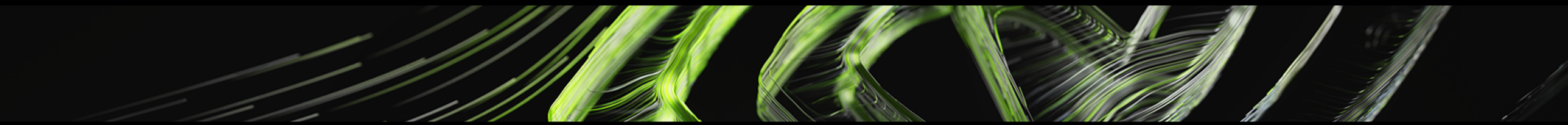


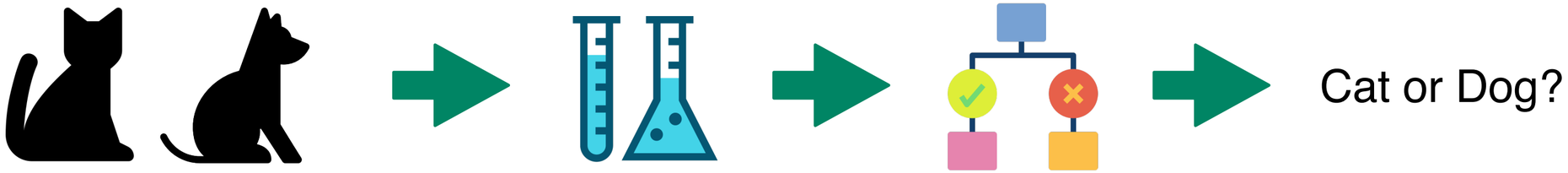
Paradigm Shift in Computer Vision and Beyond: Transformer and Attention

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

Deep Learning

Minimize the efforts in feature engineering

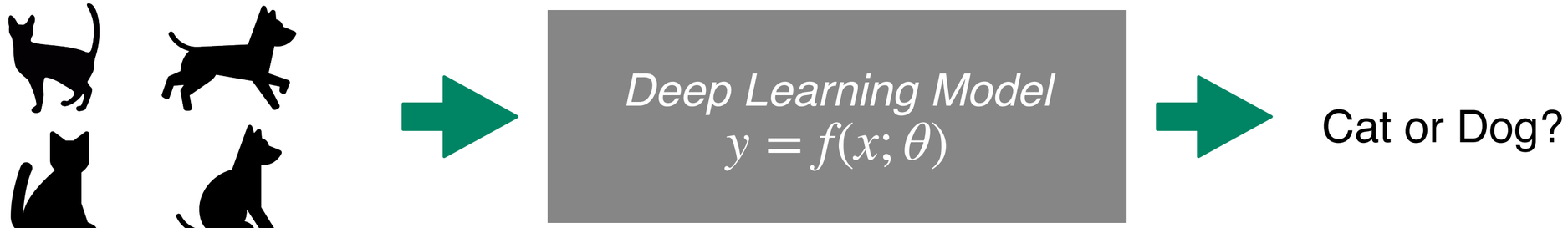
Machine Learning



Input → *Feature Engineer* → *Classifier* → *Result*

Deep Learning (with sufficient data)

A blackbox deep learning model does not imply we don't know anything about it.
Instead, we know every computation and math details of the model.
We just have no much clues about the decision boundary being made.

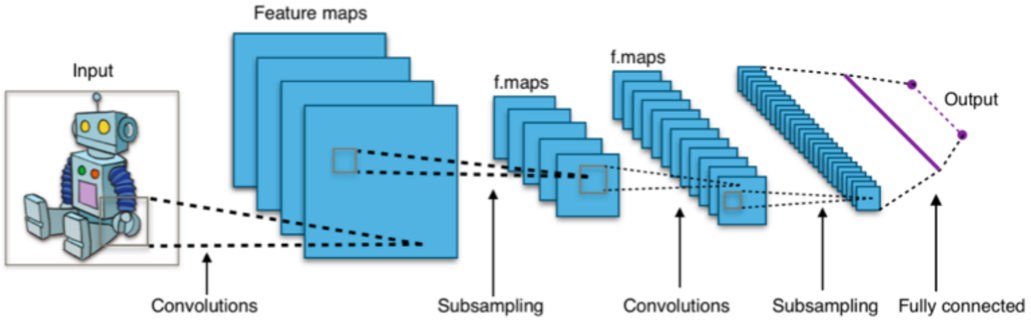


Input → *Deep Learning Model* $y = f(x; \theta)$ → *Result*

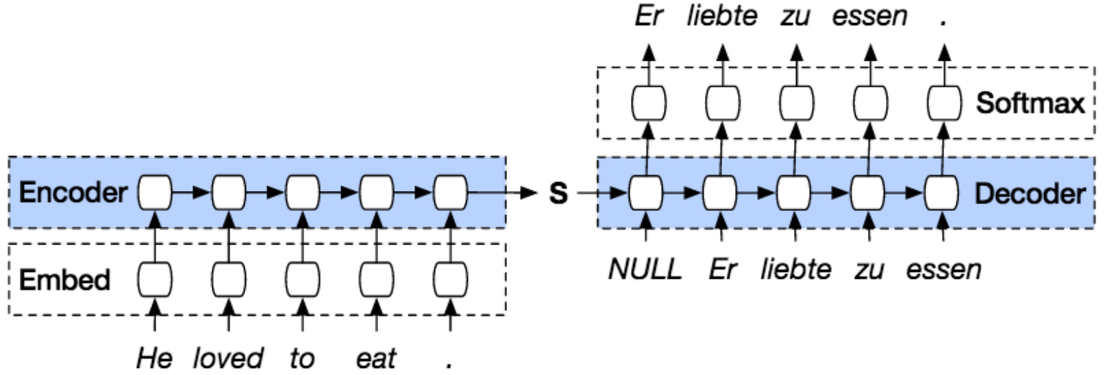
Building Deep Learning Model

Earlier than 2018

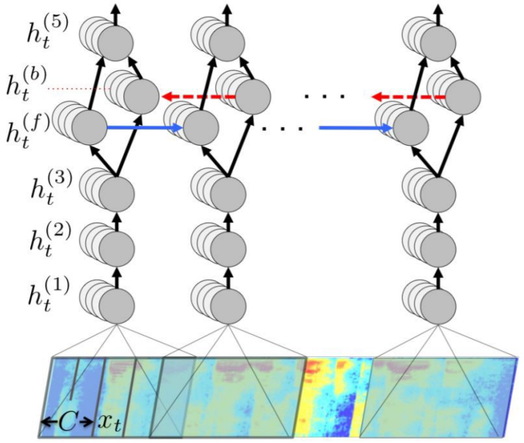
Computer Vision



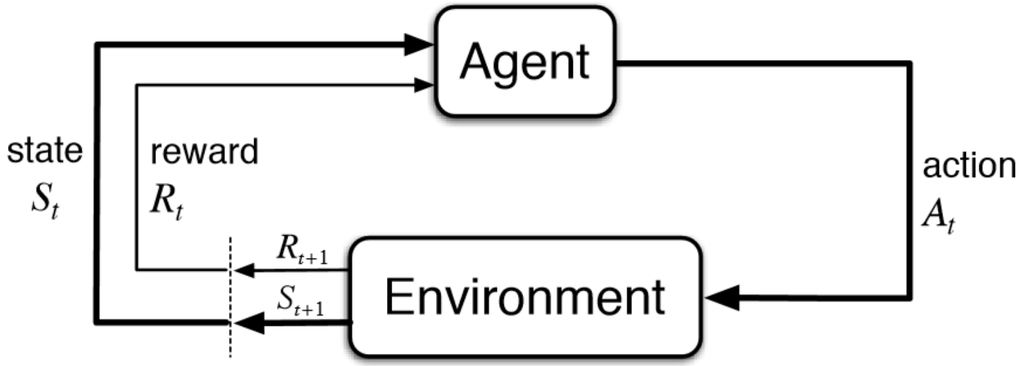
Natural Language Processing



Speech Recognition

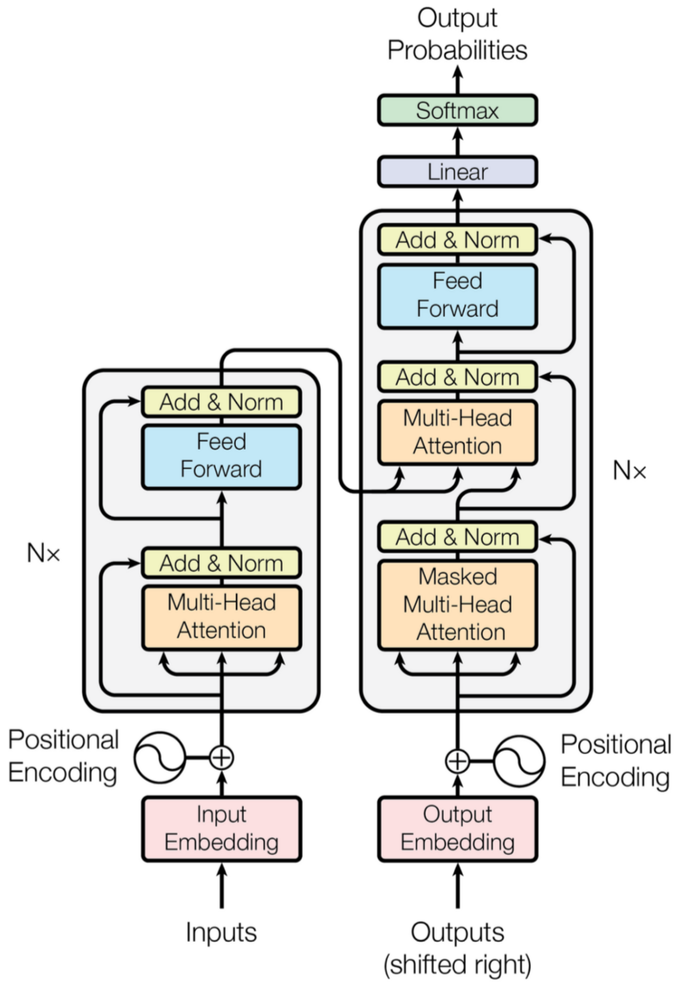


Reinforcement Learning



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

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

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

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

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

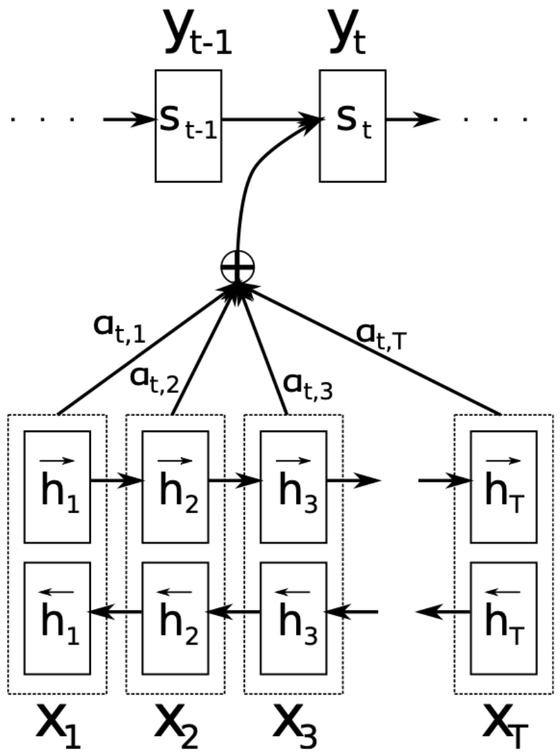
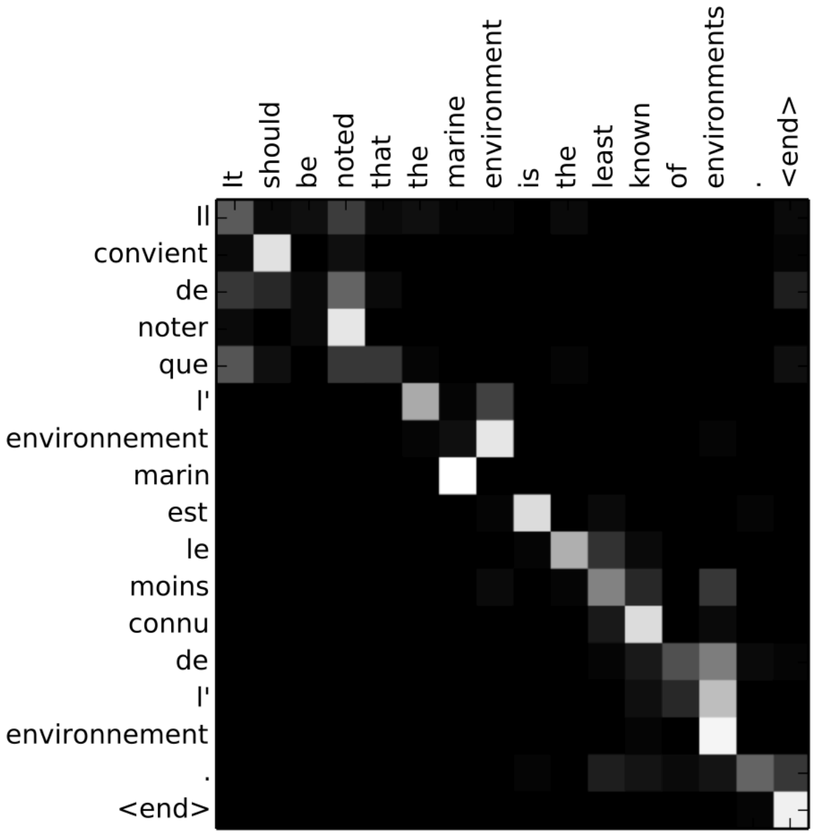


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.



