

# Deep Learning for Computer Vision

Andrii Liubonko

Grammarly\*



\* The opinions expressed in this presentation and on the following slides are solely those of the presenter and not necessarily those of Grammarly

# **Logistics**

## **4 units**

### **3 types of homework:**

- paper review [20 % of FINAL SCORE]
- notebooks [30 % of FINAL SCORE]
- mini-project [50 % of FINAL SCORE]

### **important dates:**

*31 January, 23:59*

### **course repo:**

<https://github.com/lyubonko/ucu2022cv>

# **Overview of the course**

## **Unit I**

[T] Intro

[P] pytorch

## **Unit II**

[T] CNNs in depth, Object Detection

[P] simple nets

## **Unit III**

[T] Attention in CV, Transformers

[P] classification

## **Unit IV**

[T] Generative models, Diffusion Models

[P] project structure, stable diffusion

# Content of today's lecture

- Attention Intro
  - Brief History
  - Seq2seq Attention
  - Self-Attention
  - Transformer
- Attention in CV
  - SE
  - CBAM
  - ViT
  - latest developments

# Brief History of Attention

## Seq2Seq

- Cho et al. (2014)
- Sutskever et al. (2014)



2014

## Align & Translate

- Bahdanau et al. (2015)
- Luong et al. (2015)



2015

## Self attention / Transformer

- Vaswani et al (2017)



2016

2017

## BERT

- Devlin et al (2018)



2018

## GPT-3

- Tom B. Brown et al (2020)



2020

2021

## Visual attention

- Xu et al. (2015)



## SE networks

- Hu et al. (2018)



## CBAM

- Woo et al. (2018)



## ViT

- Dosovitskiy et al. (2020)



# Brief History of Attention

Attention  
Intro

## Seq2Seq

- Cho et al. (2014)
- Sutskever et al. (2014)

## Align & Translate

- Bahdanau et al. (2015)
- Luong et al. (2015)

## Self attention / Transformer

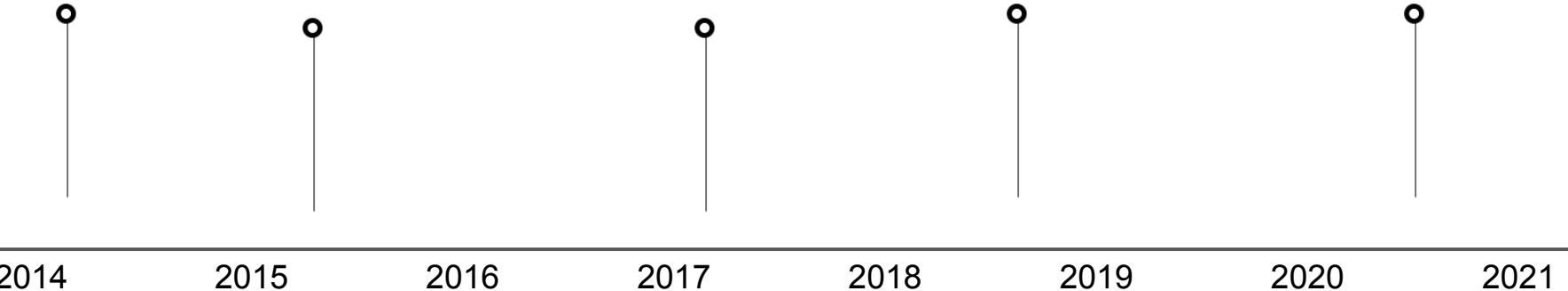
- Vaswani et al (2017)

## BERT

- Devlin et al (2018)

## GPT-3

- Tom B. Brown et al (2020)



Attention in  
CV

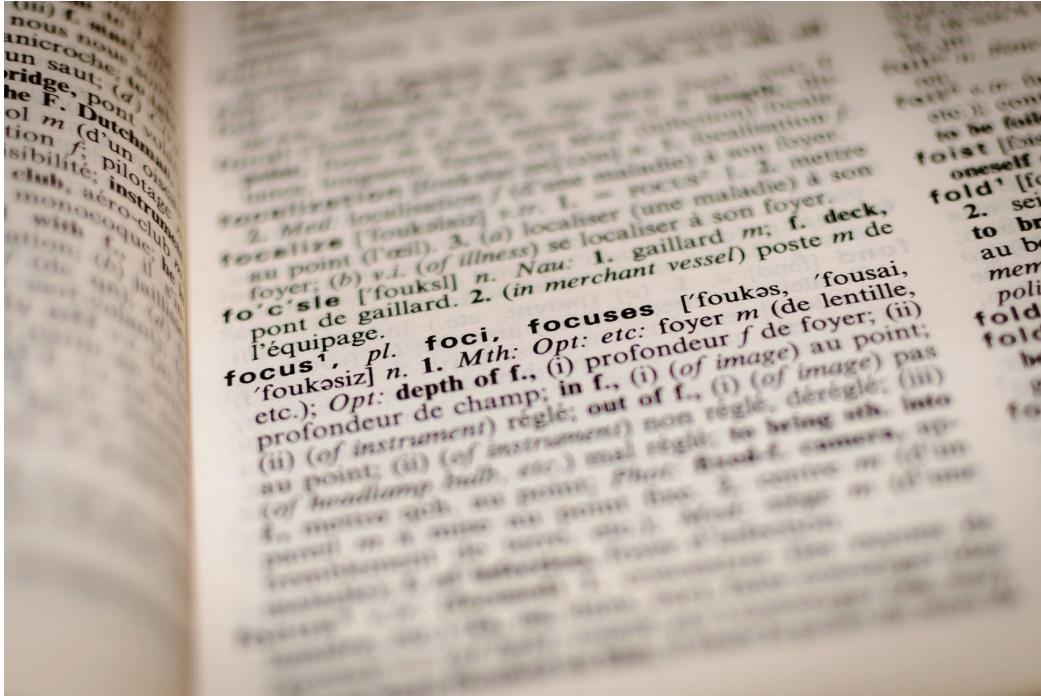
Visual attention  
• Xu et al. (2015)

SE networks  
• Hu et al. (2018)

CBAM  
• Woo et al. (2018)

ViT  
• Dosovitskiy et al. (2020)

# Attention [intro]



[image source](#)

# Attention [intro]



[image source](#)

# Attention [intro]



[image source](#)

# Attention [intro]



[image source](#)

# Attention [intro]



## Attention [intro]

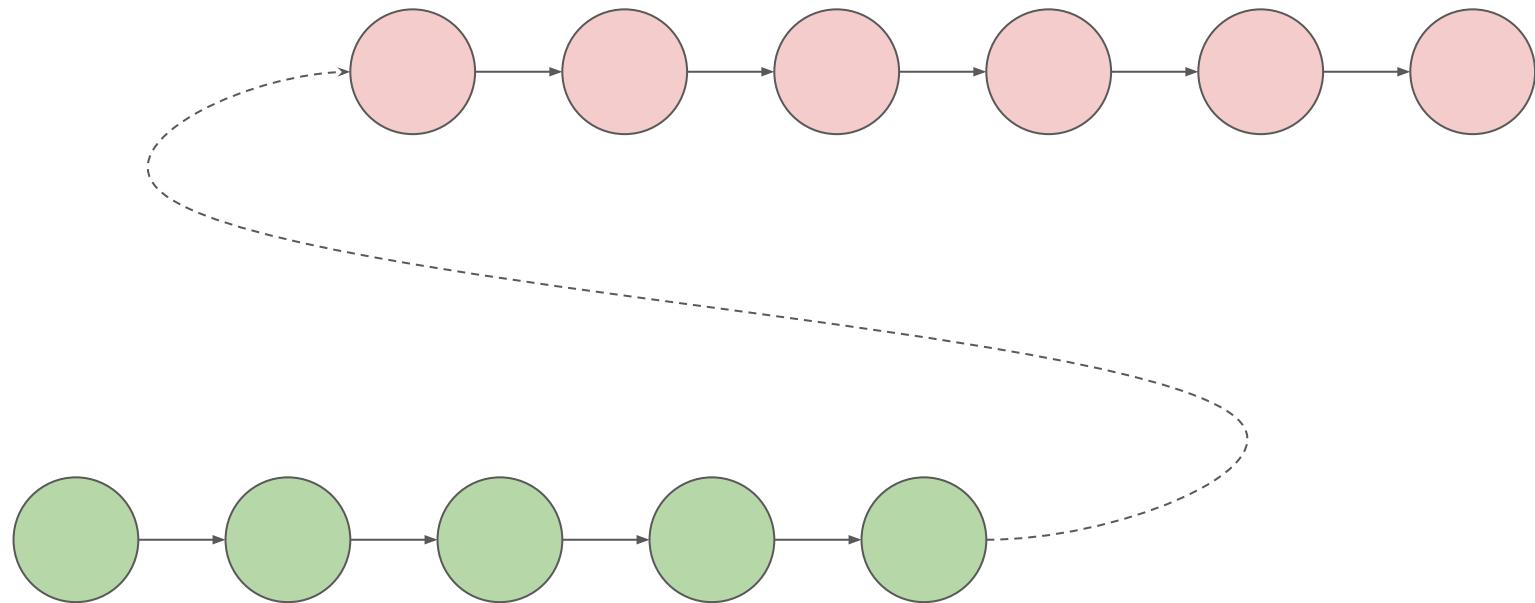
The diagram shows a horizontal bracket above the sentence, divided into three segments by vertical bars. The first segment is labeled "low attention" and covers the words "She", "is", and "eating". The third segment is labeled "high attention" and covers the words "apple.". The second segment, containing "a" and "green", is unlabeled.

high attention

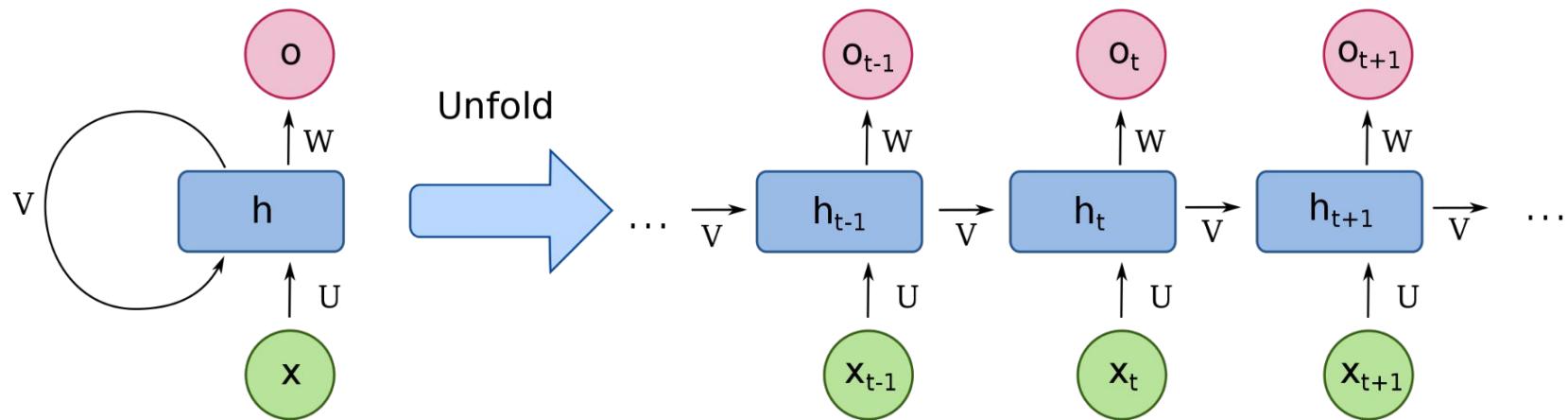
low attention

She is eating a green apple.

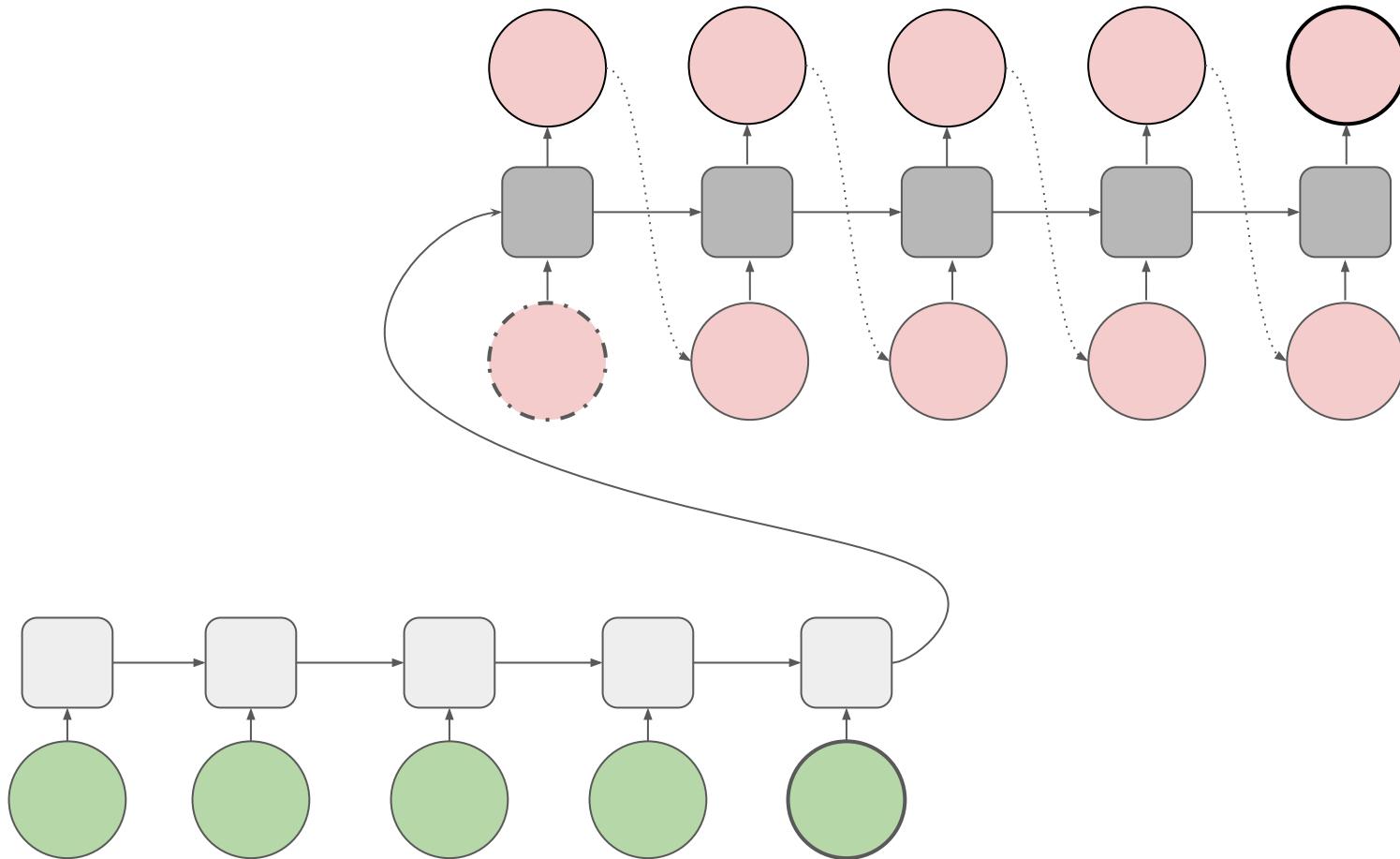
# Attention [seq2seq]



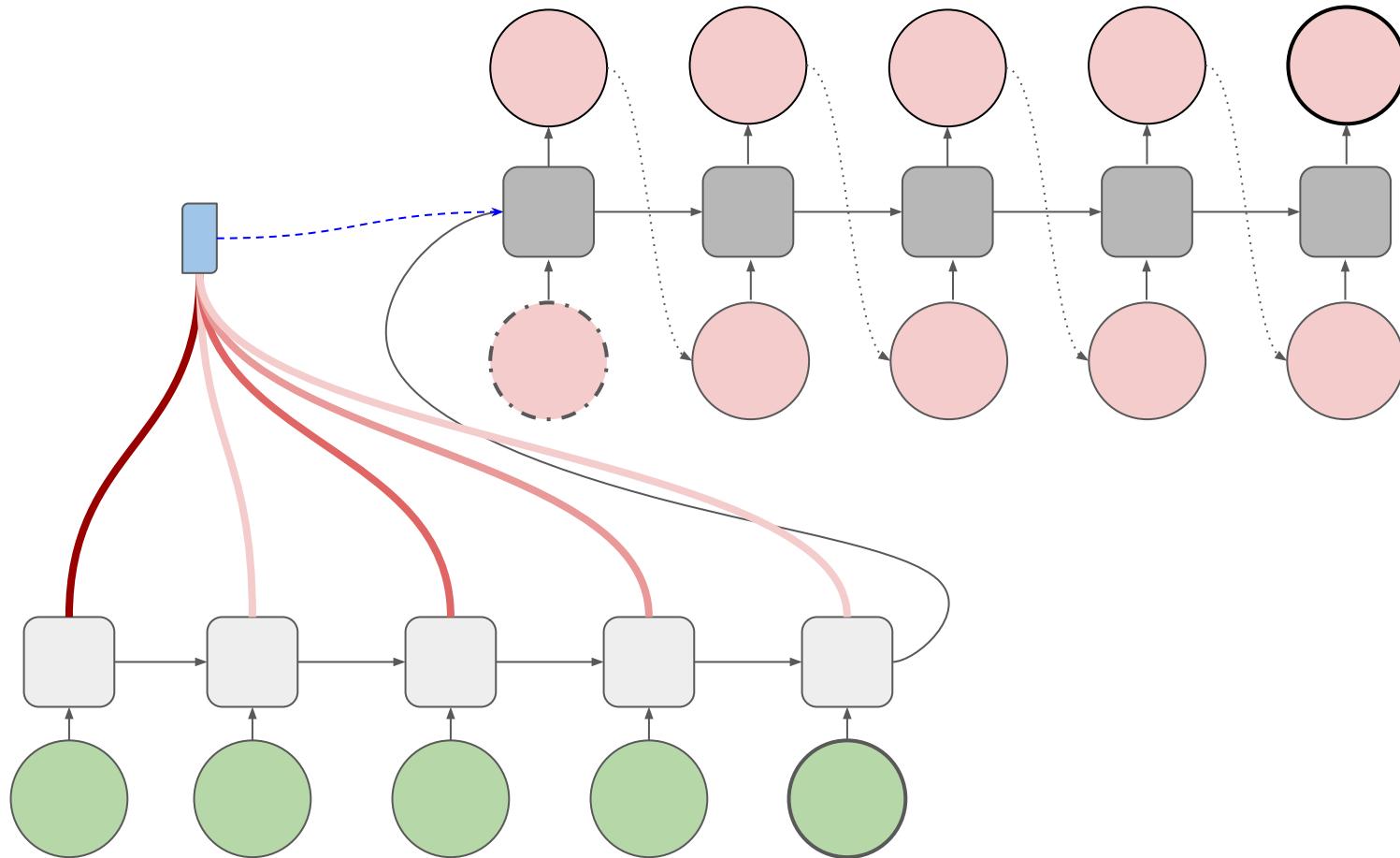
# Attention [seq2seq]



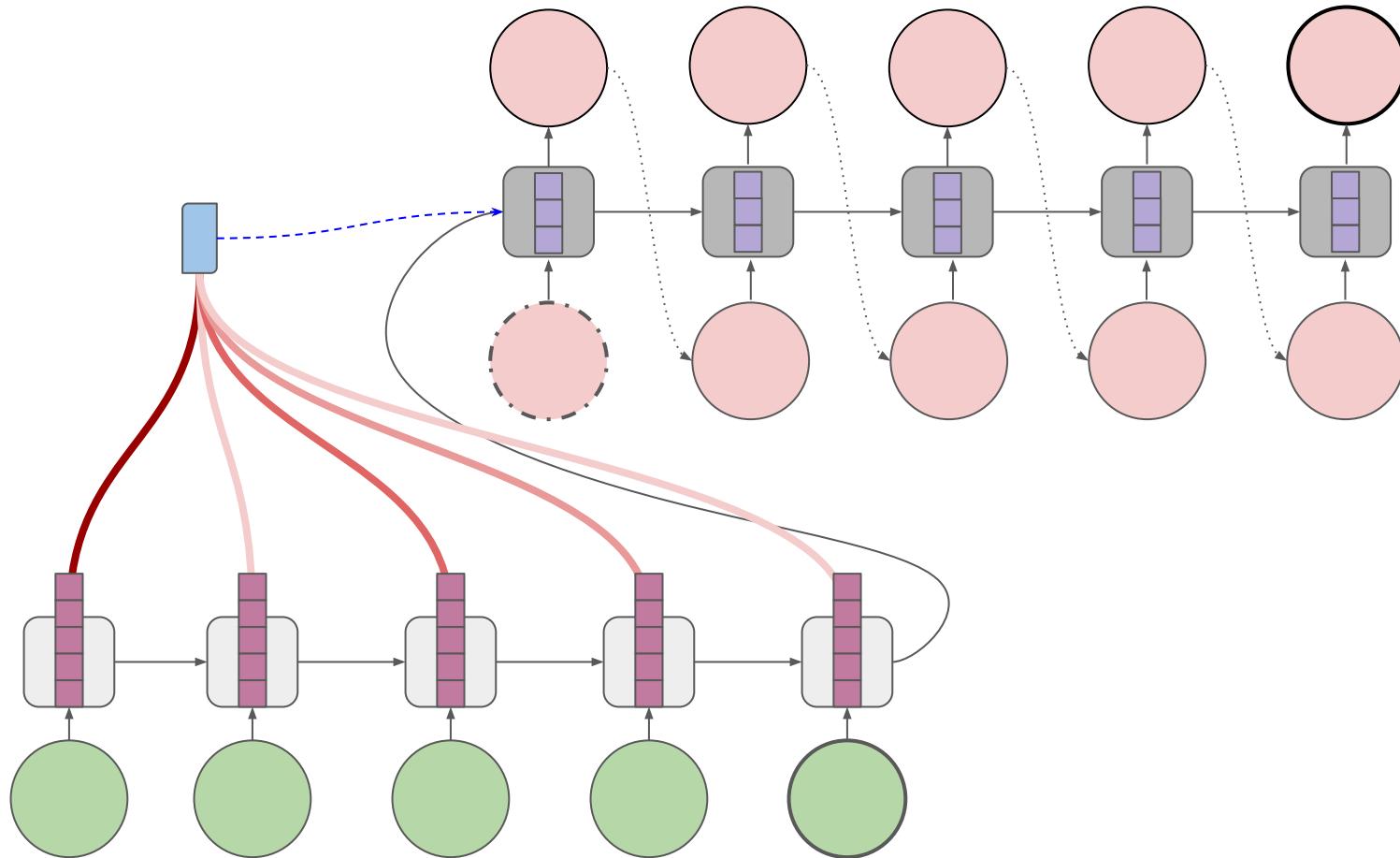
# Attention [seq2seq]



