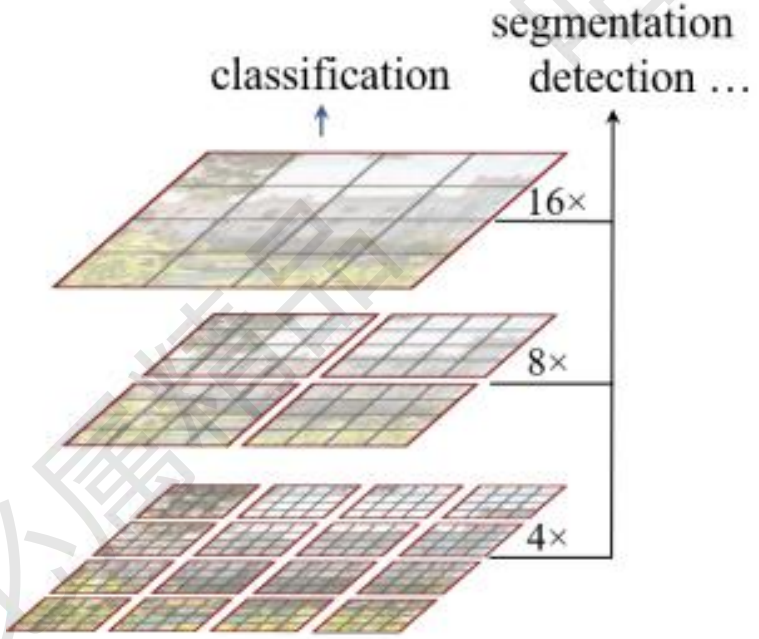
SwinTransformer

- 先来吹一波
  - 分类，分割，检测等任务中均是刷分神器
  - 官方终于开源了，各预训练模型全部给出
  - 新一代backbone，可直接套用在各项下游任务中
  - 提供大，中，小个版本模型；可以自由选择合适的



SwinTransformer

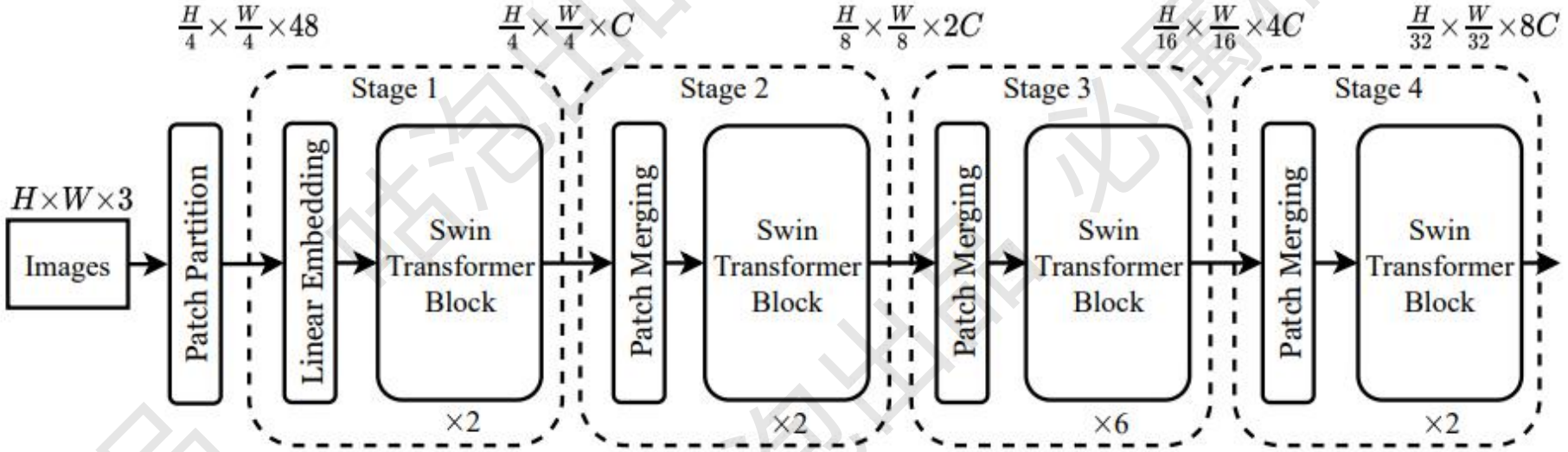
- 解决了哪些问题呢？
  - 图像中像素点太多了，如果需要更多的特征就必须构建很长的序列
  - 越长的序列算起注意力肯定越慢，这就导致了效率问题
  - 能否用窗口和分层的形式来替代长序列的方法呢？这就是它的本质
  - CNN中经常提到感受野，transformer中该如何体现呢？（答案就是分层）

