TRANSFORMER

# Vision Transformer & Meta-architecture

2023.8.15 高鑫

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

# Transformer-Based Visual Segmentation: A Survey

Xiangtai Li, Henghui Ding, Wenwei Zhang, Haobo Yuan, Jiangmiao Pang,
Guangliang Cheng, Kai Chen, Ziwei Liu, Chen Change Loy

1 **Transformer**

Self-attention, cross-attention, positional embedding, inductive bias

2 **Meta-architecture**

A unified model for classification, segmentation and detection

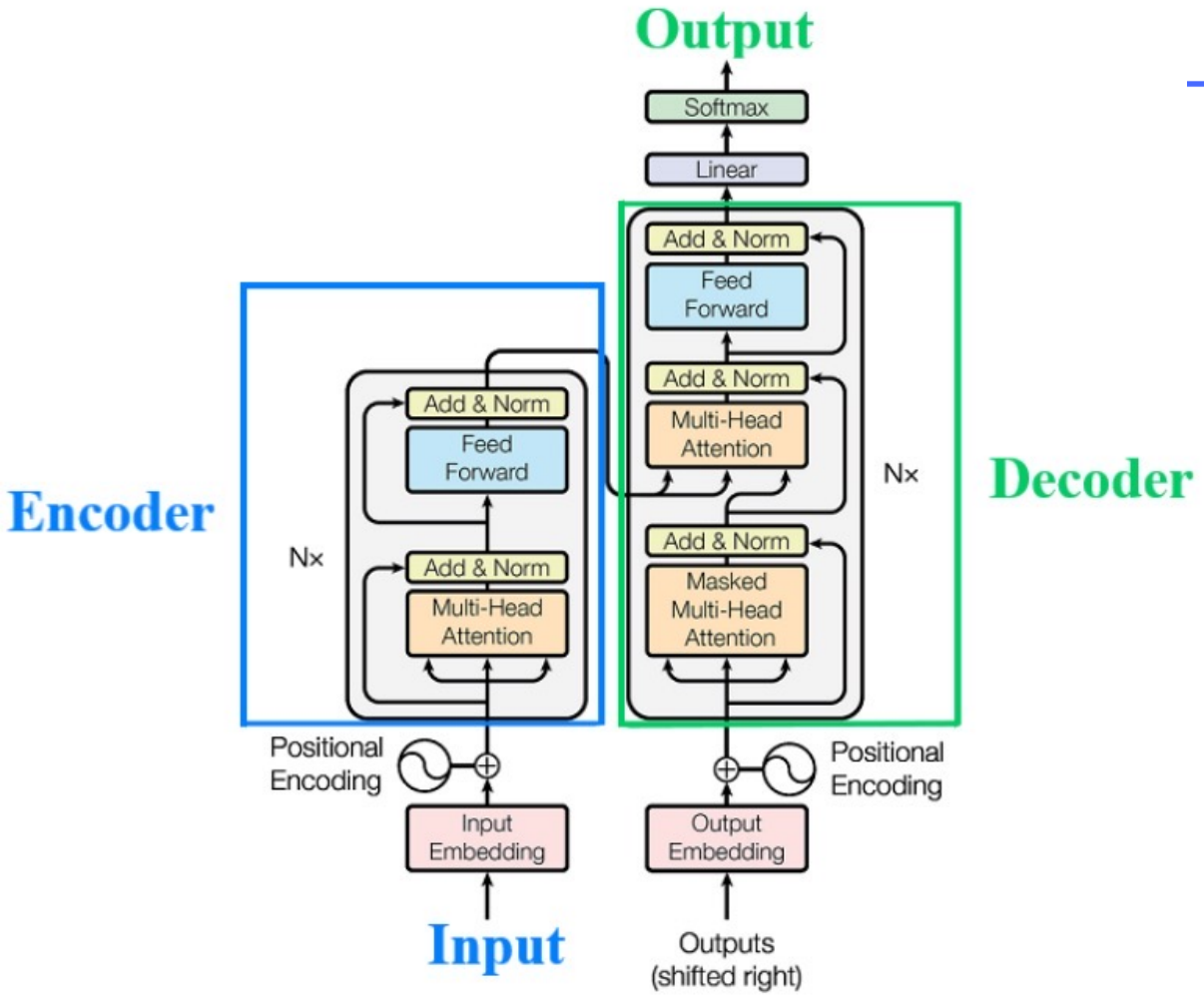
3 **Methods in three aspects**

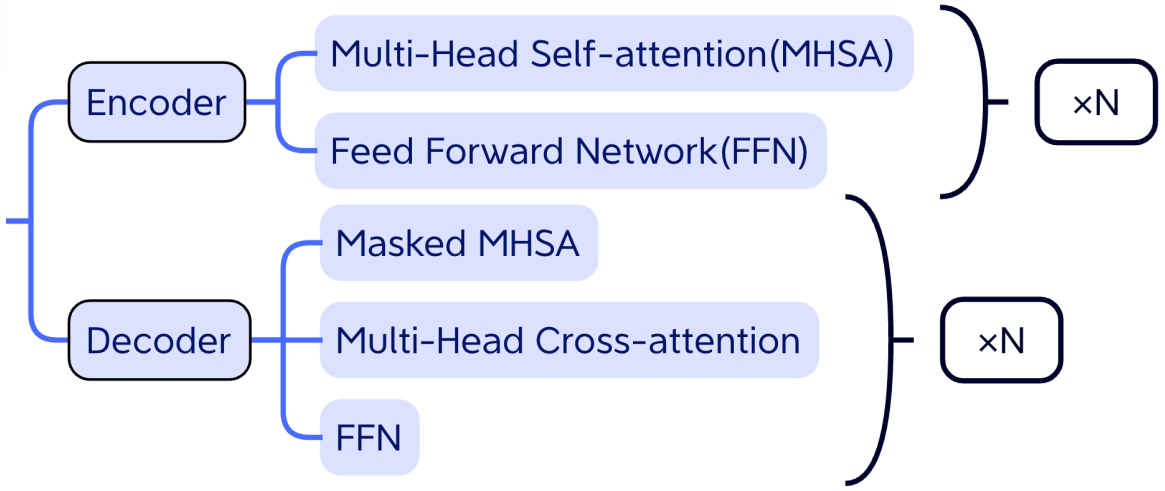
Backbone, decoder, object query

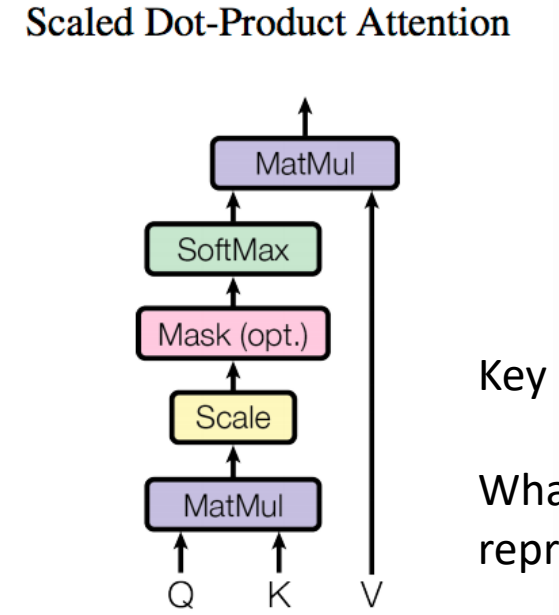
4 **Future directions**

Six future directions and possible applications

# Think Again about Transformer




$$SA(X) = \text{softmax}(\frac{Q \cdot K^T}{\sqrt{d}})V$$

Key point: Attention Mechanism

What do Q, K, and V actually represent?



Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

# Think Again about Transformer

Input

text

$[l_0, 1]$

↓ tokenize

$[l, 1]$

↓ word embedding

$[l, d_1]$

image

$[C, H, W]$

↓ patch

$[\frac{W \times H}{P^2}, P^2 \times C]$

↓ linear projection

$[\frac{W \times H}{P^2}, d_1]$

记 $l = \frac{W \times H}{P^2}$

$l$为序列长度

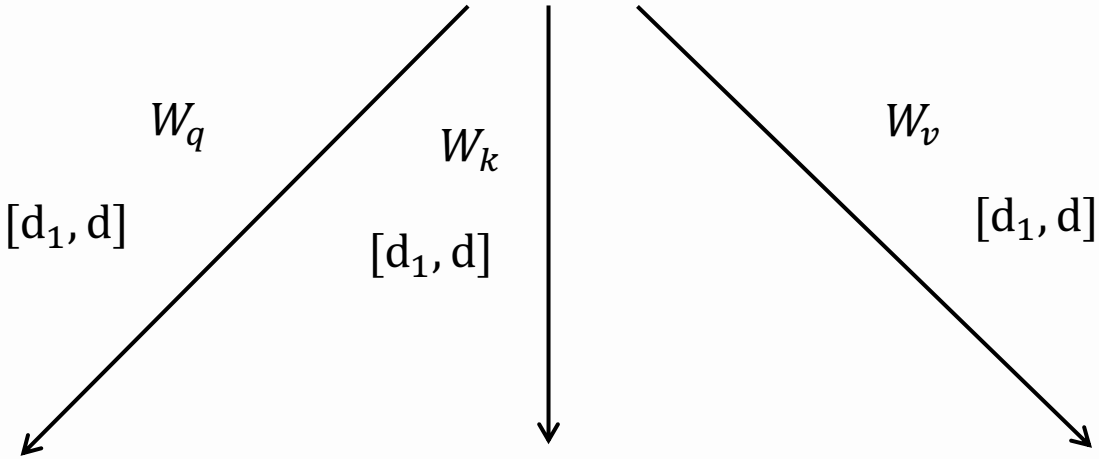
通常会选取

$d_1$= #head × d

表示 representive space 维数

e.g., 768=8×96

X




Self-attention:

$l_q = l_k = l_v = l$

$Q, K, V$ are all obtained through linear transformations of X

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

先理解 $\text{softmax}(\frac{X \cdot X^T}{\sqrt{d}})X$

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{l1} & \cdots & x_{ld} \end{bmatrix} = [x_1, \ldots, x_l]^T \in \mathbb{R}^{l \times d}, \text{其中} x_i \in \mathbb{R}^d, i = 1, \ldots, l$$

$$XX^T = \begin{bmatrix} x_1 \\ \vdots \\ x_l \end{bmatrix} [x_1 \quad \cdots \quad x_l] = \begin{bmatrix} x_1 x_1^T & \cdots & x_1 x_l^T \\ \vdots & \ddots & \vdots \\ x_l x_1^T & \cdots & x_l x_l^T \end{bmatrix} \in \mathbb{R}^{l_q \times l_k}$$

$x_1$与 $x_1, x_2, \ldots, x_l$的相似度

某一token与其它token的相似度

两个向量 $a, b$ 的内积 $ab^T = |a||b|cos < a, b > = \begin{cases} > 0, \\ = 0, \\ < 0, \end{cases}$

锐角 同向时最大
正交
钝角 反向时最小

向量 $a$ 在 $b$ 上的投影，
表示两向量之间的相似度

$[x_1 x_1^T \quad \cdots \quad x_1 x_l^T]$, 记 $z_i = \frac{x_1 x_i^T}{\sqrt{d}}$

$[z_1 \quad \cdots \quad z_l]$ ↓ softmax

$[\frac{e^{z_1}}{\sum e^{z_i}} \quad \cdots \quad \frac{e^{z_l}}{\sum e^{z_i}}]$

Attention matrix

| | 机 | 器 | 学 | 习 |
|---|---|---|---|---|
| 机 | 0.4 | 0.4 | 0.1 | 0.1 |
| 器 | 0.4 | 0.5 | 0.1 | 0.0 |
| 学 | 0.2 | 0.0 | 0.4 | 0.4 |
| 习 | 0.1 | 0.1 | 0.3 | 0.5 |

$A = \text{softmax}\left(\frac{X \cdot X^T}{\sqrt{d}}\right) \Rightarrow$ 归一化后，变成weight

$A = \text{softmax}\left(\frac{X \cdot X^T}{\sqrt{d}}\right)$ 为什么要除以$\sqrt{d}$?

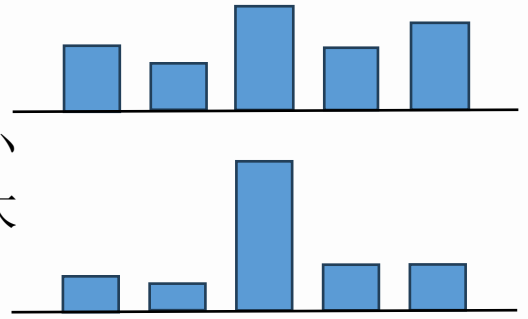
$$\mathbb{R}^{l_k} \rightarrow \mathbb{R}^{l_k}$$

$$\text{Softmax: } [z_1 \quad \dots \quad z_{l_k}] \rightarrow [s_1 \quad \dots \quad s_{l_k}], \qquad s_i = \frac{e^{z_i}}{\sum e^{z_i}}, i = 1, \dots, l_k$$

对 Softmax 求导数

$$Jacobian\ matrix: \quad J = \begin{bmatrix} s_1(1-s_1) & -s_1 s_2 & \cdots & -s_1 s_{l_k} \\ -s_2 s_1 & s_2(1-s_2) & \cdots & -s_2 s_{l_k} \\ \vdots & \vdots & \ddots & \vdots \\ -s_{l_k} s_1 & -s_{l_k} s_2 & \cdots & s_{l_k}(1-s_{l_k}) \end{bmatrix}, \qquad J = 0 \Leftrightarrow s_i = 1, s_{j \neq i} = 0, i\ in\ [1, \dots, l_k]$$

$$z_i \times scaling\ factor, i = 1, \dots, l_k \Rightarrow \left[\frac{e^{z_1}}{\sum e^{z_i}} \quad \dots \quad \frac{e^{z_l}}{\sum e^{z_i}}\right] \begin{cases} scaling\ factor < 1, \text{越小} \\ scaling\ factor > 1, \text{越大} \end{cases}$$



越smooth

越接近使J=0的条件

$X \cdot X^T$中的元素较大时，会使得softmax出现梯度消失，所以乘上一个scaling factor，乘什么呢？

假设两个独立的随机向量$X, Y$， 向量中每个元素均值为0，方差为1

$D(XY^T) = D(x_1y_1 + x_2y_2 + \dots + x_dy_d) = D(x_1y_1) + D(x_2y_2) + \dots + D(x_dy_d) = d$

除以$\sqrt{d}$ 对方差起到归一化作用

[4]To illustrate why the dot products get large, assume that the components of $q$ and $k$ are independent random variables with mean 0 and variance 1. Then their dot product, $q \cdot k = \sum_{i=1}^{d_k} q_i k_i$, has mean 0 and variance $d_k$.

$$A = \text{softmax}\left(\frac{X \cdot X^T}{\sqrt{d}}\right) \in \mathbb{R}^{l_q \times l_k}, \quad AX \in \mathbb{R}^{l_q \times d}$$

$$AX = \begin{bmatrix} a_{11} & a_{11} & \cdots & a_{1l_k} \\ a_{21} & a_{22} & \cdots & a_{2l_k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{l_q1} & a_{l_q2} & \cdots & a_{l_ql_k} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{l_v} \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + \cdots + a_{1l_k}x_{l_v} \\ a_{21}x_1 + \cdots + a_{2l_k}x_{l_v} \\ \vdots \\ a_{l_q1}x_1 + \cdots + a_{l_ql_k}x_{l_v} \end{bmatrix}, \quad \text{(必须满足 } l_k = l_v,\ l_q \text{可以不等)}$$

$$(Self\ attention\ \ l_k = l_v = l_q)$$

$\text{softmax}\left(\frac{X \cdot X^T}{\sqrt{d}}\right)X$ 得到的结果

每一个元素都变成原来序列中所有token的加权和

$$\begin{bmatrix} 机 \\ 器 \\ 学 \\ 习 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.4机 + 0.4器 + 0.1学 + 0.1习 \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}$$

