



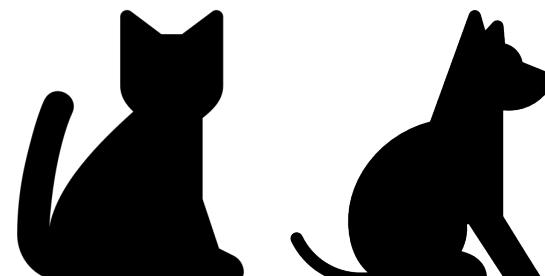
Paradigm Shift in Computer Vision and Beyond: Transformer and Attention

Tsung-Ming Tai, NVIDIA AI Technology Center (NVAITC), NVIDIA; Computer Science PhD student of UNIBZ.

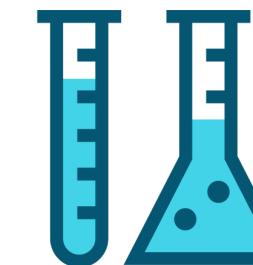
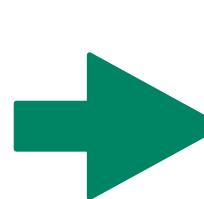
Deep Learning

Minimize the efforts in feature engineering

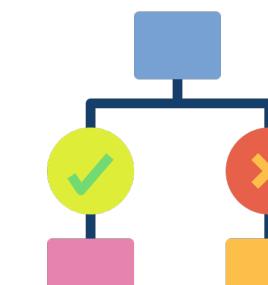
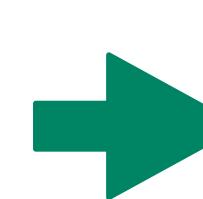
Machine Learning



Input



Feature Engineer



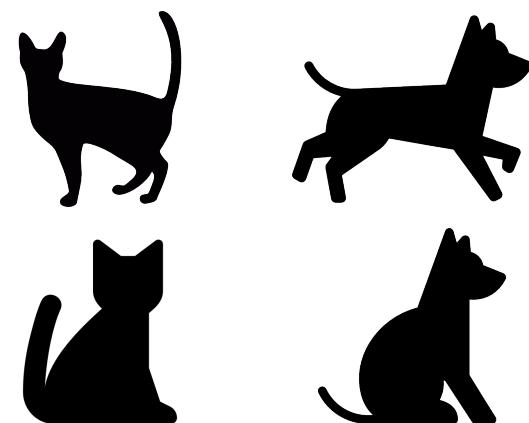
Classifier



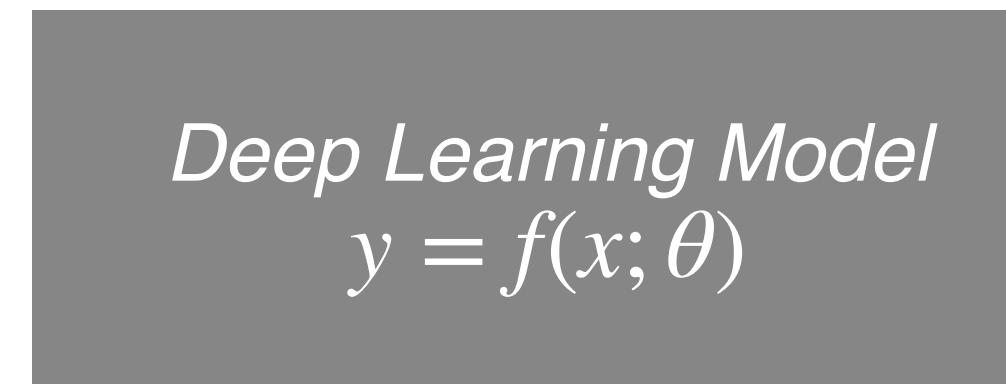
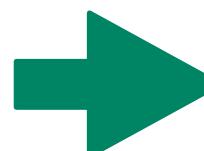
Cat or Dog?

Result

Deep Learning (with sufficient data)



Input



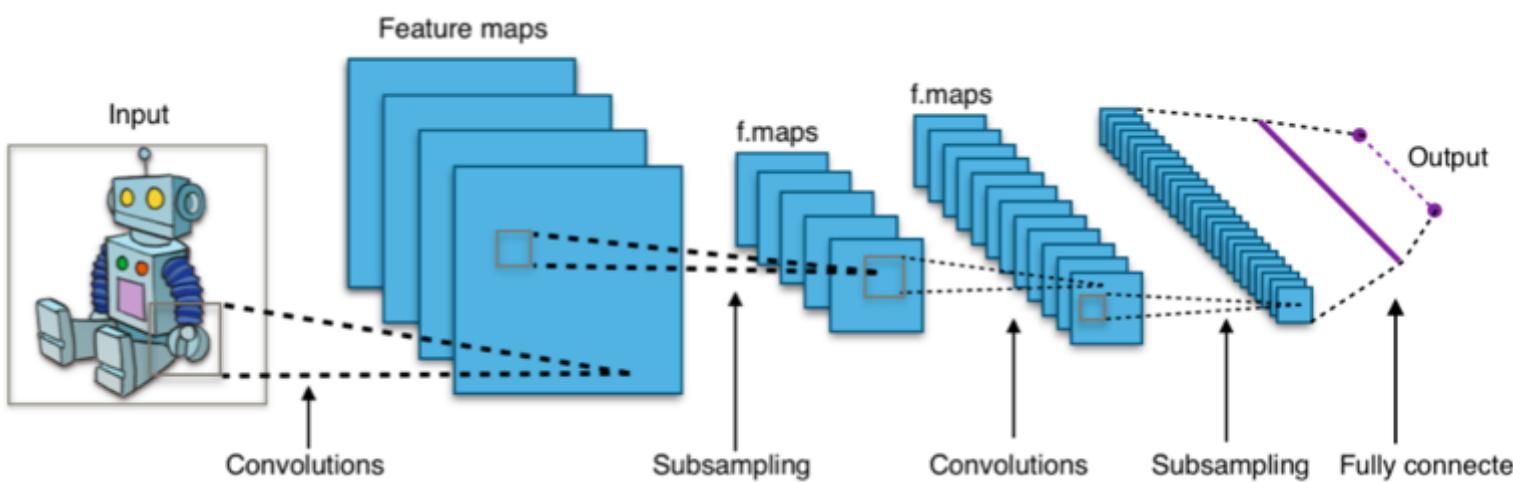
Cat or Dog?

Result

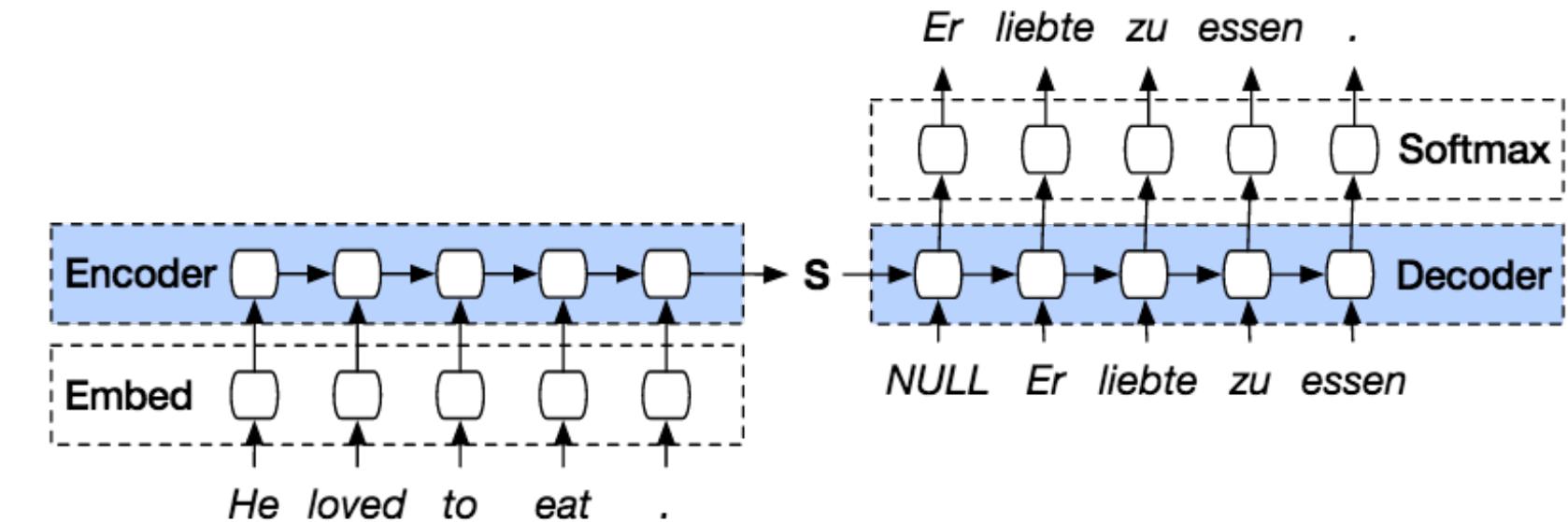
Building Deep Learning Model

Earlier than 2018

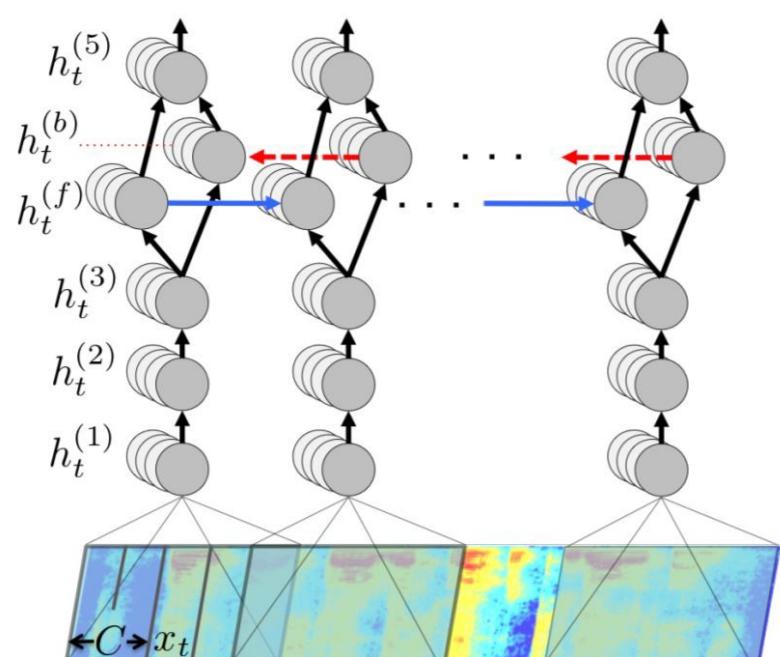
Computer Vision



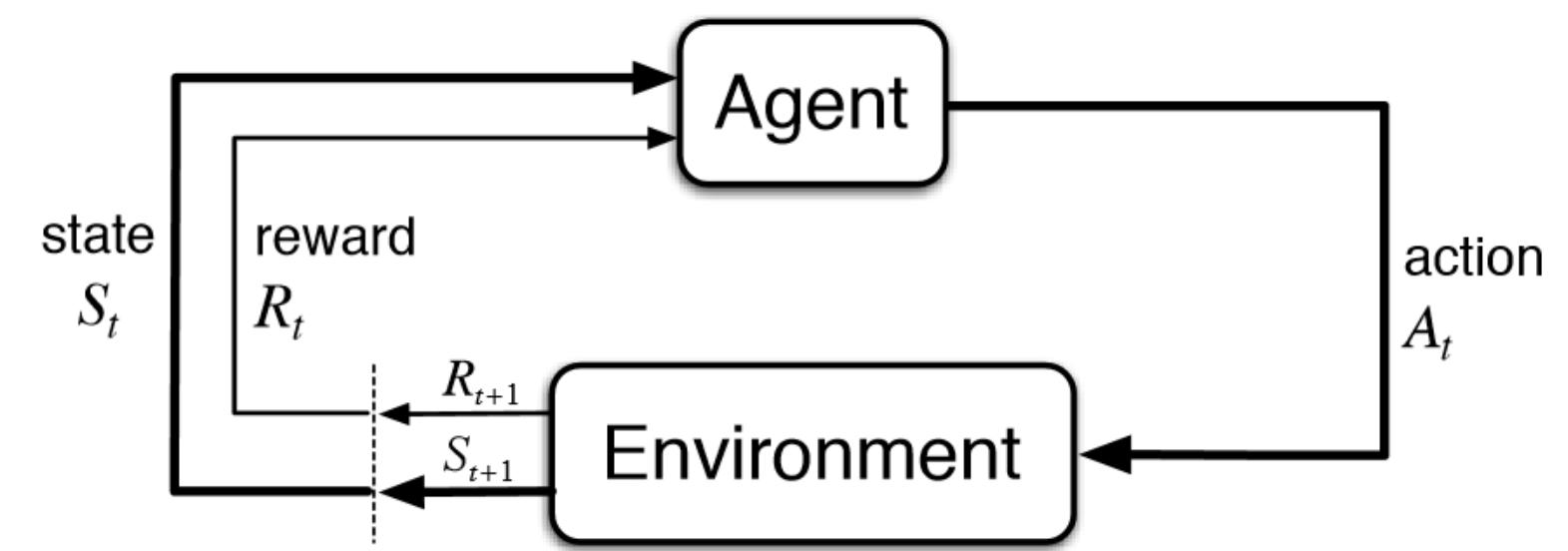
Natural Language Processing



Speech Recognition

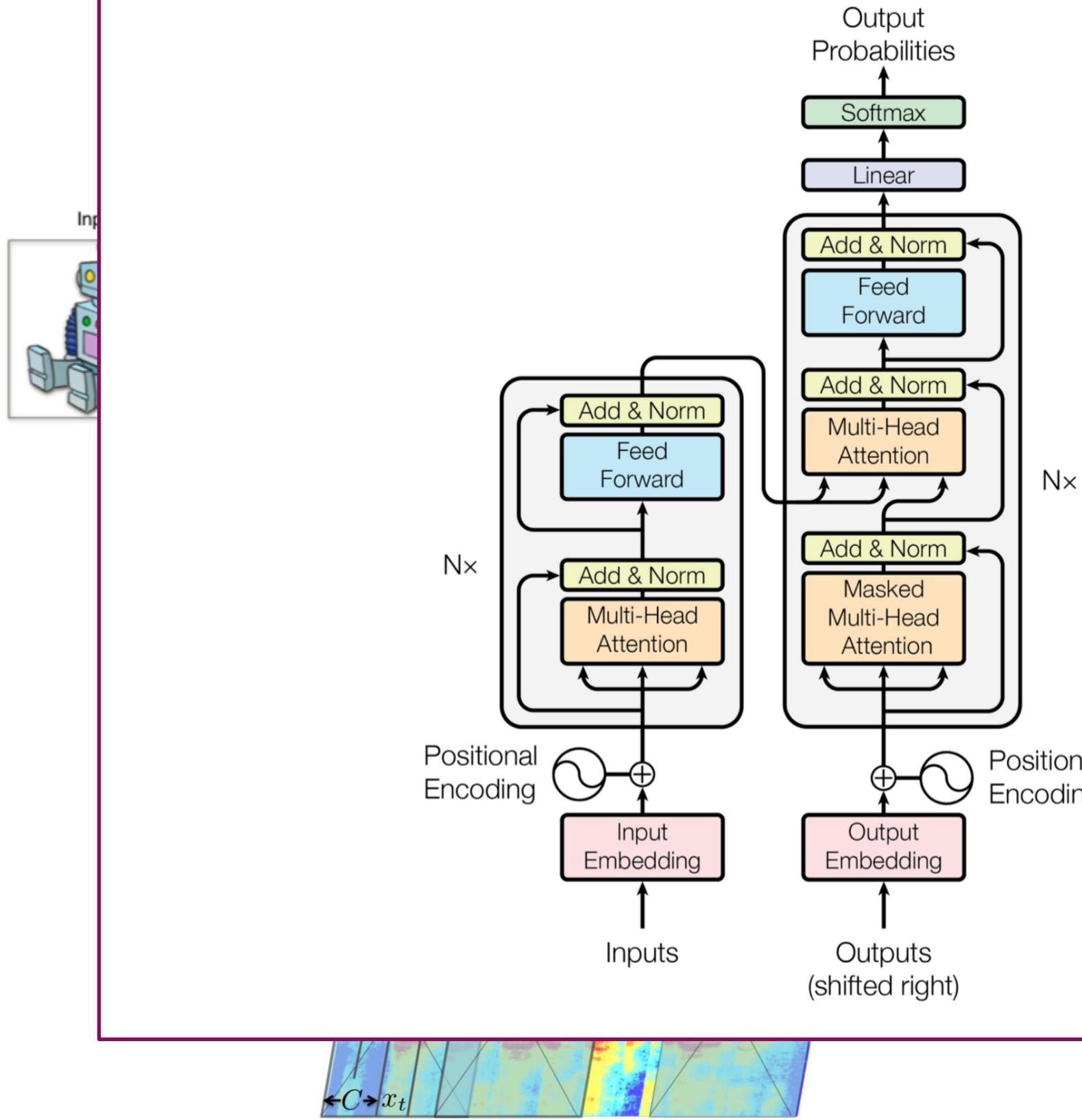


Reinforcement Learning



Building Deep Learning Model

Recently,



**"Attention Is
All You Need"**

Transformer and Attention

Attention Learns Alignment

Neural Machine Translation by Jointly Learning to Align and Translate
Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, 2014

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

$$c_i = \sum_{j=1}^T a_{ij} h_j$$

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

$$e_{ij} = a(s_{i-1}, h_j)$$

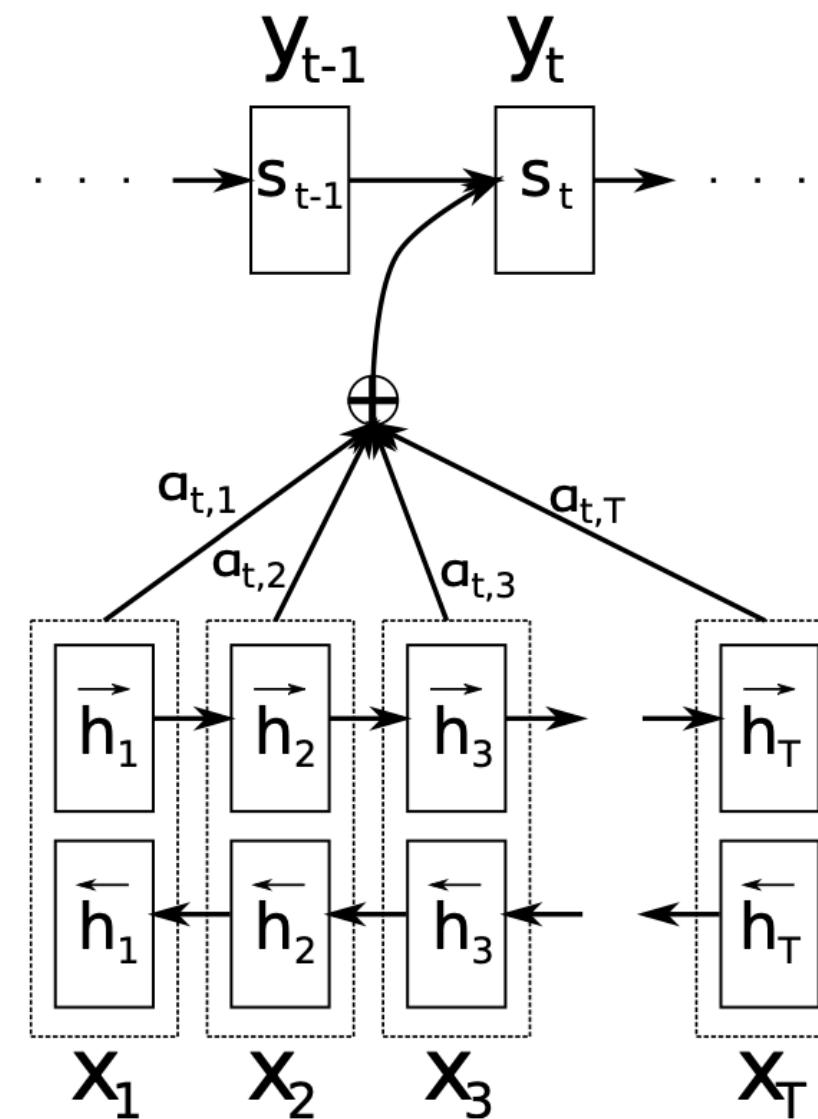
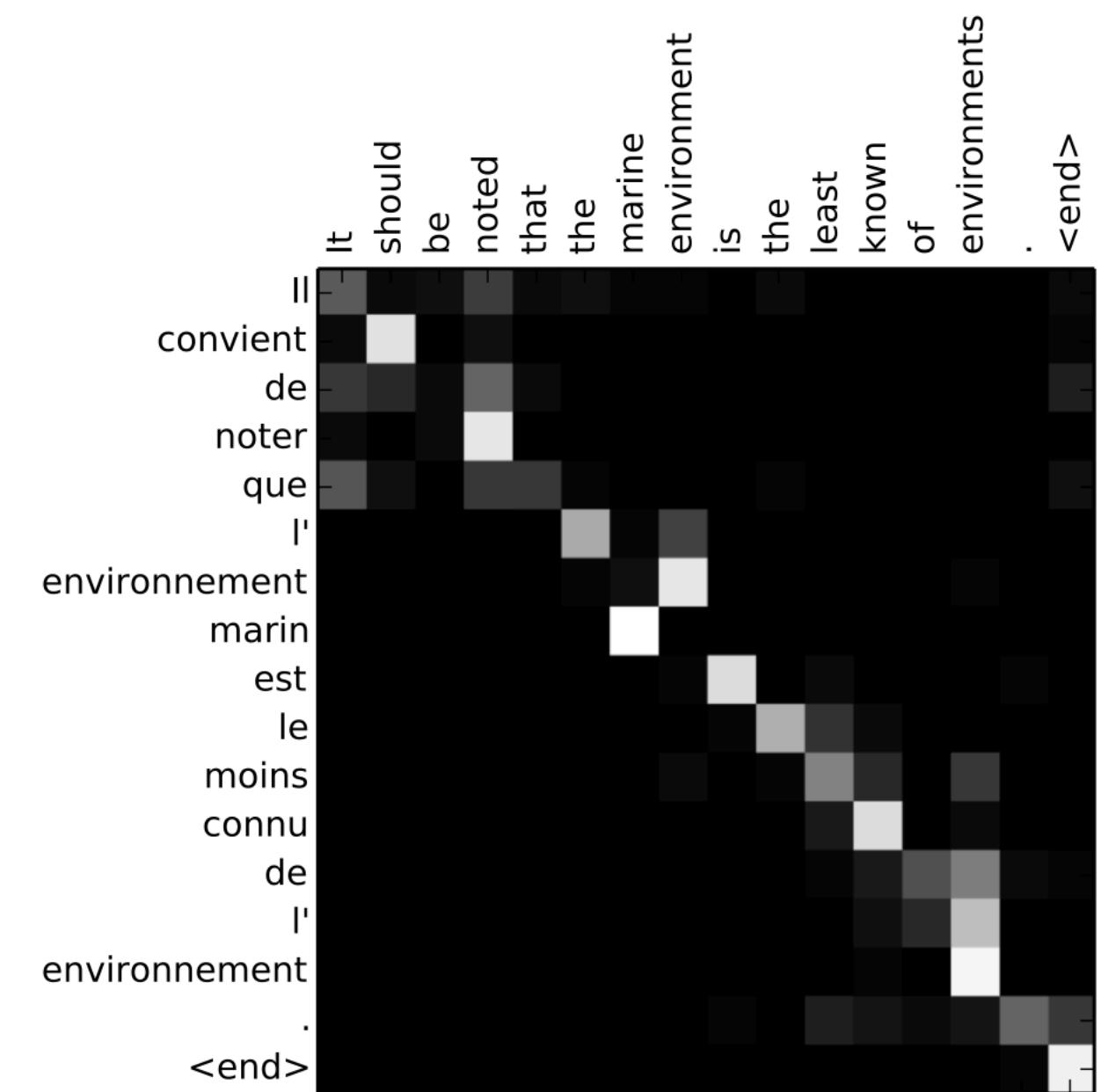


Figure 1: The graphical illustration of the proposed model trying to generate the t -th target word y_t given a source sentence (x_1, x_2, \dots, x_T) .



