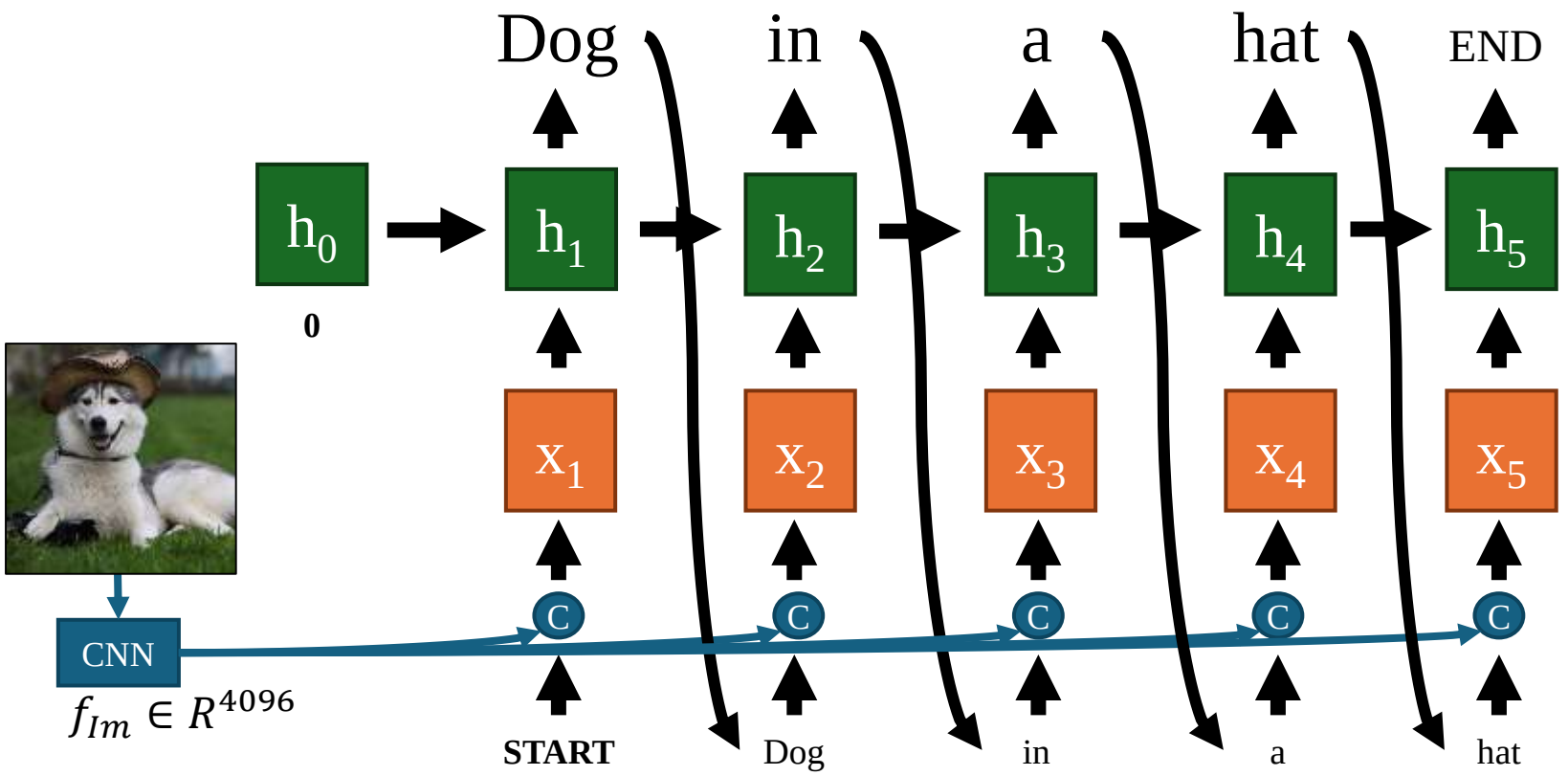
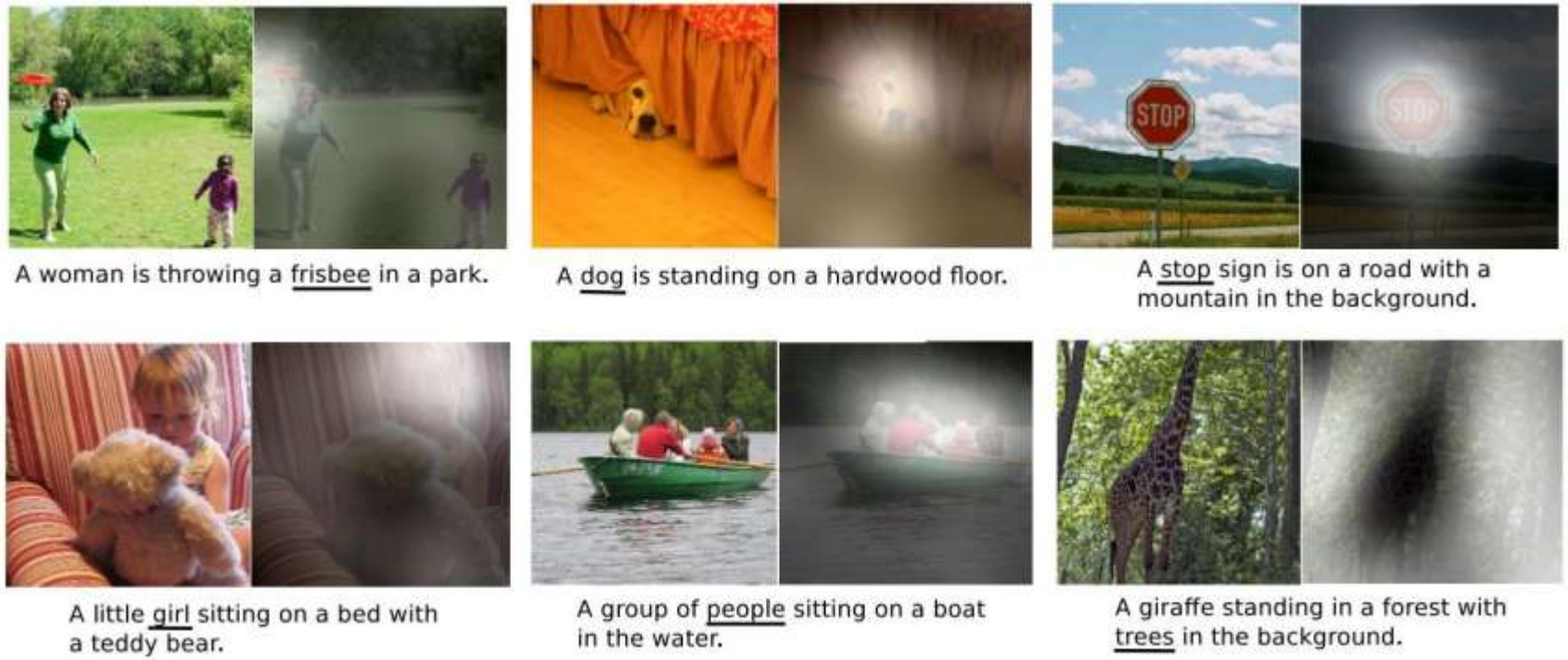
回顾：序列模型

Captioning：图像描述



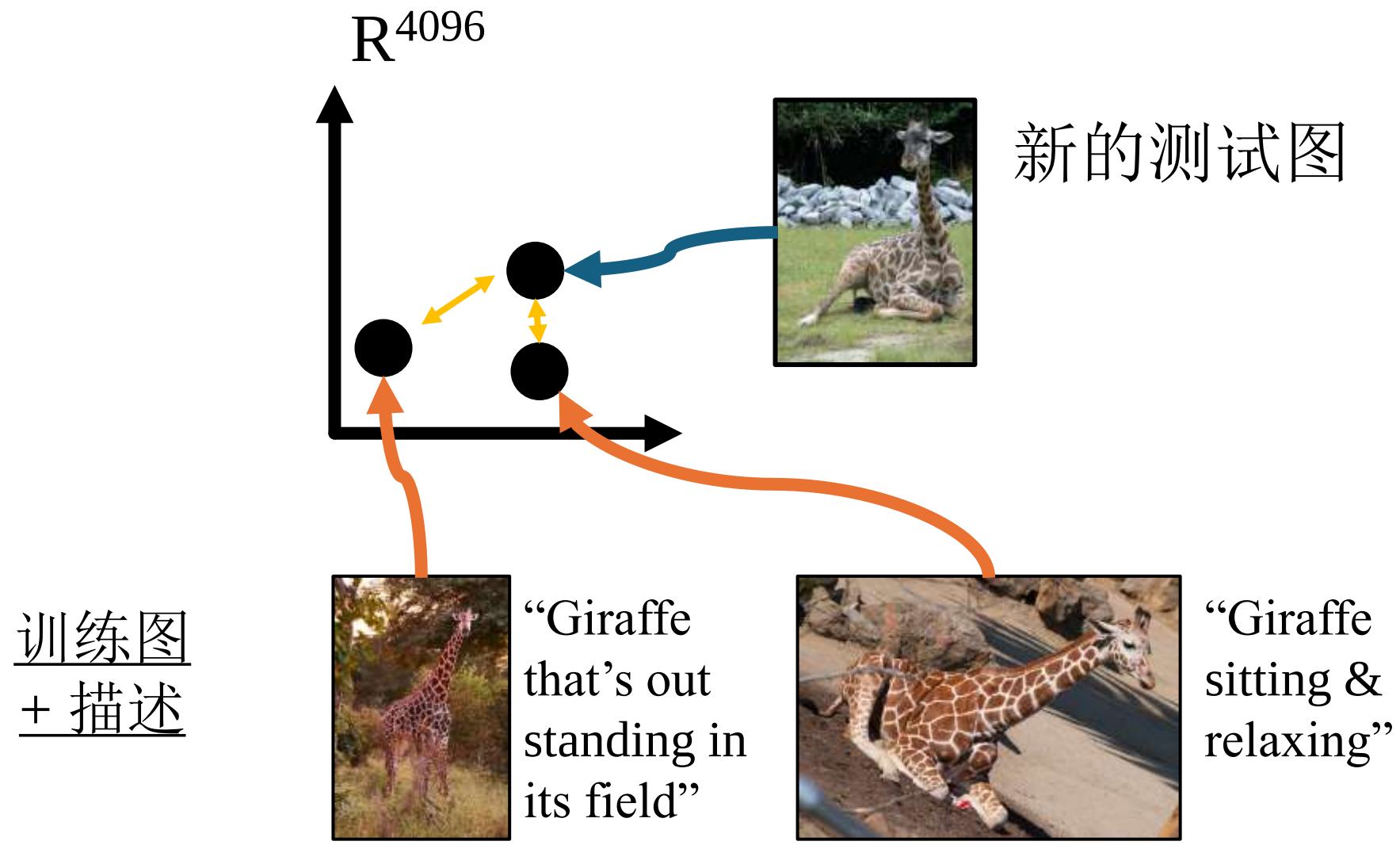
图像内容： Attention

注意力机制会让我们关注不同的空间区域

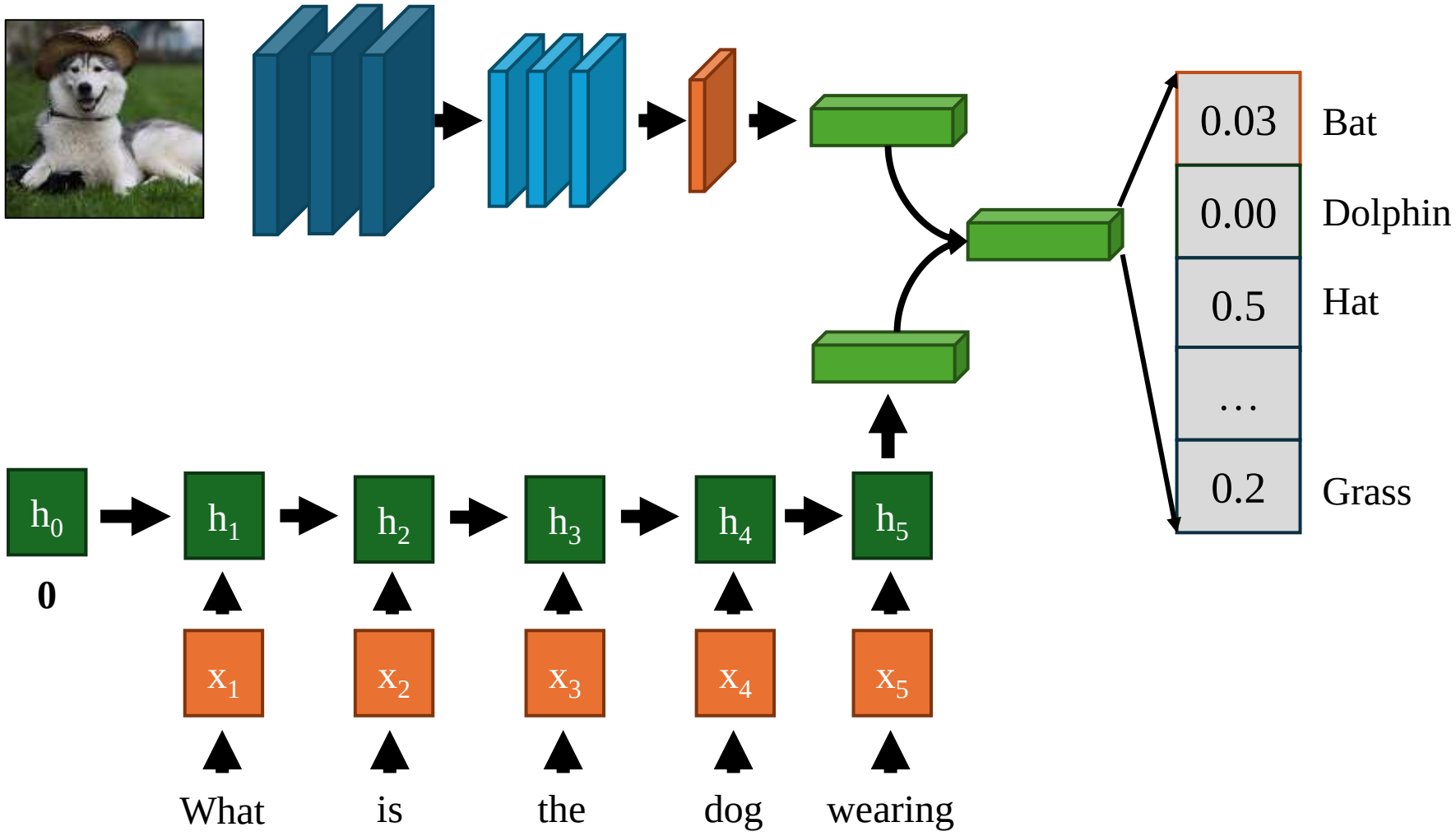


Xu et al, “Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention”, ICML 2015

另一种描述的方式：检索


R4096
新的测试图
训练图
+ 描述
“Giraffe that’s out standing in its field”
“Giraffe sitting & relaxing”

VQA模型


0.03
Bat
0.00
Dolphin
0.5
Hat
…
0.2
Grass
h0
h1
h2
h3
h4
h5
0
x1
x2
x3
x4
x5
What
is
the
dog
wearing

RNN的问题

任务：机器翻译

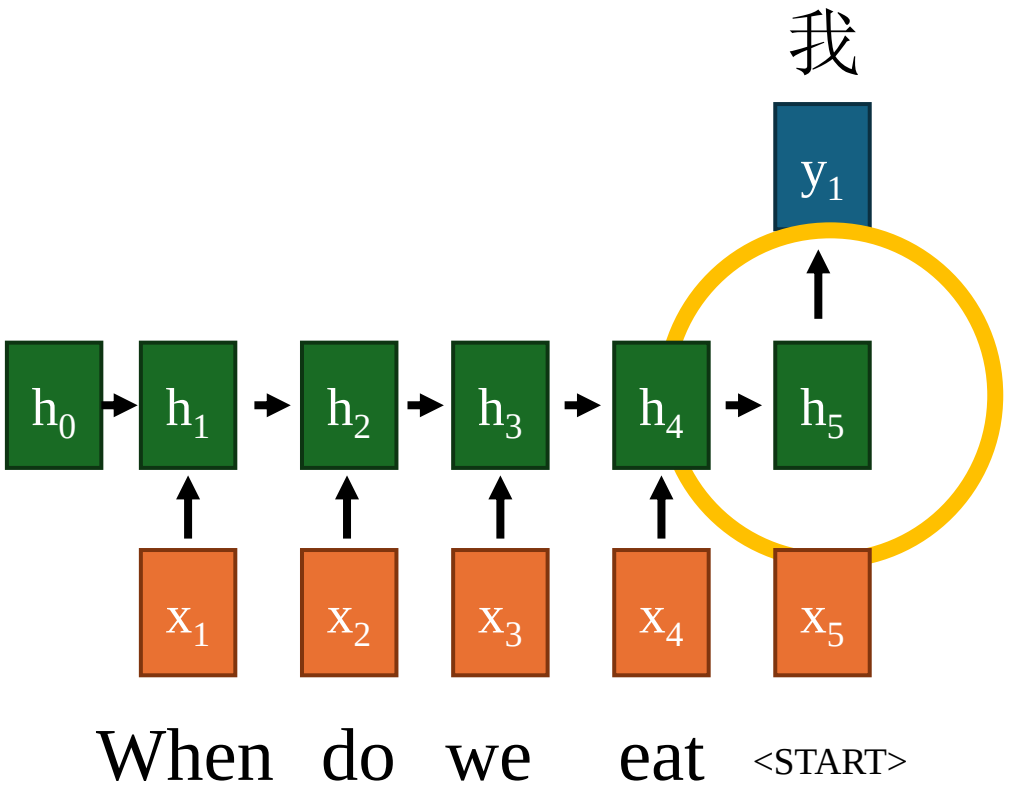


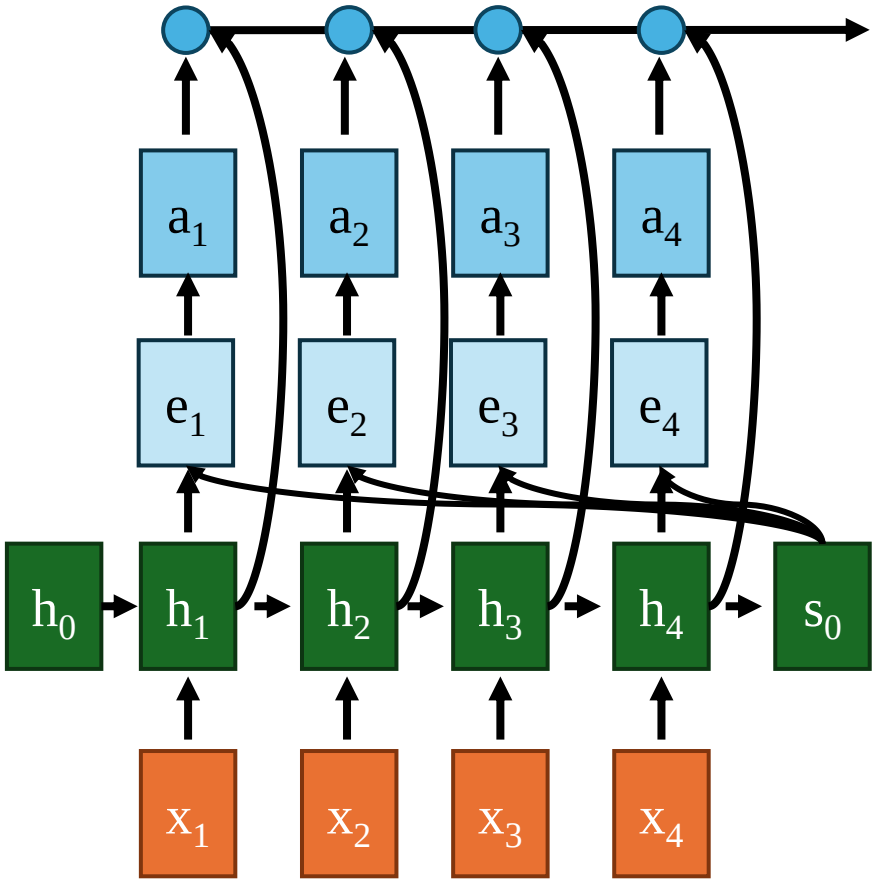
Diagram is remake of one from J. Johnson

我们需要记住每一个英文单词。

随着序列长度的增加，模型需要记住更多的信息，这使得保持长期依赖变得非常困难，这种现象通常称为**“长期依赖问题”**。

Transformer

Self-Attention



Output: $\sum_i a_i \boldsymbol{h_i}$

Attention: softmax over e_i
注意力权重

Alignment Score: $e_i = \boldsymbol{h}_i^T \boldsymbol{s}_0$
对齐得分

Diagram is remake of one from J. Johnson

Self-Attention

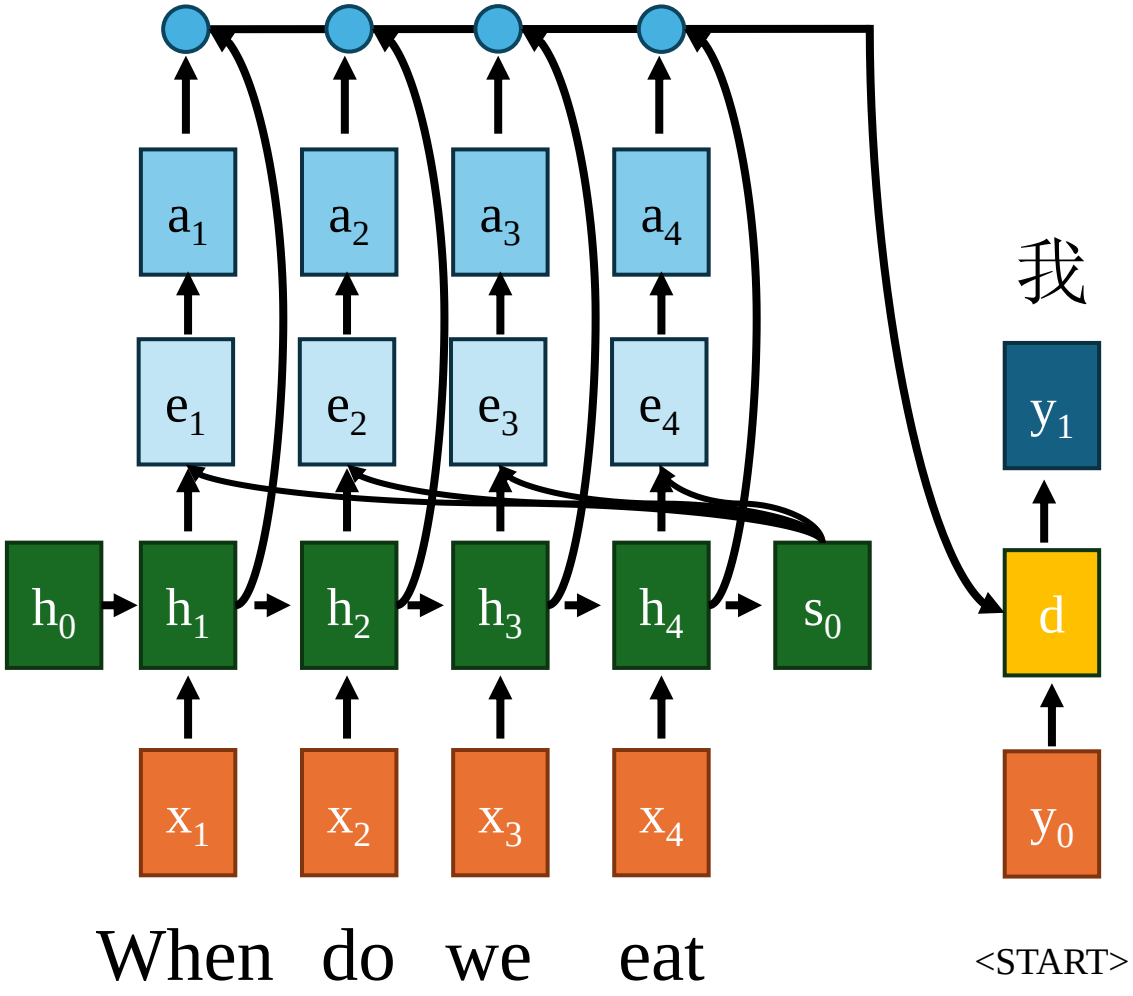
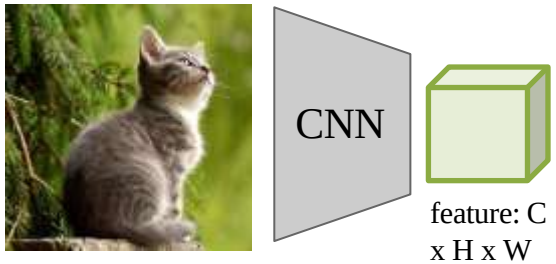