Attention Learns Alignment

Neural Machine Translation by Jointly Learning to Align and Translate

Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, 2014

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

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

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

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

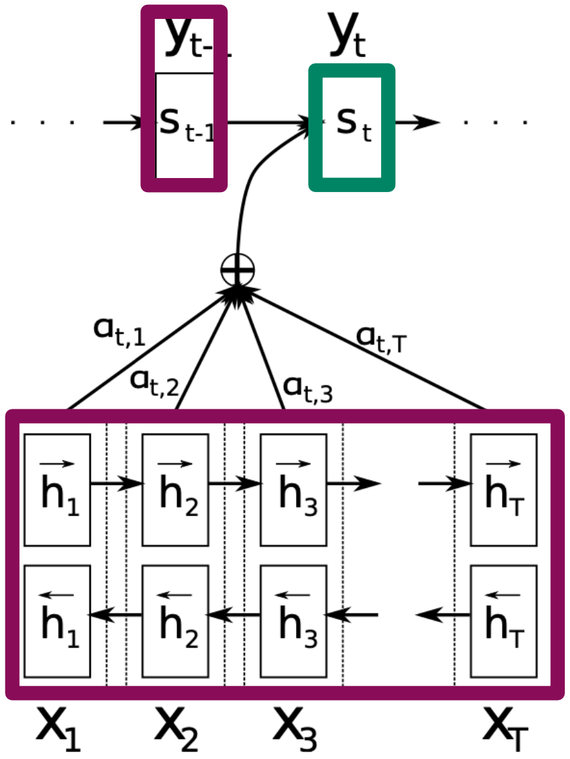
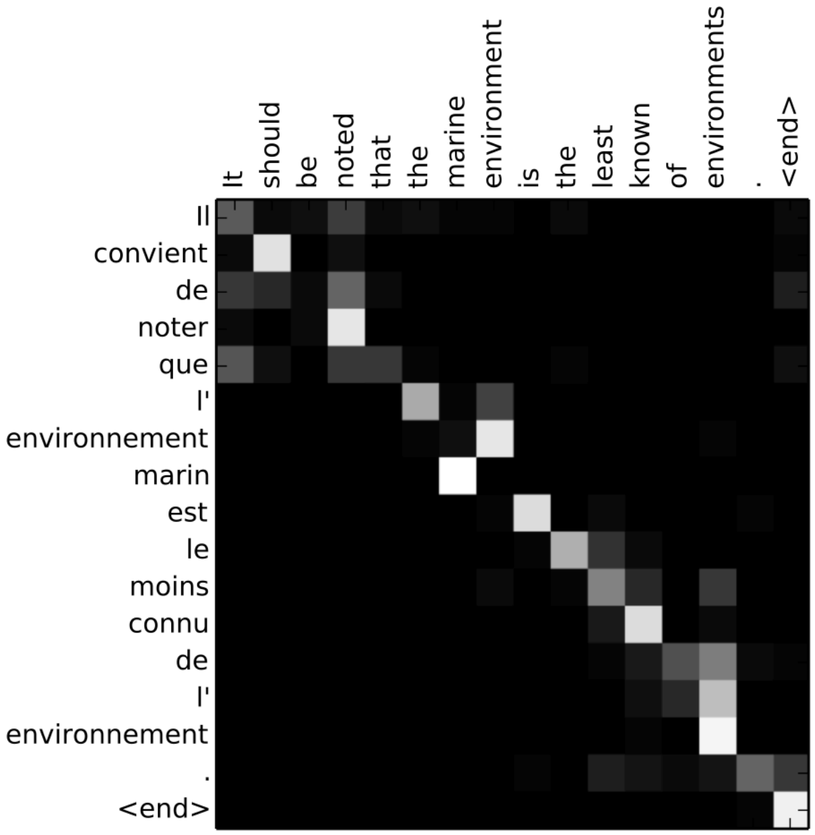


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.



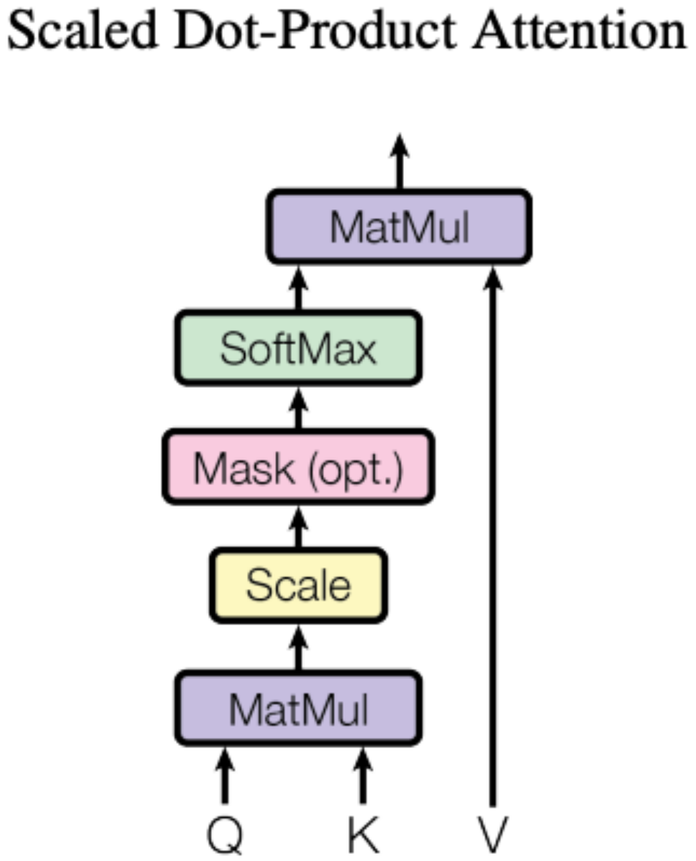
Attention

Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, 2017

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

- 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!



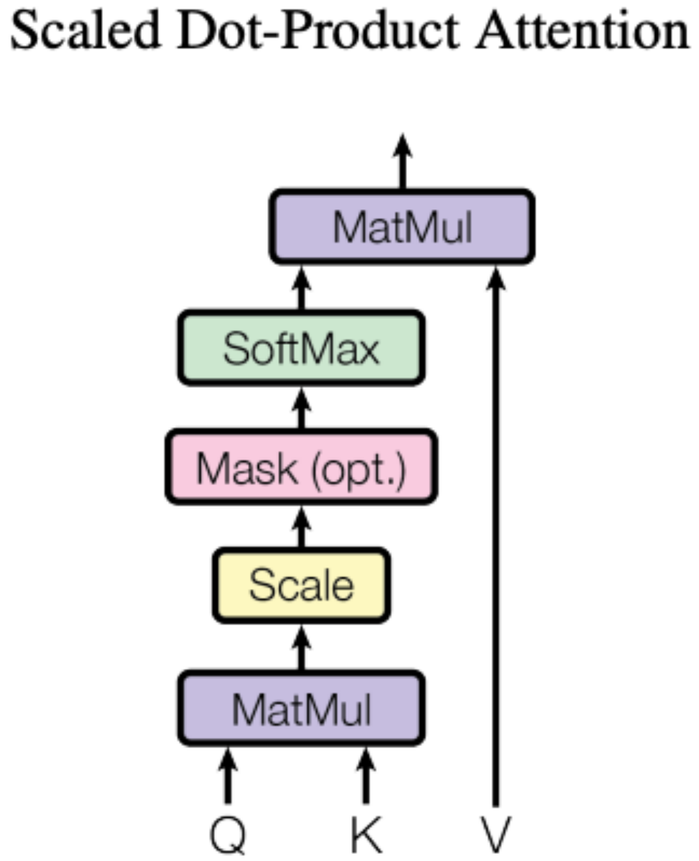
Attention

Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, 2017

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

- 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?**



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

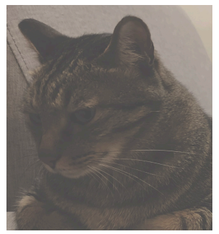
Q (Query)

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

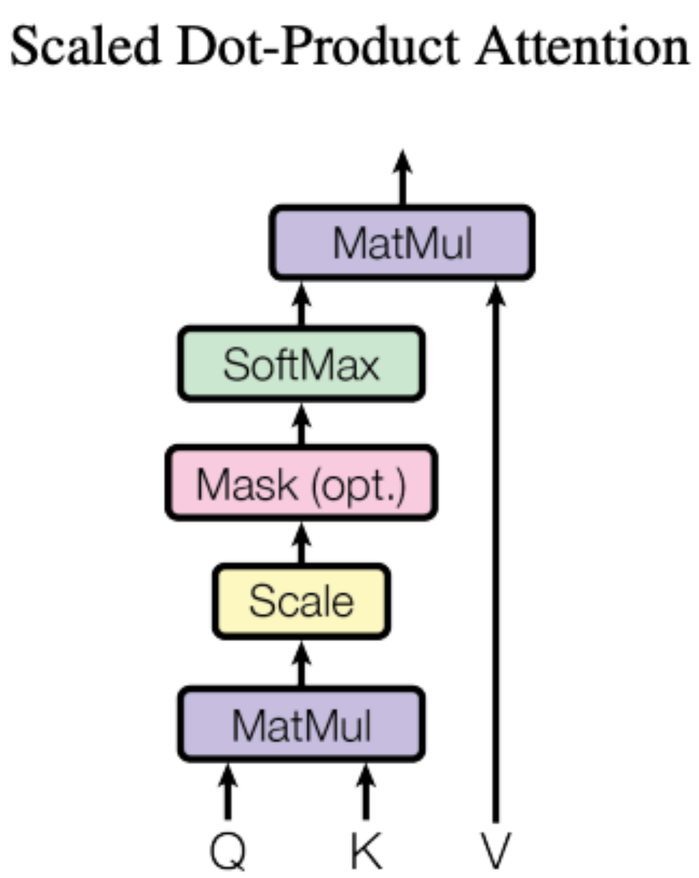
Attention

Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, 2017

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

- 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.
- **Q, K, and V presents the query, keys, and values, with the shape**
  - $Q : (b, L_q, d_k)$
  - $K : (b, L_k, d_k)$
  - $V : (b, L_k, d_v)$



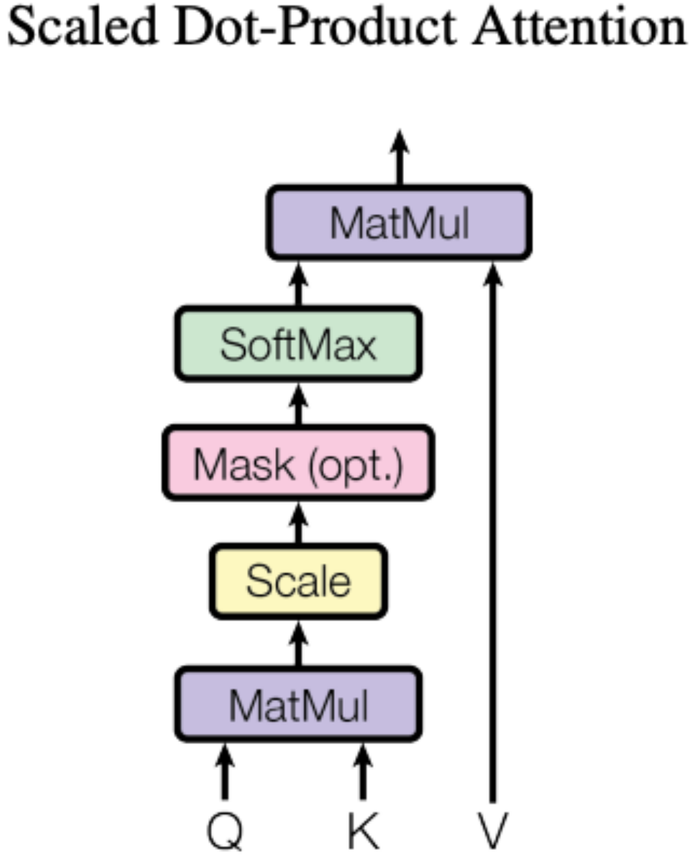
Attention

Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, 2017

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

- 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.
- **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**.

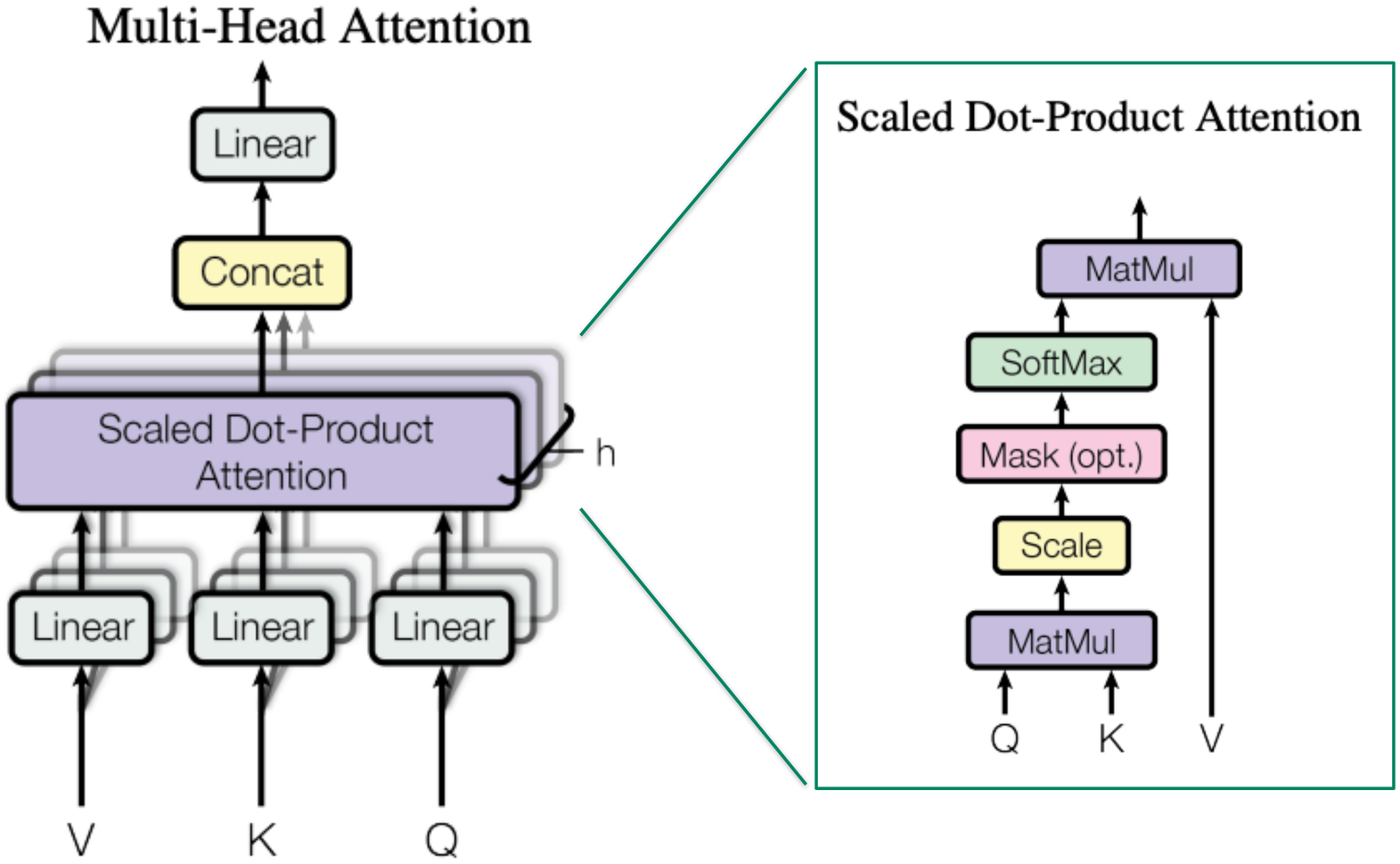
Multi-Head Attention

Attention Is All You Need

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, 2017

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

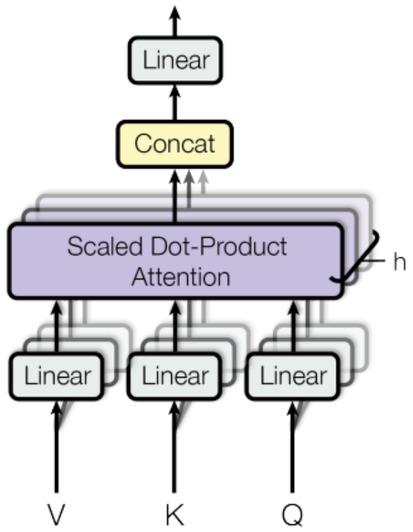
Each head is now with the dimension $\frac{d_k \text{ or } d_v}{heads}$.

