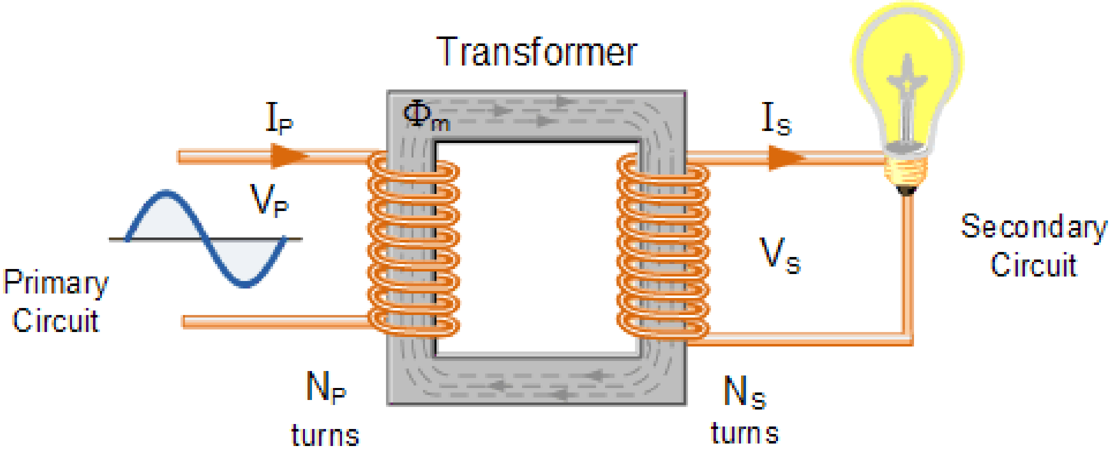
渤海大學
BOHAI UNIVERSITY

Transformer's Application In Computer Vision

Introduction

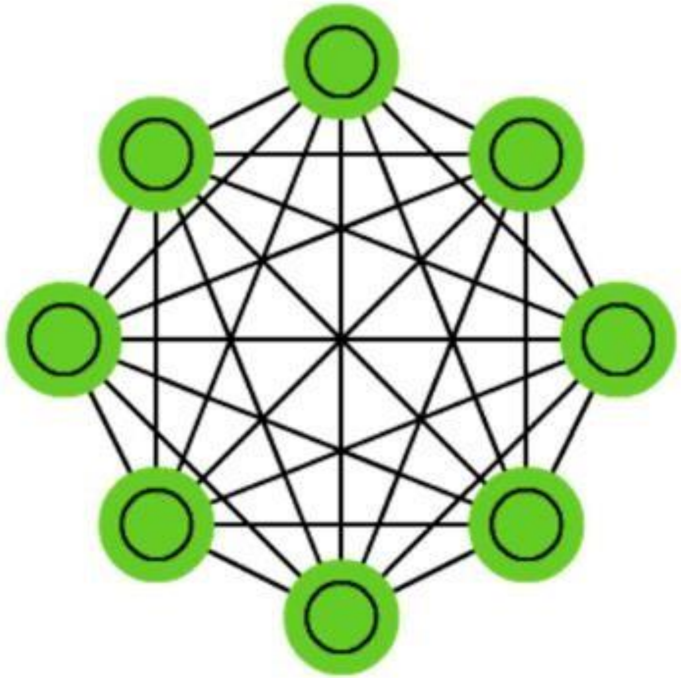
What’s Transformer?

变形金刚



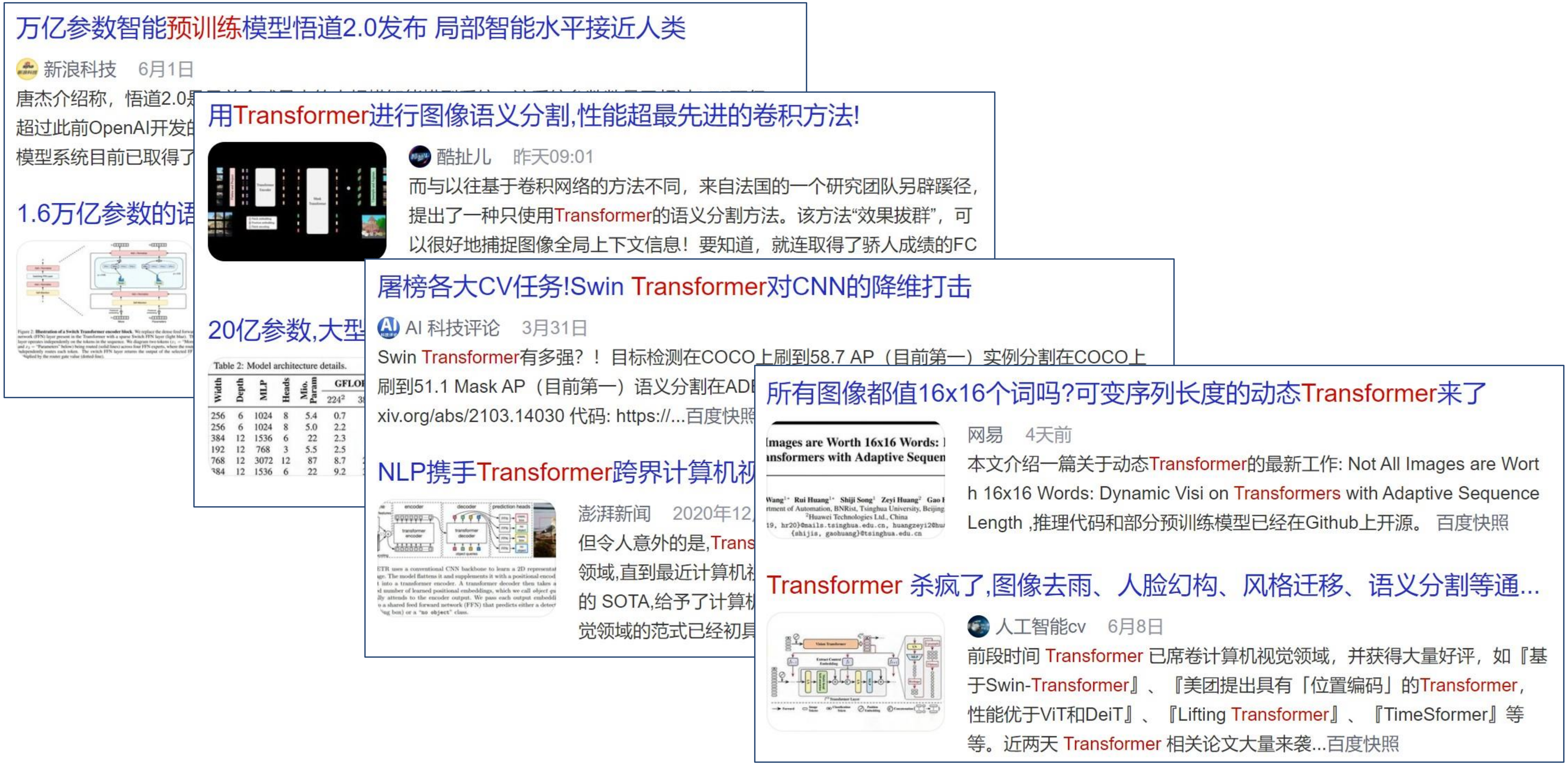
变压器

Transformer



Transformer is an architecture of A Neural Network.

Transformer



万亿参数智能预训练模型悟道2.0发布 局部智能水平接近人类

新浪科技 6月1日

唐杰介绍称，悟道2.0

超过此前OpenAI开发的

模型系统目前已取得了

1.6万亿参数的语

用Transformer进行图像语义分割,性能超最先进的卷积方法!

酷扯儿 昨天09:01

而与以往基于卷积网络的方法不同，来自法国的一个研究团队另辟蹊径，提出了一种只使用Transformer的语义分割方法。该方法"效果拔群"，可以很好地捕捉图像全局上下文信息！要知道，就连取得了骄人成绩的FC

20亿参数,大型

Table 2: Model architecture details.

Width	Depth	MLP	Heads	Mio. Param	GFLOP 224²
256	6	1024	8	5.4	0.7
256	6	1024	8	5.0	2.2
384	12	1536	6	22	2.3
192	12	768	3	5.5	2.5
768	12	3072	12	87	8.7
384	12	1536	6	22	9.2

屠榜各大CV任务!Swin Transformer对CNN的降维打击

AI 科技评论 3月31日

Swin Transformer有多强？！ 目标检测在COCO上刷到58.7 AP（目前第一）实例分割在COCO上刷到51.1 Mask AP（目前第一）语义分割在ADE

xiv.org/abs/2103.14030 代码: https://...百度快照

NLP携手Transformer跨界计算机视

澎湃新闻 2020年12

但令人意外的是,Trans

领域,直到最近计算机视

的 SOTA,给予了计算机

觉领域的范式已经初具

所有图像都值16x16个词吗?可变序列长度的动态Transformer来了

Images are Worth 16x16 Words:
nsformers with Adaptive Sequen

Wang Rui Huang Shiji Song Zeyi Huang Gao
rtment of Automation, BNRist, Tsinghua University, Beijing
Huawei Technologies Ltd., China
19, hr20}@mails.tsinghua.edu.cn, huangzeyi2@hu
{shijis, gaohuang}@tsinghua.edu.cn

网易 4天前

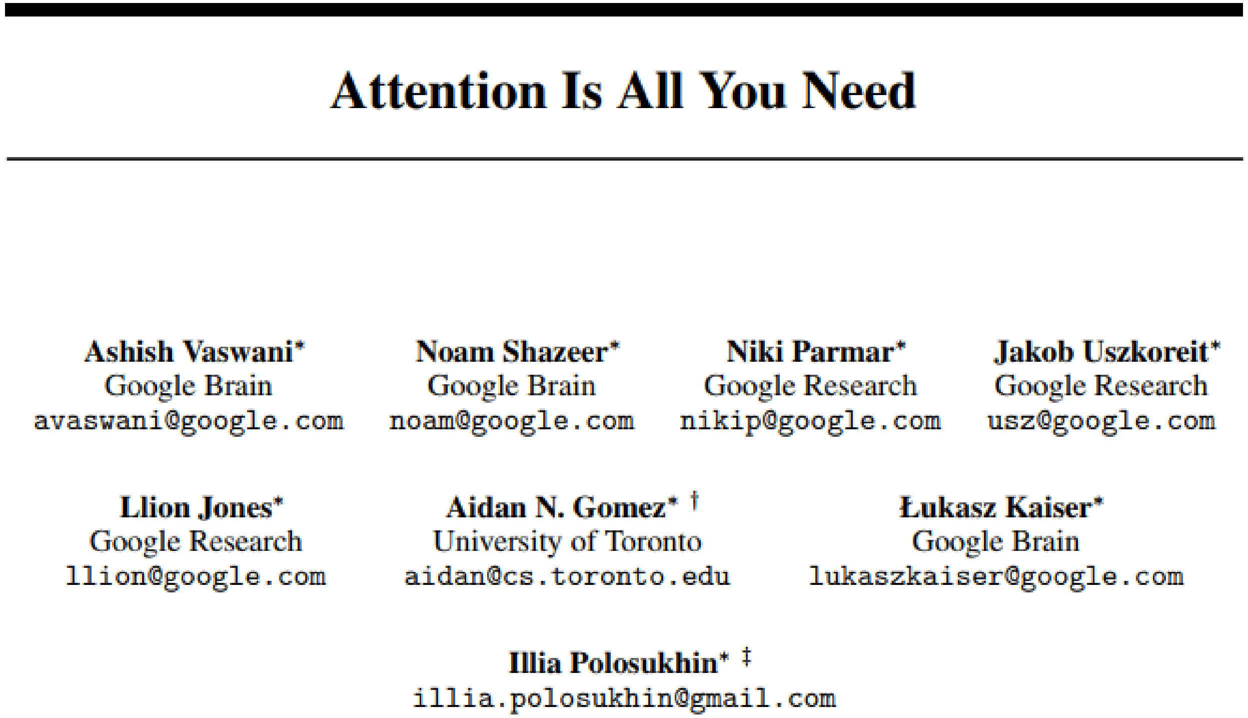
本文介绍一篇关于动态Transformer的最新工作: Not All Images are Worth 16x16 Words: Dynamic Visi on Transformers with Adaptive Sequence Length ,推理代码和部分预训练模型已经在Github上开源。 百度快照

Transformer 杀疯了,图像去雨、人脸幻构、风格迁移、语义分割等通...

人工智能cv 6月8日

前段时间 Transformer 已席卷计算机视觉领域，并获得大量好评，如『基于Swin-Transformer』、『美团提出具有「位置编码」的Transformer，性能优于ViT和DeiT』、『Lifting Transformer』、『TimeSformer』等等。近两天 Transformer 相关论文大量来袭...百度快照

Transformer



Attention Is All You Need

Ashish Vaswani*
Google Brain
`avaswani@google.com`

Noam Shazeer*
Google Brain
`noam@google.com`

Niki Parmar*
Google Research
`nikip@google.com`

Jakob Uszkoreit*
Google Research
`usz@google.com`

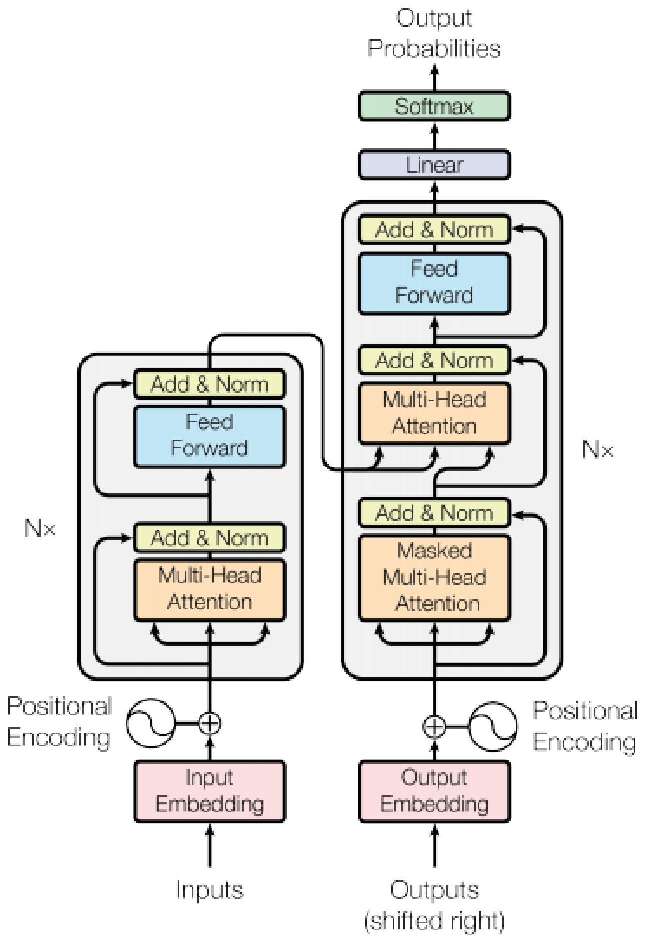
Llion Jones*
Google Research
`llion@google.com`

Aidan N. Gomez* †
University of Toronto
`aidan@cs.toronto.edu`

Łukasz Kaiser*
Google Brain
`lukaszkaiser@google.com`

Illia Polosukhin* ‡
`illia.polosukhin@gmail.com`

Vaswani, Ashish, et al. "Attention is All you Need." NIPS. 2017.



