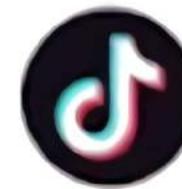


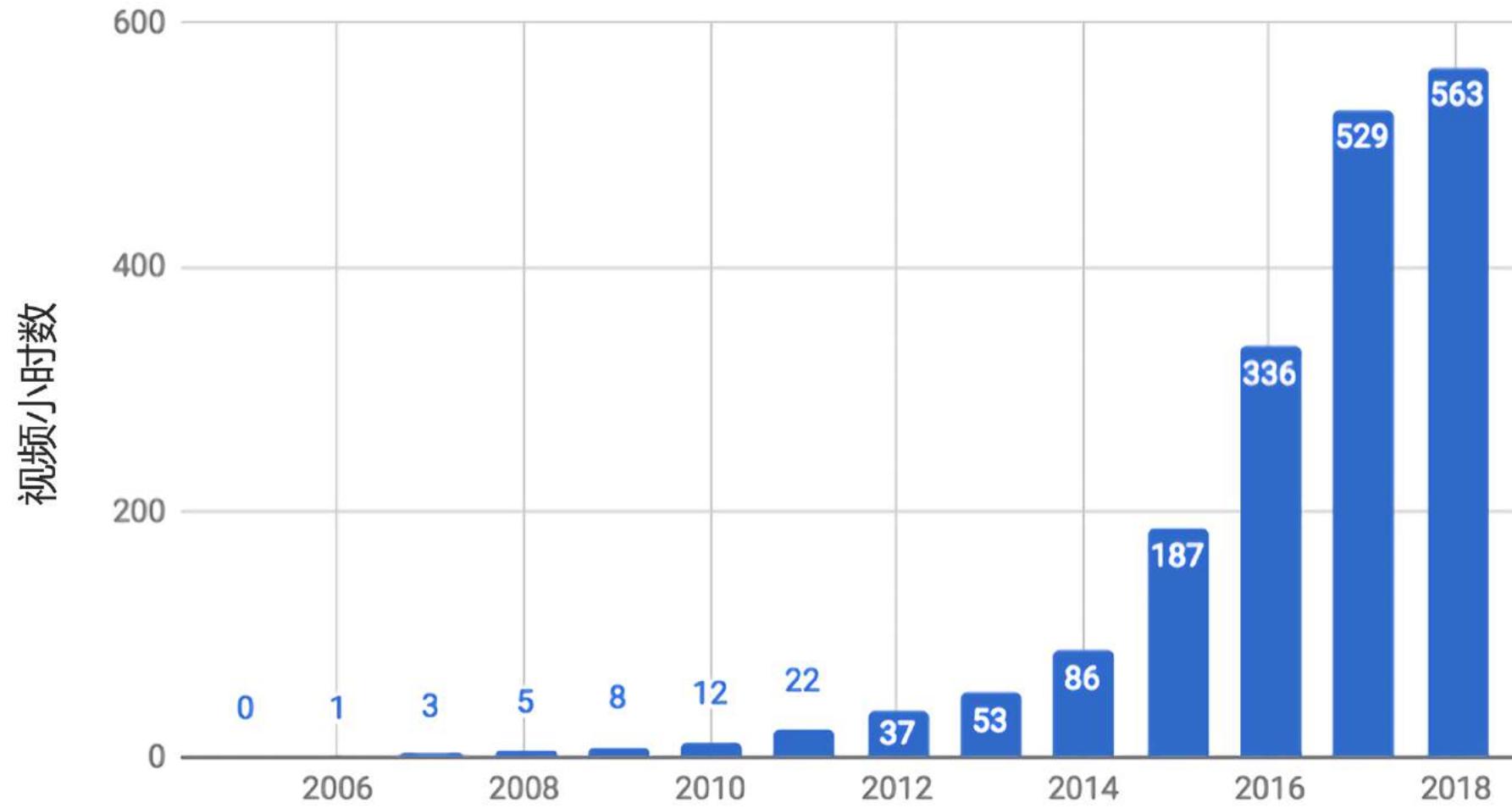
通用视觉框架OpenMMLab

第8讲 视频理解与MMAction2

林达华 教授
2021年6月

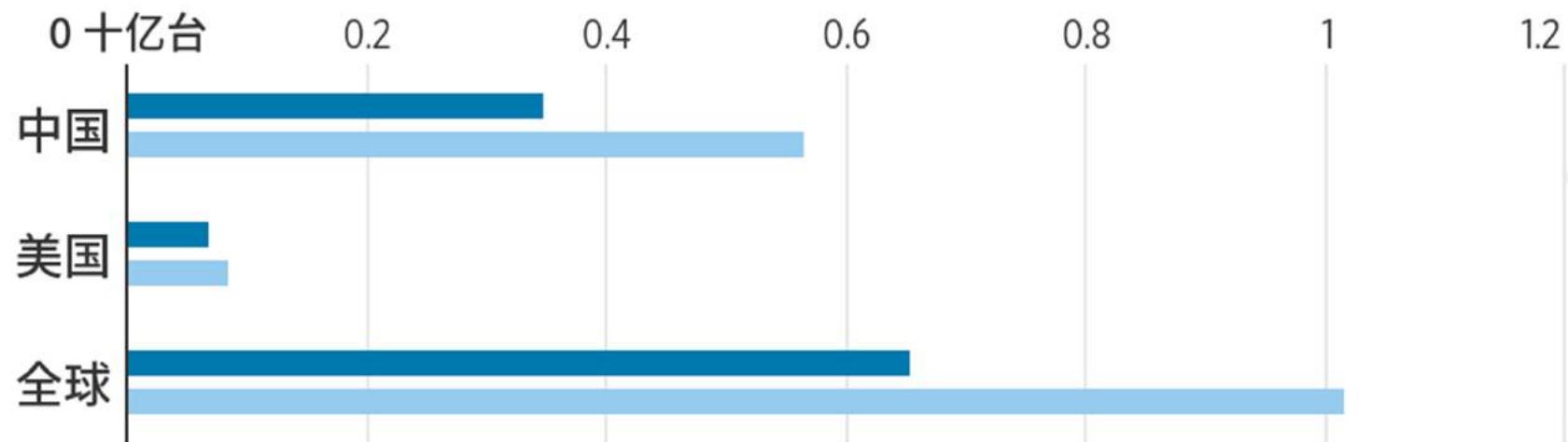


YouTube 上每分钟新增的视频时长



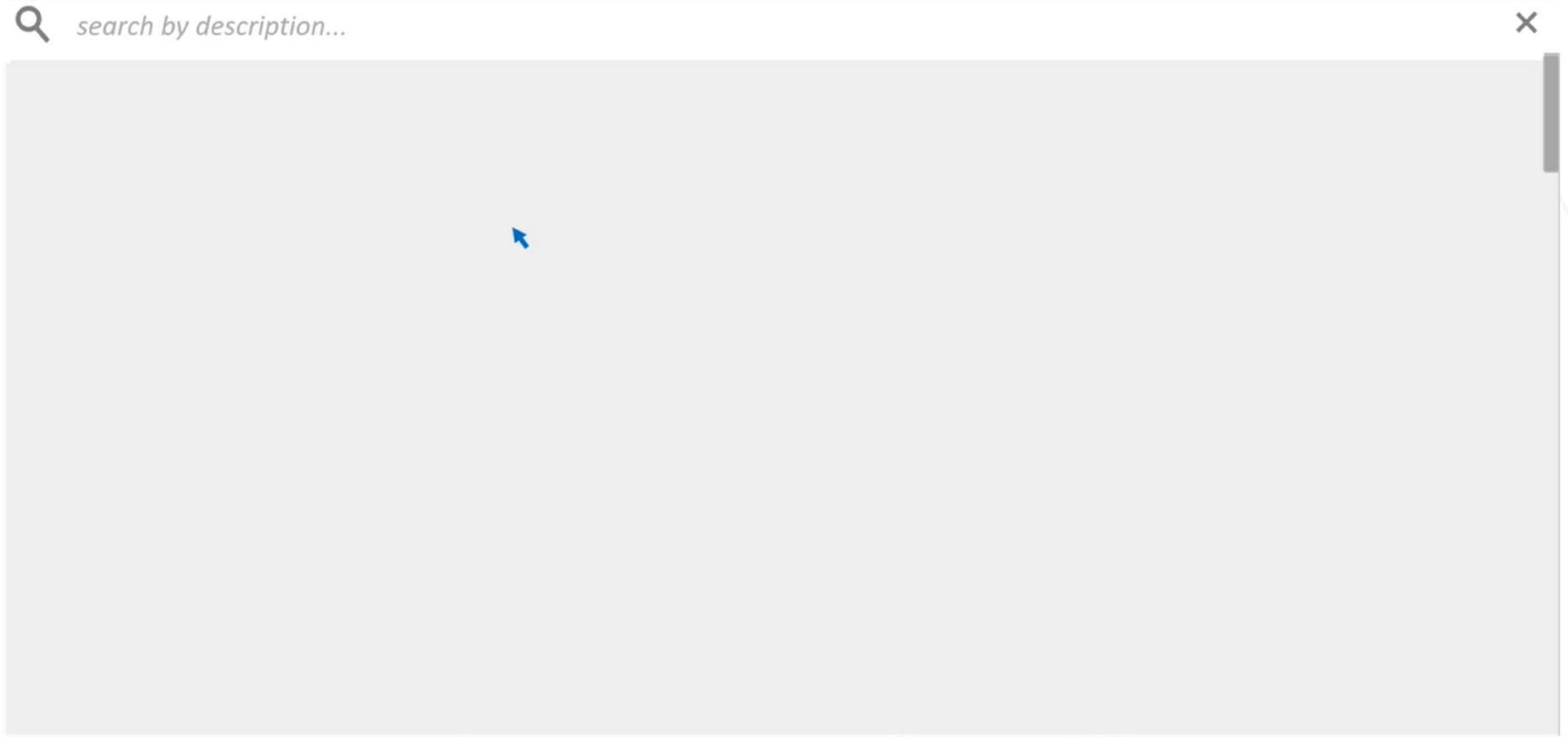
安全摄像头的安装数量

■ 2018 ■ 2021*



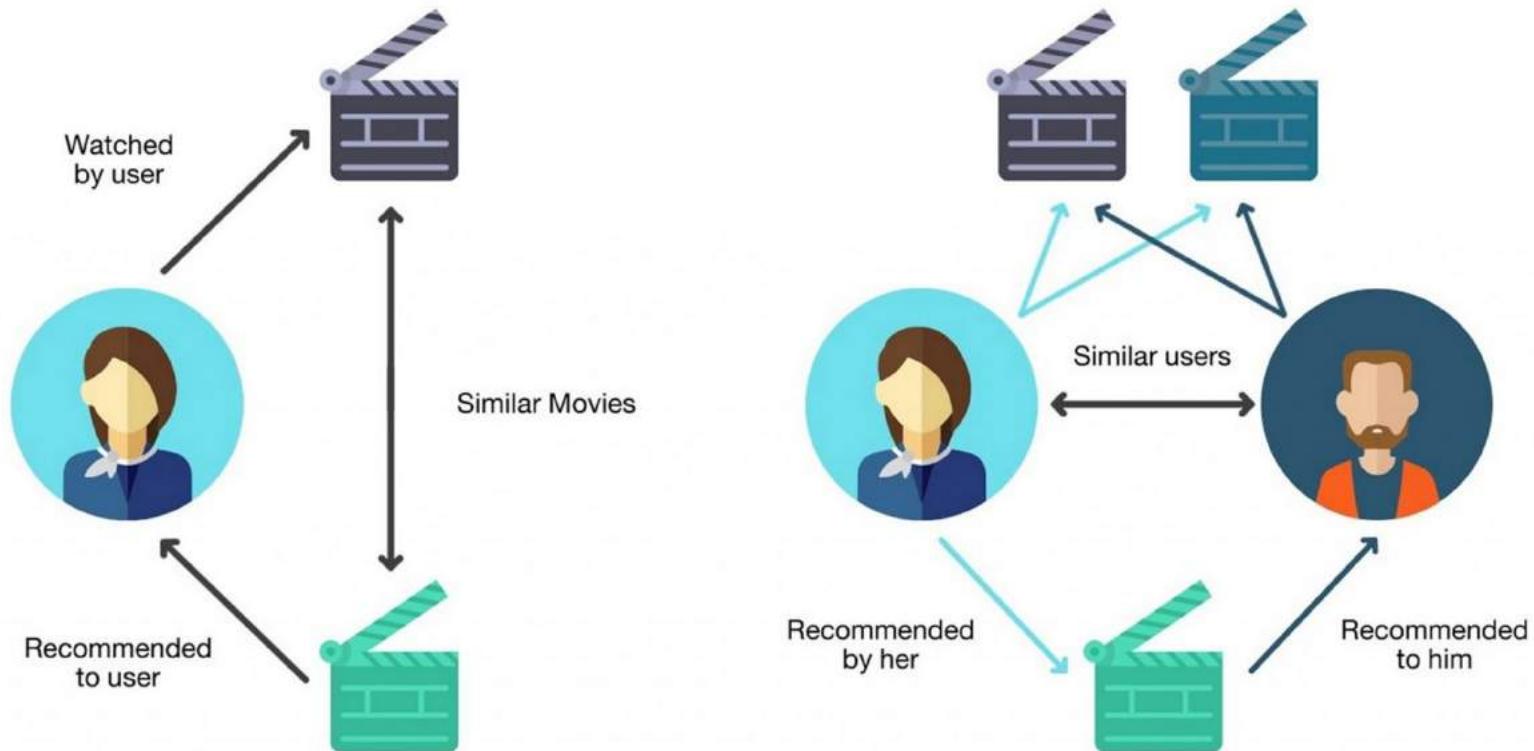
*预测值

数据来源：IHS Markit



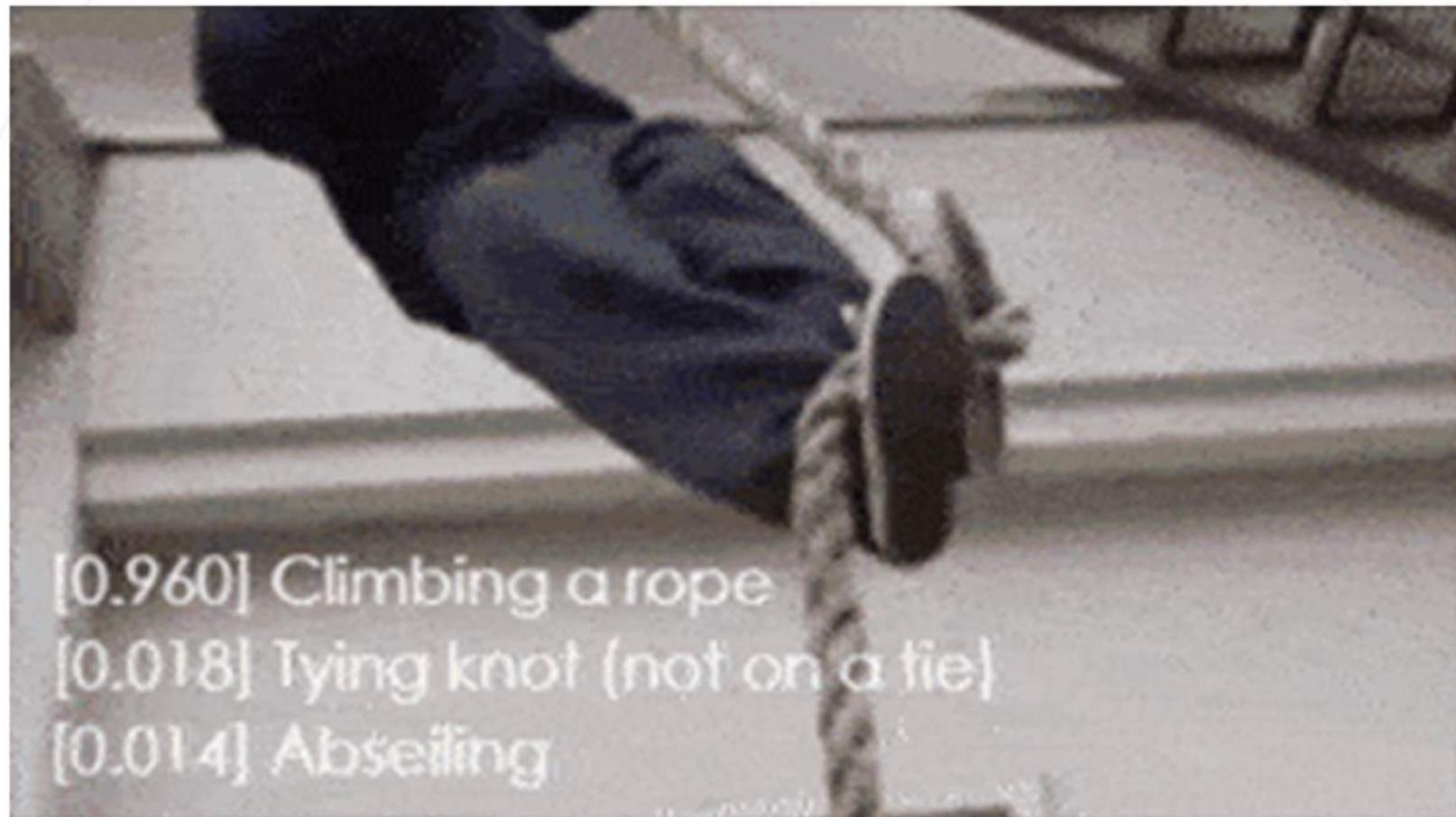


Rio Olympics 2016 – Man's 3M Spring Board Final

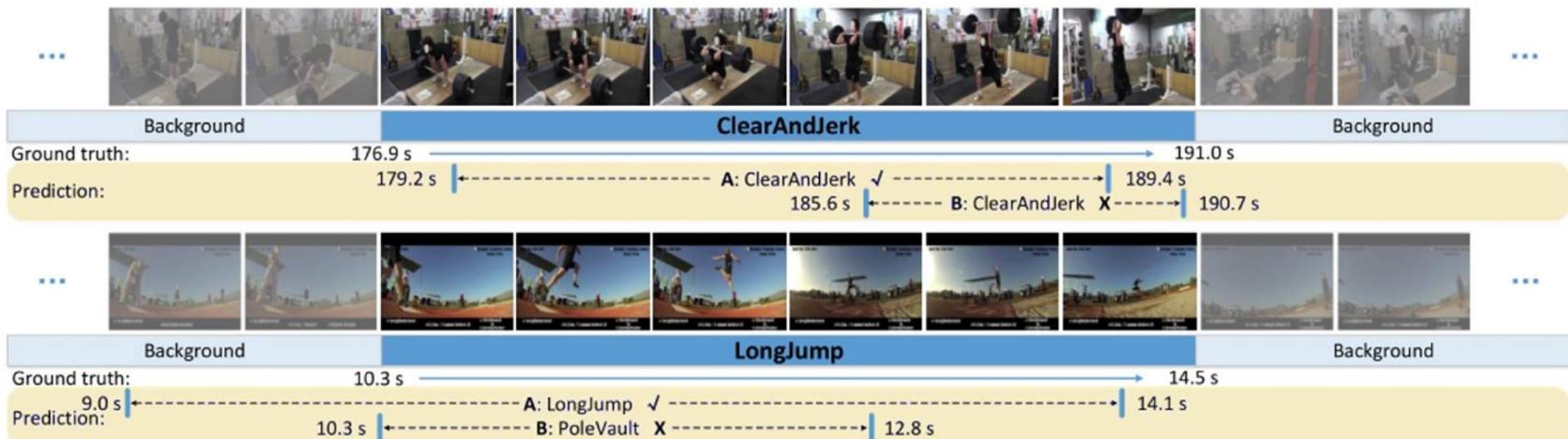


视频理解的基本任务