Attention Learns Alignment

Neural Machine Translation by Jointly Learning to Align and Translate
Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, 2014

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

$$c_i = \sum_{j=1}^T a_{ij} h_j$$

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

$$e_{ij} = a(s_{i-1}, h_j)$$

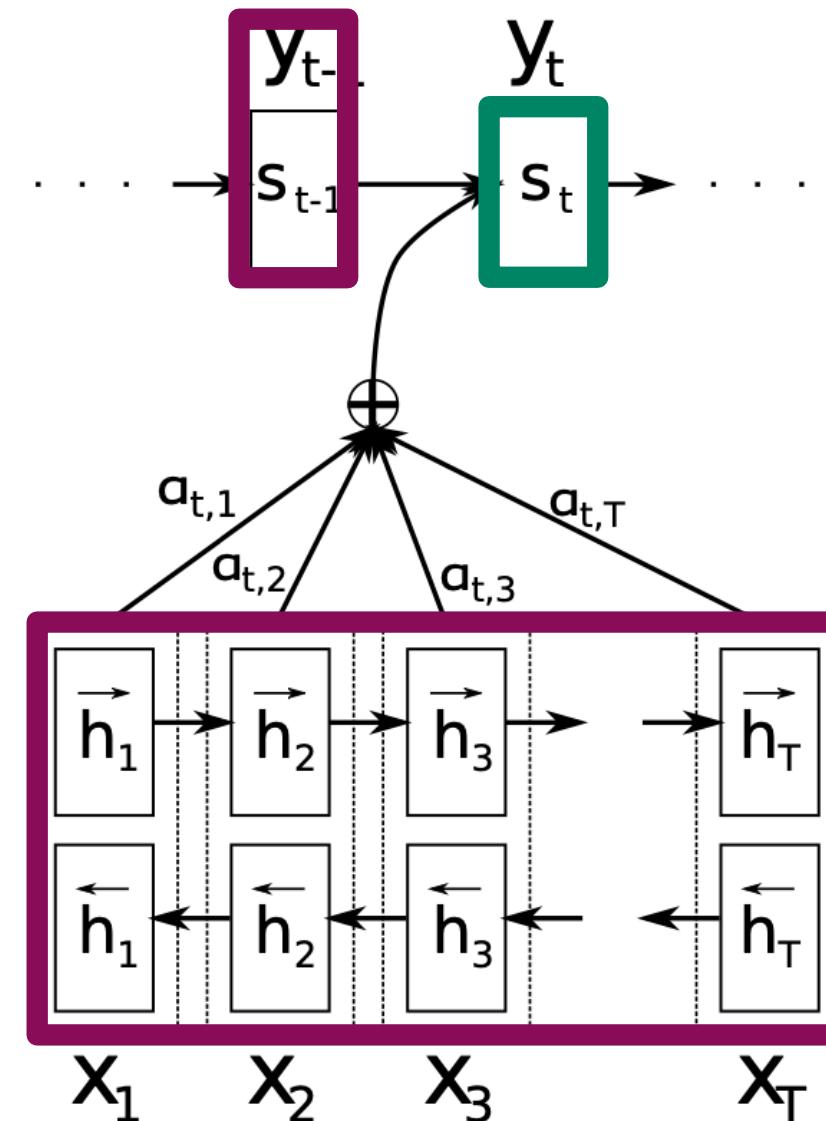
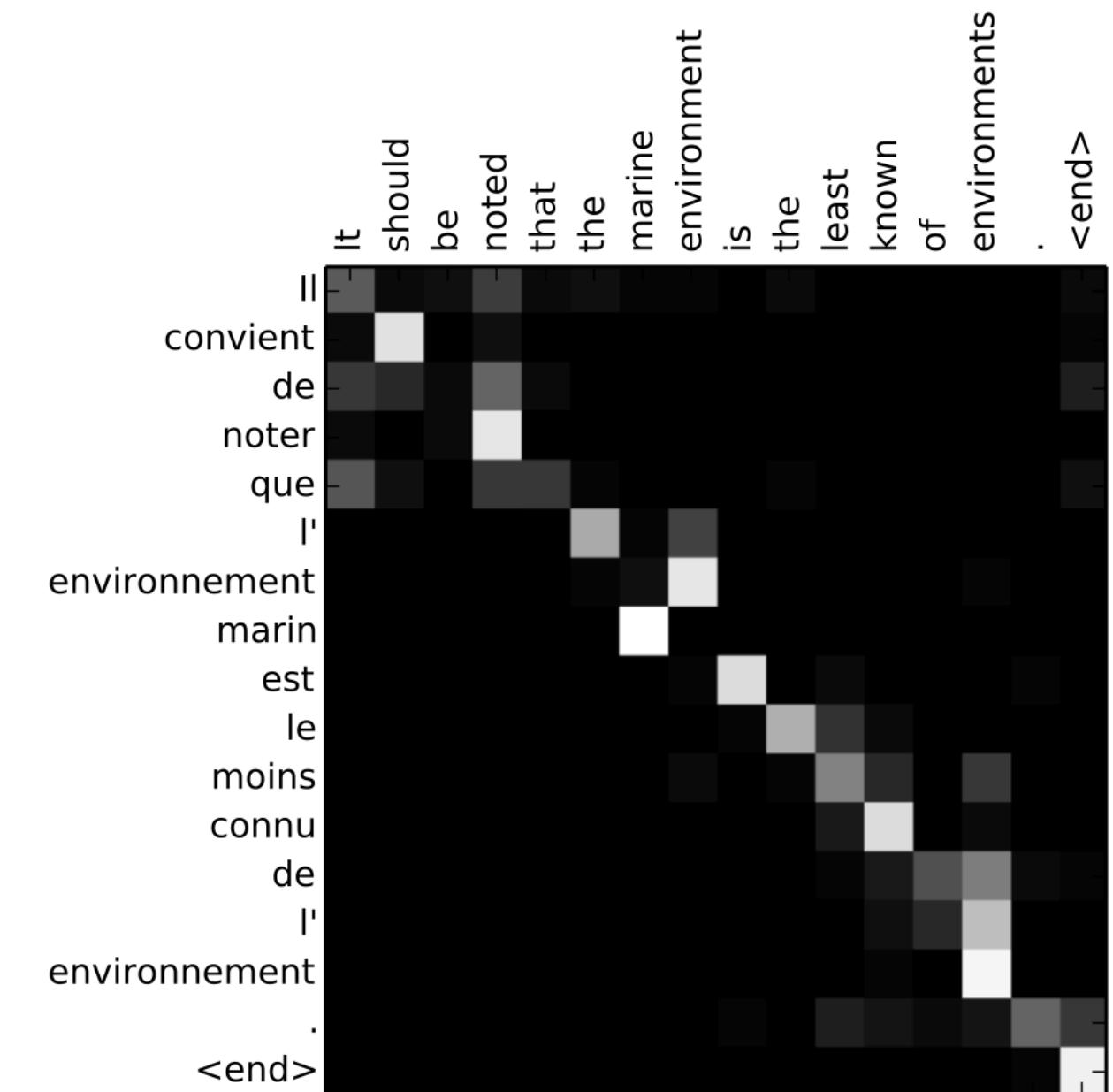


Figure 1: The graphical illustration of the proposed model trying to generate the t -th target word y_t given a source sentence (x_1, x_2, \dots, x_T) .



Attention

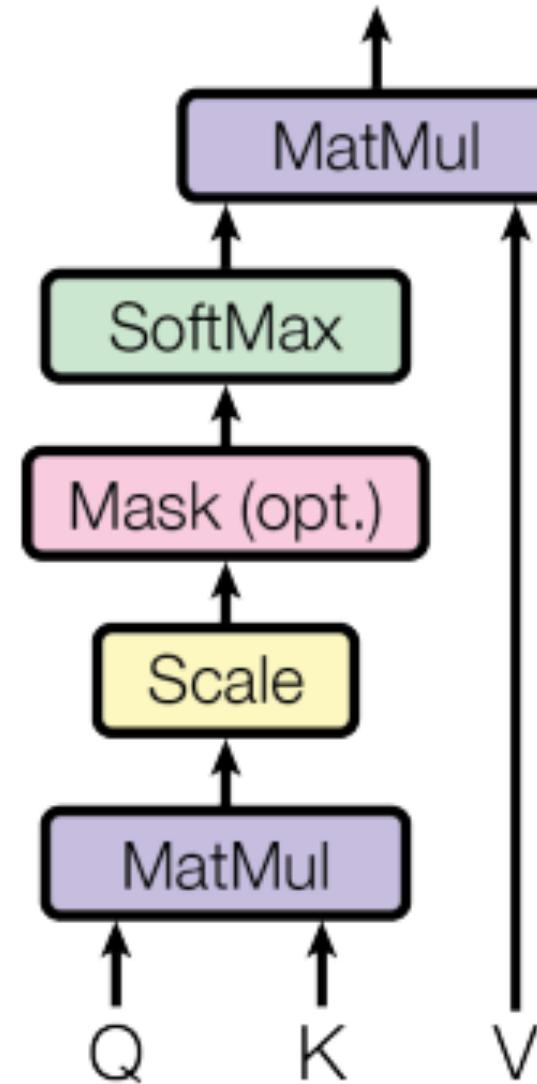
Attention Is All You Need

[Ashish Vaswani](#), [Noam Shazeer](#), [Niki Parmar](#), [Jakob Uszkoreit](#), [Llion Jones](#), [Aidan N. Gomez](#), [Łukasz Kaiser](#), [Illia Polosukhin](#), 2017

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- Assume that the components of q and k are independent random variable with mean 0 and variance 1.
- Their dot product $q \cdot k = \sum_{i=0}^d q_i k_i$, has mean 0 and variance d_k .
- Therefore, scale by $\frac{1}{\sqrt{d_k}}$ to counteract the large gradients.
- Attention is parameter-free!

Scaled Dot-Product Attention



Attention

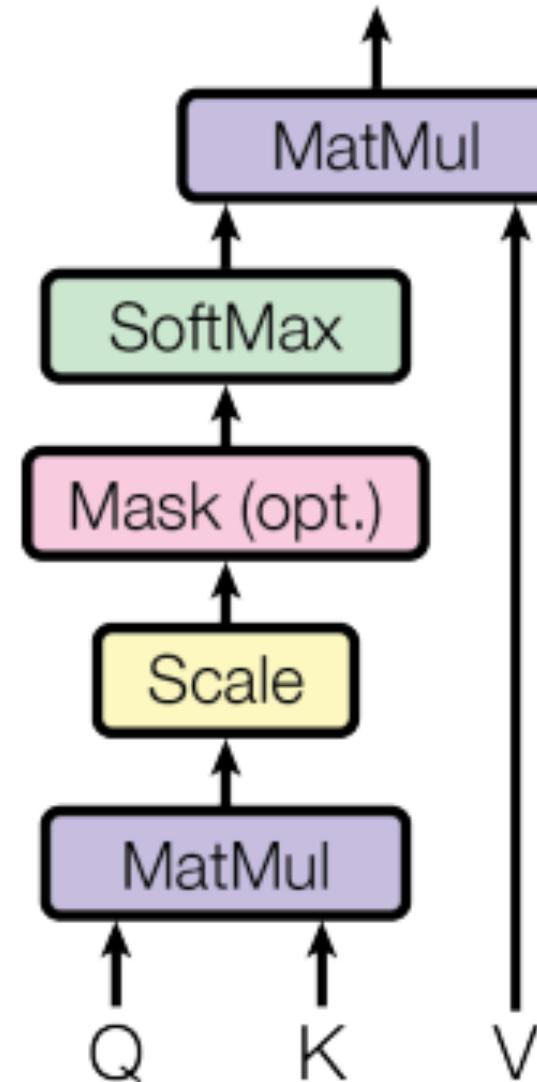
Attention Is All You Need

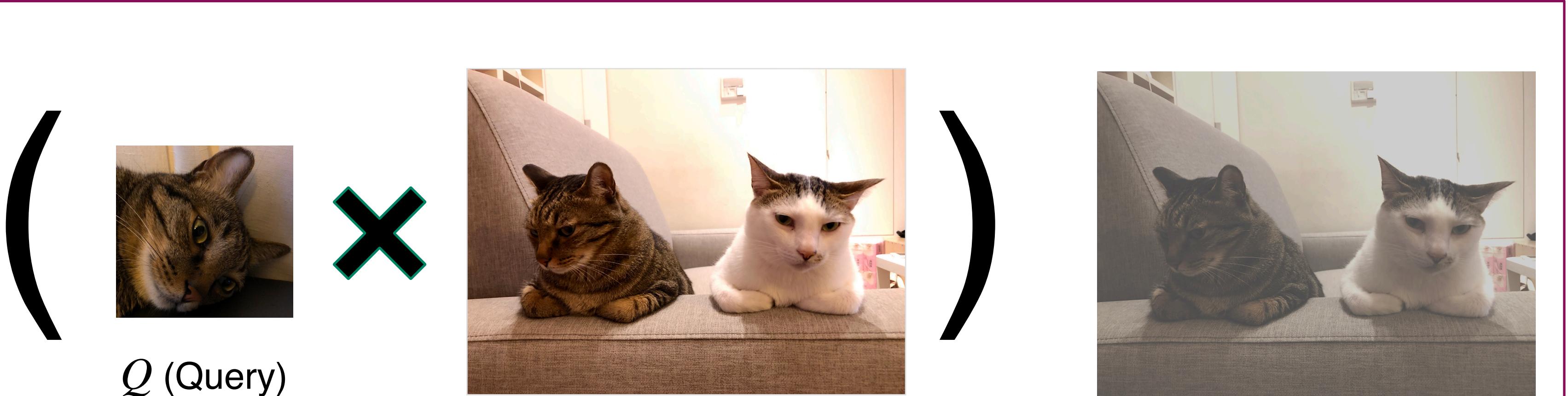
[Ashish Vaswani](#), [Noam Shazeer](#), [Niki Parmar](#), [Jakob Uszkoreit](#), [Llion Jones](#), [Aidan N. Gomez](#), [Łukasz Kaiser](#), [Illia Polosukhin](#), 2017

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- Assume that the components of q and k are independent random variable with mean 0 and variance 1.
- Their dot product $q \cdot k = \sum_{i=0}^d q_i k_i$, has mean 0 and variance d_k .
- Therefore, scale by $\frac{1}{\sqrt{d_k}}$ to counteract the large gradients.
- Attention is parameter-free!
- **But ... what exactly are Q , K , and V ?**

Scaled Dot-Product Attention





Q (Query)

K (Keys) Keys are usually derived from values, in
a form that easier to compare with query

V (Values)



Q (Query)

K (Keys) Keys are usually derived from values, in
a form that easier to compare with query

V (Values)

The dot-product between
query and keys computes the
correlation score





Q (Query)

K (Keys) Keys are usually derived from values, in a form that easier to compare with query

V (Values)

The dot-product between query and keys computes the correlation score



Then the score is applied on the value to retrieve the relevant content

Attention

Attention Is All You Need

[Ashish Vaswani](#), [Noam Shazeer](#), [Niki Parmar](#), [Jakob Uszkoreit](#), [Llion Jones](#), [Aidan N. Gomez](#), [Łukasz Kaiser](#), [Illia Polosukhin](#), 2017

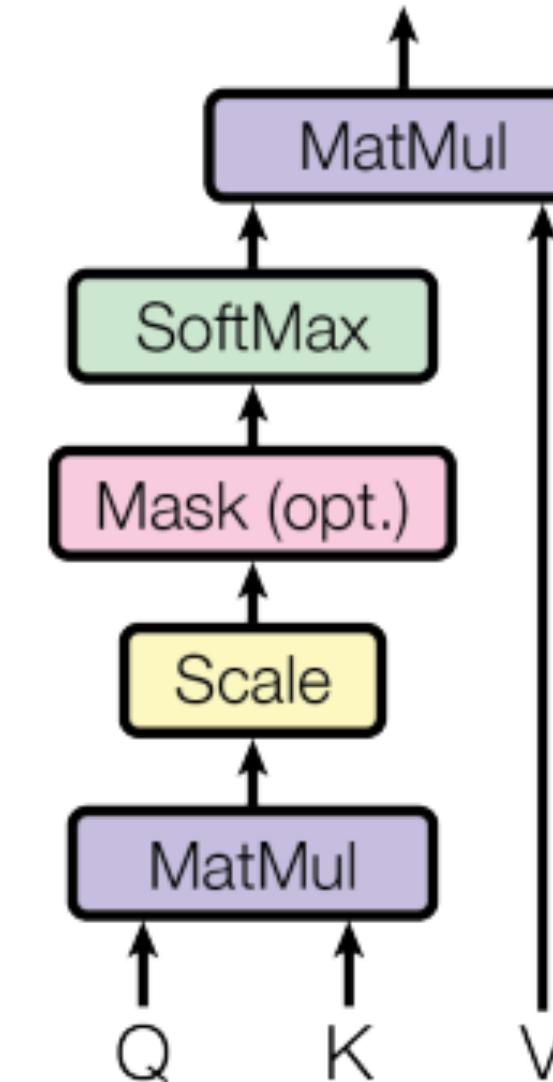
$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Scaled Dot-Product Attention

- Assume that the components of q and k are independent random variable with mean 0 and variance 1.
- Their dot product $q \cdot k = \sum_{i=0}^d q_i k_i$, has mean 0 and variance d_k .
- Therefore, scale by $\frac{1}{\sqrt{d_k}}$ to counteract the large gradients.

- **Q , K , and V presents the query, keys, and values, with the shape**

- $Q : (b, L_q, d_k)$
- $K : (b, L_k, d_k)$
- $V : (b, L_k, d_v)$



Attention

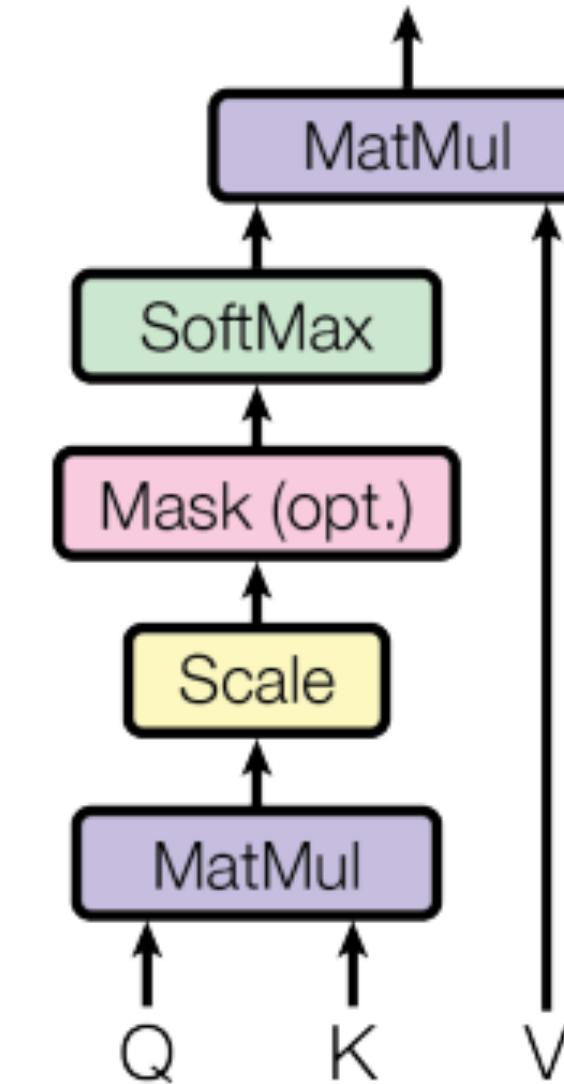
Attention Is All You Need

[Ashish Vaswani](#), [Noam Shazeer](#), [Niki Parmar](#), [Jakob Uszkoreit](#), [Llion Jones](#), [Aidan N. Gomez](#), [Łukasz Kaiser](#), [Illia Polosukhin](#), 2017

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Scaled Dot-Product Attention

- Assume that the components of q and k are independent random variable with mean 0 and variance 1.
- Their dot product $q \cdot k = \sum_{i=0}^d q_i k_i$, has mean 0 and variance d_k .
- Therefore, scale by $\frac{1}{\sqrt{d_k}}$ to counteract the large gradients.
- Q, K , and V presents the query, keys, and values, with the shape**
 - $Q : (b, L_q, d_k)$
 - $K : (b, L_k, d_k)$
 - $V : (b, L_k, d_v)$



If Q, K, V are from the same source, we say it is **Self-attention**; otherwise, it is **Cross-attention**.

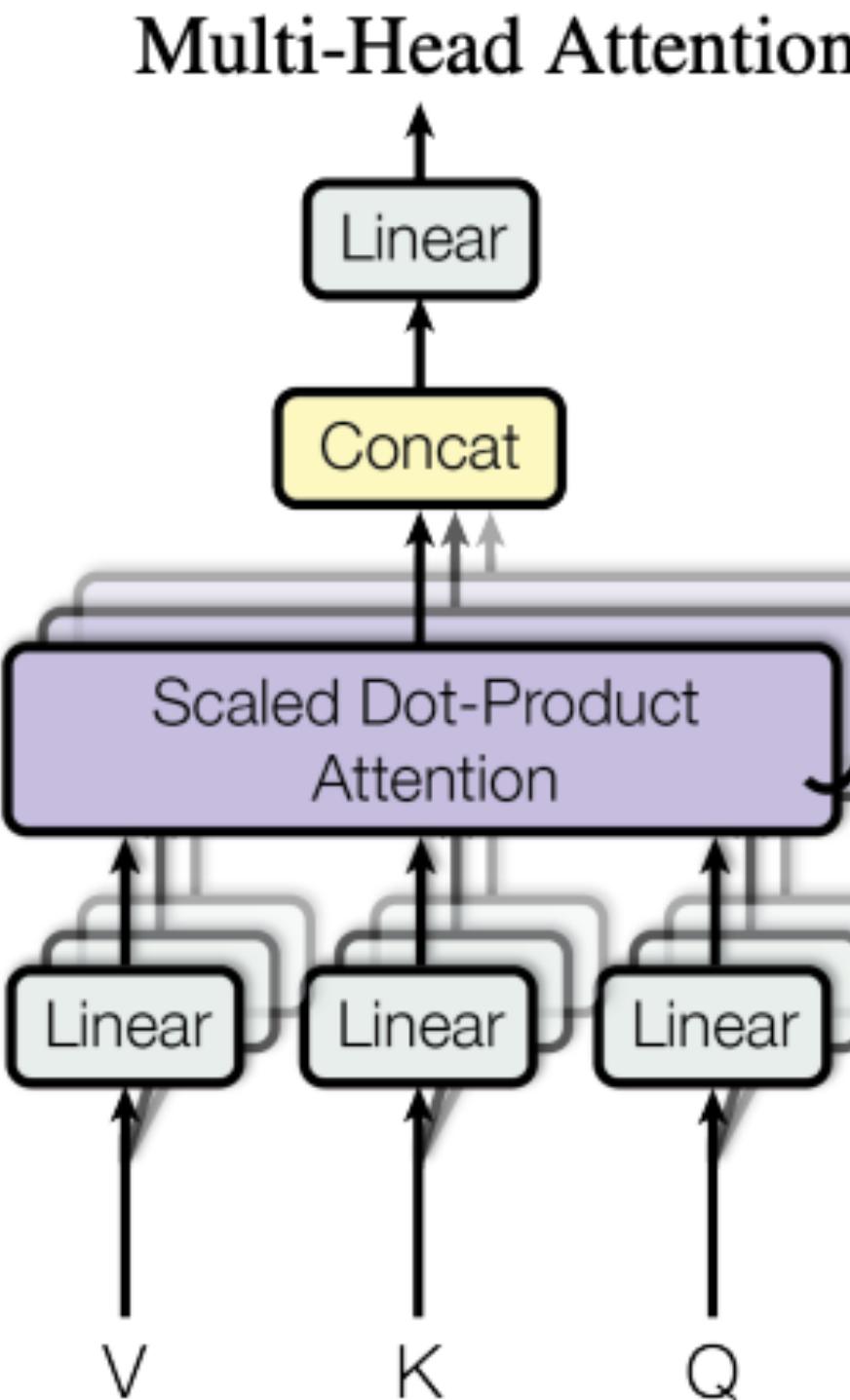
Multi-Head Attention

Attention Is All You Need

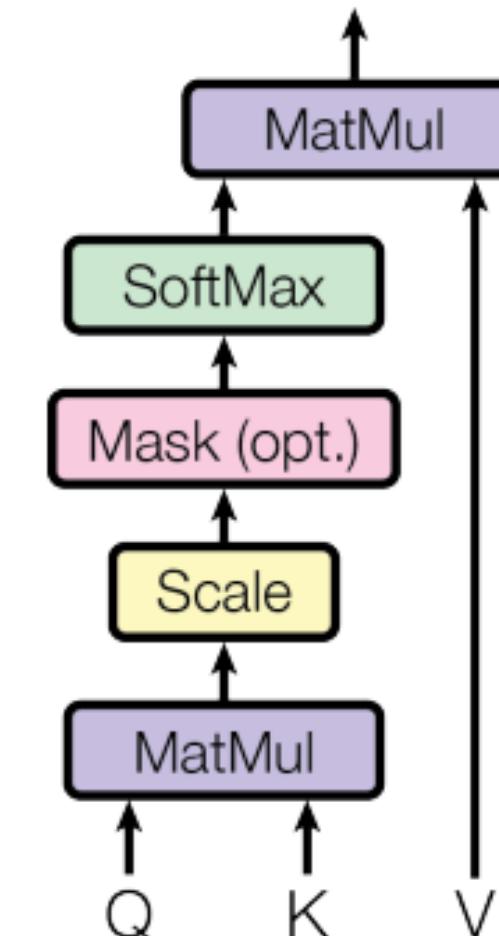
[Ashish Vaswani](#), [Noam Shazeer](#), [Niki Parmar](#), [Jakob Uszkoreit](#), [Llion Jones](#), [Aidan N. Gomez](#), [Łukasz Kaiser](#), [Illia Polosukhin](#), 2017

Multi-Head Attention is to split each of the Q, K, V into several sub-groups (heads).

