BIU NLP

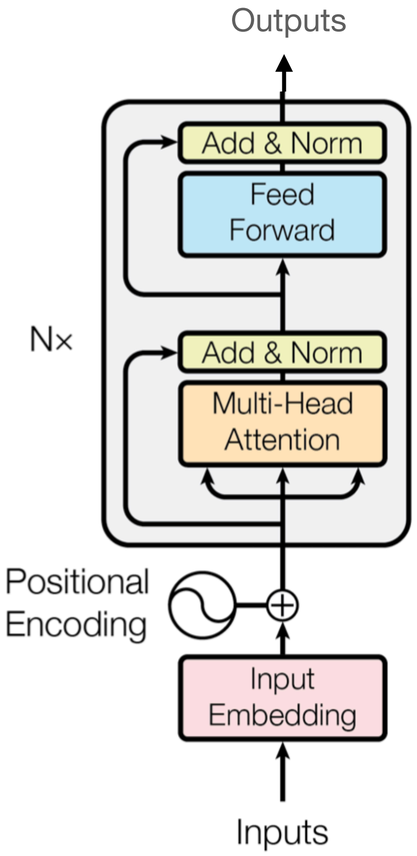
Thinking Like Transformers

Gail Weiss, Yoav Goldberg, Eran Yahav

European Research Council

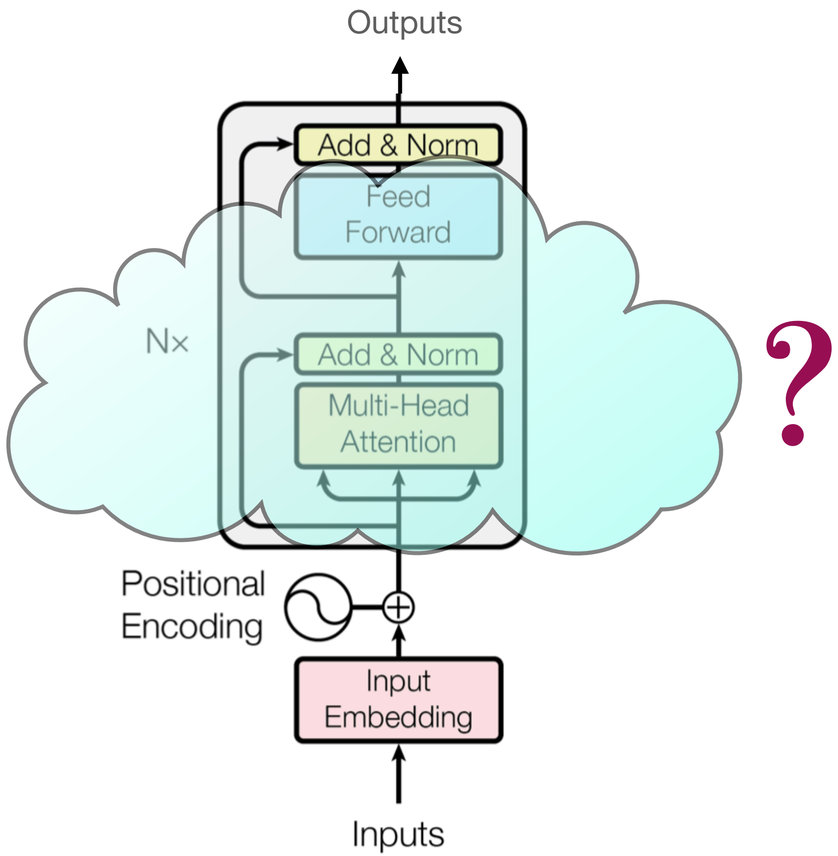
Supported by European Union's Horizon 2020 research and innovation programme, grant agreement no. 802774 (iEXTRACT)

Transformers are very effective



Transformers are very effective

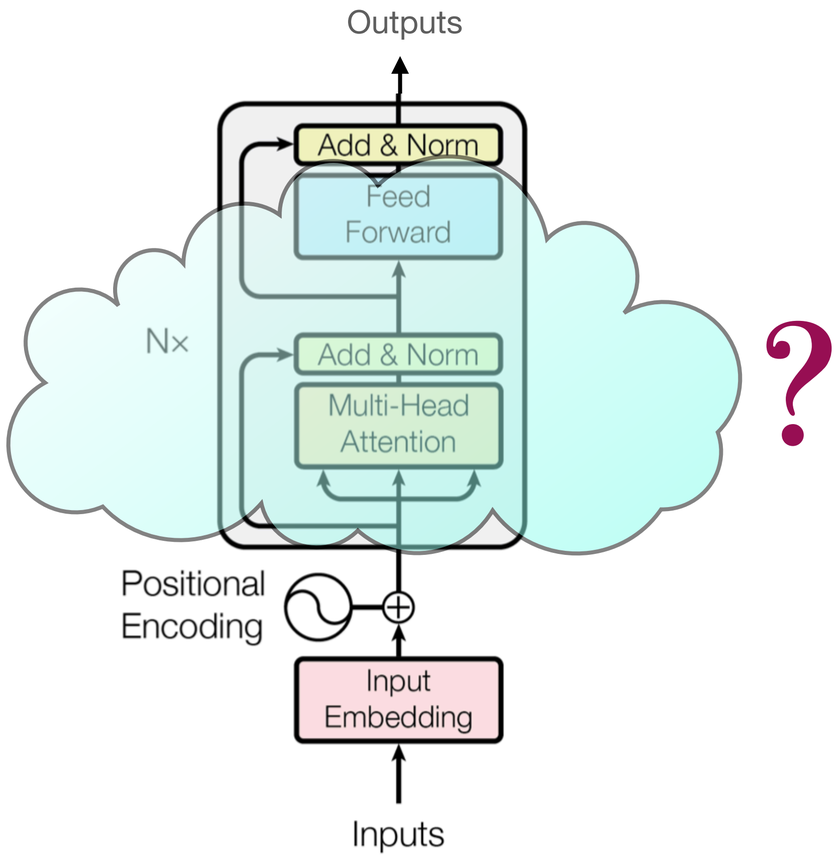
But we don't know how they work



?

Question

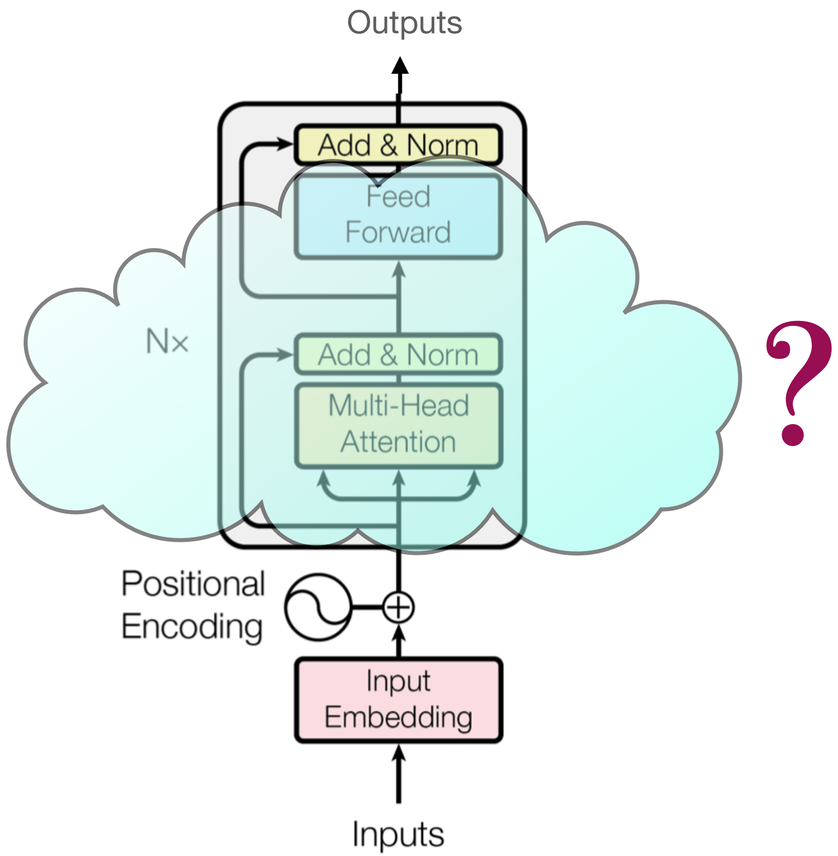
What is the computational model of the Transformer-encoder?



?

Can it solve “Reverse”?

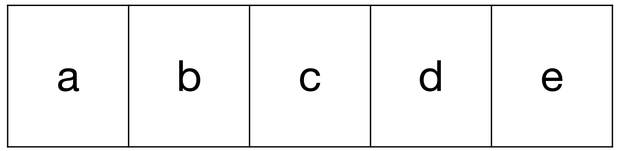
abcde ↦ edcba



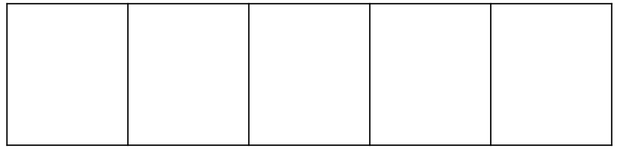
?

Can it solve “Reverse”?

abcde ↦ edcba

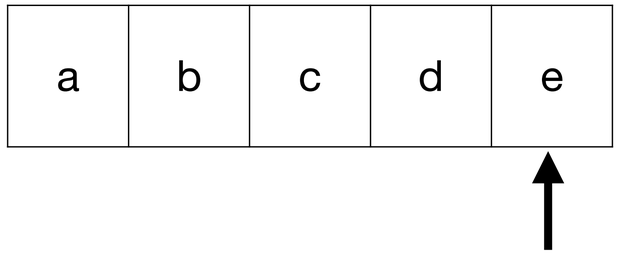


"standard" programming language

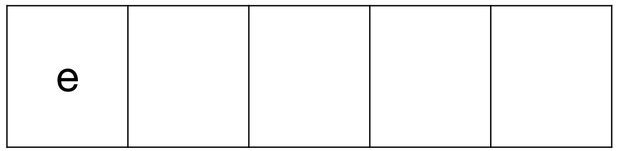


Can it solve “Reverse”?

abcde ↦ edcba

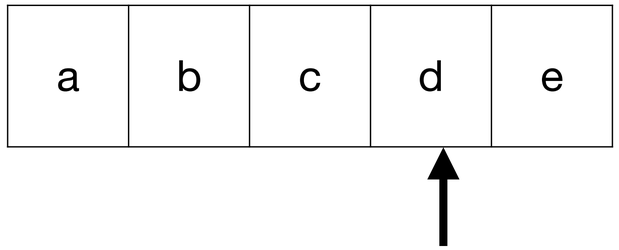


"standard" programming language

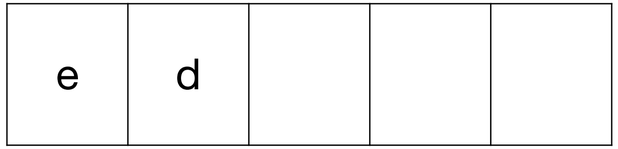


Can it solve “Reverse”?

abcde ↦ edcba

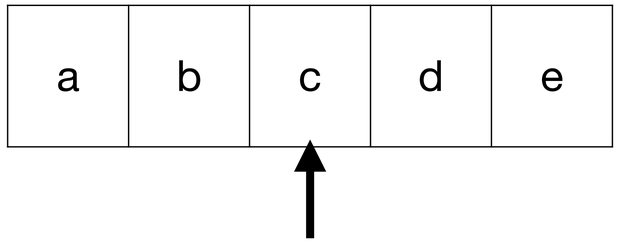


"standard" programming language

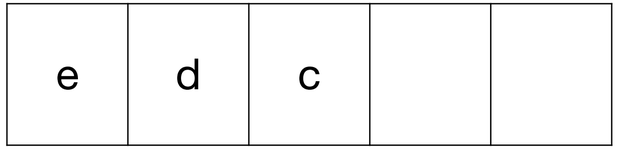


Can it solve “Reverse”?

abcde ↦ edcba

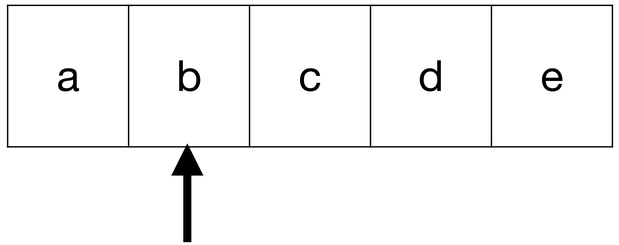


"standard" programming language

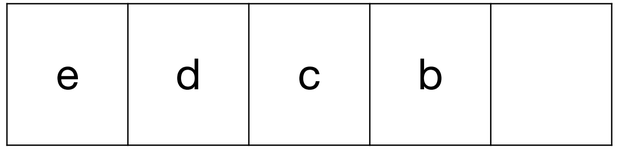


Can it solve “Reverse”?

abcde ↦ edcba

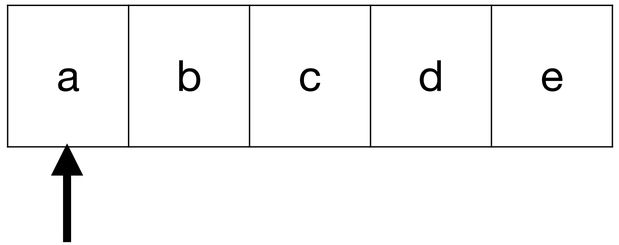


"standard" programming language

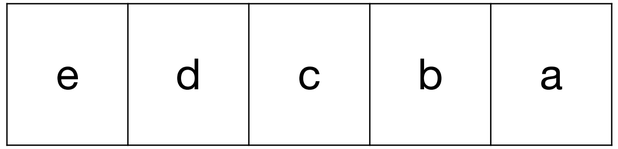


Can it solve “Reverse”?

abcde ↦ edcba

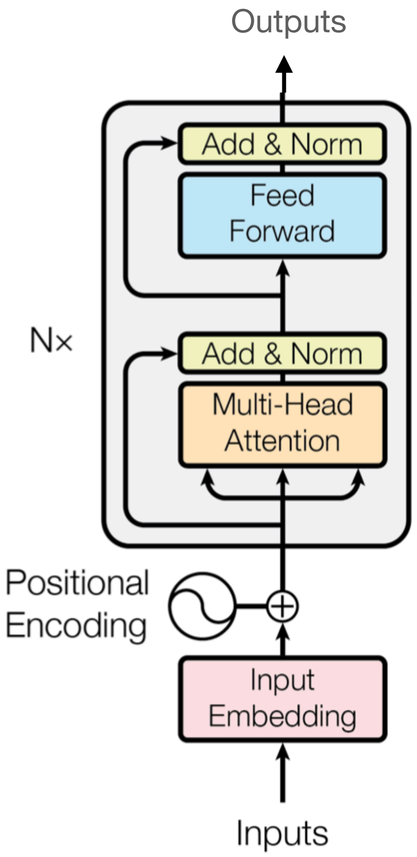


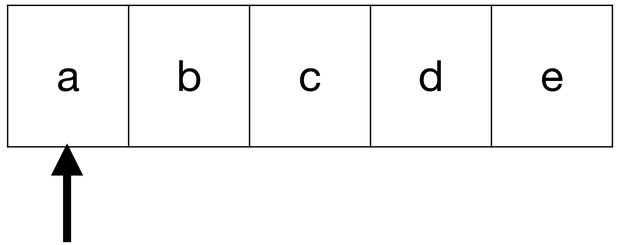
"standard" programming language



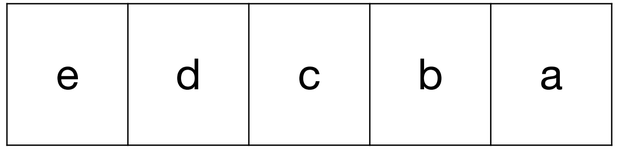
Can it solve “Reverse”?

