A Survey of LLMs

Zhao et. al

Anthony J. Dsouza

Date - 23.09.2024

Table of Contents

Table of Contents - Scaling and Need for scaling

What are LLMs?

What are LLMs?

What factors to consider when calling an LM “Large”?

What are LLMs?

What factors to consider when calling an LM “Large”?

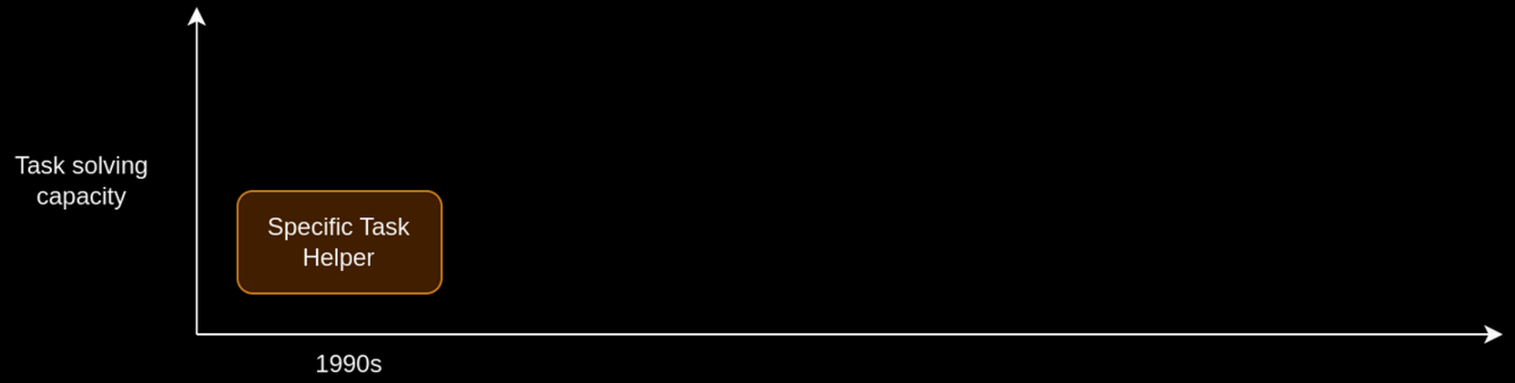
- **Size**

What are LLMs?

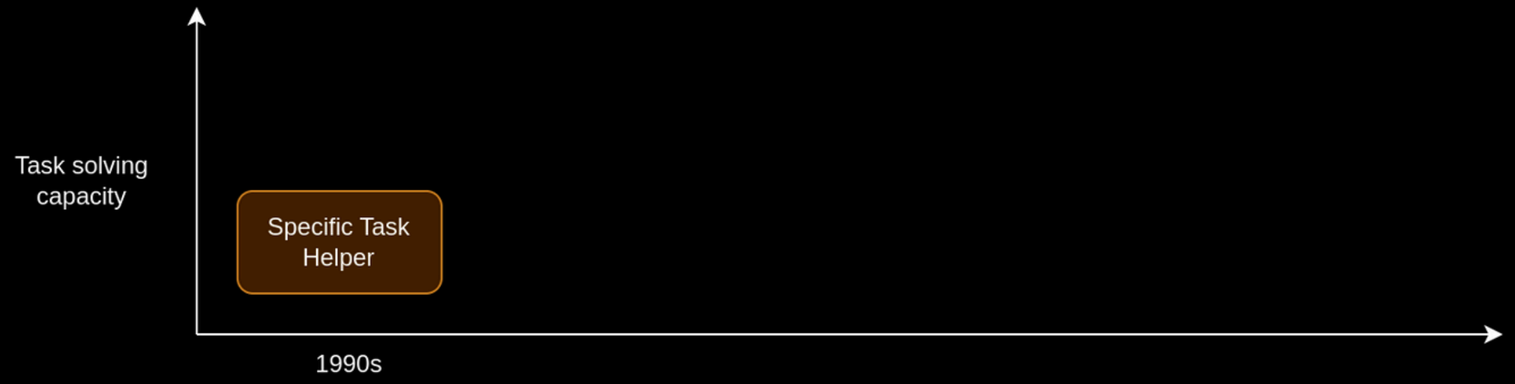
What factors to consider when calling an LM “Large”?

- **Size**
- **Abilities**

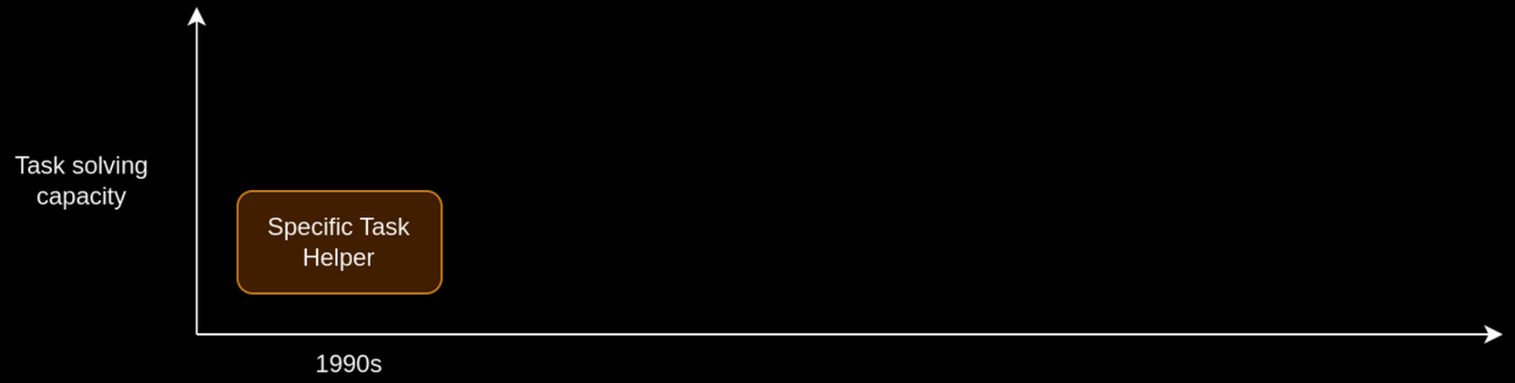
ngram models
based on probability estimation
based on _____ assumption



ngram models
based on probability estimation
based on markov assumption

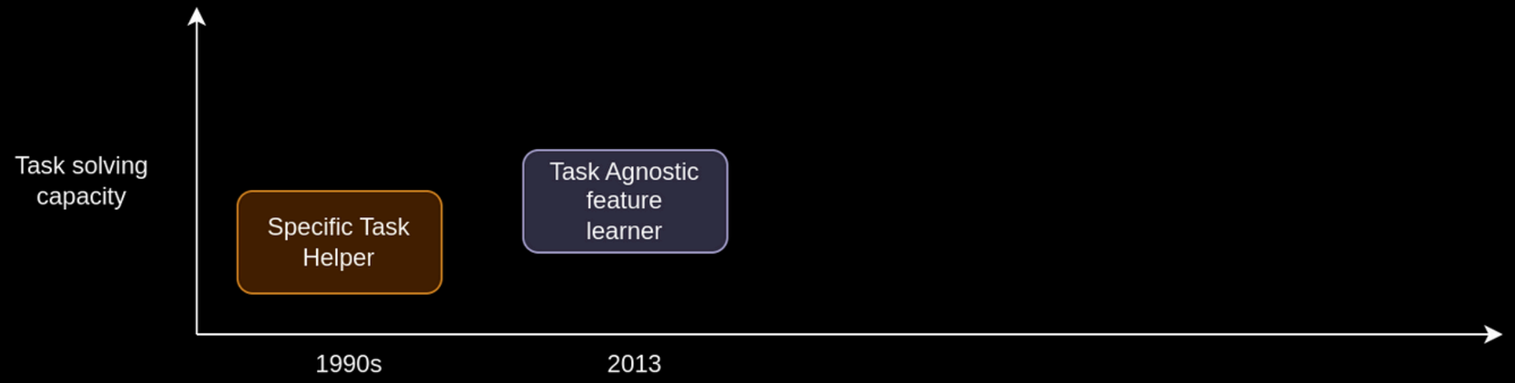


ngram models
based on probability estimation
based on markov assumption
suffer from curse of dimensionality - but why?

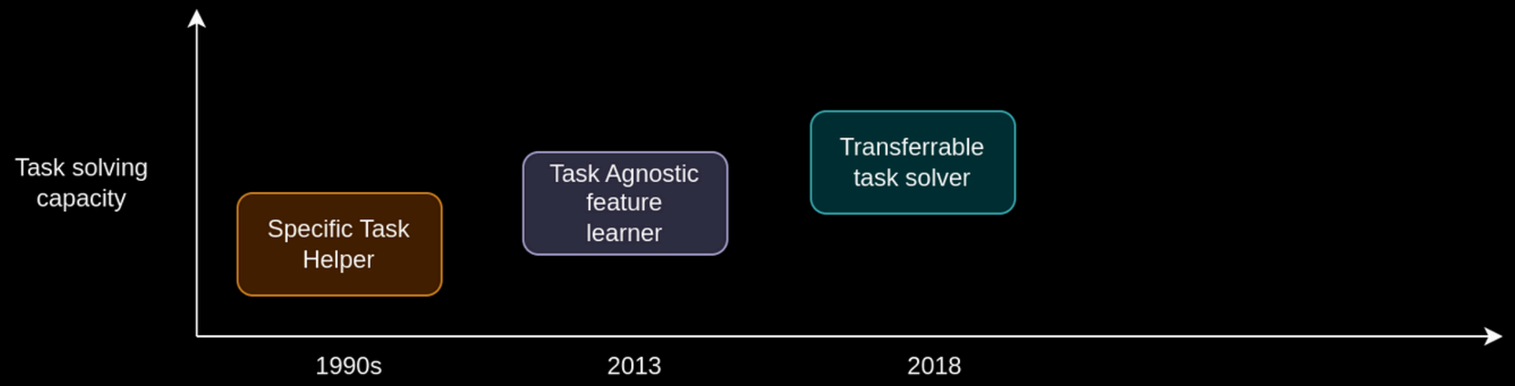


ngram models
based on probability estimation
based on markov assumption
suffer from curse of dimensionality - difficult to model all transition probabilities

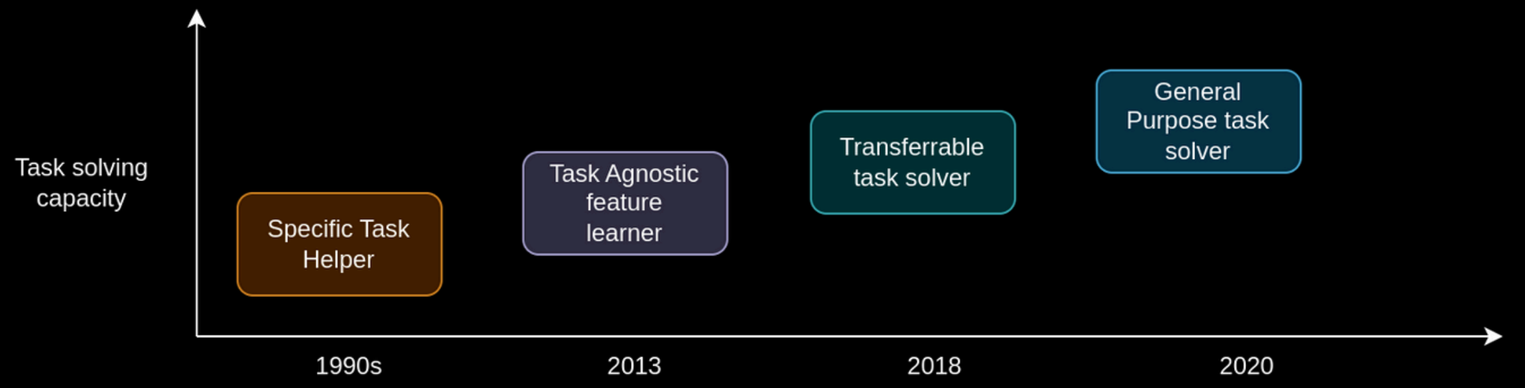
Static word representations
Neural context modelling
word2vec / GloVE
Distributed representation of words
Predicting words based upon surrounding words



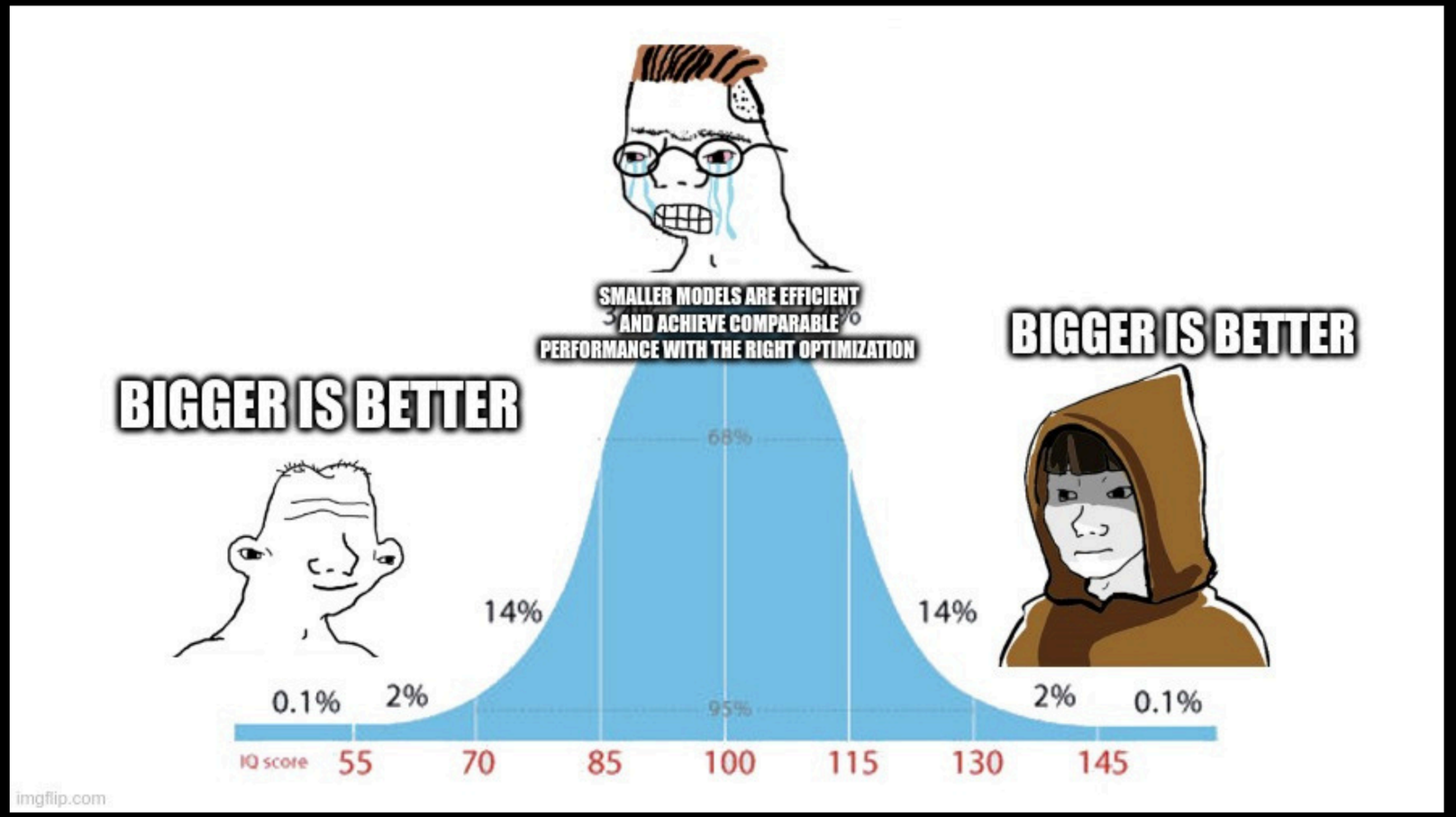
Context aware representatoins
pretrain on large corpora
finetune on task specific corpora