Development of X-formers(Attention)

2017

- Transformer

2018

- Image Transformer
- Memory Compressed Attention
- Local Transformer
- Average Attention
- Li et al., 2018

2019

- Star-Transformer
- Sparse Transformer
- BP-Transformer
- Axial Transformer
- Set Transformer
- Low-rank and locality constrained attention
- Gaussian Transformer
- Adaptive Attention Span
- Dynamic Routing

2020

- ETC
- Longformer
- BigBird
- Routing Transformer
- Reformer
- SAC
- Sparse Sinkhorn Attention
- Linear Transformer
- Performer
- Clustered Attention
- Linformer
- CSALR
- Predictive Attention Transformer
- RealFormer
- Hard-Coded Gaussian Attention
- Synthesizer
- Deshpande and Narasimhan, 2020
- Talking-head Attention
- Collaborative MHA
- Multi-Scale Transformer

2021

- RFA
- DPFP
- Informer
- Poolingformer
- Luna
- Nyströmformer
- LazyFormer
- CAMTL

🟦 sparse attention (position based)

🟧 sparse attention (content based)

🟥 linearized attention

🟩 query prototyping

🟪 memory compression

🟦 low-rank

🟨 prior attention

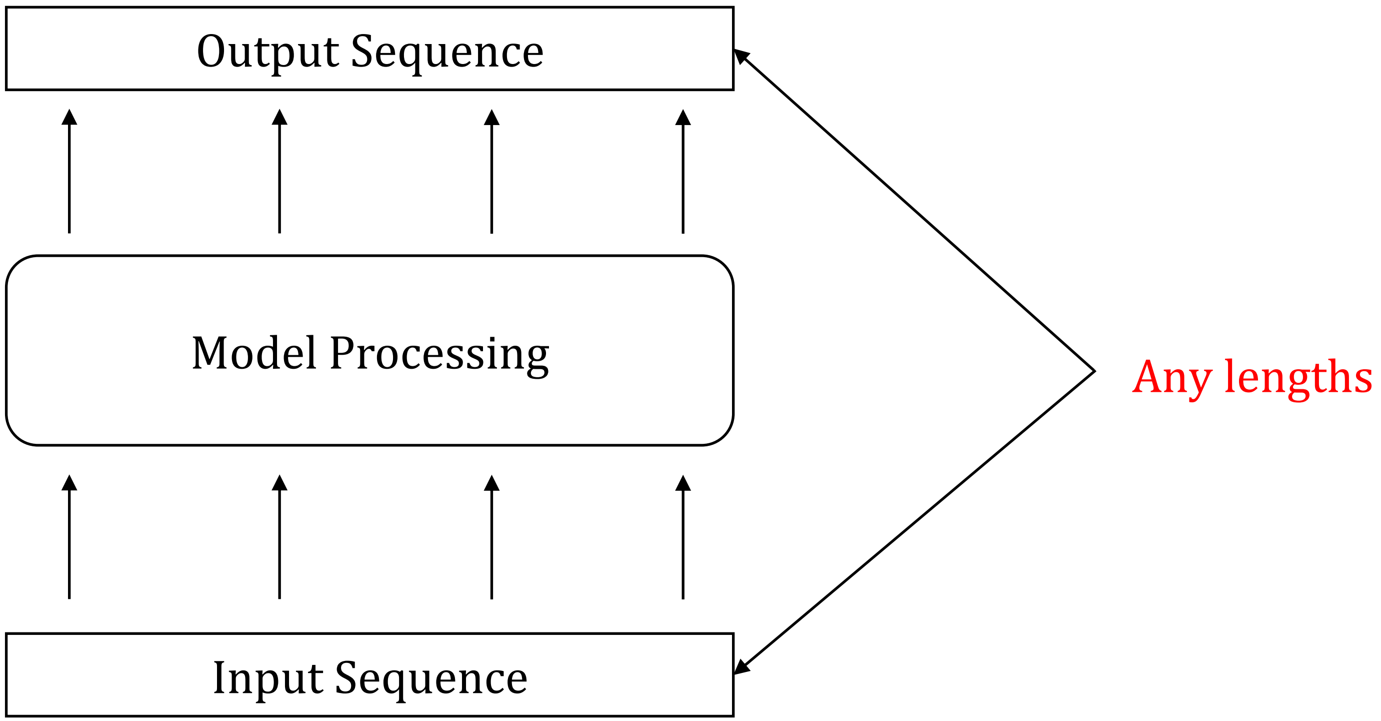
🟪 improved multi-head mechanism

Transformer Theory

The origin of Transformer

It all starts with a model called seq2seq(sequence to sequence).

Seq2seq



Output Sequence

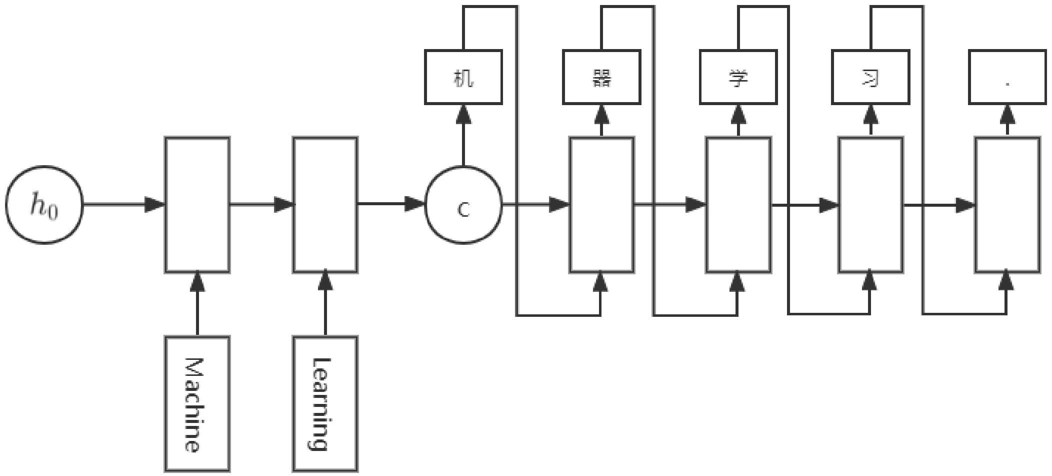
Model Processing

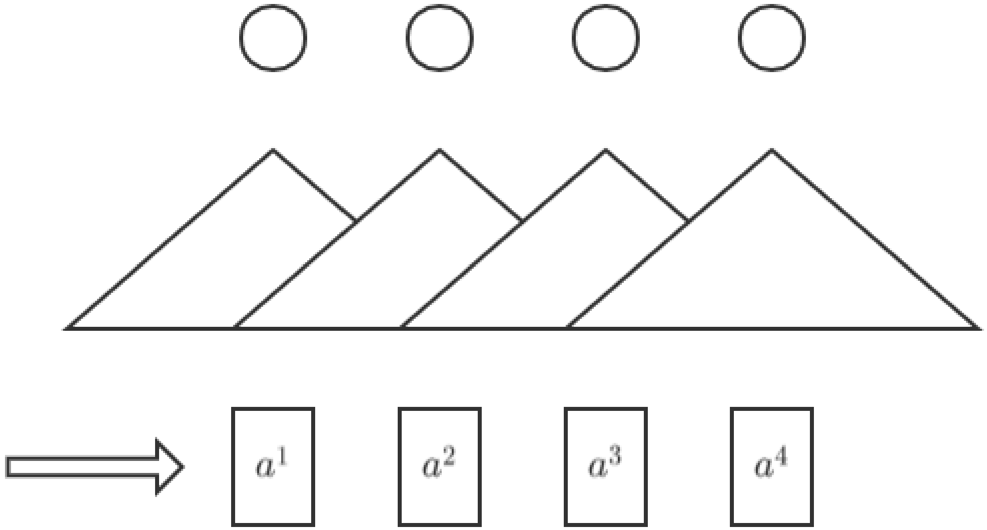
Input Sequence

Any lengths

Seq2seq

Before Transformer, People used RNN and CNN for seq2seq tasks.





Input sequence

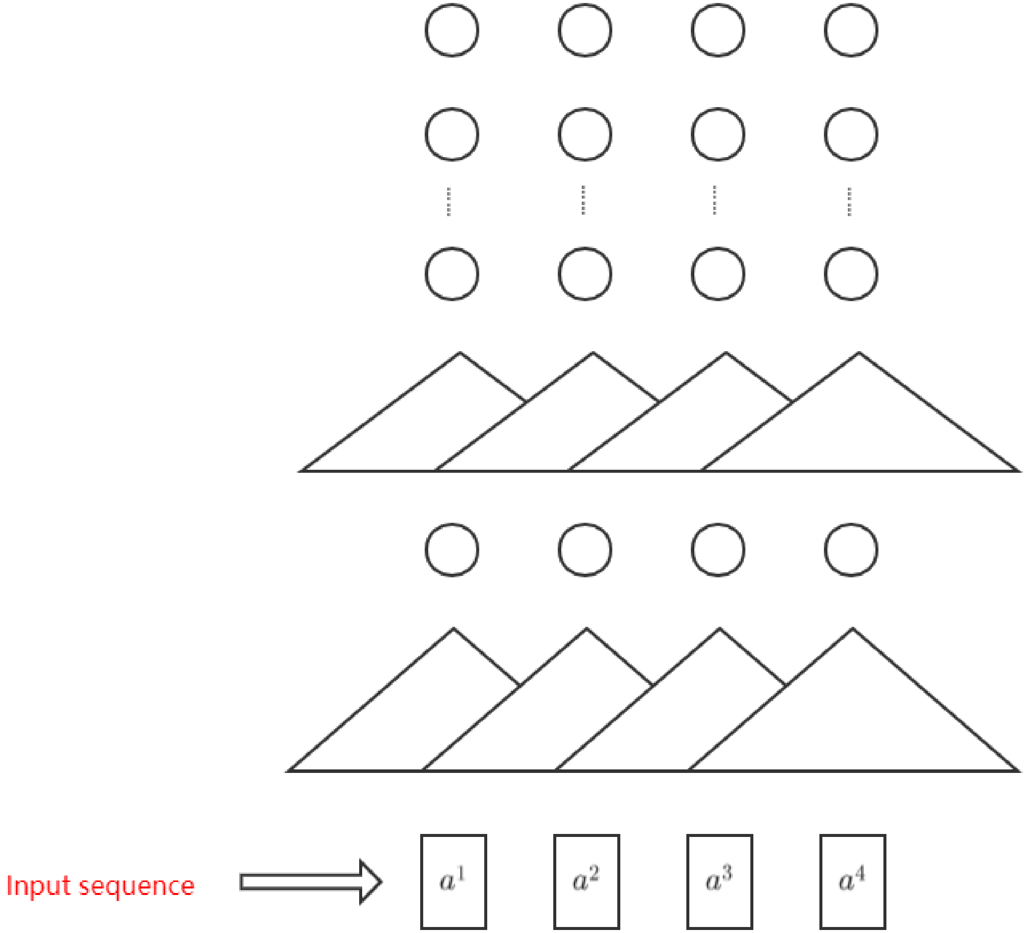
Advantages : Long Experience
Disadvantages: Less Parallel

Seq2seq

Before Transformer, People used RNN and CNN for seq2seq tasks.

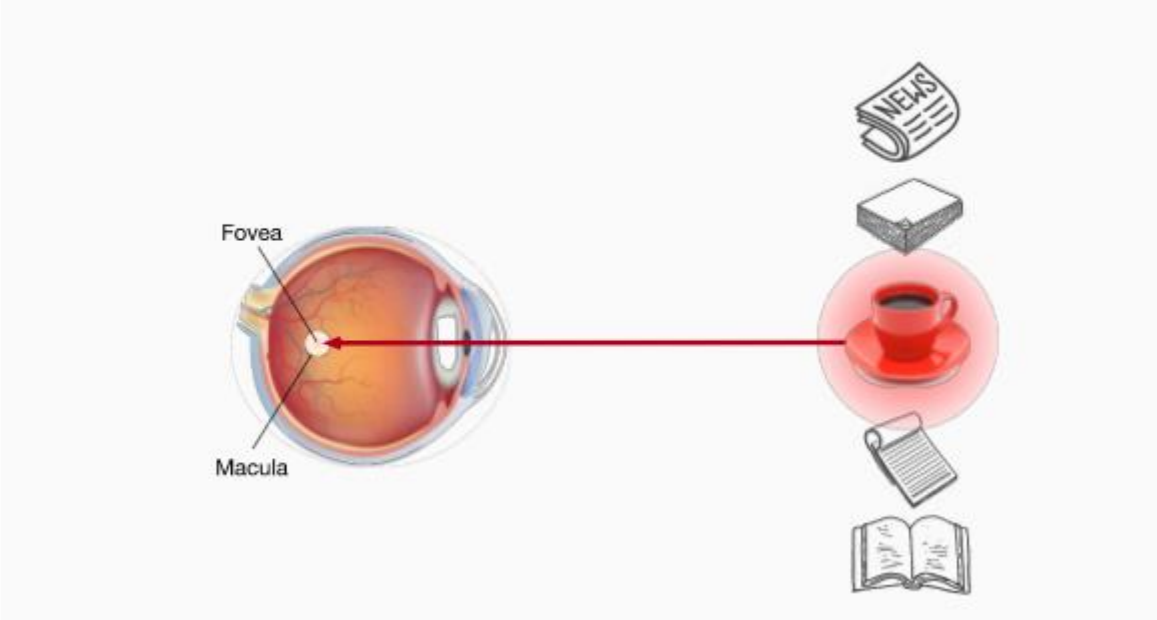
Is there a better model that can take into account long experiment and better parallelism at the same time?

Advantages : More parallel
Disadvantages： Less Experiment

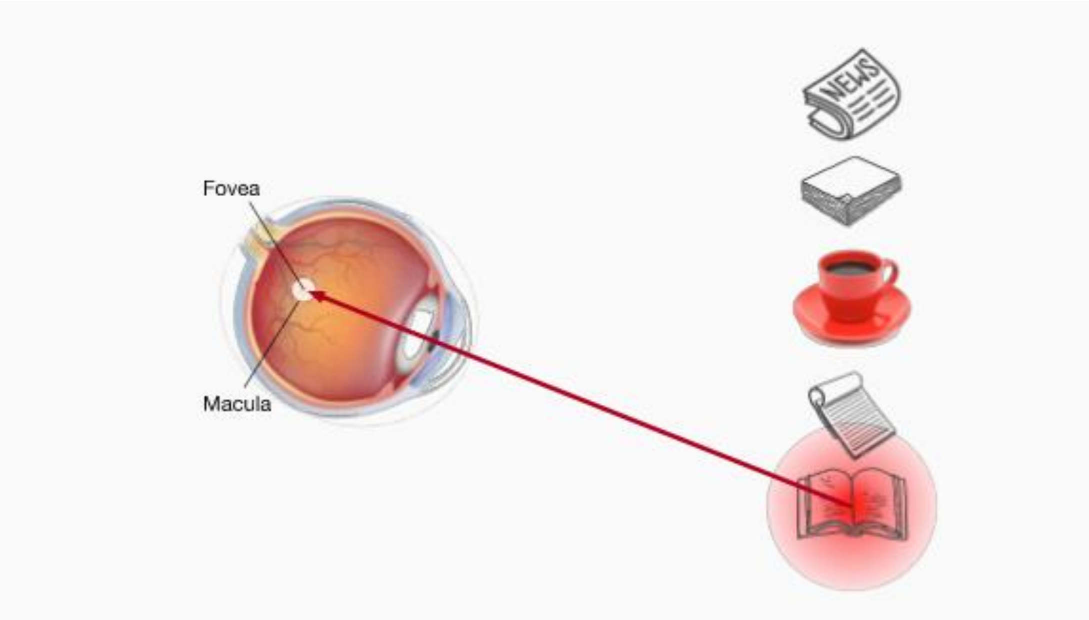


Attention Mechanism

Attention in biology



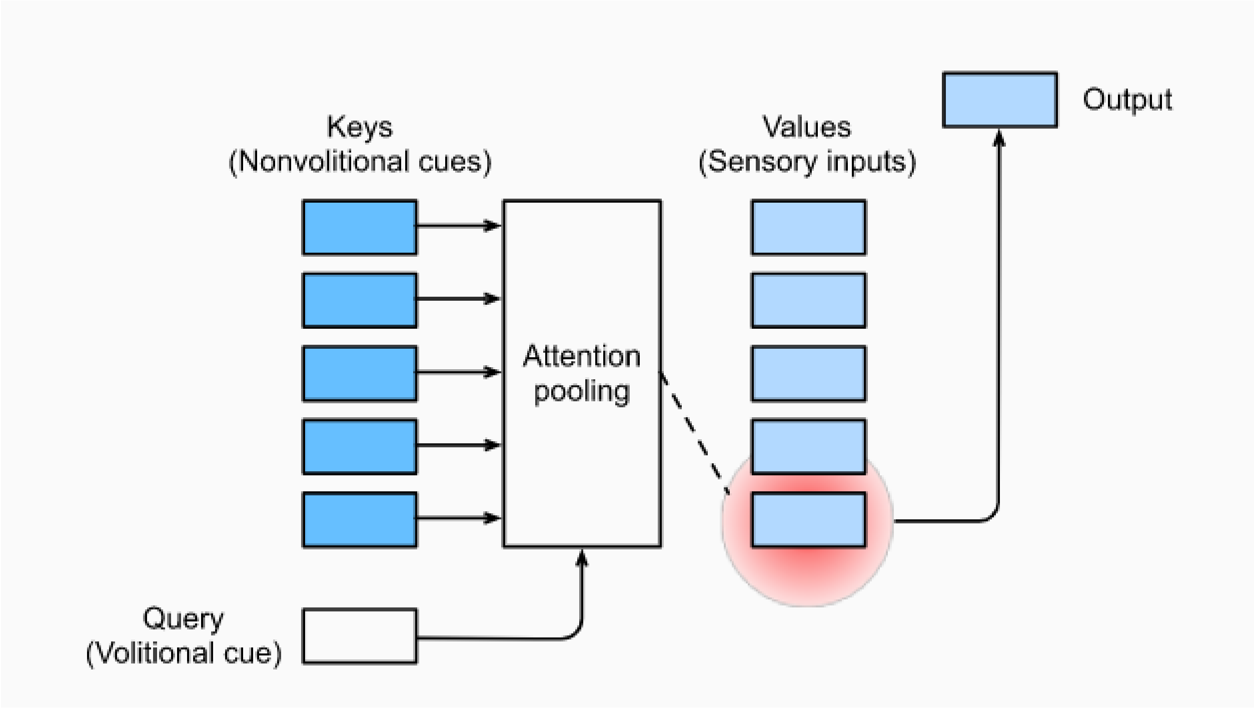
基于显著性的注意力



聚焦式注意力

Attention Mechanism

When the neural network processes a large amount of input information, it can learn from the attention mechanism of the human brain and only select some key information inputs for processing to improve the efficiency of the neural network.