# Attention [seq2seq]



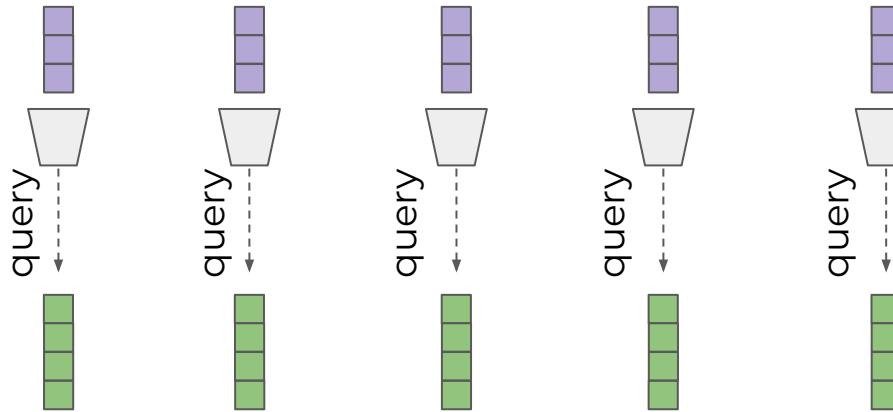
# Attention [seq2seq]



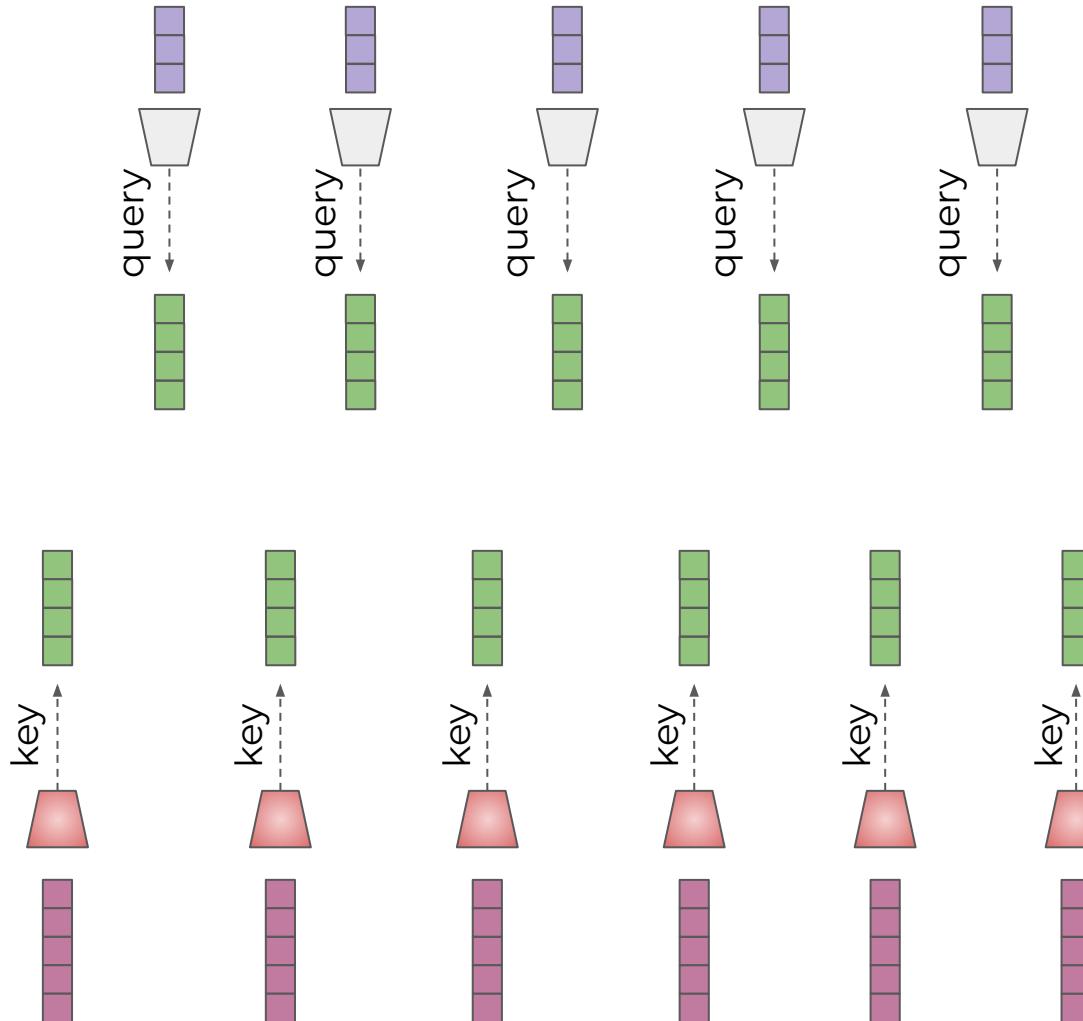
# Attention [seq2seq]



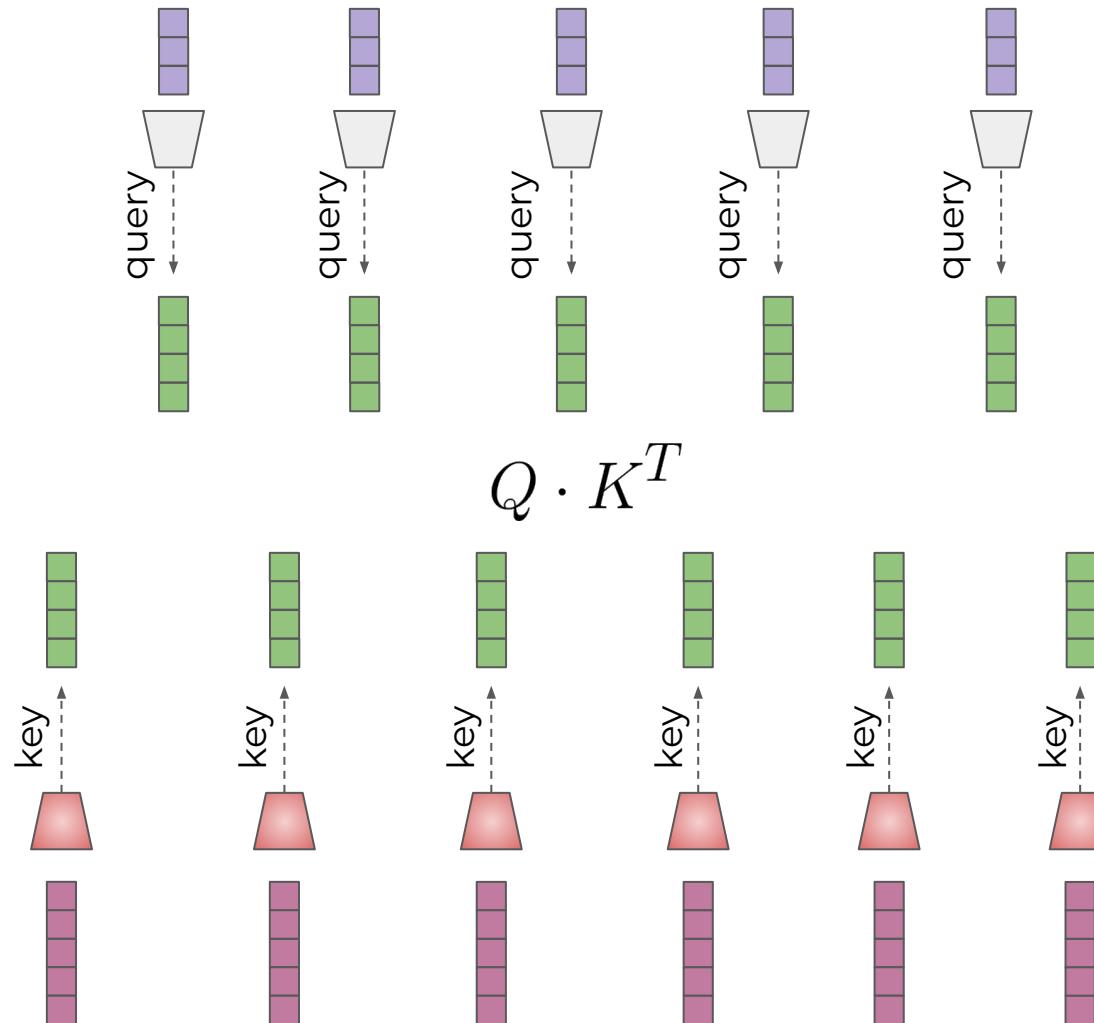
# Attention [seq2seq]



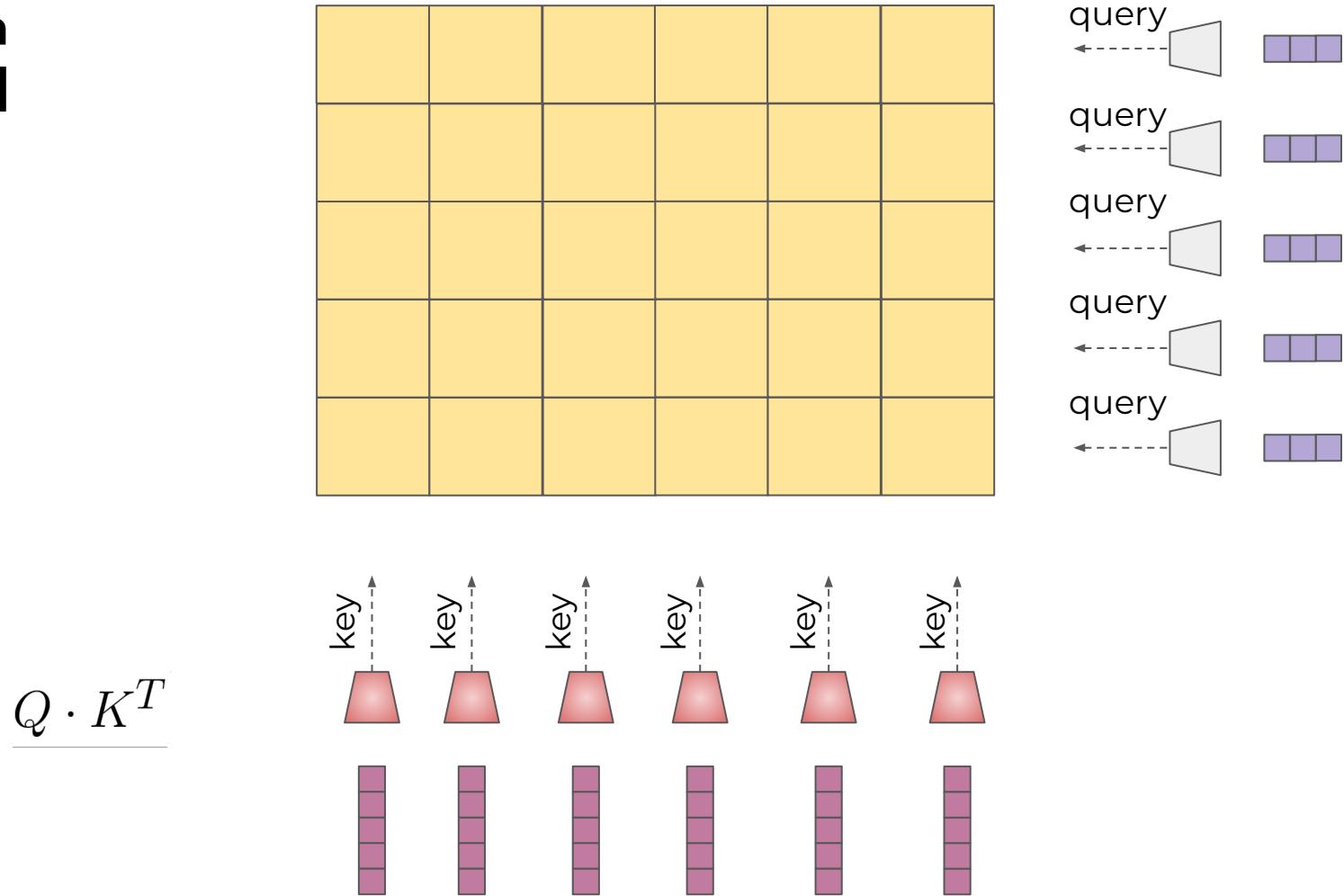
# Attention [seq2seq]



# Attention [seq2seq]

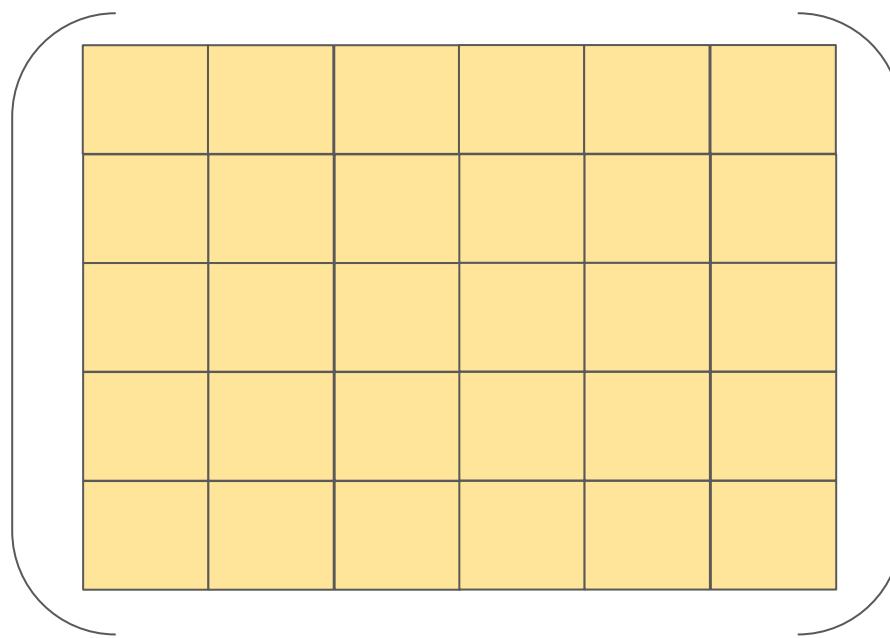


# Attention [seq2seq]



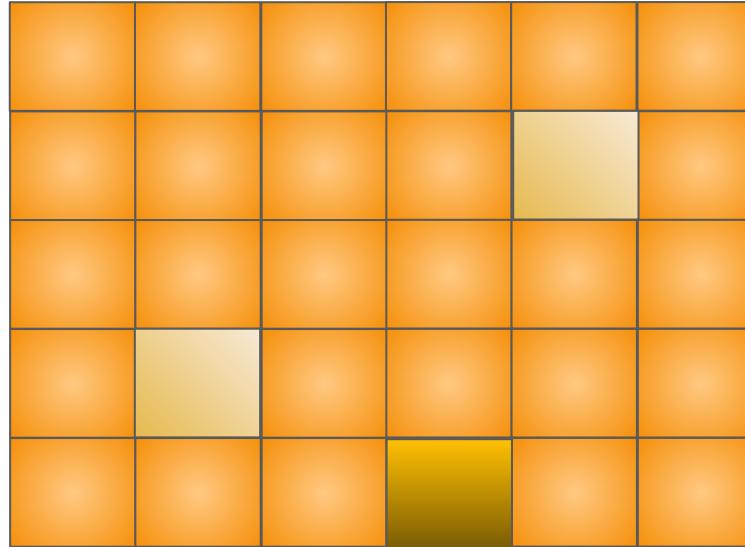
# Attention [seq2seq]

softmax



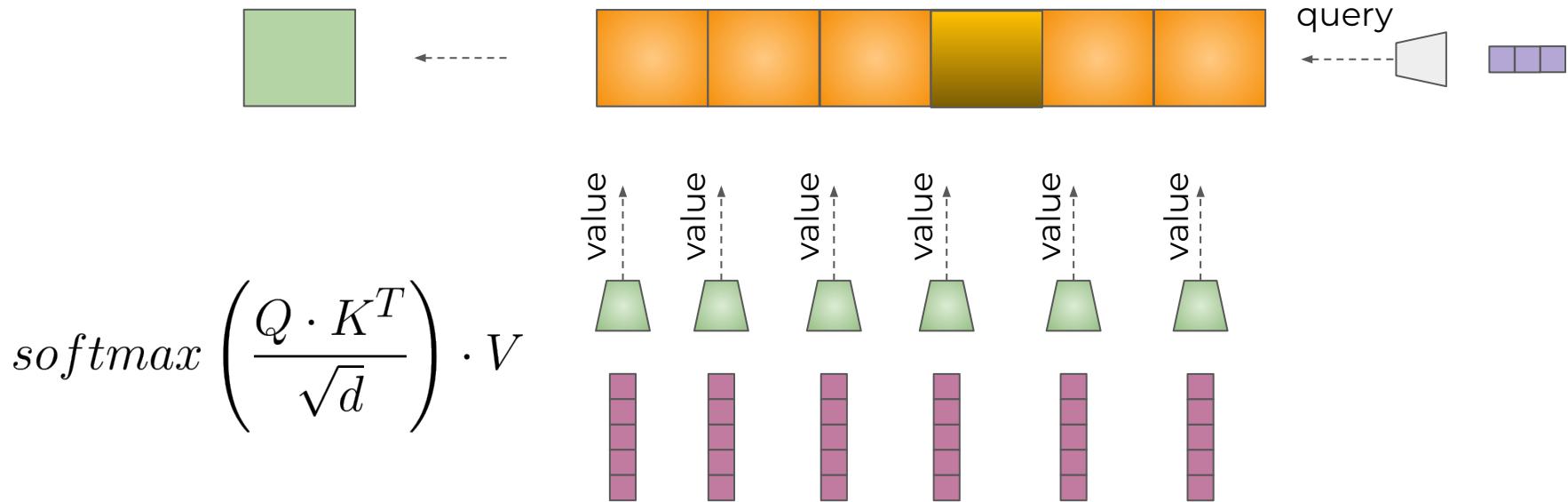
$$\text{softmax} \left( \frac{Q \cdot K^T}{\sqrt{d}} \right)$$

# Attention [seq2seq]

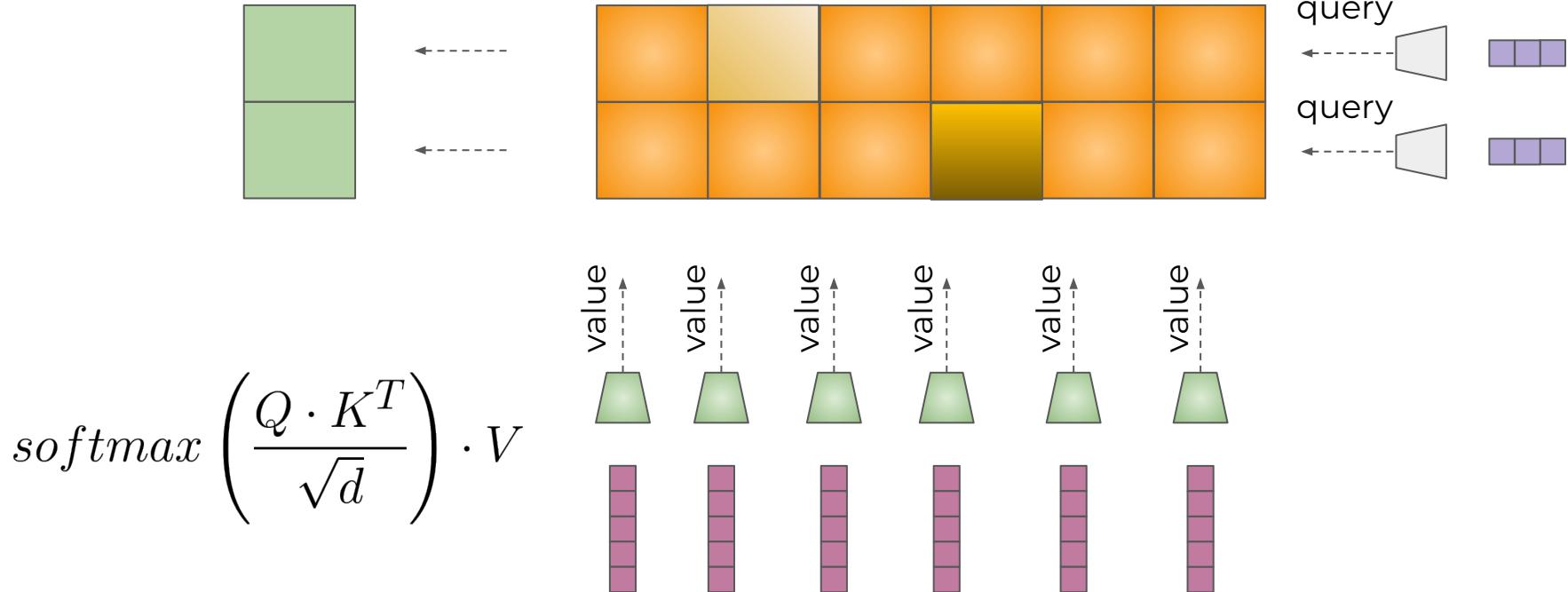


$$\text{softmax} \left( \frac{Q \cdot K^T}{\sqrt{d}} \right)$$

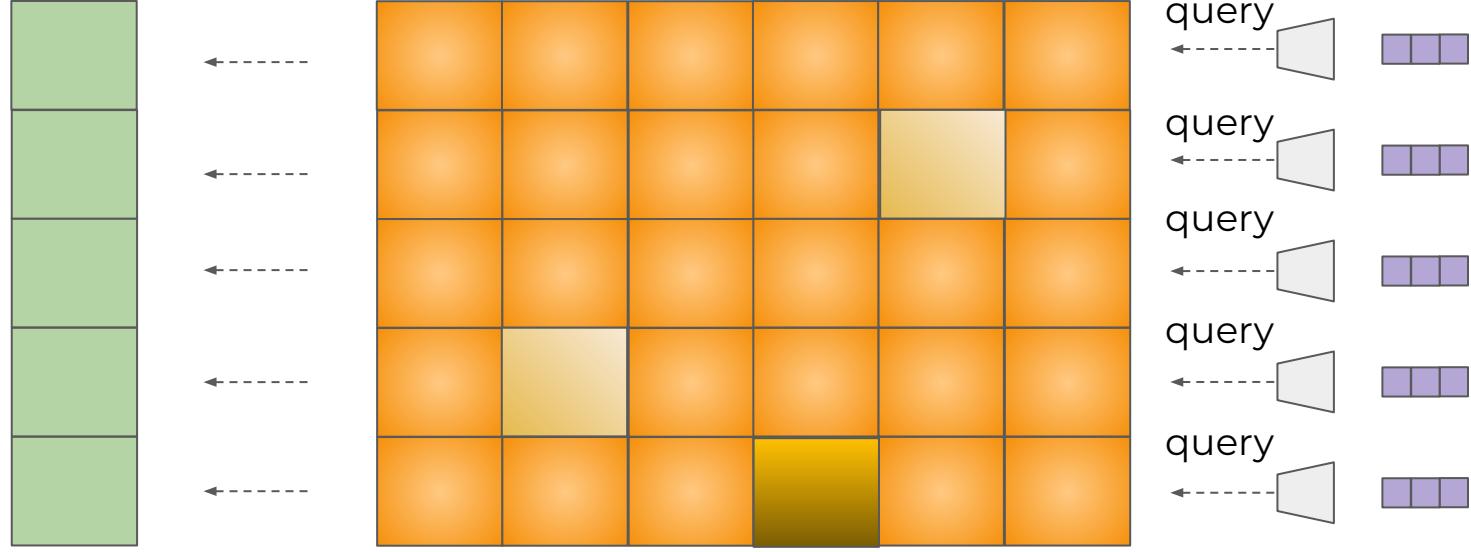
# Attention [seq2seq]