SwinTransformer

- 整体网络架构

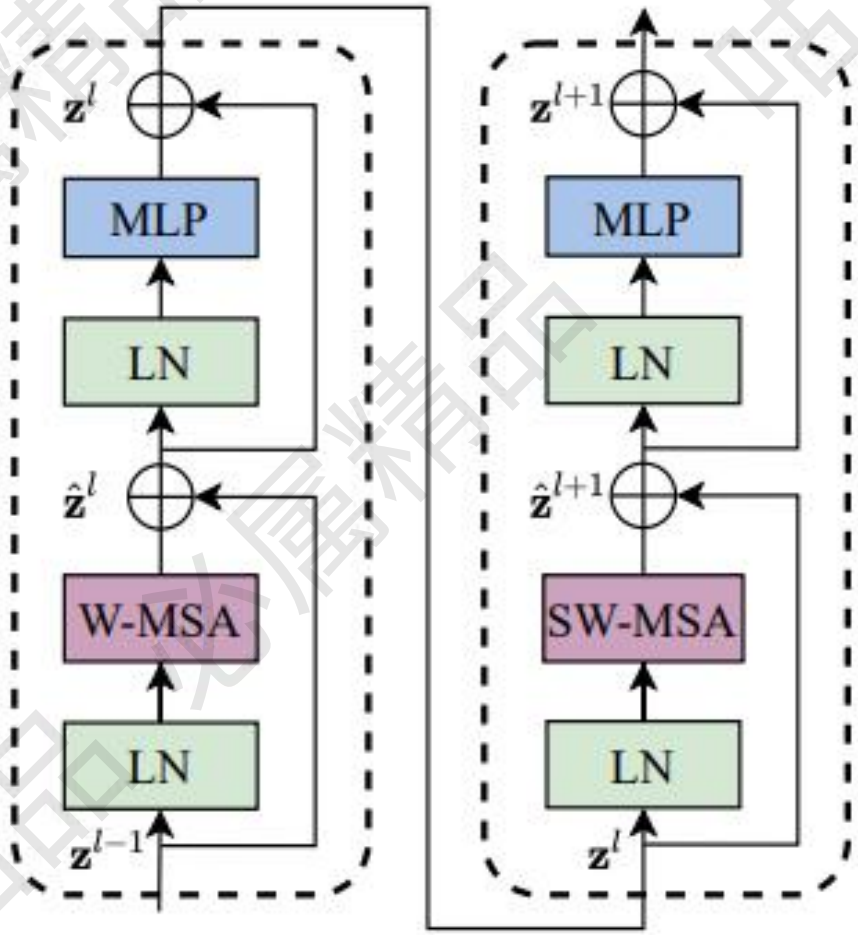
- 1.得到各Pathch特征构建序列；2.分层计算attention（逐步下采样过程）

- 其中Block是最核心的，对attention的计算方法进行了改进



SwinTransformer

- Transformer Blocks
  - 右图这俩是一个组合（得一起上）
  - W-MSA：基于窗口的注意力计算
  - SW-MSA：窗口滑动后重新计算注意力
  - 它俩串联在一起就是一个block



SwinTransformer

- Patch Embedding
  - 输入：图像数据（224，224，3）
  - 输出：（3136，96）相当于序列长度是3136个，每个的向量是96维特征
  - 通过卷积得到，Conv2d(3, 96, kernel_size=(4, 4), stride=(4, 4))
  - 3136也就是 (224/4) * (224/4)得到的，也可以根据需求更改卷积参数

SwinTransformer

- window_partition
  - 输入：特征图（56，56，96）
  - 默认窗口大小为7，所以总共可以分成8*8个窗口
  - 输出：特征图（64，7，7，96）
  - 之前的单位是序列，现在的单位是窗口（共64个窗口）