abcde ↦ edcba



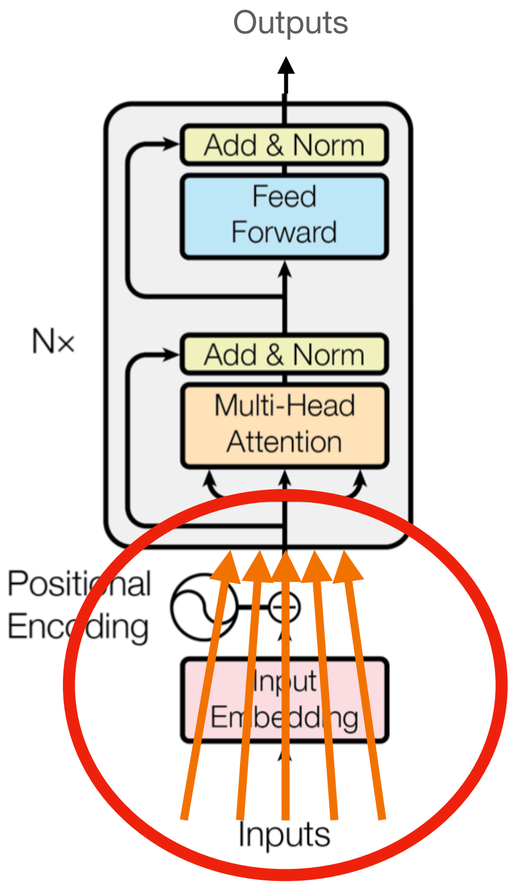


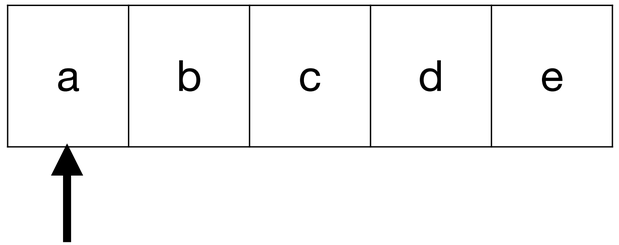
"standard" programming language



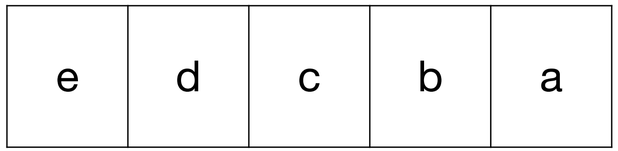
Can it solve “Reverse”?

abcde ↦ edcba



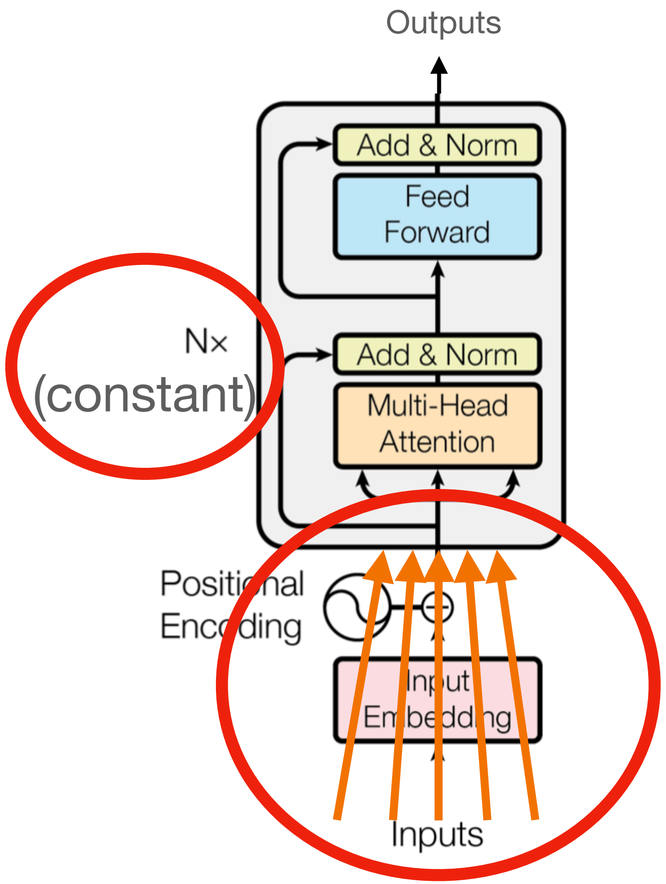


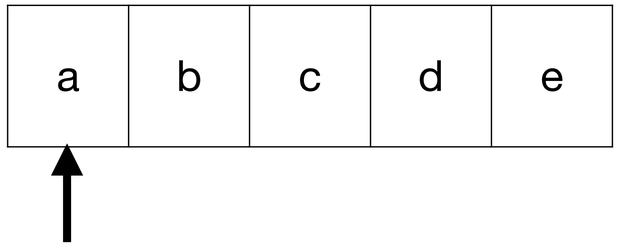
"standard" programming language



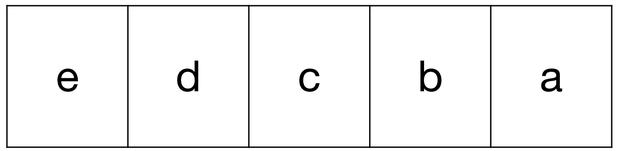
Can it solve “Reverse”?

abcde ↦ edcba



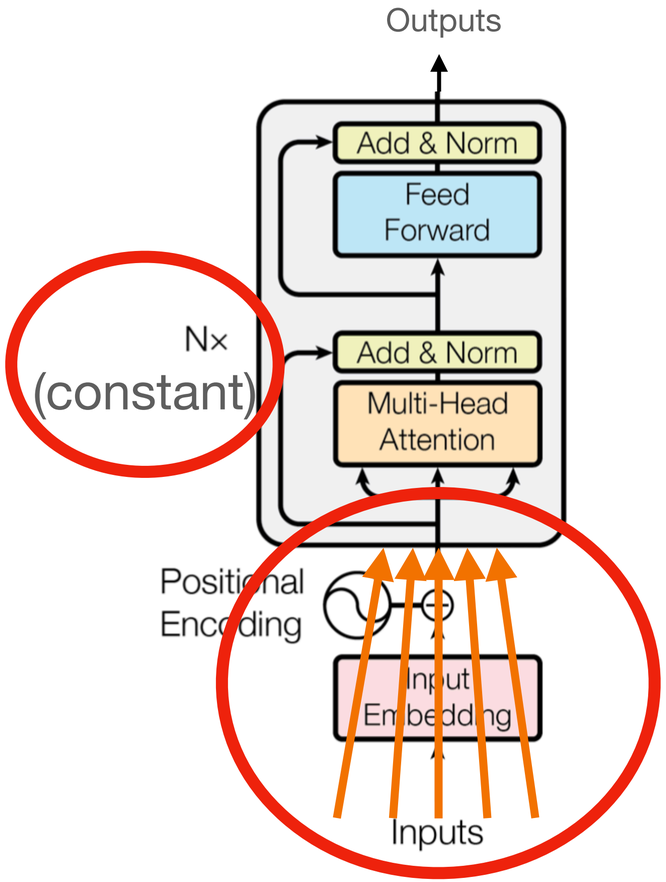


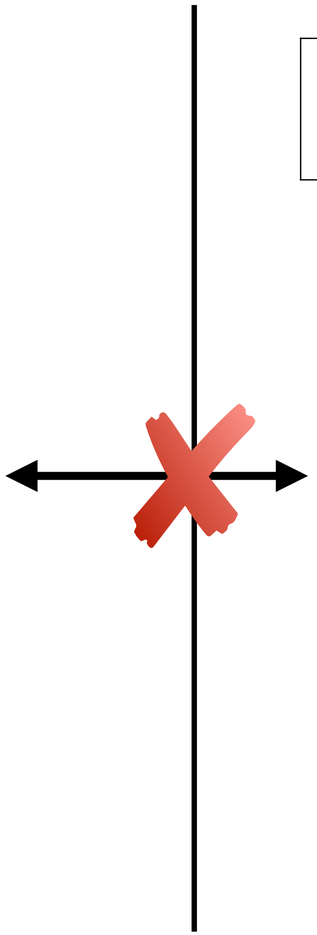
"standard" programming language

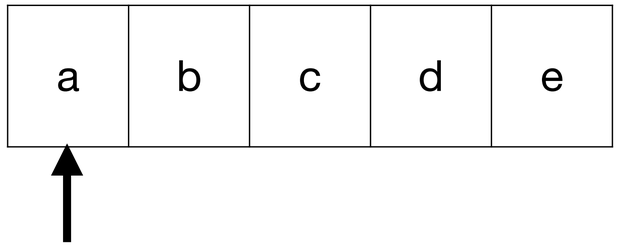


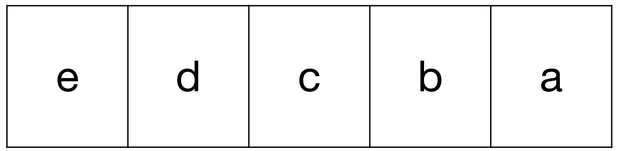
Can it solve “Reverse”?

abcde ↦ edcba





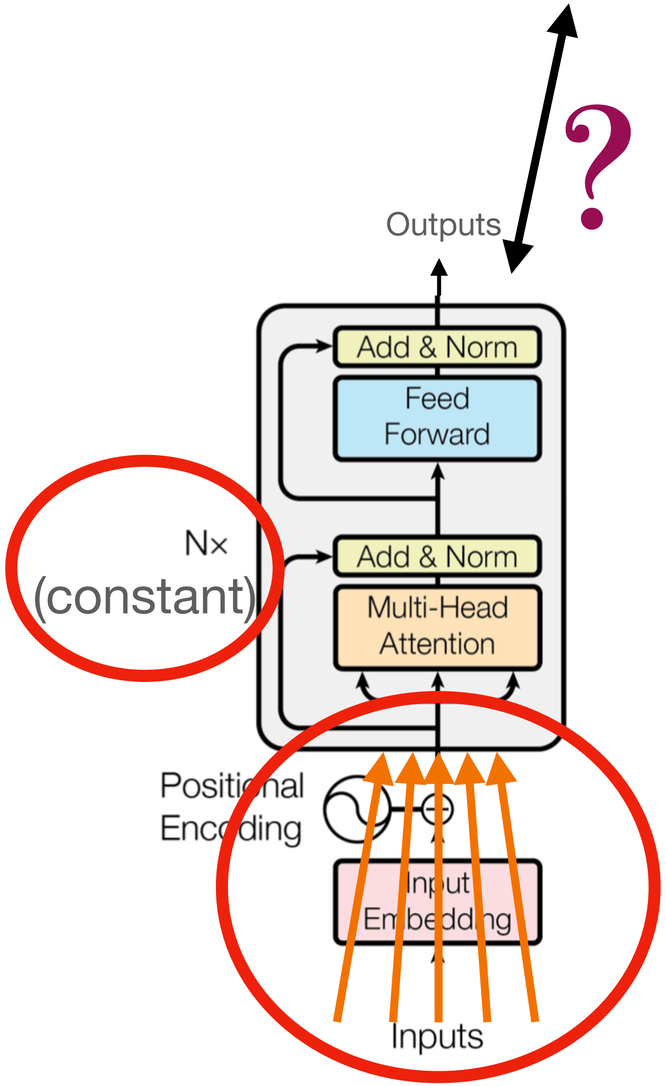


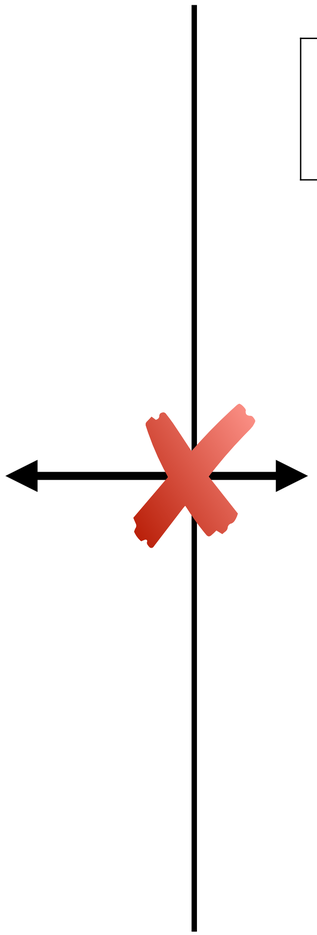


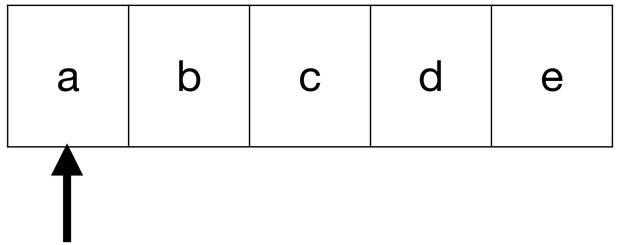
"standard" programming language

Can it solve “Reverse”?

