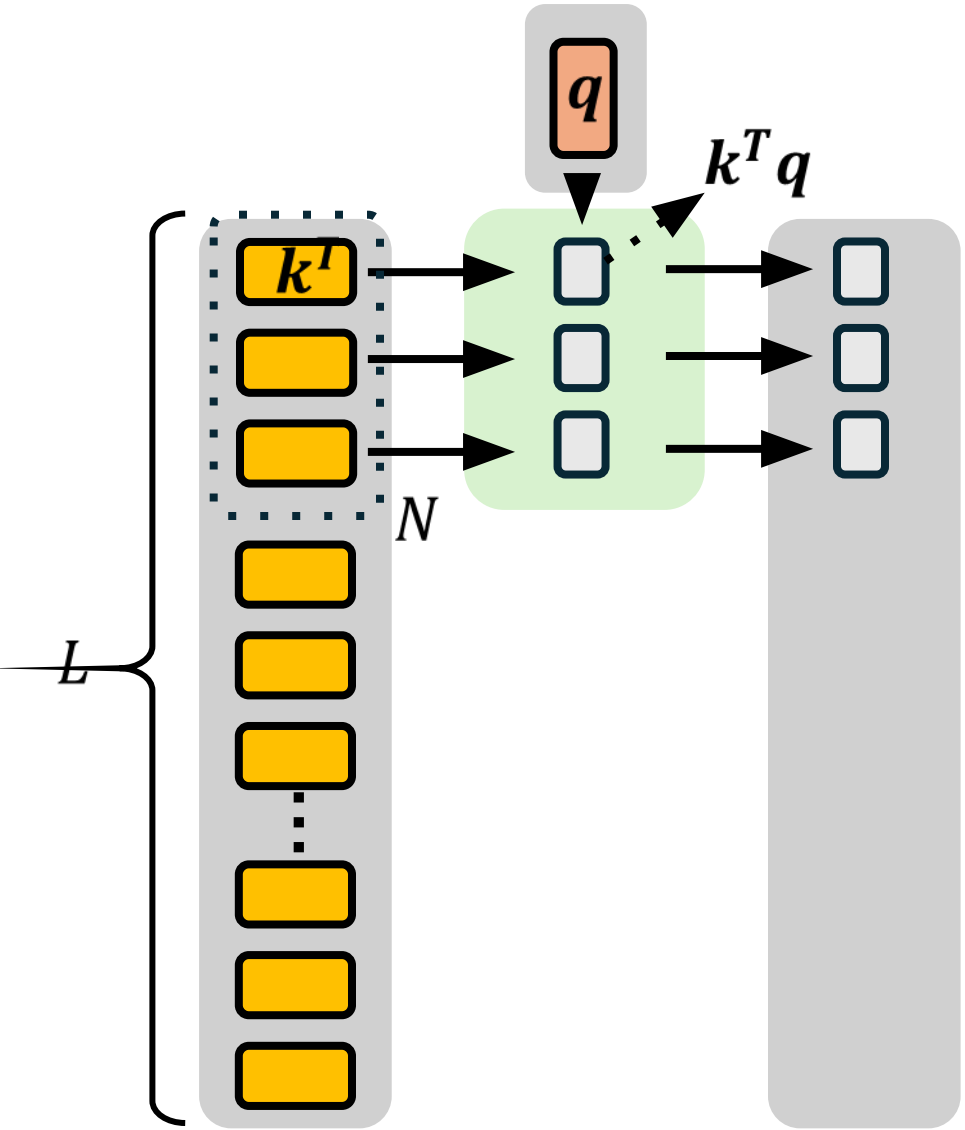
https://youtu.be/MAbGgsWKrg8?si=88lZbhk0GVJtN0Yi

Flash Attention

https://arxiv.org/abs/2205.14135

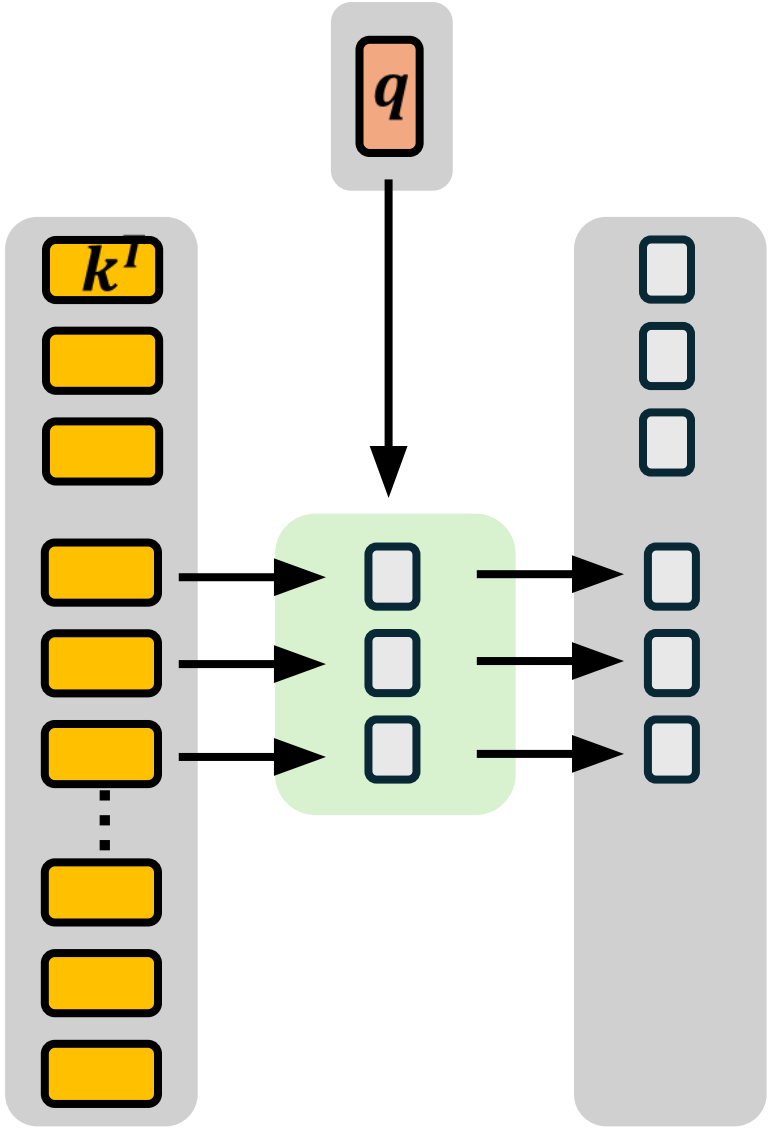
HBM

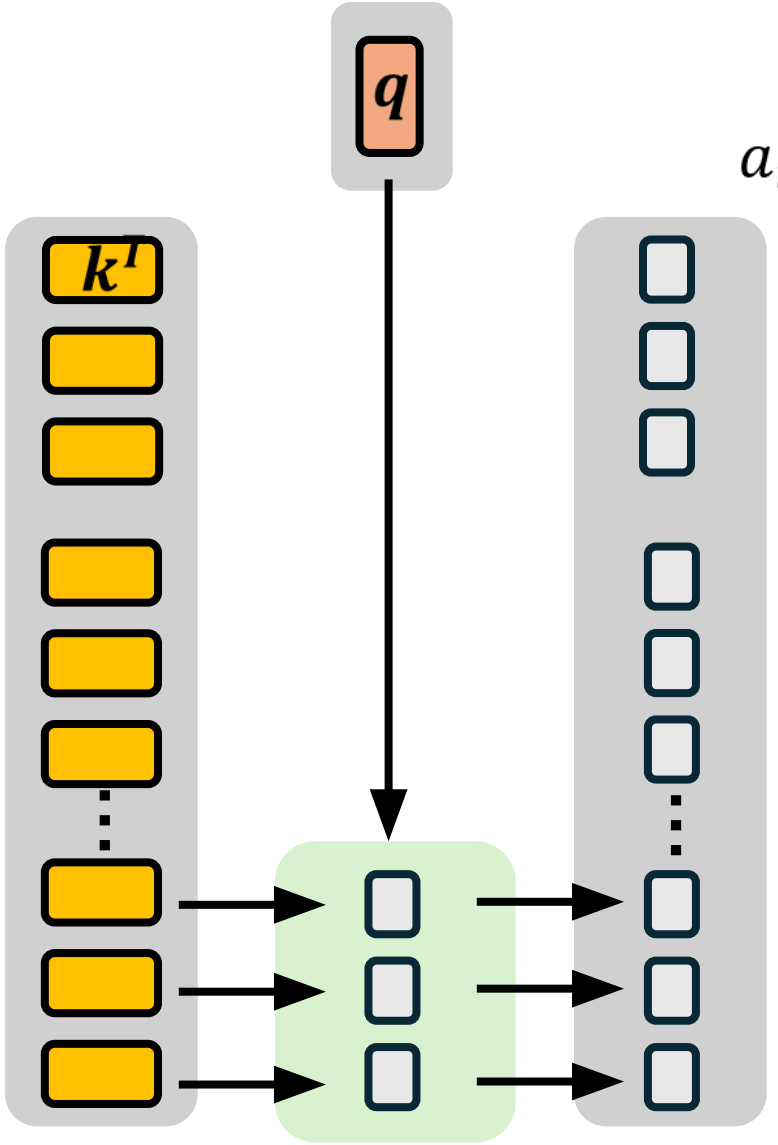
SRAM



倉庫

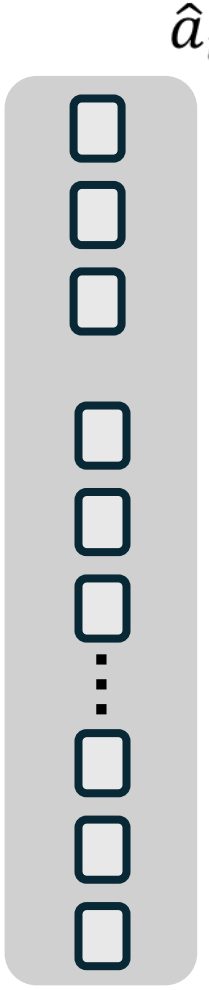
工作臺

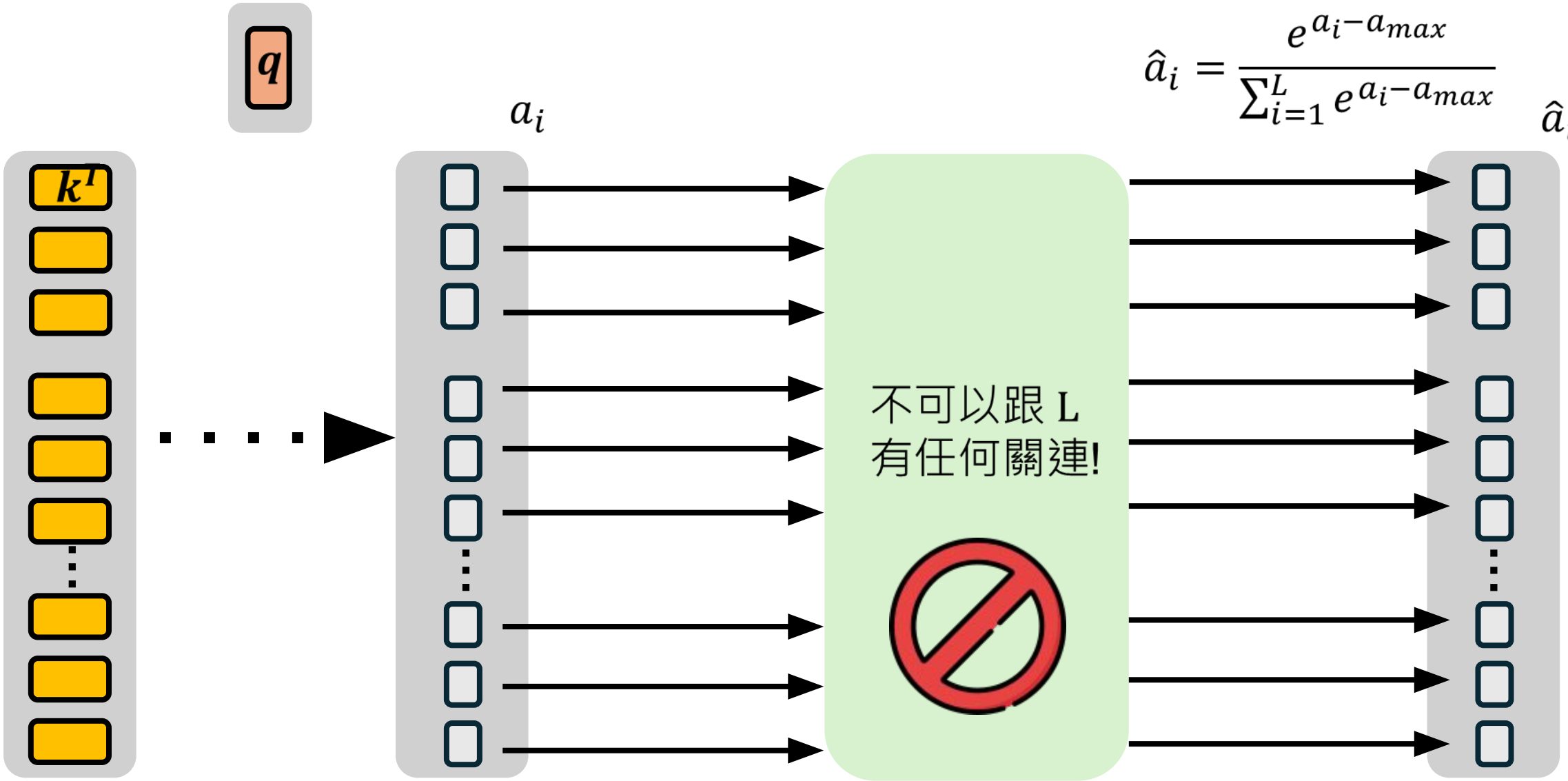

q
k^T

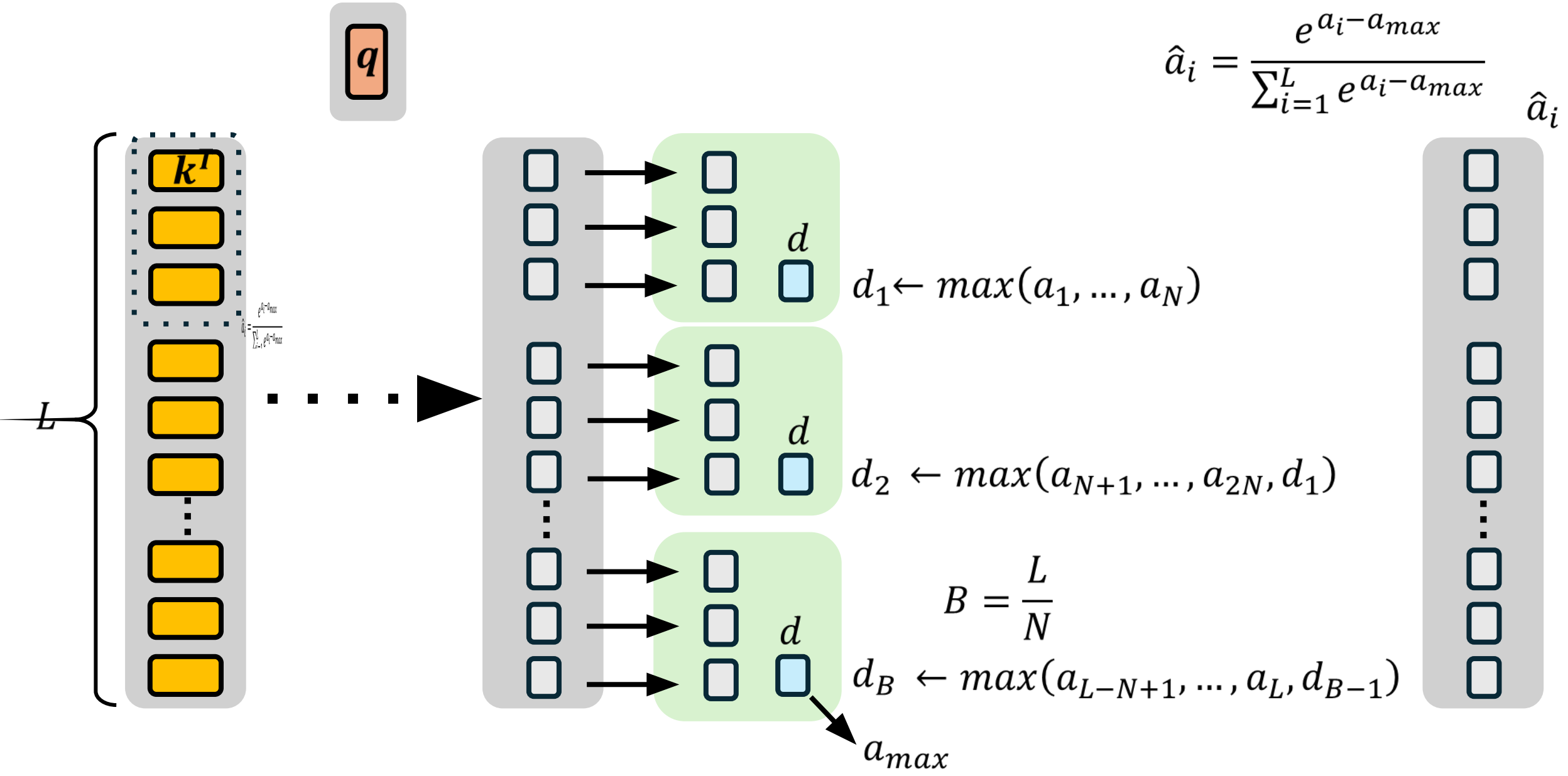


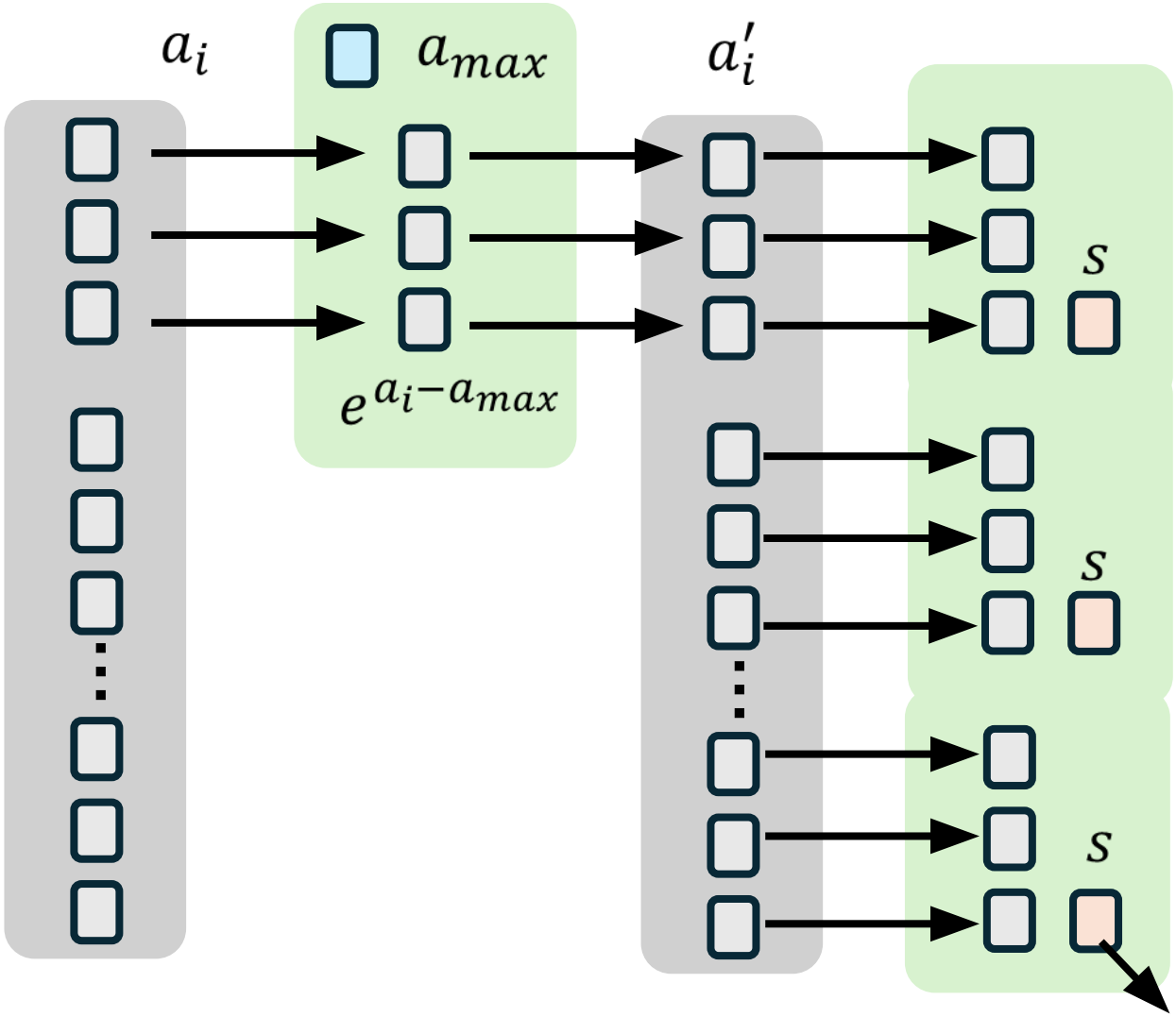
$$\hat{a}_i = \frac{e^{a_i}}{\sum_{i=1}^{L} e^{a_i}}$$

$$\hat{a}_i = \frac{e^{a_i - a_{max}}}{\sum_{i=1}^{L} e^{a_i - a_{max}}}$$




$\hat{a}_i = \frac{e^{a_i - a_{max}}}{\sum_{i=1}^{L} e^{a_i - a_{max}}}$
q
k^1
a_i
$\hat{a}_i$
不可以跟 L
有任何關連!