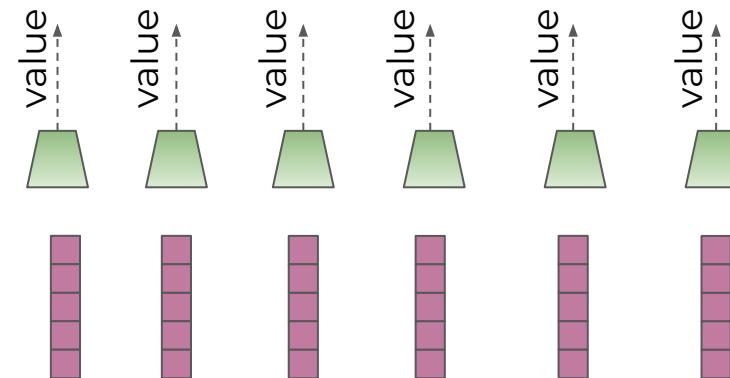
# Attention [seq2seq]



# Attention [seq2seq]



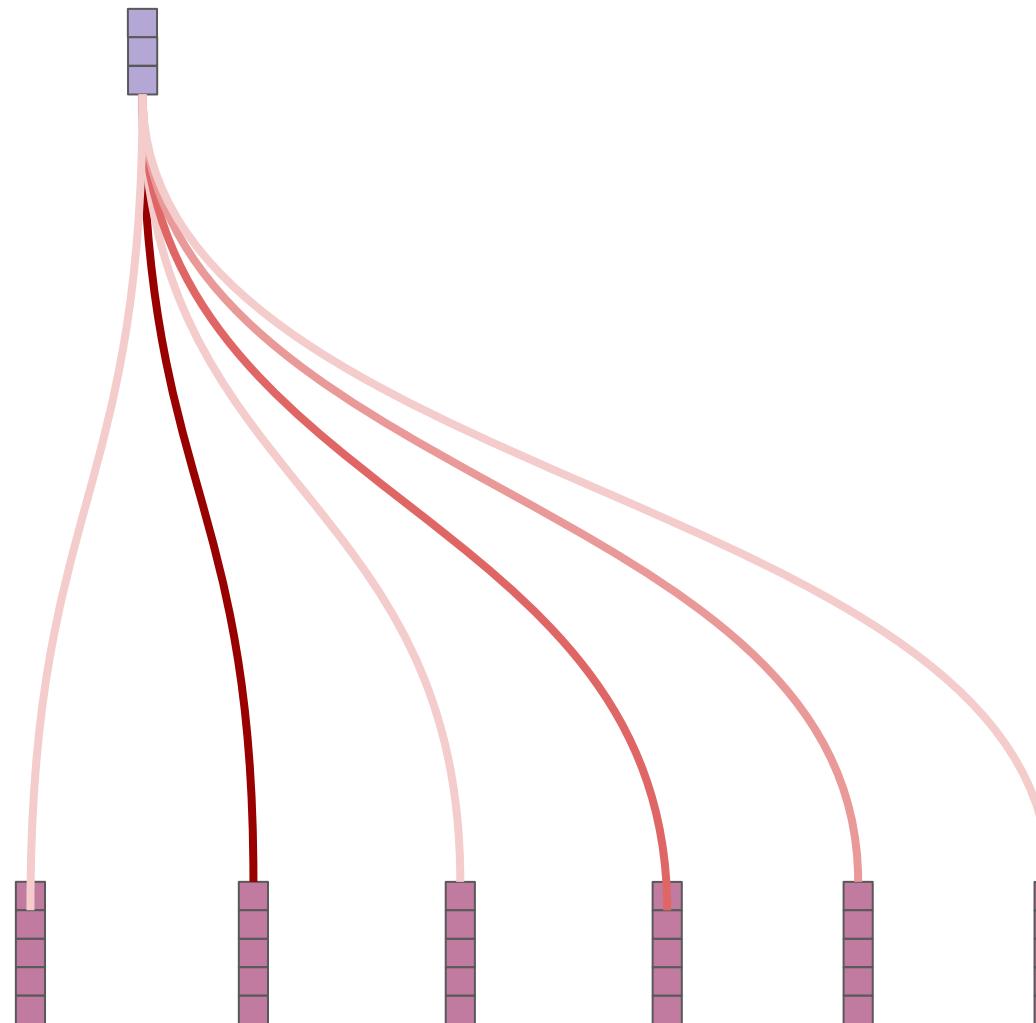
$$\text{softmax} \left( \frac{Q \cdot K^T}{\sqrt{d}} \right) \cdot V$$



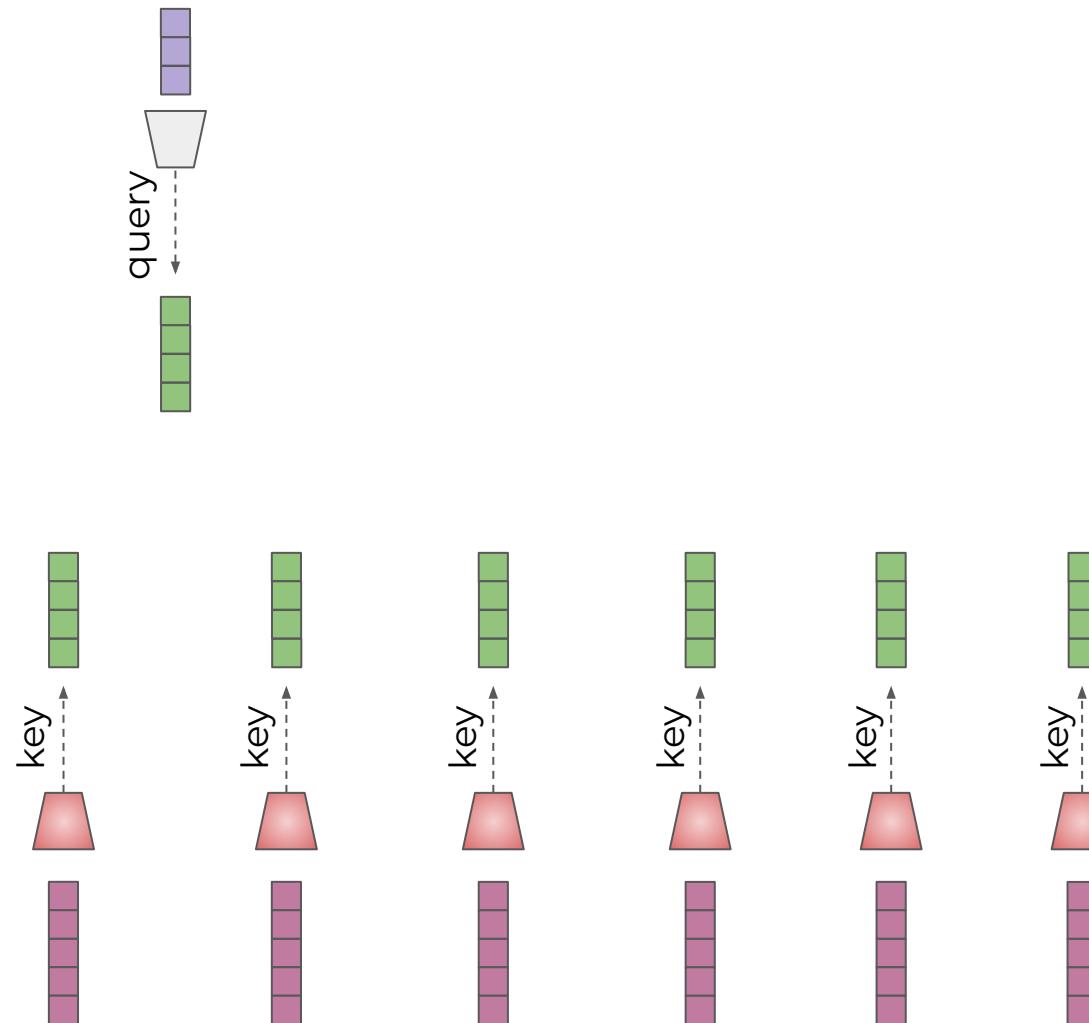
# Attention [seq2seq]

... recap ...

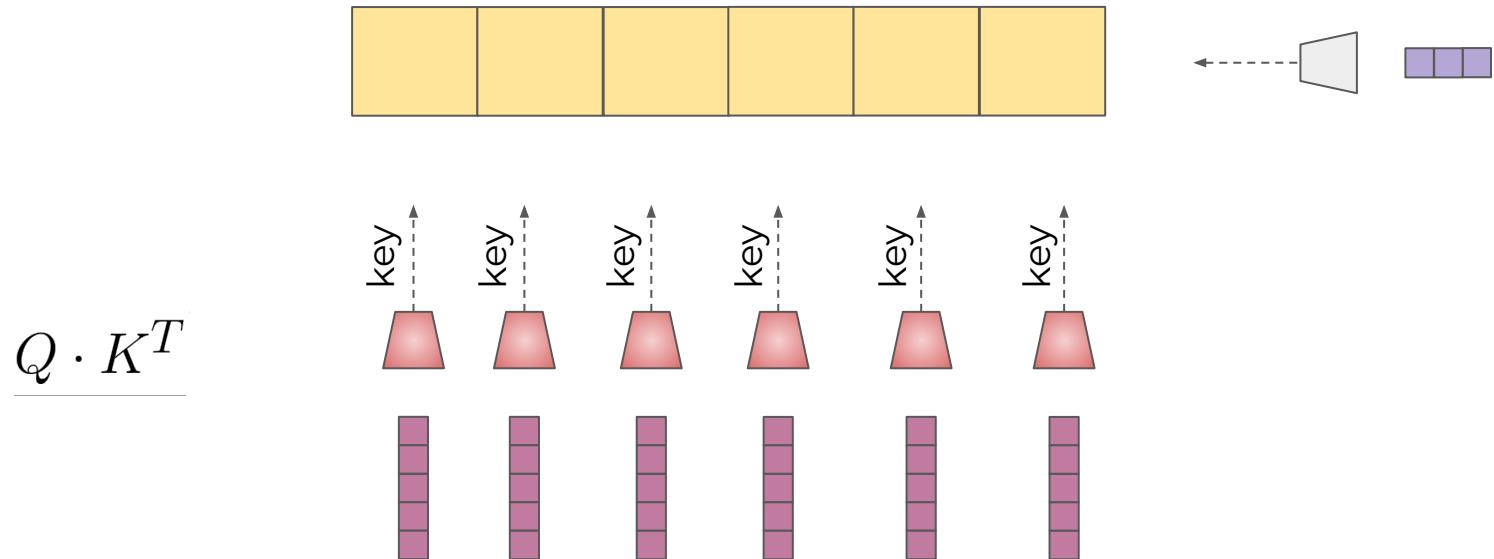
# Attention [seq2seq]



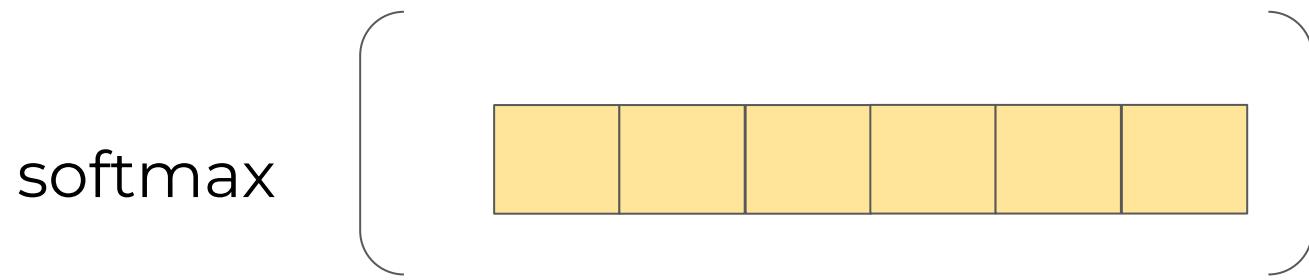
# Attention [seq2seq]



# Attention [seq2seq]



# Attention [seq2seq]



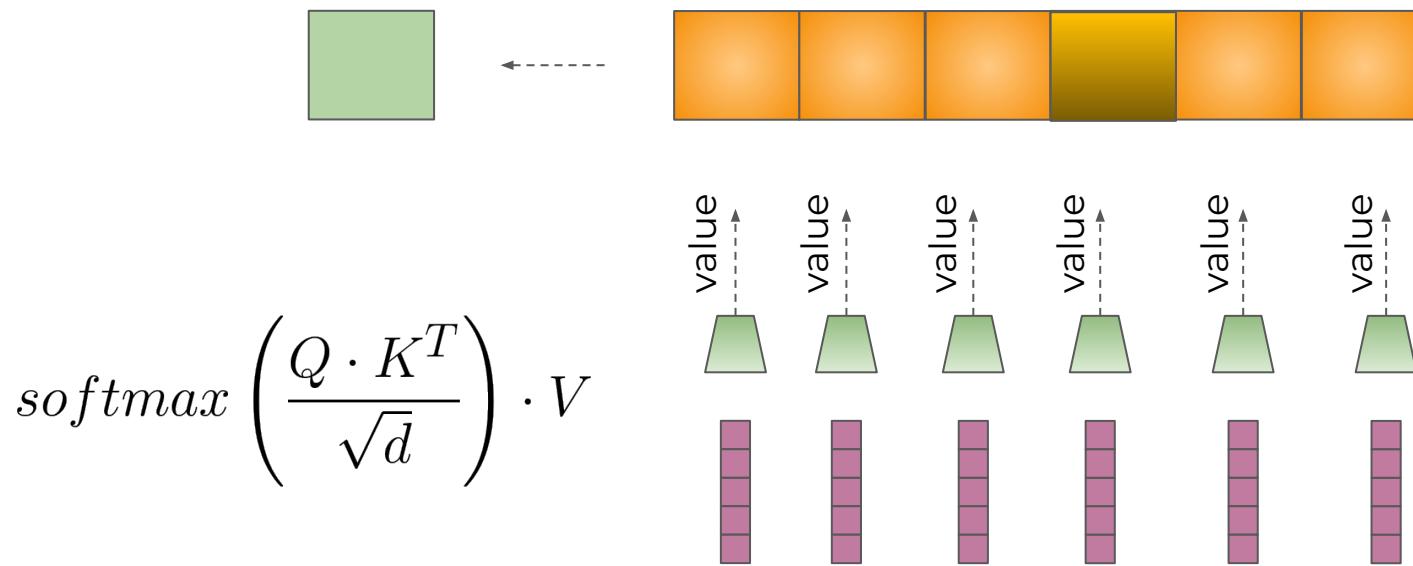
$$\text{softmax} \left( \frac{Q \cdot K^T}{\sqrt{d}} \right)$$

# Attention [seq2seq]

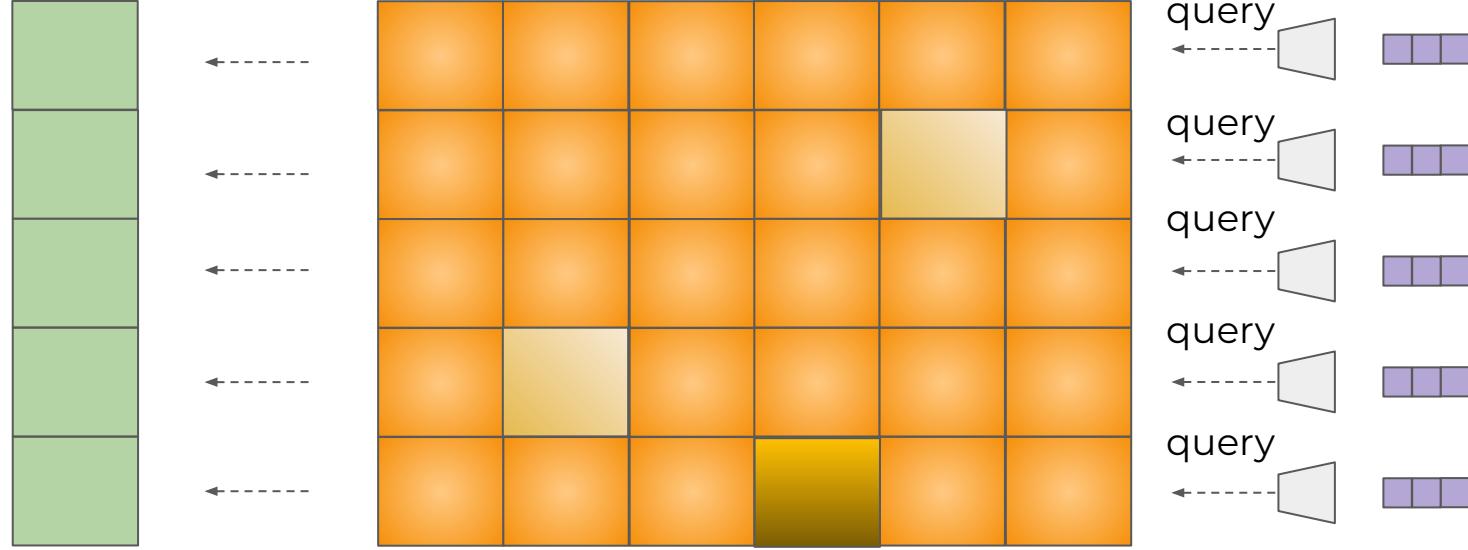


$$\text{softmax} \left( \frac{Q \cdot K^T}{\sqrt{d}} \right)$$

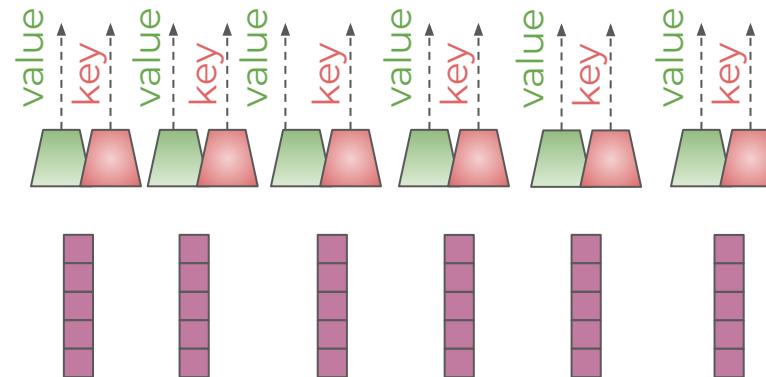
# Attention [seq2seq]



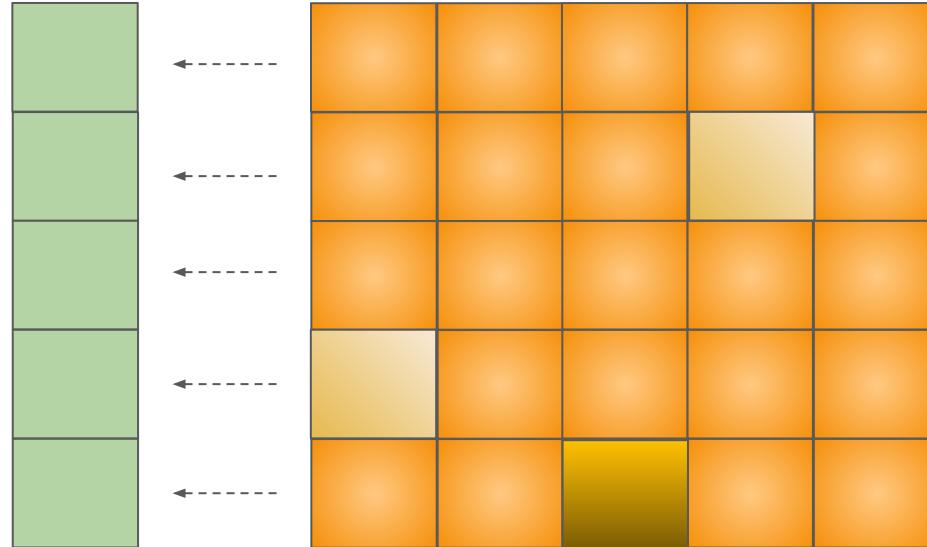
# Attention [seq2seq]