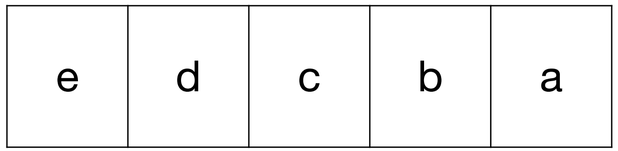
abcde ↦ edcba





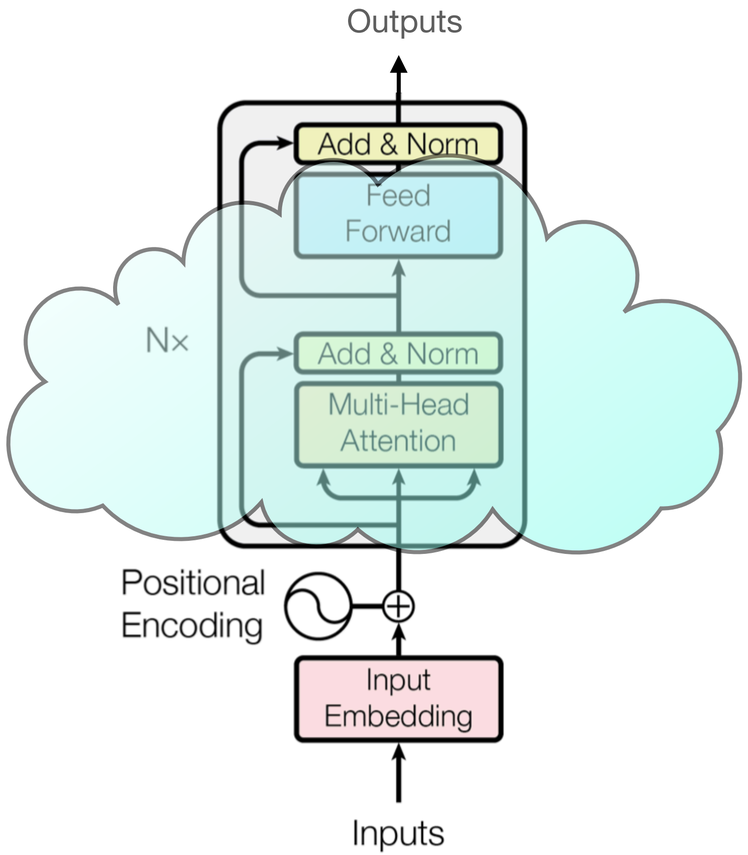


"standard" programming language



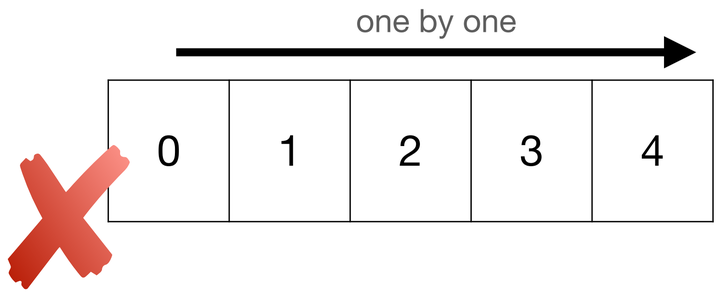
RASP

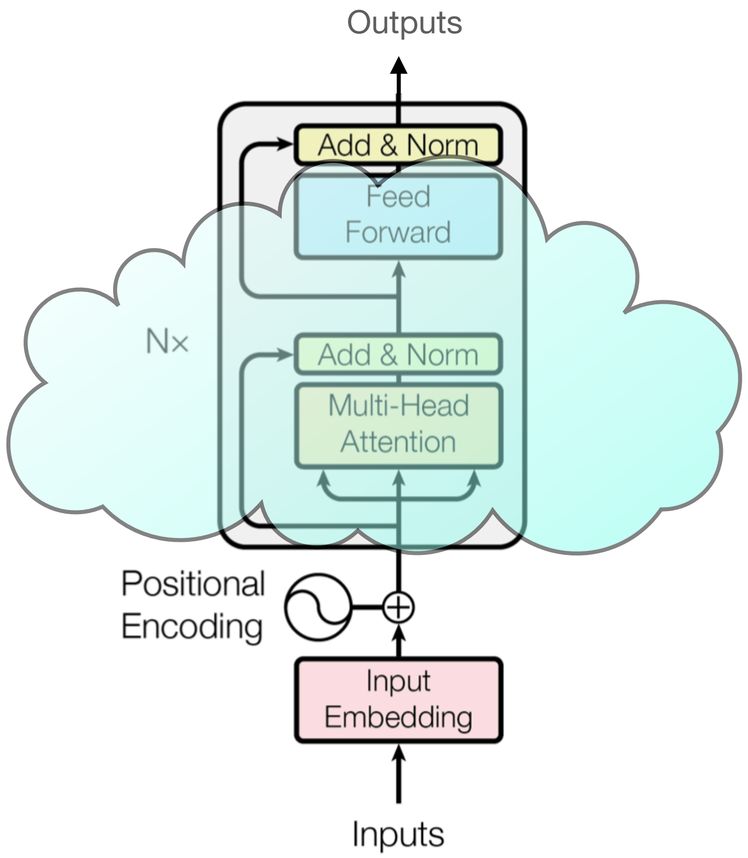
Restricted Access Sequence Processing



RASP

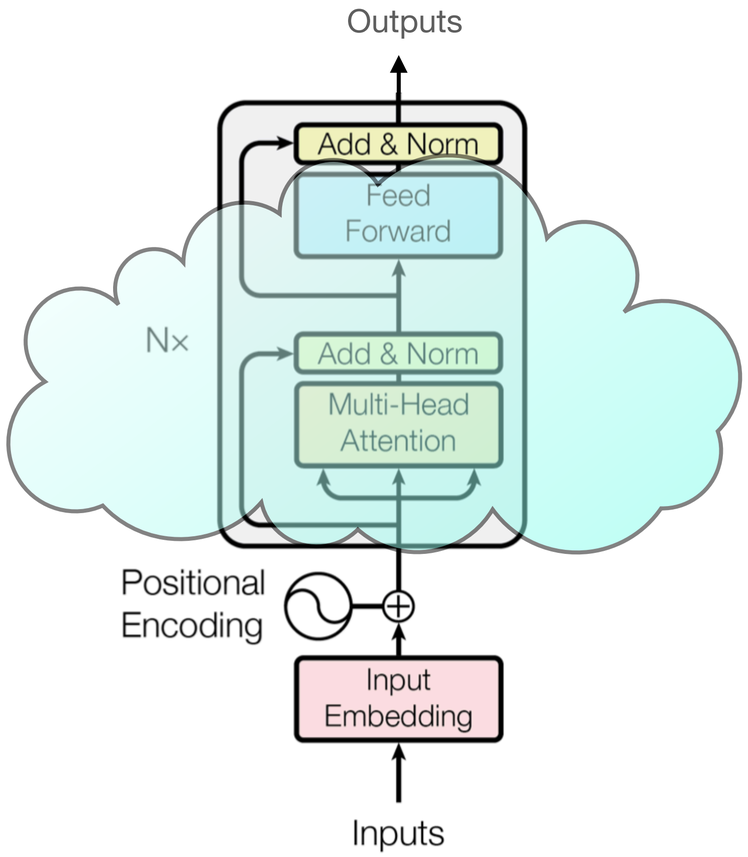
Restricted Access Sequence Processing

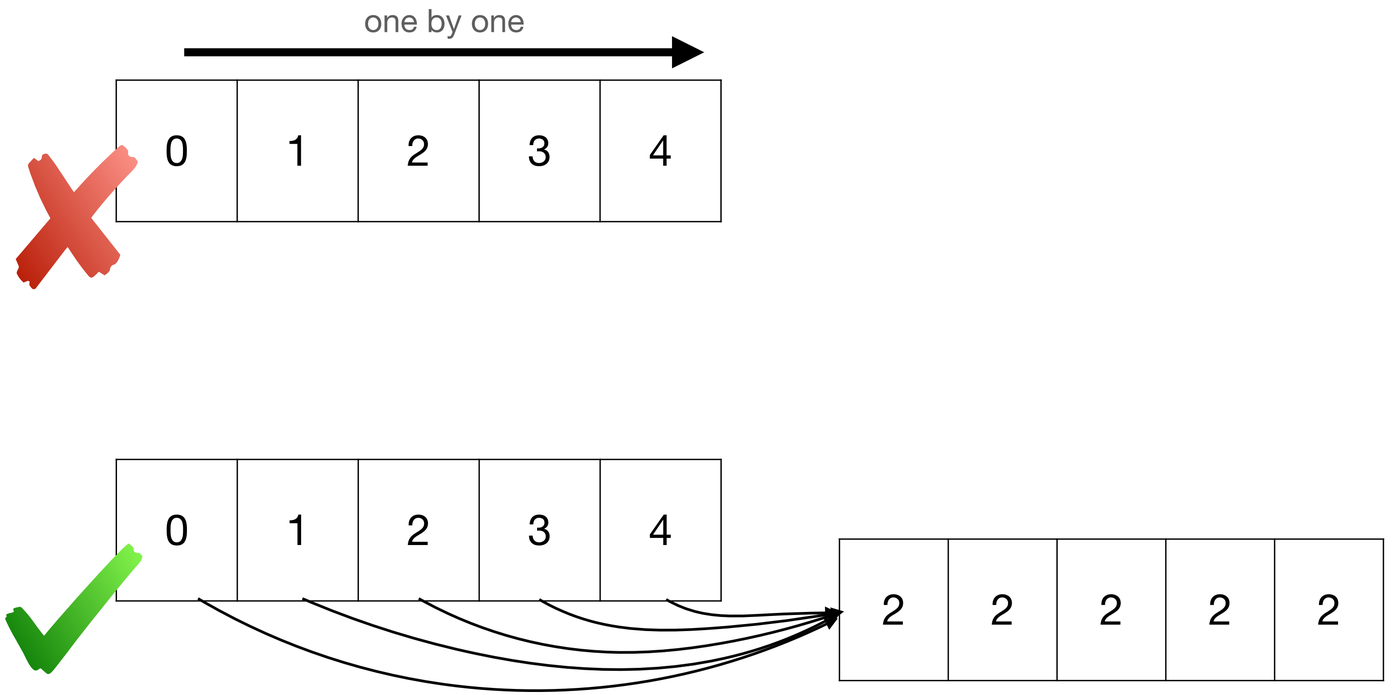




RASP

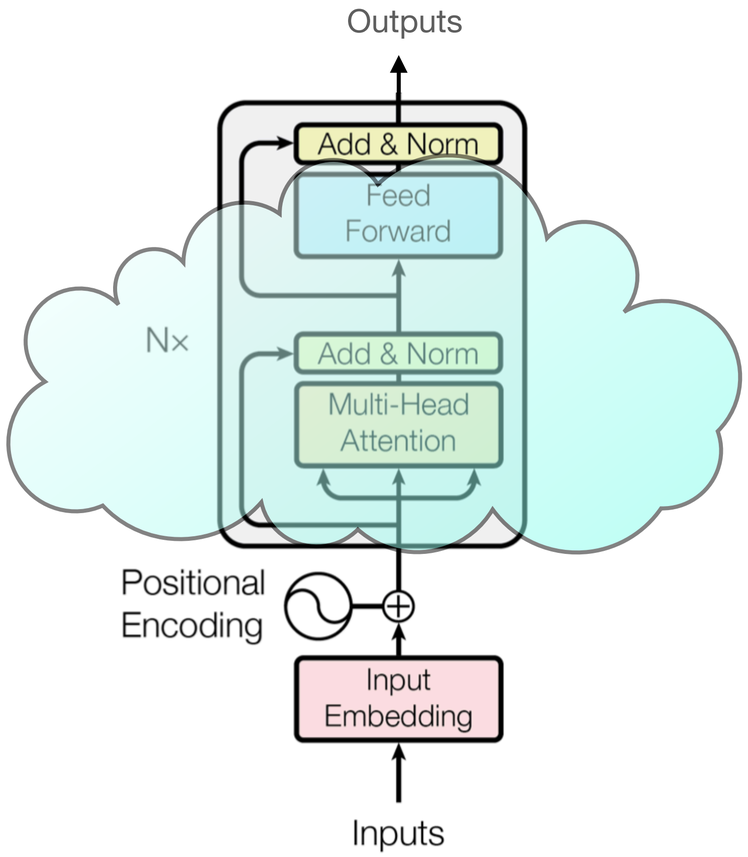
Restricted Access Sequence Processing

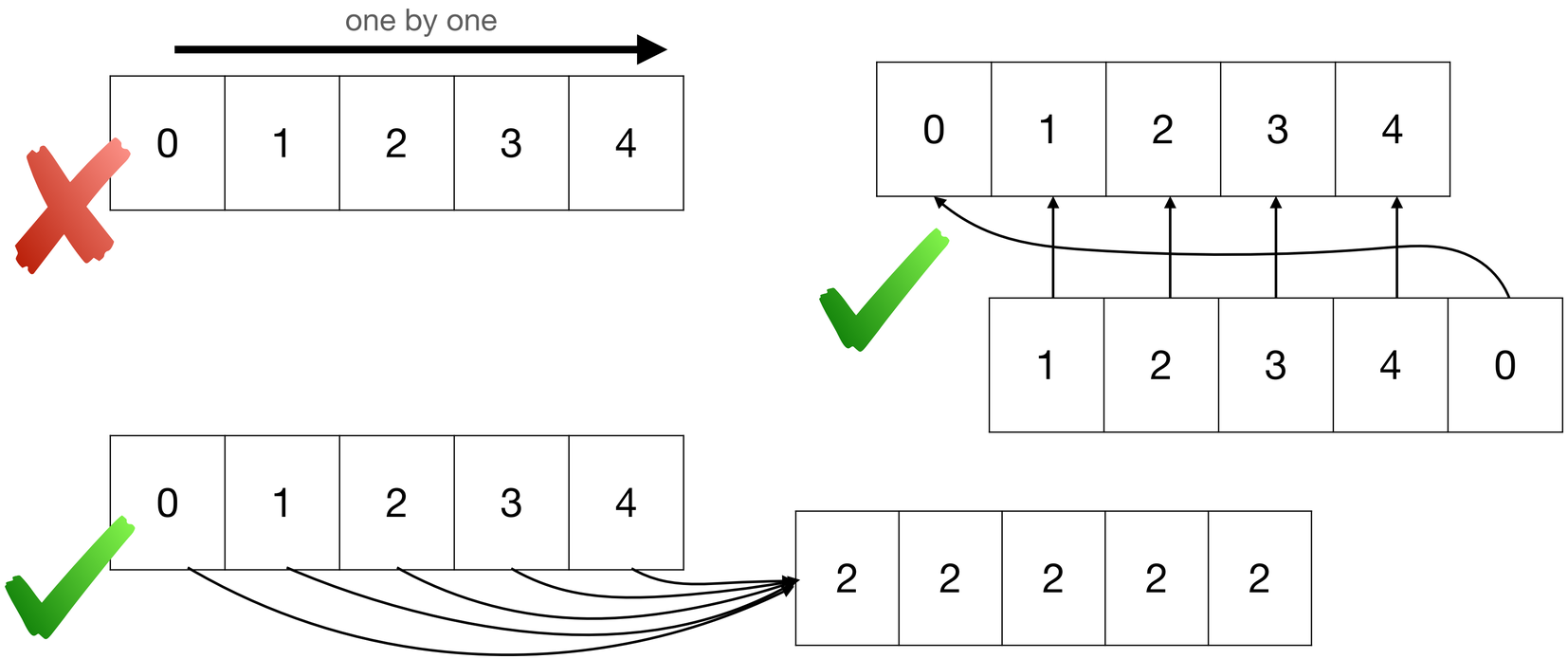




RASP

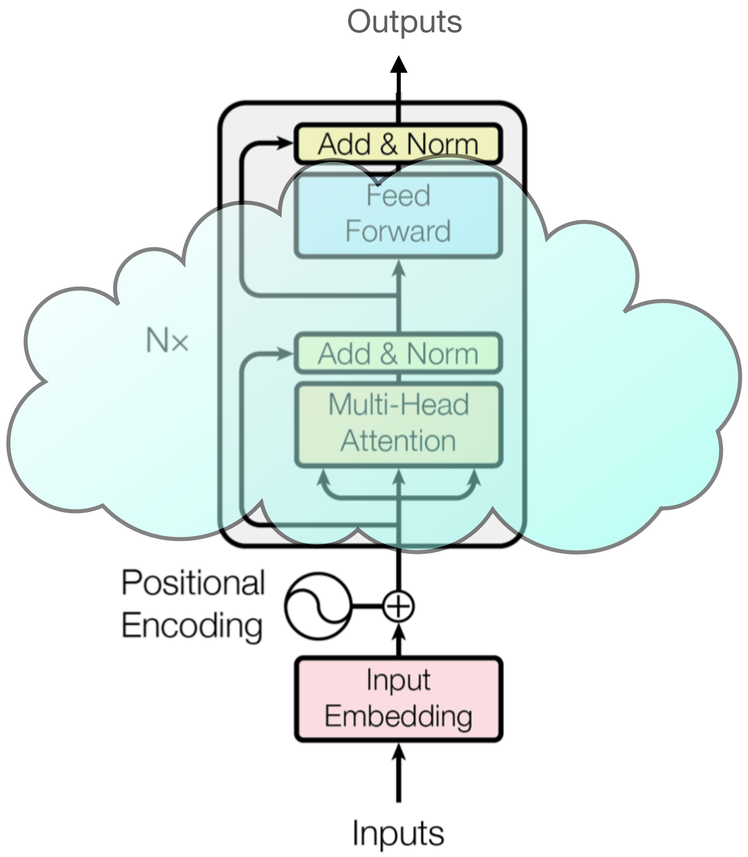
Restricted Access Sequence Processing

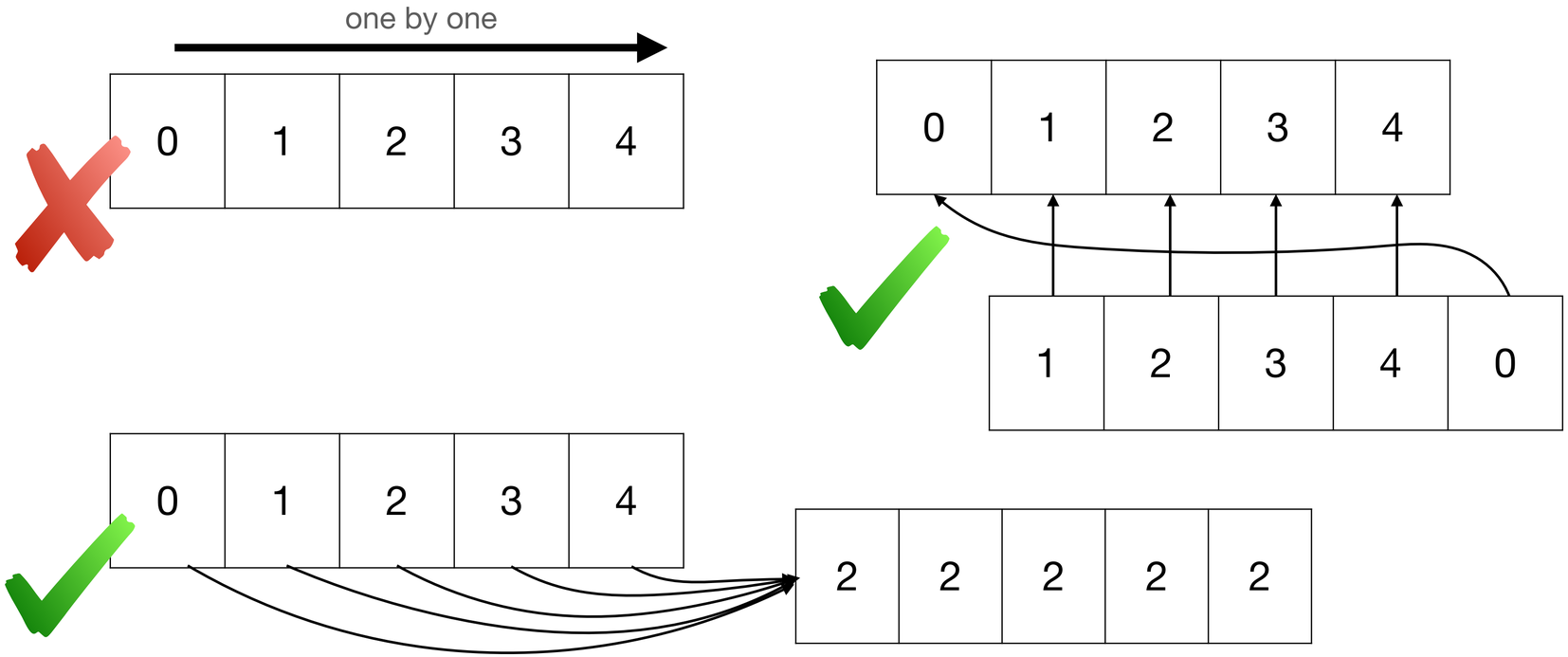




RASP

Restricted Access Sequence Processing





Symbolic:

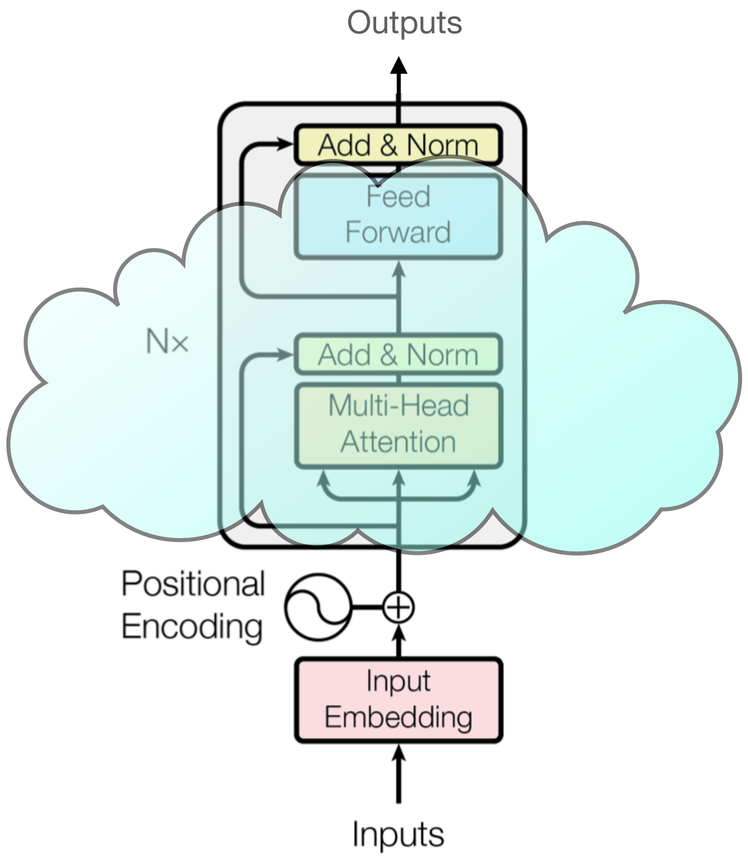
"A" 😀

instead of

[0.1,-0.2,0.65,…???] ☹️

RASP

Restricted Access Sequence Processing



Primitive Sequences:

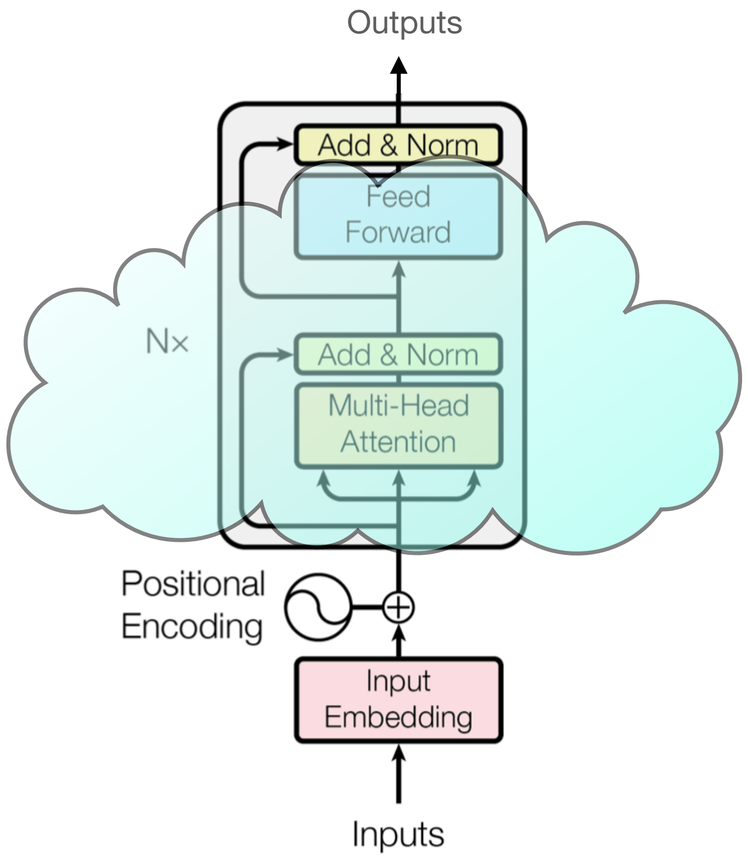
tokens = [R,A,S,P]

indices = [0,1,2,3]

length = [4,4,4,4]

RASP

Restricted Access Sequence Processing



Primitive Sequences:

tokens = [R,A,S,P]

indices = [0,1,2,3]

length = [4,4,4,4]

Elementwise Operations:

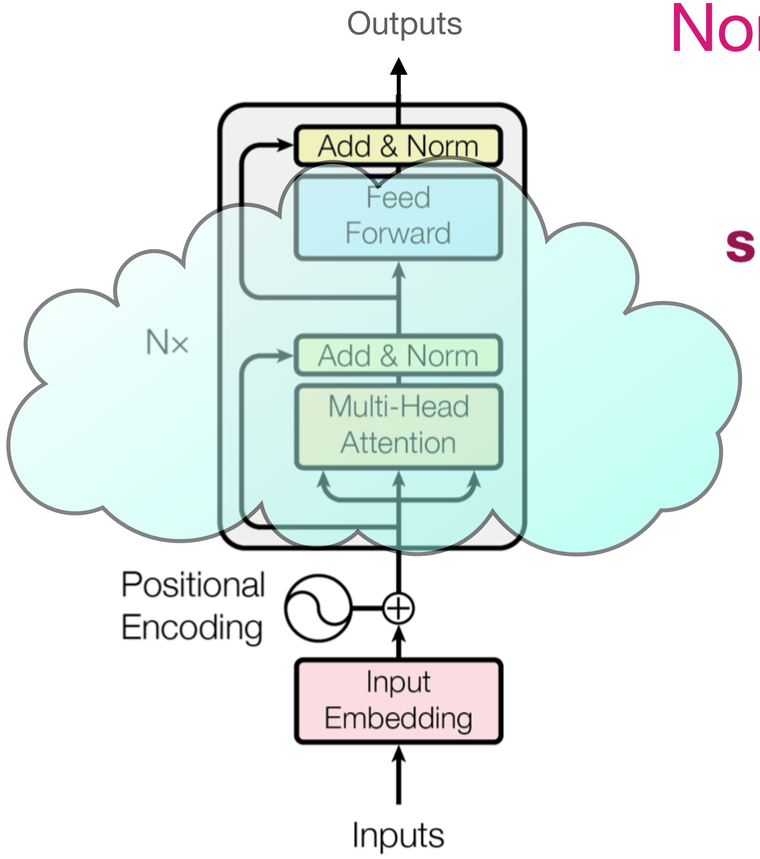
indices*2 = [0,2,4,6]

indices+length = [4,5,6,7]

...

RASP

Restricted Access Sequence Processing



Non-Elementwise Operations:

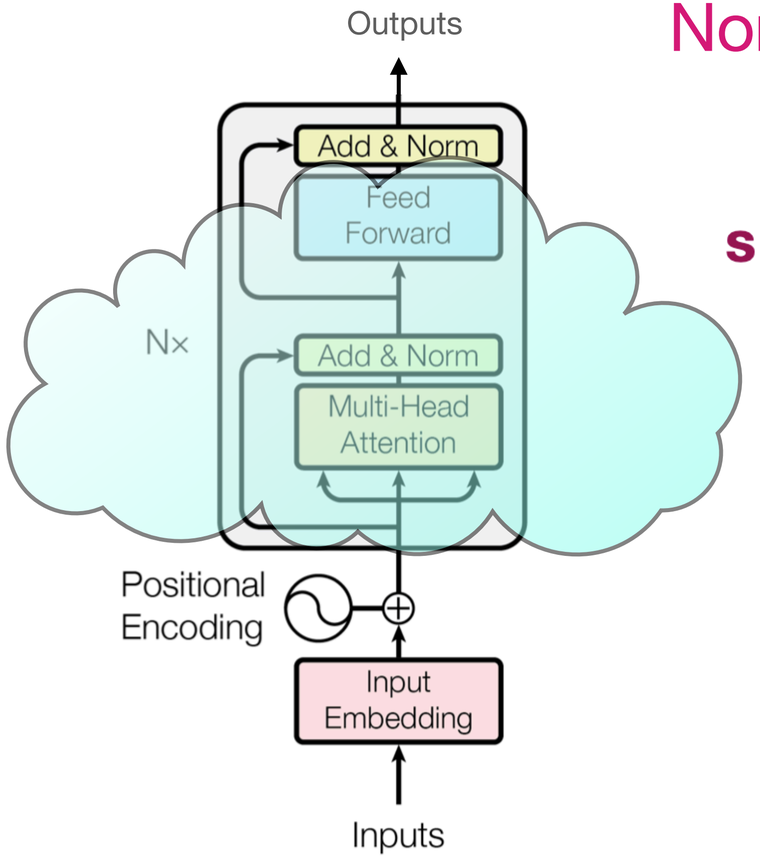
s = select([1,2,2],[0,1,2],==) **res=aggregate(s, [4,6,8])**

	1	2	2
0	F	F	F
1	T	F	F
2	F	T	T

			4	6	8				
F	F	F	4	6	8	=>	0		
T	F	F	4	6	8	=>	4	=>	[0,4,7]
F	T	T	4	6	8	=>	7		

RASP

Restricted Access Sequence Processing



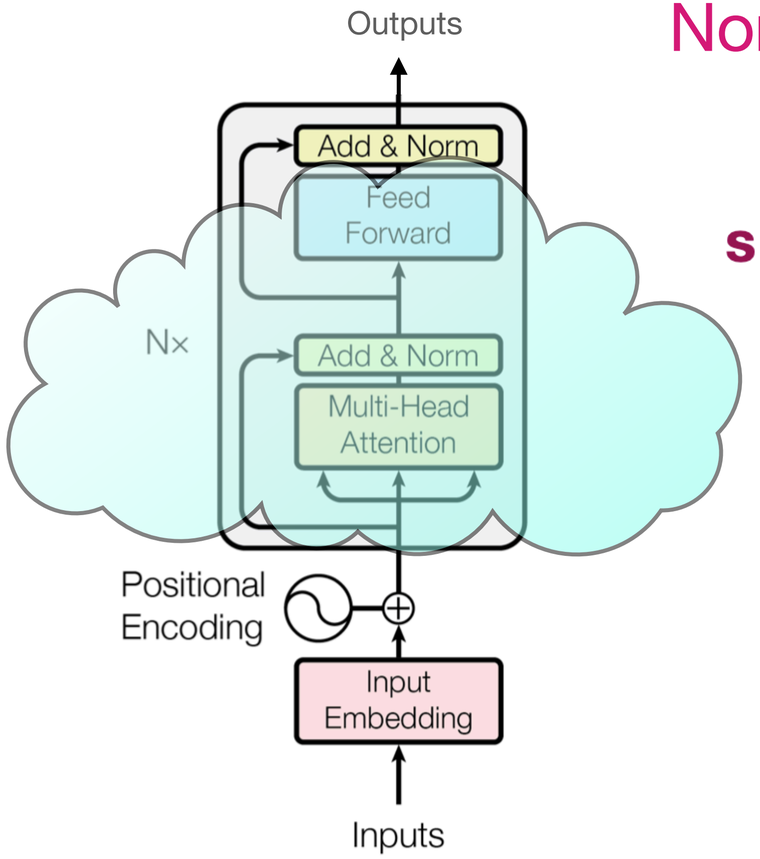
Non-Elementwise Operations:

s = select([1,2,2],[0,1,2],==)

	1	2	2
0	F	F	F
1	**T**	F	F
2	F	**T**	**T**

RASP

Restricted Access Sequence Processing



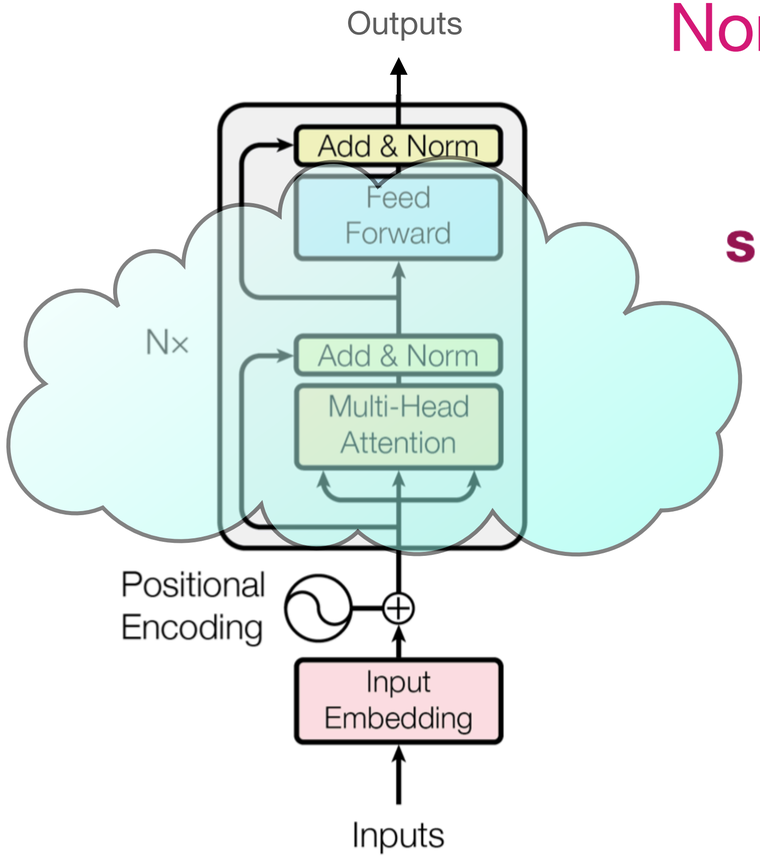
Non-Elementwise Operations:

s = select([1,2,2],[0,1,2],==)

	1	2	2
0	F	F	F
1	T	F	F
2	F	T	T

RASP

Restricted Access Sequence Processing



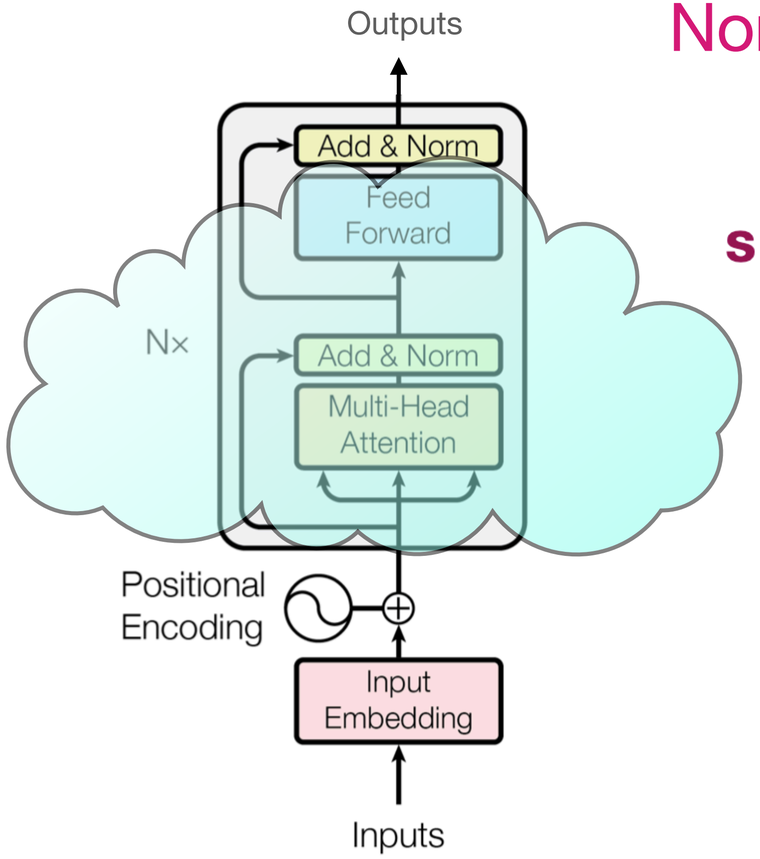
Non-Elementwise Operations:

s = select([1,2,2],[0,1,2],==)

	1	2	2
0	F	F	F
1	T	F	F
2	F	T	T

RASP

Restricted Access Sequence Processing



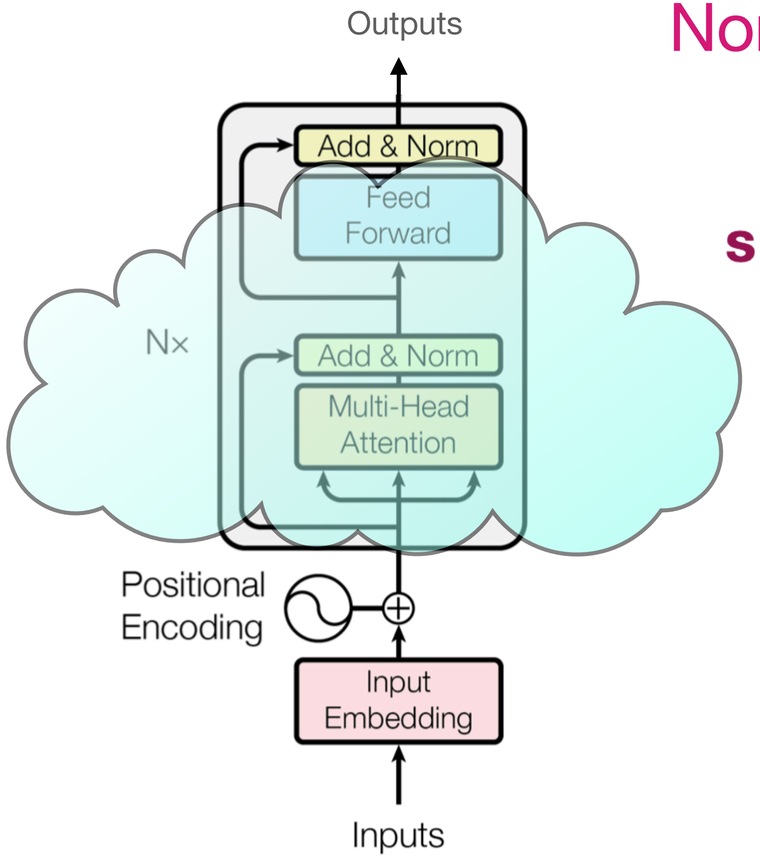
Non-Elementwise Operations:

s = select([1,2,2],[0,1,2],==)