q
k^T
L
d
d
d
$d_1 \leftarrow max(a_1, \ldots, a_N)$
$d_2 \leftarrow max(a_{N+1}, \ldots, a_{2N}, d_1)$
$B = \frac{L}{N}$
$d_B \leftarrow max(a_{L-N+1}, \ldots, a_L, d_{B-1})$
a_{max}
$\hat{a}_i = \frac{e^{a_i - a_{max}}}{\sum_{i=1}^{L} e^{a_i - a_{max}}}$
$\hat{a}_i$



$$\hat{a}_i = \frac{e^{a_i - a_{max}}}{\sum_{i=1}^{L} e^{a_i - a_{max}}}$$

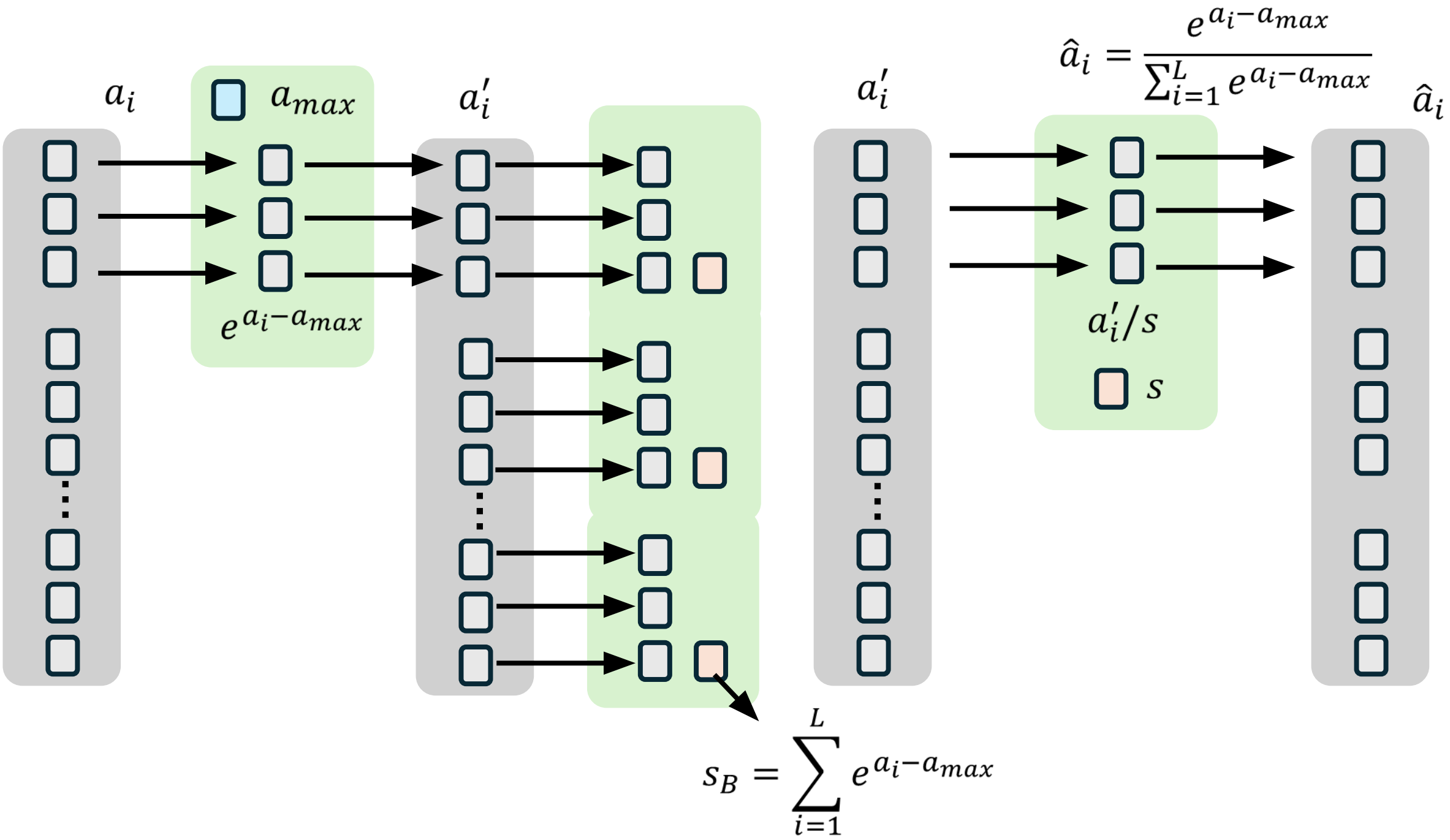
$\hat{a}_i$

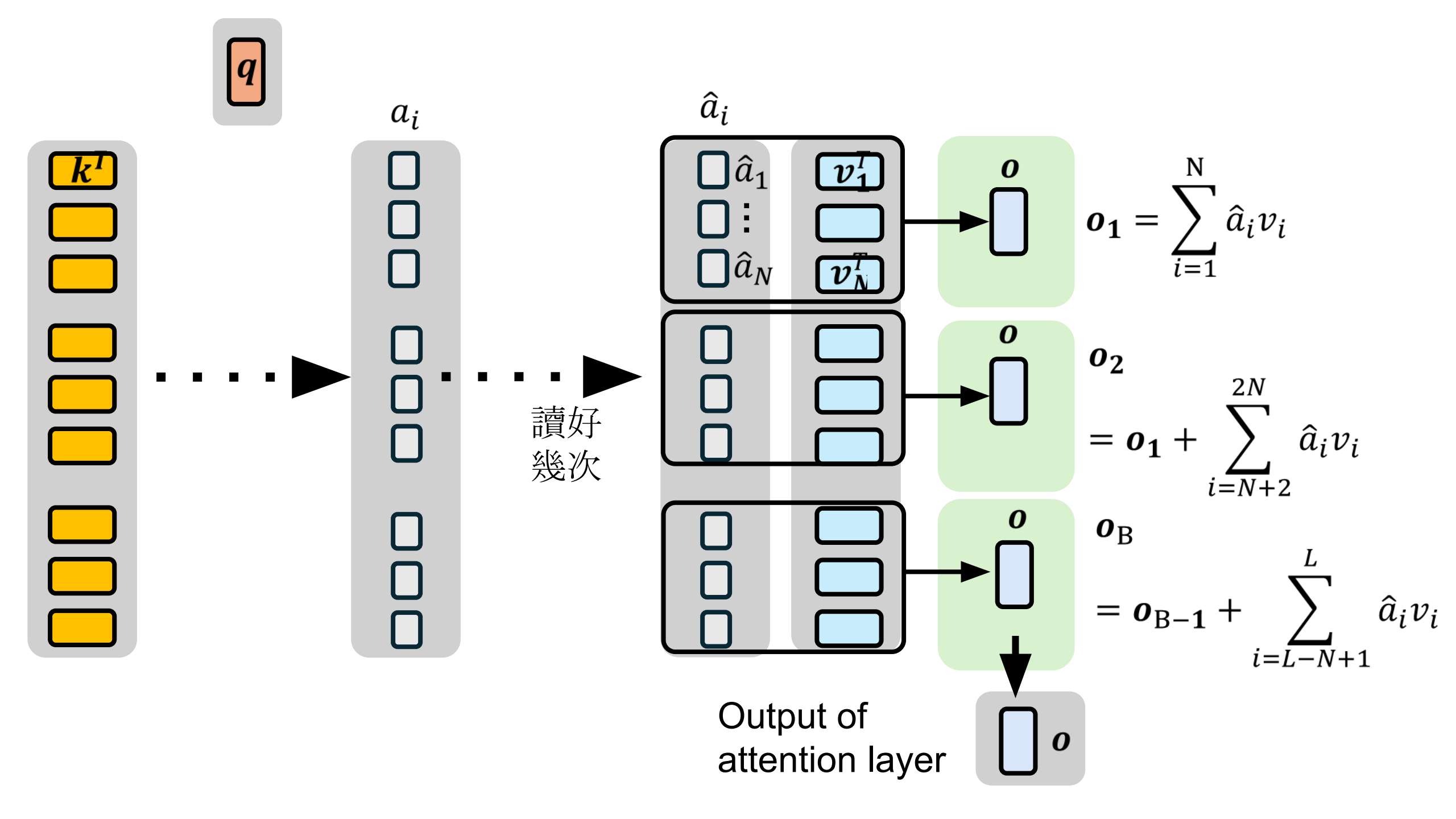
$$s_1 = \sum_{i=1}^{N} a'_i$$

$$s_2 = s_1 + \sum_{i=N+1}^{2N} a'_i$$

$$s_B = s_{B-1} + \sum_{i=L-N+1}^{L} a'_i$$

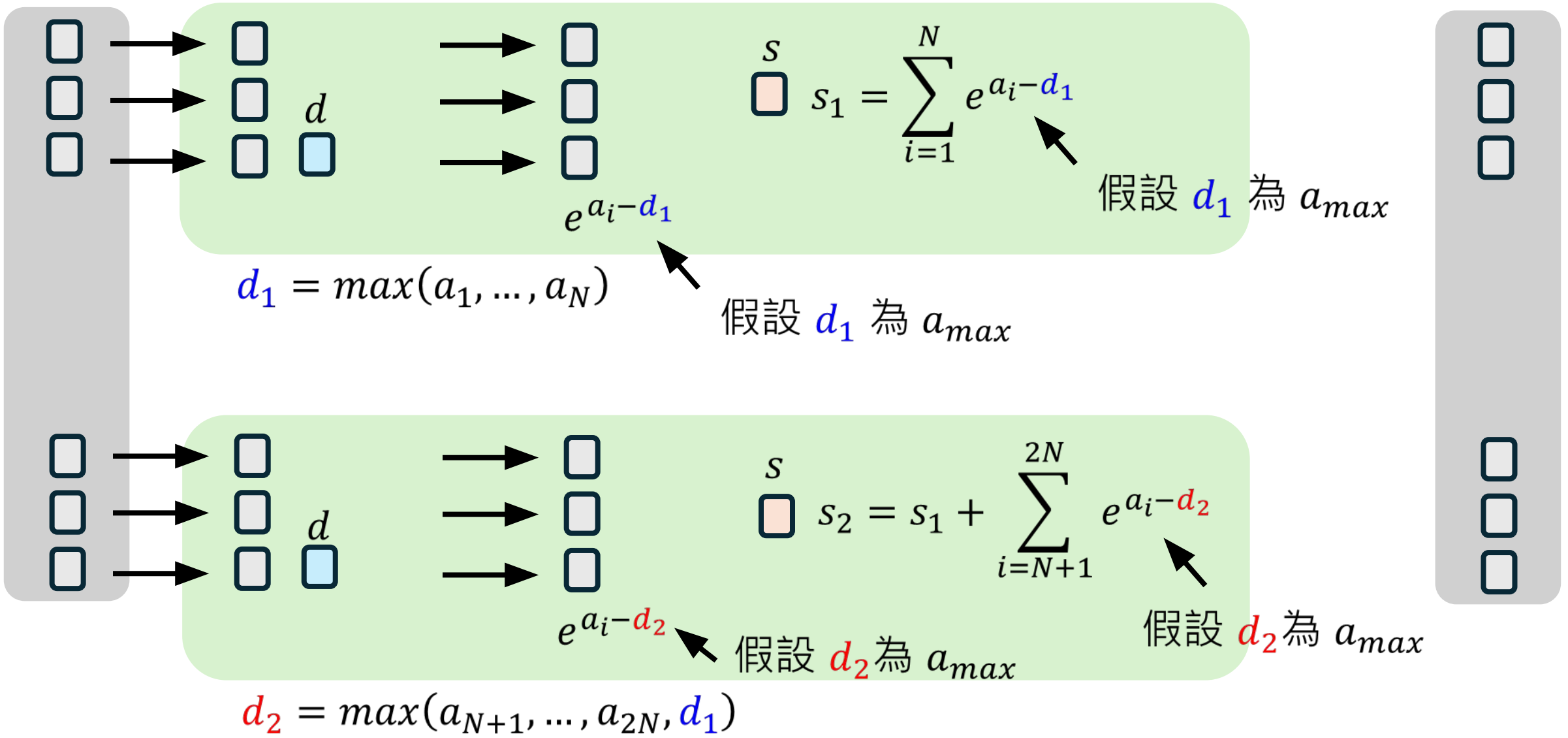
$$\sum_{i=1}^{L} a'_i = \sum_{i=1}^{L} e^{a_i - a_{max}}$$


a_i
a_{max}
a'_i
$e^{a_i - a_{max}}$
$s_B = \sum_{i=1}^{L} e^{a_i - a_{max}}$
a'_i
$\hat{a}_i = \frac{e^{a_i - a_{max}}}{\sum_{i=1}^{L} e^{a_i - a_{max}}}$
$\hat{a}_i$
a'_i / s
s


q
k^1
a_i
$\hat{a}_i$
$\hat{a}_1$
$\hat{a}_N$
v_1^I
v_N^I
o
讀好
幾次
$o_1 = \sum_{i=1}^{N} \hat{a}_i v_i$
$o_2 = o_1 + \sum_{i=N+2}^{2N} \hat{a}_i v_i$
$o_B = o_{B-1} + \sum_{i=L-N+1}^{L} \hat{a}_i v_i$
Output of
attention layer

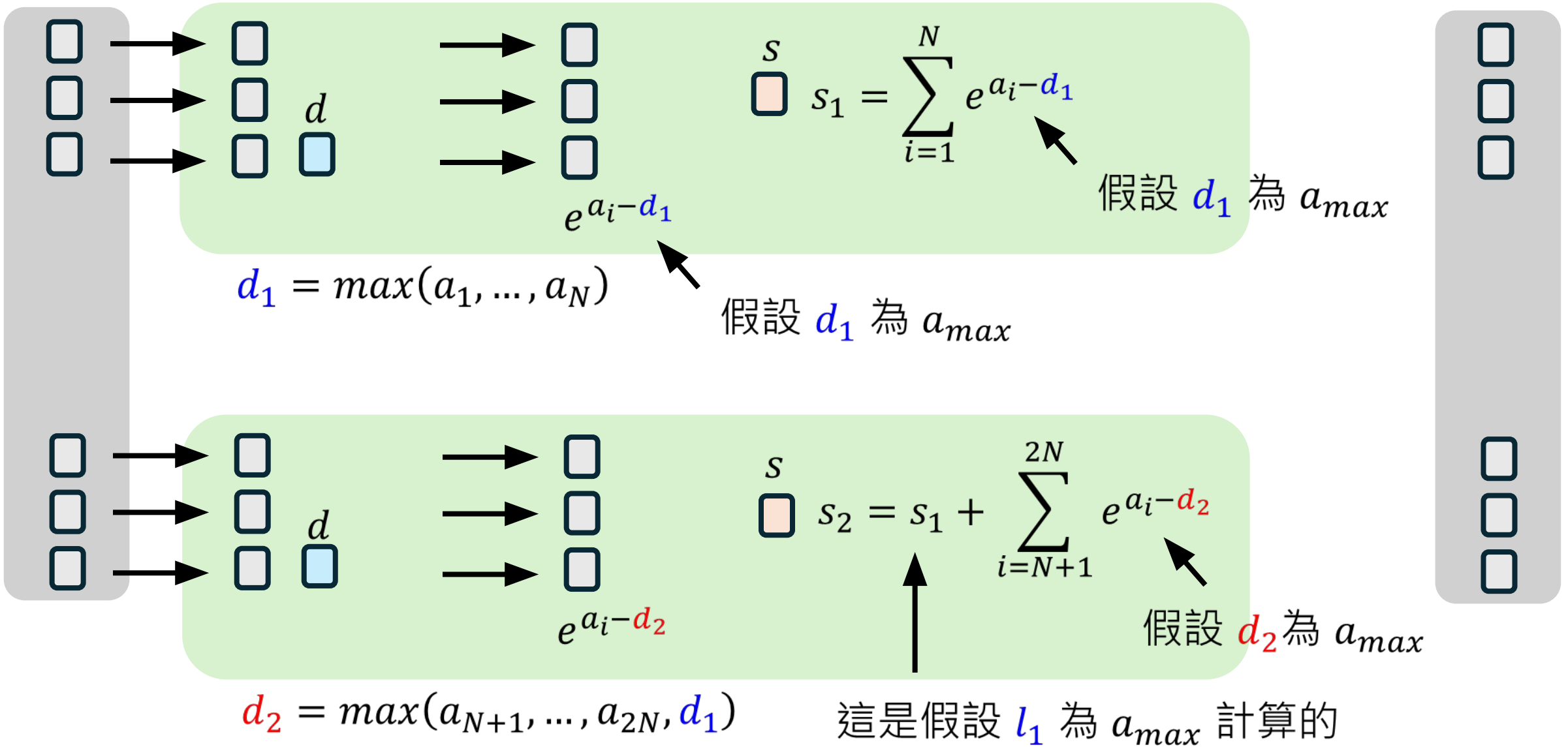
一次找出 a_{max} 和 $\sum_{i=1}^{L} e^{a_i - a_{max}}$

$$\hat{a}_i = \frac{e^{a_i - a_{max}}}{\sum_{i=1}^{L} e^{a_i - a_{max}}}$$



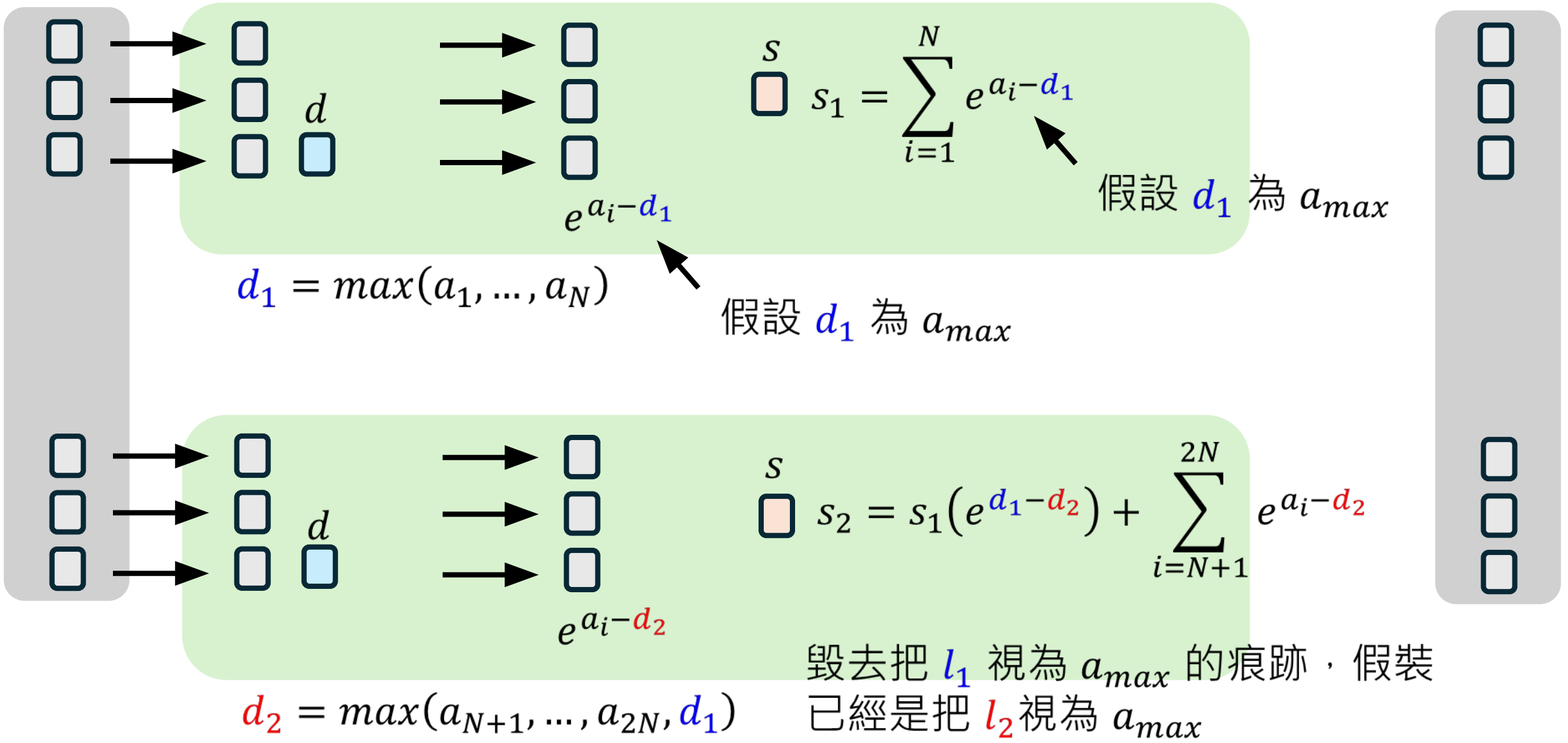
一次找出 a_{max} 和 $\sum_{i=1}^{L} e^{a_i - a_{max}}$