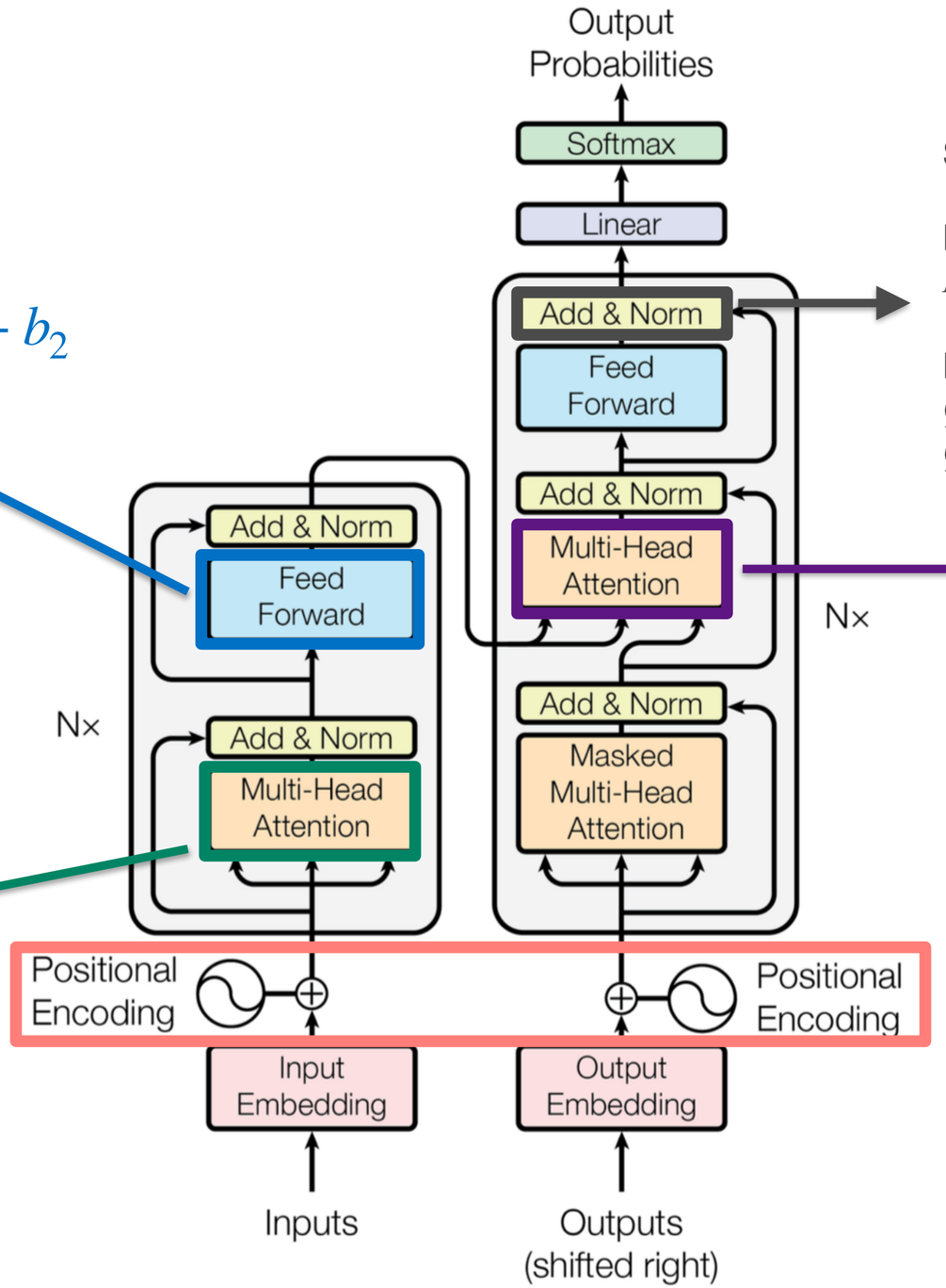


Point-Wise MLP:

$$y_i = W_2\ GELU(W_1 x + b_1) + b_2$$

Multi-Head **Self-Attention**





Shortcut + LayerNorm

PostNorm (figure):

$$LN(Q + Attn(Q, K, V))$$

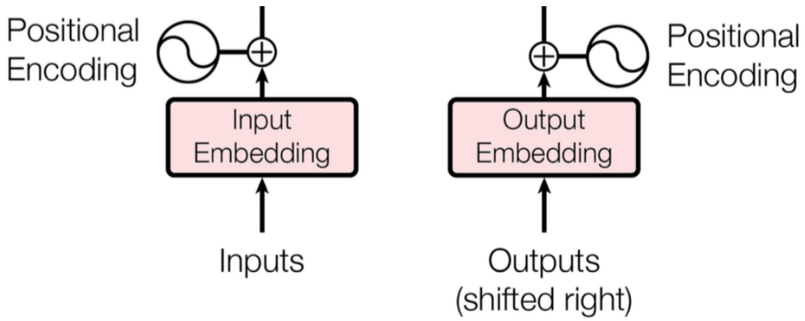
PreNorm:

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

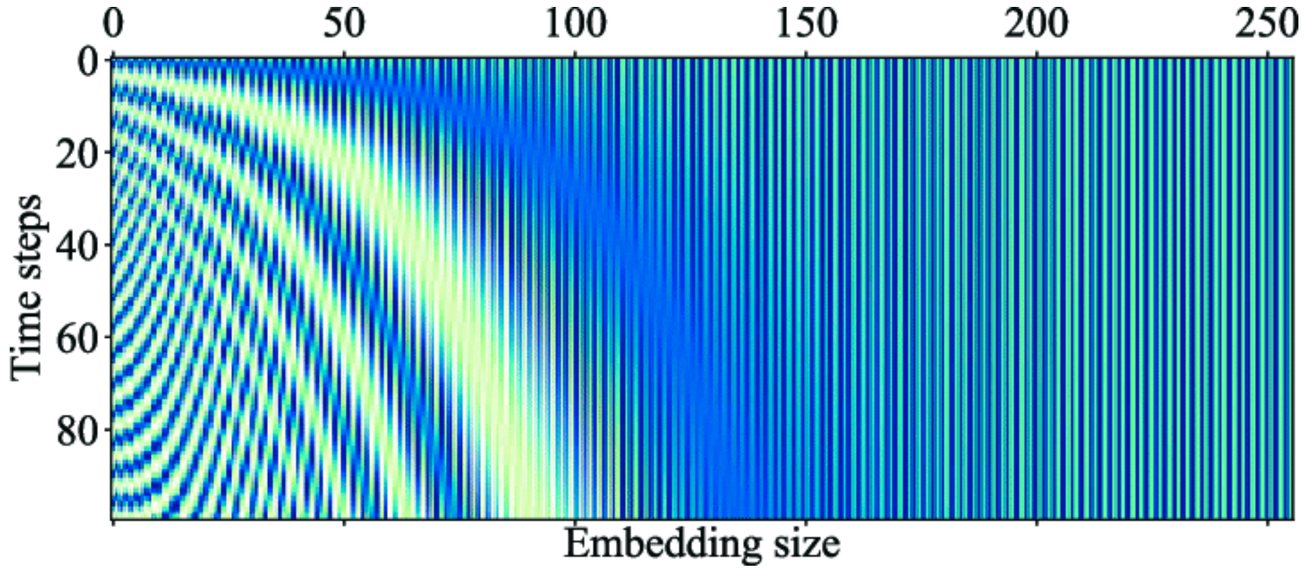
Multi-Head **Cross-Attention**

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

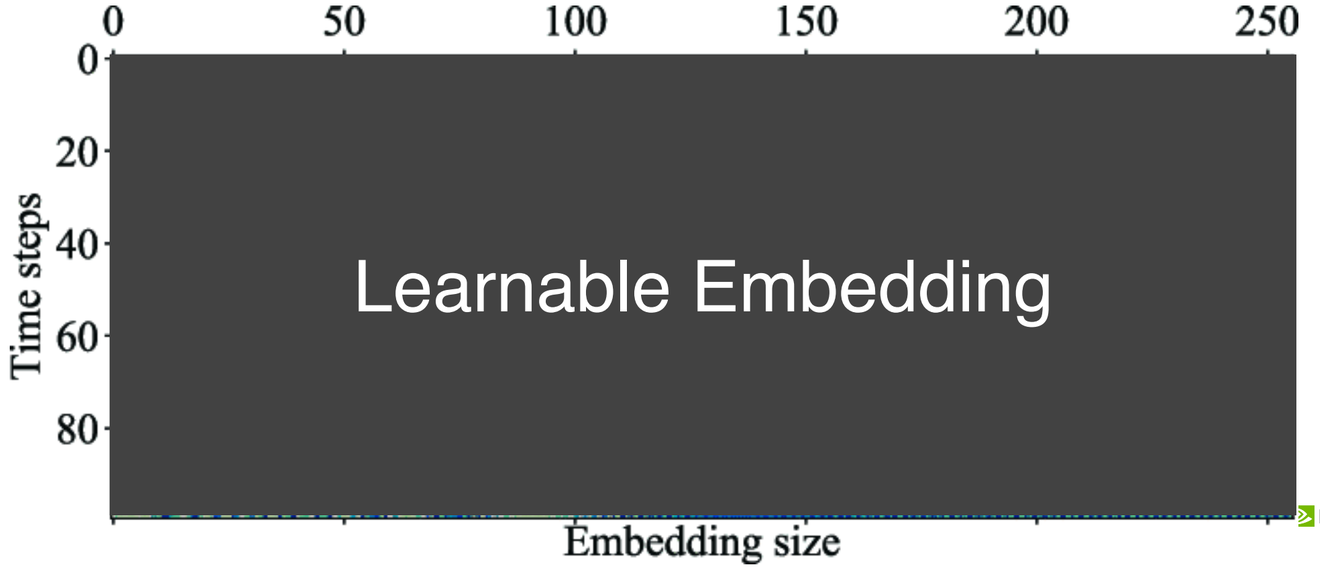
Recall: $\text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V$

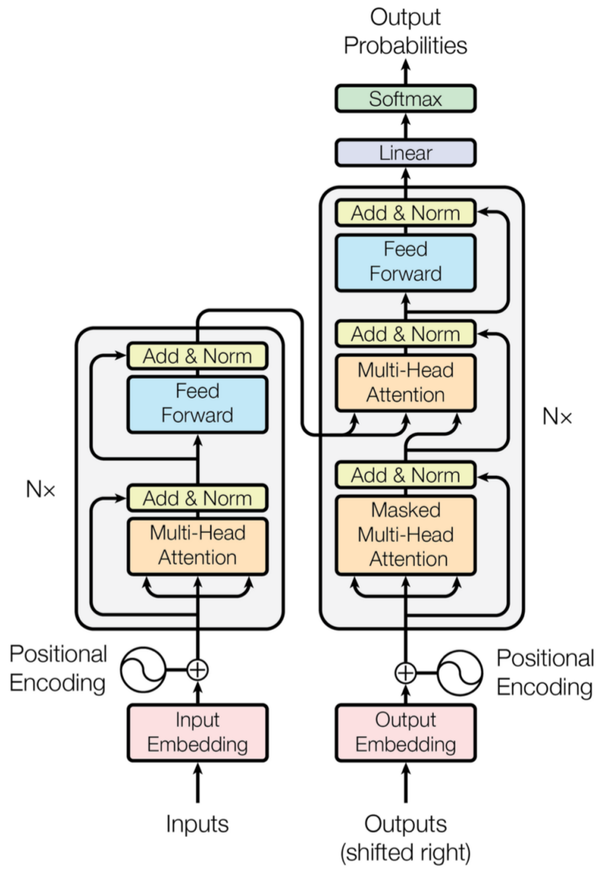


Sinusoidal Positional Encoding



General Positional Embedding





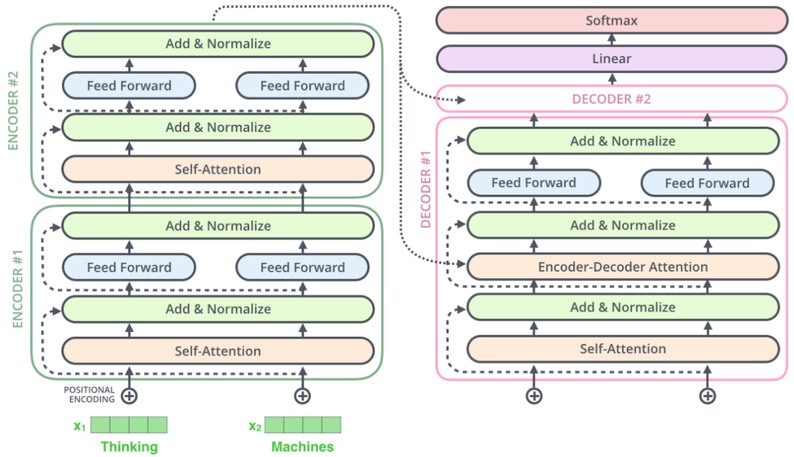


Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

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**	$2.3 \cdot 10^{19}$	

LLM Visualization

Home

Chapter: Self Attention

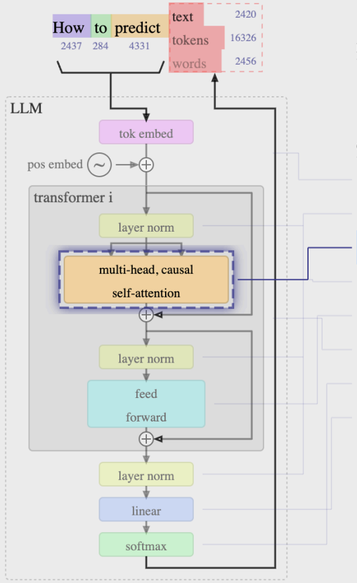


Table of Contents

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.