Scale PLM
Use diverse data

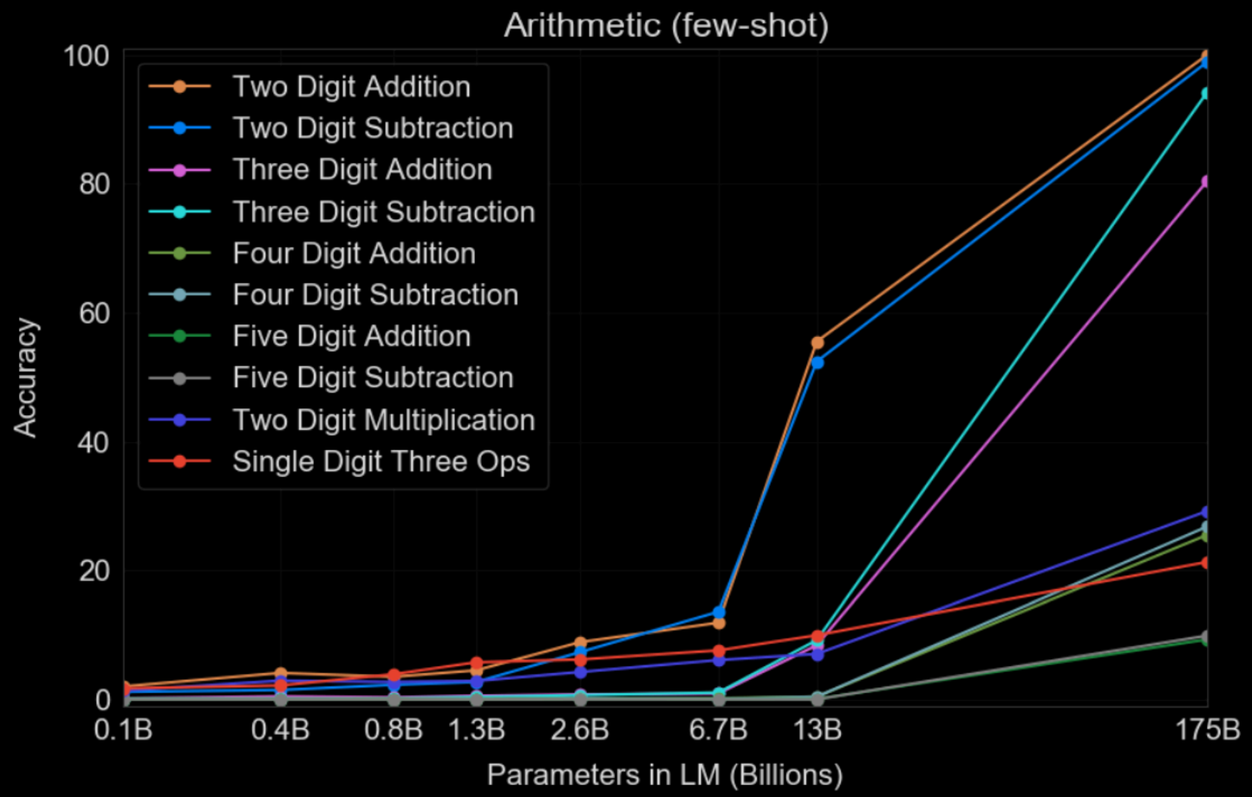


Scaling Laws


SMALLER MODELS ARE EFFICIENT AND ACHIEVE COMPARABLE PERFORMANCE WITH THE RIGHT OPTIMIZATION
BIGGER IS BETTER
BIGGER IS BETTER
68%
95%
0.1%
2%
14%
14%
2%
0.1%
IQ score
55
70
85
100
115
130
145
imgflip.com

Scaling Laws

Why do we need scaling laws?


Arithmetic (few-shot)
Two Digit Addition
Two Digit Subtraction
Three Digit Addition
Three Digit Subtraction
Four Digit Addition
Four Digit Subtraction
Five Digit Addition
Five Digit Subtraction
Two Digit Multiplication
Single Digit Three Ops
Accuracy
100
80
60
40
20
0
0.1B
0.4B
0.8B
1.3B
2.6B
6.7B
13B
175B
Parameters in LM (Billions)

KM Scaling Law

KM Scaling Law

$$L(N) = \left(\frac{N_c}{N}\right)^{\alpha N}, \alpha N \sim 0.076, N_c \sim 8.8 \times 10^{23}$$

$$L(D) = \left(\frac{D_c}{D}\right)^{\alpha D}, \alpha D \sim 0.095, D_c \sim 5.4 \times 10^{13}$$

$$L(C) = \left(\frac{C_c}{C}\right)^{\alpha C}, \alpha C \sim 0.050, C_c \sim 3.1 \times 10^{8}$$

Chinchilla Scaling Law

$$L(N, D) = E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}}$$

$$N_{opt}(C) = G\left(\frac{C}{6}\right)^{a}, D_{opt}(C) = G^{-1}\left(\frac{C}{6}\right)^{b}$$

So, which one would you use?

Scaling Laws

So, back to the question, why do we need scaling laws?

1. Predictable Scaling

Scaling Laws

So, back to the question, why do we need scaling laws?

Scaling Laws

So, back to the question, why do we need scaling laws?

1. Predictable Scaling
2. Task-level predictability

Scaling Laws

So, back to the question, why do we need scaling laws?

1. Predictable Scaling
2. Task-level predictability
3. Emergent Abilities

Scaling Laws

So, back to the question, why do we need scaling laws?

1. Predictable Scaling
2. Task-level predictability
3. Emergent Abilities
    a. In-Context Learning

Scaling Laws

So, back to the question, why do we need scaling laws?

1. Predictable Scaling
2. Task-level predictability
3. Emergent Abilities
    a. In-Context Learning
    b. Instruction Following

Scaling Laws

So, back to the question, why do we need scaling laws?

1. Predictable Scaling
2. Task-level predictability
3. Emergent Abilities
    a. In-Context Learning - @Nicholas
    b. Instruction Following
    c. Step-by-step reasoning

In-Context Learning

Please see the following text and annotation. You need to identify the first name, last name and age from the text. the format of data is provided to you

{
1: I am Eric Dawson, 31M, living in Miami,
2: I'm Fiona Gonzales, living in SF. I am 24 years old.
}

Your response should be like the following:

{
1: I am Eric [FN] Dawson [LN], 31 [age] M, living in Miami,
2: I'm Fiona [FN] Gonzales [LN], living in SF. I am 24 [age] years old.
}

Instruction following

without additional instruction

Model Input

Instruction: Write the words of the following sentence in reverse order.

Input: The dog crossed the road

Model Output

Output: road the crossed dog the

with additional instruction

Model Input

Instruction: Reformat the following comma-separated list of names in <last name>, <first name> format separated with semicolons.

Input: Cameron Wolfe, John Doe

Model Output

Output: Wolfe, Cameron; Doe, John

Chain of Thought

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Section Summary

We saw

1. History of language models
2. Scaling Laws (OpenAI and Chinchilla)
    a. OpenAI - Best possible model
    b. Chinchilla - Best possible model with provided budget
3. Why we need scaling laws
4. Emergent abilities

Questions

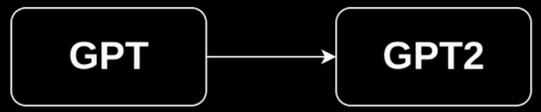
Table of Contents - Evolution of GPT

Generative Pre-training

Technical Evolution

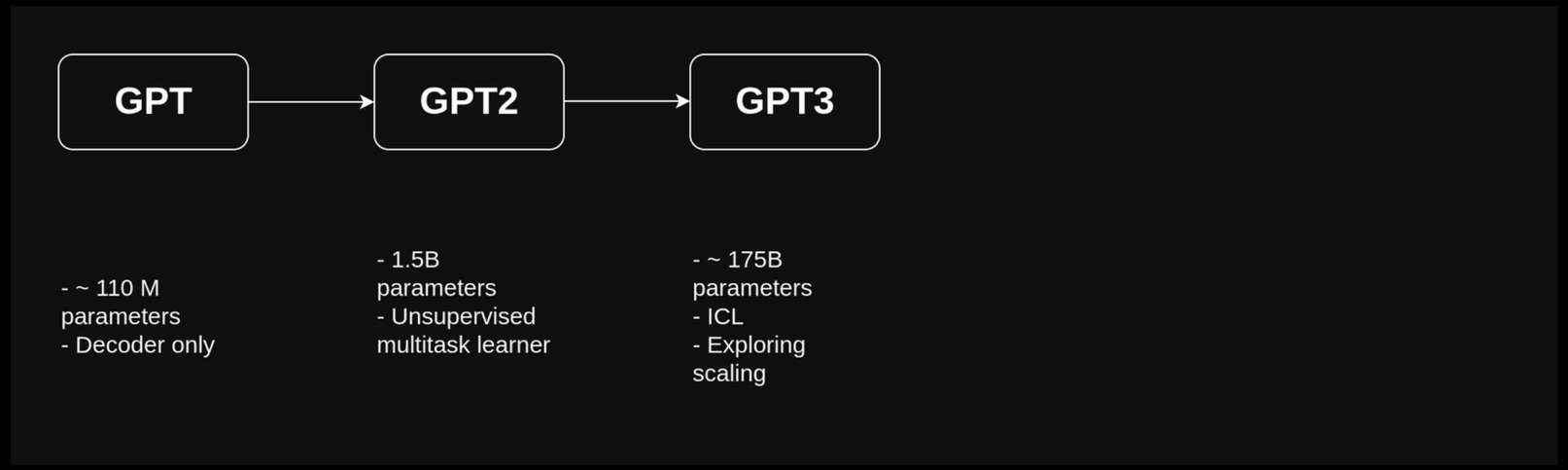
GPT

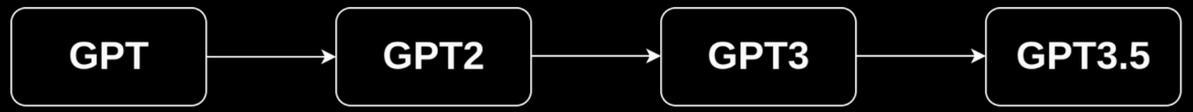
- ~ 110 M parameters
- Decoder only



- ~ 110 M parameters
- Decoder only

- 1.5B parameters
- Unsupervised multitask learner


GPT
GPT2
GPT3
- ~ 110 M parameters
- Decoder only
- 1.5B parameters
- Unsupervised multitask learner
- ~ 175B parameters
- ICL
- Exploring scaling



- ~ 110 M parameters
- Decoder only

- 1.5B parameters
- Unsupervised multitask learner

- ~ 175B parameters
- ICL
- Exploring scaling

- Code + instruction finetuning + RLHF + Chat

- ~ 110 M parameters
- Decoder only

- 1.5B parameters
- Unsupervised multitask learner

- ~ 175B parameters
- ICL
- Exploring scaling

- Code + instruction finetuning + RLHF + Chat

- Multimodal

Resources

Pretraining Corpora

Resources

Pretraining Corpora

Webpages		
Books		
Conversation		
code		

Resources

Pretraining Corpora

Webpages	General text	CommonCrawl
Books		
Conversation		
code		

Resources

Pretraining Corpora

Webpages	General text	CommonCrawl
Books	Long context understanding	BookCorpus
Conversation		
code		

Resources

Pretraining Corpora

Webpages	General text	CommonCrawl
Books	Long context understanding	BookCorpus
Conversation	reply response	RedditLinks
code		

Resources

Pretraining Corpora

Webpages	General text	CommonCrawl
Books	Long context understanding	BookCorpus
Conversation	reply response	RedditLinks
code	coding & reasoning abilities	GitHub

Resources

Fine-tuning Datasets

Resources

Fine-tuning Datasets

Instruction Finetuning

1. **NLP Task Datasets**

Resources

Fine-tuning Datasets

Instruction Finetuning

1. **NLP Task Datasets**
2. **Daily Chat**

Resources

Fine-tuning Datasets

Instruction Finetuning

1. **NLP Task Datasets**
2. **Daily Chat**
3. **Synthetic Datasets - @Ansh**

Resources

Libraries

Transformers	
DeepSpeed	
Triton	
vLLM	

Resources

Libraries

Transformers	model architecture & weights
DeepSpeed	
Triton	
vLLM	

Resources

Libraries

Transformers	model architecture & weights
DeepSpeed	efficient distributed training
Triton	
vLLM	