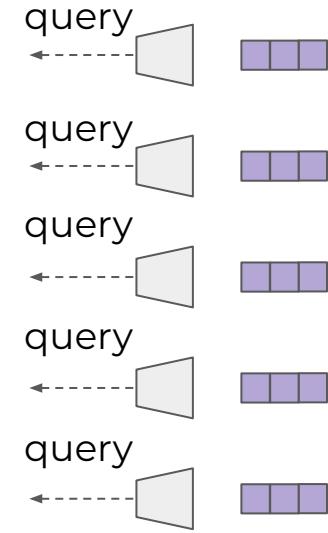
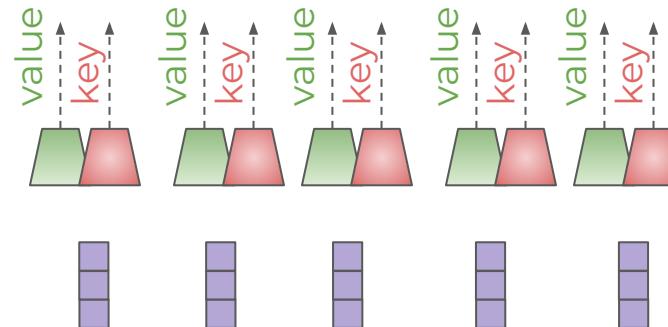
$$\text{softmax} \left( \frac{Q \cdot K^T}{\sqrt{d}} \right) \cdot V$$



# Attention [self-attention]



$$\text{softmax} \left( \frac{Q \cdot K^T}{\sqrt{d}} \right) \cdot V$$

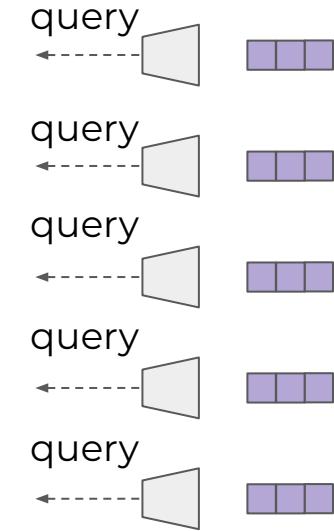
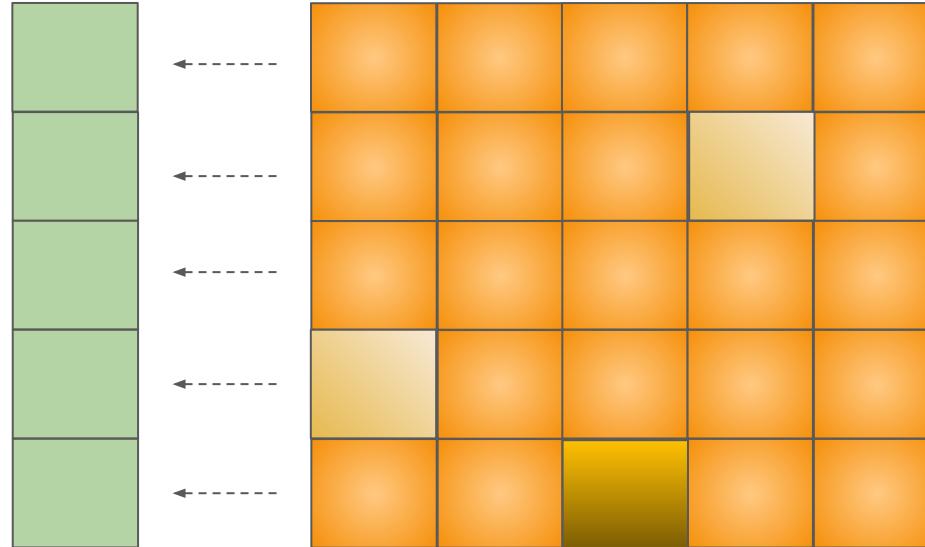


# Attention

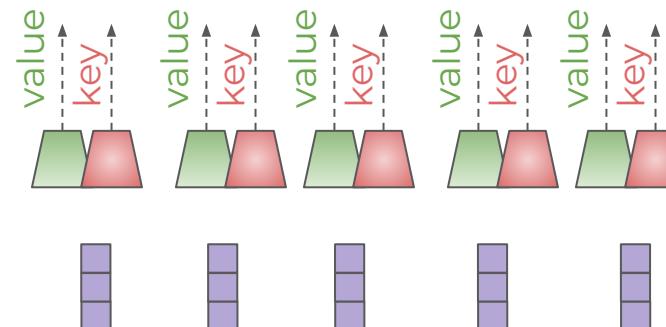
## [self-attention]

$$\text{Attention} = f(g(x), x)$$

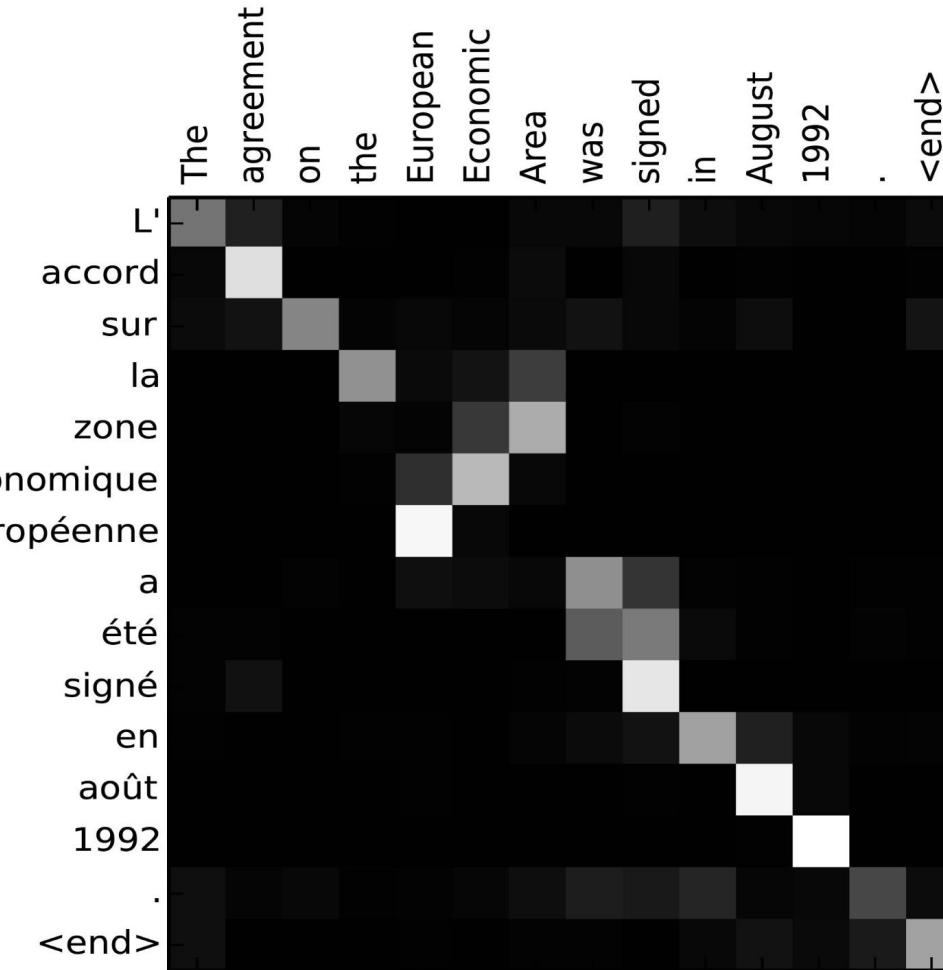
# Attention [self-attention]



$$Q, K, V = \text{Linear}(x)$$
$$g(x) = \text{Softmax}(QK)$$
$$f(g(x), x) = g(x)V$$



# Attention [seq2seq]



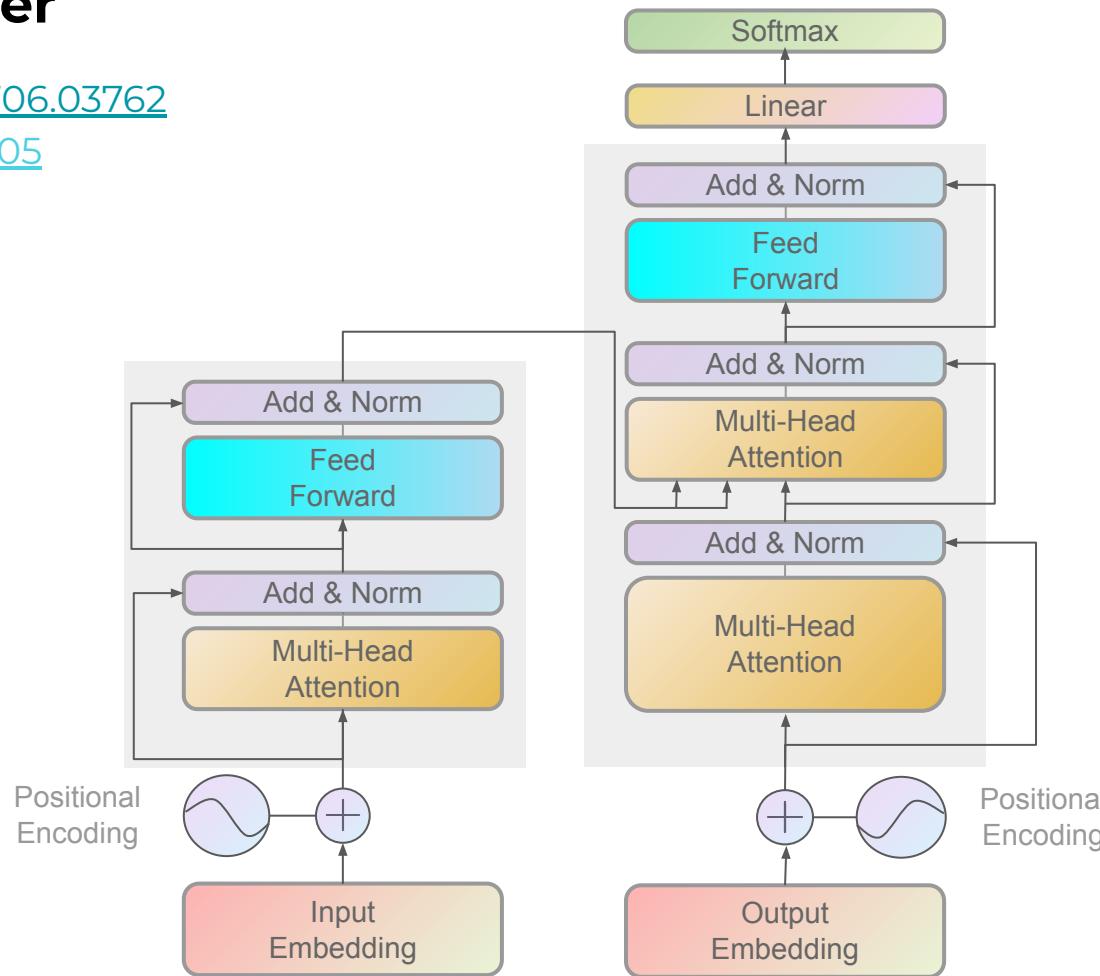
# Content of today's lecture

- Attention Intro
  - Brief History
  - Seq2seq Attention
  - Self-Attention
  - Transformer
- Attention in CV
  - SE
  - CBAM
  - ViT
  - latest developments

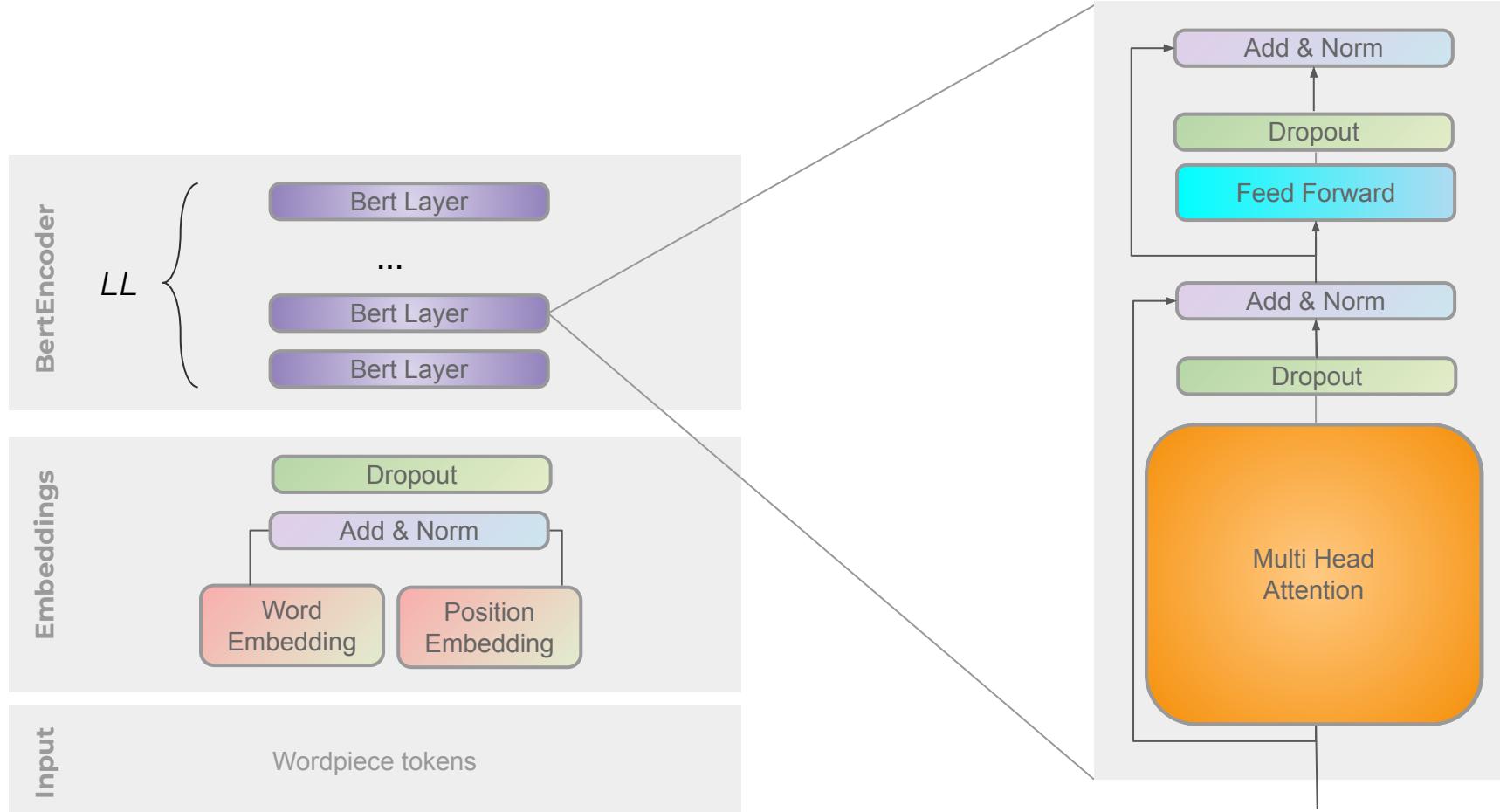
# Transformer

Transformer: [1706.03762](#)

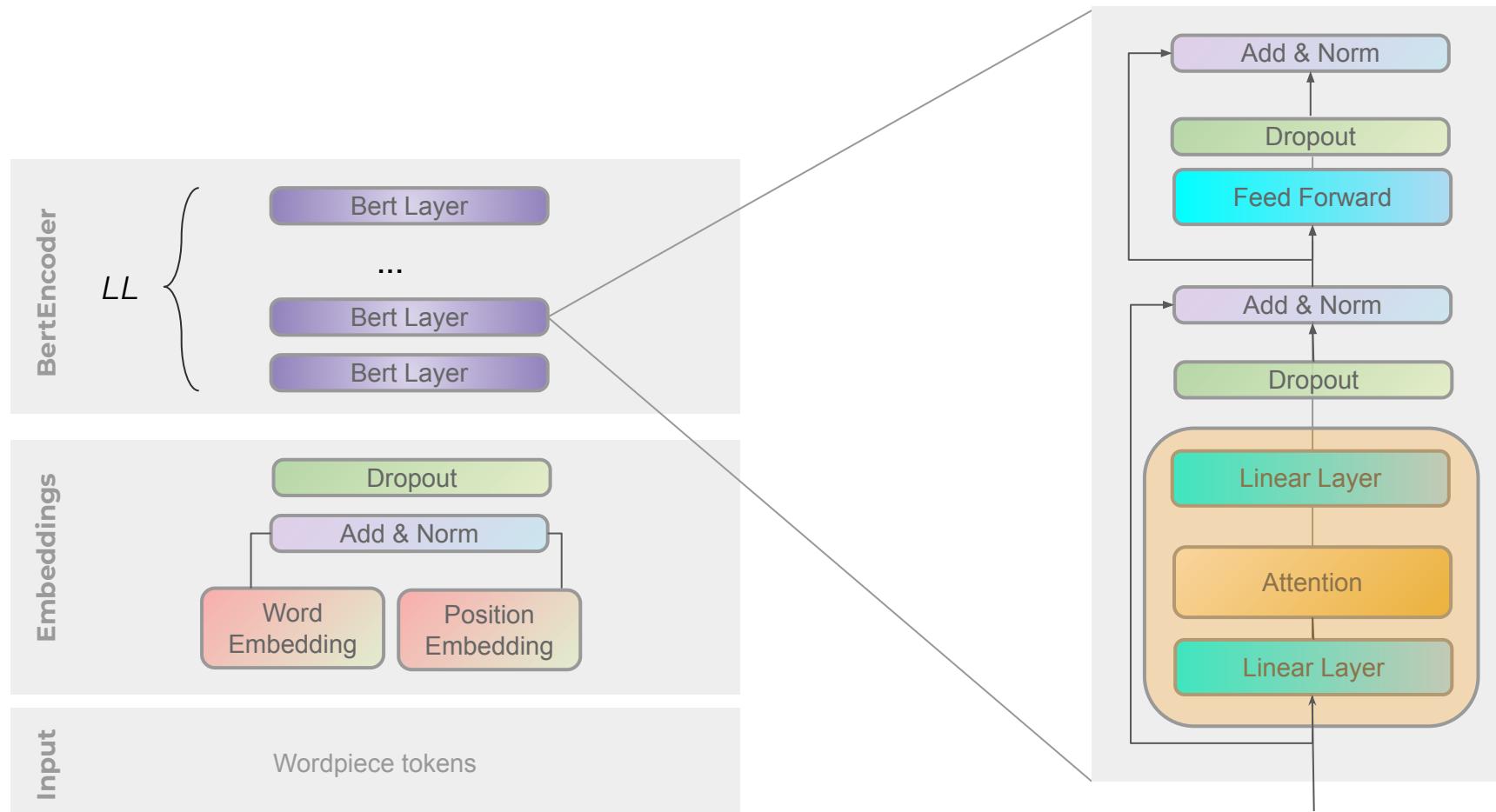
BERT: [1810.04805](#)



# Transformer [Encoder]



# Transformer [Encoder]



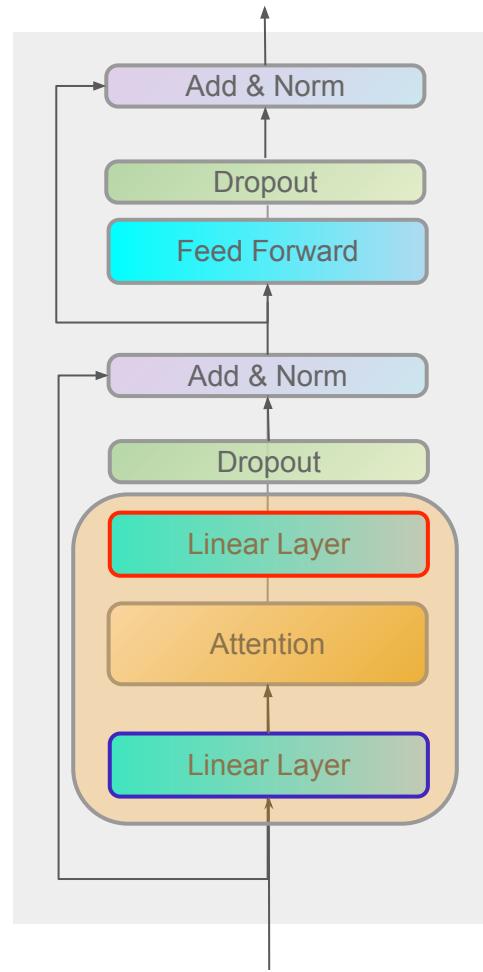
# Transformer [Multi head attention]

$$FC(X) = \text{Linear}_{[\mathbf{W}_{\text{out}}, \mathbf{b}_{\text{out}}]}(\sigma(\text{Linear}_{[\mathbf{W}_{\text{in}}, \mathbf{b}_{\text{in}}]}(X)))$$

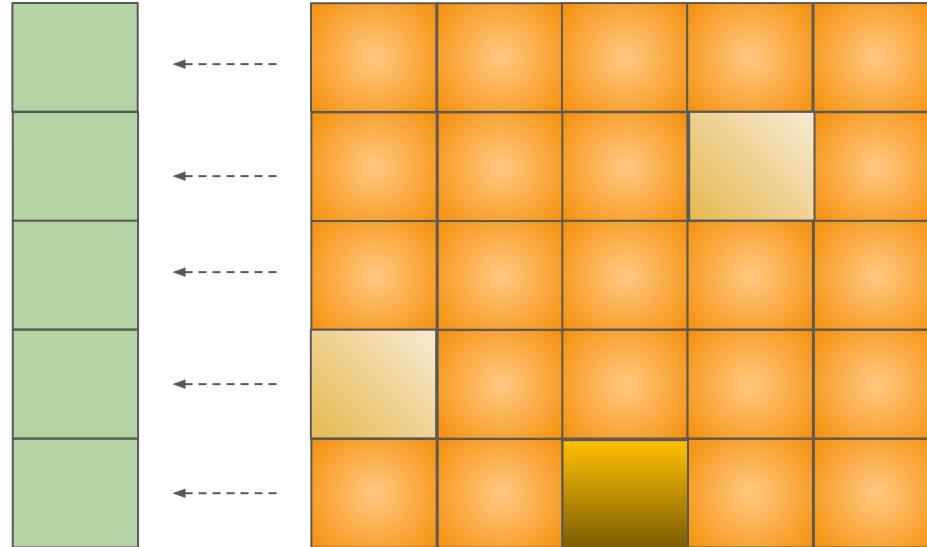
$$\text{MultiHead}(\mathbf{X}, \mathbf{X}) = \text{Linear}_{[\mathbf{W}^{\text{o}}, \mathbf{b}^{\text{o}}]} \left( \underset{i \in [A]}{\text{concat}} [\mathbf{H}^i] \right)$$