$$\hat{a}_i = \frac{e^{a_i - a_{max}}}{\sum_{i=1}^{L} e^{a_i - a_{max}}}$$



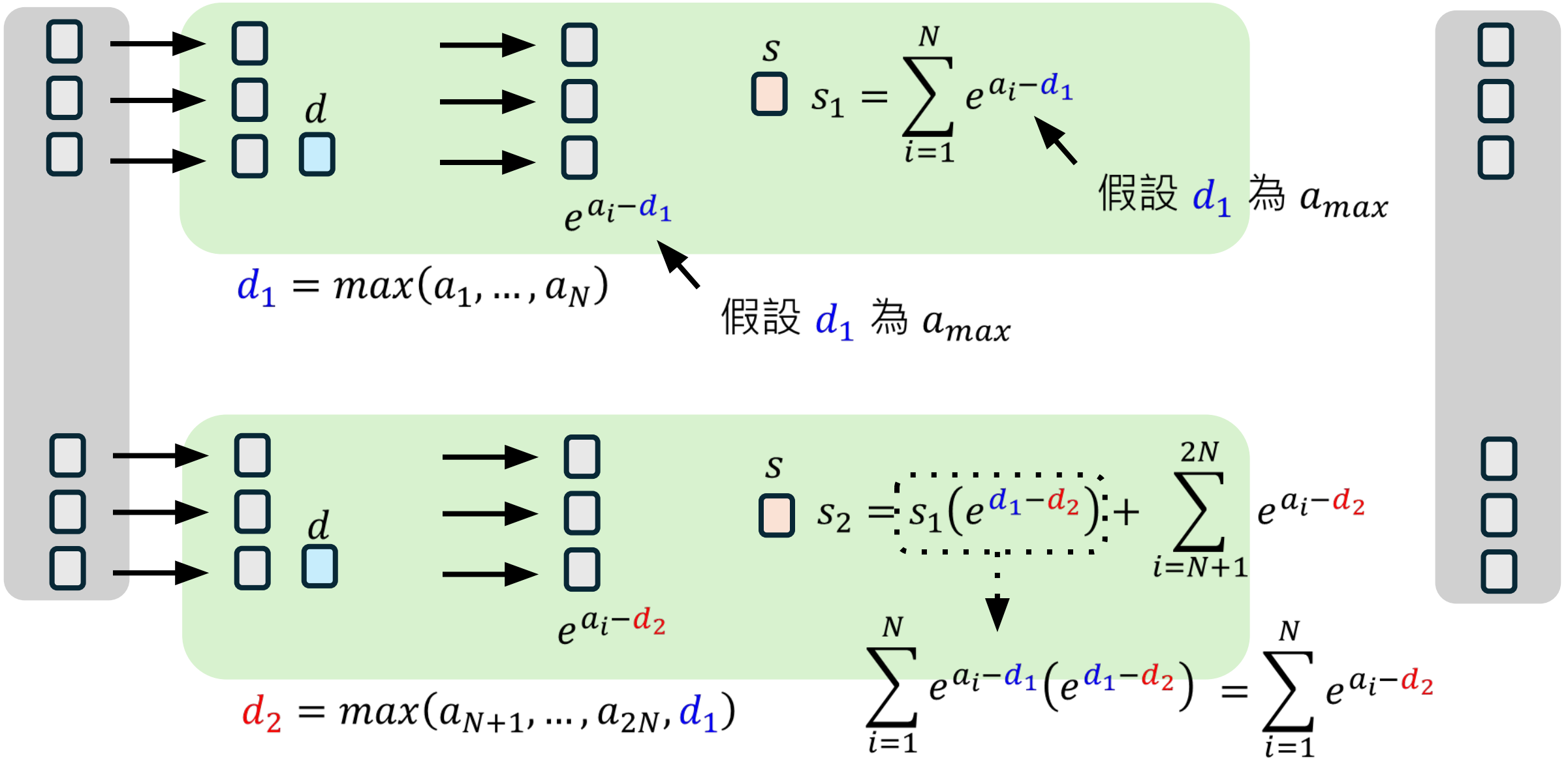
一次找出 a_{max} 和 $\sum_{i=1}^{L} e^{a_i - a_{max}}$

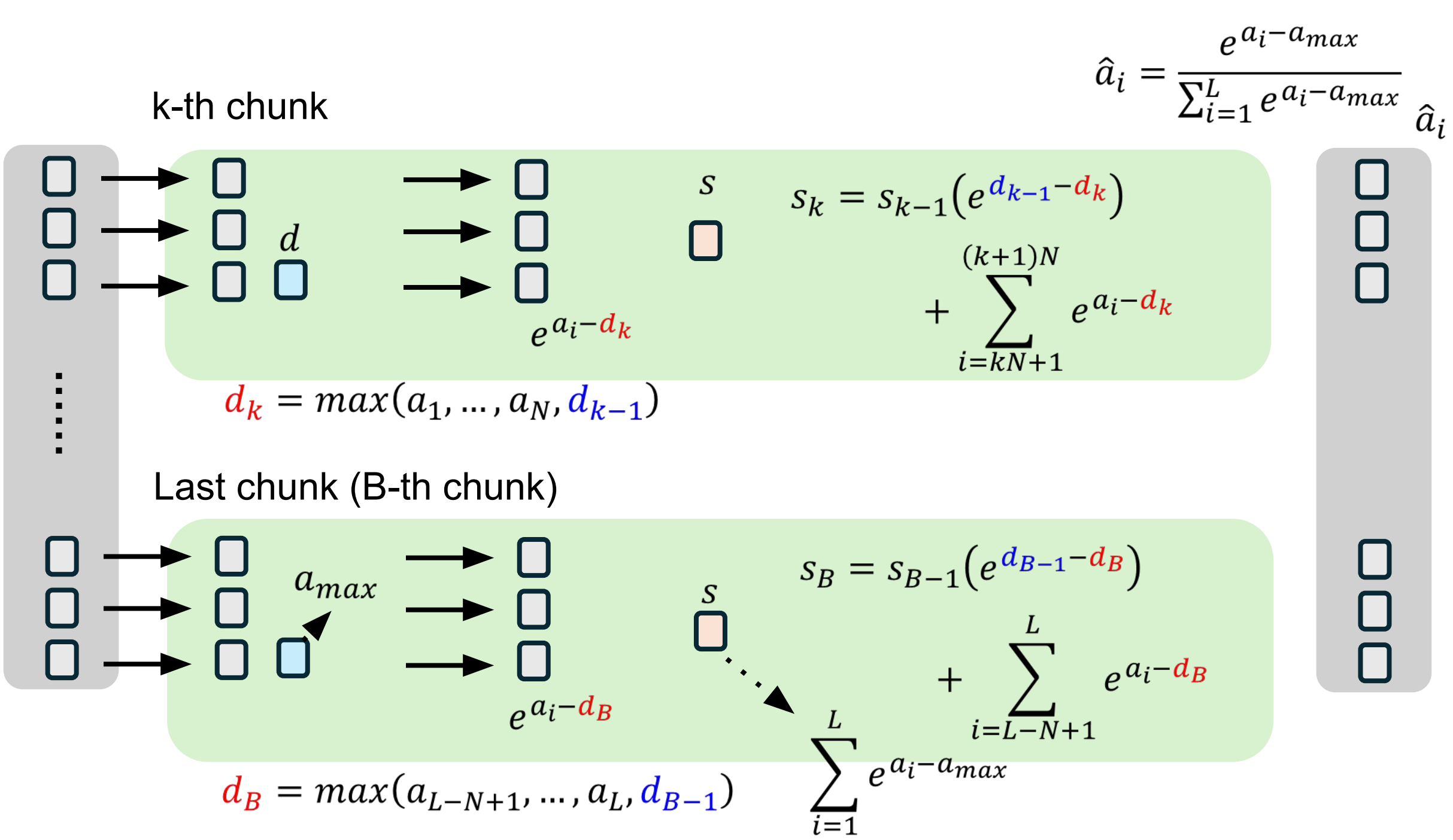
$$\hat{a}_i = \frac{e^{a_i - a_{max}}}{\sum_{i=1}^{L} e^{a_i - a_{max}}}$$

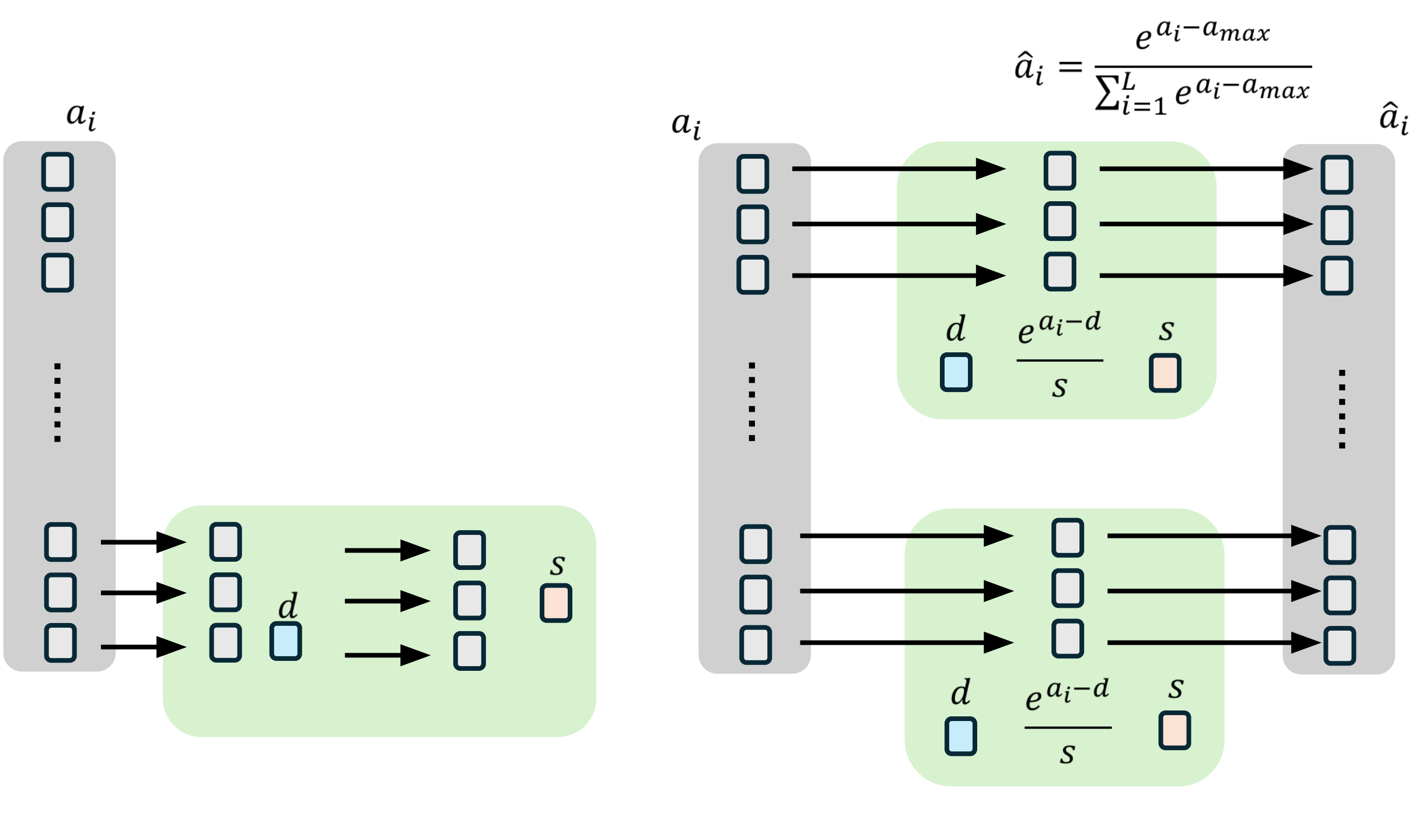


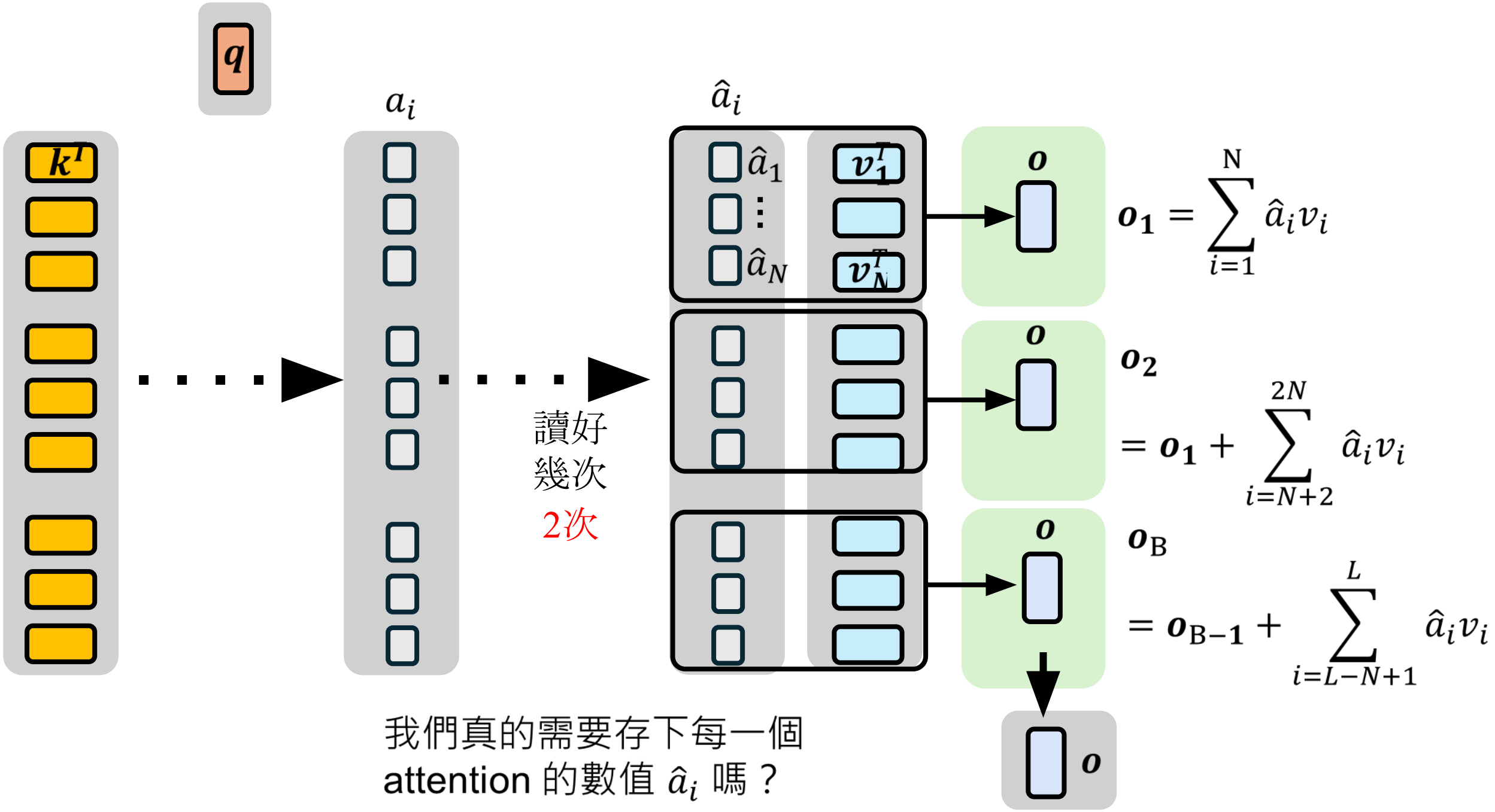
一次找出 a_{max} 和 $\sum_{i=1}^{L} e^{a_i - a_{max}}$

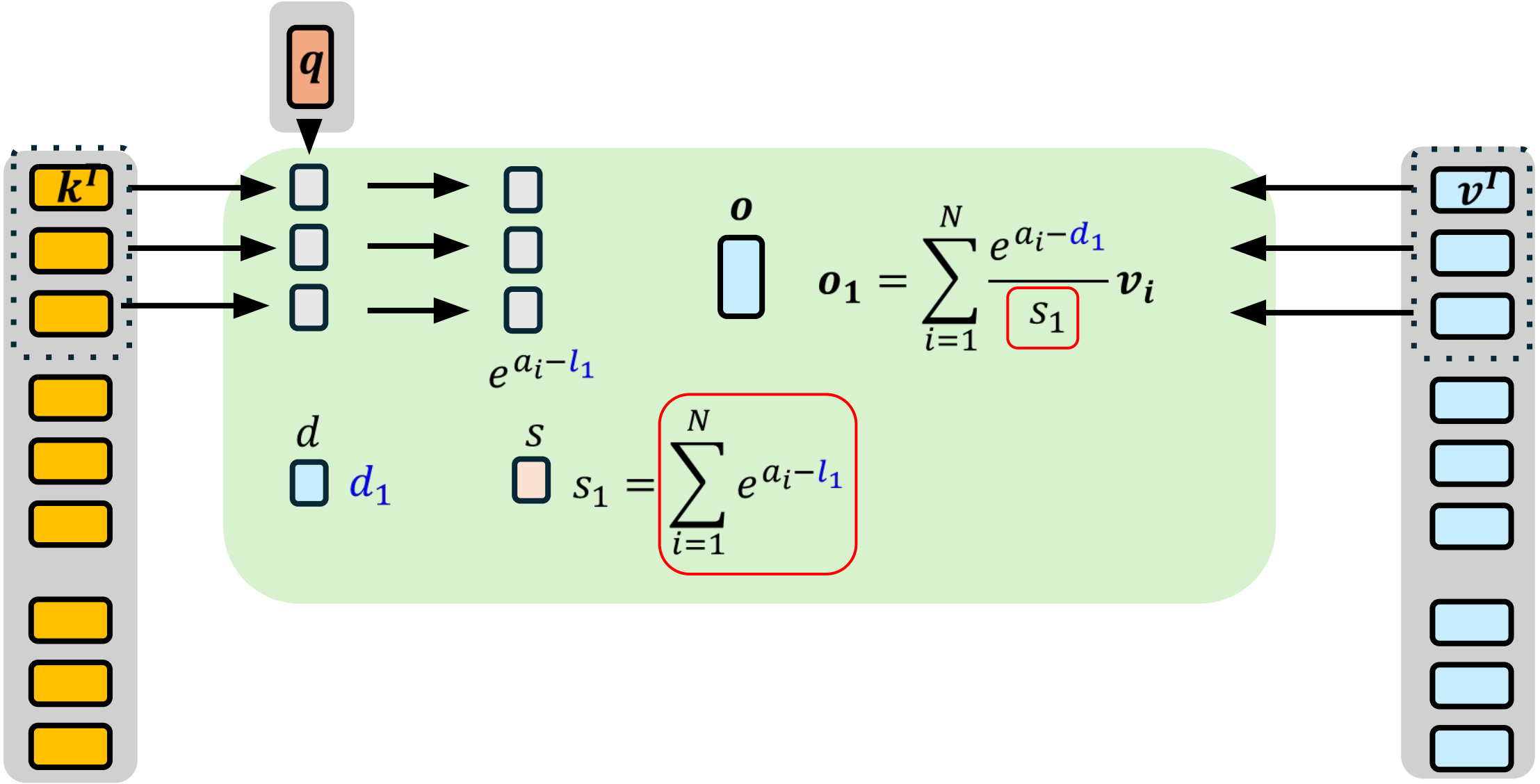
$$\hat{a}_i = \frac{e^{a_i - a_{max}}}{\sum_{i=1}^{L} e^{a_i - a_{max}}}$$