Resources

Libraries

Transformers	model architecture & weights
DeepSpeed	efficient distributed training
Triton	writing autotuned cuda kernels
vLLM	

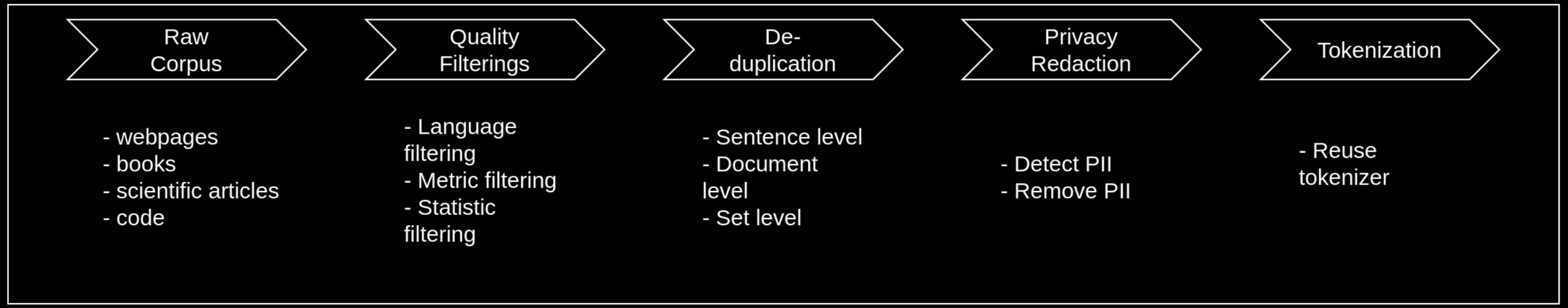
Resources

Libraries

Transformers	model architecture & weights
DeepSpeed	efficient distributed training
Triton	writing autotuned cuda kernels
vLLM	inference

Pre-Training

Data Preparation


Raw Corpus
- webpages
- books
- scientific articles
- code
Quality Filterings
- Language filtering
- Metric filtering
- Statistic filtering
De-duplication
- Sentence level
- Document level
- Set level
Privacy Redaction
- Detect PII
- Remove PII
Tokenization
- Reuse tokenizer