- ▶ 识别视频片段中出现的动作（通常为短视频）
- ▶ 视频的分类问题：What

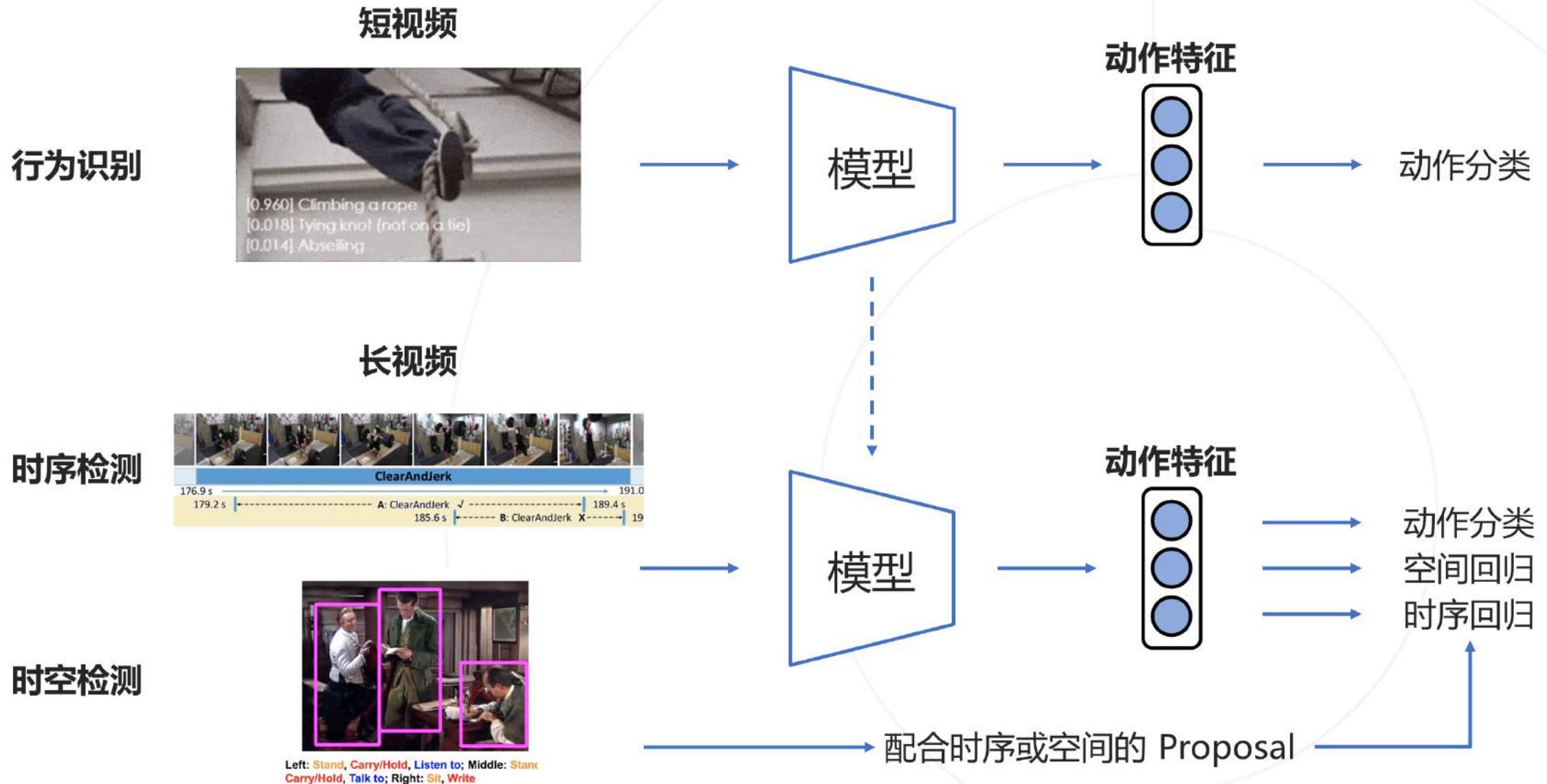


在长视频中定位特定动作出现的时间段，并对动作进行分类：When + What



- ▶ 识别并定位视频中出现的人和动作
- ▶ 视频的检测问题
What + When + Where

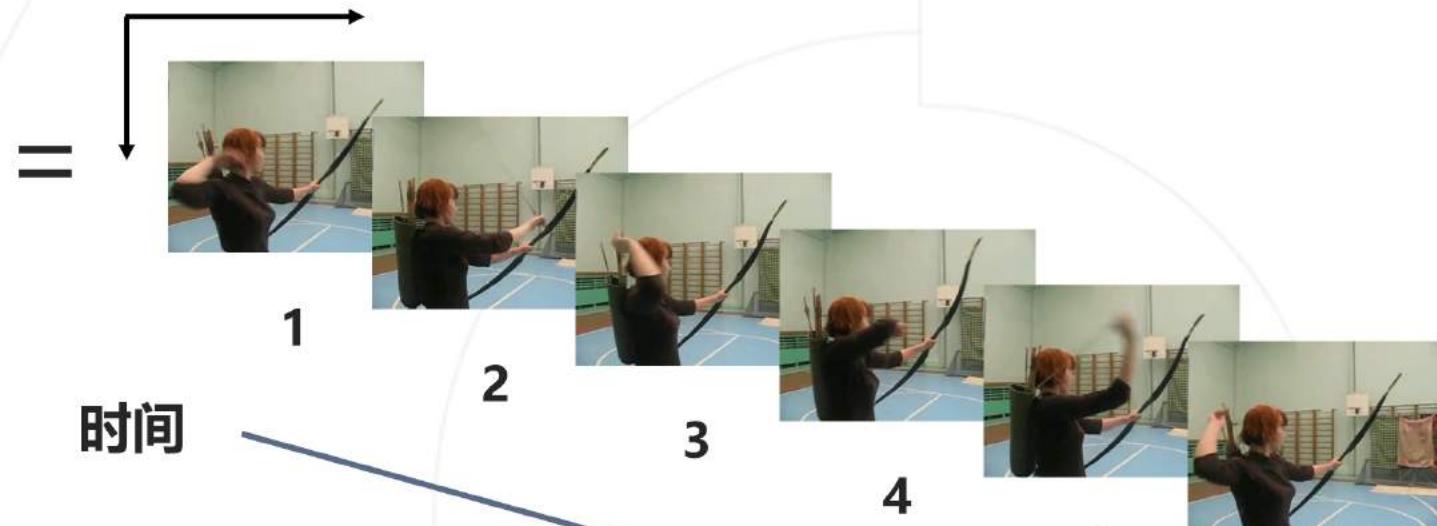




视频理解的挑战



空间



- 挑战 1：如何表示视频中的动作
- 挑战 2：如何高效处理大量视频数据
- 挑战 3：如何降低视频数据的标注成本

挑战1：如何描述视频中的动作

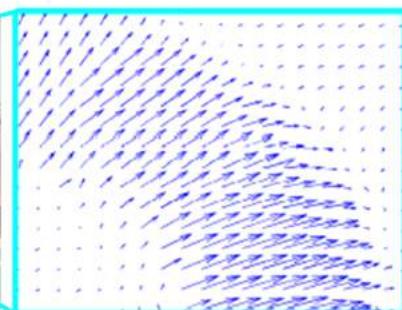
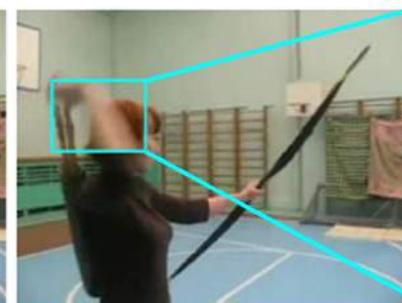
$$\text{动作} = \text{外观} + \text{运动}$$
$$\text{action} = \text{appearance} + \text{motion}$$