Each head is now with the dimension $\frac{d_k \text{ or } d_v}{\text{heads}}$.

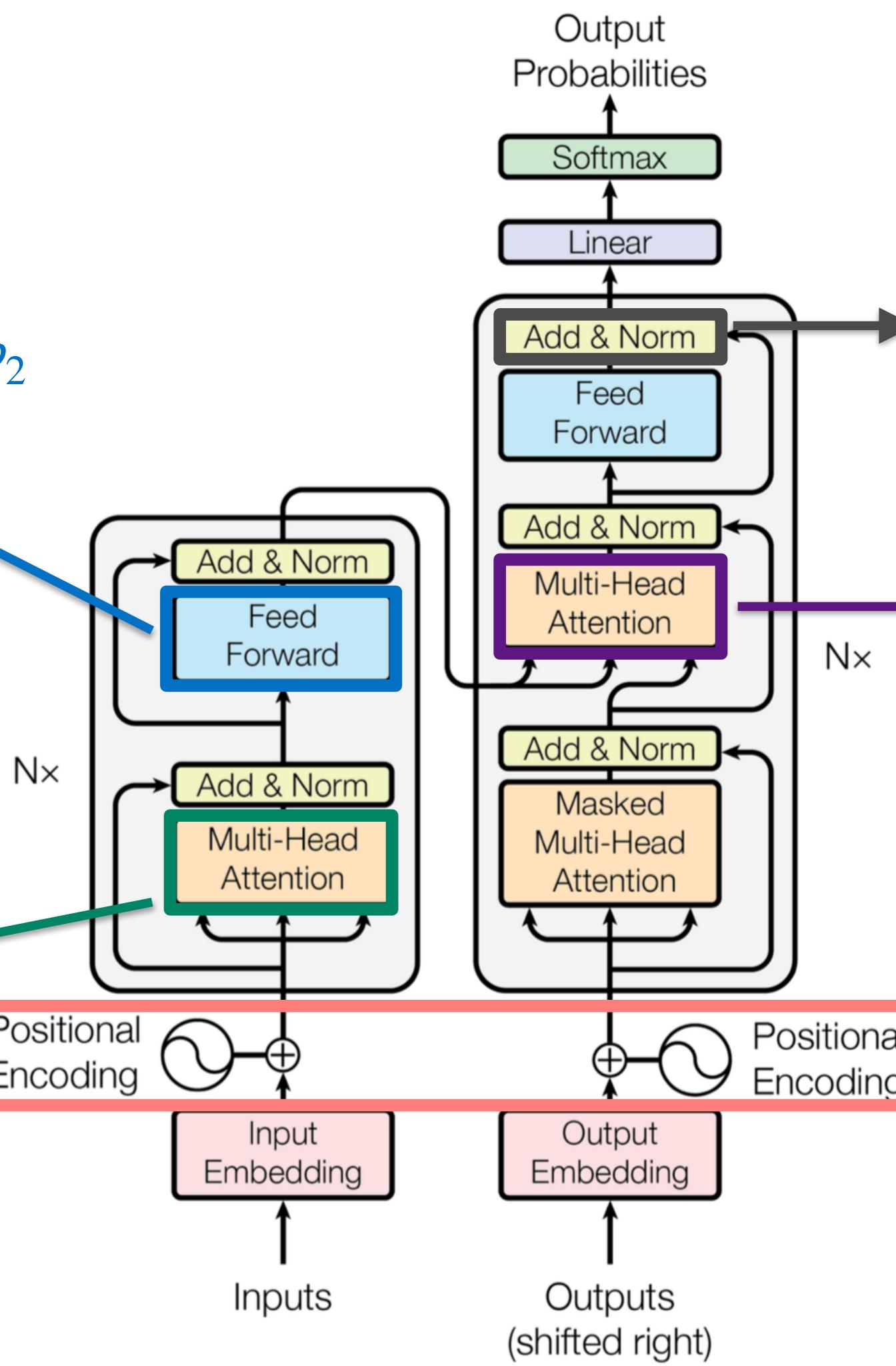
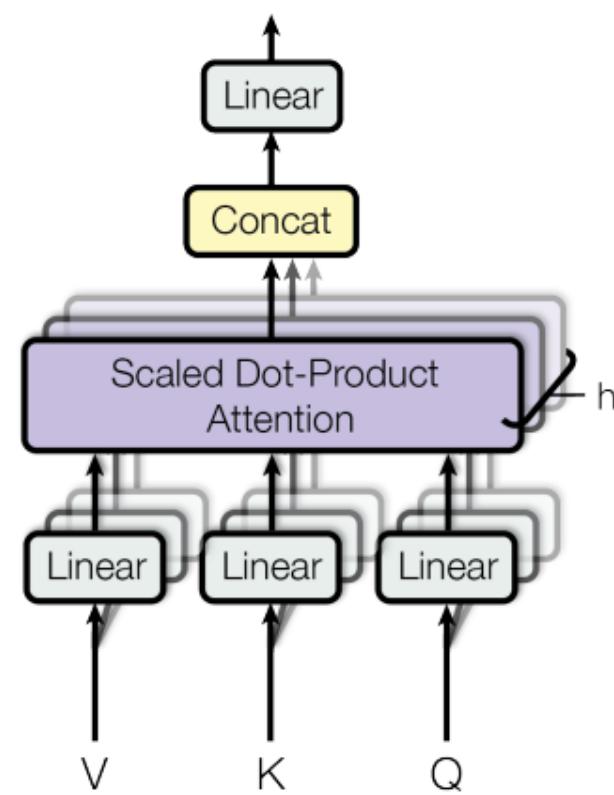


Scaled Dot-Product Attention



Point-Wise MLP:
 $y_i = W_2 \text{GELU}(W_1 x + b_1) + b_2$

Multi-Head Self-Attention



Shortcut + LayerNorm

PostNorm (figure):
 $LN(Q + Attn(Q, K, V))$

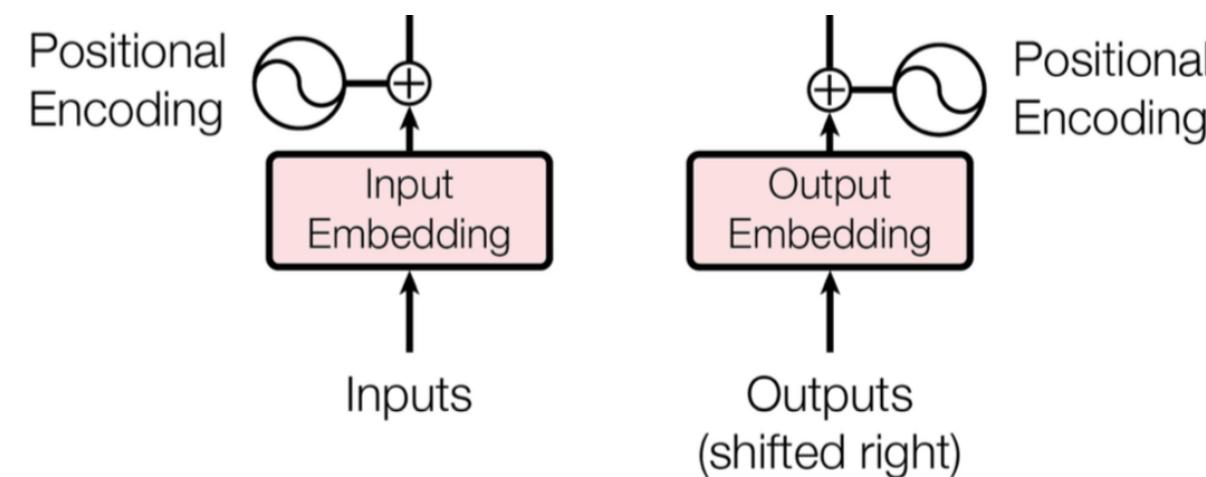
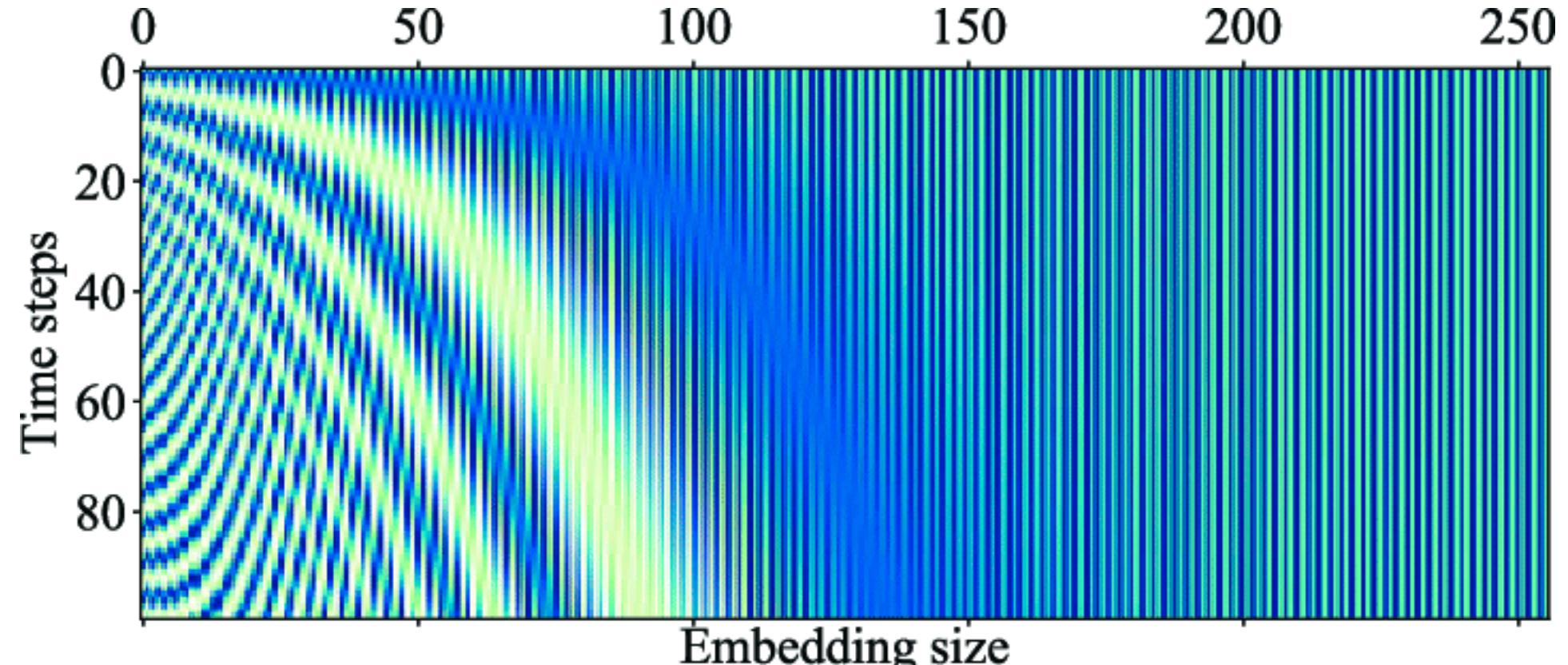
PreNorm:
 $Q', K', L' = LN(Q), LN(K), LN(V)$
 $Q' + Attn(Q', K', V')$

Multi-Head Cross-Attention

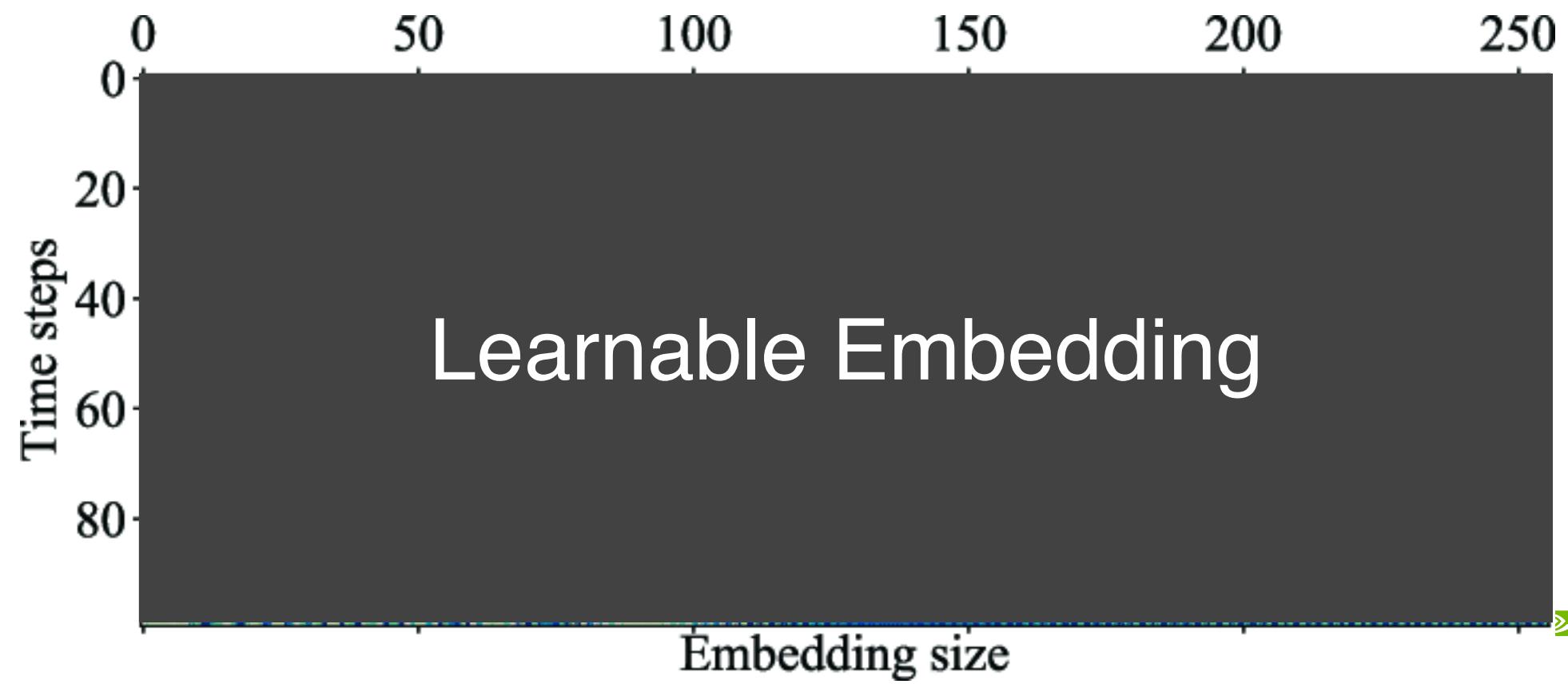
Since Q, K, V in attention is permute-invariant. Positional Encoding indicate the order appeared.

Recall: $\text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$

Sinusoidal Positional Encoding



General Positional Embedding



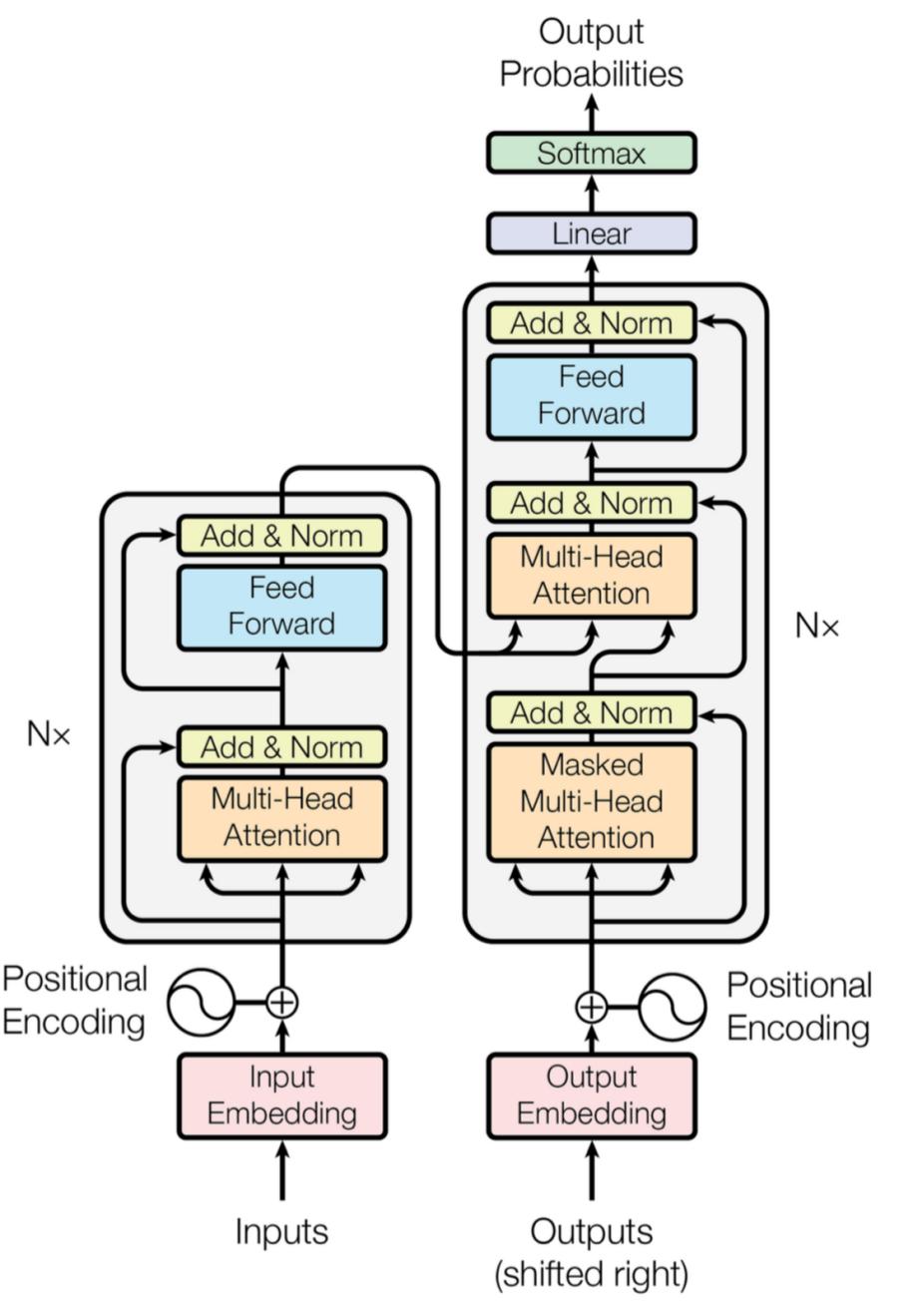
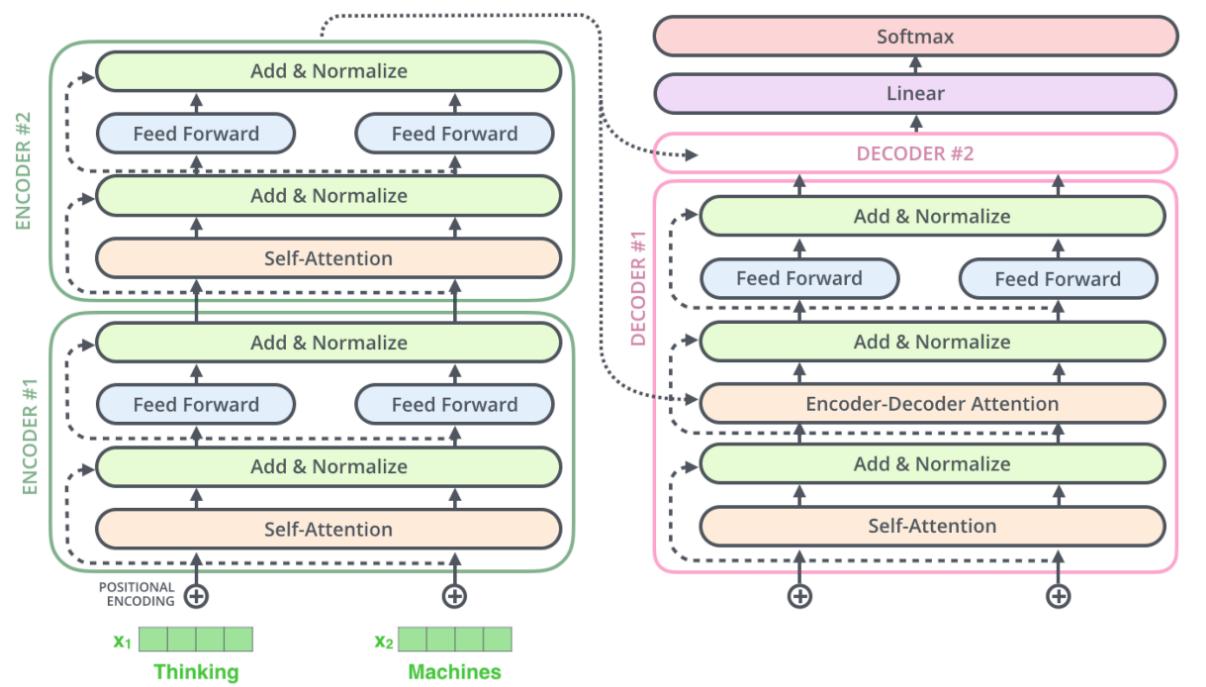
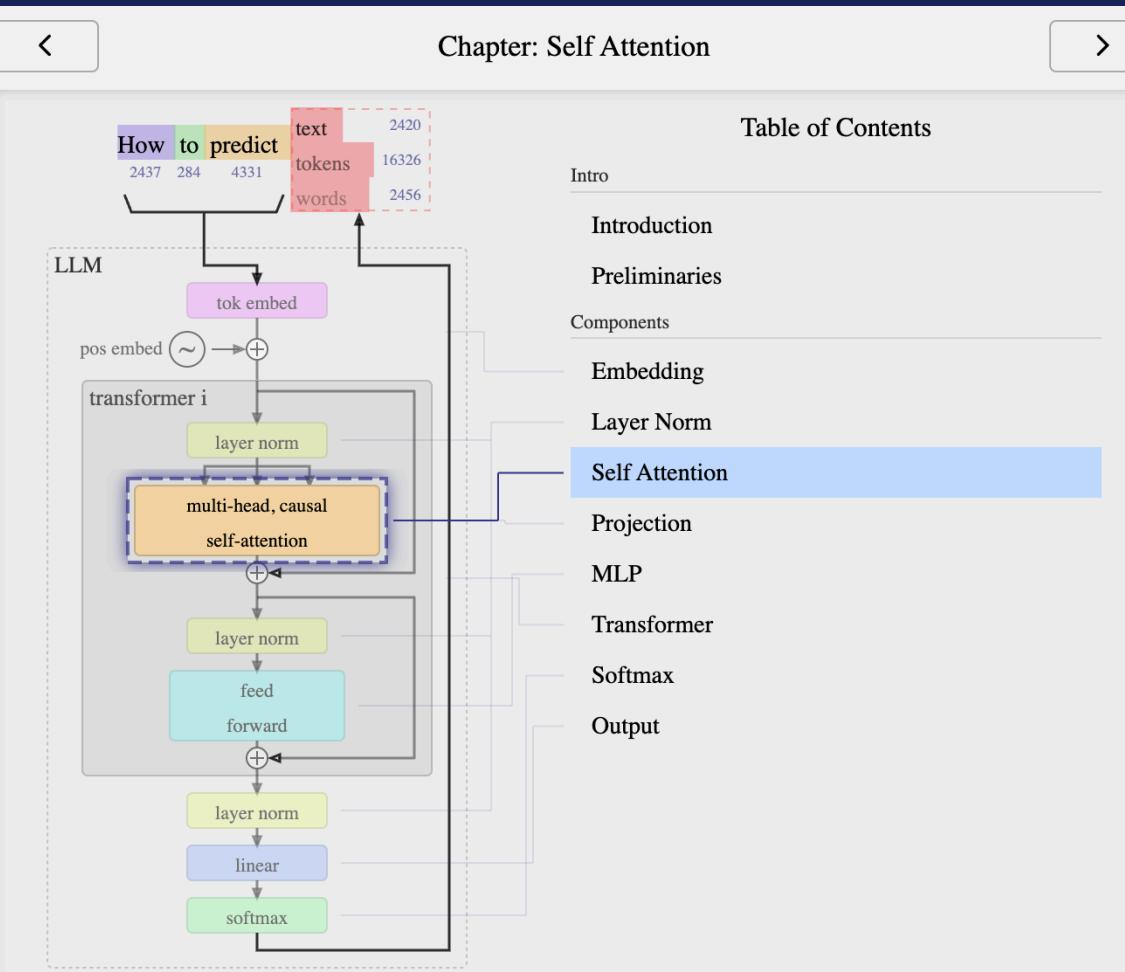


Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	41.29	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1		$3.3 \cdot 10^{18}$
Transformer (big)	28.4	41.8		$2.3 \cdot 10^{19}$

I arrived at the





The self-attention layer is perhaps the heart of the Transformer and of GPT. It's the phase where the columns in our input embedding matrix "talk" to each other. Up until now, and in all other phases, the columns can be regarded independently.