Section Summary

We saw

1. Evolution of GPT
2. Pretraining corpora
3. Finetuning Corpora
4. Data Preparation

Questions

Table of Contents - Scaling and Need for scaling

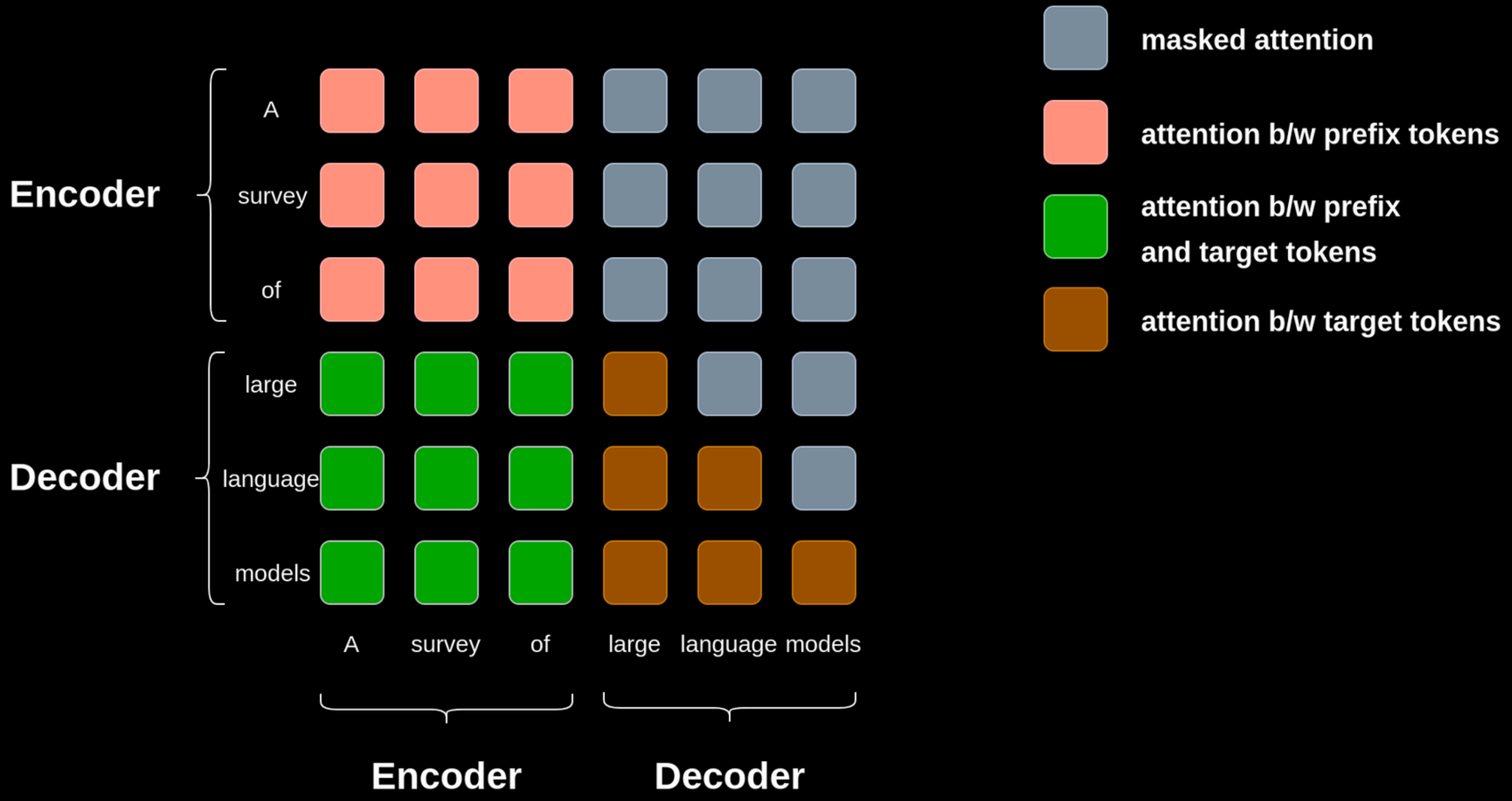
Pre-Training

Choosing the Architecture

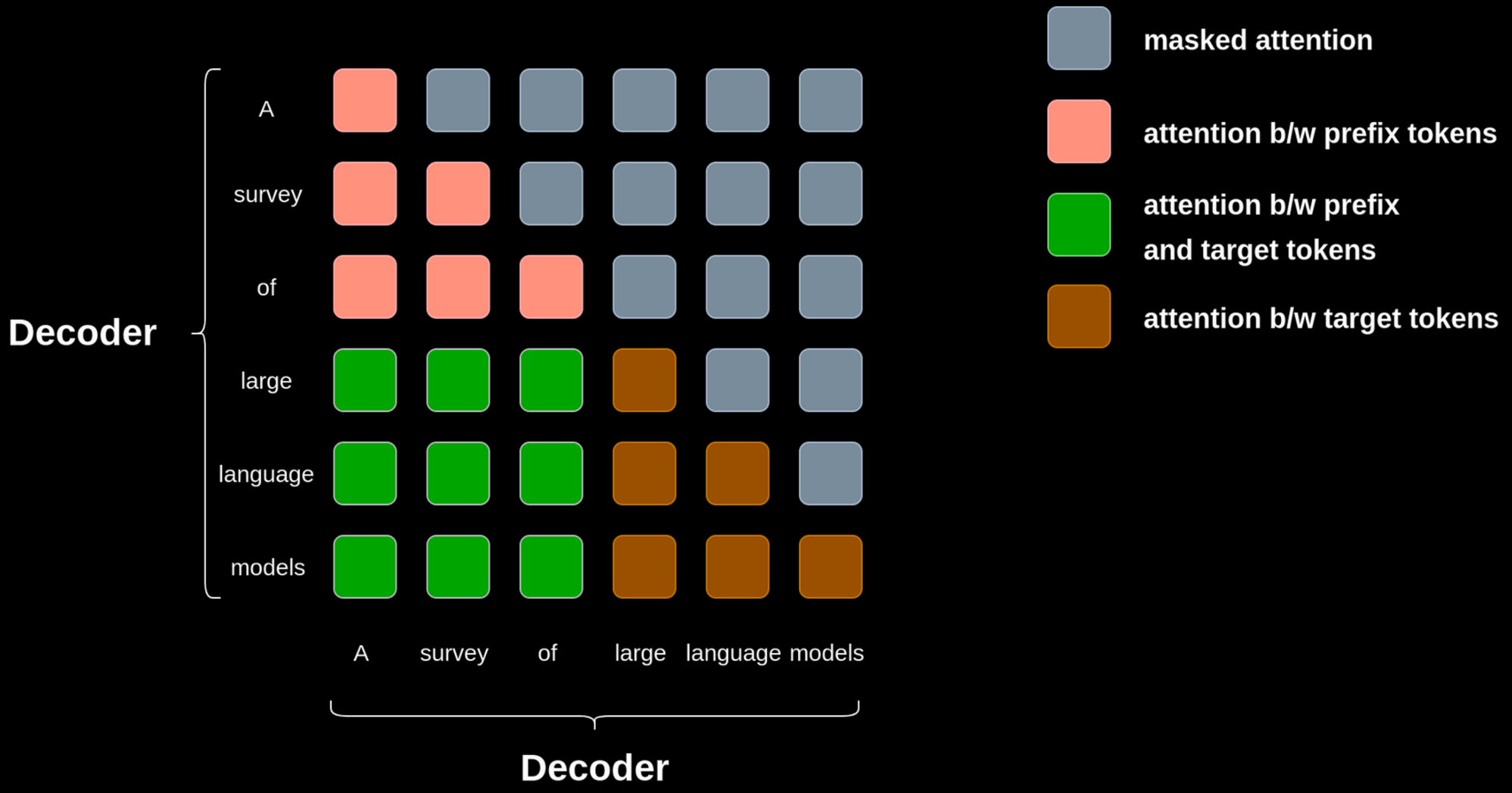
Pre-Training

Choosing the Architecture

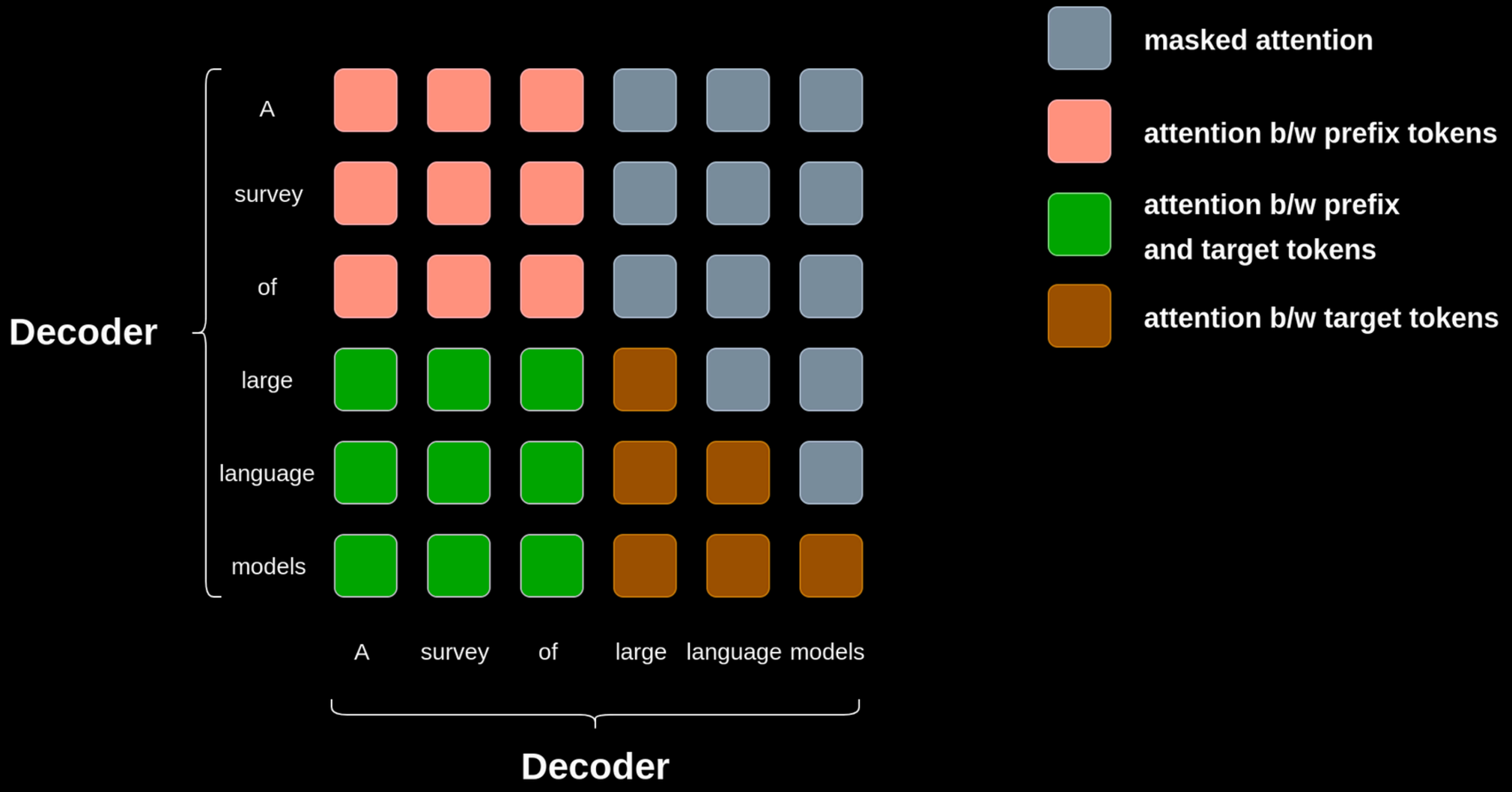
Encoder-Decoder Model



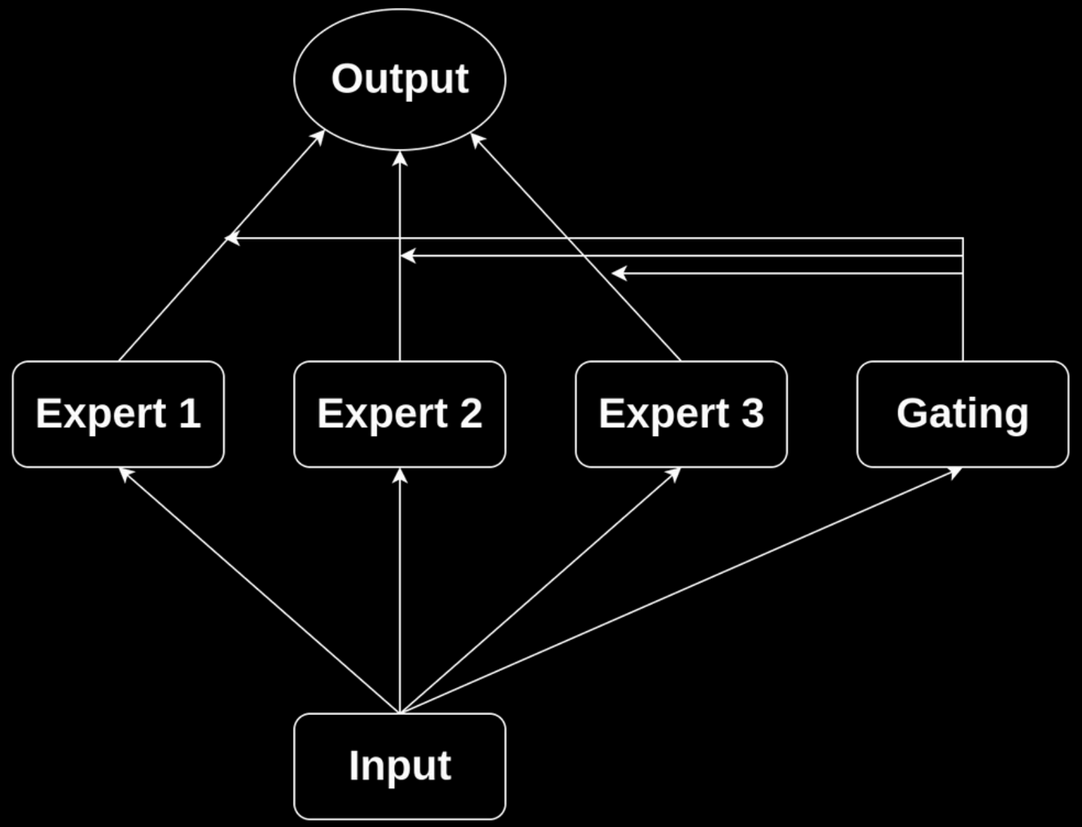
Causal Decoder Model



Non-causal Decoder Model



Mixture of Experts



Emergent Architectures

RWKV

XLSTM

H3

State Space Models

S4

And many others...!