即每个元素变成了与原token高度语义相关的token组合

那如果希望得到原token之后可能的下一个token呢？

或者其它映射规则？

增加可学习参数，
拟合映射规则

$$\text{softmax}\left(\frac{X \cdot X^T}{\sqrt{d}}\right)X \Rightarrow \text{softmax}\left(\frac{XW_q \cdot (XW_k)^T}{\sqrt{d}}\right)XW_v$$

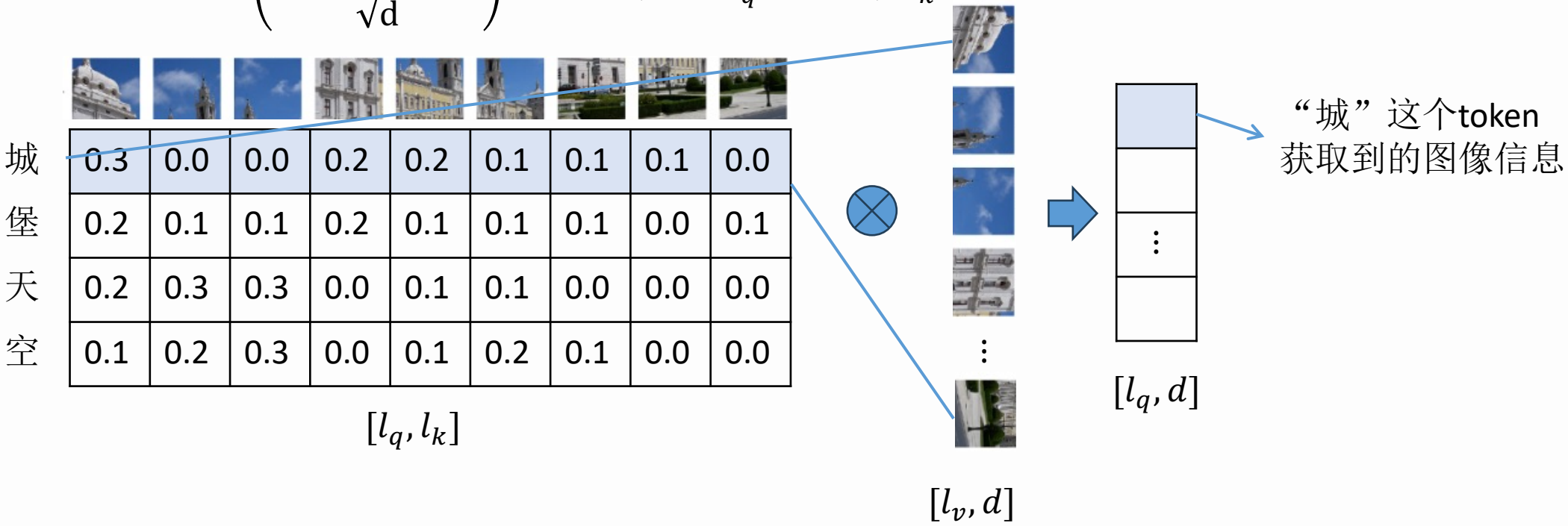
Self-attention in Transformer

# Cross Attention

对于$\mathrm{X} \in \mathbb{R}^{l_q \times d}, \mathrm{Y} \in \mathbb{R}^{l_k \times d}, l_k = l_v,$

$X, \mathrm{Y}$ 一般是不同模态的两个序列，为了保证 $l_k = l_v$，$K$ 和 $V$ 一般都由 Y 得到

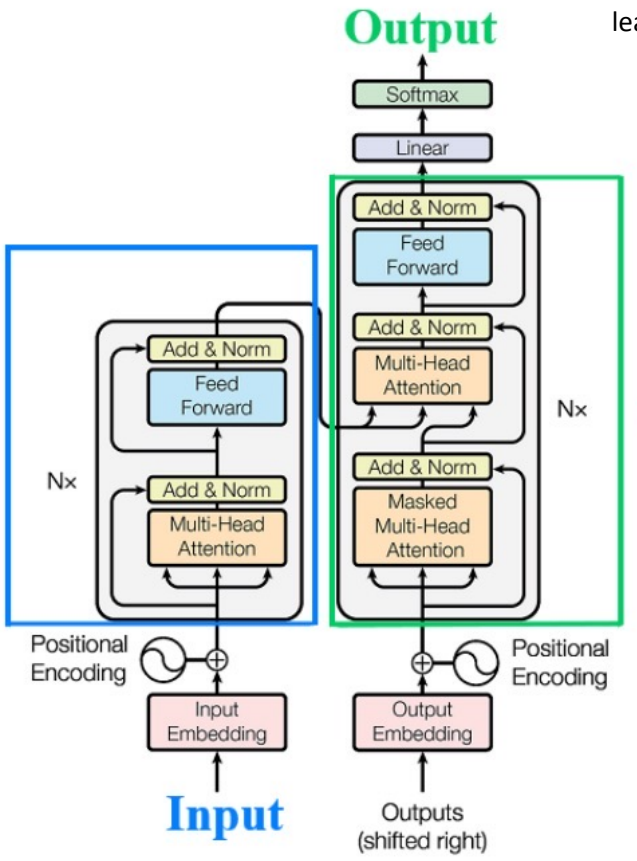
$$\mathrm{A} = \mathrm{softmax}\left(\frac{\mathrm{X}W_q \cdot (YW_k)^T}{\sqrt{\mathrm{d}}}\right) \in \mathbb{R}^{l_q \times l_k}, \qquad W_q \in \mathbb{R}^{d \times d}, W_k \in \mathbb{R}^{d \times d}$$

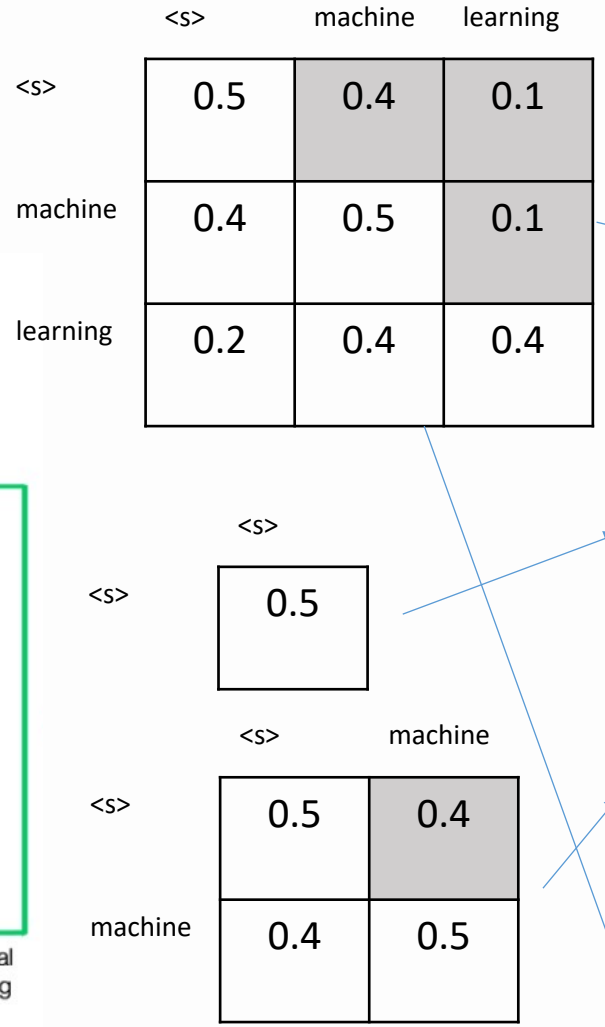



$$\mathrm{Output} = A(YW_v) \in \mathbb{R}^{l_q \times d}, \quad W_v \in \mathbb{R}^{d \times d}$$

# Masked Attention

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




Causal Mask

样本：翻译成英文：机器学习 machine learning

- Training

Input 翻译成英文：机器学习
Output(shifted right) <s> machine learning
Output machine learning <ed>
Ground Truth machine learning <ed>

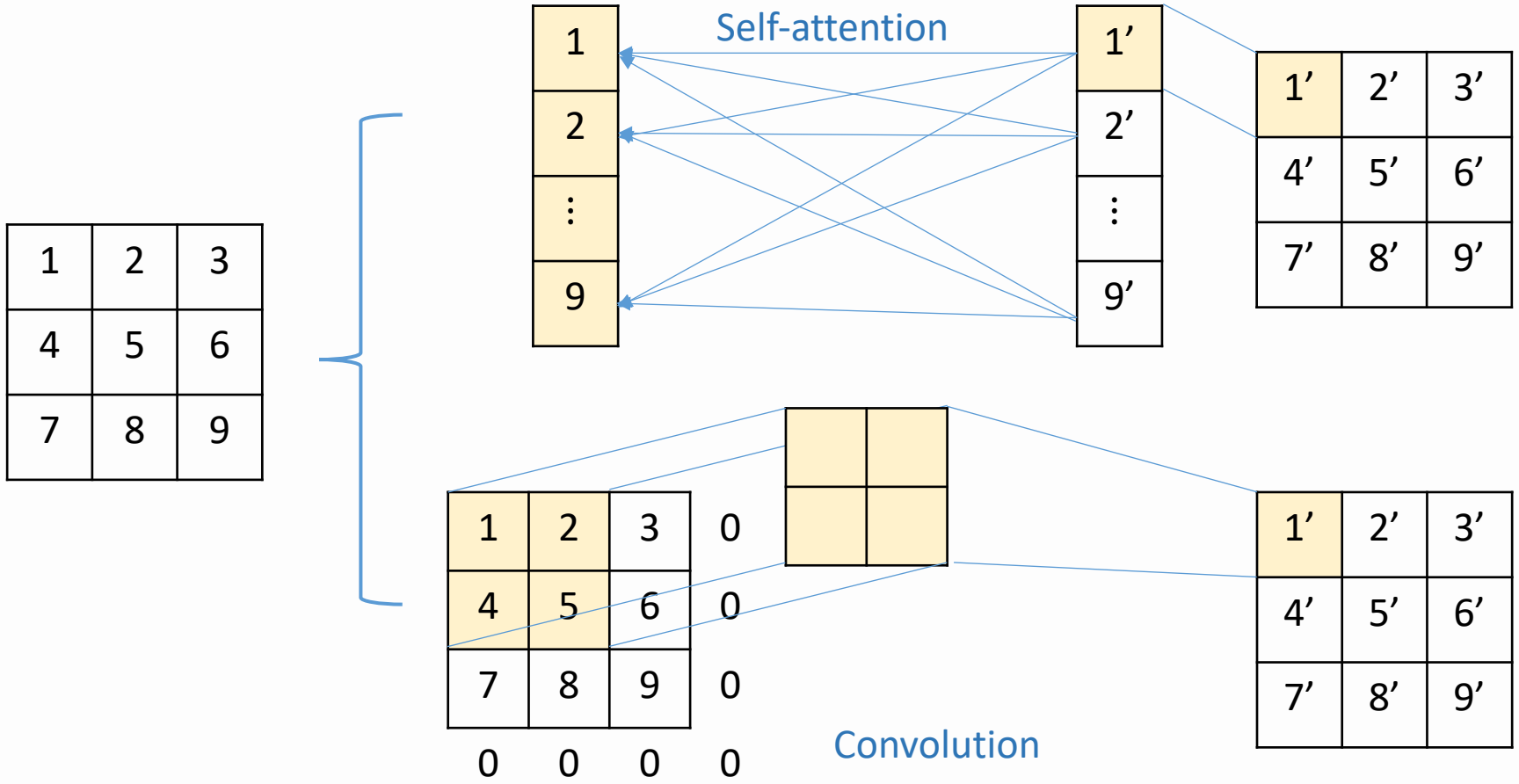
- Testing

1. Input 翻译成英文：机器学习
   Output(shifted right) <s>
   Output machine
2. Input 翻译成英文：机器学习
   Output(shifted right) <s> machine
   Output machine learning
3. Input 翻译成英文：机器学习
   Output(shifted right) <s> machine learning
   Output machine learning <ed>

输出遇到<ed>，终止token-by-token的预测过程

# Image-specific Inductive bias

"We note that Vision Transformer has much less image-specific inductive bias than CNNs. In CNNs, locality, two-dimensional neighborhood structure, and translation equivariance are baked into each layer throughout the whole model. In ViT, only MLP layers are local and translationally equivariant, while the self-attention layers are global. "




- Global

长序列全局感知 ✅

训练时间长 需要多数据 ❌

- Locality
- Translation equivariance

# Positional embedding in NLP Transformer and ViT

Transformer 位置不敏感(Position-insensitive)

1, 2, 3, 4, ……?　实际问题中可能比训练集的句子还要长，会使得模型泛化性能较差

1. 为每个时间步（token在序列中的位置）输出唯一的编码；
2. 即便序列长度不一，序列中两个时间步之间的距离应该是“恒定”的；
3. 模型可以轻易泛化到更长的序列上。

$P = [p_1, \ldots, p_l]^T \in \mathbb{R}^{l \times d}$,　　其中$p_i = [p_{i1}, \ldots, p_{id}]^T \in \mathbb{R}^d, i = 1, \ldots, l$，是一个token的位置编码

$$p_{ij} = \begin{cases} sin(\omega_k \cdot i), & j = 2k \\ cos(\omega_k \cdot i), & j = 2k+1 \end{cases}, \quad j = 0, \ldots, d-1, \quad \omega_k = \frac{1}{T^{\frac{2k}{d}}}, \quad k = 0,1, \ldots, \frac{d}{2} - 1, T = 10000$$

$$\mathrm{p}_i = \left[\sin(i), \cos(i), \sin\left(\frac{i}{10000^{\frac{2}{d}}}\right), \cos\left(\frac{i}{10000^{\frac{2}{d}}}\right), \ldots, \sin\left(\frac{i}{10000^{\frac{d-2}{d}}}\right), \cos\left(\frac{i}{10000^{\frac{d-2}{d}}}\right)\right]^T$$

# Think Again about Transformer

$$\omega_k = \frac{1}{T^{\frac{2k}{d}}}$$

令$d=4, T=4,$　　序列的位置编码:

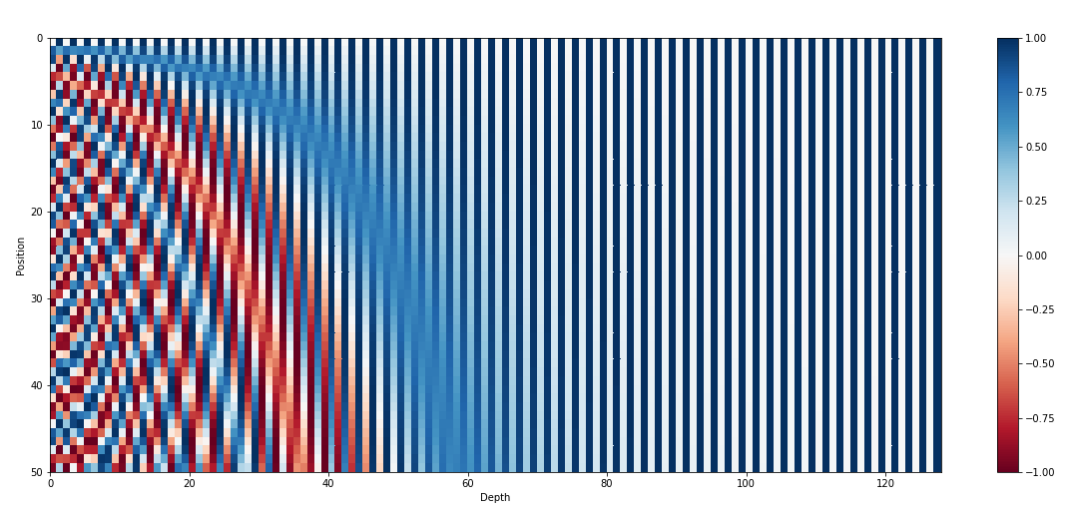
| | | | | |
|---|---|---|---|---|
| Token 0 | $\sin(\frac{0}{4^{0/4}})$ | $\cos(\frac{0}{4^{0/4}})$ | $\sin(\frac{0}{4^{2/4}})$ | $cos(\frac{0}{4^{2/4}})$ |
| Token 1 | $\sin(\frac{1}{4^{0/4}})$ | $\cos(\frac{1}{4^{0/4}})$ | $\sin(\frac{1}{4^{2/4}})$ | $\cos(\frac{1}{4^{2/4}})$ |
| Token 2 | $\sin(\frac{2}{4^{0/4}})$ | $\cos(\frac{2}{4^{0/4}})$ | $\sin(\frac{2}{4^{2/4}})$ | $\cos(\frac{2}{4^{2/4}})$ |
| Token 3 | $\sin(\frac{3}{4^{0/4}})$ | $\cos(\frac{3}{4^{0/4}})$ | $\sin(\frac{3}{4^{2/4}})$ | $\cos(\frac{3}{4^{2/4}})$ |
| … | | | | |
| Token $2\pi$ | $\sin(\frac{2\pi}{4^{0/4}})$ | $\cos(\frac{2\pi}{4^{0/4}})$ | $\sin(\frac{2\pi}{4^{2/4}})$ | $\cos(\frac{2\pi}{4^{2/4}})$ |
| … | | | | |
| Token $2\times2\pi$ | $\sin(\frac{4\pi}{4^{0/4}})$ | $\cos(\frac{4\pi}{4^{0/4}})$ | $\sin(\frac{4\pi}{4^{2/4}})$ | $\cos(\frac{4\pi}{4^{2/4}})$ |

最小的周期是$2\pi$，最大是$2\times2\pi$

整个位置编码的周期是$2\times2\pi$

旋转$\frac{\pi}{180}\times1°$　　旋转$\frac{\pi}{180}\times\frac{1}{2}°$

| | 0 | 1 | 0 | 1 |
|---|---|---|---|---|
| 0 | $\sin(1/1)$ | $\cos(1/1)$ | $\sin(1/2)$ | $\cos(1/2)$ |
| 1 | $\sin(2/1)$ | $\cos(2/1)$ | $\sin(2/2)$ | $\cos(2/2)$ |
| 2 | $\sin(3/1)$ | $\cos(3/1)$ | $\sin(3/2)$ | $cos(3/2)$ |




- Position周期 $T\times\pi$，避免训练和测试的序列长度有差别，提升泛化能力
- 位置信息基本储存在前面，随着depth加深，位置信息越来越少

# Think Again about Transformer

- We chose this function because we hypothesized it would **allow the model to easily learn to attend by relative positions**, since for any fixed offset s, $p_{i+s}$ can be represented as a **linear function** of $p_i$ .

$$\begin{bmatrix}\sin(\omega_k(i+s))\\ \cos(\omega_k(i+s))\end{bmatrix} = \begin{bmatrix}cos(\omega_k s) & sin(\omega_k s)\\ -sin(\omega_k s) & cos(\omega_k s)\end{bmatrix}\begin{bmatrix}\sin(\omega_k i)\\ \cos(\omega_k i)\end{bmatrix}$$

$p_{i+s}$ 和 $p_i$ 位置向量的内积仅与相对位置有关

$$\begin{aligned}p_{i+s}\cdot p_i &= \sum_{k=0}^{\frac{d}{2}-1}\sin(w_k\cdot i)\cdot\sin(w_k(i+s))+\cos(w_k\cdot i)\cdot\cos(w_k(i+s))\\ &= \sum_{k=0}^{\frac{d}{2}-1}\cos(w_k(i-(i+s))\\ &= \sum_{k=0}^{\frac{d}{2}-1}\cos(w_k s),\quad \text{where } w_k=\frac{1}{10000^{2k/d}}\end{aligned}$$

$p_{i+s}$ 和 $p_i$ 位置向量的欧式距离仅与相对位置有关

$$\begin{aligned}&||p_{i+s}-p_i||^2\\ &= \sum_{k=0}^{\frac{d}{2}-1}(\sin(w_k\cdot i)-\sin(w_k(i+s)))^2+(\cos(w_k\cdot i)-\cos(w_k(i+s)))^2\\ &= 4\sum_{k=0}^{\frac{d}{2}-1}[\cos(w_k(\frac{2i+s}{2})\cdot\sin(w_k(\frac{s}{2})]^2+[\sin(w_k(\frac{2i+s}{2})\cdot\sin(w_k(\frac{s}{2})]^2\\ &= 4\sum_{k=0}^{\frac{d}{2}-1}\sin^2(w_k\cdot\frac{s}{2}),\quad \text{where } w_k=\frac{1}{10000^{2k/d}}\end{aligned}$$

- Position获取到了信息

$QK^T=(XW_q)(YW_k)^T=XW_qW_k^TY^T$
表示X每个token与Y中所有token的联系

$((X+p_1)W_q)((Y+p_2)W_k)^T=XW_qW_k^TY^T+XW_qW_k^Tp_2{}^T+p_1W_qW_k^TY^T+p_1W_qW_k^Tp_2{}^T$

- 表示X每个token与Y中所有位置的联系
- 表示X每个位置与Y中所有token的联系
- 表示X每个位置与Y中所有位置的联系

# ViT: Learnable Positional Embedding

```
self.pos_embedding = nn.Parameter(torch.randn(1, num_patches+1, dim))
```

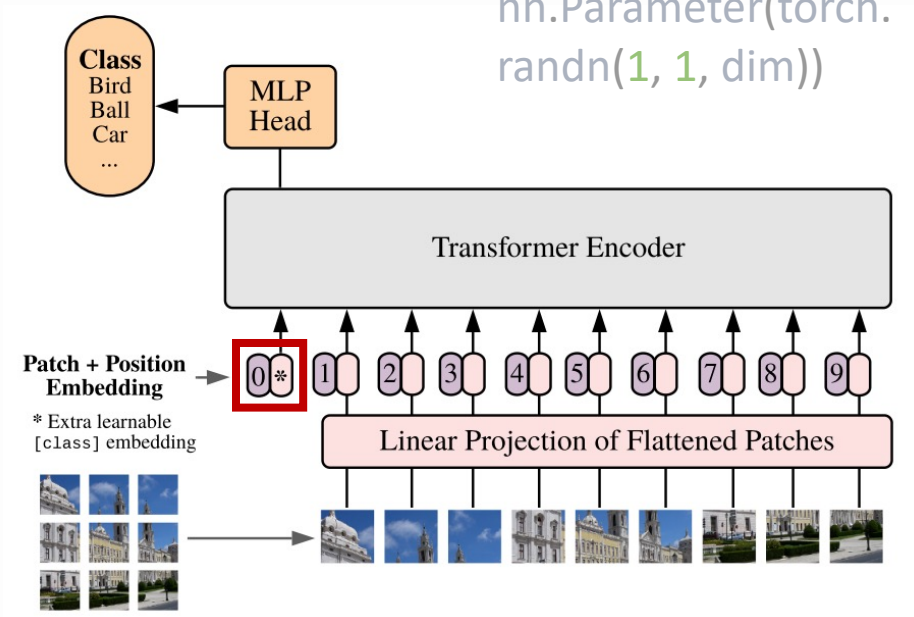
或者使用 nn.Embedding（将一个数字映射为指定维数的向量，其weight是可学习的参数）

```
self.pos_embed = nn.Embedding(num_patches+1, dim)
self.pos_embedding=self.pos_embed(range(num_patches+1)).weight
```

- 图像的序列长度基本固定
- 希望相对位置编码可以学习到图像的inductive bias

## Learnable cls token

```
self.cls_token = nn.Parameter(torch.randn(1, 1, dim))
```




## learnable query

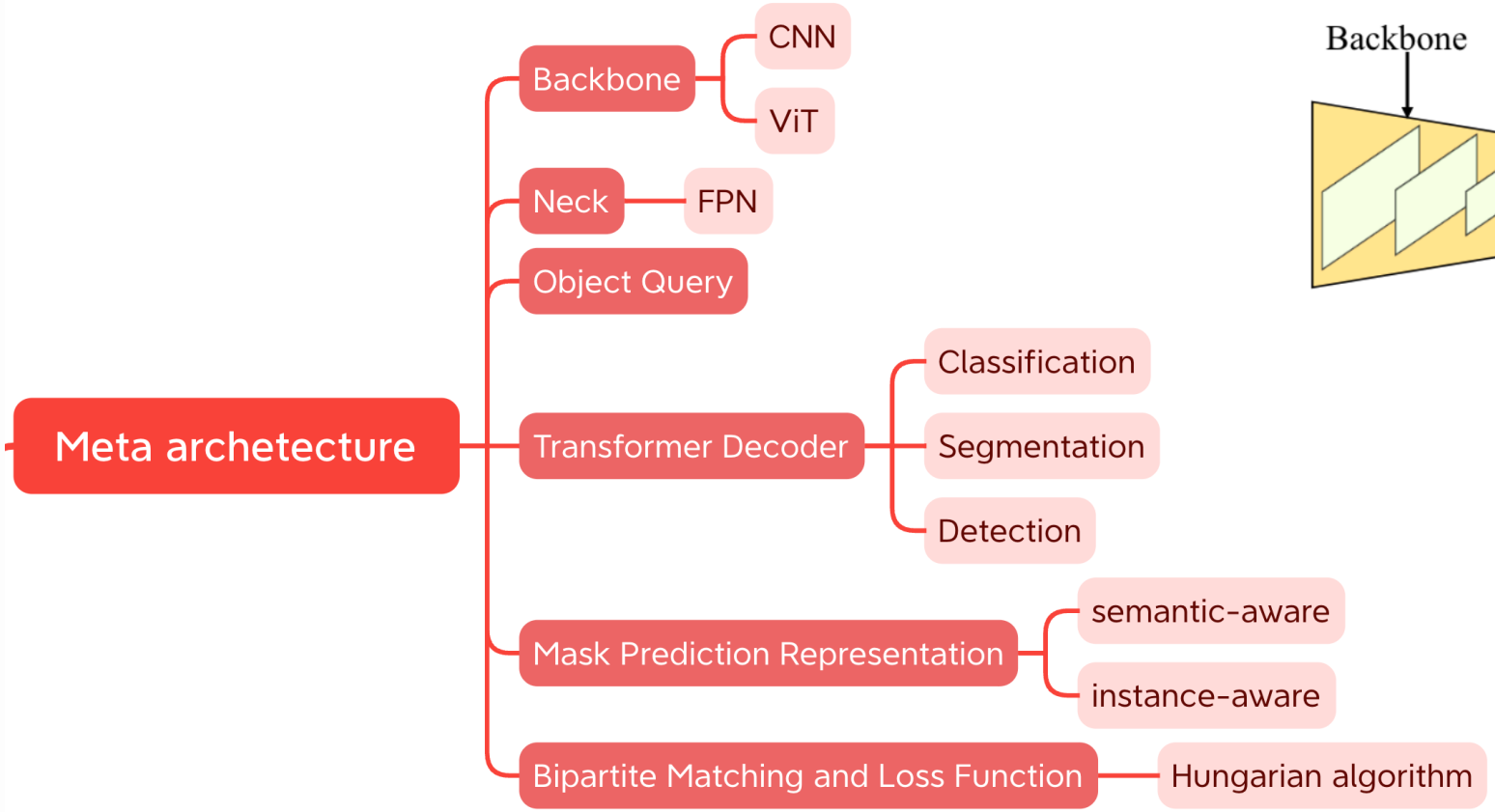
用固定长度的Query序列对Key和Value序列采样
是可学习的参数，随梯度下降更新

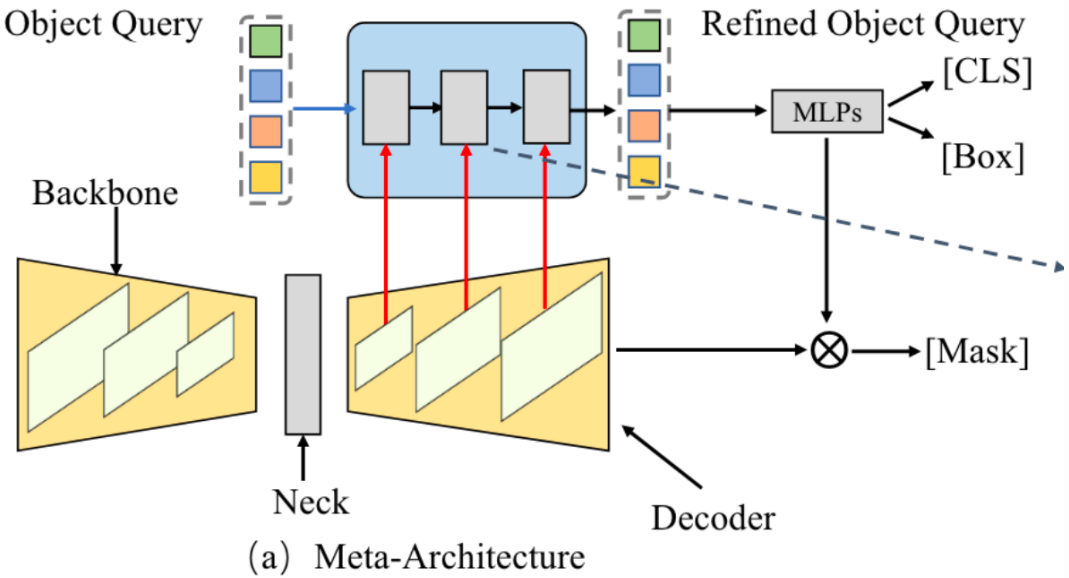
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| q1 | 0.3 | 0.0 | 0.0 | 0.2 | 0.2 | 0.1 | 0.1 | 0.1 | 0.0 |
| q2 | 0.2 | 0.3 | 0.3 | 0.0 | 0.1 | 0.1 | 0.0 | 0.0 | 0.0 |
| q3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.4 | 0.3 | 0.2 |
| q4 | 0.2 | 0.1 | 0.1 | 0.2 | 0.1 | 0.1 | 0.1 | 0.0 | 0.1 |

Attention matrix between Q and Image

Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.

# Meta architecture




How to do all the vision tasks in a unified model?