Diagram is remake of one from J. Johnson

与自注意力机制允许模型在每个时间步都考虑到整个输入序列，而不是仅仅依赖于前一个隐藏状态。这种方法提高了模型捕捉长距离依赖的能力

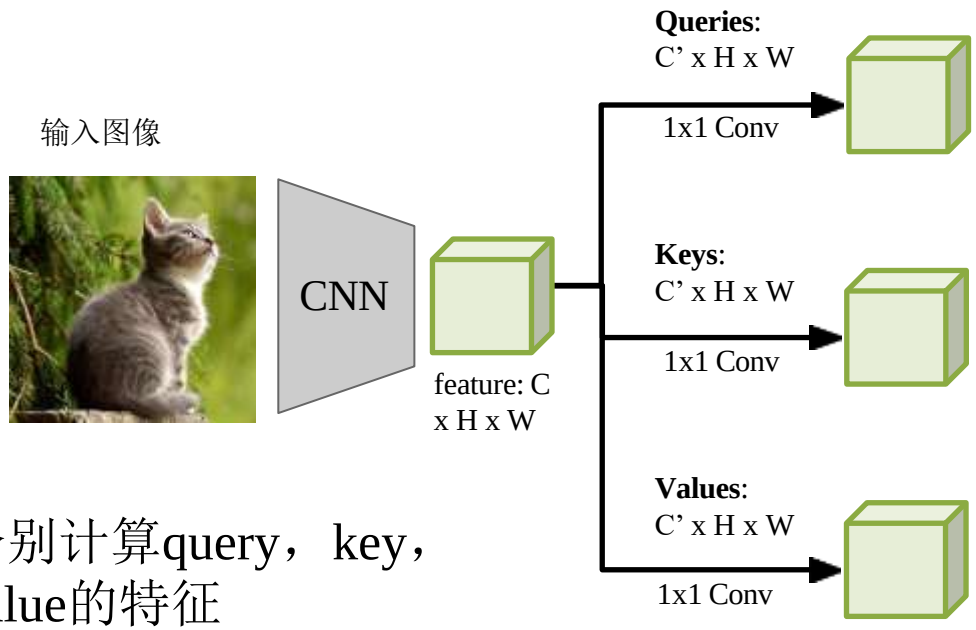
使用CNN的Self-Attention

输入图像



Zhang et al, “Self-Attention Generative Adversarial Networks”, ICML 2018

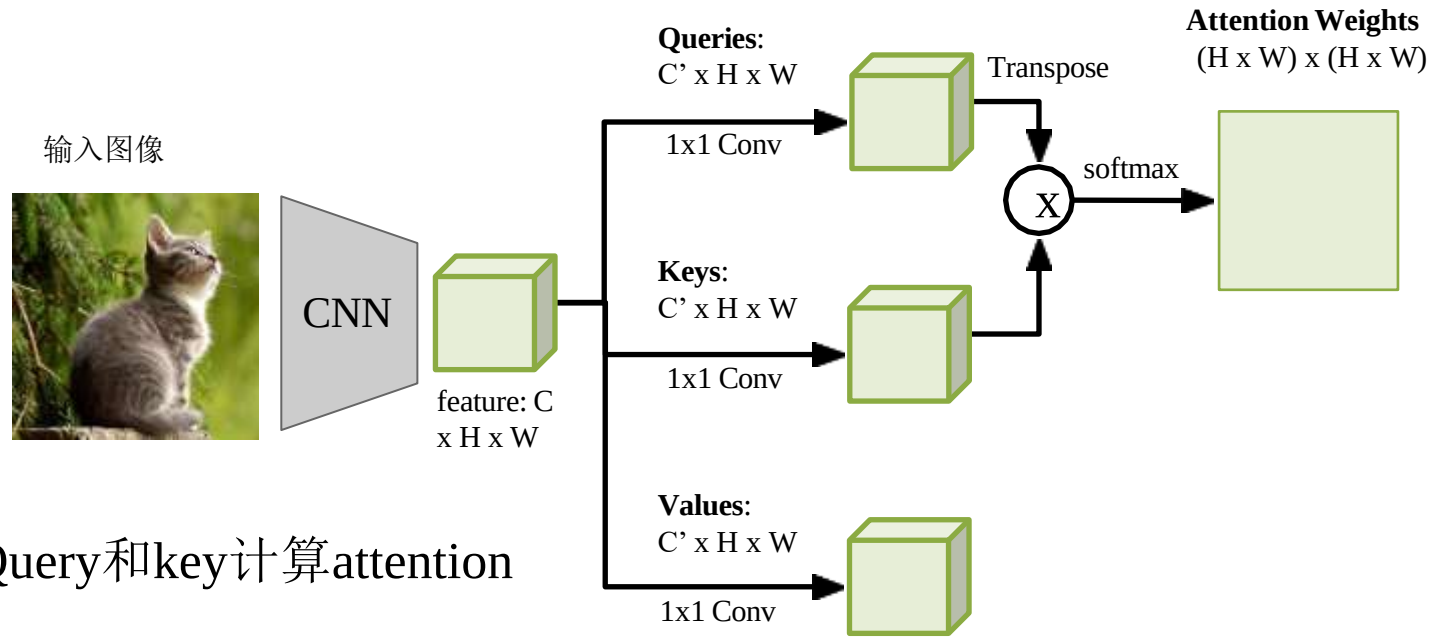
使用CNN的Self-Attention



分别计算query，key，value的特征

Zhang et al, "Self-Attention Generative Adversarial Networks", ICML 2018

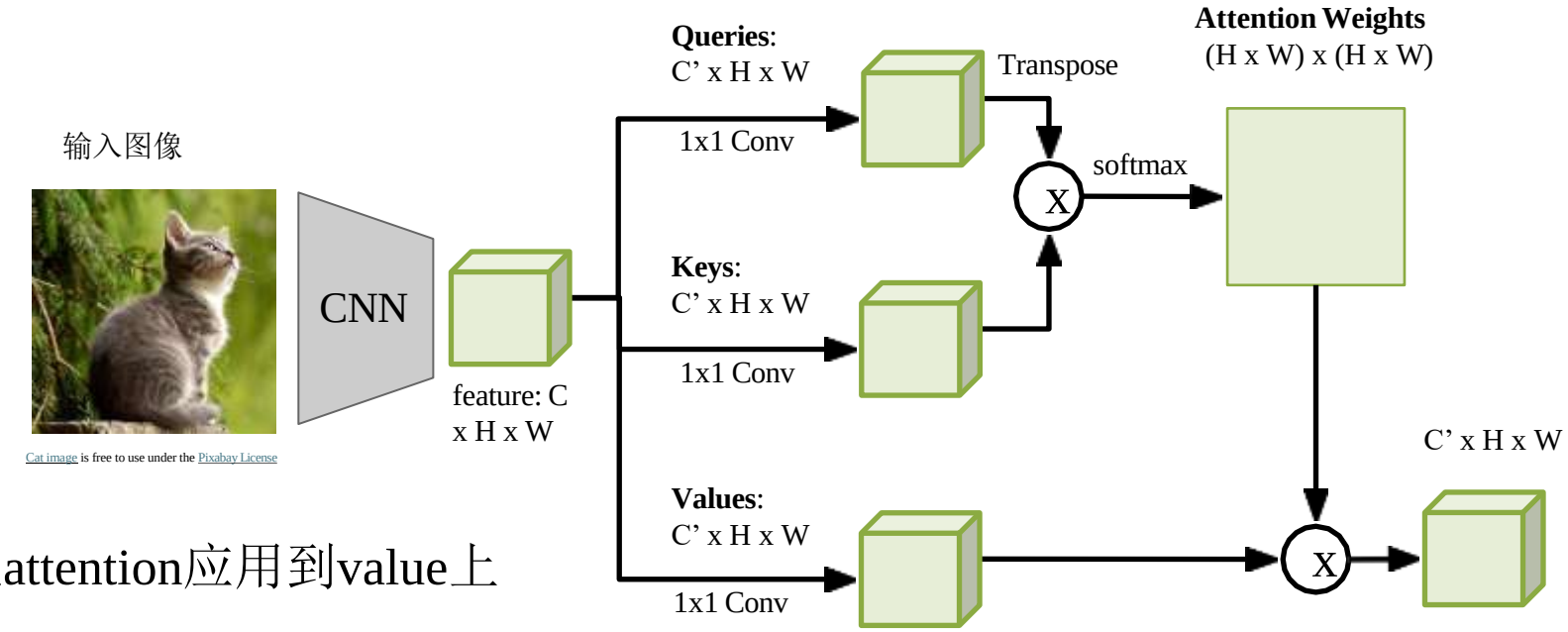
使用CNN的Self-Attention



用Query和key计算attention

Zhang et al, "Self-Attention Generative Adversarial Networks", ICML 2018

使用CNN的Self-Attention

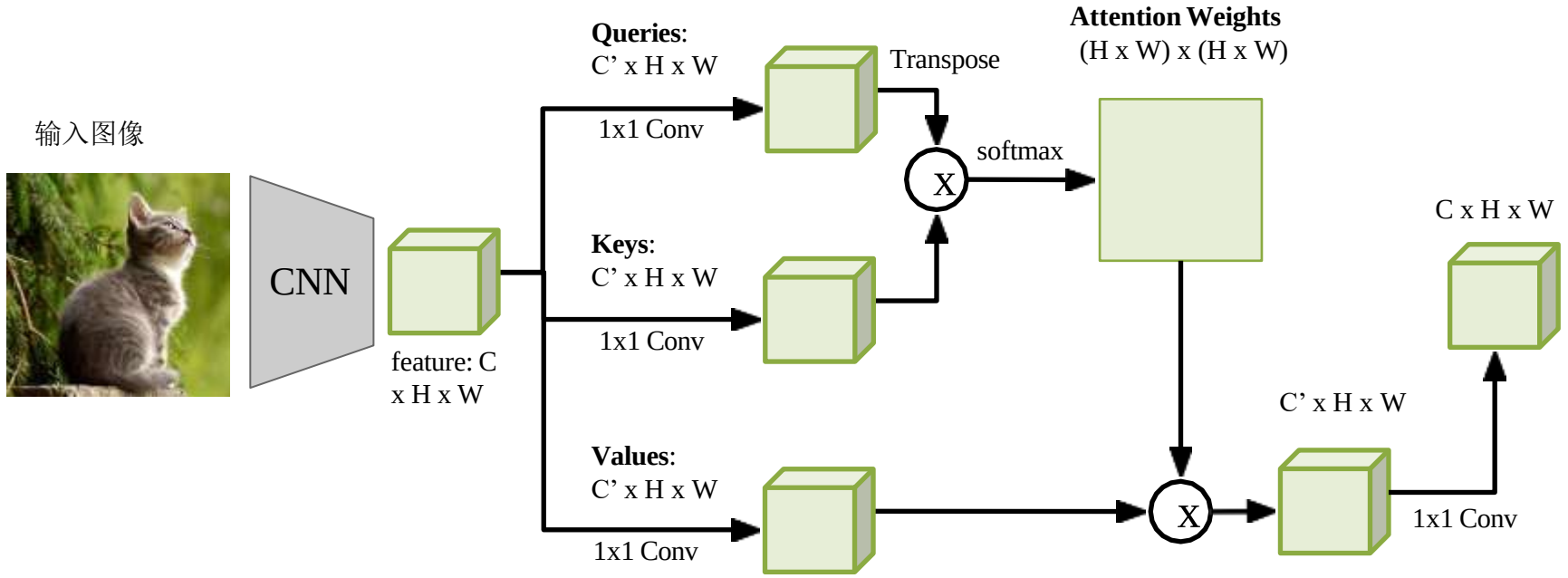


Cat image is free to use under the Pixabay License

把attention应用到value上

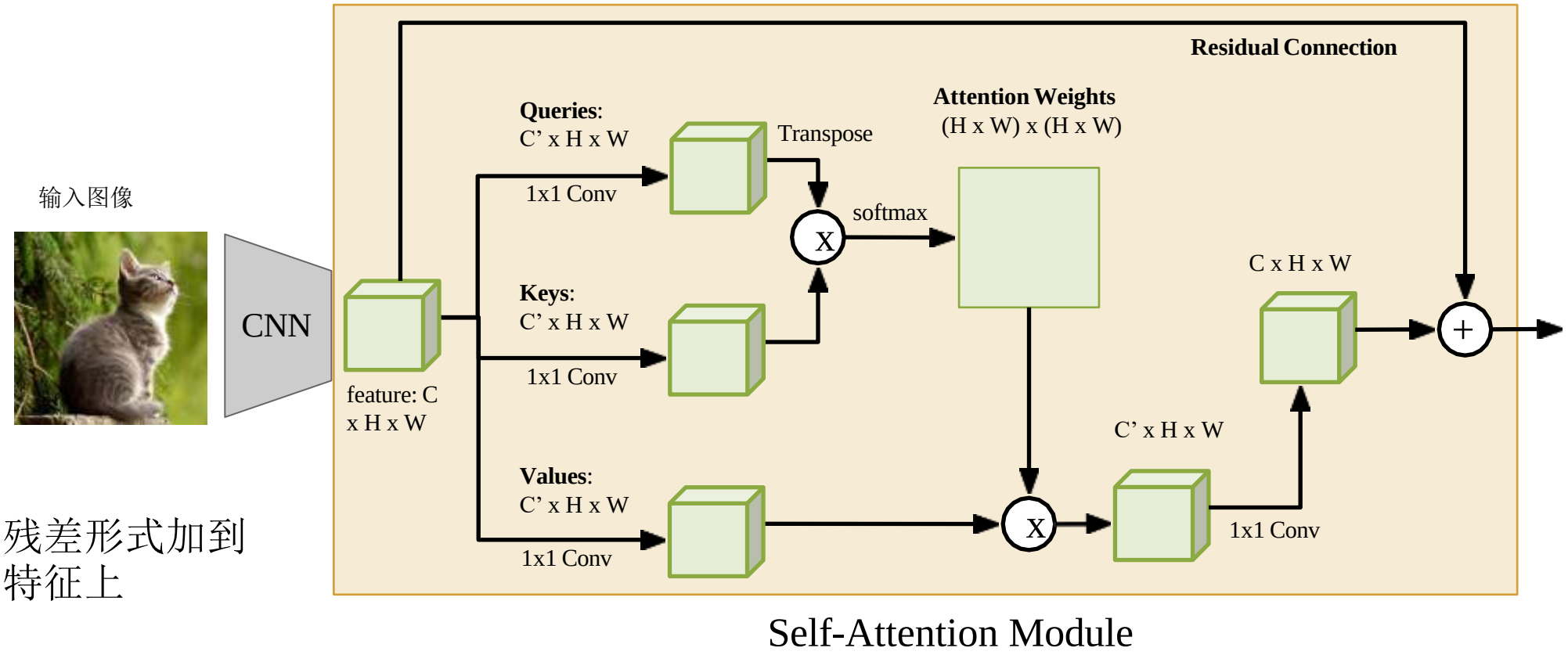
Zhang et al, "Self-Attention Generative Adversarial Networks", ICML 2018

使用CNN的Self-Attention



Zhang et al, "Self-Attention Generative Adversarial Networks", ICML 2018

使用CNN的Self-Attention

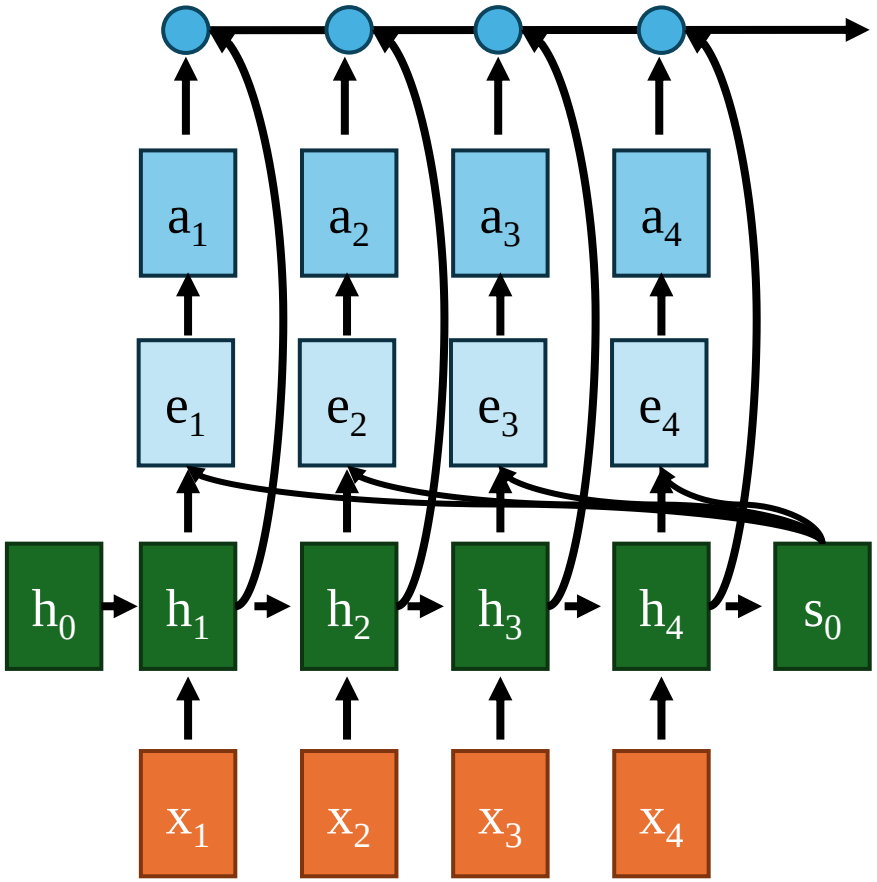


以残差形式加到原特征上

Self-Attention Module

Zhang et al, "Self-Attention Generative Adversarial Networks", ICML 2018

Transformers



Old Output: $\sum_i a_i \boldsymbol{h_i}$

New Output: $\sum_i a_i (\boldsymbol{V h_i})$

表示隐藏状态的线性变换，用于修改隐藏状态之前的加权和

Attention: softmax over e_i

Old Alignment: $e_i = \boldsymbol{h_i^T s_0}$

New: 通过$\boldsymbol{h_i^T K s_0}$ 得到 e_i

输入向量不做处理

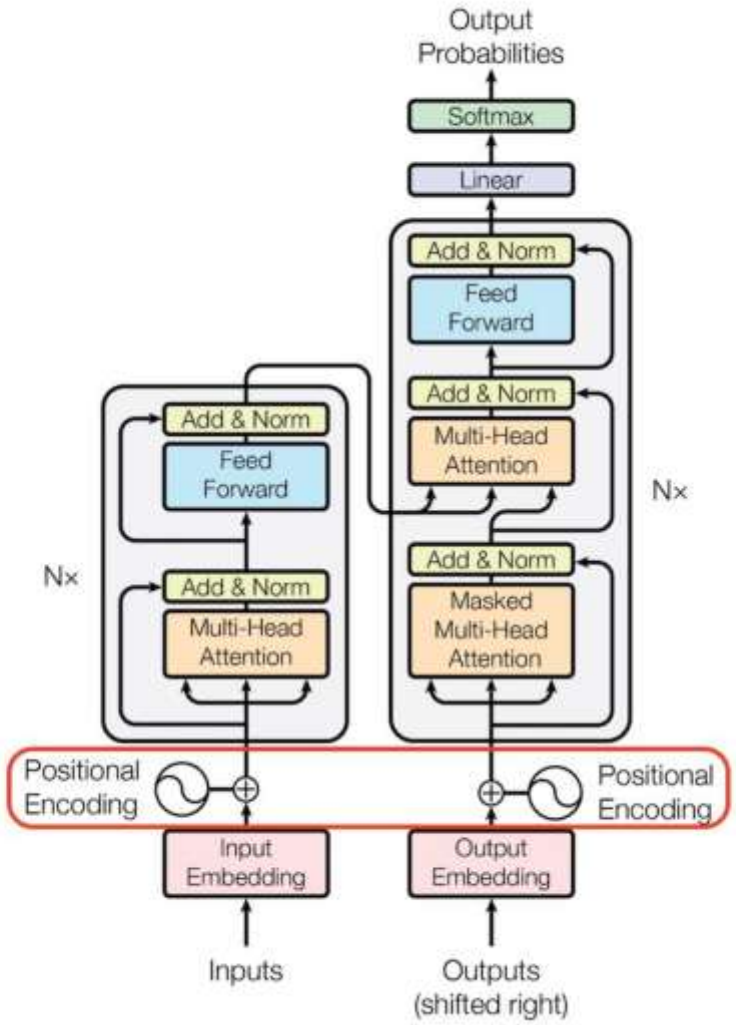
New: 对于 F-维 向量，用 $\sqrt{F}$ 调整尺度，防止梯度消失

Diagram is remake of one from J. Johnson

RNNs vs Transformer

- **RNNs**
- (+) 在较长的序列上表现较好.
- (-) 需要顺序序列
- (-) 需要顺序计算.