Press Space to continue

The first step is to produce three vectors for each of the T columns from the normalized input embedding matrix. These vectors are the Q, K, and V vectors:

Q: Query vector
K: Key vector
V: Value vector

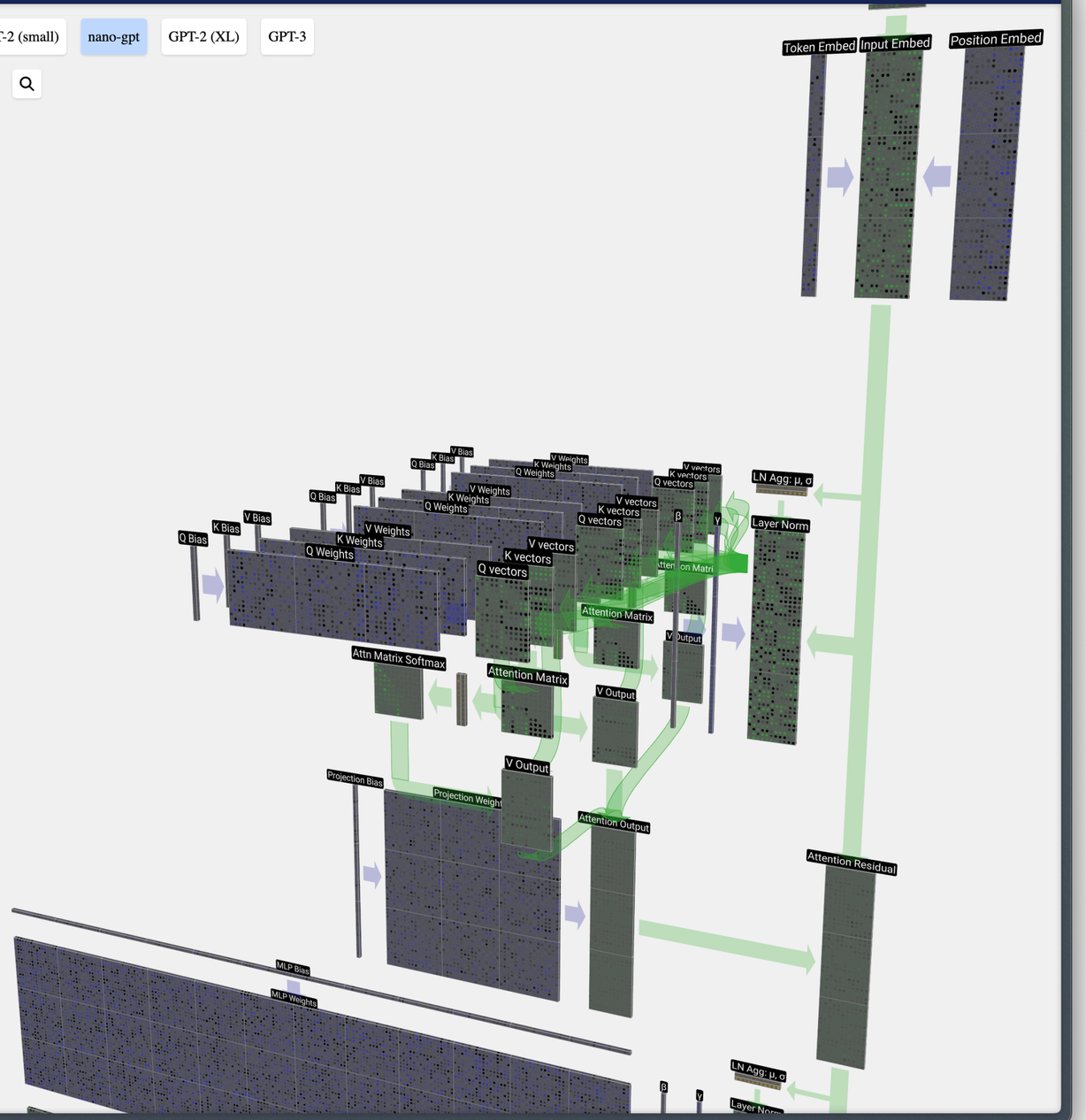
To produce one of these vectors, we perform a matrix-vector multiplication with a bias added. Each output cell is some linear combination of the input vector. E.g. for the Q vectors, this is done with a dot product between a row of the Q-weight matrix and a column of the input matrix.

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 up.

Continue

Skip

GPT-2 (small) | nano-gpt | GPT-2 (XL) | GPT-3



https://bbycroft.net/llm

NVIDIA

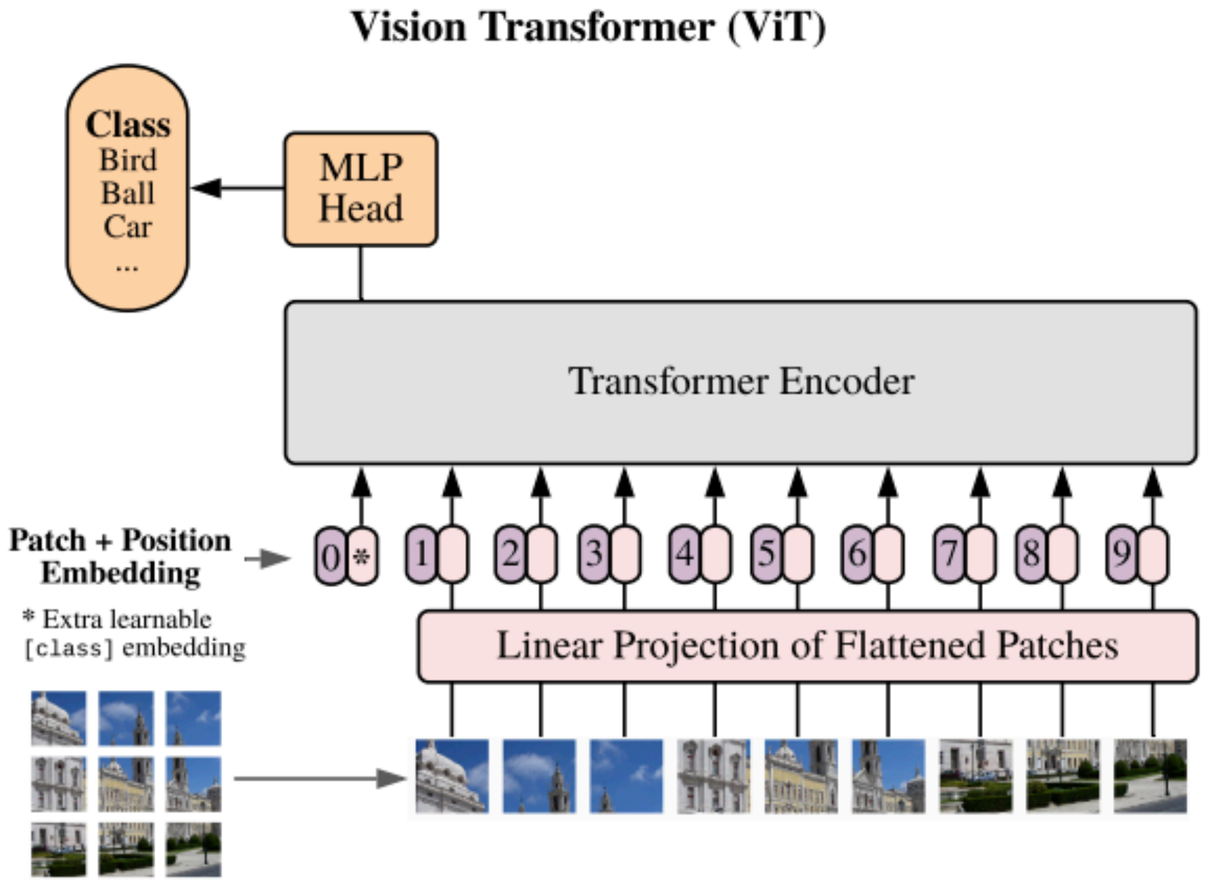
Transformer and Attention In *Computer Vision*

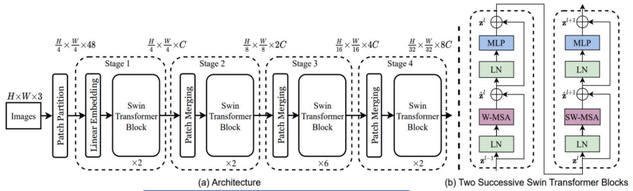
Vision Transformer

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

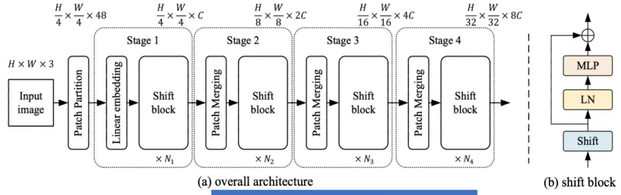
Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby, 2020

- 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)