- Classification
- Segmentation(semantic, instance, panoptic)
- Detection

# Meta architecture

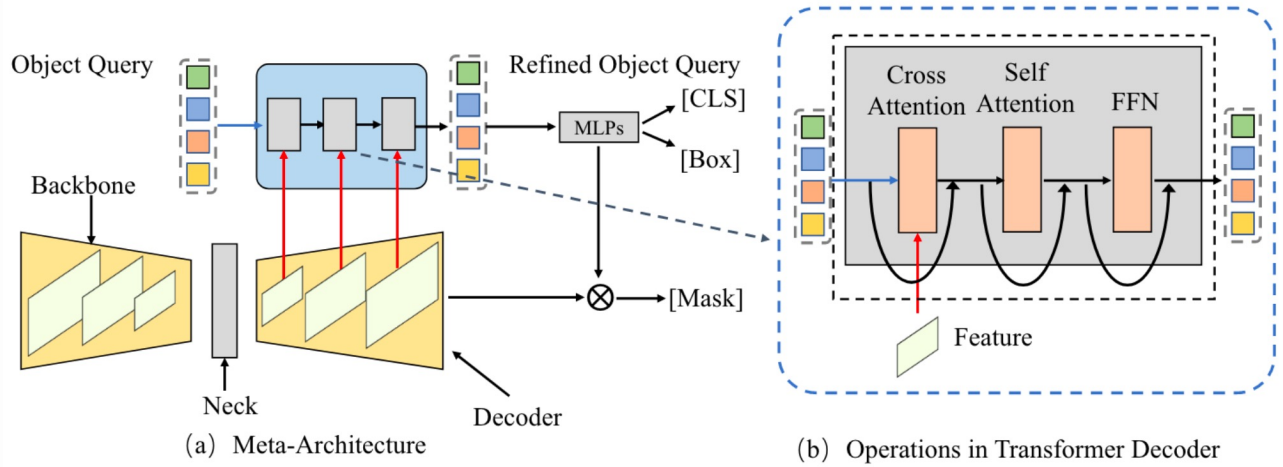



Fig. 3: Illustration of (a) meta-architecture and (b) common operations in the decoder.

- Backbone

$$[C, H, W] \Rightarrow [\frac{W \times H}{P^2}, d]$$

可以在序列前加上一个learnable cls token, 变成 $[\frac{W \times H}{P^2} + 1, d]$

Backbone 输出的 cls token($[1, d]$) 经过 MLP 进行分类，也可以后面再分类

- Neck

FPN(Feature Pyramid Network): Compute a multi-scale feature representation

$$F_1: [\frac{W}{P}, \frac{H}{P}, d] \Rightarrow F_2: [W', H', d] \Rightarrow F_3: [W, H, d]$$

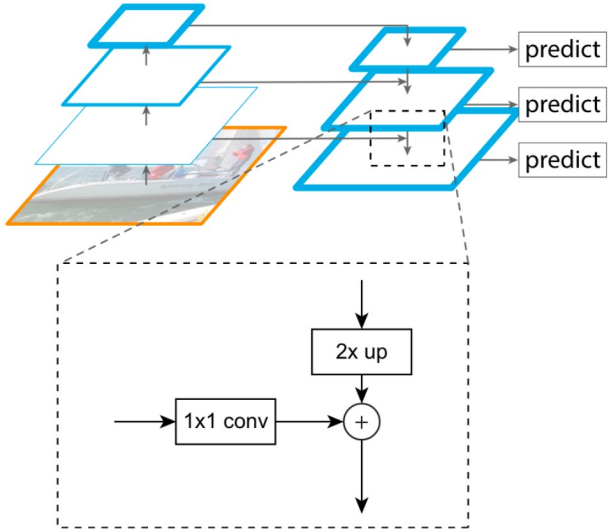



Figure 3. A building block illustrating the lateral connection and the top-down pathway, merged by addition.



Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature Pyramid Networks for Object Detection.

# Meta architecture

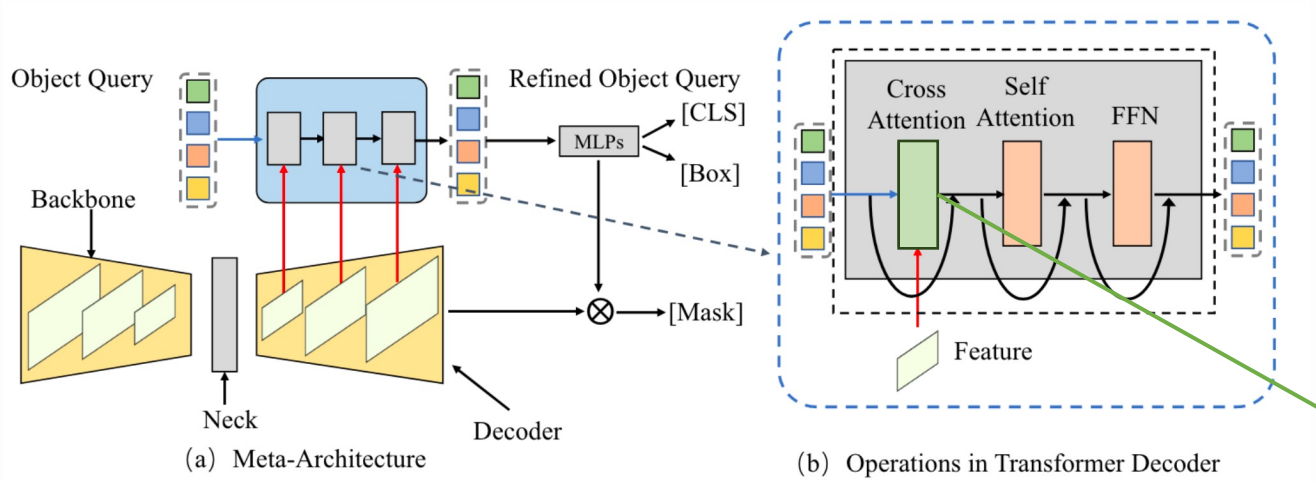



Fig. 3: Illustration of (a) meta-architecture and (b) common operations in the decoder.

- Transformer Decoder

$F_1$: $[\frac{W\times H}{P^2}, d]$ $F_2$: $[W'\times H', d]$ $F_3$: $[W\times H, d]$

Object query: $Q_0 \in \mathbb{R}^{l_q\times d}$, 0值初始化

$$K_1 = F_1 W_{1q} \in \mathbb{R}^{\frac{W\times H}{P^2}\times d},\ W_{1q} \in \mathbb{R}^{d\times d}$$
$$V_1 = F_1 W_{1v} \in \mathbb{R}^{\frac{W\times H}{P^2}\times d},\ W_{1v} \in \mathbb{R}^{d\times d}$$
$$Q_1 = \text{softmax}\left(\frac{Q_0 \cdot {K_1}^{\mathrm{T}}}{\sqrt{\mathrm{d}}}\right) V_1 \in \mathbb{R}^{l_q\times d}$$

……

$$K_3 = F_1 W_{3q} \in \mathbb{R}^{(W\times H)\times d},\ W_{3q} \in \mathbb{R}^{d\times d}$$
$$V_3 = F_1 W_{3v} \in \mathbb{R}^{(W\times H)\times d},\ W_{3v} \in \mathbb{R}^{d\times d}$$
$$Q_3 = \text{softmax}\left(\frac{Q_3 \cdot {K_3}^{\mathrm{T}}}{\sqrt{\mathrm{d}}}\right) V_3 \in \mathbb{R}^{l_q\times d}$$

- Learnable Object Query

Depending on the task and objectives, the number of learnable tokens is chosen, and the values of these tokens are altered through error-propagated gradient descent.

Attention matrix between Q and Image

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| q1 | 0.3 | 0.0 | 0.0 | 0.2 | 0.2 | 0.1 | 0.1 | 0.1 | 0.0 |
| q2 | 0.2 | 0.3 | 0.3 | 0.0 | 0.1 | 0.1 | 0.0 | 0.0 | 0.0 |
| q3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.1 | 0.4 | 0.3 | 0.2 |
| q4 | 0.2 | 0.1 | 0.1 | 0.2 | 0.1 | 0.1 | 0.1 | 0.0 | 0.1 |

Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers[C]//European conference on computer vision. Cham: Springer International Publishing, 2020: 213-229.

# Meta architecture

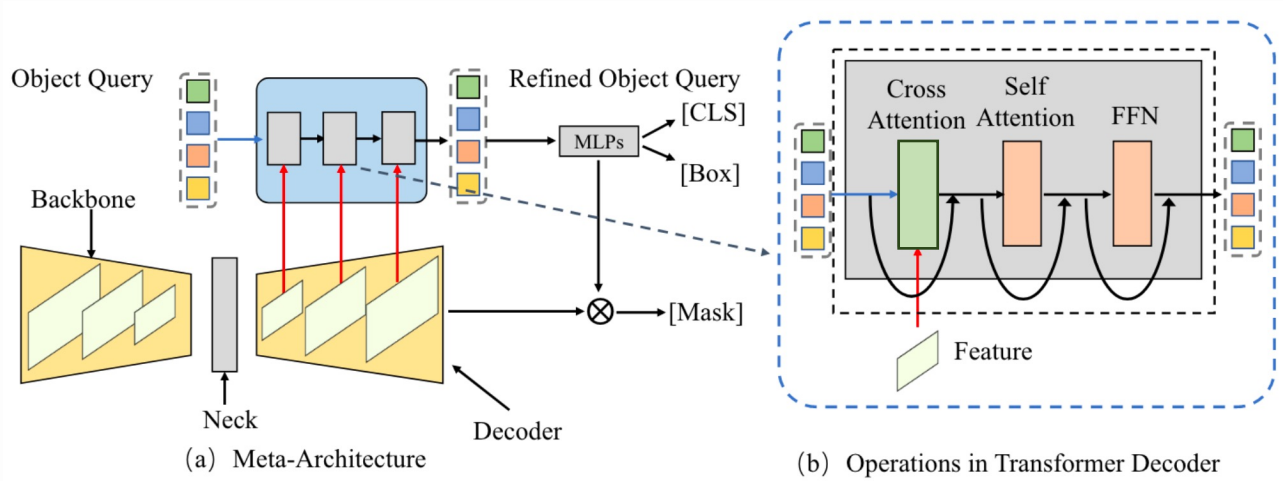



Fig. 3: Illustration of (a) meta-architecture and (b) common operations in the decoder.

- Bipartite Matching (Hungarian algorithm)

$Instance\ Mask \in \mathbb{R}^{l_q \times H \times W}$

$Ground\ Truth \in \mathbb{R}^{C \times H \times W}$, $l_q > C$, 每个样本C不同

Cost matrix

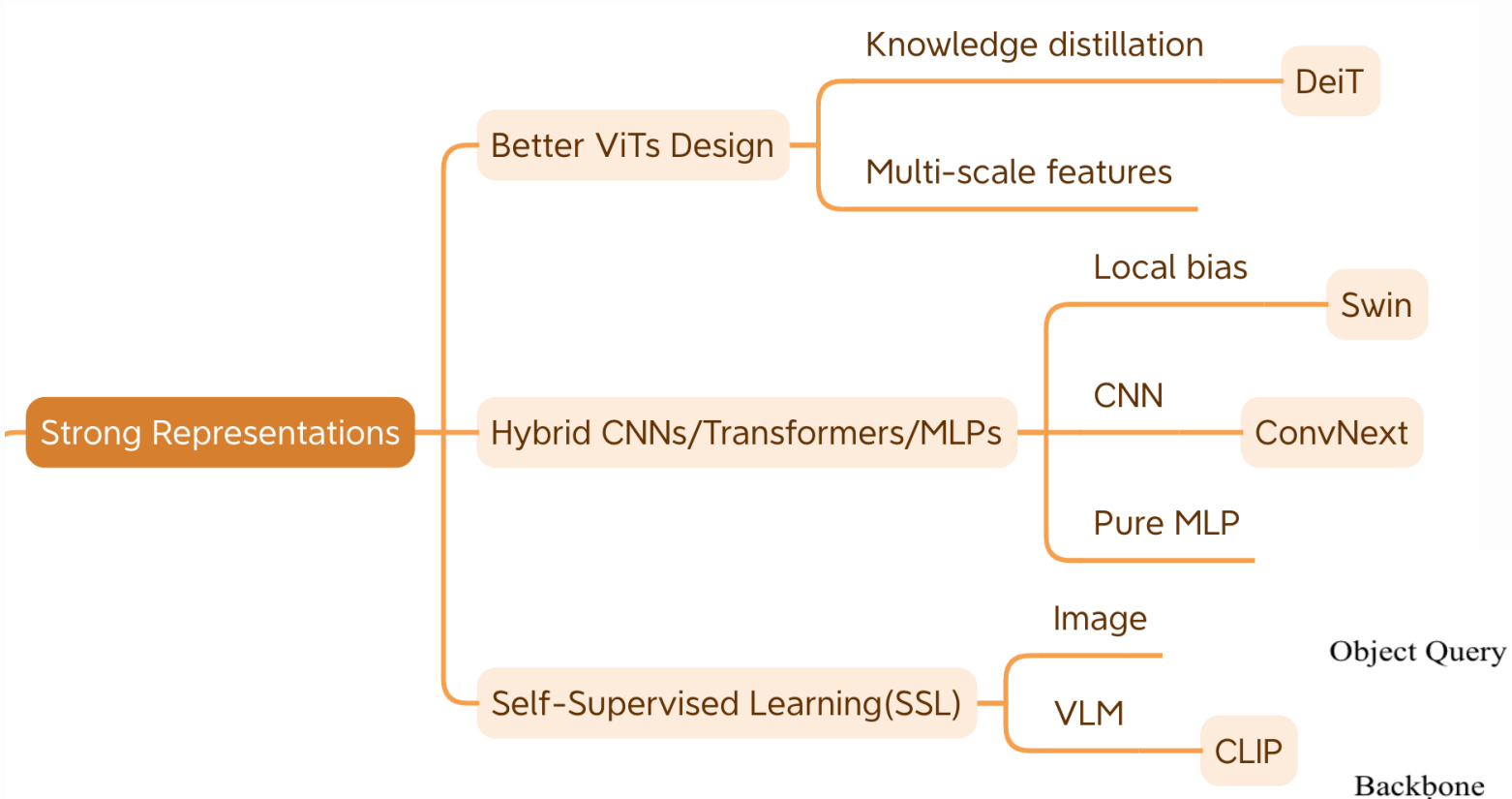
| | $M_1$ | $M_2$ | $M_3$ | $M_4$ | …… | $M_{l_q-1}$ | $M_{l_q}$ |
|---|---|---|---|---|---|---|---|
| $G_1$ | | | | | | | |
| $G_2$ | | | | | | | |
| ⋮ | | | | | | | |
| $G_c$ | | | | | | | |
| $\emptyset$ | | | | | | | |