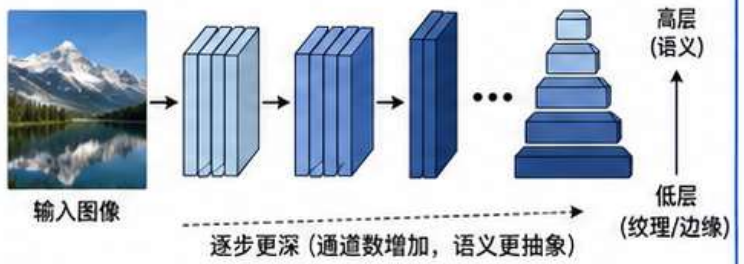
- **Transformer:**
- (+) 同样在较长的序列上表现较好.
- (+) 可以在没有顺序的序列上应用.
- (+) 并行计算.
- (-) 需要大量内存. (但是可以用GPU加速)



为什么 Transformer 主导的网络通常更宽、参数更多?

核心原因：CNN 主要编码局部特征；Transformer 需要在更高维空间中承载多头注意力、MLP / MoE 与更丰富的相关性组合。

1. CNN：更依赖深度累积

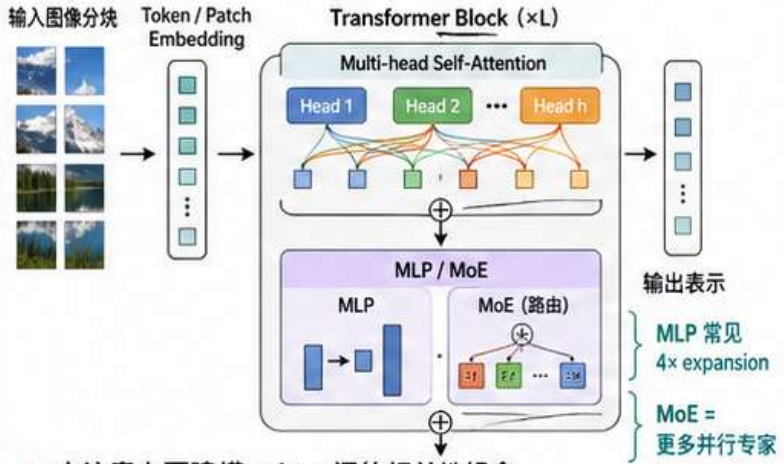


- 局部卷积核，天然局部归纳偏置
- 主要建模边缘、纹理、局部模式
- 感受野通常依靠层数逐步扩大
- 组合能力更多由“加深网络”获得
- 因此常见形态：较窄，但可以较深

局部特征表达，不一定需要很高维度。

CNN:
靠深度累积
局部模式

2. Transformer：更依赖宽度承载关系建模

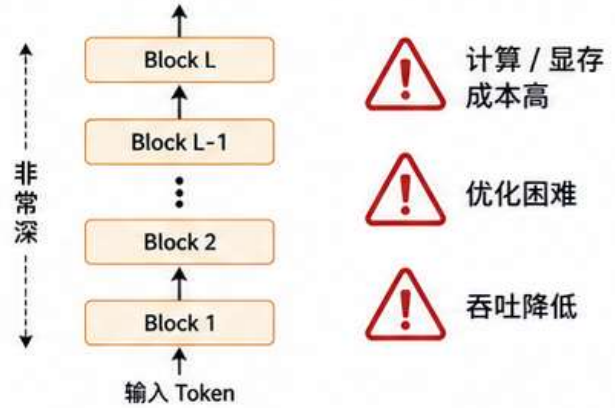


- 自注意力要建模 token 间的相关性组合
- Multi-head 让不同头关注不同关系模式
- 更大的 hidden width 提供更丰富的子空间
- MLP / MoE 进一步把参数量推高
- 因此常见形态：较宽、更大

宽度越大，越能容纳更多注意力组合与路由能力。

Transformer:
靠宽度承载
多样关系

3. 为什么通常不会一味更深?



- 注意力的计算、显存与通信成本更高
- 宽模型已经具备较强表达能力
- 继续加深的边际收益常低于加宽
- 过深设计更难优化，训练吞吐也更差
- 所以主流设计常是：又宽又浅，但总体很大

深度受成本与优化约束，宽度更直接提升容量。

典型例子（示意）

	ResNet-50	ViT-L/16
主导算子	Conv	Self-Attention
深度	50 层	24 Blocks
宽度 / 表征	中等通道宽度	Hidden size 1024
多头 / 专家	无	16 heads，MLP 4096
参数量	约 25M	约 307M
设计倾向	较深、较窄	较宽、较浅、总体更大

结论： CNN 主要靠**深度**逐层组合局部特征；Transformer 主要靠**宽度**承载多头注意力、MLP / MoE 与高维关系建模，因此更容易走向**“更宽、参数更多”**的主导范式。

Large-scale Language Models 大语言模型

- 在极大的尺度上训练的语言模型
- GPT: Generative Pretrained Transformers
- GPT-3:
  - 410 billion tokens 的训练数据
  - 17 billion 的参数量
- “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜” Bender et al. https://dl.acm.org/doi/pdf/10.1145/3442188.3445922

Transformers

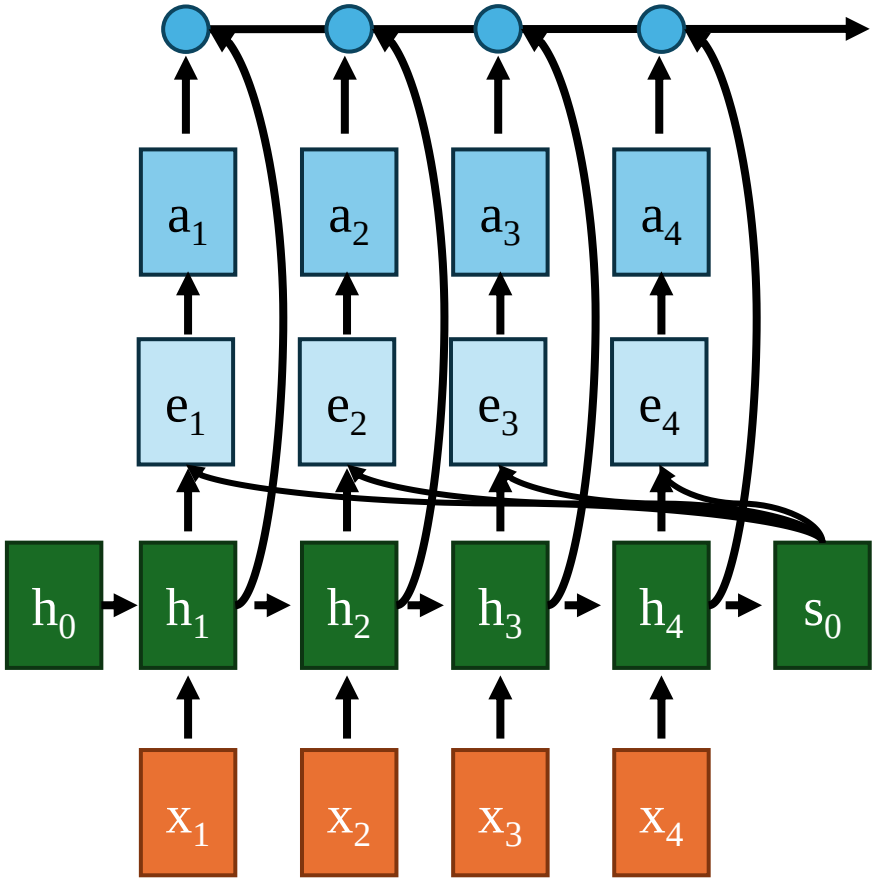


Diagram is remake of one from J. Johnson

Alignment score 可以体现每个token的位置吗?

Alignment Score: $e_i = \boldsymbol{h}_i^T \boldsymbol{s}_0$

Solution: 把位置信息加入到计算alignment score的过程中

Positional Encodings 位置编码

位置编码（Positional Encoding），其目的是给模型提供序列中各个元素位置的信息。f(p, 2i)和f(p, 2i + 1)分别定义了每个位置p的编码，其中i是维度索引，而d是编码的维度总数。对于偶数维度（使用2i），位置p的编码是通过正弦函数来计算的；对于奇数维度（使用2i + 1），使用余弦函数。

$$f(p,2i) = \sin\left(\frac{p}{10000^{2i/d}}\right) \quad f(p,2i+1) = \cos\left(\frac{p}{10000^{2i/d}}\right)$$

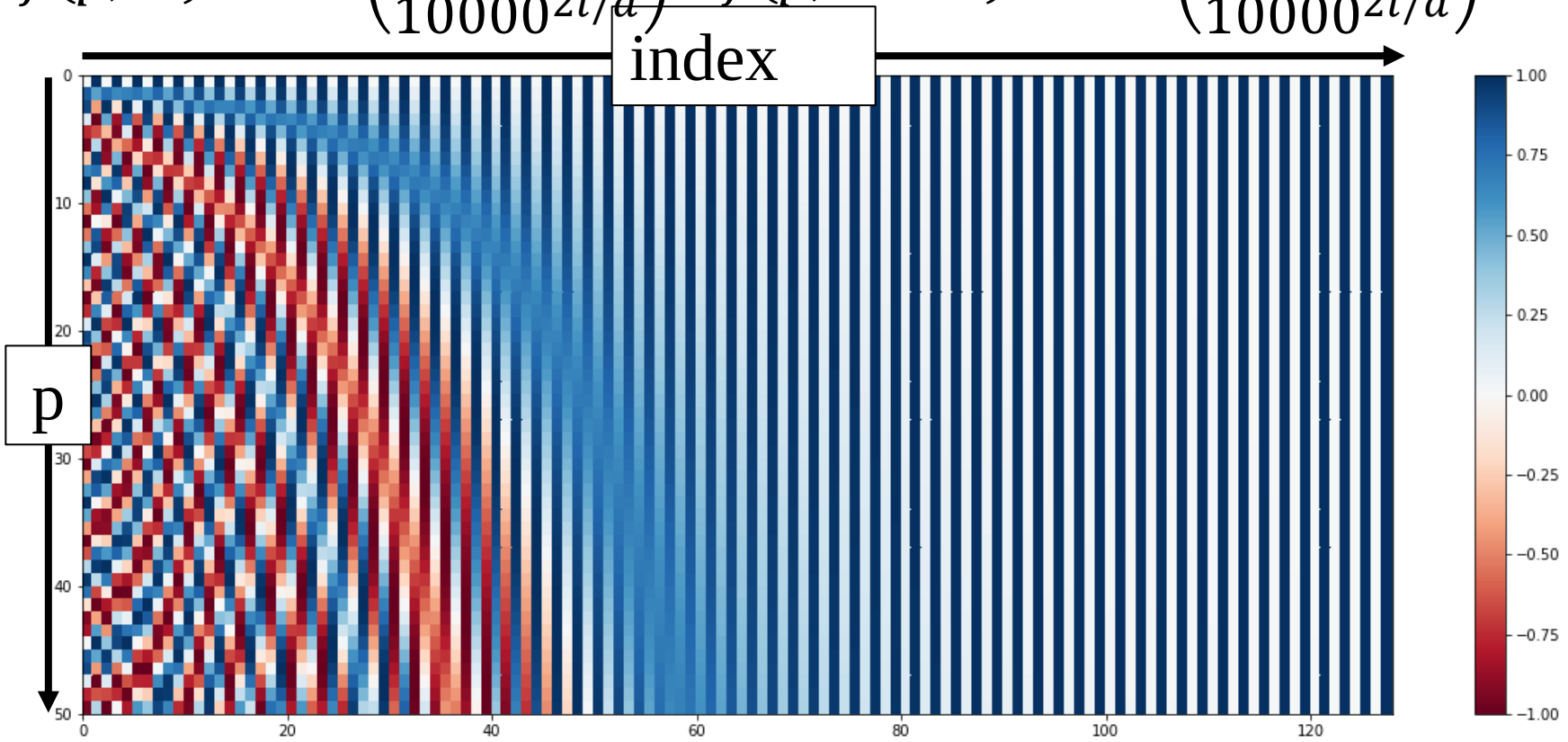


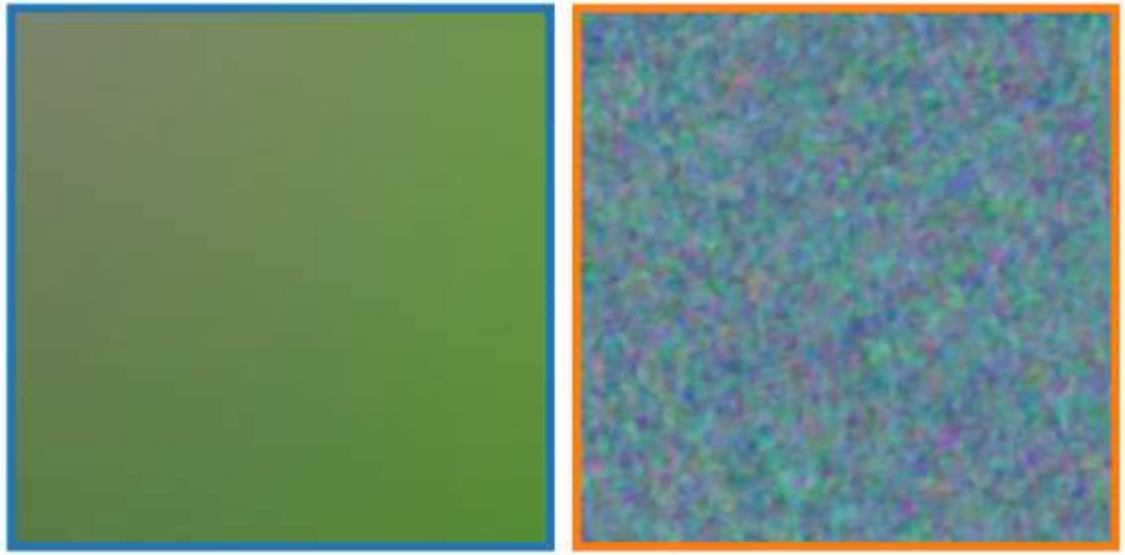
Diagram: Amirhoseein Kazemjejad, https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

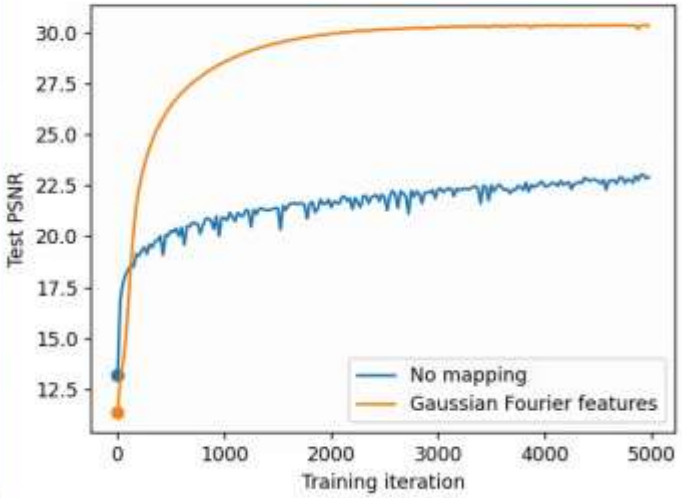
Positional Encodings 与 图像

需要学习从(x,y) 到 (r,g,b) 的映射

Learned on (x,y)

w/positional encodings)





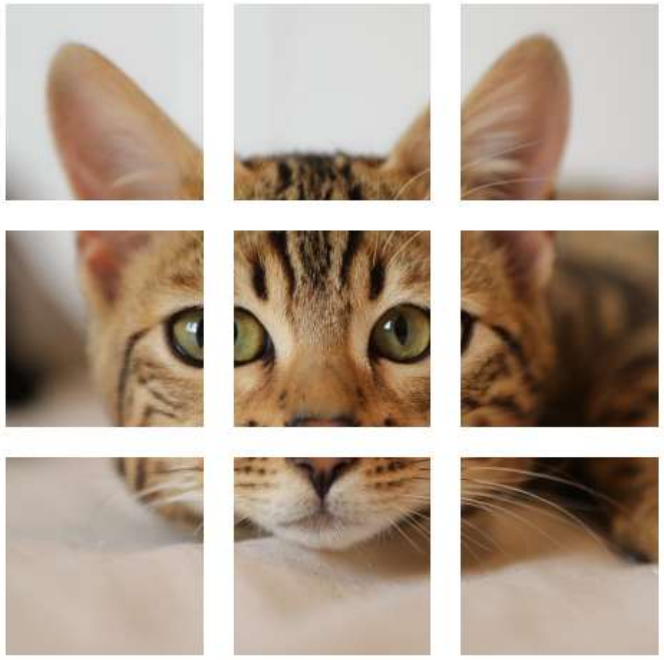
Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains, Tancik et al. NeurIPS 2020. See also Implicit Neural Representations with Periodic Activation Functions, Sitzmann et al. NeurIPS 2020.

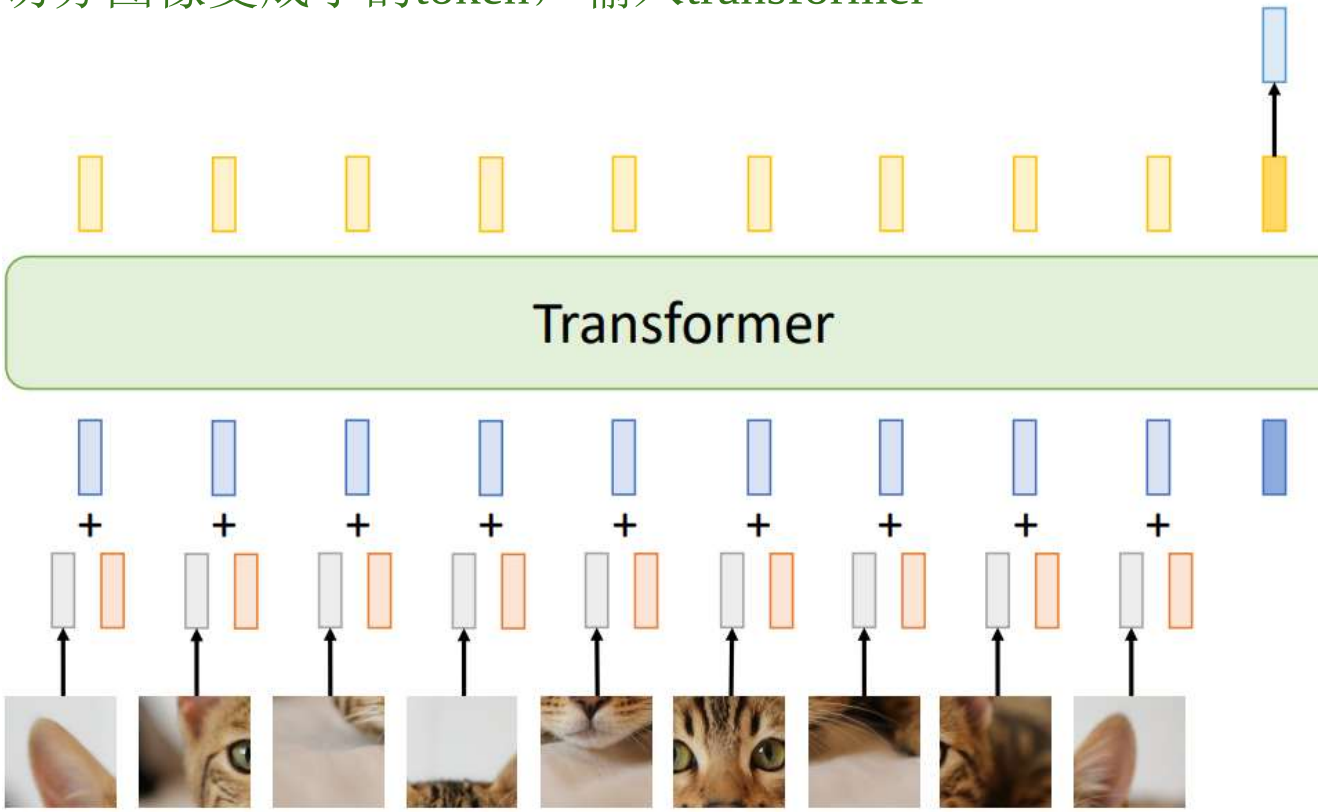
怎样把Transformer用到图像上?

把图像切成小的局部，然后放到Transformer里

Vision Transformers

切分图像变成小的token，输入transformer

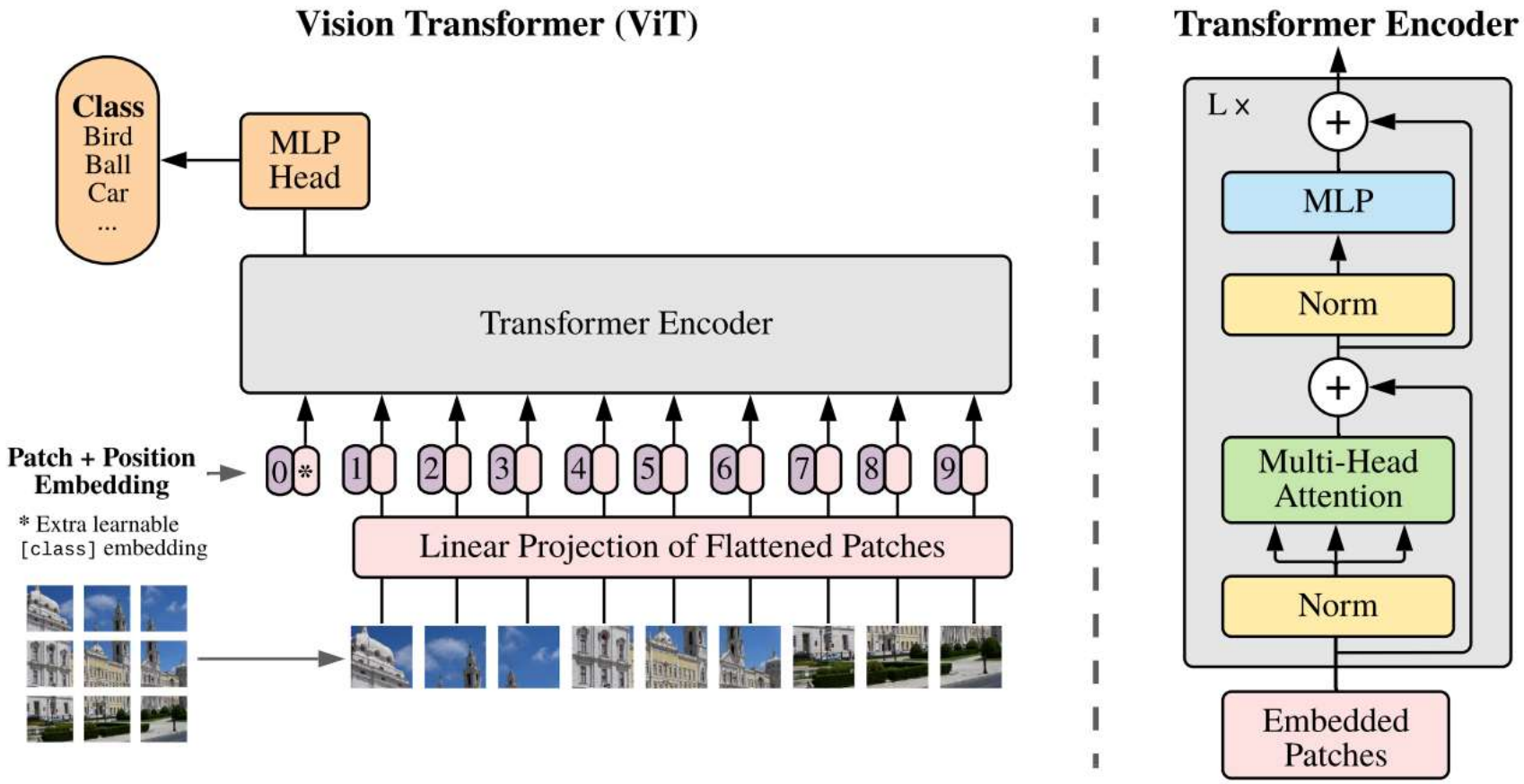




Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ArXiv 2020 Colab link to an implementation of vision transformers

Vision Transformer （ViT）

核心思想: 把图像转换为image tokens



An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. Dosovitskiy et al. ICLR 2021.