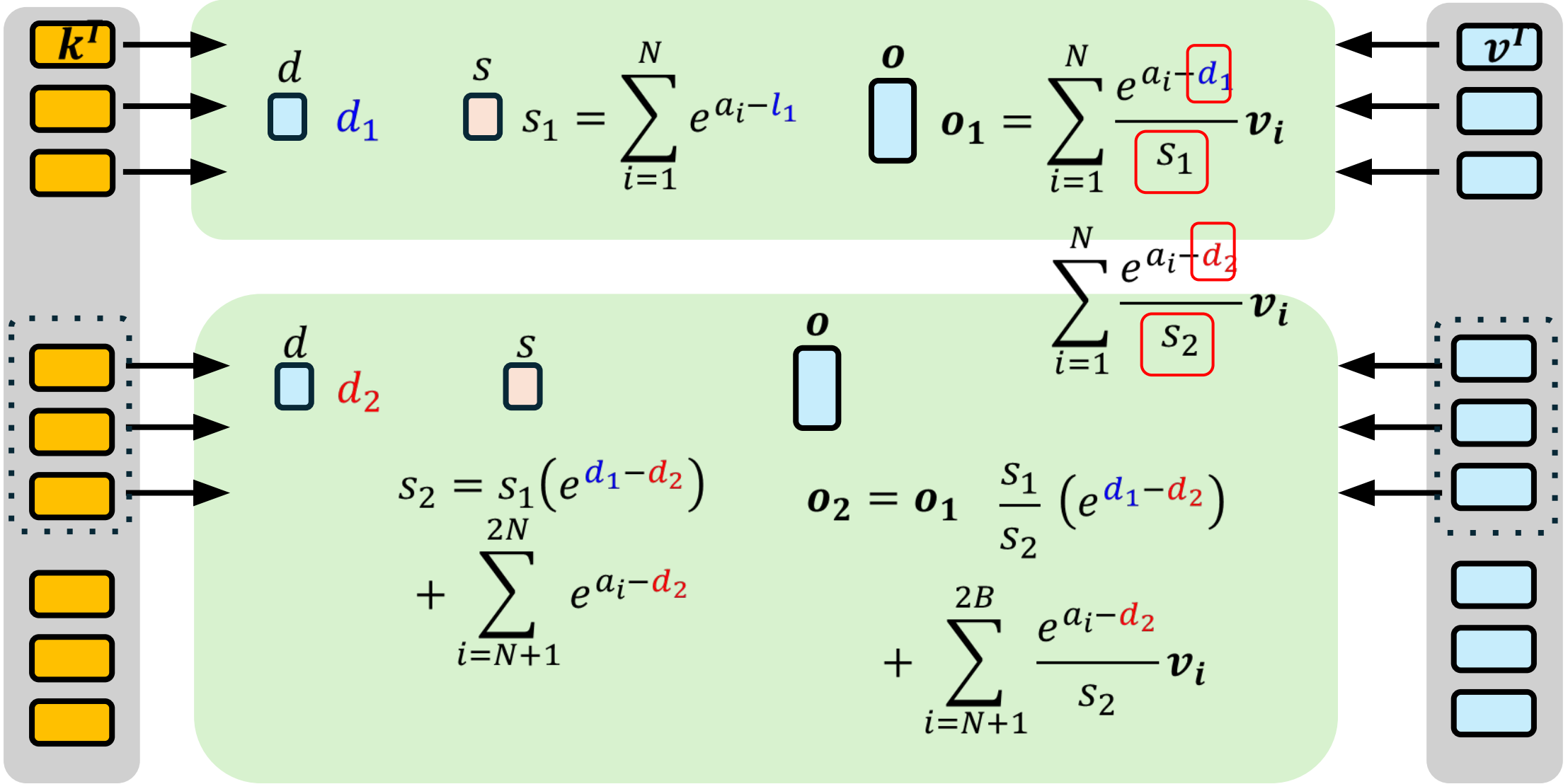


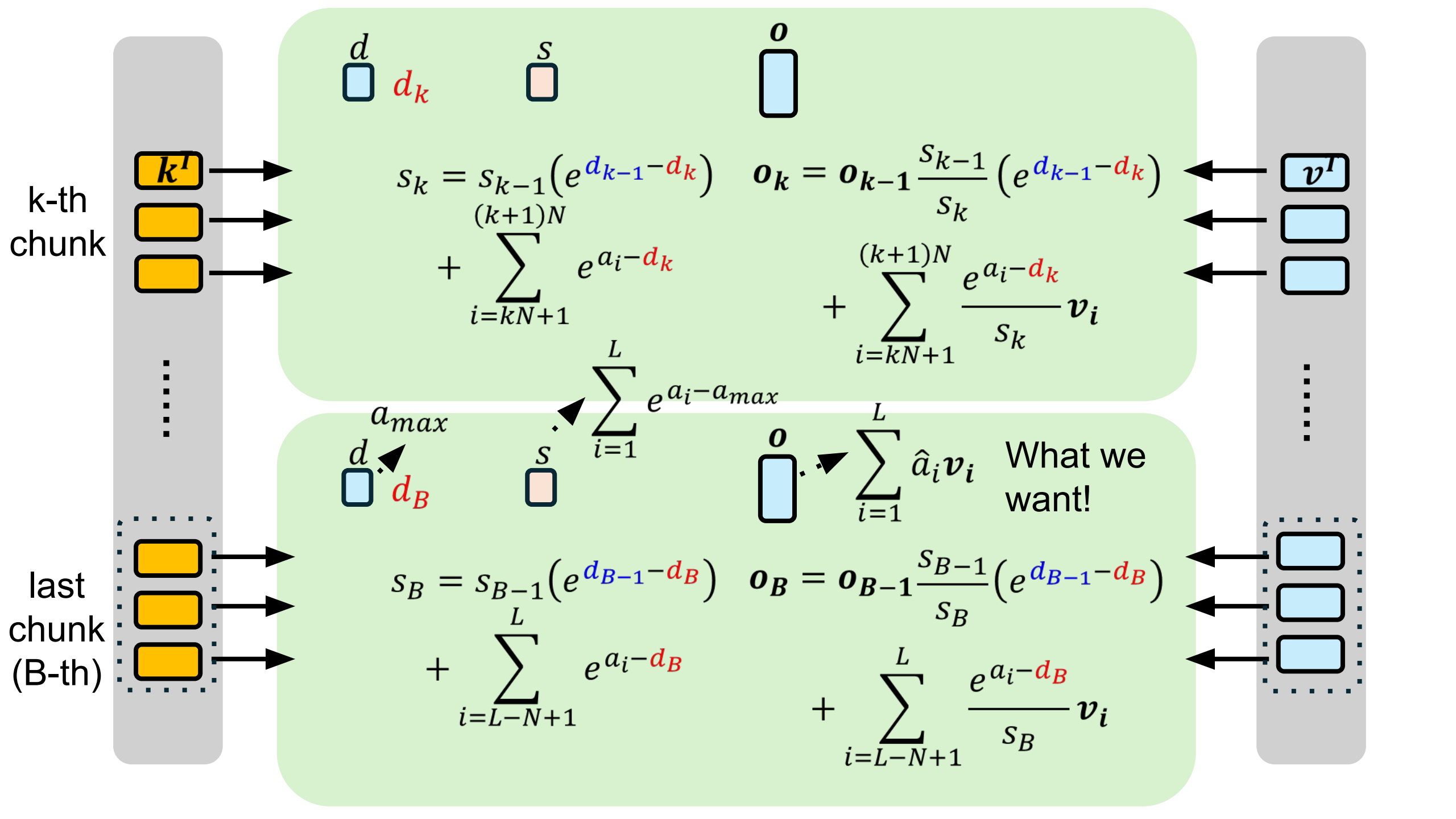

$\hat{a}_i = \frac{e^{a_i - a_{max}}}{\sum_{i=1}^{L} e^{a_i - a_{max}}}$
$\hat{a}_i$
k-th chunk
d
s
$e^{a_i - d_k}$
$s_k = s_{k-1}\left(e^{d_{k-1} - d_k}\right) + \sum_{i=kN+1}^{(k+1)N} e^{a_i - d_k}$
$d_k = max(a_1, \ldots, a_N, d_{k-1})$
Last chunk (B-th chunk)
a_{max}
s
$e^{a_i - d_B}$
$s_B = s_{B-1}\left(e^{d_{B-1} - d_B}\right) + \sum_{i=L-N+1}^{L} e^{a_i - d_B}$
$\sum_{i=1}^{L} e^{a_i - a_{max}}$
$d_B = max(a_{L-N+1}, \ldots, a_L, d_{B-1})$


a_i
d
s
$\hat{a}_i = \frac{e^{a_i - a_{max}}}{\sum_{i=1}^{L} e^{a_i - a_{max}}}$
a_i
$\hat{a}_i$
d
$\frac{e^{a_i - d}}{s}$
s
d
$\frac{e^{a_i - d}}{s}$
s


q
k^1
a_i
$\hat{a}_i$
$\hat{a}_1$
$\hat{a}_N$
v_1^I
v_N^I
讀好
幾次
2次
o
o
o
o
$o_1 = \sum_{i=1}^{N} \hat{a}_i v_i$
$o_2 = o_1 + \sum_{i=N+2}^{2N} \hat{a}_i v_i$
$o_B = o_{B-1} + \sum_{i=L-N+1}^{L} \hat{a}_i v_i$
我們真的需要存下每一個
attention 的數值 $\hat{a}_i$ 嗎？


q
k^1
v^1
$e^{a_i-l_1}$
o
$o_1 = \sum_{i=1}^{N} \frac{e^{a_i-d_1}}{s_1} v_i$
d
d_1
s
$s_1 = \sum_{i=1}^{N} e^{a_i-l_1}$

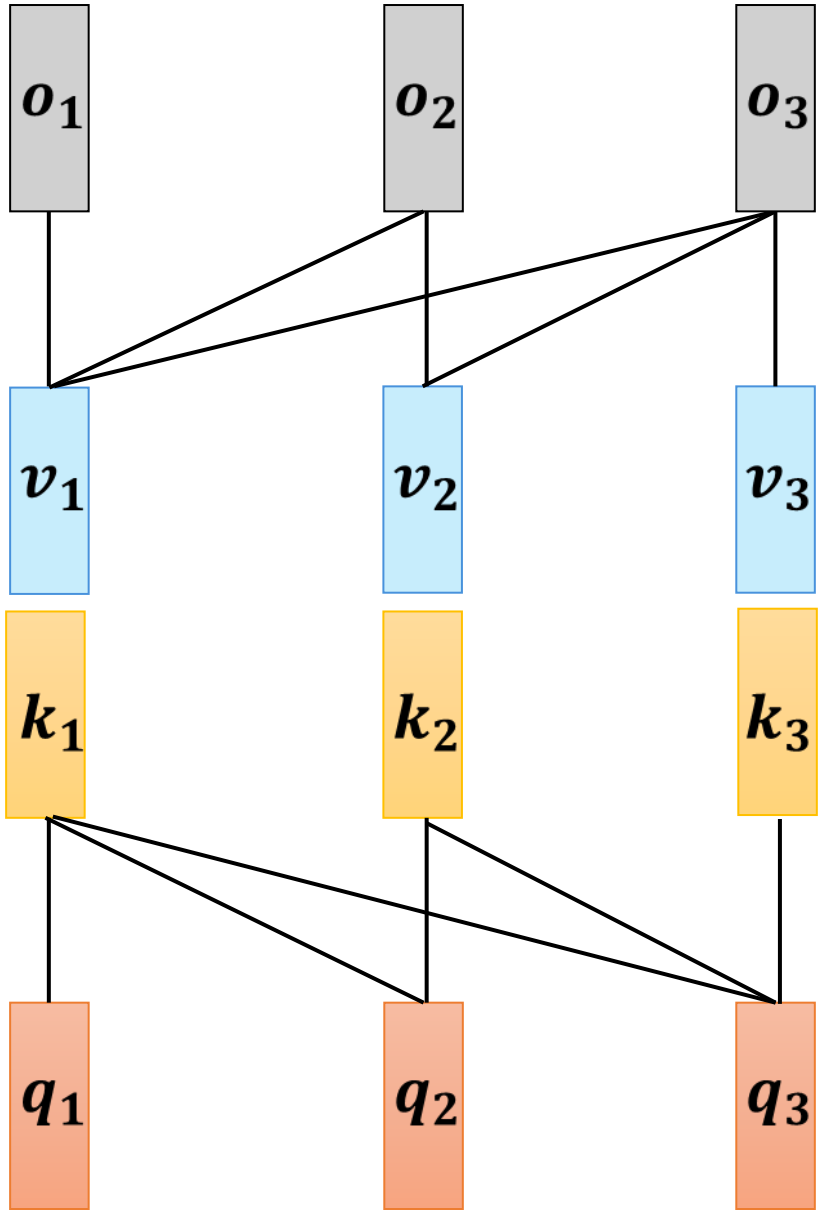

k^I
v^I
d
d_1
s
$s_1 = \sum_{i=1}^{N} e^{a_i - l_1}$
o
$o_1 = \sum_{i=1}^{N} \frac{e^{a_i - d_1}}{s_1} v_i$
d
d_2
s
o
$\sum_{i=1}^{N} \frac{e^{a_i - d_2}}{s_2} v_i$
$s_2 = s_1 (e^{d_1 - d_2}) + \sum_{i=N+1}^{2N} e^{a_i - d_2}$
$o_2 = o_1 \frac{s_1}{s_2} (e^{d_1 - d_2}) + \sum_{i=N+1}^{2B} \frac{e^{a_i - d_2}}{s_2} v_i$

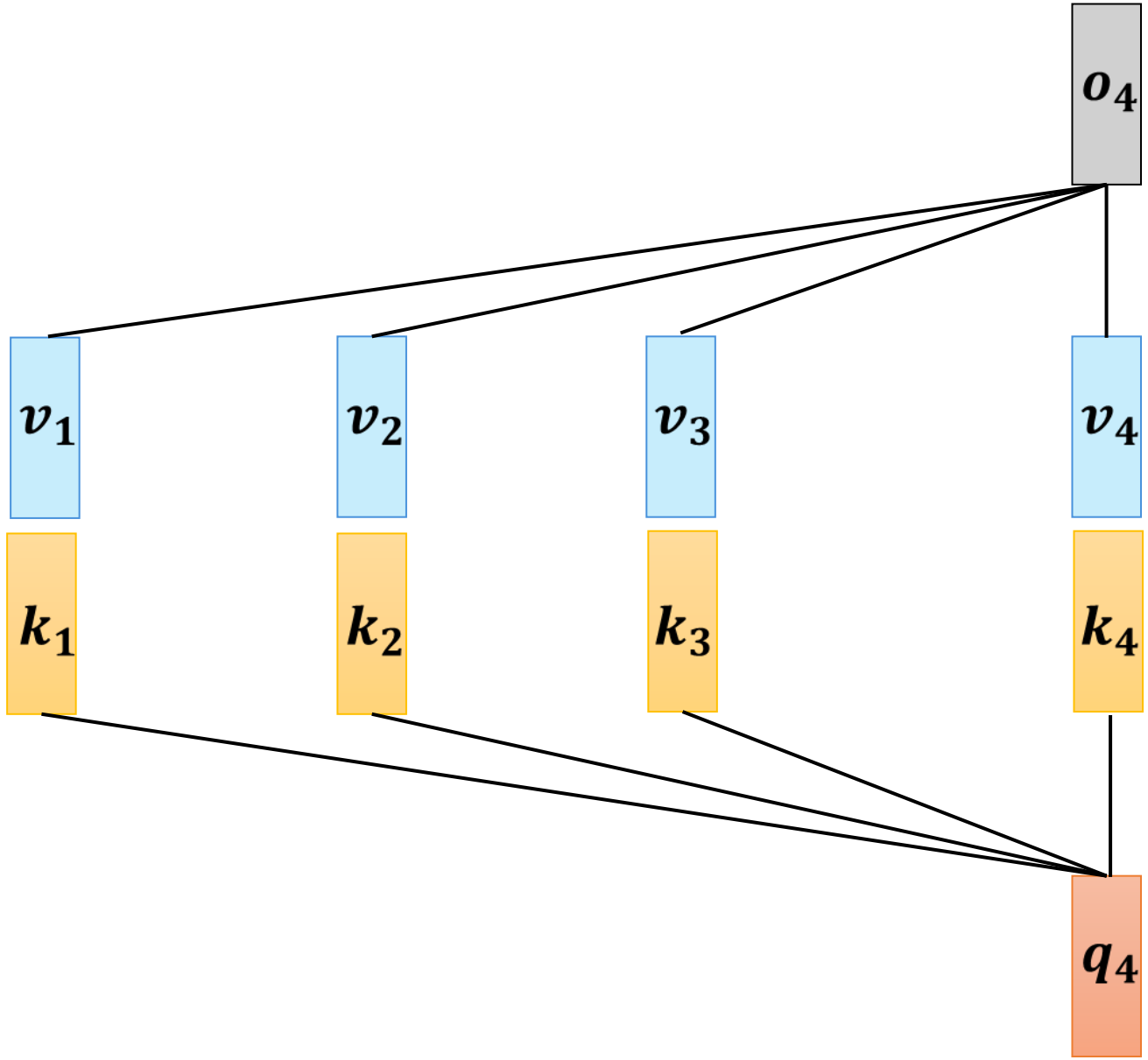

k-th chunk
last chunk (B-th)
k^T
v^T
d
d_k
s
o
$s_k = s_{k-1}\left(e^{d_{k-1}-d_k}\right) + \sum_{i=kN+1}^{(k+1)N} e^{a_i-d_k}$
$o_k = o_{k-1}\frac{s_{k-1}}{s_k}\left(e^{d_{k-1}-d_k}\right) + \sum_{i=kN+1}^{(k+1)N} \frac{e^{a_i-d_k}}{s_k} v_i$
a_{max}
$\sum_{i=1}^{L} e^{a_i-a_{max}}$
d_B
$\sum_{i=1}^{L} \hat{a}_i v_i$
What we want!
$s_B = s_{B-1}\left(e^{d_{B-1}-d_B}\right) + \sum_{i=L-N+1}^{L} e^{a_i-d_B}$
$o_B = o_{B-1}\frac{s_{B-1}}{s_B}\left(e^{d_{B-1}-d_B}\right) + \sum_{i=L-N+1}^{L} \frac{e^{a_i-d_B}}{s_B} v_i$