Pre-Training

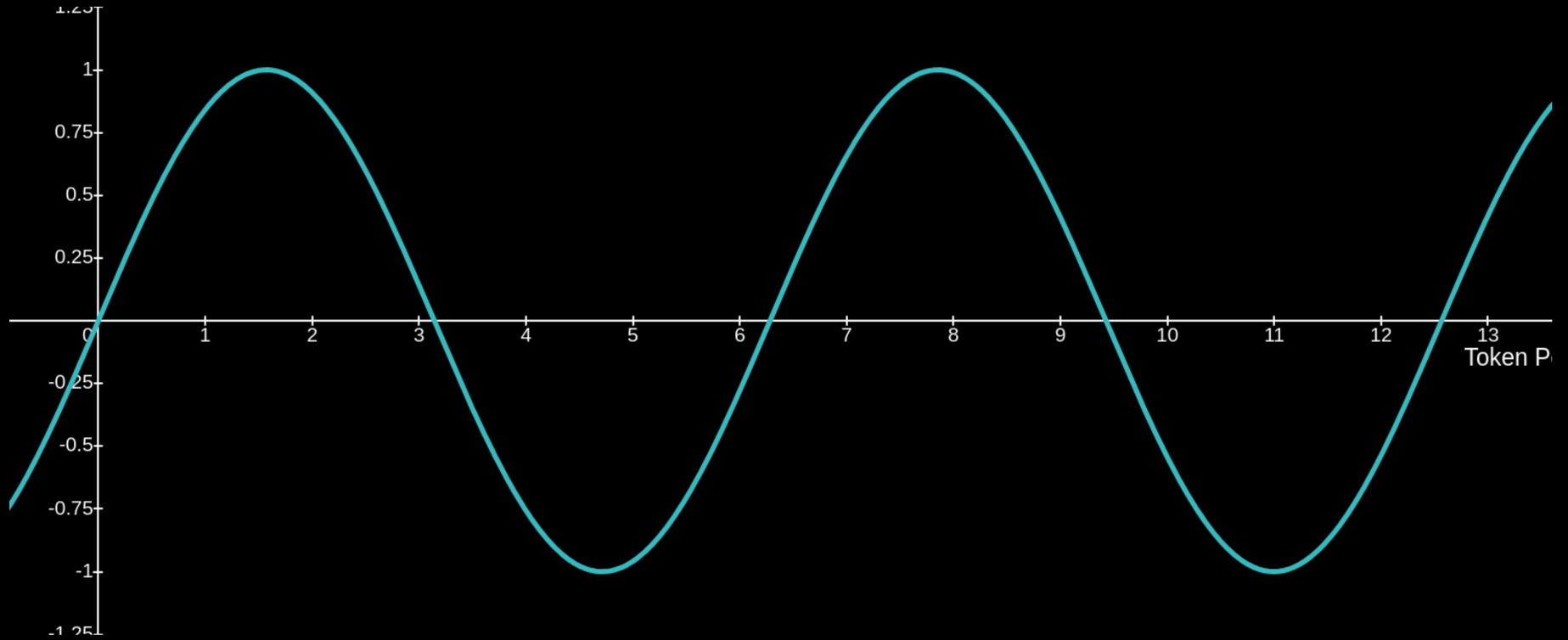
Position Embeddings

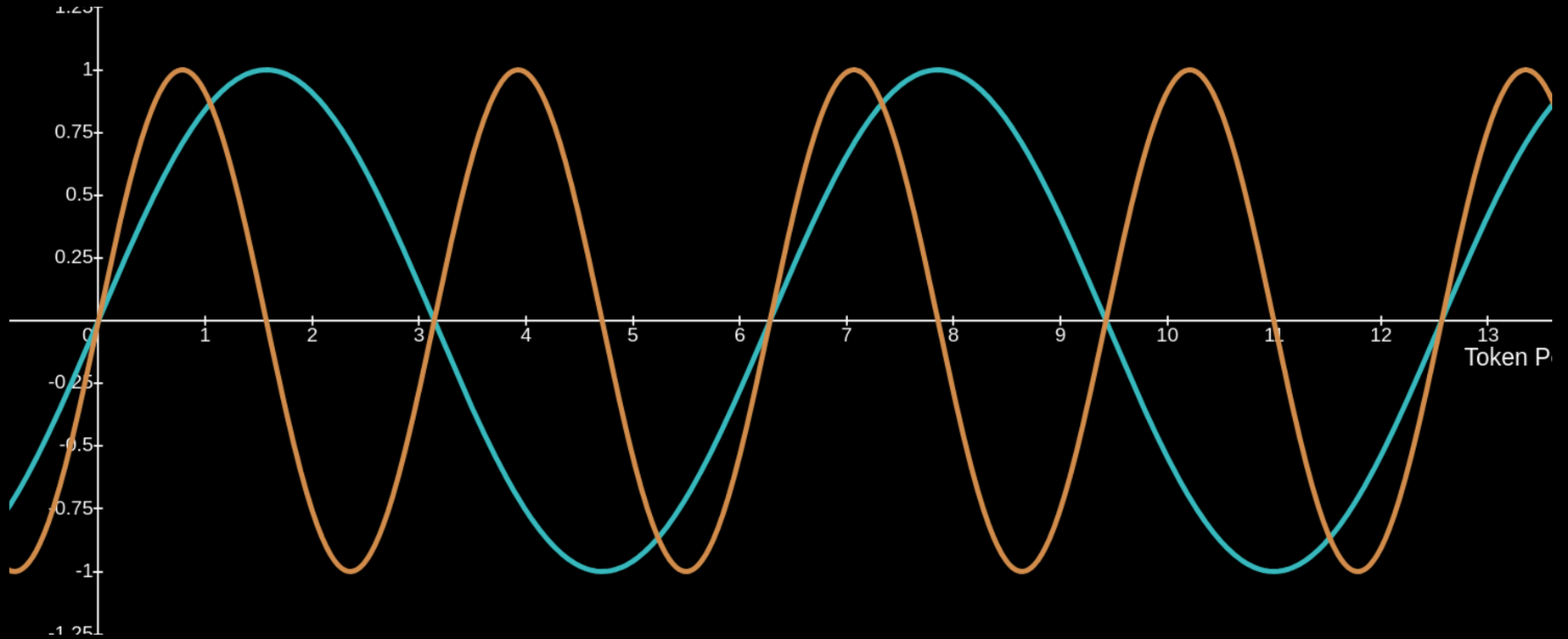
Absolute Position Embeddings

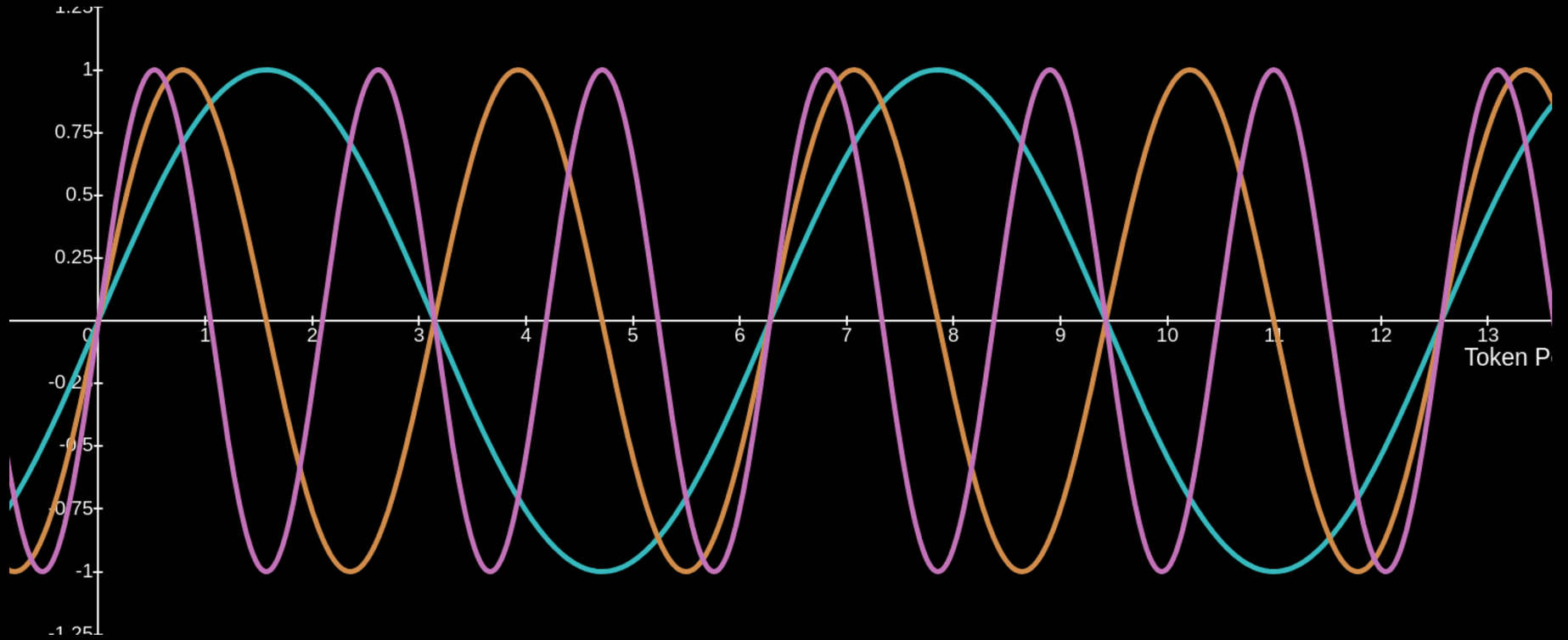
Sinusoidal

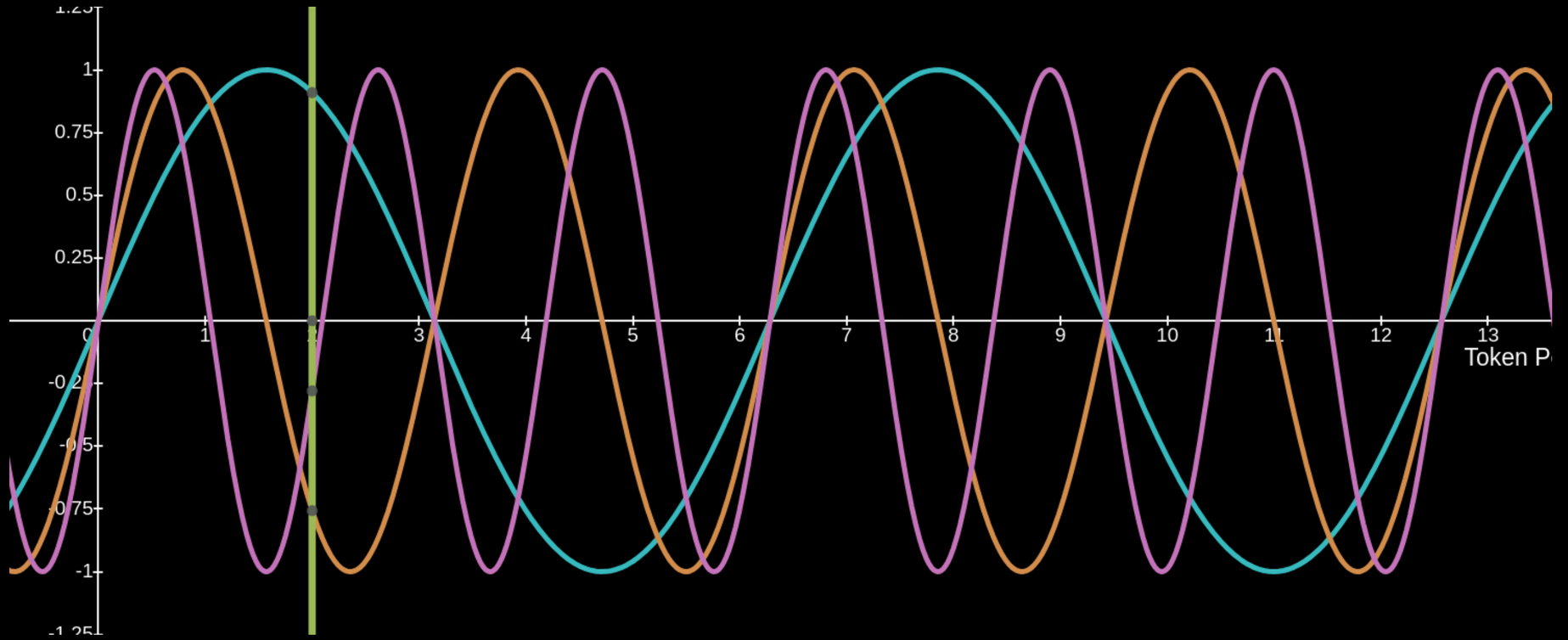
Learned

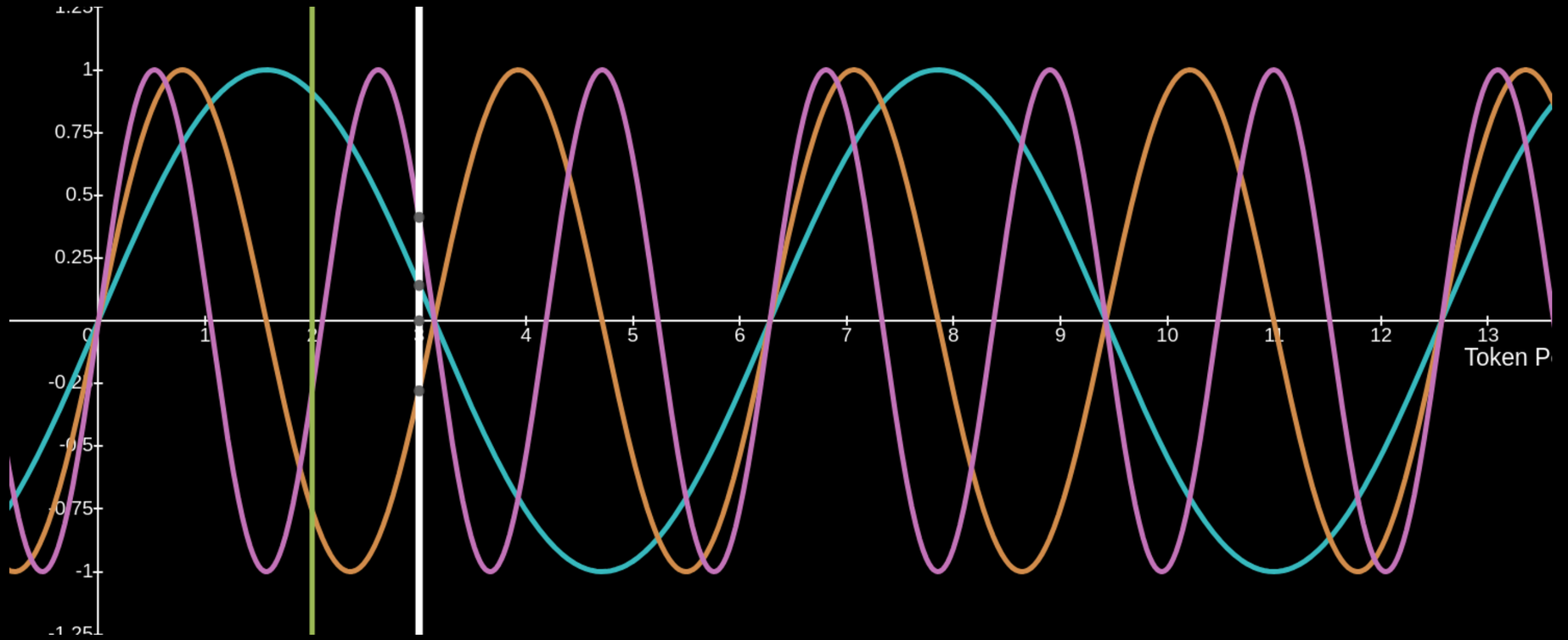
An Intuition for Sinusoidal Embeddings

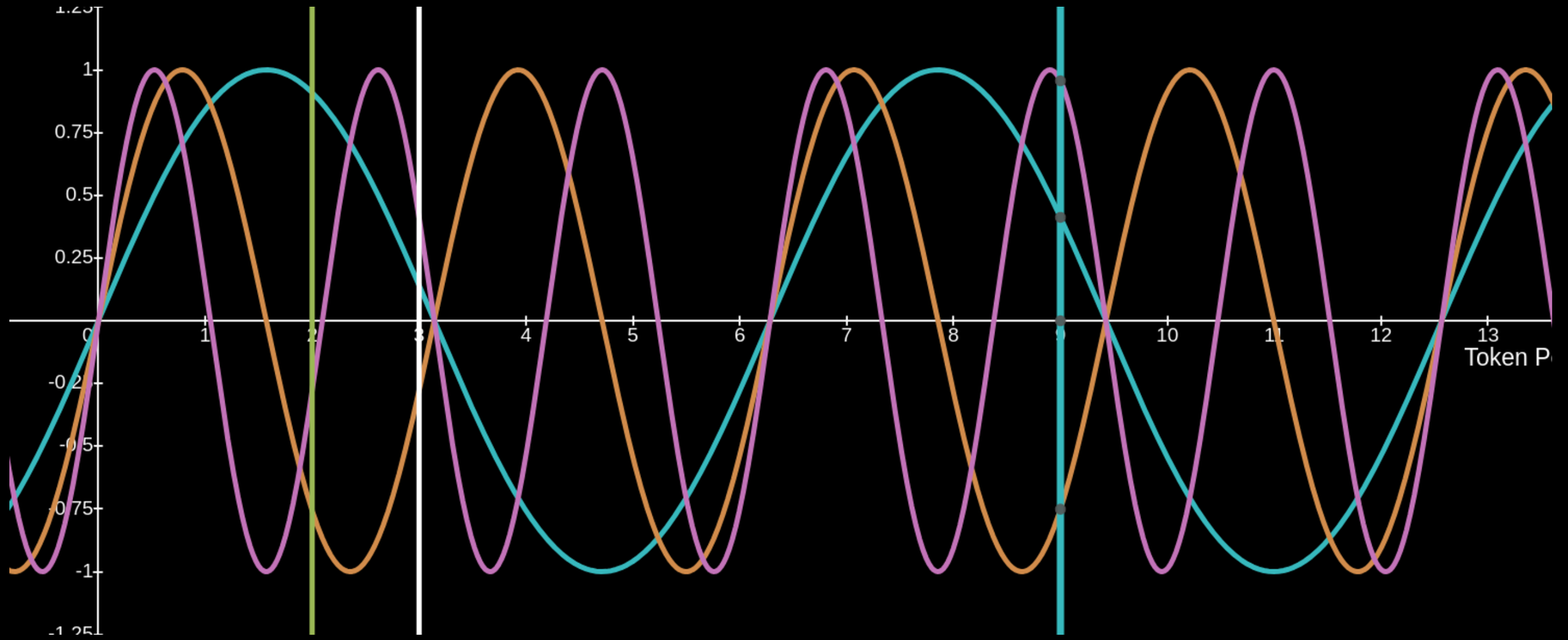

1
0.75
0.5
0.25
0
1
2
3
4
5
6
7
8
9
10
11
12
13
-0.25
-0.5
-0.75
-1
Token P

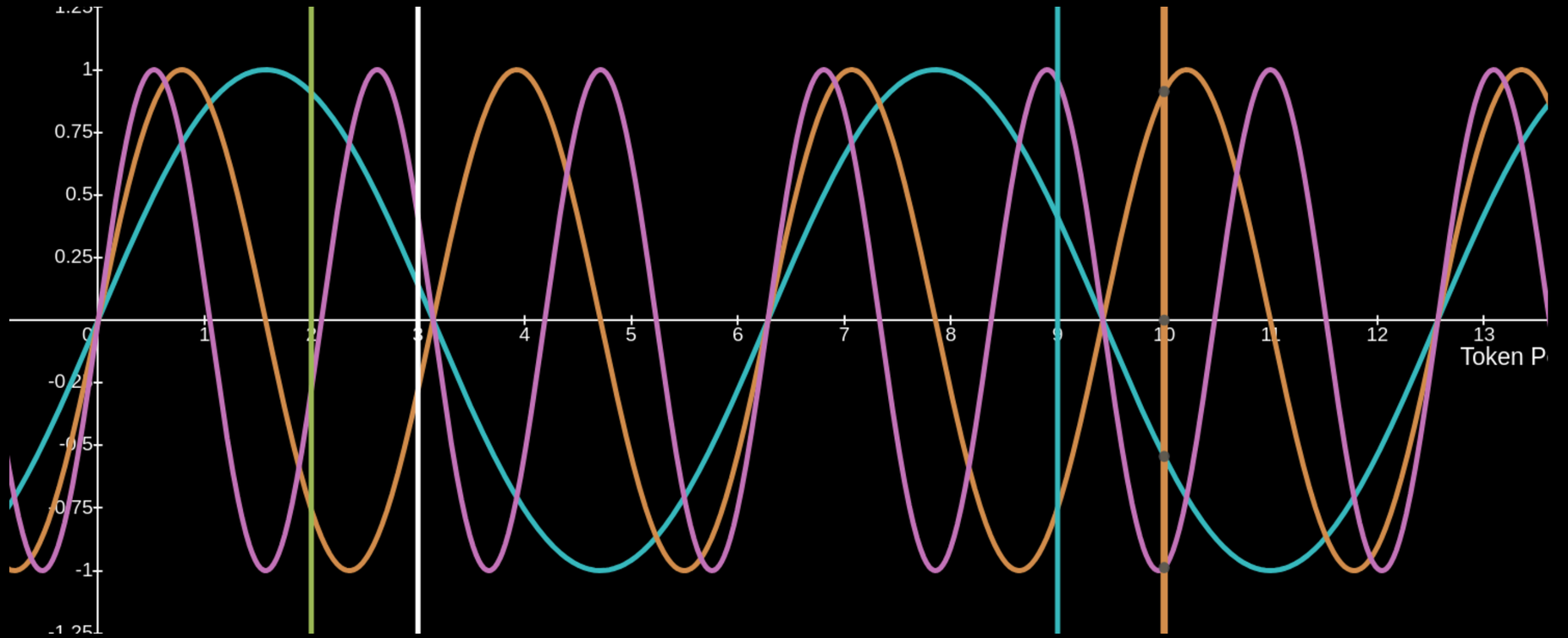

1
0.75
0.5
0.25
0
1
2
3
4
5
6
7
8
9
10
11
12
13
-0.5
-0.75
-1

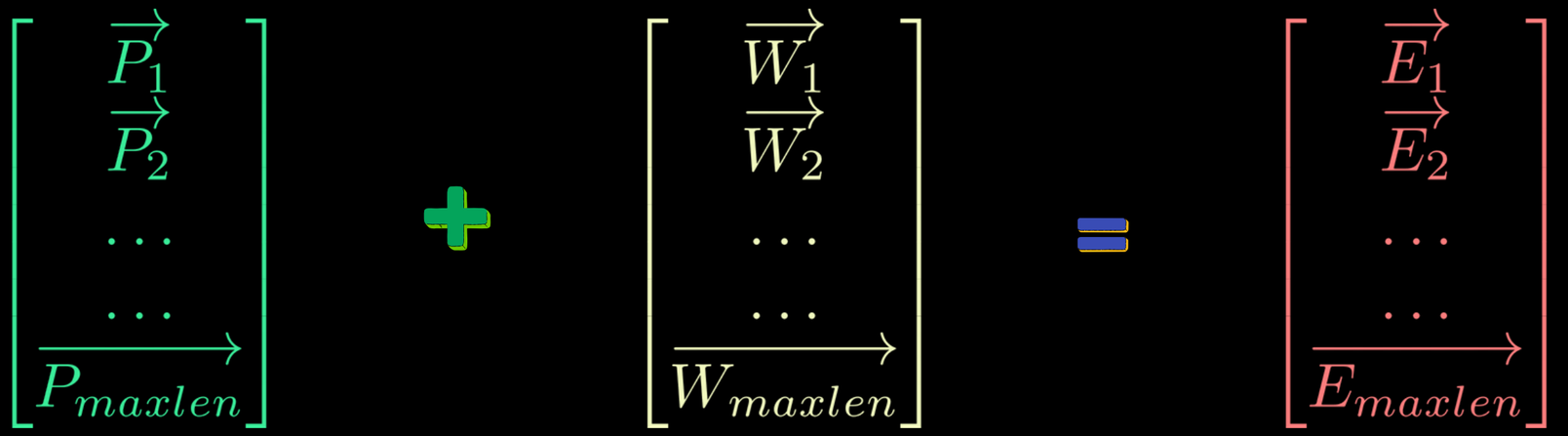

1
0.75
0.5
0.25
0
1
2
3
4
5
6
7
8
9
10
11
12
13
Token P
-0.2
-0.5
-0.75
-1


1
0.75
0.5
0.25
0
1
2
3