ViTs vs. ResNets

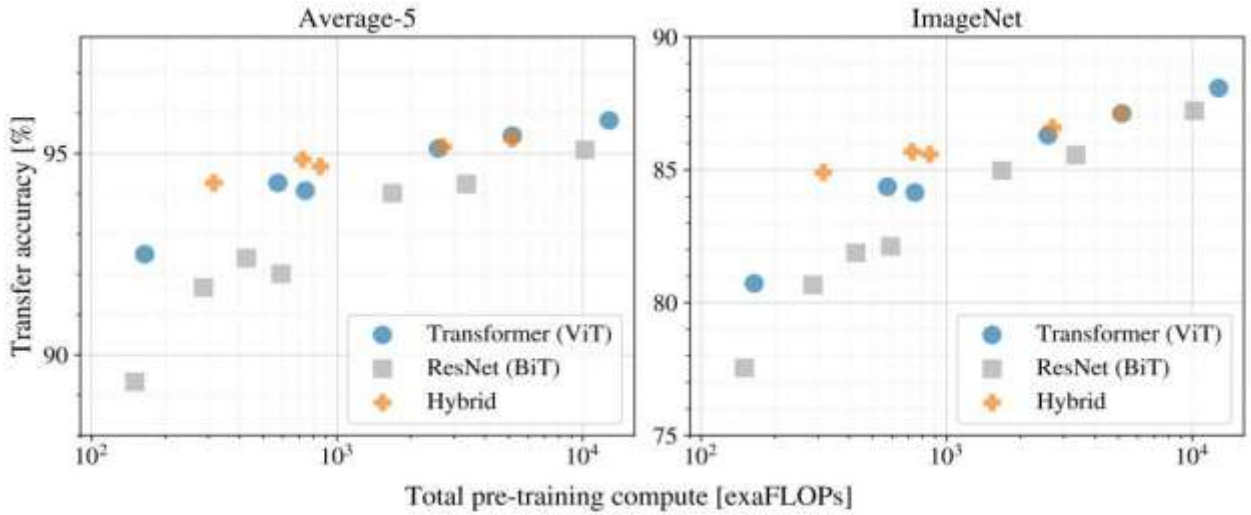
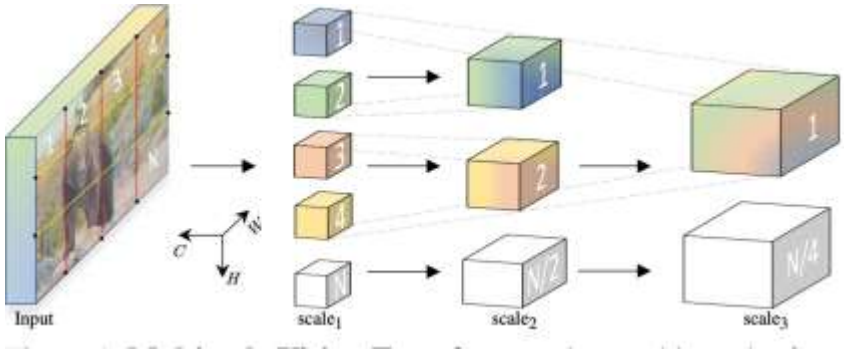


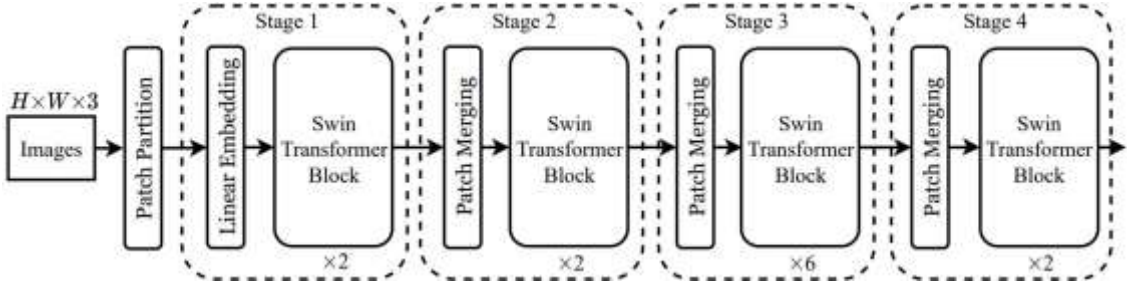
Figure 5: Performance versus cost for different architectures: Vision Transformers, ResNets, and hybrids. Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanishes for larger models.

Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, ArXiv 2020 Colab link to an implementation of vision transformers

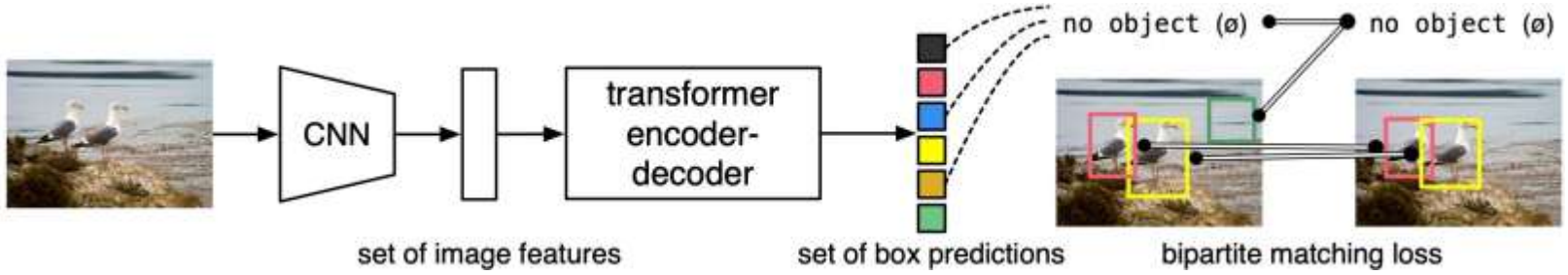
Vision Transformers (ViT)



Fan et al, “Multiscale Vision Transformers”, ICCV 2021

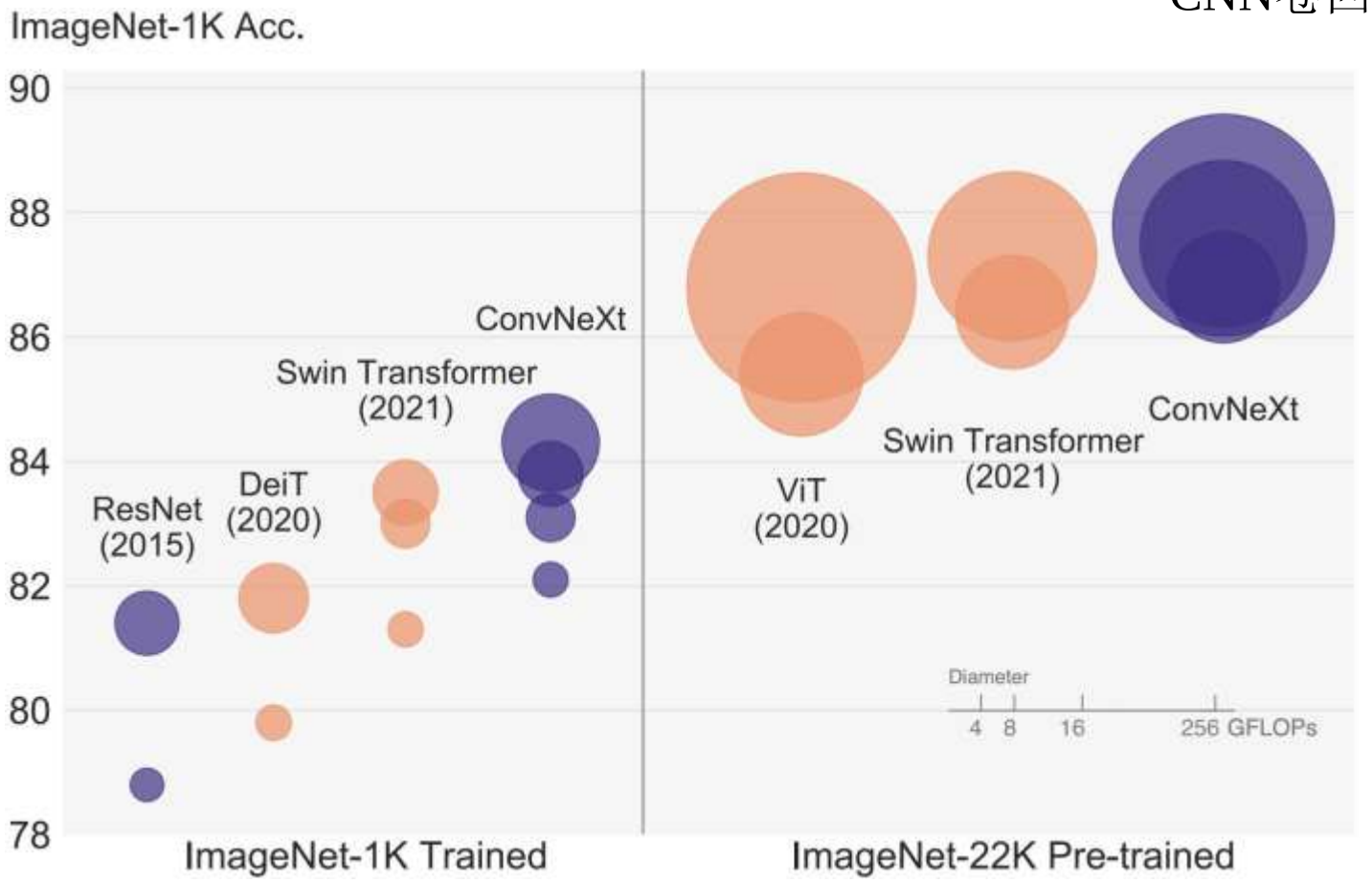


Liu et al, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”, CVPR 2021



Carion et al, “End-to-End object detection with Transformers”, ECCV 2020

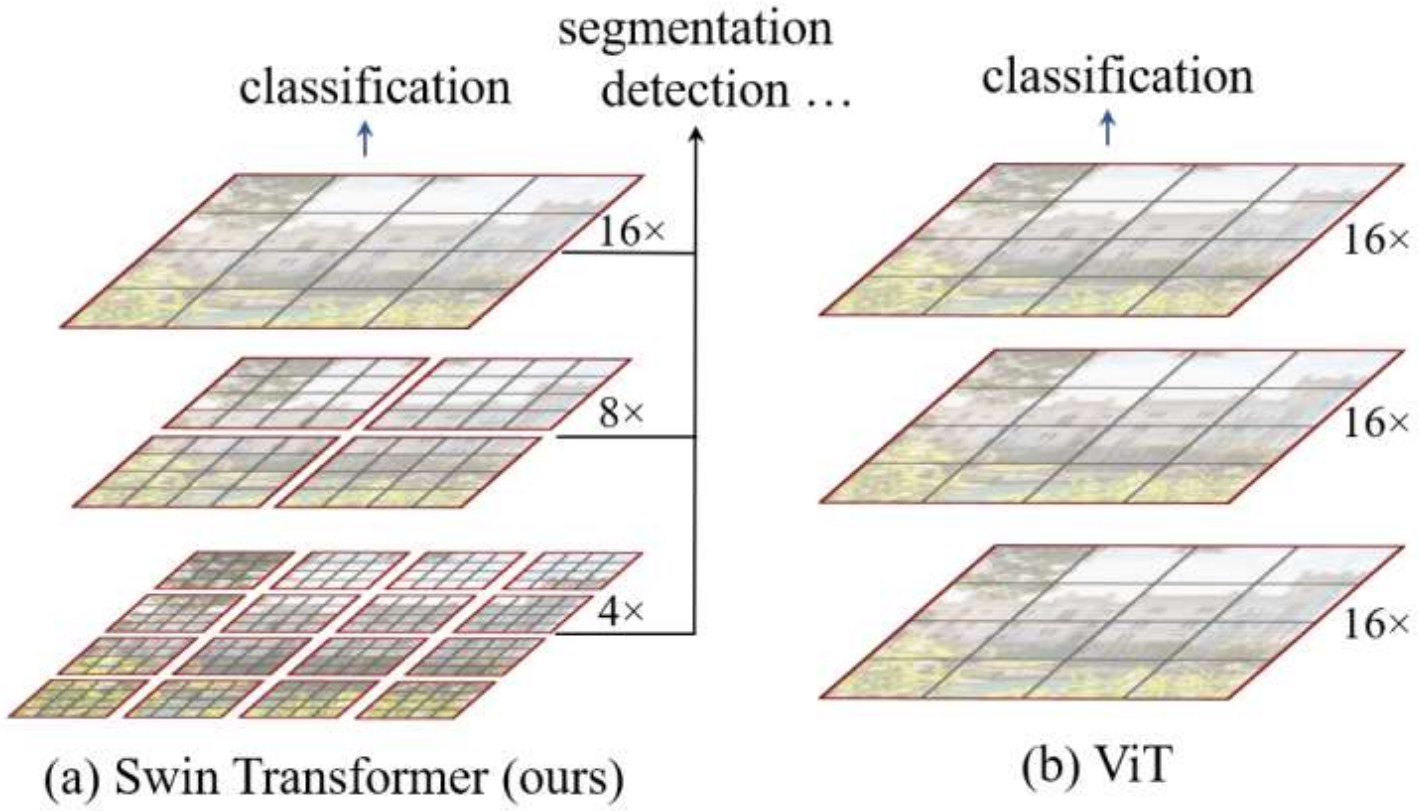
CNN卷回来了!



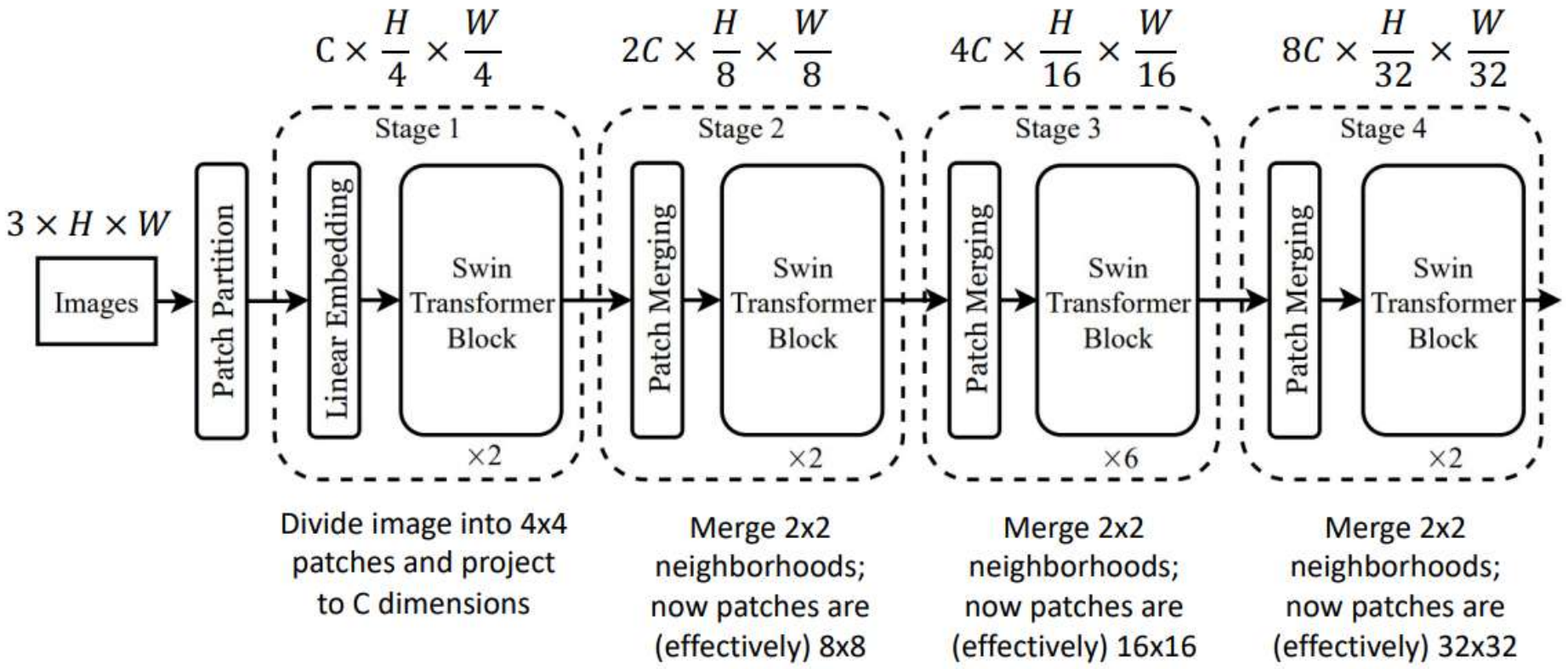
A ConvNet for the 2020s. Liu et al. CVPR 2022

Swin Transformers

分阶段在不同尺度内部做transformer

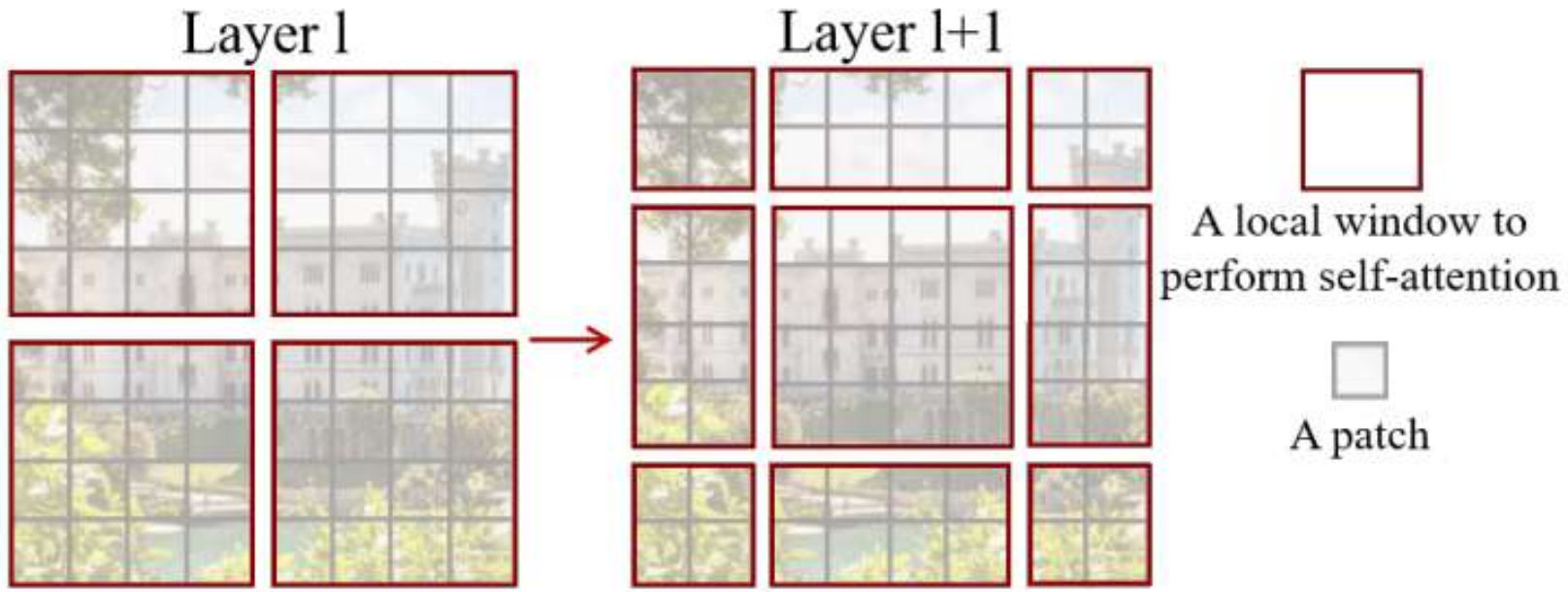


Swin Transformers



Swin Transformers

Shift window避免信息只在固定窗口传递



视频任务

视频分类：对比图像分类

Images: Recognize **objects**

Dog
Cat
Fish
Truck

Videos: Recognize **actions**

Swimming
Running
Jumping
Eating
Standing

Running video is in the public domain

问题: 视频太大了

- 视频每秒大约有30帧（~30fps）

Input video:
T x 3 x H x W

Size of uncompressed video (3 bytes per pixel):

SD (640 x 480): **~1.5 GB per minute**
HD (1920 x 1080): ~**10 GB per minute**

问题: 视频太大了

- 视频每秒大约有30帧（~30fps）

Input video:
T x 3 x H x W

Size of uncompressed video (3 bytes per pixel):

SD (640 x 480): **~1.5 GB per minute**
HD (1920 x 1080): **~10 GB per minute**

解决方案 短视频 低分辨率**:**
low fps and low spatial resolution
e.g. T = 16, H=W=112
(3.2 seconds at 5 fps, 588 KB)

