Efficient LLM Decoding

Large Language Models: Introduction and Recent Advances

ELL881 · AIL821

Yatin Nandwani
Research Scientist, IBM Research

Training Vs Inference in LLMs

Forward Pass through an LLM

Transformer based LLM (ϑ)

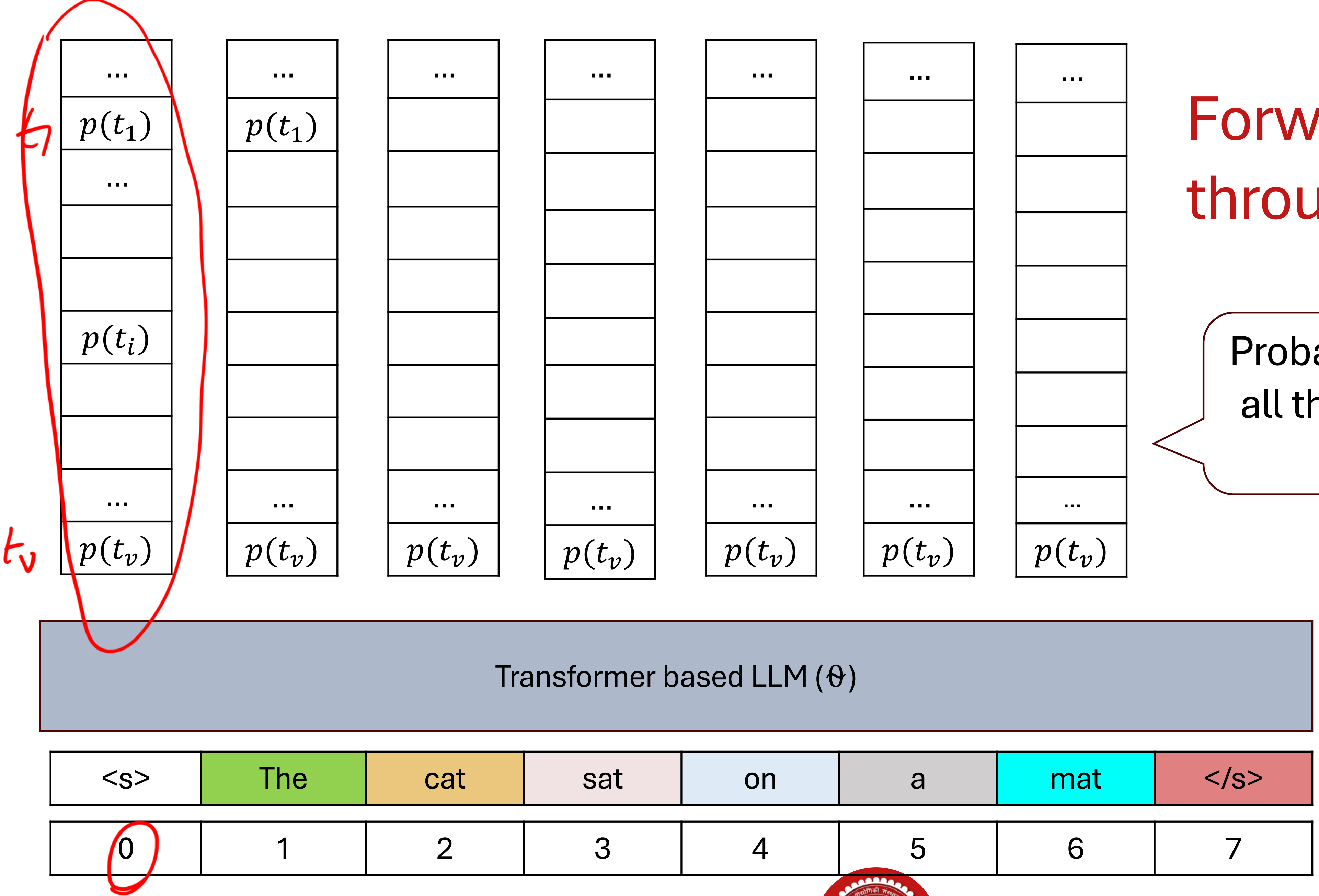
<s>	The	cat	sat	on	a	mat	</s>
0	1	2	3	4	5	6	7

Forward Pass through an LLM

Probability distribution over all the tokens at each step (simultaneously)

Transformer based LLM (ϑ)

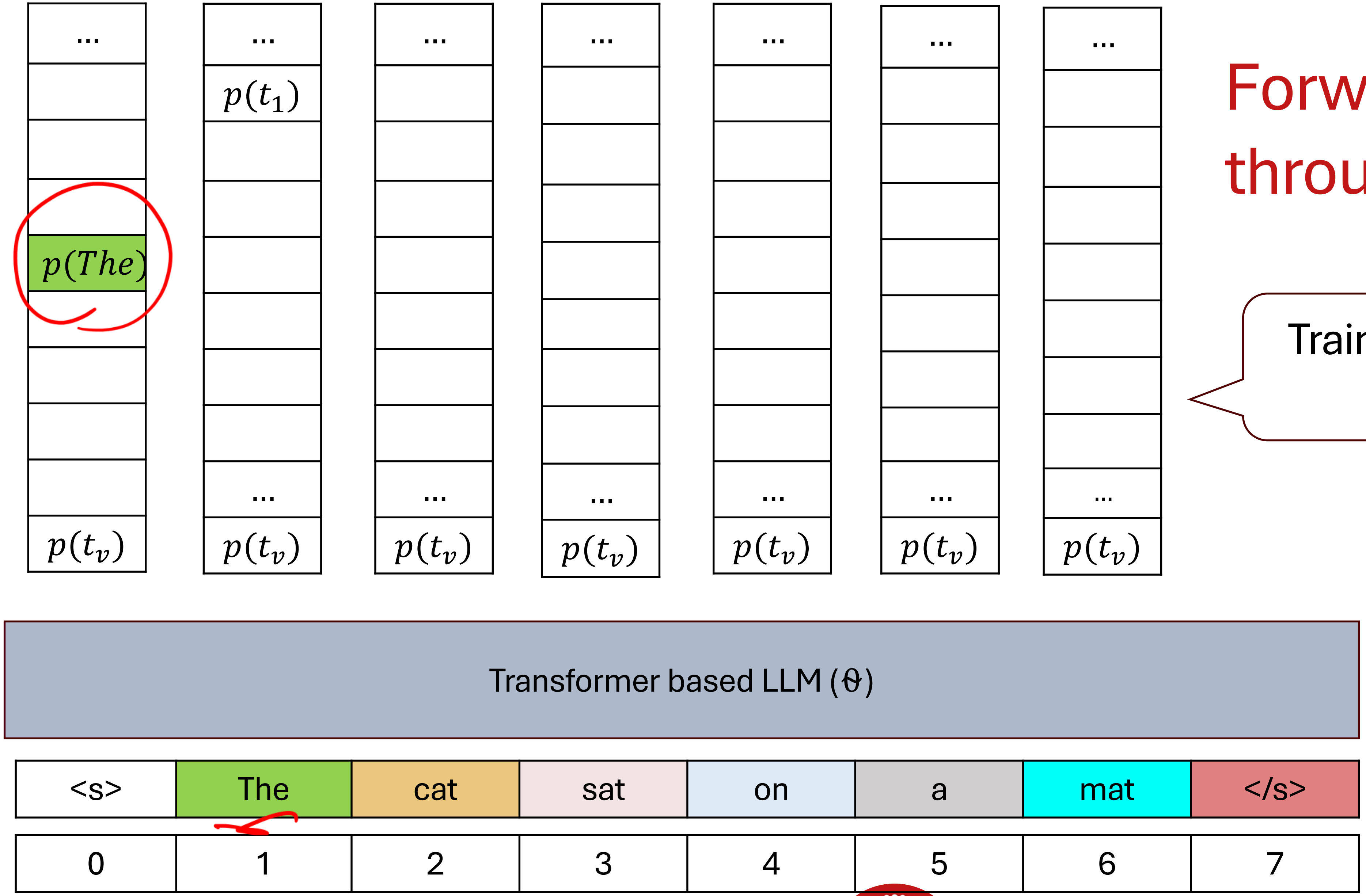
<s>	The	cat	sat	on	a	mat	</s>
0	1	2	3	4	5	6	7



Forward Pass through an LLM

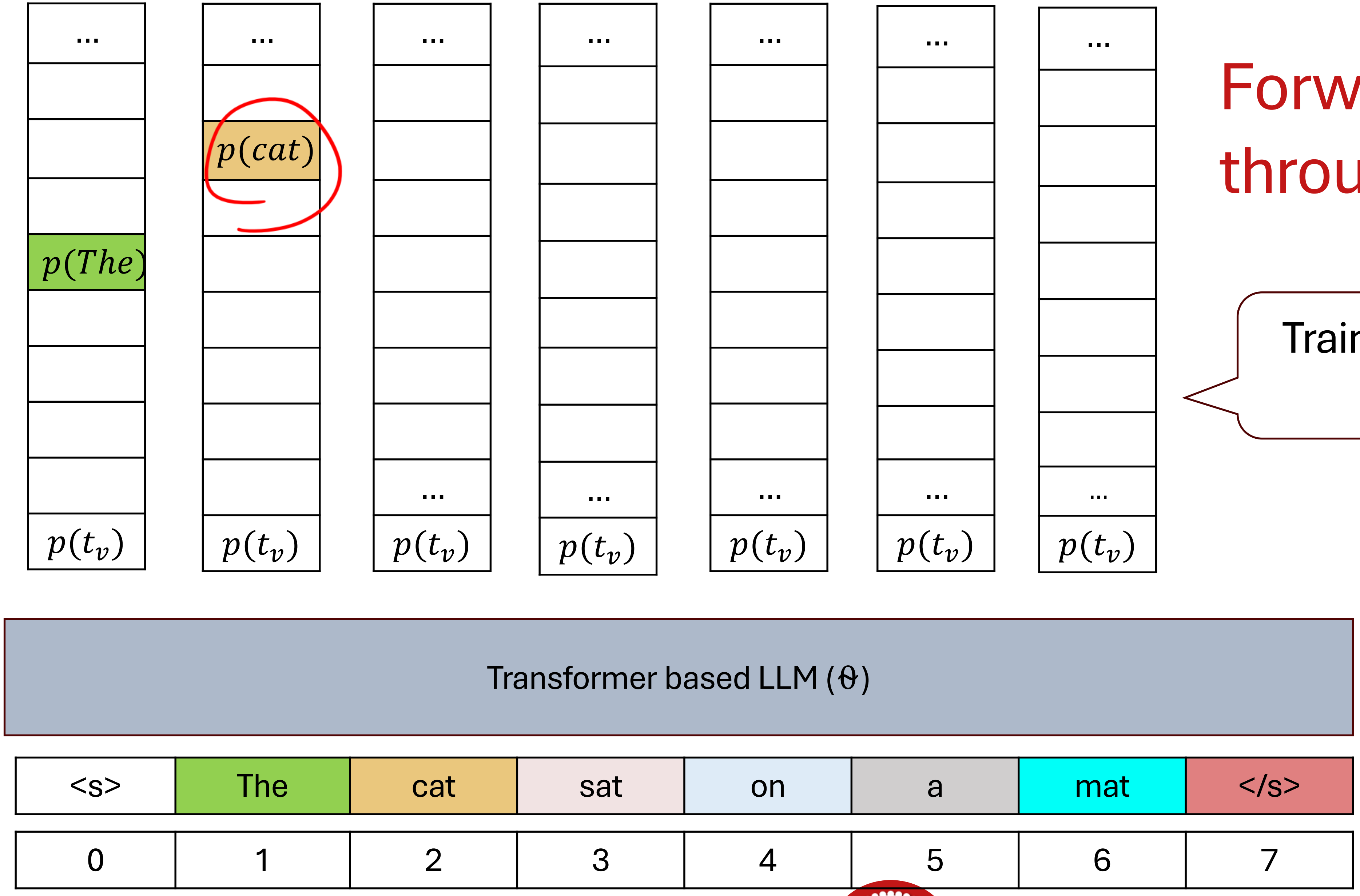
Probability distribution over all the tokens at each step (simultaneously)

Forward Pass through an LLM



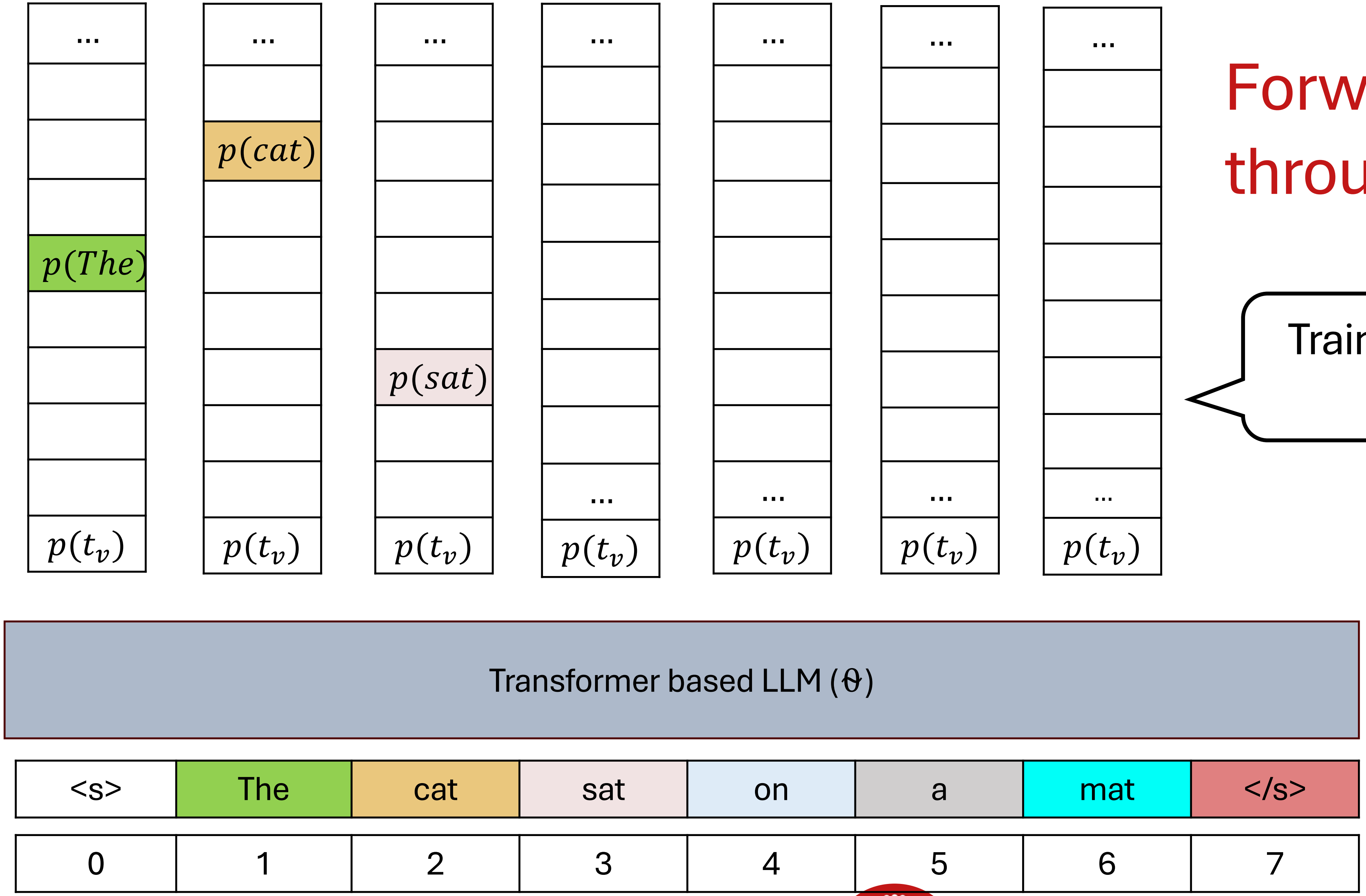
Train to maximize prob. of *The* at step 0

Forward Pass through an LLM



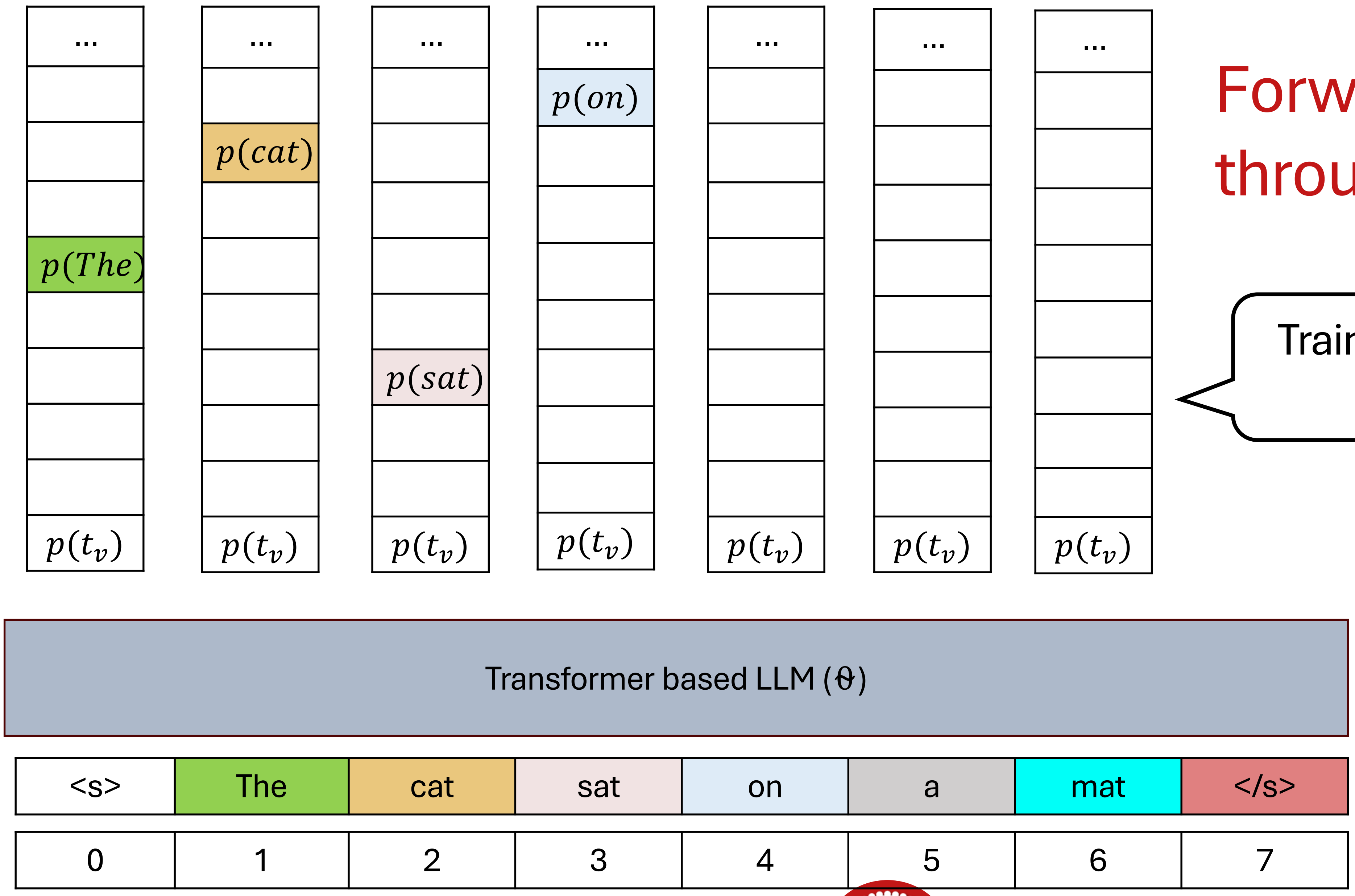
Train to maximize prob. of *cat* at step 1

Forward Pass through an LLM



Train to maximize prob. of *sat* at step 2

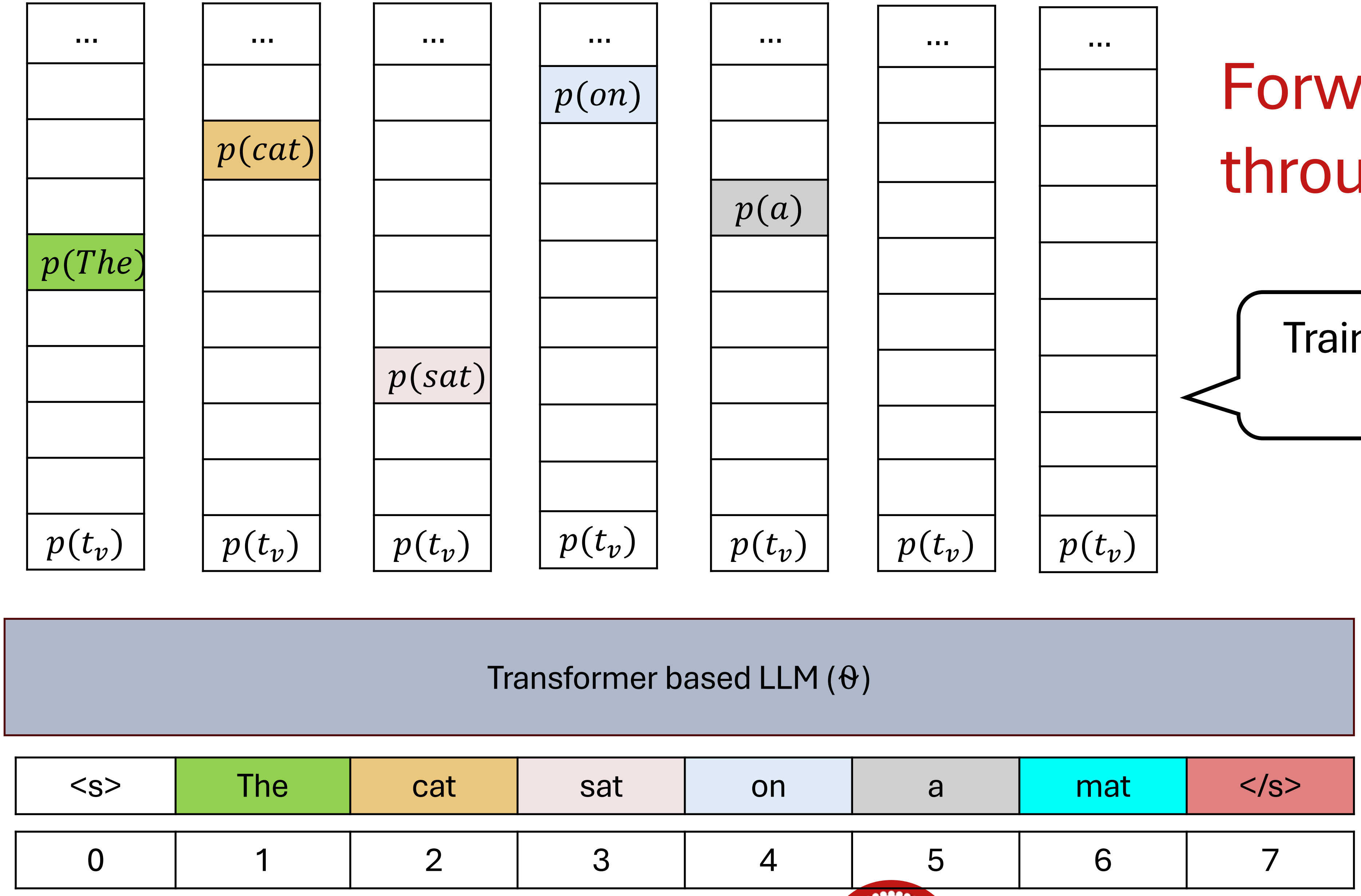
Forward Pass through an LLM



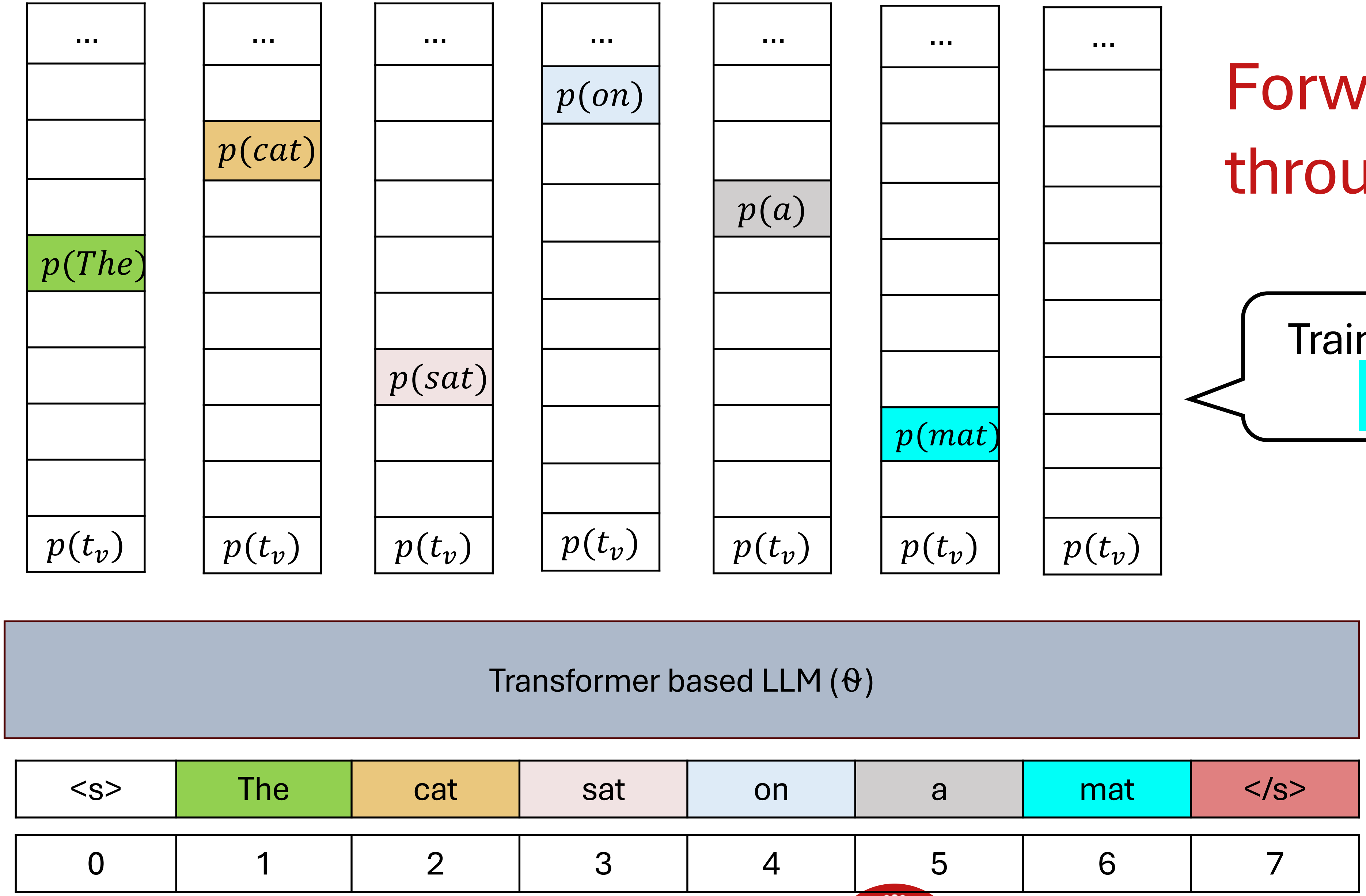
Train to maximize prob. of *on* at step 3

Forward Pass through an LLM

Train to maximize prob. of a at step 4

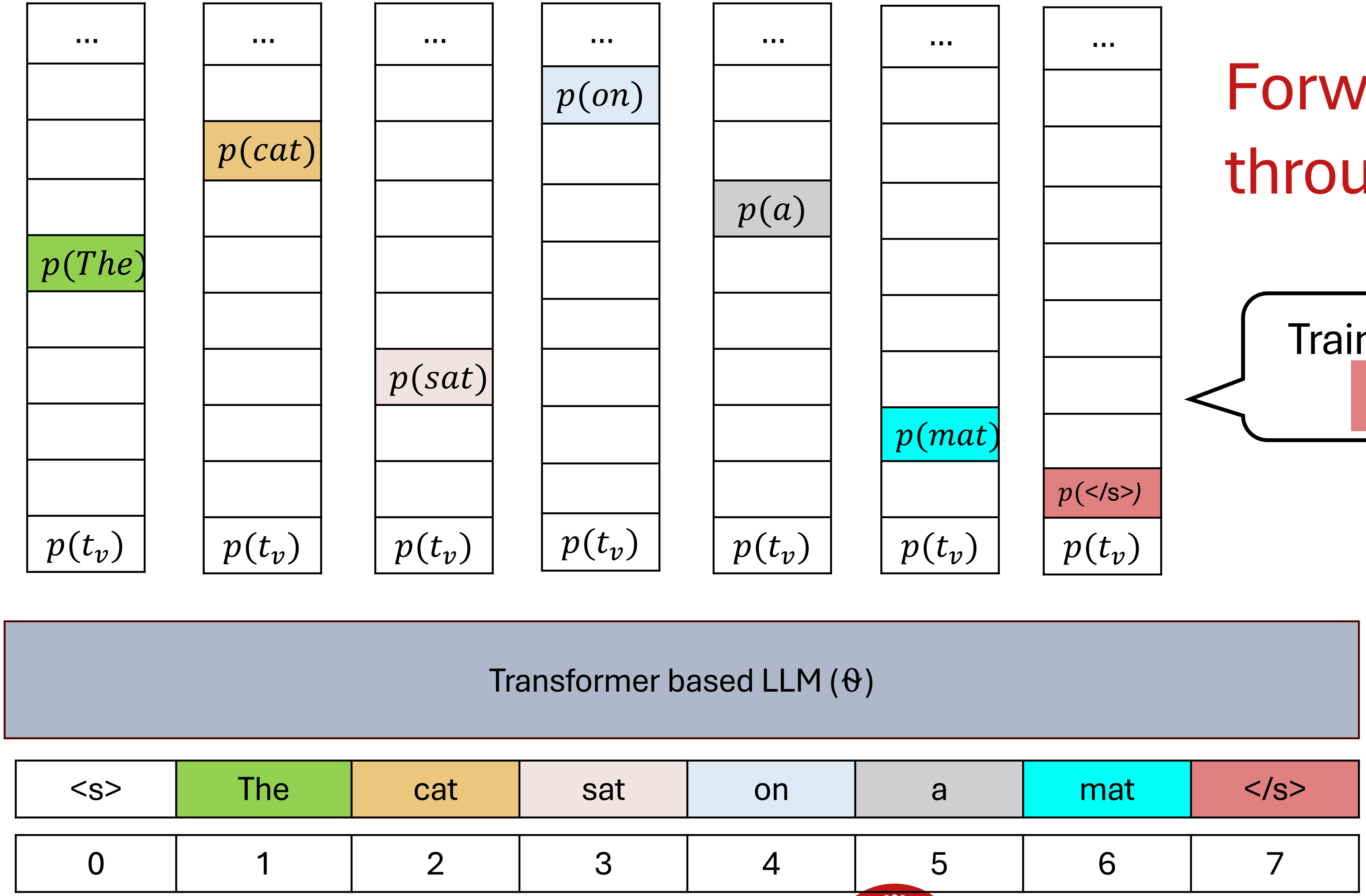


Forward Pass through an LLM



Train to maximize prob. of *mat* at step 5

Forward Pass through an LLM



Train to maximize prob. of </s> at step 6

Inference through an LLM

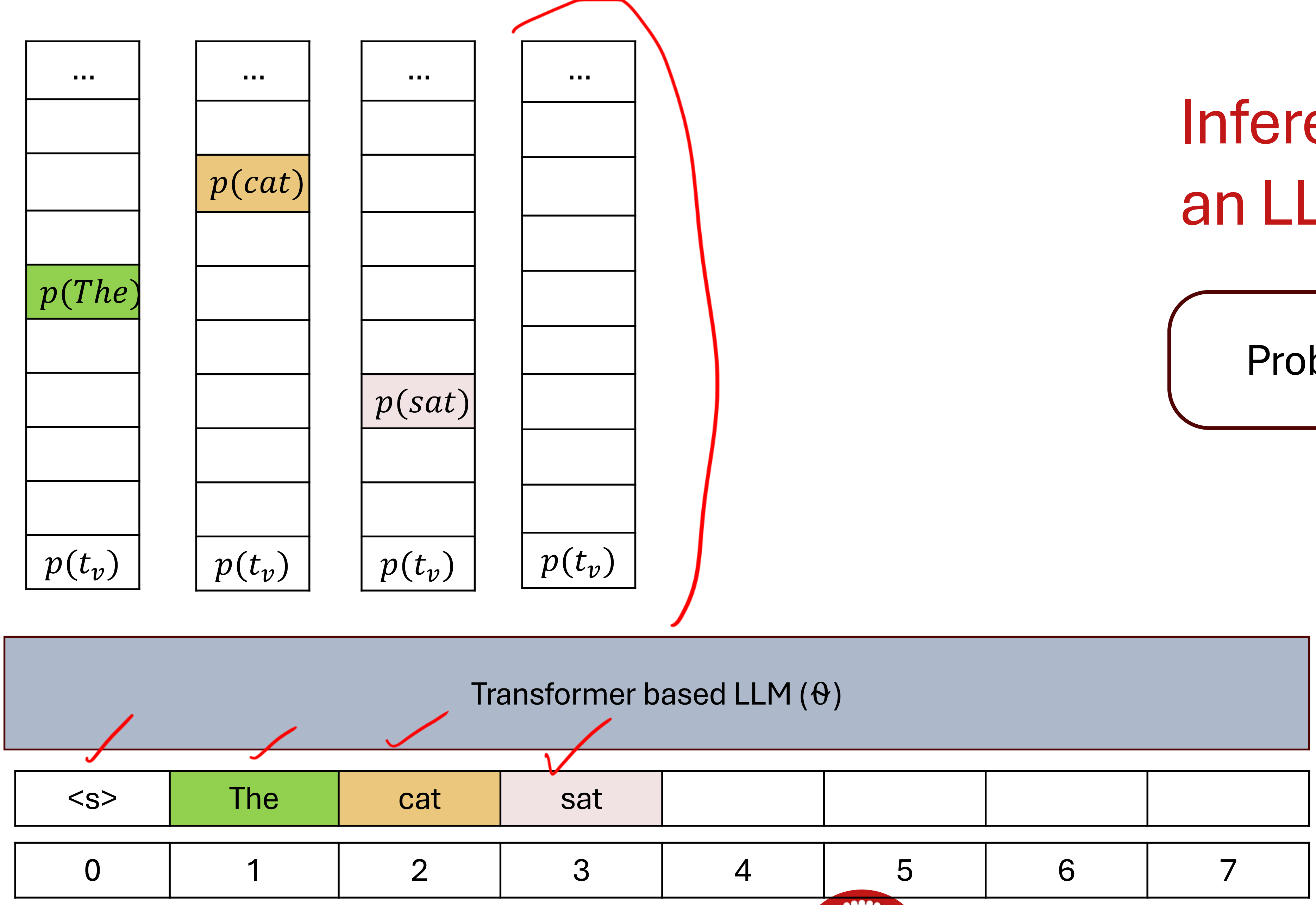
Forward Pass (#1)

Transformer based LLM (ϑ)