The self-attention layer is made up of several heads, and we'll focus on one of them for now.

The first step is to produce three vectors for each of the 1 columns from the normalized input embedding

Q: Query vector

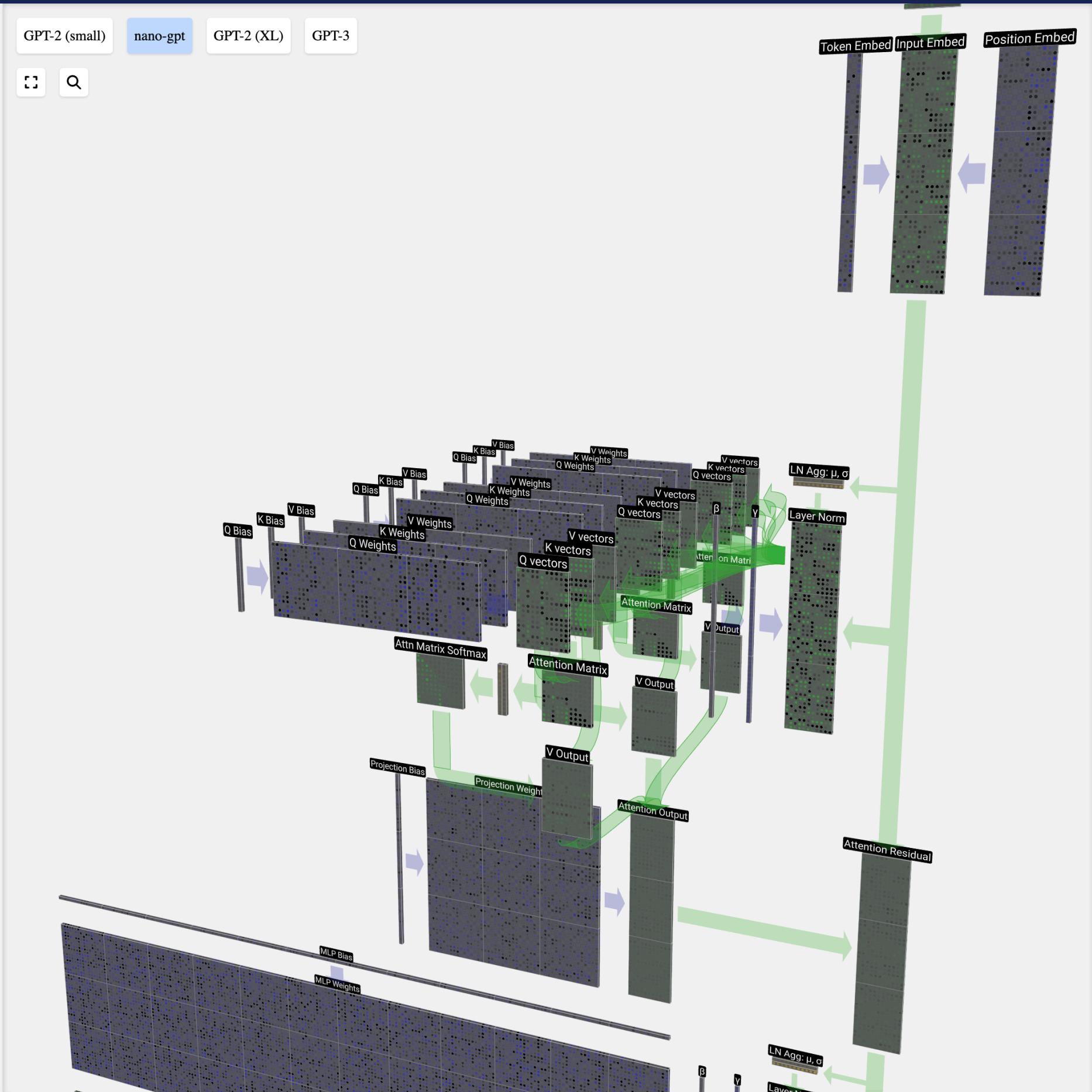
K: Key vector

To produce one of these vectors, we perform a matrix-vector multiplication with a bias added. Each output cell is some linear combination of the input vector. E.g. for the [Q vectors](#), this is done with a dot product between a row of the [Q weight matrix](#) and a column of the [input matrix](#).

The dot product operation, which we'll see a lot of, is quite simple: We pair each element from the first vector with the corresponding element from the second vector, multiply the pairs together and then add the results

Continue

Skip



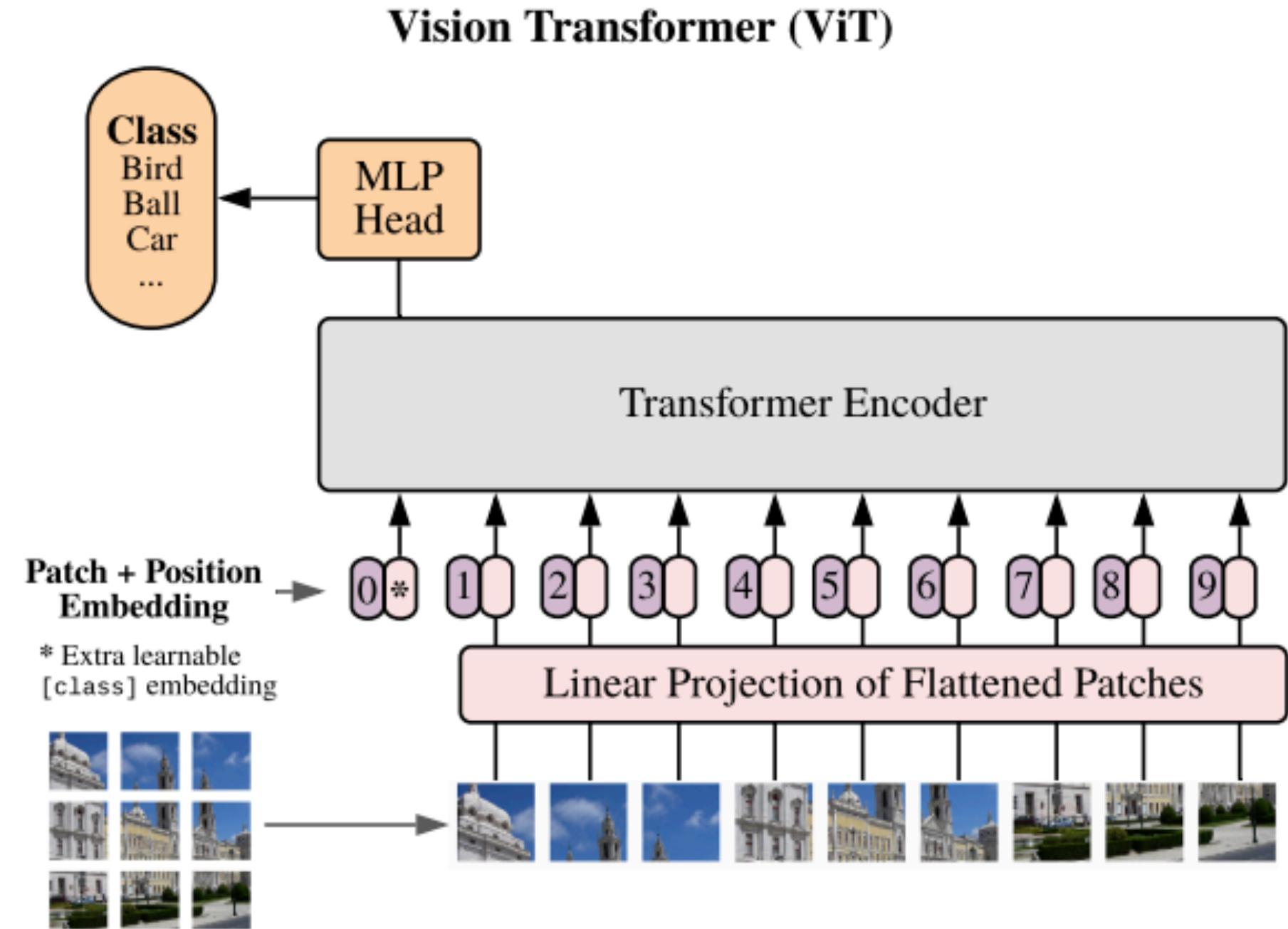
Transformer and Attention In *Computer Vision*

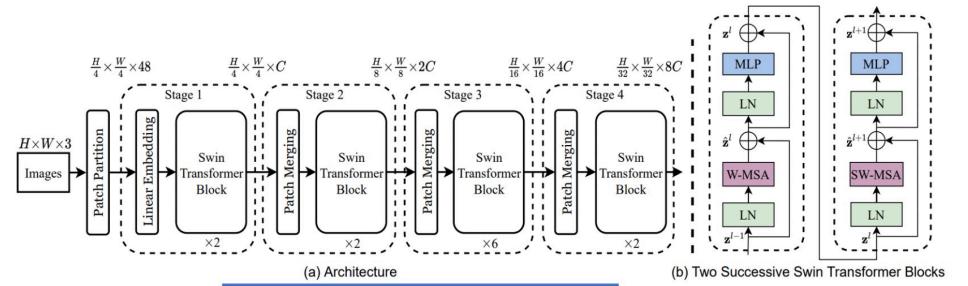
Vision Transformer

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

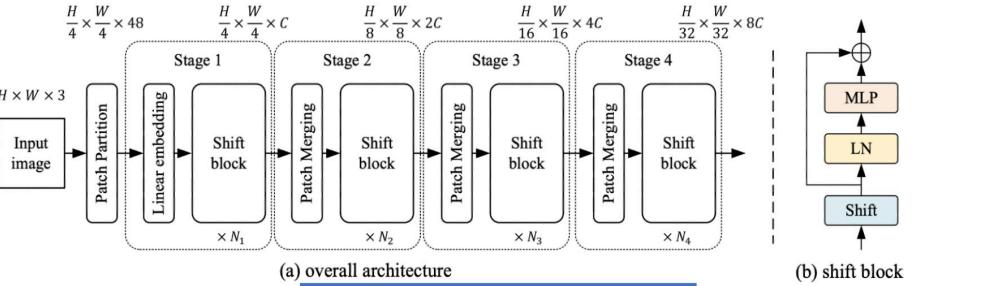
Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby, 2020

- Encoder-only transformer.
- Anything you can tokenize, you can feed to Transformer
 - How to tokenize an image?
 - For (224,224) pixels, that's 50k sequence length!
 - Thus, most works restrict attention to local pixel neighborhood, by cutting it into patches of (16, 16)

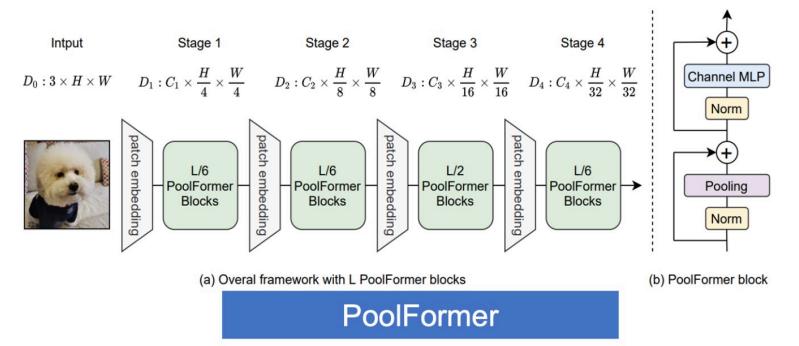




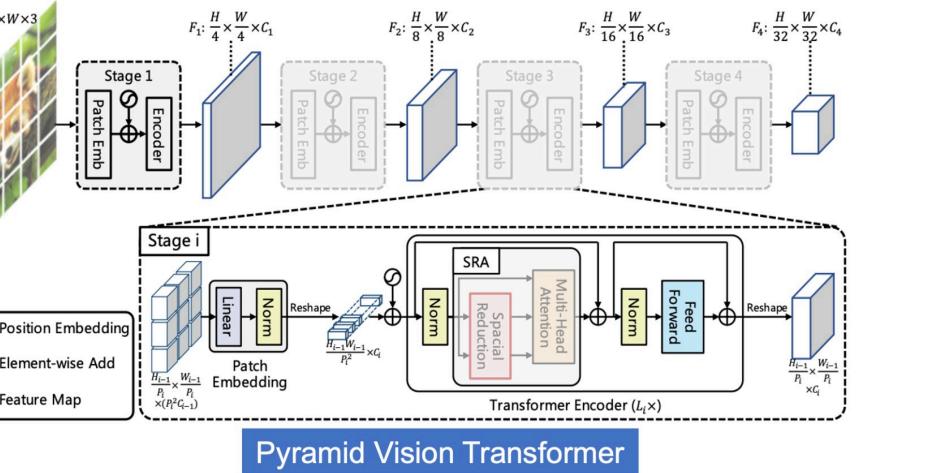
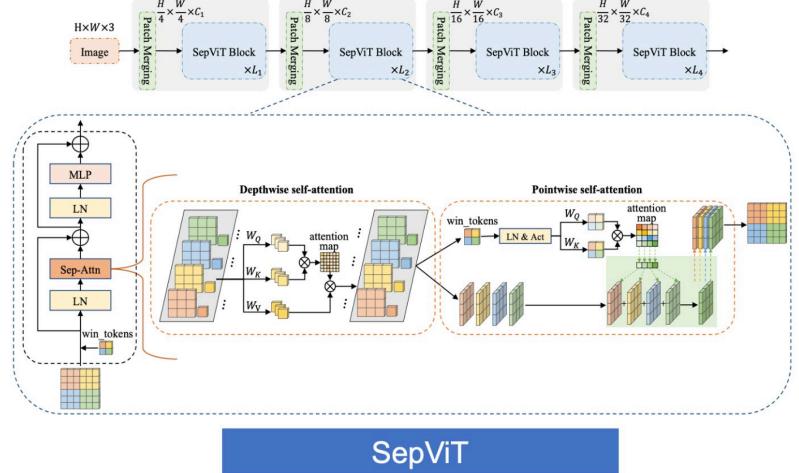
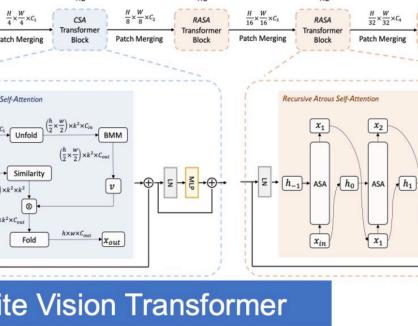
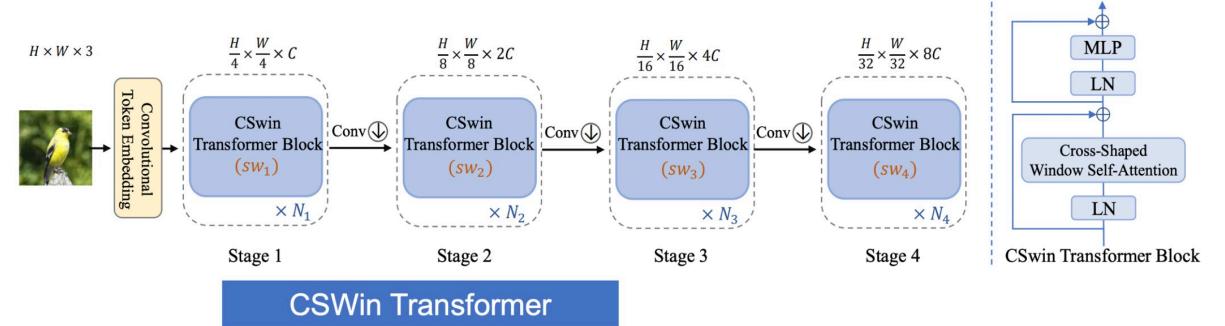
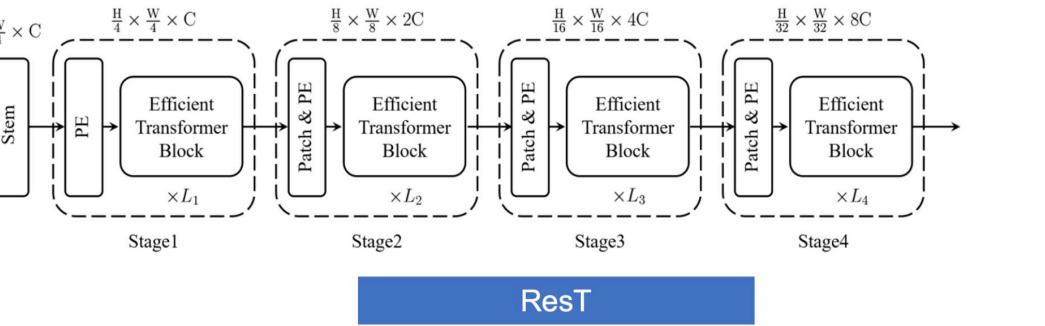
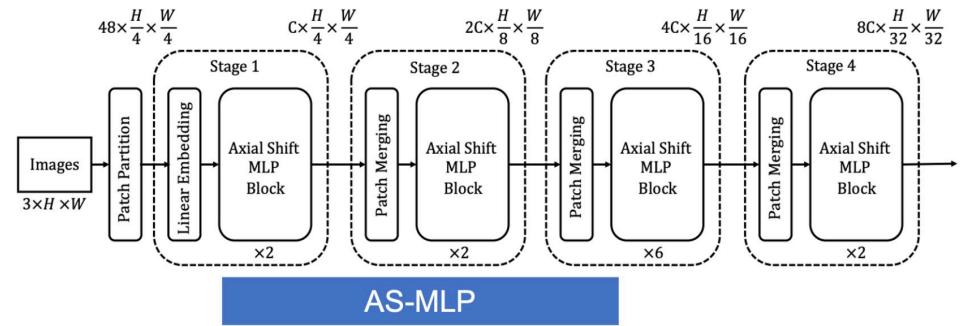
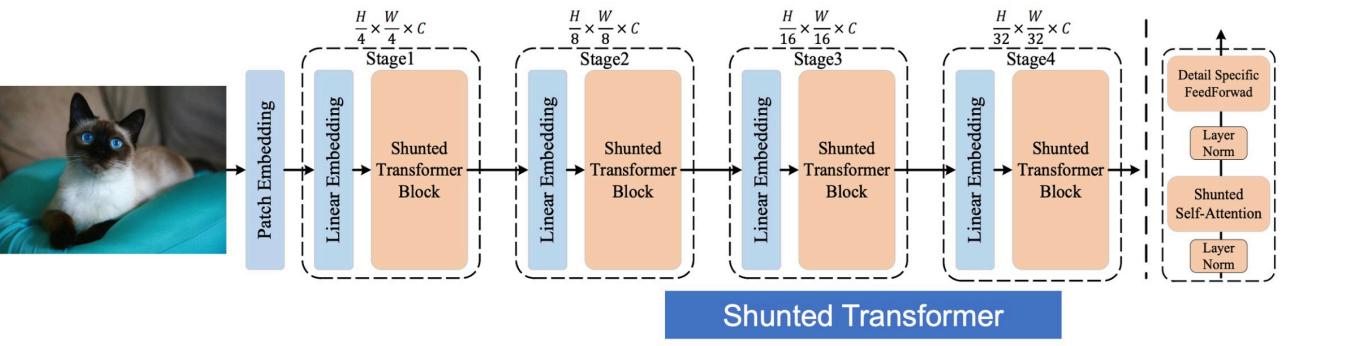
Swin Transformer



ShiftViT



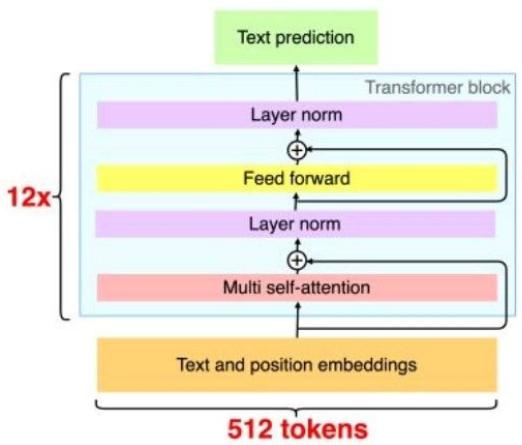
PoolFormer



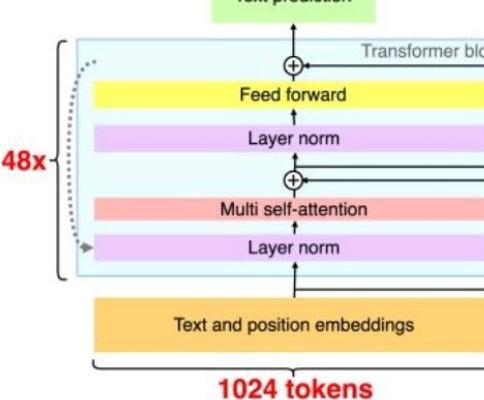
Explosion of works doing Vision Transformers on images