在视频上训练CNN

原视频：很长，很高的FPS

在视频上训练CNN

原视频：很长，很高的FPS

训练：短视频、低FPS

在视频上训练CNN

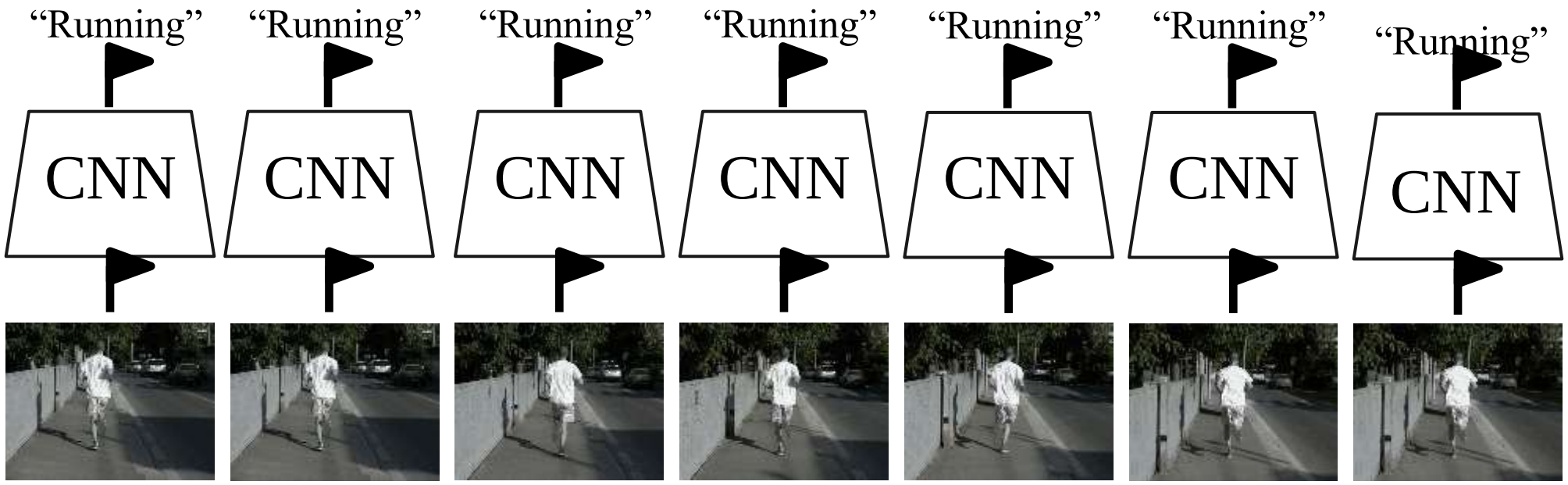
原视频：很长，很高的FPS

训练：短视频、低FPS

测试：在不同的短视频窗口预测，平均重叠部分

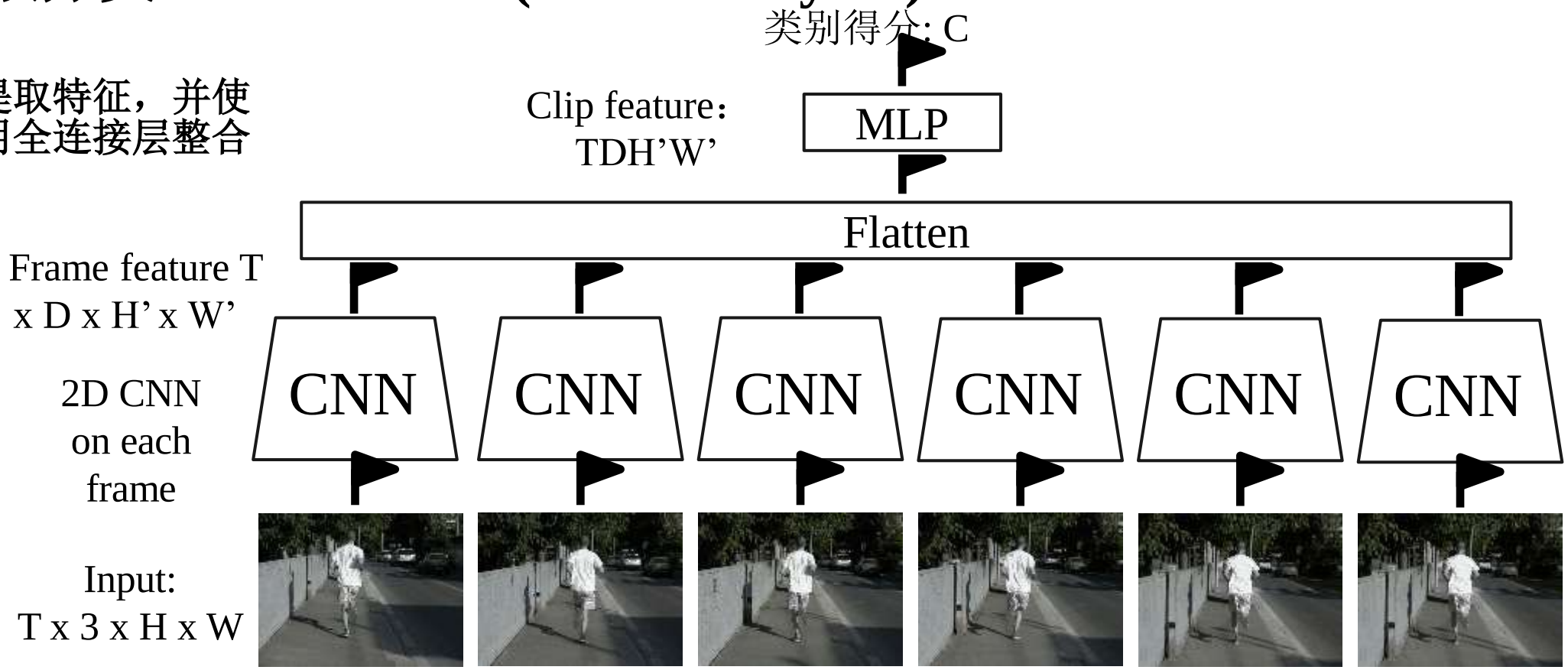
视频分类: 单帧CNN

- 想法: 训练2D CNN，然后取均值



视频分类: Late Fusion (with FC layers)

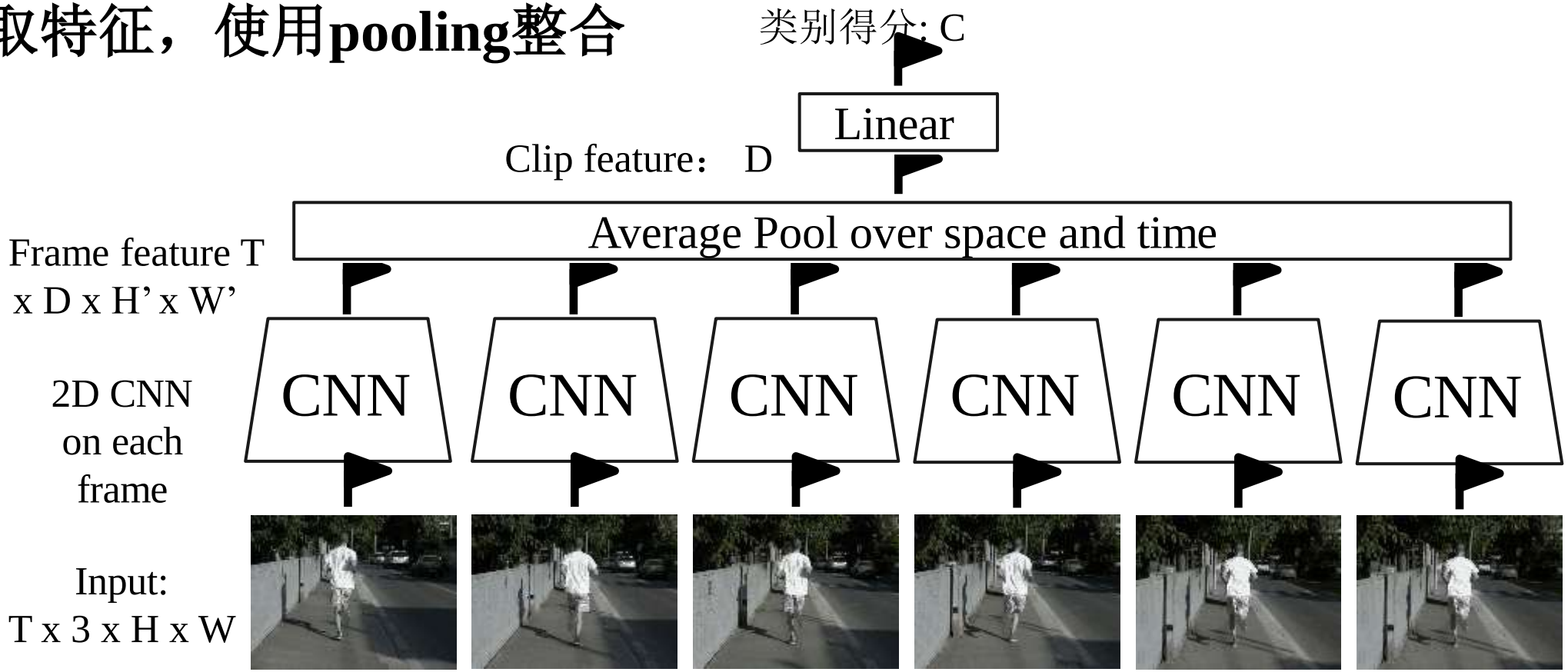
- 提取特征，并使用全连接层整合



Karpathy et al, "Large-scale 视频分类 with 卷积 Neural Networks", CVPR 2014

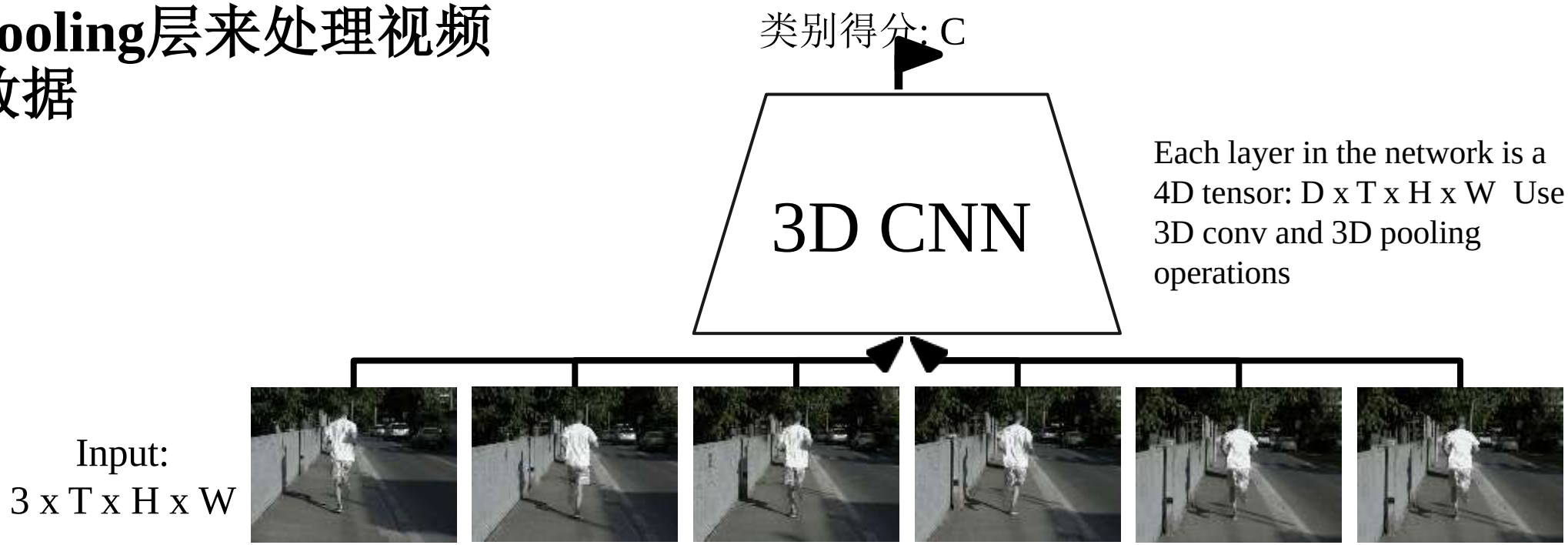
视频分类: Late Fusion (with pooling)

- 提取特征，使用pooling整合



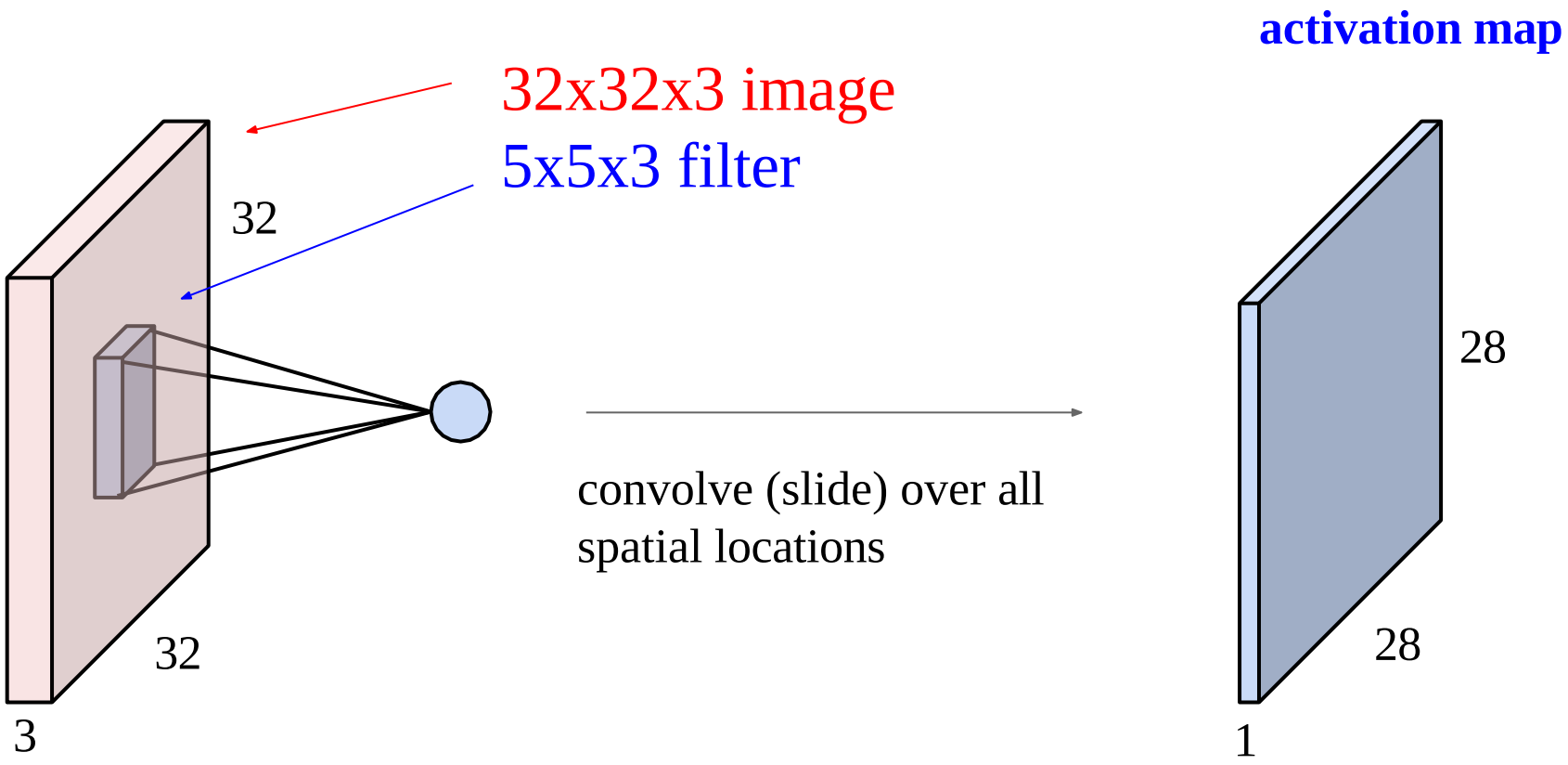
视频分类: 3D CNN

- 使用3维版本的卷积、pooling层来处理视频数据

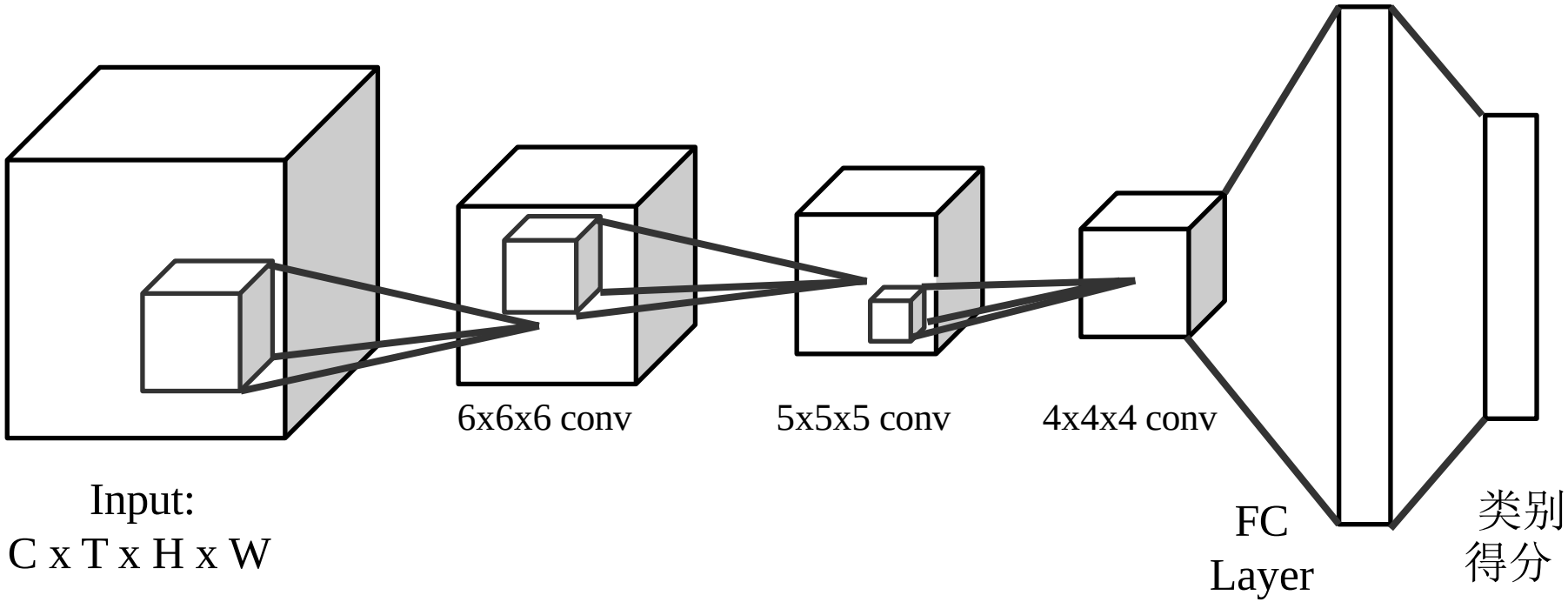


Ji et al, “3D Convolution Neural Networks for Human Action Recognition”, TPAMI 2010 ; Karpathy et al, “Large-scale 视频分类 with 卷积 Neural Networks”, CVPR 2014

2D卷积


activation map
32x32x3 image
5x5x3 filter
32
32
3
convolve (slide) over all
spatial locations
28
28
1

3D 卷积

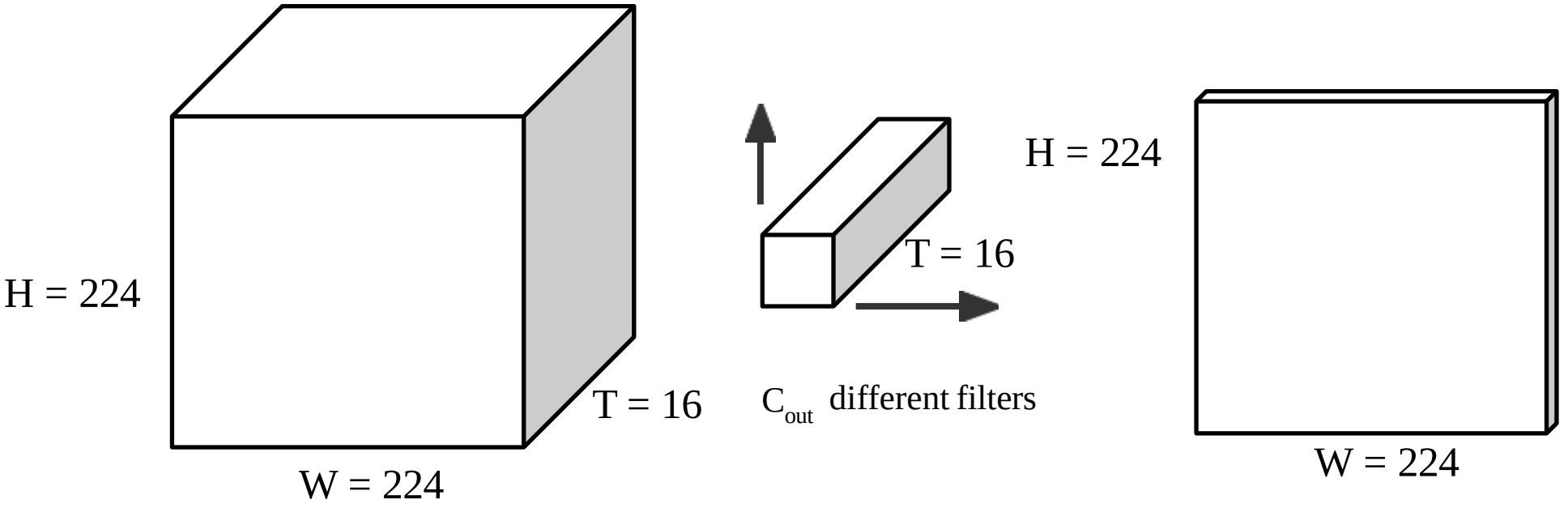

6x6x6 conv
5x5x5 conv
4x4x4 conv
Input:
C x T x H x W
FC
Layer
类别
得分

3D 卷积：所有帧全部考虑在内

Input: C_{in} x T x H x W
(3D grid with C_{in}-dim feat at each point)

Weight:
C_{out} x C_{in} x T x 3 x 3
Slide over x and y

Output:
C_{out} x H x W
2D grid with C_{out} – dim feat at each point

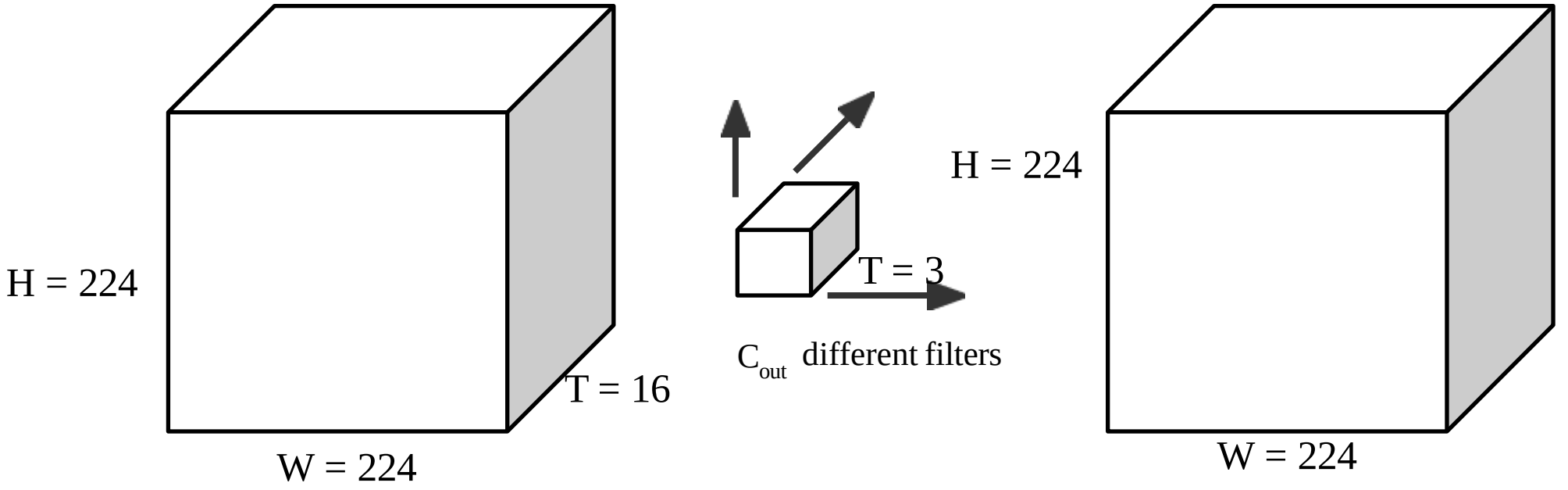


3D 卷积：时间上也做卷积

Input: C_{in} x T x H x W
(3D grid with C_{in}-dim
feat at each point)