<s>	The	cat	sat				
0	1	2	3	4	5	6	7

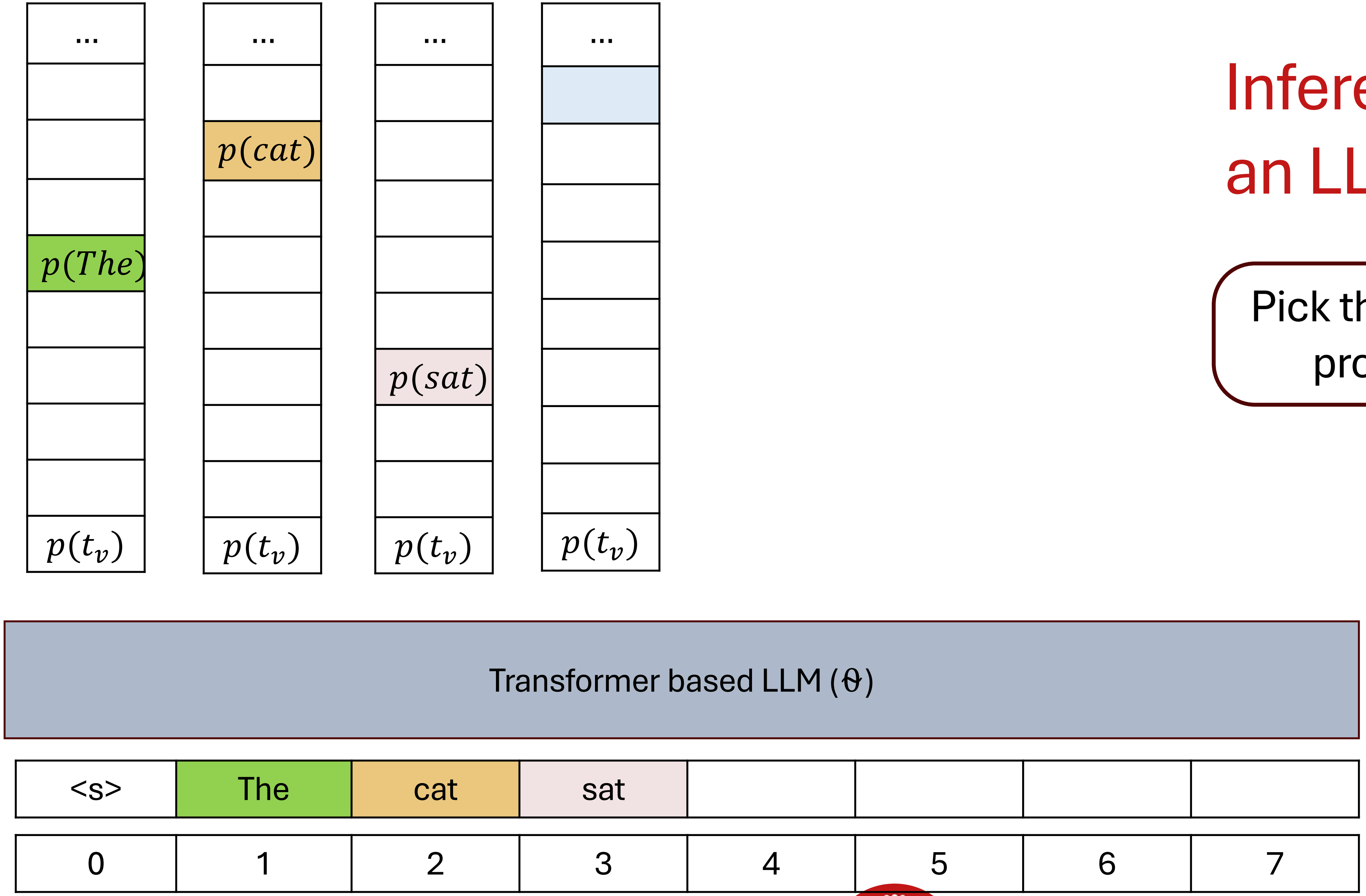
Inference through an LLM

Prob. Dist. at all steps



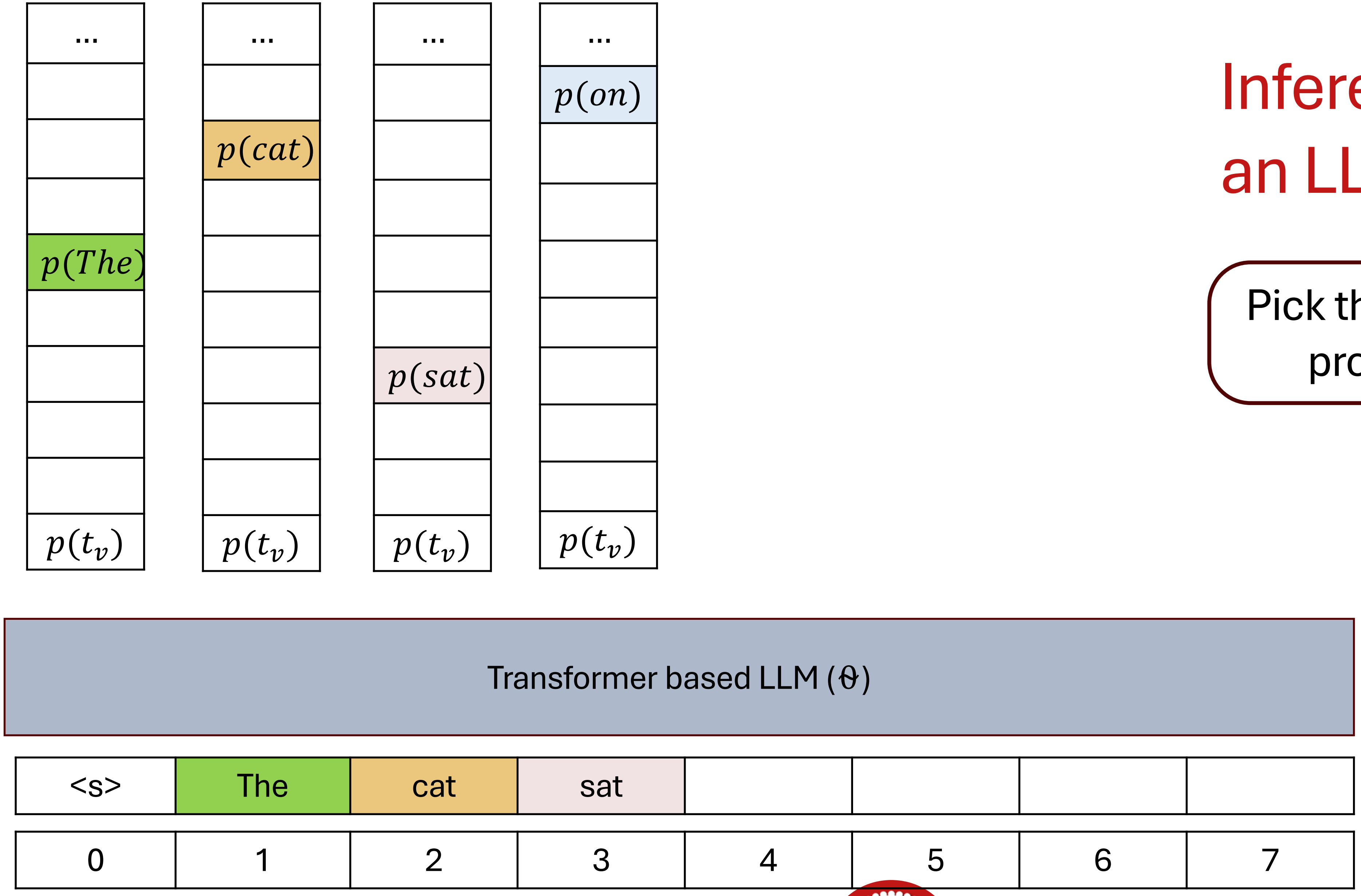
Inference through an LLM

Pick the token having max. probability at step 3



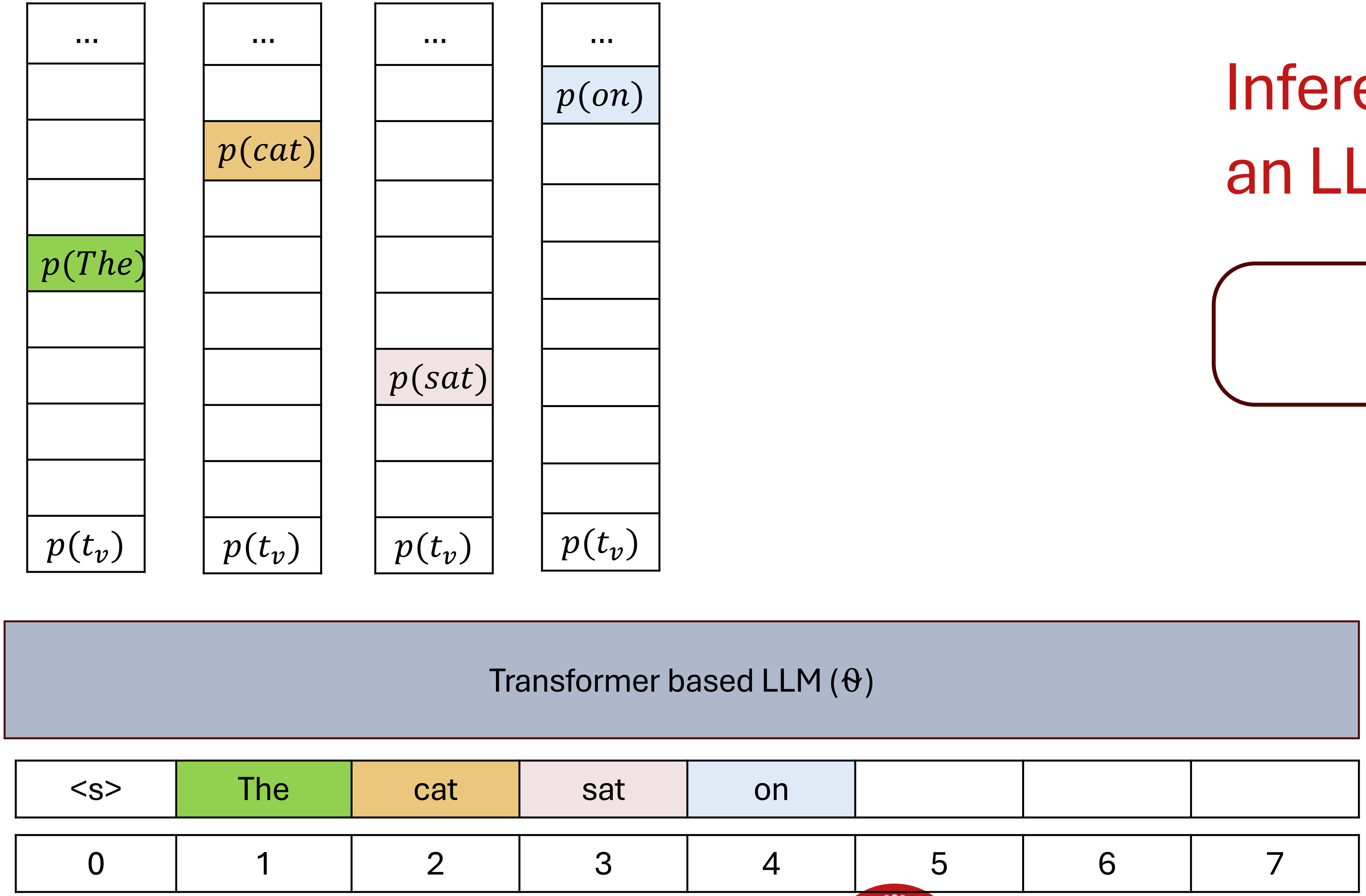
Inference through an LLM

Pick the token having max. probability at step 3



Inference through an LLM

Fill at step 4



Inference through an LLM

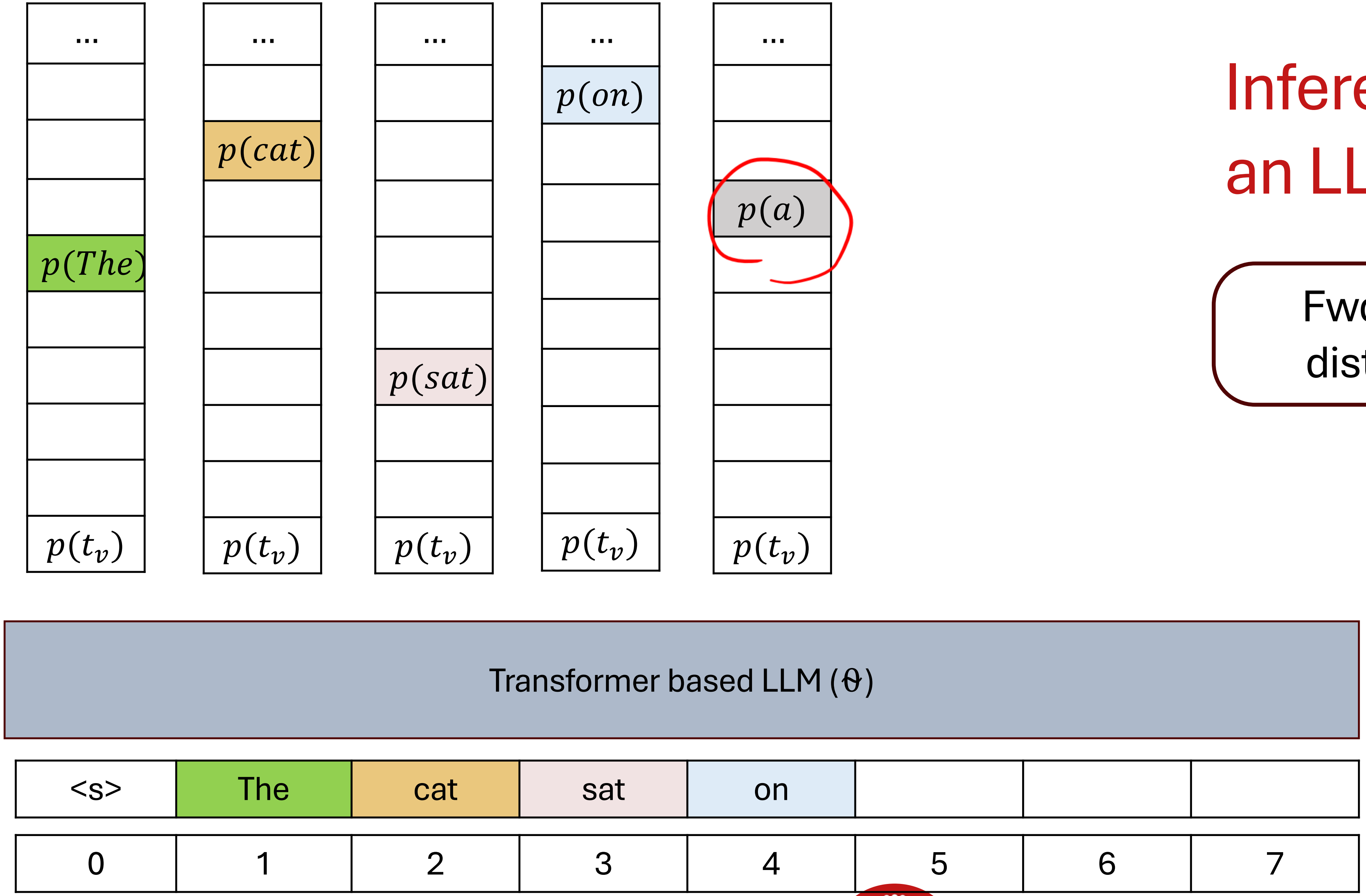
Fwd. Pass (#2)

Transformer based LLM (ϑ)

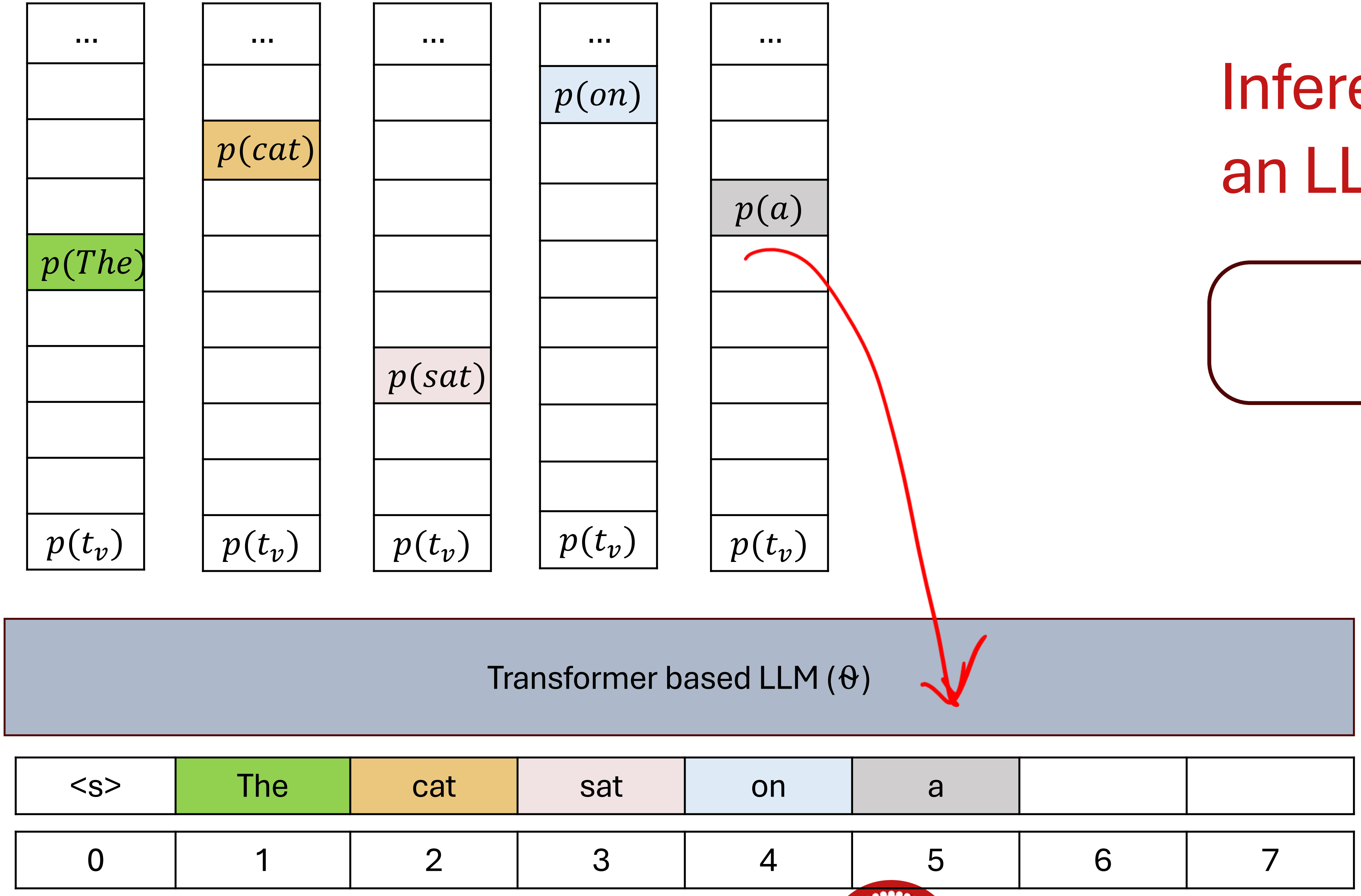
<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

Fwd. pass (#2) to get distribution at step 4



Inference through an LLM



Fill at step 5

Inference through an LLM

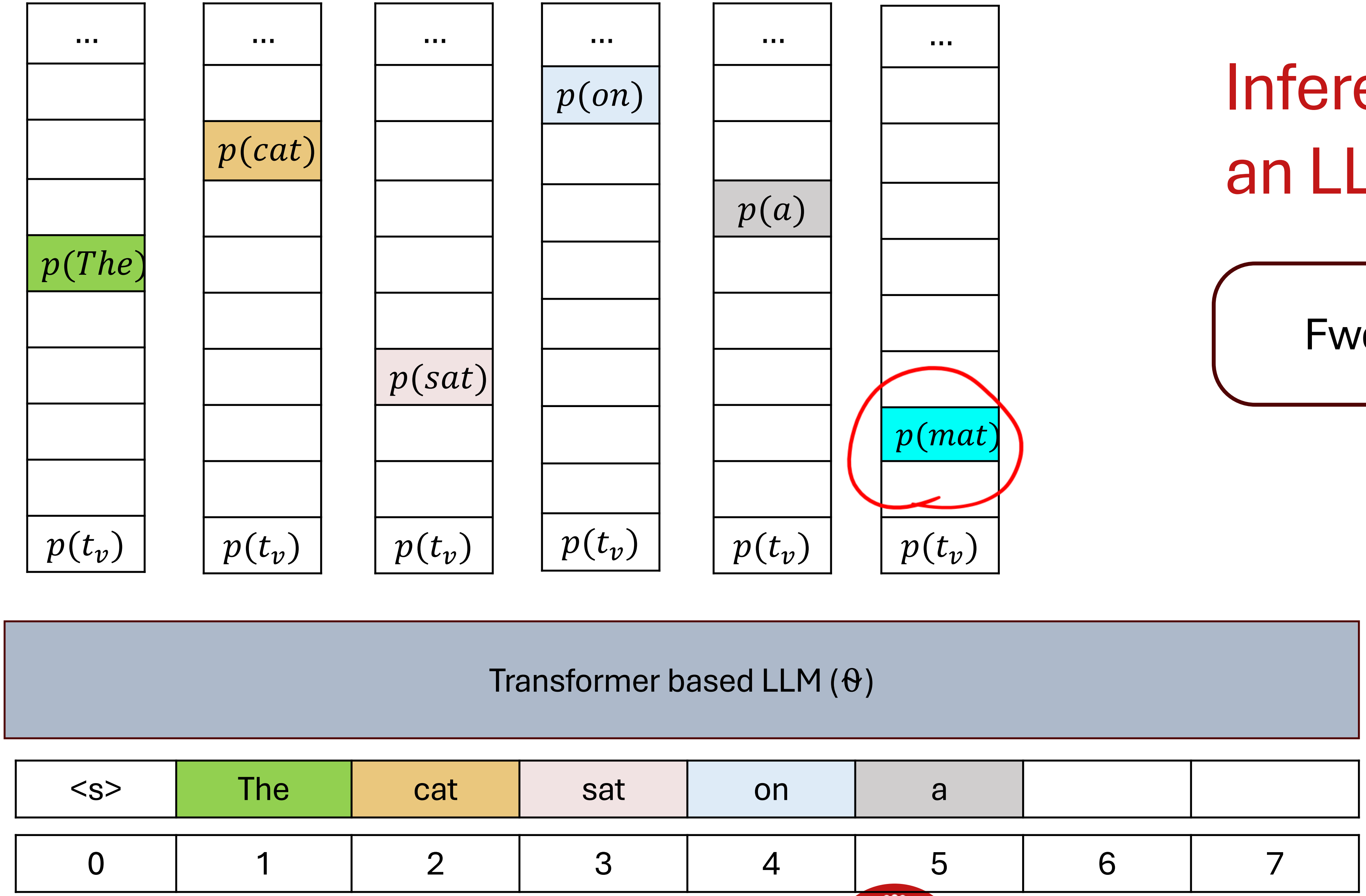
Fwd. pass again (#3)

Transformer based LLM (θ)

<s>	The	cat	sat	on	a		
0	1	2	3	4	5	6	7

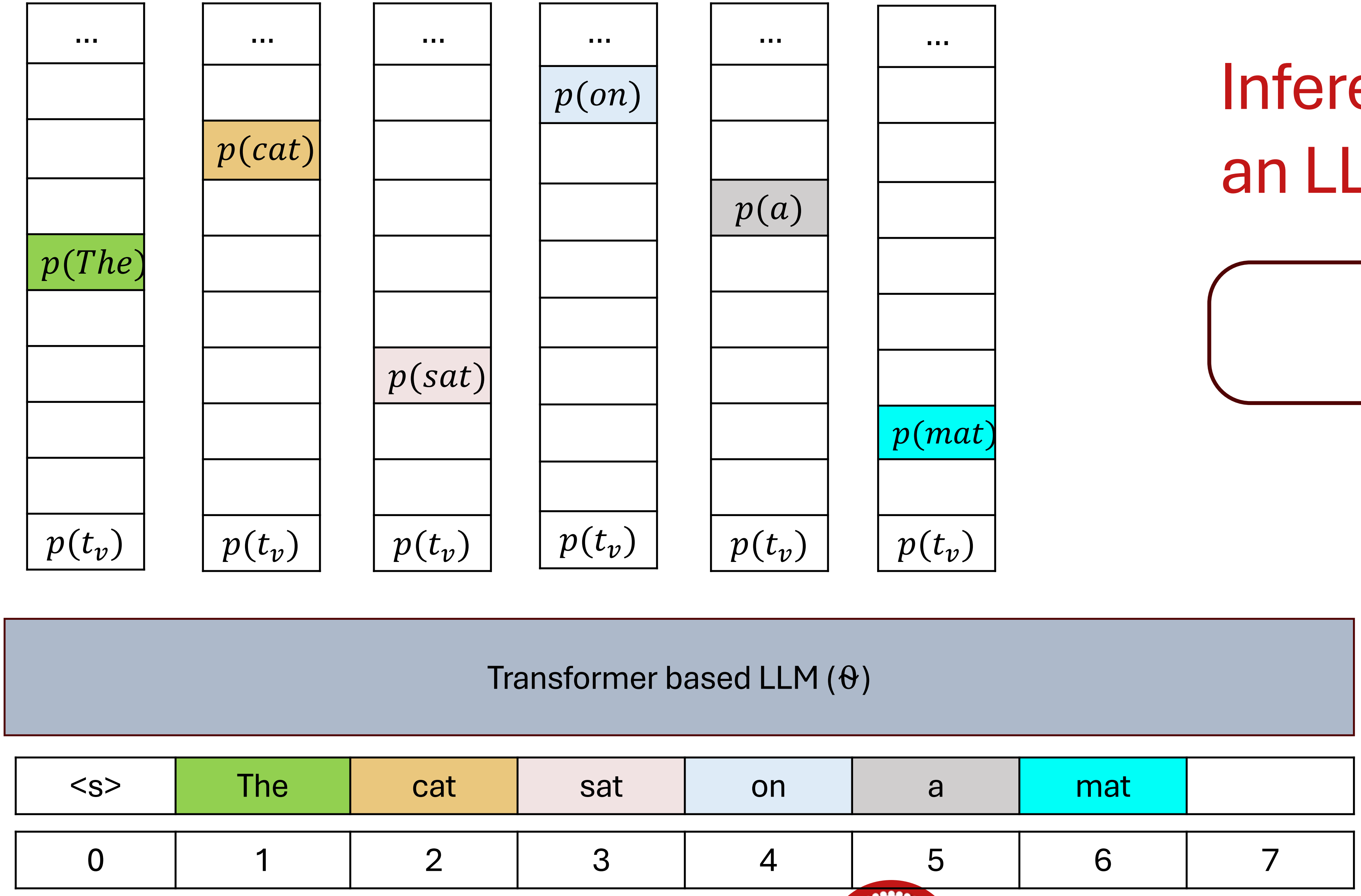
Inference through an LLM

Fwd. pass again (#3)



Inference through an LLM

Fill at step 6



Inference through an LLM

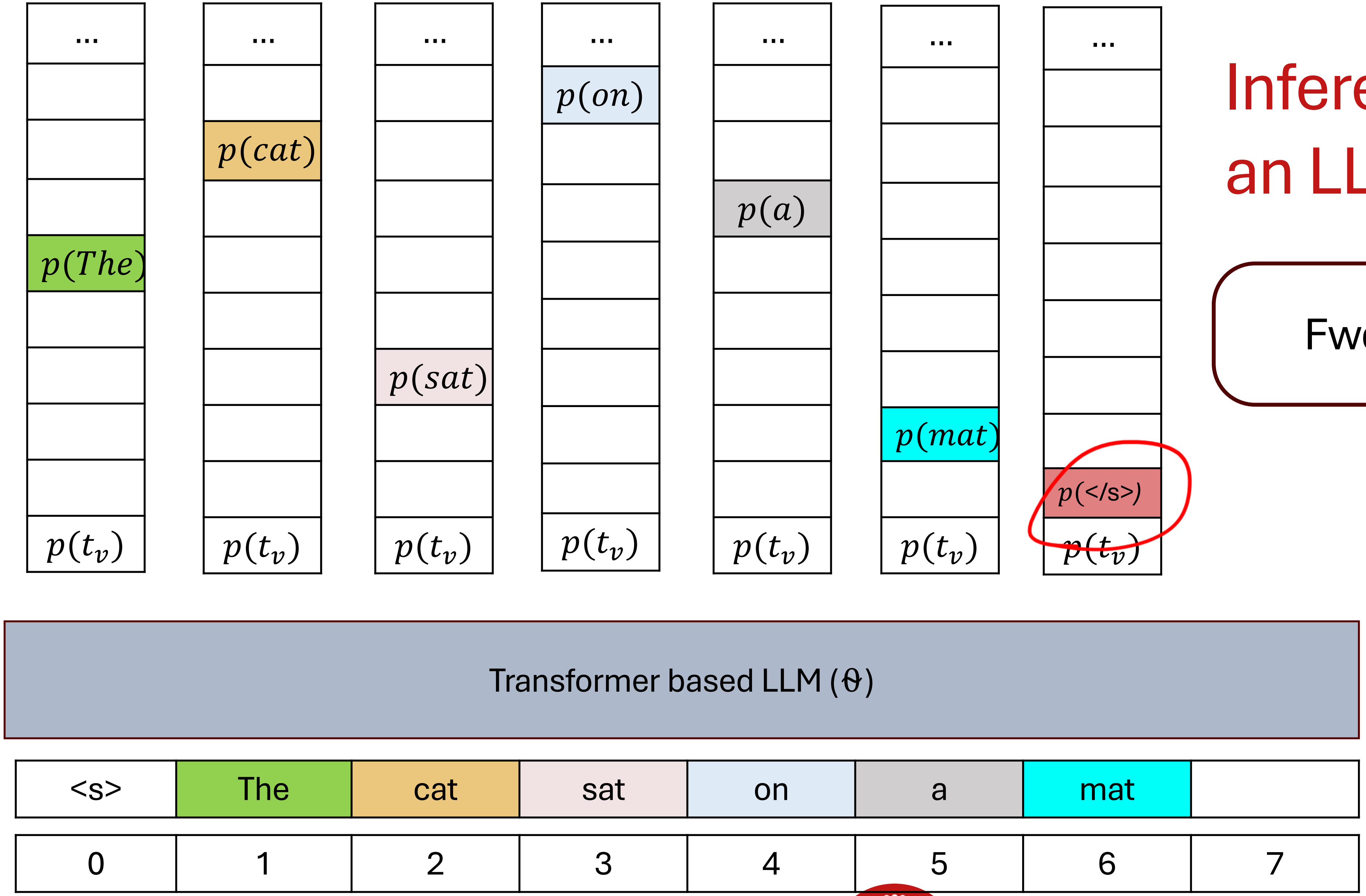
Fwd. pass again (#4)

Transformer based LLM (ϑ)

<s>	The	cat	sat	on	a	mat	
0	1	2	3	4	5	6	7

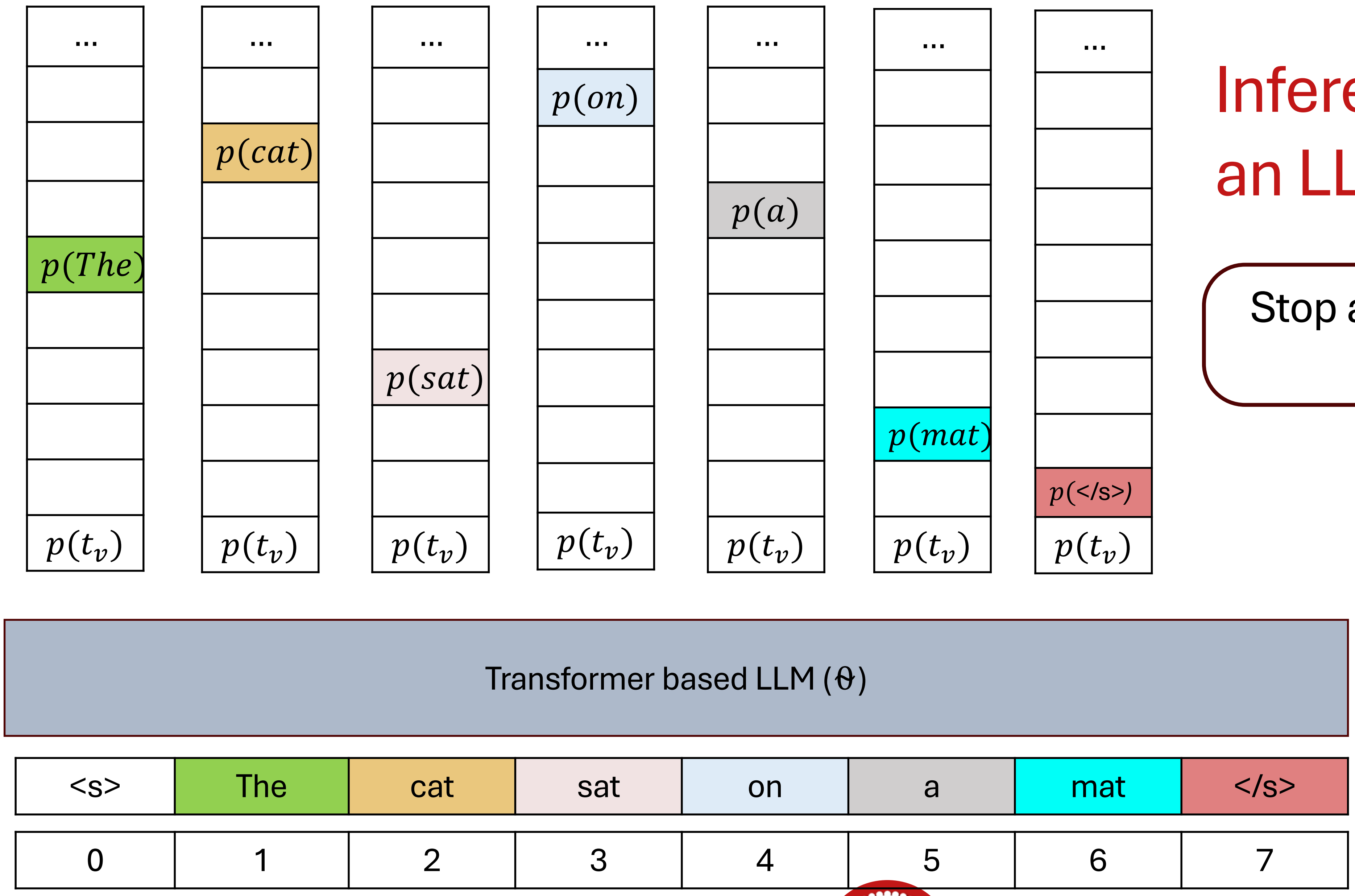
Inference through an LLM

Fwd. pass again (#4)