视频中的动作
action

两方面表现



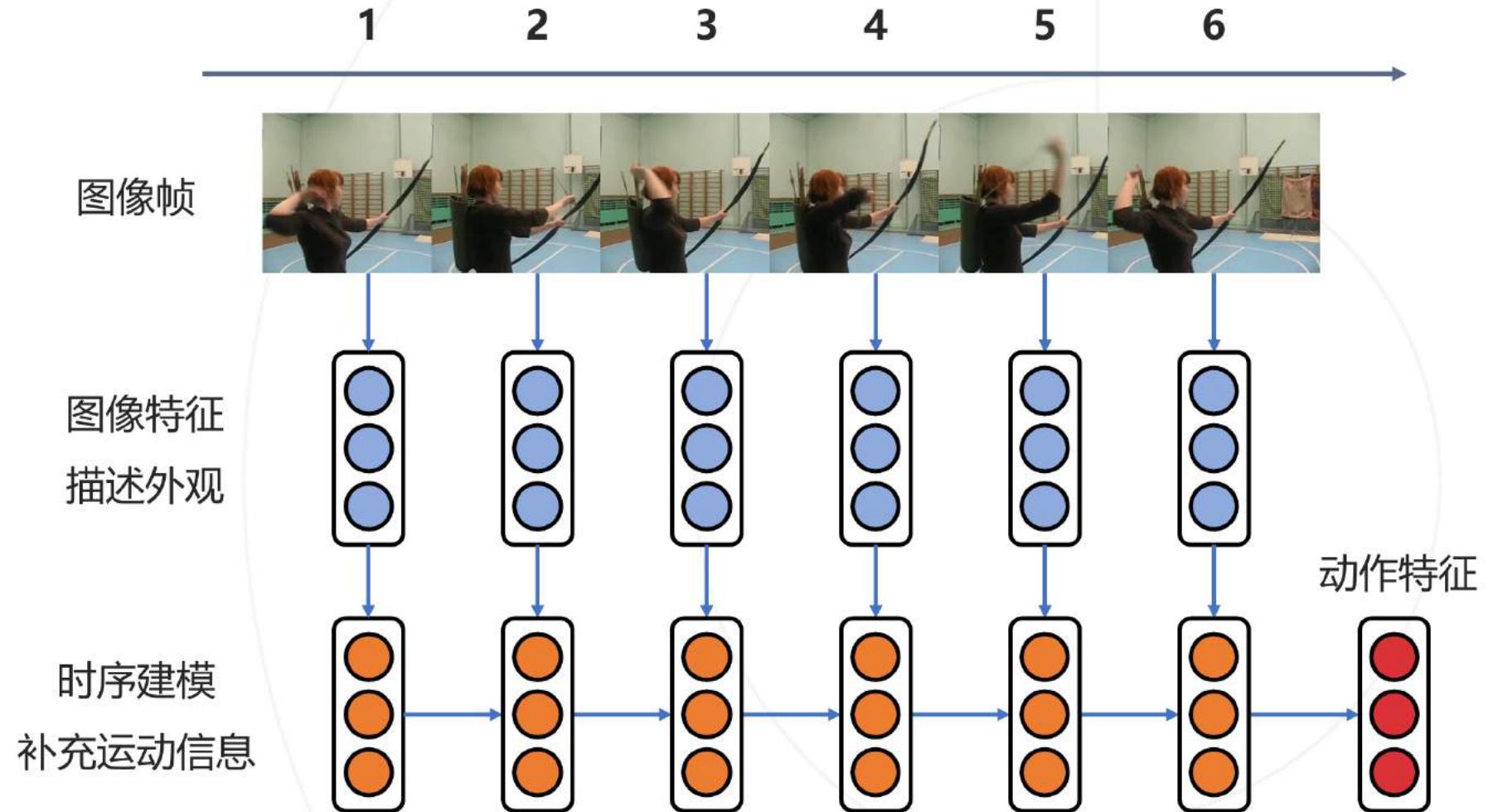
图像帧



帧间运动

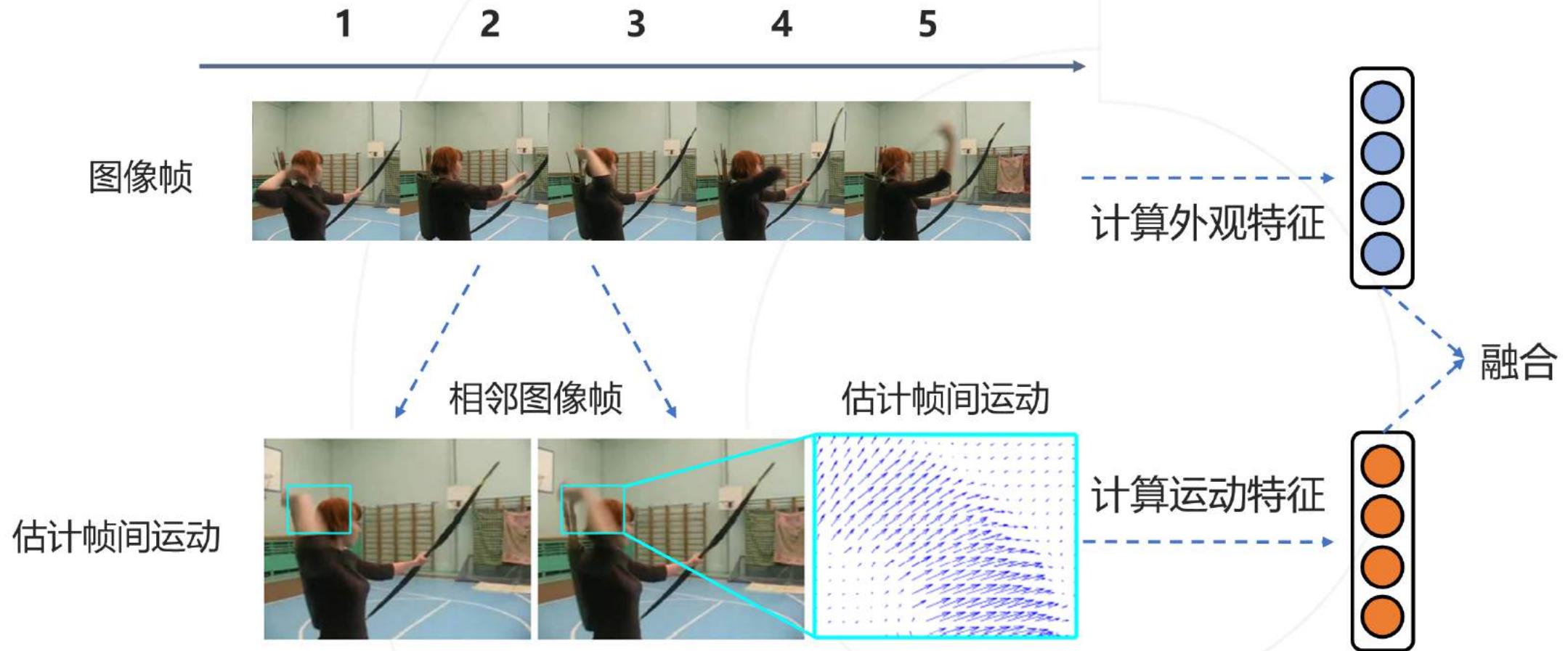
挑战1：如何描述视频中的动作

思路1：独立提取图像特征，再进行时序建模

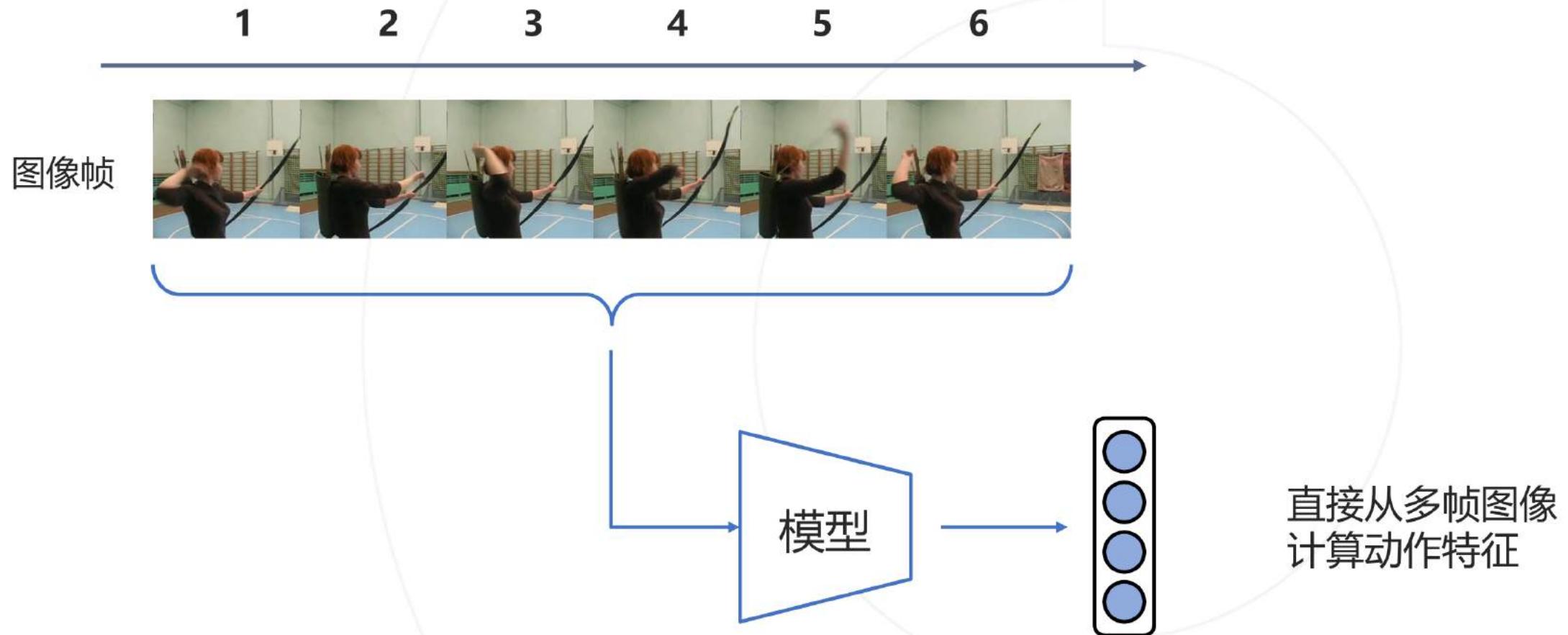


挑战1：如何描述视频中的动作

思路2：估计像素的运动，再提取运动特征，再与外观特征融合



思路3：从多帧图像直接计算运动特征

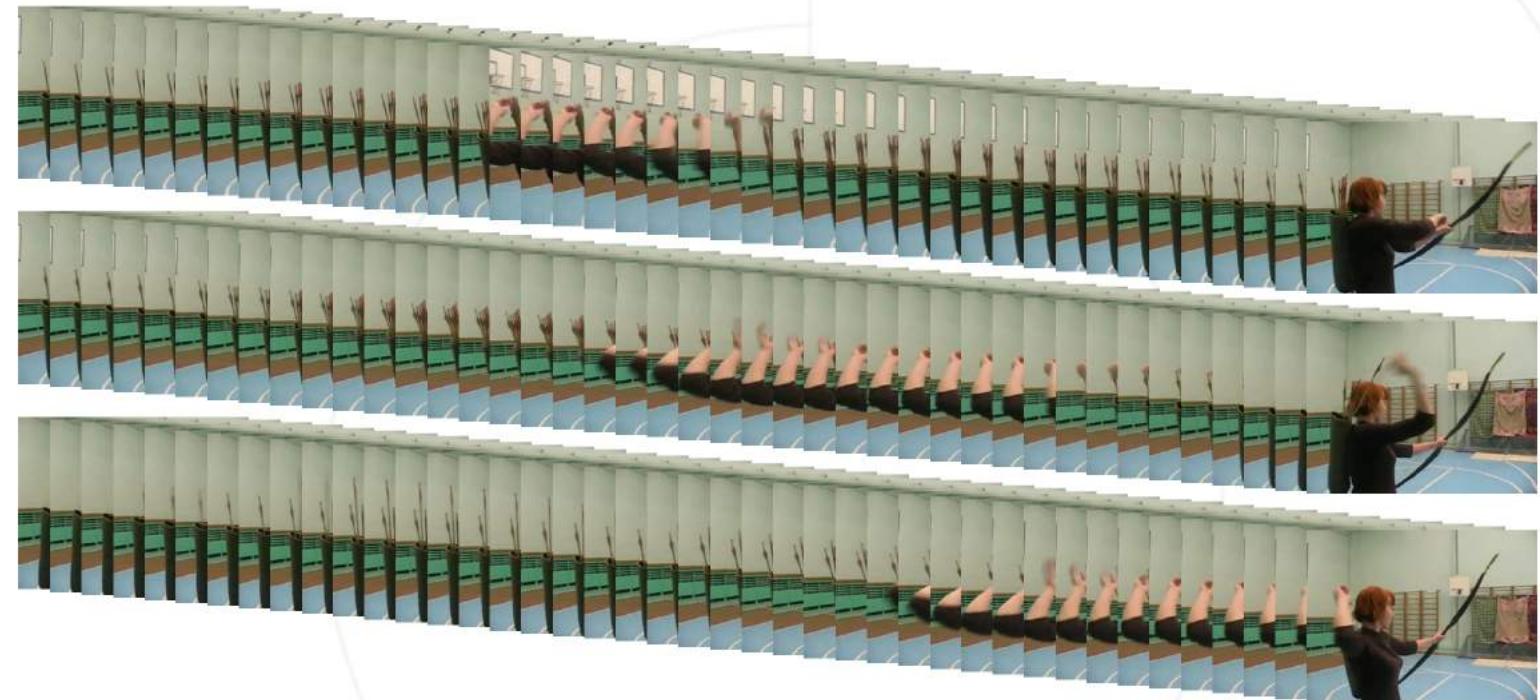


挑战2：如何高效处理视频数据

视频的数据量远大于图像，在计算方面提出了挑战



=

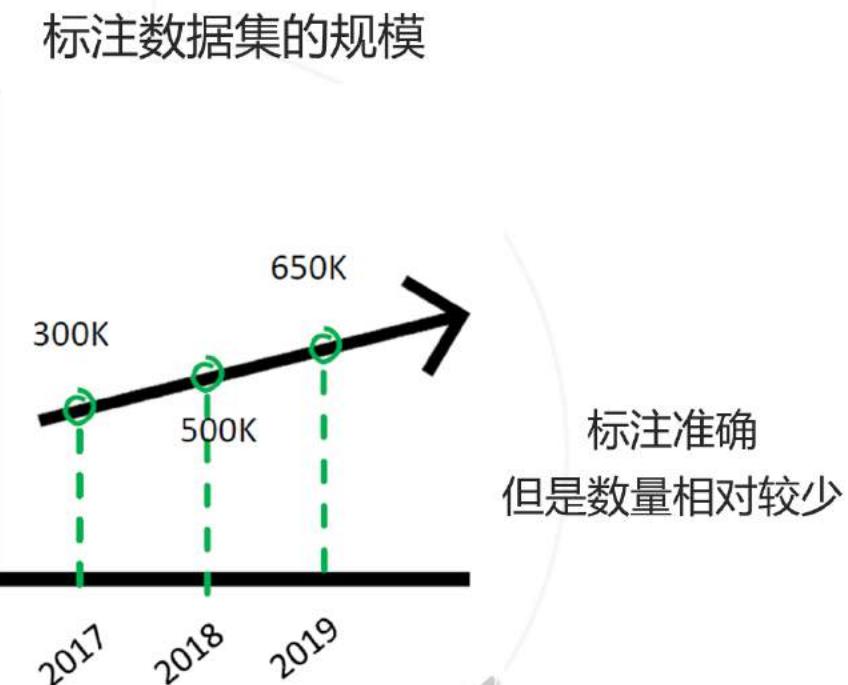
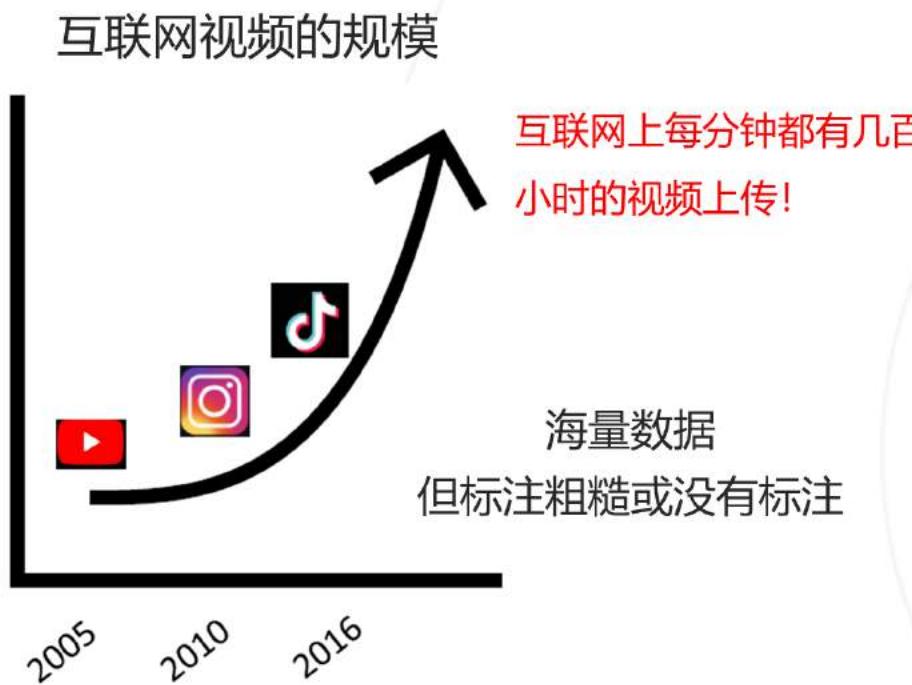


5 秒短视频

150+ 张图像

挑战3：如何有效利用互联网上的海量视频

视频数据的标注是昂贵的；网络上有海量的视频数据，能否有效利用这些数据训练模型？

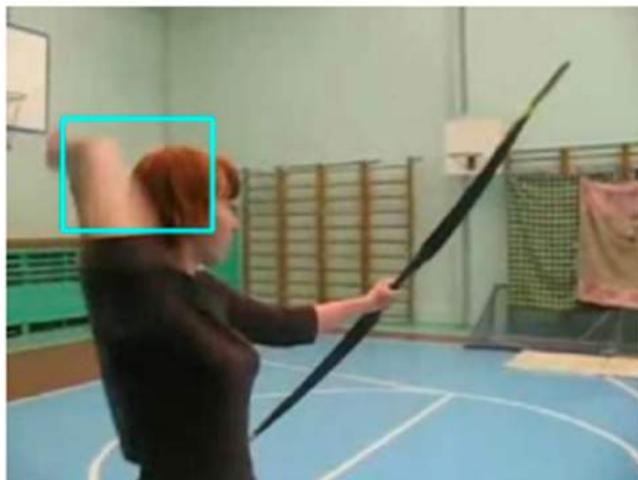