Attention Mechanism

The calculation of the attention mechanism can be divided into two steps：

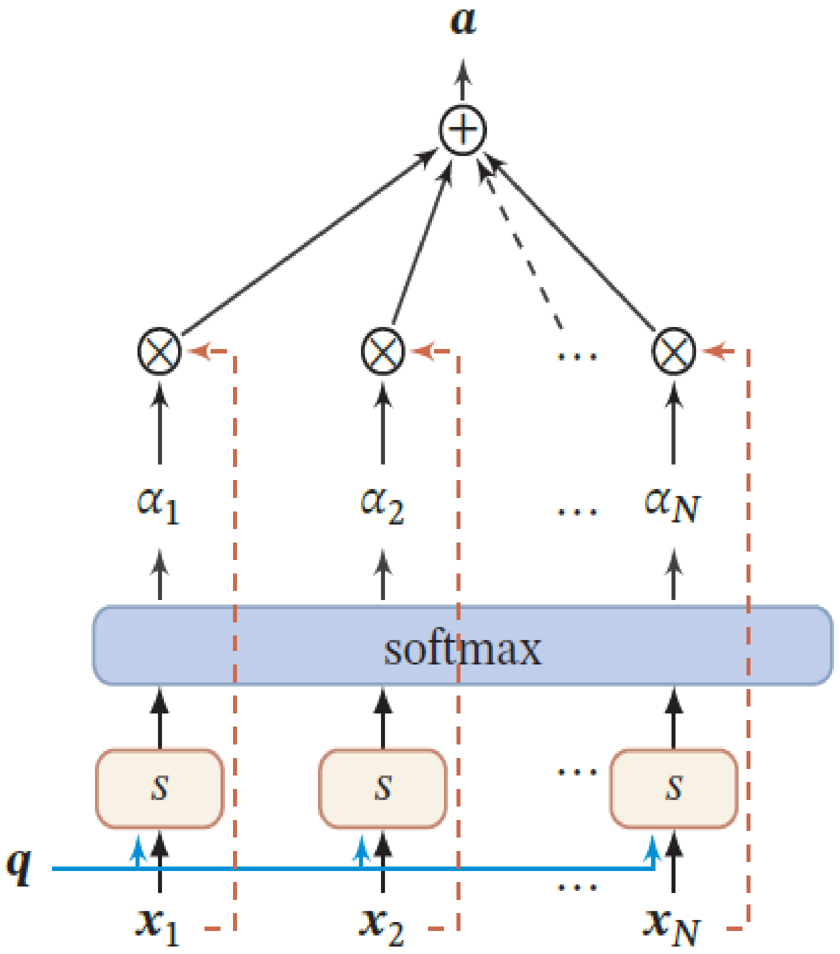
- ➢ Calculate the attention distribution on all input information

$$\begin{aligned}\alpha_n &= p(z = \mathrm{n}|\boldsymbol{X}, \boldsymbol{q}) \\ &= \mathrm{softmax}(s(\boldsymbol{x}_n, \boldsymbol{q})) \\ &= \frac{exp(s(\boldsymbol{x}_n, \boldsymbol{q}))}{\sum_{j=1}^{N}(s(\boldsymbol{x}_n, \boldsymbol{q}))}\end{aligned}$$

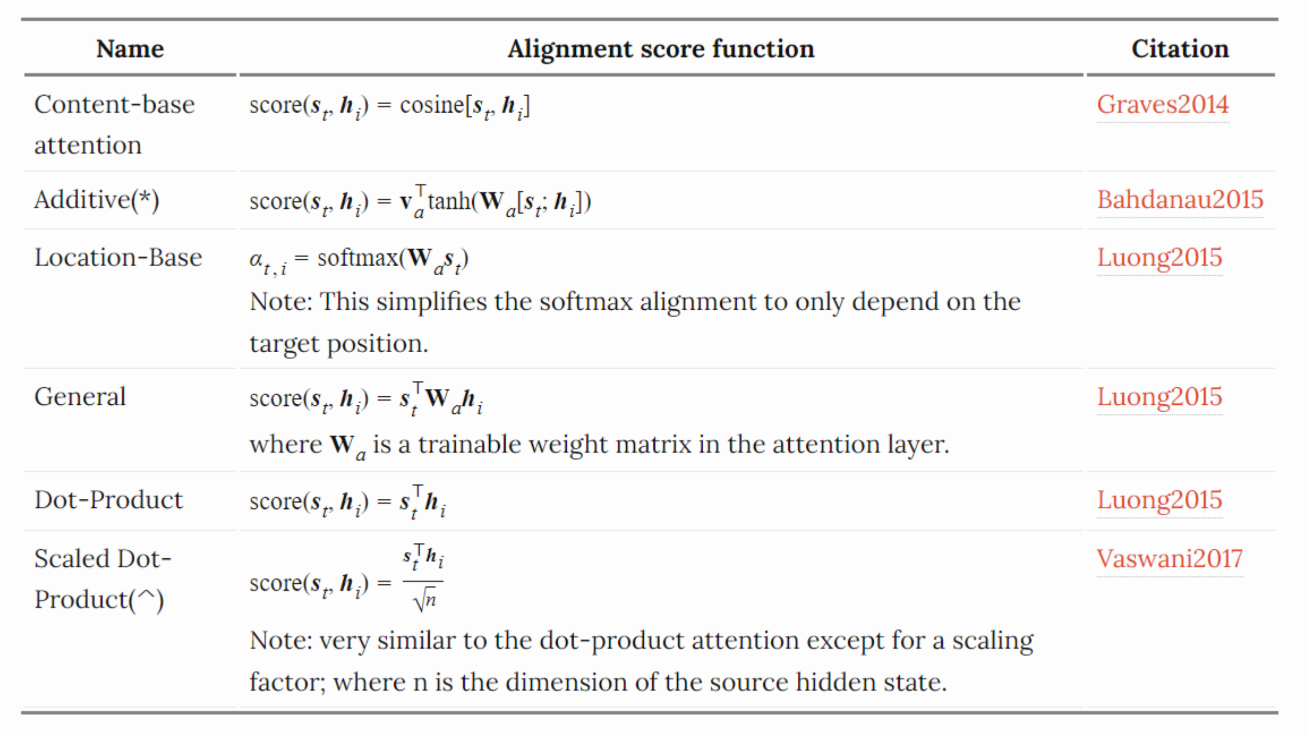
α_n *is* *attention distribution*， $\mathrm{s}(\boldsymbol{x}, \boldsymbol{q})$ is *attention scoring function*

- ➢ Calculate the weighted average of the input information according to the attention distribution

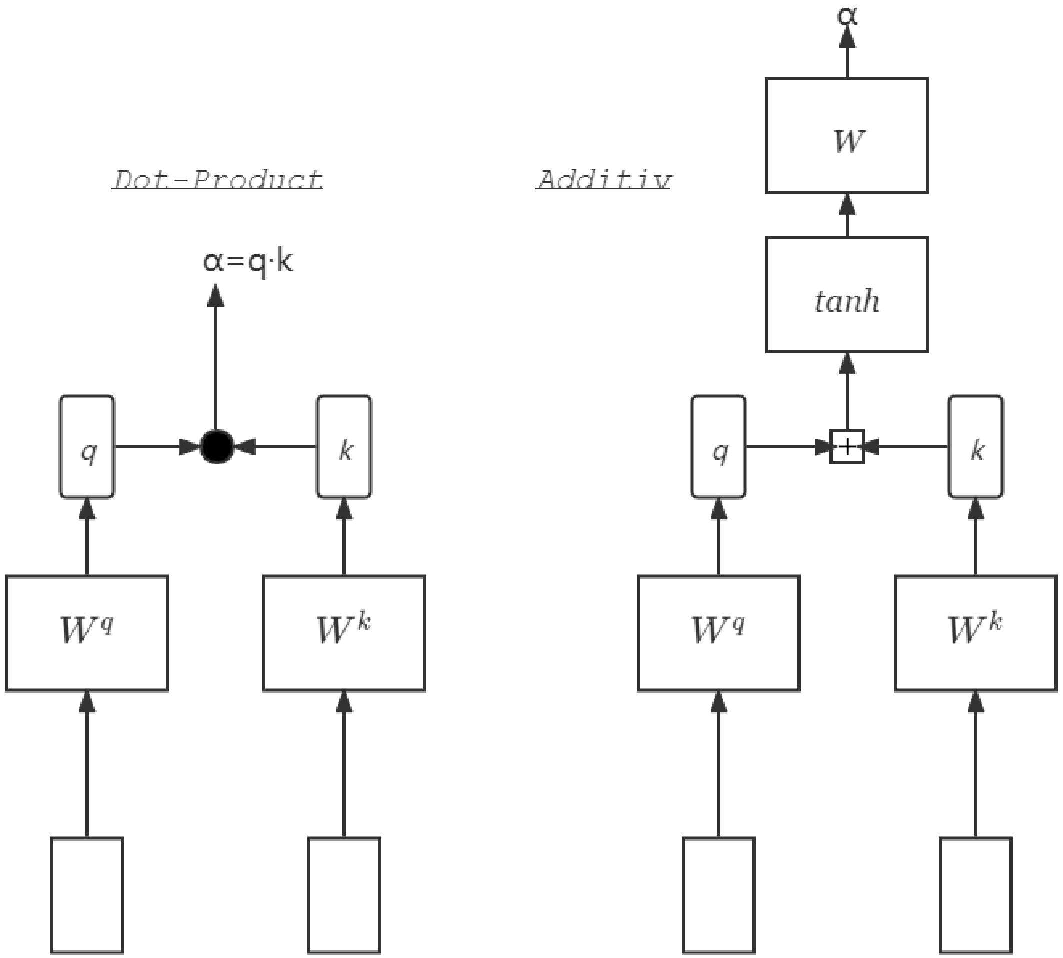
$$\begin{aligned}att(\boldsymbol{X}, \boldsymbol{q}) &= \sum_{n=1}^{N} \alpha_n \boldsymbol{x_n}, \\ &= E_{z\sim p(z|\boldsymbol{X},\boldsymbol{q})}[\boldsymbol{x}_z].\end{aligned}$$



Attention Mechanism



Name	Alignment score function	Citation
Content-base attention	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \text{cosine}[\boldsymbol{s}_t, \boldsymbol{h}_i]$	Graves2014
Additive(*)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\boldsymbol{s}_t; \boldsymbol{h}_i])$	Bahdanau2015
Location-Base	$\alpha_{t,i} = \text{softmax}(\mathbf{W}_a \boldsymbol{s}_t)$ Note: This simplifies the softmax alignment to only depend on the target position.	Luong2015
General	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \mathbf{W}_a \boldsymbol{h}_i$ where $\mathbf{W}_a$ is a trainable weight matrix in the attention layer.	Luong2015
Dot-Product	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \boldsymbol{h}_i$	Luong2015
Scaled Dot-Product(^)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \frac{\boldsymbol{s}_t^\top \boldsymbol{h}_i}{\sqrt{n}}$ Note: very similar to the dot-product attention except for a scaling factor; where n is the dimension of the source hidden state.	Vaswani2017



Self-attention and Transformer

Key-value pair attention

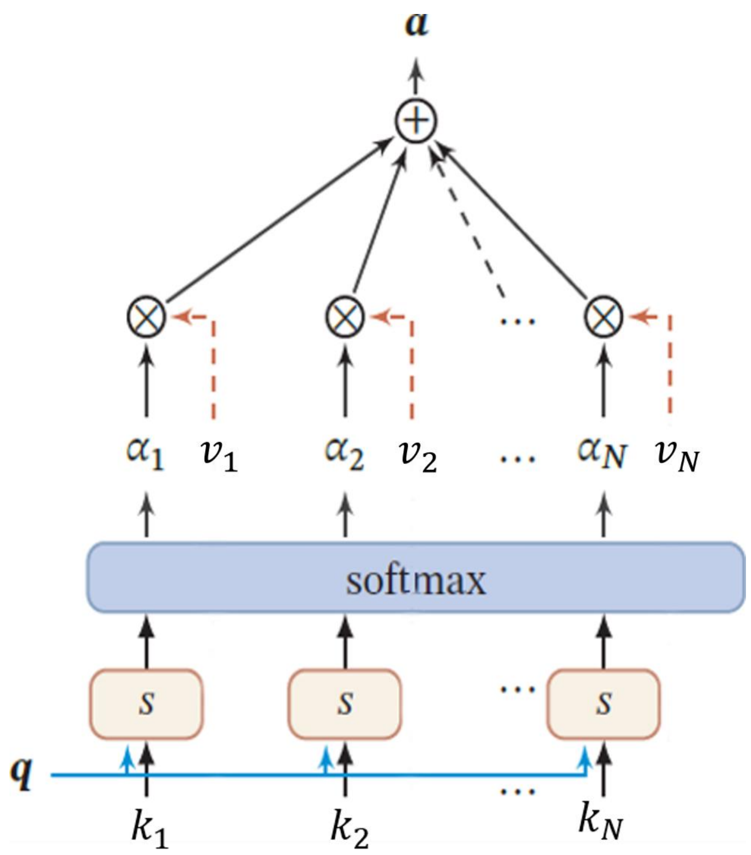
Use key-value pairs to represent input information, "key" to calculate the attention distribution α_n, and "value" to calculate aggregate information.

Use $(K, V) = [(k_1, v_1), \cdots, (k_N, v_N)]$ to represent the N group input information, the attention function is:

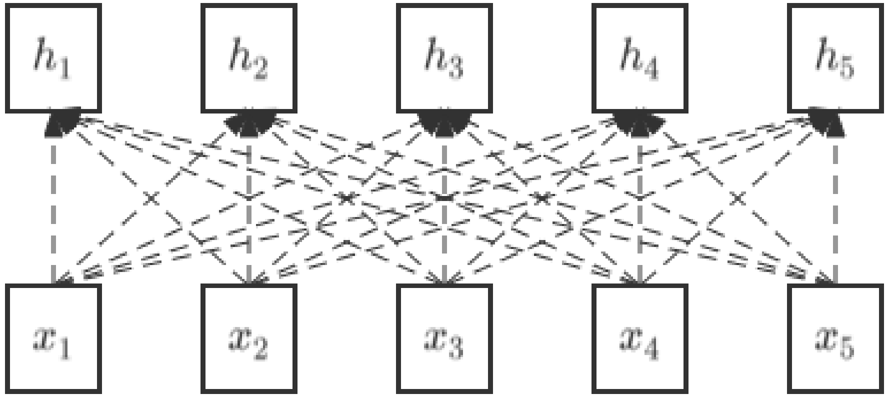
$$att\big((\boldsymbol{K}, \boldsymbol{V}), \boldsymbol{q}\big) = \sum_{n=1}^{N} \alpha_n \boldsymbol{v}_n ,$$

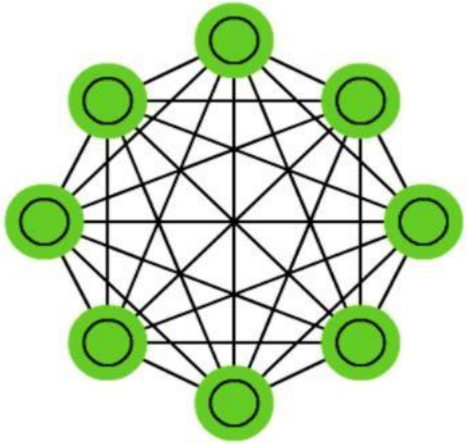
$$= \sum_{n=1}^{N} \frac{exp(s(\boldsymbol{k}_n, \boldsymbol{q}))}{\sum_j exp\big(s(\boldsymbol{k}_j, \boldsymbol{q})\big)} \boldsymbol{v}_n .$$

$s(\boldsymbol{k}_n, \boldsymbol{q})$ is attention scoring function.