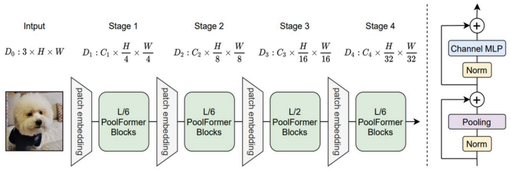




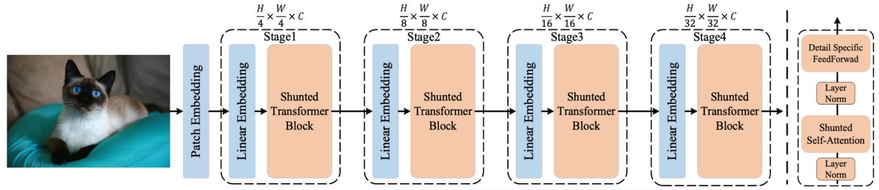
Swin Trasnformer



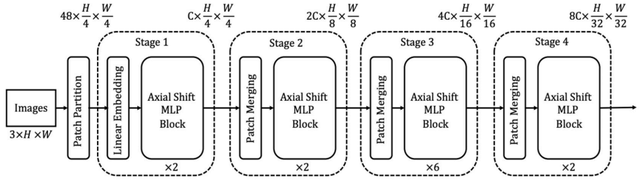
ShiftViT



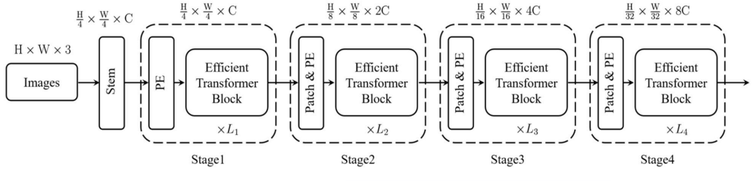
PoolFormer



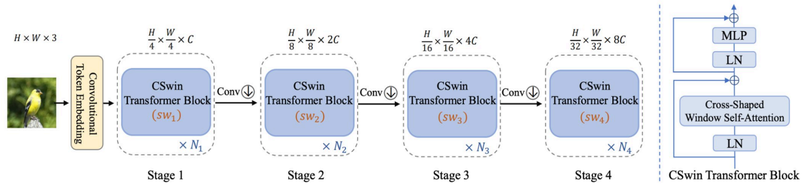
Shunted Transformer



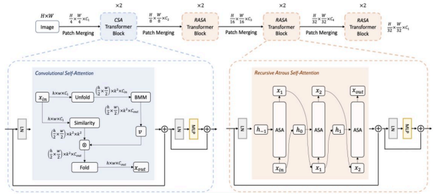
AS-MLP



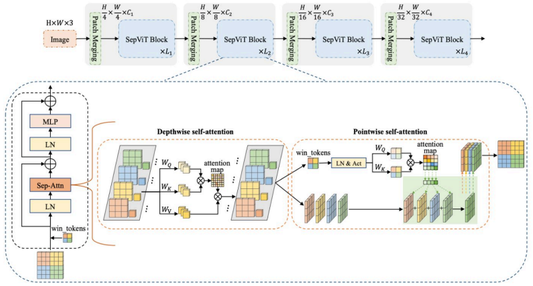
ResT



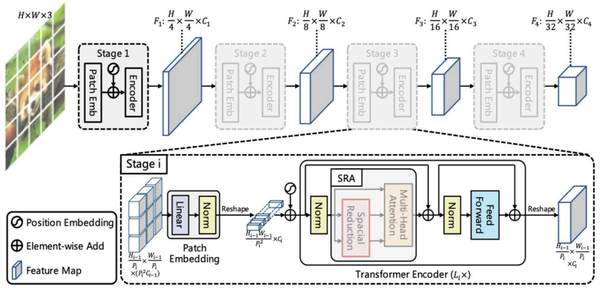
CSWin Transformer



Lite Vision Transformer

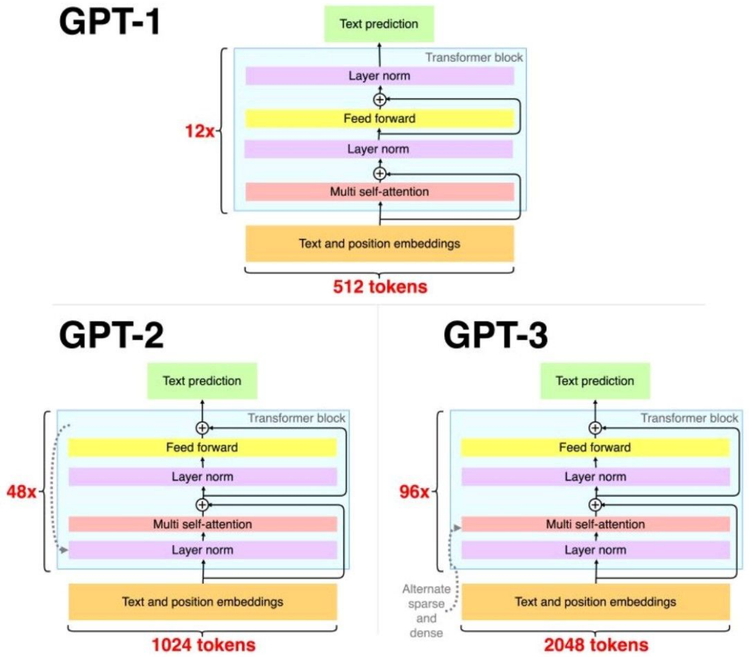


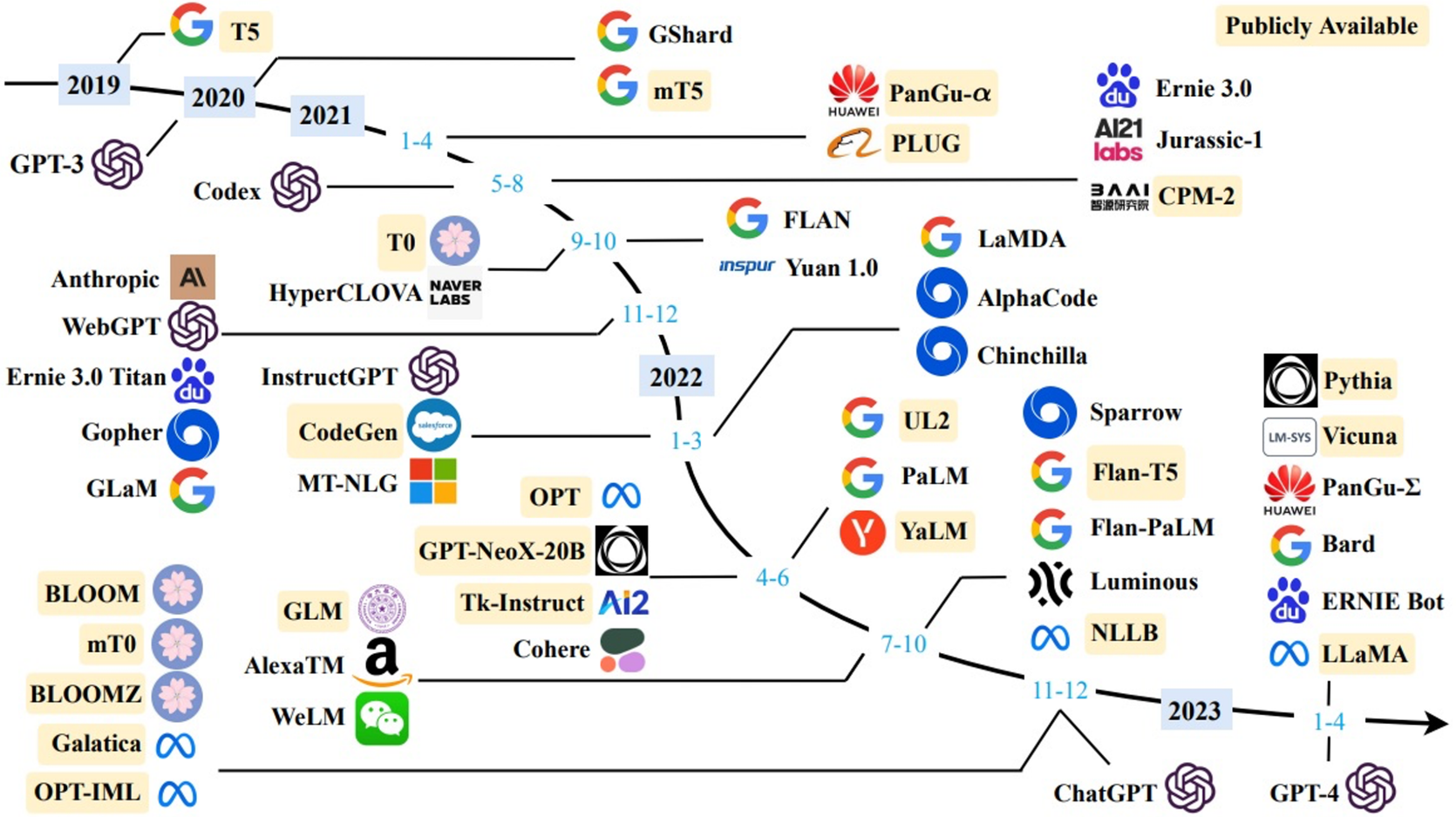
SepViT



Pyramid Vision Transformer

Explosion of works doing Vision Transformers on images





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

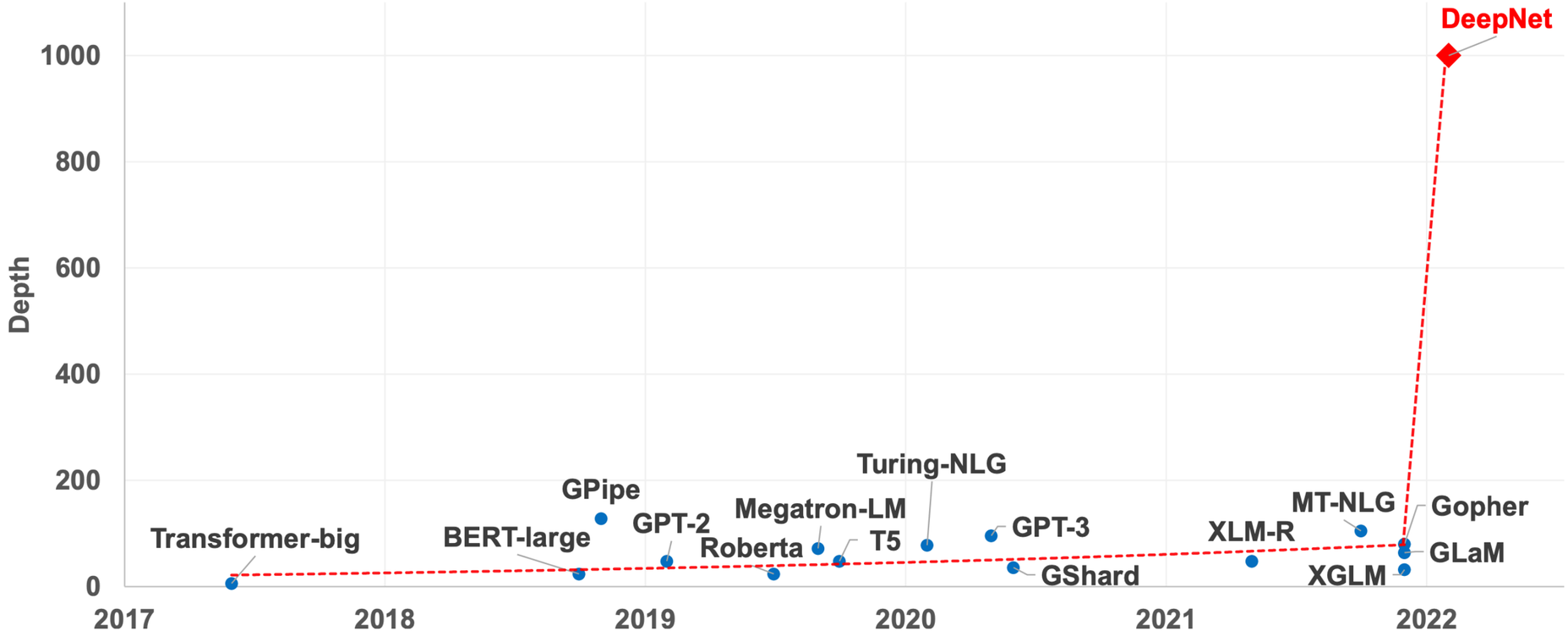
Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Furu Wei, 2022

- Normalization and Initialization matters.