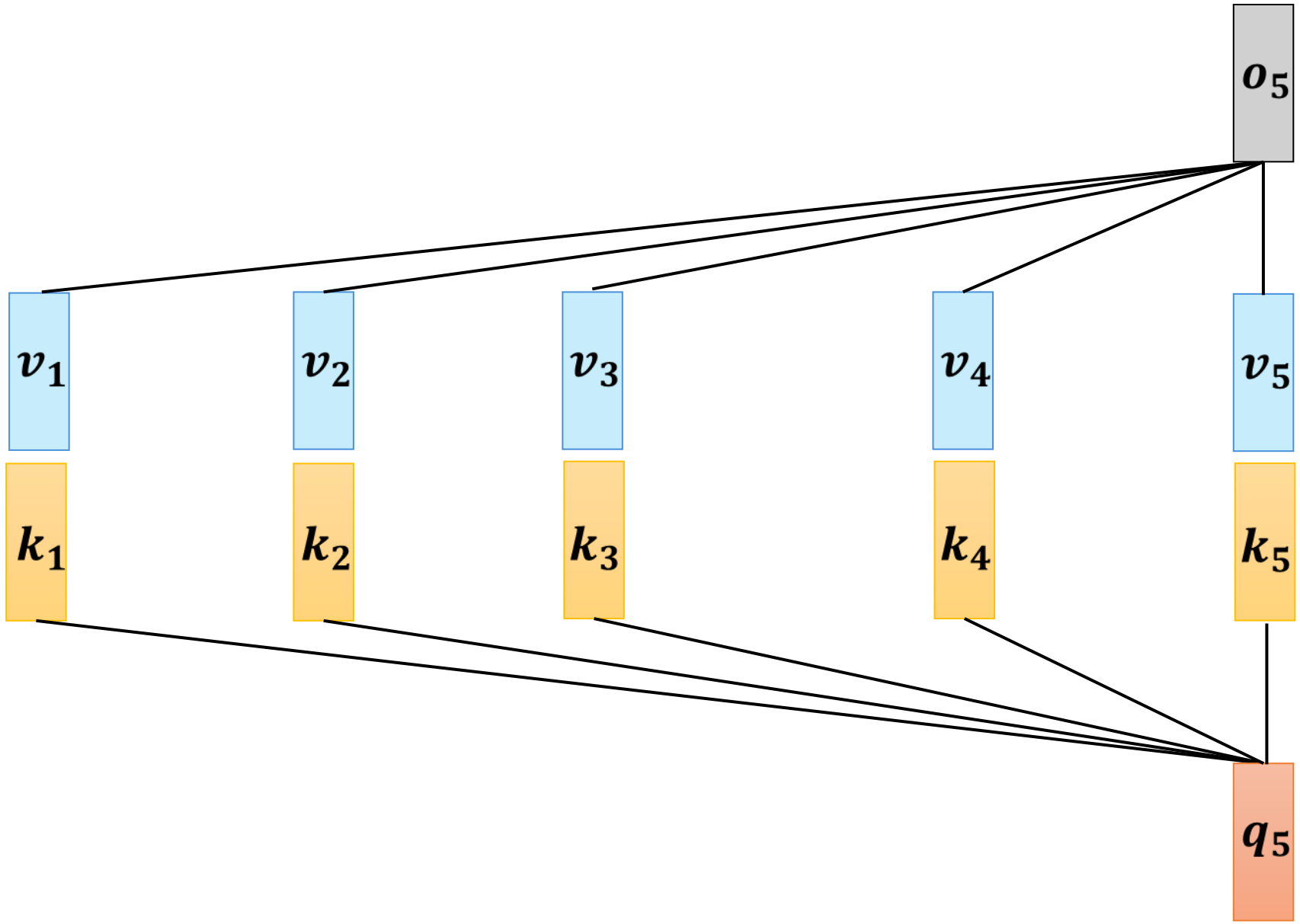
範例程式

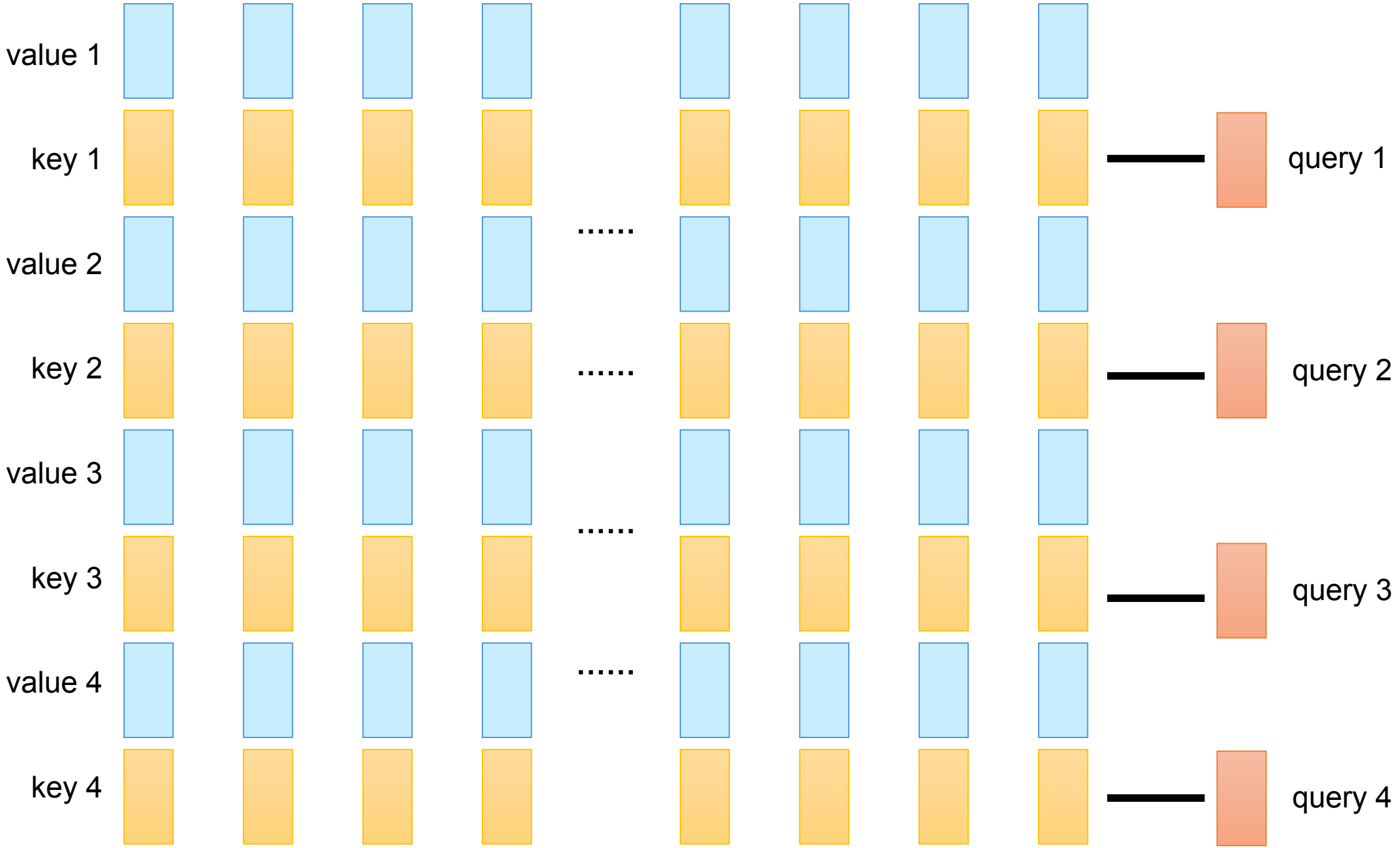
- https://colab.research.google.com/drive/1KoeKKIXSXI9b-pYg0kun3-uLQkP6p_hC?usp=sharing

KV Cache


o_1
o_2
o_3
v_1
v_2
v_3
k_1
k_2
k_3
q_1
q_2
q_3


o_4
v_1
v_2
v_3
v_4
k_1
k_2
k_3
k_4
q_4


o_5
v_1
v_2
v_3
v_4
v_5
k_1
k_2
k_3
k_4
k_5
q_5


value 1
key 1
value 2
key 2
value 3
key 3
value 4
key 4
query 1
query 2
query 3
query 4

HBM

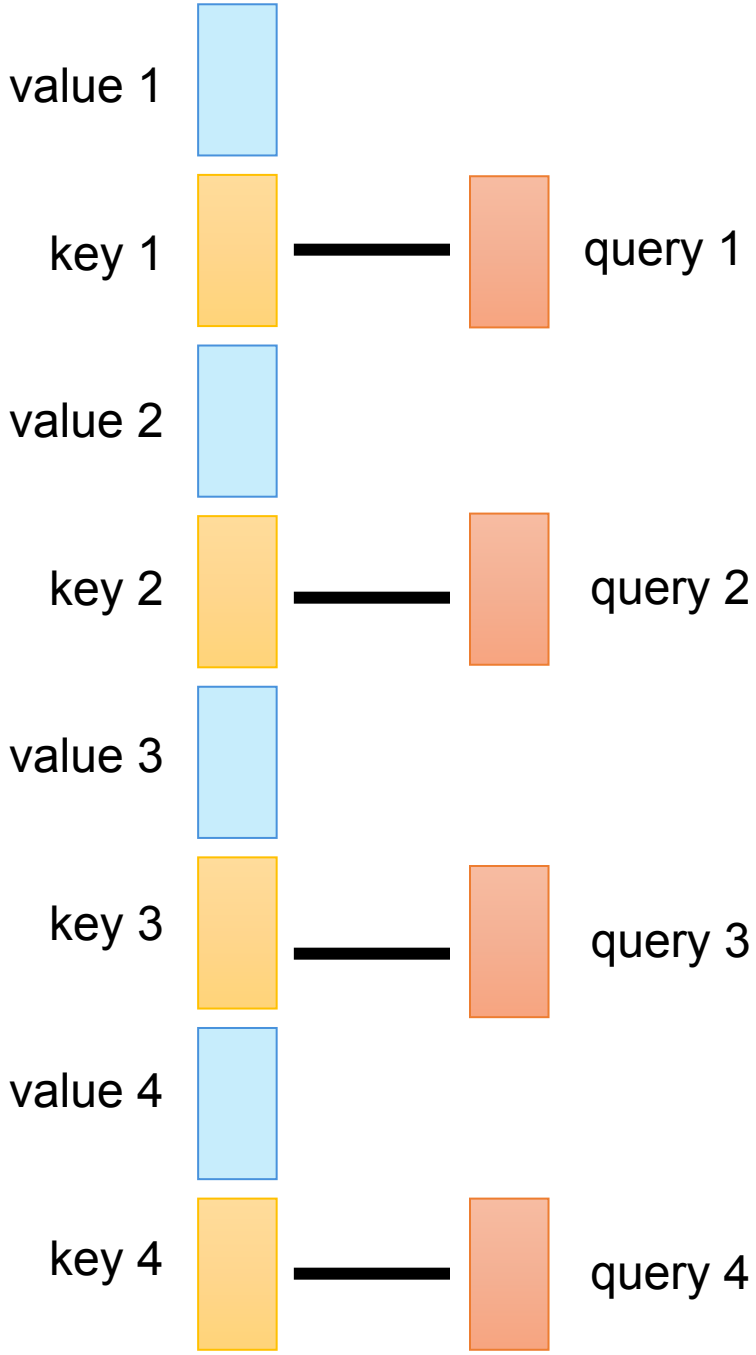
SRAM

Gemma 2

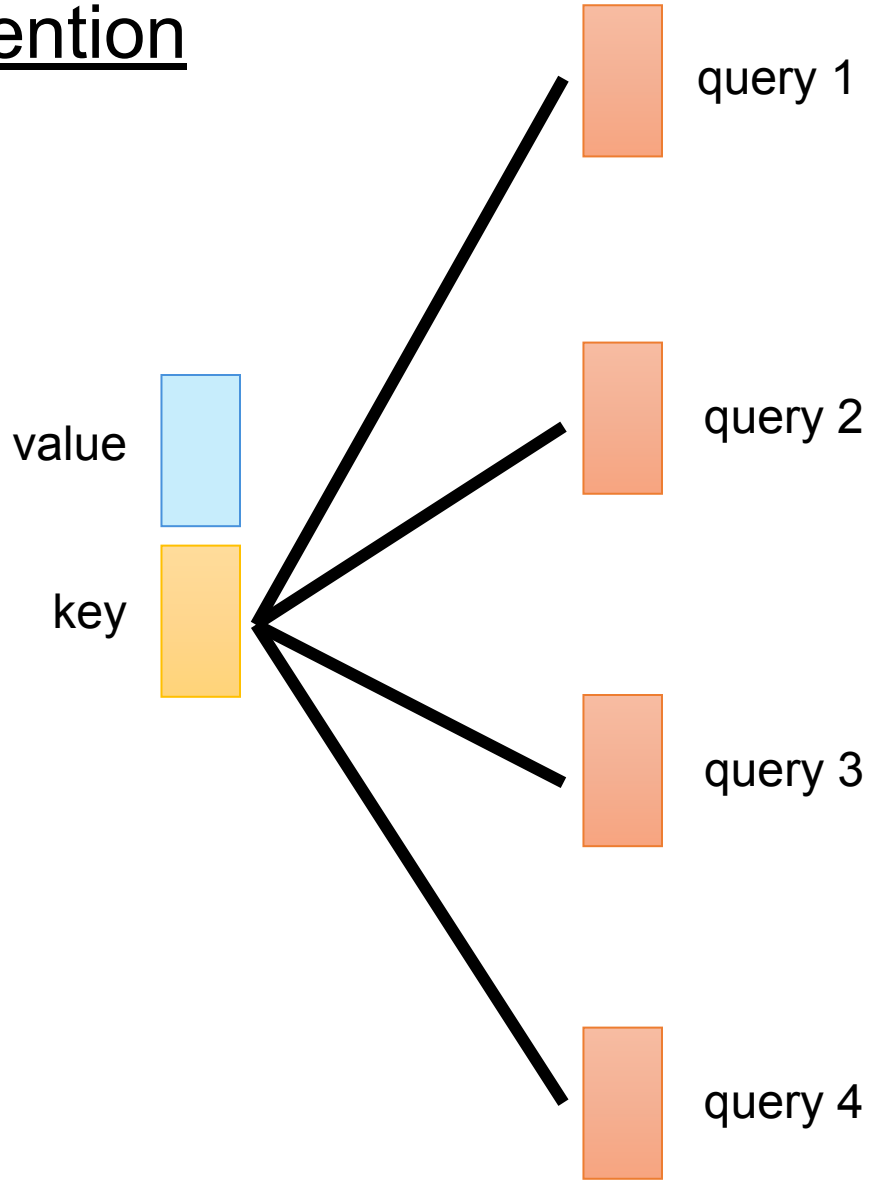
Parameters	2B	9B	27B
d_model	2304	3584	4608
Layers	26	42	46
Pre-norm	yes	yes	yes
Post-norm	yes	yes	yes
Non-linearity	GeGLU	GeGLU	GeGLU
Feedforward dim	18432	28672	73728
Head type	GQA	GQA	GQA
Num heads	8	16	32
Num KV heads	4	8	16
Head size	256	256	128
Global att. span	8192	8192	8192
Sliding window	4096	4096	4096
Vocab size	256128	256128	256128
Tied embedding	yes	yes	yes

Each toke:
46 (layers) x 32 (heads) x 128 (dimension) x 2 (FP16) x 2 (value, key) = 753664 bytes (736 KB, 0.72MB)

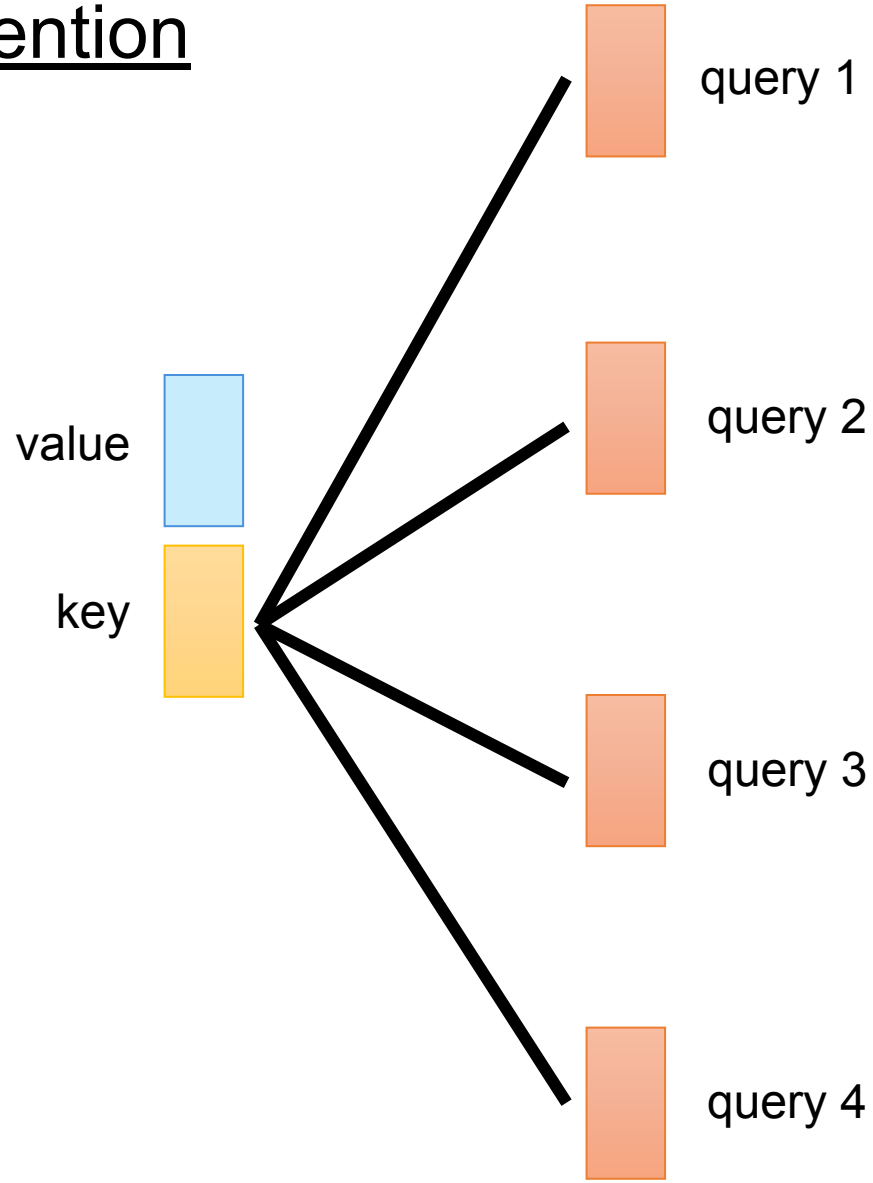
A100 (80GB):
About 114k tokens



Multi-query attention

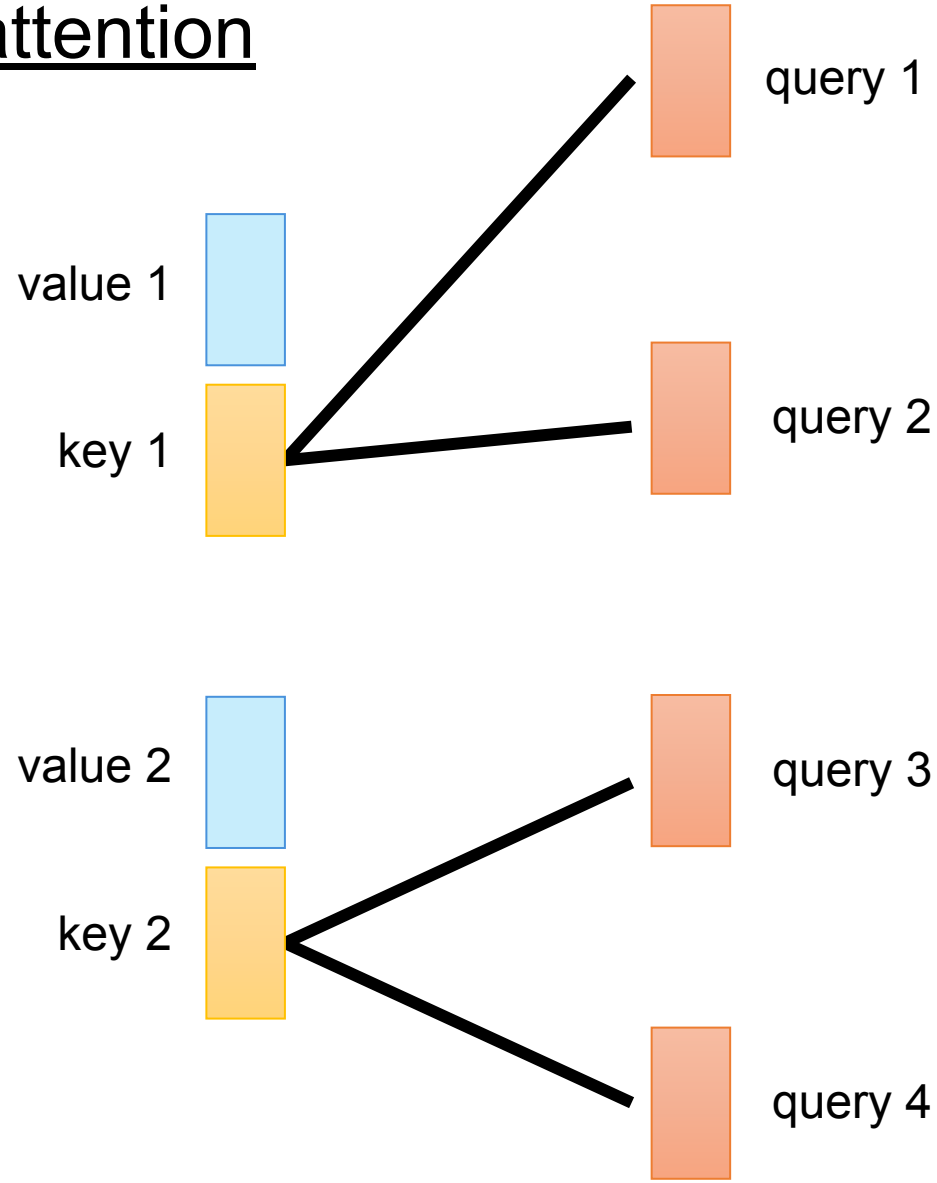


Multi-query attention

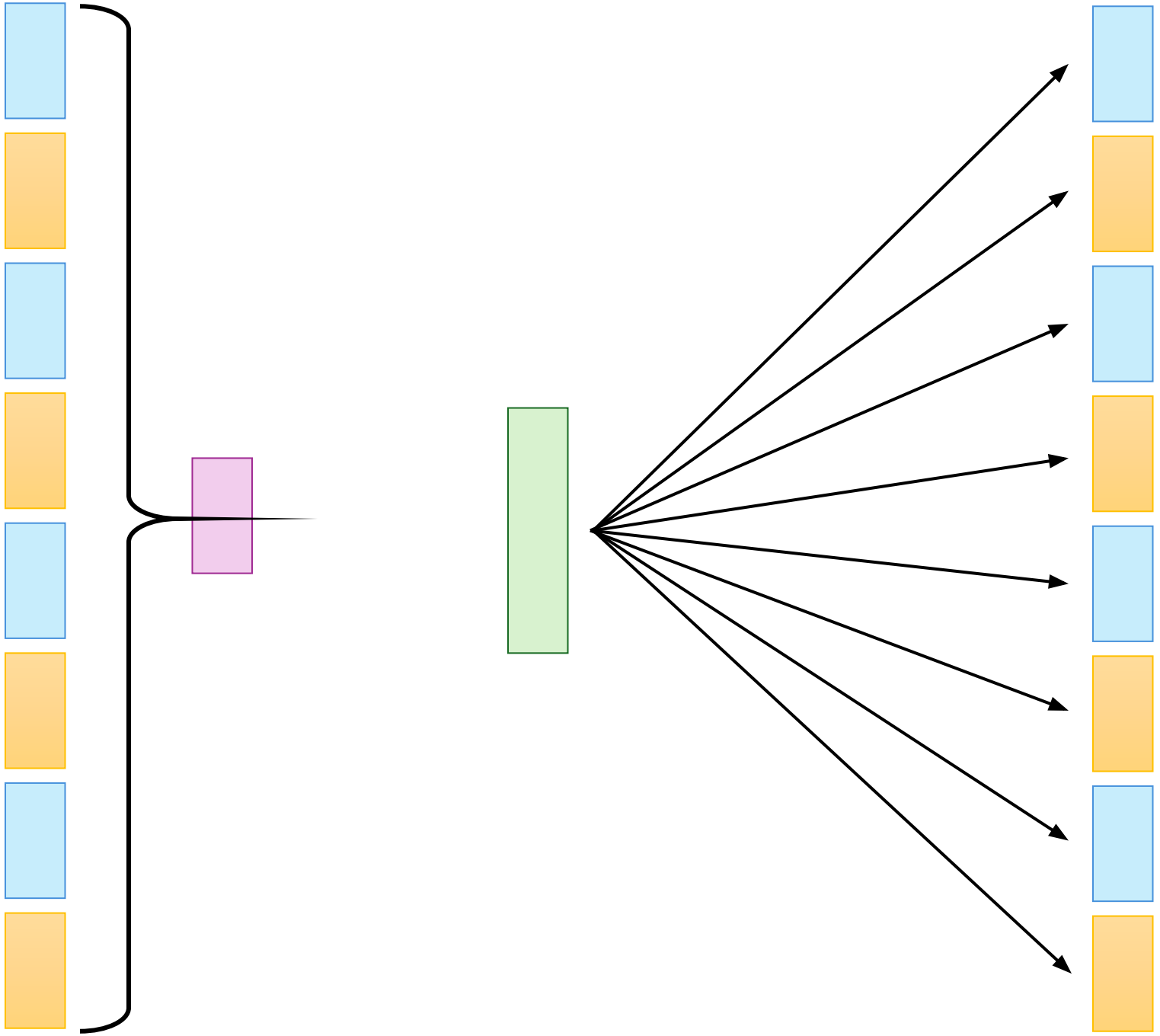


Llama, Gemma, etc.