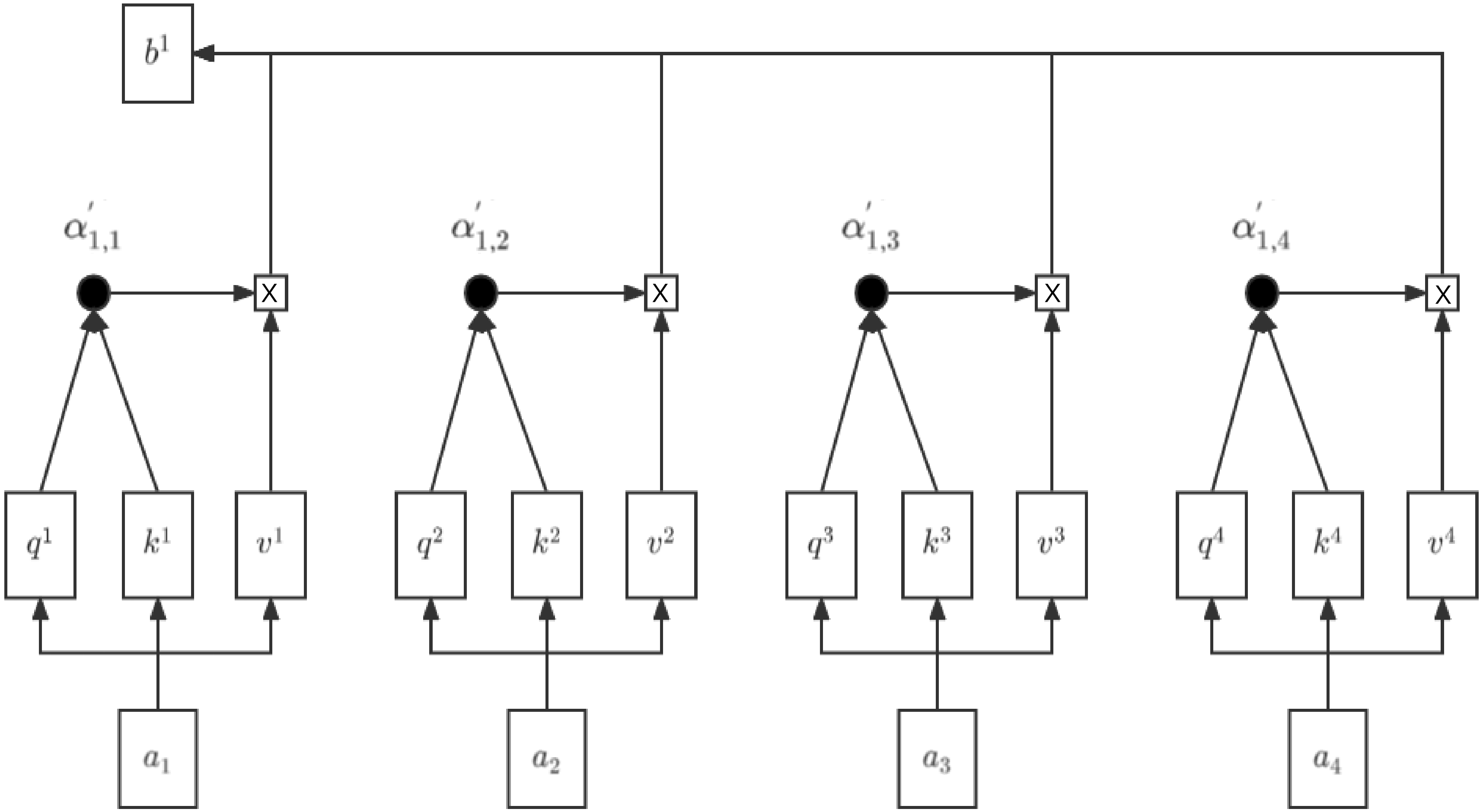
Self-Attention





Calculate the dynamic weights(α_{ij}) between elements.

Self-Attention



$$\boldsymbol{b^1} = \sum_i \alpha'_{1,i}\boldsymbol{v^i}$$

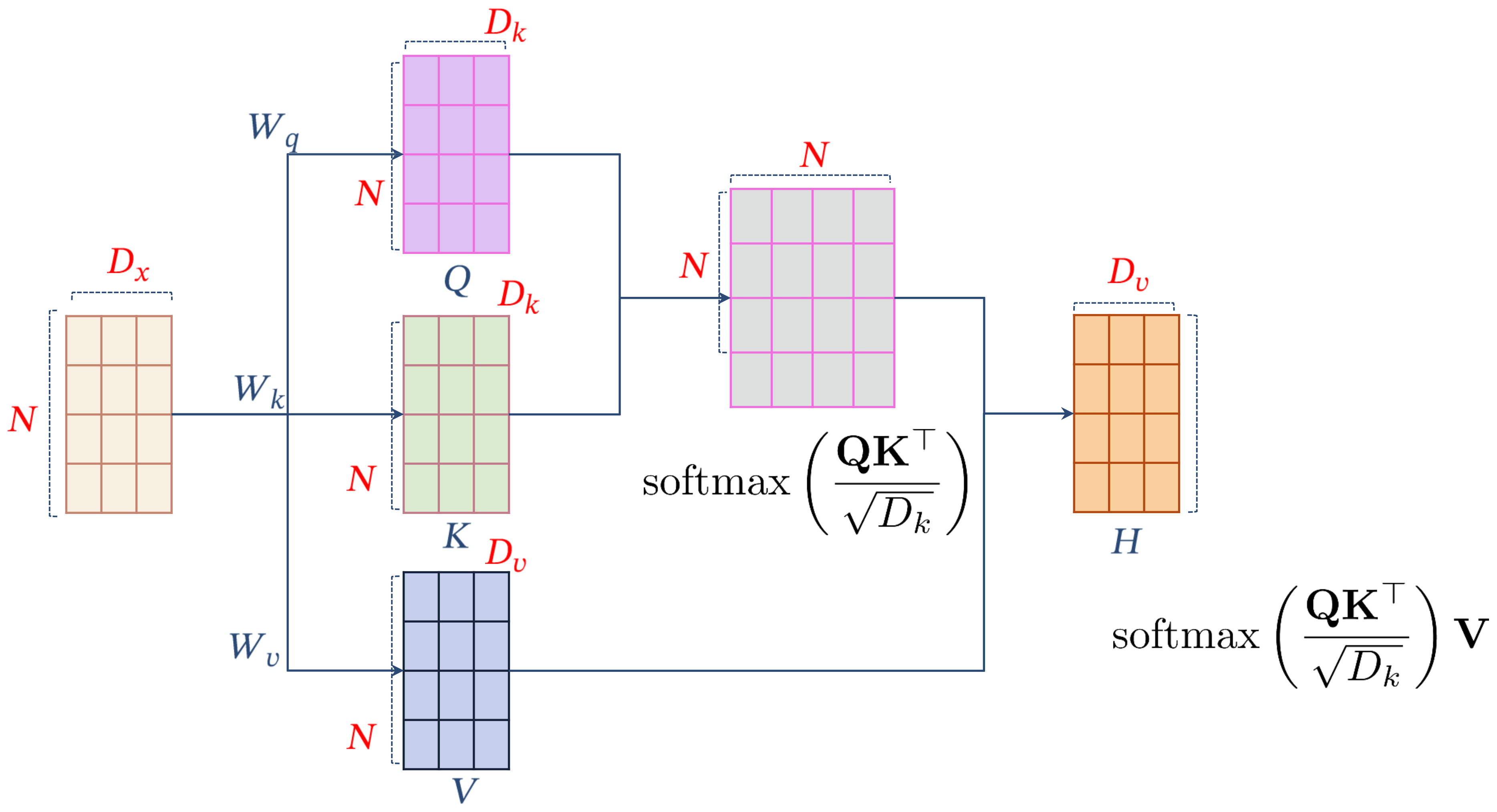
$\mathrm{q}^1 = a_1 \times \mathrm{W_q}$、$\mathrm{k}^1 = a_1 \times \mathrm{W_k}$
$\mathrm{v}^1 = a_1 \times \mathrm{W_v}$

$\mathrm{q}^2 = a_2 \times \mathrm{W_q}$、$\mathrm{k}^2 = a_2 \times \mathrm{W_k}$
$\mathrm{v}^2 = a_2 \times \mathrm{W_v}$

$\mathrm{q}^3 = a_3 \times \mathrm{W_q}$、$\mathrm{k}^3 = a_3 \times \mathrm{W_k}$
$\mathrm{v}^3 = a_3 \times \mathrm{W_v}$

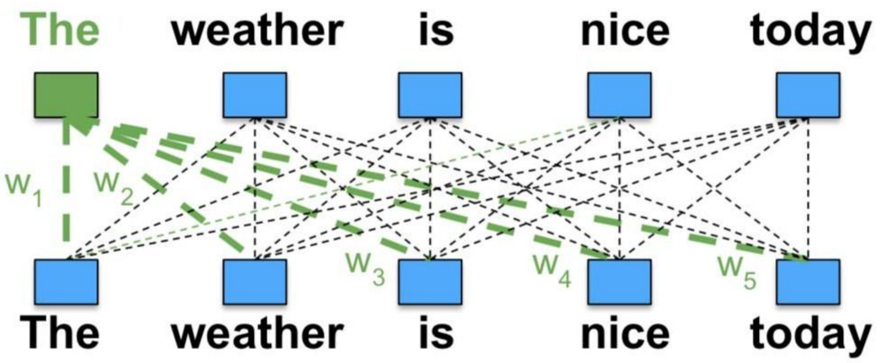
$\mathrm{q}^4 = a_4 \times \mathrm{W_q}$、$\mathrm{k}^4 = a_4 \times \mathrm{W_k}$
$\mathrm{v}^4 = a_4 \times \mathrm{W_v}$

Self-Attention



D_k
W_q
N
D_x
N
Q
D_k
W_k
N
K
D_v
W_v
N
V
N
N
softmax $\left(\frac{\mathbf{QK}^\top}{\sqrt{D_k}}\right)$
D_v
H
softmax $\left(\frac{\mathbf{QK}^\top}{\sqrt{D_k}}\right)\mathbf{V}$

A self-attention instance to model language

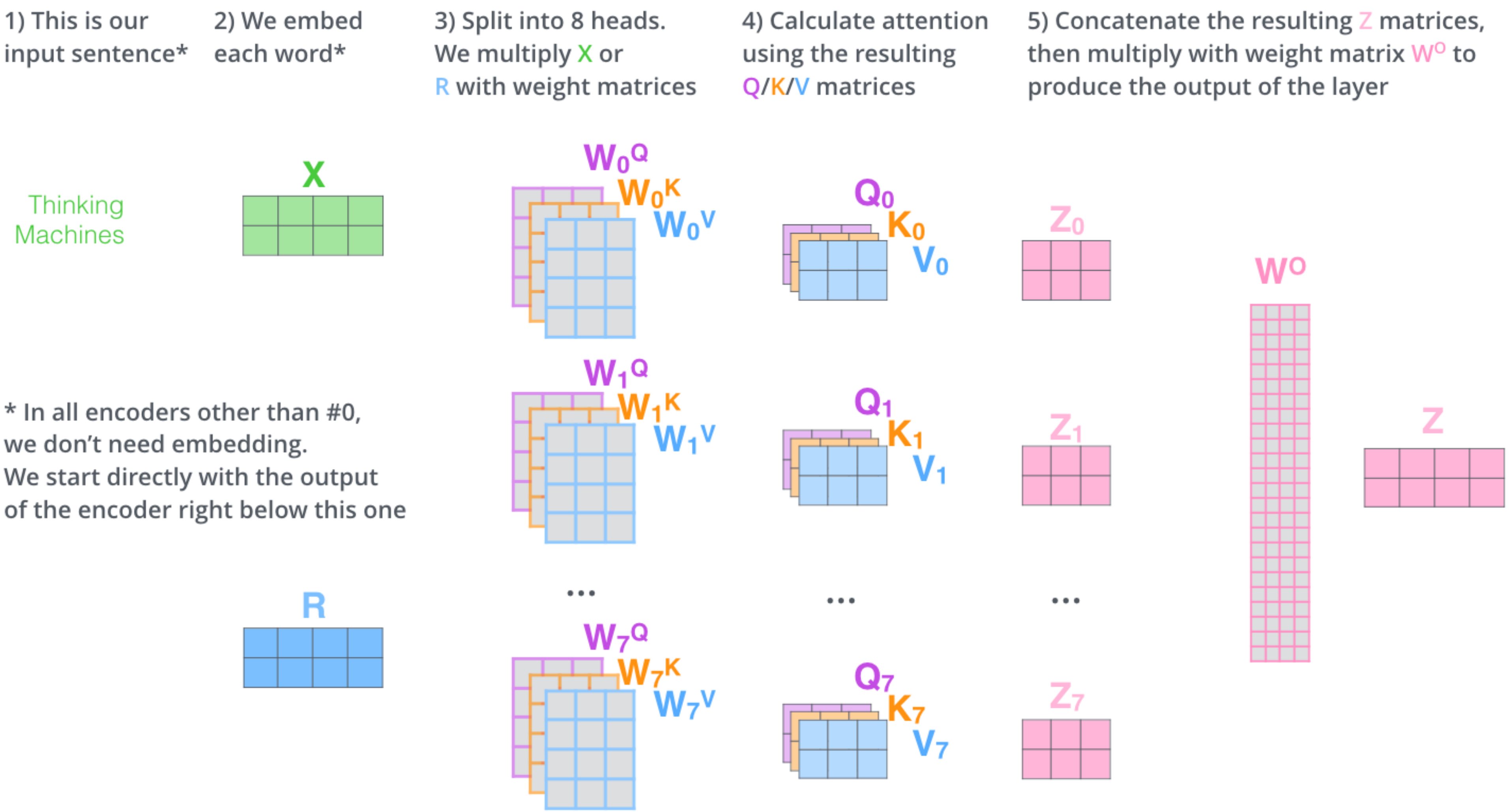


$$w_1 , w_2 , w_3 , w_4 , w_5 = \text{softmax}\left(\underbrace{\begin{bmatrix}0.6 & 0.2 & 0.8\end{bmatrix}}_{\text{The}} \times \begin{array}{ccccc} 0.6 & 0.2 & 0.9 & 0.4 & 0.4 \\ 0.2 & 0.3 & 0.1 & 0.1 & 0.1 \\ 0.8 & 0.1 & 0.8 & 0.4 & 0.6 \\ \text{The} & \text{weather} & \text{is} & \text{nice} & \text{today} \end{array}\right)$$

$$\underbrace{\begin{bmatrix}1.8\\2.3\\0.4\end{bmatrix}}_{\text{The}} = w_1 \times \underbrace{\begin{bmatrix}0.6\\0.2\\0.8\end{bmatrix}}_{\text{The}} + w_2 \times \underbrace{\begin{bmatrix}0.2\\0.3\\0.1\end{bmatrix}}_{\text{weather}} + w_3 \times \underbrace{\begin{bmatrix}0.9\\0.1\\0.8\end{bmatrix}}_{\text{is}} + w_4 \times \underbrace{\begin{bmatrix}0.4\\0.1\\0.4\end{bmatrix}}_{\text{nice}} + w_5 \times \underbrace{\begin{bmatrix}0.4\\0.1\\0.6\end{bmatrix}}_{\text{today}}$$

pic source：http://fuyw.top/NLP_02_QANet/

Multi-Head Attention



1) This is our input sentence*

2) We embed each word*

3) Split into 8 heads. We multiply X or R with weight matrices

4) Calculate attention using the resulting Q/K/V matrices

5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

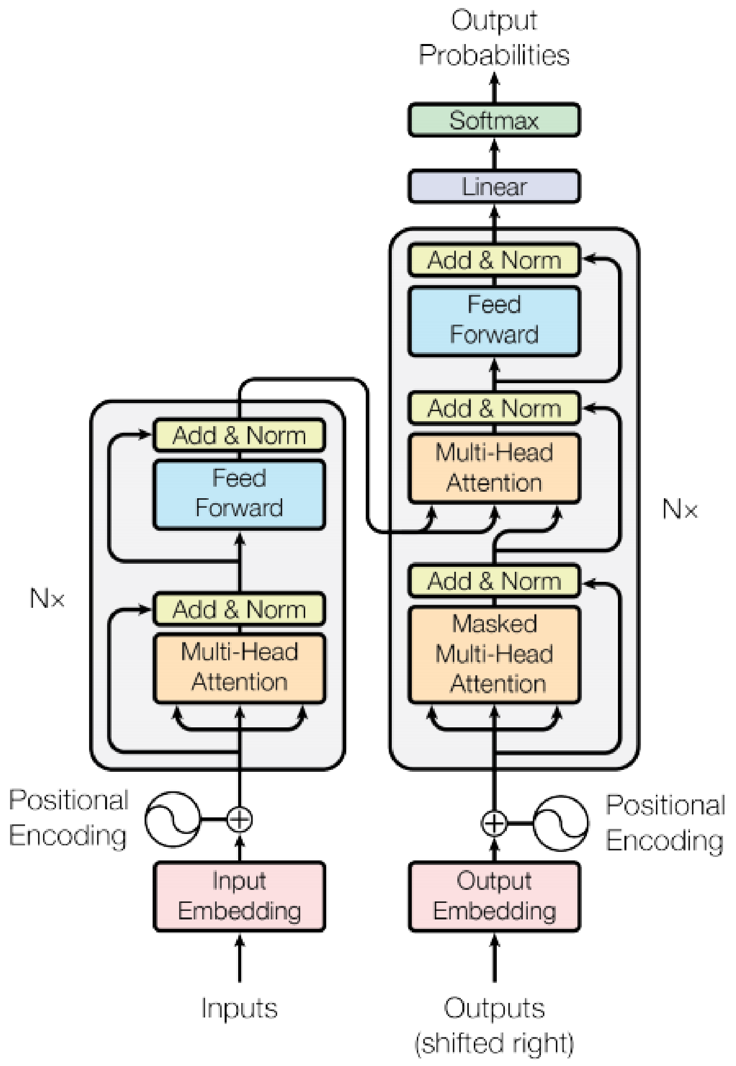
FFN、Positional Encoding and Sublayer

Both in Encoder and Decoder ,there are three module called FFN、Sublayer and Positional encoding.