	1	2	2
0	F	F	F
1	T	F	F
2	F	T	T

RASP

Restricted Access Sequence Processing



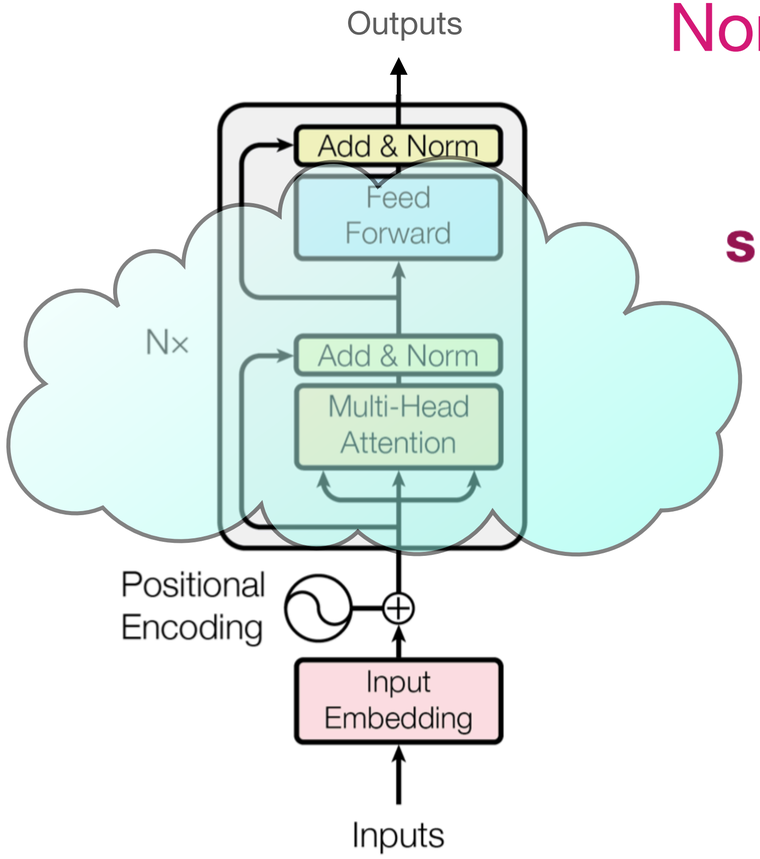
Non-Elementwise Operations:

s = select([1,2,2],[0,1,2],==)

	1	2	2
0	F	F	F
1	T	F	F
2	F	T	T

RASP

Restricted Access Sequence Processing



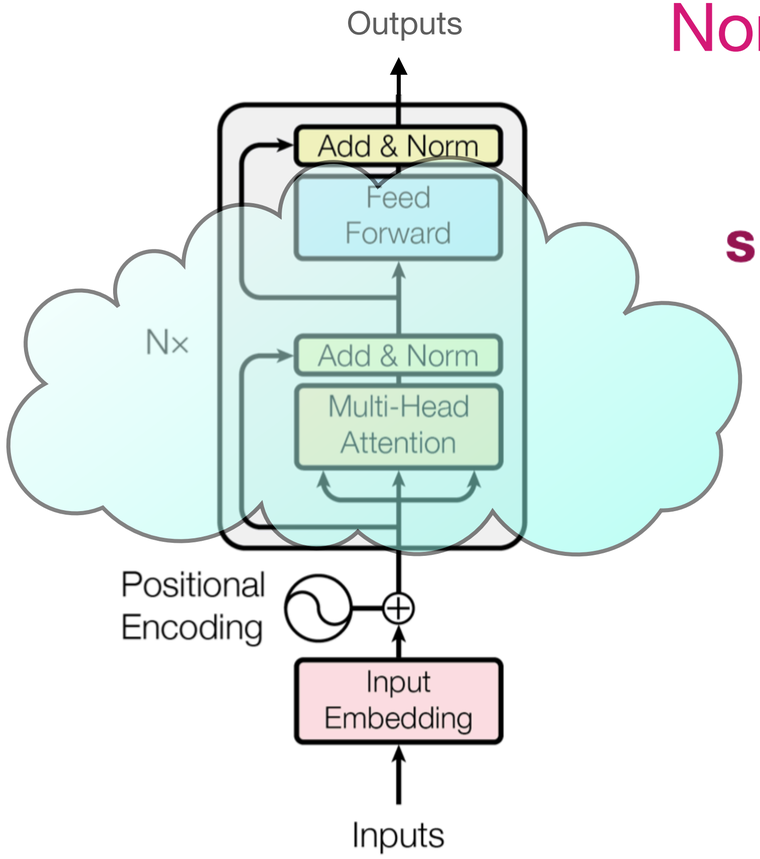
Non-Elementwise Operations:

s = select([1,2,2],[0,1,2],==)

	1	2	2
0	F	F	F
1	T	F	F
2	F	T	T

RASP

Restricted Access Sequence Processing



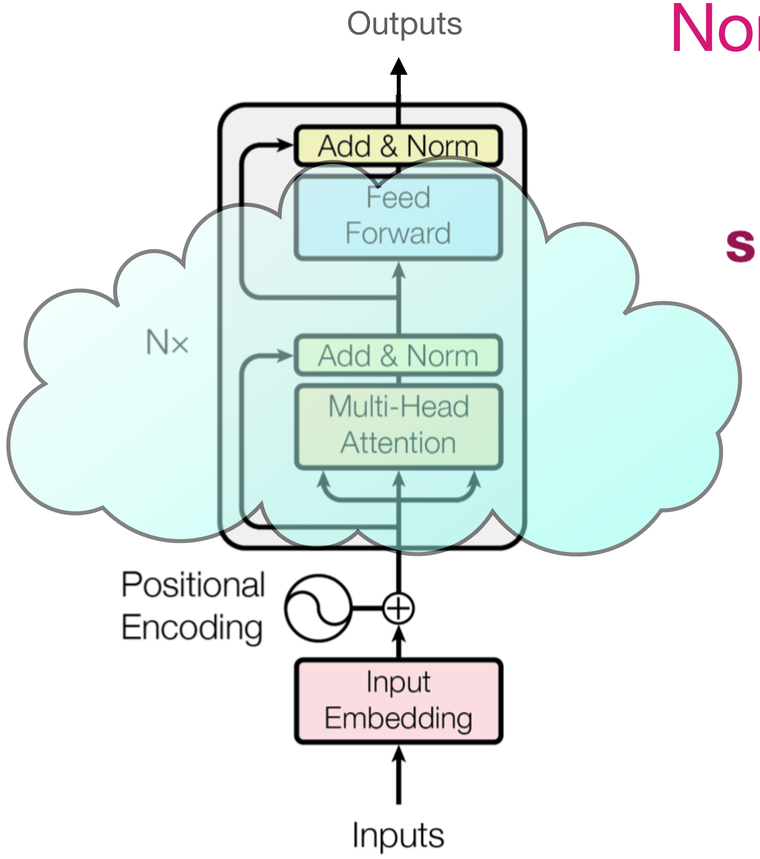
Non-Elementwise Operations:

s = select([1,2,2],[0,1,2],==)

	1	2	2
0	F	F	F
1	T	F	F
2	F	T	T

RASP

Restricted Access Sequence Processing



Non-Elementwise Operations:

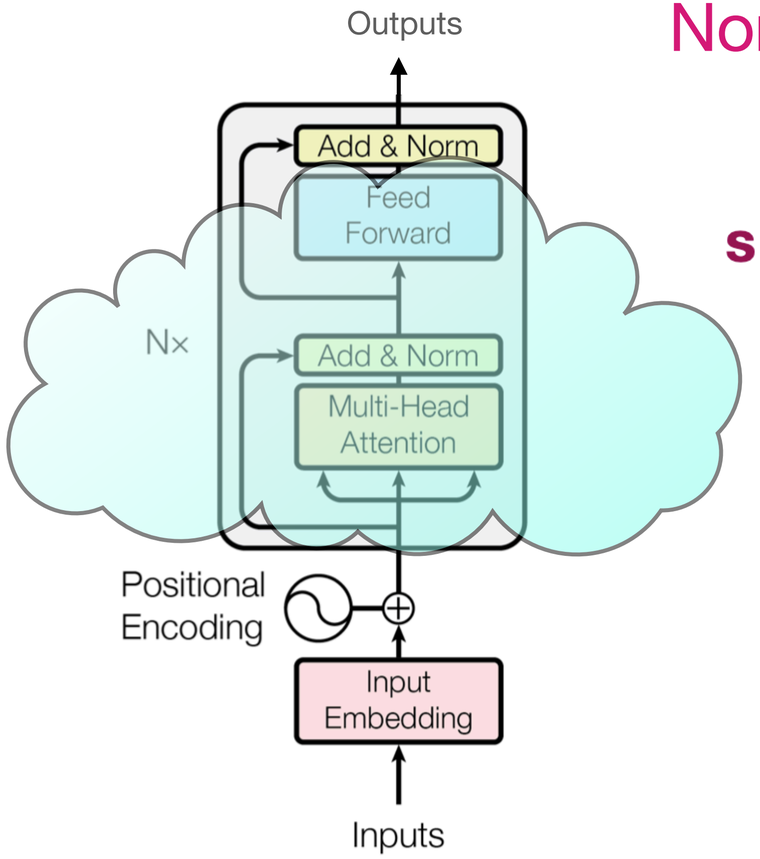
s = select([1,2,2],[0,1,2],==) **res=aggregate(s, [4,6,8])**

	1	2	2
0	F	F	F
1	T	F	F
2	F	T	T

			4	6	8				
F	F	F	4	6	8	=>	0		
T	F	F	4	6	8	=>	4	=>	[0,4,7]
F	T	T	4	6	8	=>	7		

RASP

Restricted Access Sequence Processing



Non-Elementwise Operations:

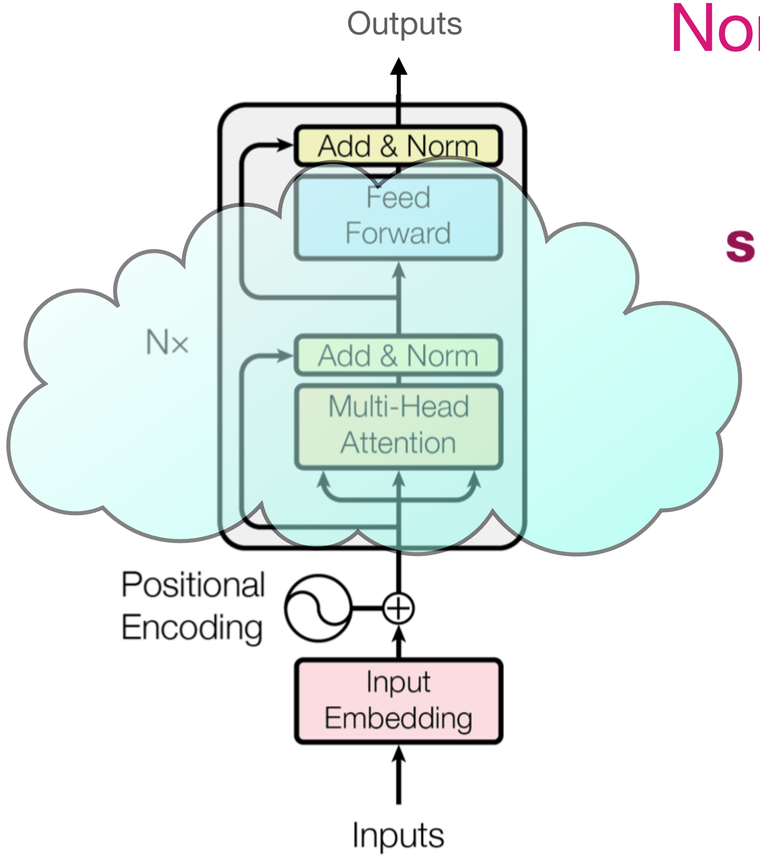
s = select([1,2,2],[0,1,2],==) **res=aggregate(s, [4,6,8])**

	1	2	2
0	F	F	F
1	T	F	F
2	F	T	T

			4	6	8			
F	F	F	4	6	8	=>	0	
T	F	F	4	6	8	=>	4	=> [0,4,7]
F	T	T	4	6	8	=>	7	

RASP

Restricted Access Sequence Processing



Non-Elementwise Operations:

s = select([1,2,2],[0,1,2],==) **res=aggregate(s, [4,6,8])**

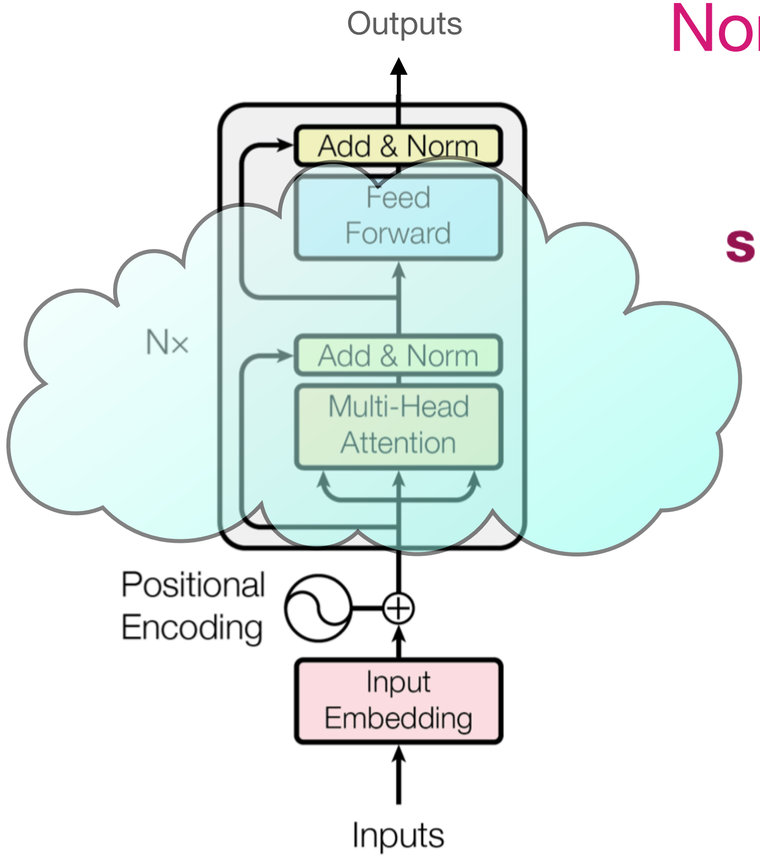
	1	2	2
0	F	F	F
1	T	F	F
2	F	T	T

			4	6	8		
F	F	F	4	6	8	=>	0
T	F	F	4	6	8	=>	4
F	T	T	4	6	8	=>	7

=> **[0,4,7]**

RASP

Restricted Access Sequence Processing



Non-Elementwise Operations:

s = select([1,2,2],[0,1,2],==) **res=aggregate(s, [4,6,8])**

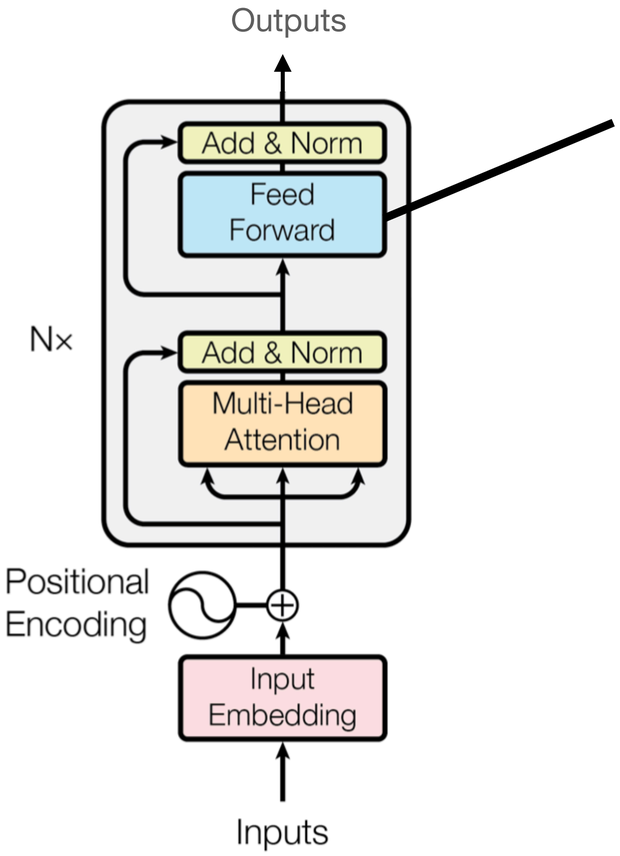
	1	2	2
0	F	F	F
1	T	F	F
2	F	T	T

			4	6	8		
F	F	F	4	6	8	=>	0
T	F	F	4	6	8	=>	4
F	T	T	4	6	8	=>	7

=> **[0,4,7]**

RASP

Restricted Access Sequence Processing



Feed Forward Sublayer

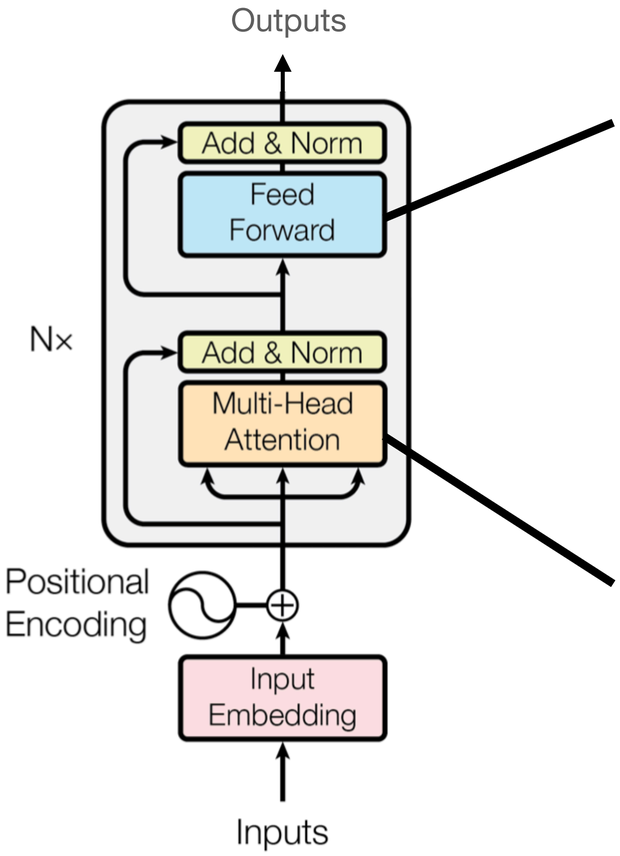
indices*2 = [0,2,4,6]

indices+length = [4,5,6,7]

...