例：IG-65M 数据集从网络爬取，包含6500万视频，在规模上百倍于严谨标注的学术数据集

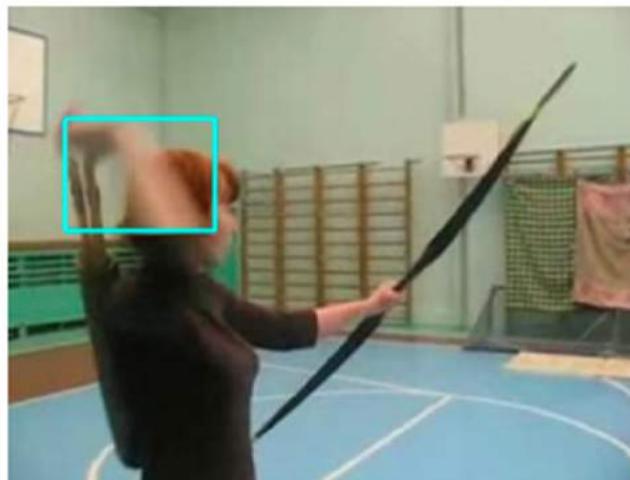
光流：捕捉视频中的运动

光流是视频中**物体运动**的描述，是图像平面上的向量场

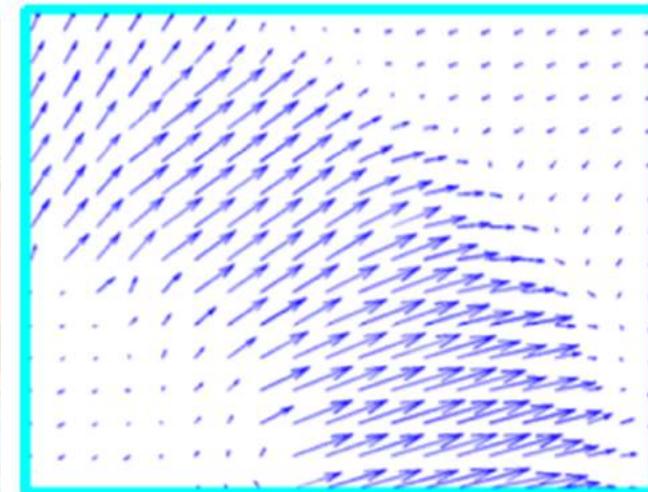
光流通常基于相邻图像帧进行估计得到



t 时刻

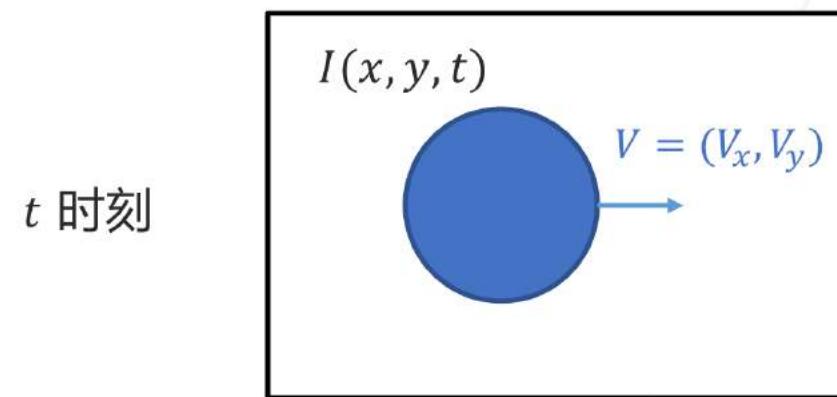


$t + \Delta t$ 时刻



光流
手臂的运动

t 时刻物体位于 (x, y) 位置, $t + \Delta t$ 时刻移动到了 $(x + \Delta x, y + \Delta y)$ 位置

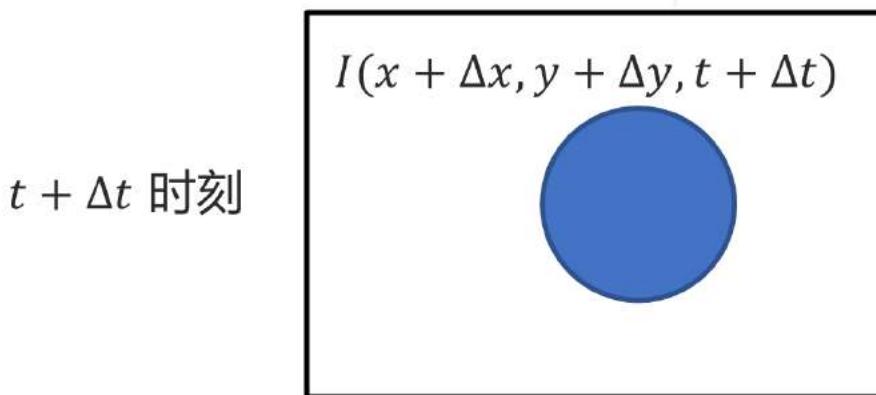


一阶近似

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t)$$

$$\frac{\partial I}{\partial x} \Delta x + \frac{\partial I}{\partial y} \Delta y + \frac{\partial I}{\partial t} \Delta t = 0$$