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{Q \cdot K^T}{\sqrt{d}} \right) \cdot V$$

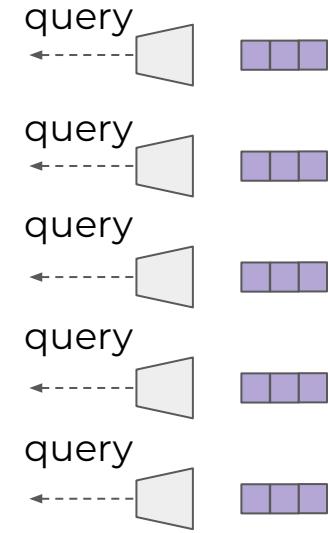
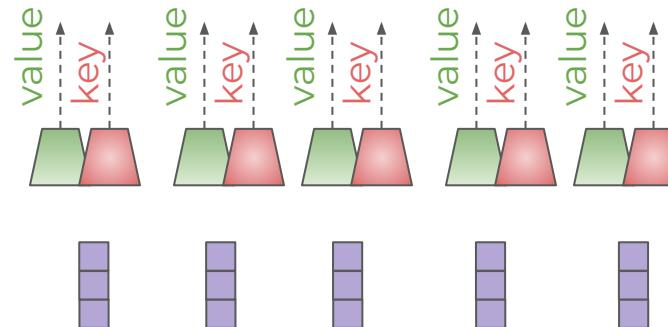
$$\begin{aligned} \mathbf{H}^i &= \text{Attention}(\text{Linear}_{[\mathbf{W}_Q^{(i)}, \mathbf{b}_Q^{(i)}]}(X), \\ &\quad \text{Linear}_{[\mathbf{W}_K^{(i)}, \mathbf{b}_K^{(i)}]}(X), \\ &\quad \text{Linear}_{[\mathbf{W}_V^{(i)}, \mathbf{b}_V^{(i)}]}(X)) \end{aligned}$$



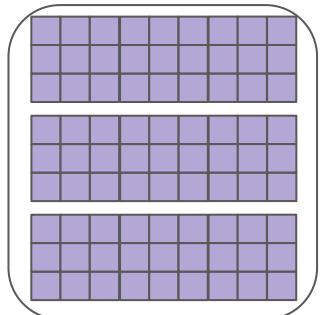
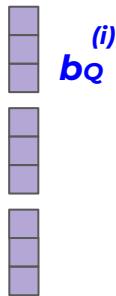
# Attention [self-attention]



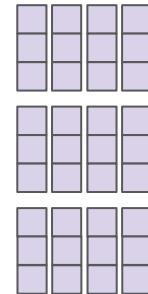
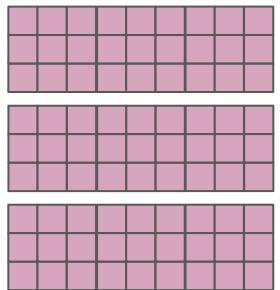
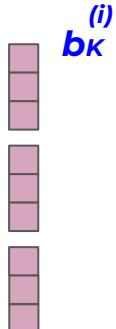
$$\text{softmax} \left( \frac{Q \cdot K^T}{\sqrt{d}} \right) \cdot V$$



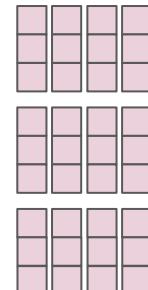
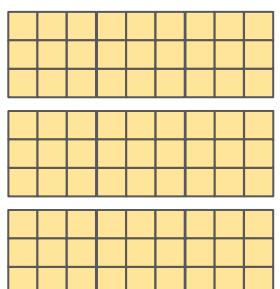
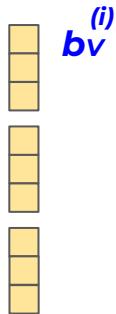
# Transformer [Multi-Head Attention (MHA)]

 $W_Q^{(i)}$  $b_Q^{(i)}$ 

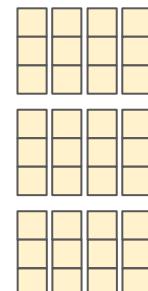
⋮

 $Q^{(i)}$  $W_K^{(i)}$  $b_K^{(i)}$ 

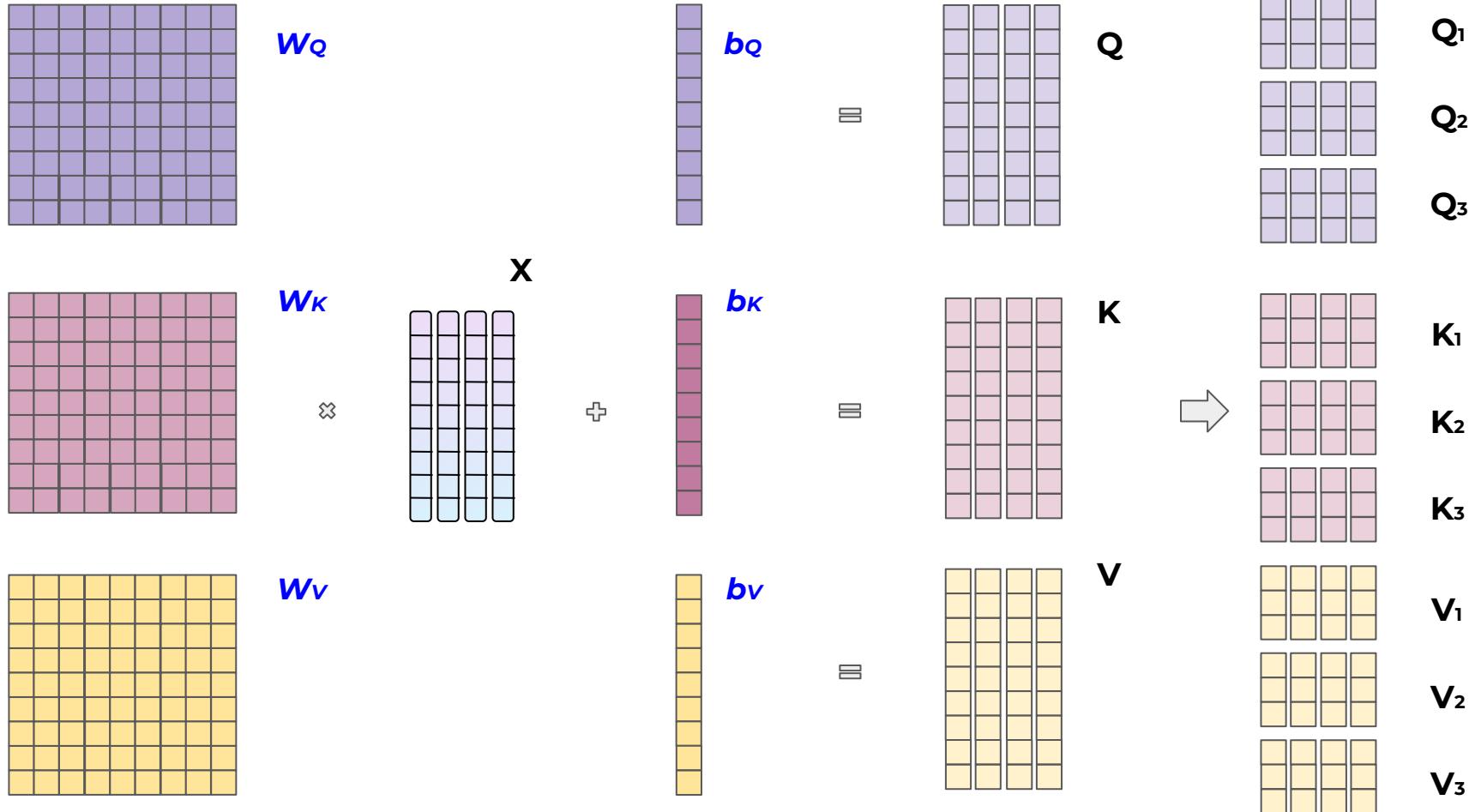
⋮

 $K^{(i)}$  $W_V^{(i)}$  $b_V^{(i)}$ 

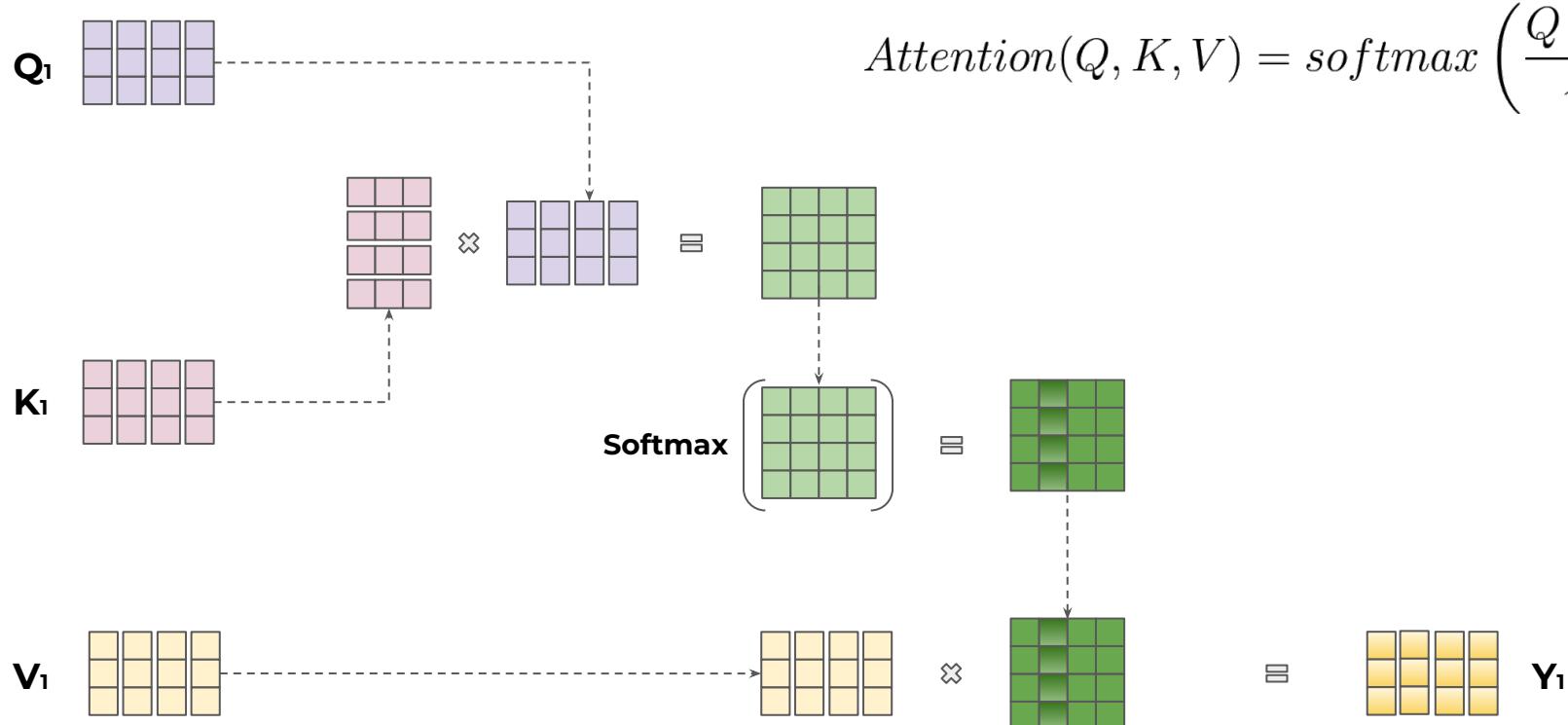
⋮

 $V^{(i)}$  $i = 1 \dots \#head$

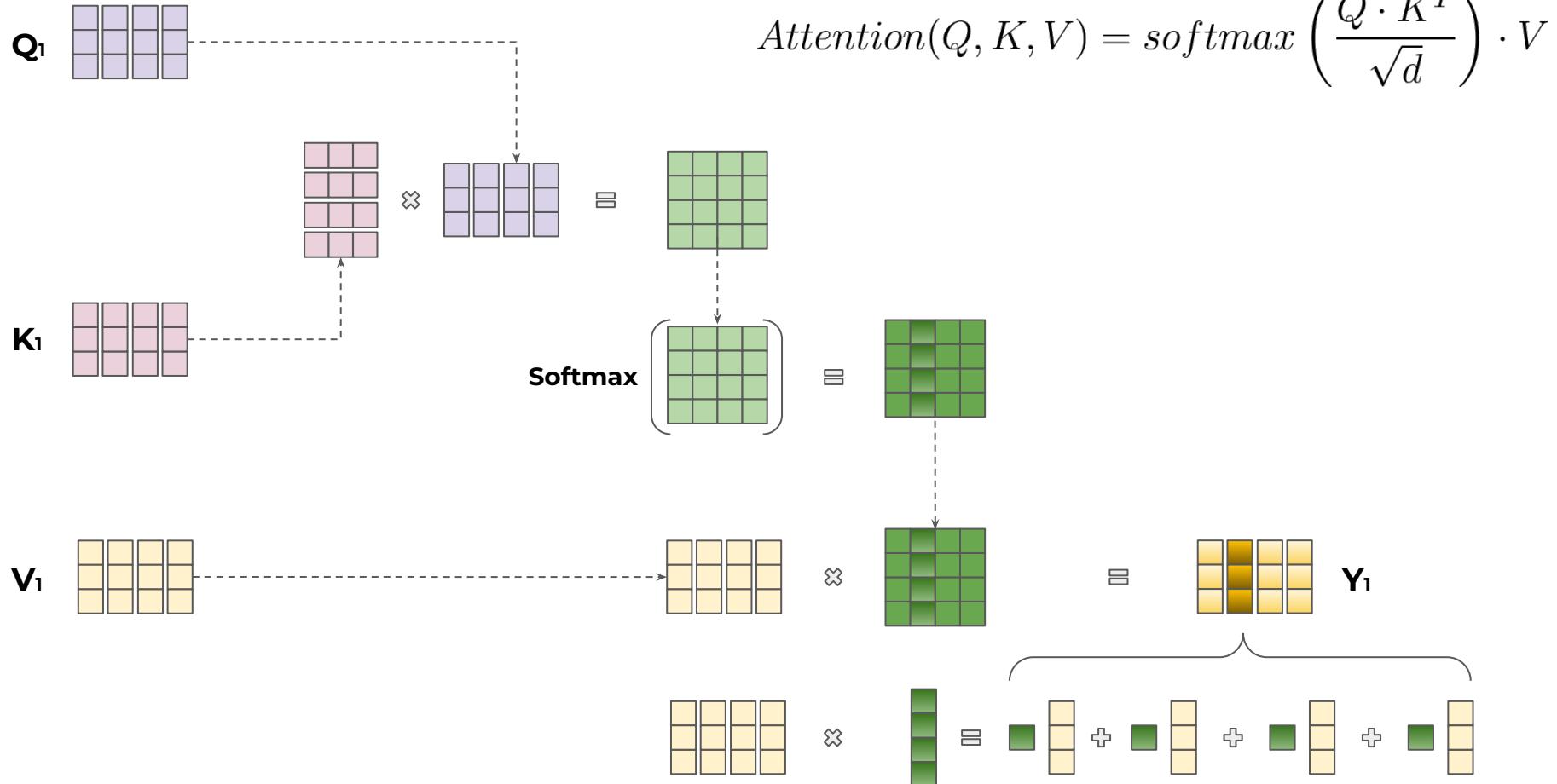
# Transformer [Multi-Head Attention (MHA)]



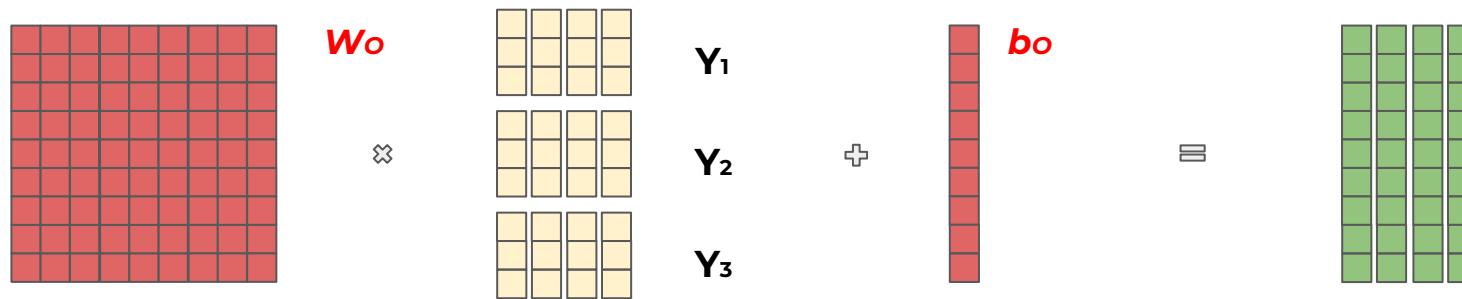
# Transformer [Single Head Attention]



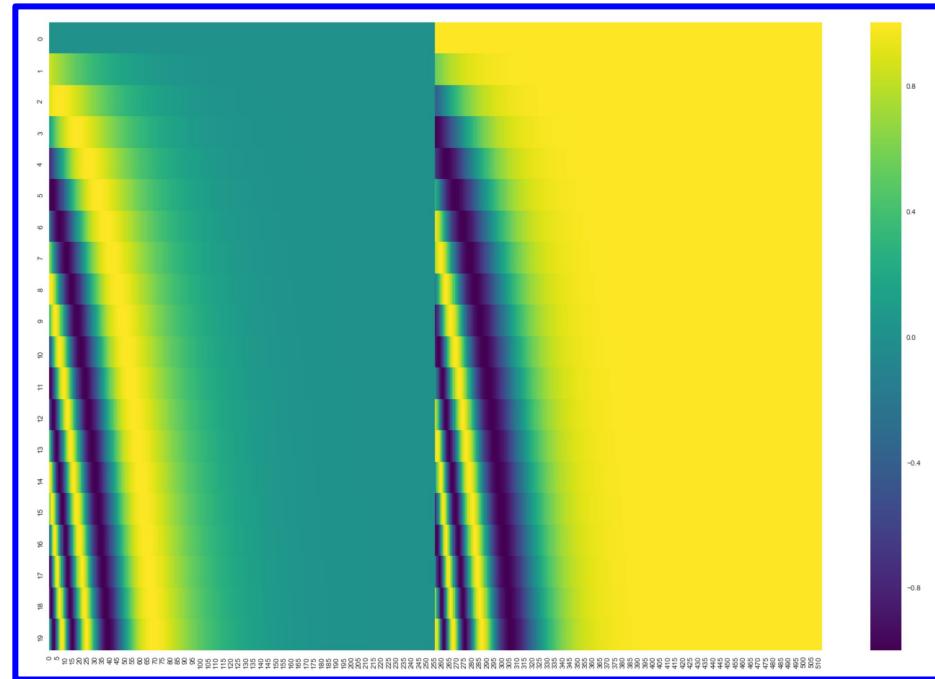
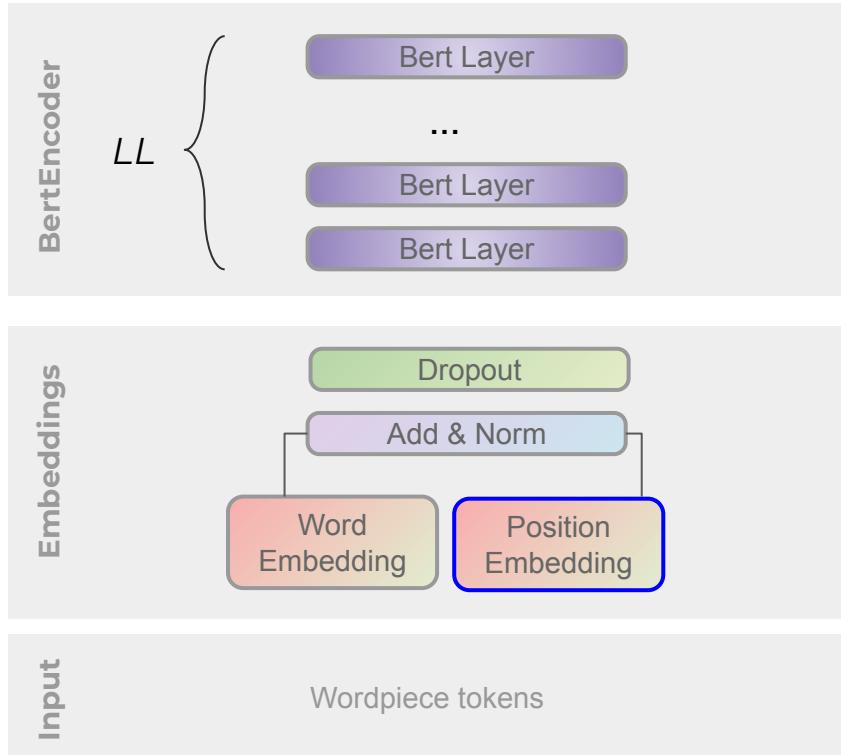
# Transformer [Single Head Attention]