SwinTransformer

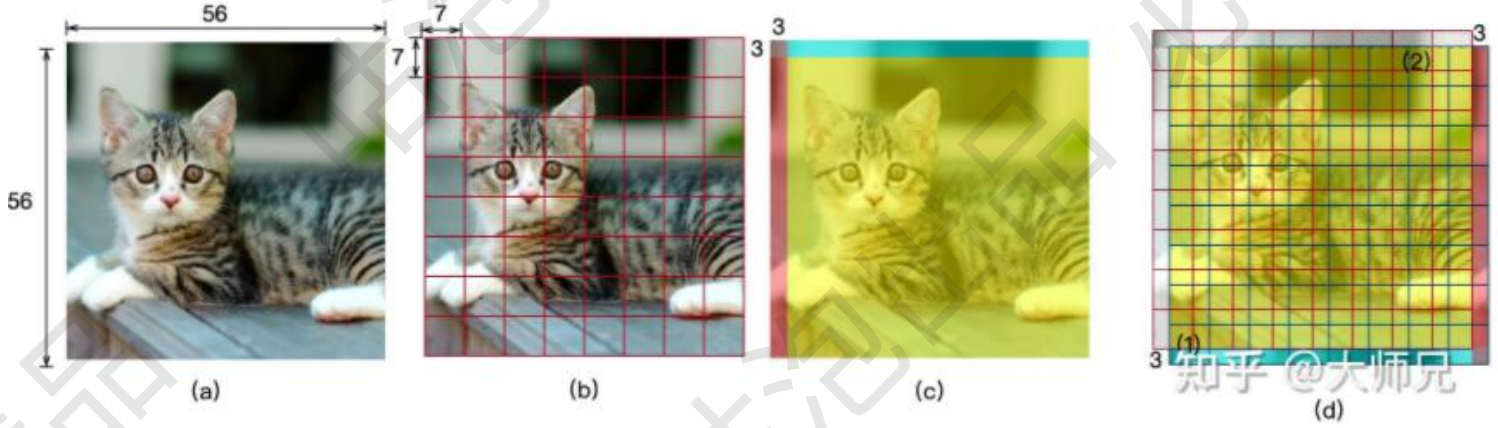
- W-MSA（Window Multi-head Self Attention）
  - 对得到的窗口，计算各个窗口自己的自注意力得分
  - qkv三个矩阵放在一起了：（3，64，3，49，32）
  - 3个矩阵，64个窗口，heads为3，窗口大小7*7=49，每个head特征96/3=32
  - attention结果为：（64，3，49，49） 每个头都会得出每个窗口内的自注意力

SwinTransformer

- window_reverse
  - 通过得到的attention计算得到新的特征（64，49，96）
  - 总共64个窗口，每个窗口7*7的大小，每个点对应96维向量
  - window_reverse就是通过reshape操作还原回去（56，56，96）
  - 这就得到了跟输入特征图一样的大小，但是其已经计算过了attention

SwinTransformer

- SW-MSA（Shifted Window）
  - 为什么要shift？原来的window都是算自己内部的
  - 这样就会导致只有内部计算，没有它们之间的关系
  - 容易上模型局限在自己的小领地，可以通过shift操作来改善（下图来自知乎）