除以 Δt 再取极限



$$\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y = - \frac{\partial I}{\partial t}$$

三个偏导数可以根据相邻图像帧计算出

一个方程两个未知数, 欠定问题, 也称为孔径问题 (aperture problem)

- 需要额外约束求解

□ 两幅灰度图

分别表示X、Y分量
亮度越大幅度越大



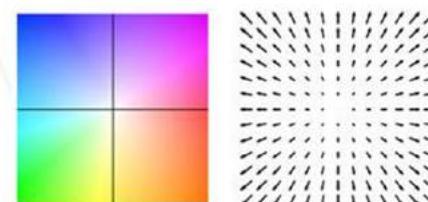
□ 向量场

常用于稀疏光流



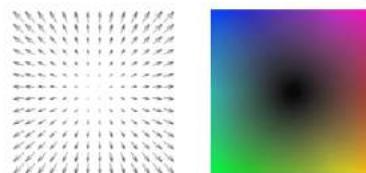
□ 单幅彩色图

用亮度表示速度大小
用色彩表示速度方向



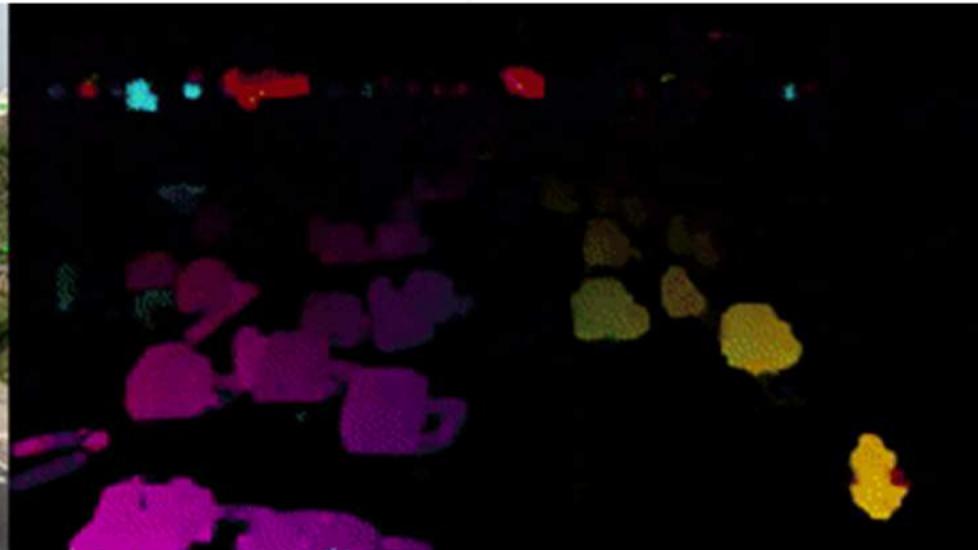
光流的两种类型

颜色表示方向
亮度表示大小



稀疏光流

跟踪少量感兴趣点



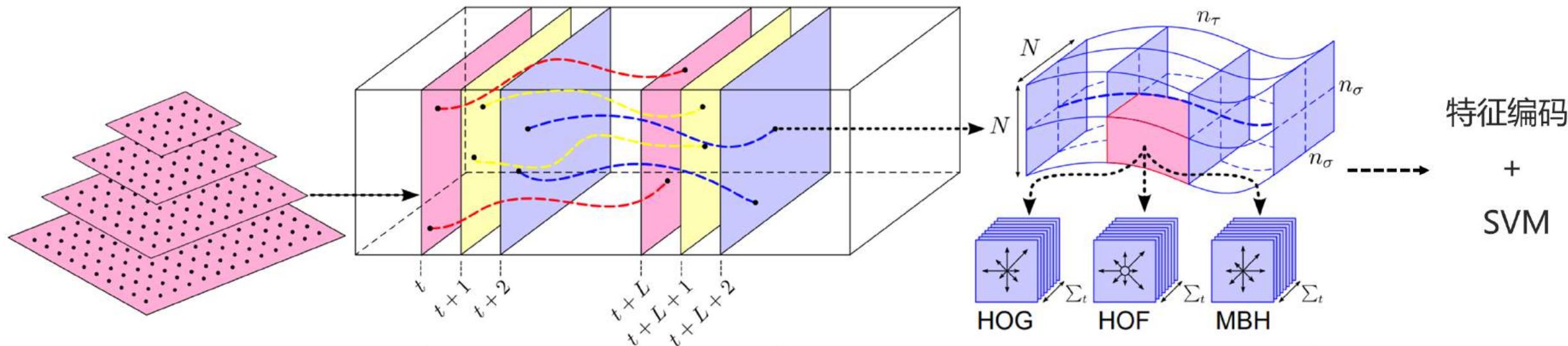
稠密光流

估算所有像素的光流

Dense Trajectories 是手工设计的视频特征，是深度学习前时代的主流方法

基本思路：

- 使用光流估计物体的运动轨迹，再在轨迹的时空邻域内统计图像和光流的直方图，作为动作特征



1. 多尺度密集选取
兴趣点

2. 以兴趣点为起点，根据光流追踪物体
在 $L=15$ 帧内的运动轨迹

3. 在轨迹的时空邻域，计
算不同的图像和光流特征

4. 对特征进行编码，训练
分类器完成分类任务

1. 在 8 个空间尺度上，每隔 5 像素设置一个兴趣点
2. 移除平滑区域的兴趣点，因为这些位置没有运动



第一个尺度上的稠密采样示例

- 对于每个兴趣点，利用光流估计该点在下一帧的位置

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + \boxed{(M * \omega_t)}|_{(x_t, y_t)}$$

下一帧位置 当前位置 3×3 邻域内 光流的中值

序列 $(P_t, P_{t+1}, P_{t+2}, \dots)$ 构成该兴趣点的运动轨迹

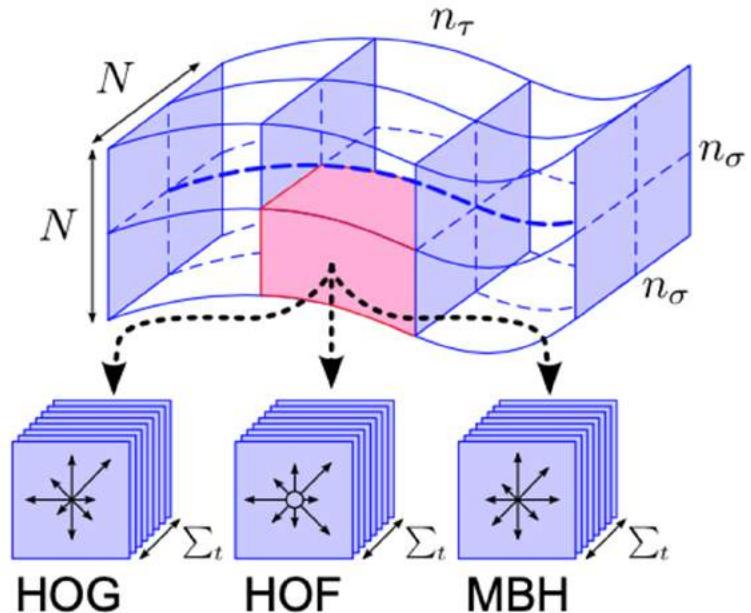
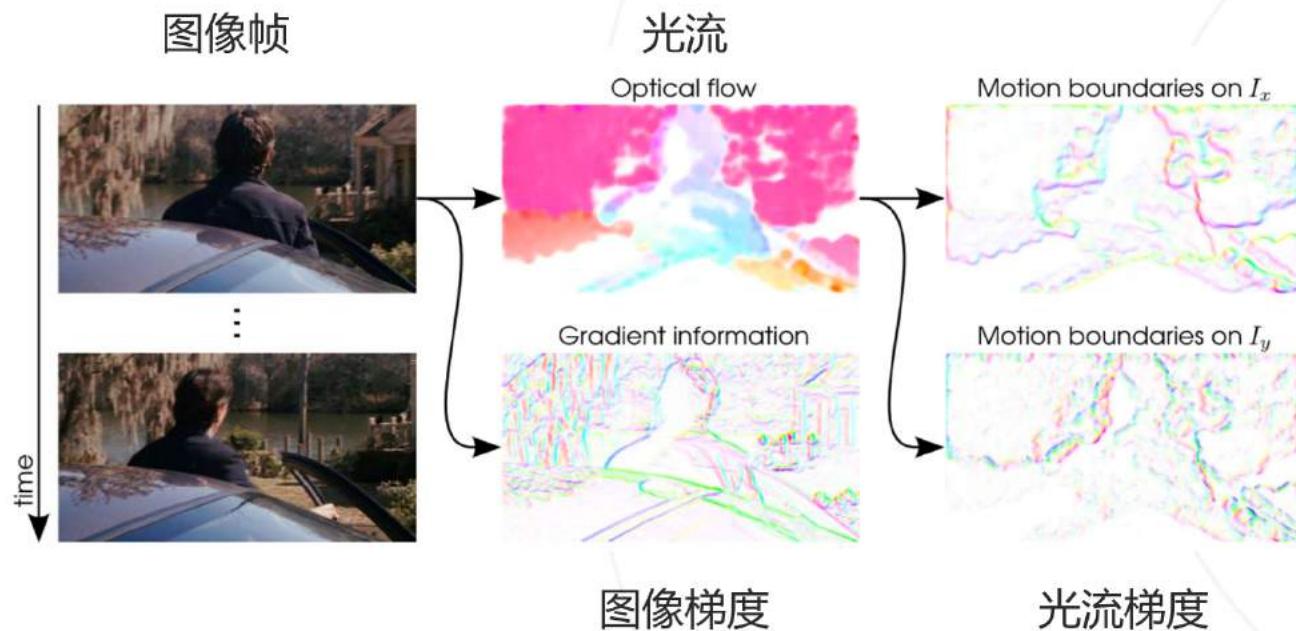
- 归一化帧间位移，得到轨迹的**形状特征**， $2L=30$ 维度

$$\Delta P_t = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} - y_t)$$

$$T = \frac{(\Delta P_t, \dots, \Delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} \|\Delta P_j\|}$$

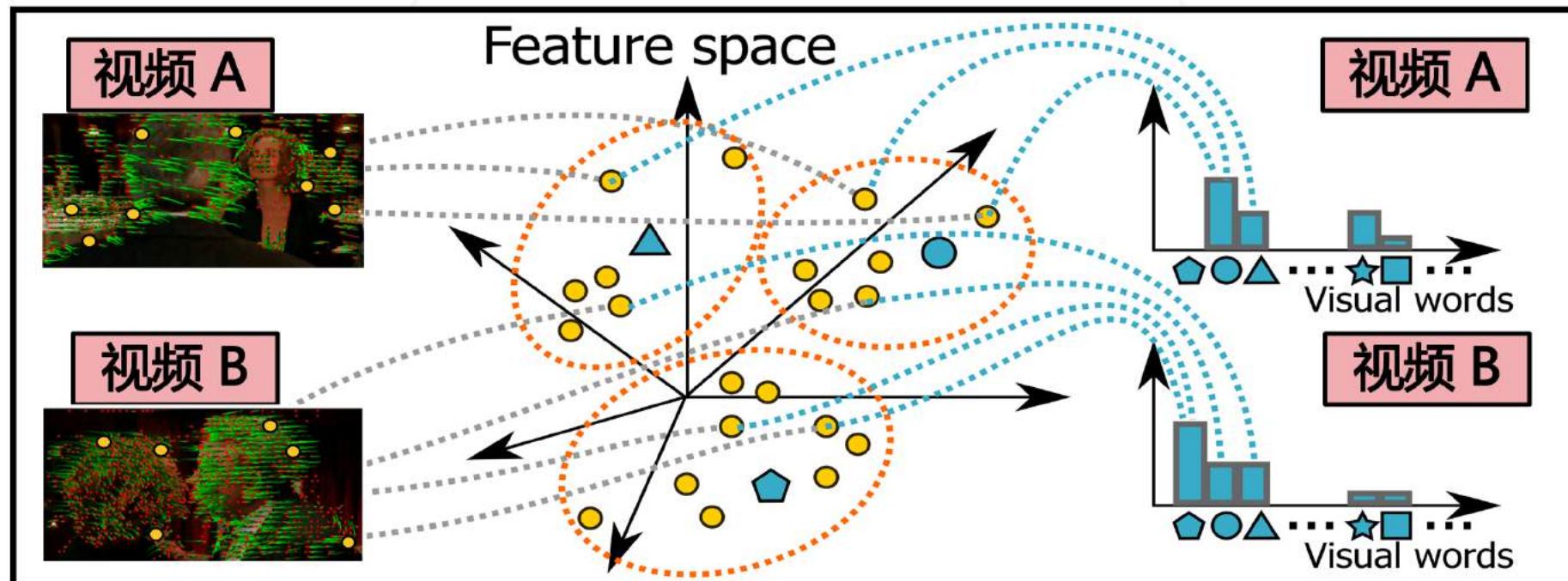
按轨迹对齐的局部特征

- 取轨迹周围 $N \times N \times L$ 的时空区域，切分成 $n_\sigma \times n_\sigma \times n_r$ 块；
- 在每个时空块内计算局部图像或视频特征；
- 拼接所有区块的局部特征，作为该轨迹的局部特征，共 426 维；