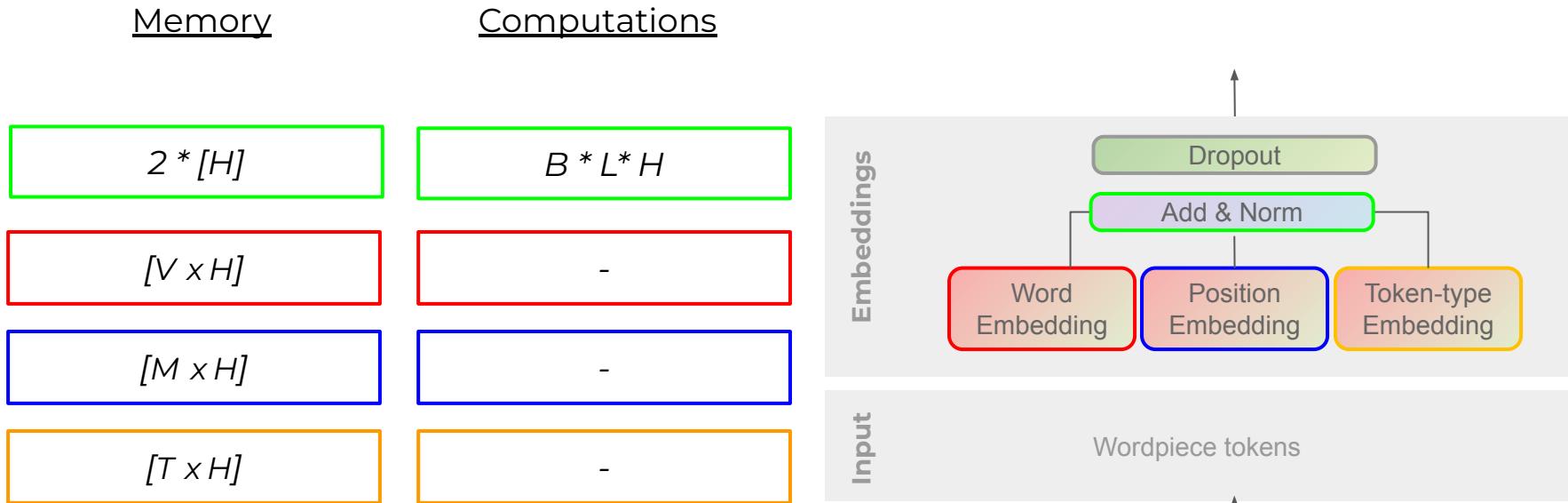
# Transformer [Multi-Head Attention (MHA)]



# Transformer [Encoder]



# Transformer [Embeddings]



B - batch\_size

H - hidden

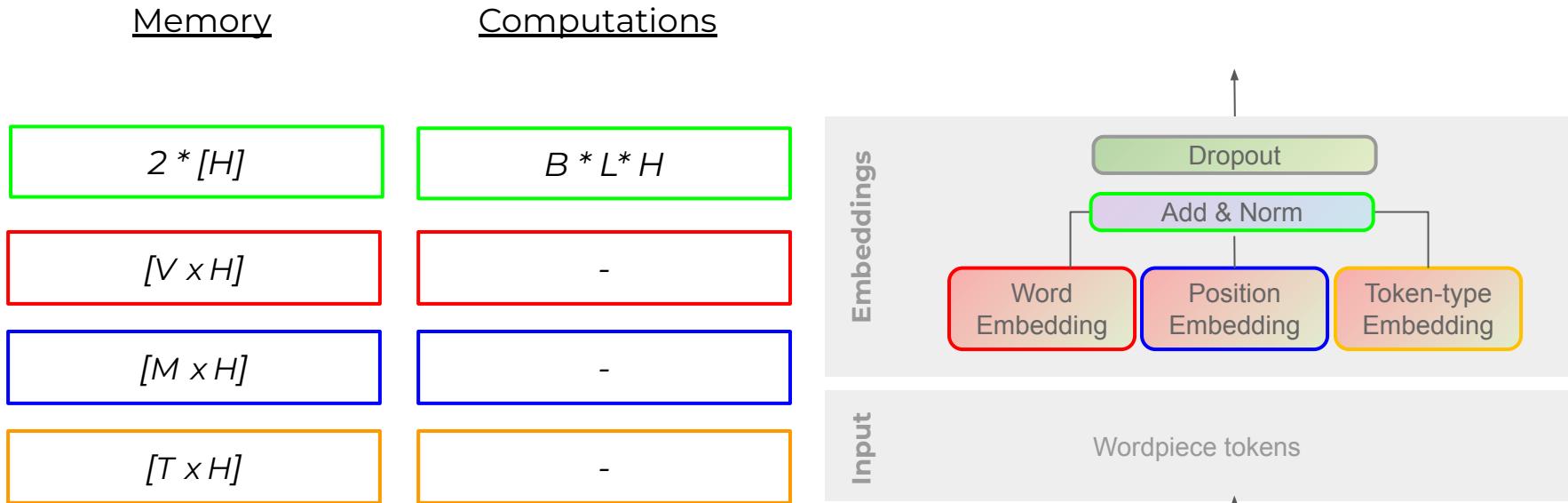
V - vocabulary size

M - max token length

T - number of types

[bert-base: 768]  
[bert-base: 50265]  
[bert-base: 514]  
[bert-base: 1]

# Transformer [Embeddings]



B - batch\_size

H - hidden

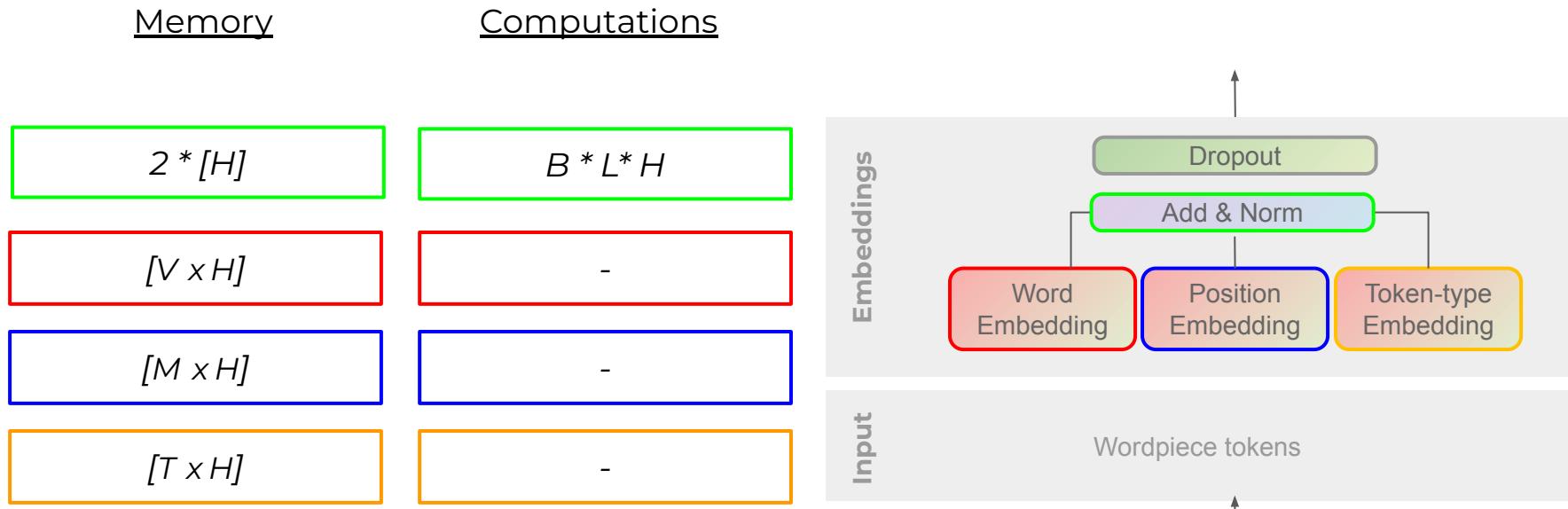
V - vocabulary size

M - max token length

T - number of types

[bert-base: 768]  
[bert-base: 50265]  
[bert-base: 514]  
[bert-base: 1]

# Transformer [Positional Encoding]



B - batch\_size

H - hidden

V - vocabulary size

M - max token length

T - number of types

[bert-base: 768]

[bert-base: 50265]

[bert-base: 514]

[bert-base: 1]

# Transformer [MHA]

Memory	Computations
$2 * [H]$	$B * L * H$
$[A * S + 1 \times H]$	$B * L * (A * S) * H$
-	$B * A * L * L * S * 2$
$3 * [H + 1 \times A * S]$	$B * L * H * (A * S) * 3$

B - batch\_size

L - batch\_length

H - hidden

A - num\_heads

S - head\_size (= H/A)

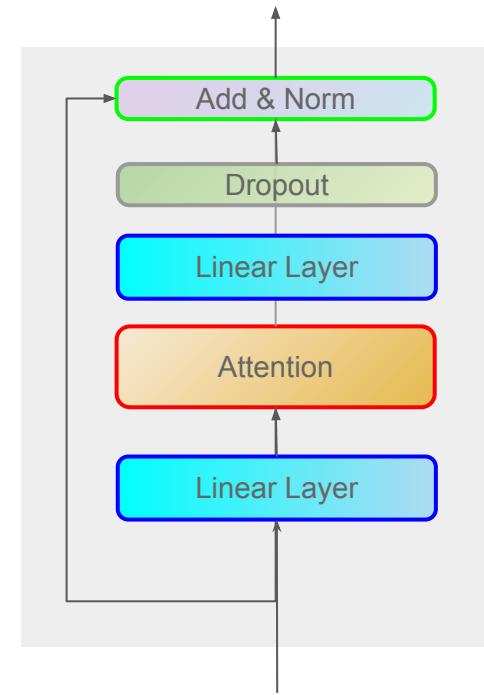
I - intermediate FC size

[bert-base: 768]

[bert-base: 12]

[bert-base: 64]

[bert-base: 3072]



# Transformer [FC]

Memory

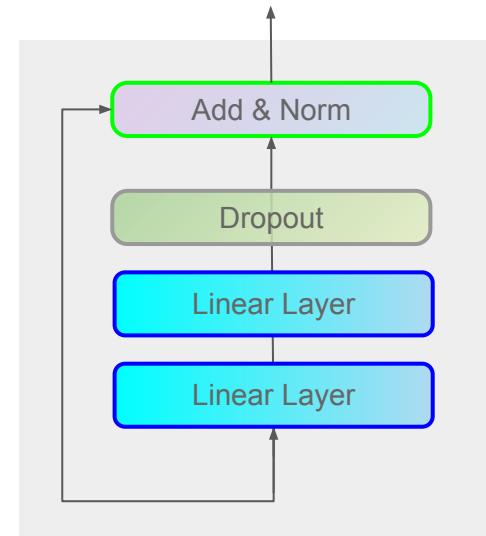
$$2 * [H]$$

Computations

$$B * L * H$$

$$[H + 1 \times I] + [I + 1 \times H]$$

$$2 * B * L * H * I$$



B - batch\_size  
L - batch\_length

H - hidden  
A - num\_heads  
S - head\_size (= H/A)  
I - intermediate FC size  
[bert-base: 768]  
[bert-base: 12]  
[bert-base: 64]  
[bert-base: 3072]

# Transformer [summary]

	<u>#params [M]</u>	<u>mocs [B]</u>
LL	FC-LayerNorm	$2 * H$
	FC-Out	$(1 + H) * I$
	FC-In	$(1 + I) * H$
	Atten-LayerNorm	$2 * H$
	Atten-LinearOutput	$(1 + H) * H$
	Atten-AttenScoreValue	0
	Atten-AttenScore	0
	Atten-LinearInput	$(1 + H) * H * 3$
	Embedding-LayerNorm	$2 * H$
	Embedding-Word	$V * H$

# Transformer [Memory & Runtime of BERT]

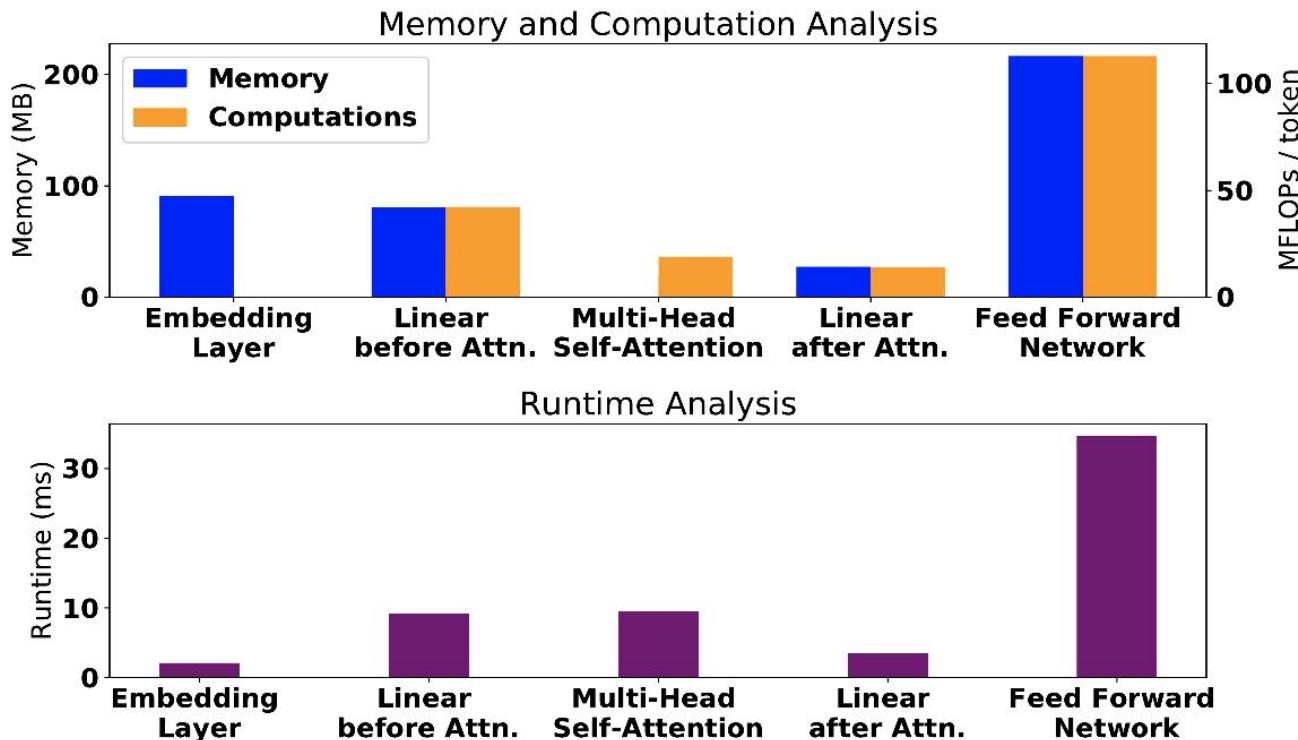
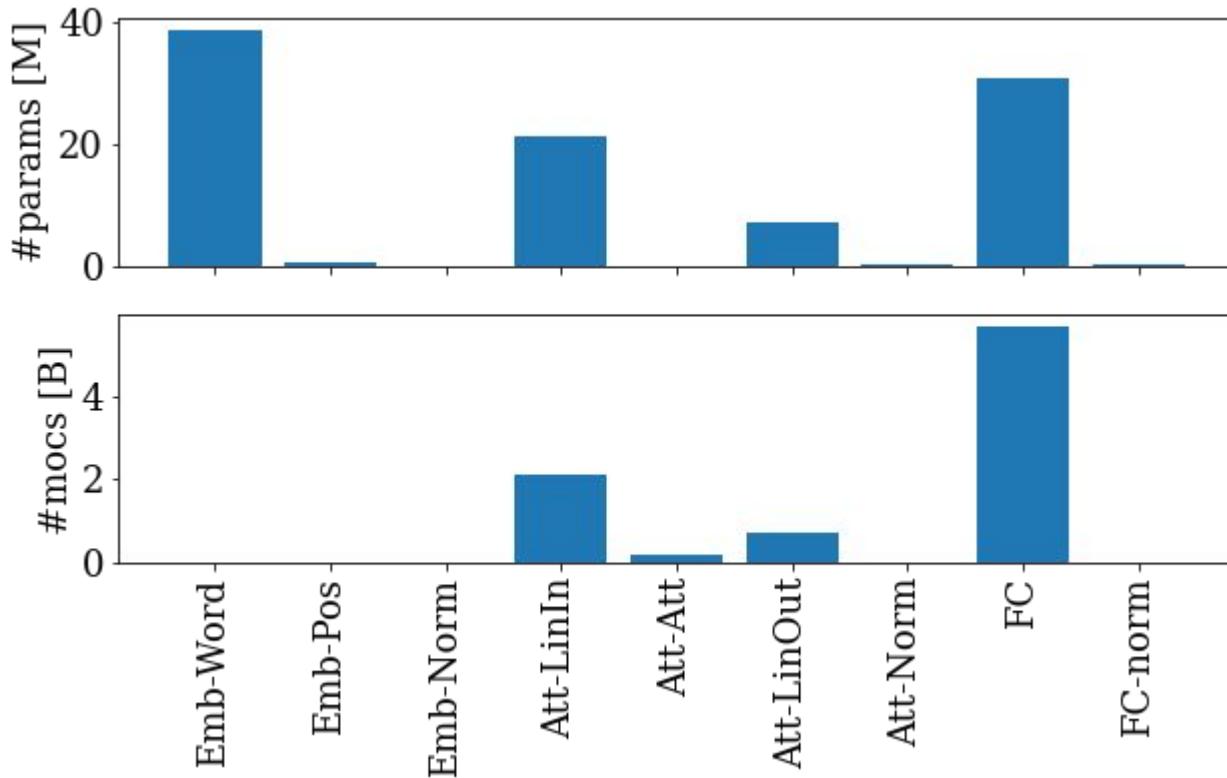
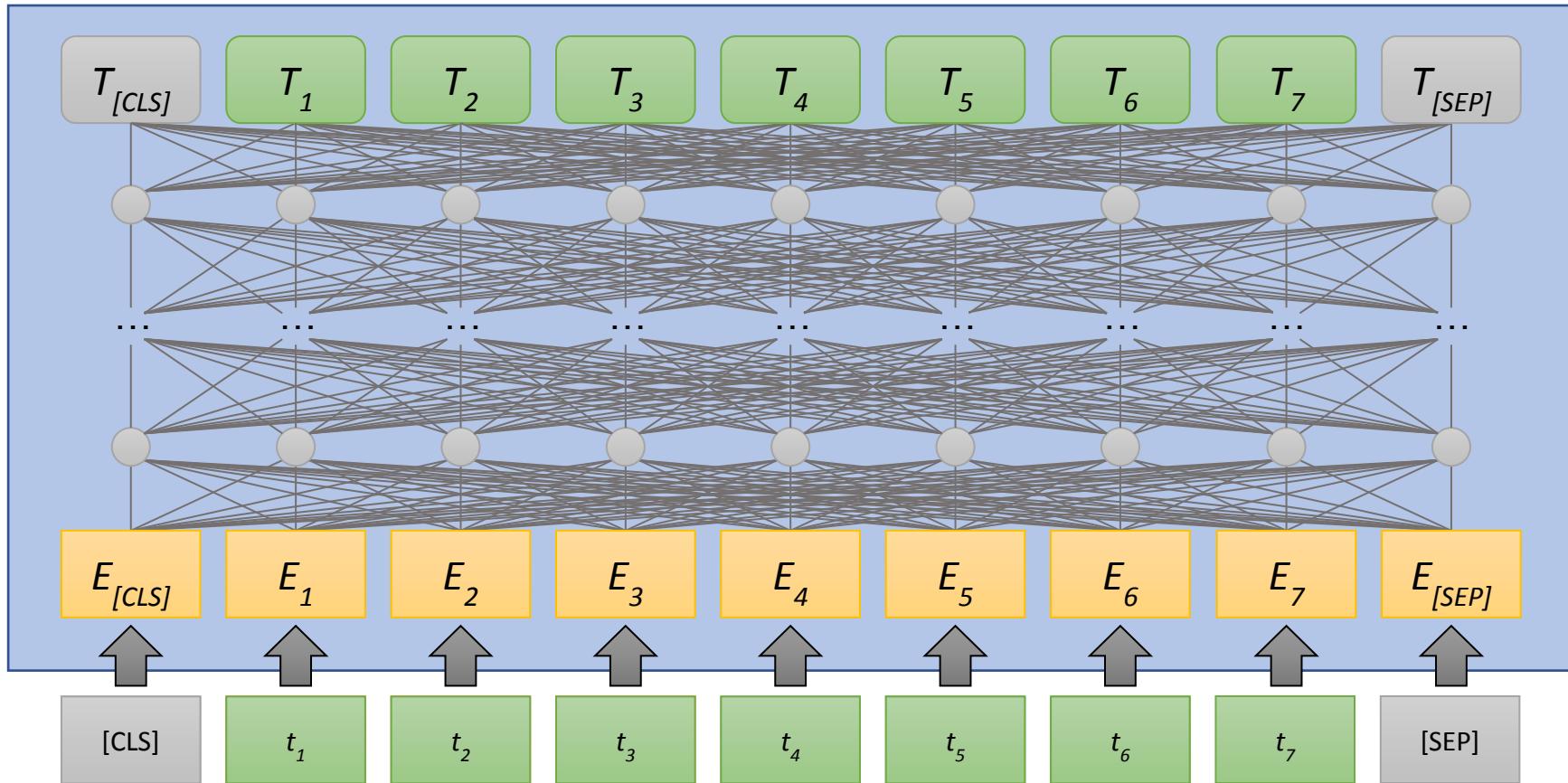


image source: [2002.11985](https://arxiv.org/abs/1609.08039)

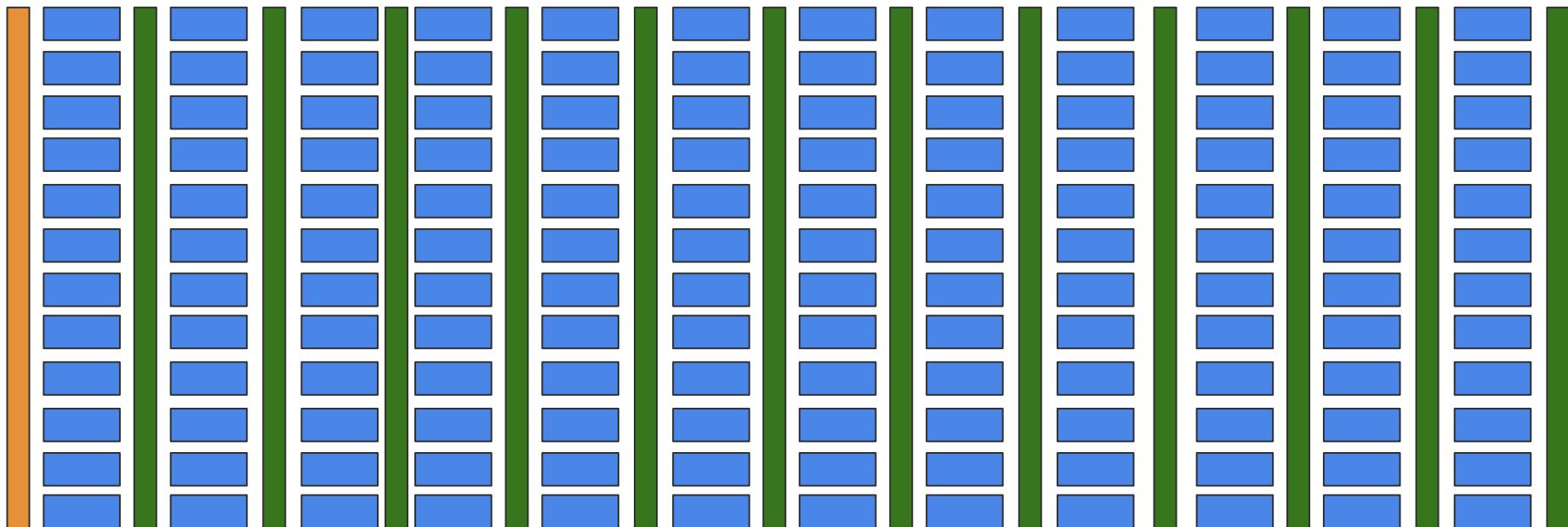
# Transformer [summary] [bert-Base] [B=1, L = 100]



# Transformer [another view]



# Transformer [another view]



$L = 12$  (bert-base),  $12 \times 12 = 144$  heads

# ChatGPT



Chris | SocialAudioGuy WE/ME @socialaudioguy · 19h

...