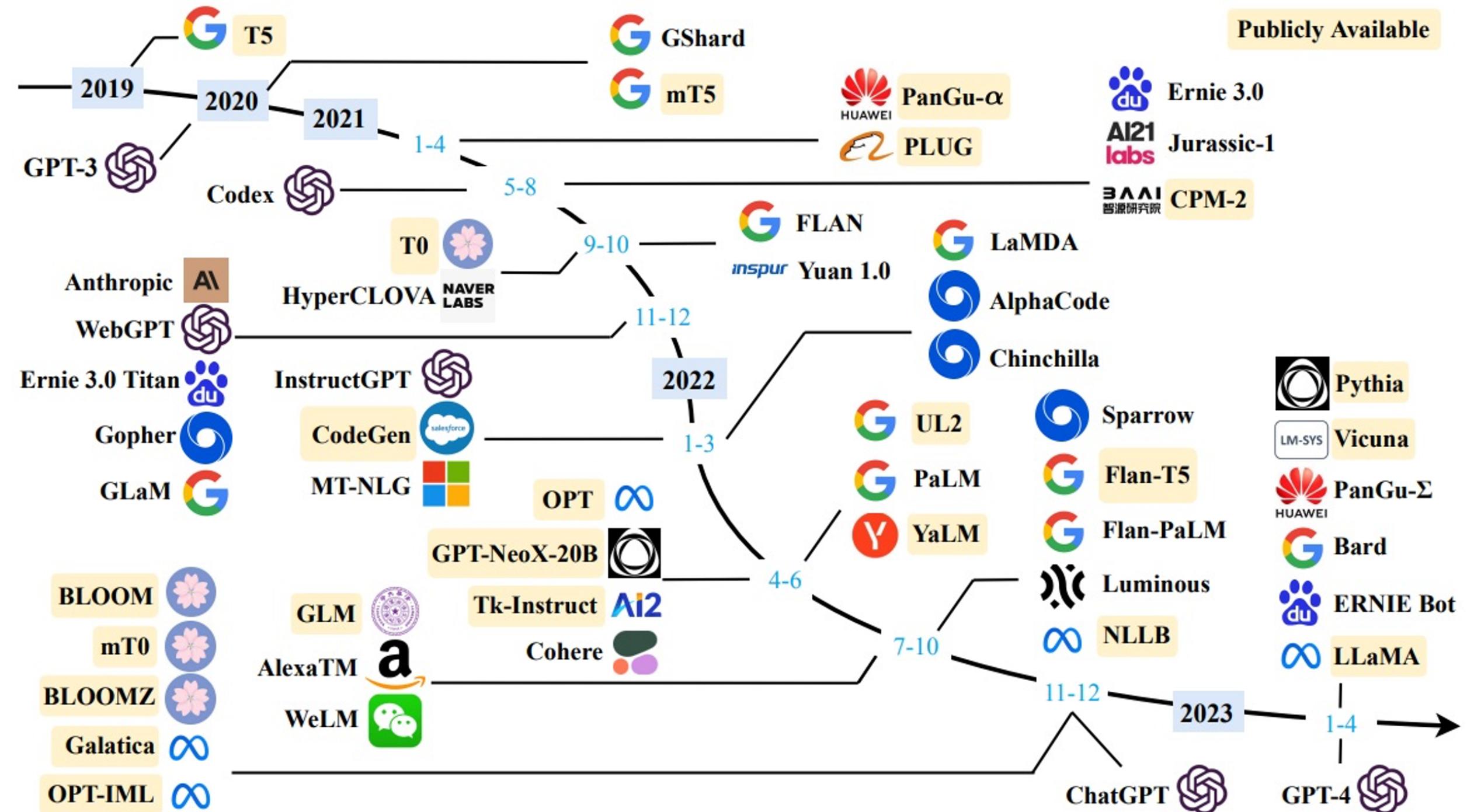
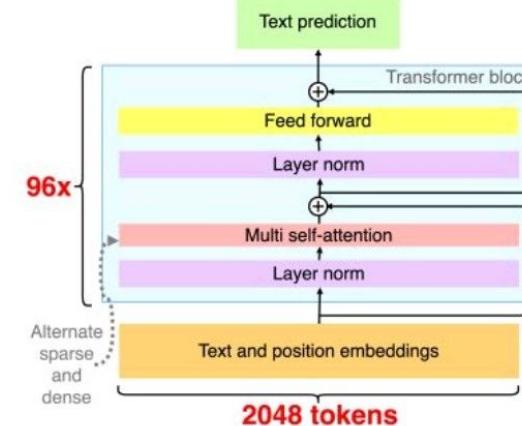
GPT-1



GPT-2



GPT-3



Zhao, Wayne Xin, et al. "A survey of large language models." *arXiv preprint arXiv:2303.18223* (2023).

Transformer Was Not Deep ... Until Now

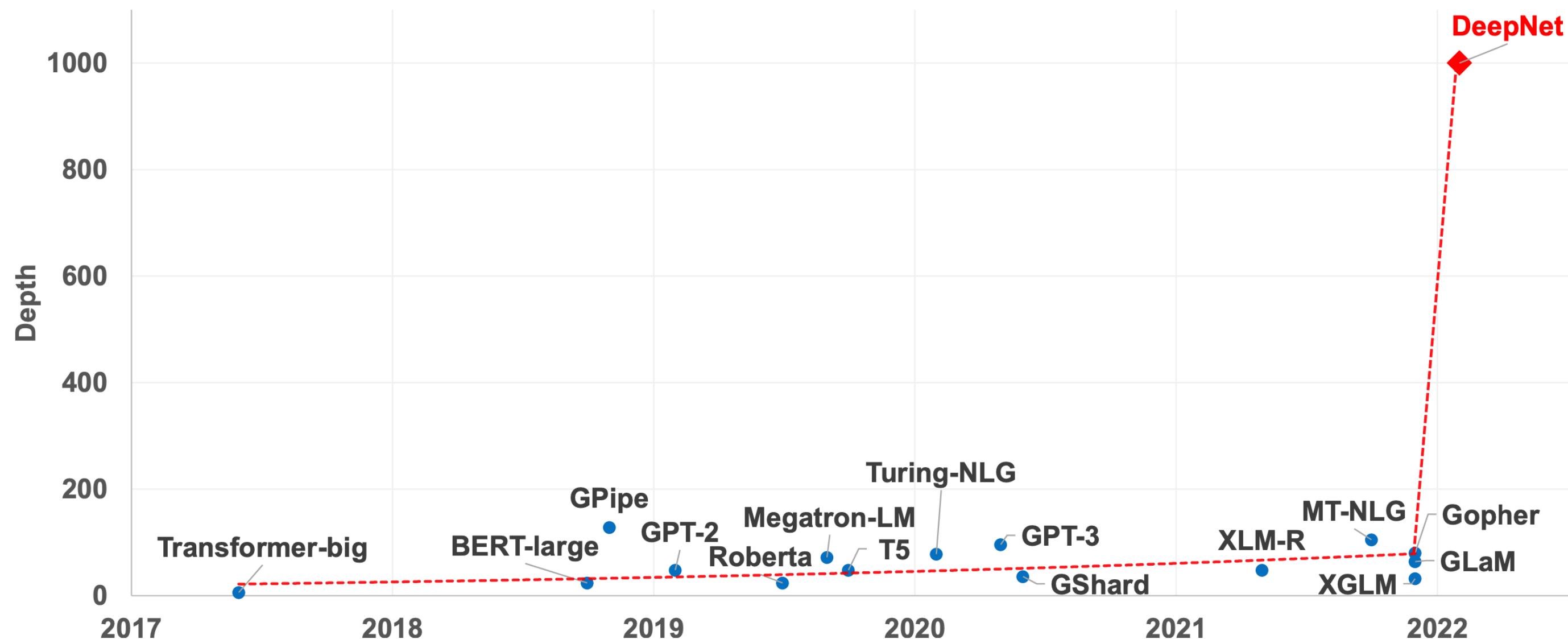
DeepNet: Scaling Transformers to 1,000 Layers
[Hongyu Wang](#), [Shuming Ma](#), [Li Dong](#), [Shaohan Huang](#), [Dongdong Zhang](#), [Furu Wei](#), 2022

- Normalization and Initialization matters.

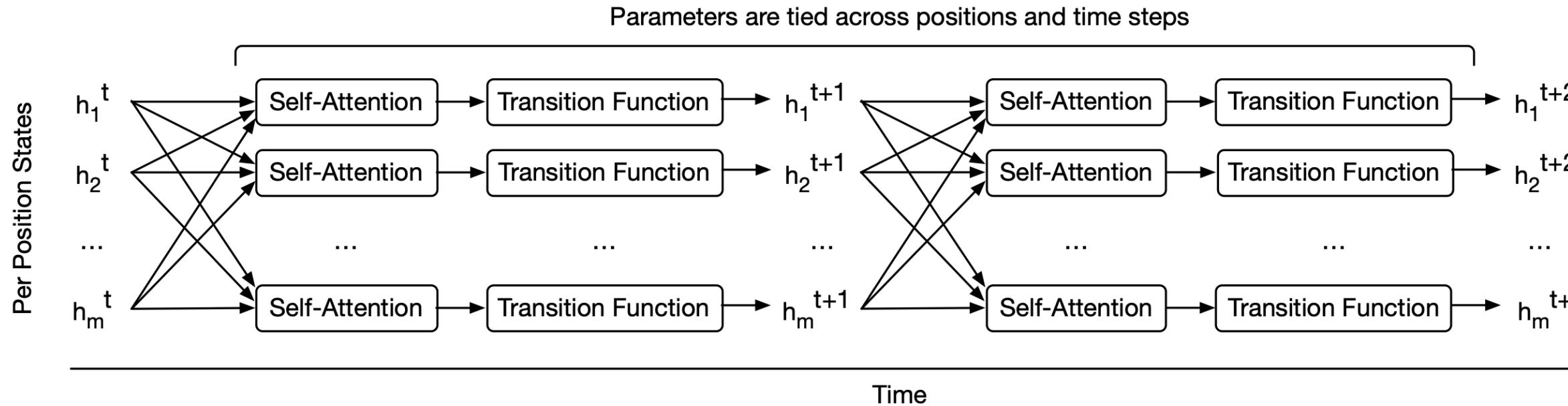
```
def deepnorm(x):
    return LayerNorm(x *  $\alpha$  + f(x))

def deepnorm_init(w):
    if w is ['ffn', 'v_proj', 'out_proj']:
        nn.init.xavier_normal_(w, gain= $\beta$ )
    elif w is ['q_proj', 'k_proj']:
        nn.init.xavier_normal_(w, gain=1)
```

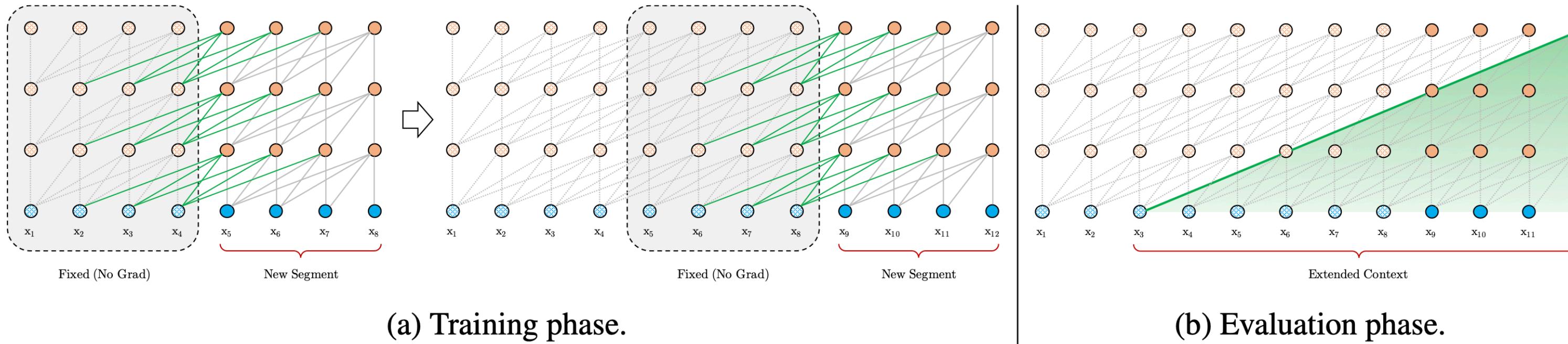
Architectures	Encoder	Decoder	
	α	β	α
Encoder-only (e.g., BERT)	$(2N)^{\frac{1}{4}}$	$(8N)^{-\frac{1}{4}}$	-
Decoder-only (e.g., GPT)	-	-	$(2M)^{\frac{1}{4}}$ $(8M)^{-\frac{1}{4}}$
Encoder-decoder (e.g., NMT, T5)	$0.81(N^4 M)^{\frac{1}{16}}$	$0.87(N^4 M)^{-\frac{1}{16}}$	$(3M)^{\frac{1}{4}}$ $(12M)^{-\frac{1}{4}}$



Transformer Meets Recurrent



Dehghani, Mostafa, et al. "Universal Transformers." *International Conference on Learning Representations*. 2018.



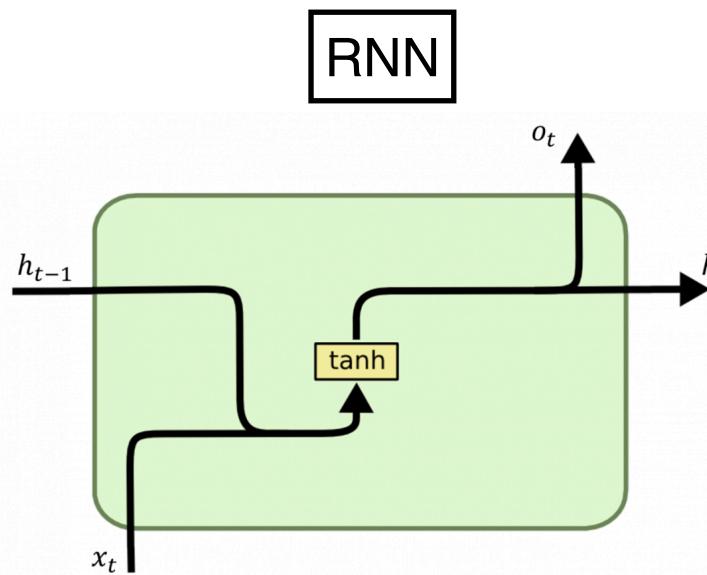
Dai, Zihang, et al. "Transformer-xl: Attentive language models beyond a fixed-length context." *arXiv preprint arXiv:1901.02860* (2019).

Side Note: Recurrent Networks

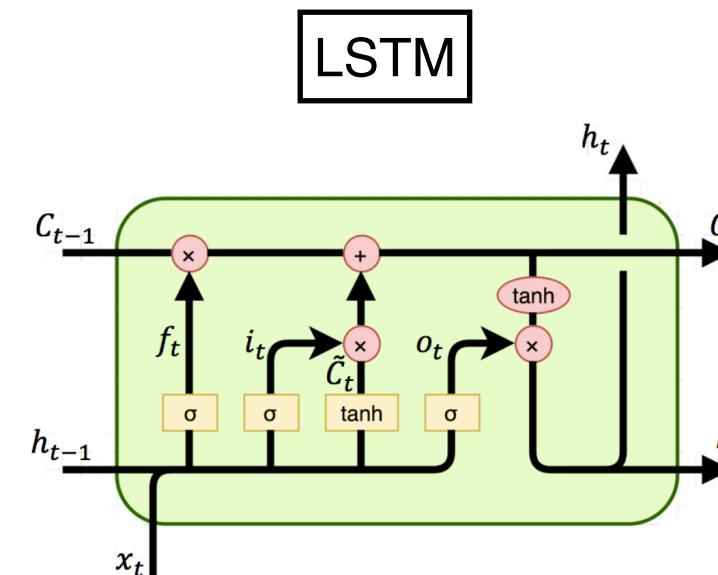
- Recurrent neural networks update its internal memory with a shared function f
- Generally, we can present a recurrent network by

$$h_t = f(x_t; h_{t-1})$$

- For example,



$$h_t = \sigma_h(i_t) = \sigma_h(U_h x_t + V_h h_{t-1} + b_h)$$



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

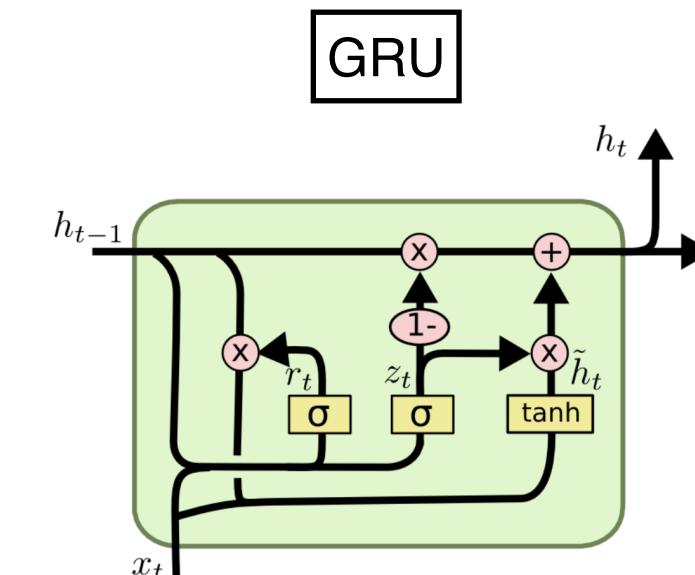
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

$$h_t = o_t \odot \tanh(C_t)$$



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

Side Note: Recurrent Networks

- Recurrent neural networks update its internal memory with a shared function f
- Generally, we can present a recurrent network by

$$h_t = f(x_t; h_{t-1})$$

- The above equation describes a Markov chain. We can also extend to higher-order Markov model.

$$h_t = f(x_t; h_{t-1}, h_{t-2}, h_{t-3}, \dots, h_{t-S})$$

- In this case, an aggregation function $\phi(\cdot)$ is introduced to summarize the past S states,

$$h_t = f(x_t + \phi(h_{t-1}, h_{t-2}, h_{t-3}, \dots, h_{t-S}))$$

- Where $\phi(\cdot)$ can be:
 - Linear
 - Soltani, Rohollah, and Hui Jiang. "Higher order recurrent neural networks." *arXiv preprint arXiv:1605.00064* (2016).
 - Polynomial
 - Yu, Rose, et al. "Long-term forecasting using tensor-train rnns." *Arxiv* (2017).
 - Tensor-Train Convolutions
 - Su, Jiahao, et al. "Convolutional tensor-train lstm for spatio-temporal learning." *Advances in Neural Information Processing Systems* 33 (2020): 13714-13726.

Side Note: Recurrent Networks

- Recurrent neural networks update its internal memory with a shared function f
- Generally, we can present a recurrent network by

$$h_t = f(x_t; h_{t-1})$$

- The above equation describes a Markov chain. We can also extend to higher-order Markov model.

$$h_t = f(x_t; h_{t-1}, h_{t-2}, h_{t-3}, \dots, h_{t-S})$$

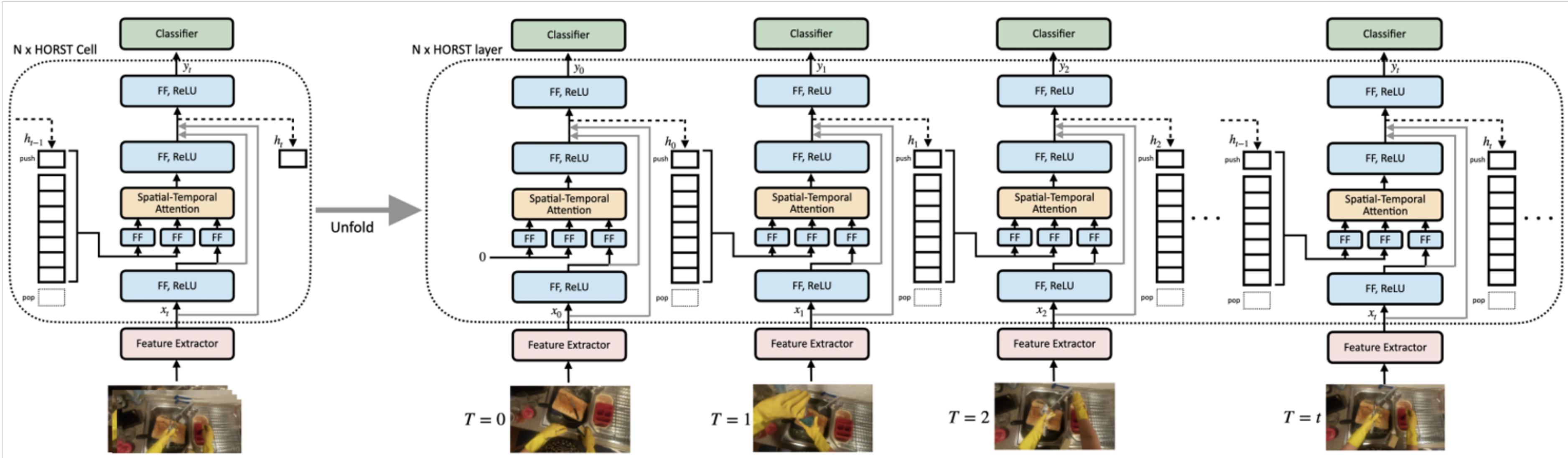
- In this case, an aggregation function $\phi(\cdot)$ is introduced to summarize the past S states,

$$h_t = f(x_t + \phi(h_{t-1}, h_{t-2}, h_{t-3}, \dots, h_{t-S}))$$

- Where $\phi(\cdot)$ can be:
 - Linear
 - Soltani, Rohollah, and Hui Jiang. "Higher order recurrent neural networks." *arXiv preprint arXiv:1605.00064* (2016).
 - Polynomial
 - Yu, Rose, et al. "Long-term forecasting using tensor-train rnns." *Arxiv* (2017).
 - Tensor-Train Convolutions
 - Su, Jiahao, et al. "Convolutional tensor-train lstm for spatio-temporal learning." *Advances in Neural Information Processing Systems* 33 (2020): 13714-13726.
 - **Self-Attention 😺?**

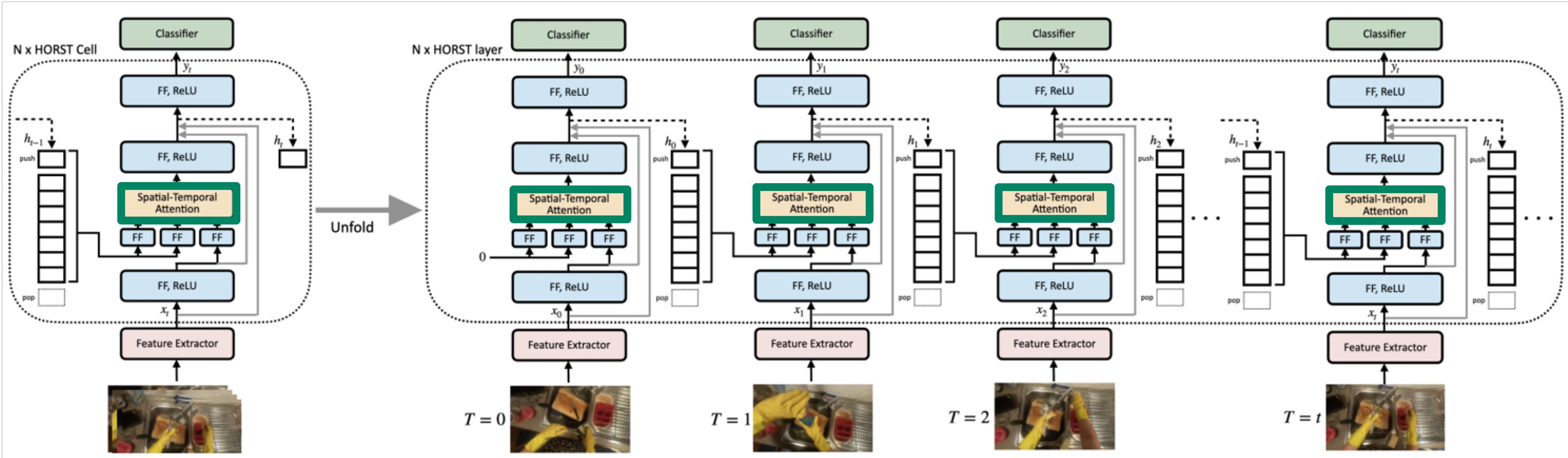
Transformer Meets (Higher-Order) Recurrent

Higher Order Recurrent Space-Time Transformer for Video Action Prediction
Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Oswald Lanz, 2021