- HOG：图像的梯度的直方图
- HOF：光流的直方图
- MBH：光流的梯度的直方图

- 每个视频中包含数万个轨迹（且数目不定），每个轨迹包含 30 维轨迹特征和 426 维局部特征
- 使用**特征词袋** (bag of visual feature)，得到视频的运动特征



每个图像包含不
定数目的特征点

在特征空间聚类
聚类中心定义为**视觉词语**

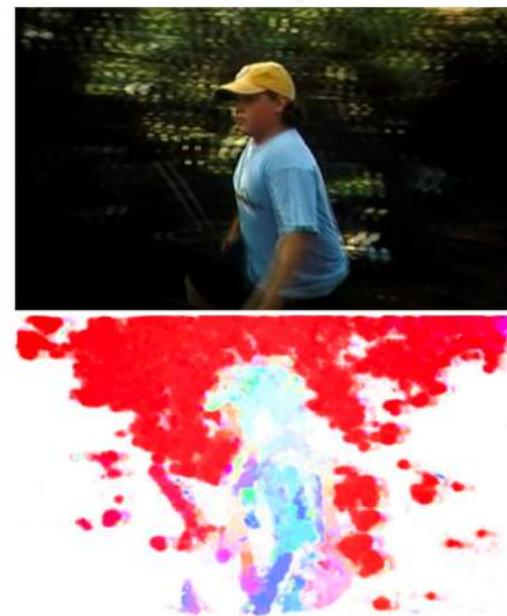
统计视频中视觉词语的词频
该直方图为图像的特征

iDT (improved dense trajectory) 在 Dense Trajectory 的基础上进行了一些改进，在行为识别等任务上可以取得更好的效果

- 增加相机运动估计，并在光流和轨迹中移除相机运动的影响
- 检测人体，只是用人体以外的区域估计相机运动

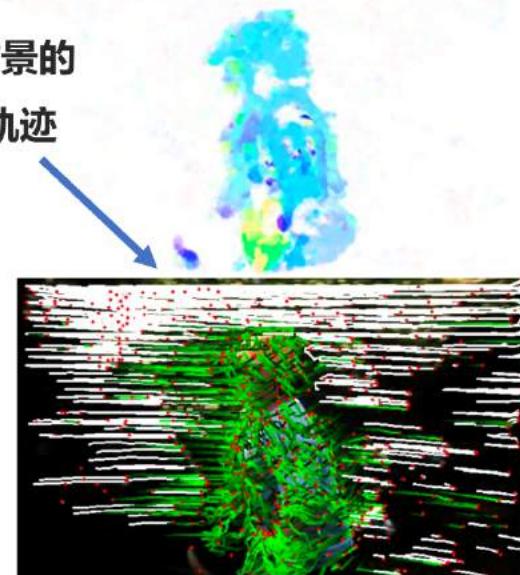
	HMDB51	UCF50
DT	46.6%	84.5%
iDT	57.2%	91.2%