SwinTransformer

- 位移中的细节

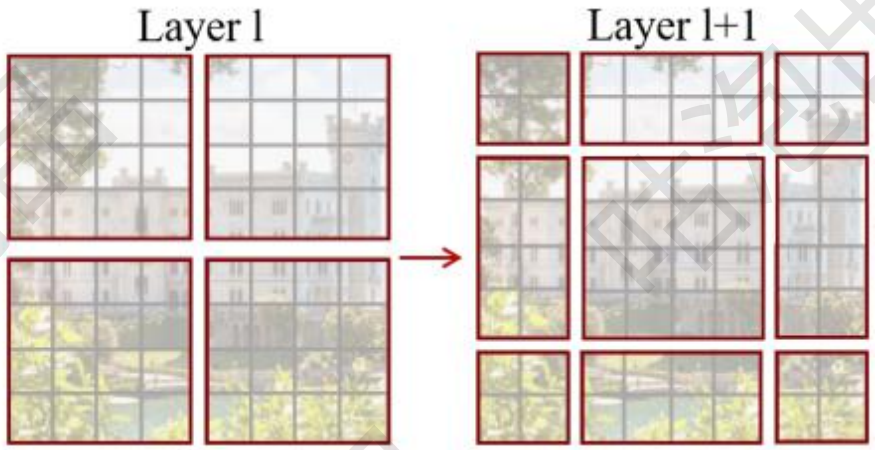
- 位移就是像素点挪一下位置：

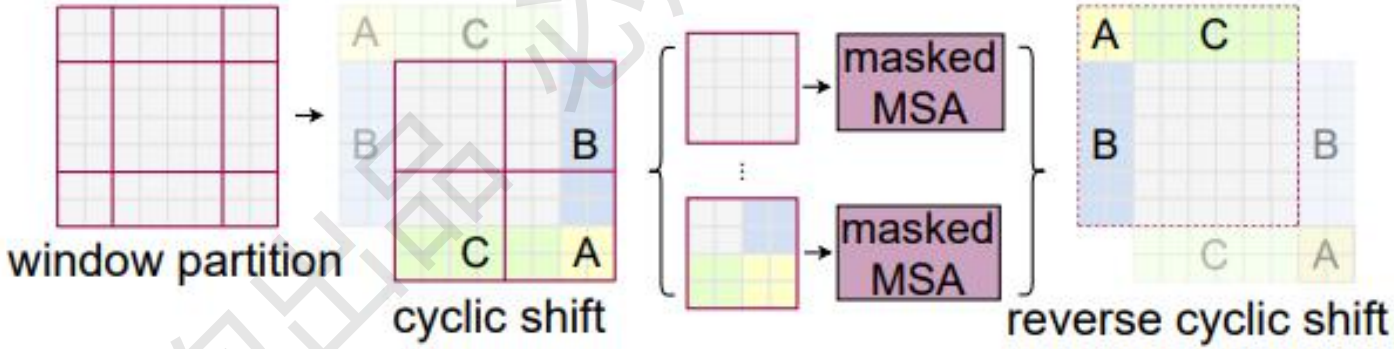
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

→

5	6	7	8
9	10	11	12
13	14	15	16
1	2	3	4

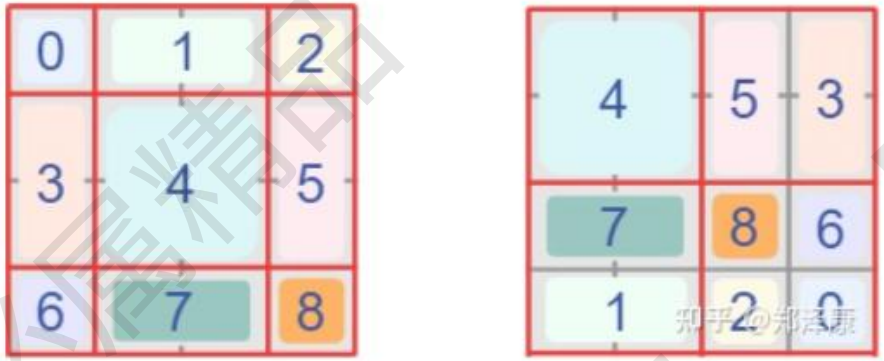
- 窗口移动后，还有点小问题，例如原来4个，现在9个了，计算量怎么解决呢?