Transformer Meets (Higher-Order) Recurrent

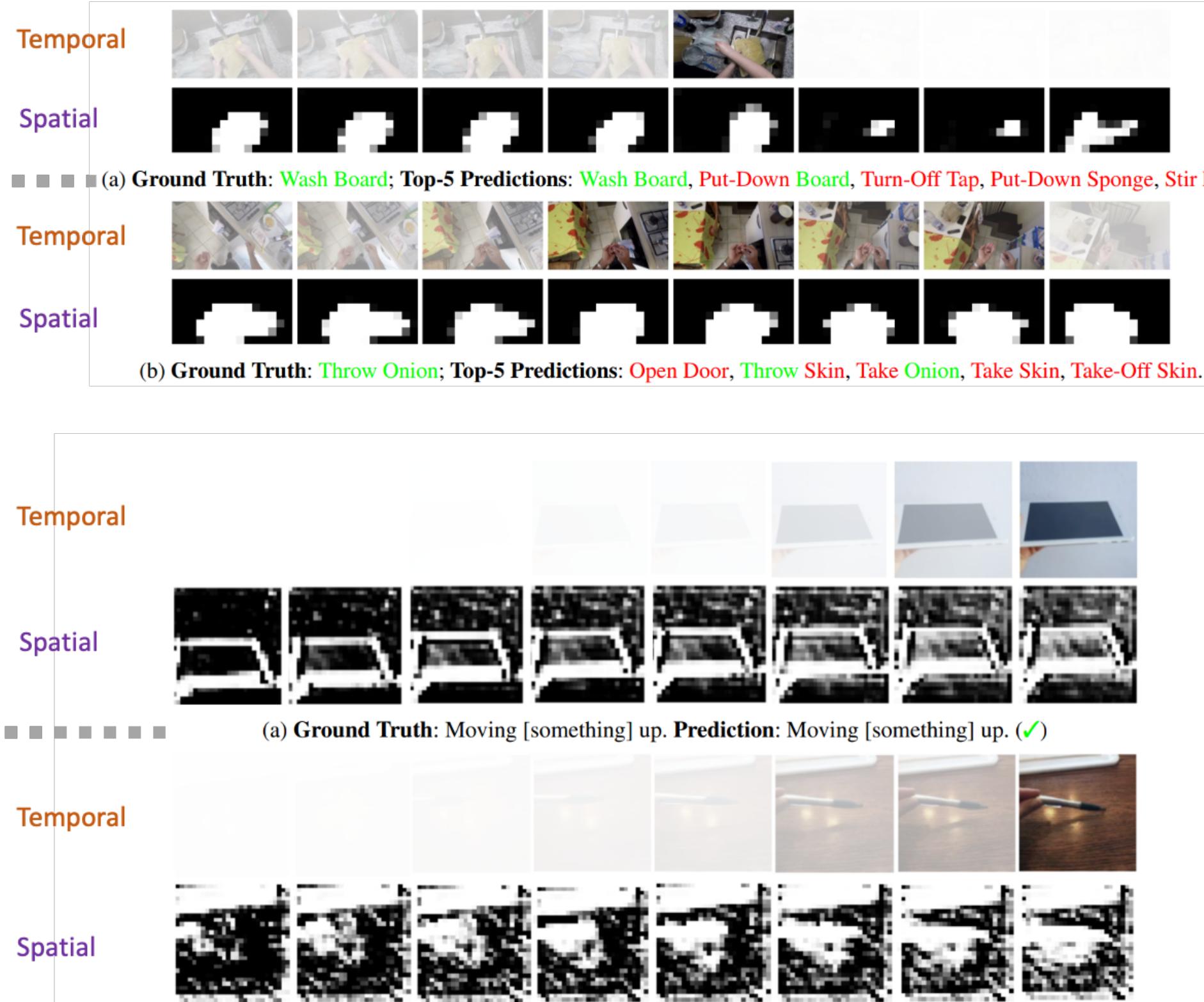
Higher Order Recurrent Space-Time Transformer for Video Action Prediction
 Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Oswald Lanz, 2021



- Spatial Filtering
 - $f_\chi(X) = \text{sigmoid}(\theta_\chi * [X_{avg}; X_{max}] + b_\chi)$
- Self-Filtering (X): $f_\chi(X) \cdot X$
- Cross-Filtering (X, Y): $f_\chi(X) \cdot Y$
- Spatial Branch
 - $\hat{Q} = \text{GAP}(\text{cross_filtering}(Q, K))$
 - $S(Q, K) = \text{sigmoid}(\hat{Q}^T K)$
- Temporal Branch
 - $T(Q, K) = \text{softmax}(\frac{\text{self_filtering}(Q)^T \text{self_filtering}(K)}{\sqrt{C}})$
- Spatial-Temporal Attention
 - $STATT(Q, K, V) = (S(Q, K) \cdot T(Q, K)) \cdot V$

Transformer Meets (Higher-Order) Recurrent

Higher Order Recurrent Space-Time Transformer for Video Action Prediction
 Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Oswald Lanz, 2021



SSv2

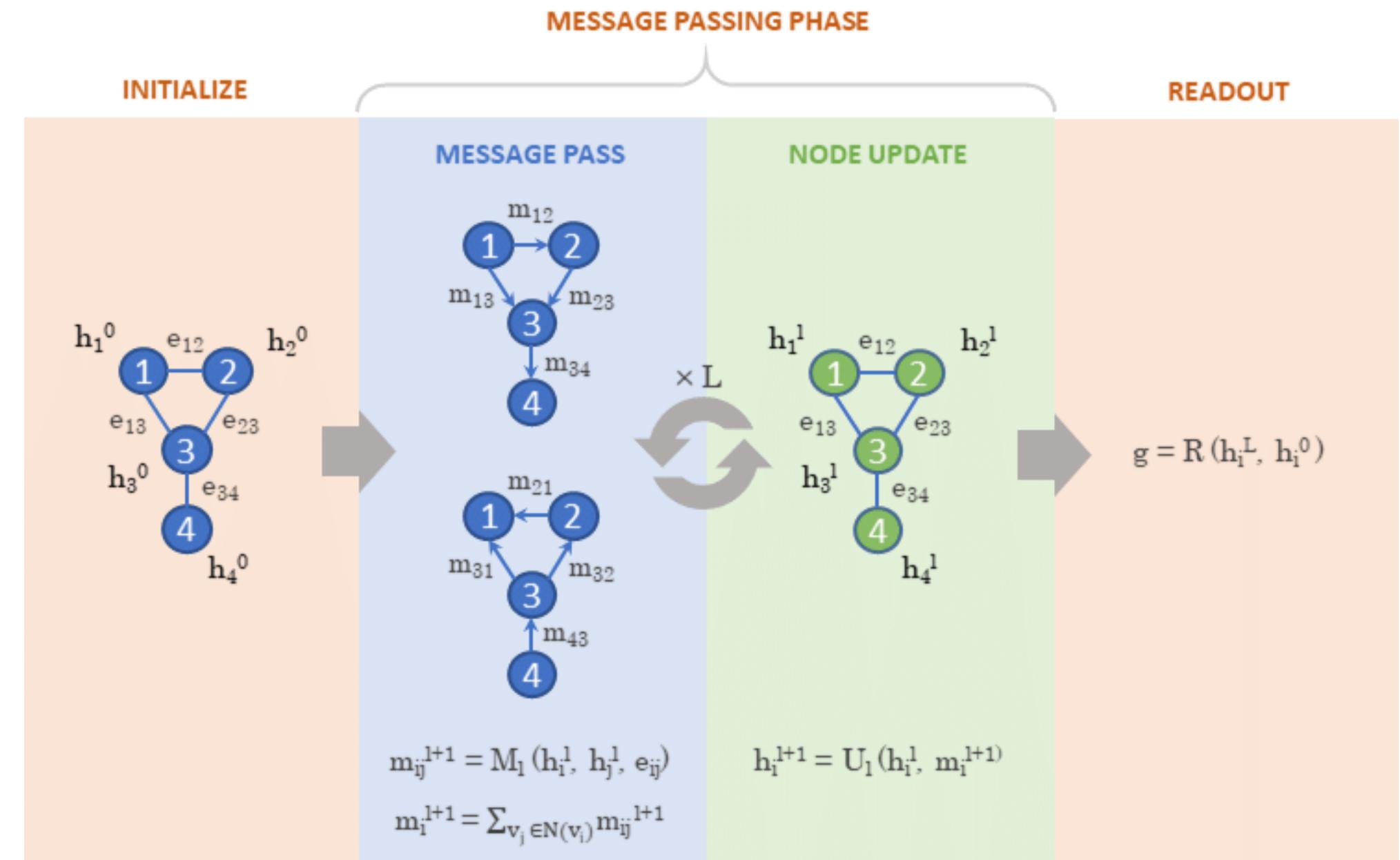
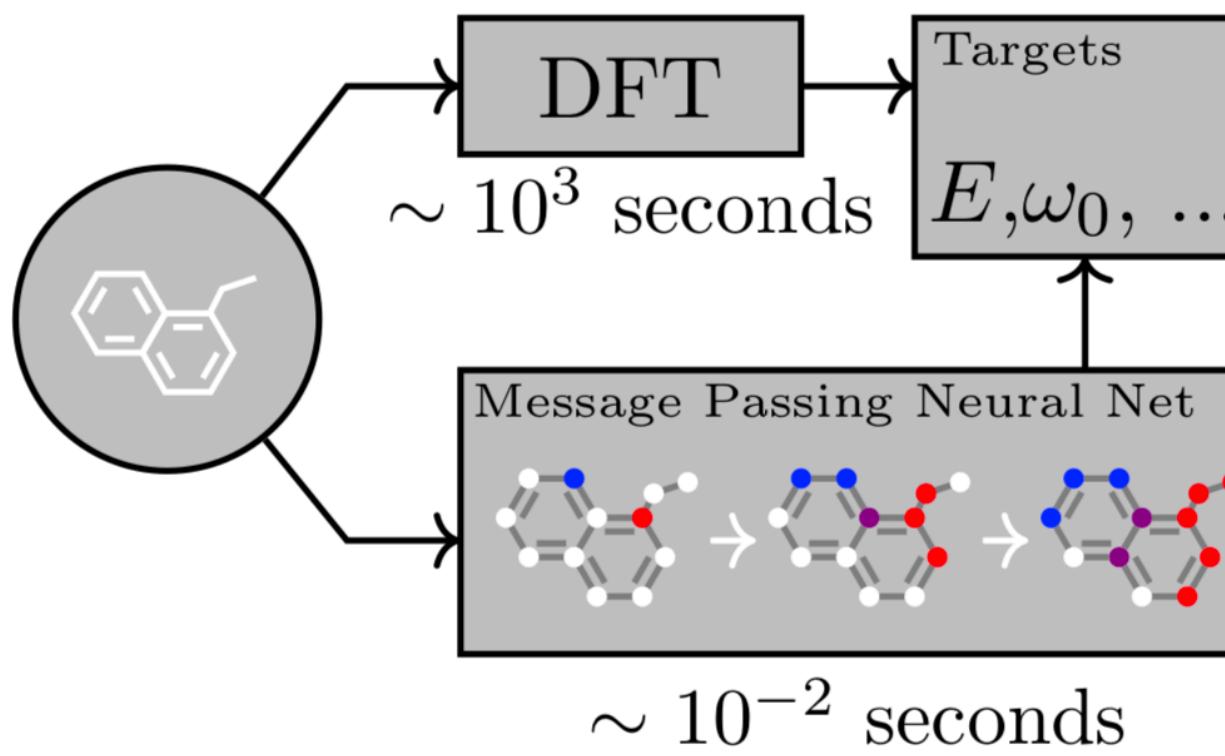
	$S = 1$	$S = 3$	$S = 8$	$S = 12$	$S = 16$
Acc. (%)	30.7	31.2	33.9	33.8	32.7

Attention	#Layers	Acc. (%)
Joint Space-Time [3]	2	30.4
Space-Time Decomposition	2	32.2
Temporal only	4	32.7
Spatial only	4	31.2
Divided Space-Time † [3]	4	25.8
Space-Time Decomposition	4	33.9

EK100

Method	Backbone	Pretrain	Verb	Noun	Action
RU-RGB	<u>BNInc</u>	<u>In1k</u>	27.5	29.0	13.3
AVT-h	<u>BNInc</u>	<u>In1k</u>	27.3	30.7	13.6
AVT-h	AVT-b	In1k	28.2	29.3	13.4
AVT-h	AVT-b	In21k+In1k	28.7	32.3	14.4
AVT-h	AVT-b	In21k	30.2	31.7	14.9
AVT-h	irCSN152	IG65M	25.5	28.1	12.8
HORST	<u>BNInc</u>	<u>In1k</u>	26.2	29.5	14.1
HORST	ConvNeXt	<u>In1k</u>	28.9	33.0	15.6

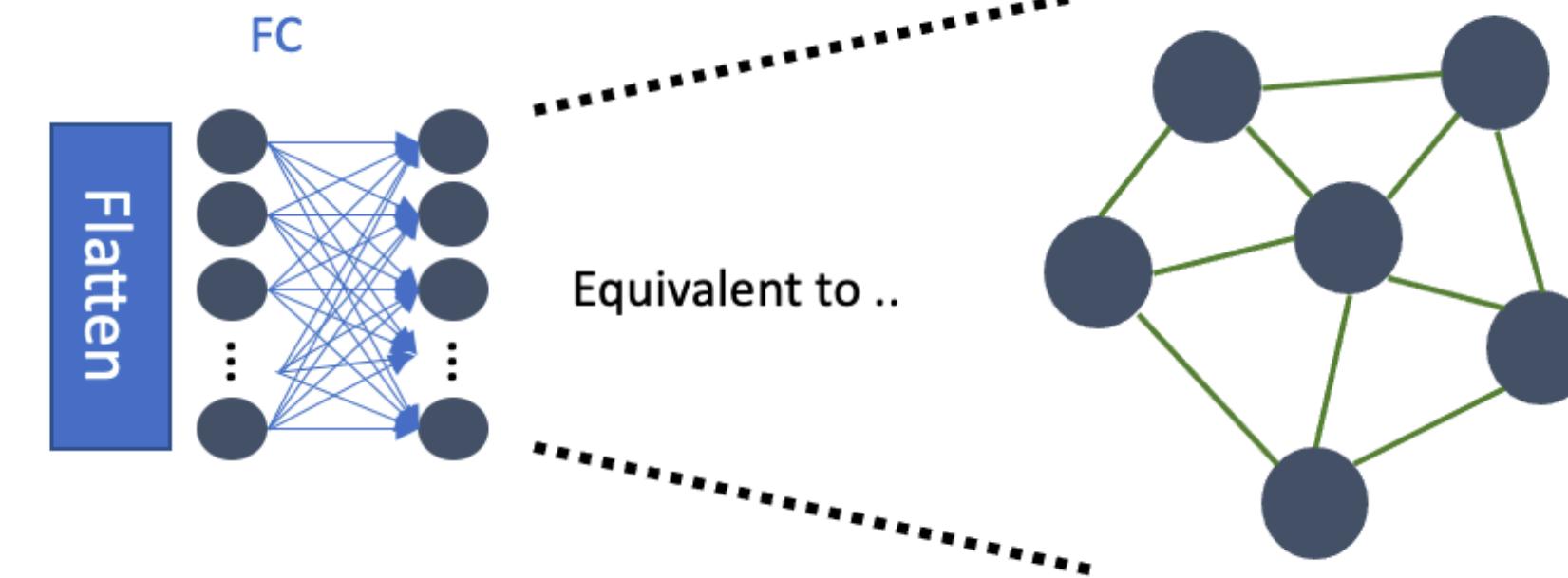
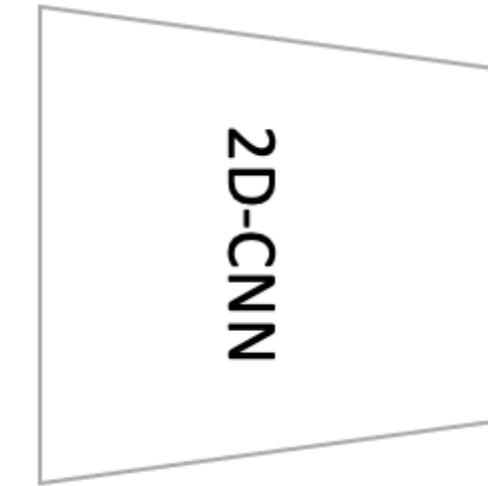
Self-Attention in Message Passing Neural Network



Self-Attention in a Graph Learning

Unified Recurrence Modeling for Video Action Anticipation

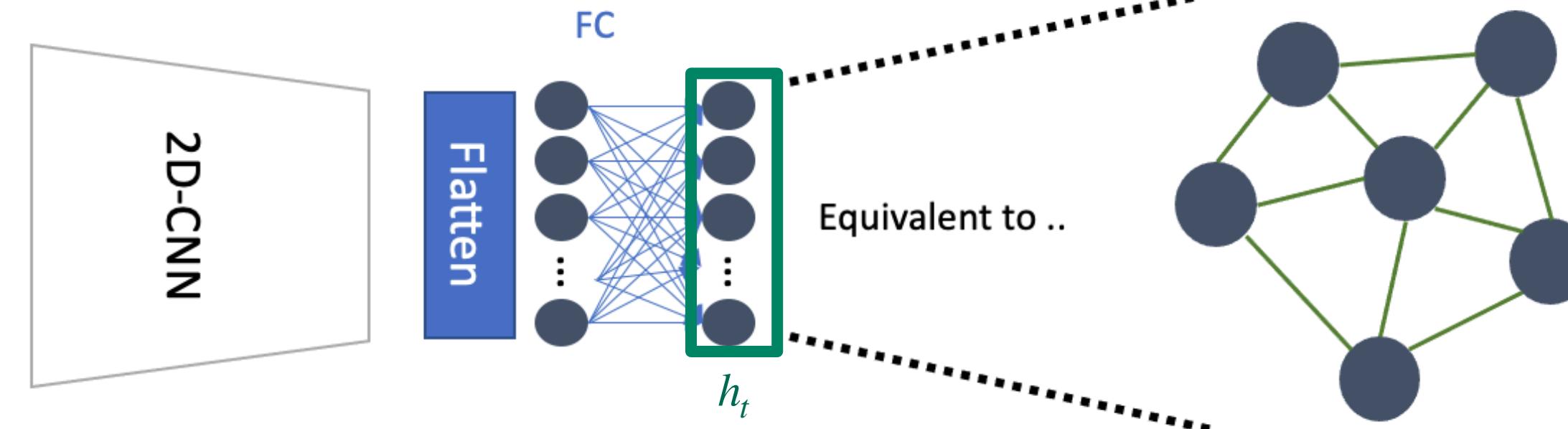
Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Simon See, Oswald Lanz, 2022



Self-Attention in a Graph Learning

Unified Recurrence Modeling for Video Action Anticipation

Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Simon See, Oswald Lanz, 2022



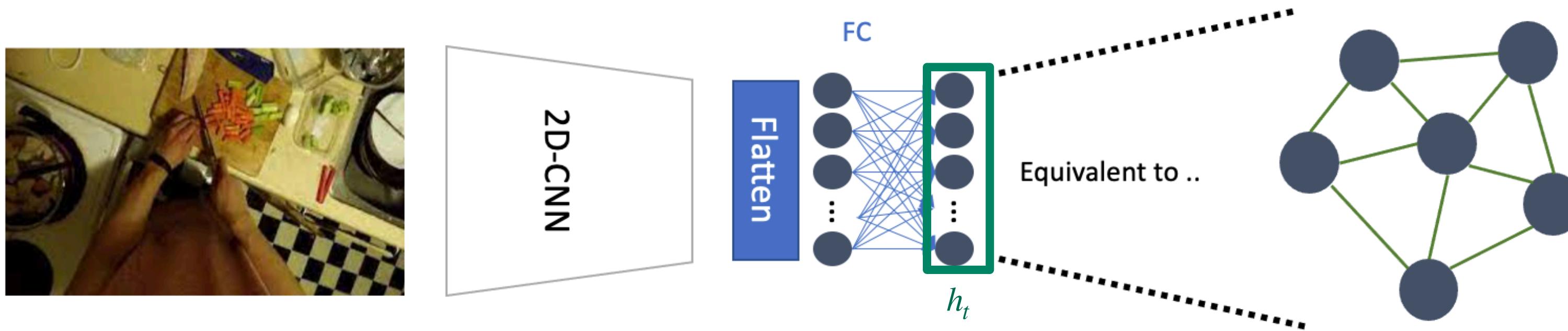
The tokenize representation.

It can be serves as Q, K, V in self-attention :)

Self-Attention in a Graph Learning

Unified Recurrence Modeling for Video Action Anticipation

Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Simon See, Oswald Lanz, 2022

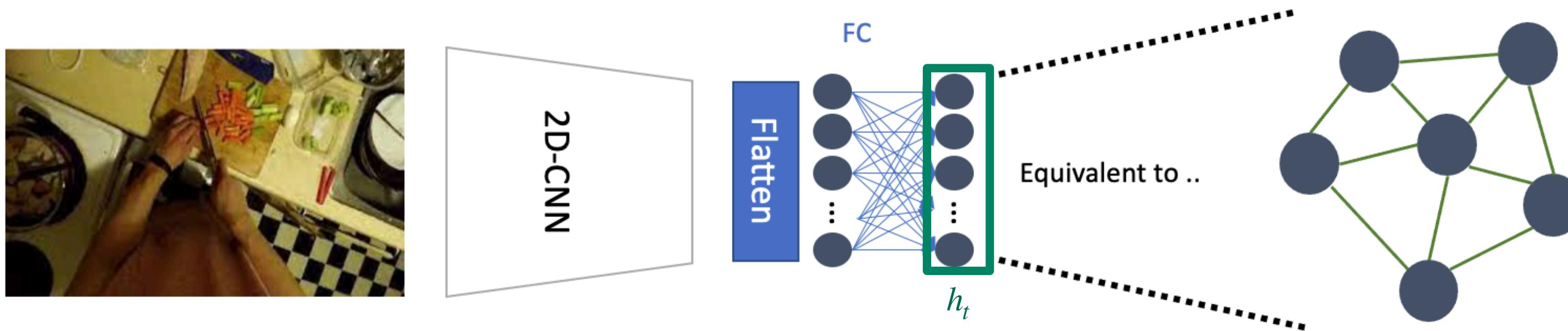


- We adopted Message-Passing Framework (Gilmer, et al 2017) for graph:

- Message function M_t : $m^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_v w)$
- Update function U_t : $h_v^{t+1} = U_t(h_v^t, m^{t+1})$
- Readout function R : $y^{t+1} = R(\{h_v^{t+1}; v \in G\})$

Self-Attention in a Graph Learning

Unified Recurrence Modeling for Video Action Anticipation
Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Simon See, Oswald Lanz, 2022



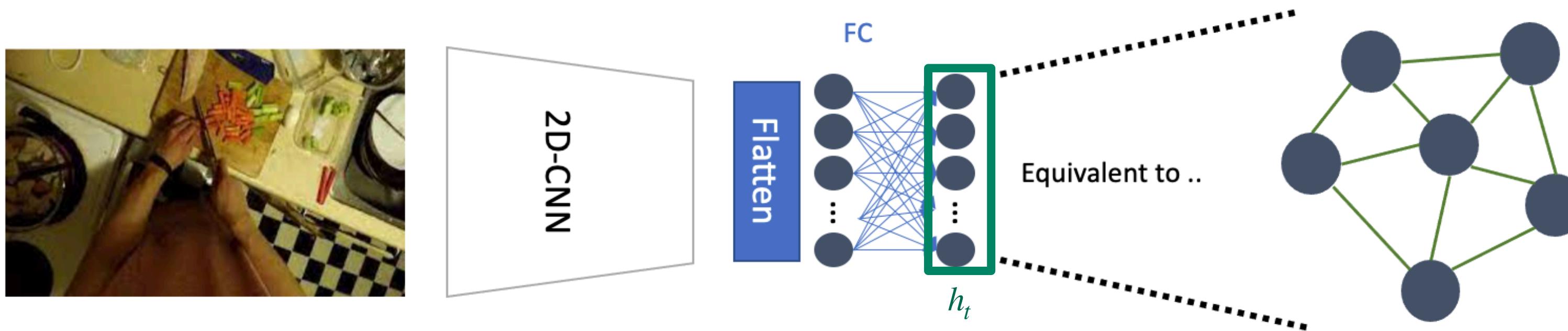
- We adopted Message-Passing Framework (Gilmer, et al 2017) for graph:

- Message function M_t : $m^{t+1} = \sum_{v \in N(v)} M_t(h_v^t, h_w^t, e_v w)$ Generalize $M(\cdot)$ by attention ($Q = h_t, K = h_t, V = h_t$)
- Update function U_t : $h_v^{t+1} = U_t(h_v^t, m^{t+1})$
- Readout function R : $y^{t+1} = R(\{h_v^{t+1}; v \in G\})$

Self-Attention in a Graph Learning

Unified Recurrence Modeling for Video Action Anticipation

Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Simon See, Oswald Lanz, 2022



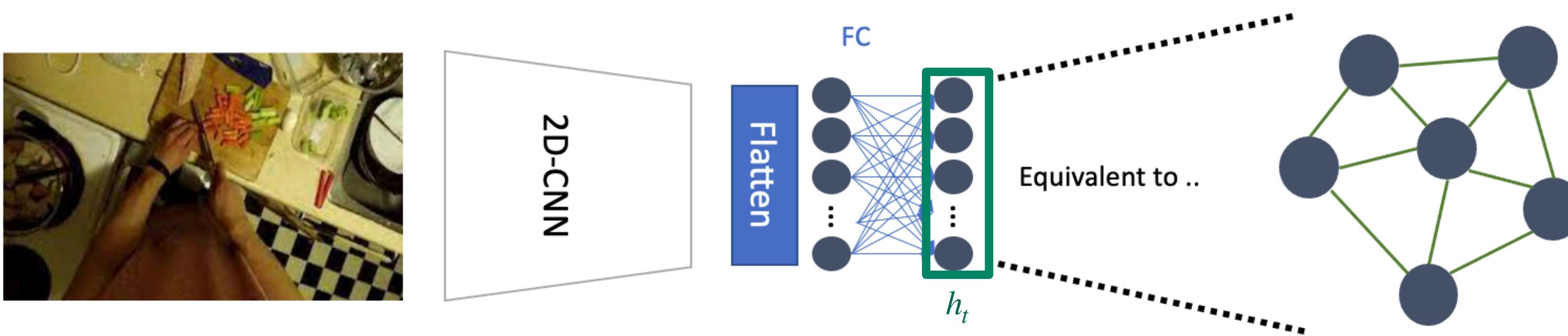
- We adopted Message-Passing Framework (Gilmer, et al 2017) for graph:

- Message function M_t : $m^{t+1} = \sum_{v \in N(v)} M_t(h_v^t, h_w^t, e_v w)$ Generalize $M(.)$ by attention ($Q = h_t, K = h_t, V = h_t$)
- Update function U_t : $h_v^{t+1} = U_t(h_v^t, m^{t+1})$ Generalize $U(.), R(.)$ by attention as well.
- Readout function R : $y^{t+1} = R(\{h_v^{t+1}; v \in G\})$

Self-Attention in a Graph Learning

Unified Recurrence Modeling for Video Action Anticipation

Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Simon See, Oswald Lanz, 2022

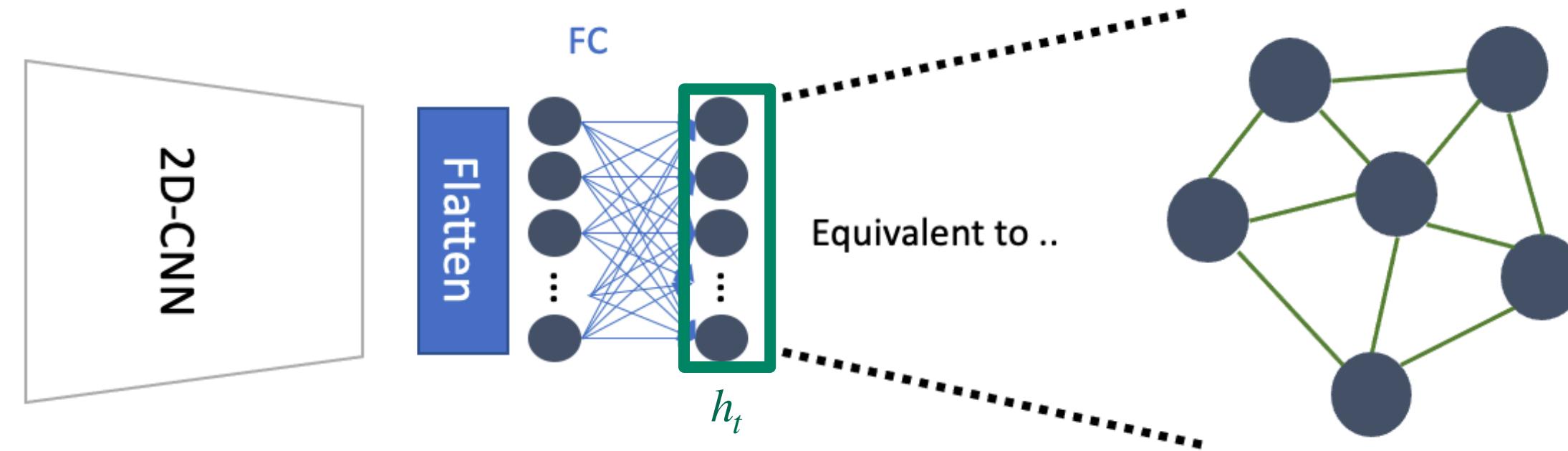


- We adopted Message-Passing Framework (Gilmer, et al 2017) for graph:
 - Message function M : $m_{t+1} = M(h_t)$
 - Update function U : $h_{t+1} = U(h_t, m_{t+1})$
 - Readout function R : $y_{t+1} = R(h_{t+1})$

Self-Attention in a Graph Learning

Unified Recurrence Modeling for Video Action Anticipation

Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Simon See, Oswald Lanz, 2022

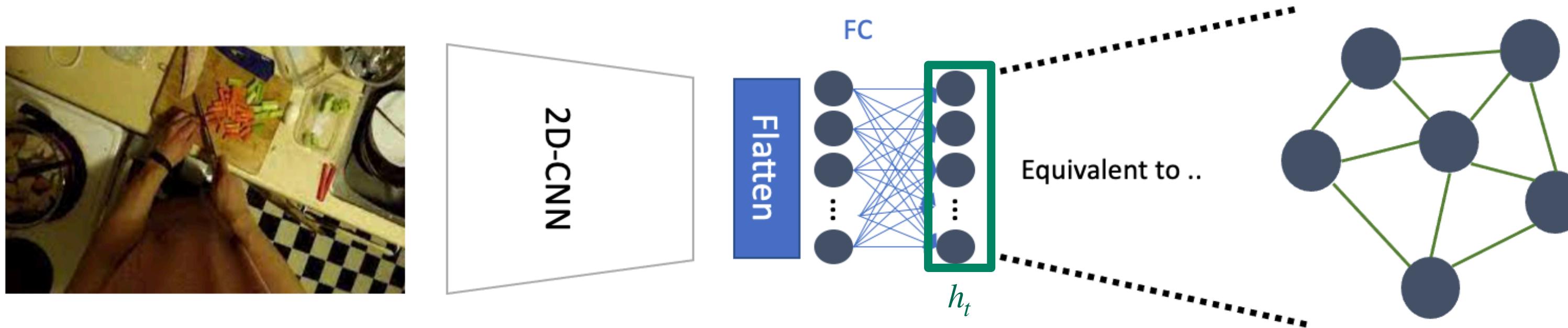


- We adopted Message-Passing Framework (Gilmer, et al 2017) for graph:
 - Message function M : $m_{t+1} = M(h_t)$
 - Update function U : $h_{t+1} = U(h_t, m_{t+1})$
 - Readout function R : $y_{t+1} = R(h_{t+1})$
- **How about the edges?**

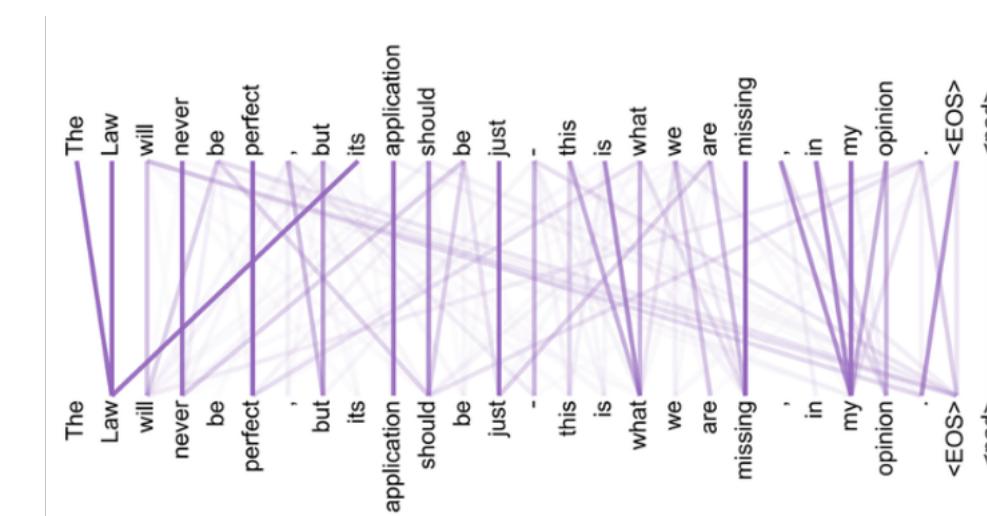
Self-Attention in a Graph Learning

Unified Recurrence Modeling for Video Action Anticipation

Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Simon See, Oswald Lanz, 2022



- We adopted Message-Passing Framework (Gilmer, et al 2017) for graph:
 - Message function M : $m_{t+1} = M(h_t)$
 - Update function U : $h_{t+1} = U(h_t, m_{t+1})$
 - Readout function R : $y_{t+1} = R(h_{t+1})$
- **How about the edges?**



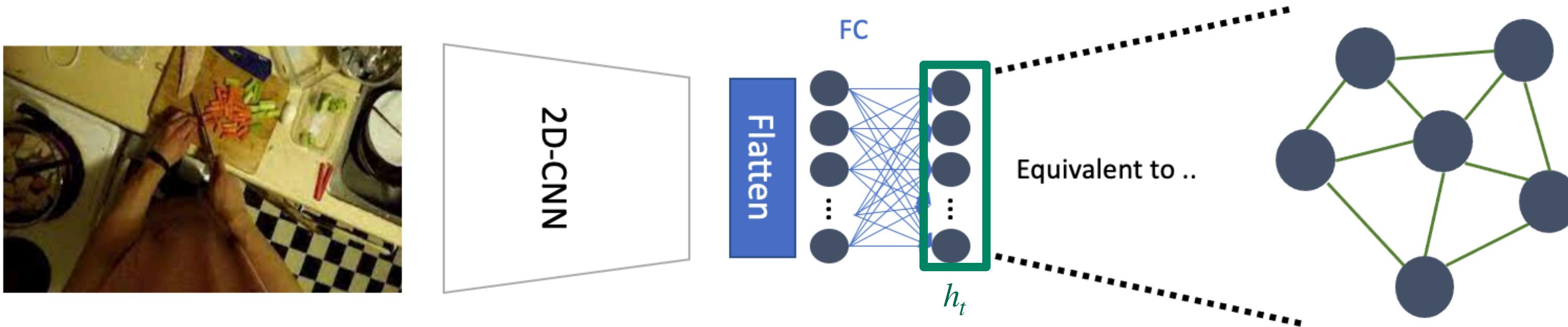
Vaswani et al., 2017

Self-Attention do learn the edge
relationship implicitly by
computing the dot-product score

Self-Attention in a Graph Learning

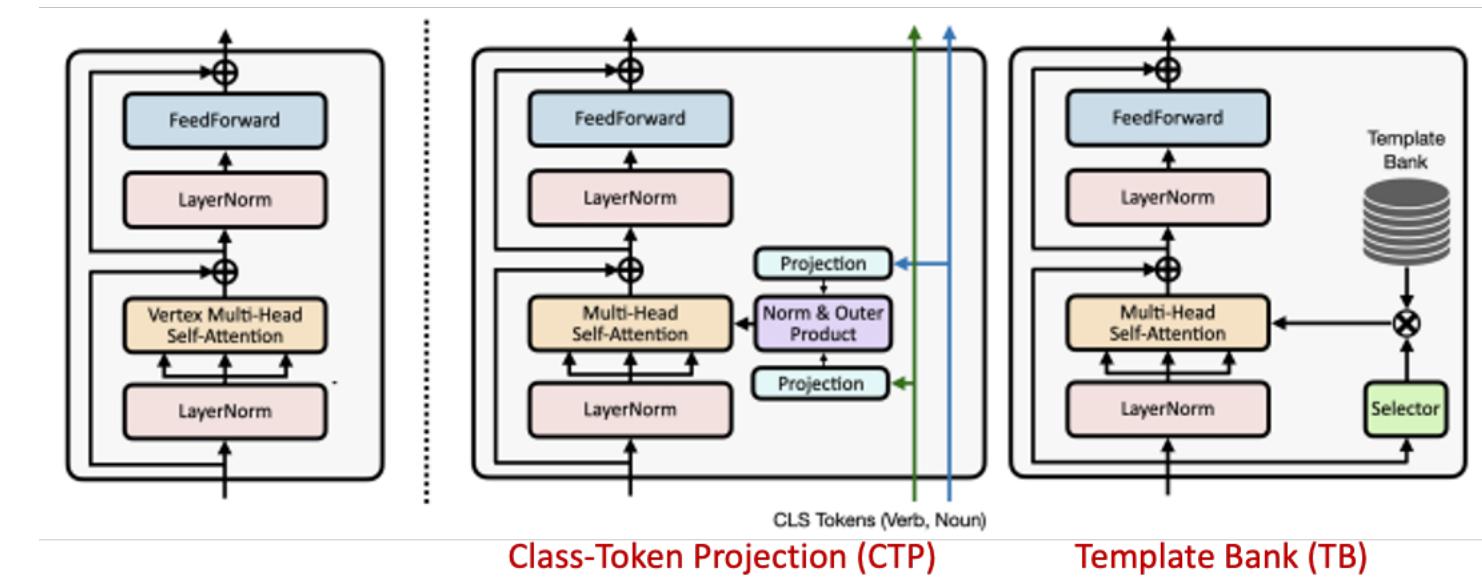
Unified Recurrence Modeling for Video Action Anticipation

Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Simon See, Oswald Lanz, 2022



- We adopted Message-Passing Framework (Gilmer, et al 2017) for graph:

- Message function M : $m_{t+1} = M(h_t, \widehat{\mathbf{A}}_t)$
- Update function U : $h_{t+1} = U(h_t, m_{t+1})$
- Readout function R : $y_{t+1} = R(h_{t+1})$



CTP:

$$V = f_v(CLSS_{verb}^t)$$
$$N = f_n(CLSS_{noun}^t)$$
$$\widehat{\mathbf{A}}^t = V \otimes N$$

TB:

$$w = softmax(f_{select}(x^t))$$

$$\widehat{\mathbf{A}}^t = \sum_{i=0}^s w_i \cdot B_i$$

(Learnable templates)

We achieved 2nd of EPIC-Kitchen Anticipation Challenge and Presented at CVPR'22

NVIDIA-UNIBZ Submission for EPIC-KITCHENS-100 Action Anticipation Challenge 2022

Tsung-Ming Tai, Oswald Lanz, Giuseppe Fiameni, Yi-Kwan Wong, Sze-Sen Poon, Cheng-Kuang Lee, Ka-Chun Cheung, Simon See

Action Anticipation	①	SCUT (hrgdscs)	Zeyu Jiang Changxing Ding	South China University of Technology South China University of Technology
	②	NVIDIA-UNIBZ (corcovadoming)	Tsung-Ming Tai Oswald Lanz Giuseppe Fiameni Yi-Kwan Wong Sze-Sen Poon Cheng-Kuang Lee Ka-Chun Cheung Simon See	NVIDIA, Free University of Bozen-Bolzano Free University of Bozen-Bolzano NVIDIA NVIDIA NVIDIA NVIDIA NVIDIA NVIDIA
	③	ICL-SJTU (Shawn0822)	Xiao Gu Yao Guo Zeju Li Jianing Qiu Benny Lo Guang-Zhong Yang	Imperial College London Shanghai Jiao Tong University Imperial College London Imperial College London Imperial College London Shanghai Jiao Tong University

Model	MT5R (%)
(a) HORST Family with all modalities	17.47
(b) MPNNEL Family with all modalities	18.19
(a) + (b)	19.52
(a) + (b) and weightings 1.2x on all RGB models	19.61

We achieved 2nd place in the EPIC-Kitchen competition for Video Action Anticipation by ensembling both modelings!

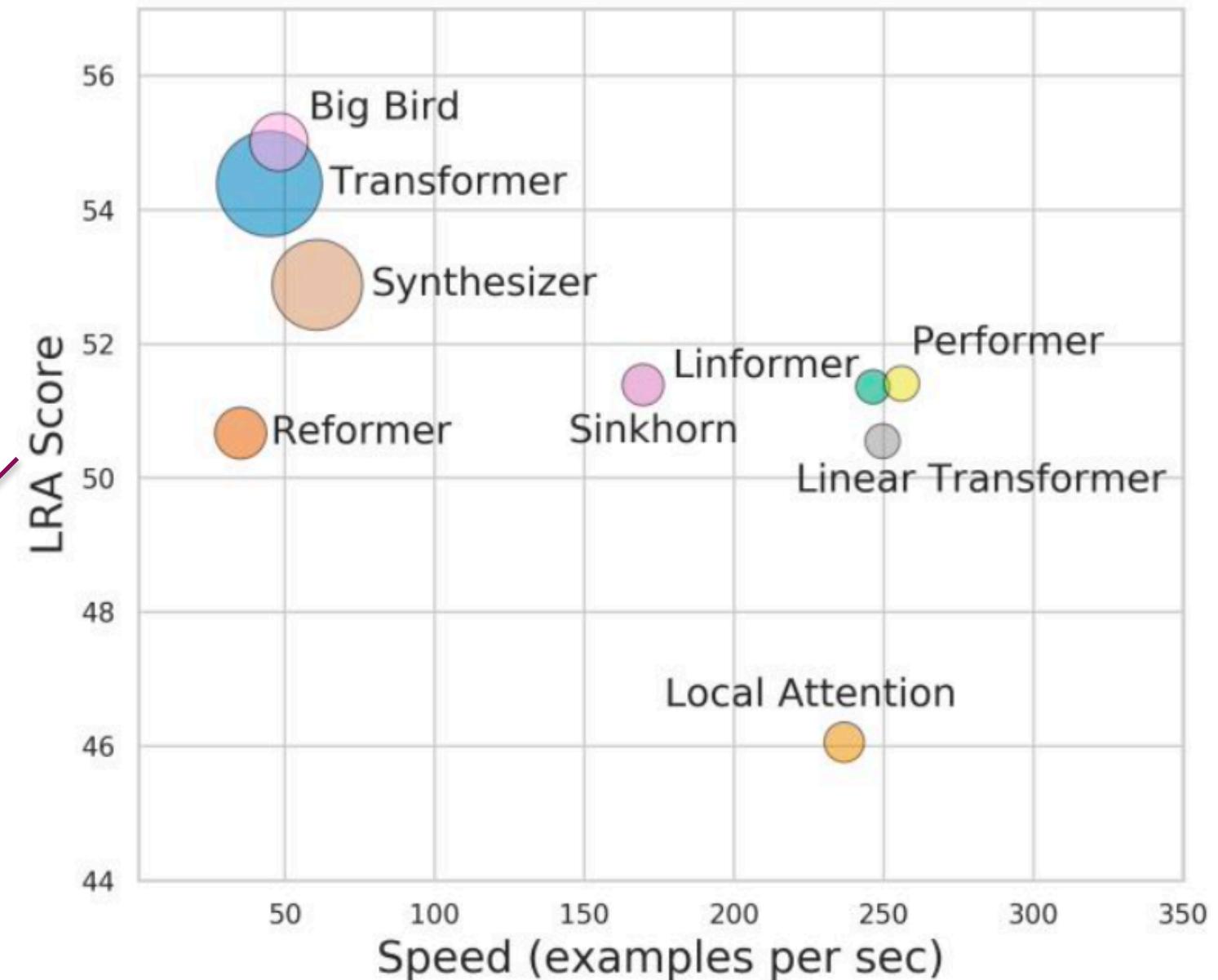
Open Topics in Transformers

Efficiency

Transformer is computational heavy

- The complexity of self-attention is $O(n^2)$
 - Where n presents the numbers of tokens
- It is too heavy for
 - Processing whole articles
 - Full videos
 - High resolution images
- Current methods provide trade-off between speed and quality. No clear solution so far.

Can be interpreted as effective receptive field



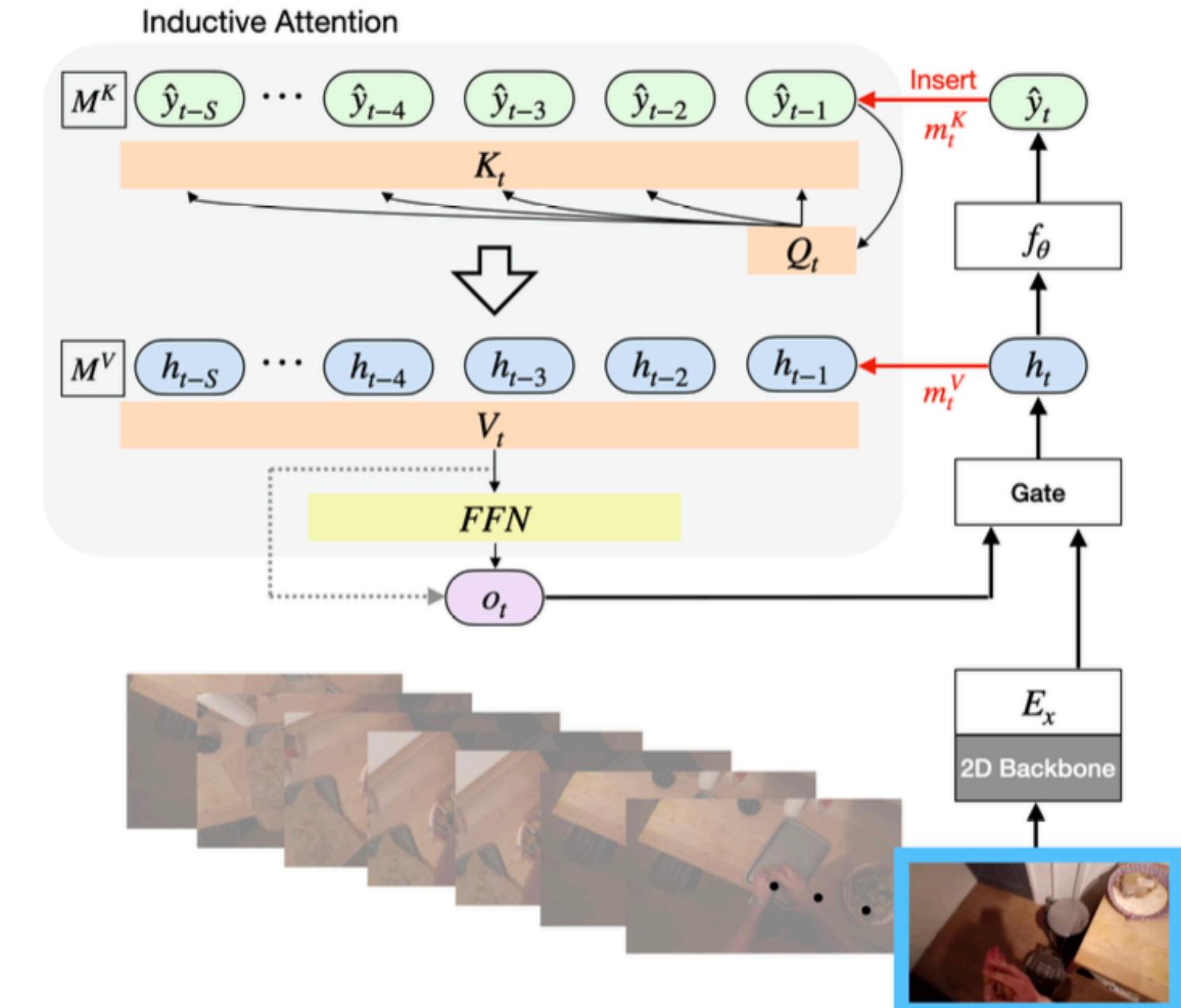
About Query

What is the suitable query choice?