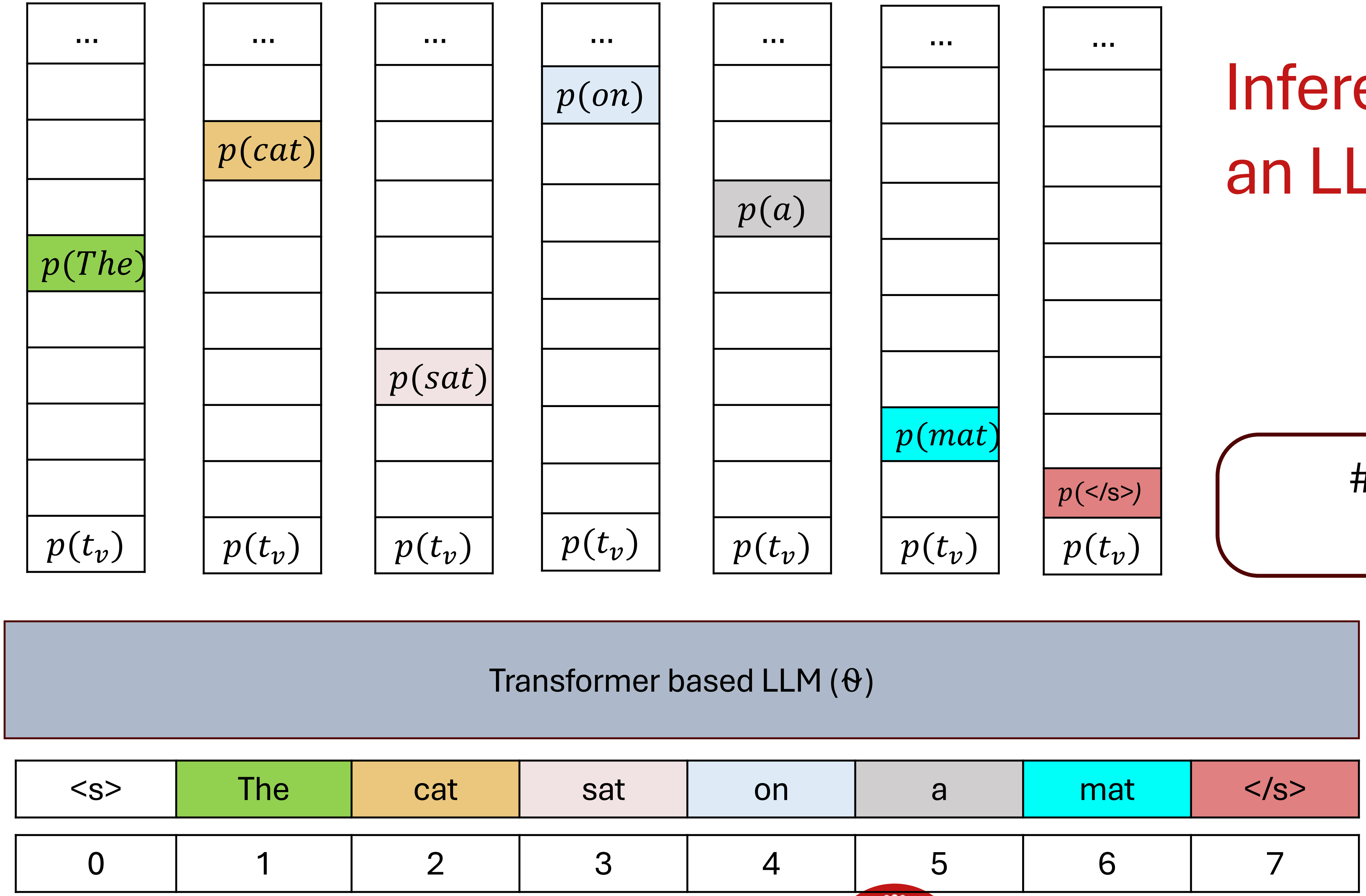


Inference through an LLM

Stop at end of seq. token: </s>



Inference through an LLM



Fwd Passes: 4

#Tokens: 4

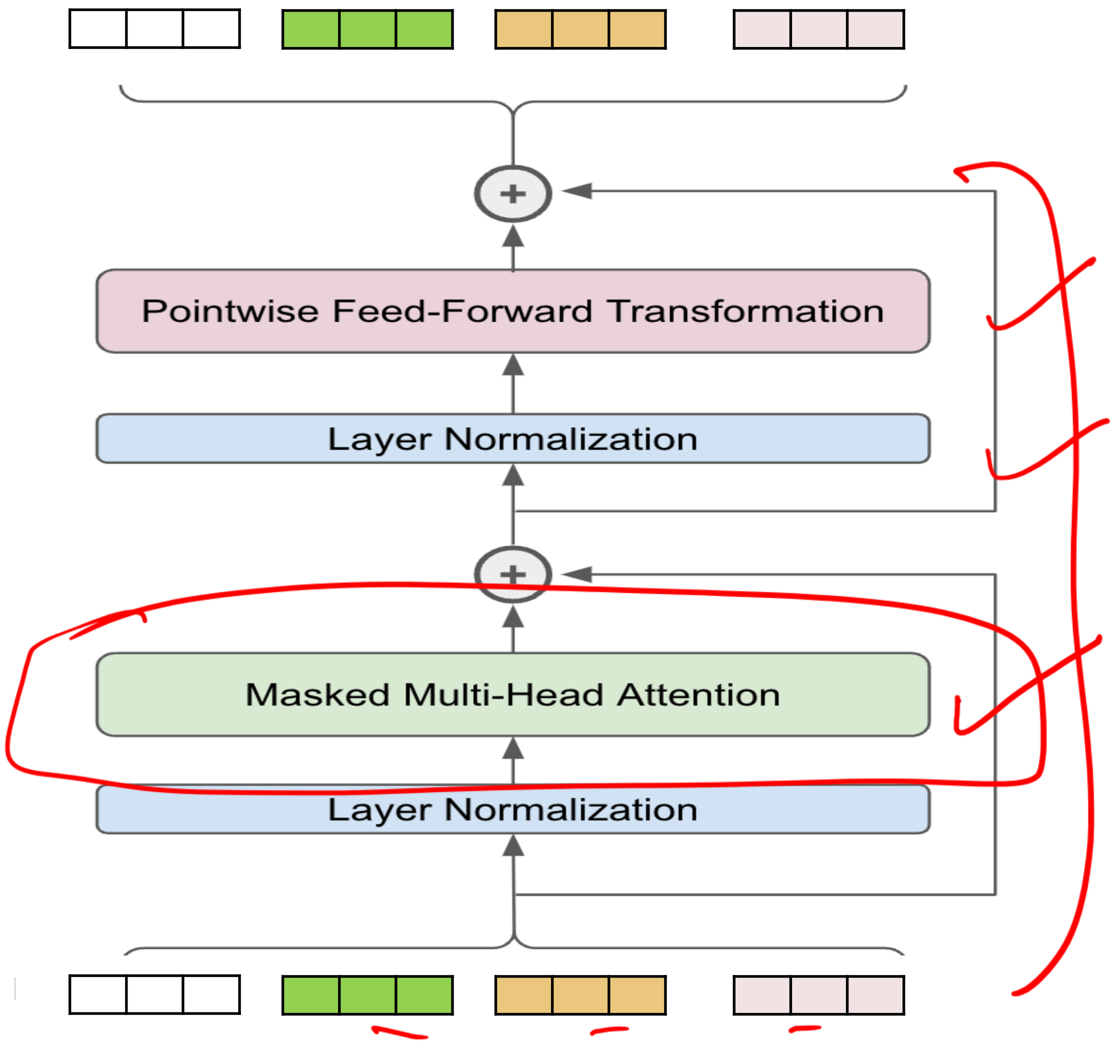
Inference through an LLM

- 4 forward passes for 4 tokens
- Not feasible at production scale
- Let us revisit forward pass through and see if we can optimize
- We will focus on attention layer as that is the bottleneck

Fwd Passes: 4
#Tokens: 4

Transformer based LLM (θ)

<s>	The	cat	sat	on	a	mat	</s>
0	1	2	3	4	5	6	7

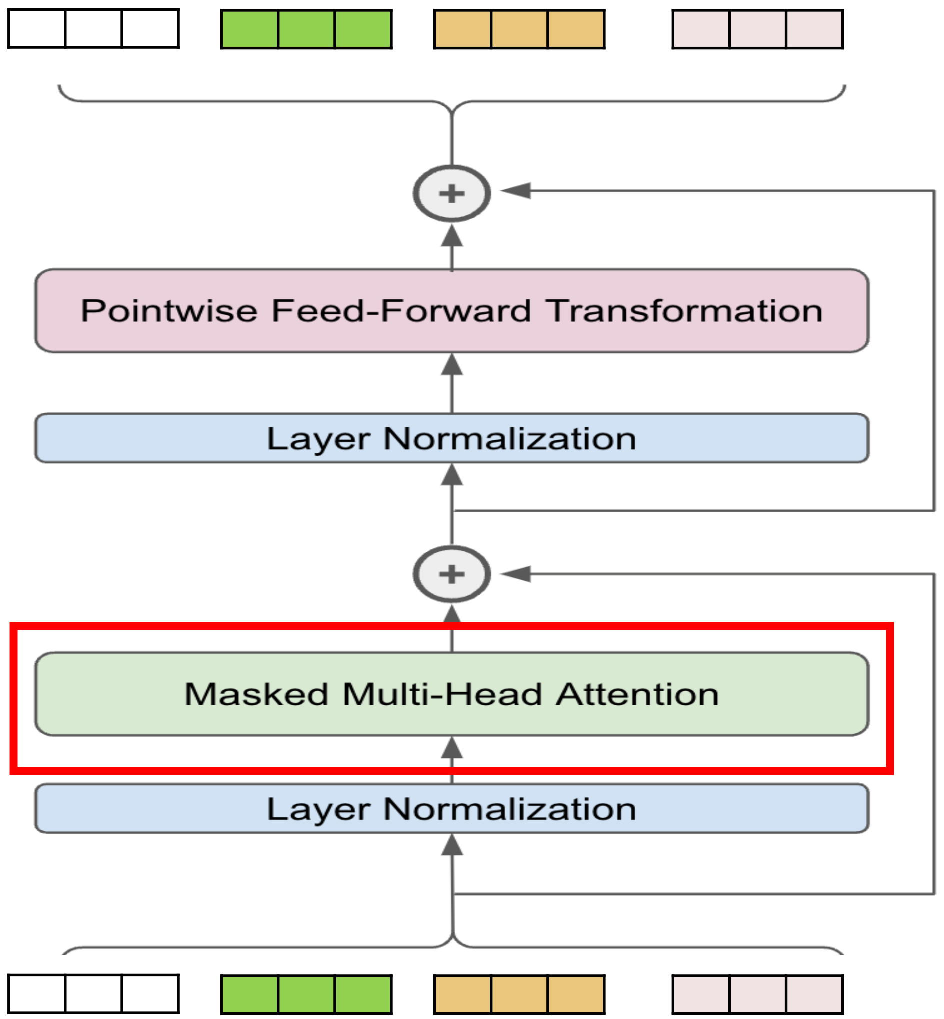


Inference through an LLM

Forward Pass #1

<s>	The	cat	sat				
0	1	2	3	4	5	6	7

Content credits: https://cameronrwolfe.substack.com/p/decoder-only-transformers-the-workhorse



Inference through an LLM

Forward Pass #1

<s>	The	cat	sat				
0	1	2	3	4	5	6	7

Content credits: https://cameronrwolfe.substack.com/p/decoder-only-transformers-the-workhorse

Inference through an LLM

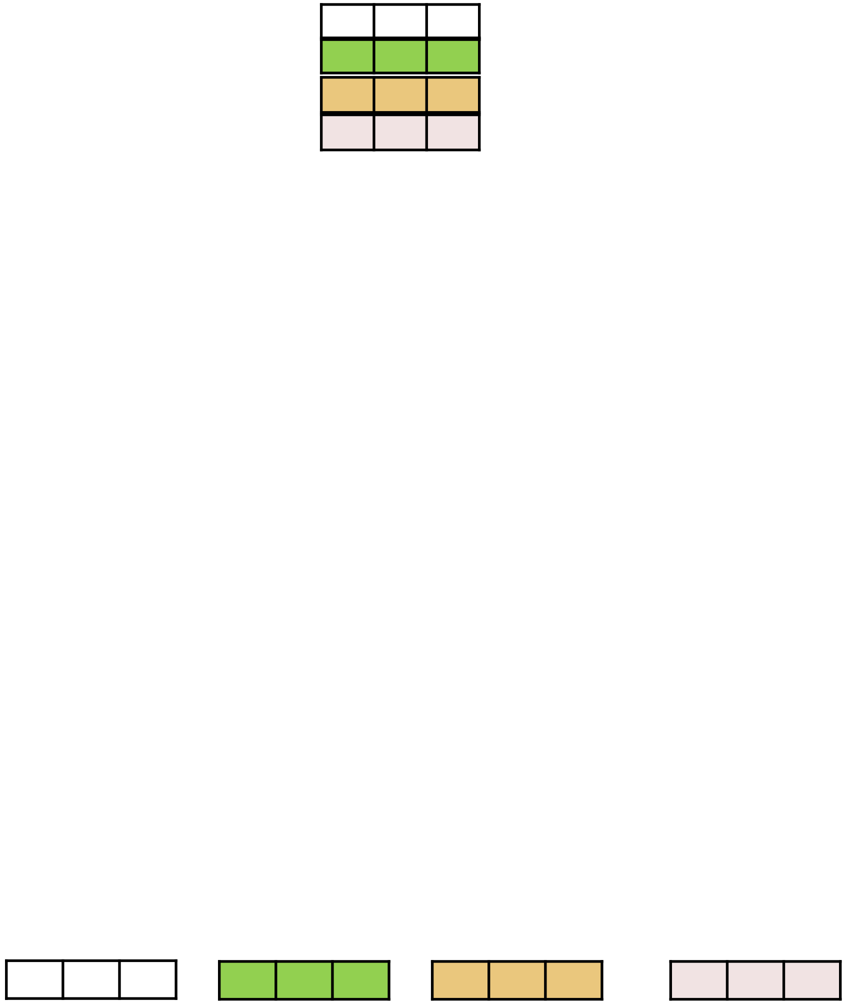
Forward Pass #1

<s>	The	cat	sat				
0	1	2	3	4	5	6	7

Content credits: https://cameronrwolfe.substack.com/p/decoder-only-transformers-the-workhorse

Inference through an LLM

Forward Pass #1



<s>	The	cat	sat				
0	1	2	3	4	5	6	7

Content credits: https://cameronrwolfe.substack.com/p/decoder-only-transformers-the-workhorse

Inference through an LLM

Forward Pass #1

W_Q = <s> | The | cat | sat — **Q**: $4 \times d$ dim.

W_K = <s> | The | cat | sat — **K**: $4 \times d$ dim.

W_V = <s> | The | cat | sat — **V**: $4 \times d$ dim.

<s>	The	cat	sat				
0	1	2	3	4	5	6	7

Content credits: https://cameronrwolfe.substack.com/p/decoder-only-transformers-the-workhorse

Inference through an LLM

Forward Pass #1

Q: *4 x d* dim.

<s>
The
cat
sat

V: *4 x d* dim.

<s>
The
cat
sat

<s>	The	cat	sat

K^T: *d x 4* dim.

<s>	The	cat	sat				
0	1	2	3	4	5	6	7

Inference through an LLM

Forward Pass #1

Q: *4 x d* dim.

<s>
The
cat
sat

A: 4 x 4 dim.

$$A = softmax\left(\frac{QK^T}{\sqrt{d}}\right)$$

V: *4 x d* dim.

<s>
The
cat
sat

<s>	The	cat	sat

K^T: *d x 4* dim.

<s>	The	cat	sat				
0	1	2	3	4	5	6	7

Inference through an LLM

Forward Pass #1

Q: *4 x d* dim.

<s>
The
cat
sat

A: 4 x 4 dim.

1			
0.2	0.8		
0.1	0.3	0.6	
0.01	0.19	0.3	0.5

V: *4 x d* dim.

<s>
The
cat
sat

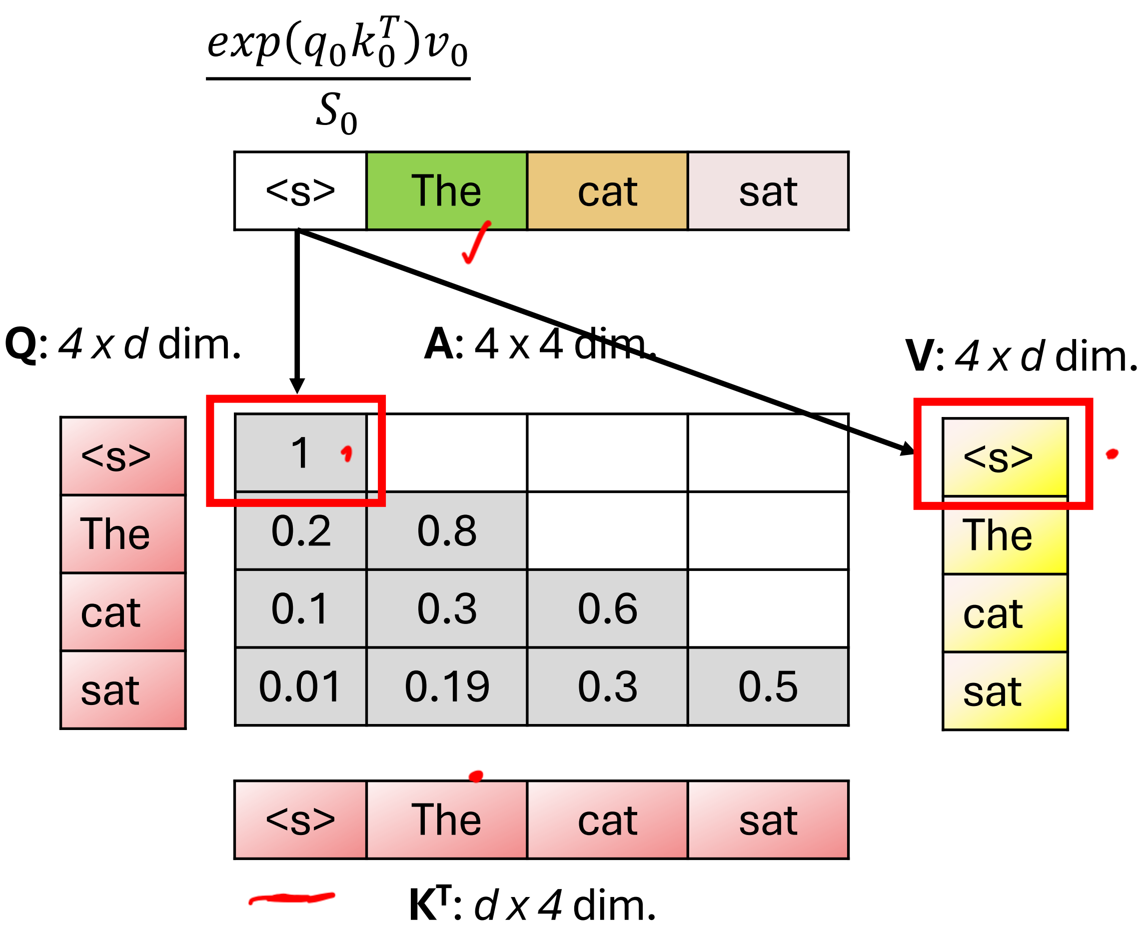
<s>	The	cat	sat

K^T: *d x 4* dim.

<s>	The	cat	sat				
0	1	2	3	4	5	6	7

Inference through an LLM

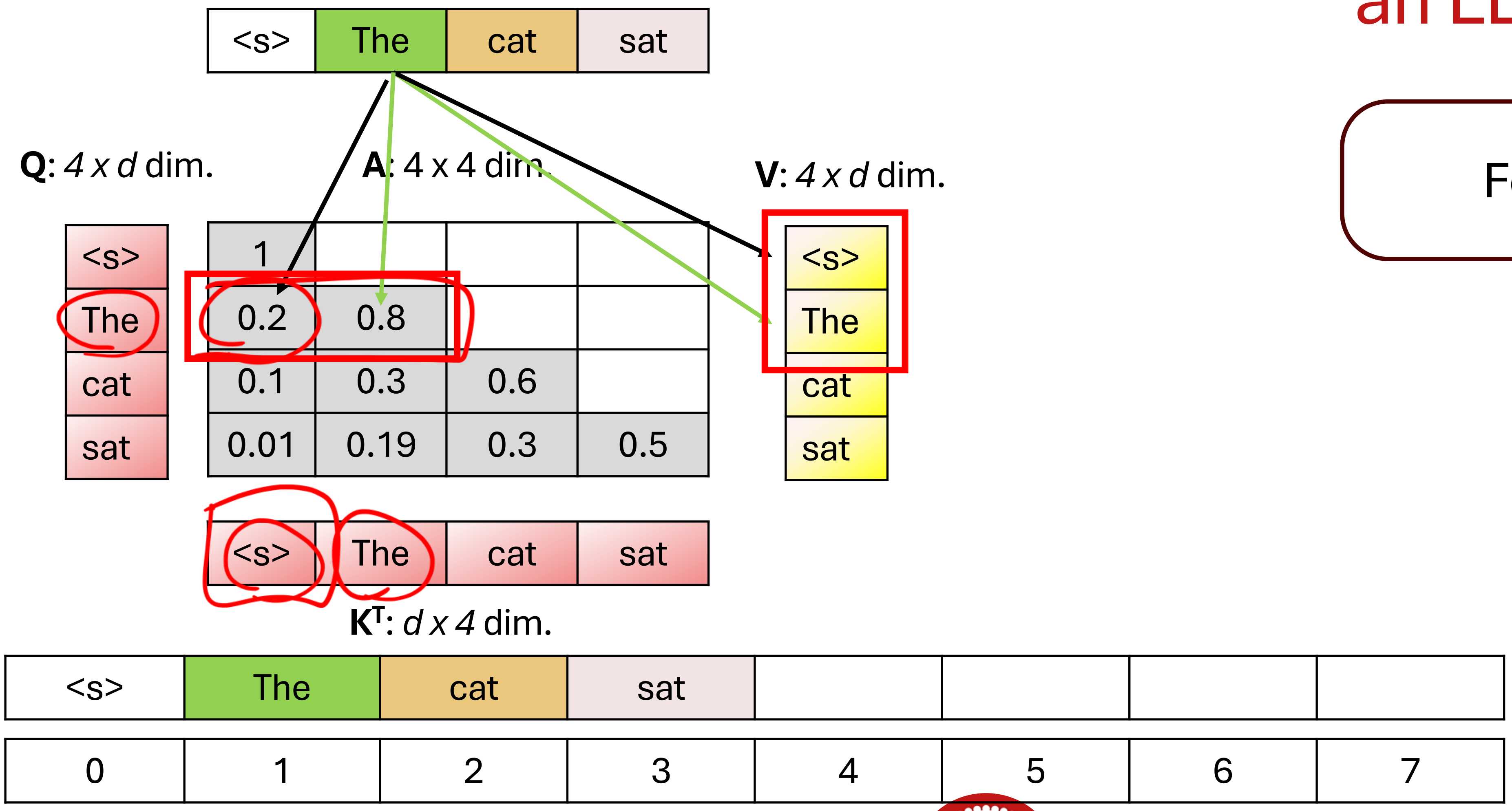
Forward Pass #1



<s>	The	cat	sat				
0	1	2	3	4	5	6	7

Inference through an LLM

$$\frac{exp(q_1 k_0^T)v_0}{S_1} + \frac{exp(q_1 k_1^T)v_1}{S_1}$$

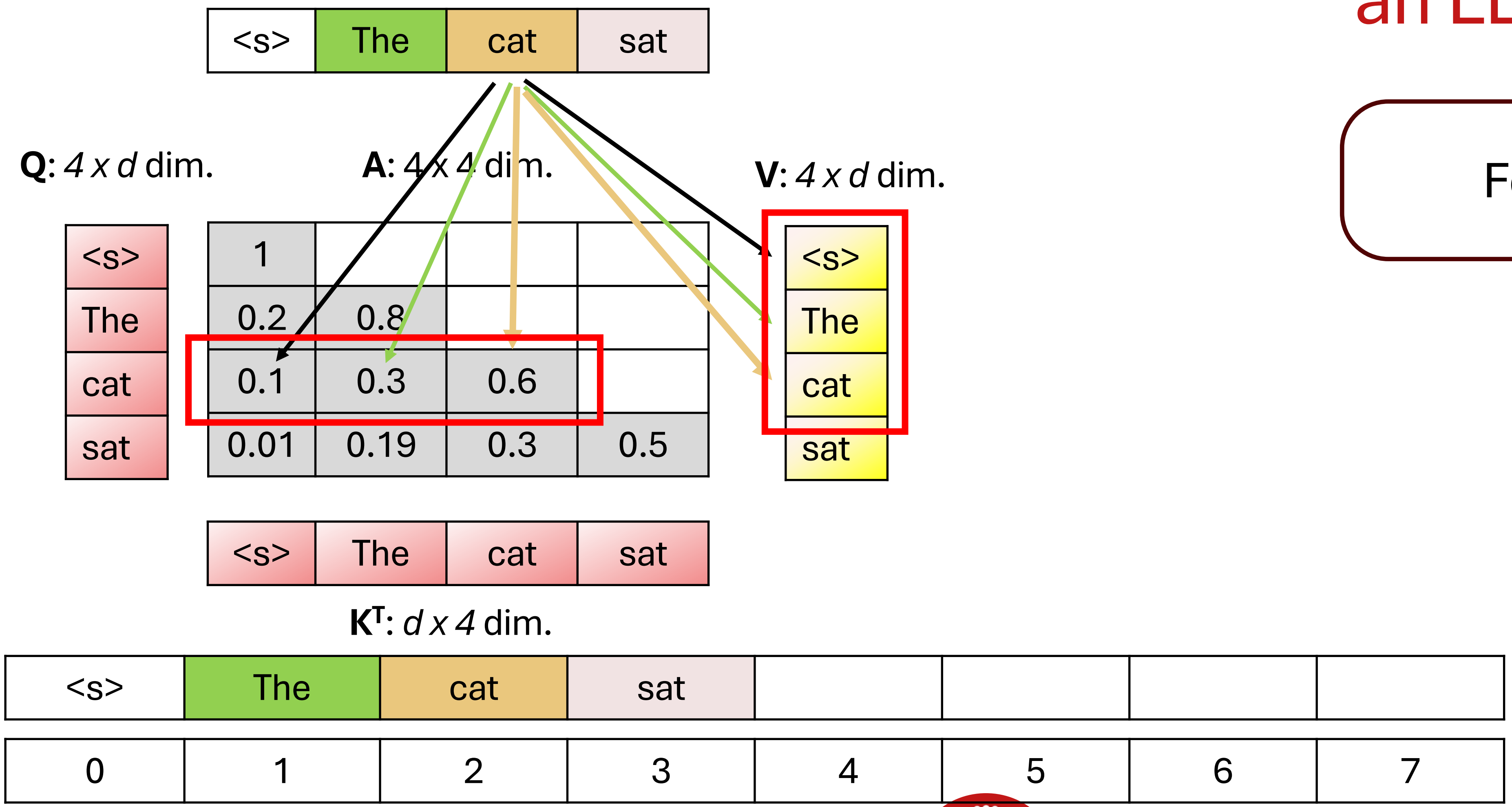


Forward Pass #1

Inference through an LLM

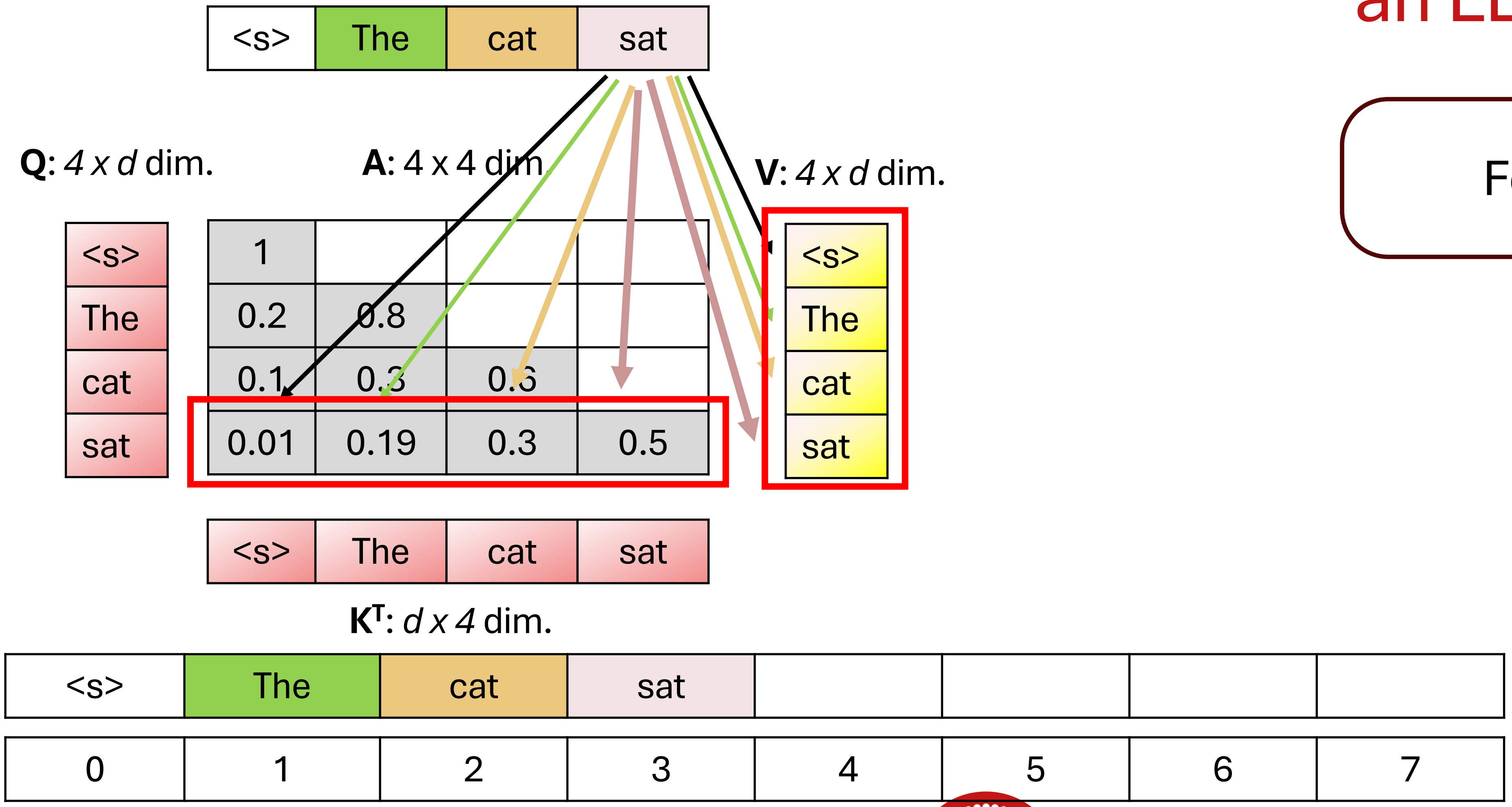
Forward Pass #1

$$\frac{exp(q_2 k_0^T) v_0}{S_2} + \frac{exp(q_2 k_1^T) v_1}{S_2} + \frac{exp(q_2 k_2^T) v_2}{S_2}$$

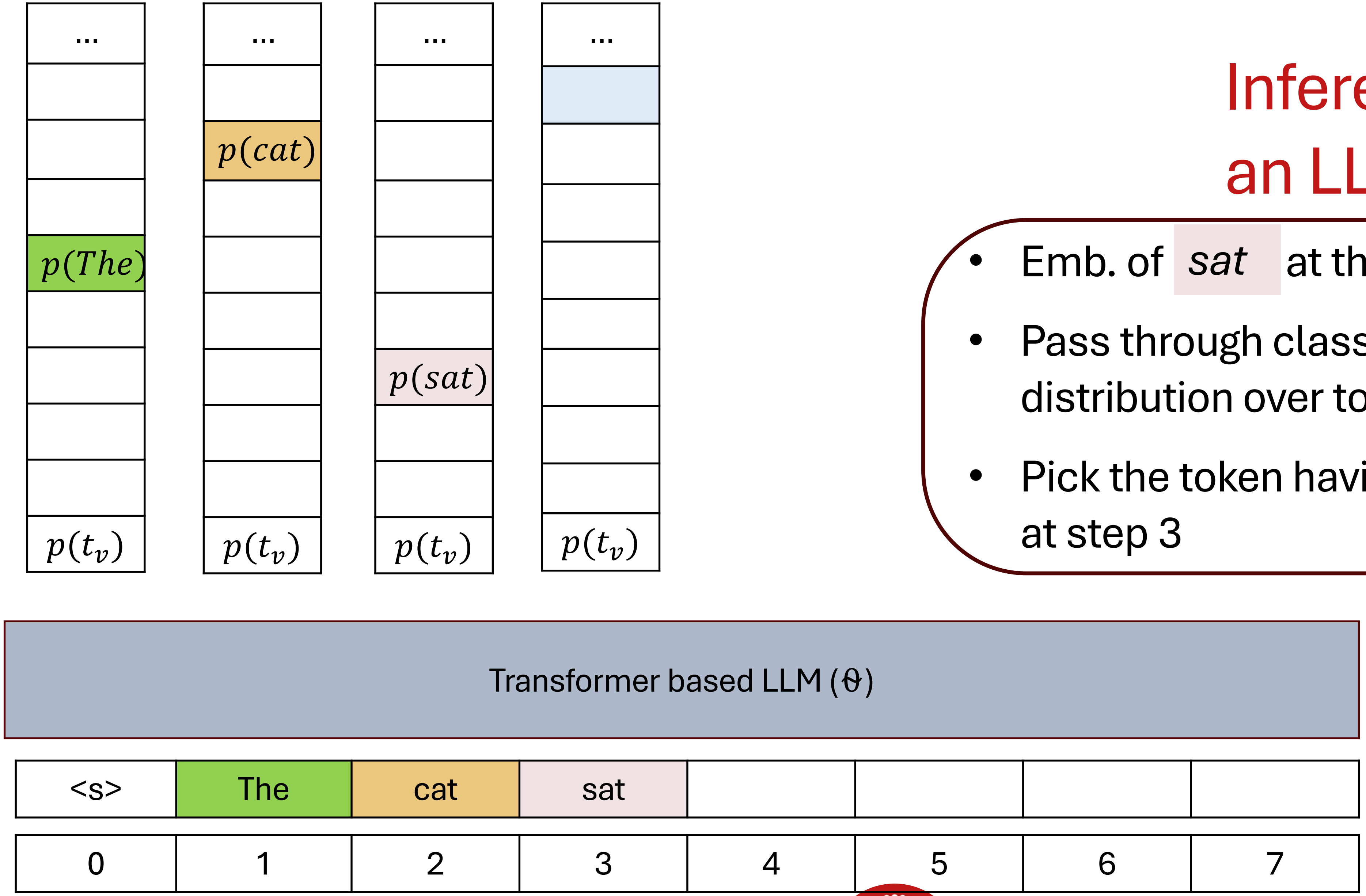


Inference through an LLM

$$\frac{exp(q_3k_0^T)v_0}{S_3} + \frac{exp(q_3k_1^T)v_1}{S_3} + \frac{exp(q_3k_2^T)v_2}{S_3} + \frac{exp(q_3k_3^T)v_3}{S_3}$$