RASP

Restricted Access Sequence Processing



Feed Forward Sublayer

Attention Head

indices*2 = [0,2,4,6]

indices+length = [4,5,6,7]

…

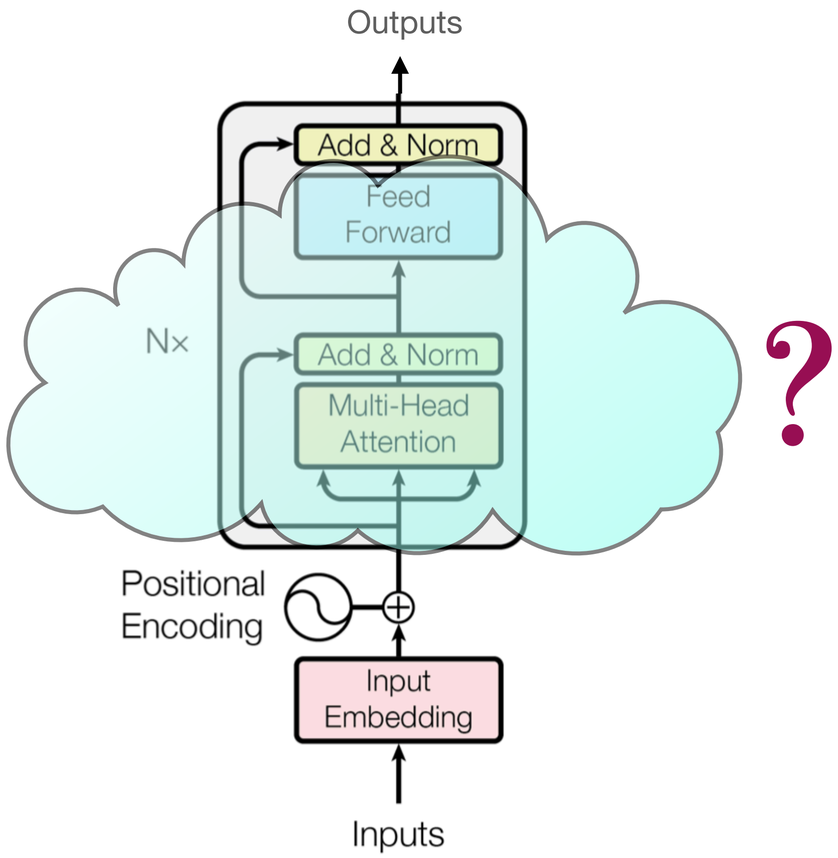
s = select([1,2,2],[0,1,2],==) **res=aggregate(s, [4,6,8])**

	1	2	2
0	F	F	F
1	T	F	F
2	F	T	T

			4	6	8			
F	F	F	4	6	8	=>	0	
T	F	F	4	6	8	=>	4	=> [0,4,7]
F	T	T	4	6	8	=>	7	

Can it solve “Reverse”?

abcde ↦ edcba

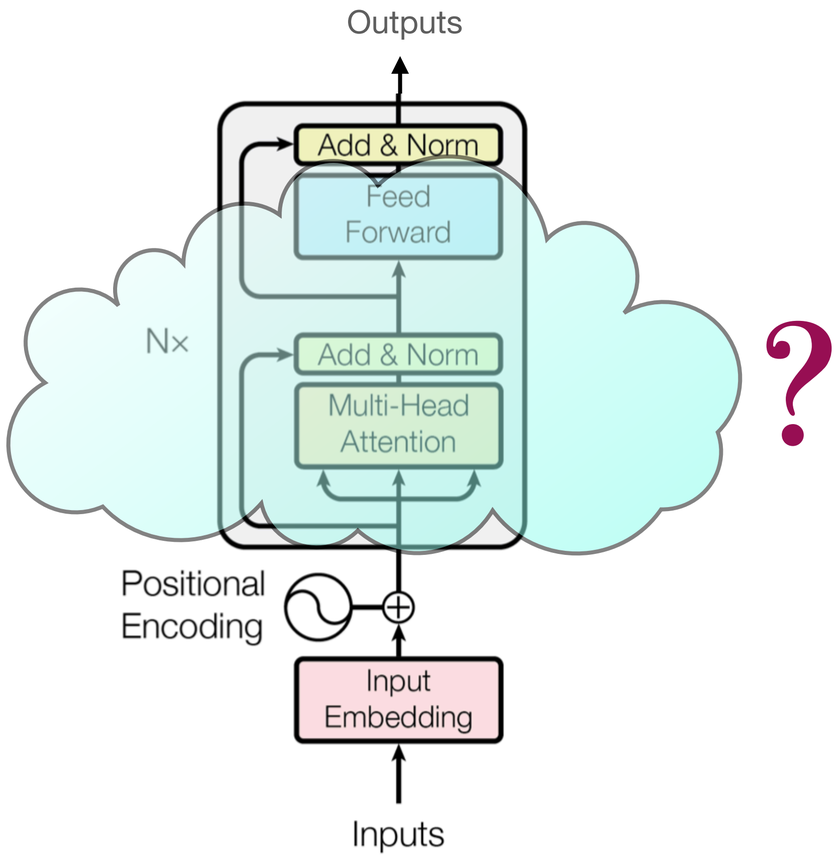


?

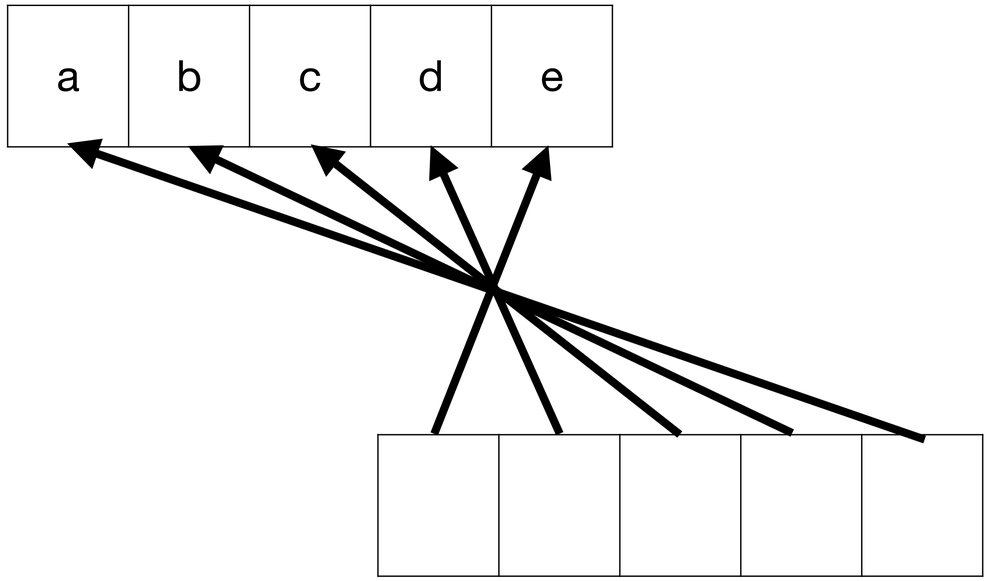
a	b	c	d	e

Can it solve “Reverse”?

abcde ↦ edcba

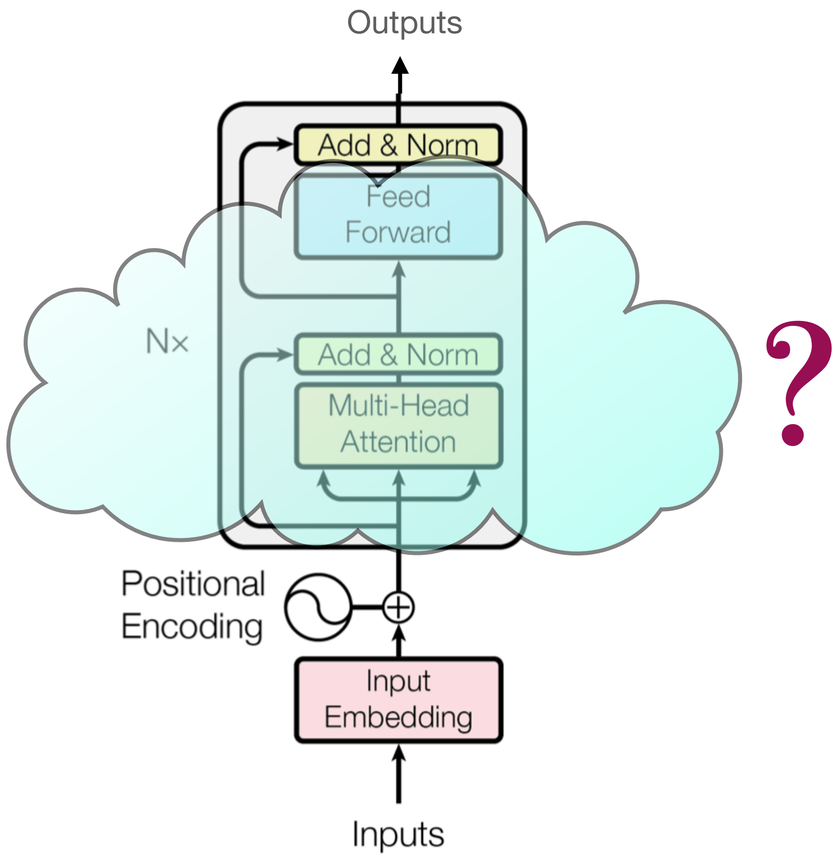


?

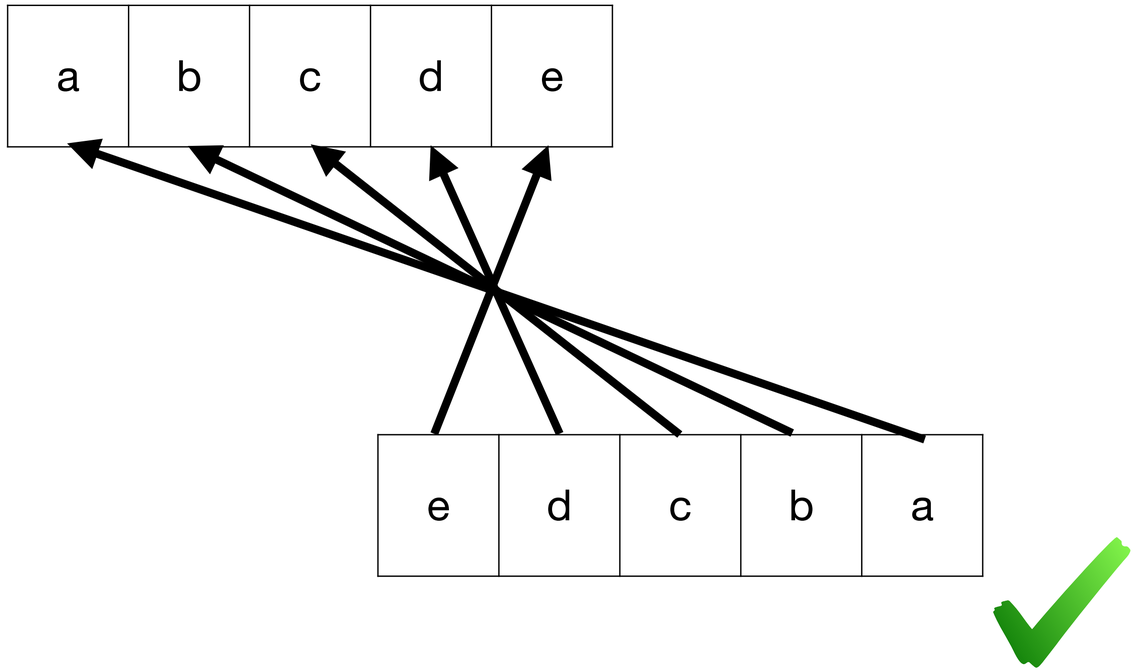


Can it solve “Reverse”?

abcde ↦ edcba



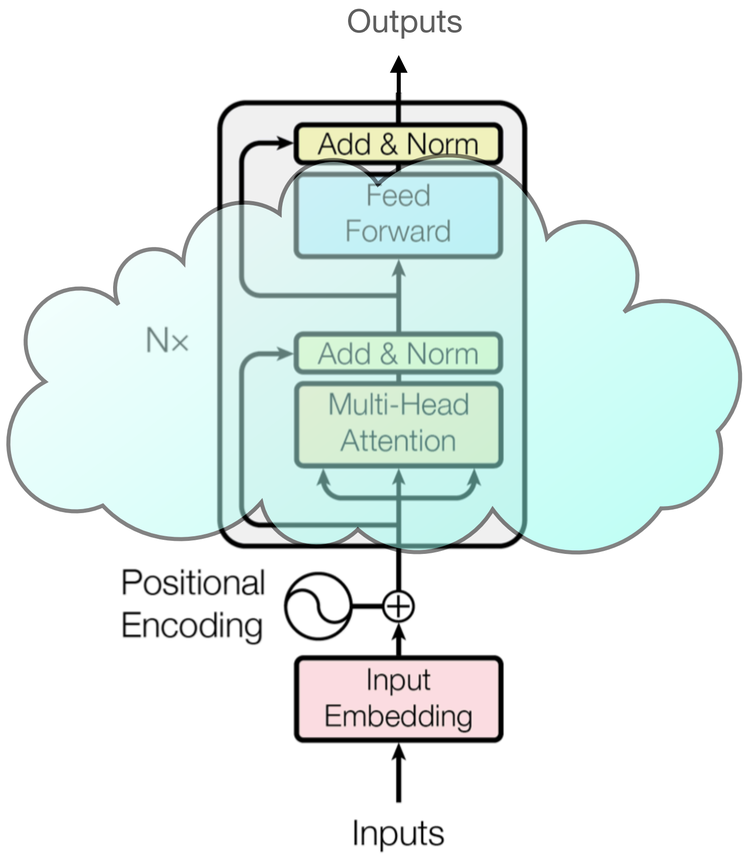
?