Here is a ChatGPT-created **poem** based on Twitter and **ChatGPT**. Very cool!



Sure! Here's a poem about Twitter in the form of Shakespearean verse, featuring social audio:

Twitter, oh Twitter, platform of wit  
Where thoughts and words are shared in bits  
A haven for the clever and the wise  
Where social audio doth oft arise

In 140 characters or less  
We share our thoughts with finesse  
But with the power of social audio  
We can share our thoughts with aural glow

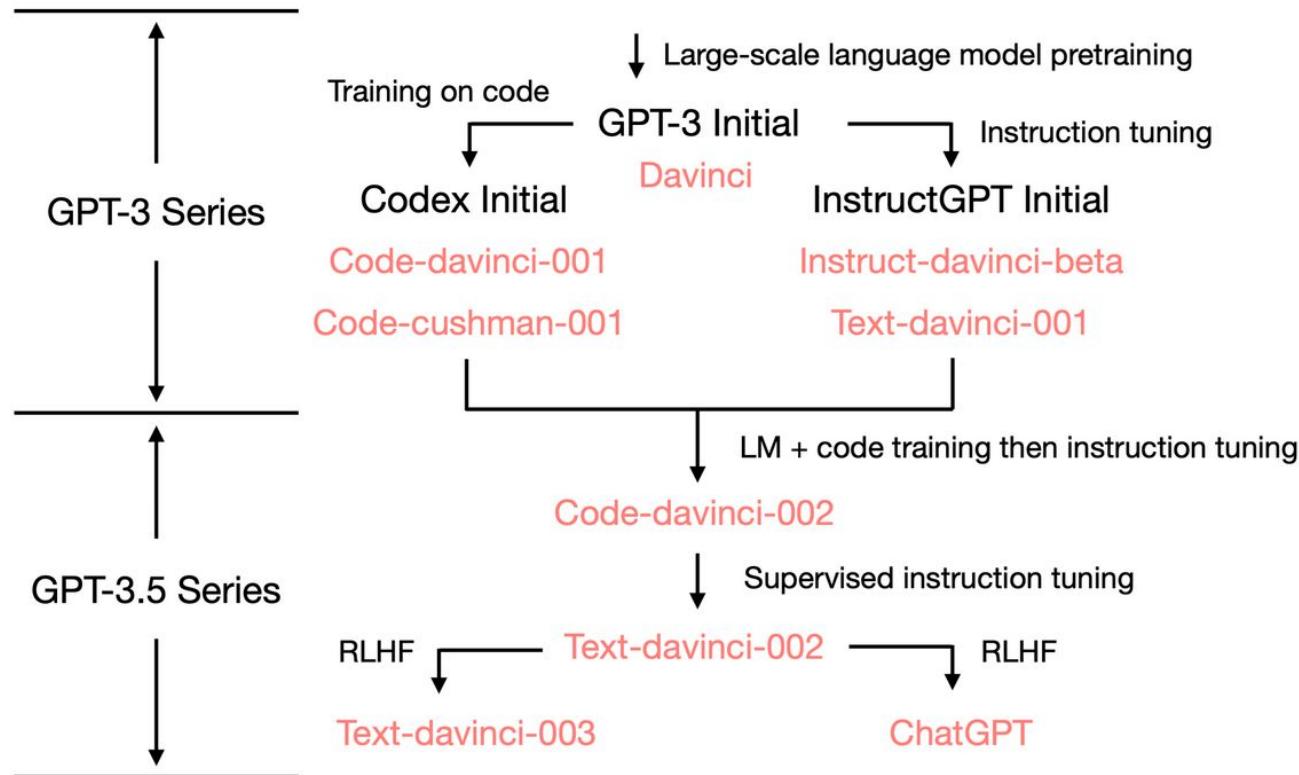
Listeners tune in, with ear so keen  
To hear the musings of the Twitter queen  
Her voice doth ring with clarity and grace  
As she shares her thoughts with the human race

Twitter, oh Twitter, thou art a delight  
A place to share, to listen, and to write  
With social audio, the possibilities abound  
To share our thoughts with all around

So let us tweet and listen with glee  
On this platform of wit and levity  
Twitter, oh Twitter, thou art our delight  
Forevermore, in day and night

[ChatGPT: Optimizing Language Models for Dialogue](#)

# GPT-\*

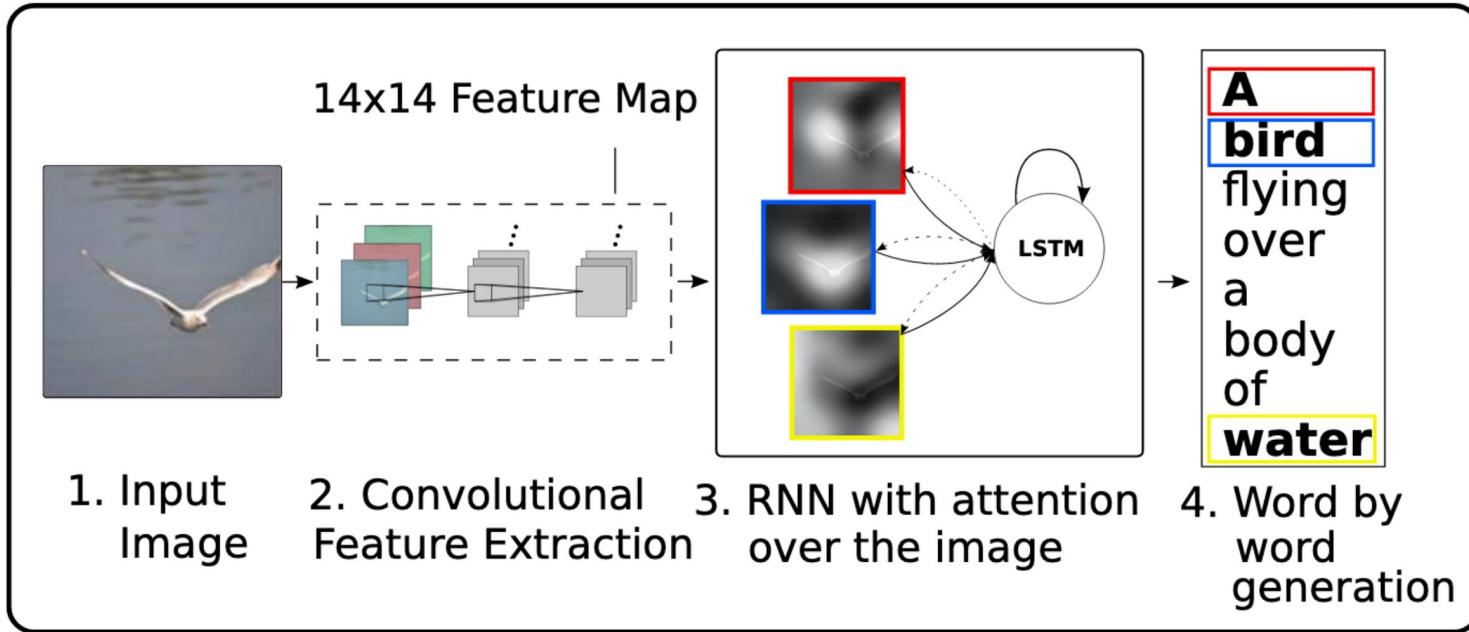


How does GPT Obtain its Ability? Tracing Emergent Abilities of Language Models to their Sources

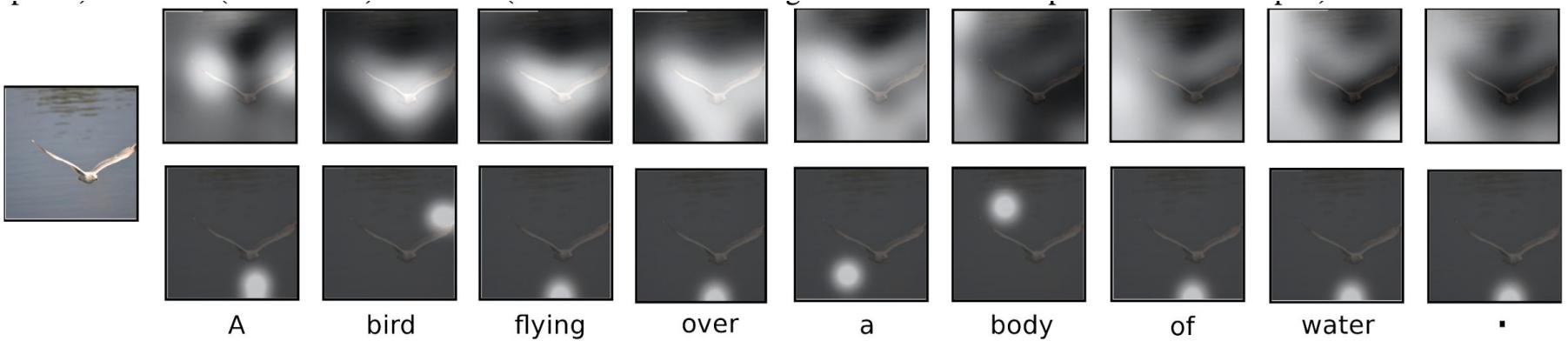
# Content of today's lecture

- Attention Intro
  - Brief History
  - Seq2seq Attention
  - Self-Attention
  - Transformer, BERT
- Attention in CV
  - SE
  - CBAM
  - ViT
  - latest developments

# Attention [seq2seq, visual]



# Attention [seq2seq, visual]



1502.03044

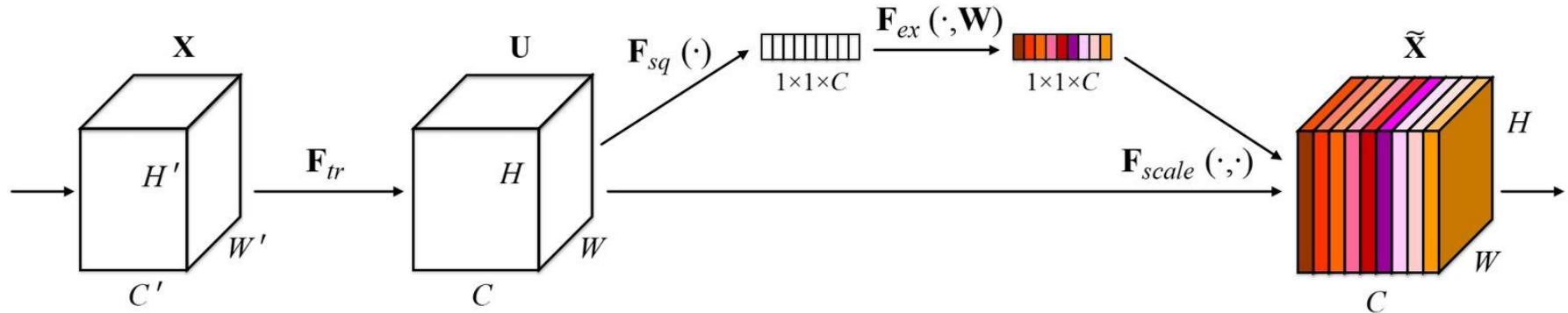
# Attention [seq2seq, visual]



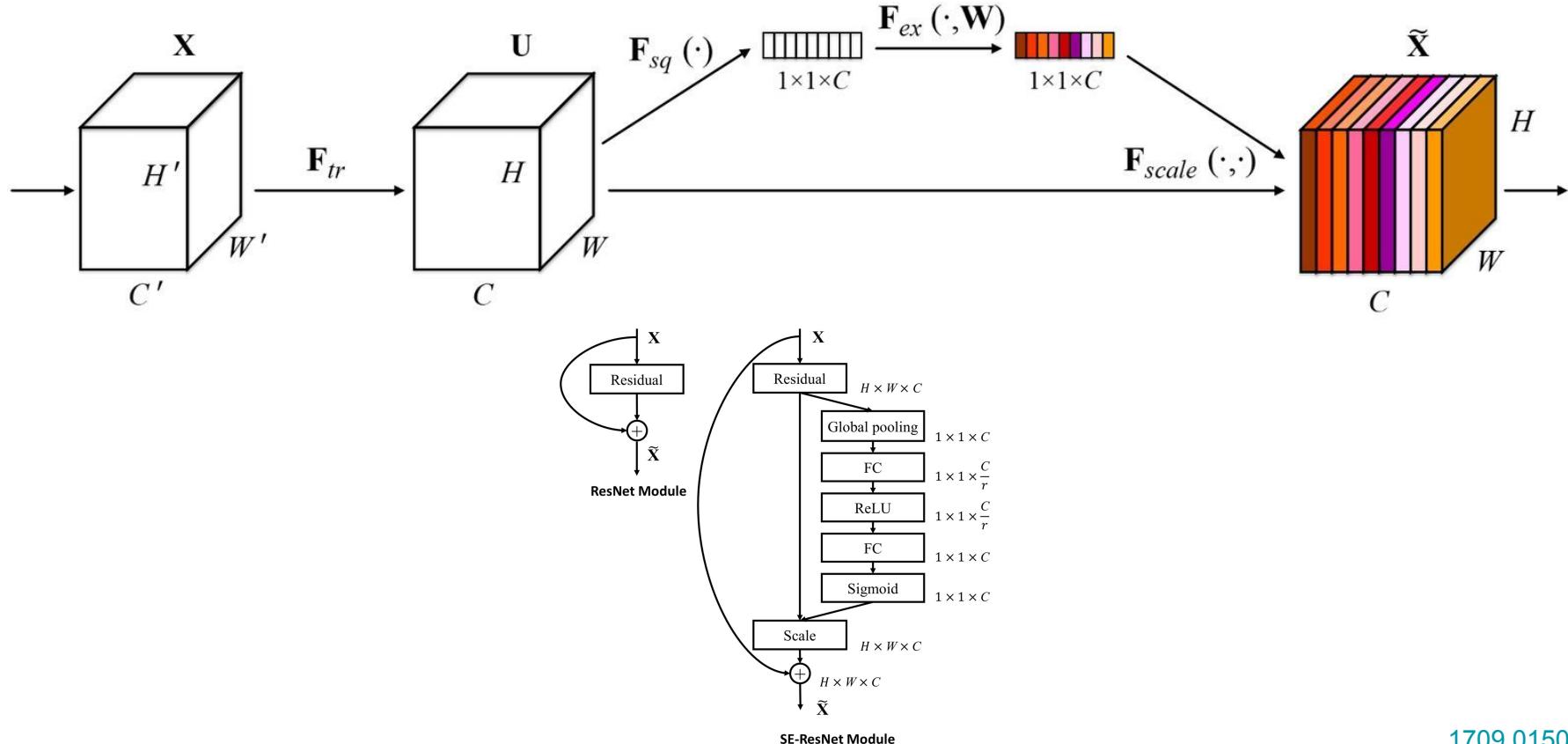
A woman is throwing a frisbee in a park.

1502.03044

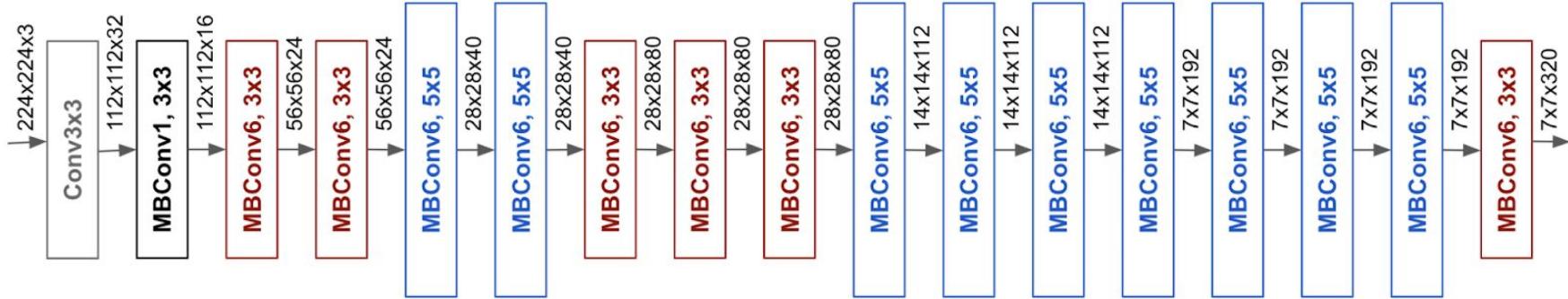
# Squeeze-and-Excitation (SE) Block



# Squeeze-and-Excitation (SE) Block



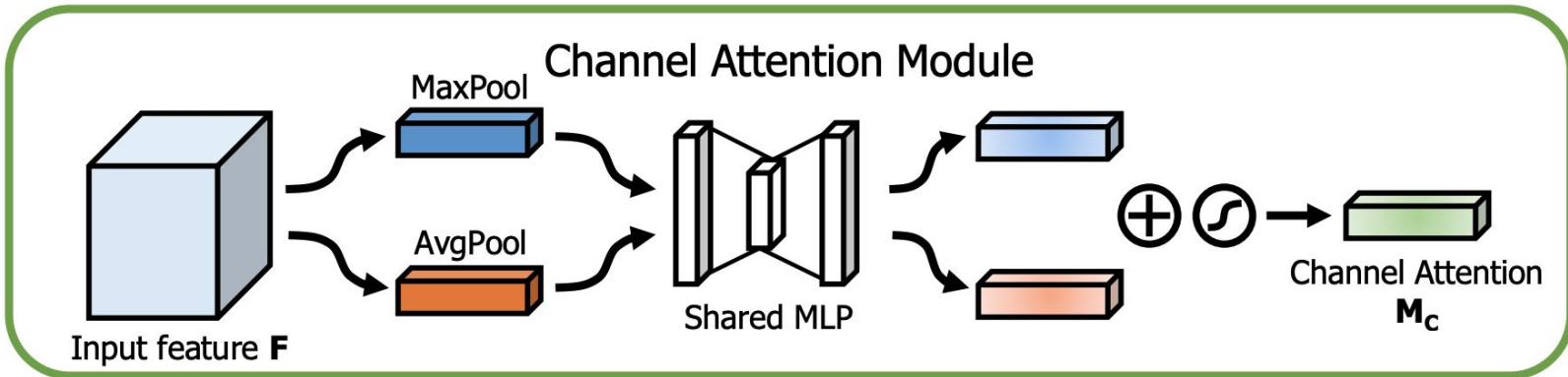
# Squeeze-and-Excitation (SE) Block



**EfficientNet**



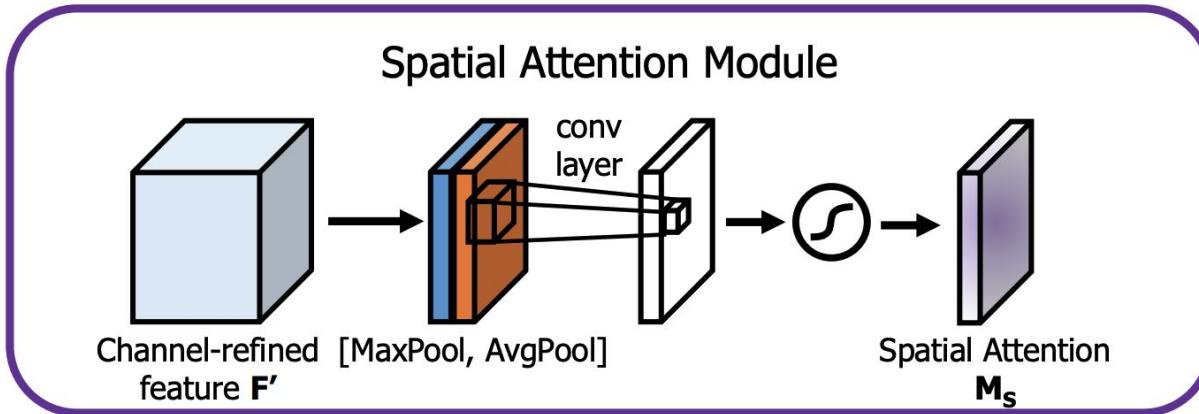
# CBAM (Channel Attention Module)



$$\mathbf{M}_c(\mathbf{F}) = \sigma(\text{MLP}(\text{AvgPool}(\mathbf{F})) + \text{MLP}(\text{MaxPool}(\mathbf{F})))$$

$$\mathbf{M}_c(\mathbf{F}) = \sigma(\mathbf{W}_1(\mathbf{W}_0(\mathbf{F}_{avg}^c)) + \mathbf{W}_1(\mathbf{W}_0(\mathbf{F}_{max}^c)))$$