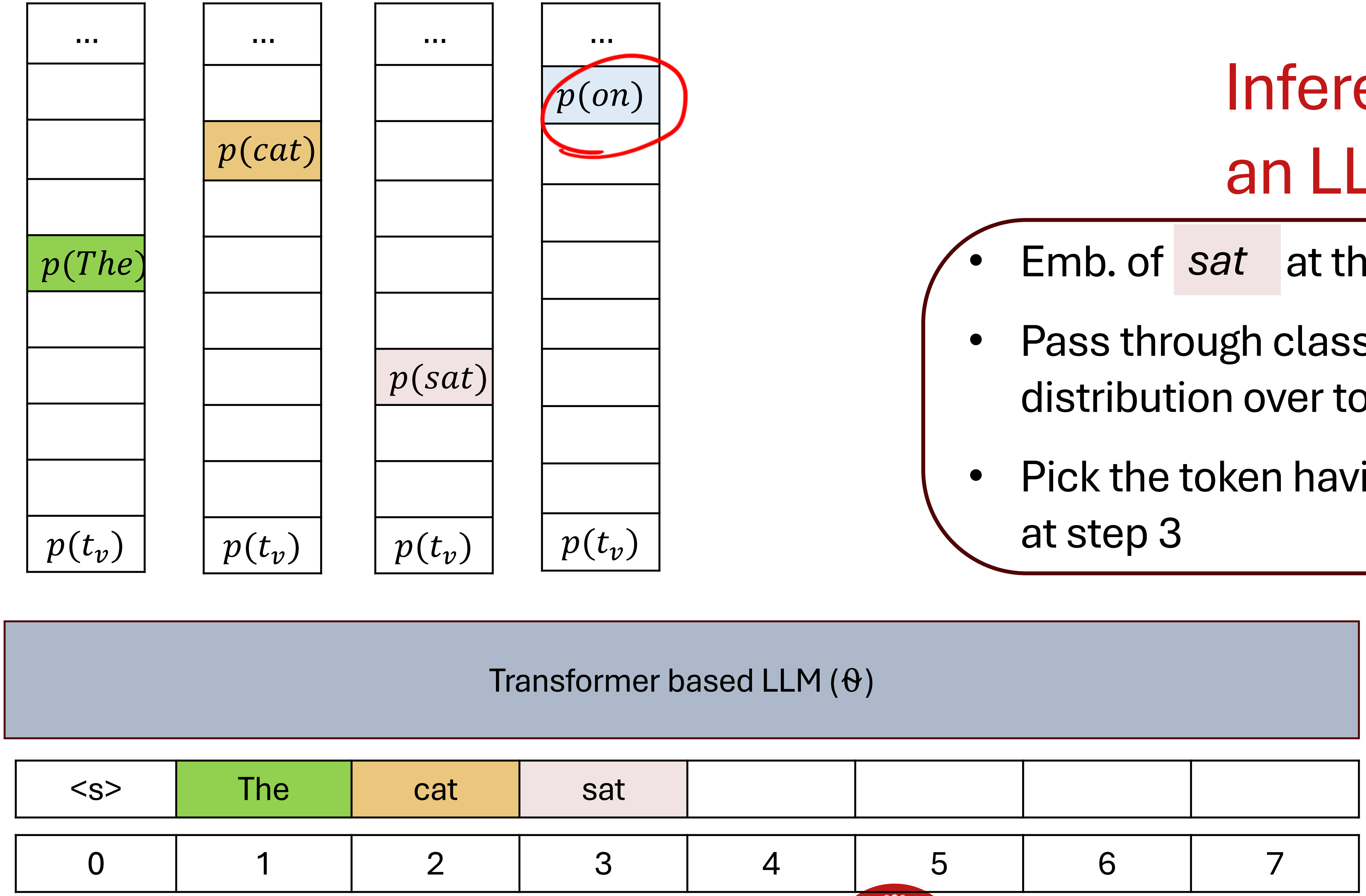


Forward Pass #1



Inference through an LLM

- Emb. of *sat* at the last layer
- Pass through classifier to get distribution over tokens
- Pick the token having max. probability at step 3

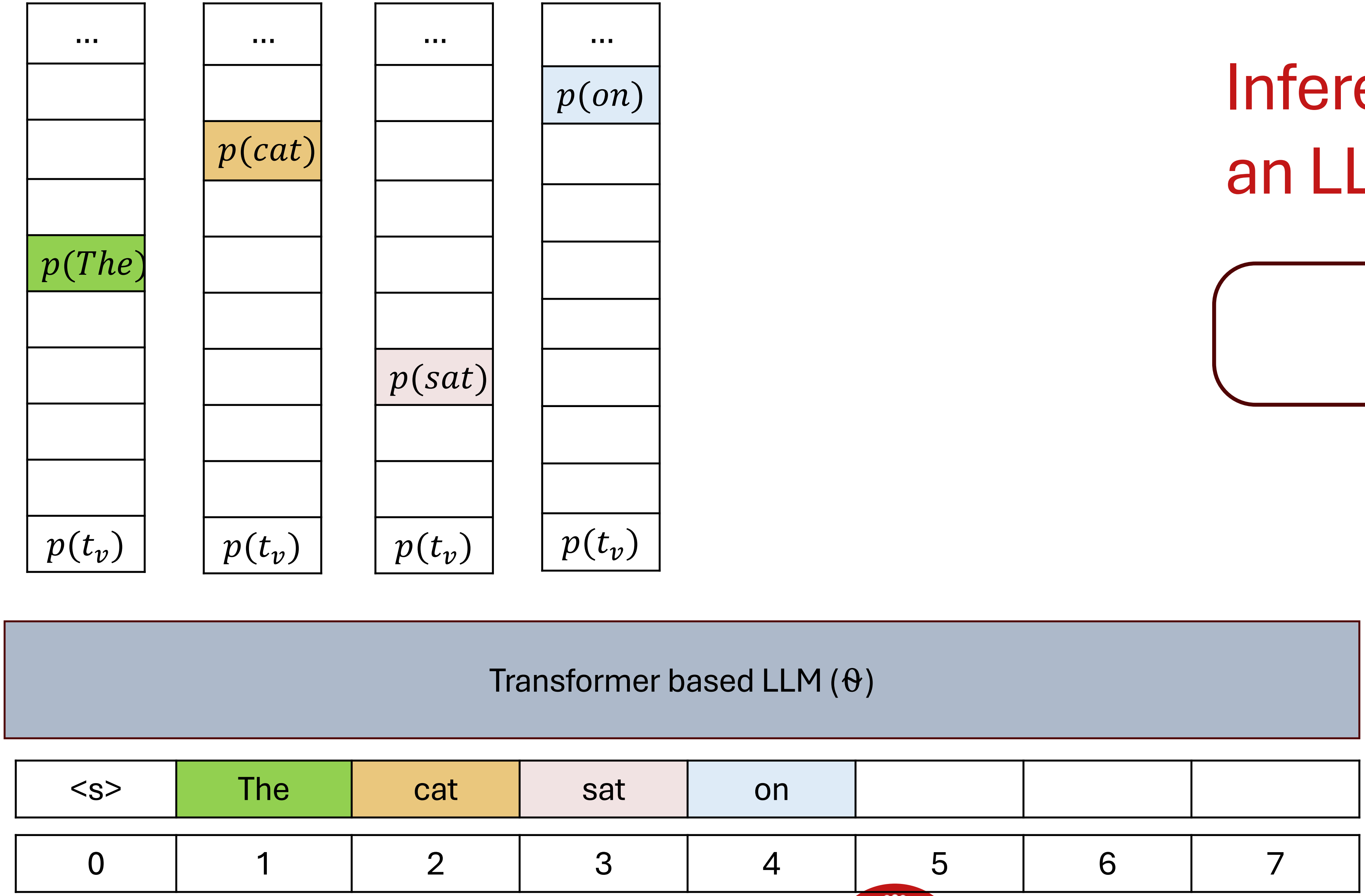


Inference through an LLM

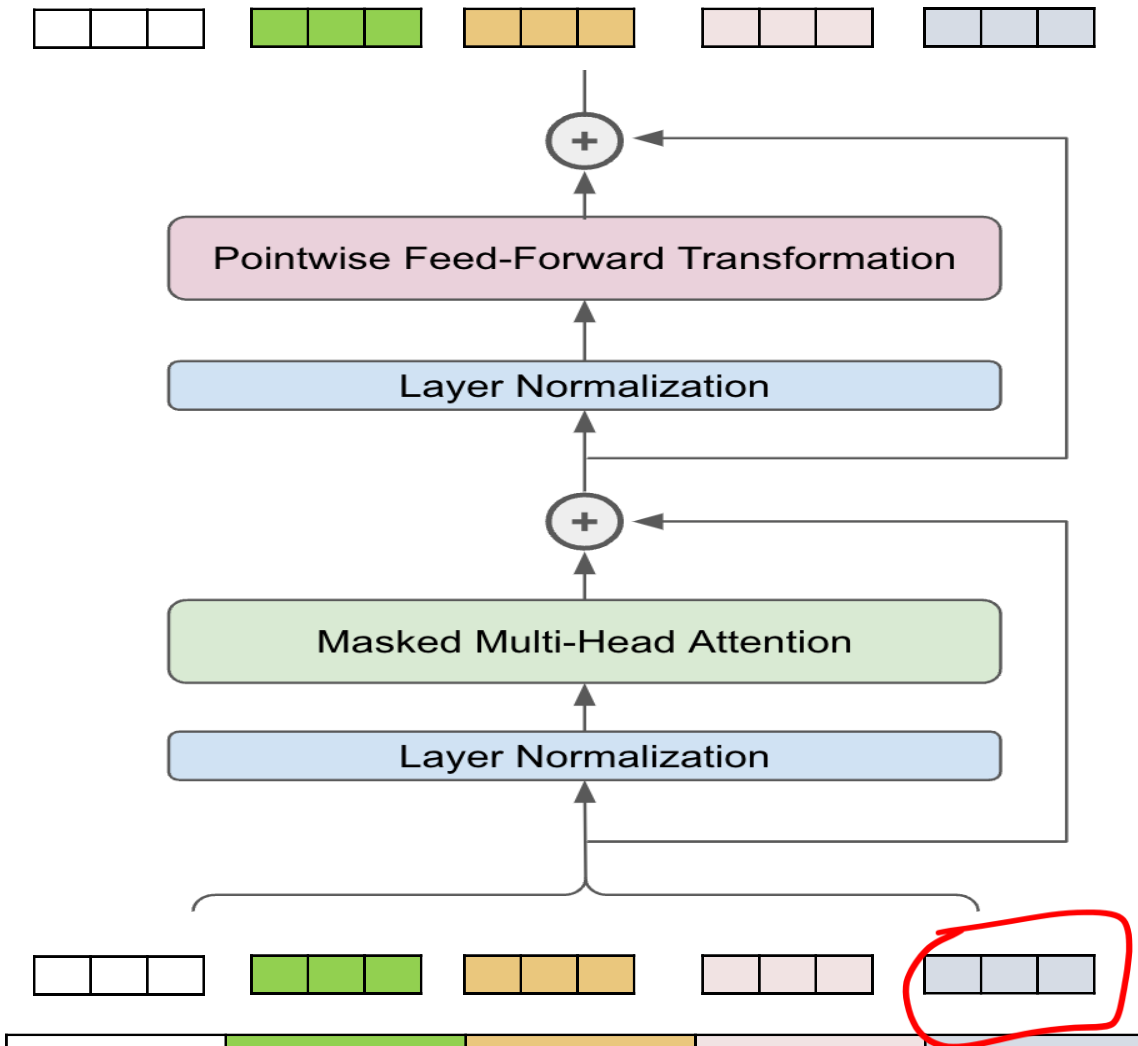
- Emb. of *sat* at the last layer
- Pass through classifier to get distribution over tokens
- Pick the token having max. probability at step 3

Inference through an LLM

Fill at step 4



Inference through an LLM



<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Forward Pass #2

Content credits: https://cameronrwolfe.substack.com/p/decoder-only-transformers-the-workhorse

Inference through an LLM

- A lot of computation already done in Fwd. pass #1

Forward Pass #2

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

$$A = softmax\left(\frac{QK^T}{\sqrt{d}}\right)$$

V: *5 x d* dim.

<s>
The
cat
sat
on

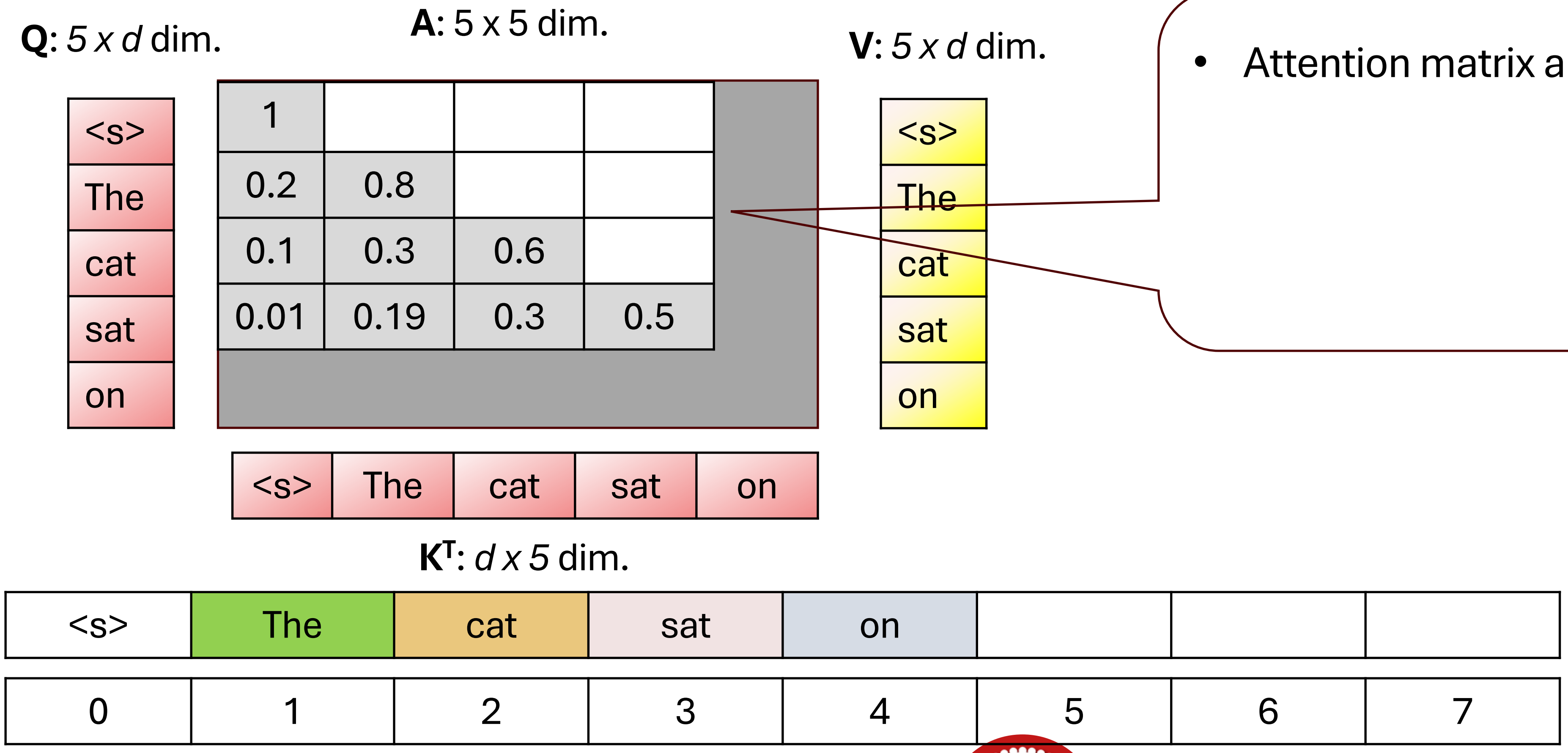
<s>	The	cat	sat	on

K^T: *d x 5* dim.

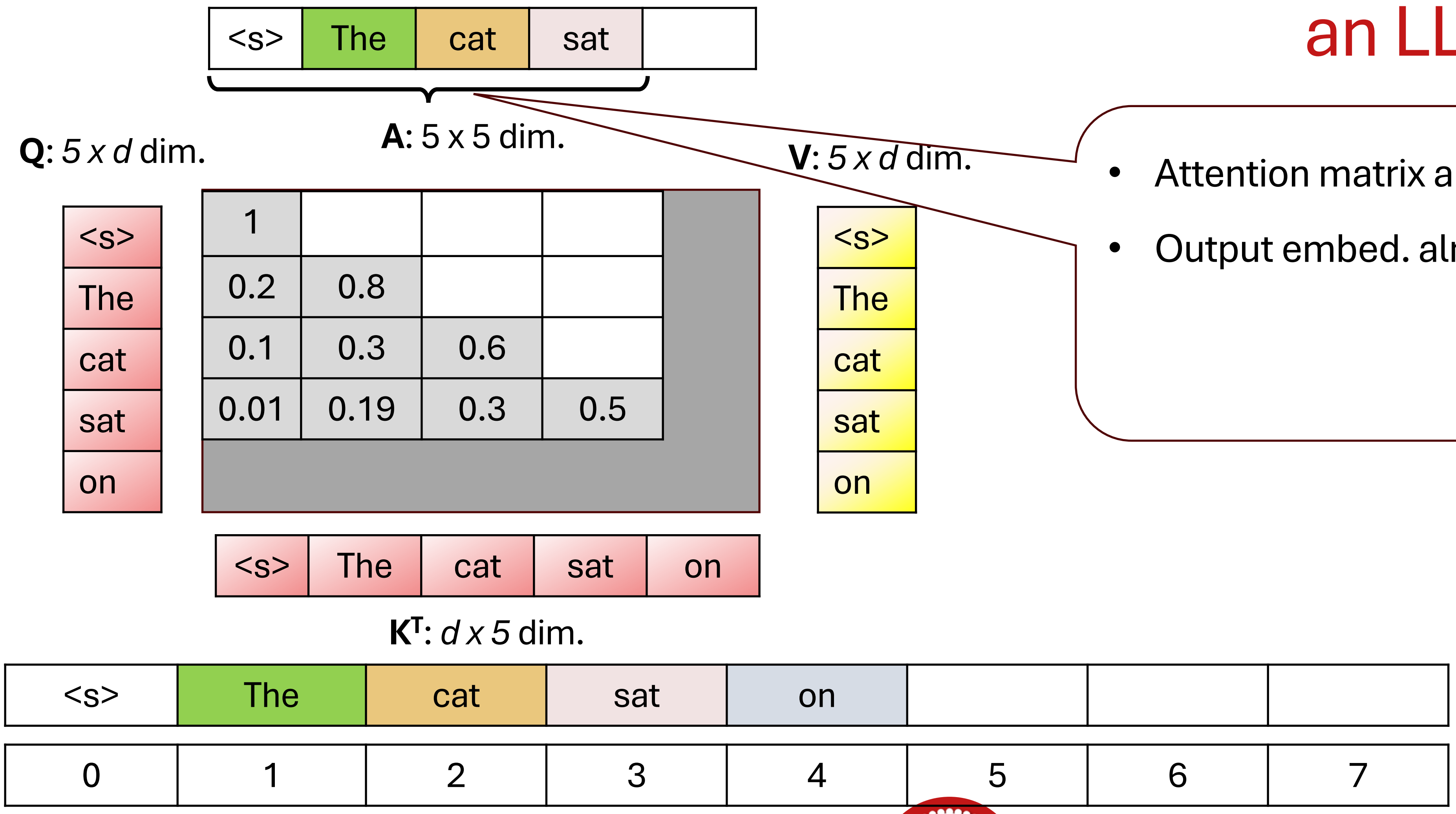
<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

- Attention matrix already computed in #1

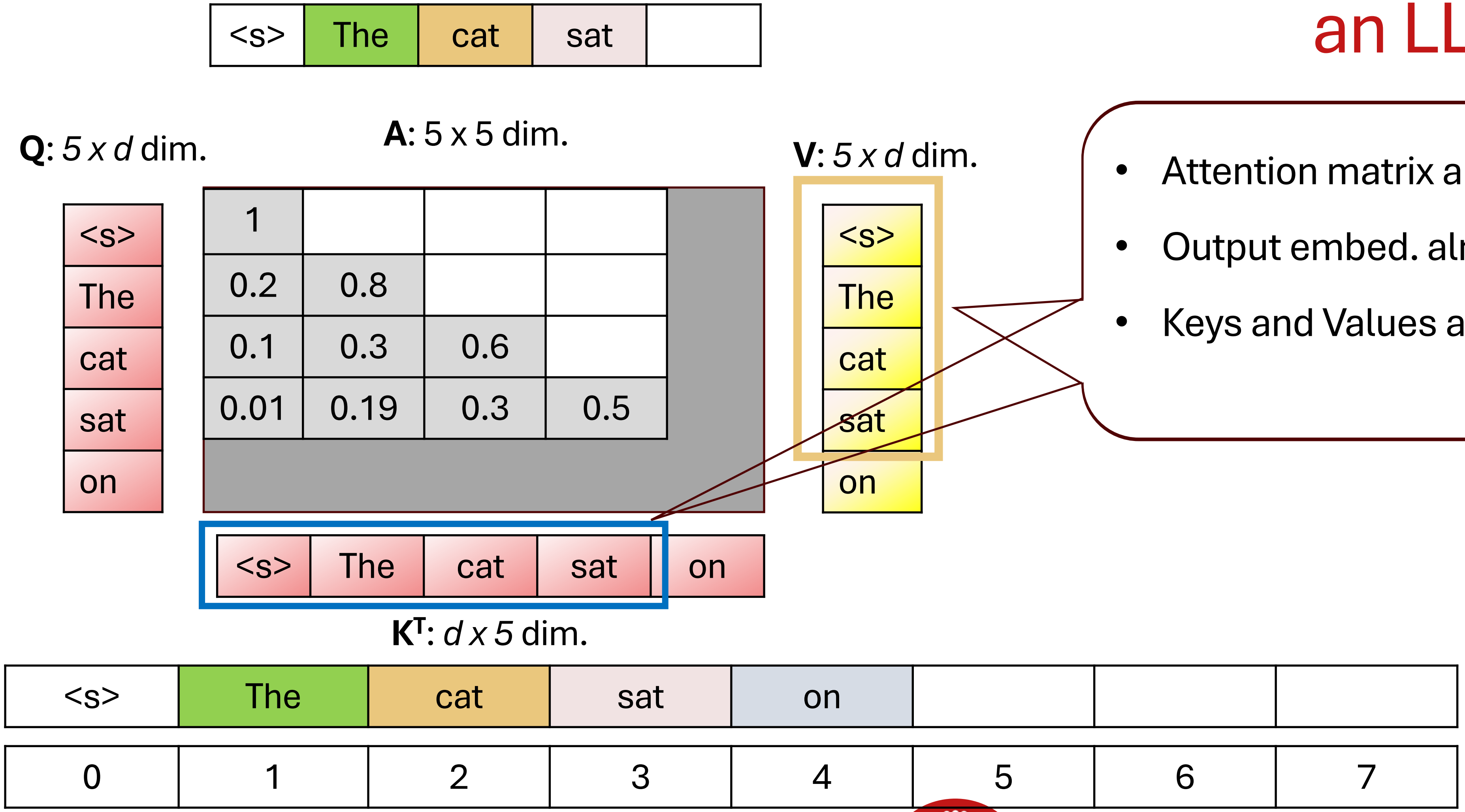


Inference through an LLM



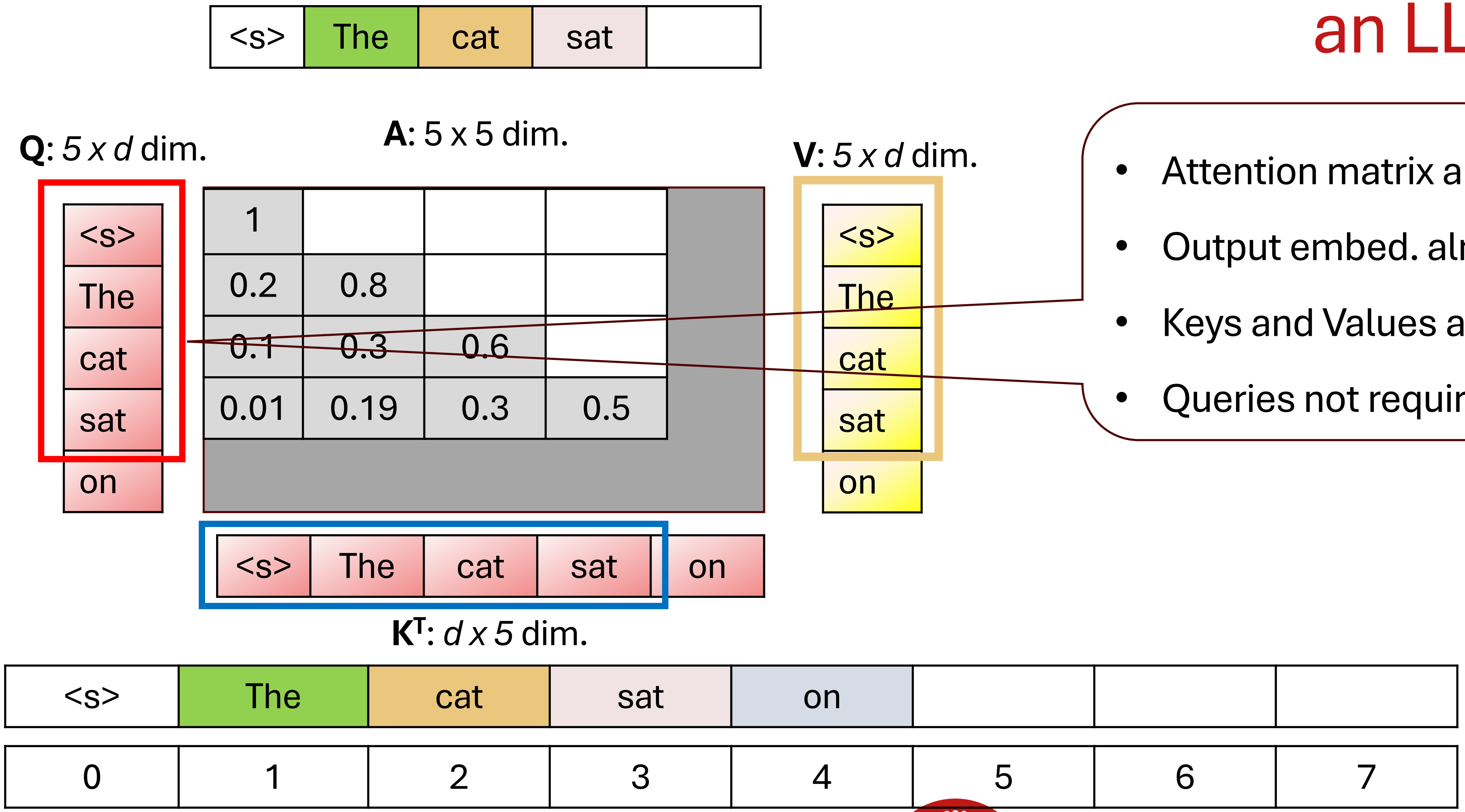
- Attention matrix already computed in #1
- Output embed. already computed in #1

Inference through an LLM



- Attention matrix already computed in #1
- Output embed. already computed in #1
- Keys and Values already computed in #1

Inference through an LLM



- Attention matrix already computed in #1
- Output embed. already computed in #1
- Keys and Values already computed in #1
- Queries not required in #2

Inference through an LLM

- Cache the already computed matrices

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	

V: *5 x d* dim.

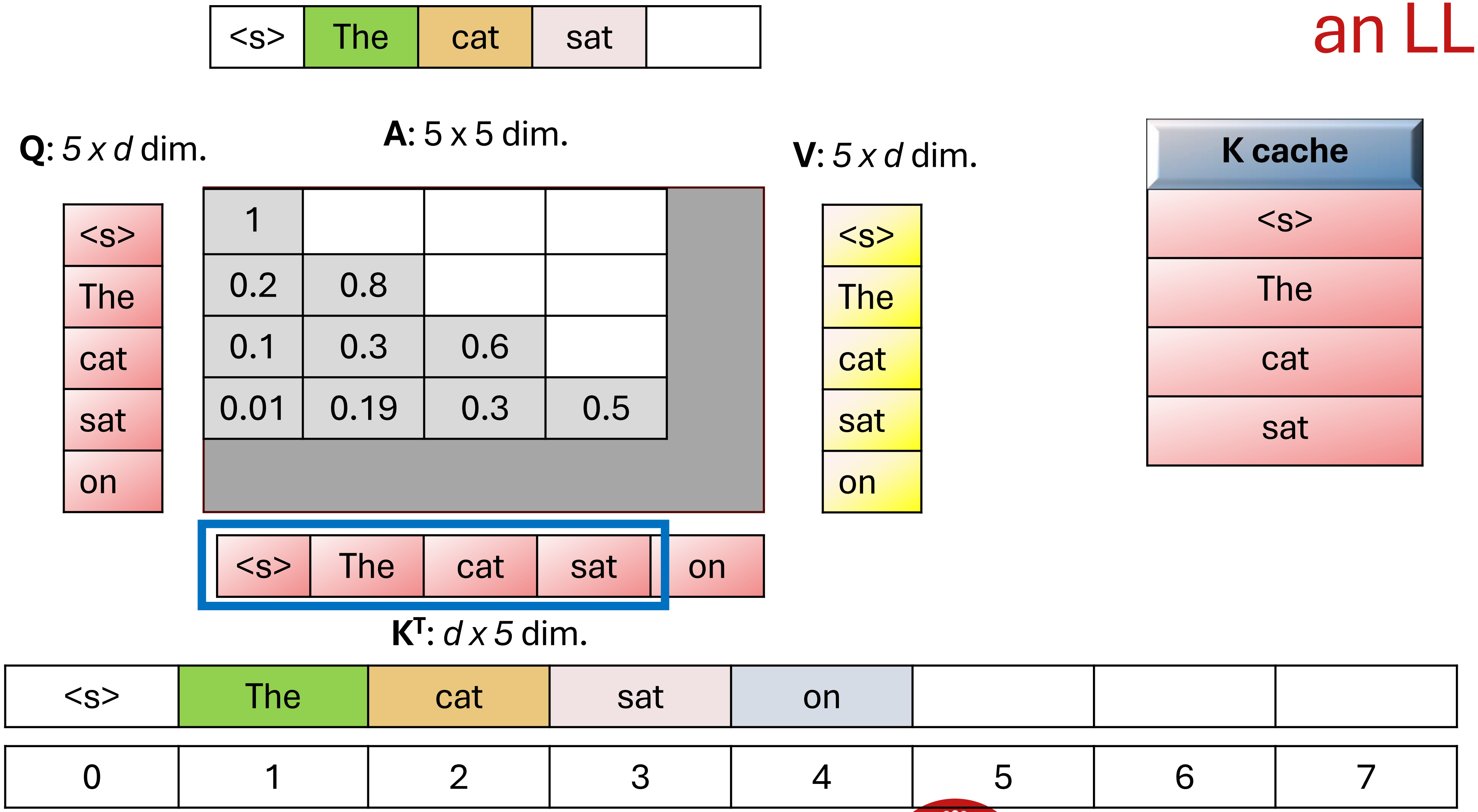
<s>
The
cat
sat
on

<s>	The	cat	sat	on

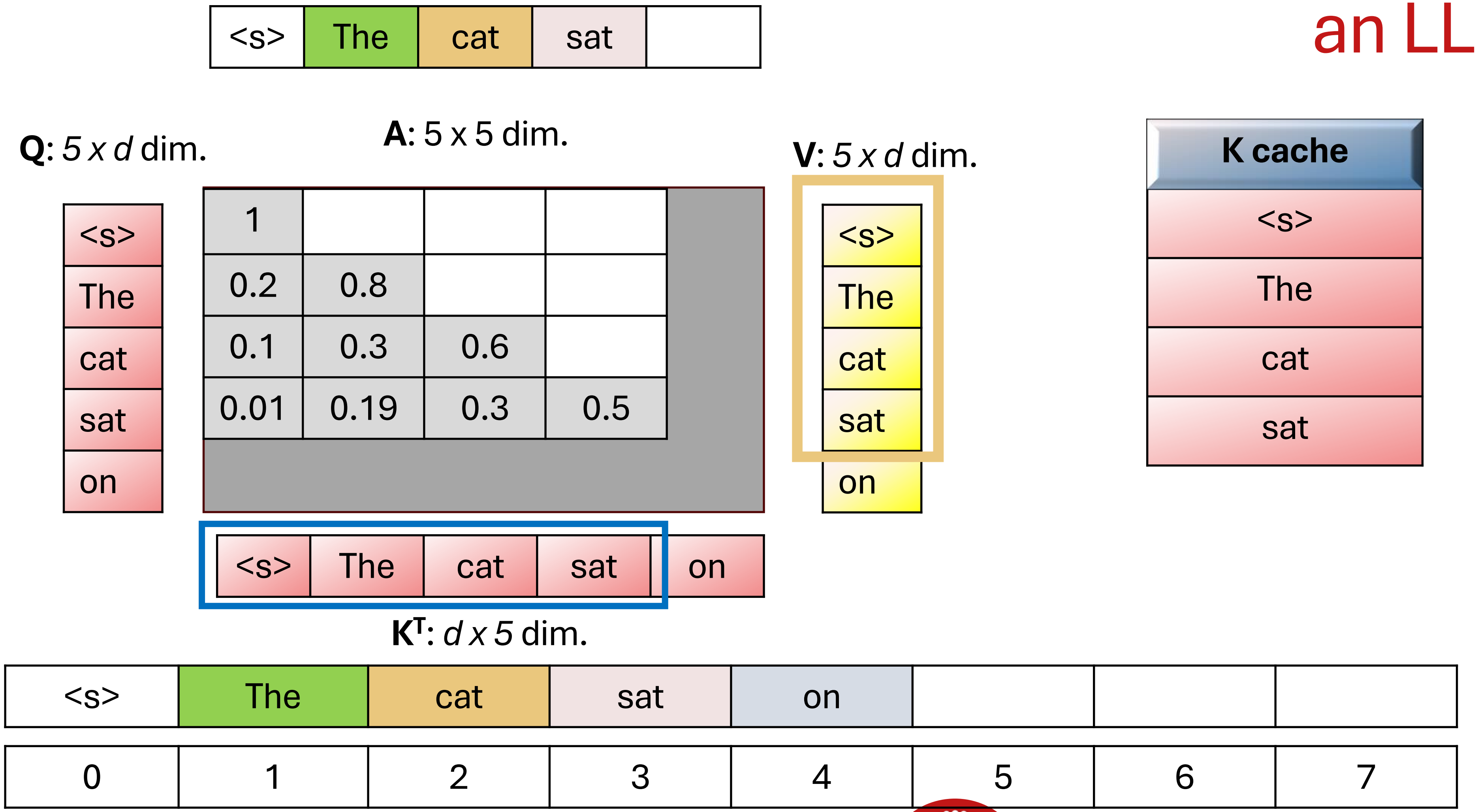
K^T: *d x 5* dim.