- Query triggers the transformer to pay attention on the meaningful contents of inputs.
 - Attention retrieves the content conditional on the query.
 - The choice of query is critical.
- Usually, the query is using the whole context (i.e., on images), or the current observation (i.e., on videos).
- Although the optimal choice of query is under-explored. We provide another direction, which leverages the target as query.

$\text{Attention}(X_t, X_{<t}, X_{<t})$

$\text{InductiveAttention}(\textcolor{red}{Y}_t, \textcolor{red}{Y}_{<t}, X_{<t})$



Tai, et al 2022. (will be released soon)



Top-5 Positive
 squeeze cloth
 wash sink
 unroll cloth
 gather rubbish
 put sink

Top-5 Positive
 wash cloth
 wash top
 unroll cloth
 dry hand
 dry hand
 wash plug

Top-5 Positive
 wash cloth
 unroll cloth
 dry hand
 wash hand
 wash top

Top-5 Positive
 squeeze cloth
 unroll cloth
 wash plug
 pull cup
 hang hand

Top-5 Positive
 wash cloth
 wash sink
 dry hand
 take cloth
 put cloth

Top-5 Positive
 squeeze cloth
 wash plug
 wash cloth
 hang hand
 pull cup

Top-5 Positive
 squeeze cloth
 wash cloth
 hang hand
 pull cup
 squeeze sponge

Top-5 Negative
 apply bowl
 gather broccoli
 insert bean:green
 check pie
 close freezer

Top-5 Negative
 let-go sponge
 mix lettuce
 eat pork
 check box
 soak plate

Top-5 Negative
 gather meat
 gather broccoli
 eat broccoli
 wash food
 insert pan:dust

Top-5 Negative
 sort broccoli
 eat pork
 coat oil
 brush dough
 eat broccoli

Top-5 Negative
 throw pear
 put sheets
 pat omelette
 unroll omelette
 move omelette

Top-5 Negative
 roll omelette
 wrap box
 eat broccoli
 turn ring:onion
 pat noodle

Top-5 Negative
 roll lemon
 pat omelette
 eat pork
 turn-on liquid:washing
 roll omelette

wash sink
 squeeze cloth
 wash top
 wash cloth
 squeeze sponge

Top-5 Positive
 wash top
 wash cloth
 throw food
 take cloth
 throw rubbish

Top-5 Positive
 squeeze cloth
 unroll cloth
 wash oven
 shake hand
 gather rubbish

Top-5 Positive
 squeeze cloth
 wash sink
 wash floor
 drink beer
 remove rubbish

Top-5 Positive
 wash sink
 wash cloth
 squeeze sponge
 wash top
 wash sponge

Top-5 Positive
 squeeze cloth
 squeeze sponge
 wash top
 wash oven
 put sponge

Top-5 Positive
 wash sink
 dry hand
 take cloth
 wash oven
 gather rubbish

Top-5 Positive
 wash sink
 wash top
 dry hand
 wash oven
 wash hand

Top-5 Negative
 shake rubbish
 wash plug
 shake straw
 dry cloth
 hang hand

Top-5 Negative
 peel squash
 insert nut
 take nut
 cut apple
 look bag

Top-5 Negative
 insert banana
 peel sausage
 unroll omelette
 cut banana
 hang hand

Top-5 Negative
 shake peach
 squeeze garlic
 carry bag
 wash straw
 squeeze caper

Top-5 Negative
 dry scissors
 search salt
 carry bag
 wash straw
 hang cup

Top-5 Negative
 fill liquid
 hang hand
 squeeze can
 shake liquid
 shake courgette

Top-5 Negative
 adjust plate
 dry sponge
 move funnel
 empty container
 filter plate

wash sink
squeeze cloth
 wash top
 wash cloth
 squeeze sponge

Top-5 Positive
 wash top
 wash cloth
 squeeze cloth
 squeeze sponge
 take cloth

Top-5 Positive
 wash top
 squeeze sponge
 unroll cloth
 wash cloth
 wash sponge
 take cloth

Top-5 Positive
 wash top
 unroll cloth
 wash oven
 dry hand
 squeeze sponge

Top-5 Positive
 wash sink
 unroll cloth
 pour liquid:washing
 wash hand
 dry hand

Top-5 Positive
 squeeze sponge
 squeeze cloth
 wash sponge
 turn-off tap
 take cloth

Top-5 Positive
 wash top
 squeeze sponge
 wash cloth
 dry hand
 take cloth

Top-5 Positive
 wash top
 wash cloth
 squeeze sponge
 take cloth
 dry hand

Top-5 Negative
 shake straw
 shake rubbish
 throw kale
 hang hand
 move stand

Top-5 Negative
 shake straw
 shake rubbish
 shake caper
 wash plug
 pat napkin

Top-5 Negative
 shake straw
 brush sink
 adjust straw
 shake caper
 take alarm

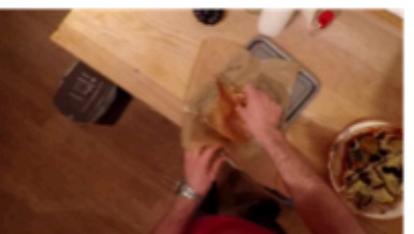
Top-5 Negative
 throw squash
 smell candle
 let-go sponge
 soak plate
 shake caper

Top-5 Negative
 shake straw
 throw squash
 shake rubbish
 wash straw
 drop sponge

Top-5 Negative
 shake straw
 shake rubbish
 brush sink
 move filter
 move coffee

Top-5 Negative
 hang hand
 move coffee
 shake straw
 squeeze caper
 shake rubbish

wash sink
squeeze cloth
wash top
 wash cloth
 squeeze sponge



Top-5 Positive
apply olive
put mushroom
put potato
throw rubbish
throw skin

Top-5 Positive
take pizza
apply olive
apply basil
take basil
open oven

Top-5 Positive
take pizza
apply olive
apply basil
check pizza
take basil

Top-5 Positive
put pizza
flatten dough
apply olive
fold bag
put potato

Top-5 Positive
put pizza
take pizza
apply basil
apply olive
take basil

Top-5 Positive
put pizza
apply olive
open pizza
put fruit
close bin

Top-5 Positive
apply olive
cut pizza
remove pizza
put pin:rolling
put food

Top-5 Negative
look bottle
sort fork
open kale
put parsley
insert oven

Top-5 Negative
check board:chopping
search book
rip wrap:plastic
insert ring:onion
insert cloth

Top-5 Negative
look sauce
remove jar
look can
insert holder
lift dishwasher

Top-5 Negative
turn-on dishwasher
sort fork
check drawer
scrub scissors
look box

Top-5 Negative
eat bread
press coriander
take candle
open corn
look box

Top-5 Negative
pat spatula
insert ring:onion
move omelette
check pie
put pie

(GT: take paper)
Rank 28 of 3806

put pizza
take pizza
insert pizza
cut pizza
remove pizza

Top-5 Positive
put pizza
take pizza
insert pizza
close bin
open pizza

Top-5 Positive
put pizza
insert pizza
check pizza
open pizza
wrap plate

Top-5 Positive
insert pizza
remove pizza
cut pizza
flatten dough
open oven

Top-5 Positive
flatten dough
cut pizza
apply dough
remove pizza
put fruit

Top-5 Positive
insert pizza
flatten dough
cut pizza
apply olive
put fruit

Top-5 Positive
take pizza
insert pizza
cut pizza
put sausage
remove pizza

Top-5 Positive
put pizza
take pizza
insert pizza
drink beer
open pizza

Top-5 Negative
cut corn
pat garlic
press onion
remove stalk
check ginger
check onion

Top-5 Negative
cut corn
press onion
remove stalk
apply onion
sort onion

Top-5 Negative
move pen
put pear
take toaster
eat broccoli
remove cucumber

Top-5 Negative
rip wrap:plastic
divide pizza
unroll clothes
move fridge
dry food

Top-5 Negative
dry food
divide pizza
unroll clothes
move fridge
dry food

Top-5 Negative
remove floor
look bottle
sort broccoli
pour bean:green
brush dough

Top-5 Negative
cut corn
sprinkle tomato
close oregano
put pear
scrape skin

put pizza
take pizza
insert pizza
cut pizza
remove pizza

Top-5 Positive
apply olive
cut pizza
remove pizza
apply basil
take aubergine

Top-5 Positive
take pizza
insert pizza
check pizza
open pizza
open oven

Top-5 Positive
put pizza
insert pizza
check pizza
cut pizza
open pizza

Top-5 Positive
flatten dough
insert pizza
take pizza
fold bag
divide dough

Top-5 Positive
remove pizza
insert pizza
take pizza
cut pizza
apply basil

Top-5 Positive
put pizza
open pizza
close bin
drink beer
remove salad

Top-5 Positive
apply olive
take pizza
cut pizza
put fruit
put food

Top-5 Negative
squeeze garlic
pat burger
squeeze sausage
mix sausage
sort onion

Top-5 Negative
cut corn
press onion
apply onion
sort onion
squeeze garlic

Top-5 Negative
cut corn
press onion
apply onion
sort onion
remove stalk

Top-5 Negative
search plate
lift foil
cut pork
sort pizza
hold tray

Top-5 Negative
cut corn
attach tray
take toaster
rub sauce
shake tomato

Top-5 Negative
cut corn
close pepper
put shell:egg
insert shell:egg
press onion

put pizza
take pizza
insert pizza
cut pizza
remove pizza

Summary

- Due to data scale grows every year, people employs deep neural networks to automatically extract the features. It not only minimizes the efforts for feature engineering, but also enabling an end-to-end trainable pipeline for solving tasks, achieving remarkable achievements in this decade.

Summary

- Due to data scale grows every year, people employs deep neural networks to automatically extract the features. It not only minimizes the efforts for feature engineering, but also enabling an end-to-end trainable pipeline for solving tasks, achieving remarkable achievements in this decade.
- Researchers tailored different deep learning architectures for different tasks. The appearance of transformer model unifying and establishing baselines for various of applications.

Summary

- Due to data scale grows every year, people employs deep neural networks to automatically extract the features. It not only minimizes the efforts for feature engineering, but also enabling an end-to-end trainable pipeline for solving tasks, achieving remarkable achievements in this decade.
- Researchers tailored different deep learning architectures for different tasks. The appearance of transformer model unifying and establishing baselines for various of applications.
- Self-attention, the building block of transformer, is parameter-free and permute-invariant, thus the positional encoding is required.

Summary

- Due to data scale grows every year, people employs deep neural networks to automatically extract the features. It not only minimizes the efforts for feature engineering, but also enabling an end-to-end trainable pipeline for solving tasks, achieving remarkable achievements in this decade.
- Researchers tailored different deep learning architectures for different tasks. The appearance of transformer model unifying and establishing baselines for various of applications.
- Self-attention, the building block of transformer, is parameter-free and permute-invariant, thus the positional encoding is required.
- Every thing you can tokenize, you can feed into the transformer

Summary

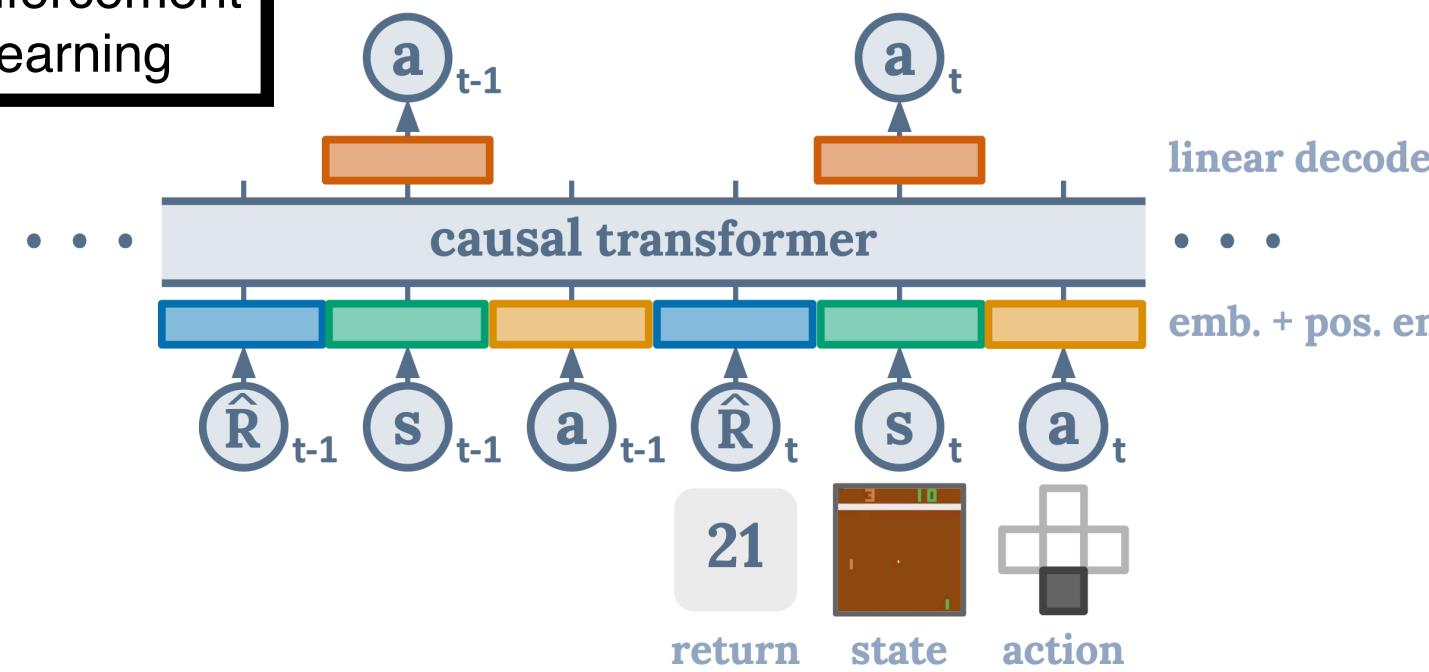
- Due to data scale grows every year, people employs deep neural networks to automatically extract the features. It not only minimizes the efforts for feature engineering, but also enabling an end-to-end trainable pipeline for solving tasks, achieving remarkable achievements in this decade.
- Researchers tailored different deep learning architectures for different tasks. The appearance of transformer model unifying and establishing baselines for various of applications.
- Self-attention, the building block of transformer, is parameter-free and permute-invariant, thus the positional encoding is required.
- Every thing you can tokenize, you can feed into the transformer
- The next challenge in transformer is to efficiently apply on the high dimensional data, e.g., whole articles, full videos, high resolution images.

Summary

- Due to data scale grows every year, people employs deep neural networks to automatically extract the features. It not only minimizes the efforts for feature engineering, but also enabling an end-to-end trainable pipeline for solving tasks, achieving remarkable achievements in this decade.
- Researchers tailored different deep learning architectures for different tasks. The appearance of transformer model unifying and establishing baselines for various of applications.
- Self-attention, the building block of transformer, is parameter-free and permute-invariant, thus the positional encoding is required.
- Every thing you can tokenize, you can feed into the transformer
- The next challenge in transformer is to efficiently apply on the high dimensional data, e.g., whole articles, full videos, high resolution images.
- We introduced some methods combining the transformer and recurrent, which is suitable for the (spatial-temporal) series modeling. We also shown the transformer can be viewed as a graph modeling method.

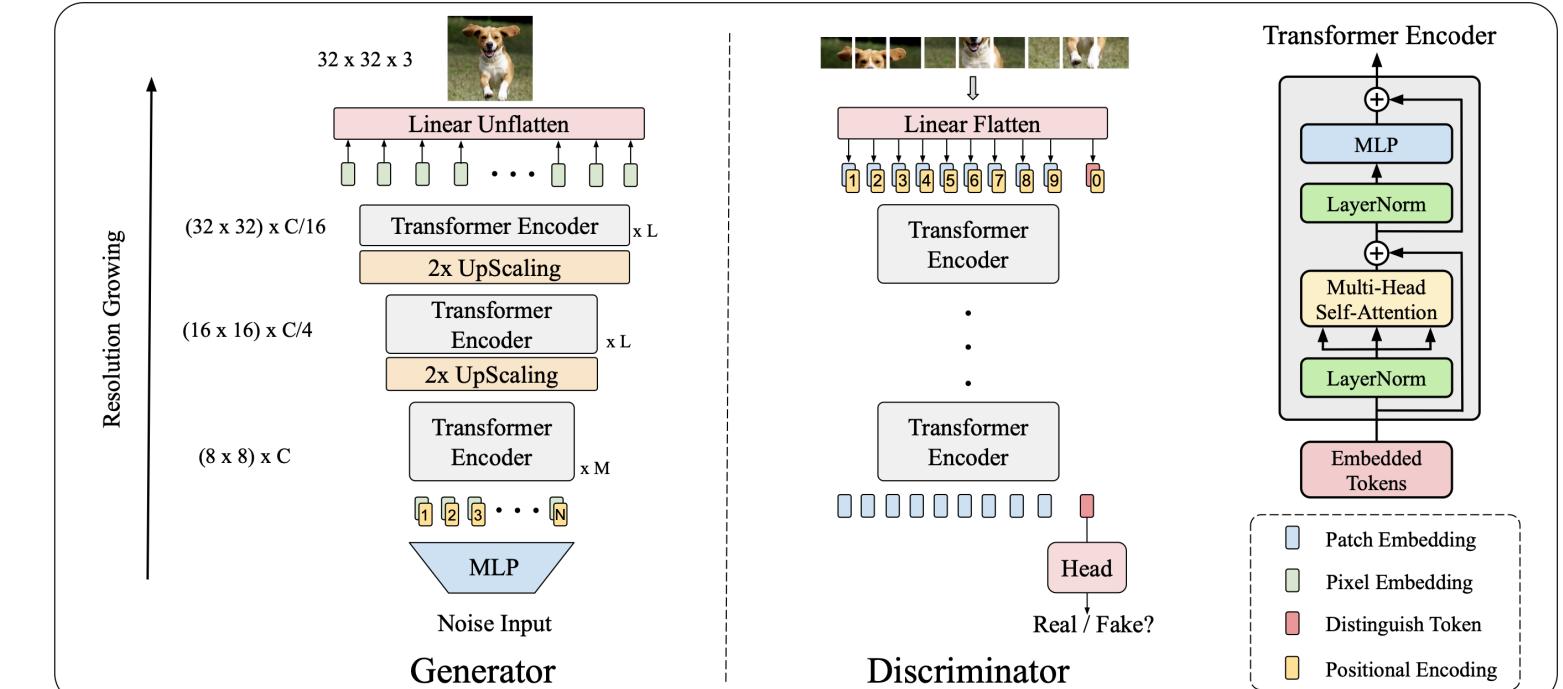
More Transformer Applications

Reinforcement Learning



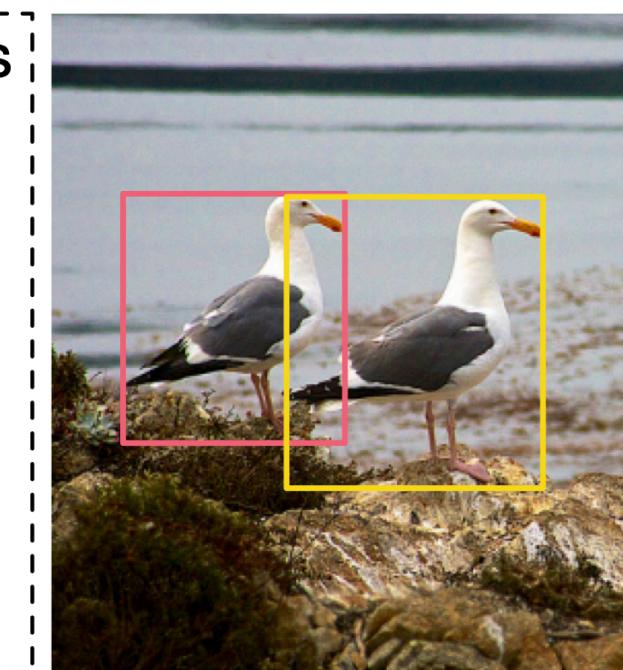
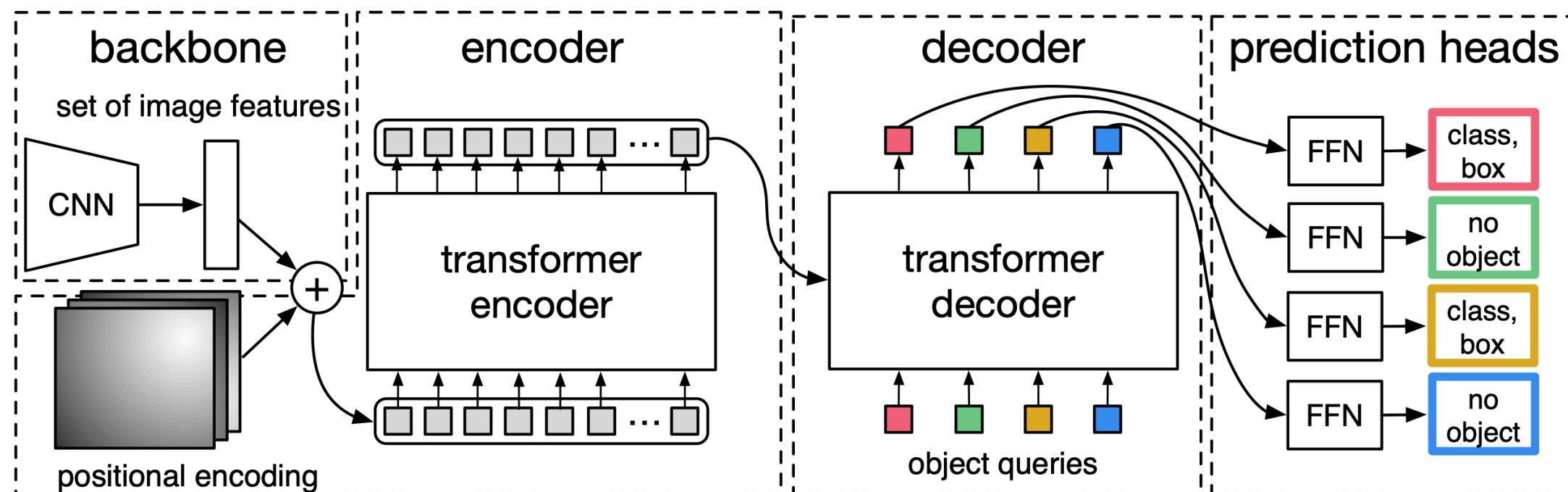
Chen, Lili, et al. "Decision transformer: Reinforcement learning via sequence modeling." *Advances in neural information processing systems* 34 (2021): 15084-15097.

Generative Models



Jiang, Yifan, Shiyu Chang, and Zhangyang Wang. "Transgan: Two transformers can make one strong gan." *arXiv preprint arXiv:2102.07074* 1.3 (2021).

Object Detection



Carion, Nicolas, et al. "End-to-end object detection with transformers." *European conference on computer vision*. Springer, Cham, 2020.

References

- Public tutorial slides of transformer by Lucas Beyer: lucasheyer.be/transformer
- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).
- Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
- Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).
- Dehghani, Mostafa, et al. "Universal transformers." *arXiv preprint arXiv:1807.03819* (2018).
- Dai, Zihang, et al. "Transformer-xl: Attentive language models beyond a fixed-length context." *arXiv preprint arXiv:1901.02860* (2019).
- Tai, Tsung-Ming, et al. "Higher Order Recurrent Space-Time Transformer for Video Early Action Prediction." *2022 IEEE International Conference on Image Processing (ICIP)*.
- Tai, Tsung-Ming, et al. "Unified Recurrence Modeling for Video Action Anticipation." *International Conference on Pattern Recognition (ICPR)*, 2022.
- Tai, Tsung-Ming, et al. "NVIDIA-UNIBZ Submission for EPIC-KITCHENS-100 Action Anticipation Challenge 2022." *arXiv preprint arXiv:2206.10869* (2022).



NVIDIA®