4
5
6
7
8
9
10
11
12
13
Token P

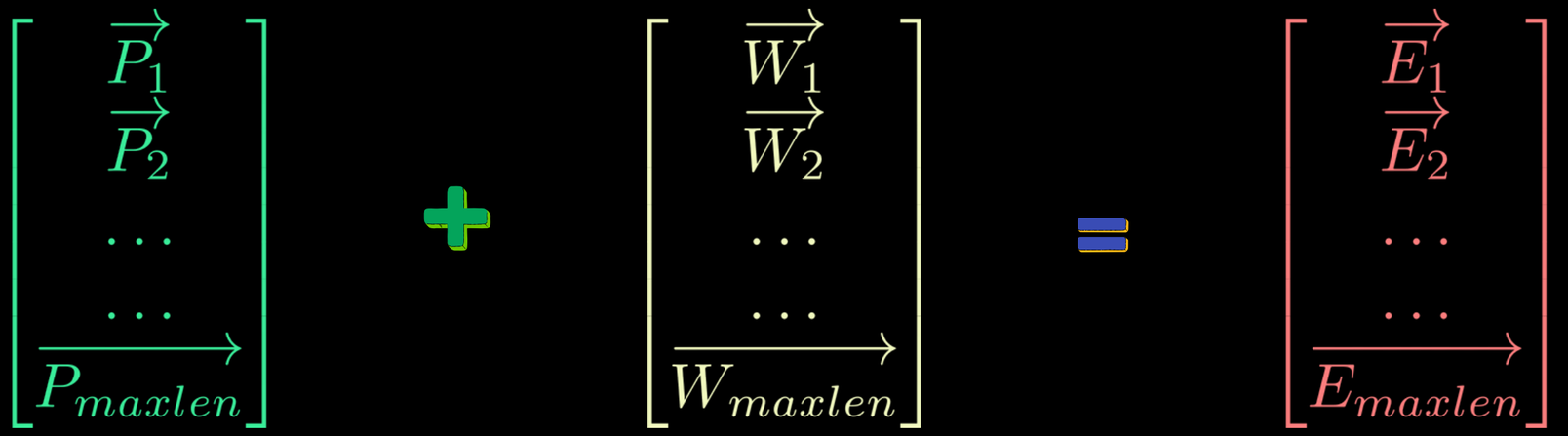

1
0.75
0.5
0.25
0
1
2
3
4
5
6
7
8
9
10
11
12
13
Token P
-0.2
-0.5
-0.75
-1


1
0.75
0.5
0.25
0
1
2
3
4
5
6
7
8
9
10
11
12
13
Token P
-0.2
-0.5
-0.75
-1


1
0.75
0.5
0.25
0
1
2
3
4
5
6
7
8
9
10
11
12
13
Token P


$\overrightarrow{P_1}$
$\overrightarrow{P_2}$
$\overrightarrow{P_{maxlen}}$
$\overrightarrow{W_1}$
$\overrightarrow{W_2}$
$\overrightarrow{W_{maxlen}}$
$\overrightarrow{E_1}$
$\overrightarrow{E_2}$
$\overrightarrow{E_{maxlen}}$

An Intuition for Learned Embeddings

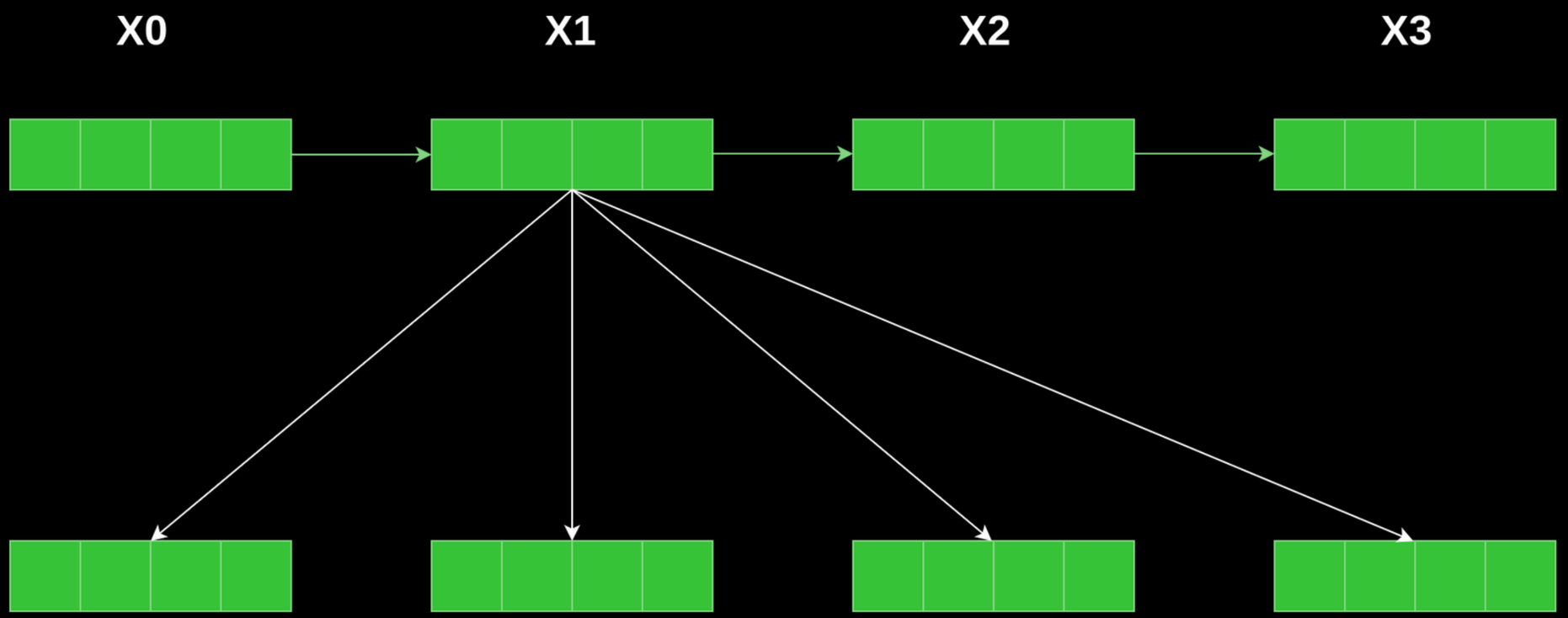


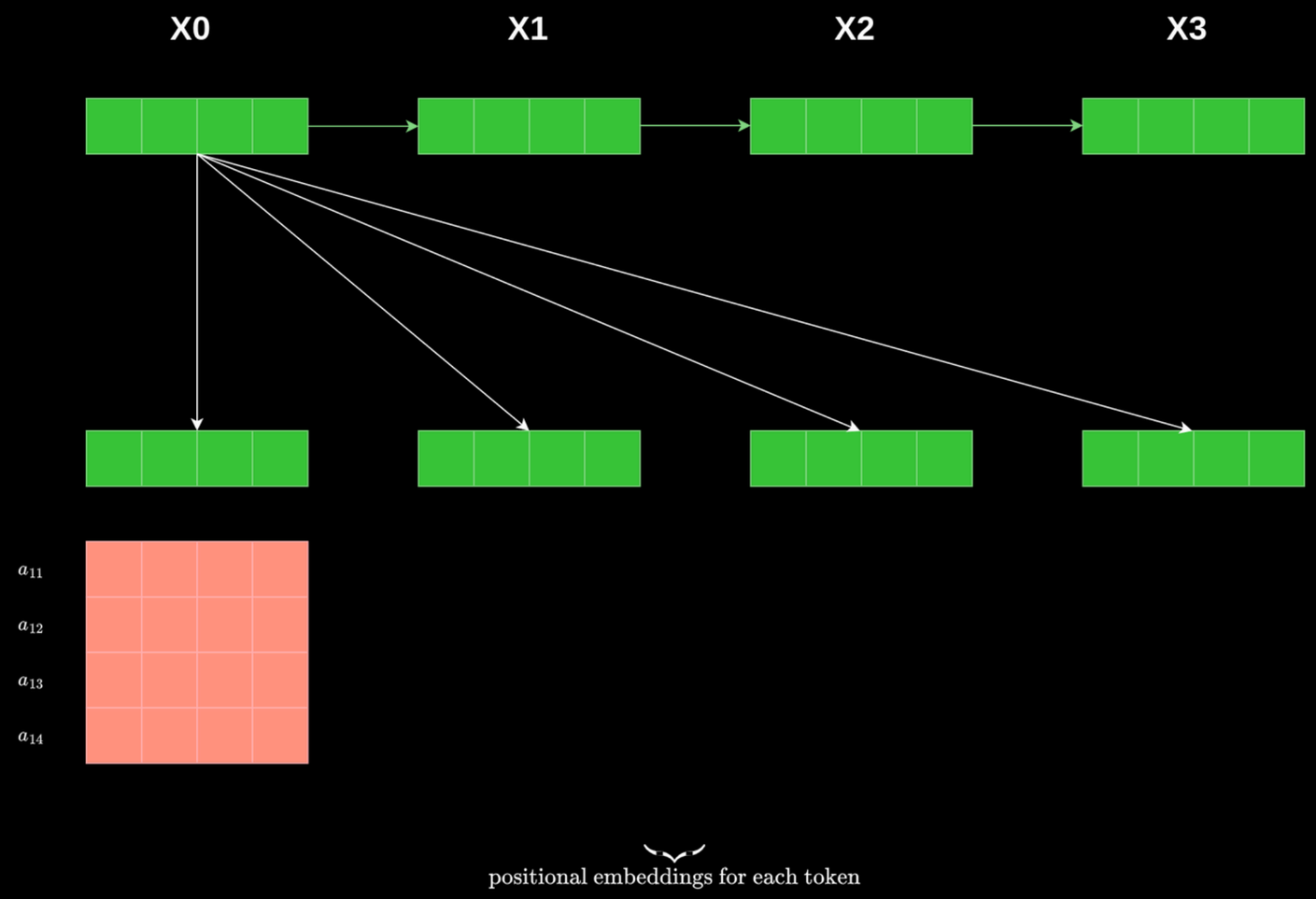
But these are learned along
with other params of the model