$$\mathrm{FNN}(x) = \max(0, xW_1 + b_1)\,W_2 + b_2$$

$$sub - layer = LayerNorm(x + sublayer(x))$$

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$
$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$



Compare with other models

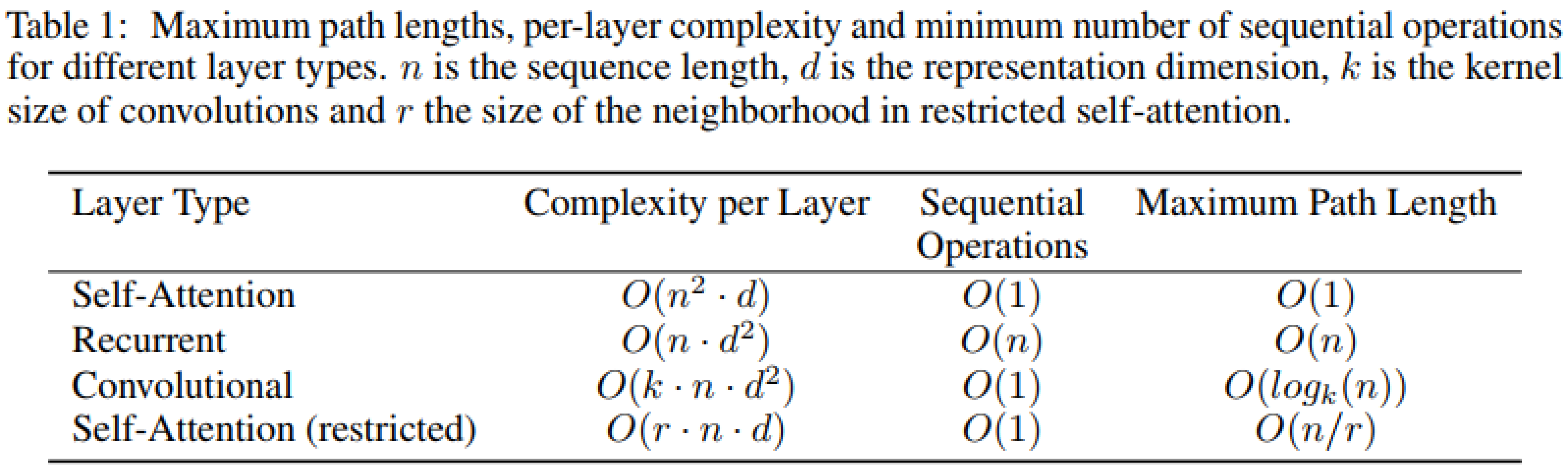


Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

- Self-Attention has constant max path length (like fully connected layer), which is suitable for long-range dependency modeling.
- It is more parallelizable than recurrent layers.
- It has global receptive field, thus doesn’t require stacking layers to model global dependencies (like convolutional layers).

Disadvantages of Transformer

➢ The acquisition of local information is not as strong as RNN and CNN.

Transformer is good at modeling two elements in a long distance, unlike RNN, which models similar elements and ignores elements that are far away.

➢ Modeling for long sequence inputs consumes more resources.

Transformer has too much calculation, and the complexity of self-attention is $O(T^2)$. Long input training is inefficient.

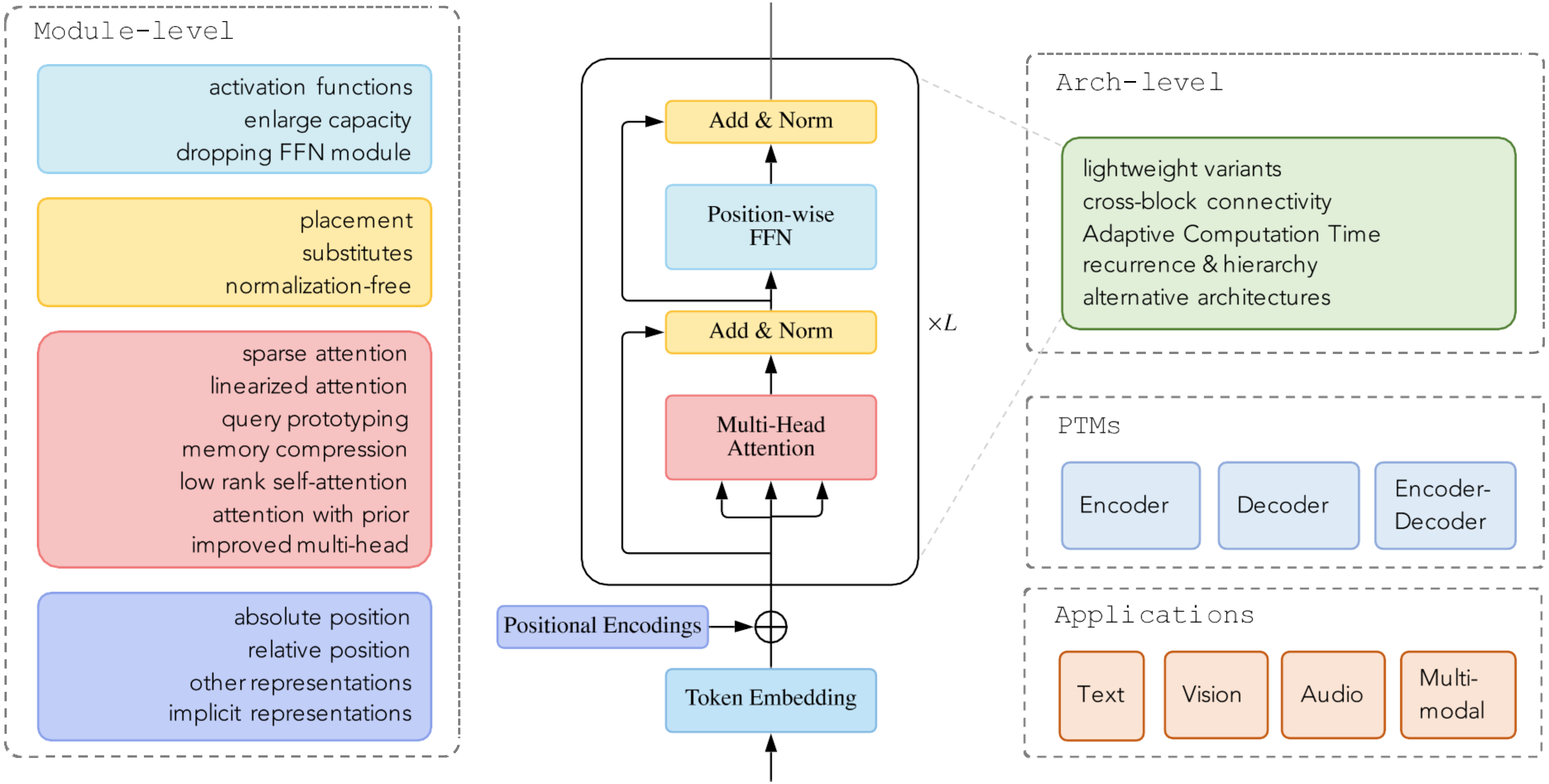
➢ Problem with Positional Encoding.

Position coding only serves as the index information of the location of the element, and the semantic information of the location of the element is not expressed.

➢ The top-level gradient disappears.

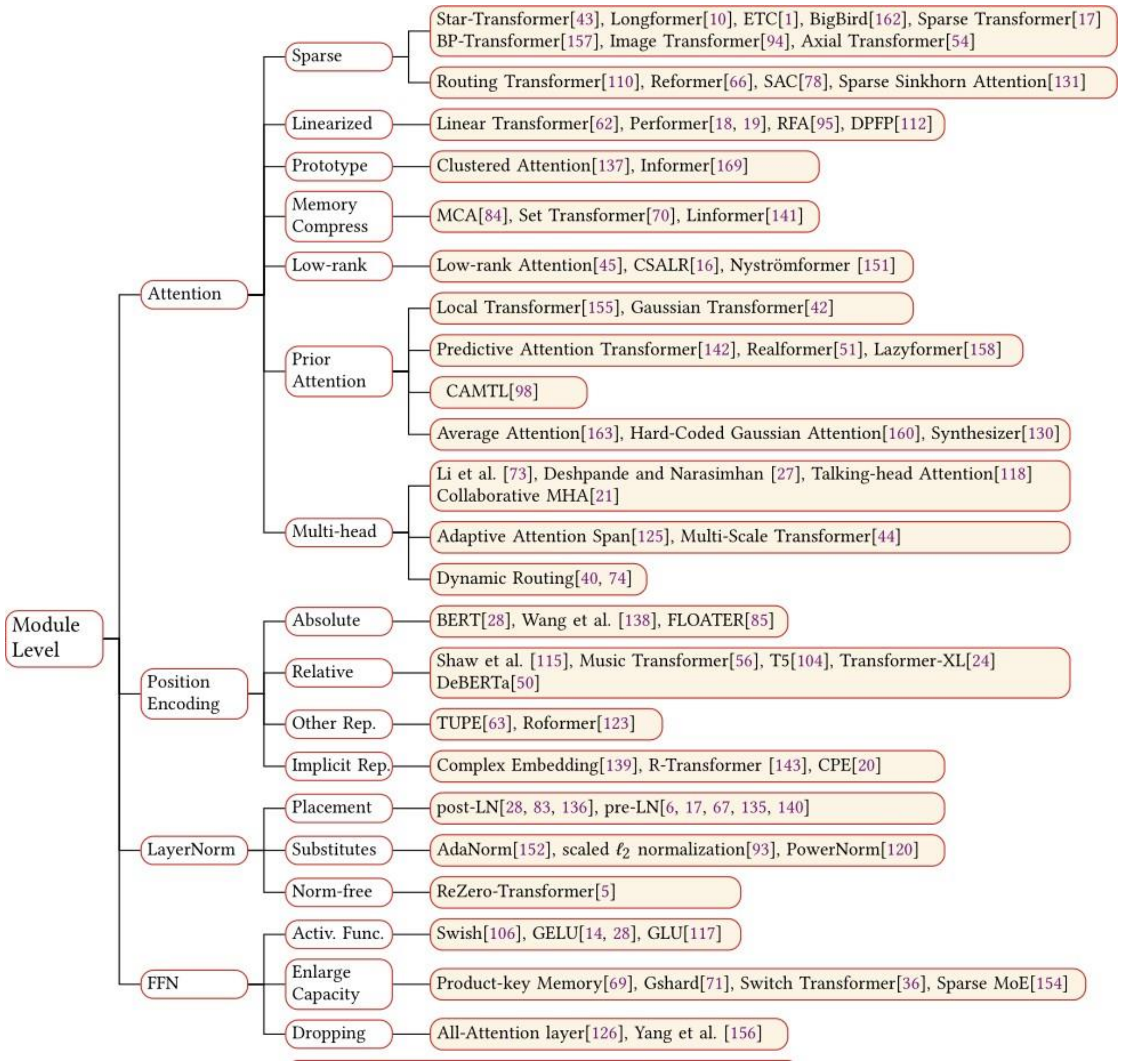
The layer normalization module is located between the two residual modules. Therefore, there is no direct connection between the final output layer and the previous Transformer layer, and the gradient flow will be blocked by the layer normalization module.

A taxonomy of X-formers(Transformer variants)

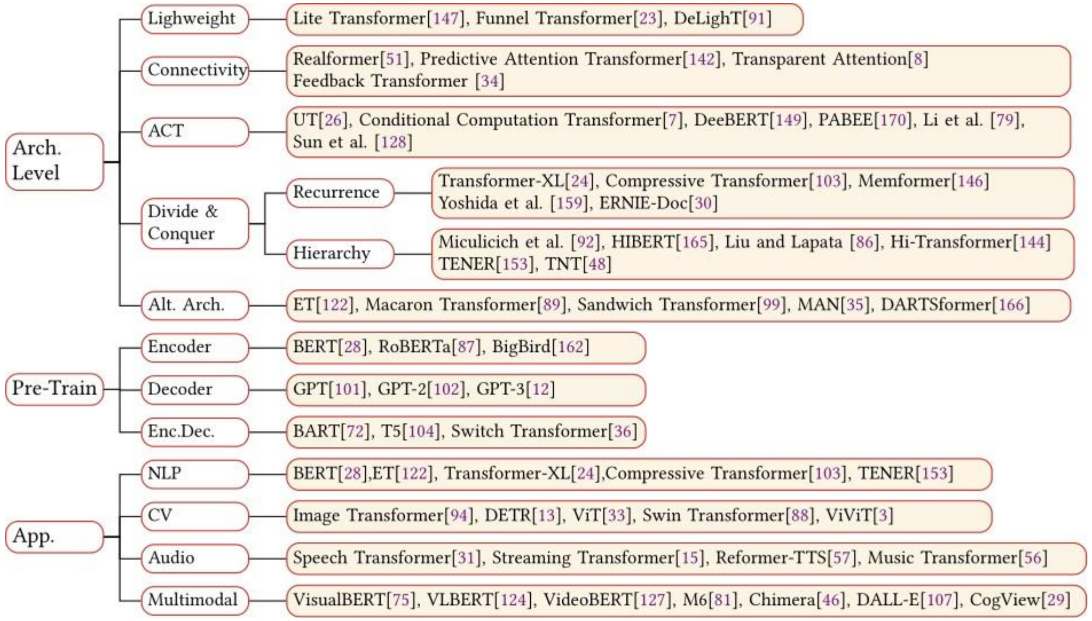


X-formers

Summary



- Module Level
  - Attention
    - Sparse
      - Star-Transformer[43], Longformer[10], ETC[1], BigBird[162], Sparse Transformer[17] BP-Transformer[157], Image Transformer[94], Axial Transformer[54]
      - Routing Transformer[110], Reformer[66], SAC[78], Sparse Sinkhorn Attention[131]
    - Linearized
      - Linear Transformer[62], Performer[18, 19], RFA[95], DPFP[112]
    - Prototype
      - Clustered Attention[137], Informer[169]
    - Memory Compress
      - MCA[84], Set Transformer[70], Linformer[141]
    - Low-rank
      - Low-rank Attention[45], CSALR[16], Nyströmformer [151]
    - Prior Attention
      - Local Transformer[155], Gaussian Transformer[42]
      - Predictive Attention Transformer[142], Realformer[51], Lazyformer[158]
      - CAMTL[98]
      - Average Attention[163], Hard-Coded Gaussian Attention[160], Synthesizer[130]
    - Multi-head
      - Li et al. [73], Deshpande and Narasimhan [27], Talking-head Attention[118] Collaborative MHA[21]
      - Adaptive Attention Span[125], Multi-Scale Transformer[44]
      - Dynamic Routing[40, 74]
  - Position Encoding
    - Absolute
      - BERT[28], Wang et al. [138], FLOATER[85]
    - Relative
      - Shaw et al. [115], Music Transformer[56], T5[104], Transformer-XL[24] DeBERTa[50]
    - Other Rep.
      - TUPE[63], Roformer[123]
    - Implicit Rep.
      - Complex Embedding[139], R-Transformer [143], CPE[20]
  - LayerNorm
    - Placement
      - post-LN[28, 83, 136], pre-LN[6, 17, 67, 135, 140]
    - Substitutes
      - AdaNorm[152], scaled ℓ_2 normalization[93], PowerNorm[120]
    - Norm-free
      - ReZero-Transformer[5]
  - FFN
    - Activ. Func.
      - Swish[106], GELU[14, 28], GLU[117]
    - Enlarge Capacity
      - Product-key Memory[69], Gshard[71], Switch Transformer[36], Sparse MoE[154]
    - Dropping
      - All-Attention layer[126], Yang et al. [156]



- Arch. Level
  - Lighweight
    - Lite Transformer[147], Funnel Transformer[23], DeLighT[91]
  - Connectivity
    - Realformer[51], Predictive Attention Transformer[142], Transparent Attention[8] Feedback Transformer [34]
  - ACT
    - UT[26], Conditional Computation Transformer[7], DeeBERT[149], PABEE[170], Li et al. [79], Sun et al. [128]
  - Divide & Conquer
    - Recurrence
      - Transformer-XL[24], Compressive Transformer[103], Memformer[146] Yoshida et al. [159], ERNIE-Doc[30]
    - Hierarchy
      - Miculicich et al. [92], HIBERT[165], Liu and Lapata [86], Hi-Transformer[144] TENER[153], TNT[48]
  - Alt. Arch.
    - ET[122], Macaron Transformer[89], Sandwich Transformer[99], MAN[35], DARTSformer[166]
- Pre-Train
  - Encoder
    - BERT[28], RoBERTa[87], BigBird[162]
  - Decoder
    - GPT[101], GPT-2[102], GPT-3[12]
  - Enc.Dec.
    - BART[72], T5[104], Switch Transformer[36]
- App.
  - NLP
    - BERT[28],ET[122], Transformer-XL[24],Compressive Transformer[103], TENER[153]
  - CV
    - Image Transformer[94], DETR[13], ViT[33], Swin Transformer[88], ViViT[3]
  - Audio
    - Speech Transformer[31], Streaming Transformer[15], Reformer-TTS[57], Music Transformer[56]
  - Multimodal
    - VisualBERT[75], VLBERT[124], VideoBERT[127], M6[81], Chimera[46], DALL-E[107], CogView[29]

Improvement Methods

- Model Efficiency
  - lightweight attention (e.g. sparse attention variants) and Divide-and-conquer methods(e.g., recurrent and hierarchical mechanism).
- Model Generalization
  - Since the transformer is a flexible architecture and makes few assumptions on the structural bias of input data, it is hard to train on small-scale data.
  - introducing structural bias or regularization, pre-training on large-scale unlabeled data, etc.
- Model Adaptation
  - adapting the Transformer to specific downstream tasks and applications.