RASP

Restricted Access Sequence Processing

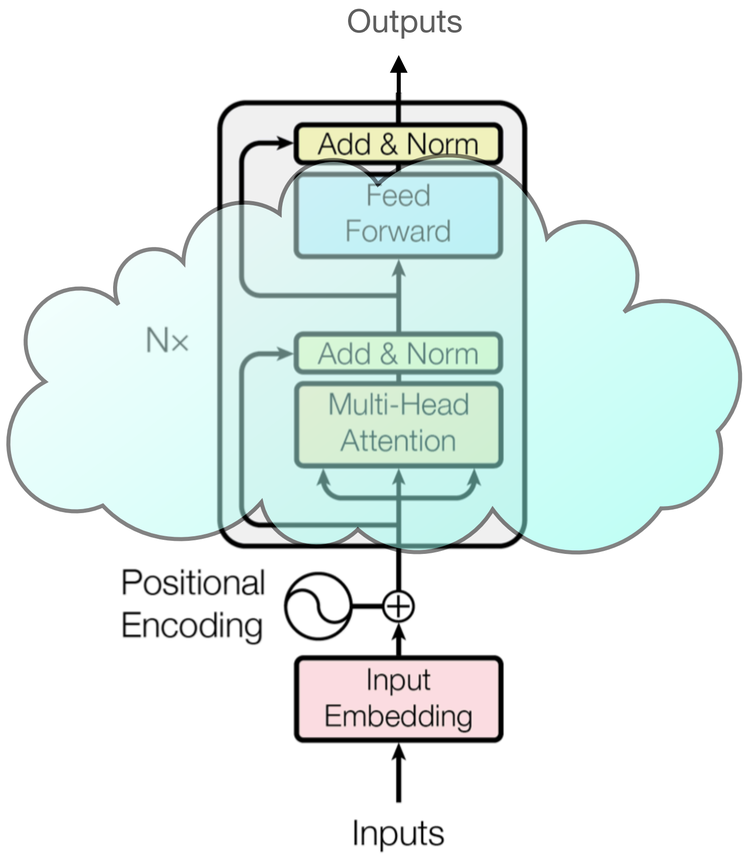
```
target = length-indices-1;
```
= [3,2,1,0]



RASP

Restricted Access Sequence Processing

```
target = length-indices-1;
flip = select(target,indices,==);
```



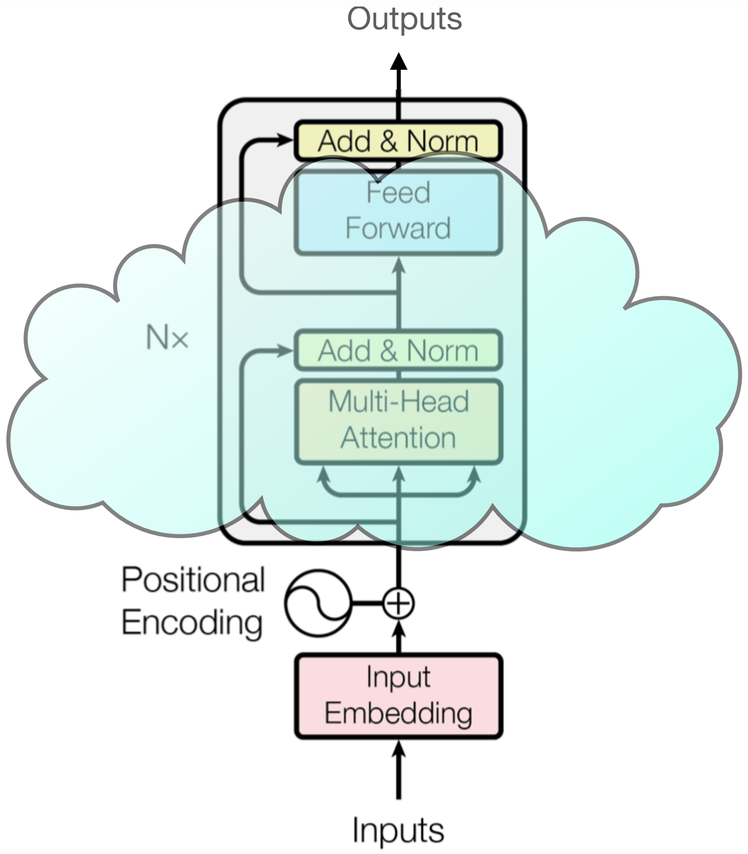
flip = select([3,2,1,0],[0,1,2,3],==)

	3	2	1	0
0	F	F	F	T
1	F	F	T	F
2	F	T	F	F
3	T	F	F	F

RASP

Restricted Access Sequence Processing

```
target = length-indices-1;
flip = select(target,indices,==);
reverse = aggregate(flip,tokens);
```



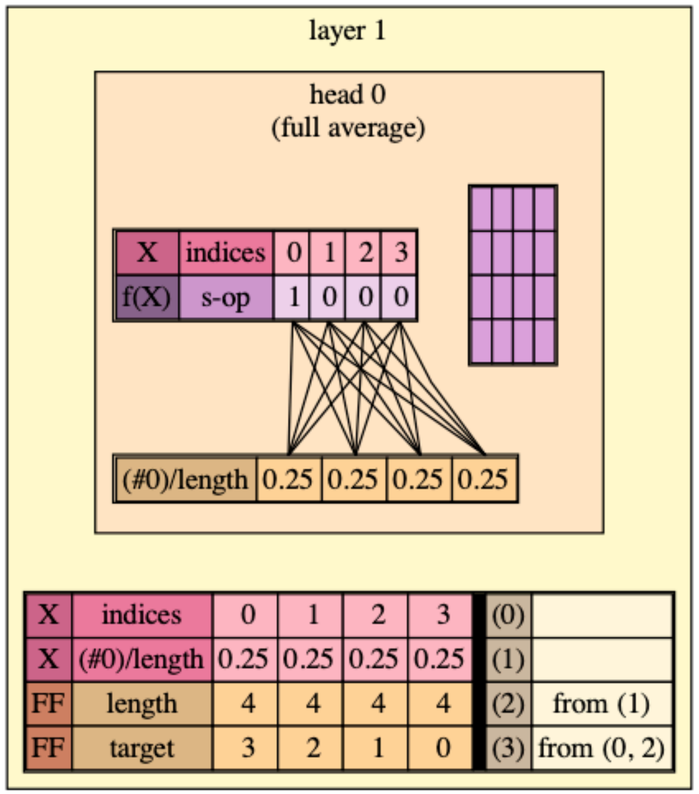
reverse=aggregate(flip, [R,A,S,P])

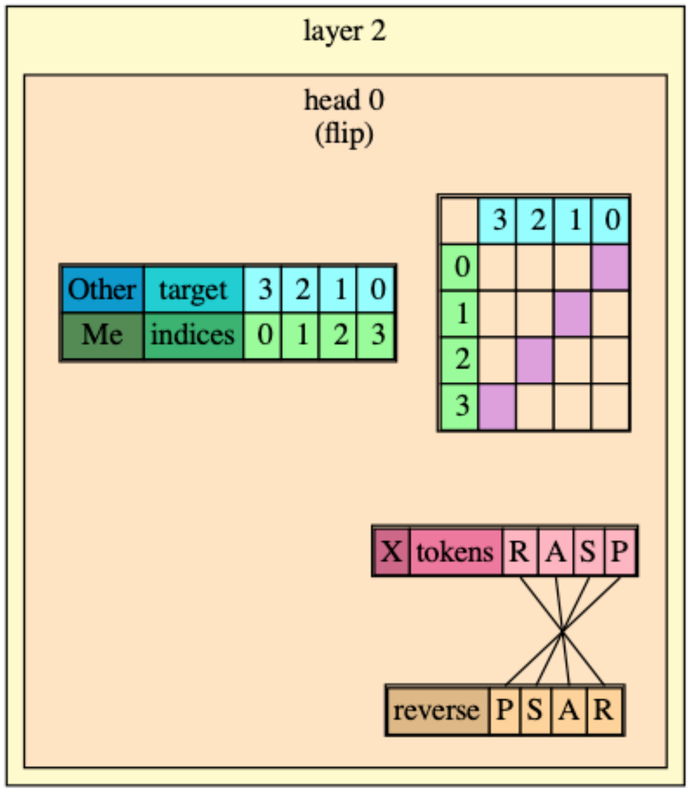
```
          R A S P
F F F T   R A S P  =>  P
F F T F   R A S P  =>  S  =>  [P,S,A,R]
F T F F   R A S P  =>  A
T F F F   R A S P  =>  R
```

RASP

Restricted Access Sequence Processing

```
target = length-indices-1;
flip = select(target,indices,==);
reverse = aggregate(flip,tokens);
```

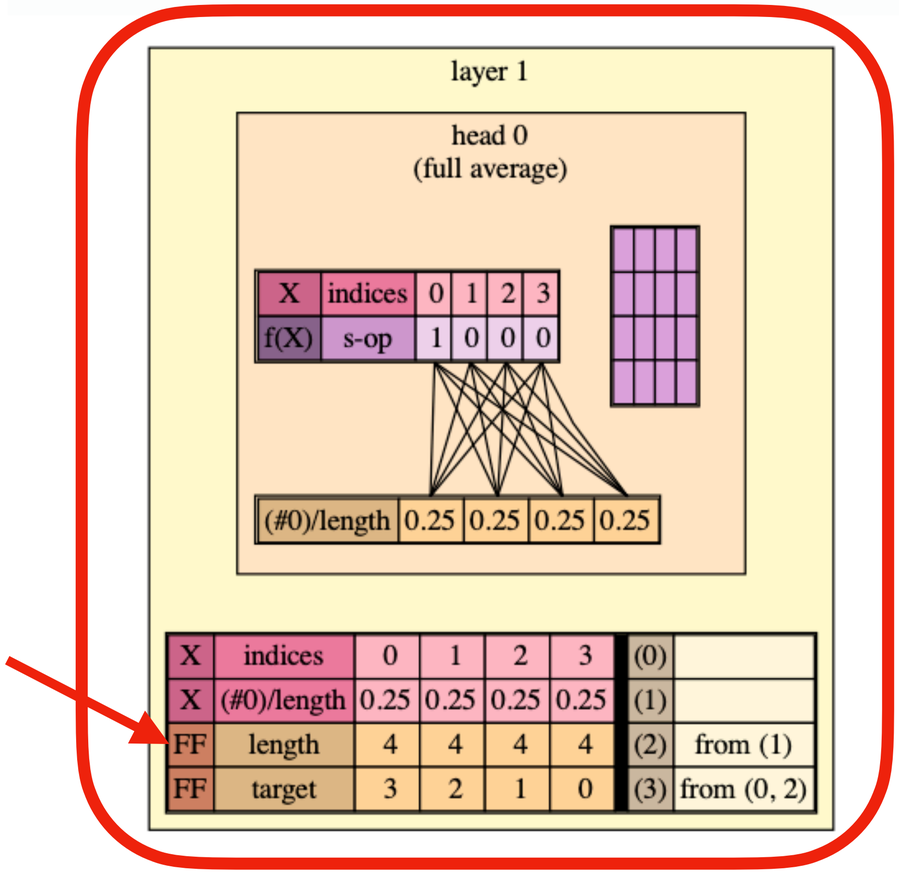


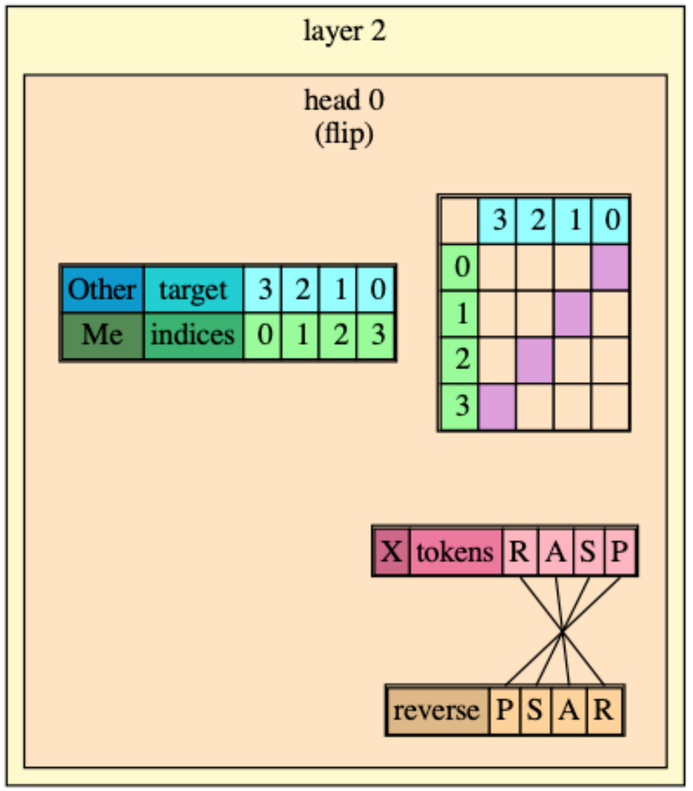


RASP

Restricted Access Sequence Processing

```
target = length-indices-1;
flip = select(target,indices,==);
reverse = aggregate(flip,tokens);
```

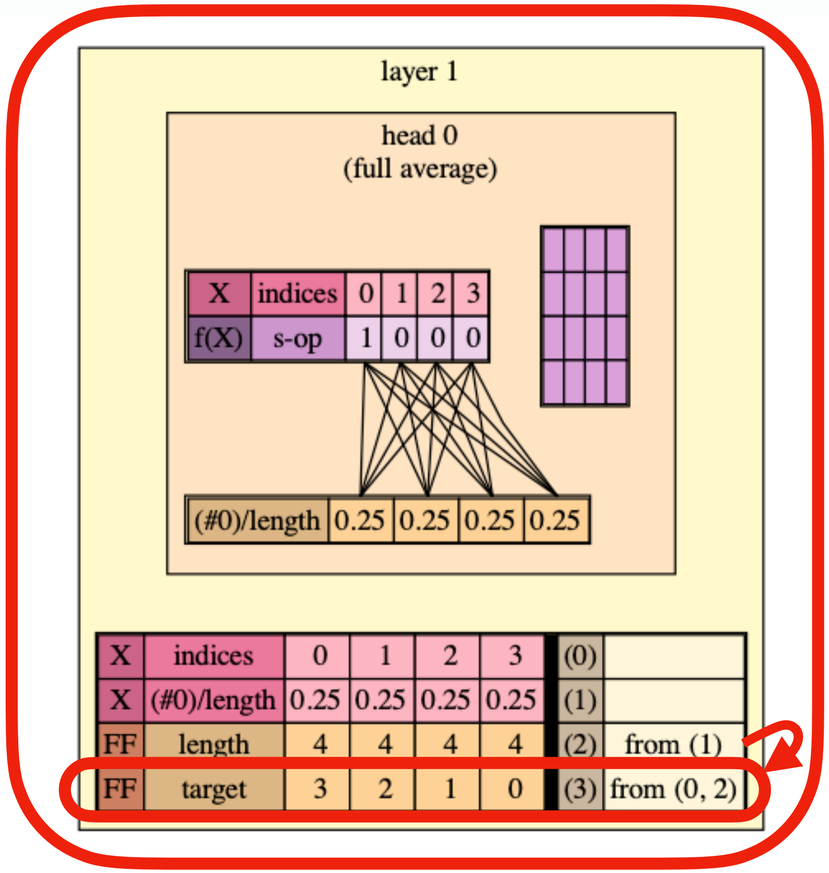


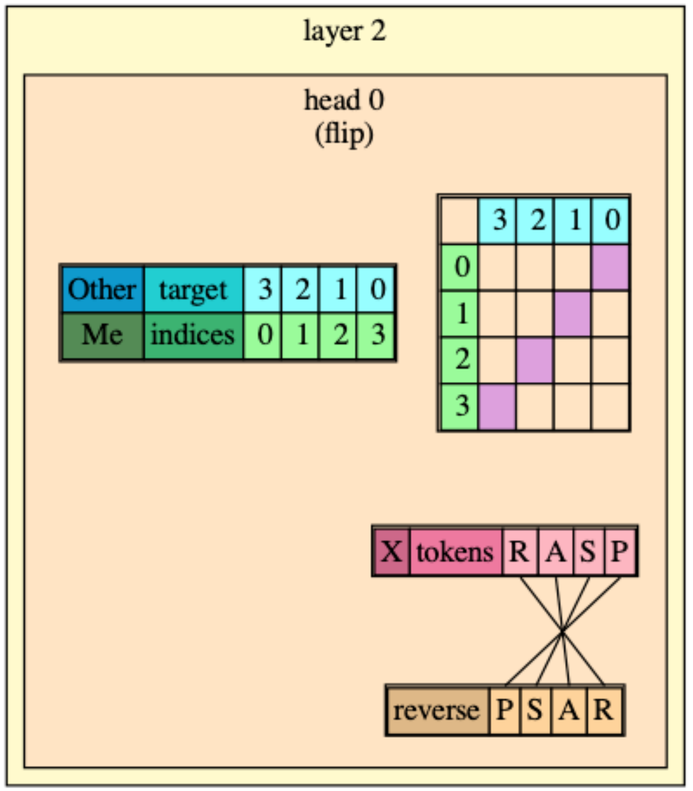


RASP

Restricted Access Sequence Processing

```
target = length-indices-1;
flip = select(target,indices,==);
reverse = aggregate(flip,tokens);
```