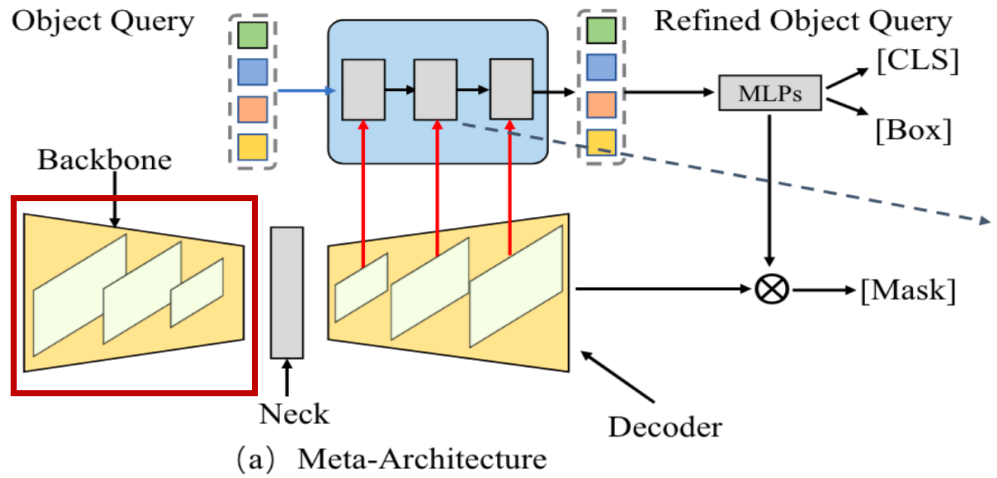
$Detection$是类似的，Cost则由cls和box定义

$Q_3 \in \mathbb{R}^{l_q \times d}$

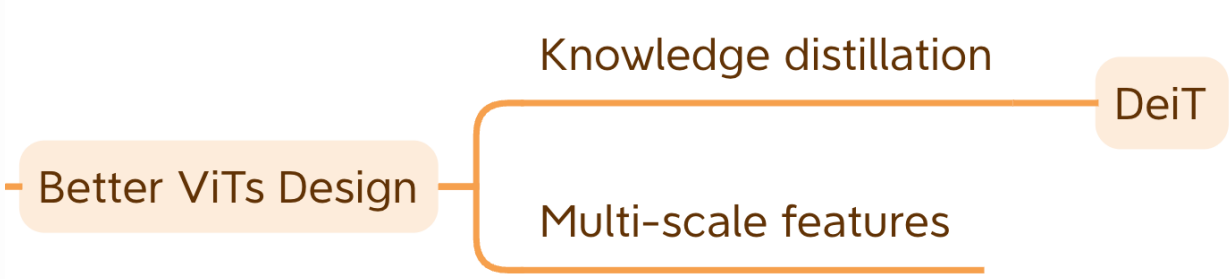
- Classification $cls = mlp(Q_3)$
- Detection $cls,\ box = mlp(Q_3) \in \mathbb{R}^{l_q \times 5}$, $l_q$为一个预设的数，如100，表示最多instance
- Segmentation
  - Instance $Mask = Sigmoid(Q_3 F_3^T) \in \mathbb{R}^{l_q \times H \times W}$, $F_3 \in \mathbb{R}^{H \times W \times d}$, $l_q$ 为一个预设的数，如100，表示最多instance
  - Semantic $Mask = Sigmoid(Q_3 F_3^T) \in \mathbb{R}^{l_q \times H \times W}$, $F_3 \in \mathbb{R}^{H \times W \times d}$, $l_q$为类别

# 1. Strong Representations



# 1. Strong Representations




- DeiT
- 提出了针对 ViT 的教师-学生蒸馏训练策略
- 提出了 token-based distillation 方法

"The transformers do not generalize well when trained on insufficient amounts of data."

$$\mathcal{L}_{\text{global}} = (1-\lambda)\mathcal{L}_{\text{CE}}(\psi(Z_{\text{s}}), y) + \lambda\tau^2 \text{KL}(\psi(Z_{\text{s}}/\tau), \psi(Z_{\text{t}}/\tau)). \quad (2)$$

$$\mathcal{L}_{\text{global}}^{\text{hardDistill}} = \frac{1}{2}\mathcal{L}_{\text{CE}}(\psi(Z_s), y) + \frac{1}{2}\mathcal{L}_{\text{CE}}(\psi(Z_s), y_{\text{t}}). \quad (3)$$

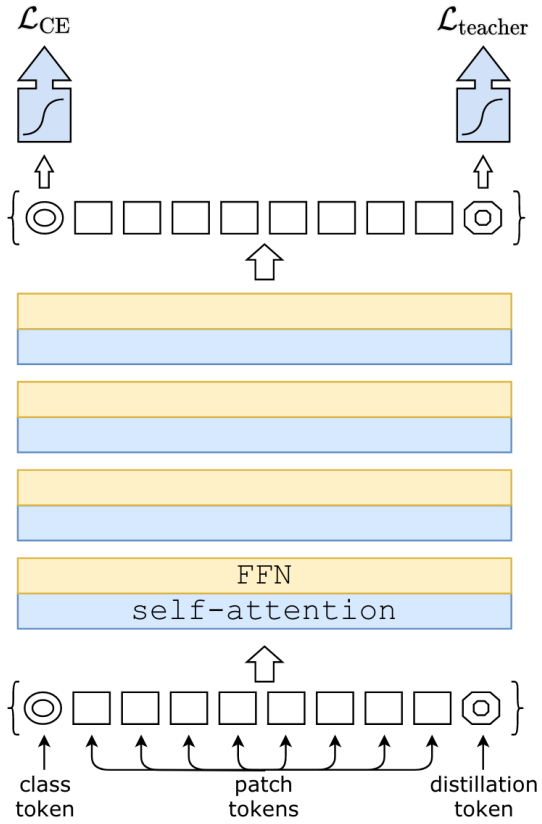


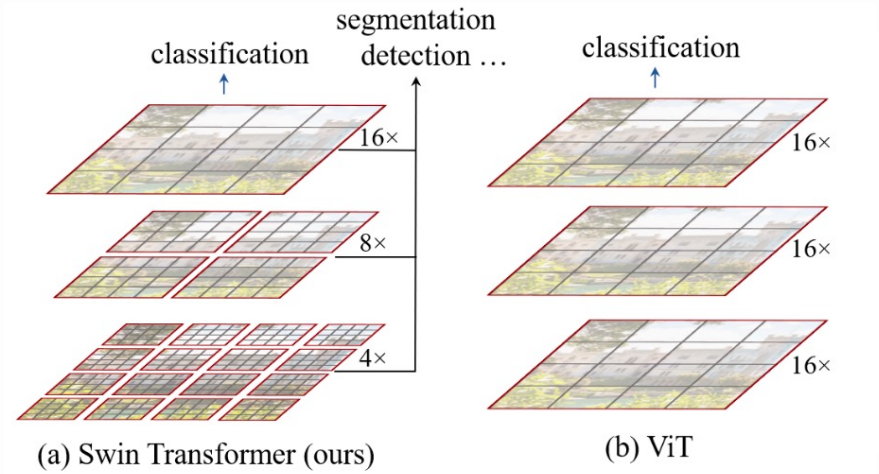

Figure 2: Our distillation procedure: we simply include a new *distillation token*. It interacts with the class and patch tokens through the self-attention layers. This distillation token is employed in a similar fashion as the class token, except that on output of the network its objective is to reproduce the (hard) label predicted by the teacher, instead of true label. Both the class and distillation tokens input to the transformers are learned by back-propagation.

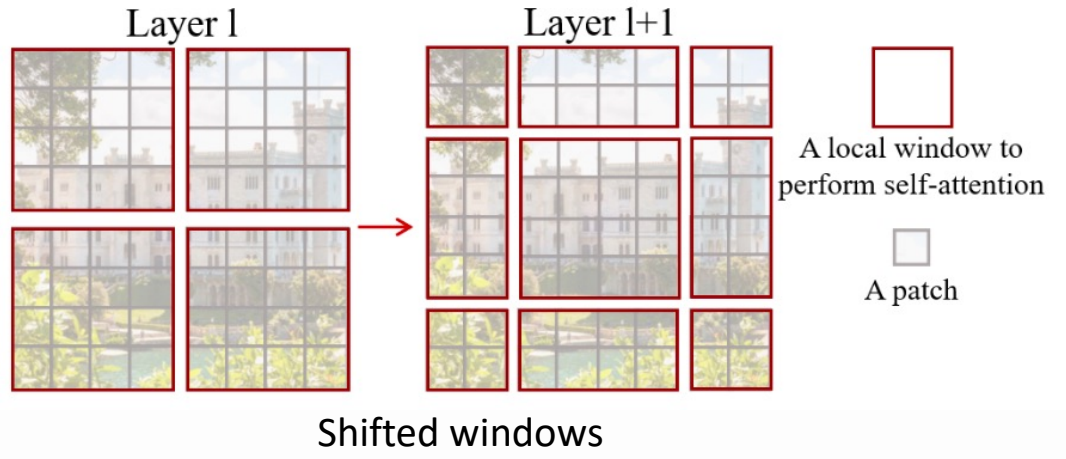


Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H., 2021. Training data-efficient image transformers & distillation through attention.

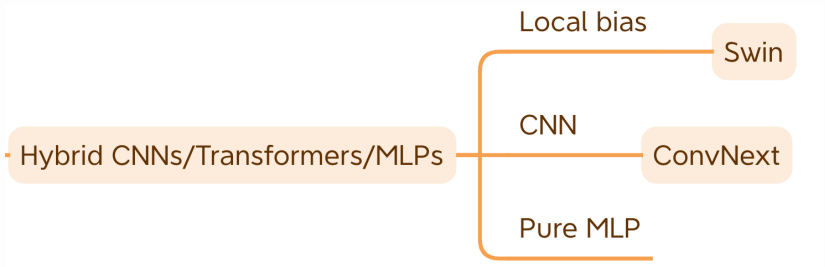
# 1. Strong Representations

- Swin Transformer
- General backbone




- 在ViT中引入inductive bias
- Patch merging
- Shifted windows



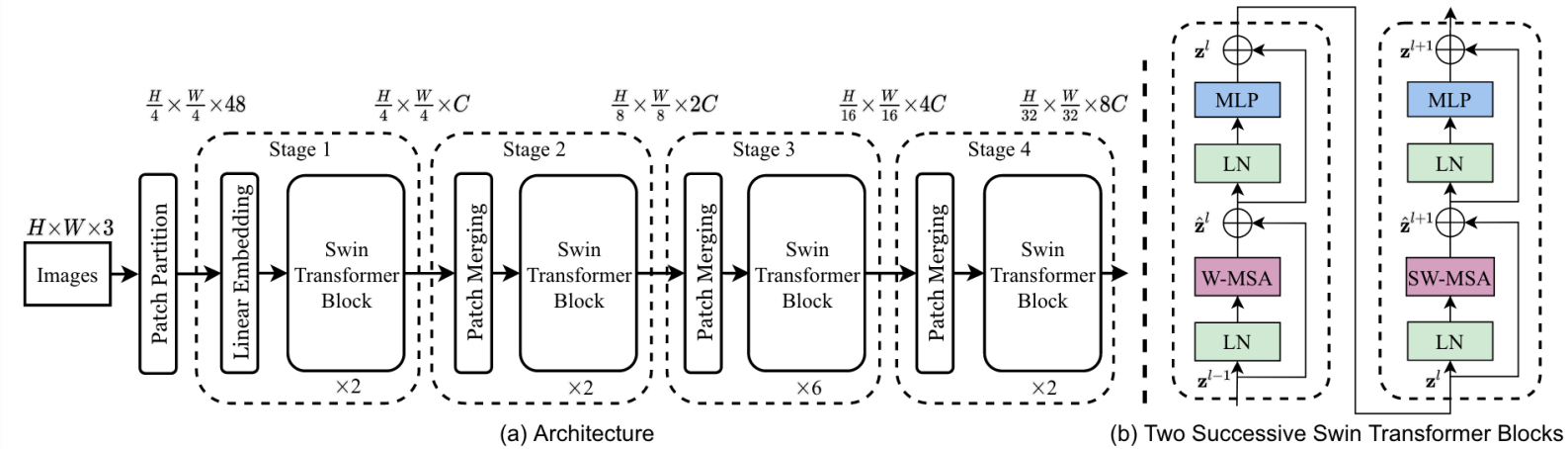




Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10012-10022.

# 1. Strong Representations

- ConvNeXt, facebook
- 用训练ViT的一些灵感来改造CNN

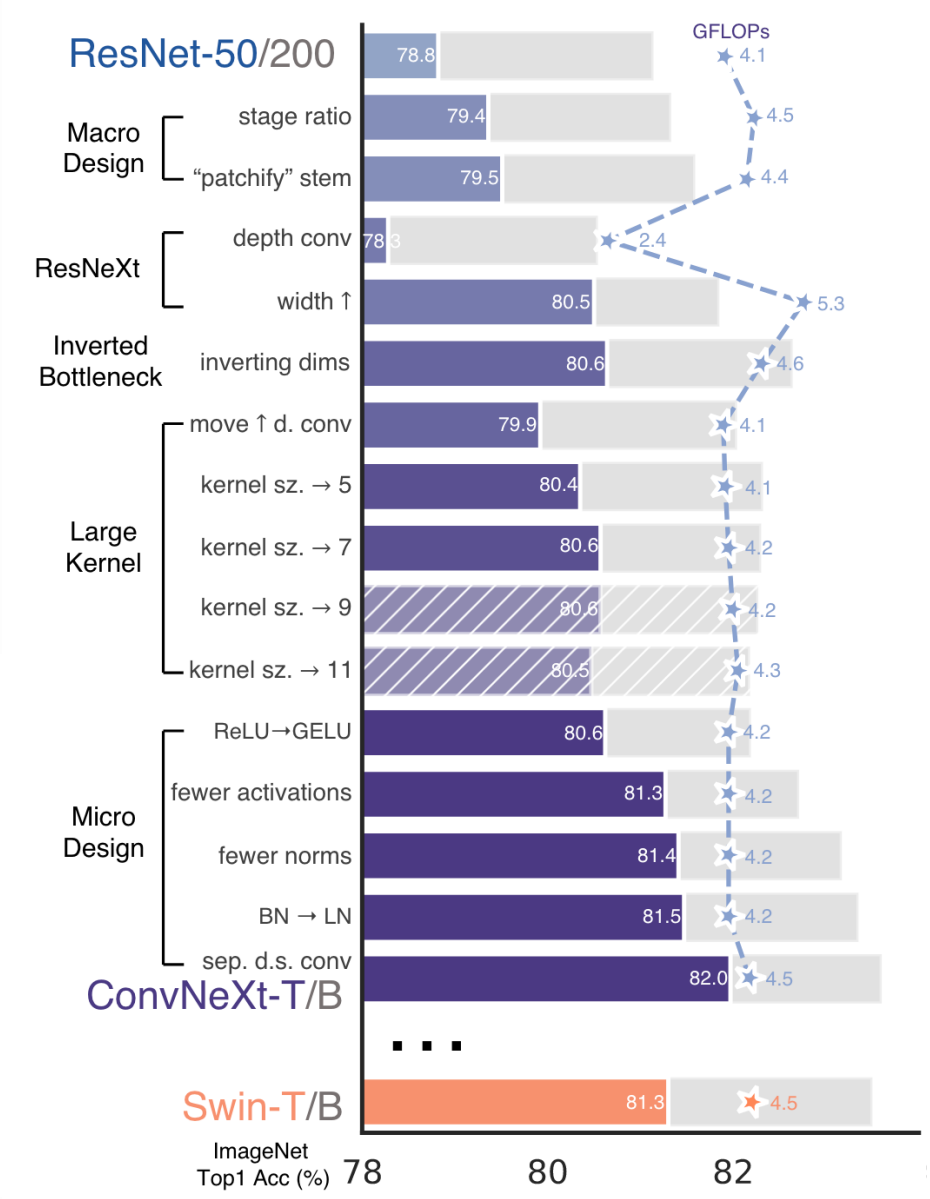
■ Training techniques: Optimization strategy, hyper-parameter settings (学习DeiT和Swin Transformer)

AdamW optimizer, data augmentation techniques such as Mixup, Cutmix, RandAugment, Random Erasing, and regularization schemes including Stochastic Depth and Label Smoothing.