Weight:
C_{out} x C_{in} x 3 x 3 x 3
Slide over x and y

Output:
C_{out} x T x H x W
3D grid with C_{out}–dim
feat at each point



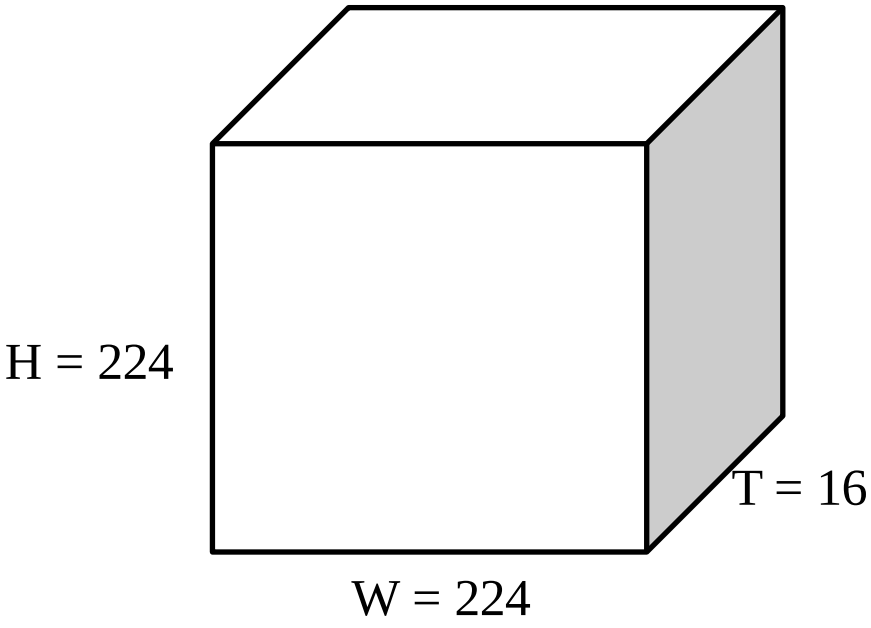
C_{out} different filters

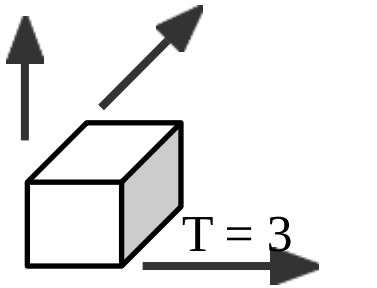
3D 卷积：同样也可以做可视化！

Input: C_{in} x T x H x W
(3D grid with C_{in}-dim feat at each point)

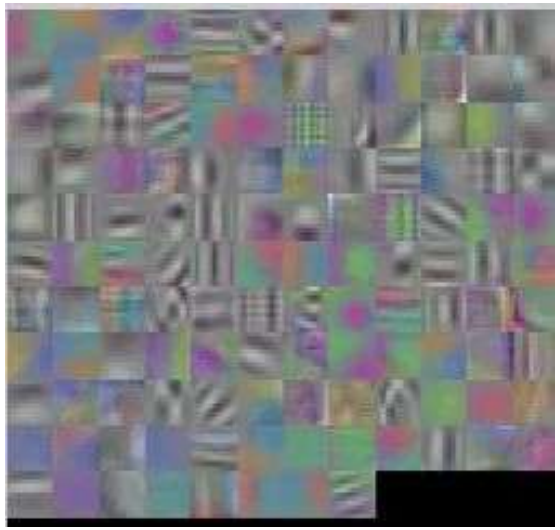
Weight:
C_{out} x C_{in} x 3 x 3 x 3
Slide over x and y

First-layer filters have shape 3 (RGB) x 4 (frames) x 5 x 5 (space)
Can visualize as video clips!





C_{out} different filters



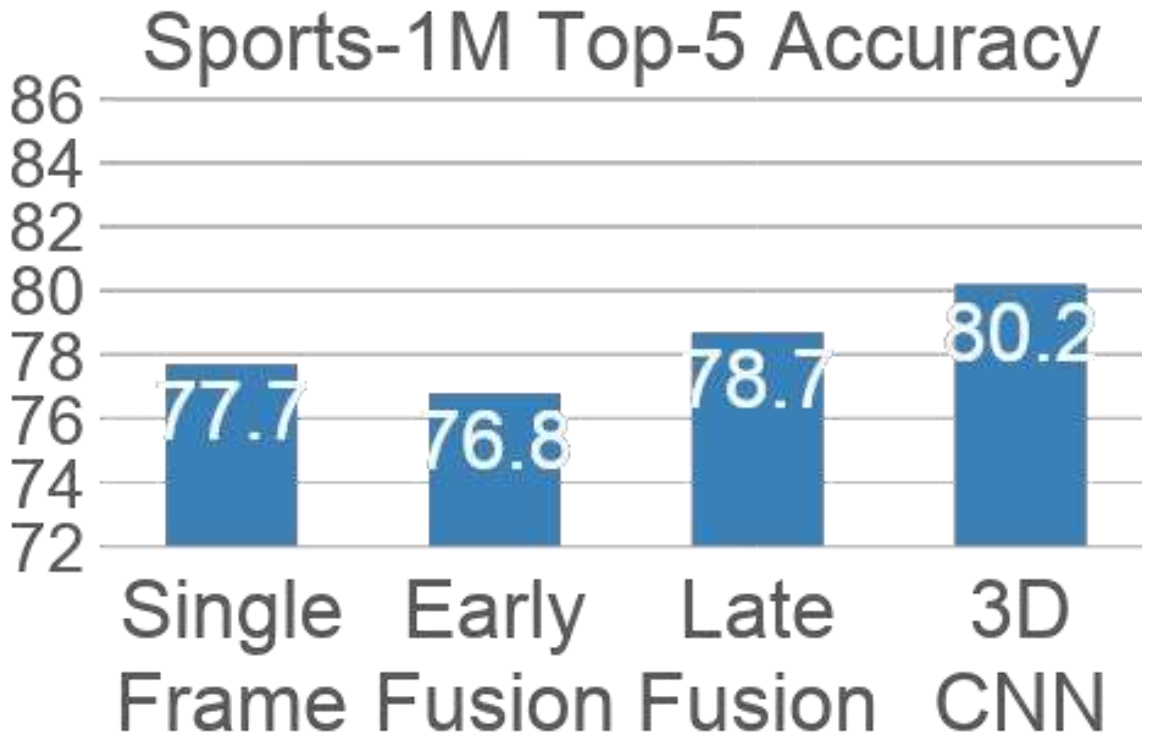
Sports-1M

1 million YouTube videos annotated with labels for 487 different types of sports

Ground Truth
Correct prediction
Incorrect prediction

Karpathy et al, “Large-scale 视频分类 with 卷积 Neural Networks”, CVPR 2014

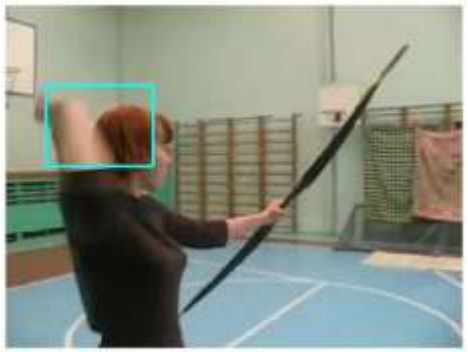
3D CNN



Karpathy et al, “Large-scale 视频分类 with 卷积 Neural Networks”, CVPR 2014

光流

Image at frame t



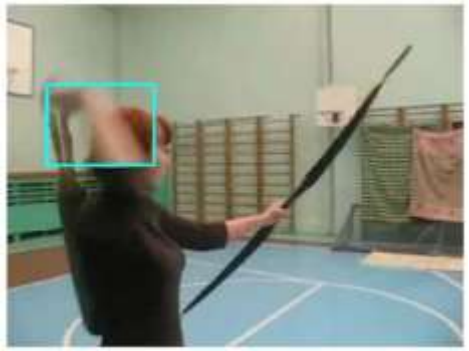
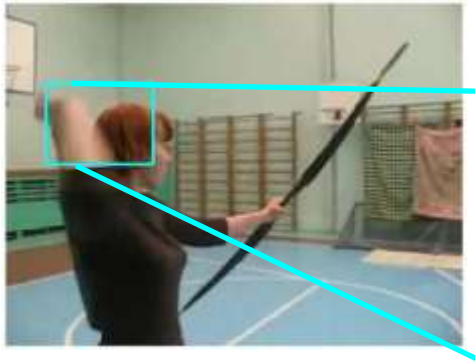


Image at frame t+1

光流

Image at frame t



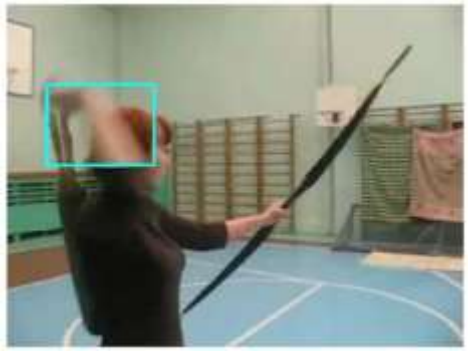
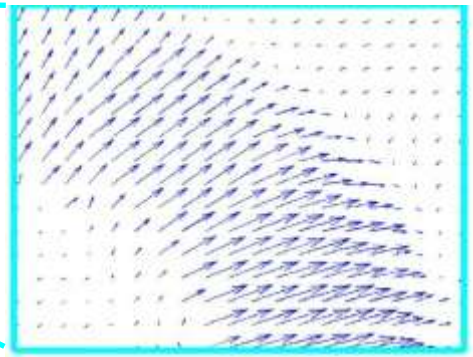


Image at frame t+1

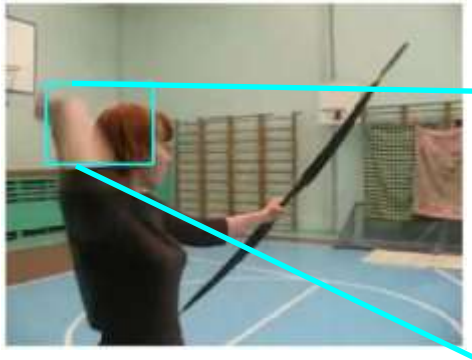


下一帧该像素会出现在哪里：$F(x, y) = (dx, dy)$

$I_{t+1}(x+dx, y+dy) = I_t(x, y)$

光流

Image at frame t



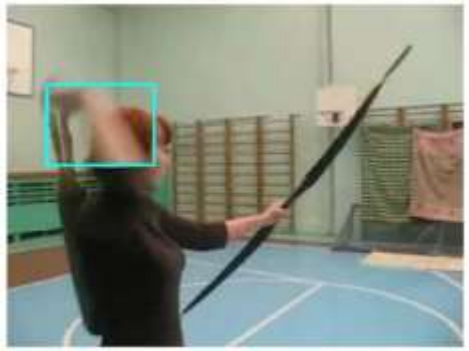
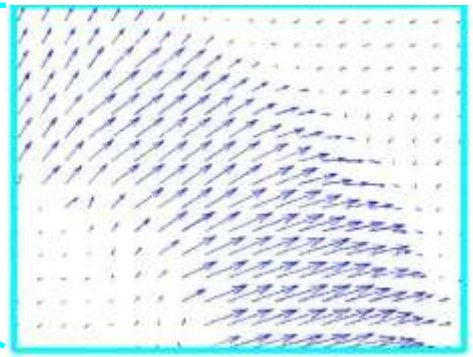


Image at frame t+1



下一帧该像素会出现在哪里：$F(x, y) = (dx, dy)$

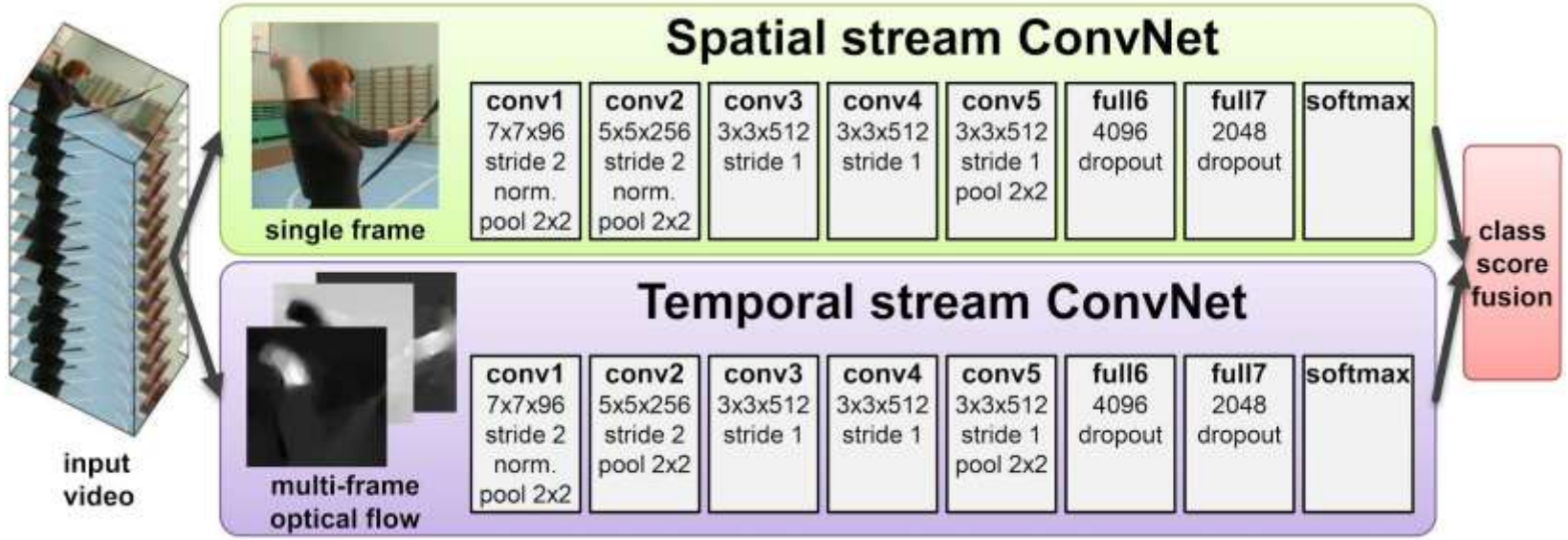
$I_{t+1}(x+dx, y+dy) = I_t(x, y)$

Horizontal flow dx

Vertical Flow dy

Two-Stream 网络：分开处理运动和外表

Input: Single Image
3 x H x W

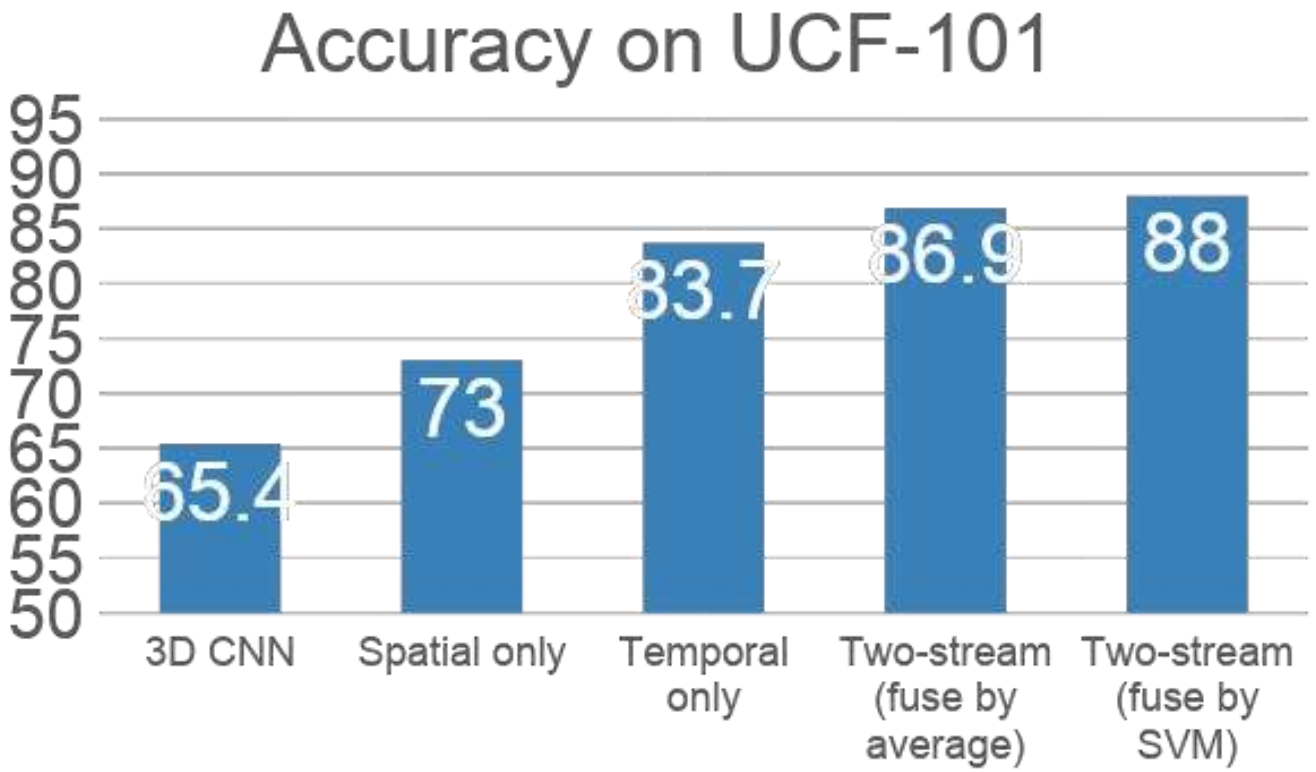


Input: Stack of optical flow:
[2*(T-1)] x H x W

Early fusion: First 2D conv processes all flow images

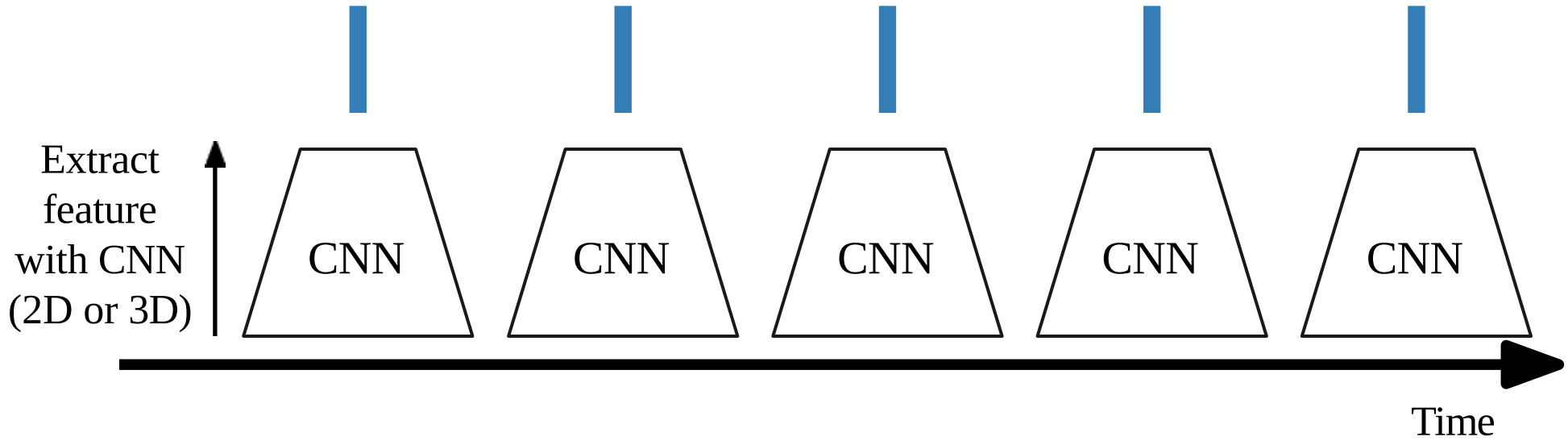
Simonyan and Zisserman, “Two-stream 卷积 networks for action recognition in videos”, NeurIPS 2014

Two-Stream 网络：分开处理运动和外表

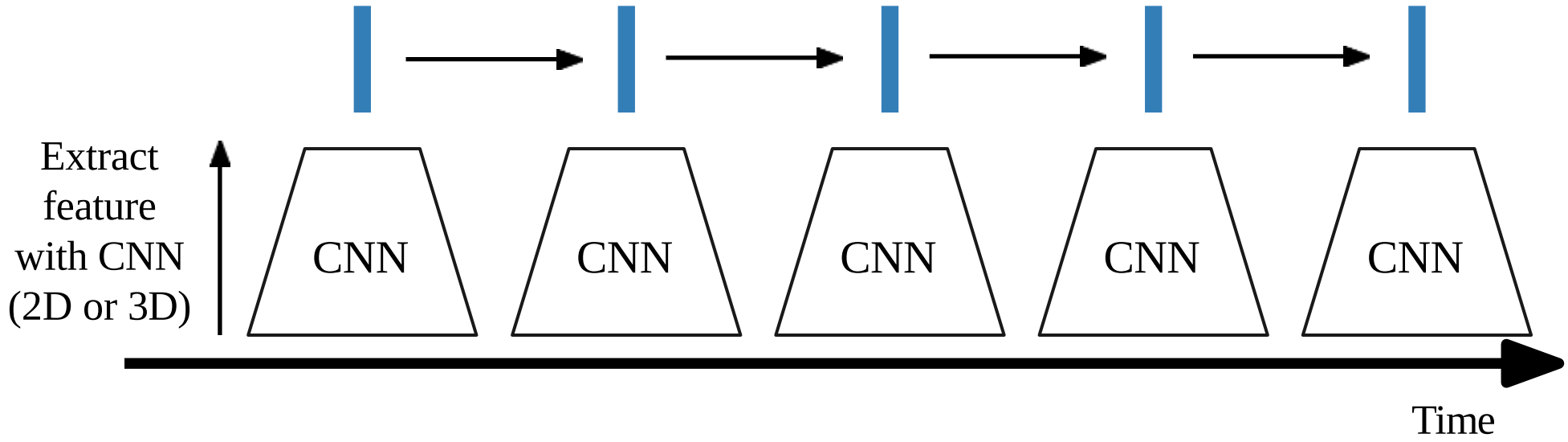


Simonyan and Zisserman, “Two-stream 卷积 networks for action recognition in videos”, NeurIPS 2014