Application in computer vision

Application to Image

Introduction

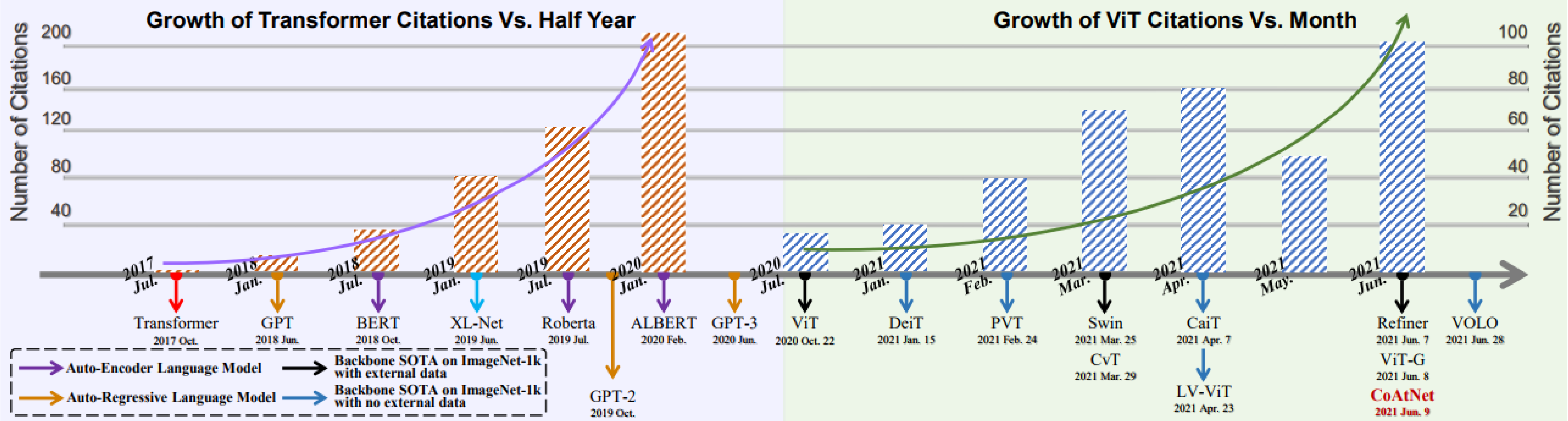
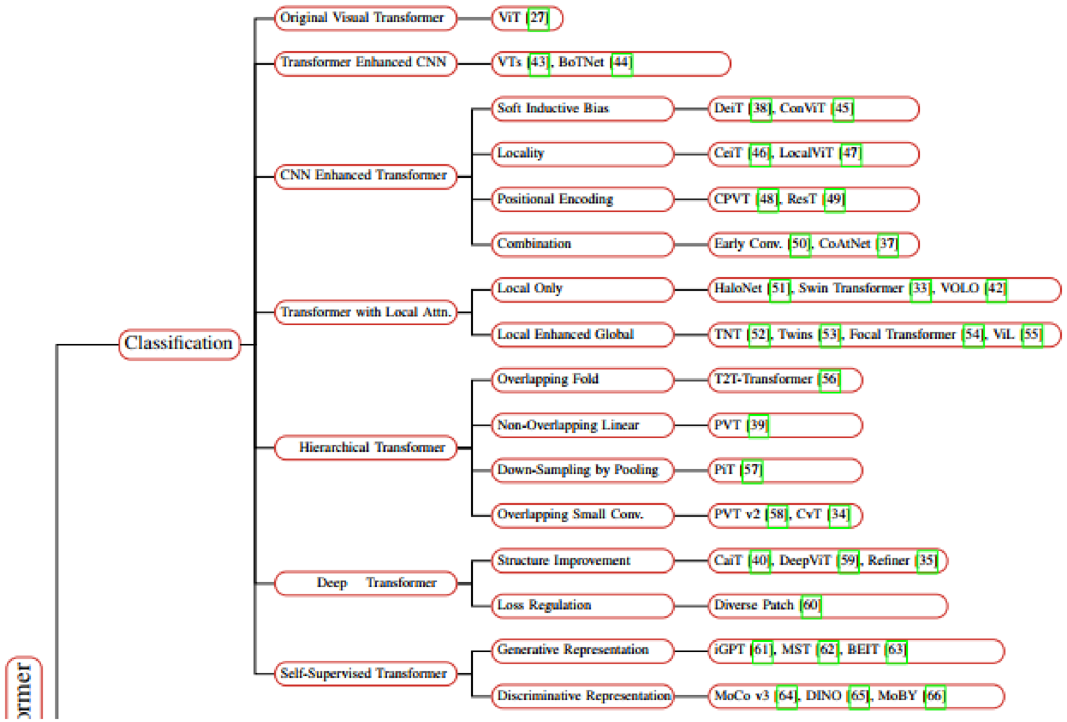


Fig. 1. Odyssey of Transformer application & Growth of both Transformer [1] and ViT [27] citations according to Google Scholar. (Upper Left) Growth of Transformer citations in multiple conference publication including: NIPS, ACL, ICML, IJCAI, ICLR, and ICASSP. (Upper Right) Growth of ViT citations in Arxiv publications. (Bottom Left) Odyssey of language model [1]–[8]. (Bottom Right) Odyssey of visual Transformer backbone where the black [27], [33]–[37] is the SOTA with external data and the blue [38]–[42] refers to the SOTA without external data (best viewed in color).

Inspired by the prominent developments of Transformer in NLP , many researchers attempt to introduce Transformer into image classification、detection and segmentation.

Taxonomy of Transformers



ormer

Classification

Original Visual Transformer
ViT [27]

Transformer Enhanced CNN
VTs [43], BoTNet [44]

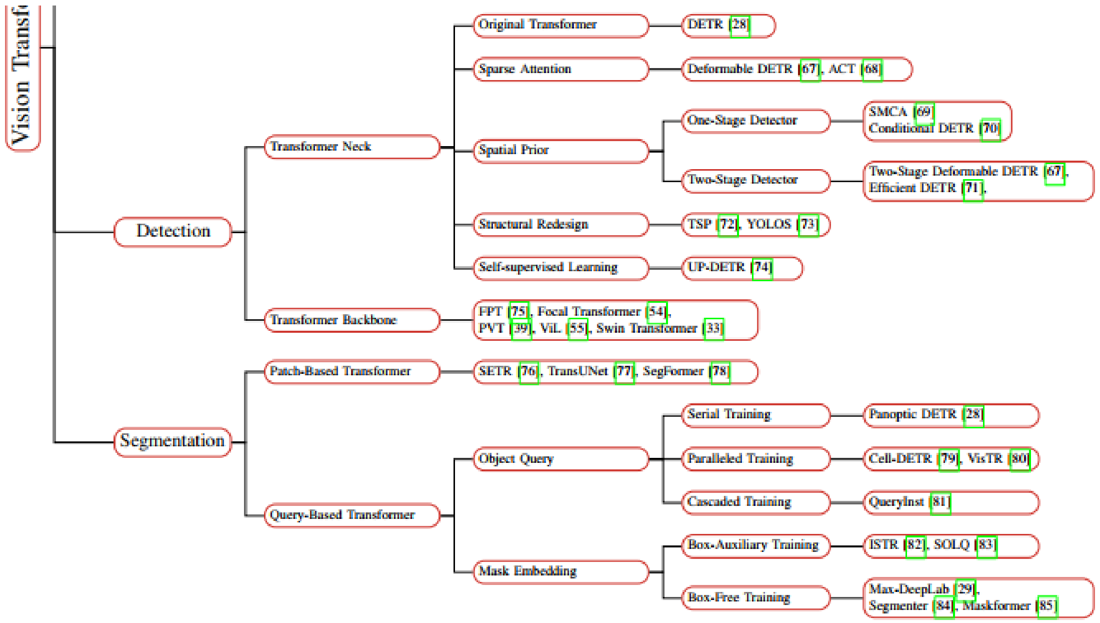
CNN Enhanced Transformer
Soft Inductive Bias
DeiT [38], ConViT [45]
Locality
CeiT [46], LocalViT [47]
Positional Encoding
CPVT [48], ResT [49]
Combination
Early Conv. [50], CoAtNet [37]

Transformer with Local Attn.
Local Only
HaloNet [51], Swin Transformer [33], VOLO [42]
Local Enhanced Global
TNT [52], Twins [53], Focal Transformer [54], ViL [55]

Hierarchical Transformer
Overlapping Fold
T2T-Transformer [56]
Non-Overlapping Linear
PVT [39]
Down-Sampling by Pooling
PiT [57]
Overlapping Small Conv.
PVT v2 [58], CvT [34]

Deep Transformer
Structure Improvement
CaiT [40], DeepViT [59], Refiner [35]
Loss Regulation
Diverse Patch [60]

Self-Supervised Transformer
Generative Representation
iGPT [61], MST [62], BEIT [63]
Discriminative Representation
MoCo v3 [64], DINO [65], MoBY [66]



Vision Transf

Detection

Transformer Neck
Original Transformer
DETR [28]
Sparse Attention
Deformable DETR [67], ACT [68]
Spatial Prior
One-Stage Detector
SMCA [69]
Conditional DETR [70]
Two-Stage Detector
Two-Stage Deformable DETR [67],
Efficient DETR [71],
Structural Redesign
TSP [72], YOLOS [73]
Self-supervised Learning
UP-DETR [74]

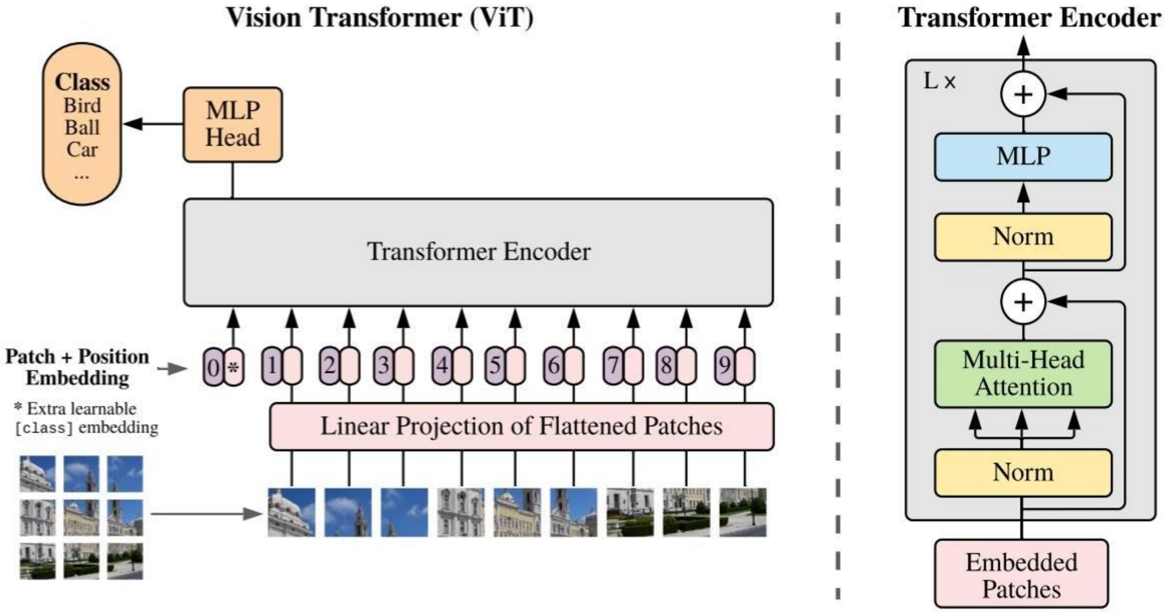
Transformer Backbone
FPT [75], Focal Transformer [54],
PVT [39], ViL [55], Swin Transformer [33]

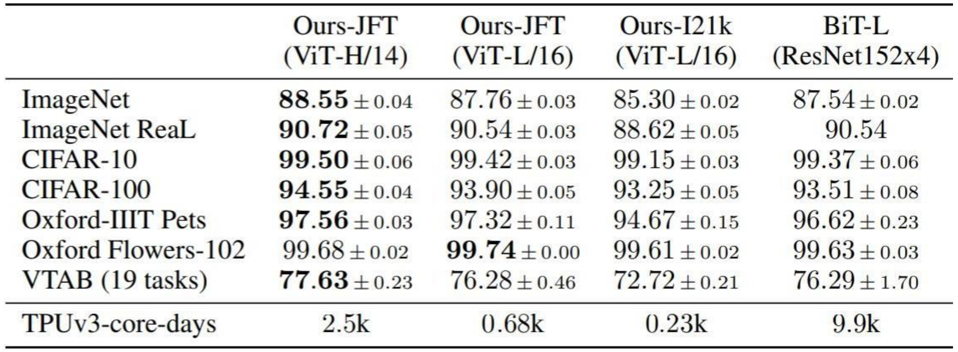
Segmentation

Patch-Based Transformer
SETR [76], TransUNet [77], SegFormer [78]

Query-Based Transformer
Object Query
Serial Training
Panoptic DETR [28]
Paralleled Training
Cell-DETR [79], VisTR [80]
Cascaded Training
QueryInst [81]
Mask Embedding
Box-Auxiliary Training
ISTR [82], SOLQ [83]
Box-Free Training
Max-DeepLab [29],
Segmenter [84], Maskformer [85]

Transformer For Classification





	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k

Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).

Vit-based improvements

- Transformer Enhanced CNN
  - Transformer has theoretically more powerful modeling capabilities than CNN.

    Relate Papers: VTs and BoTNet
- CNN Enhanced Transformer
  - The strong inductive bias limits CNN. This kind of work attempts to use appropriate convolution bias to enhance Transformer and accelerate its convergence . Soft Inductive Bias (DeiT and ConViT)、 Locality (CeiT and LocalViT)、 Positional Encoding (CPVT and ResT)、 Combination (Early Conv and CoAyNet)
- Transformer with Local Attn
  - In order to enhance the local feature extraction ability and retain the unconvolutional structure, some works try to adapt the patch structure through the local self-attention mechanism.

    Relate Papers: Local Only (HaloNet , Swin Transformer , VOLO)、 Local Enhanced Global (TNT ,Twins Focal Transformer ,ViL)

Vit-based improvements

- Hierarchical Transformer
  - Since ViT inherits the original columnar structure with a fixed resolution throughout the network, it ignores fine-grained features and brings expensive computational costs. Following the hierarchical CNN, recent work applies a similar structure to Transformer.