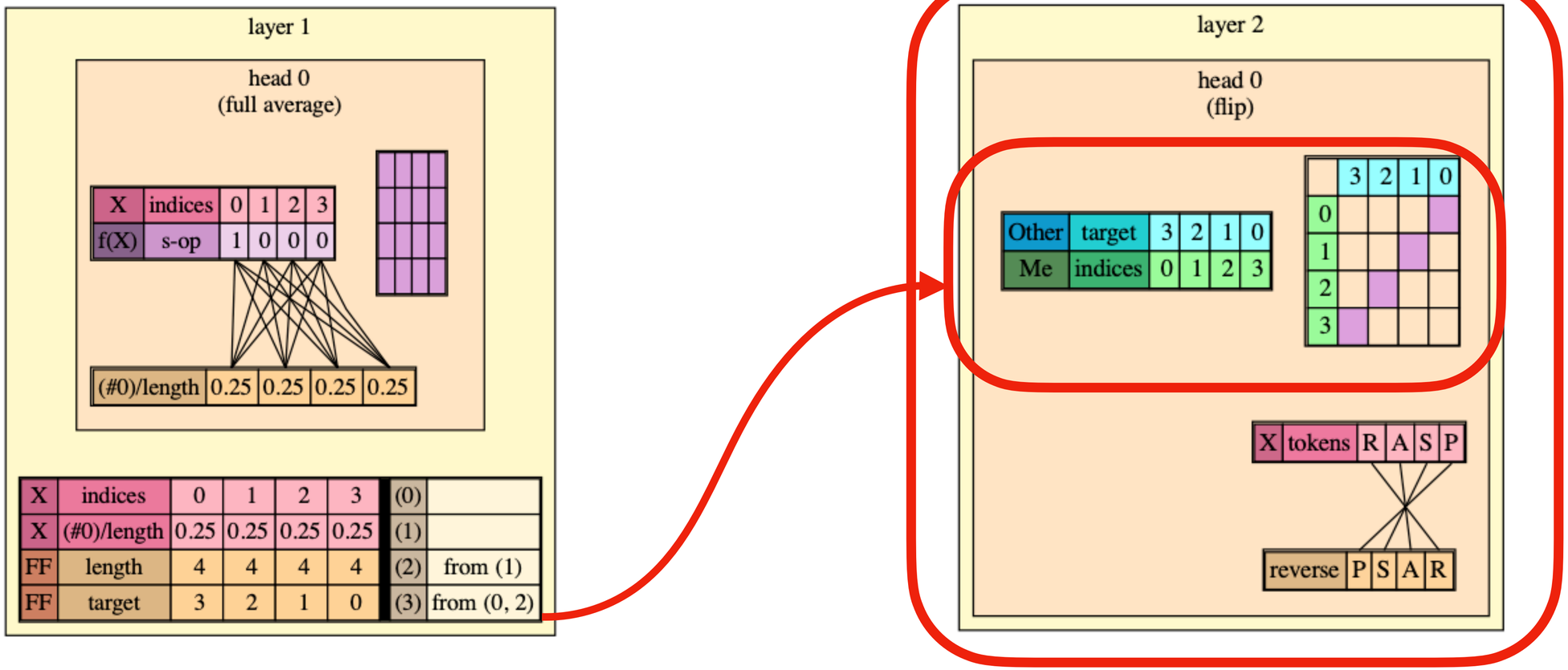




RASP

Restricted Access Sequence Processing

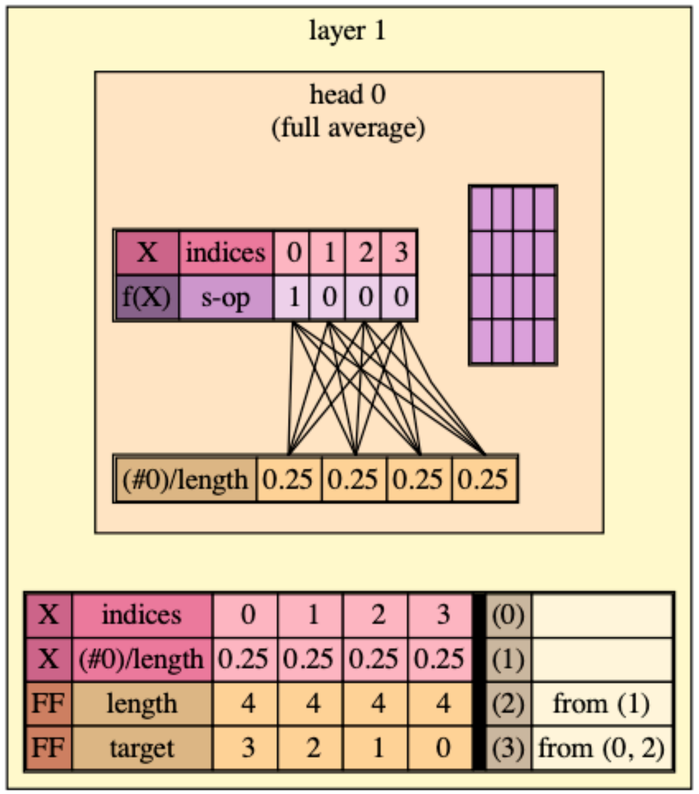
```
target = length-indices-1;
flip = select(target,indices,==);
reverse = aggregate(flip,tokens);
```

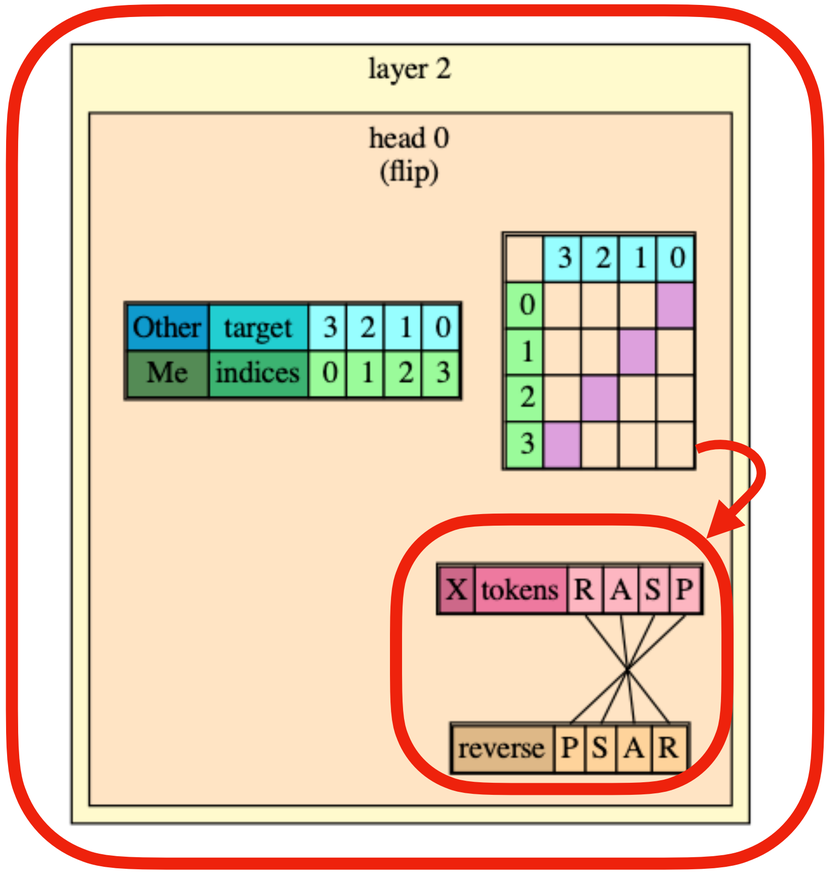


RASP

Restricted Access Sequence Processing

```
target = length-indices-1;
flip = select(target,indices,==);
reverse = aggregate(flip,tokens);
```

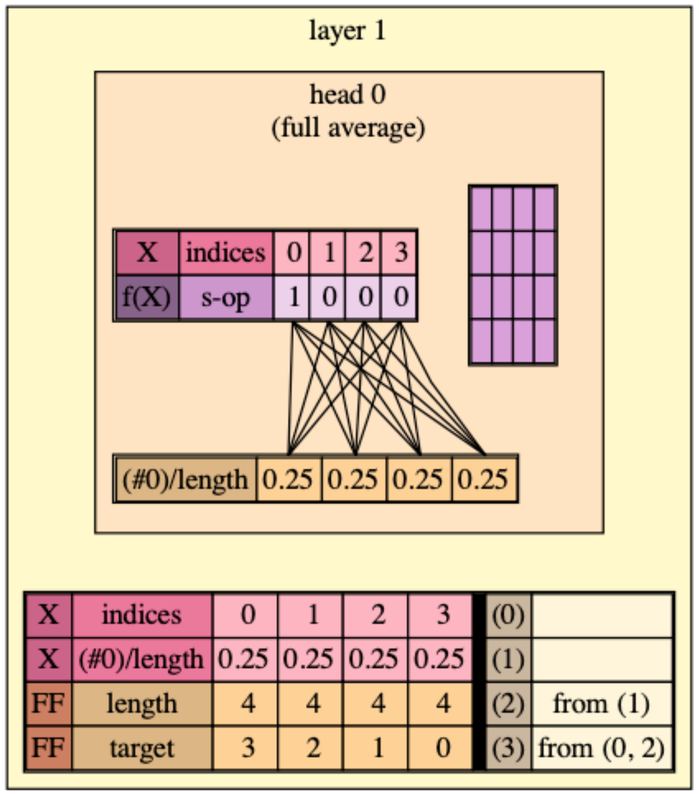


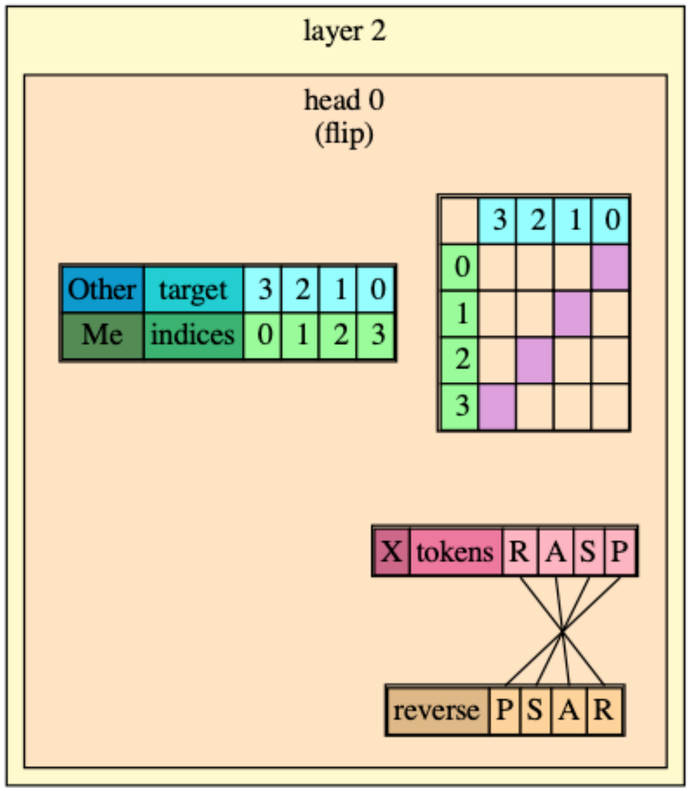


RASP

Restricted Access Sequence Processing

```
target = length-indices-1;
flip = select(target,indices,==);
reverse = aggregate(flip,tokens);
```

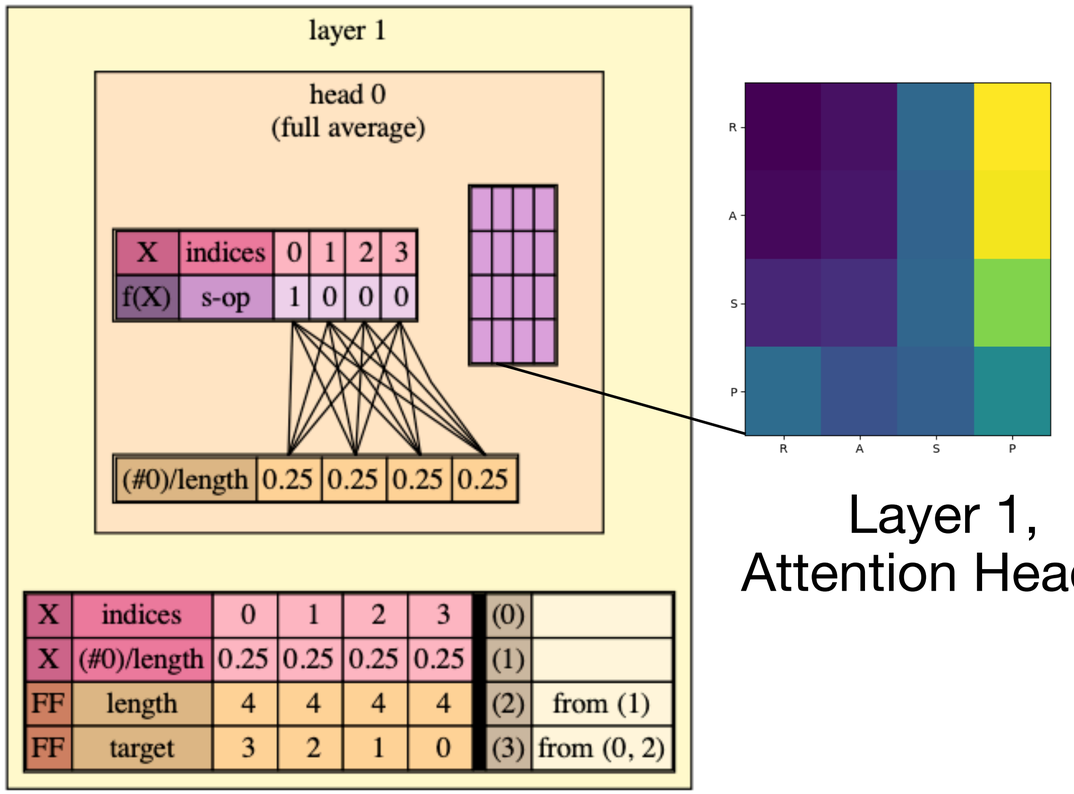




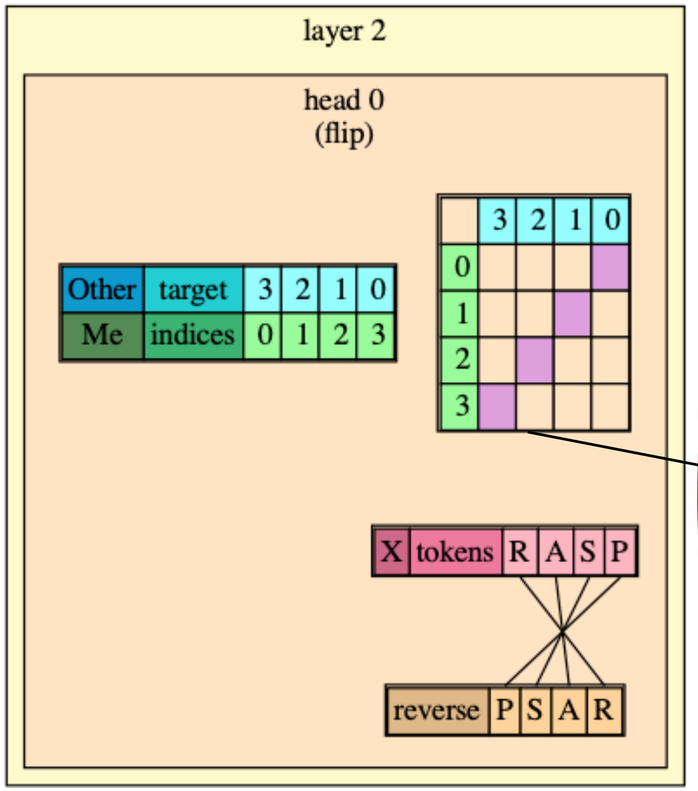
RASP

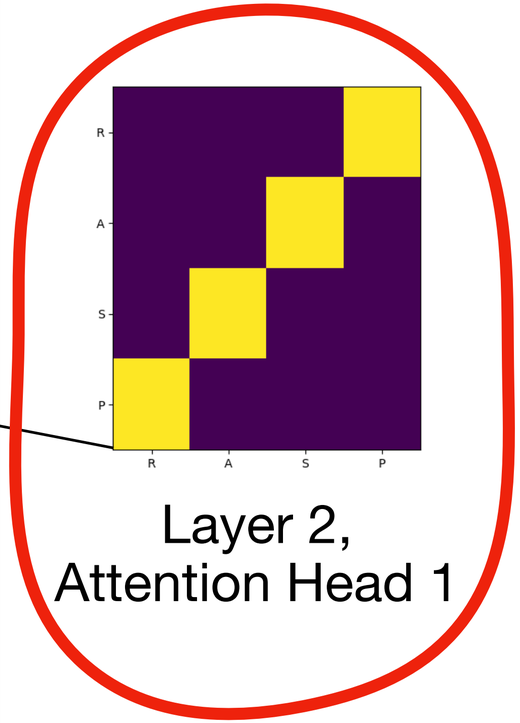
Restricted Access Sequence Processing

```
target = length-indices-1;
flip = select(target,indices,==);
reverse = aggregate(flip,tokens);
```



Layer 1,
Attention Head 1





Layer 2,
Attention Head 1

Conclusion

Try it out!! :

github.com/tech-srl/RASP

- RASP: abstraction for transformer-encoder
- Solve formal tasks in transformer-encoders
  - Even Dyck-k for arbitrary k and unbounded depth!
- Translate to neural transformer-encoders architectures
  - Exact weights still required

More details in paper!