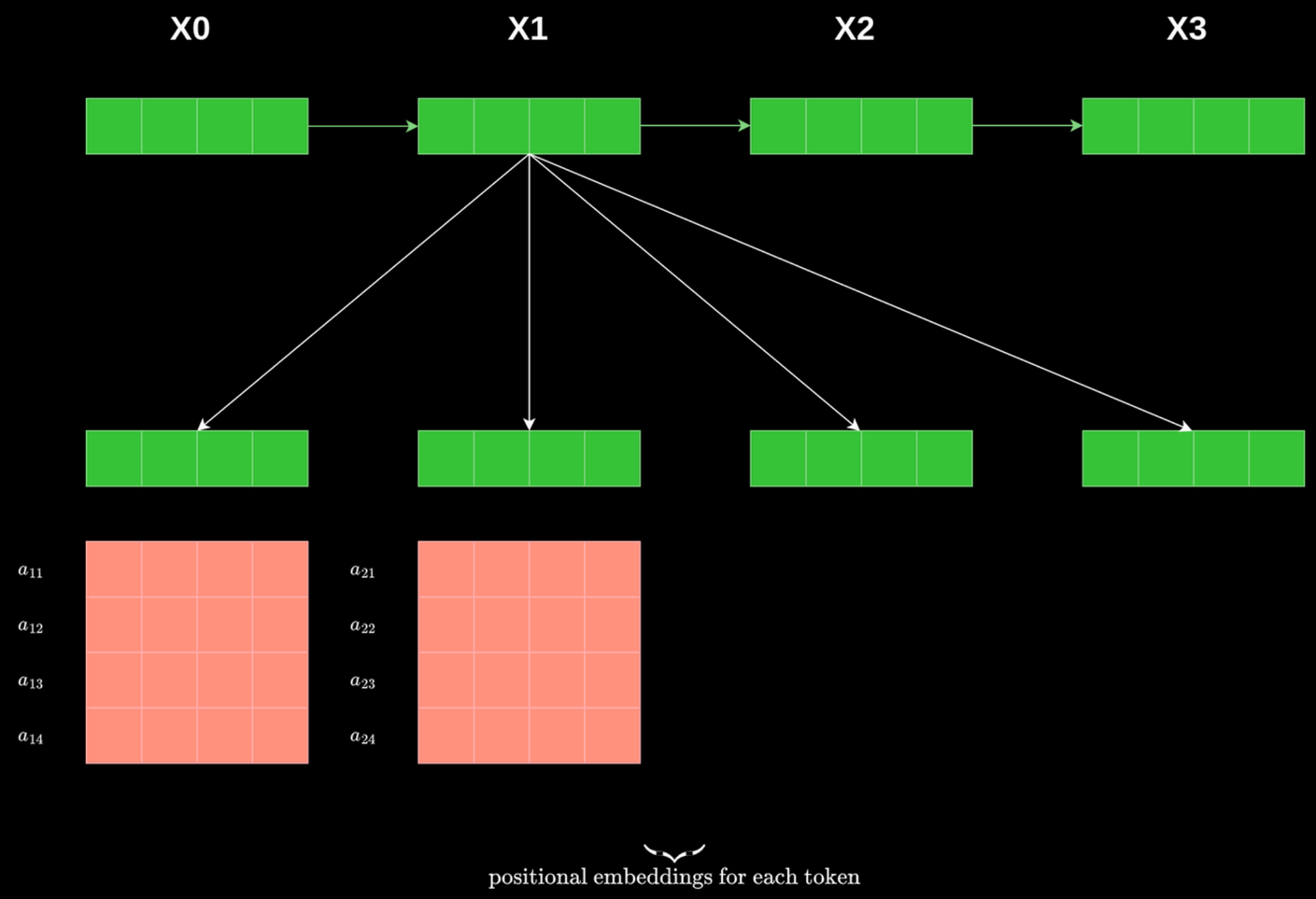
Pre-Training

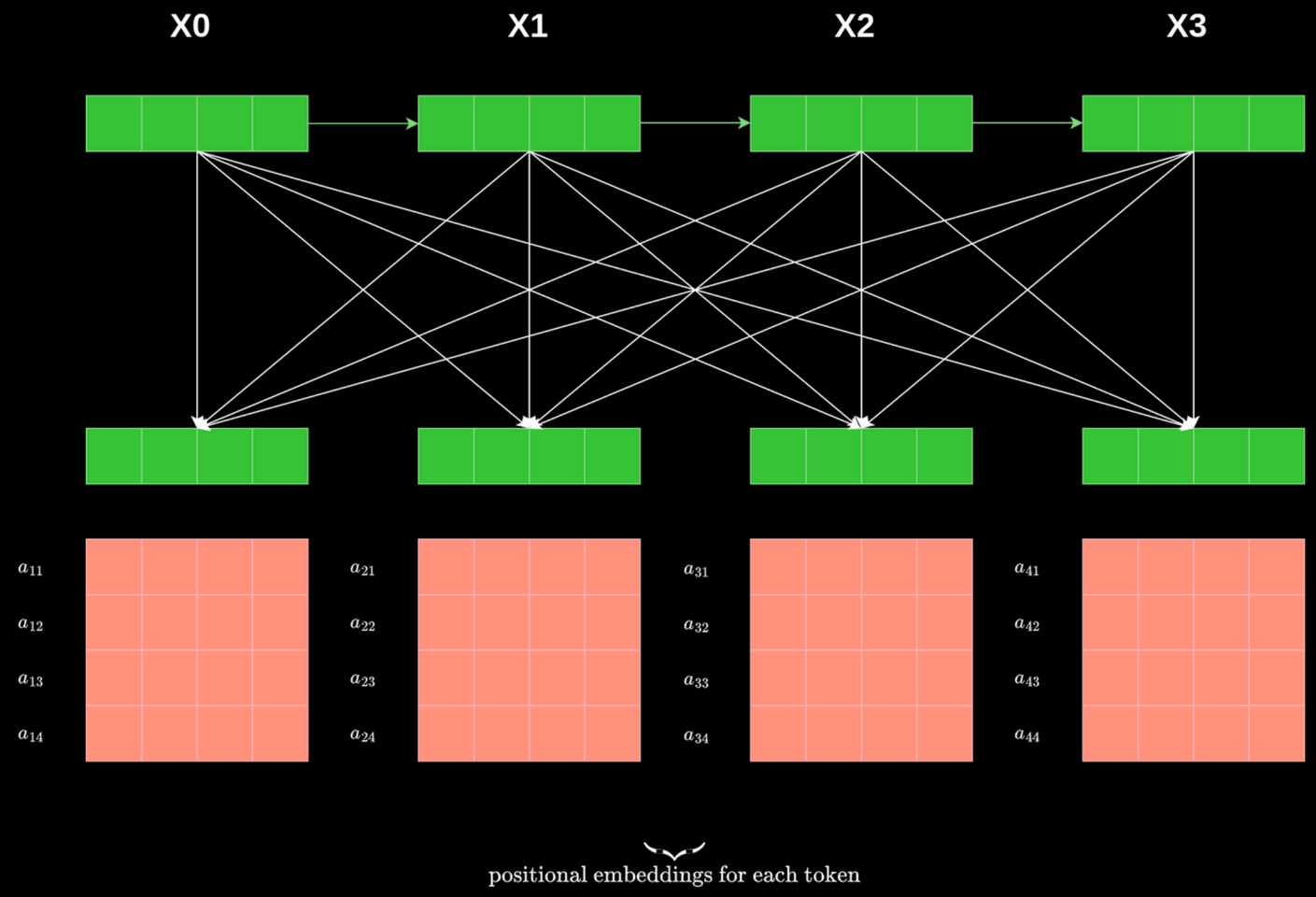
Position Embeddings

Relative Position Embeddings

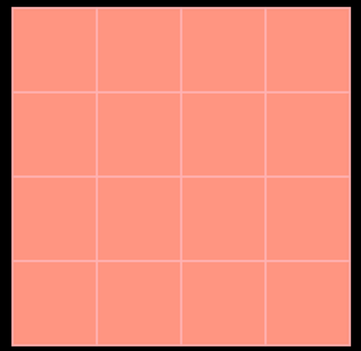

X0
X1
X2
X3


X0
X1
X2
X3
a_{11}
a_{12}
a_{13}
a_{14}
positional embeddings for each token


X0
X1
X2
X3
a_{11}
a_{12}
a_{13}
a_{14}
a_{21}
a_{22}
a_{23}
a_{24}
positional embeddings for each token


X0
X1
X2
X3
a_{11}
a_{12}
a_{13}
a_{14}
a_{21}
a_{22}
a_{23}
a_{24}
a_{31}
a_{32}
a_{33}
a_{34}
a_{41}
a_{42}
a_{43}
a_{44}
positional embeddings for each token

$$e_{ij} = \frac{\overbrace{x_i W^Q}^{\text{Query Projection}} \quad \left(\quad \overbrace{x_j W^K}^{\text{Key Projection}} \quad \right)^\top}{\sqrt{d_z}}$$



$$e_{ij} = \frac{\overbrace{x_i W^Q}^{\text{Query Projection}} \left(\overbrace{x_j W^K}^{\text{Key Projection}} + \overbrace{a_{ij}^K}^{\text{Key position vector}} \right)^\top}{\sqrt{d_z}}$$

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})}$$

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})}$$

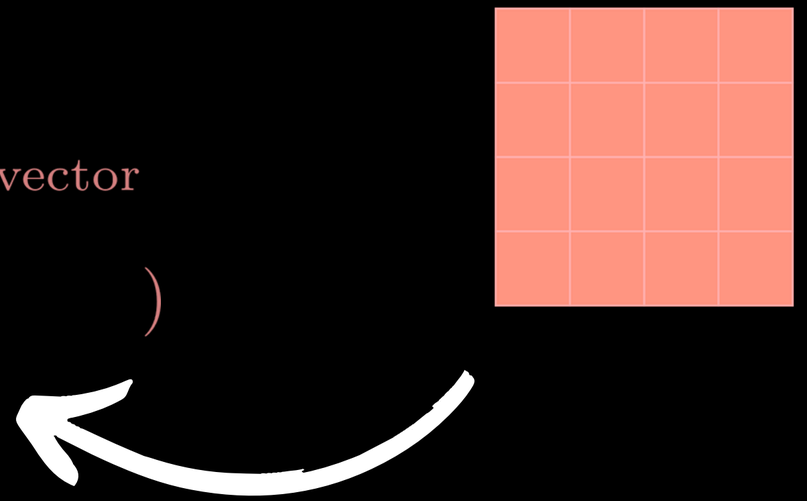
What is this operation called?

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})}$$

What is this operation called?

SoftMax!

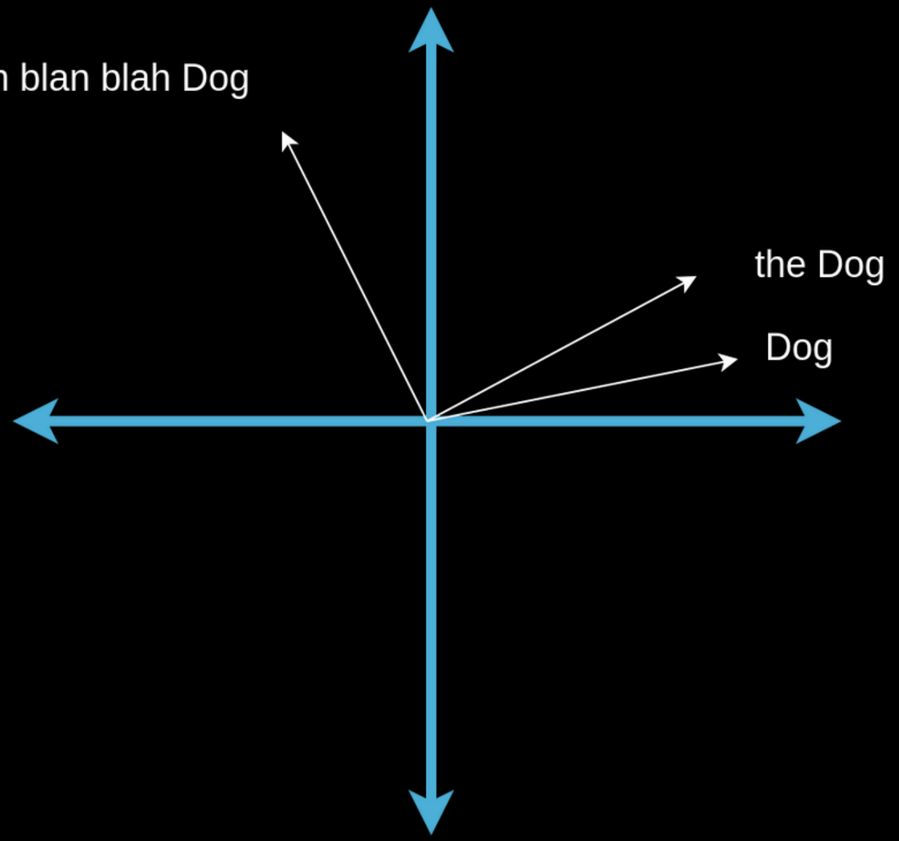
$$z_i = \sum_{j=1}^{n} \alpha_{ij} (\overbrace{x_j W^V}^{\text{Value Projection}} + \overbrace{a_{ij}^V}^{\text{Value position vector}})$$

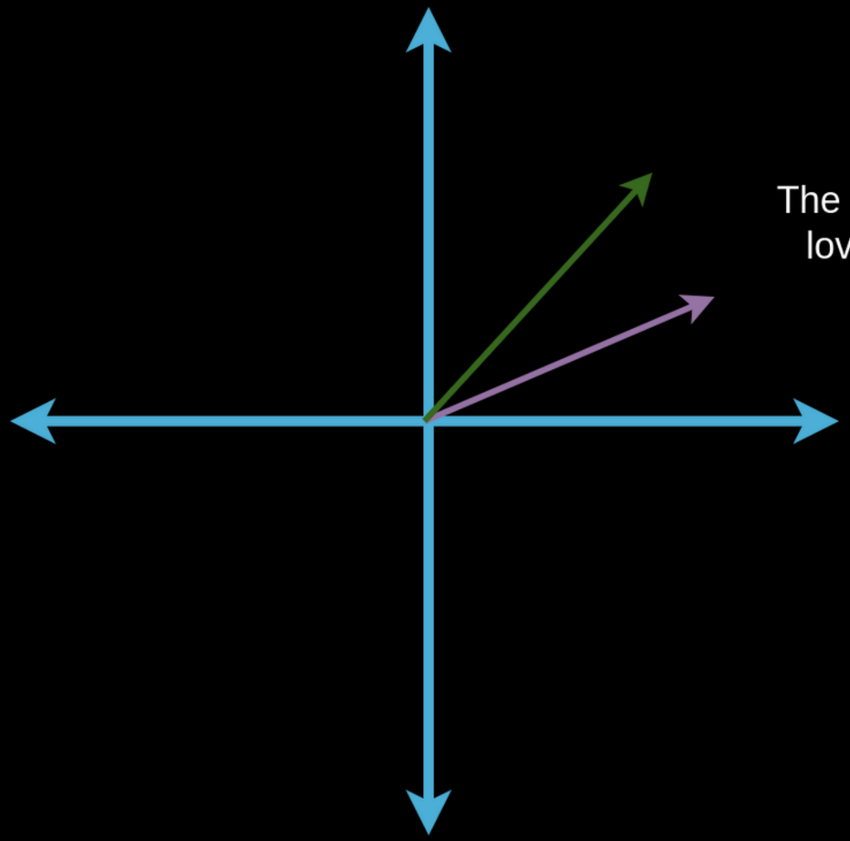


Pre-Training

Position Embeddings

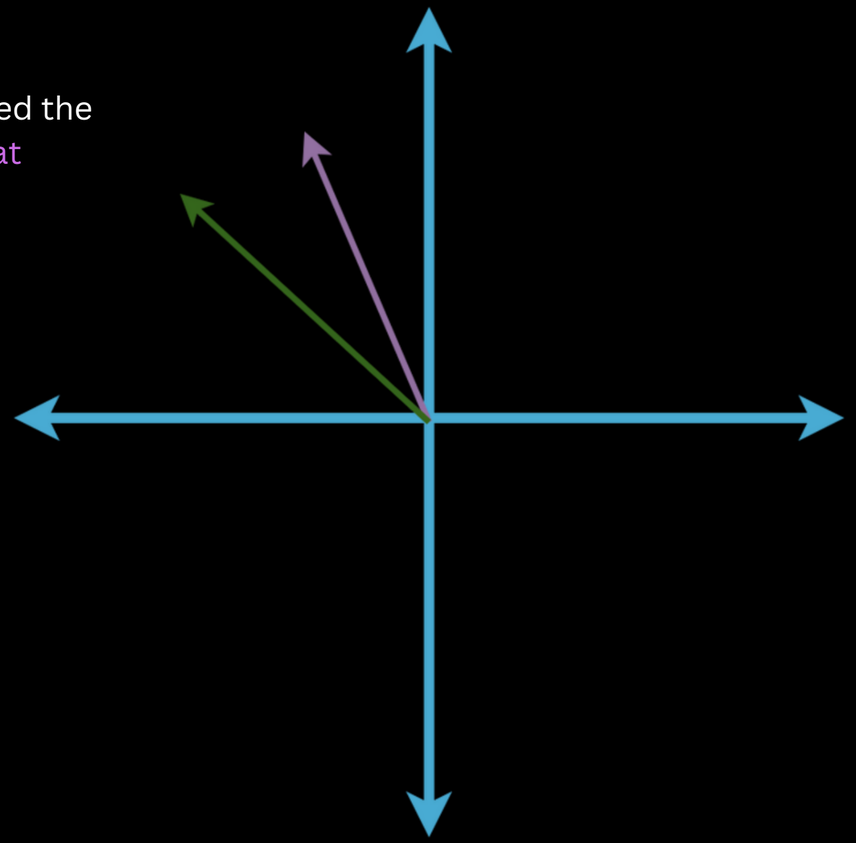
Rotary Position Embeddings (RoPE)


blah blan blah Dog
the Dog
Dog



The **dog** and cat **were** loved by the owner

The owner loved the
dog and cat



$$f_{q,k}(x_m, m) = \overbrace{\begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix}}^{\text{Rotation Matrix}} \begin{pmatrix} W_{qk}^{(11)} & W_{qk}^{(12)} \\ W_{qk}^{(21)} & W_{qk}^{(22)} \end{pmatrix} \overbrace{\begin{pmatrix} x_m^{(1)} \\ x_m^{(2)} \end{pmatrix}}^{\text{Vector we want to rotate}}$$

So, what would you use?