■ Macro Design

■ Micro Design

| model | image size | #param. | FLOPs | throughput (image / s) | IN-1K top-1 acc. |
|---|---|---|---|---|---|
| ImageNet-1K trained models | | | | | |
| • RegNetY-16G [54] | $224^2$ | 84M | 16.0G | 334.7 | 82.9 |
| • EffNet-B7 [71] | $600^2$ | 66M | 37.0G | 55.1 | 84.3 |
| • EffNetV2-L [72] | $480^2$ | 120M | 53.0G | 83.7 | 85.7 |
| ○ DeiT-S [73] | $224^2$ | 22M | 4.6G | 978.5 | 79.8 |
| ○ DeiT-B [73] | $224^2$ | 87M | 17.6G | 302.1 | 81.8 |
| ○ Swin-T | $224^2$ | 28M | 4.5G | 757.9 | 81.3 |
| • ConvNeXt-T | $224^2$ | 29M | 4.5G | 774.7 | **82.1** |
| ○ Swin-S | $224^2$ | 50M | 8.7G | 436.7 | 83.0 |
| • ConvNeXt-S | $224^2$ | 50M | 8.7G | 447.1 | **83.1** |
| ○ Swin-B | $224^2$ | 88M | 15.4G | 286.6 | 83.5 |
| • ConvNeXt-B | $224^2$ | 89M | 15.4G | 292.1 | **83.8** |
| ○ Swin-B | $384^2$ | 88M | 47.1G | 85.1 | 84.5 |
| • ConvNeXt-B | $384^2$ | 89M | 45.0G | 95.7 | **85.1** |
| • ConvNeXt-L | $224^2$ | 198M | 34.4G | 146.8 | **84.3** |
| • ConvNeXt-L | $384^2$ | 198M | 101.0G | 50.4 | **85.5** |






Liu Z, Mao H, Wu C Y, et al. A convnet for the 2020s[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 11976-11986.

# 1. Strong Representations

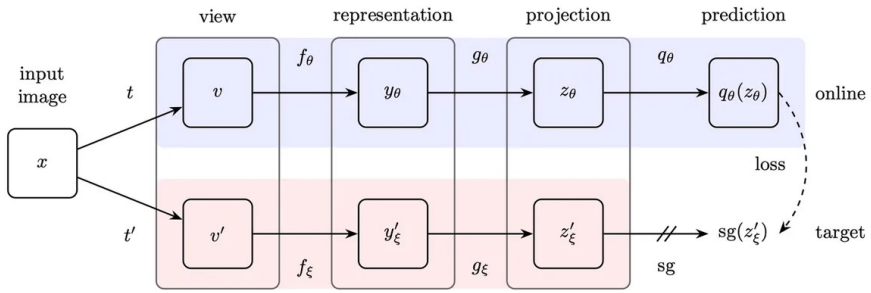
- Self-Supervised Learning(SSL)
  - Image
  - VLM
  - CLIP

**Masked Autoencoders Are Scalable Vision Learners**

Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick
*equal technical contribution †project lead
Facebook AI Research (FAIR)

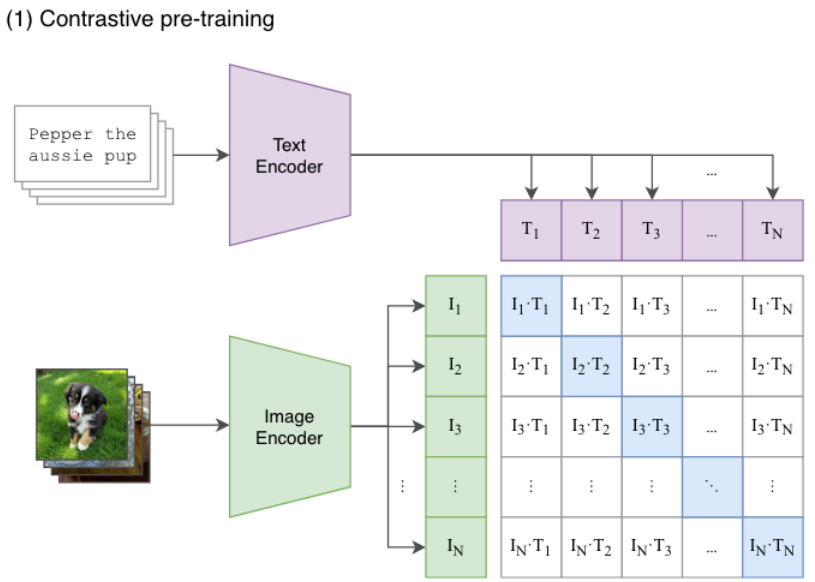
**Bootstrap Your Own Latent**
**A New Approach to Self-Supervised Learning**

Jean-Bastien Grill*[1] , Florian Strub*[1] , Florent Altché*[1] , Corentin Tallec*[1] , Pierre H. Richemond*[1,2]
Elena Buchatskaya[1] , Carl Doersch[1] , Bernardo Avila Pires[1] , Zhaohan Daniel Guo[1]
Mohammad Gheshlaghi Azar[1], Bilal Piot[1], Koray Kavukcuoglu[1] , Rémi Munos[1] , Michal Valko[1]
[1]DeepMind [2]Imperial College
[jbgrill,fstrub,altche,corentint,richemond]@google.com



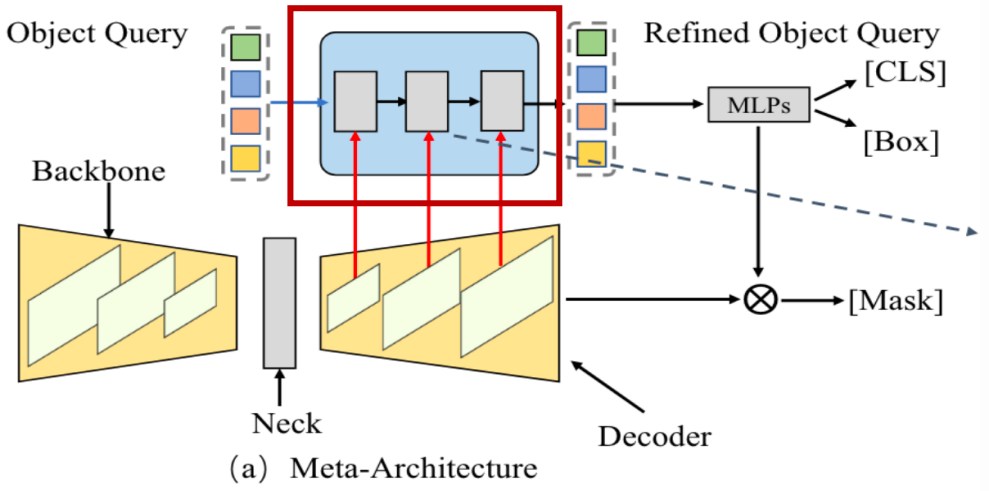

**Learning Transferable Visual Models From Natural Language Supervision**

Alec Radford *[1] Jong Wook Kim *[1] Chris Hallacy [1] Aditya Ramesh [1] Gabriel Goh [1] Sandhini Agarwal [1]
Girish Sastry [1] Amanda Askell [1] Pamela Mishkin [1] Jack Clark [1] Gretchen Krueger [1] Ilya Sutskever [1]



# 2. Interaction design in Decoder

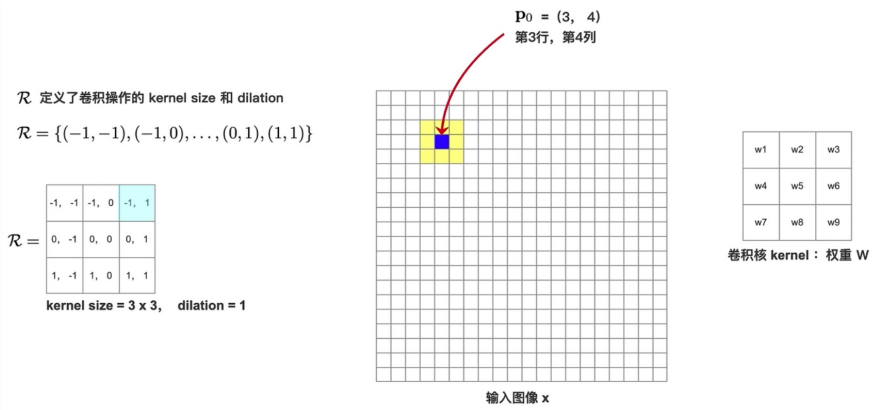
- Interaction Design in Decoder
  - Improved Cross-Attention Design
    - Deformable DETR
    - MaskFormer, Mask2Former
  - Spatial-Temporal Cross-Attention Design
    - message token
    - IFC



# 2. Interaction design in Decoder

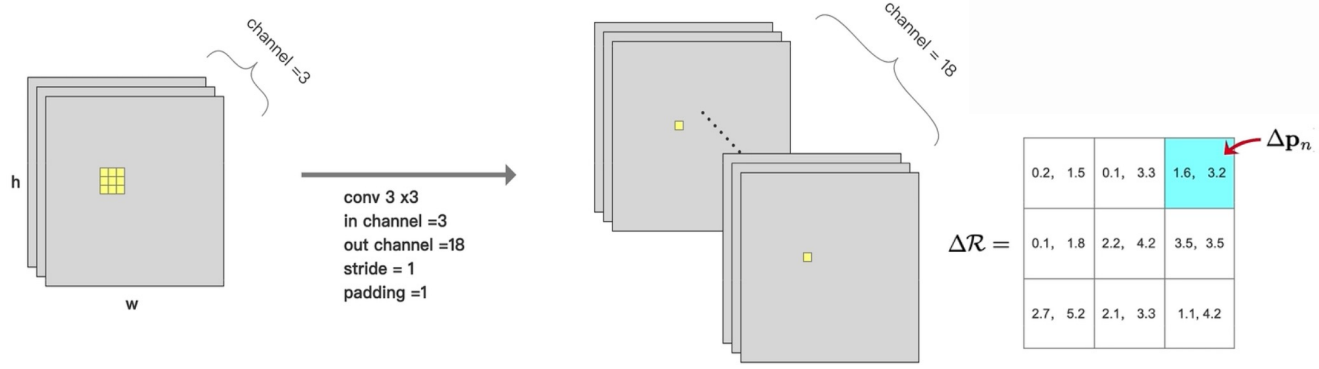
Deformable DETR • 提出了deformable Attention

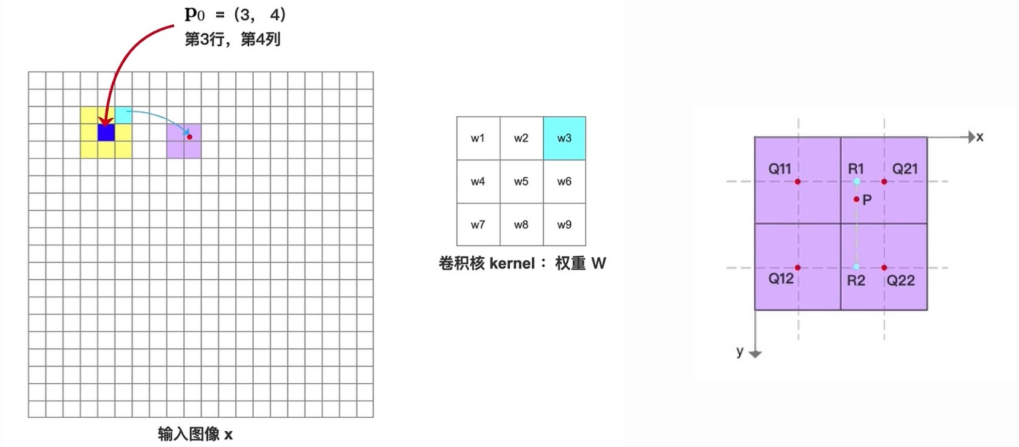
普通卷积公式： $\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n),$




可变形卷积公式： $\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta\mathbf{p}_n).$

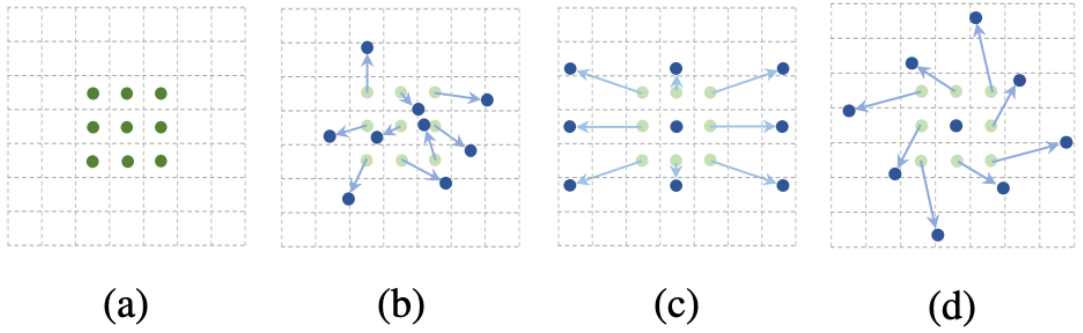
$\Delta\mathbf{p}_n$ 是学习到的值，是浮点型数据，由图像经过普通卷积计算得到




$\mathbf{p} = \mathbf{p}_0 + \mathbf{p}_n + \Delta\mathbf{p}_n$ = (3, 4) + (-1, 1) + (1.6, 3.2) = (3.6, 8.2)

(3.6, 8.2) 临近的 4个像素点分别为 (3, 8) 、 (3, 9) 、 (4, 8) 、 (4, 9)