分类精度提升



原始光流包含背景的运动

去除背景的
运动轨迹

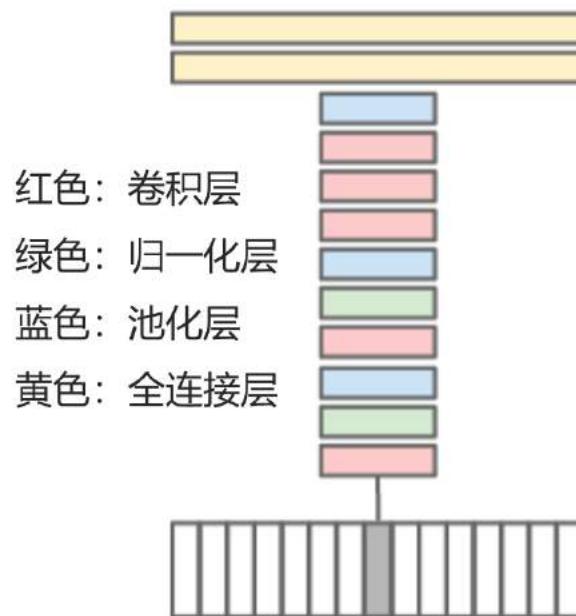


去除背景的运动轨迹

深度学习下的视频理解

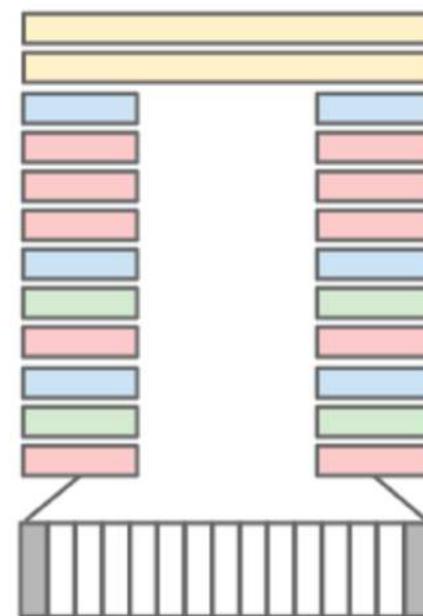
首次使用深度学习解决视频分类任务，但性能不如 iDT 等手工设计的特征

基本思路：使用 AlexNet 提取图像帧的特征，再通过不同方法融合不同图像帧的特征



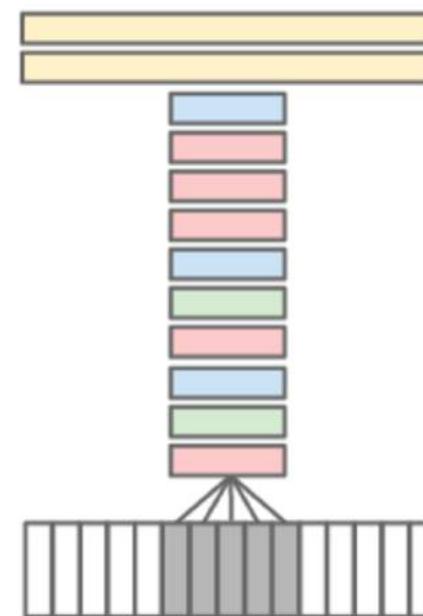
单帧

只有外观特征



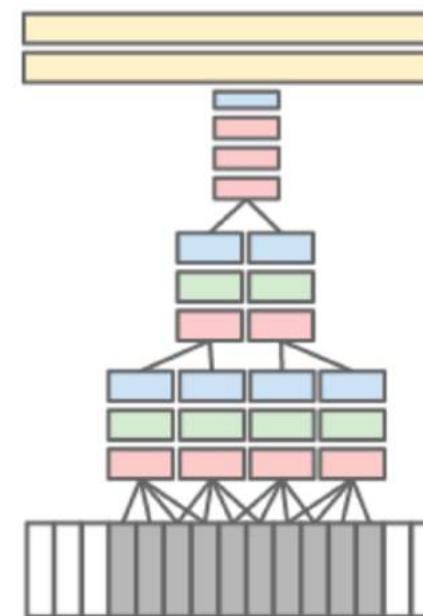
晚融合

在网络高层融合不同
帧的特征



早融合

在网络低层融合不同
帧的特征

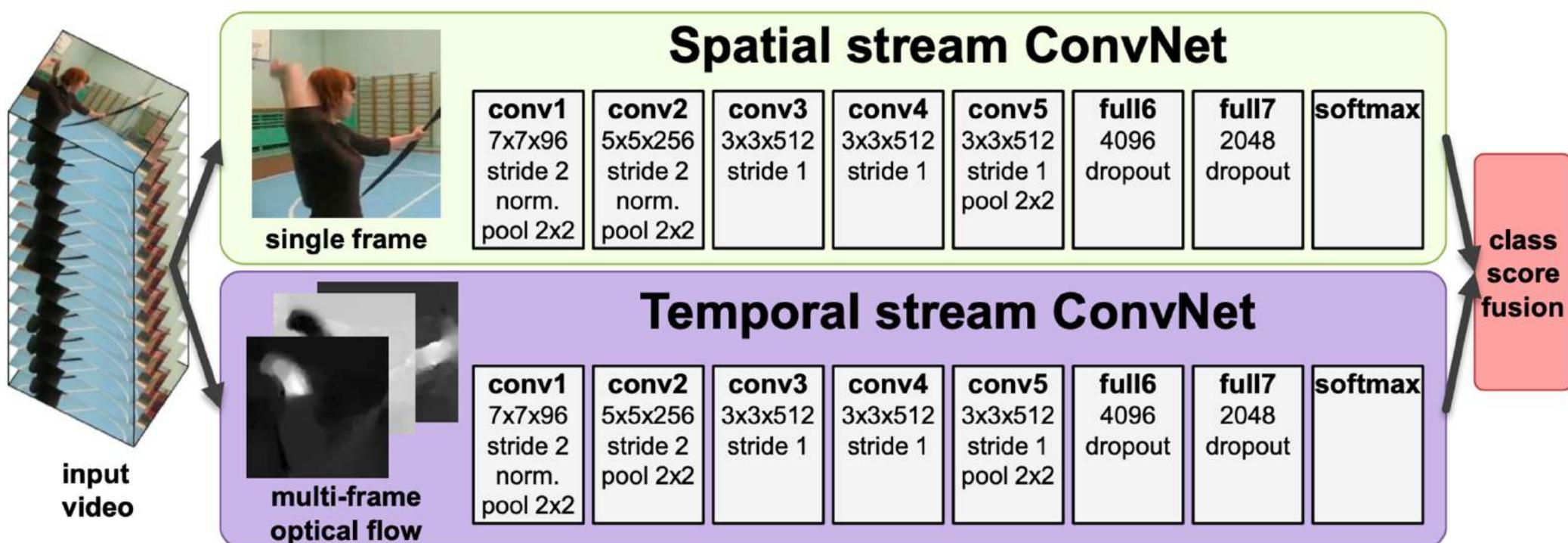


慢融合

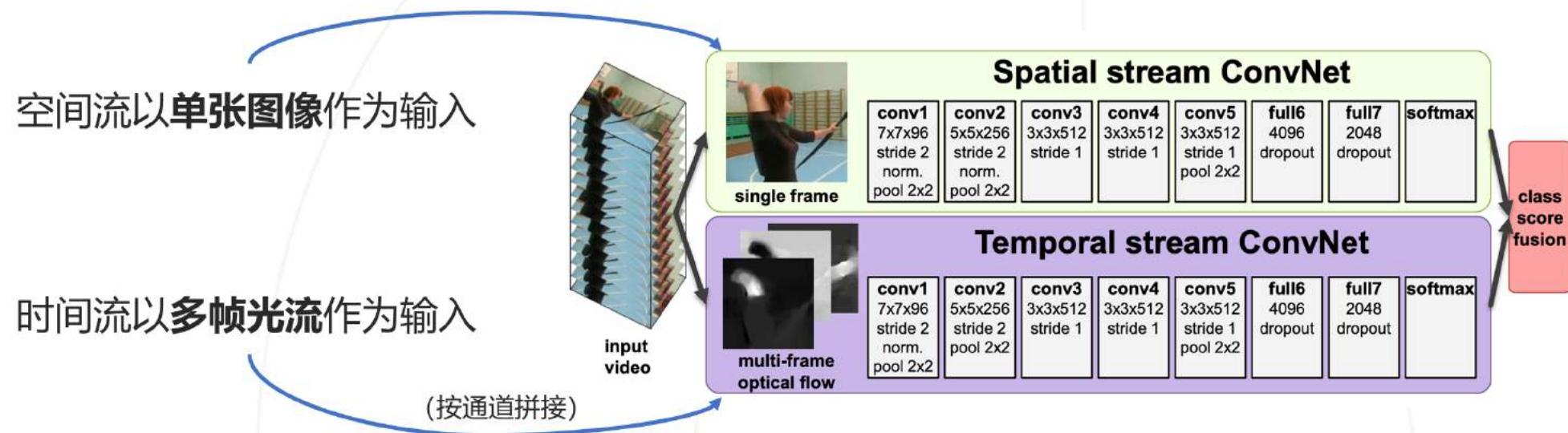
分级融合不同帧的特征

显式使用运动信息（光流）：

- 使用两个神经网络，一个基于图像计算外观特征，另一个基于光流计算运动特征
- 平均两个网络的分类概率作为最终结果



双流模型将运动建模为单帧图像和多帧光流的函数



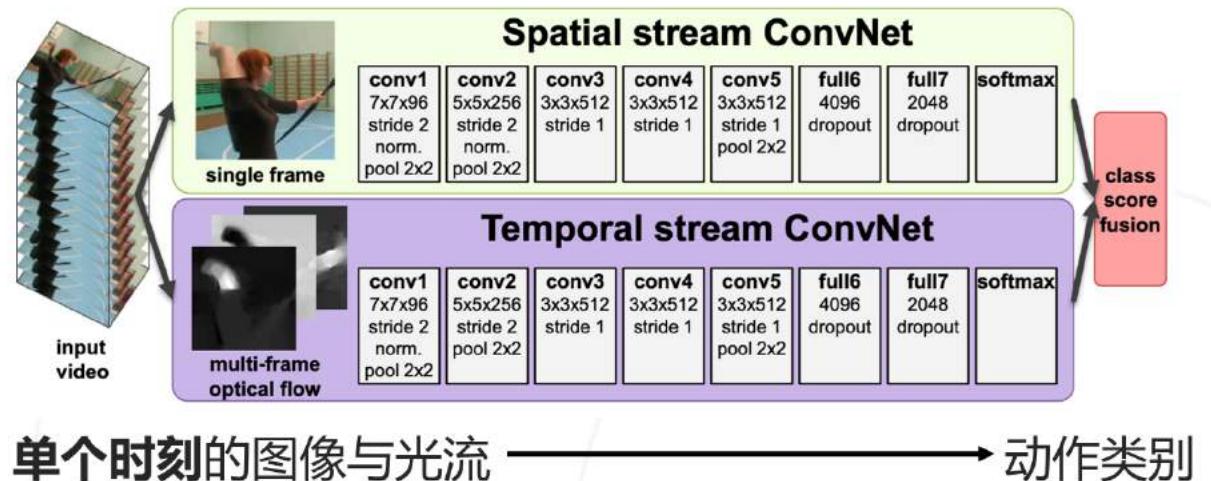
训练时：在视频中随机选取一个时刻，前传对应时刻的图像和光流

测试时：在全部时刻进行预测，再平均所有时刻的分类概率

	HMDB51	UCF101
iDT	57.2%	85.9%
双流网络	59.4%	88.0%

性能超过 iDT 传统方法

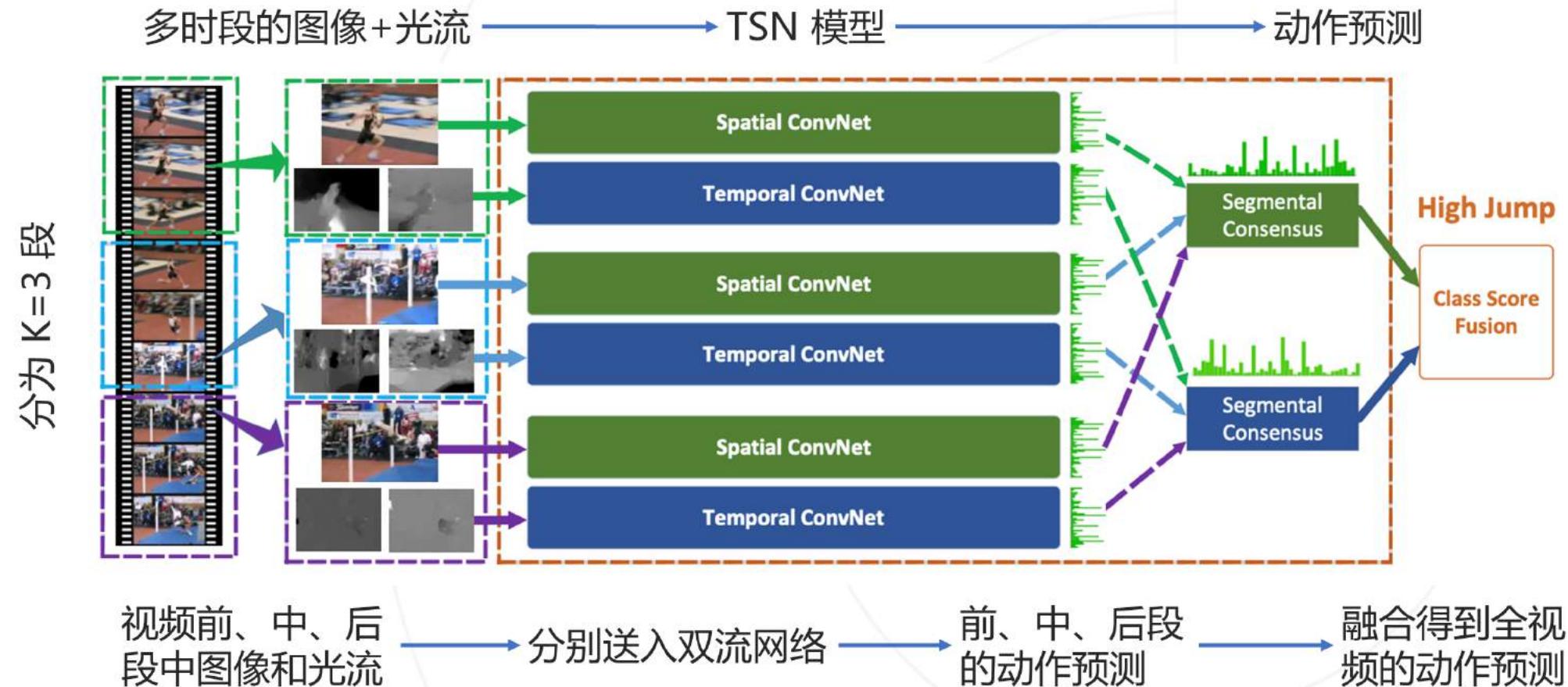
问题：双流网络只有短时建模，动作由单一时刻的图像和光流所确定

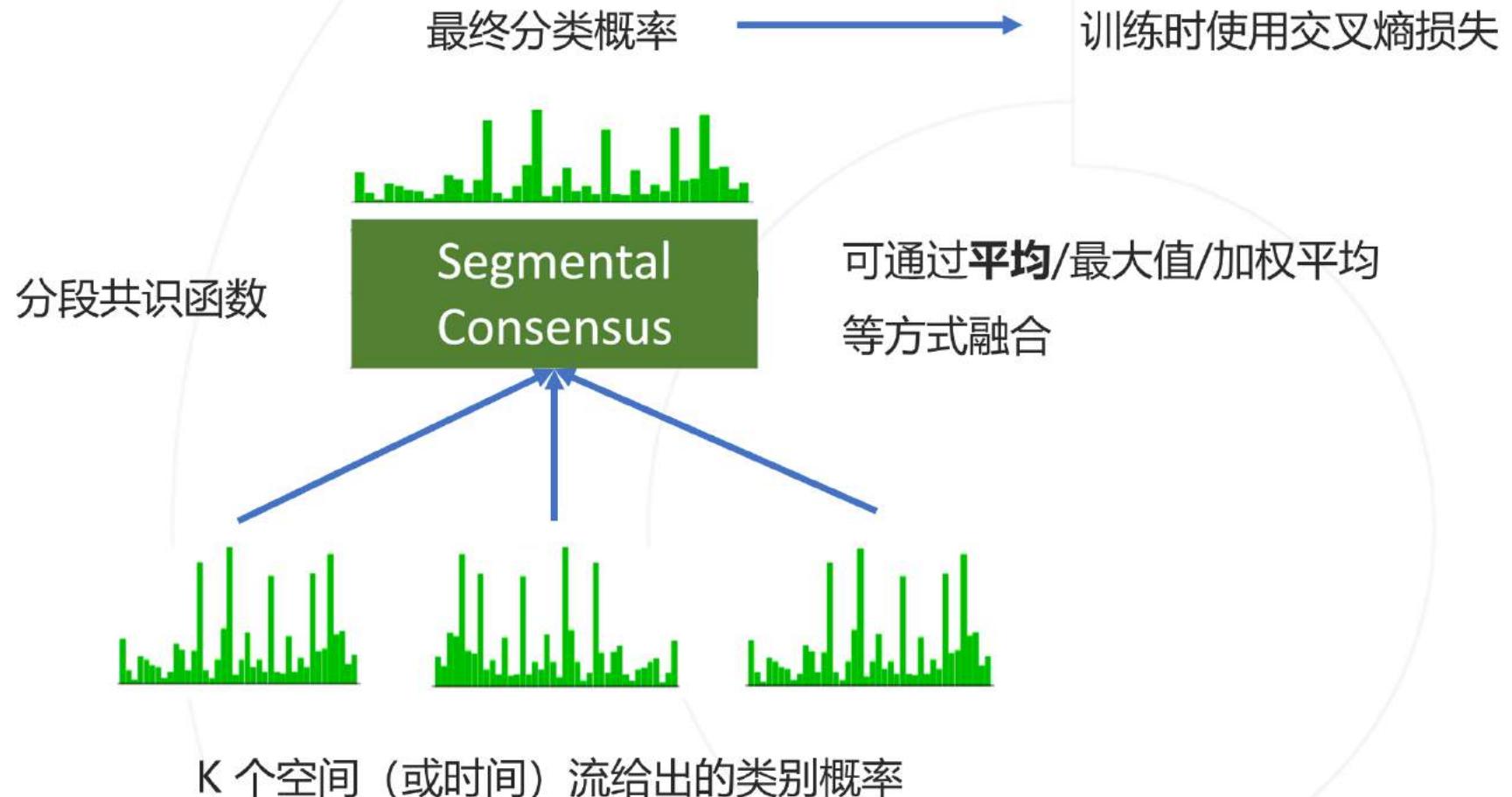


例：跳远视频的开头是助跑，训练时如果采样时刻恰好在开头，则使用助跑的视频与跳远的标签训练了一次迭代

更好的方案：长时建模，动作应由整个时间段内的图像和运动信息所确定

时序分段网络 (Temporal Segment Networks) 将动作建模为视频中多个时段的图像与光流的函数
→ 显式长时建模





相比Two-Stream, TSN 探索了更多的输入配置

- 空间流输入增加了 RGB 差值图像
- 时间流输入参照 iDT, 增加了去除运动估计的光流图



RGB图像

RGB差值图像

光流图像

去除运动估计的光流图像

不同配置在UCF-101上的分类精度

	输入配置	精度
空间流	RGB图像	84.5%
	RGB差值	83.8%
时间流	RGB图像+差值	87.3%
	光流	87.2%
	去除运动估计的光流	86.9%
	光流以及 去除运动估计的光流	87.8%

相比双流神经网络，TSN 在两个数据集上的表现都有大幅提升

模型	HMDB51	UCF101
Two-stream	59.4%	88.0%
TSN(RGB+Flow)	68.5%	94.0%
TSN(RGB+Flow+Warped Flow)	69.4%	94.2%

3D 卷积网络

相邻视频帧

$T: T + t$



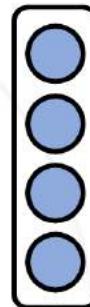
相邻光流帧

估计光流



时间流网络

T 时刻
运动特征



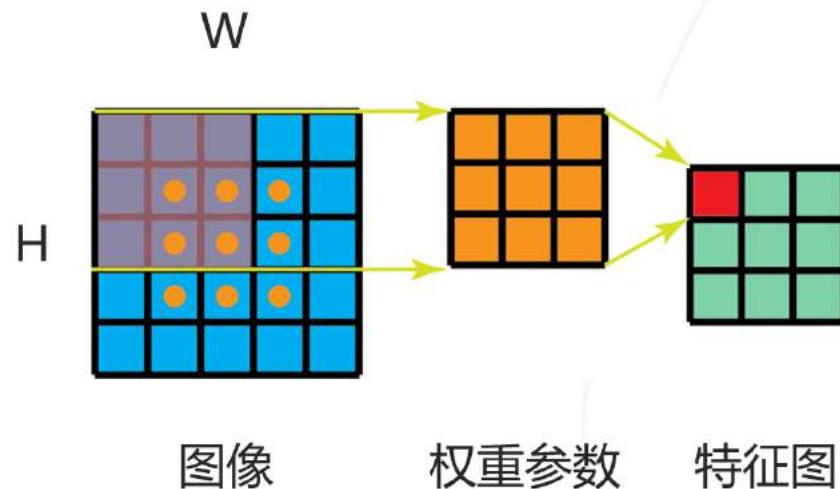
(t, h, w, c)

形状的数组

直接从多帧图像计算出特征?