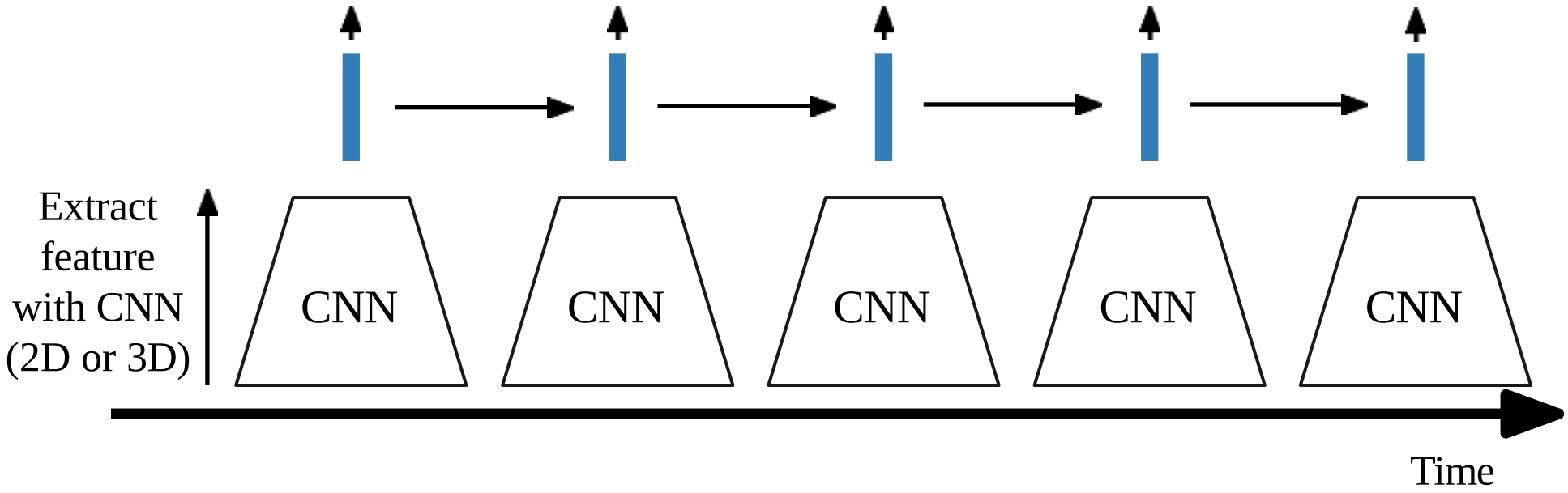
提取长远的时序特征



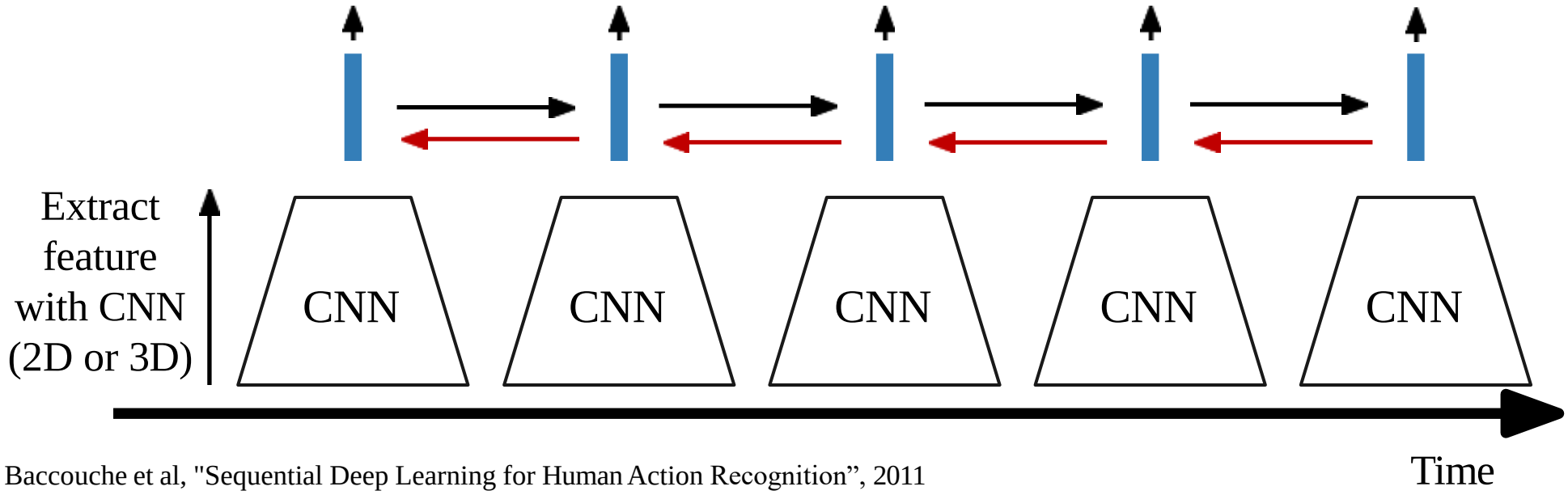
提取长远的时序特征



提取长远的时序特征

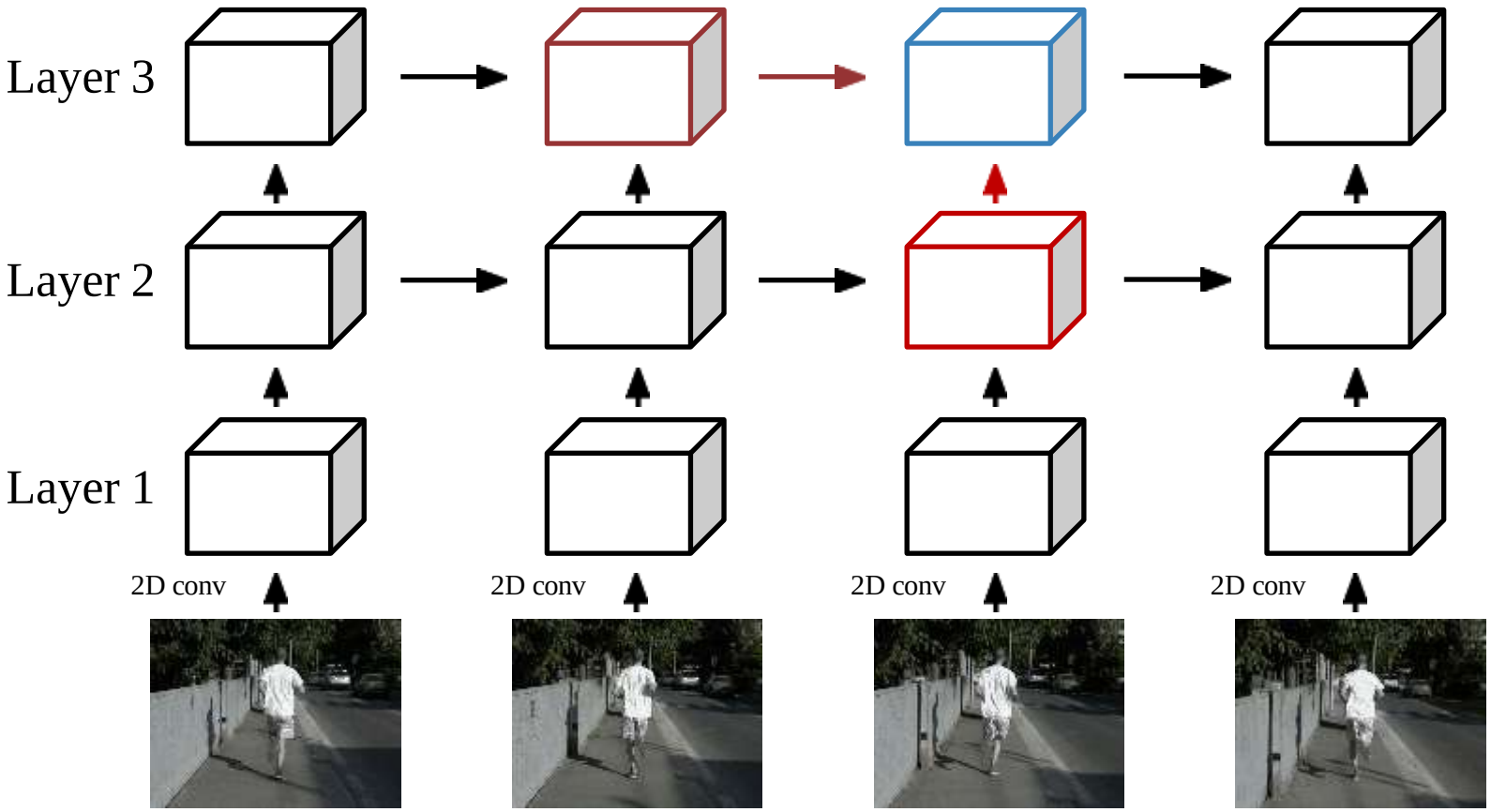


提取长远的时序特征



Baccouche et al, "Sequential Deep Learning for Human Action Recognition", 2011
Donahue et al, "Long-term recurrent convolution networks for visual recognition and description", CVPR 2015

RNN



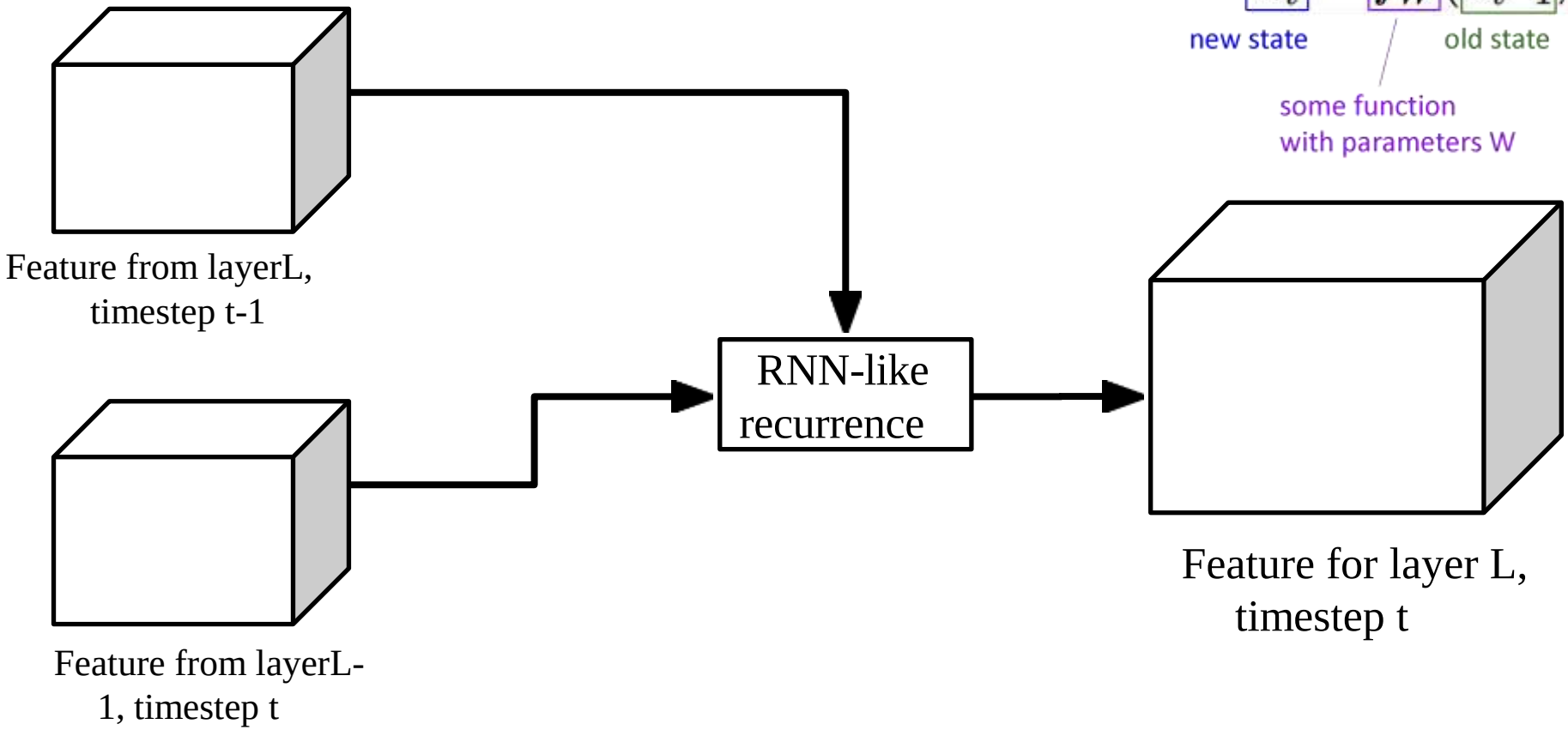
RNN

Recall: Recurrent Network

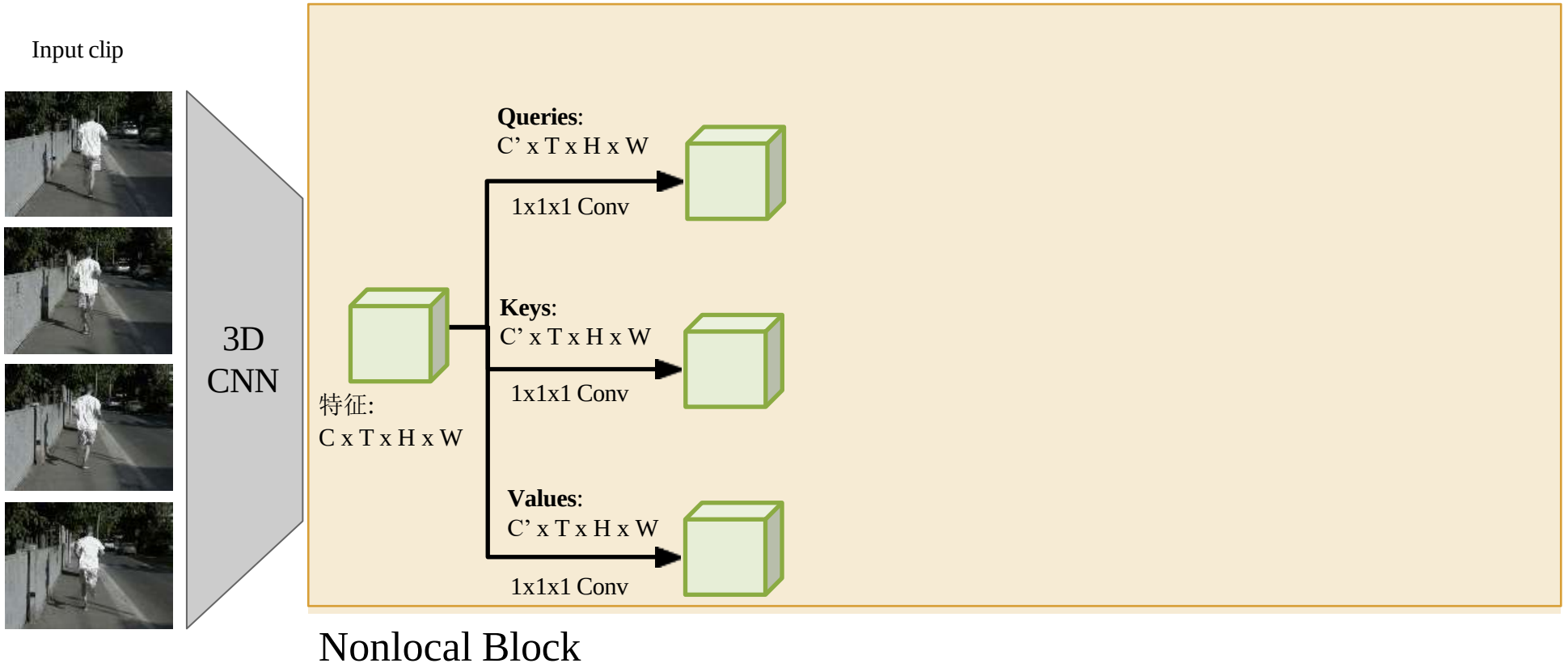
$$h_t = f_W(h_{t-1}, x_t)$$

new state — old state

some function with parameters W

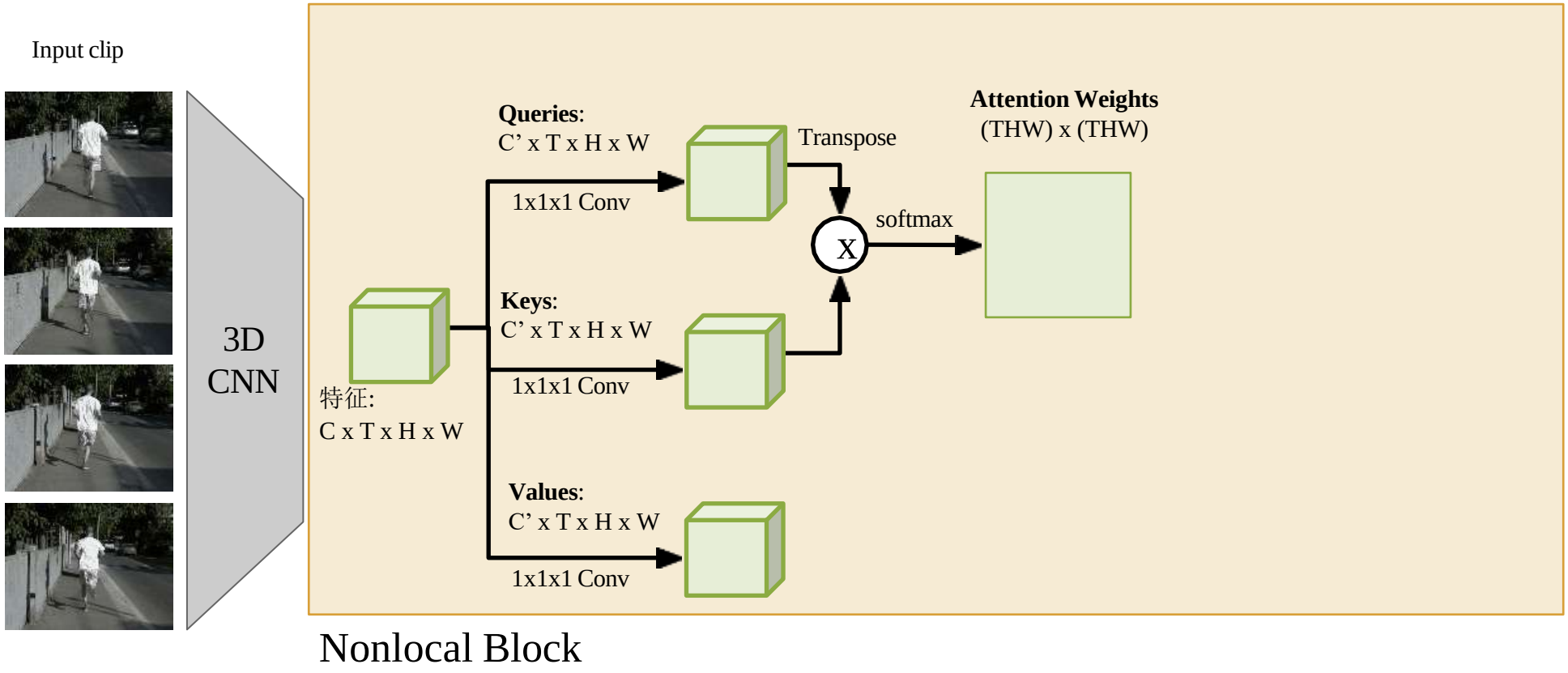


Spatio-Temporal Self-Attention (Nonlocal Block)



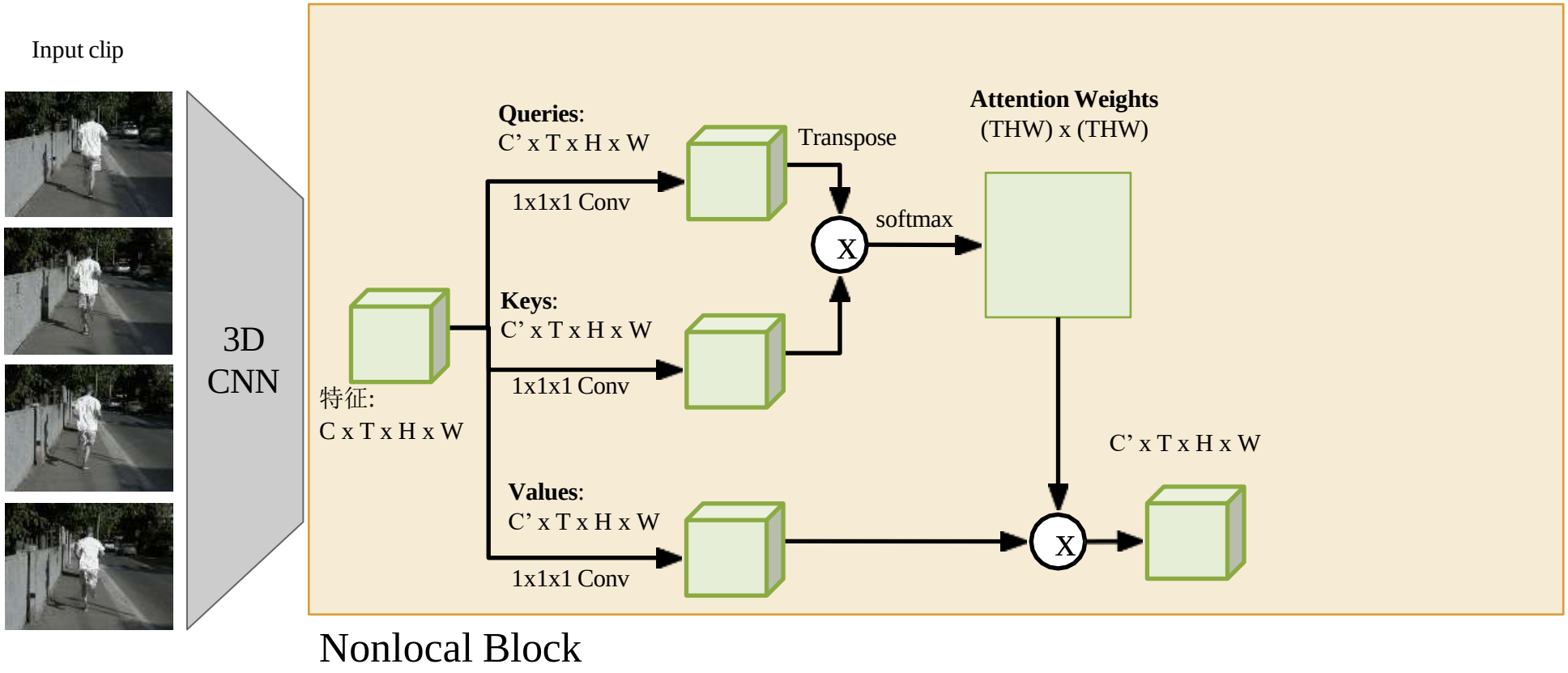
Wang et al, "Non-local neural networks", CVPR 2018