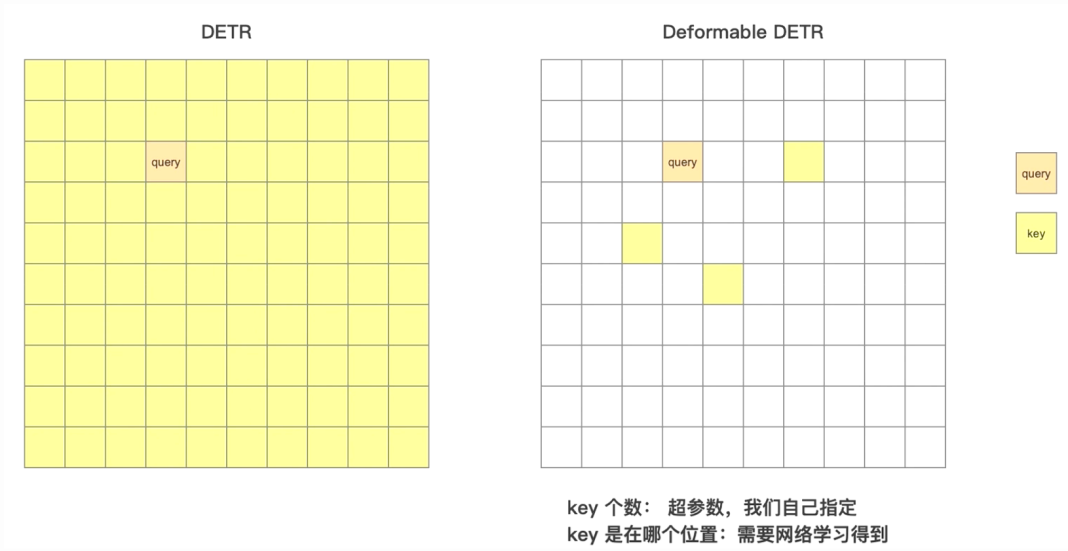


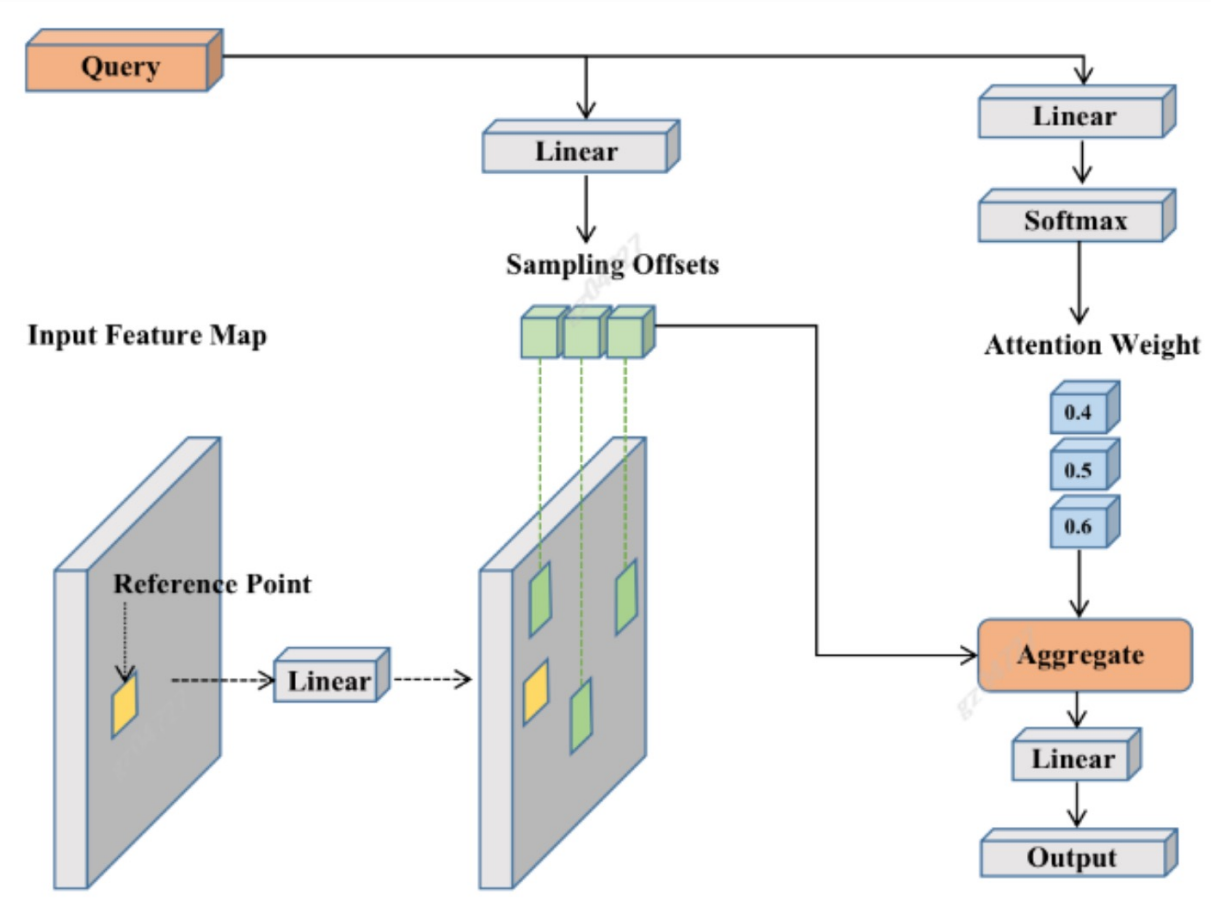

Dai J, Qi H, Xiong Y, et al. Deformable convolutional networks[C]//Proceedings of the IEEE international conference on computer vision. 2017: 764-773.

# 2. Interaction design in Decoder

Deformable DETR • 提出了deformable Attention

Attention modules only attend to a small set of key sampling points around a reference.






Zhu X, Su W, Lu L, et al. Deformable DETR: Deformable Transformers for End-to-End Object Detection[C]//International Conference on Learning Representations. 2020.

# 2. Interaction design in Decoder

IFC message token $[f_t^0, m_t^0] \in \mathbb{R}^{(HW+M)\times C}, \quad t \in \{1, 2, \cdots, T\},$

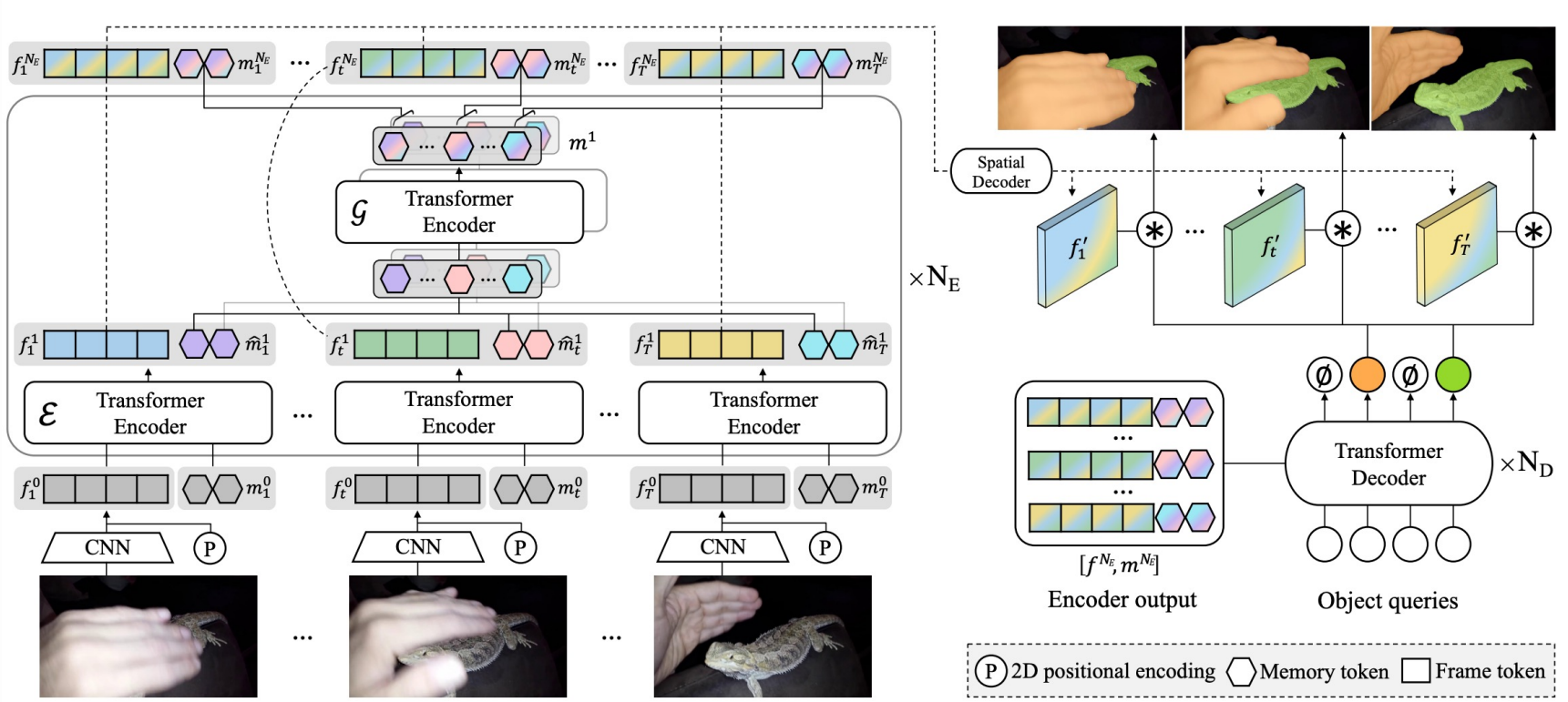
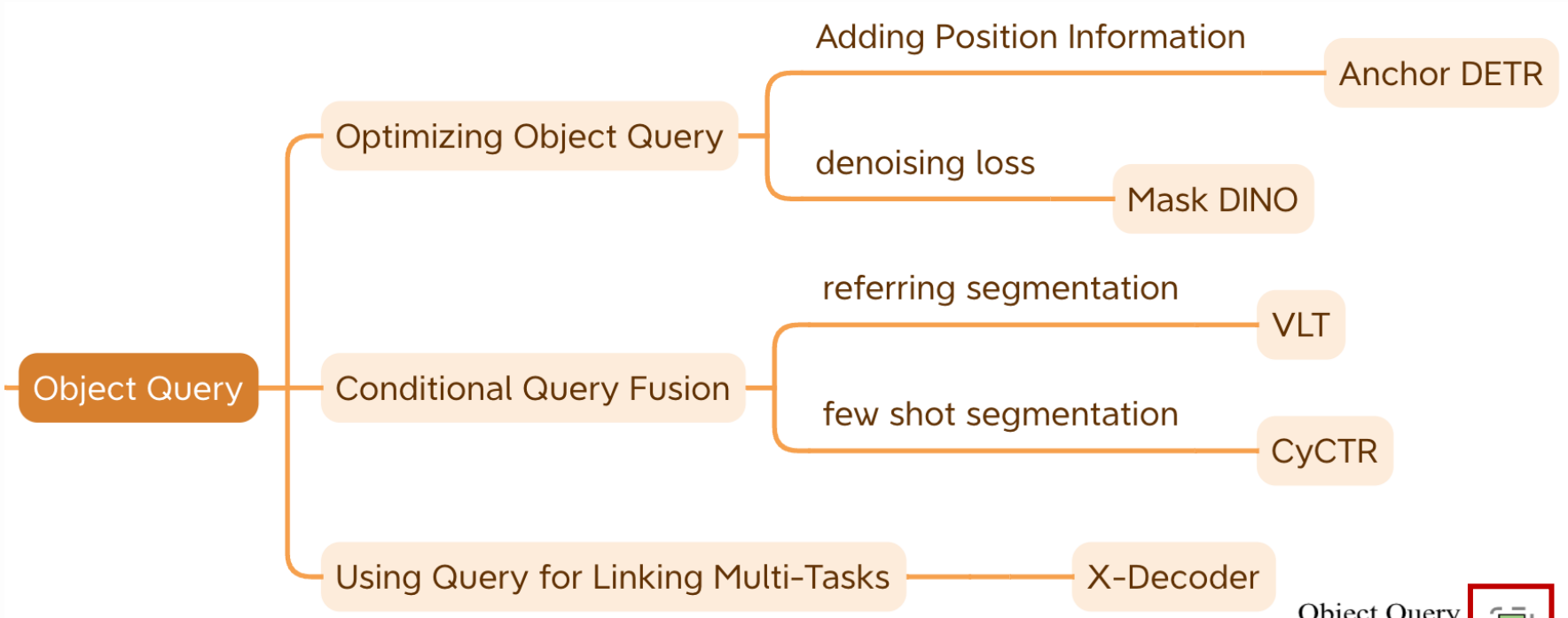


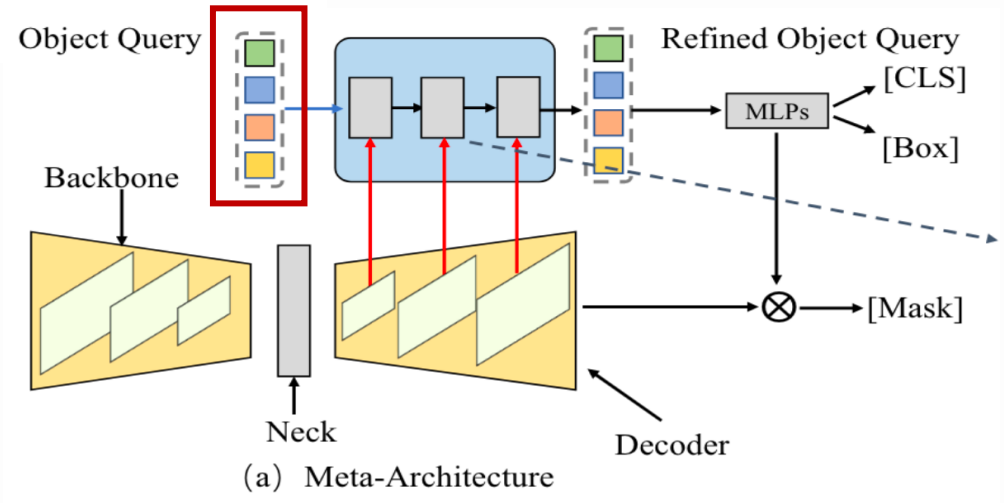

Figure 1: Overview of IFC framework. Our transformer encoder block has two components: 1) Encode-Receive ($\mathcal{E}$) simultaneously encodes frame tokens and memory tokens. 2) Only memory tokens pass Gather-Communicate ($\mathcal{G}$) to perform communications between frames. The outputs from the stack of $N_E$ encoder blocks goes into two modules, spatial decoder and transformer decoder, to generate segmentation masks.



Hwang S, Heo M, Oh S W, et al. Video instance segmentation using inter-frame communication transformers[J]. Advances in Neural Information Processing Systems, 2021, 34: 13352-13363.

# 3. Object Query



# 3. Object Query

## X-decoder

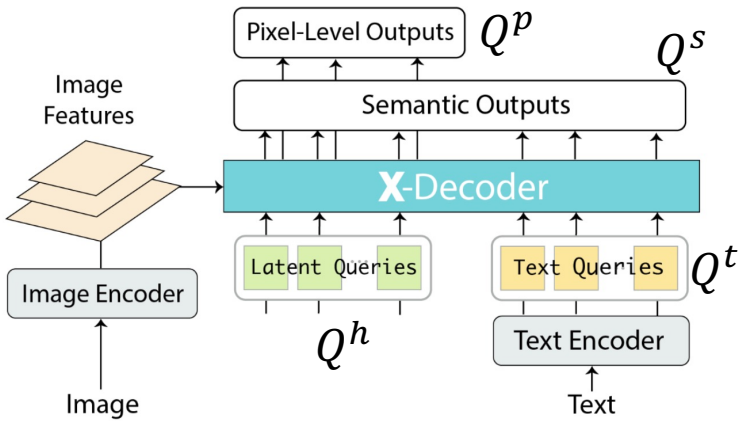



**Figure 2.** Overall pipeline for our model. It consists of an image encoder, a text encoder and our own designed X-Decoder.