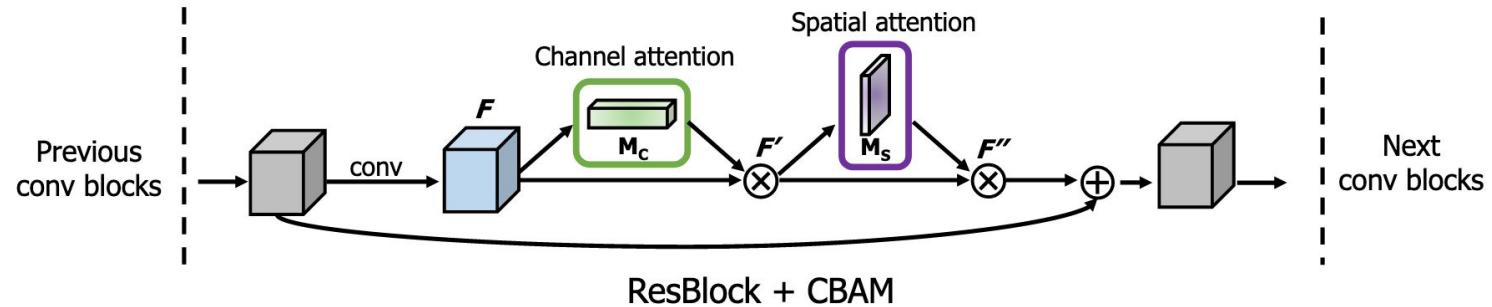
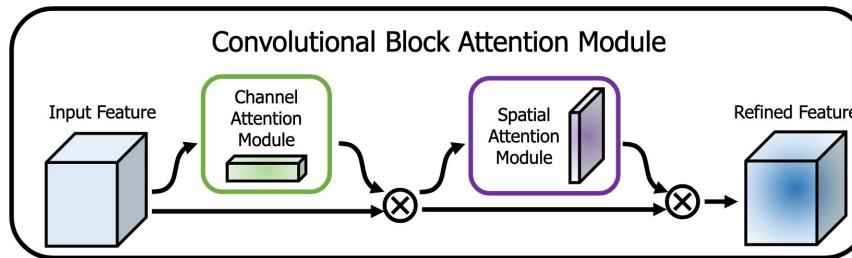
# CBAM (Spatial Attention Module)



$$\mathbf{M}_s(F) = \sigma(f^{7x7}([\text{AvgPool}(F); \text{MaxPool}(F)]))$$

$$\mathbf{M}_s(F) = \sigma(f^{7x7}([\mathbf{F}_{avg}^s; \mathbf{F}_{max}^s]))$$

# CBAM



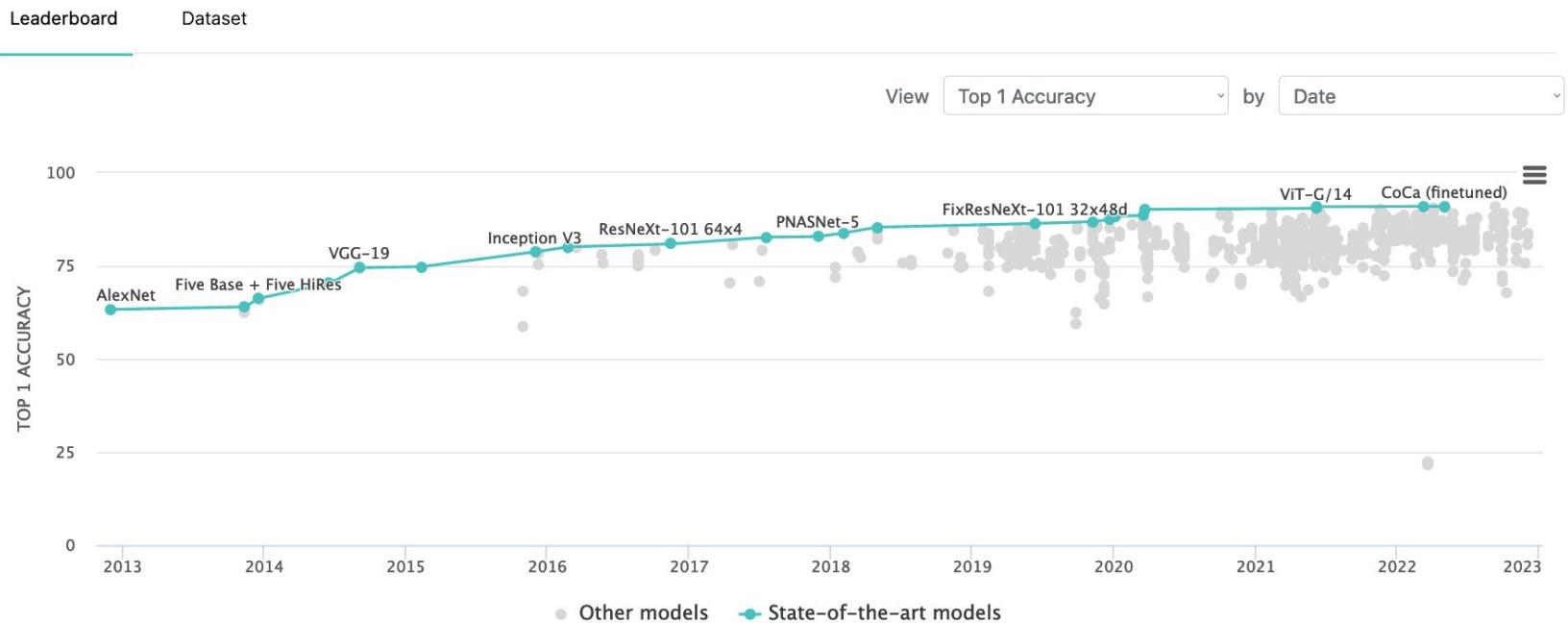
CBAM

	Tailed frog	Toilet tissue	Loudspeaker	Spider web	American egret	Tank	Seawall	Space heater
Input image								
ResNet50								
	P=0.80736	P=0.11857	P=0.65681	P=0.22357	P=0.64185	P=0.14763	P=0.92236	P=0.01176
ResNet50 + SE								
	P=0.87240	P=0.14643	P=0.77550	P=0.25093	P=0.70827	P=0.15367	P=0.97166	P=0.26611
ResNet50 + CBAM								
	P = 0.96340	P = 0.19994	P = 0.93707	P = 0.35248	P = 0.87490	P = 0.53005	P = 0.99085	P = 0.59662

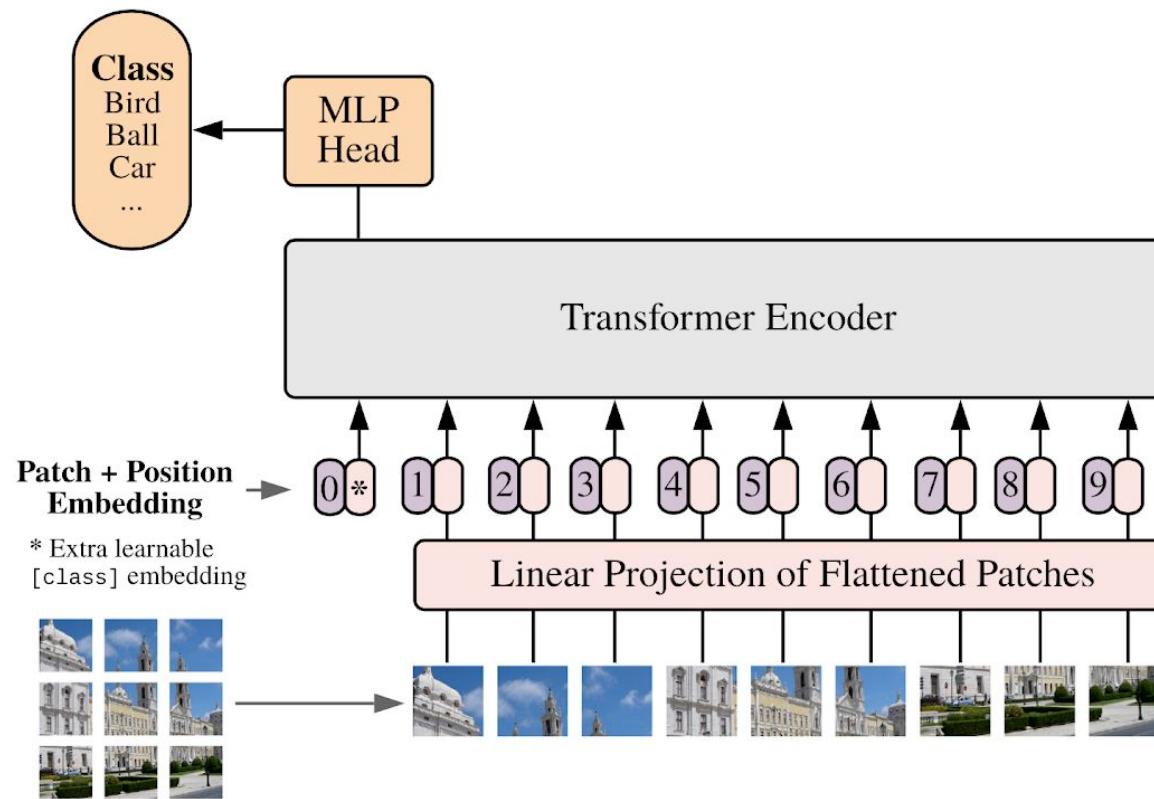
# Content of today's lecture

- Attention Intro
  - Brief History
  - Seq2seq Attention
  - Self-Attention
  - Transformer, BERT
- Attention in CV
  - SE
  - CBAM
  - ViT
  - latest developments

# Image Classification on ImageNet



# ViT

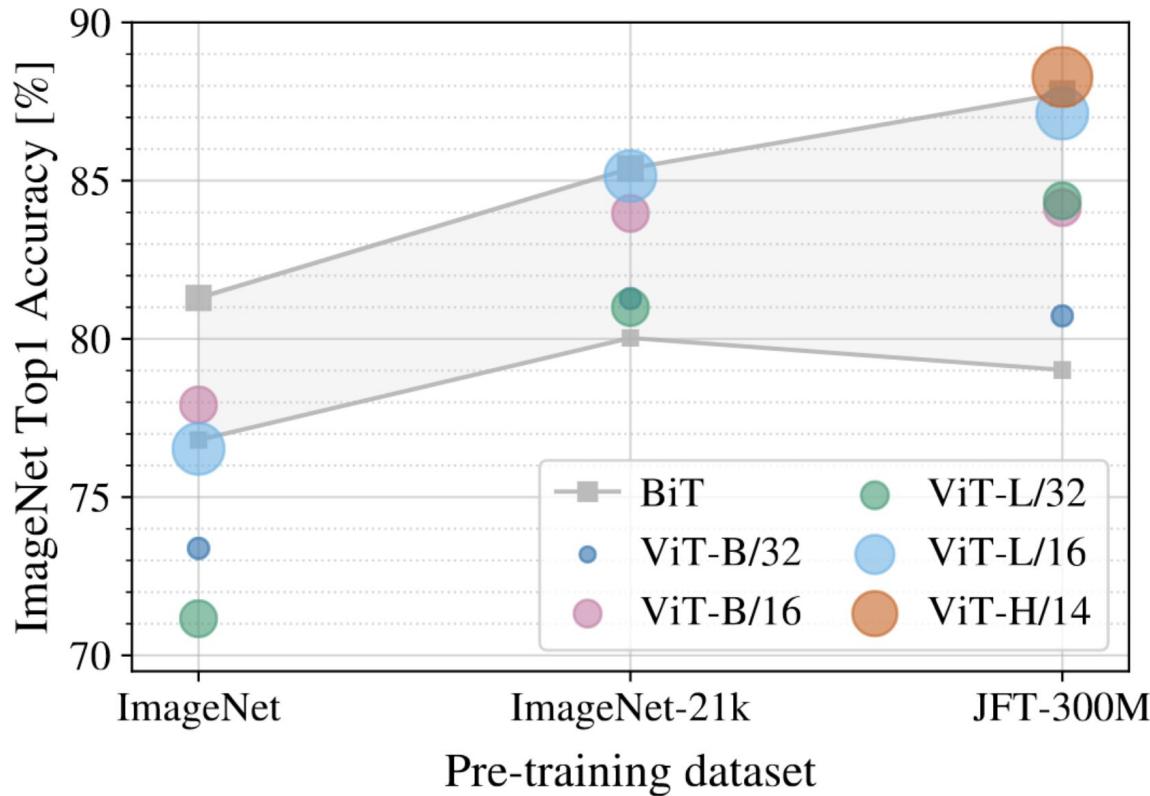


Model	Layers	Hidden size $D$	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

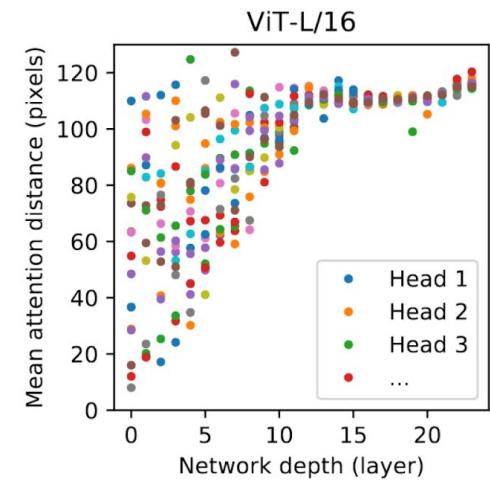
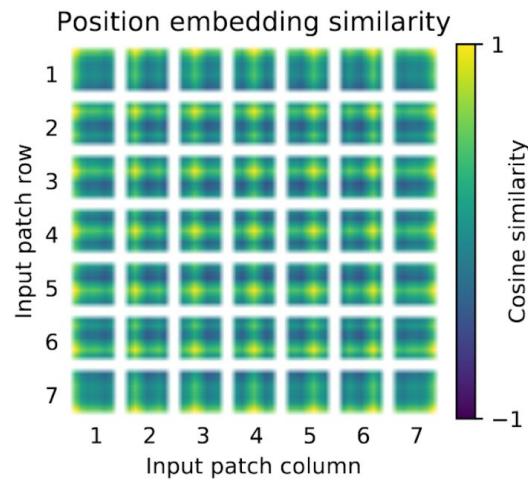
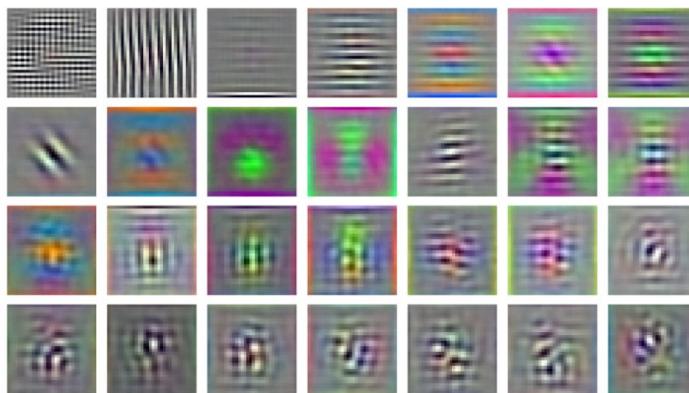
	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21K (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	<b>88.55</b> $\pm$ 0.04	87.76 $\pm$ 0.03	85.30 $\pm$ 0.02	87.54 $\pm$ 0.02	88.4/88.5*
ImageNet ReaL	<b>90.72</b> $\pm$ 0.05	90.54 $\pm$ 0.03	88.62 $\pm$ 0.05	90.54	90.55
CIFAR-10	<b>99.50</b> $\pm$ 0.06	99.42 $\pm$ 0.03	99.15 $\pm$ 0.03	99.37 $\pm$ 0.06	—
CIFAR-100	<b>94.55</b> $\pm$ 0.04	93.90 $\pm$ 0.05	93.25 $\pm$ 0.05	93.51 $\pm$ 0.08	—
Oxford-IIIT Pets	<b>97.56</b> $\pm$ 0.03	97.32 $\pm$ 0.11	94.67 $\pm$ 0.15	96.62 $\pm$ 0.23	—
Oxford Flowers-102	99.68 $\pm$ 0.02	<b>99.74</b> $\pm$ 0.00	99.61 $\pm$ 0.02	99.63 $\pm$ 0.03	—
VTAB (19 tasks)	<b>77.63</b> $\pm$ 0.23	76.28 $\pm$ 0.46	72.72 $\pm$ 0.21	76.29 $\pm$ 1.70	—
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

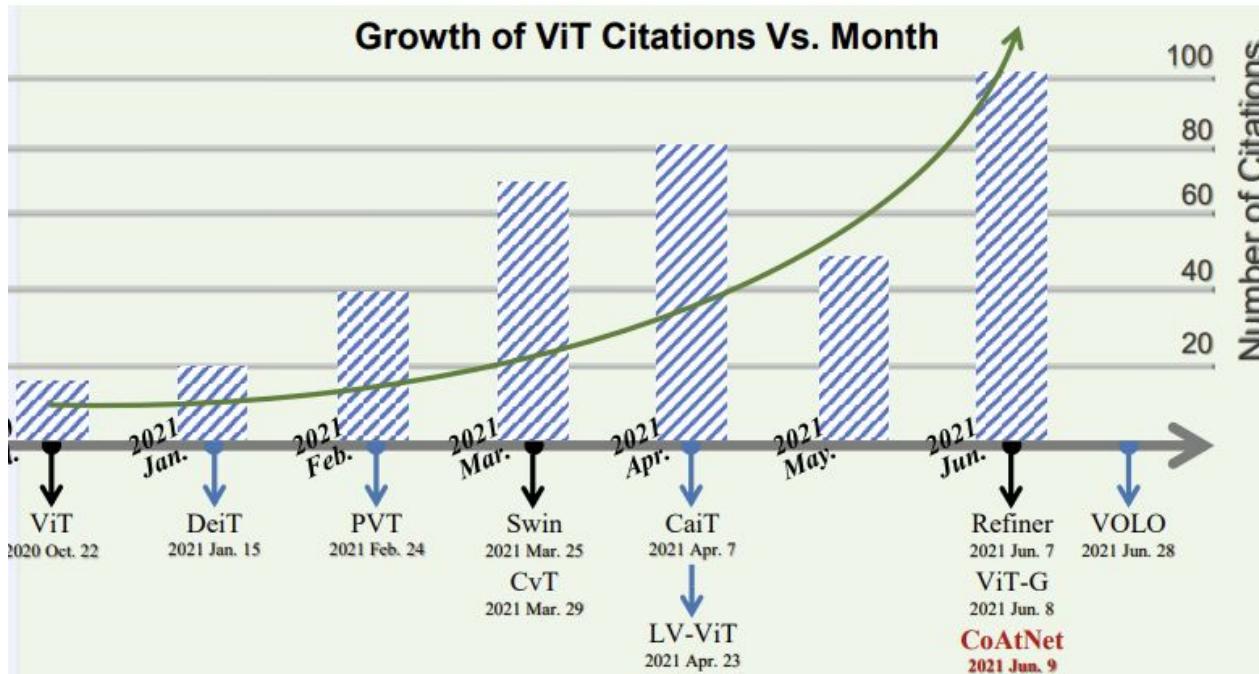
# ViT

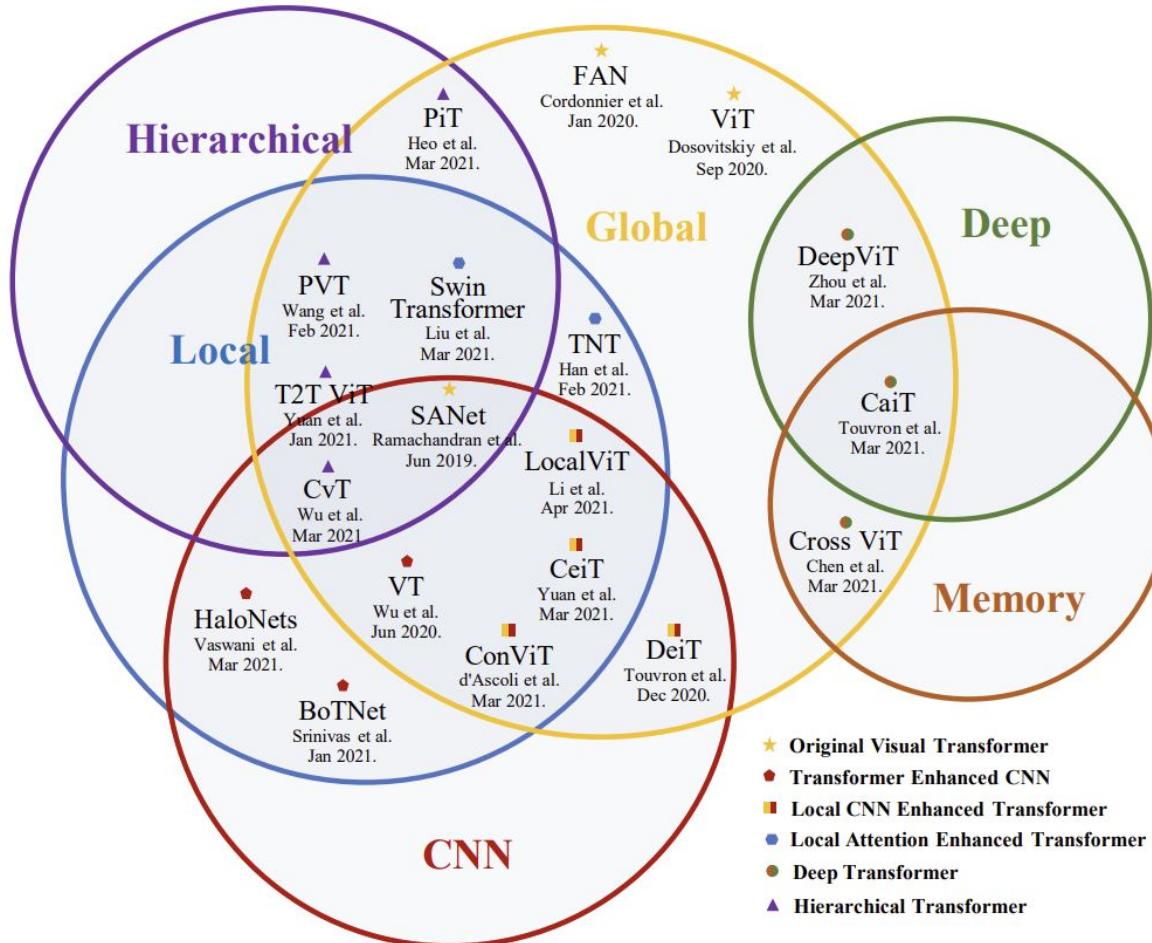


# ViT

RGB embedding filters  
(first 28 principal components)







# Content of today's lecture

- Attention Intro
  - Brief History
  - Seq2seq Attention
  - Self-Attention
  - Transformer, BERT
- Attention in CV
  - SE
  - CBAM
  - ViT
  - latest developments

# Summary

- Pay attention to attention :)  
Currently one of the main important paradigms in Deep Learning
- Self-attention dominates NLP and recently came to prominence in CV