Scaling beyond 3000 tokens

Scaling beyond 3000 tokens

Increase context length
slowly during training

Scaling beyond 3000 tokens

Increase context length slowly during training

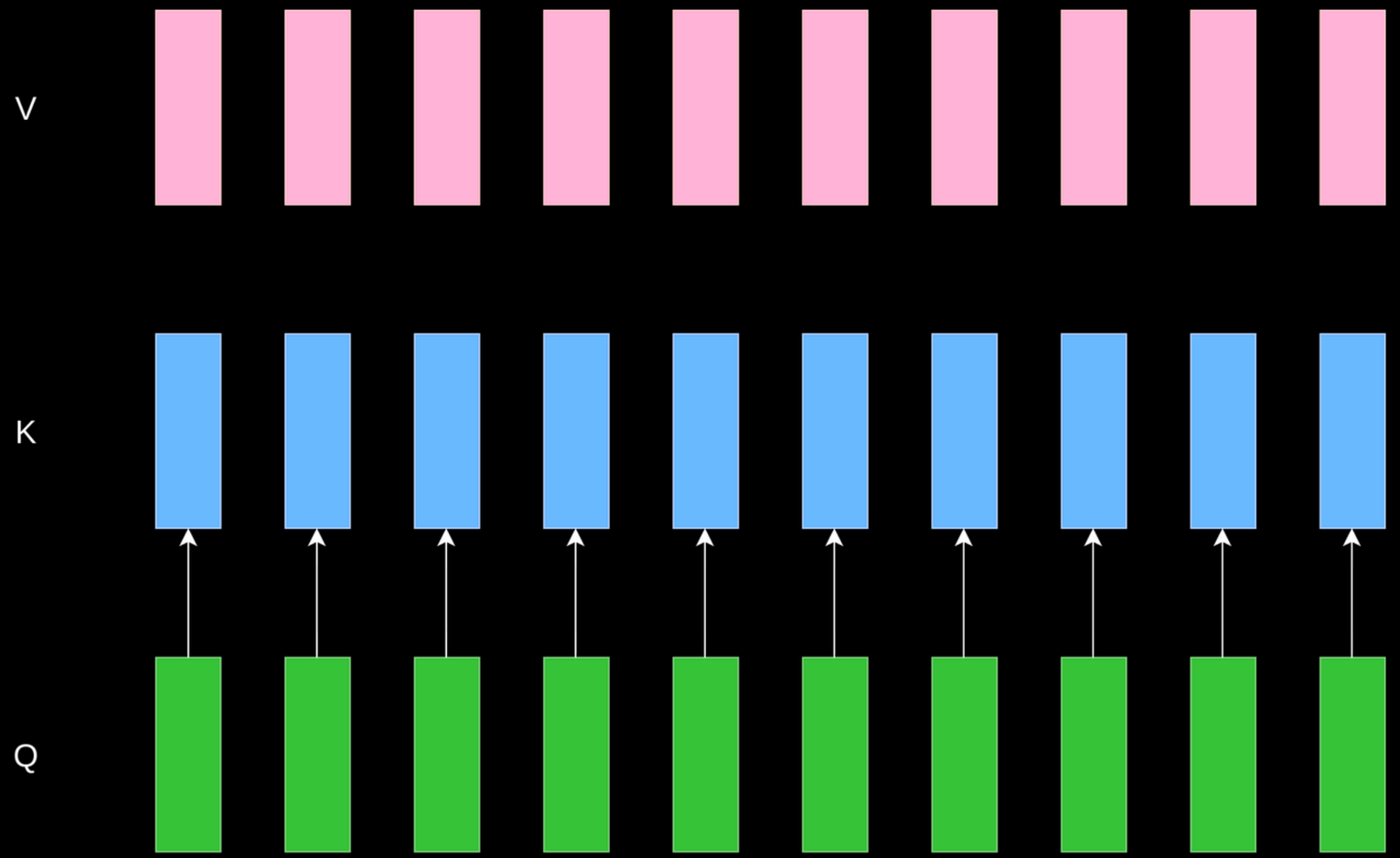
Position interpolation by multiplying with a factor

Pre-Training

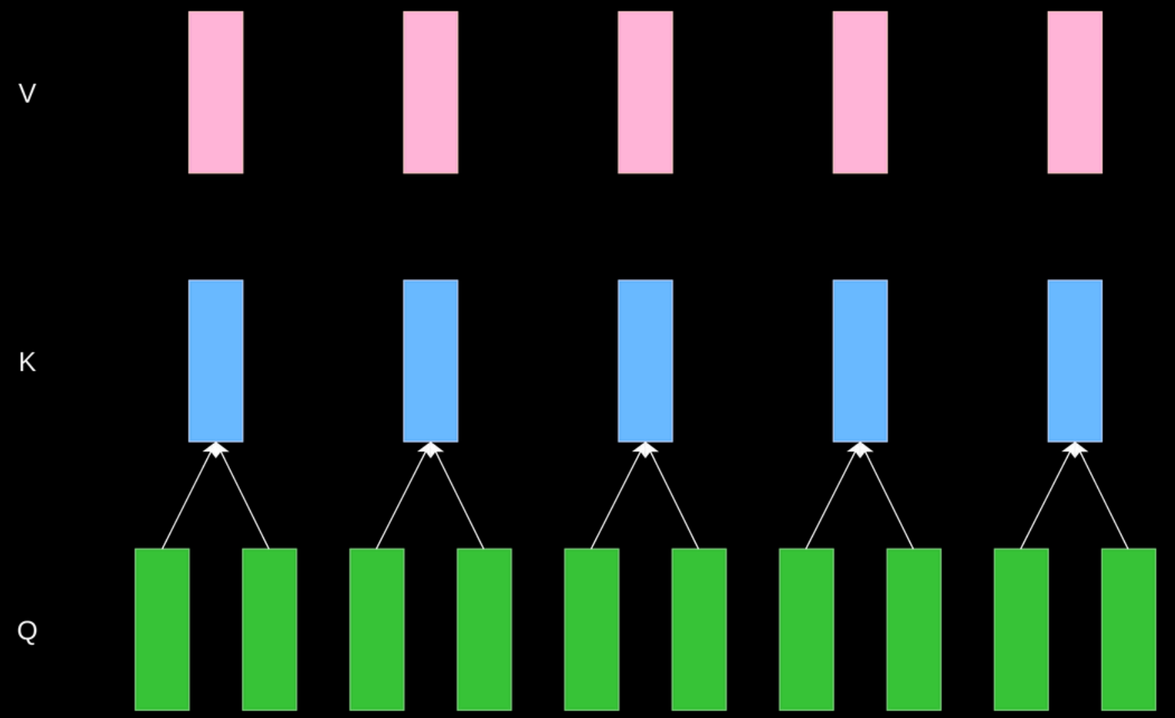
Attention Mechanism

Full Attention

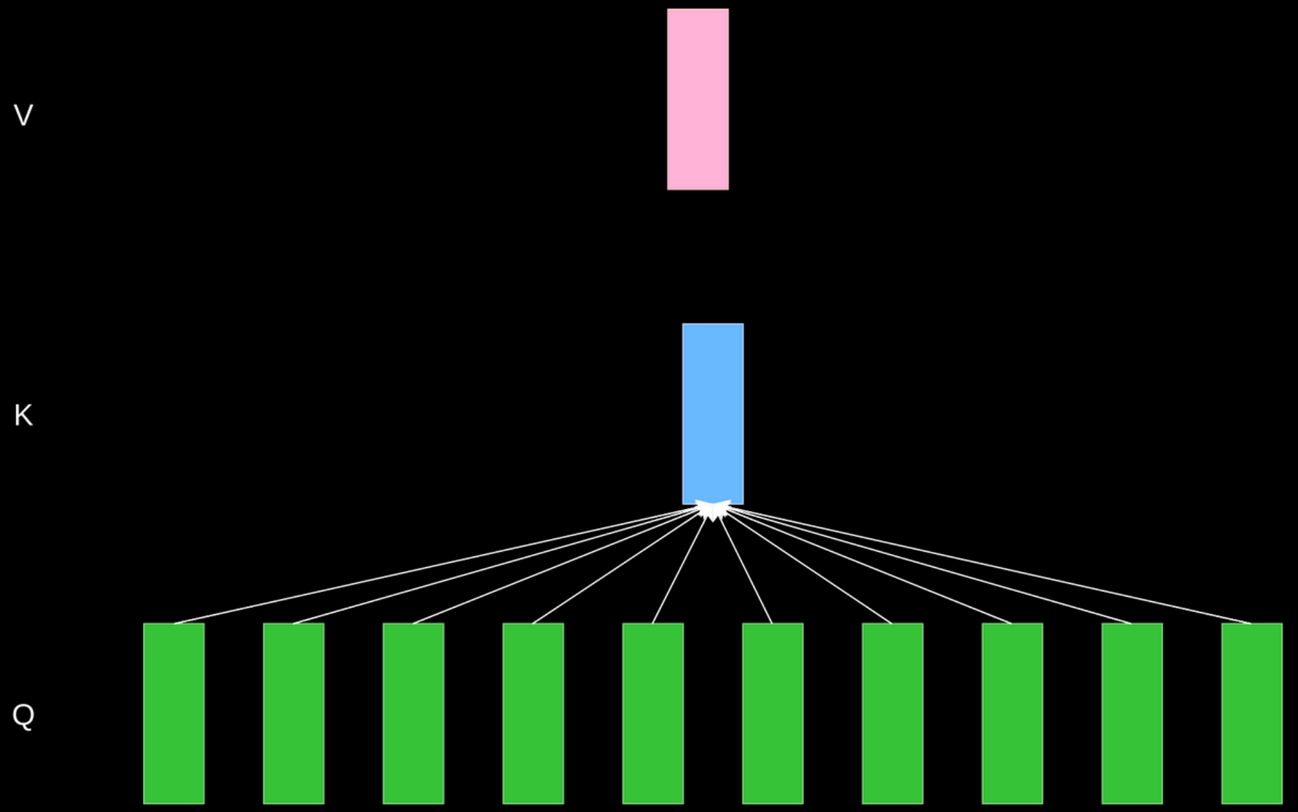
$$Attention(q, k, v) = softmax\left(\frac{\overbrace{Q}^{\text{Query Projection}} \quad \overbrace{K^\top}^{\text{Key Projection}}}{\underbrace{\sqrt{d_k}}_{\text{Scaling factor}}}\right) \overbrace{V}^{\text{Value Projection}}$$


V
K
Q

Grouped Query Attention

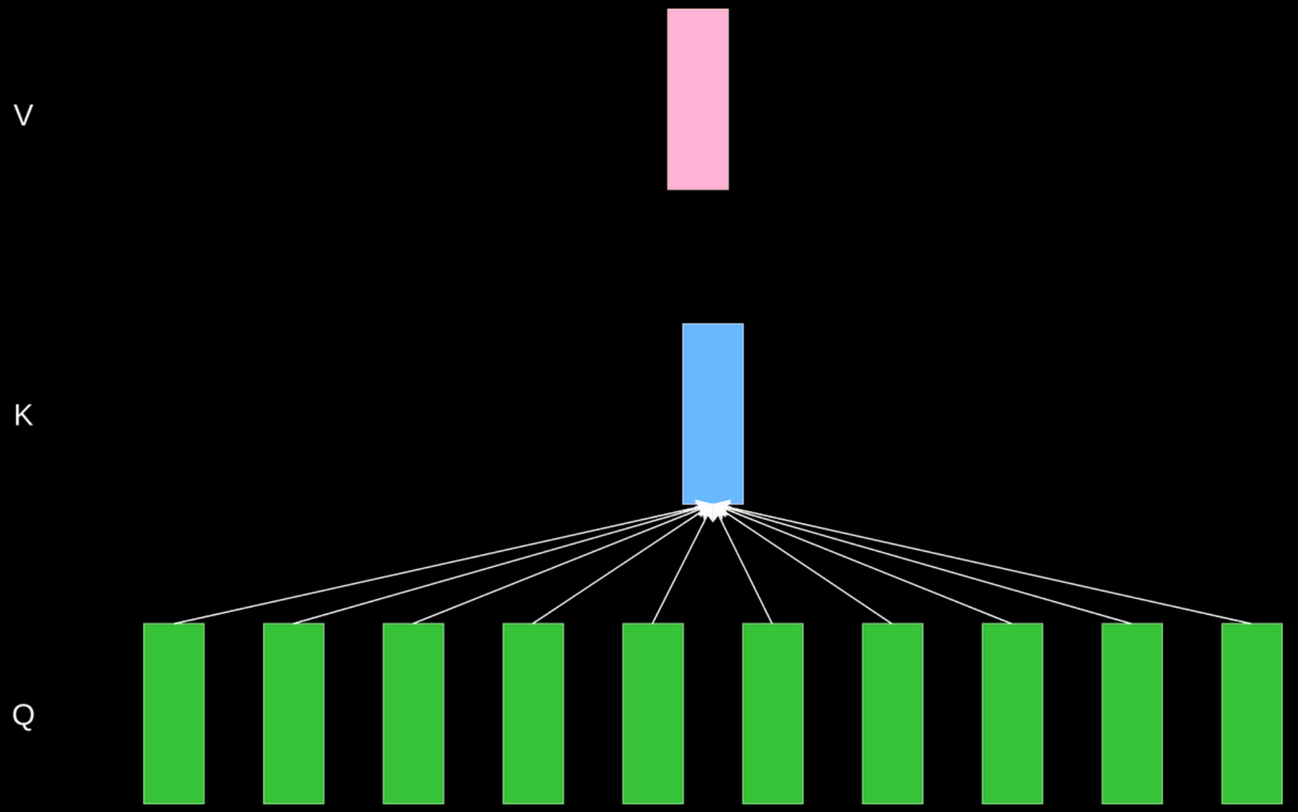


Multi Query Attention



Multi Query Attention

Do you think this reduces the quality of generation?



Section Summary

We saw

1. Various architectures for our LLM
2. Positional Embeddings and ow to scale beyond 3000 tokens
3. Attention mechanisms

Questions

Conclusion

- Pretraining a LLM is a daunting task
- Scaling laws need to be derived for optimal compute usage.
- Data quality is very important!
- Choosing a good architecture is important as it determines how much compute you use!

Thank You
for
Listening