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

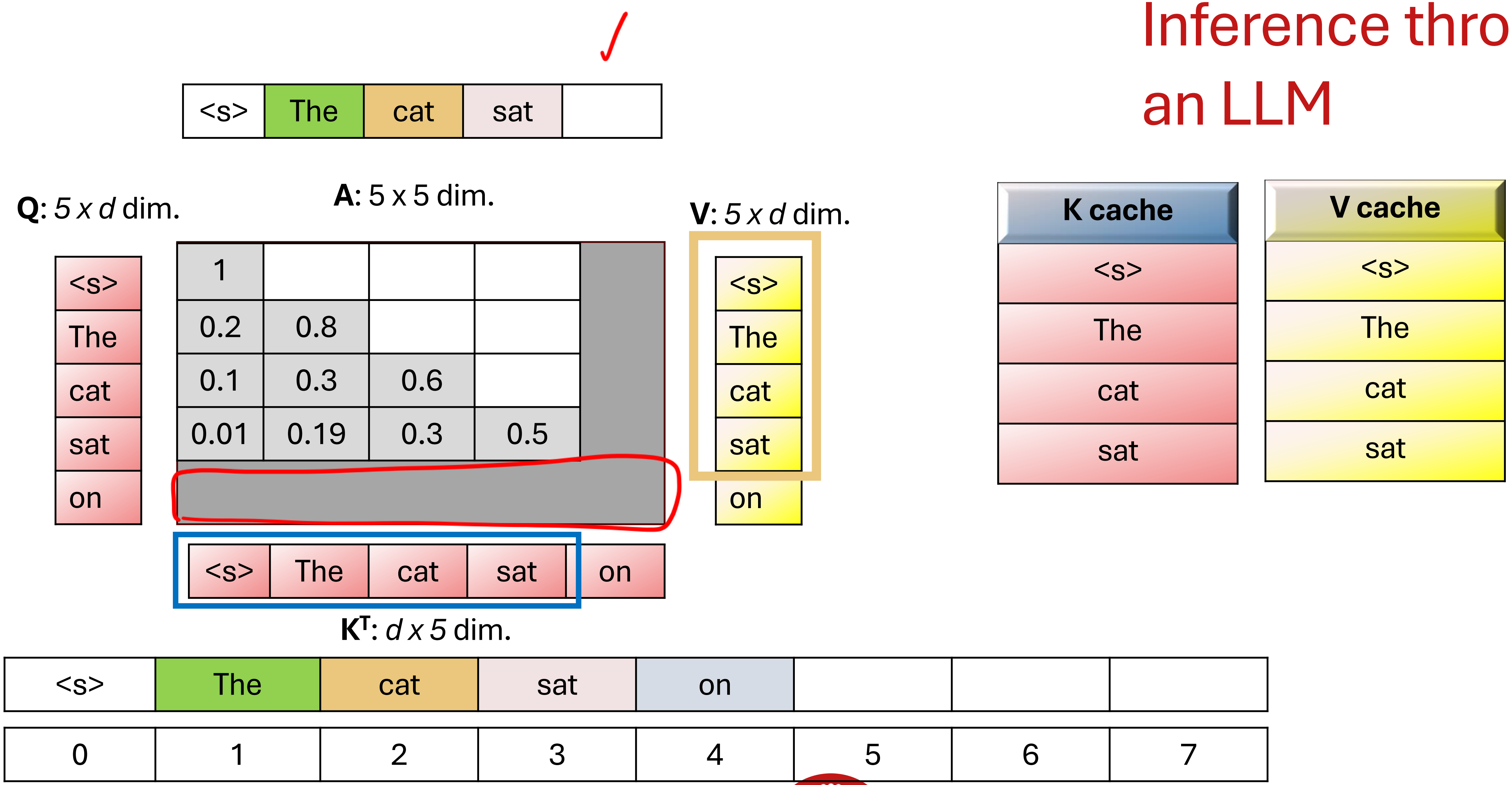
Inference through an LLM



Inference through an LLM



Inference through an LLM



Inference through an LLM

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
$\boxed{\text{on}}$

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	

V: *5 x d* dim.

<s>
The
cat
sat
on

<s>	The	cat	sat	on

K^T: *d x 5* dim.

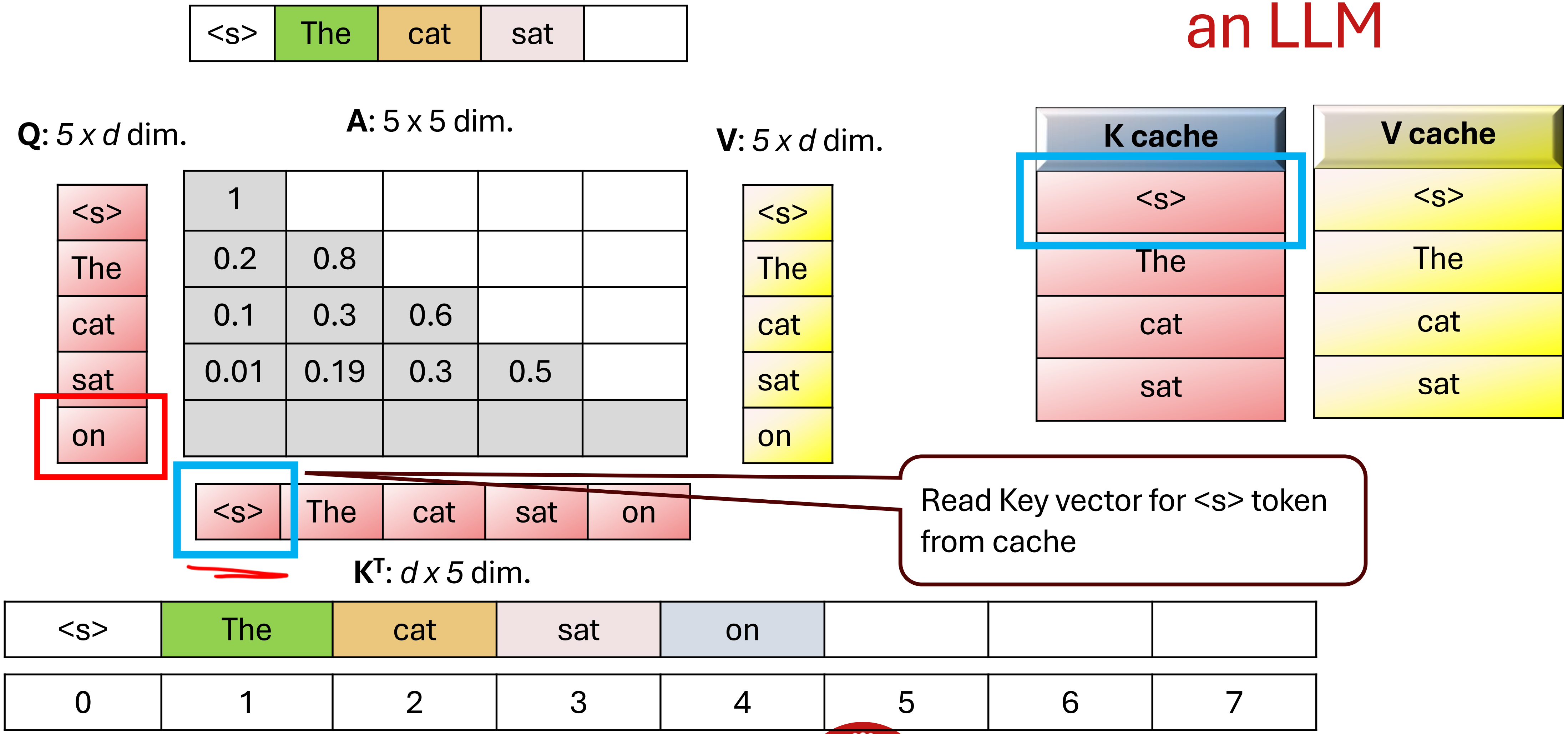
K cache
<s>
The
cat
sat

V cache
<s>
The
cat
sat

Compute Query vector for token on

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM



Inference through an LLM

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	
$q_4 k_0^T$				

<s>	The	cat	sat	on

K^T: *d x 5* dim.

V: *5 x d* dim.

<s>
The
cat
sat
on

K cache
<s>
The
cat
sat

V cache
<s>
The
cat
sat

Dot product to compute attention score b/w query “on” and key “<s>”

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	
$q_4 k_0^T$	$q_4 k_1^T$			

<s>	The	cat	sat	on

K^{T}: *d x 5* dim.

V: *5 x d* dim.

<s>
The
cat
sat
on

K cache
<s>
The
cat
sat

V cache
<s>
The
cat
sat

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	
$q_4 k_0^T$	$q_4 k_1^T$	$q_4 k_2^T$		

<s>	The	cat	sat	on

K^T: *d x 5* dim.

V: *5 x d* dim.

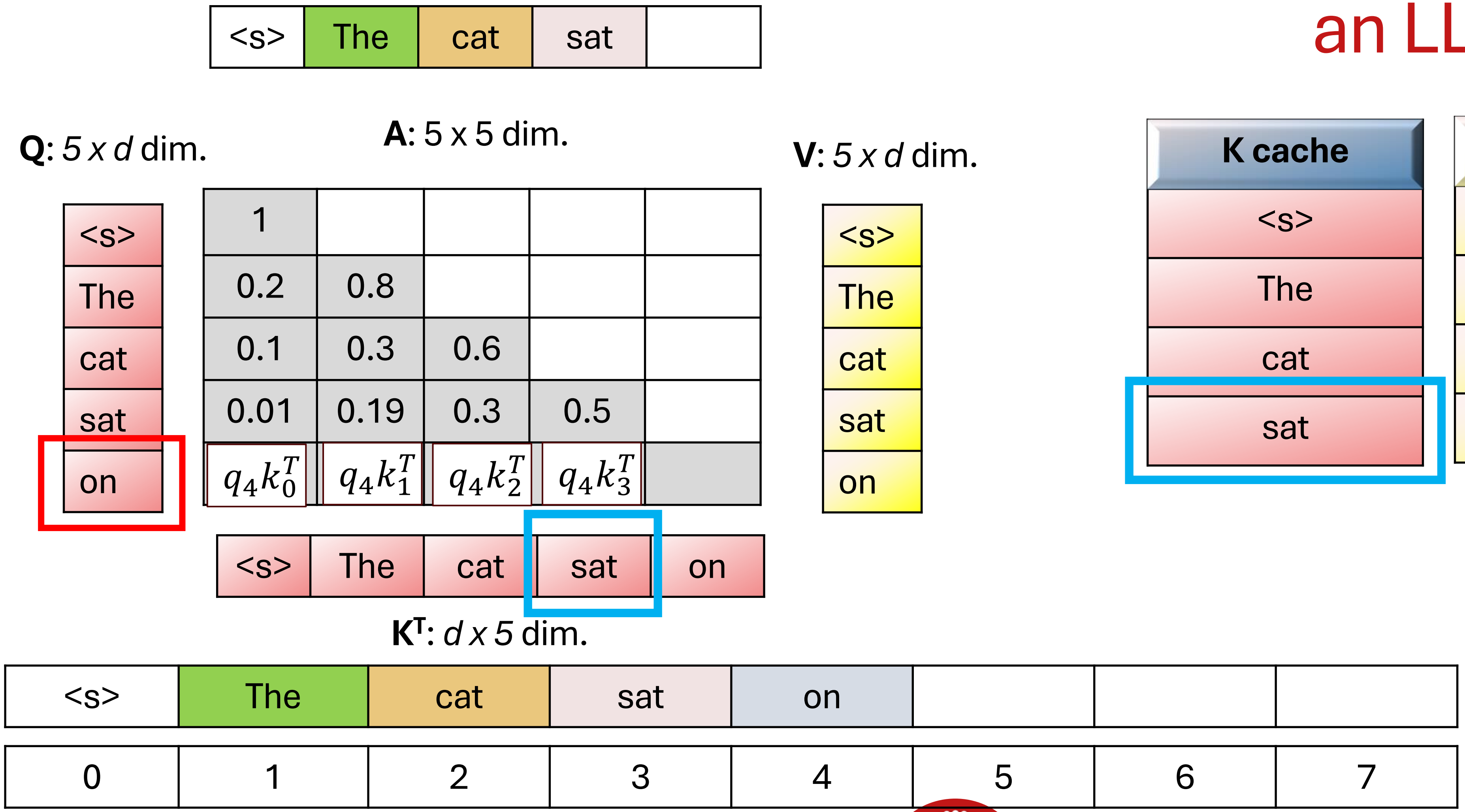
<s>
The
cat
sat
on

K cache
<s>
The
cat
sat

V cache
<s>
The
cat
sat

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM



Inference through an LLM

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	
$q_4 k_0^T$	$q_4 k_1^T$	$q_4 k_2^T$	$q_4 k_3^T$	$q_4 k_4^T$

<s>	The	cat	sat	on

K^T: *d x 5* dim.

V: *5 x d* dim.

<s>
The
cat
sat
on

K cache
<s>
The
cat
sat

V cache
<s>
The
cat
sat

Compute the Key emb. of token “on”

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	
$q_4k_0^T$	$q_4k_1^T$	$q_4k_2^T$	$q_4k_3^T$	$q_4k_4^T$

<s>	The	cat	sat	on

K^T: *d x 5* dim.

V: *5 x d* dim.

<s>
The
cat
sat
on

K cache
<s>
The
cat
sat
on

V cache
<s>
The
cat
sat

Add the Key emb. of token “on” to the cache

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	
0.03	0.07	0.1	0.3	0.4

<s>	The	cat	sat	on

K^T: *d x 5* dim.

V: *5 x d* dim.

<s>
The
cat
sat
on

Convert attn. scores to probability

K cache
<s>
The
cat
sat
on

V cache
<s>
The
cat
sat

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	
0.03	0.07	0.1	0.3	0.4

<s>	The	cat	sat	on

K^T: *d x 5* dim.

V: *5 x d* dim.

<s>
The
cat
sat
on

K cache
<s>
The
cat
sat
on

V cache
<s>
The
cat
sat

Load Value vectors from V cache

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

$$\frac{(q_4 k_0^T) v_0}{S_4}$$

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	
0.03	0.07	0.1	0.3	0.4

<s>	The	cat	sat	on

K^T: *d x 5* dim.

V: *5 x d* dim.

<s>
The
cat
sat
on

K cache
<s>
The
cat
sat
on

V cache
<s>
The
cat
sat

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

$$\frac{(q_4 k_0^T) v_0}{S_4} + \frac{(q_4 k_1^T) v_1}{S_4}$$

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	
0.03	$\boxed{0.07}$	0.1	0.3	0.4

<s>	The	cat	sat	on

K^T: *d x 5* dim.

V: *5 x d* dim.

<s>
$\boxed{\text{The}}$
cat
sat
on

K cache
<s>
The
cat
sat
on

V cache
<s>
$\boxed{\text{The}}$
cat
sat

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

$$\frac{(q_4k_0^T)v_0}{S_4}+\frac{(q_4k_1^T)v_1}{S_4}+\frac{(q_4k_2^T)v_2}{S_4}$$

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	
0.03	0.07	0.1	0.3	0.4

<s>	The	cat	sat	on

K^T: *d x 5* dim.

V: *5 x d* dim.

<s>
The
cat
sat
on

K cache
<s>
The
cat
sat
on

V cache
<s>
The
cat
sat

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

$$\frac{(q_4 k_0^T) v_0}{S_4} + \frac{(q_4 k_1^T) v_1}{S_4} + \frac{(q_4 k_2^T) v_2}{S_4} + \frac{(q_4 k_3^T) v_3}{S_4}$$

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	
0.03	0.07	0.1	$\boxed{0.3}$	0.4

<s>	The	cat	sat	on

K^T: *d x 5* dim.

V: *5 x d* dim.

<s>
The
cat
$\boxed{\text{sat}}$
on

K cache
<s>
The
cat
sat
on

V cache
<s>
The
cat
$\boxed{\text{sat}}$

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

$$\frac{(q_4 k_0^T)v_0}{S_4} + \frac{(q_4 k_1^T)v_1}{S_4} + \frac{(q_4 k_2^T)v_2}{S_4} + \frac{(q_4 k_3^T)v_3}{S_4}$$

<s>	The	cat	sat	

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	
0.03	0.07	0.1	0.3	$\boxed{0.4}$

<s>	The	cat	sat	on

K^T: *d x 5* dim.

V: *5 x d* dim.

<s>
The
cat
sat
$\boxed{\text{on}}$

K cache
<s>
The
cat
sat
on

V cache
<s>
The
cat
sat

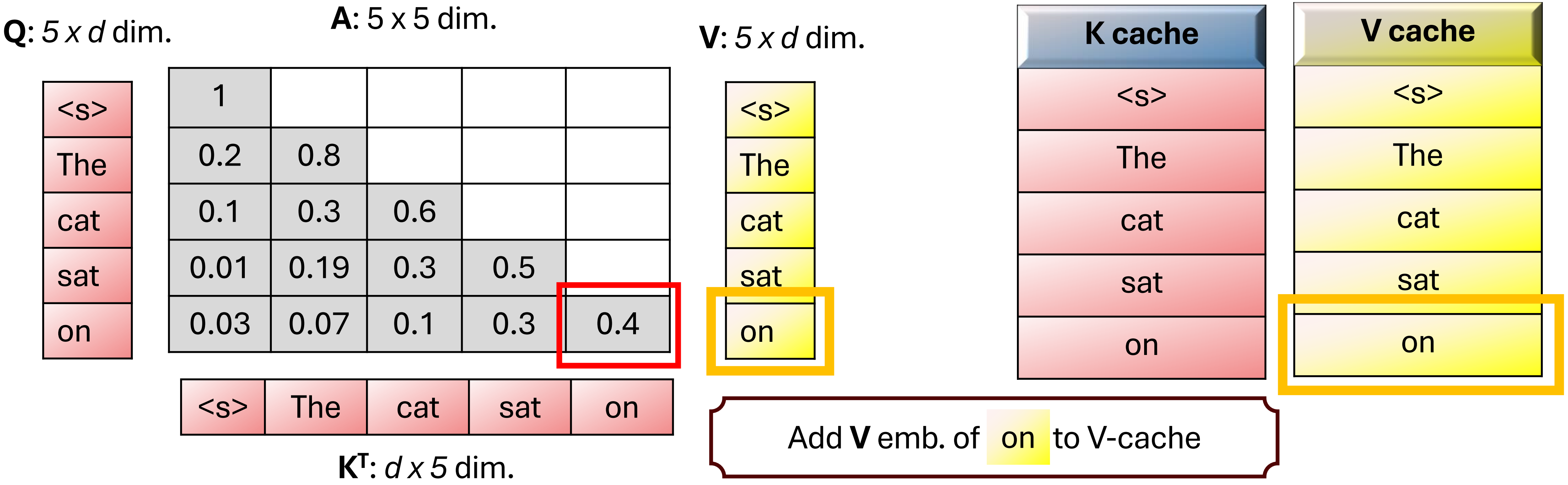
Compute **V** emb. of on

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

$$\frac{(q_4 k_0^T) v_0}{S_4} + \frac{(q_4 k_1^T) v_1}{S_4} + \frac{(q_4 k_2^T) v_2}{S_4} + \frac{(q_4 k_3^T) v_3}{S_4}$$

<s>	The	cat	sat	



<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

Inference through an LLM

$$\frac{(q_4k_0^T)v_0}{S_4}+\frac{(q_4k_1^T)v_1}{S_4}+\frac{(q_4k_2^T)v_2}{S_4}+\frac{(q_4k_3^T)v_3}{S_4}+\frac{(q_4k_4^T)v_4}{S_4}$$

<s>	The	cat	sat	on

Q: *5 x d* dim.

<s>
The
cat
sat
on

A: 5 x 5 dim.

1				
0.2	0.8			
0.1	0.3	0.6		
0.01	0.19	0.3	0.5	
0.03	0.07	0.1	0.3	0.4

<s>	The	cat	sat	on

K^T: *d x 5* dim.

V: *5 x d* dim.