Input image $\mathbf{I} \in \mathcal{R}^{H \times W \times 3}$, image encoder $\mathbf{Enc}_I$ => features $\mathbf{Z} \in \mathcal{R}^{H'W' \times d}$

Textual query $\mathbf{T}$ , text encoder $\mathbf{Enc}_T$ => $\mathbf{Q^t} = \langle q_1^t, \cdots, q_n^t \rangle$

Latent queries $\mathbf{Q^h} = \langle q_1^h, \cdots, q_m^h \rangle$ are fed to our X-Decoder to predict the outputs:

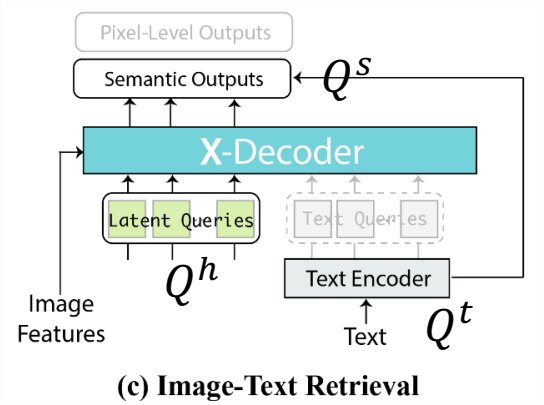
$$\langle \mathbf{O^p}, \mathbf{O^s} \rangle = \mathbf{XDec}(\langle \mathbf{Q^h}, \mathbf{Q^t} \rangle; \mathbf{Z})$$

where $\mathbf{O^p}$ and $\mathbf{O^s}$ are the pixel-level masks and token-level semantics, respectively

Latent queries: m learnable queries, the last latent query to **extract the global image representation** and the remaining for **generic segmentation**

$Q^h$ Z Z $\Rightarrow Q^s$

- Image-text Retrieval




**(c) Image-Text Retrieval**

The last valid token feature of $\mathbf{Q^t}$ from the text encoder to represent a text as $\hat{q}^t$

The last entry in $\mathbf{O}^s$ derived from X-Decoder as $\hat{o}^s$

$B$ pairs of features $\left\langle \hat{q}_i^t, \hat{o}_i^s \right\rangle_{i=1}^{B}$ for a minibatch of $B$ image-text pairs

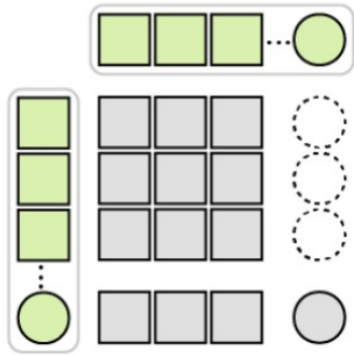
Dot-product between these $B \times B$ feature pairs => affinity matrix $\mathbf{S}_{it} \in \mathcal{R}^{B \times B}$

Compute the bidirectional cross-entropy loss:

$$\mathcal{L}_{it} = \mathbf{CE}(\mathbf{S}_{it}, \mathbf{y}_{it}) + \mathbf{CE}(\mathbf{S}_{it}^T, \mathbf{y}_{it})$$

where $\mathbf{y}_{it}$ are the class labels corresponding to diagonal entries in $\mathbf{S}_{it}$

Self-attention mask



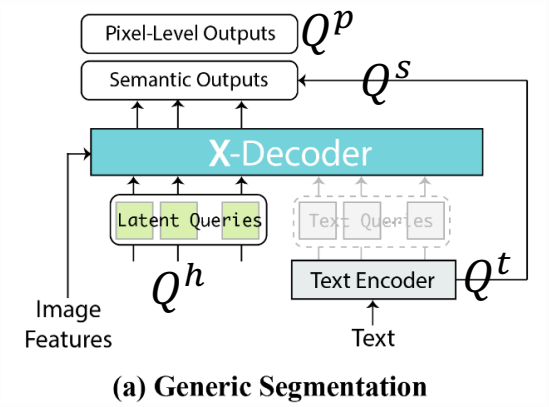
**(a) Seg + Ret**



Zou X, Dou Z Y, Yang J, et al. Generalized decoding for pixel, image, and language[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 15116-15127.

# 3. Object Query

- Generic segmentation



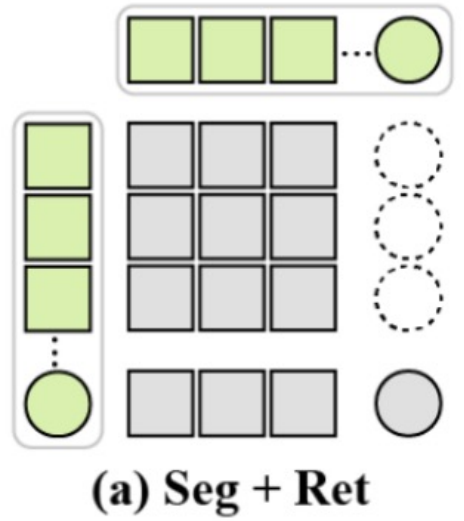
(a) Generic Segmentation

Self attention $\underset{Q}{Q^h}, \underset{K}{Q^h}, \underset{V}{Q^h} \Longrightarrow Q^{h\prime}$

Cross attention $\underset{Q}{Q^{h\prime}}, \underset{K}{Z}, \underset{V}{Z} \Rightarrow O^s$

$Sigmoid(\widehat{Q^s}\, Z^T) = O^p$, m − 1个instance mask

Bipartite Matching (Hungarian algorithm)


(a) Seg + Ret

Self-attention mask

Mask classification

Encode all $C$ class names including "background" into $C$ text queries
Take the last valid token feature from each to represent the concept

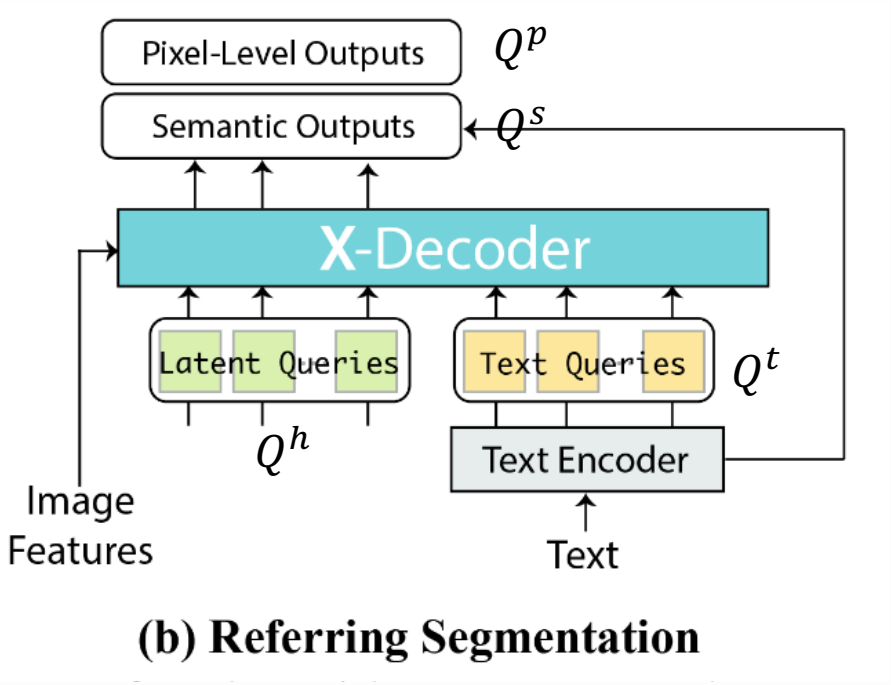
Use the first $(m-1)$ query in $O^s$ to compute the dot-product
Between these outputs and concept embeddings => affinity matrix $\mathbf{S}_{cls} \in \mathcal{R}^{(m-1)\times C}$

Compute the loss $\mathcal{L}_{cls} = \mathbf{CE}(\mathbf{S}_{cls}, \mathbf{y}_{cls})$, with the ground-truth class $\mathbf{y}_{cls}$

Semantic Seg: combine the instance seg output and class label

# 3. Object Query

- Referring Segmentation




**(b) Referring Segmentation**

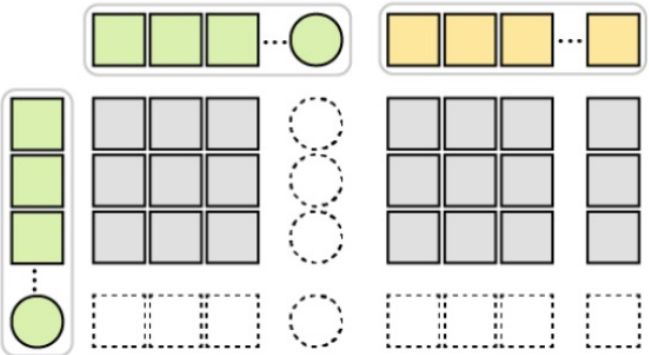
Conditional Self attention $\overset{Q}{Q^h}, \overset{K}{< Q^h, Q^t >}, \overset{V}{< Q^h, Q^t >} \Longrightarrow Q^{h\prime}$

Cross attention $\overset{Q}{Q^{h'}}, \overset{K}{Z}, \overset{V}{Z} \Rightarrow O^s$

$Sigmoid\left(\widehat{Q^s}\, Z^T\right) = O^p,$ $\mathrm{m}-1$个instance mask

Bipartite Matching (Hungarian algorithm)

Self-attention mask

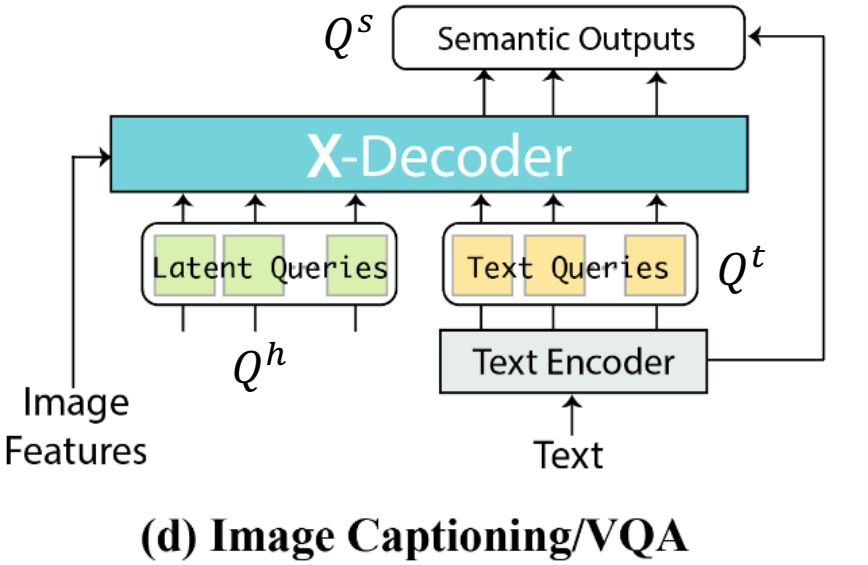


**(b) Referring Segmentation**

Other than the text used as a condition, everything else is the same as generic segmentation.

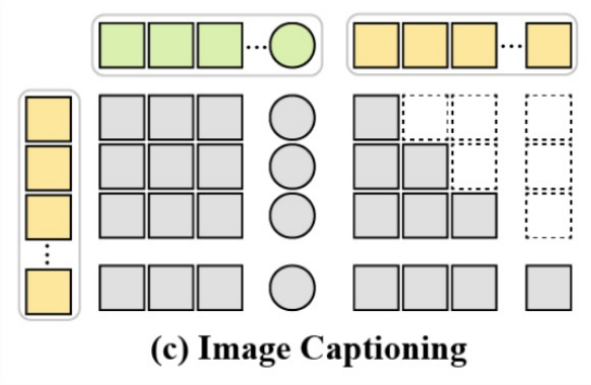
# 3. Object Query

- Image Captioning/VQA




(d) Image Captioning/VQA

Conditional Self attention $\overset{Q}{Q^t}, \overset{K}{< Q^h, Q^t >}, \overset{V}{< Q^h, Q^t >} \Longrightarrow Q^{h\prime}$



(c) Image Captioning

Self-attention mask

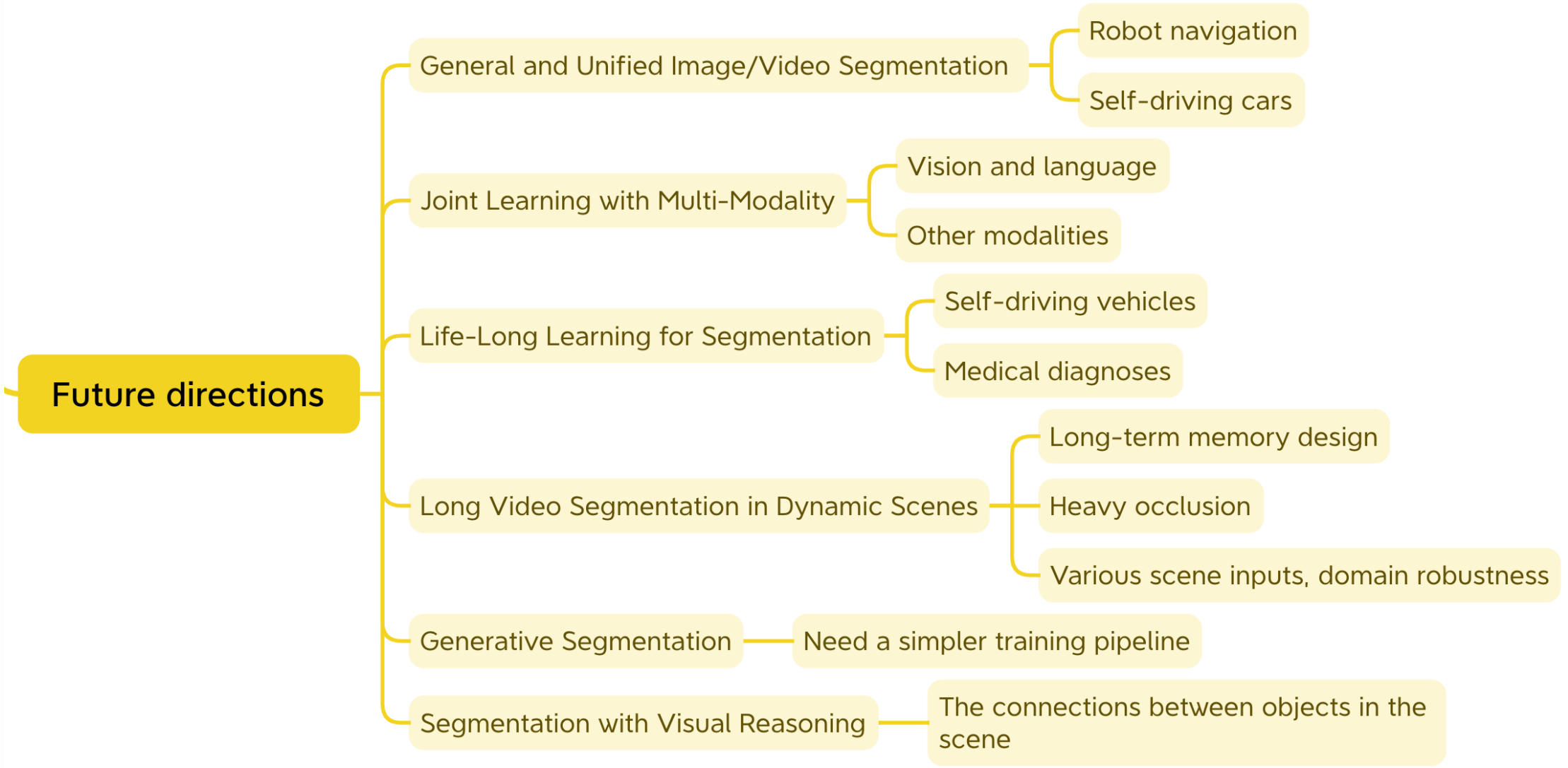
For free text image caption

Cross attention $\overset{Q}{Q^{h\prime}}, \overset{K}{\mathrm{Z}}, \overset{V}{\mathrm{Z}} \Rightarrow O^s$

The caption prediction follows a causal masking strategy while VQA use the last query in $O^s$ to predict the answer for VQA.

All types of tasks are trained together, with the distinction lying solely in the combination of different modules and computation methods.

# FUTURE DIRECTIONS



# Thanks

8.15 Journal Club