Group-query attention

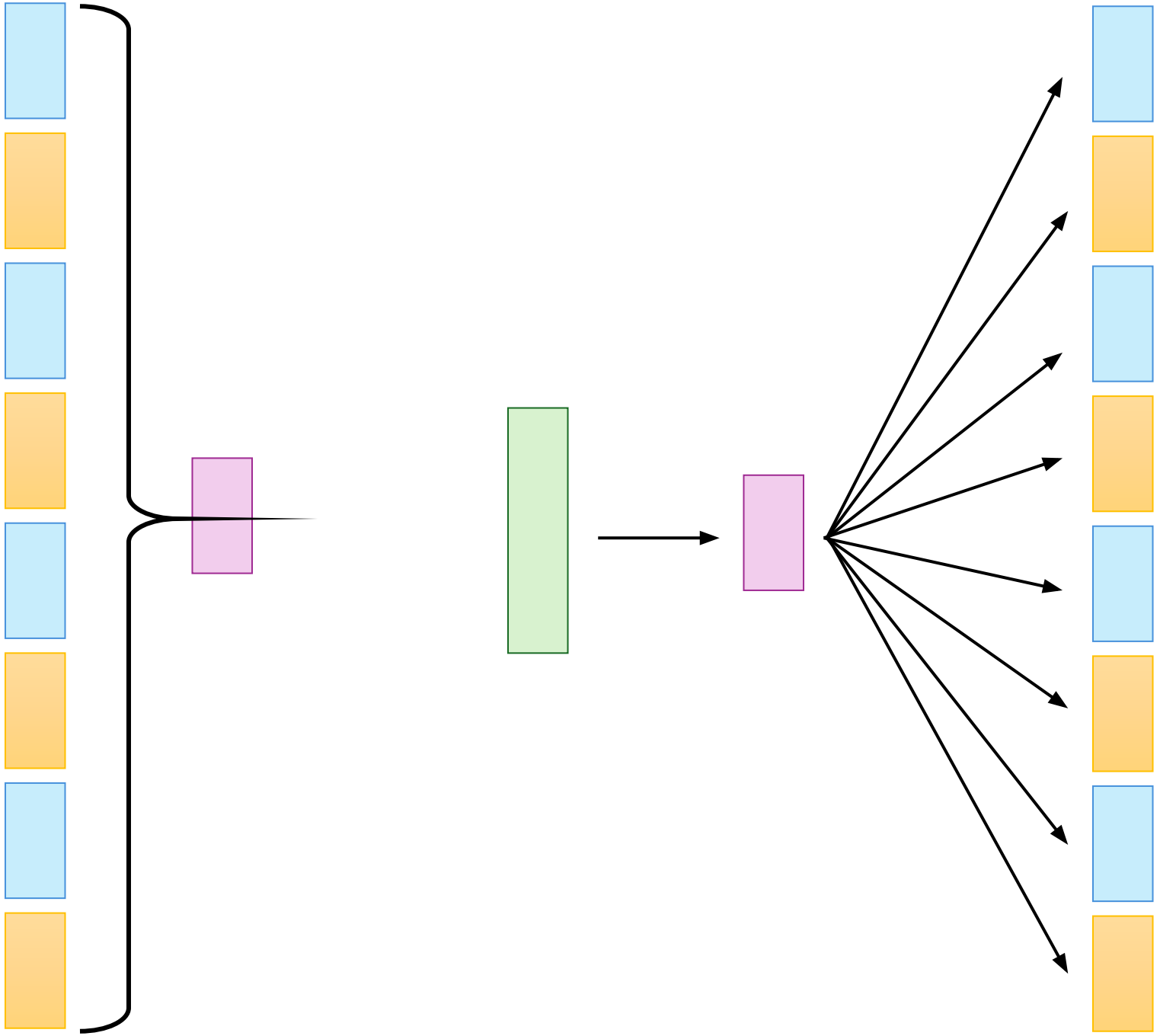


Multi-head Latent Attention



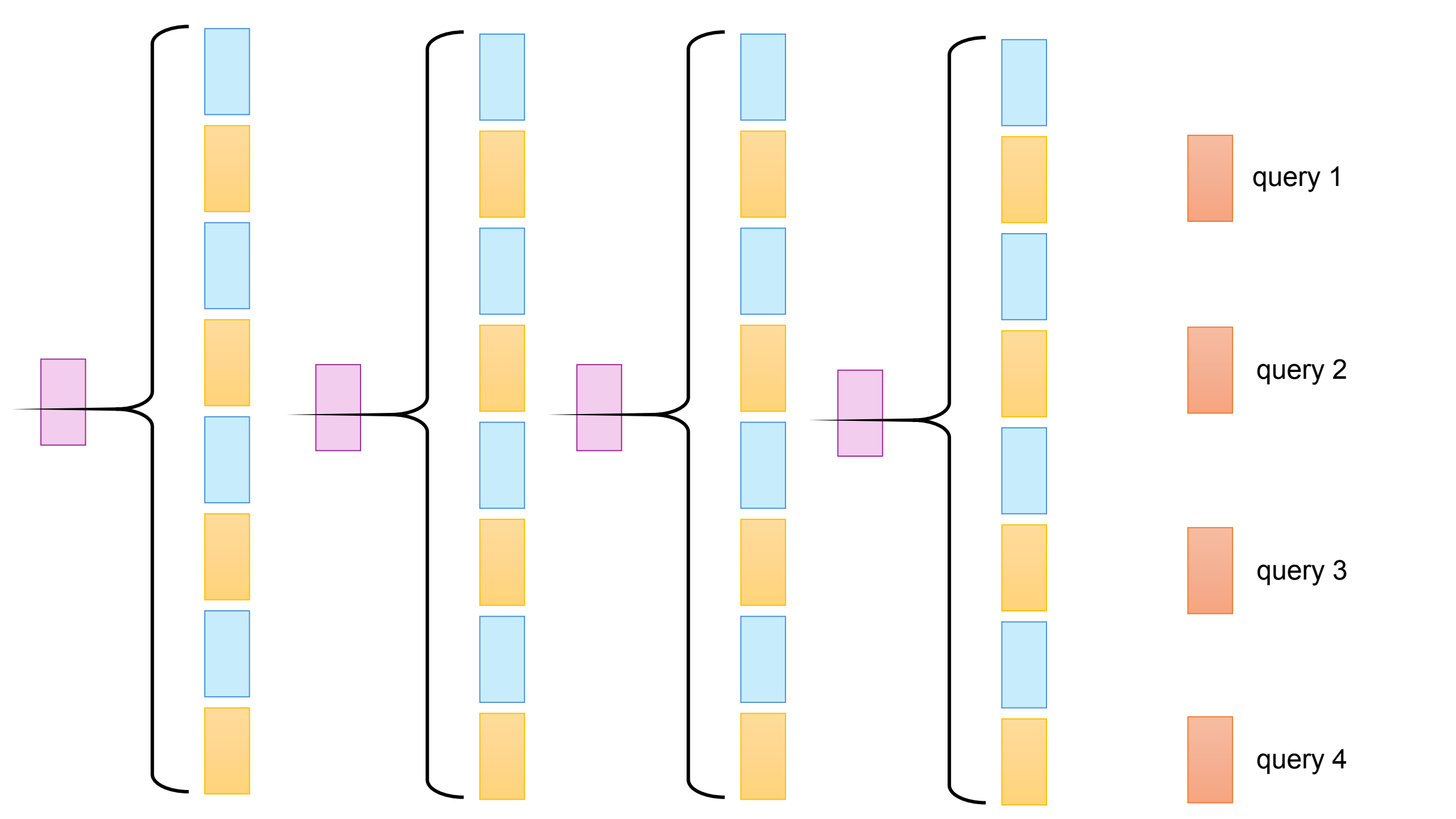
https://arxiv.org/abs/2405.04434

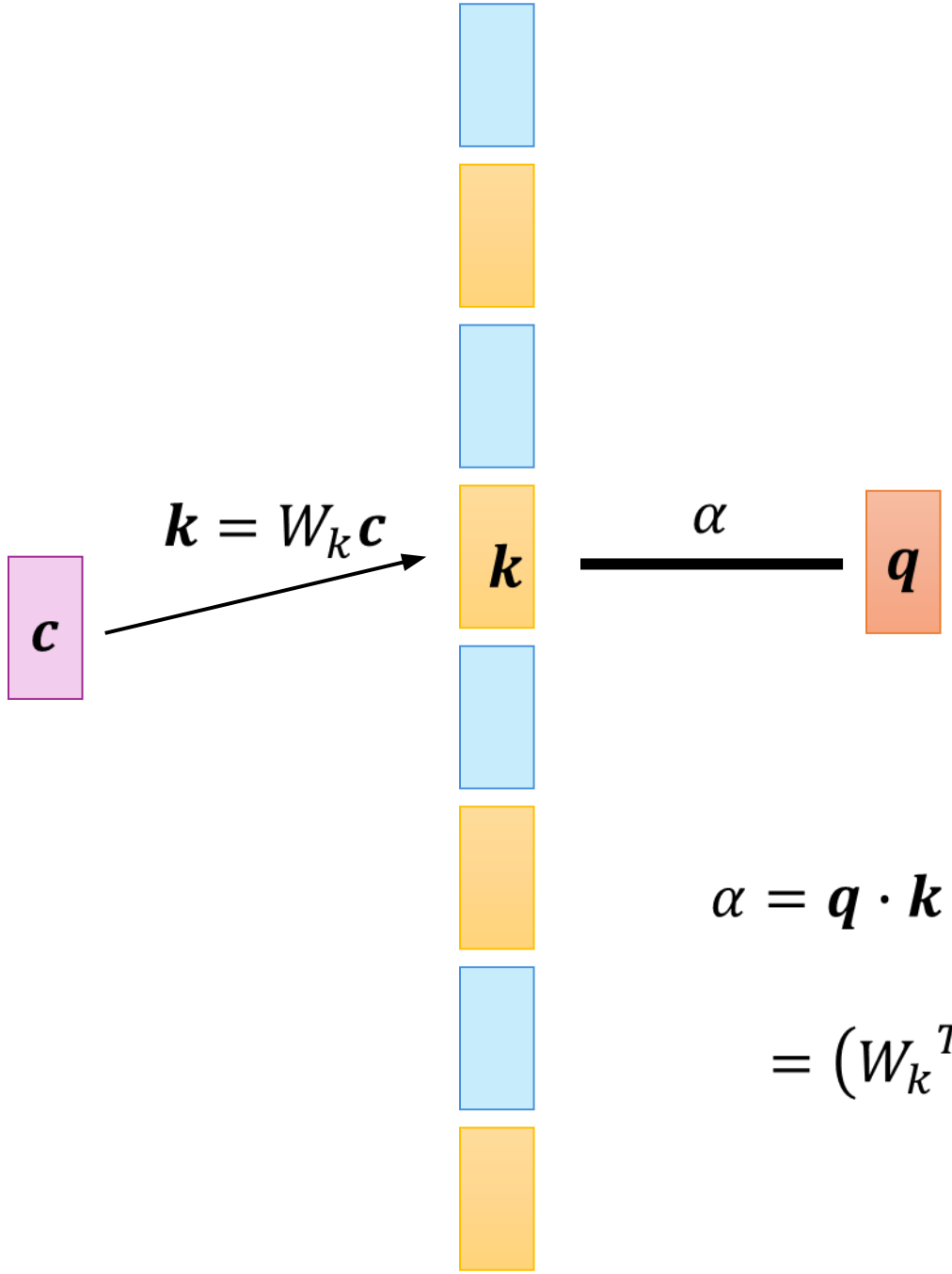
Multi-head Latent Attention

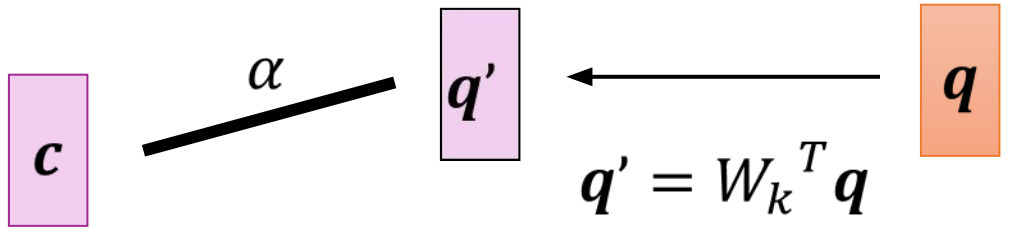


DeepSeek

https://arxiv.org/abs/2405.04434

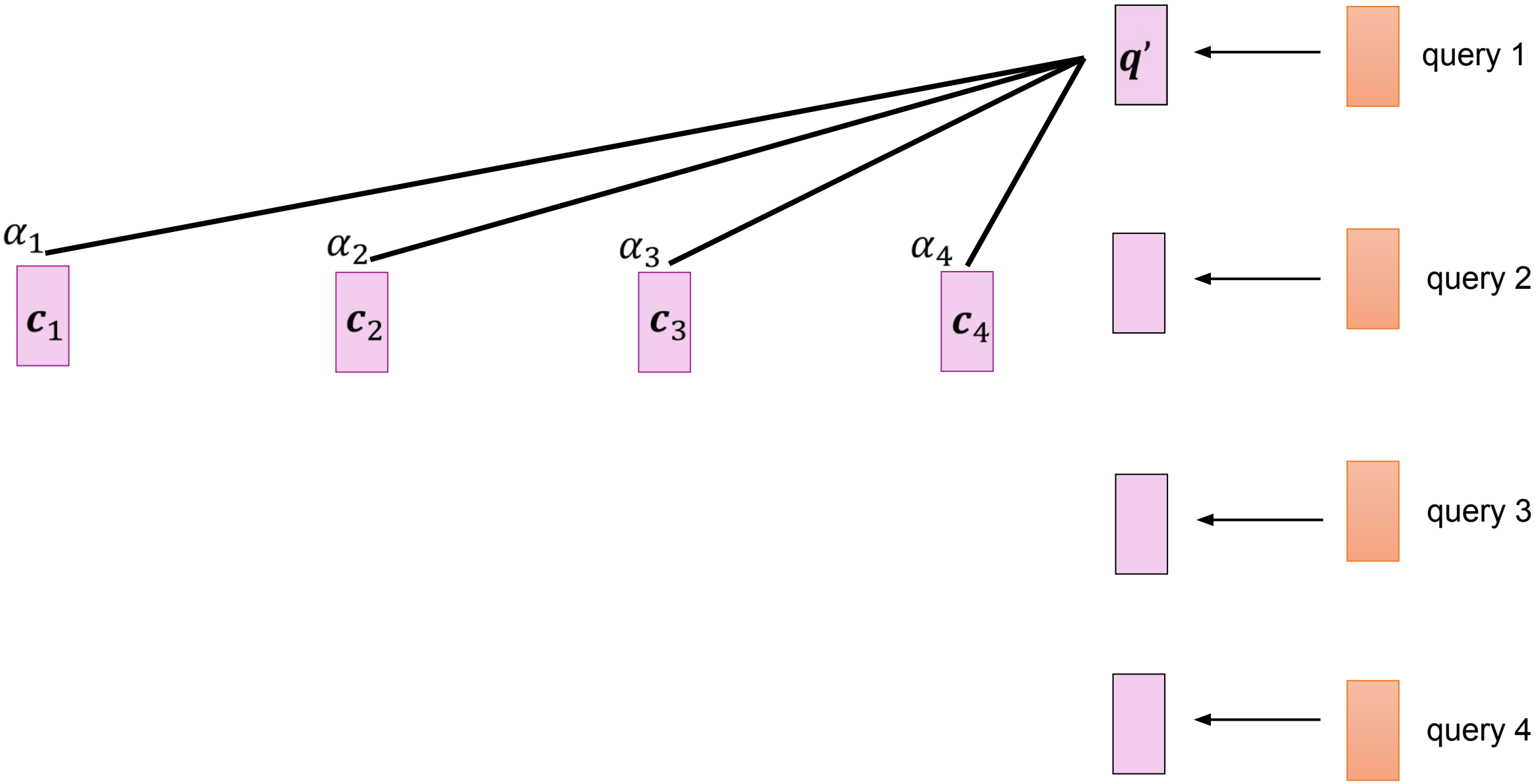

query 1
query 2
query 3
query 4

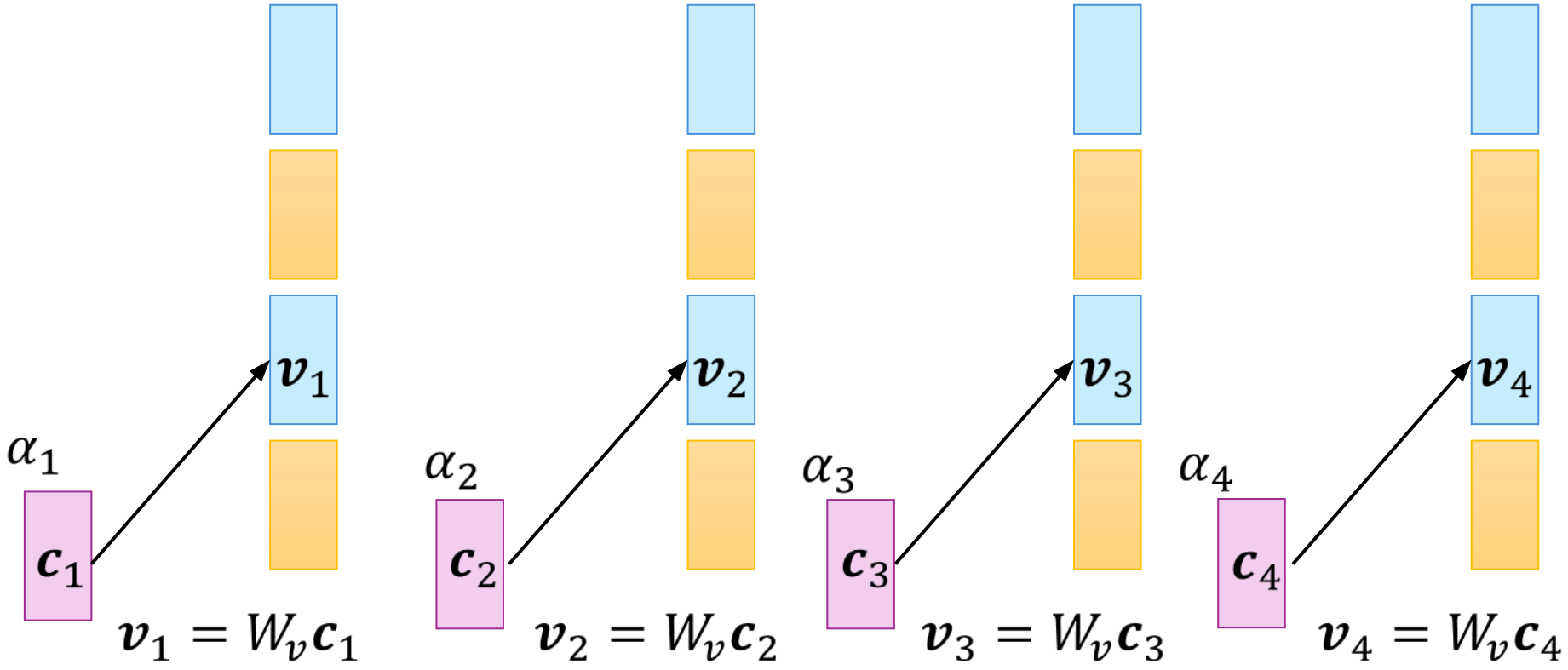




$$\alpha = \boldsymbol{q} \cdot \boldsymbol{k} = \boldsymbol{q}^T \boldsymbol{k} = \boldsymbol{q}^T W_k \boldsymbol{c}$$