Spatio-Temporal Self-Attention (Nonlocal Block)



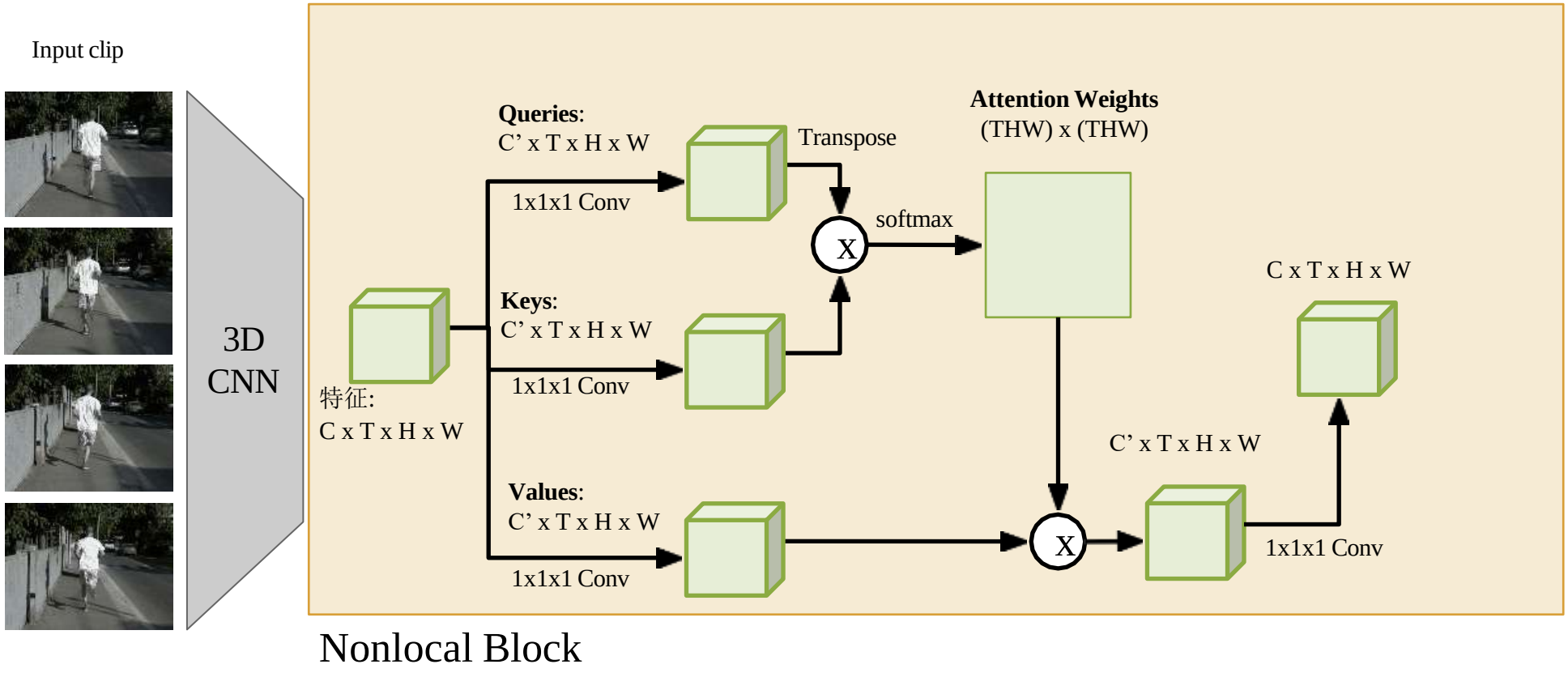
Wang et al, "Non-local neural networks", CVPR 2018

Spatio-Temporal Self-Attention (Nonlocal Block)



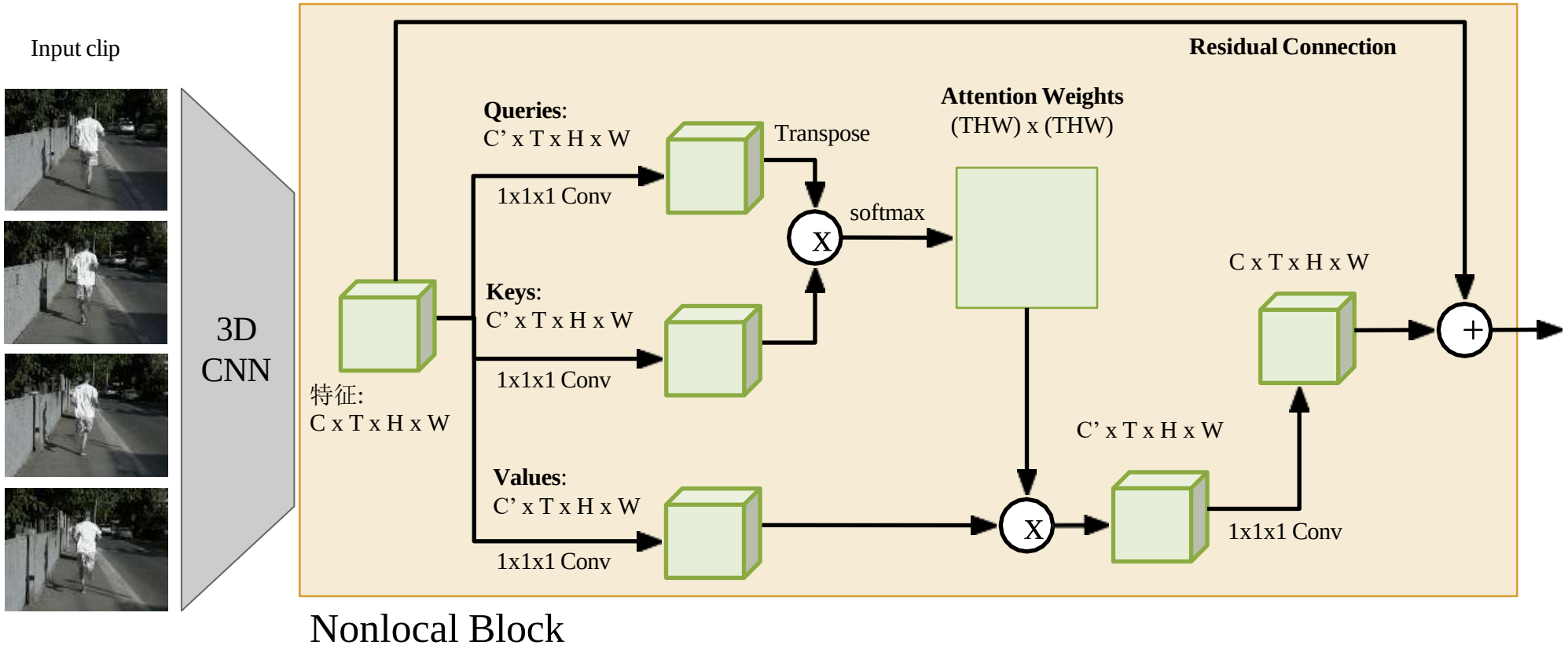
Wang et al, "Non-local neural networks", CVPR 2018

Spatio-Temporal Self-Attention (Nonlocal Block)



Wang et al, "Non-local neural networks", CVPR 2018

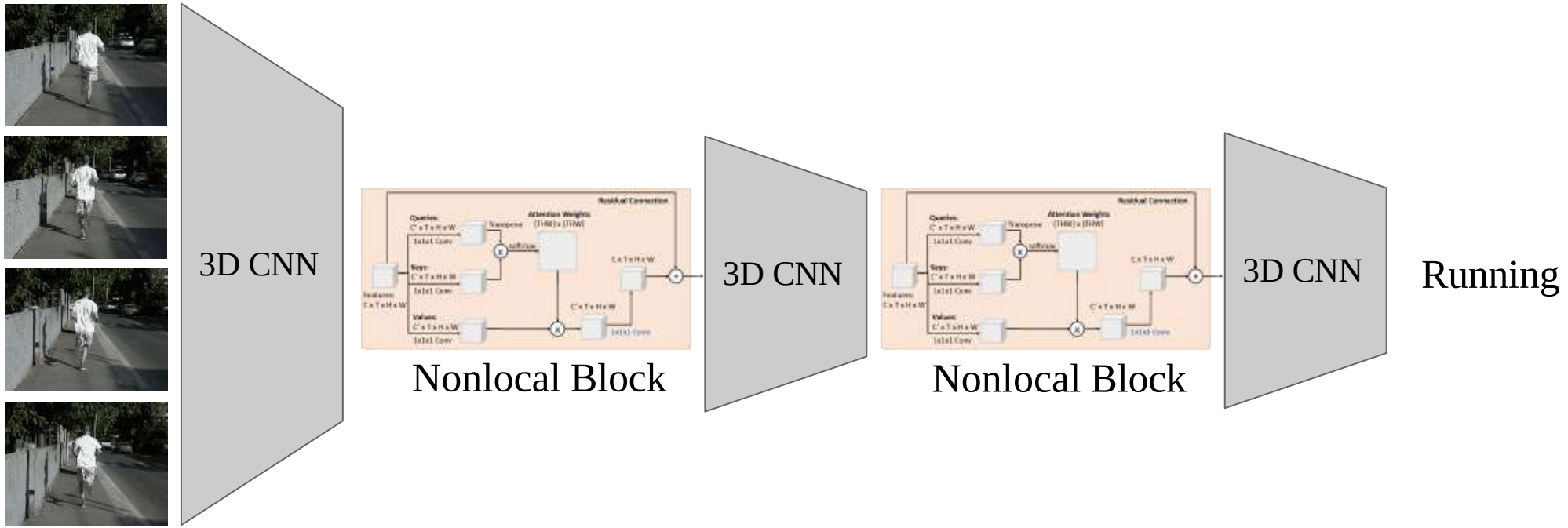
Spatio-Temporal Self-Attention (Nonlocal Block)



Wang et al, "Non-local neural networks", CVPR 2018

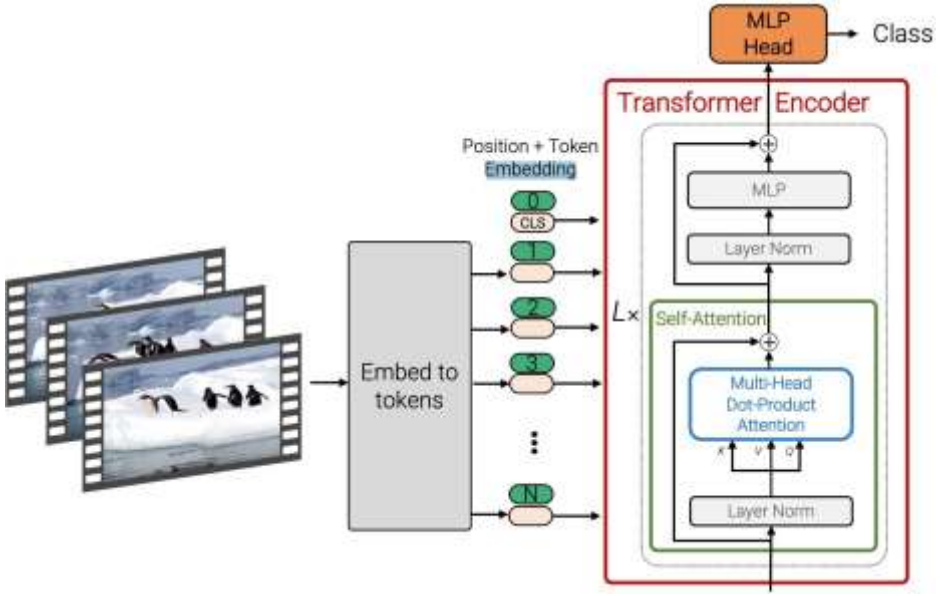
Spatio-Temporal Self-Attention (Nonlocal Block)

Input clip

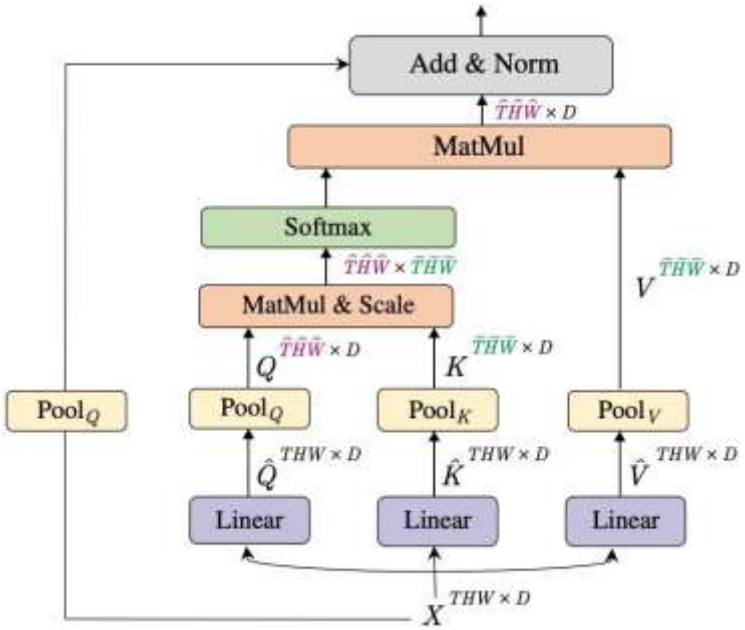


Wang et al, “Non-local neural networks”, CVPR 2018

Vision Transformers for Video

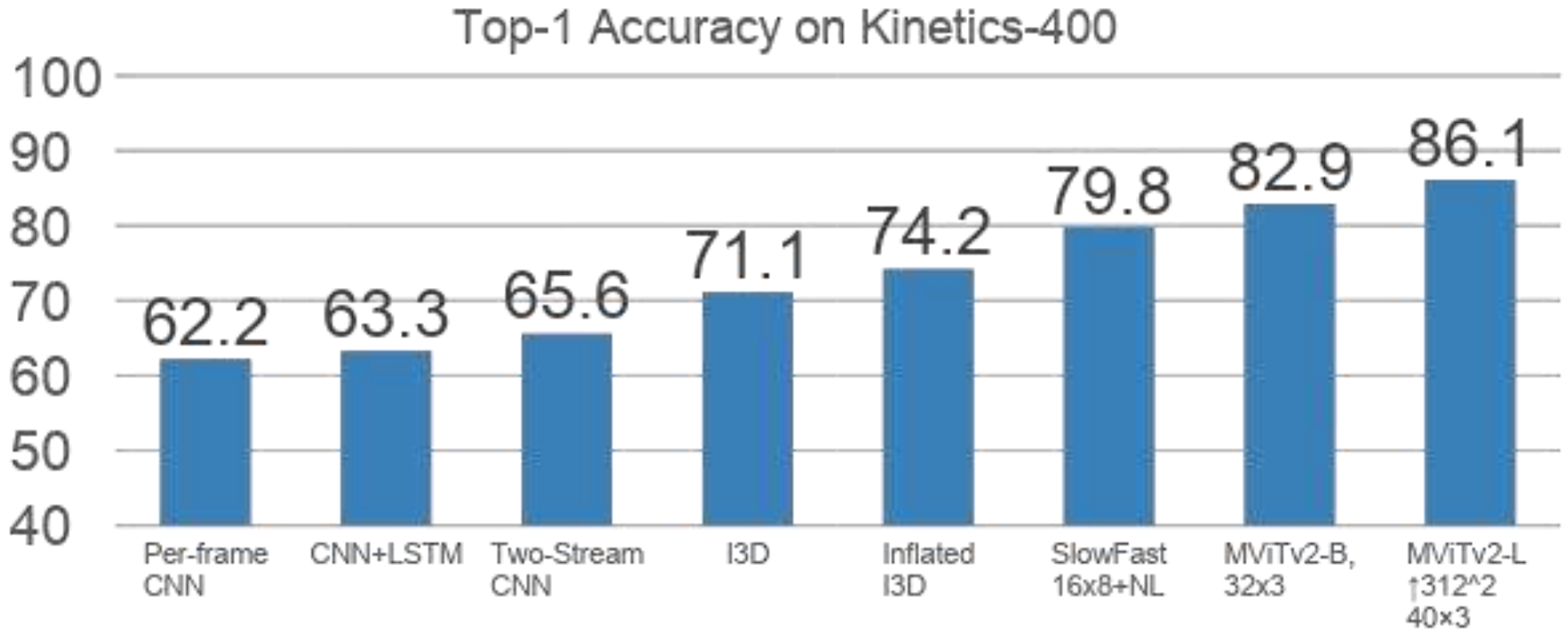


Bertasius et al, "Is Space-Time Attention All You Need for Video Understanding?", ICML 2021 Arnab et al, "ViViT: A Video Vision Transformer", ICCV 2021
Neimark et al, "Video Transformer Network", ICCV 2021



Fan et al, "Multiscale Vision Transformers", ICCV 2021 Li et al, "MViTv2: Improved Multiscale Vision Transformers for claissification and Detection", CVPR 2022

Vision Transformers for Video



Li et al, “MViTv2: Improved Multiscale Vision Transformers for 分类 and Detection”, CVPR 2022

可视化 视频模型

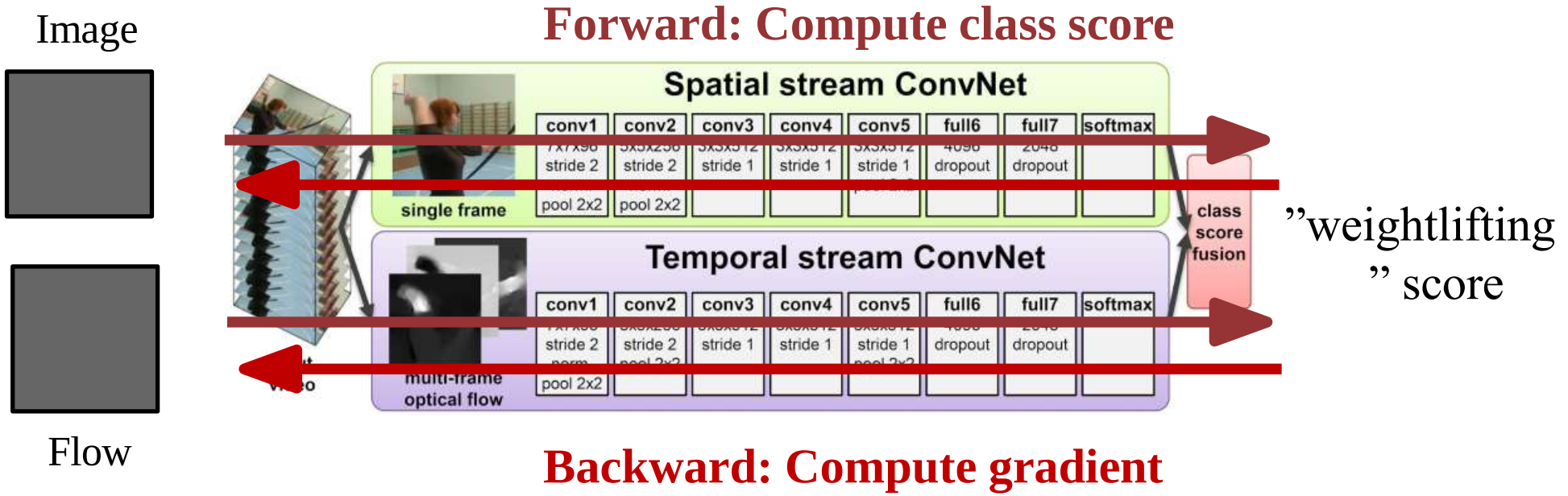
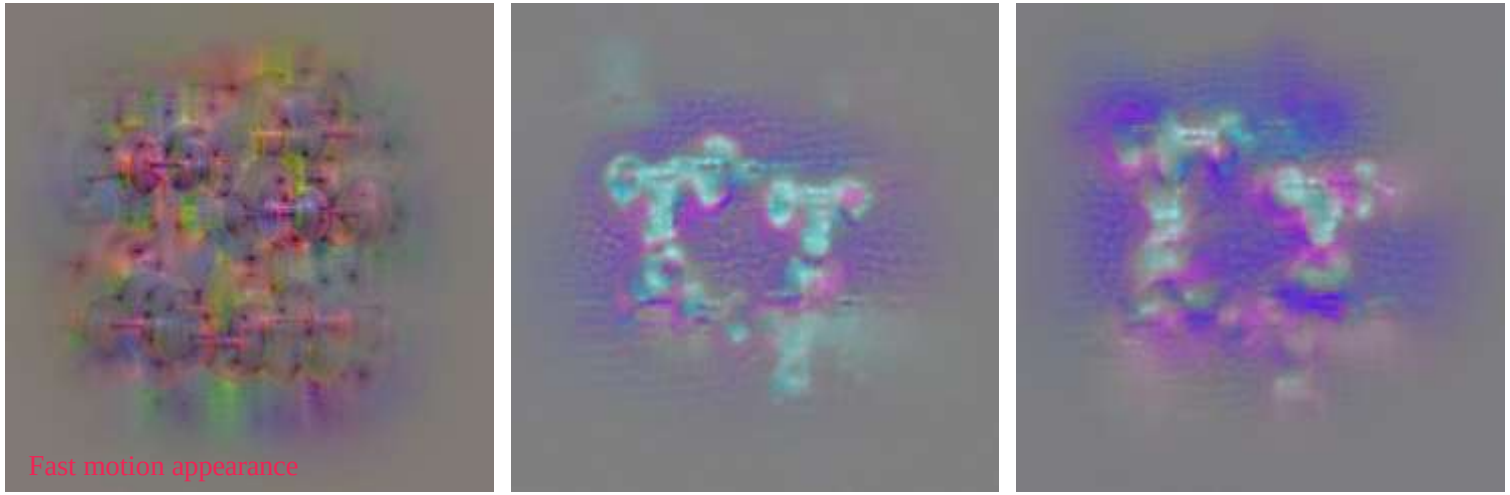


Image
Flow
Forward: Compute class score
Spatial stream ConvNet
conv1
conv2
conv3
conv4
conv5
full6
full7
softmax
stride 2
stride 2
stride 1
stride 1
stride 1
dropout
dropout
pool 2x2
pool 2x2
single frame
Temporal stream ConvNet
conv1
conv2
conv3
conv4
conv5
full6
full7
softmax
stride 2
stride 2
stride 1
stride 1
stride 1
dropout
dropout
pool 2x2
multi-frame optical flow
class score fusion
”weightlifting ” score
Backward: Compute gradient

猜猜是什么动作？

Appearance “Slow” motion “Fast” motion



猜猜是什么动作？

Weightlifting

Appearance “Slow” motion “Fast” motion

猜猜是什么动作？

Appearance “Slow” motion “Fast” motion

猜猜是什么动作？

Apply Eye Makeup

Appearance “Slow” motion “Fast” motion