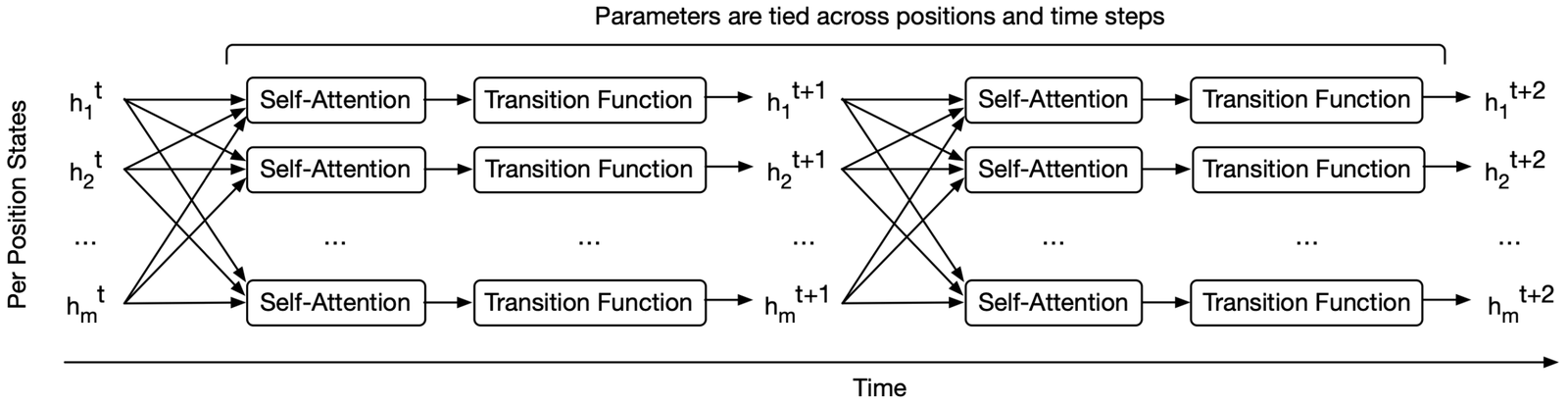
```
def deepnorm(x):
    return LayerNorm(x * α + f(x))

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

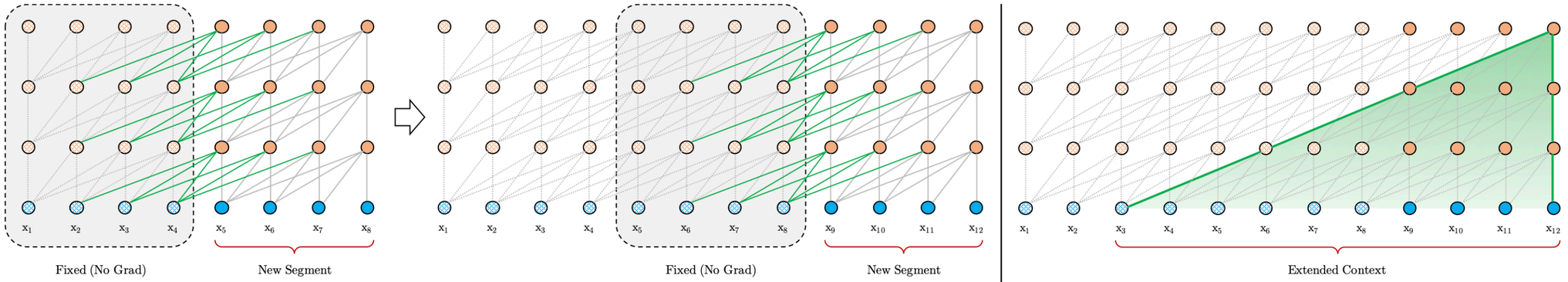
Architectures	Encoder α	Encoder β	Decoder α	Decoder β
Encoder-only (e.g., BERT)	$(2N)^{\frac{1}{4}}$	$(8N)^{-\frac{1}{4}}$	-	-
Decoder-only (e.g., GPT)	-	-	$(2M)^{\frac{1}{4}}$	$(8M)^{-\frac{1}{4}}$
Encoder-decoder (e.g., NMT, T5)	$0.81(N^4M)^{\frac{1}{16}}$	$0.87(N^4M)^{-\frac{1}{16}}$	$(3M)^{\frac{1}{4}}$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



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).

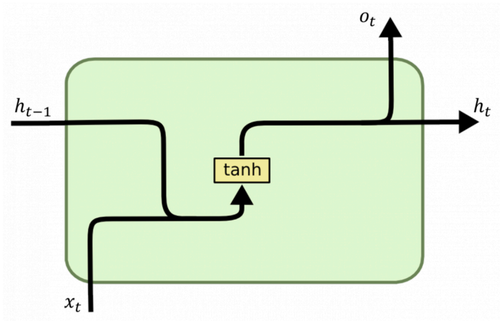
Side Note: Recurrent Networks

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

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

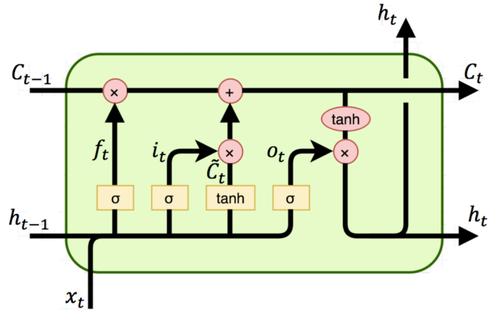
- For example,

RNN



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

LSTM



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

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

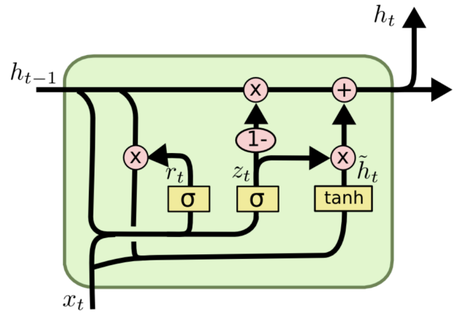
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

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

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

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

GRU



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

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

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

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

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.

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

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

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

- Where $\phi(\,.\,)$ 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.

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

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

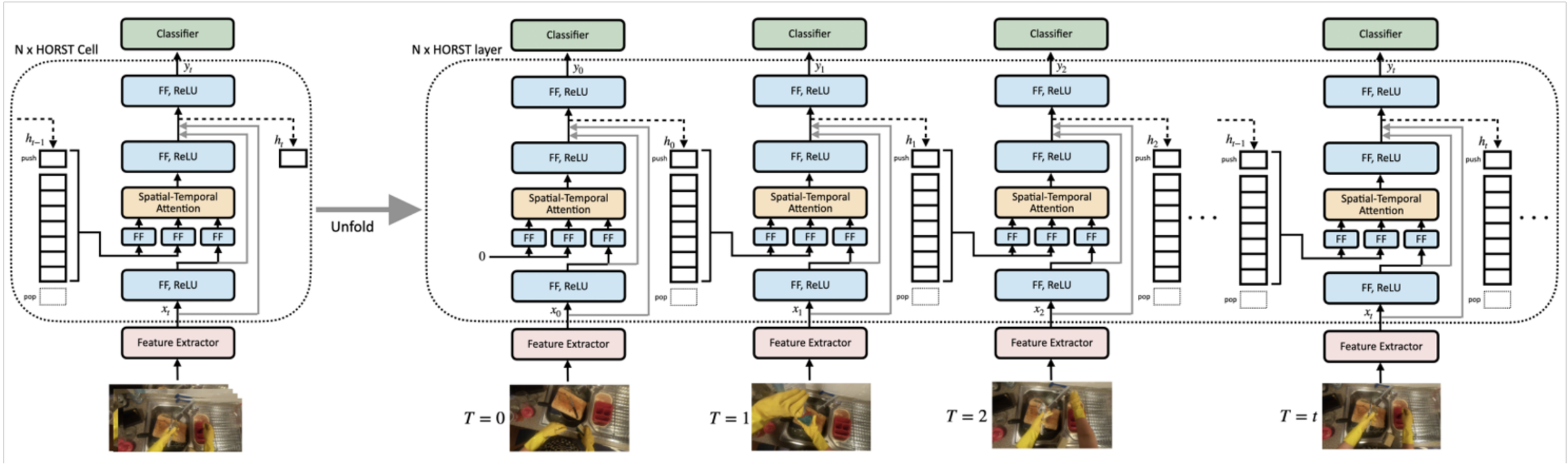
$$h_t = f(x_t + \phi(h_{t-1}, h_{t-2}, h_{t-3}, \ldots, h_{t-S}))$$

- Where $\phi(\,.\,)$ 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

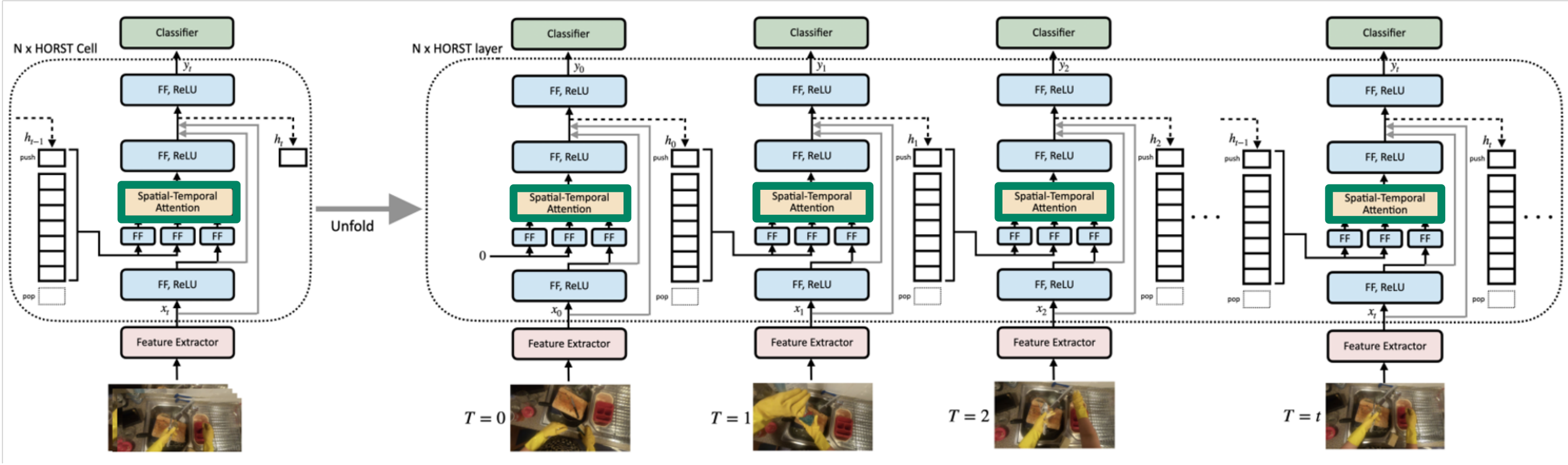
Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Oswald Lanz, 2021



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
  - $f_\chi(X) = sigmoid(\theta_\chi * [X_{avg}; X_{max}] + b_\chi)$
- Self-Filtering (X): $f_\chi(X) \cdot X$
- Cross-Filtering (X, Y): $f_\chi(X) \cdot Y$

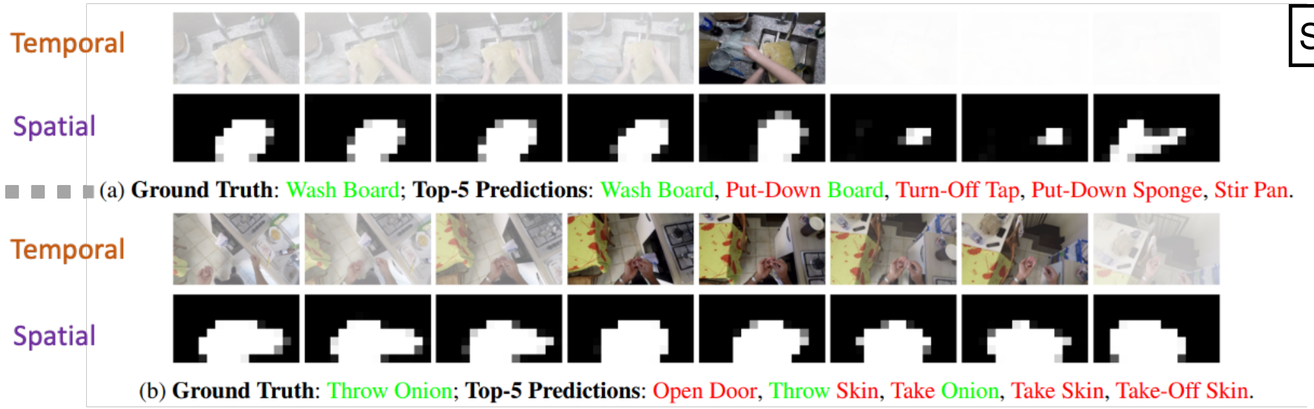
- Spatial Branch
  - $\hat{Q} = GAP(cross\_filtering(Q, K))$
  - $S(Q, K) = sigmoid(\hat{Q}^T K)$
- Temporal Branch
  - $T(Q, K) = softmax(\frac{self\_filtering(Q)^T self\_filtering(K)}{\sqrt{C}})$

- Spatial-Temporal Attention
  - $STATT(Q, K, V) = \big(S(Q, K) \cdot T(Q, K)\big) \cdot V$

Transformer Meets (Higher-Order) Recurrent

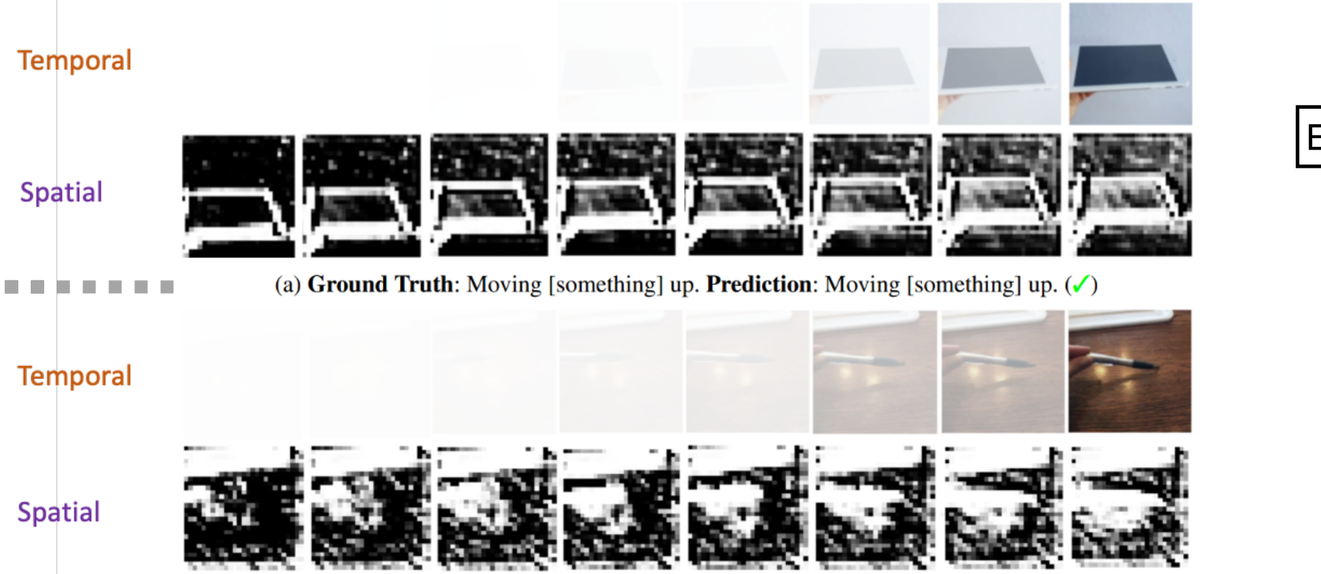
Higher Order Recurrent Space-Time Transformer for Video Action Prediction

Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Oswald Lanz, 2021



(a) **Ground Truth**: Wash Board; **Top-5 Predictions**: Wash Board, Put-Down Board, Turn-Off Tap, Put-Down Sponge, Stir Pan.

(b) **Ground Truth**: Throw Onion; **Top-5 Predictions**: Open Door, Throw Skin, Take Onion, Take Skin, Take-Off Skin.



(a) **Ground Truth**: Moving [something] up. **Prediction**: Moving [something] up. (✓)

(b) **Ground Truth**: Putting [something] on a flat surface without letting it roll. **Prediction**: Putting [something] on a surface. (✗)

SSv2

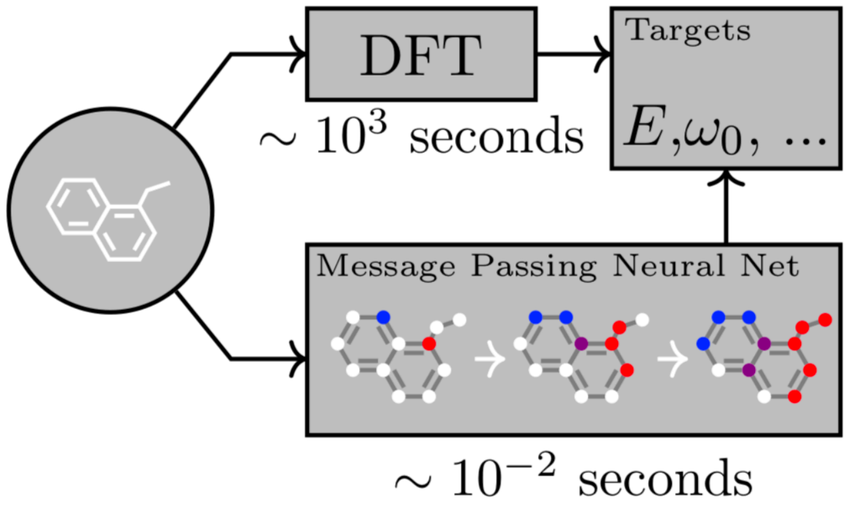
	$S=1$	$S=3$	$S=8$	$S=12$	$S=16$
Acc. (%)	30.7	31.2	**33.9**	33.8	32.7

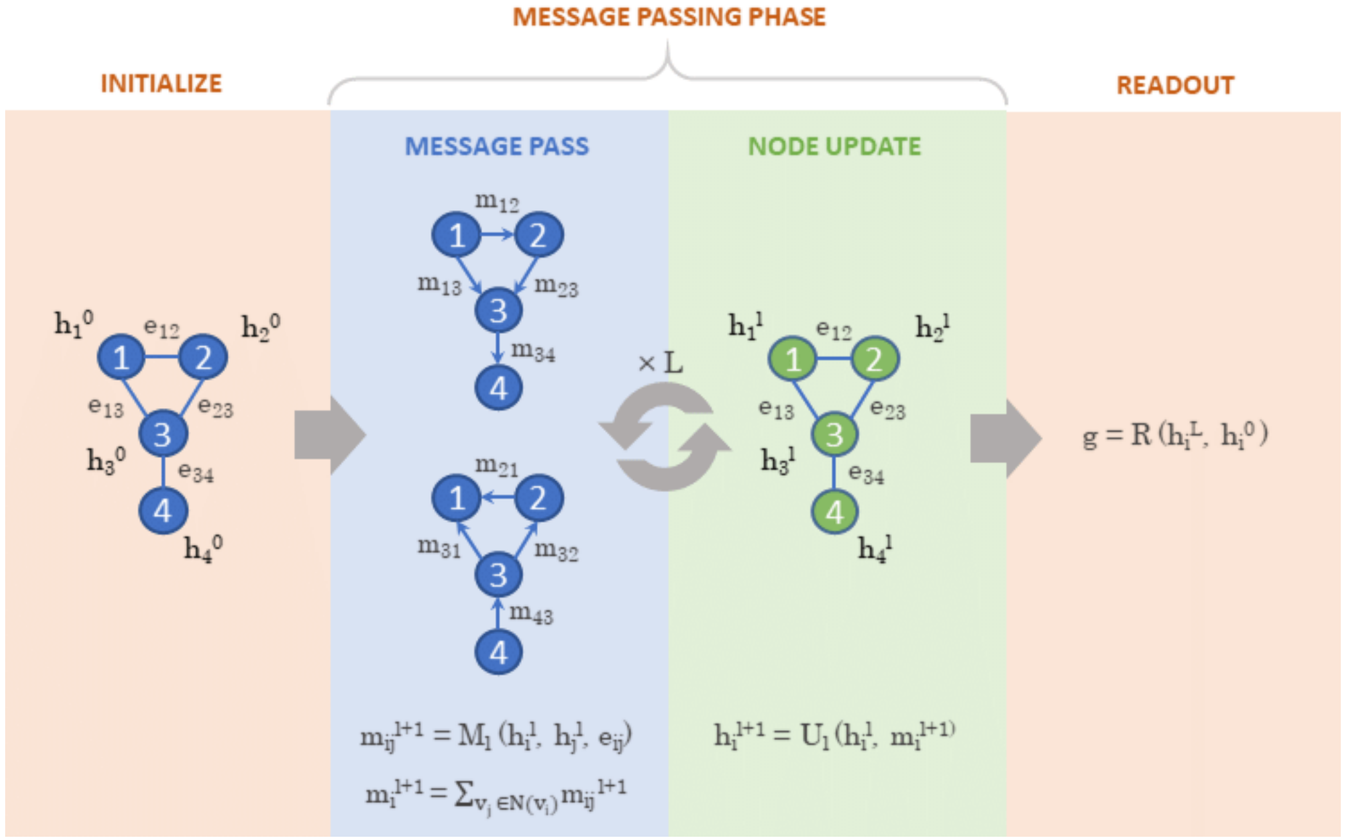
Attention	#Layers	Acc. (%)
Joint Space-Time [3]	2	30.4
Space-Time Decomposition	2	**32.2**
Temporal only	4	32.7
Spatial only	4	31.2
Divided Space-Time † [3]	4	25.8
Space-Time Decomposition	4	**33.9**

EK100

Method	Backbone	Pretrain	Verb	Noun	Action
RU-RGB	BNInc	In1k	27.5	29.0	13.3
AVT-h	BNInc	In1k	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	BNInc	In1k	26.2	29.5	14.1
HORST	ConvNeXt	In1k	28.9	**33.0**	**15.6**

Self-Attention in Message Passing Neural Network

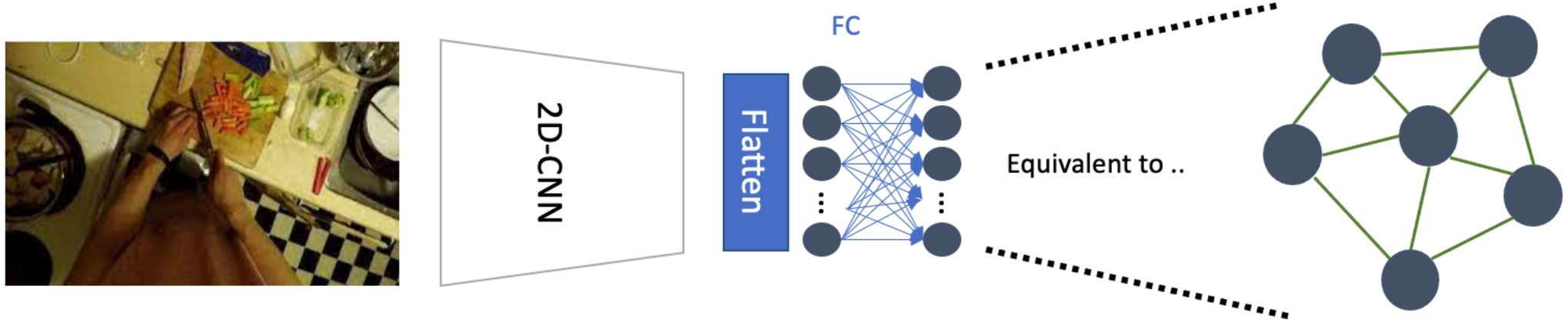




Self-Attention in a Graph Learning

Unified Recurrence Modeling for Video Action Anticipation

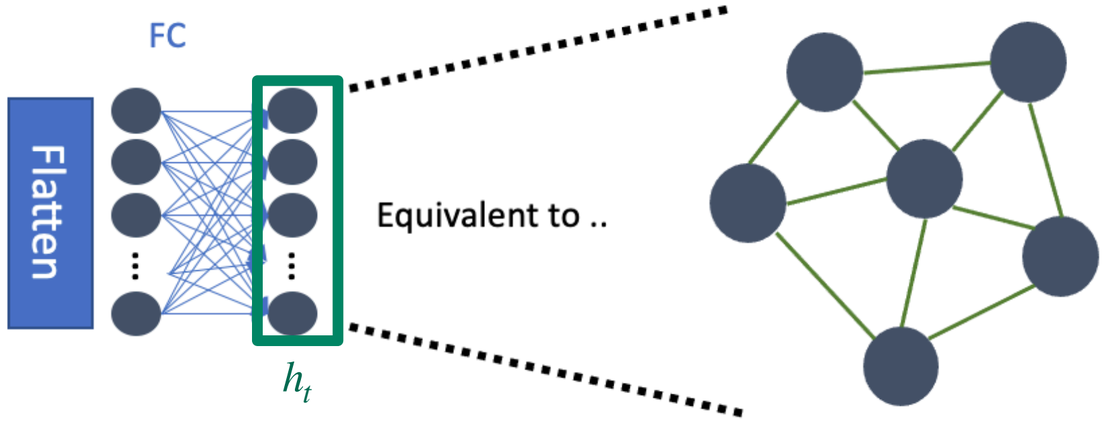
Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Simon See, Oswald Lanz, 2022



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



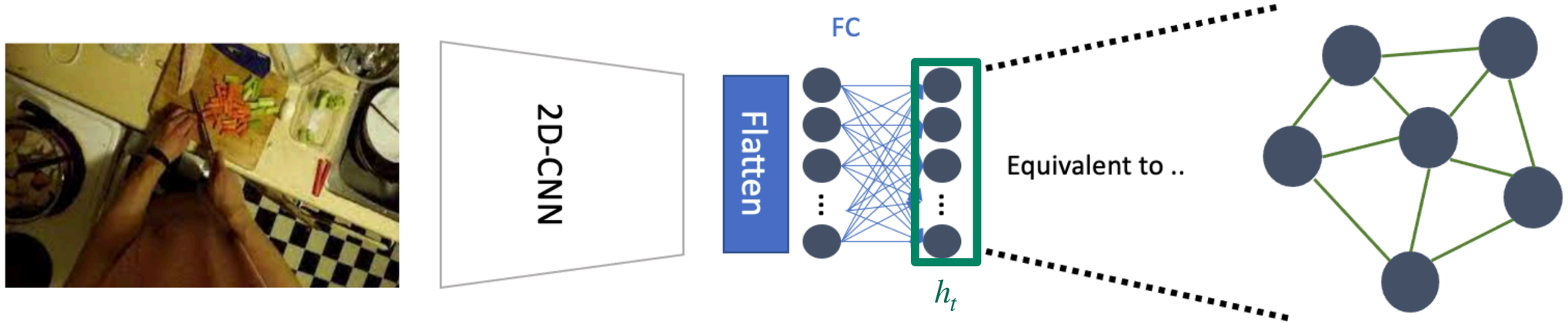
The tokenize representation.

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

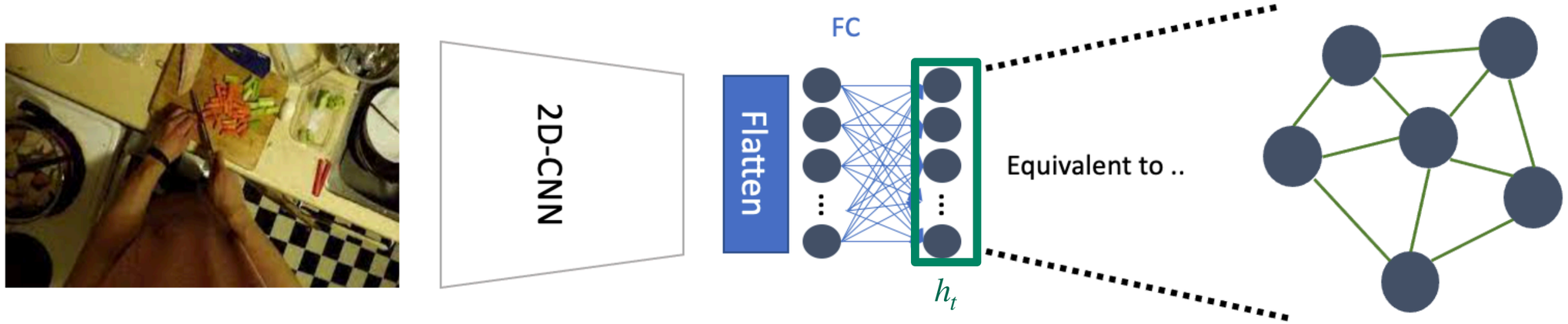


- We adopted Message-Passing Framework (Gilmer, et al 2017) for graph:
  - Message function M_t: $m^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_v w)$
  - Update function U_t: $h_v^{t+1} = U_t(h_v^t, m^{t+1})$
  - Readout function R: $y^{t+1} = R(\{h_v^{t+1}; v \in G\})$

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

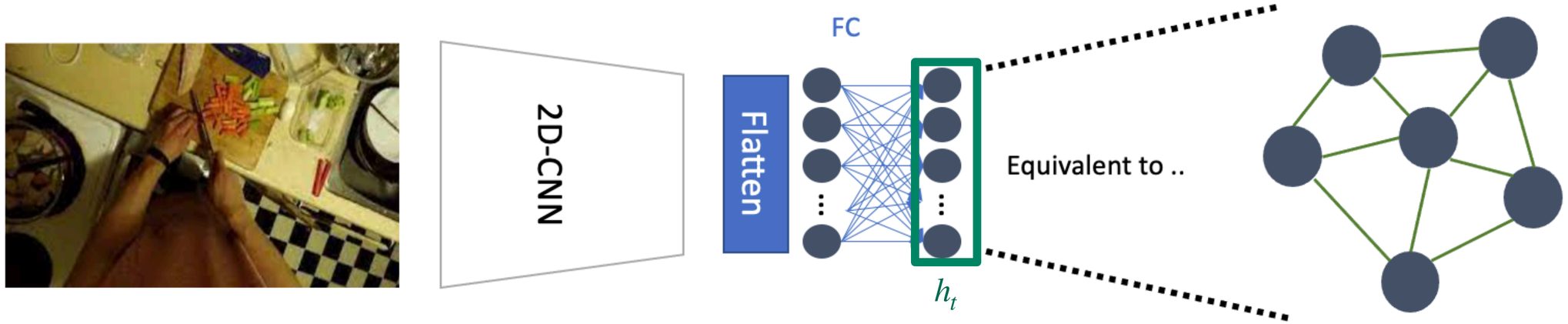