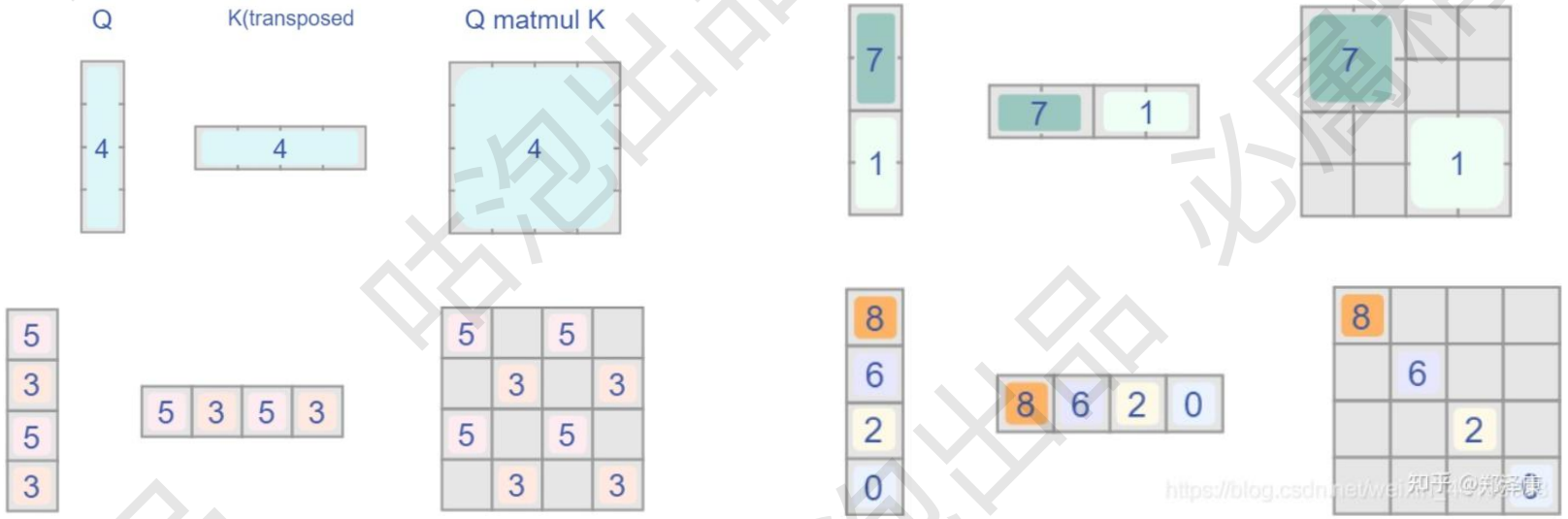
SwinTransformer

- 位移中的细节



- 首先得到新窗口，并对其做位移操作

- 在计算时，只需要计算自己窗口的，其他的都都是无关的

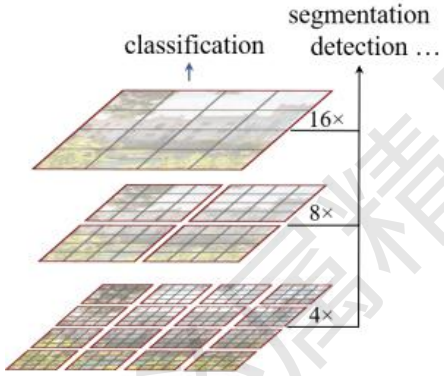


SwinTransformer

- 位移中的细节
  - 只需要设置好对应位置的mask，让其值为负无穷即可（softmax）
  - 输出结果同样为（56，56，96）
  - 不要忘记，计算完特征后需要对图像进行还原，也就是还原平移
  - 这俩组合就是SwinTransformer中的核心计算模块

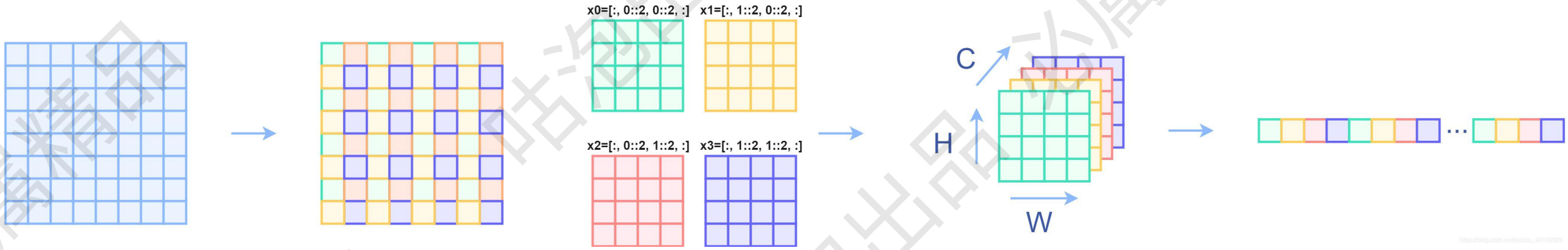
SwinTransformer

- PatchMerging



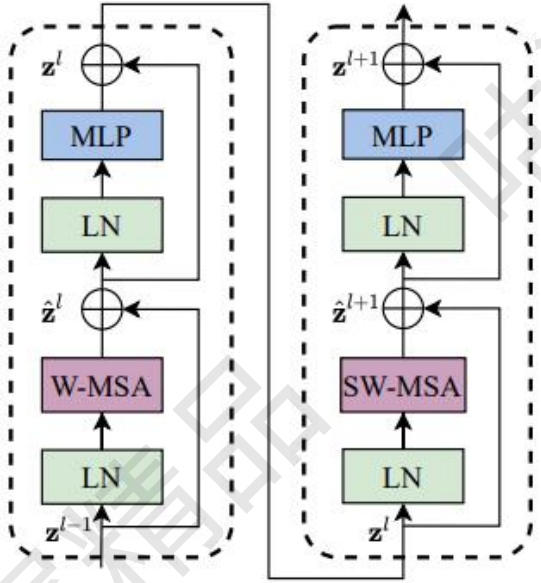
- 还记得咱们之前说的分层吧

- 也就是下采样操作，但是不同于池化，这个相当于间接的
（对H和W维度进行间隔采样后拼接在一起，得到H/2,W/2,C*4）



SwinTransformer

- 分层计算



- 一次下采样后（3136->784也就是56*56->28*28）
- 然后继续走这两兄弟，也就是各个stage的流程
- 最后根据任务来选择合适的head层即可（分类，分割，检测等）