$$= \left(W_k{}^T \boldsymbol{q}\right)^T \boldsymbol{c} = \left(W_k{}^T \boldsymbol{q}\right) \cdot \boldsymbol{c}$$


q'
query 1
query 2
query 3
query 4
α_1
α_2
α_3
α_4
c_1
c_2
c_3
c_4

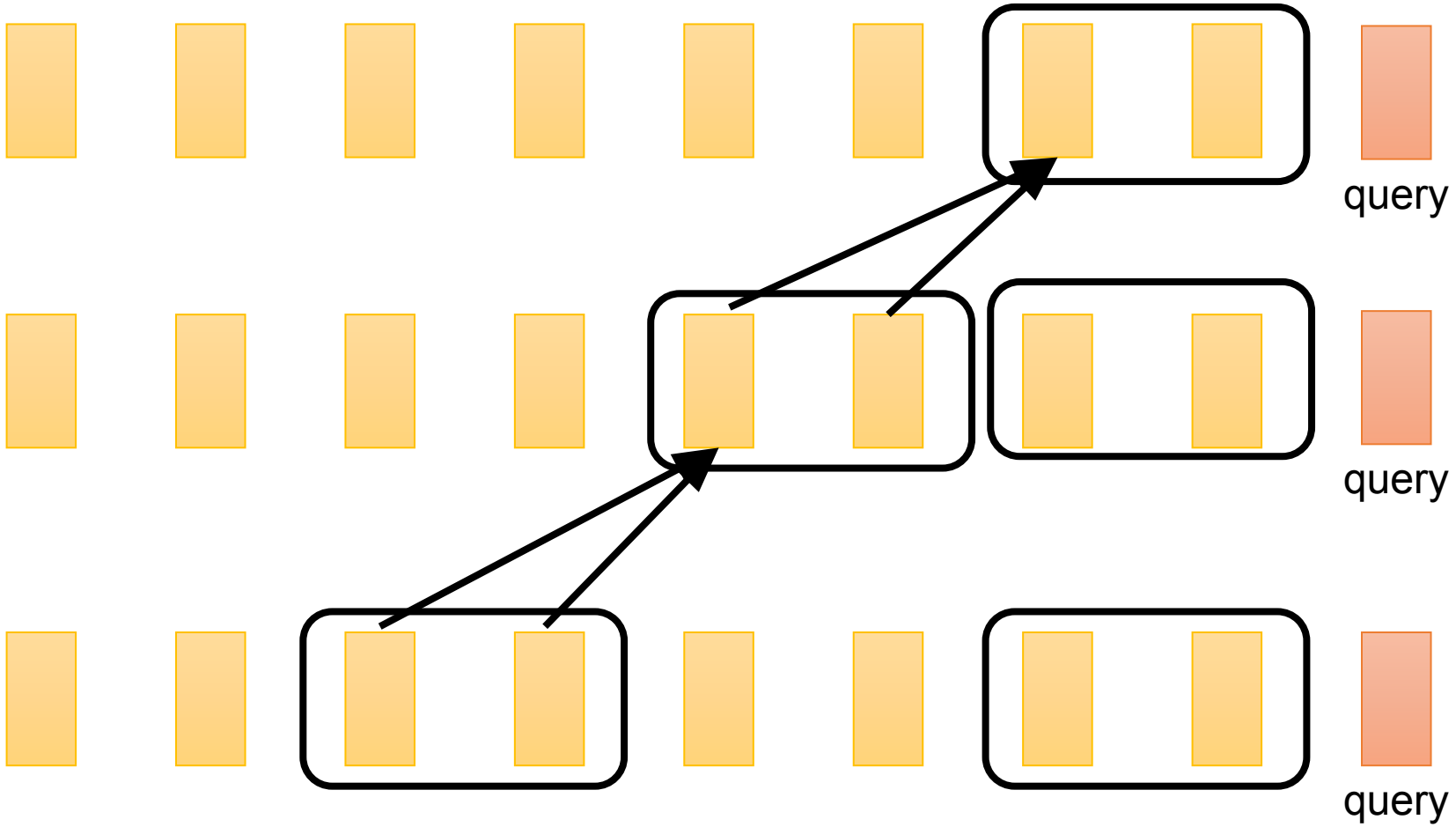


$$\boldsymbol{o} = \alpha_1\boldsymbol{v}_1 + \alpha_2\boldsymbol{v}_2 + \alpha_3\boldsymbol{v}_3 + \alpha_4\boldsymbol{v}_4$$

$$= \alpha_1 W_v\boldsymbol{c}_1 + \alpha_2 W_v\boldsymbol{c}_2 + \alpha_3 W_v\boldsymbol{c}_3 + \alpha_4 W_v\boldsymbol{c}_4$$

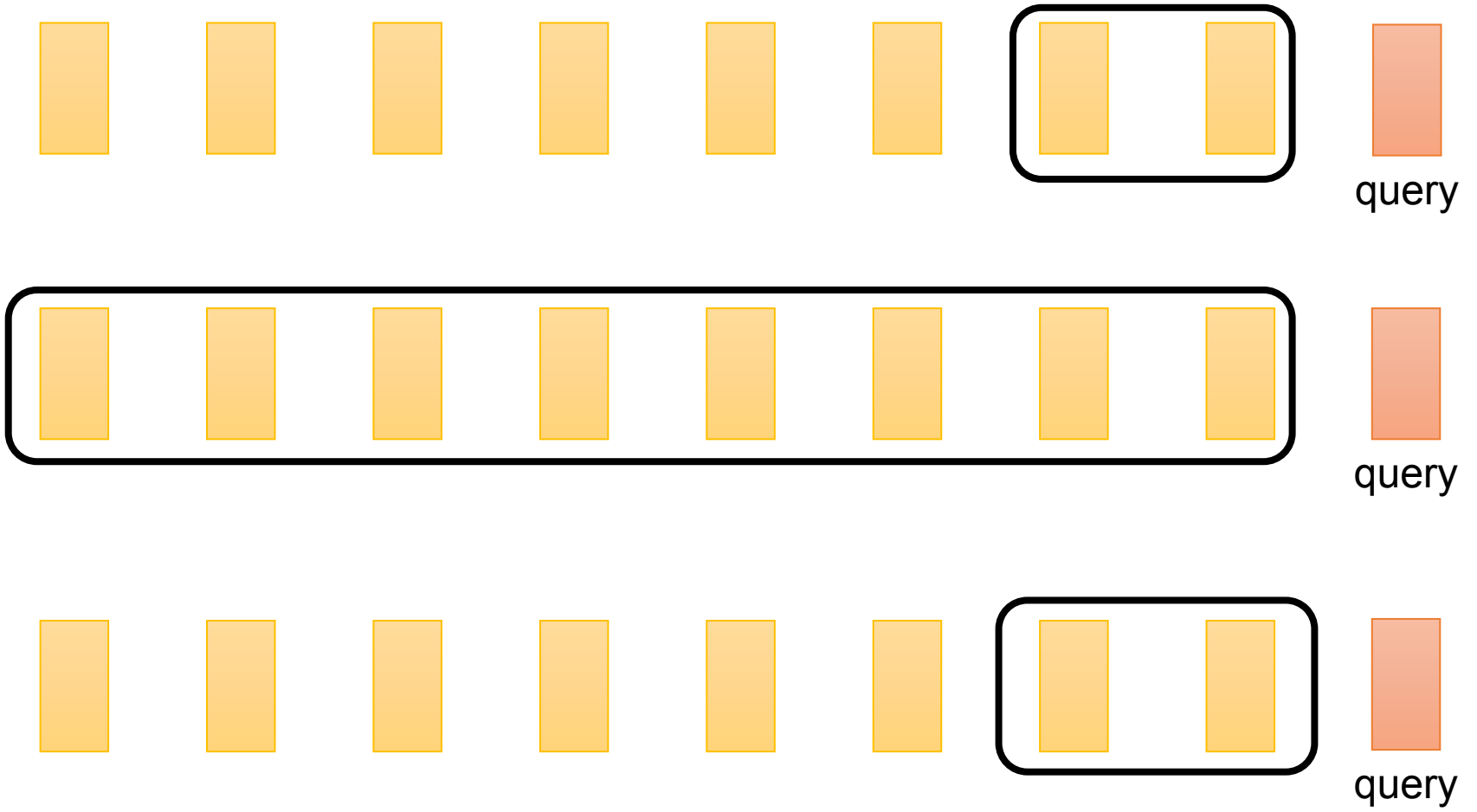
$$= W_v(\alpha_1\boldsymbol{c}_1 + \alpha_2\boldsymbol{c}_2 + \alpha_3\boldsymbol{c}_3 + \alpha_4\,\boldsymbol{c}_4)$$

Sliding Window Attention



Mistral 7B

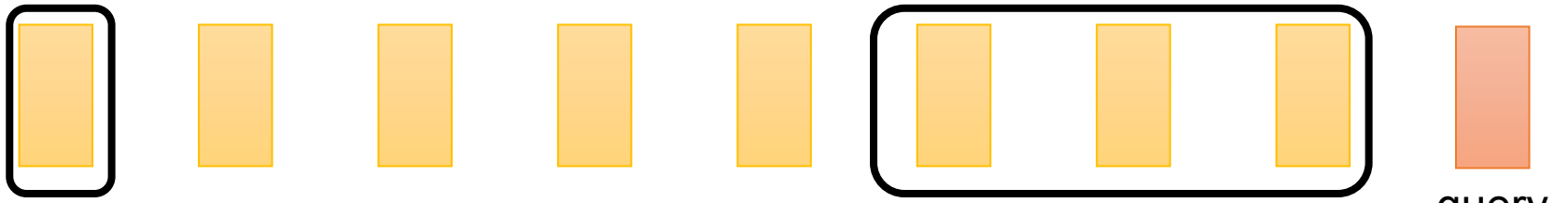
Sliding Window Attention

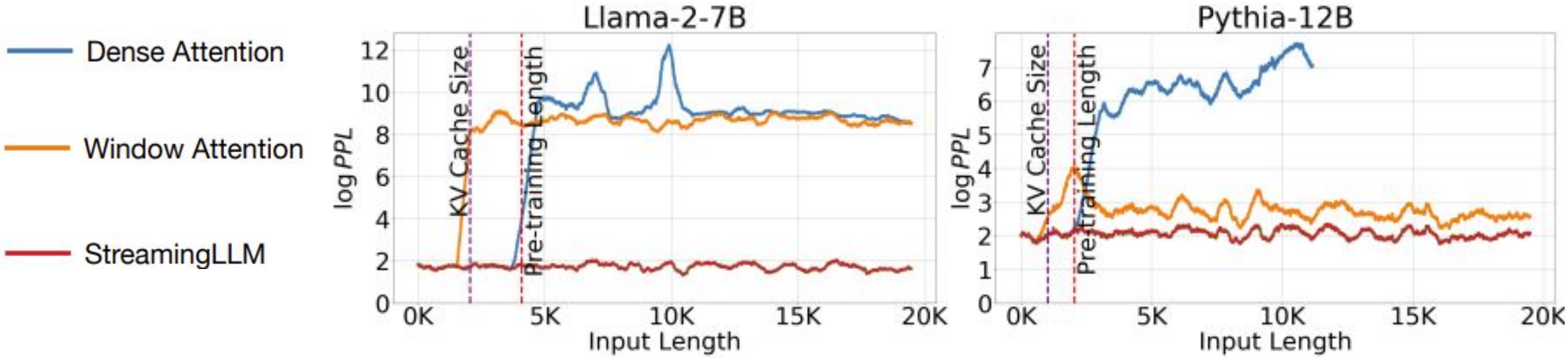


GPT-OSS

Streaming LLM

https://arxiv.org/abs/2309.17453



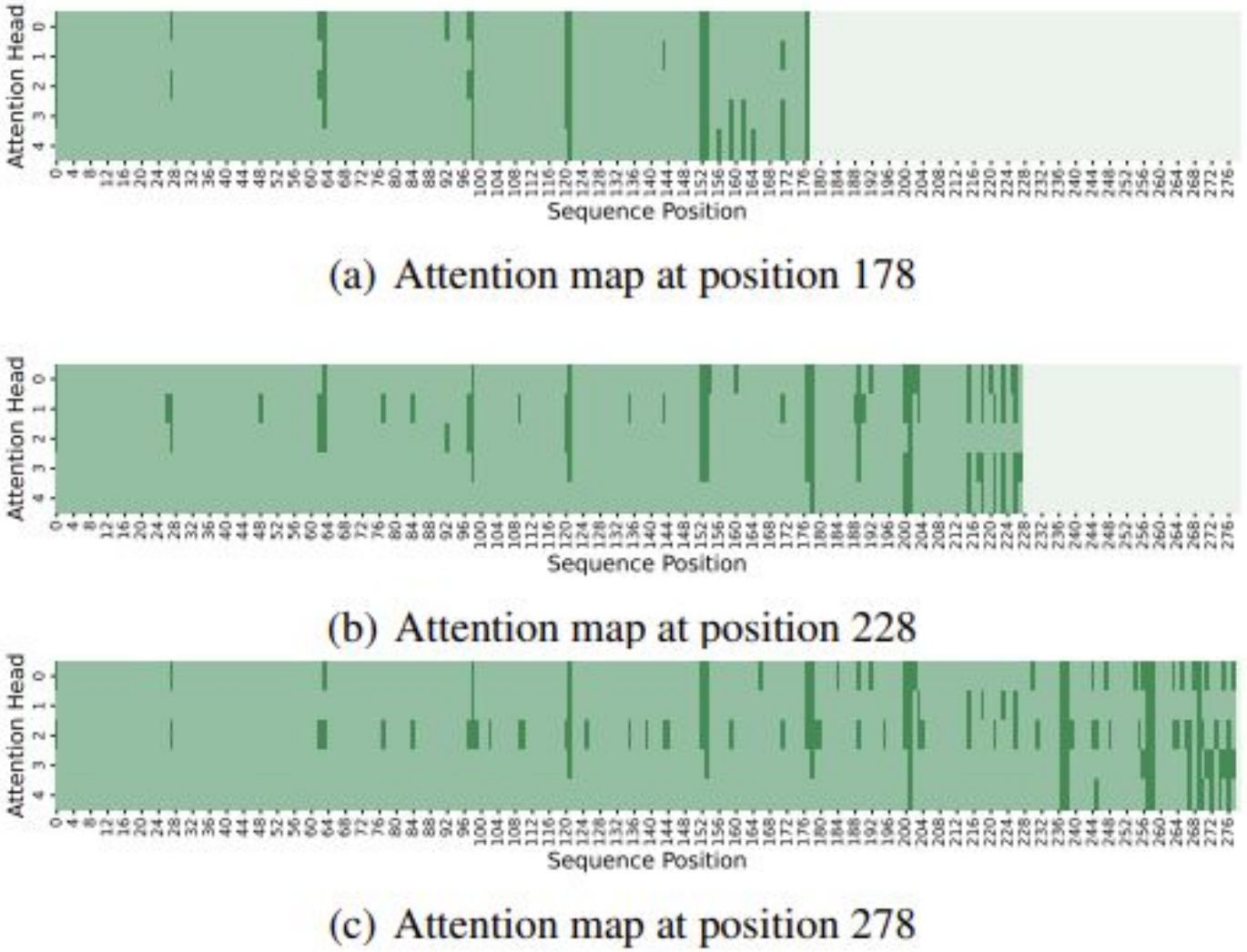


Pruning KV Cache

Scissorhands: https://arxiv.org/abs/2305.17118

H2O: https://arxiv.org/abs/2306.14048

- 每次只有小部分的 Tokens 有 Attention
- 少數 token 會反覆吸走大量 attention

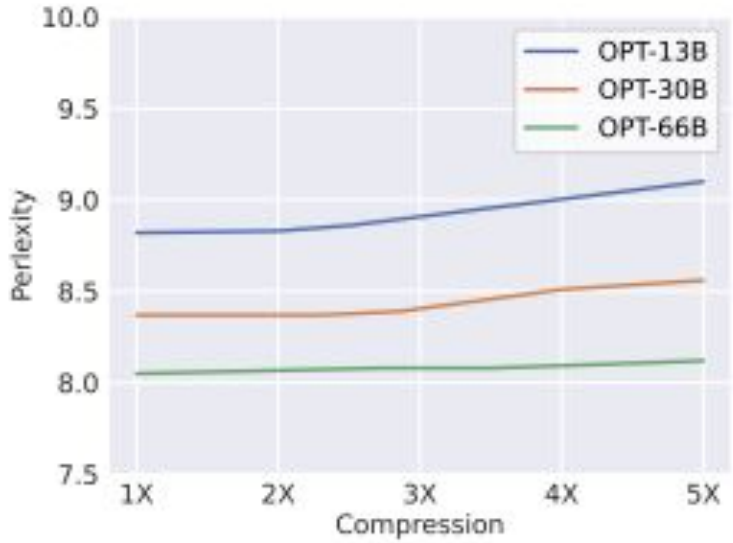


(a) Attention map at position 178

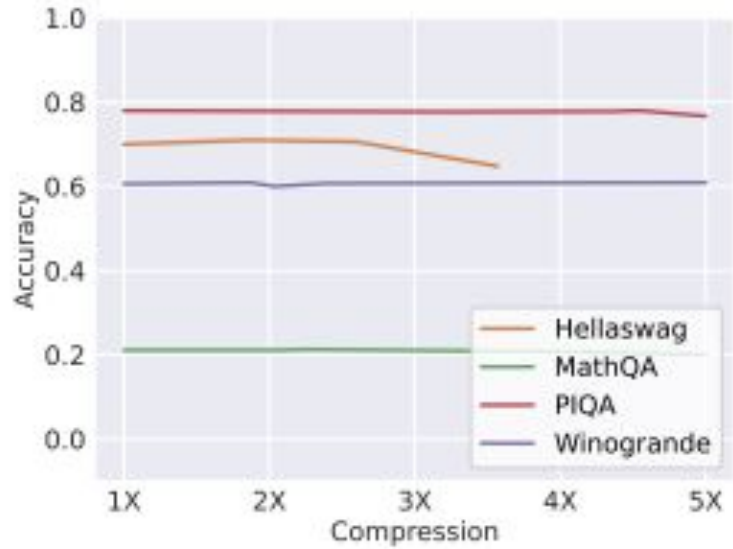
(b) Attention map at position 228

(c) Attention map at position 278

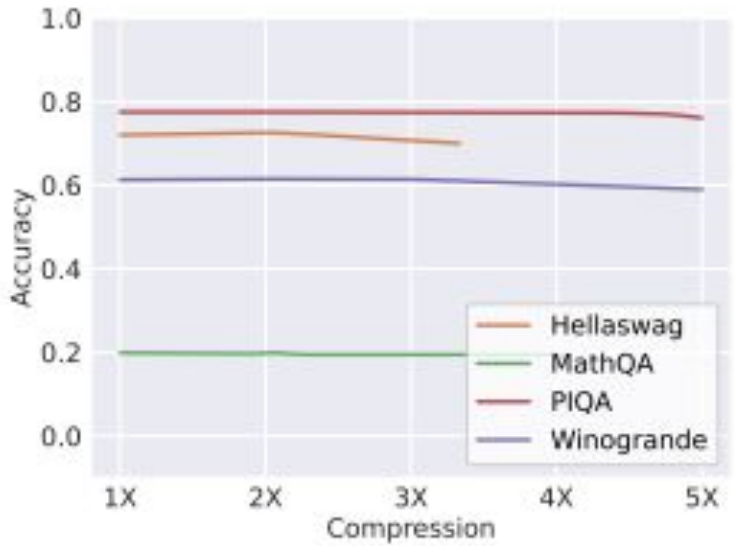
Scissorhands: https://arxiv.org/abs/2305.17118