- We adopted Message-Passing Framework (Gilmer, et al 2017) for graph:
  - Message function M_t: $m^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_v w)$ Generalize $M(\,.\,)$ by $attention(Q = h_t, K = h_t, V = h_t)$
  - Update function U_t: $h_v^{t+1} = U_t(h_v^t, m^{t+1})$
  - Readout function R: $y^{t+1} = R(\{h_v^{t+1}; v \in G\})$

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



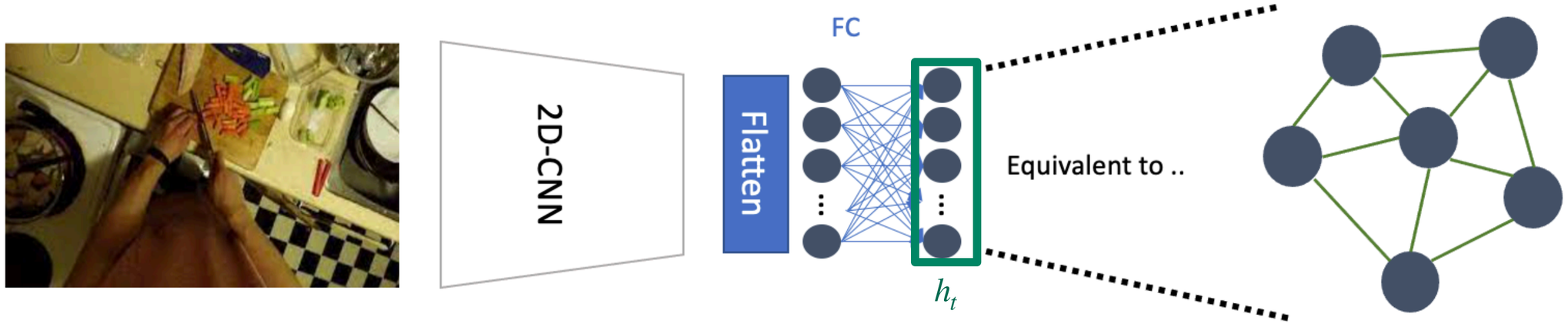
- We adopted Message-Passing Framework (Gilmer, et al 2017) for graph:
  - Message function M_t: $m^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_v w)$ Generalize $M(\,.\,)$ by $attention(Q = h_t, K = h_t, V = h_t)$
  - Update function U_t: $h_v^{t+1} = U_t(h_v^t, m^{t+1})$
  - Readout function R: $y^{t+1} = R(\{h_v^{t+1}; v \in G\})$

Generalize $U(\,.\,), R(\,.\,)$ by attention as well.

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

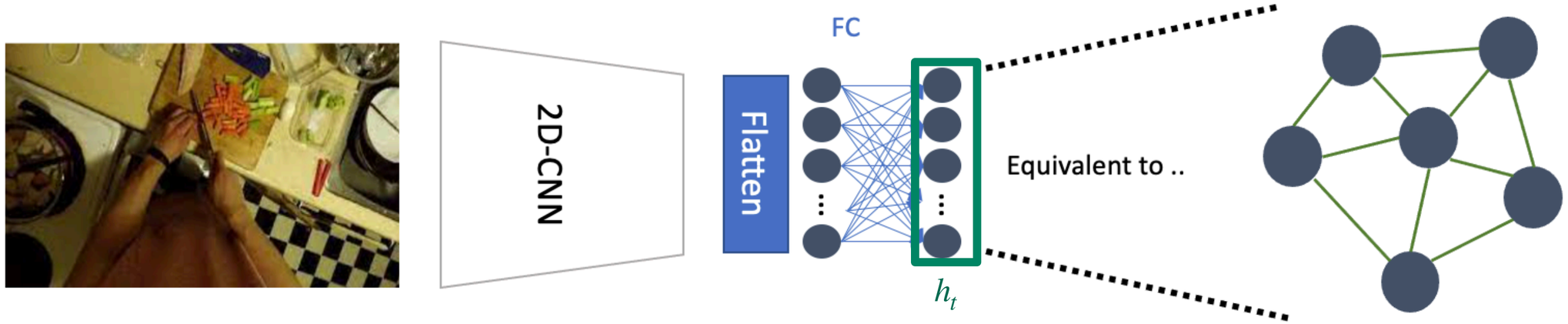


- We adopted Message-Passing Framework (Gilmer, et al 2017) for graph:
  - Message function M: $m_{t+1} = M(h_t)$
  - Update function U: $h_{t+1} = U(h_t, m_{t+1})$
  - Readout function R: $y_{t+1} = R(h_{t+1})$

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



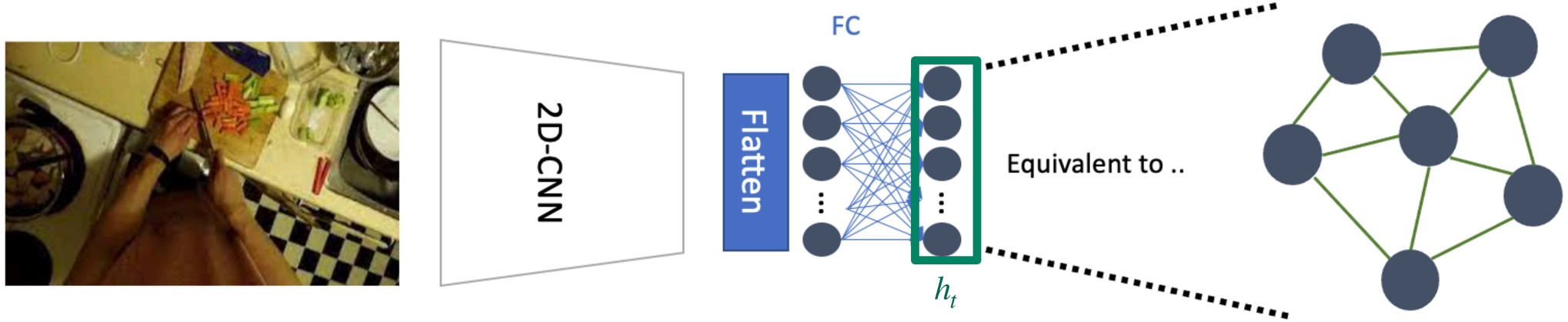
- We adopted Message-Passing Framework (Gilmer, et al 2017) for graph:
  - Message function M: $m_{t+1} = M(h_t)$
  - Update function U: $h_{t+1} = U(h_t, m_{t+1})$
  - Readout function R: $y_{t+1} = R(h_{t+1})$

  - **How about the edges?**

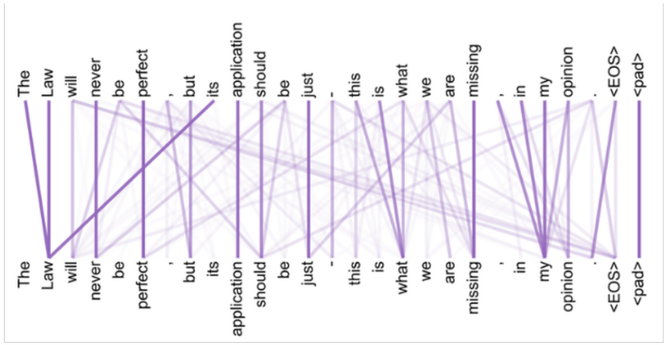
Self-Attention in a Graph Learning

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?**



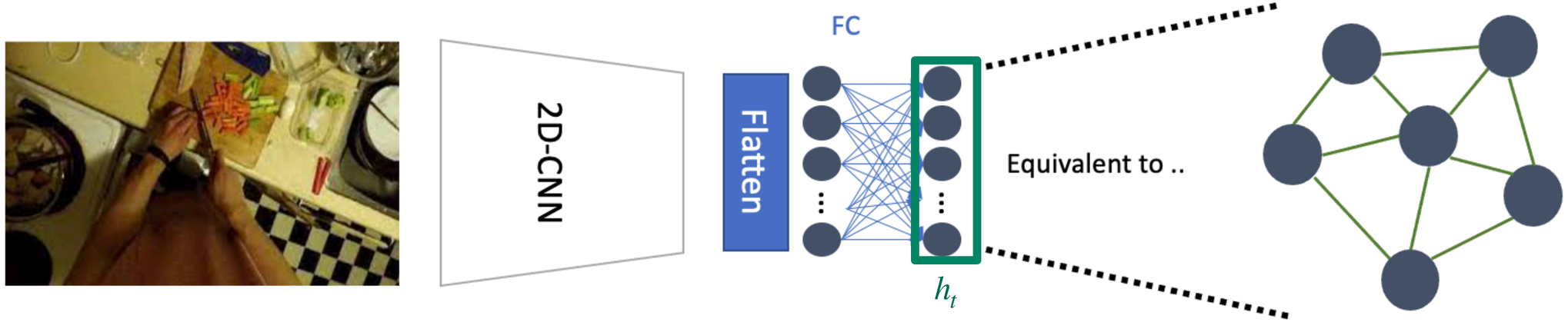
Vaswani et al., 2017

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

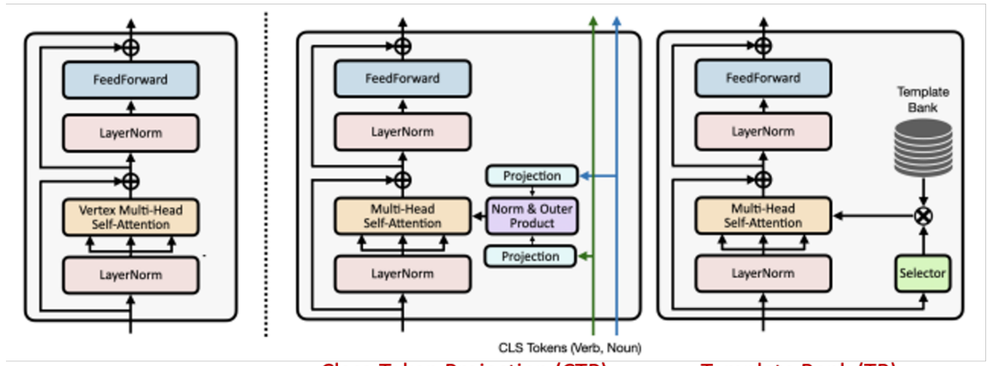
Self-Attention in a Graph Learning

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



CTP:

$V = f_v(CLS^t_{verb})$

$N = f_n(CLS^t_{noun})$

$\widehat{A^t} = V \otimes N$

TB:

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

$\widehat{A^t} = \sum_{i=0}^{s} w_i \cdot B_i$ (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	Zeyu Jiang	South China University of Technology
	(hrgdscs)	Changxing Ding	South China University of Technology
②	NVIDIA-UNIBZ	Tsung-Ming Tai	NVIDIA, Free University of Bozen-Bolzano
	(corcovadoming)	Oswald Lanz	Free University of Bozen-Bolzano
		Giuseppe Fiameni	NVIDIA
		Yi-Kwan Wong	NVIDIA
		Sze-Sen Poon	NVIDIA
		Cheng-Kuang Lee	NVIDIA
		Ka-Chun Cheung	NVIDIA
		Simon See	NVIDIA
③	ICL-SJTU	Xiao Gu	Imperial College London
	(Shawn0822)	Yao Guo	Shanghai Jiao Tong University
		Zeju Li	Imperial College London
		Jianing Qiu	Imperial College London
		Benny Lo	Imperial College London
		Guang-Zhong Yang	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

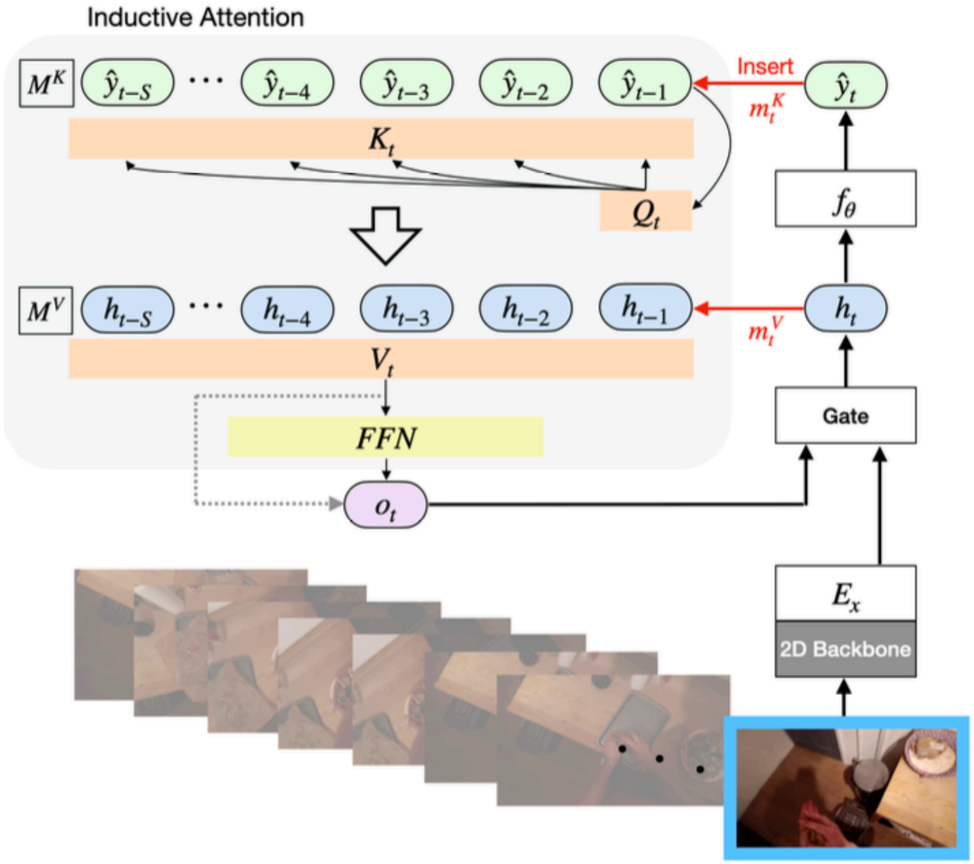
What is the suitable query choice?

- Query triggers the transformer to pay attention on the meaningful contents of inputs.
  - Attention relieves 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.

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

$$InductiveAttention(Y_t, Y_{<t}, X_{<t})$$



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

Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive
squeeze cloth	wash cloth	wash cloth	squeeze cloth	wash cloth	squeeze cloth	squeeze cloth
wash sink	wash top	unroll cloth	unroll cloth	wash sink	wash plug	wash cloth
unroll cloth	unroll cloth	dry hand	wash plug	dry hand	wash cloth	hang hand
gather rubbish	dry hand	wash hand	pull cup	take cloth	pull cup	pull cup
put sink	wash plug	wash top	hang hand	put cloth	hang hand	squeeze sponge
Top-5 Negative	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**
apply bowl	let-go sponge	gather meat	sort broccoli	throw pear	roll omelette	roll lemon
gather broccoli	mix lettuce	gather broccoli	eat pork	put sheets	wrap box	pat omelette
insert bean:green	eat pork	eat broccoli	coat oil	pat omelette	eat broccoli	eat pork
check pie	check box	wash food	brush dough	unroll omelette	turn ring:onion	turn-on liquid:washing
close freezer	soak plate	insert pan:dust	eat broccoli	move omelette	pat noodle	roll omelette

wash sink
squeeze cloth
wash top
wash cloth
squeeze sponge

Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive
wash top	squeeze cloth	squeeze cloth	wash sink	squeeze cloth	wash sink	wash sink
wash cloth	unroll cloth	wash sink	wash cloth	squeeze sponge	dry hand	wash top
throw food	wash oven	wash floor	squeeze sponge	wash top	take cloth	dry hand
take cloth	shake hand	drink beer	wash top	wash oven	wash oven	wash oven
throw rubbish	gather rubbish	remove rubbish	wash sponge	put sponge	gather rubbish	wash hand
Top-5 Negative	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**
shake rubbish	peel squash	insert banana	shake peach	dry scissors	fill liquid	adjust plate
wash plug	insert nut	peel sausage	squeeze garlic	search salt	hang hand	dry sponge
shake straw	take nut	unroll omelette	carry bag	turn-on water	squeeze can	move funnel