<s>
The
cat
sat
on

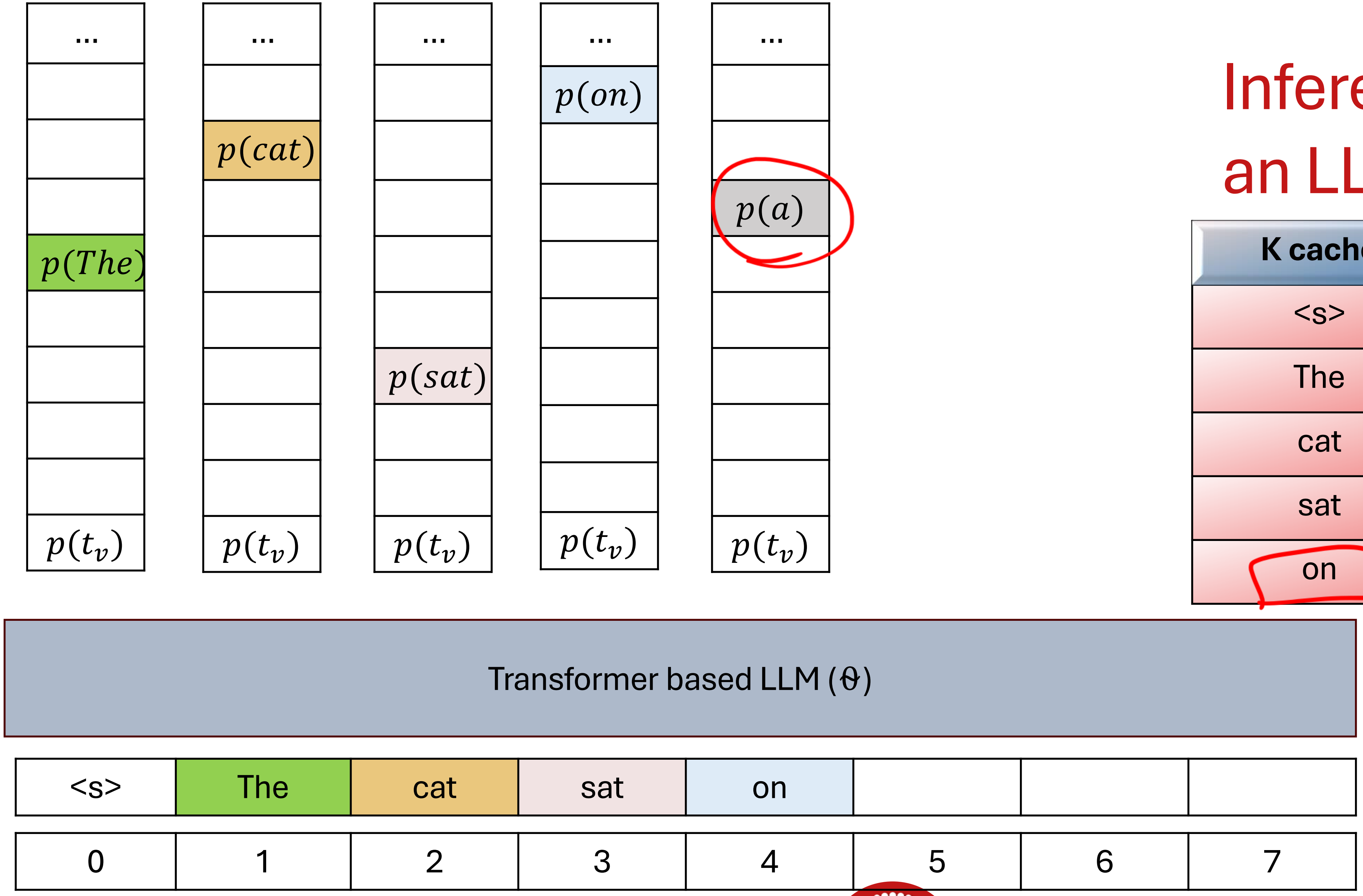
K cache
<s>
The
cat
sat
on

V cache
<s>
The
cat
sat
on

We get output emb. of on

<s>	The	cat	sat	on			
0	1	2	3	4	5	6	7

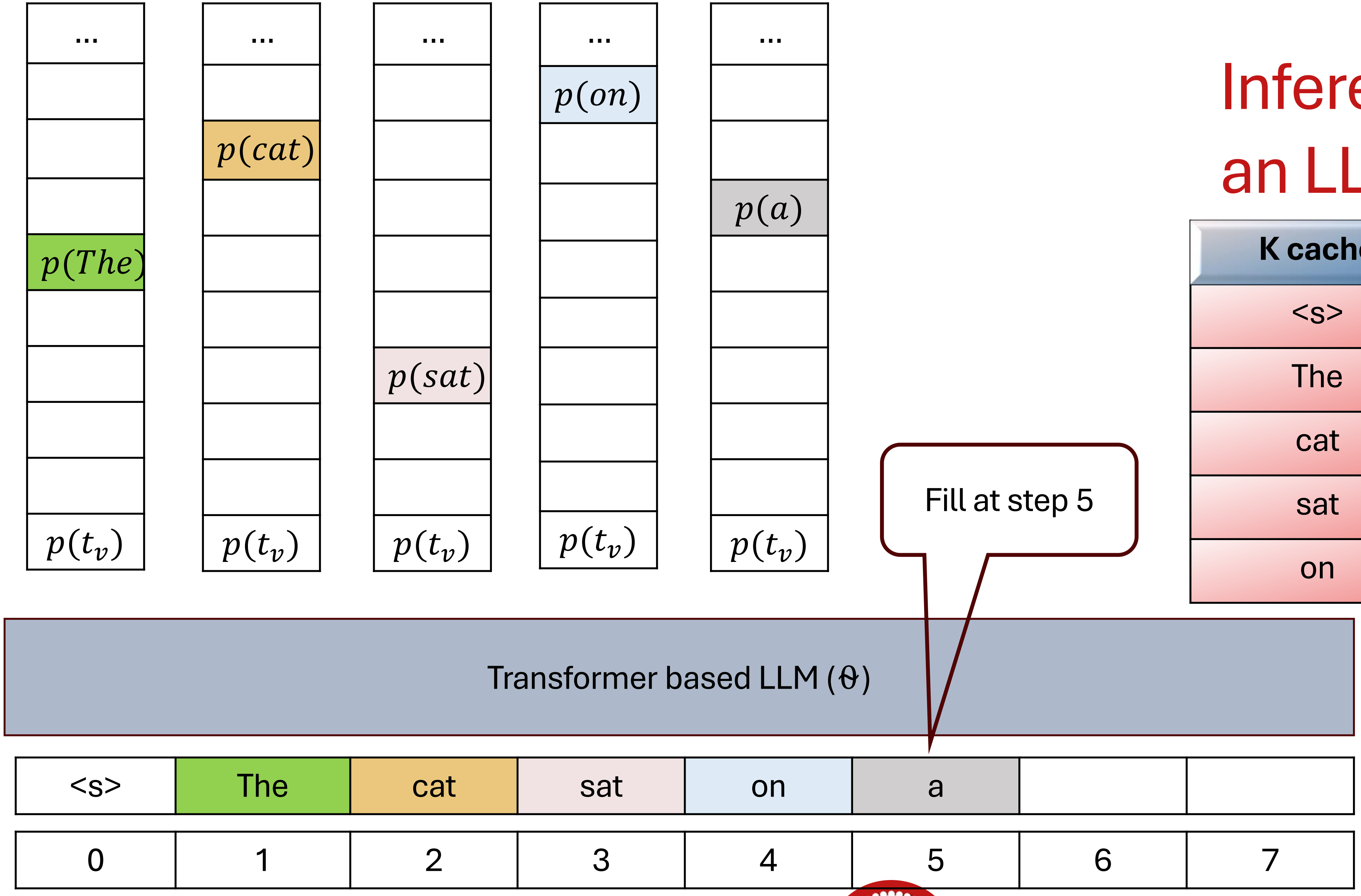
Inference through an LLM



K cache
<s>
The
cat
sat
on

V cache
<s>
The
cat
sat
on

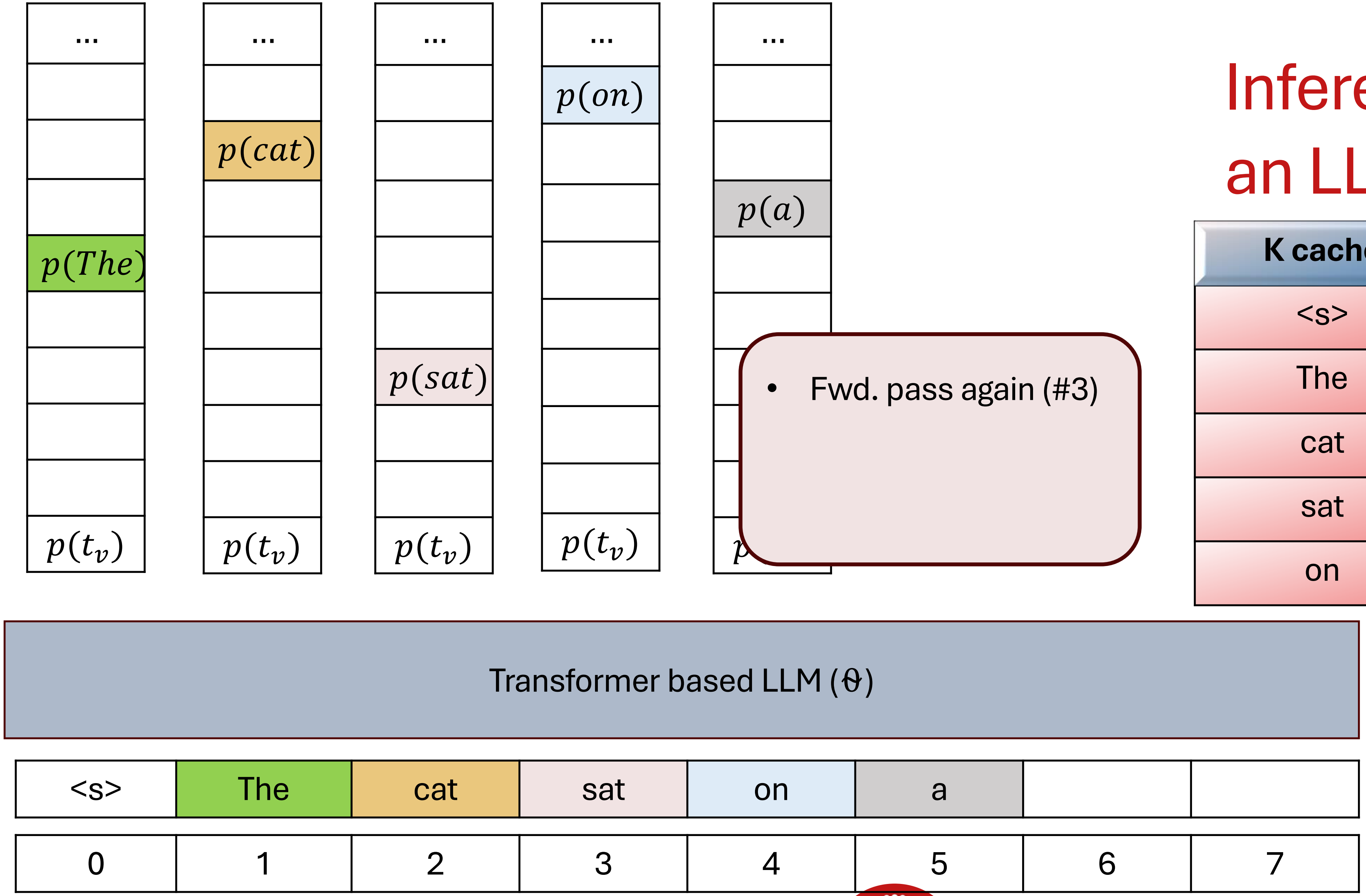
Inference through an LLM

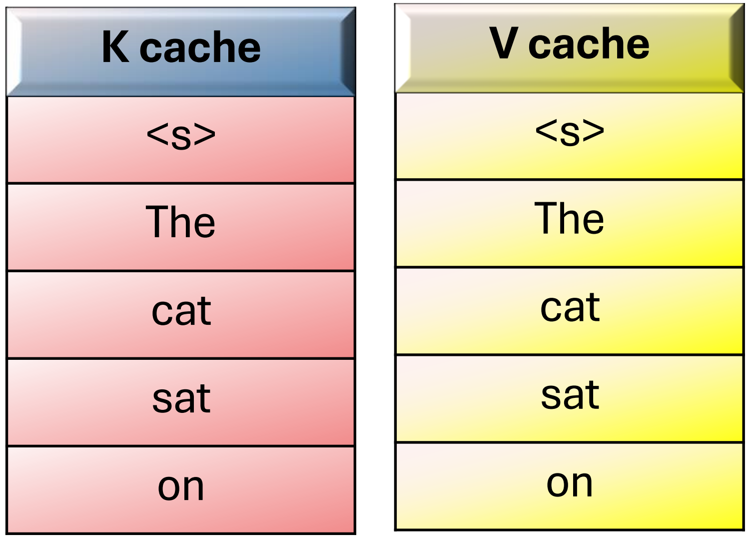


K cache
<s>
The
cat
sat
on

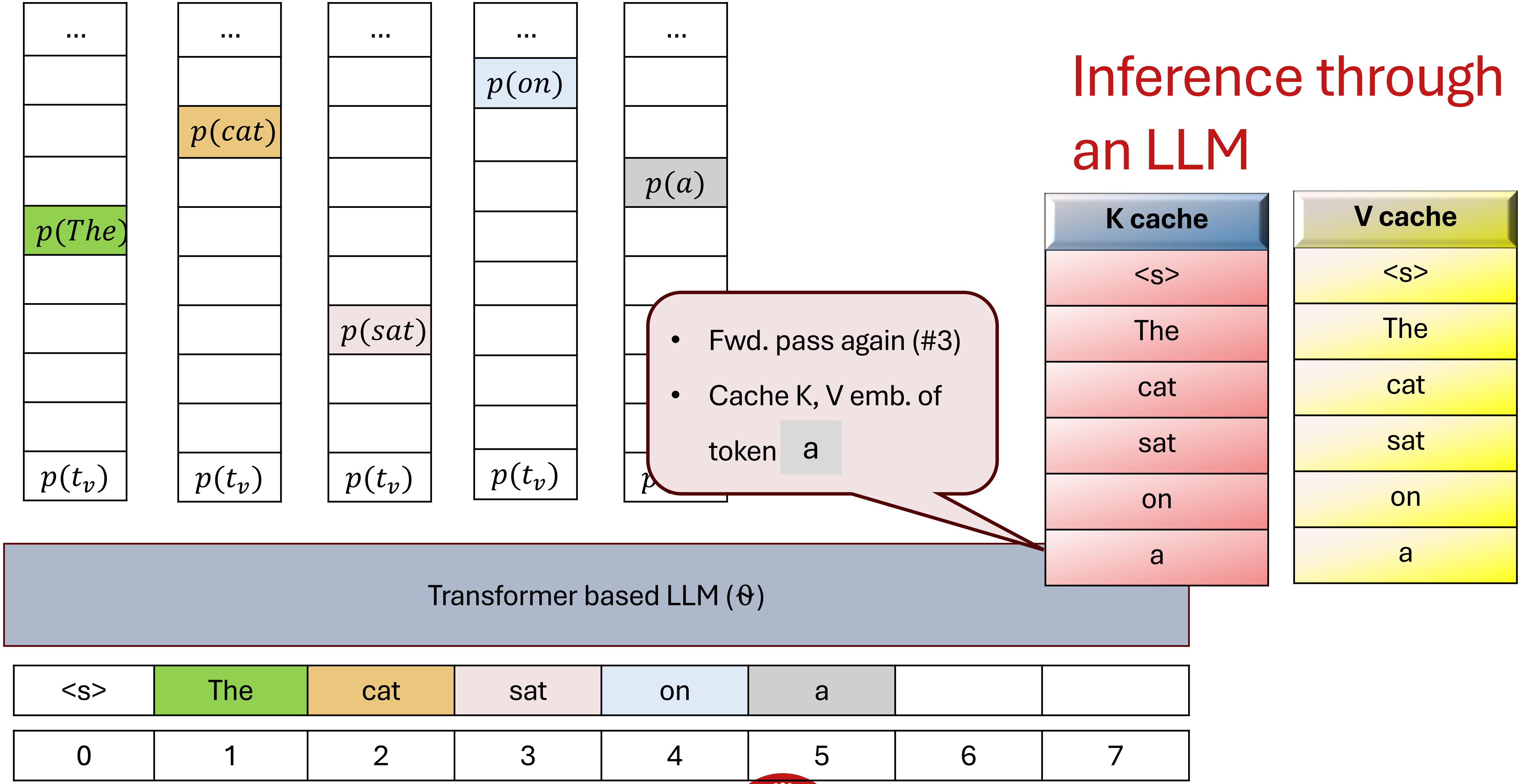
V cache
<s>
The
cat
sat
on

Inference through an LLM

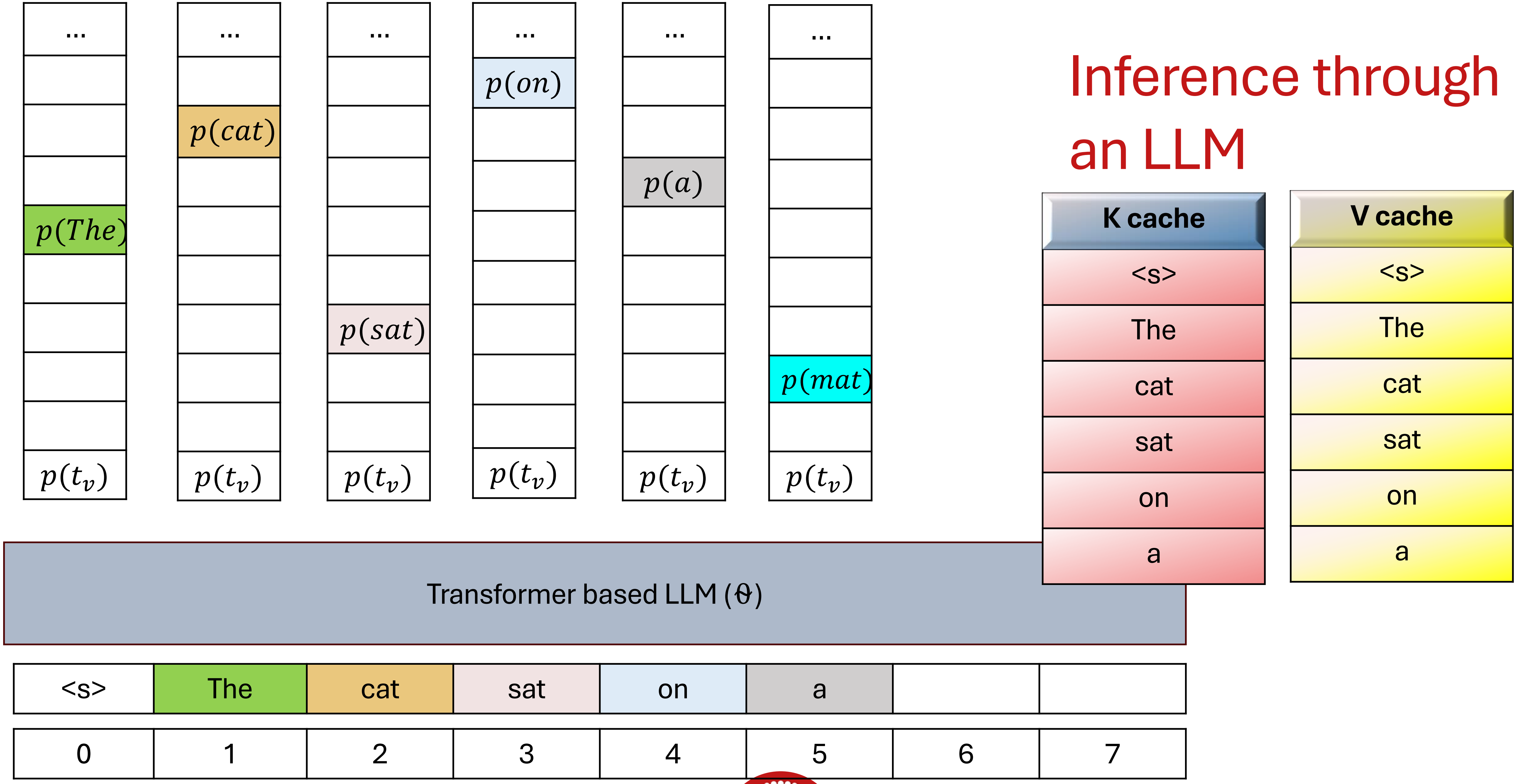




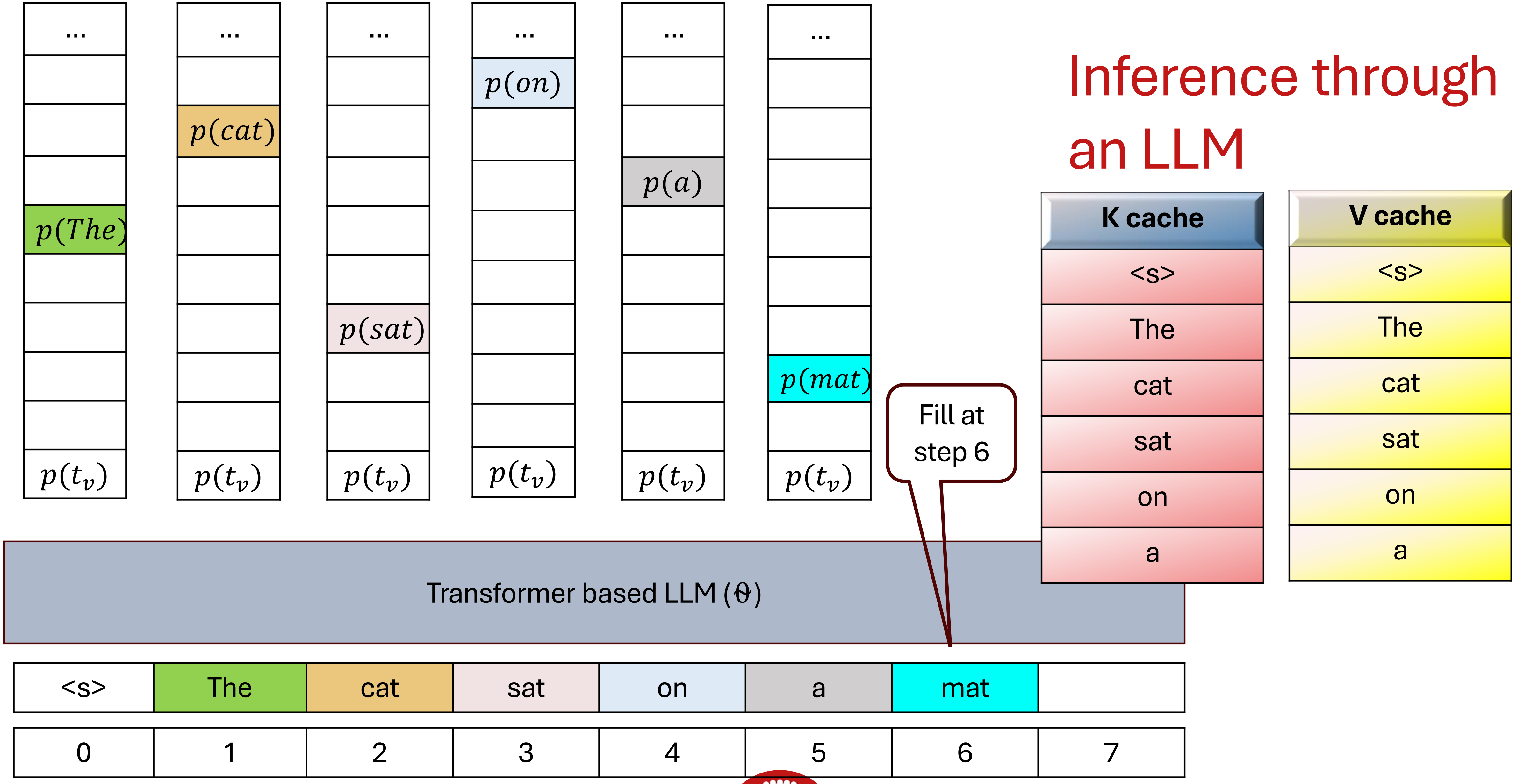
Inference through an LLM



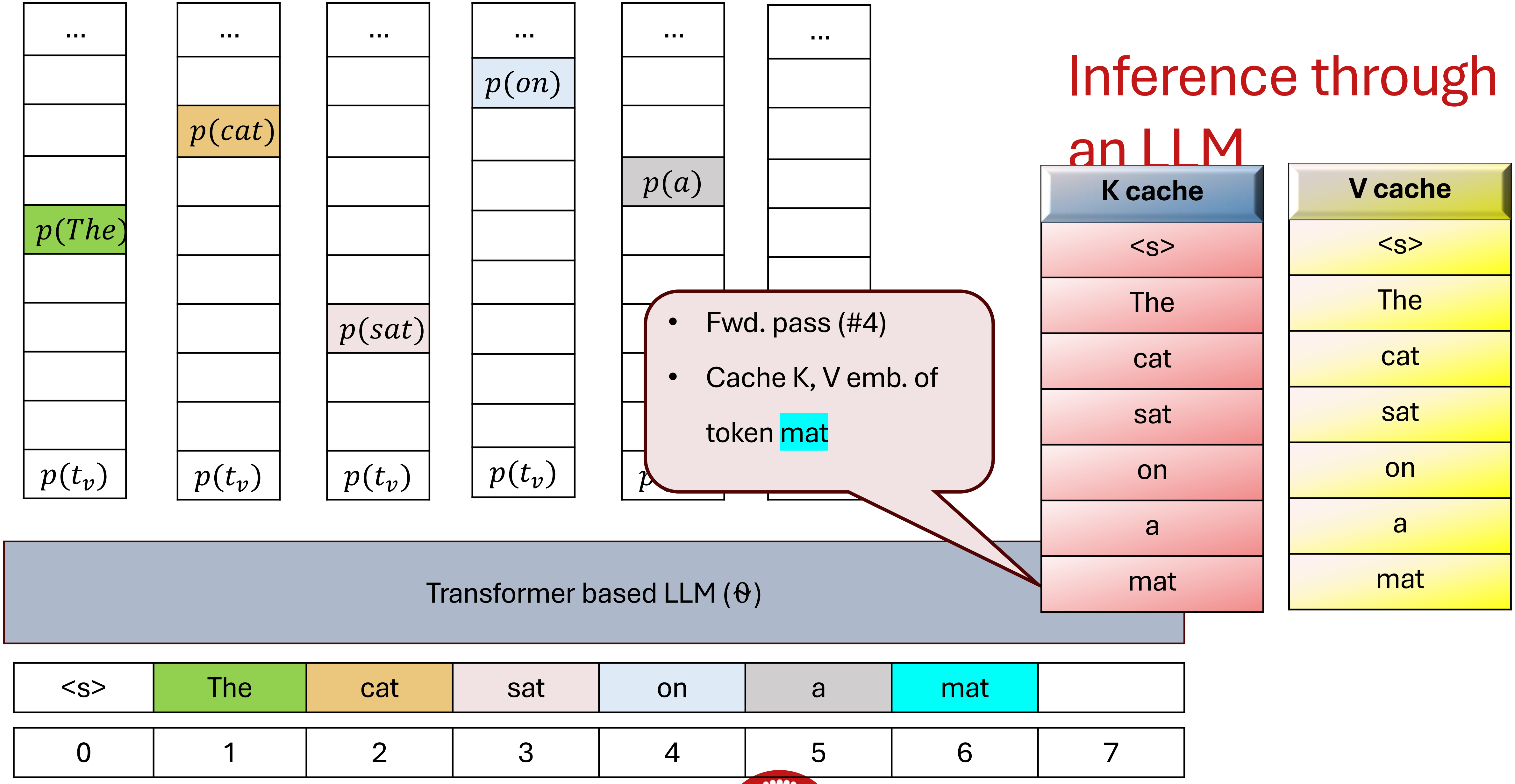
Inference through an LLM



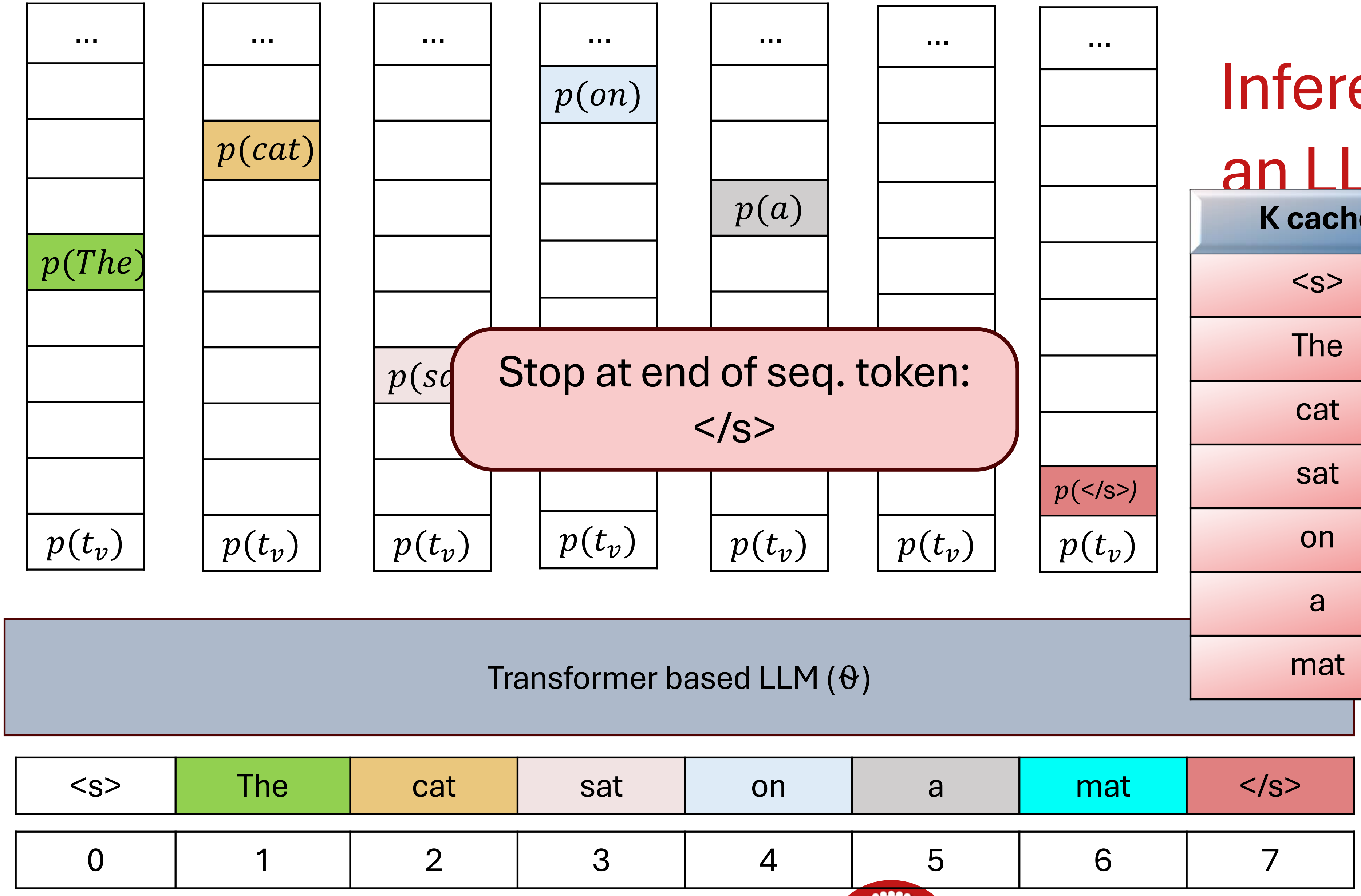
Inference through an LLM



Inference through an LLM

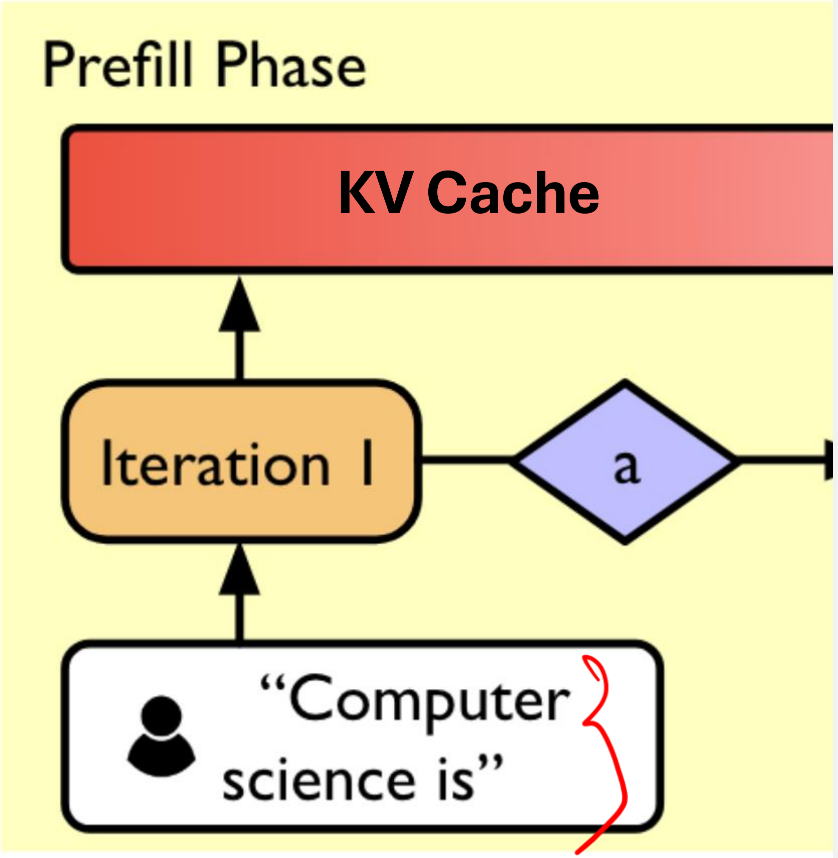


Inference through an LLM



K cache	V cache
<s>	<s>
The	The
cat	cat
sat	sat
on	on
a	a
mat	mat

Two stages of LLM inference

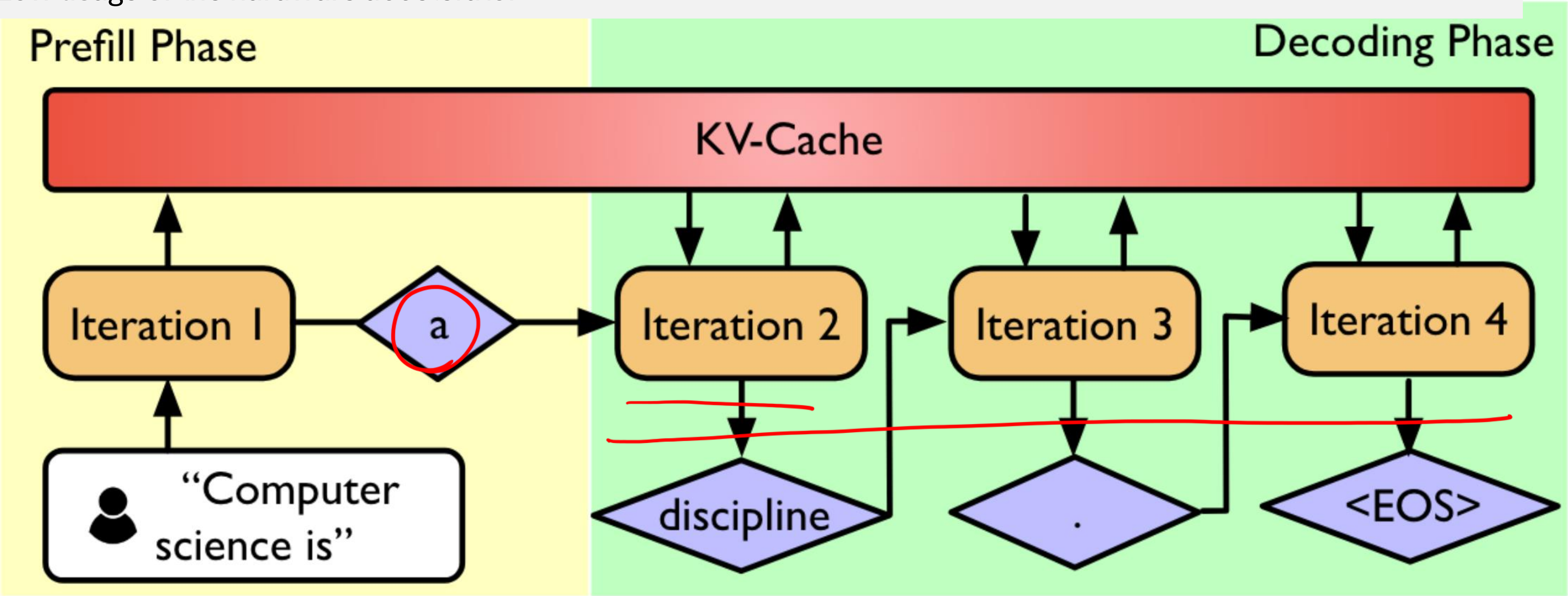


Content credits: Li et al, 2024 LLM Inference Serving: Survey of Recent Advances

- 1st forward pass (Pre-fill step) **Highly parallel**
  - The entire prompt is embedded and encoded – High latency
  - Multi-head attention computes the keys and values (KV)
  - Large matrix multiplication, high usage of the hardware accelerator

Remaining forward passes (Output generation): **sequential**

- The answer is generated one token at a time – Low latency per step
- Each generated token is appended to the previous input
- The process is repeated until the stopping criteria is met (max. length or EOS)
- Low usage of the hardware accelerator



Content credits: Li et al, 2024 LLM Inference Serving: Survey of Recent Advances and Opportunities

Inference through an LLM

- 1st forward pass (Pre-fill step) **Highly parallel**
  - The entire prompt is embedded and encoded – High latency
  - Multi-head attention computes the keys and values (KV)
  - Large matrix multiplication, high usage of the hardware accelerator
- Remaining forward passes (Output generation): **sequential**
  - The answer is generated one token at a time – Low latency per step
  - Each generated token is appended to the previous input
  - The process is repeated until the stopping criteria is met (max. length or EOS)
  - Low usage of the hardware accelerator

Content credits: https://www.slideshare.net/slideshow/julien-simon-deep-dive-optimizing-llm-inference-69d3/270921961

Memory Usage of KV cache

$$2 * \underline{precision} * \underline{N_{layers}} * \underline{d_{model}} * \underline{seqlen} * \underline{batch}$$

k/v

2	:	Two matrices for K and V
$precision$	:	bytes per parameter (e.g. 4 for fp32)
N_{layers}	:	layers in the model
d_{model}	:	dimension of embeddings
$seqlen$	:	length of context in tokens
$batch$	:	batch size

Content credits: https://www.youtube.com/watch?v=80bIUggRJf4&t=1s&ab_channel=EfficientNLP

Memory Usage of KV cache: **Example OPT-13B**

$$2 * precision * N_{layers} * d_{model} * seqlen * batch$$

2	:	Two matrices for K and V	**2 (KV)**
$precision$	:	bytes per parameter (e.g. 4 for fp32)	**2 bytes (fp16)** ✓
N_{layers}	:	layers in the model	**40 layers** ✓
d_{model}	:	dimension of embeddings	**5120 dim.**
$seqlen$	:	length of context in tokens	**2048 tokens**
$batch$	:	batch size	**10**

Content credits: https://www.youtube.com/watch?v=80bIUggRJf4&t=1s&ab_channel=EfficientNLP

Memory Usage of KV cache: **Example OPT-13B**

$$2 * precision * N_{layers} * d_{model} * seqlen * batch$$

KV Cache: 17 GB

Model Size: 2*13 = 26 GB

On a 40GB A100

- 65% (26GB) used by model parameters
- ~30% (12 GB) available for KV cache
- Expected throughput ~ 8 batch size of 2048 tokens

2 (KV)
2 bytes (fp16)
40 layers
5120 dim.
2048 tokens
10

Content credits: https://www.youtube.com/watch?v=80bIUggRJf4&t=1s&ab_channel=EfficientNLP

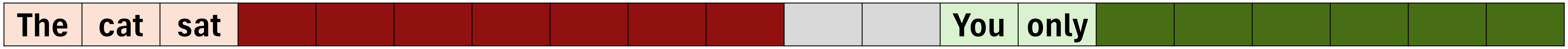
Memory Management of KV Cache

Prompt A : *"The cat sat"*
Max Tokens: *2048*

Prompt B : *"You only"*
Max Tokens: *512*

Memory Management of KV Cache

Tensor operations require contiguous memory



Tensor operations require contiguous memory

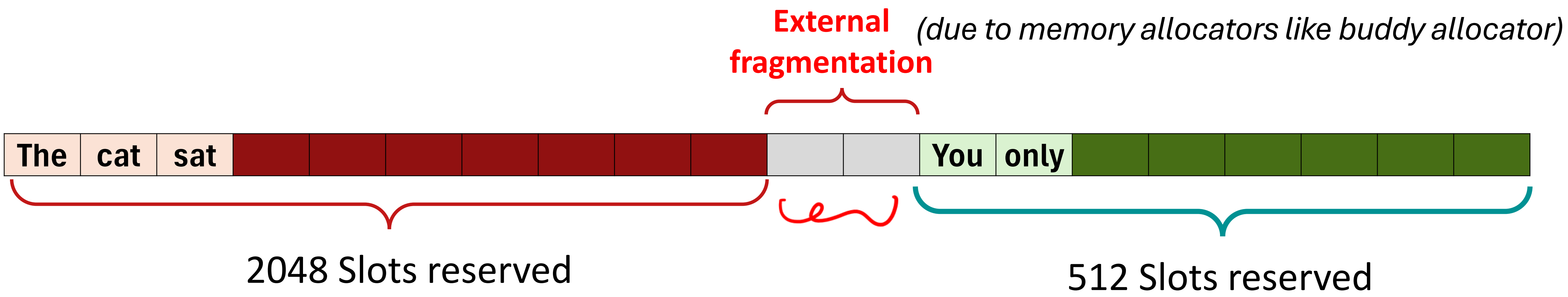
2048 Slots reserved

512 Slots reserved

Prompt A : ***"The cat sat"***
Max Tokens: *2048*

Prompt B : ***"You only"***
Max Tokens: *512*

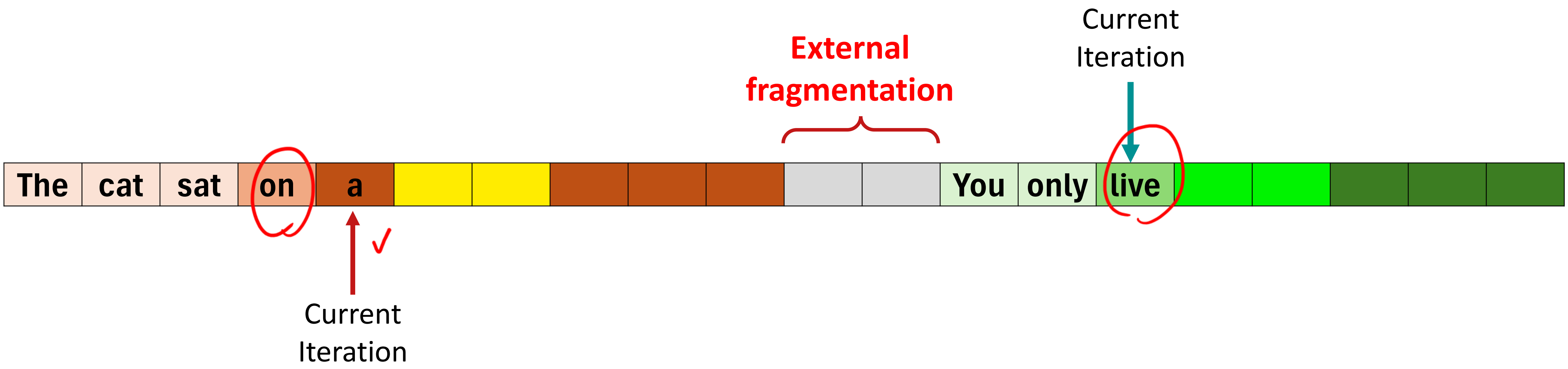
Memory Management of KV Cache



Prompt A : ***"The cat sat"***
Max Tokens: *2048*

Prompt B : ***"You only"***
Max Tokens: *512*

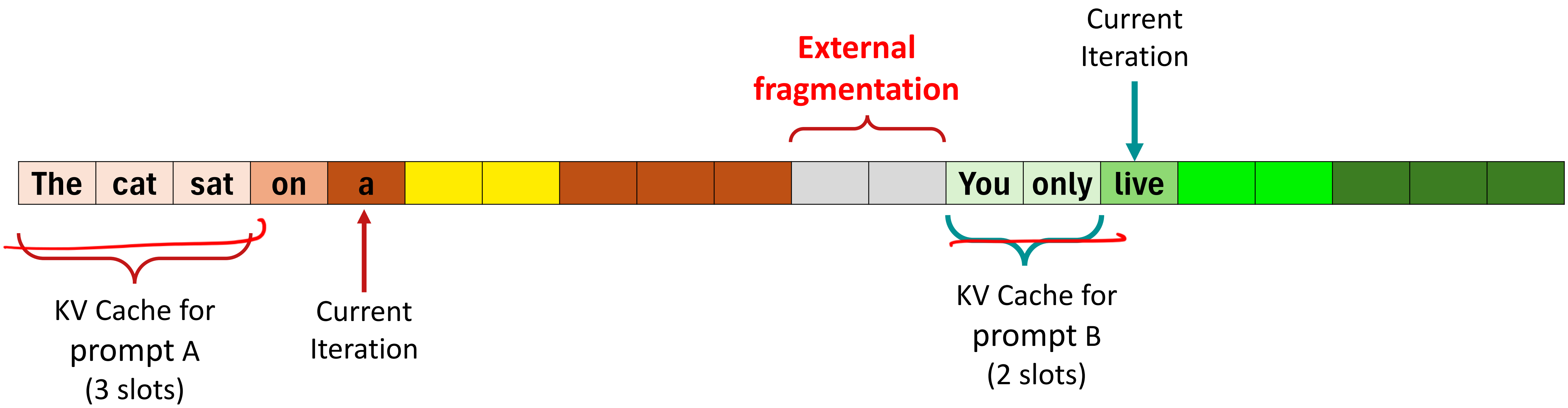
Memory Management of KV Cache



Prompt A : *"The cat sat"*
Max Tokens: *2048*

Prompt B : *"You only"*
Max Tokens: *512*

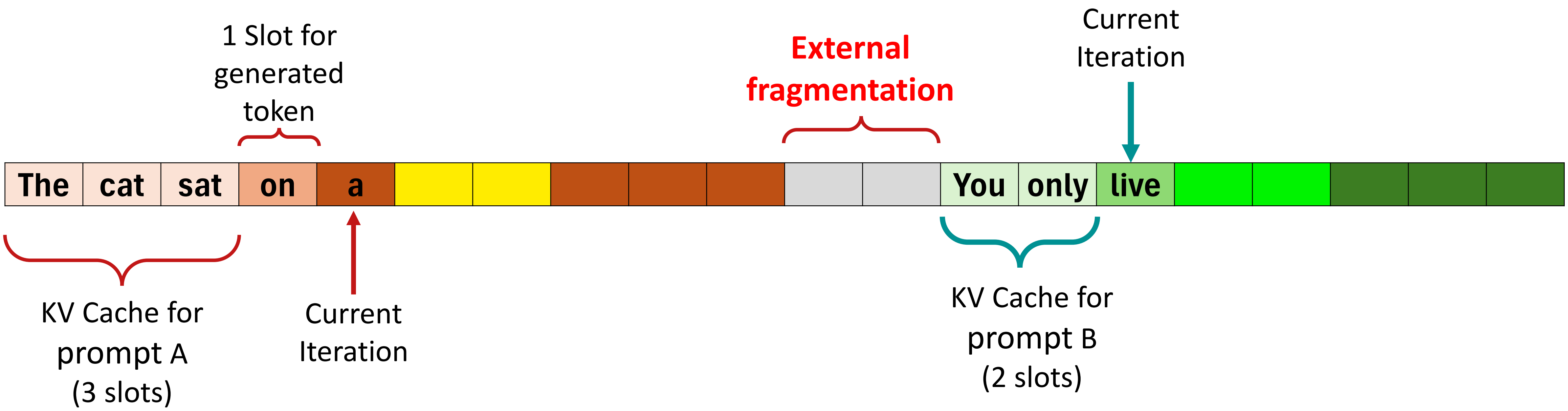
Memory Management of KV Cache



Prompt A : *"The cat sat"*
Max Tokens: *2048*

Prompt B : *"You only"*
Max Tokens: *512*

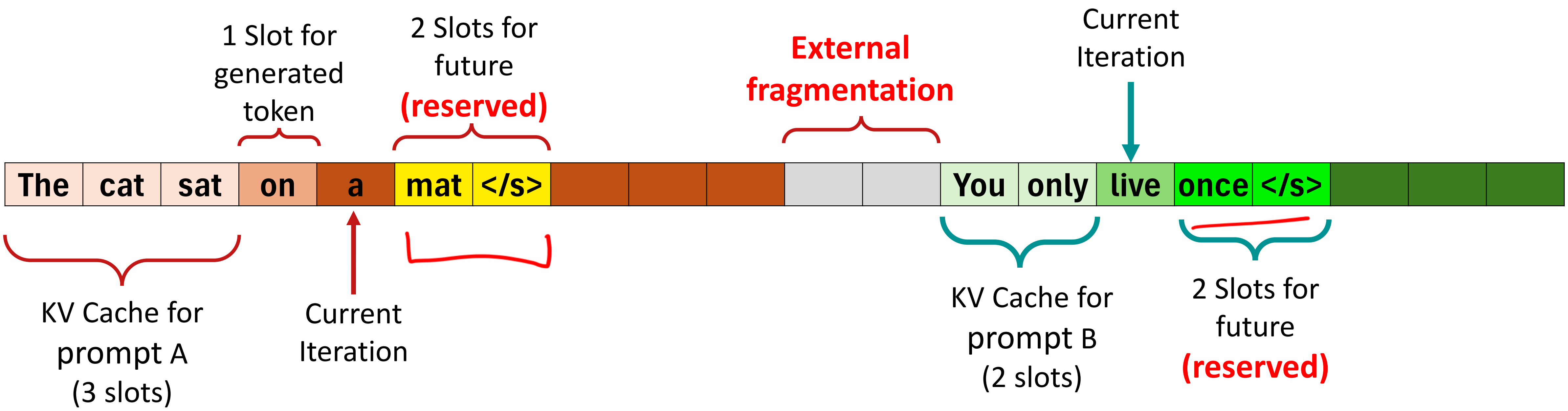
Memory Management of KV Cache



Prompt A : *"The cat sat"*
Max Tokens: *2048*

Prompt B : *"You only"*
Max Tokens: *512*

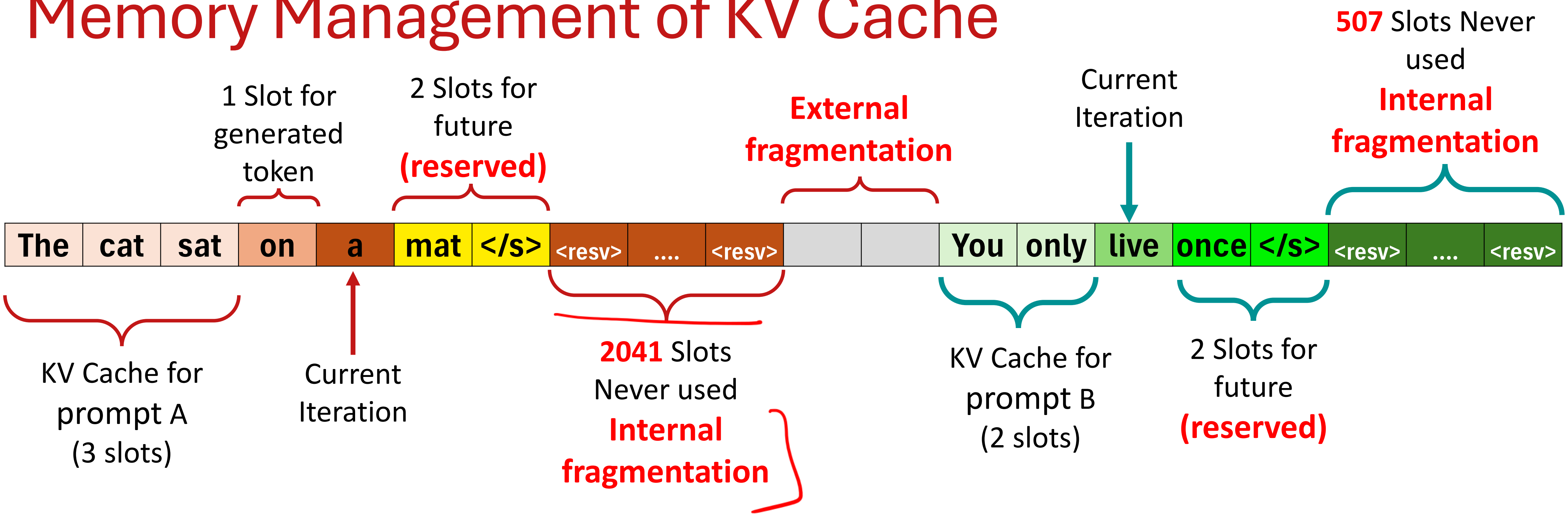
Memory Management of KV Cache



Prompt A : *"The cat sat"*
Max Tokens: *2048*

Prompt B : *"You only"*
Max Tokens: *512*

Memory Management of KV Cache



Prompt A : *"The cat sat"*
Max Tokens: *2048*

Prompt B : *"You only"*
Max Tokens: *512*

Memory Management of KV Cache

Chunk Pre-allocation scheme

- KV cache stored in contiguous memory
- Chunks of memory allocated statically, based on max. tokens.
- Actual input or eventual output length ignored while allocating memory

Memory Management of KV Cache

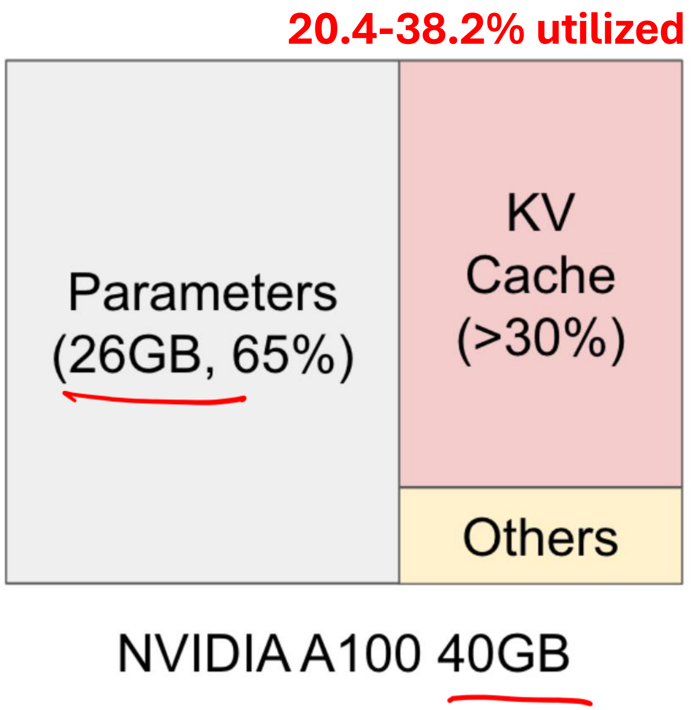
Chunk Pre-allocation scheme

- KV cache stored in contiguous memory
- Chunks of memory allocated statically, based on max. tokens.
- Actual input or eventual output length ignored while allocating memory

Results in 3 types of memory wastes –

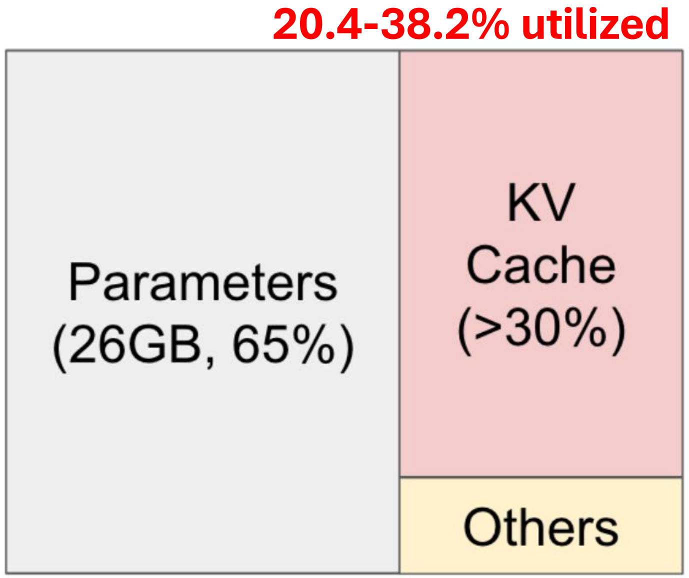
- **Reserved slots** for future tokens
- **Internal fragmentation** due to over-provisioning for maximum sequence lengths
- **External fragmentation** from the memory allocator.