    Relate Papers : Overlapping Fold (T2T-Transformer)、Non-Overlapping Linear (PVT)、Down-Sampling by Pooling (PiT)、Overlapping Small Conv. (PVT v2 ,CvT)
- Deep Transformer
  - Increasing the depth of the model allows the network to learn more complex representations . Analyze the similarity of cross-patch (Diverse Patch) and cross-layer (Refiner, DeepViT) and the contribution of residual block (CaiT) to study its scalability
- Self-supervised Transformer
  - Recent work has also tried to design various self-supervised learning schemes for the visual Transformer in generative (iGPT, BEiT) and discriminative (MoCo v3, DINO).

Backbone of visual Transformer

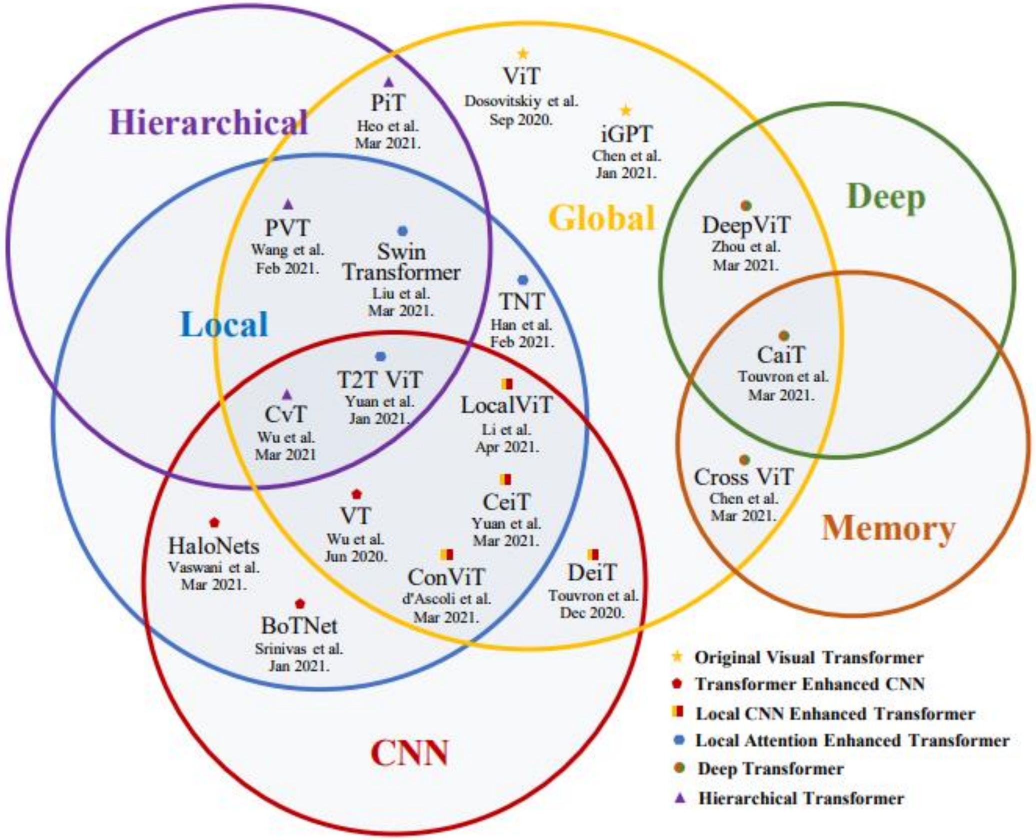


Fig. 5. Taxonomy of Visual Transformer Backbone (best viewed in color).

Since Vit proposed in October 2020, the number of applied research on Transformer in computer vision has exploded . More than 40 kinds of Transformer Model were proposed until now.

Various variants of Transformer performance

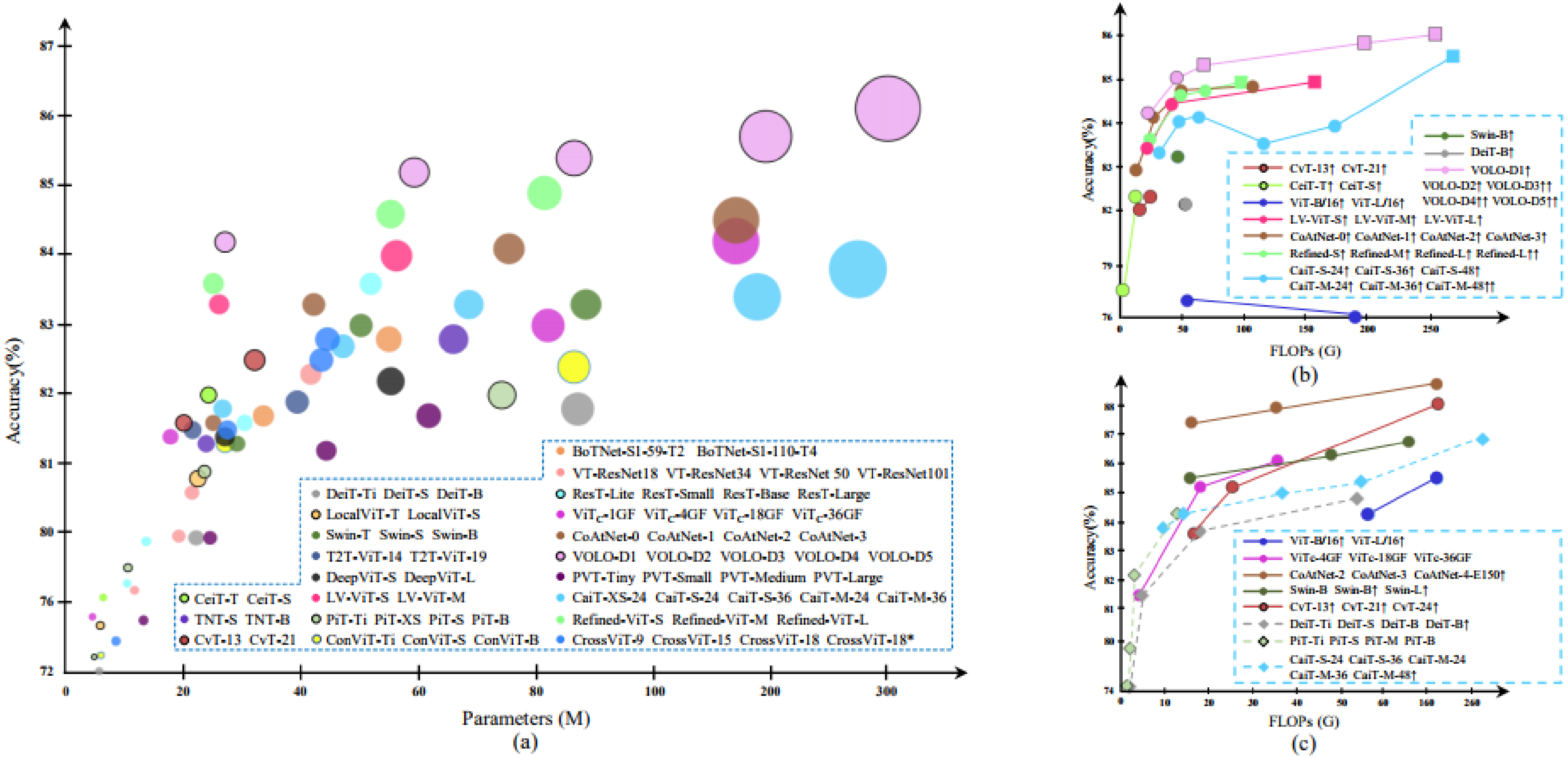
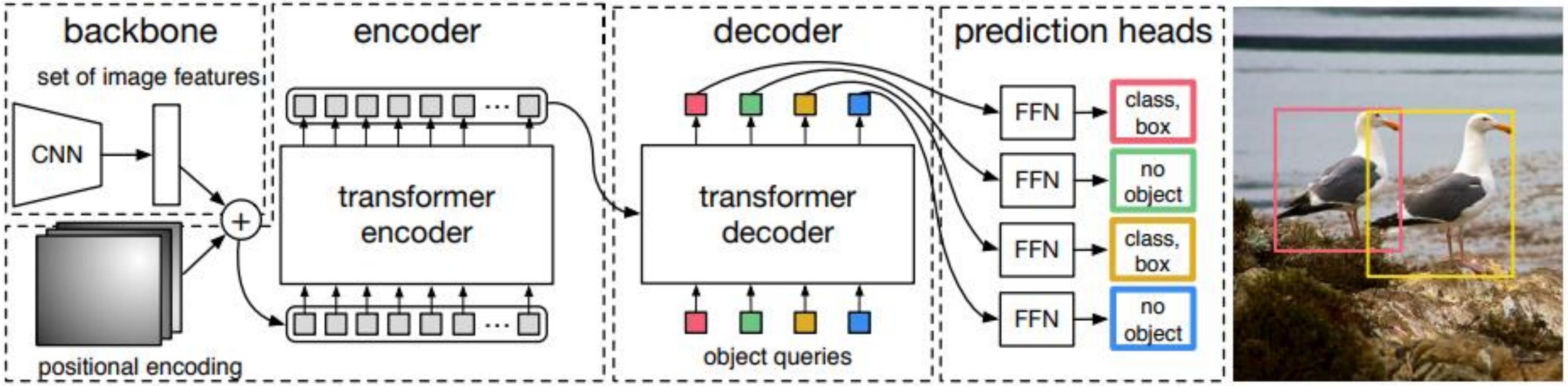
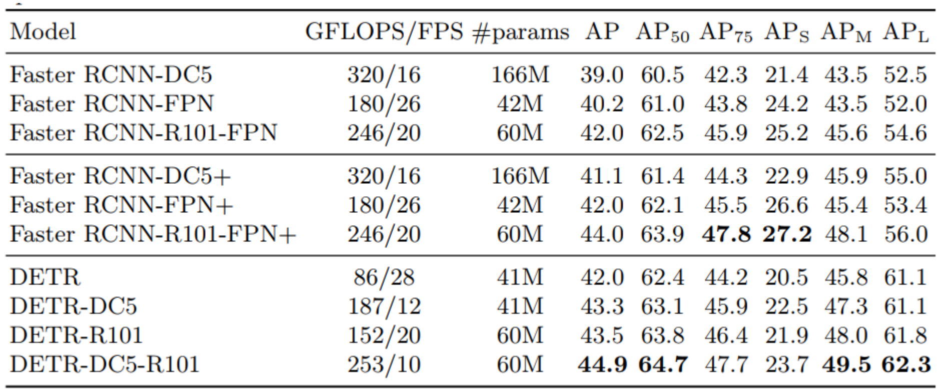


Fig. 10. Comparisons of recent visual Transformers on ImageNet-1k benchmark, including ViT [27], DeiT [38], BoTNet [44], VTs [43], ConViT [45], CeiT [46], LocalViT [47], TNT [52], Swin [33], PiT [57], T2T-ViT [56], PVT [39], CvT [34], DeepViT [59], CaiT [40], Cross ViT [108] (best viewed in color). (a) The bubble plot of the mentioned models with 224^2 resolution input, the size of cycle denotes GFLOPs. (b) Comparison of visual Transformers with high-resolution inputs, the square indicates 448^2 input resolution. (c) The accuracy plot of some pre-trained models on ImageNet-21k.

Transformer For Detection

DETR：





Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

Carion, Nicolas, et al." End-to-End Object Detection with Transformers. " arXiv preprint arXiv:2005.12872(2020).

DETR-based improvements

- Spares Transformer
  - In DETR, the dense interaction between decoder embedding and global features consumes a lot of computing resources and slows down the convergence speed of DETR. Deformable DETR and ACT rely on sparse attention of data to solve the above problems.
  - Several important improvements of sparse attention: Deformable DETR, ACT, SMCA, Conditional DETR, Two-Stage Deformable DETR and Efficient DETR.
- Structural Redesign
  - Some works have redesigned the structure of only the encoder to directly avoid the problem of the decoder. For example, TSP inherits the idea of ensemble prediction, and abandons the decoder and target query. YOLOS combines the encoder-decoder neck of DETR and the encoder-only backbone of ViT to redesign the encoder-only detector.
- Self-supervised Learning

DETR-based improvements

- Some kinds of based Transformers Backbones can be easily incorporated into various frameworks (for example, MaskR-CNN, RetinaNet, DETR, etc.) to perform dense prediction tasks.
- The hierarchical structure constructs the Transformer as a process from high resolution to low resolution to learn multi-scale features, such as PVT.
- The local enhancement structure constructs the backbone as a combination of local to global to effectively extract short-range and long-range visual dependencies and avoid secondary calculation overhead, such as Swin-Transformer, ViL, and Focal Transformer.
- FPT uses three attention components to model cross-space and scale interactions, including self-attention, top-down cross-attention, and bottom-up cross-channel attention. In the end, further improvements were made on many SOTA models.

Transformer For Segmentation

- Patch-Based Transformer

In order to expand the receptive field, CNN requires a large stack of decoders to map high-level features to the original spatial resolution. In contrast, Transformer relying on global modeling capabilities, the patch-based Transformer treats the input images as patch sequences and sends them to a columnar Transformer encoder. This resolution-invariant strategy enables Transformer to include only a relatively simple decoder and achieve ideal performance for segmentation tasks.

In addition, some works (SETR, TransUNet, Segformer) try to study the best combination between patch-based Transformer and different segmentation frameworks (Mask R-CNN, U-net).

Transformer For Segmentation

- Query-Based Transformer

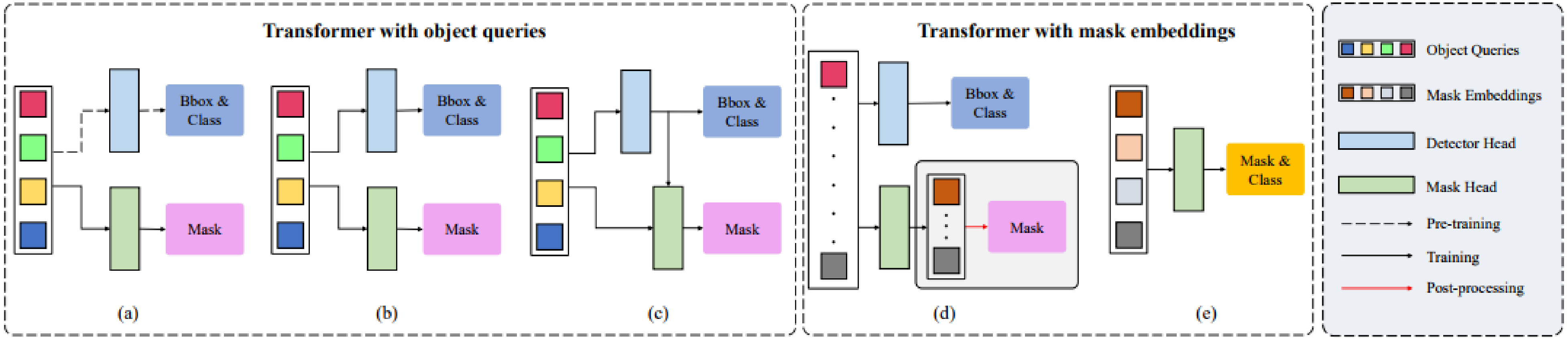


Fig. 13. Query-based frameworks for segmentation tasks. (a) The model is firstly trained on detection task and then fine-tuned on segmentation task following a serial training process. (b) The model is supervised on two dependent task simultaneously. (c) The cascade hybrid task model generates a fine-grained mask based on the coarse region predicted by detector head. (d) The query embeddings are dependently supervised by mask embeddings and boxes. (e) The box-free model directly predicts masks without box branch and views segmentation task as a mask prediction problem.

Summary

- For classification, the deep-level Transformer backbone can effectively reduce the computational complexity and avoid excessive smoothing of features in the deep layer. At the same time, the early convolution is sufficient to capture low-level features, which can significantly enhance robustness and reduce the computational complexity of shallow layers. In addition, both convolutional projection and local attention mechanism can improve the locality of Transformer. The former may also be a new method to replace position coding.
- For detection, the Transformer neck benefits from the encoder-decoder structure, which is less computationally expensive than the encoder-only Transformer detector. Therefore, the decoder is necessary, but due to its slow convergence speed, only a few stacks are needed. In addition, sparse attention is conducive to reducing the computational complexity and accelerating the convergence of Transformer, while the spatial prior is conducive to the performance of Transformer, and the convergence speed is slightly faster.
- For segmentation, the encoder-decoder Transformer model can unify the three segmentation subtasks into a mask prediction problem through a series of learnable mask embeddings. This frameless method has achieved the latest SOTA (MaskFormer) in multiple benchmarks. In addition, the box-based Transformer's cascaded model for specific hybrid tasks is proven to achieve higher performance in instance segmentation tasks.