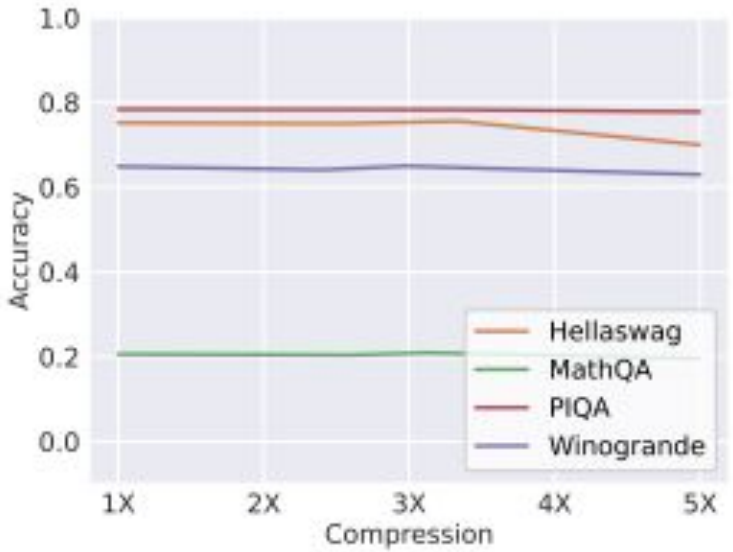
(a) Language Modeling



(b) OPT-6B Five shot

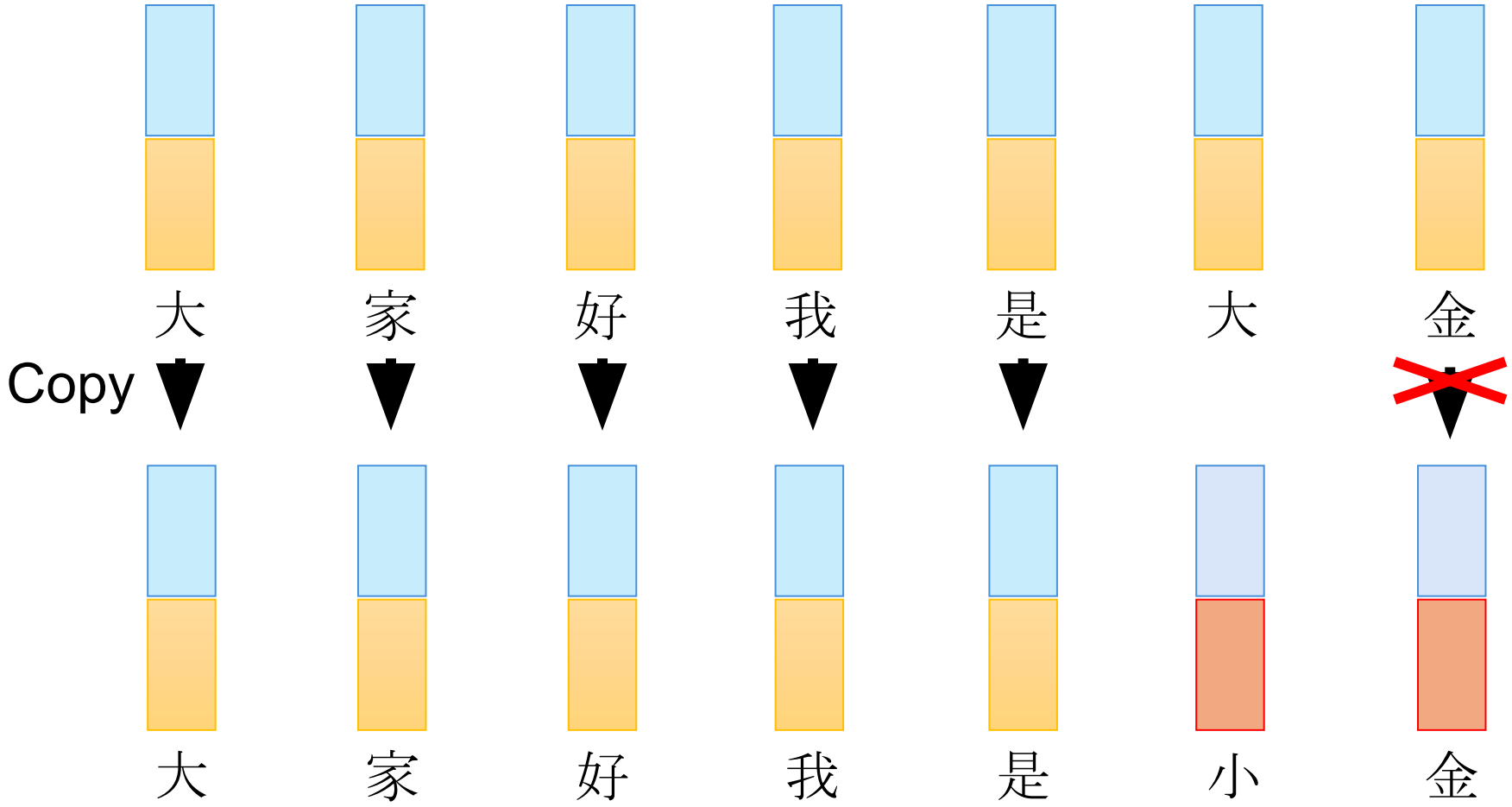


(c) OPT-13B Five shot



(d) OPT-30B Five shot

跨對話的 Cache



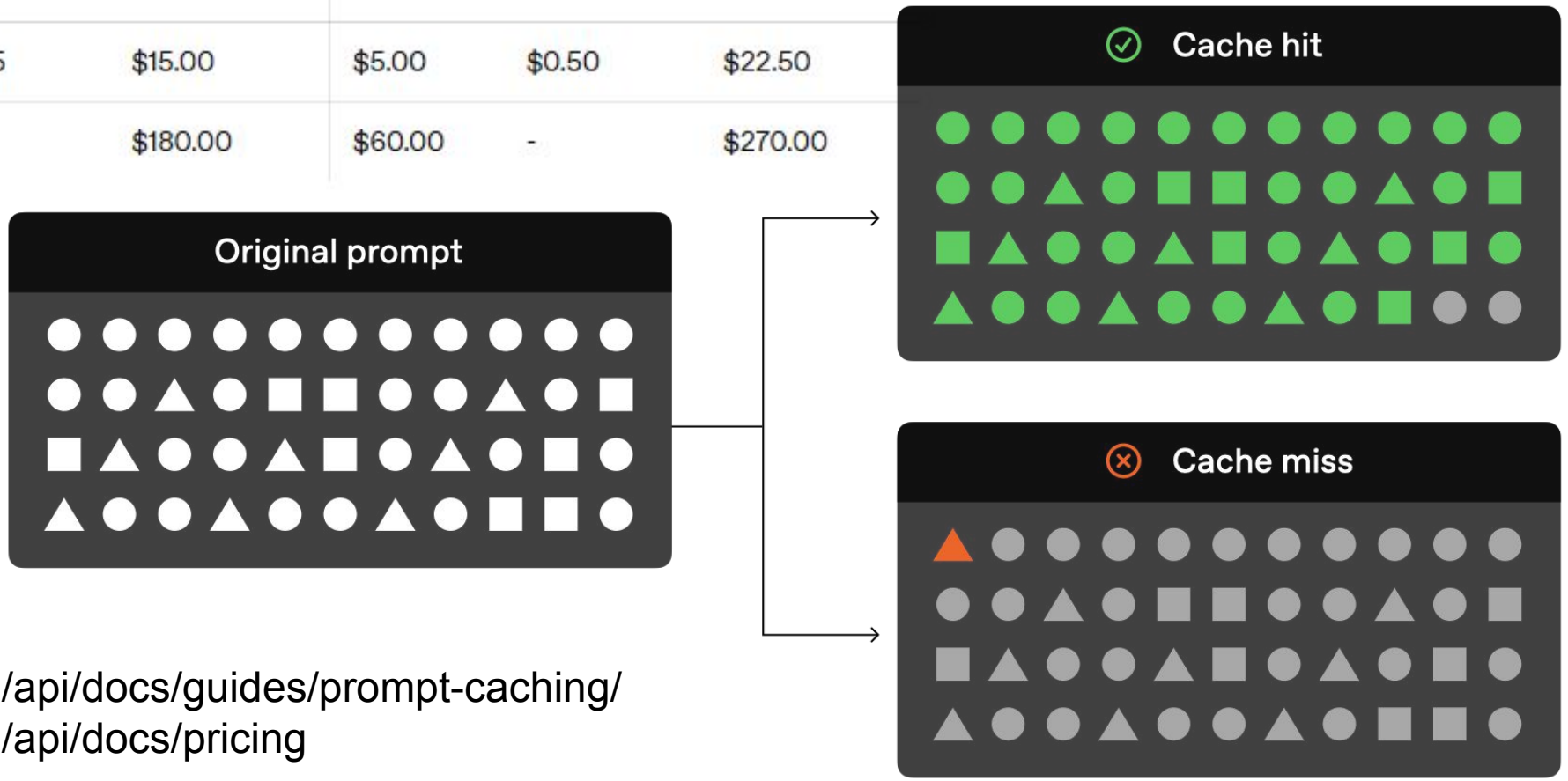
Text tokens

Prices per 1M tokens.

Standard | Batch | Flex | Priority

Our latest models

	Short context			Long context		
Model	Input	Cached input	Output	Input	Cached input	Output
gpt-5.4	$2.50	$0.25	$15.00	$5.00	$0.50	$22.50
gpt-5.4-pro	$30.00	-	$180.00	$60.00	-	$270.00



https://developers.openai.com/api/docs/guides/prompt-caching/
https://developers.openai.com/api/docs/pricing

甚麼時候跨對話的 Cache 可以發揮作用

我是小金 …

自我介紹

我是小金 …

我是小金 …
人生目標 ….

自我介紹

語言
模型

System prompt

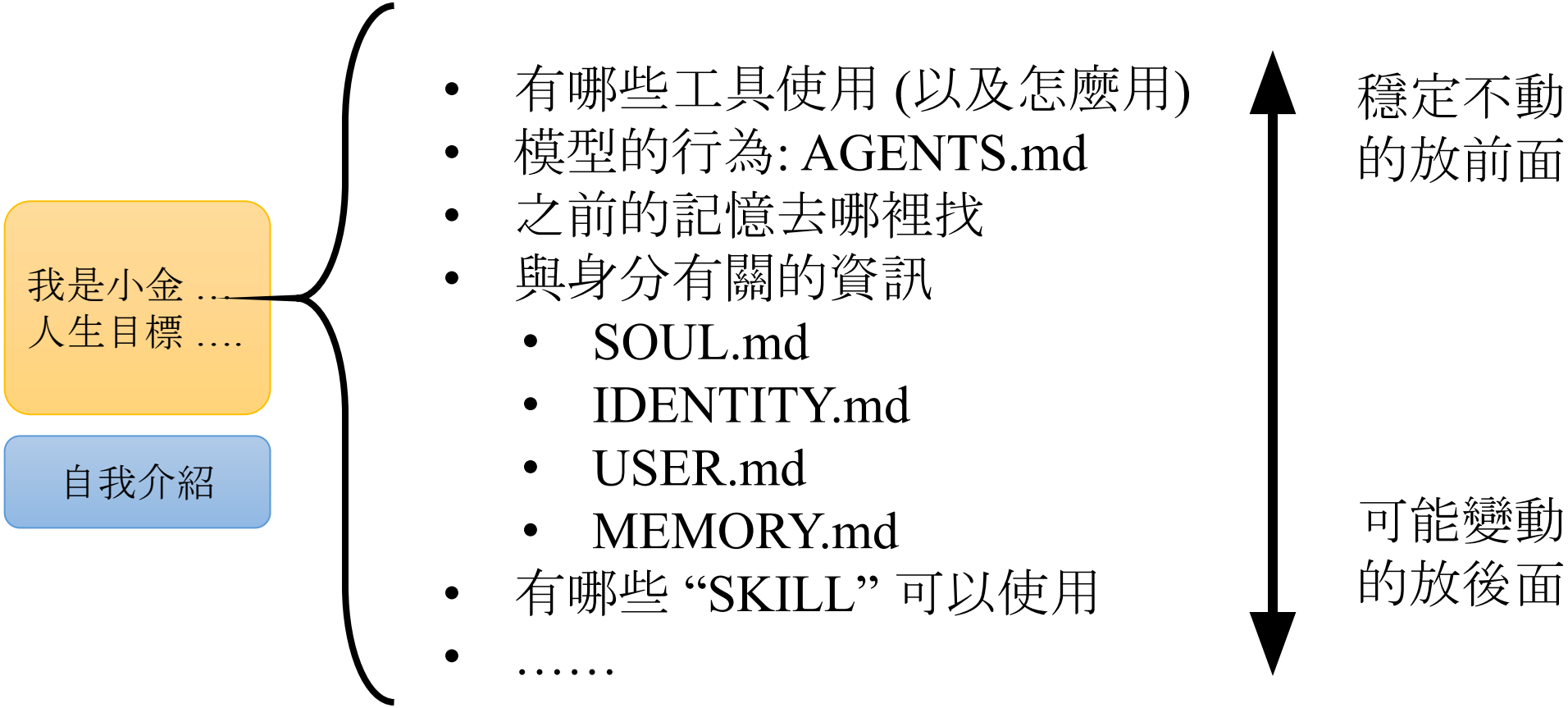
每次呼叫語言模型的 prompt 都會附上

Cache Hit!

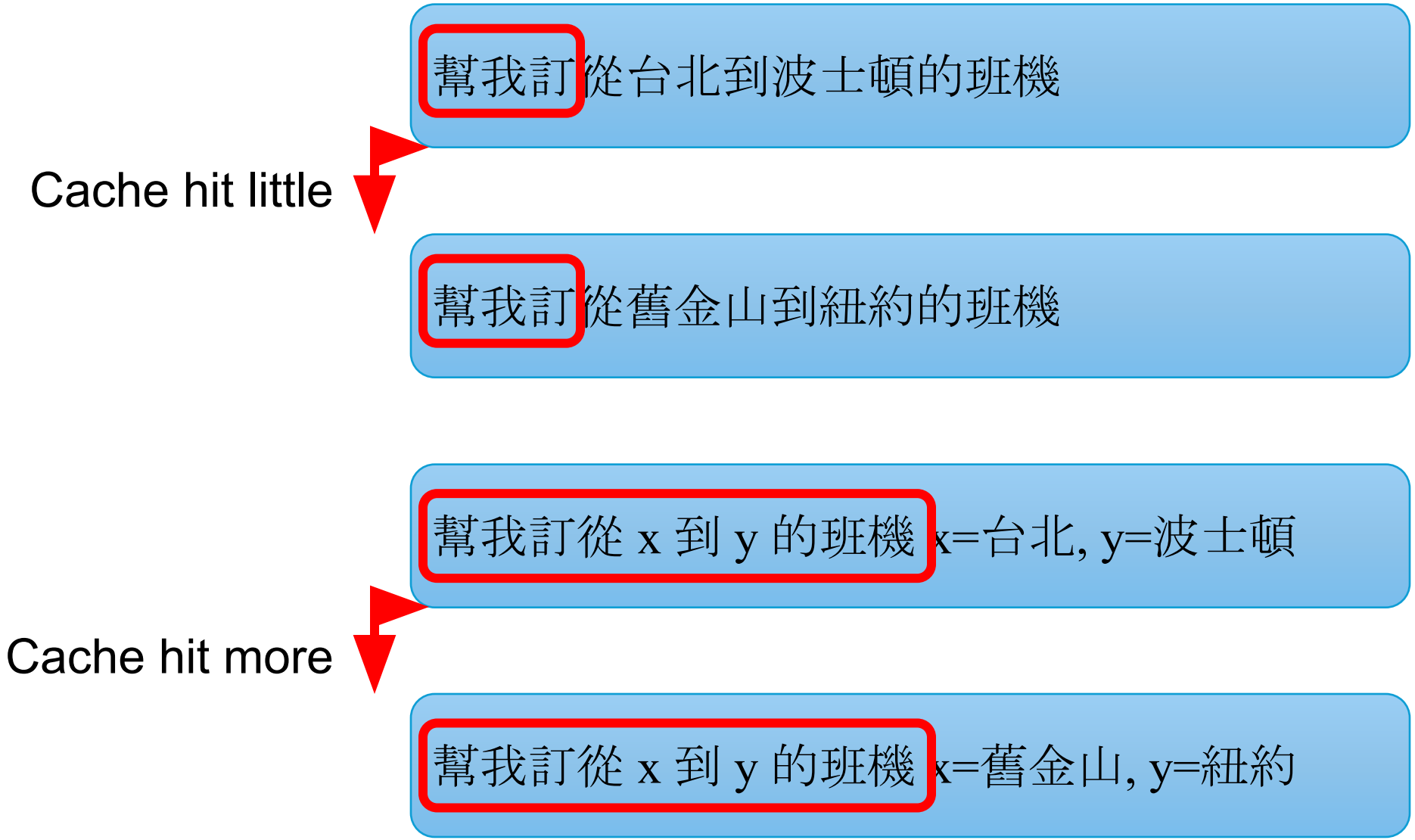
使用 AI Agent 的時候!

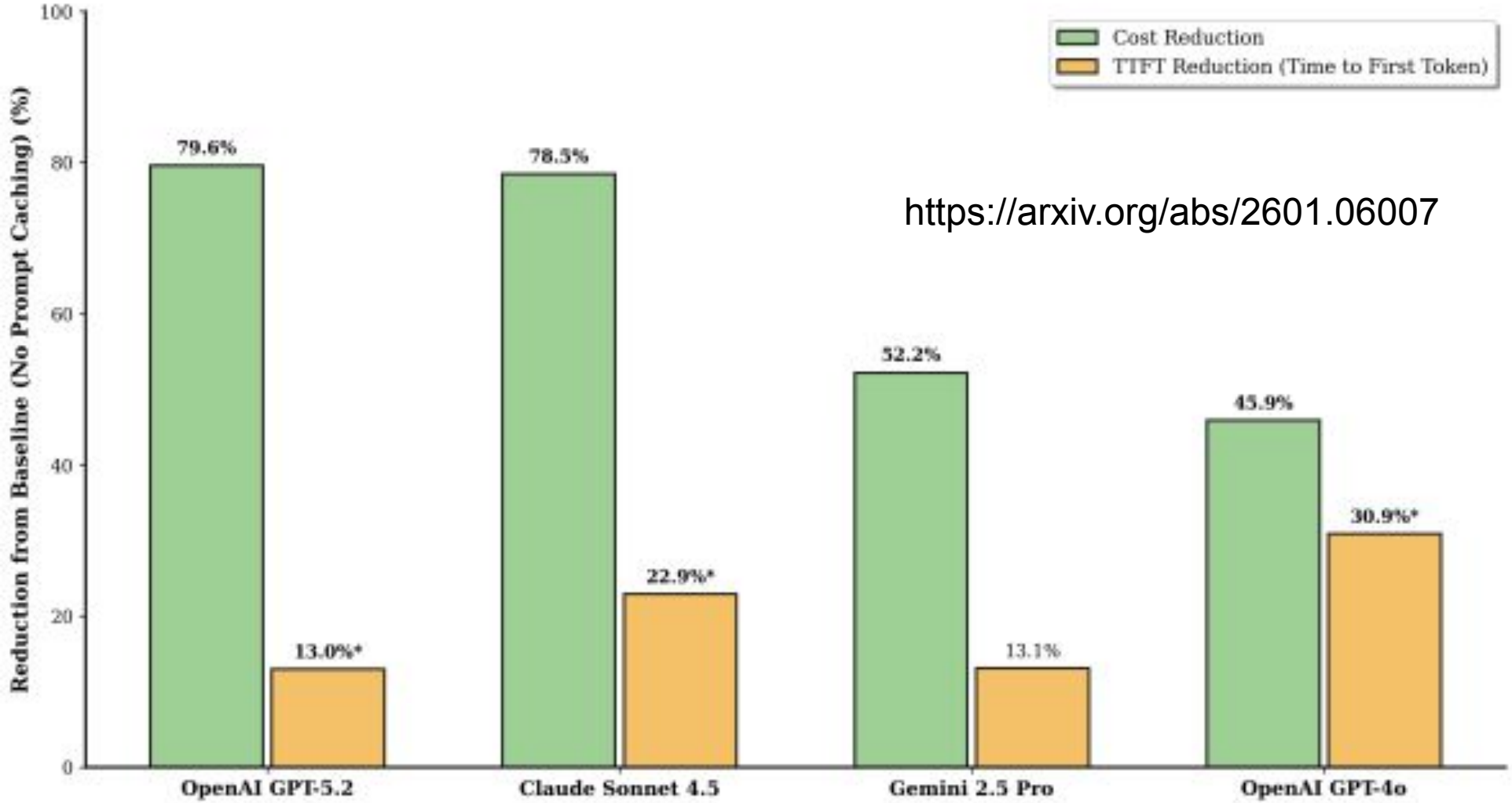
自身相關資訊的文字檔

甚麼時候跨對話的 Cache 可以發揮作用



https://github.com/openclaw/openclaw/issues/27732


幫我訂從台北到波士頓的班機
Cache hit little
幫我訂從舊金山到紐約的班機
幫我訂從 x 到 y 的班機 x=台北, y=波士頓
Cache hit more
幫我訂從 x 到 y 的班機 x=舊金山, y=紐約



https://arxiv.org/abs/2601.06007

結論

	方法	是否改變原有的 Attention	是否需要訓練模型	其他代價
Flash Attention	少搬資料	X	X	一點額外運算+一點點燒腦
KV Cache	儲存已經算出來的 key 和 value	X	X	占用記憶體
Multi-query attention	多個 query 共享 key 和 value	O	O	可能明顯傷害模型能力
Group-query attention		O	O	
Multi-head Latent Attention	壓縮 key 和 value	O	O	
Sliding Window Attention	改變 Attention 範圍	O	?	
Streaming LLM		O	?	
Pruning KV Cache	丟棄 key 和 value	O	X	可能明顯傷害模型能力
Speculative Decoding	用小模型來預言生成結果	X (理論上)	X	小模型還是需要耗費額外算力