Memory Layout for 13B-OPT model on A100 (40GB)

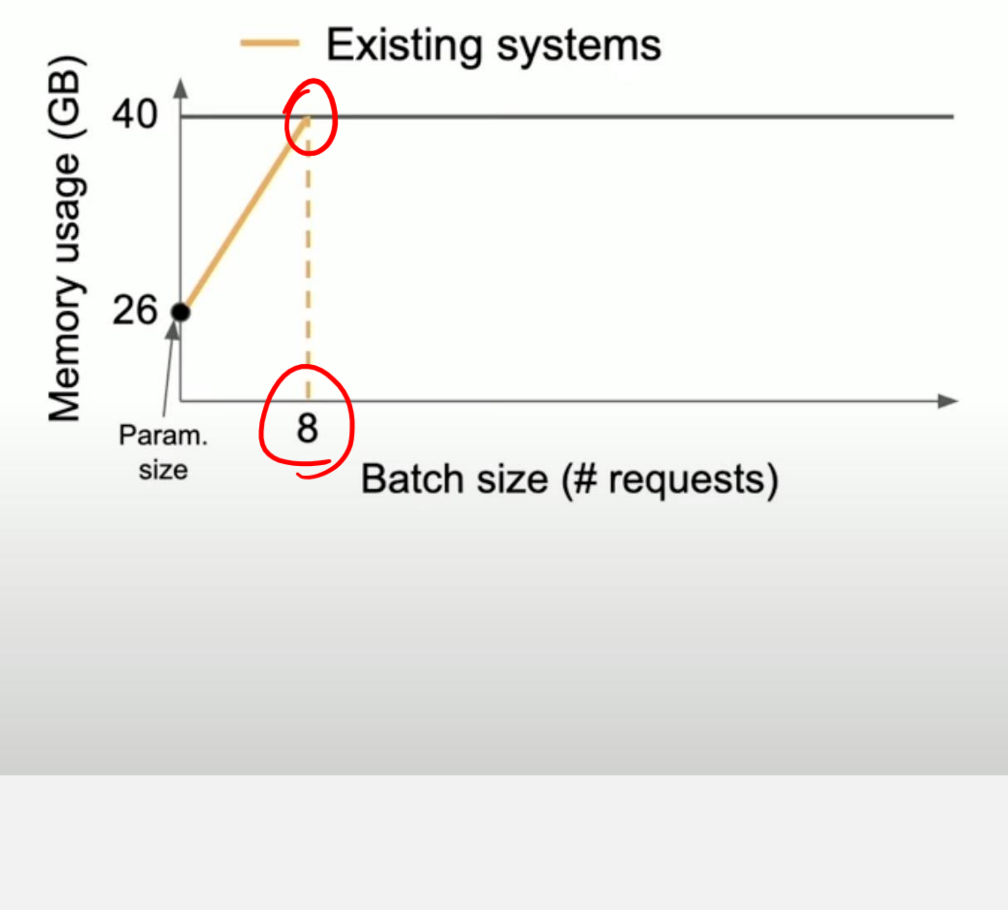


Content credits: https://www.youtube.com/watch?v=5ZlavKF_98U&t=1646s&ab_channel=Anyscale

Memory Layout for 13B-OPT model on A100 (40GB)



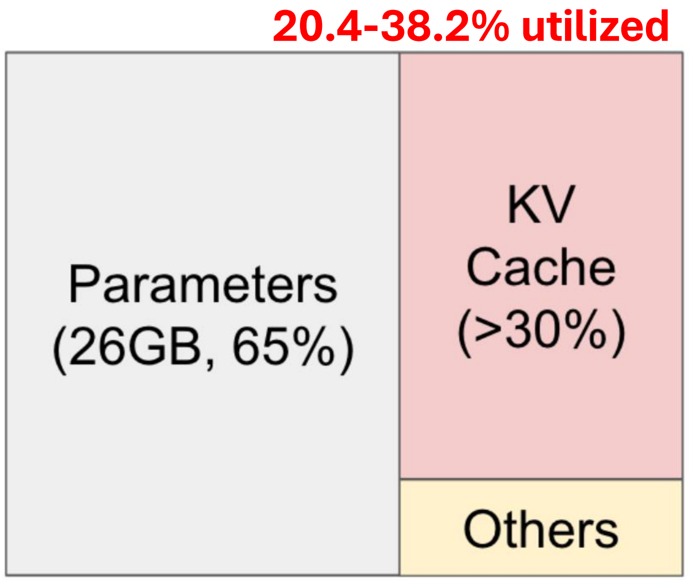
NVIDIA A100 40GB



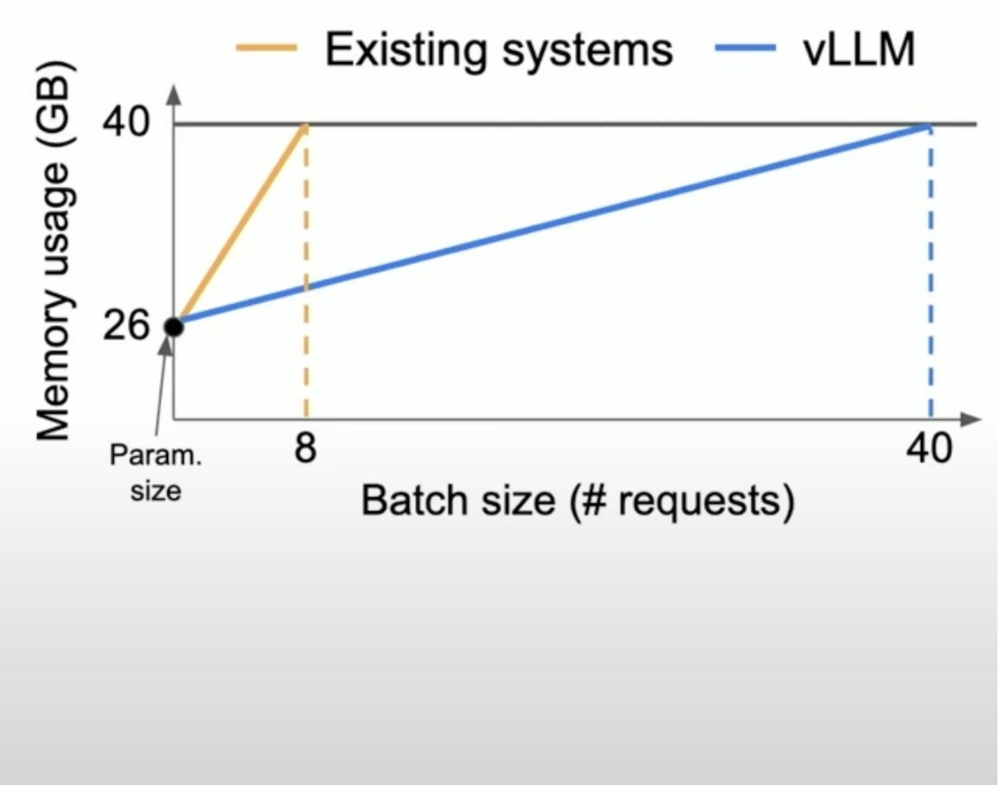
Existing systems

- max batch size - 8

Memory Layout for 13B-OPT model on A100 (40GB)



NVIDIA A100 40GB

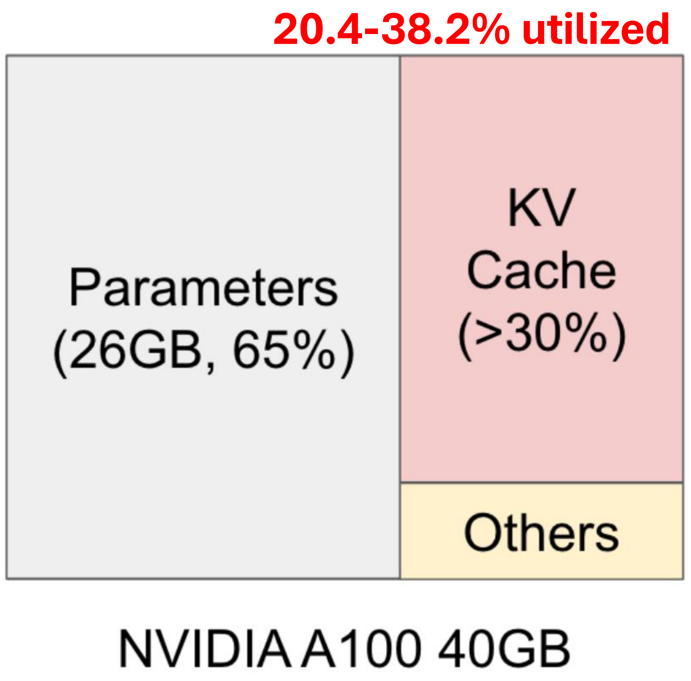


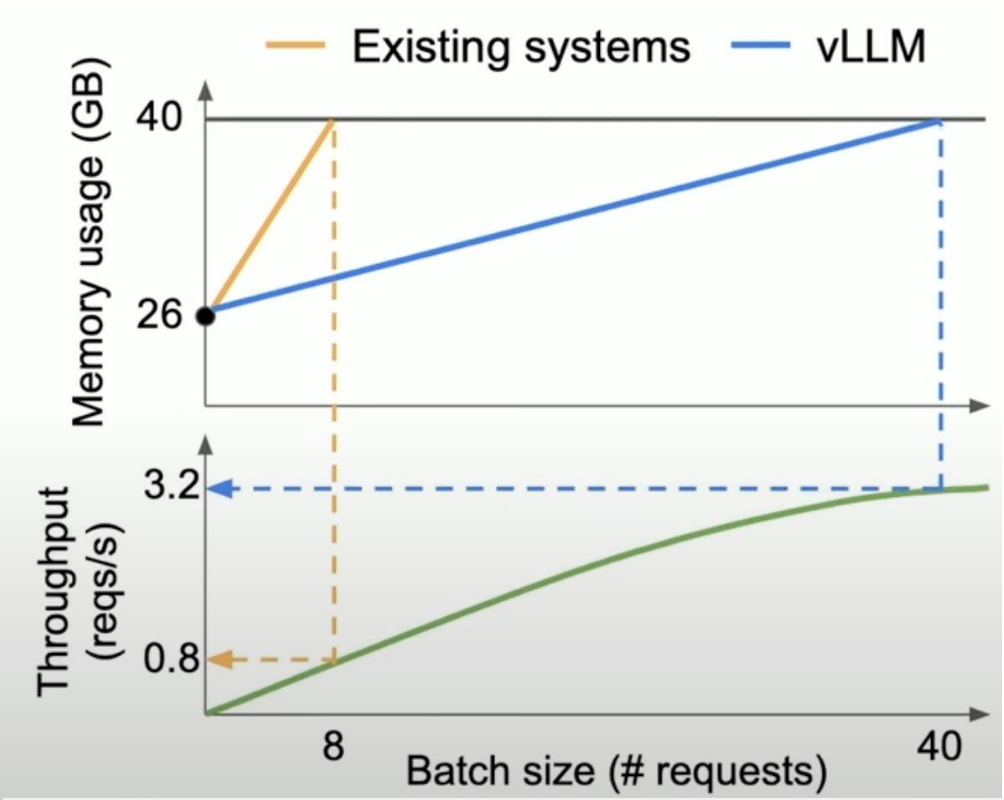
Existing systems
- max batch size - 8

vLLM (paged attention)
- Max batch size ~ 40

Content credits: https://www.youtube.com/watch?v=5ZlavKF_98U&t=1646s&ab_channel=Anyscale

Memory Layout for 13B-OPT model on A100 (40GB)





Existing systems

- max batch size - 8
- ~ 0.8 requests / sec

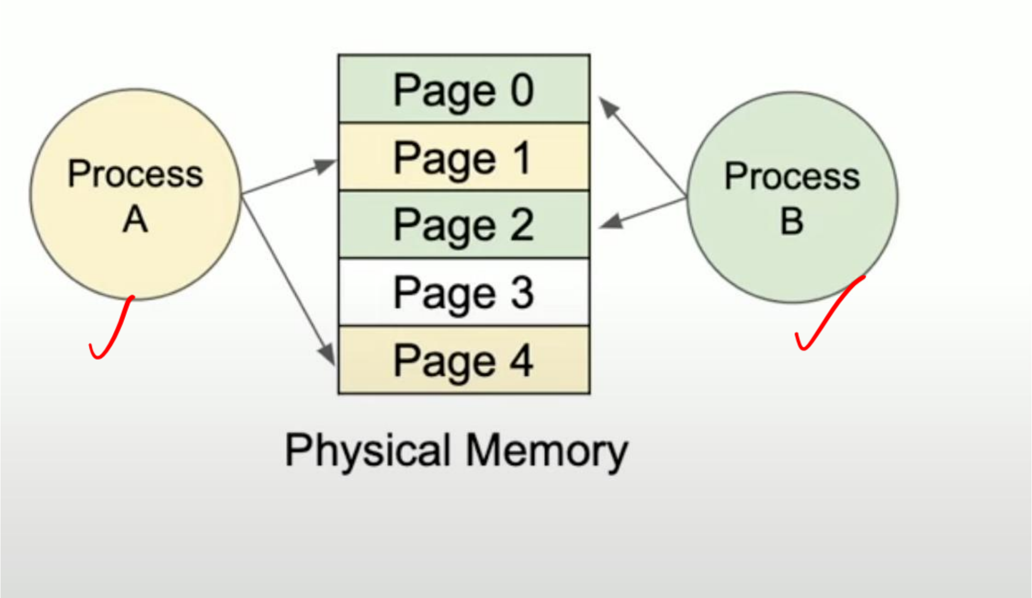
vLLM (paged attention)

- Max batch size ~ 38
- ~ 3.2 requests per sec

Content credits: https://www.youtube.com/watch?v=5ZlavKF_98U&t=1646s&ab_channel=Anyscale

vLLM: Efficient KV cache management

Inspired by Virtual memory and paging

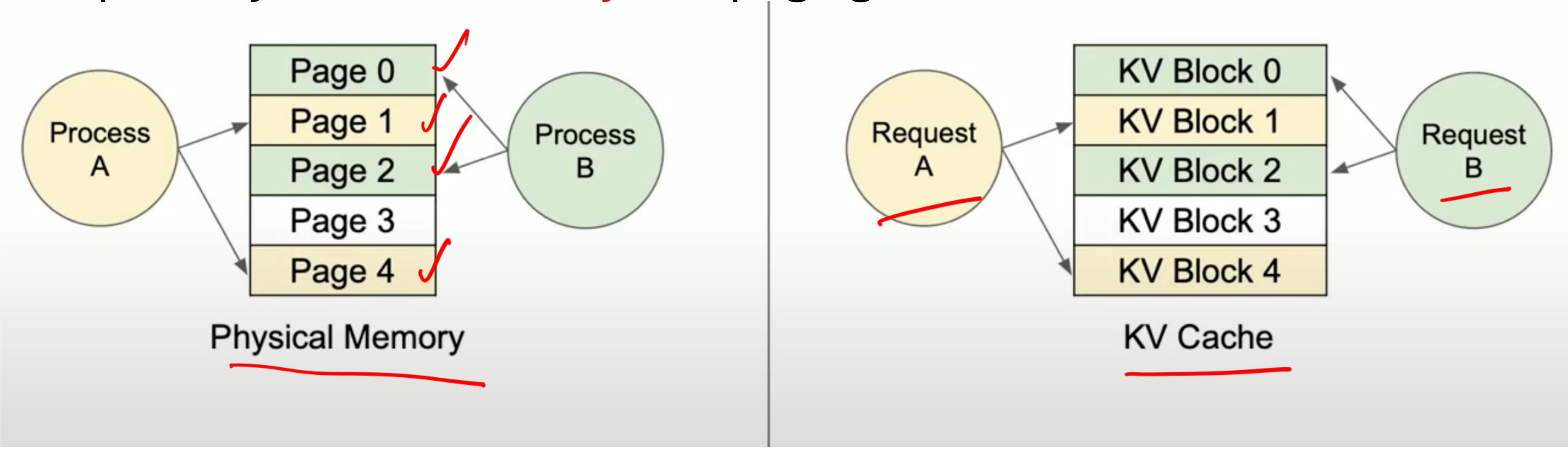


Memory management in OS

Content credits: https://www.youtube.com/watch?v=5ZlavKF_98U&t=1646s&ab_channel=Anyscale

vLLM: Efficient KV cache management

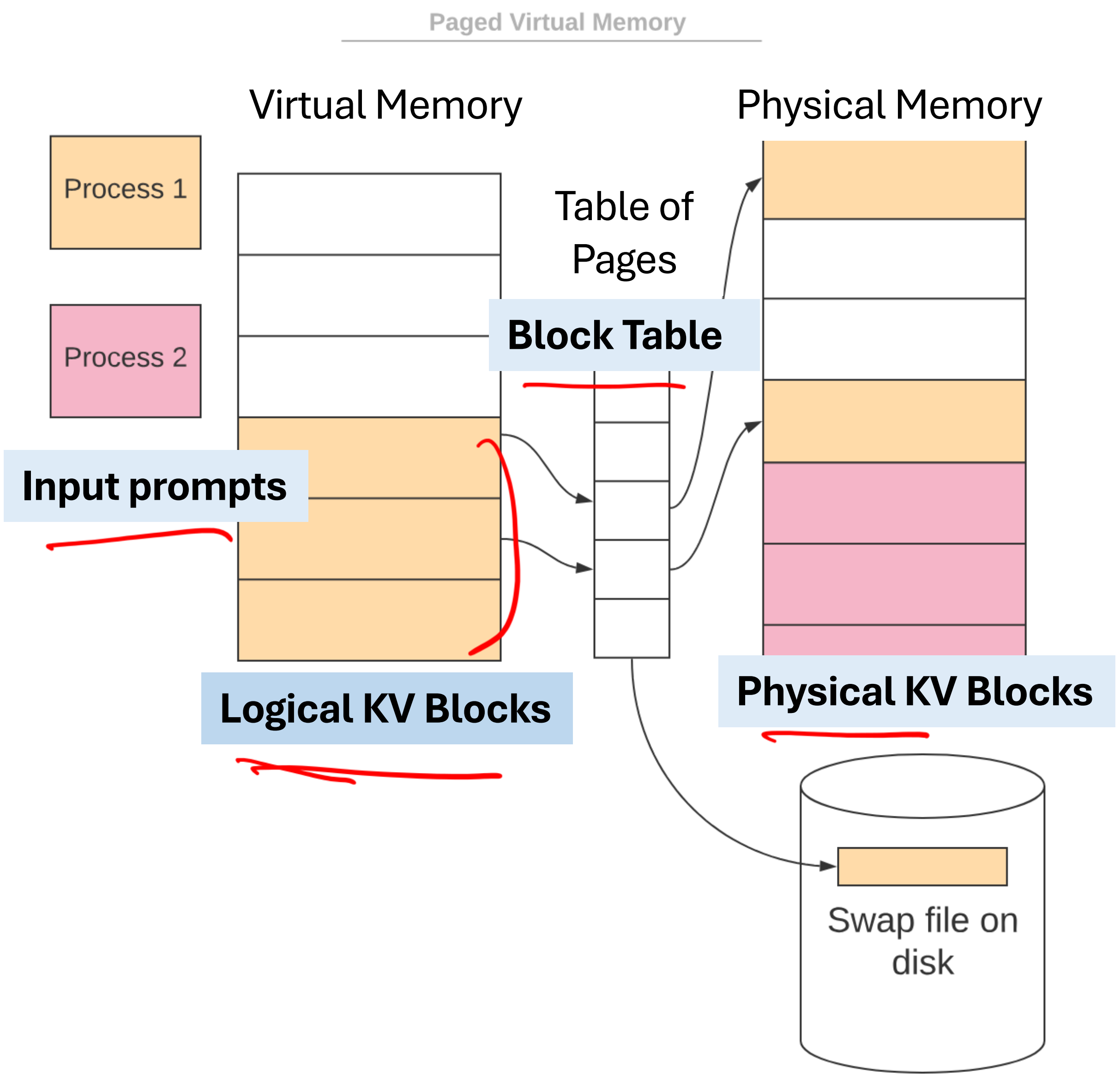
Inspired by Virtual memory and paging



Memory management in OS

Memory management in vLLM

Content credits: https://www.youtube.com/watch?v=5ZlavKF_98U&t=1646s&ab_channel=Anyscale



Efficient KV cache management

Inspired by Virtual memory and paging

- ❑ Processes as **incoming requests (**input to the model**)**
- ❑ Virtual Memory to **Logical KV Blocks**
- ❑ Physical Memory to **Physical KV Blocks**
- ❑ Page table to **Block Table**

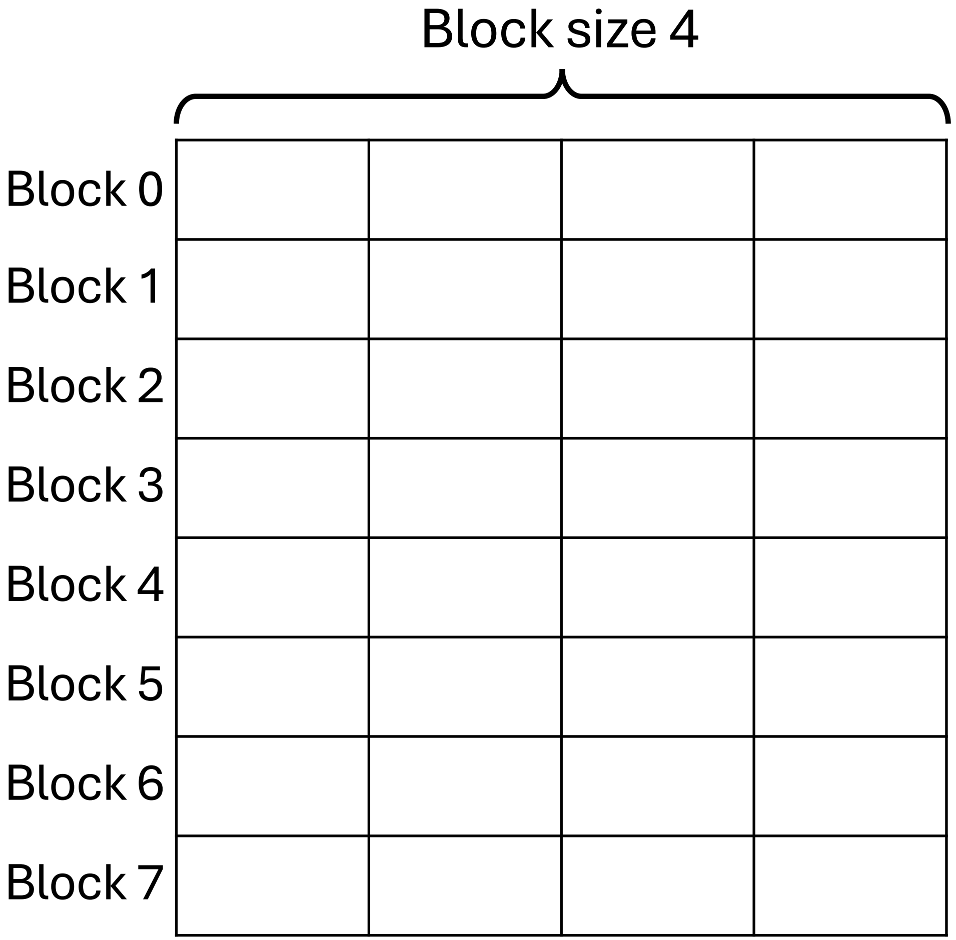
Content credits: https://youtu.be/yVXtLTcdO1Q?si=XO2Dk-VYOShUMH1u

KV Blocks

KV Cache

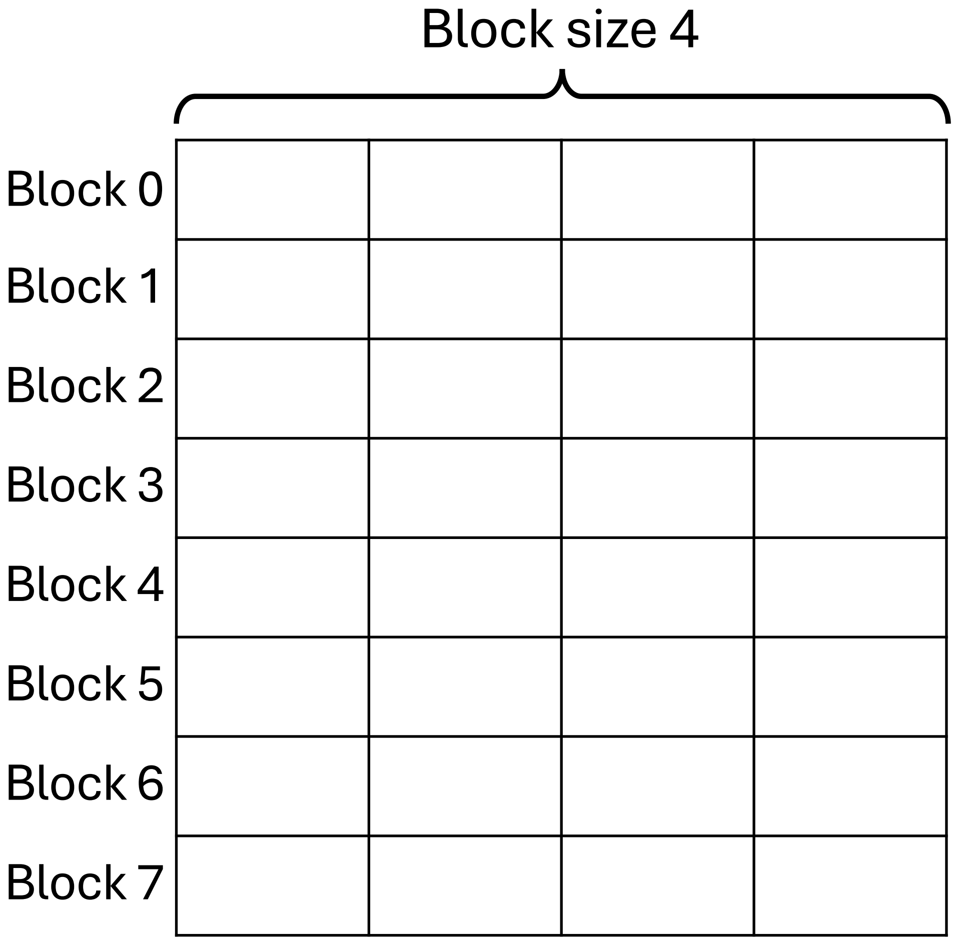
Content credits: https://www.youtube.com/watch?v=5ZlavKF_98U&t=1646s&ab_channel=Anyscale

KV Blocks



Content credits: https://www.youtube.com/watch?v=5ZlavKF_98U&t=1646s&ab_channel=Anyscale

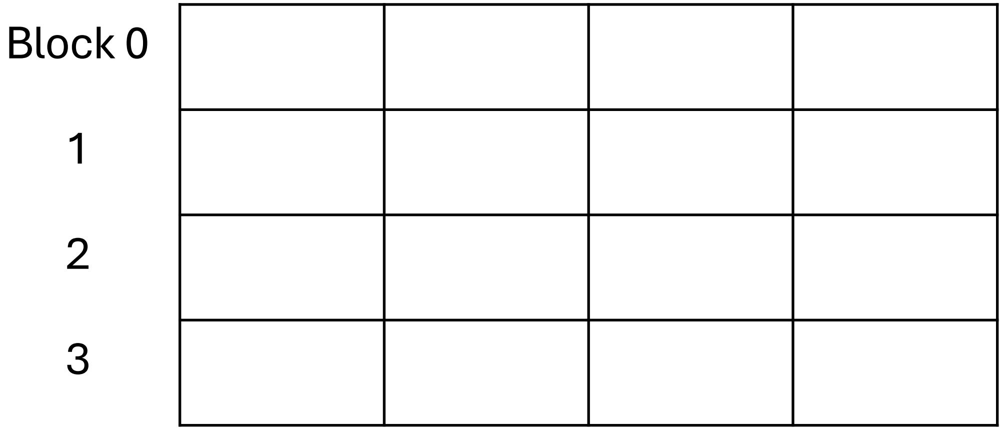
KV Blocks



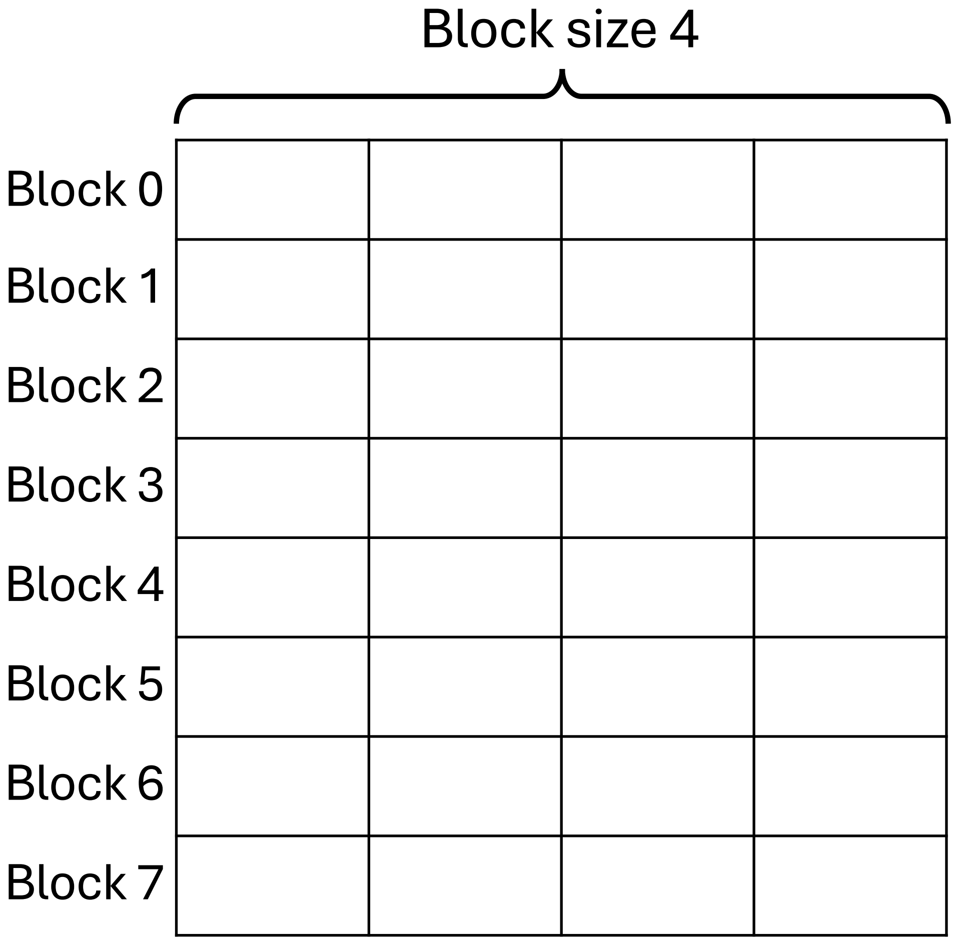
Physical KV Blocks

Content credits: https://www.youtube.com/watch?v=5ZlavKF_98U&t=1646s&ab_channel=Anyscale