Application to Video

Classification For Video（ViViT）

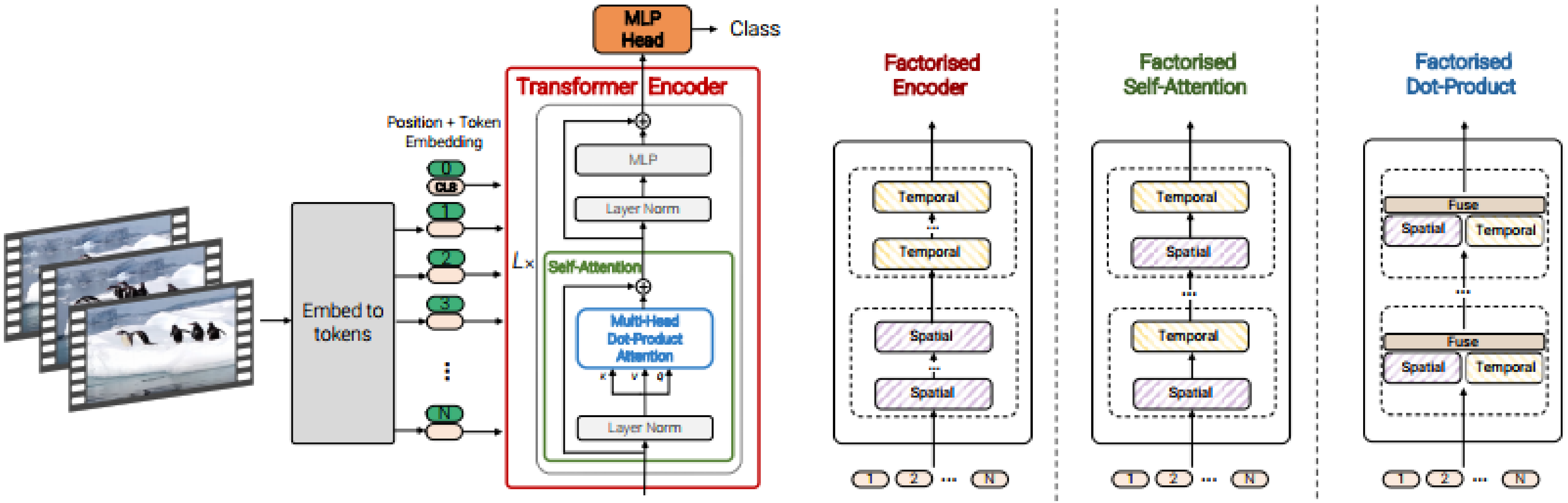


Figure 1: We propose a pure-transformer architecture for video classification, inspired by the recent success of such models for images [15]. To effectively process a large number of spatio-temporal tokens, we develop several model variants which factorise different components of the transformer encoder over the spatial- and temporal-dimensions. As shown on the right, these factorisations correspond to different attention patterns over space and time.

Model design of ViViT

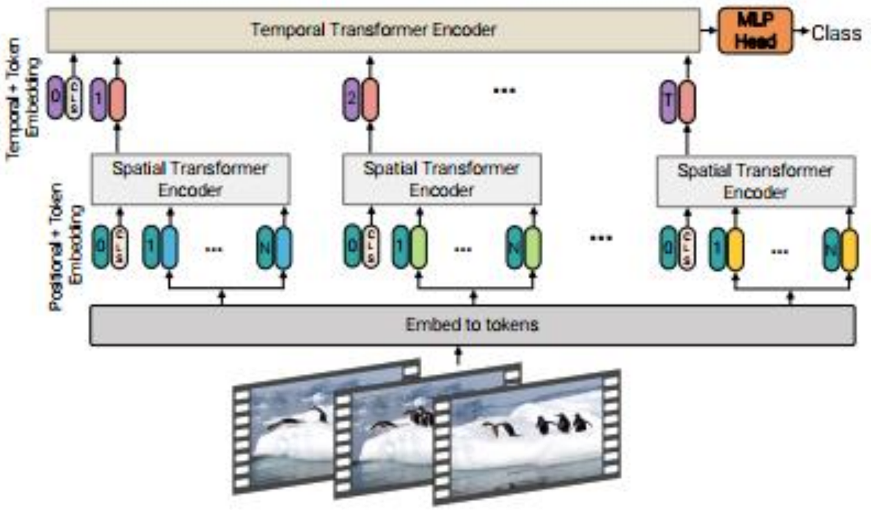

Temporal Transformer Encoder
MLP Head
Class
Temporal + Token Embedding
0
1
2
...
T
Spatial Transformer Encoder
Spatial Transformer Encoder
Spatial Transformer Encoder
Positional + Token Embedding
0
1
...
N
...
Embed to tokens

Figure 4: Factorised encoder (Model 2). This model consists of two transformer encoders in series: the first models interactions between tokens extracted from the same temporal index to produce a latent representation per time-index. The second transformer models interactions between time steps. It thus corresponds to a "late fusion" of spatial- and temporal information.

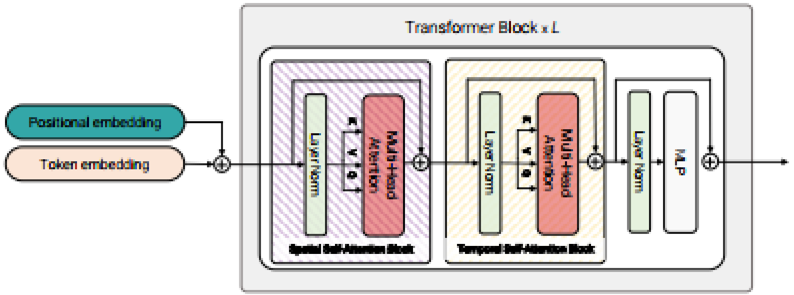


Figure 5: Factorised self-attention (Model 3). Within each transformer block, the multi-headed self-attention operation is factorised into two operations (indicated by striped boxes) that first only compute self-attention spatially, and then temporally.

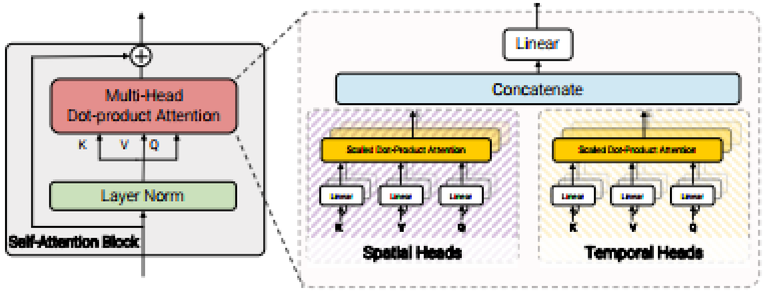
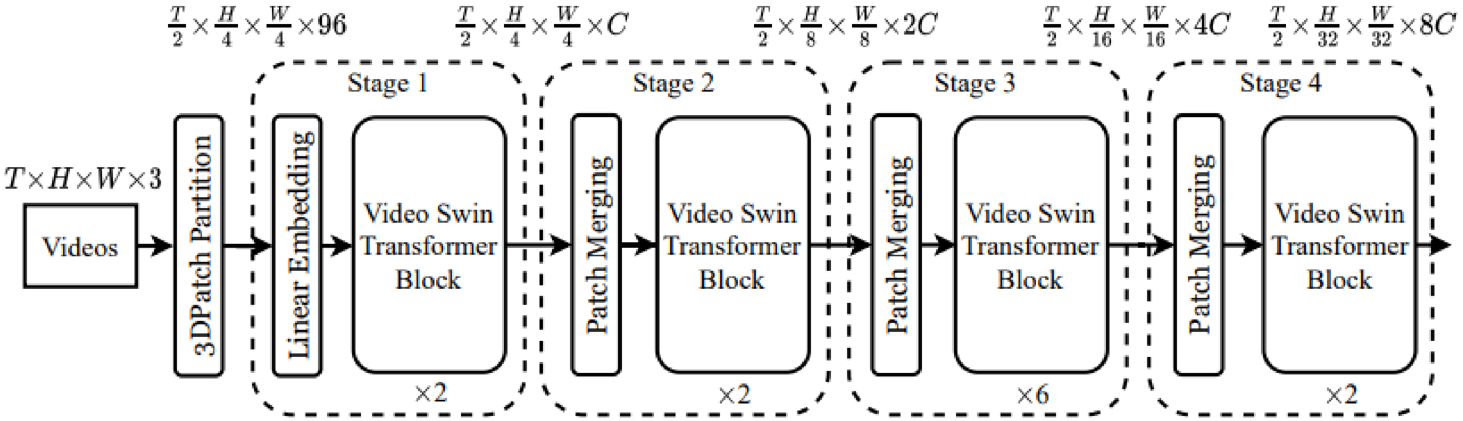


Figure 6: Factorised dot-product attention (Model 4). For half of the heads, we compute dot-product attention over only the spatial axes, and for the other half, over only the temporal axis.

Classification For Video（Video Swin Transformer）



$\frac{T}{2}\times\frac{H}{4}\times\frac{W}{4}\times 96$
$\frac{T}{2}\times\frac{H}{4}\times\frac{W}{4}\times C$
$\frac{T}{2}\times\frac{H}{8}\times\frac{W}{8}\times 2C$
$\frac{T}{2}\times\frac{H}{16}\times\frac{W}{16}\times 4C$
$\frac{T}{2}\times\frac{H}{32}\times\frac{W}{32}\times 8C$
Stage 1
Stage 2
Stage 3
Stage 4
$T\times H\times W\times 3$
Videos
3DPatch Partition
Linear Embedding
Video Swin Transformer Block
Patch Merging
×2
×2
×6
×2

Figure 1: Overall architecture of Video Swin Transformer (tiny version, referred to as Swin-T).

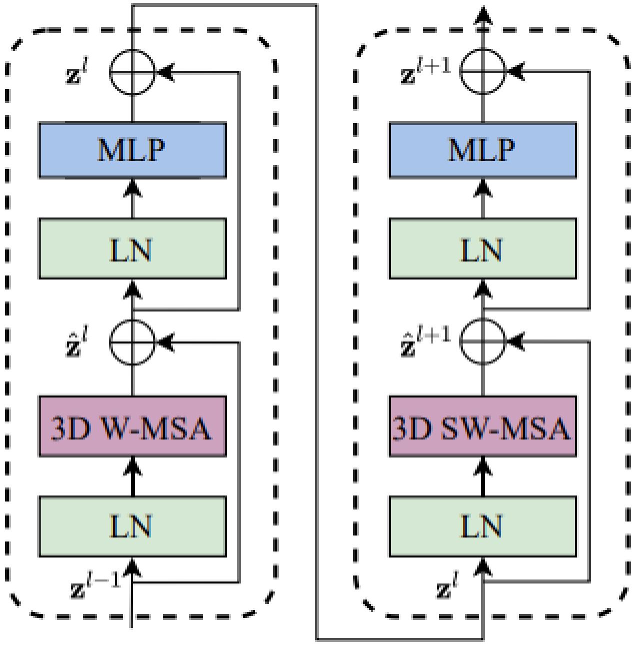


Figure 2: An illustration of two successive Video Swin Transformer blocks.

Detection For Video（VisTR）

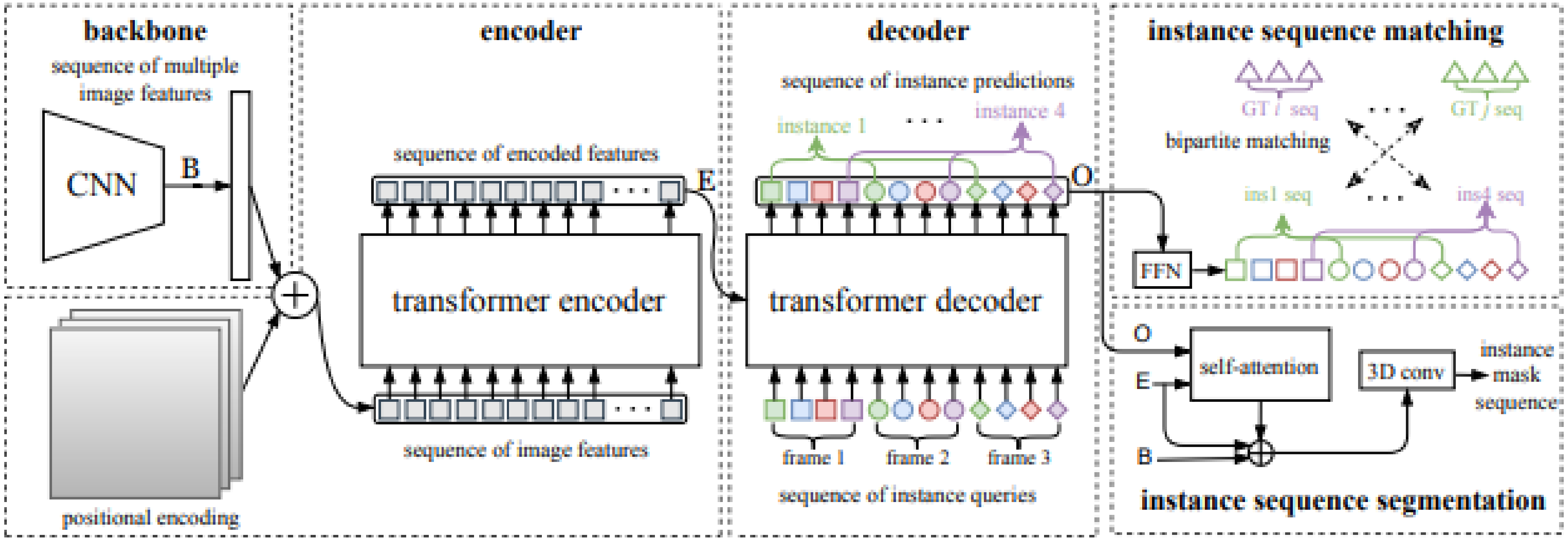


Figure 2 – The overall architecture of VisTR. It contains four main components: 1) a CNN backbone that extracts feature representation of multiple images; 2) an encoder-decoder Transformer that models the relations of pixel-level features and decodes the instance-level features; 3) an instance sequence matching module that supervises the model; and 4) an instance sequence segmentation module that outputs the final mask sequences (Best viewed on screen).

Future outlook

Future overlook

The Visual Transformer method has made great progress and has shown promising results that approach or exceed the SOTA CNN method on multiple benchmarks . But the technology is still immature, and the CV field is still dominated by CNN.

- Set Prediction

The one-to-one label assignment in the ensemble prediction strategy leads to unstable training in the early process, which may reduce the accuracy of the final result. Using other label assignments and losses to improve ensemble prediction may help new detection frameworks.

- Self-supervised learning

The self-supervised Transformer pre-training standardizes the NLP field and has achieved great success in various applications . There is no specific supervised learning method for CV, and the encoder-decoder Transformer for self-supervised learning is worthy of our further study.

Future overlook

In order to reveal and utilize Transformer's capabilities, many solutions have been proposed in recent years. These methods have shown excellent performance on various visual tasks including basic image classification, high-level vision, low-level vision and video processing. However, the potential of Transformer for computer vision has not been fully explored, and there are still some challenges to be solved. Although researchers have proposed many Transformer-based models to solve computer vision tasks, these works are groundbreaking solutions and there is still much room for improvement.

In addition, most of the existing Visual transformer models are designed to handle a single task. Many NLP models (such as GPT-3) have shown that Transformer can handle multiple tasks in one model. IPT in the CV field can also handle a variety of low-level vision tasks, such as super-resolution, image noise reduction, and water drainage. We believe that a model can involve more tasks.

Reference

Reference

[1].Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). ViViT: A Video Vision Transformer. arXiv:2103.15691.

[2].Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. arXiv:2005.12872.

[3].Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., . . . Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929.

[4].Liu, Y., Zhang, Y., Wang, Y., Hou, F., Yuan, J., Tian, J., . . . He, Z. (2021). A Survey of Visual Transformers. arXiv:2111.06091.

[5].Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., & Hu, H. (2021). Video Swin Transformer. arXiv:2106.13230.

[6].Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., & Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nat Methods, 15*(12), 1053-1058.

[7].Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., & Xia, H. (2020). End-to-End Video Instance Segmentation with Transformers. arXiv:2011.14503.

[8].Lin, T., Wang, Y., Liu, X., & Qiu, X. (2021). A Survey of Transformers. arXiv:2106.04554. Retrieved from

[9].https://ui.adsabs.harvard.edu/abs/2021arXiv210604554L

[10].https://zhuanlan.zhihu.com/p/433048484

Reference

[11].http://nlp.seas.harvard.edu/2018/04/03/attention.html#attention
[12].http://jalammar.github.io/illustrated-transformer/
[13].https://www.cnblogs.com/wxkang/p/14195057.html
[14].邱锡鹏,2020.神经网络与深度学习[M].机械工业出版社
[15].李沐,阿斯顿·张.et al.2019动手学深度学习[M].人民邮电出版社