dry cloth	cut apple	cut banana	wash straw	throw squash	shake liquid	empty container
hang hand	look bag	hang hand	squeeze caper	hang cup	shake courgette	filter plate

wash sink
squeeze cloth
wash top
wash cloth
squeeze sponge

Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive
wash top	wash top	wash top	wash sink	squeeze sponge	wash top	wash top
wash cloth	squeeze sponge	unroll cloth	unroll cloth	squeeze cloth	squeeze sponge	wash cloth
squeeze cloth	wash cloth	wash oven	pour liquid:washing	wash sponge	wash cloth	squeeze sponge
squeeze sponge	wash sponge	dry hand	wash hand	turn-off tap	dry hand	take cloth
take cloth	take cloth	squeeze sponge	dry hand	take cloth	take cloth	dry hand
Top-5 Negative	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**
shake straw	shake straw	shake straw	throw squash	shake straw	shake straw	hang hand
shake rubbish	shake rubbish	brush sink	smell candle	throw squash	shake rubbish	move coffee
throw kale	shake caper	adjust straw	let-go sponge	shake rubbish	brush sink	shake straw
hang hand	wash plug	shake caper	soak plate	wash straw	move filter	squeeze caper
move stand	pat napkin	take alarm	shake caper	drop sponge	move coffee	shake rubbish

wash sink
squeeze cloth
wash top
wash cloth
squeeze sponge

(GT: take paper)
Rank 28 of 3806

Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive
apply olive	take pizza	take pizza	put pizza	put pizza	put pizza	apply olive
put mushroom	apply olive	apply olive	flatten dough	take pizza	apply olive	cut pizza
put potato	apply basil	apply basil	apply olive	apply basil	open pizza	remove pizza
throw rubbish	take basil	check pizza	fold bag	apply olive	put fruit	put pin:rolling
throw skin	open oven	take basil	put potato	take basil	close bin	put food
Top-5 Negative	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**
look bottle	check board:chopping	look sauce	turn-on dishwasher	eat bread	pat spatula	look can
sort fork	search book	remove jar	sort fork	press coriander	insert ring:onion	take backpack
open kale	rip wrap:plastic	look can	check drawer	take candle	move omelette	take pear
put parsley	insert ring:onion	insert holder	scrub scissors	open corn	check pie	insert cap
insert oven	insert cloth	lift dishwasher	look box	look box	put pie	pull box

put pizza
take pizza
insert pizza
cut pizza
remove pizza

Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive
put pizza	put pizza	insert pizza	flatten dough	insert pizza	take pizza	put pizza
take pizza	insert pizza	remove pizza	cut pizza	flatten dough	insert pizza	take pizza
insert pizza	check pizza	cut pizza	apply dough	cut pizza	cut pizza	insert pizza
close bin	open pizza	flatten dough	remove pizza	apply olive	put sausage	drink beer
open pizza	wrap plate	open oven	put fruit	put fruit	remove pizza	open pizza
Top-5 Negative	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**
cut corn	cut corn	move pen	rip wrap:plastic	dry food	remove floor	cut corn
pat garlic	press onion	put pear	divide pizza	look bottle	throw oil	sprinkle tomato
press onion	remove stalk	take toaster	unroll clothes	sort broccoli	throw butter	close oregano
check ginger	apply onion	eat broccoli	move fridge	pour bean:green	take pear	put pear
check onion	sort onion	remove cucumber	dry food	brush dough	close paper	scrape skin

put pizza
take pizza
insert pizza
cut pizza
remove pizza

Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive	Top-5 Positive
apply olive	take pizza	put pizza	flatten dough	remove pizza	put pizza	apply olive
cut pizza	insert pizza	insert pizza	insert pizza	insert pizza	open pizza	take pizza
remove pizza	check pizza	check pizza	take pizza	take pizza	close bin	cut pizza
apply basil	open pizza	cut pizza	fold bag	cut pizza	drink beer	put fruit
take aubergine	open oven	open pizza	divide dough	apply basil	remove salad	put food
Top-5 Negative	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**	**Top-5 Negative**
squeeze garlic	cut corn	cut corn	search plate	cut corn	cut corn	check board:chopping
pat burger	press onion	press onion	lift foil	attach tray	close pepper	unroll clothes
squeeze sausage	apply onion	apply onion	cut pork	take toaster	put shell:egg	move fridge
mix sausage	sort onion	sort onion	sort pizza	rub sauce	insert shell:egg	scrape salmon
sort onion	squeeze garlic	remove stalk	hold tray	shake tomato	press onion	insert oven

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.

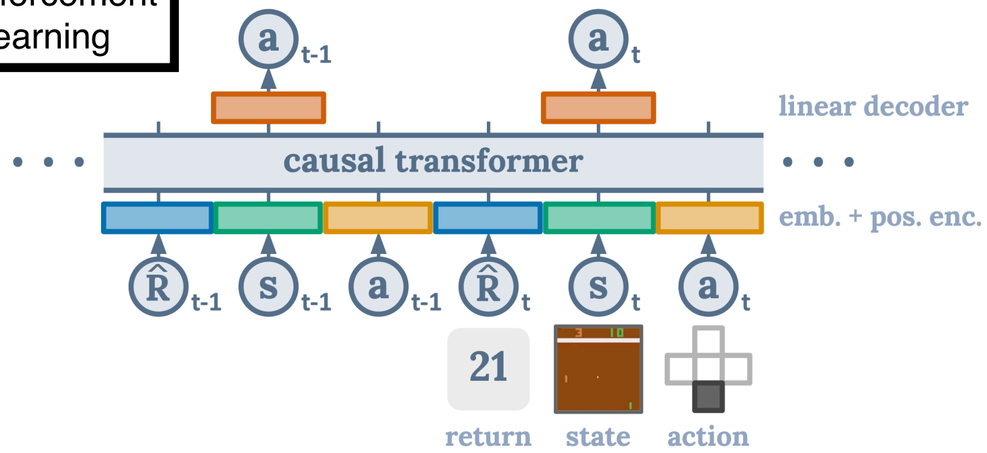
- Every thing you can tokenize, you can feed into the transformer

- 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.

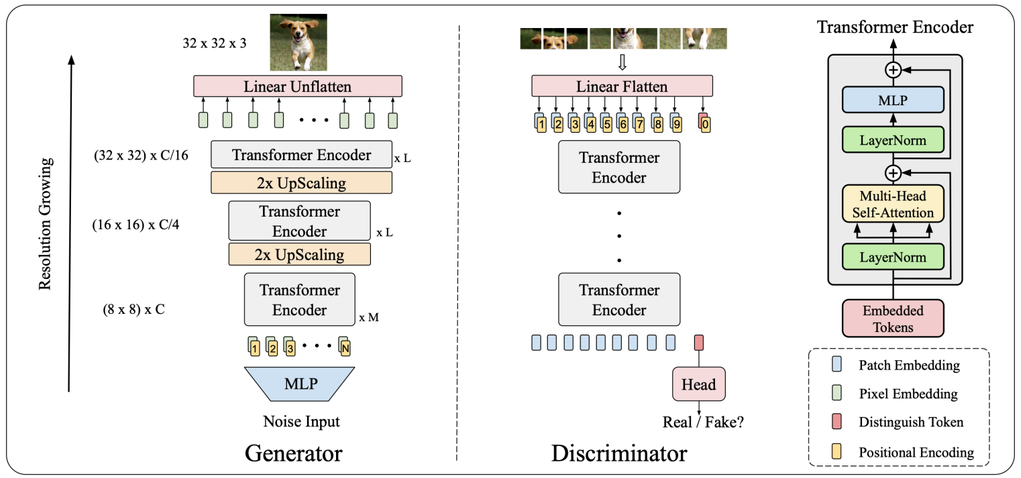
More Transformer Applications

Reinforcement Learning



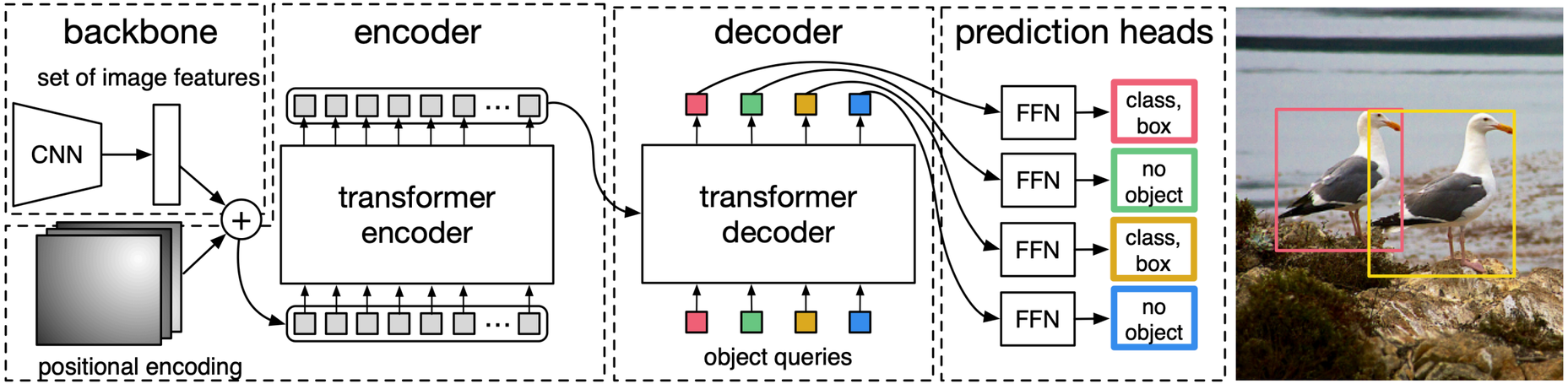
Chen, Lili, et al. "Decision transformer: Reinforcement learning via sequence modeling." *Advances in neural information processing systems* 34 (2021): 15084-15097.

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: lucasb.eyer.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