Physical vs Logical KV Blocks



Logical KV Blocks



Physical KV Blocks

Content credits: https://www.youtube.com/watch?v=5ZlavKF_98U&t=1646s&ab_channel=Anyscale

Physical vs Logical KV Blocks

Block 0				
1				
2				
3				

Logical KV Blocks

Phys. Block	# Filled

Block Table

Block size 4

Block 0				
Block 1				
Block 2				
Block 3				
Block 4				
Block 5				
Block 6				
Block 7				

Physical KV Blocks

Content credits: https://www.youtube.com/watch?v=5ZlavKF_98U&t=1646s&ab_channel=Anyscale

Physical vs Logical KV Blocks

Prompt: "*Today we are learning about LLMs and*"

Block 0	Today	we	are	learning
1	about	LLMs	and	
2				
3				

Logical KV Blocks

Phys. Block	# Filled

Block Table

Block size 4

Block 0				
Block 1				
Block 2				
Block 3				
Block 4				
Block 5				
Block 6				
Block 7				

Physical KV Blocks

Content credits: https://youtu.be/yVXtLTcdO1Q?si=XO2Dk-VYOShUMH1u

Physical vs Logical KV Blocks

Prompt: "*Today we are learning about LLMs and*"

Block 0	Today	we	are	learning
1	about	LLMs	and	
2				
3				

Logical KV Blocks

Phys. Block	# Filled
7	4

Block Table

Block size 4

Block 0				
1				
2				
3				
4				
5				
6				
7	Today	we	are	learning

Physical KV Blocks

Content credits: https://youtu.be/yVXtLTcdO1Q?si=XO2Dk-VYOShUMH1u

Physical vs Logical KV Blocks

Prompt: "*Today we are learning about LLMs and*"

Block 0	Today	we	are	learning
1	about	LLMs	and	
2				
3				

Logical KV Blocks

Phys. Block	# Filled
7	4
1	3

Block Table

Block size 4

Block 0				
1	about	LLMs	and	
2				
3				
4				
5				
6				
7	Today	we	are	learning

Physical KV Blocks

Content credits: https://youtu.be/yVXtLTcdO1Q?si=XO2Dk-VYOShUMH1u

Physical vs Logical KV Blocks

Prompt: "*Today we are learning about LLMs and*"
Completion: "*memory*"

Block 0	Today	we	are	learning
1	about	LLMs	and	memory
2				
3				

Logical KV Blocks

Phys. Block	# Filled
7	4
1	4

Block Table

Block size 4

Block 0				
1	about	LLMs	and	
2				
3				
4				
5				
6				
7	Today	we	are	learning

Physical KV Blocks

Content credits: https://youtu.be/yVXtLTcdO1Q?si=XO2Dk-VYOShUMH1u

Physical vs Logical KV Blocks

Prompt: "*Today we are learning about LLMs and*"
Completion: "*memory*"

Logical KV Blocks

Block 0	Today	we	are	learning
1	about	LLMs	and	memory
2				
3				

Block Table

Phys. Block	# Filled
7	4
1	4

Physical KV Blocks

Block size 4

Block 0				
1	about	LLMs	and	memory
2				
3				
4				
5				
6				
7	Today	we	are	learning

Content credits: https://youtu.be/yVXtLTcdO1Q?si=XO2Dk-VYOShUMH1u

Physical vs Logical KV Blocks

Prompt: "*Today we are learning about LLMs and*"
Completion: "*memory on*"

Block 0	Today	we	are	learning
1	about	LLMs	and	memory
2	on			
3				

Logical KV Blocks

Phys. Block	# Filled
7	4
1	4

Block Table

Block size 4

Block 0				
1	about	LLMs	and	memory
2				
3				
4				
5				
6				
7	Today	we	are	learning

Physical KV Blocks

Content credits: https://youtu.be/yVXtLTcdO1Q?si=XO2Dk-VYOShUMH1u

Physical vs Logical KV Blocks

Prompt: "*Today we are learning about LLMs and*"
Completion: "*memory on*"

Block 0	Today	we	are	learning
1	about	LLMs	and	memory
2	on			
3				

Logical KV Blocks

Phys. Block	# Filled
7	4
1	4
5	1

Block Table

Block size 4

Block 0				
1	about	LLMs	and	memory
2				
3				
4				
5				
6				
7	Today	we	are	learning

Allocated on demand

Physical KV Blocks

Content credits: https://youtu.be/yVXtLTcdO1Q?si=XO2Dk-VYOShUMH1u

Physical vs Logical KV Blocks

Prompt: "*Today we are learning about LLMs and*"
Completion: "*memory on*"

Block 0	Today	we	are	learning
1	about	LLMs	and	memory
2	on			
3				

Logical KV Blocks

Phys. Block	# Filled
7	4
1	4
5	1

Block Table

Block size 4

Block 0				
1	about	LLMs	and	memory
2				
3				
4				
5	on			
6				
7	Today	we	are	learning

Allocated on demand

Physical KV Blocks

Physical vs Logical KV Blocks

Prompt: "*Today we are learning about LLMs and*"
Completion: "*memory on demand*"

Block 0	Today	we	are	learning
1	about	LLMs	and	memory
2	on	demand		
3				

Logical KV Blocks

Phys. Block	# Filled
7	4
1	4
5	2

Block Table

Block size 4

Block 0				
1	about	LLMs	and	memory
2				
3				
4				
5	on	demand		
6				
7	Today	we	are	learning

Physical KV Blocks

Content credits: https://youtu.be/yVXtLTcdO1Q?si=XO2Dk-VYOShUMH1u

Physical vs Logical KV Blocks

Logical KV Blocks

Block 0	Today	we	are	learning
1	about	LLMs	and	memory
2	on	demand	</s>	
3				

Block Table

Phys. Block	# Filled
7	4
1	4
5	2

Physical KV Blocks

Block size 4

Block 0				
1	about	LLMs	and	memory
2				
3				
4				
5	on	demand		
6				
7	Today	we	are	learning

Prompt A: "*Today we are learning about LLMs and*"
Completion: "***memory on demand </s>***"

Content credits: https://youtu.be/yVXtLTcdO1Q?si=XO2Dk-VYOShUMH1u

Physical vs Logical KV Blocks

Block 0	Today	we	are	learning
1	about	LLMs	and	memory
2	on	demand	</s>	
3				

Logical KV Blocks

Phys. Block	# Filled
7	4
1	4
5	2

Block Table

Block size 4

Block 0				
1	about	LLMs	and	memory
2				
3				
4				
5	on	demand		
6				
7	Today	we	are	learning

Internal fragmentation

Physical KV Blocks

Prompt A: "*Today we are learning about LLMs and*"
Completion: "***memory on demand </s>***"

Content credits: https://youtu.be/yVXtLTcdO1Q?si=XO2Dk-VYOShUMH1u

0	Today	we	are	learning
1	about	LLMs	and	
2				
3				

Logical KV Blocks - B

0	Today	we	are	learning
1	about	LLMs	and	memory
2	on	demand	</s>	
3				

Logical KV Blocks - A

Phys. Block	# Filled
7	4
1	4
5	2

Block Table -A

Block size 4

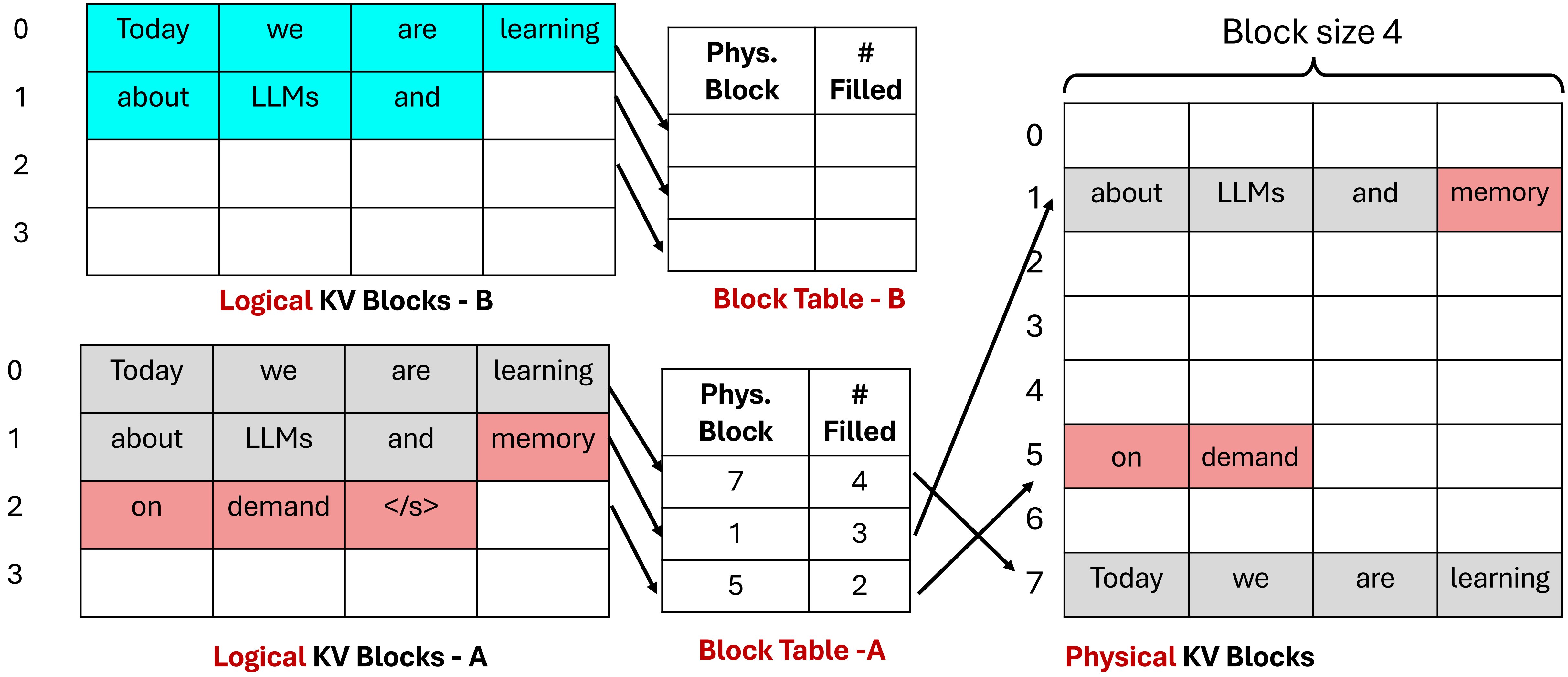
0				
1	about	LLMs	and	memory
2				
3				
4				
5	on	demand		
6				
7	Today	we	are	learning

Physical KV Blocks

Prompt A: "*Today we are learning about LLMs and*"
Completion: "***memory on demand</s>***"

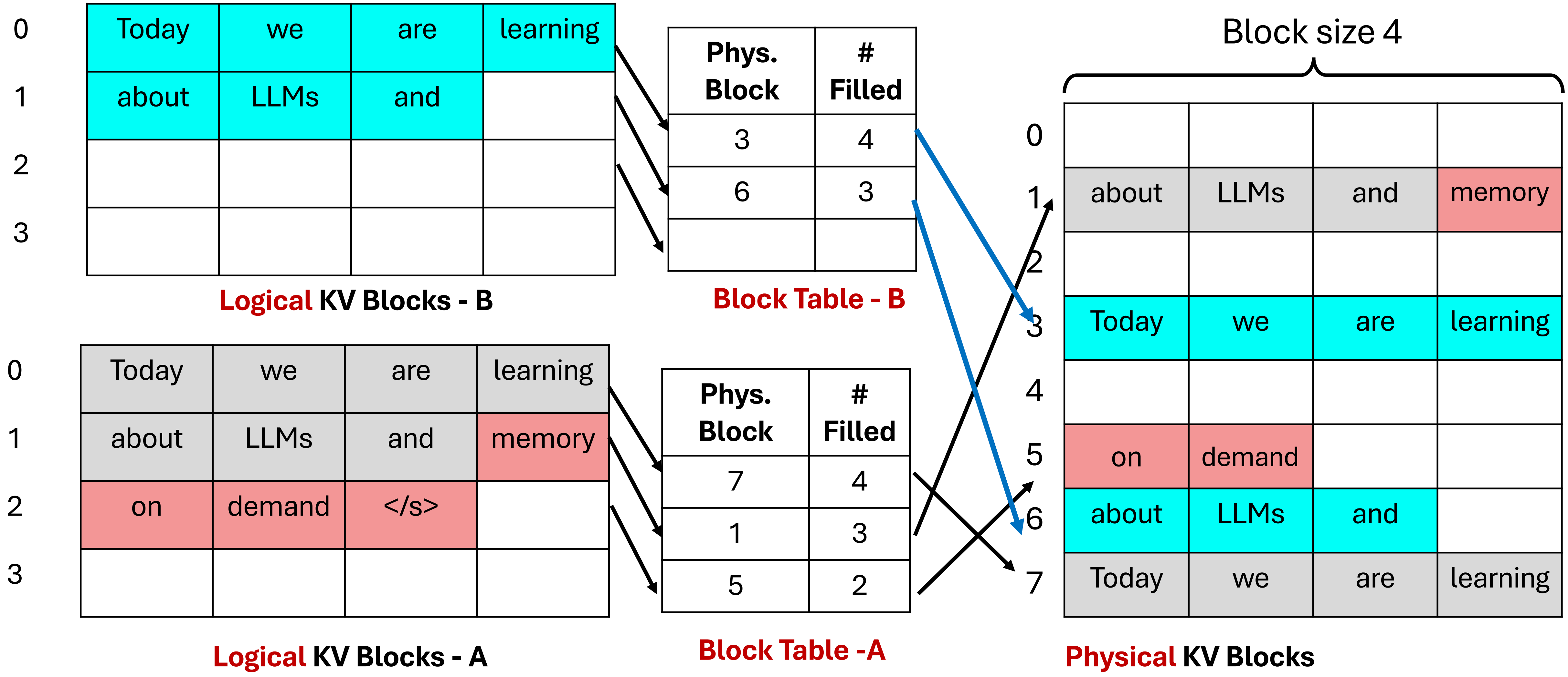
Prompt B: "*Today we are learning about LLMs and*"
Completion:

Content credits: https://youtu.be/yVXtLTcdO1Q?si=XO2Dk-VYOShUMH1u



Prompt A: "*Today we are learning about LLMs and*"
Completion: "***memory on demand </s>***"

Prompt B: "*Today we are learning about LLMs and*"
Completion:



Prompt A: "*Today we are learning about LLMs and*"
Completion: "*memory on demand </s>*"

Prompt B: "*Today we are learning about LLMs and*"
Completion:

0	Today	we	are	learning
1	about	LLMs	and	memory
2	management	</s>		
3				

Logical KV Blocks - B

Phys. Block	# Filled
3	4
6	4
2	1

Block Table - B

0	Today	we	are	learning
1	about	LLMs	and	memory
2	on	demand	</s>	
3				

Logical KV Blocks - A

Phys. Block	# Filled
7	4
1	3
5	2

Block Table -A

Block size 4

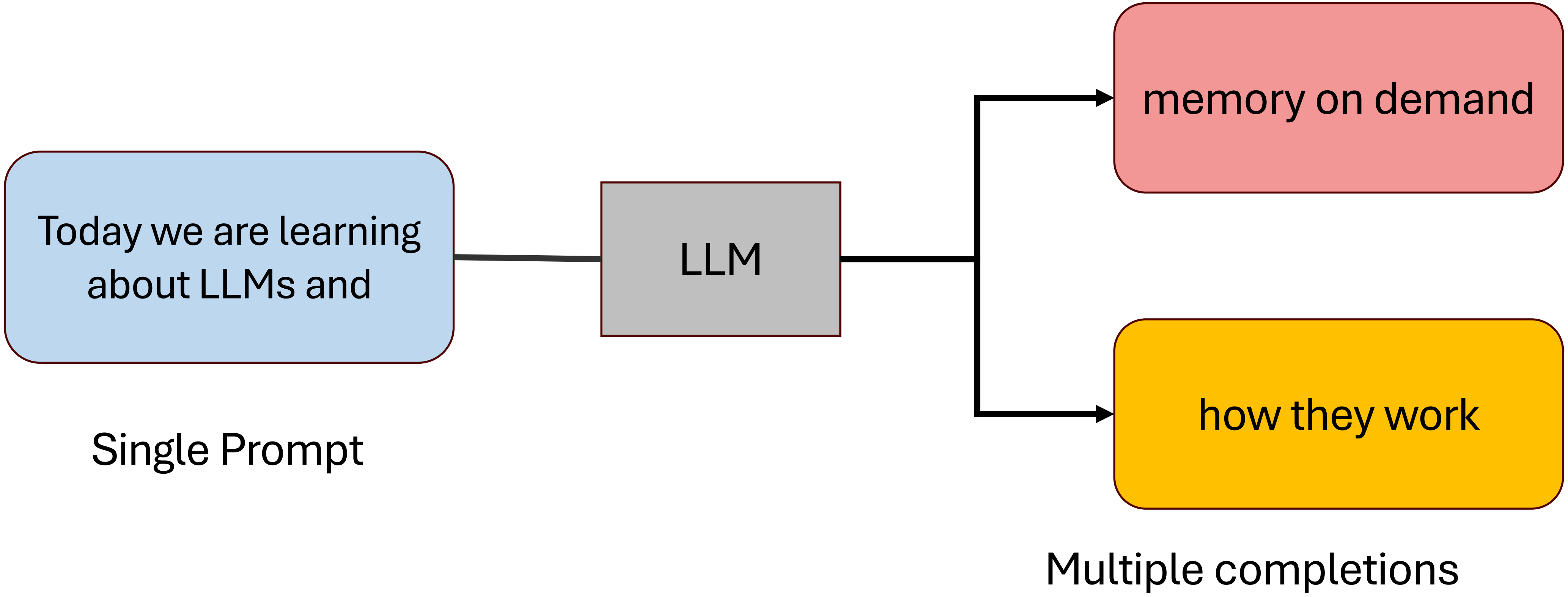
0				
1	about	LLMs	and	memory
2	management			
3	Today	we	are	learning
4				
5	on	demand		
6	about	LLMs	and	memory
7	Today	we	are	learning

Physical KV Blocks

Prompt A: "*Today we are learning about LLMs and*"
Completion: "*memory on demand </s>*"

Prompt B: "*Today we are learning about LLMs and*"
Completion: "*memory management </s>*"

Dynamic block mapping enables sharing



Sharing KV blocks in parallel sampling

B.T. for Ky#

Phys. Block	# Filled
5	4
7	3

Today	we	are	learning
about	LLMs	and	memory

Logical KV Blocks - A

0				
1				
2				
3				
4				
5	Today	we	are	learni ng
6				
7	about	LLMs	and	memory

Physical KV Blocks

B.T. for Ky.B.

Phys. Block	# Filled
~~7~~ 5	4
~~5~~ 7	3

Today	we	are	learning
about	LLMs	and	how

Logical KV Blocks - B

Sharing KV blocks in parallel sampling

Phys. Block	# Filled
5	4
7	3

Today	we	are	learning
about	LLMs	and	

Logical KV Blocks - A

0				
1				
2				
3				
4				
5	Today	we	are	learni ng
7	about	LLMs	and	

Ref count: 2

Physical KV Blocks

Phys. Block	# Filled
7	4
5	3

Today	we	are	learning
about	LLMs	and	

Logical KV Blocks - B

Sharing KV blocks in parallel sampling

Phys. Block	# Filled
5	4
7	3

Today	we	are	learning
about	LLMs	and	memory

Logical KV Blocks - A

0				
1				
2				
3				
4				
5	Today	we	are	learni ng
7	about	LLMs	and	

Ref count: 2

Physical KV Blocks

Phys. Block	# Filled
7	4
5	3

Today	we	are	learning
about	LLMs	and	how

Logical KV Blocks - B

Sharing KV blocks in parallel sampling

Phys. Block	# Filled
5	4
7	3

Today	we	are	learning
about	LLMs	and	memory

Logical KV Blocks - A

0				
1				
2				
3				
4				
5	Today	we	are	learni ng
6				
7	about	LLMs	and	

Ref count: 2 → 1

Physical KV Blocks

Phys. Block	# Filled
7	4
5	3

Today	we	are	learning
about	LLMs	and	how

Logical KV Blocks - B

Copy-on-write

Sharing KV blocks in parallel sampling

Phys. Block	# Filled
5	4
7	3

Today	we	are	learning
about	LLMs	and	memory

Logical KV Blocks - A

copy –on-write

Ref count: 1

0				
1				
2	about	LLMs	and	
3				
4				
5	Today	we	are	learning
6				
7	about	LLMs	and	

Physical KV Blocks

Phys. Block	# Filled
7	4
5	3

Today	we	are	learning
about	LLMs	and	how

Logical KV Blocks - B

Sharing KV blocks in parallel sampling

Phys. Block	# Filled
5	4
7	3

Today	we	are	learning
about	LLMs	and	memory

Logical KV Blocks - A

copy –on-write

0				
1				
2	about	LLMs	and	memo ry
3				
4				
5	Today	we	are	learni ng
6				
7	about	LLMs	and	

Ref count: 1

Physical KV Blocks

Phys. Block	# Filled
7	4
5	3

Today	we	are	learning
about	LLMs	and	how

Logical KV Blocks - B

Sharing KV blocks in parallel sampling

Phys. Block	# Filled
5	4
2	4

Today	we	are	learning
about	LLMs	and	memory

Logical KV Blocks - A

copy –on-write

0				
2	about	LLMs	and	memory
3				
4				
5	Today	we	are	learning
6				
	about	LLMs	and	

Refcount-2

Ref count: 1

Physical KV Blocks

Phys. Block	# Filled
7	4
5	3

Today	we	are	learning
about	LLMs	and	how

Logical KV Blocks - B

Sharing KV blocks in parallel sampling

Phys. Block	# Filled
5	4
2	4

Today	we	are	learning
about	LLMs	and	memory

Logical KV Blocks - A

copy –on-write

0				
2	about	LLMs	and	memory
3				
4				
5	Today	we	are	learning
6				
	about	LLMs	and	how

Ref count: 1

Physical KV Blocks

Phys. Block	# Filled
7	4
5	3

Today	we	are	learning
about	LLMs	and	how

Logical KV Blocks - B

Sharing KV blocks in parallel sampling

Phys. Block	# Filled
5	4
2	4
0	2

Today	we	are	learning
about	LLMs	and	memory
on	demand		

Logical KV Blocks - A

0	on	demand		
1				
2	about	LLMs	and	memo ry
3				
4	they	work		
5	Today	we	are	learni ng
6				
7	about	LLMs	and	how

Physical KV Blocks

Phys. Block	# Filled
~~7~~ 5	4
~~5~~ 7	4
4	2

Today	we	are	learning
about	LLMs	and	how
they	work		

Logical KV Blocks - B

Sharing KV blocks in parallel sampling

Phys. Block	# Filled
5	4
2	4
0	2

Today	we	are	learning
about	LLMs	and	memory
on	demand		

Logical KV Blocks - A

0	on	demand		
1				
2	about	LLMs	and	memo ry
3				
4	they	work		
5	Today	we	are	learni ng
7	about	LLMs	and	how

Ref count: 2

Physical KV Blocks

Phys. Block	# Filled
7	4
5	4
4	2

Today	we	are	learning
about	LLMs	and	how
they	work		

Logical KV Blocks - B

2048

The cat sat

Extern frag

We only 512

Memory efficiency of vLLMs

✓Minimal internal fragmentation

- Only happens at the last block of a sequence
- **# wasted tokens / seq < block size**
  - Sequence: O(100) or O(1000) tokens
  - Block size: 16 or 32 tokens

✓No external fragmentation

✓On average, wasted space < **4%** of KV cache

✓3-5x improved memory utilization!

0	Today	we	are	learning
1	about	LLMs	and	memory
2	management			
3				

Internal fragmentation

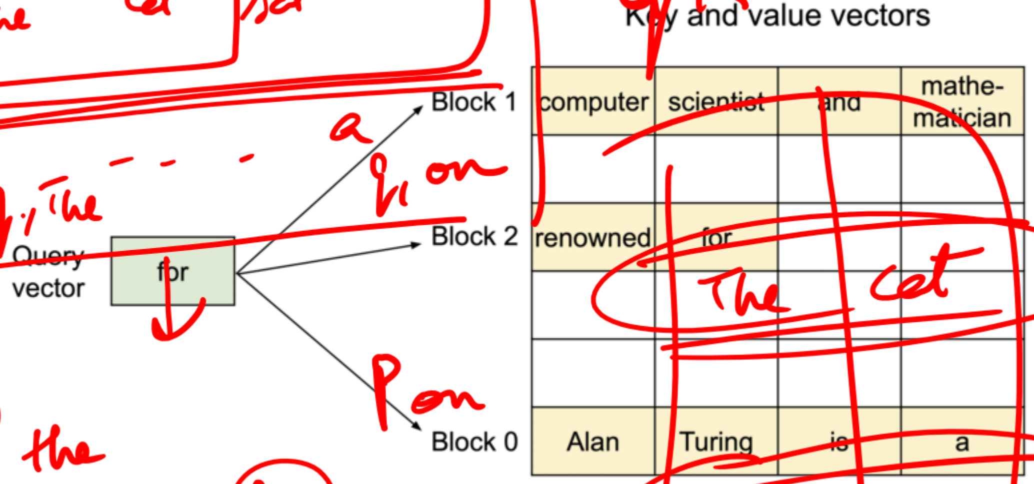
Content credits: https://www.youtube.com/watch?v=5ZlavKF_98U&t=1646s&ab_channel=Anyscale

Paged Attention

- Tensor operations require contiguous memory
- How to compute attention *softmax* across fragmented memory?
- Paged Attention!

$W_q a = q_a$

qK^T



$\sum p_i v_i$

$$softmax([A_1, A_2]) = [\alpha\, softmax(A_1), \beta\, softmax(A_2)]$$

$$softmax([A_1, A_2]) \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \alpha\, softmax(A_1) * V_1 + \beta\, softmax(A_2) * V_2$$

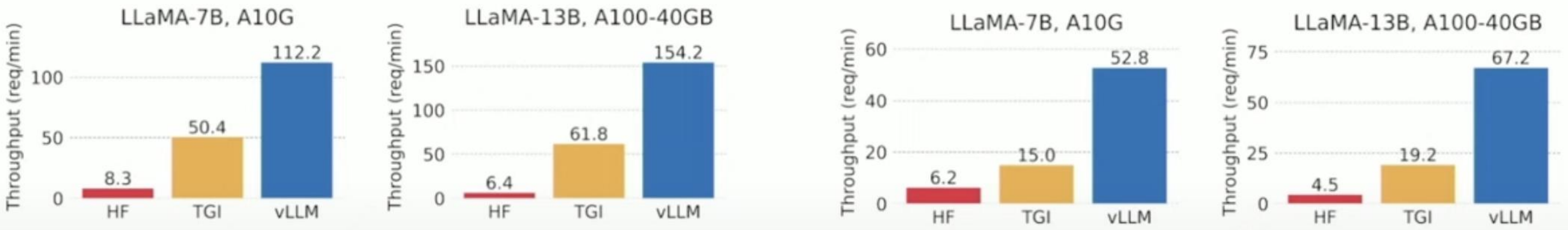
How vLLM & Paged Attention results in efficient inference?

Reduce memory fragmentation with paging

Further reduce memory usage with sharing

Comparison with HuggingFace and TGI (2023)

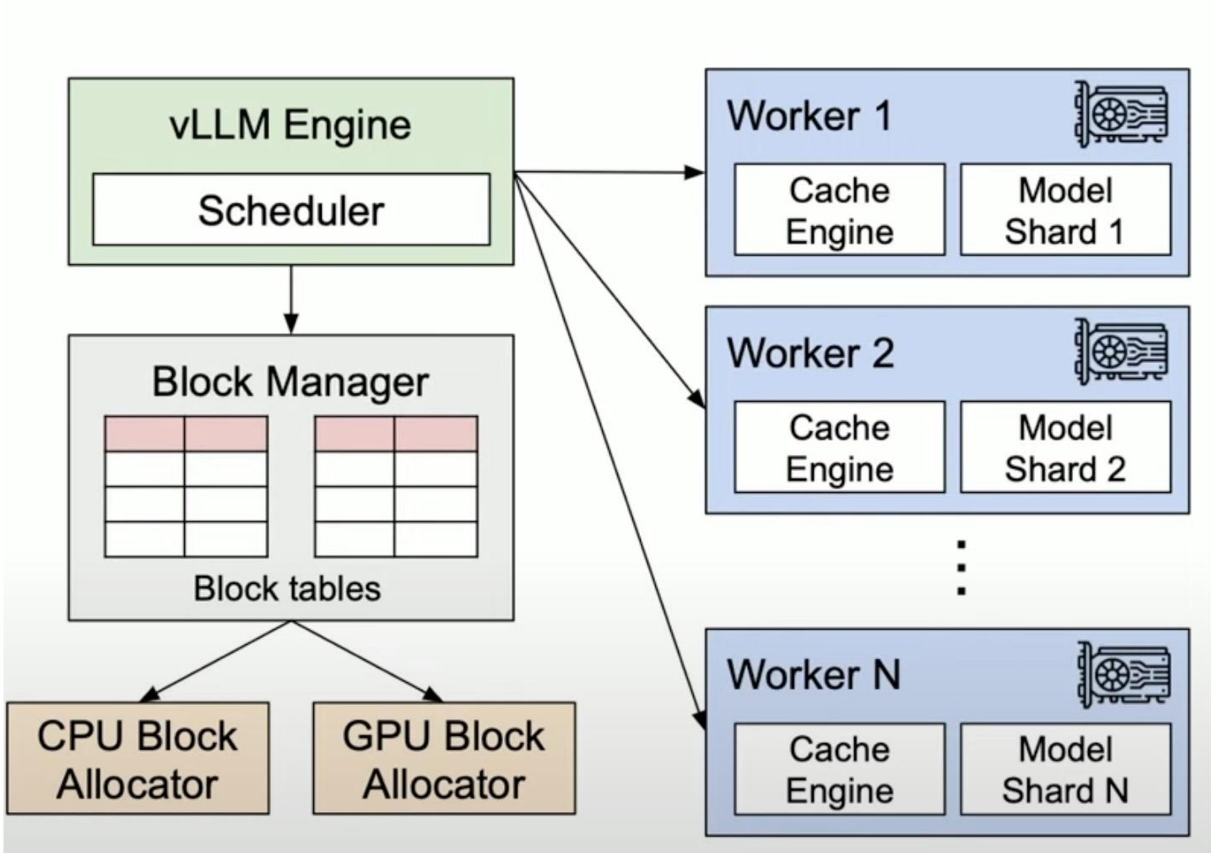
- Up to **24x** higher throughput than HuggingFace (HF)
- Up to **3.5x** higher throughput than Text Generation Inference (TGI)



Serving throughput when each request asks for 1 output completion.

Serving throughput when each request asks for 3 output completions.

System Architecture and Implementation

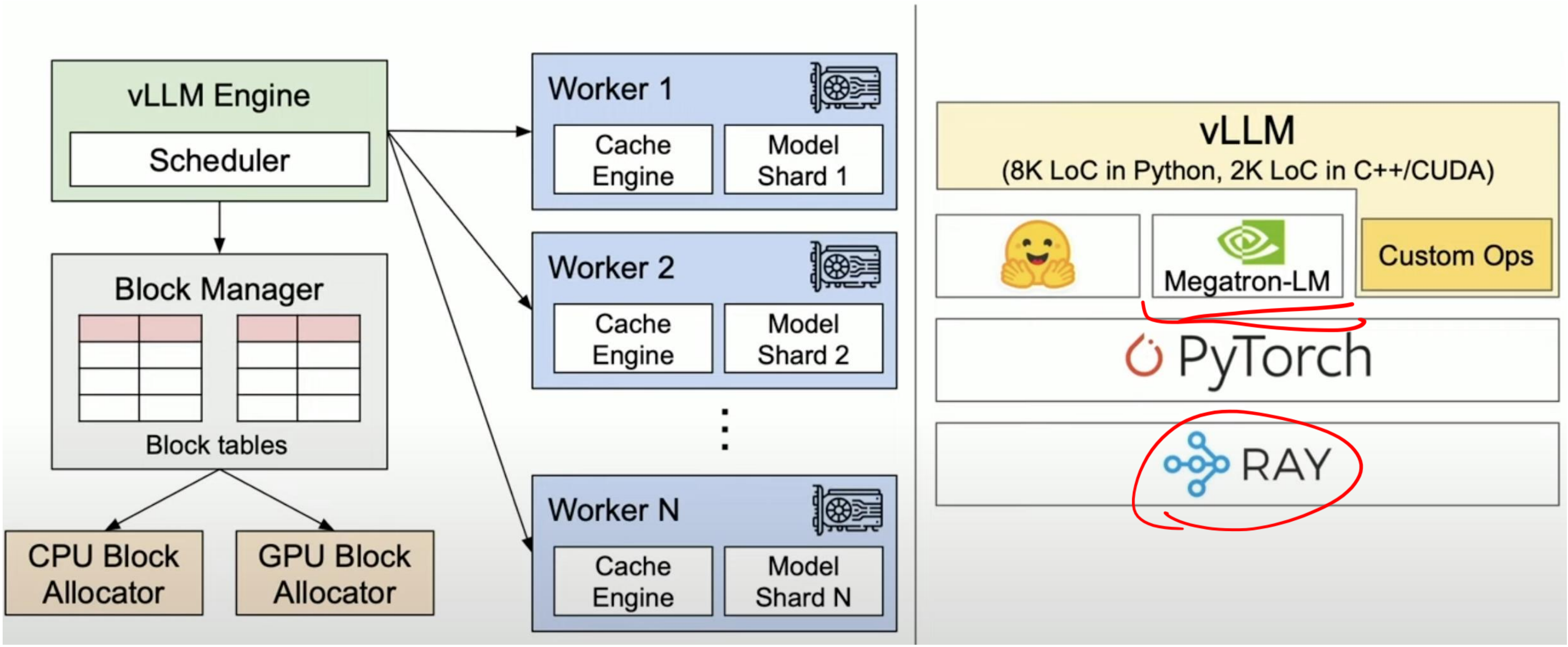


End to end llm serving engine

3 components –

- A frontend
- A distributed model executor
- A scheduler

A centralized engine that managers block table


vLLM Engine
Scheduler
Block Manager
Block tables
CPU Block Allocator
GPU Block Allocator
Worker 1
Cache Engine
Model Shard 1
Worker 2
Cache Engine
Model Shard 2
Worker N
Cache Engine
Model Shard N
vLLM
(8K LoC in Python, 2K LoC in C++/CUDA)
Megatron-LM
Custom Ops
PyTorch
RAY

Till now…

- **KV caching** – avoids re-computation of Keys and Value matrices
- **Paged Attention and vLLM -** efficient memory management
- Can we speed up attention computation?
- Flash Attention?

Redundant computation

GPU KV cache

Efficient memory management

Speedy computation

Speculative decoding