2D 卷积

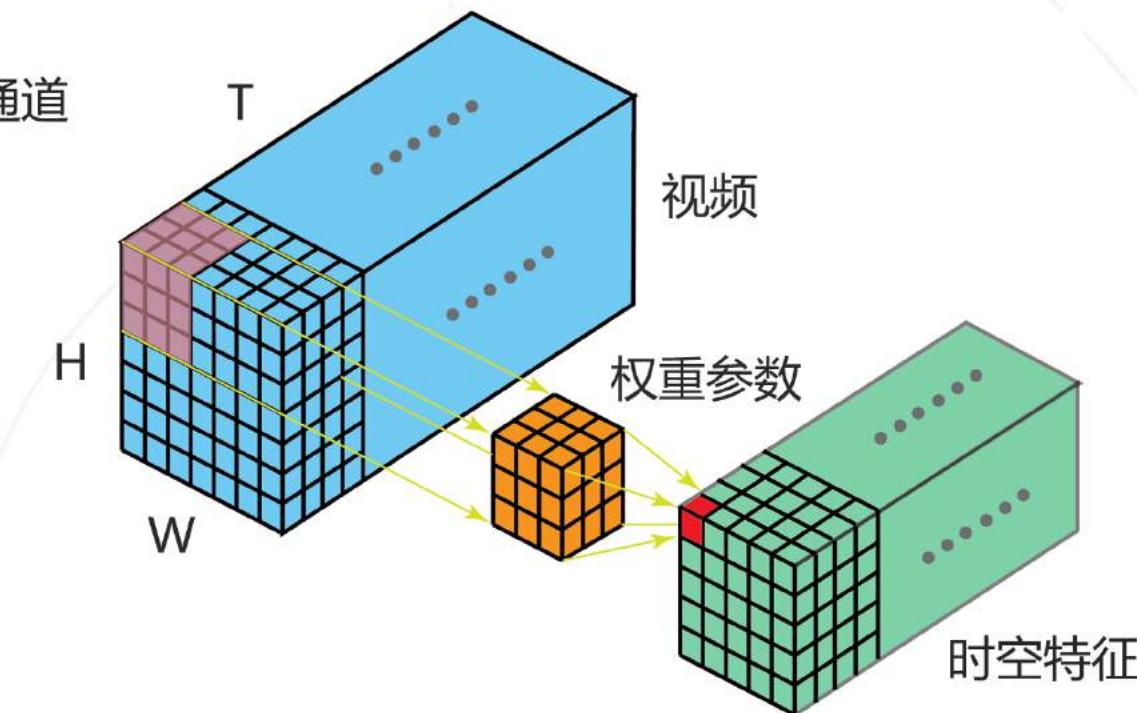
单通道



空间上相邻的像素，经过加权求和，
得到对应位置的特征

3D 卷积

单通道



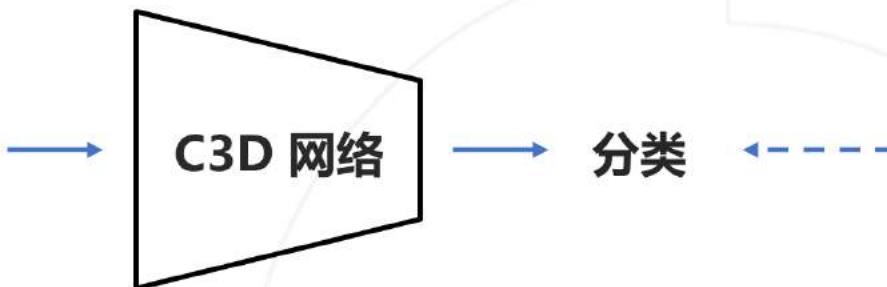
空间和时间上相邻的像素，经过加权求和，
得到对应位置和时刻的特征

早期使用 3D 卷积网络进行视频分类的模型

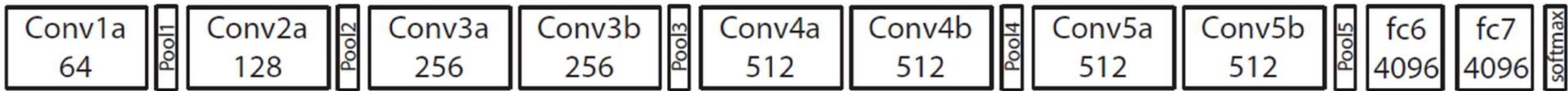
连续多帧图像



训练时从视频中随机抽取



使用 Sports-1M 数据集训练
包含 1.1M 视频, 487 类别



自主设计的网络结构，包含 8 个三维卷积层，5 个三维池化层，2 个全连接层

卷积层使用 $3 \times 3 \times 3$ 大小的卷积核，池化基于 $1 \times 2 \times 2$ 或 $2 \times 2 \times 2$ 的局部区域

一般情况下，C3D 的性能不如双流神经网络，需要额外配合一些技术才可以超过双流网络

一些原因：

- 三维卷积网络参数更多、更难训练
- 自主设计结构、从头训练，不能利用成功预训练的图像分类模型

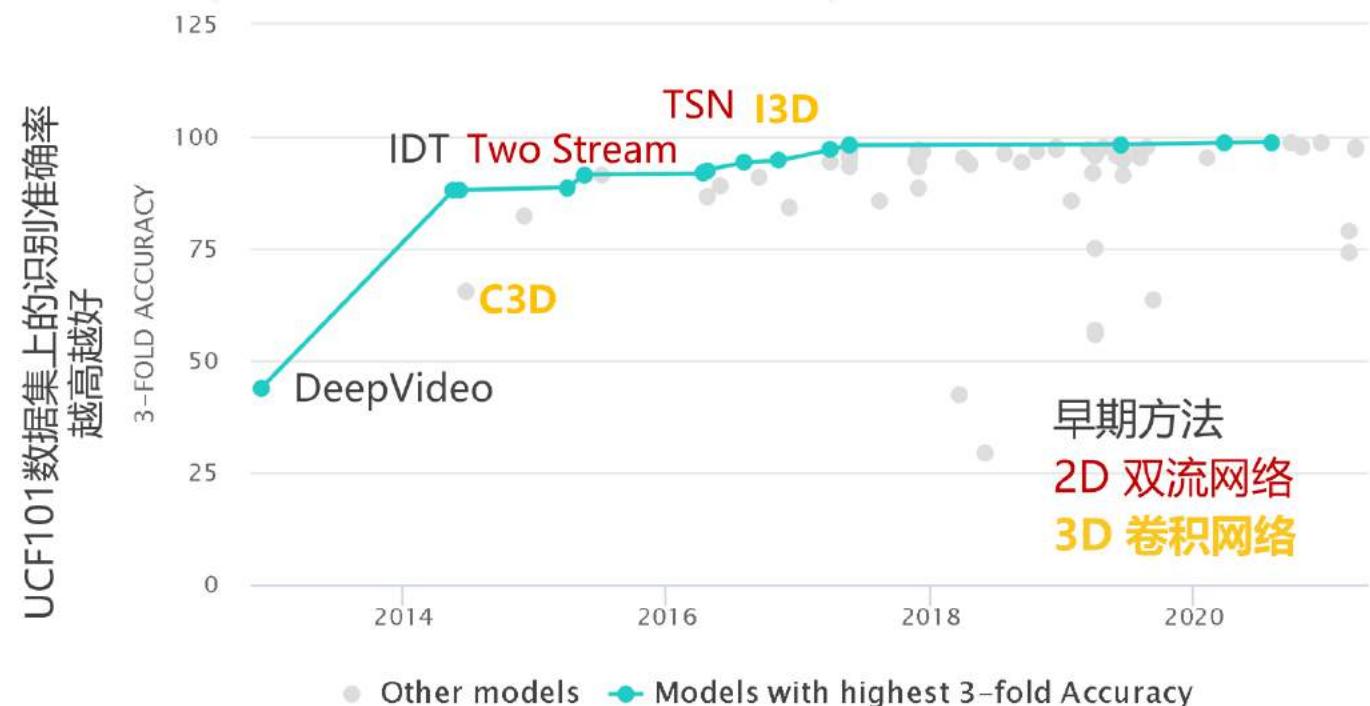
	算法	精度
只使用图像	空间流网络	72.6 %
	C3D (单网络)	82.3 %
	C3D (三网络)	85.2 %
图像+光流	双流神经网络	88.0 %
	C3D (三网络) + iDT	90.4 %

UCF101 数据集上的分类精度

2017年，DeepMind 提出 I3D 模型；

与 C3D 不同，I3D 中的 3D 网络由图像分类的 2D 网络“膨胀”得来，从而可以充分利用已有的图像分类模型；

I3D 的性能全面超越 C3D 以及同时期的 2D 卷积网络方法，自此 3D 卷积网络逐渐成为主流。



2D 卷积

单通道
单卷积核

i ₁₁	i ₁₂	i ₁₃	i ₁₄	i ₁₅
i ₂₁	i ₂₂	i ₂₃	i ₂₄	i ₂₅
i ₃₁	i ₃₂	i ₃₃	i ₃₄	i ₃₅
i ₄₁	i ₄₂	i ₄₃	i ₄₄	i ₄₅
i ₅₁	i ₅₂	i ₅₃	i ₅₄	i ₅₅

*

w ₁₁	w ₁₂	w ₁₃
w ₂₁	w ₂₂	w ₂₃
w ₃₁	w ₃₂	w ₃₃

=

o ₁₁	o ₁₂	o ₁₃
o ₂₁	o ₂₂	o ₂₃
o ₃₁	o ₃₂	o ₃₃

在时间维度堆叠

3D 卷积

单通道
单卷积核

⋮	⋮	⋮	⋮	⋮	⋮	5
i ₁₁	i ₁₂	i ₁₃	i ₁₄	i ₁₅	⋮	5
i ₂₁	i ₂₂	i ₂₃	i ₂₄	i ₂₅	⋮	5
i ₃₁	i ₃₂	i ₃₃	i ₃₄	i ₃₅	⋮	5
i ₄₁	i ₄₂	i ₄₃	i ₄₄	i ₄₅	⋮	5
i ₅₁	i ₅₂	i ₅₃	i ₅₄	i ₅₅	⋮	5

*

在时间维度堆叠

 $\frac{1}{3}$

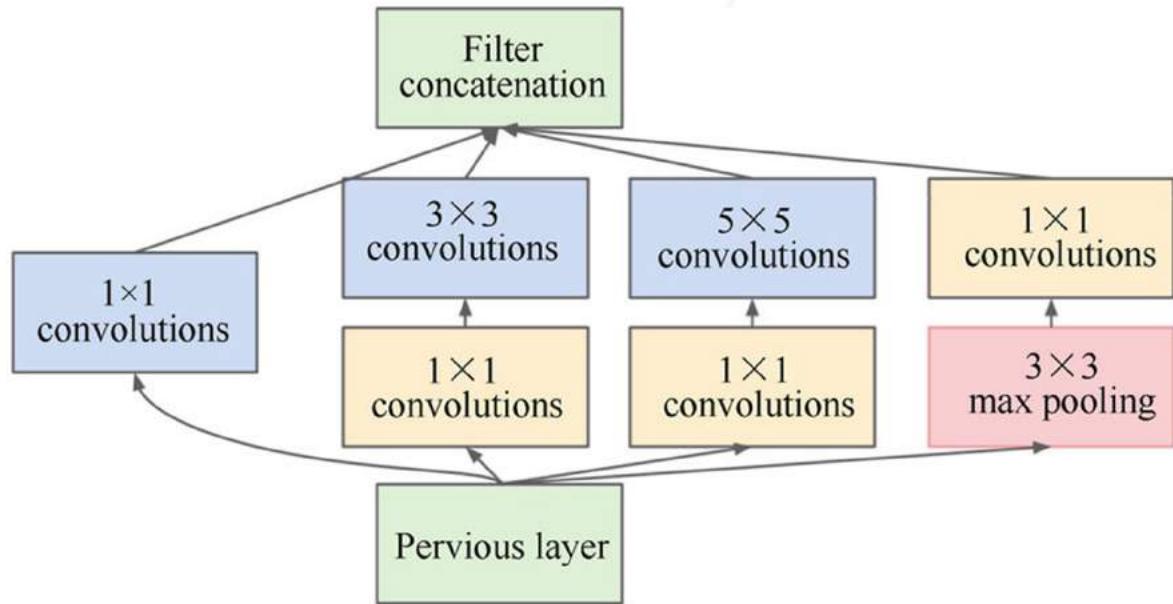
w ₁₁	w ₁₂	w ₁₃	⋮	⋮	⋮	13
w ₂₁	w ₂₂	w ₂₃	⋮	⋮	⋮	23
w ₃₁	w ₃₂	w ₃₃	⋮	⋮	⋮	33

=

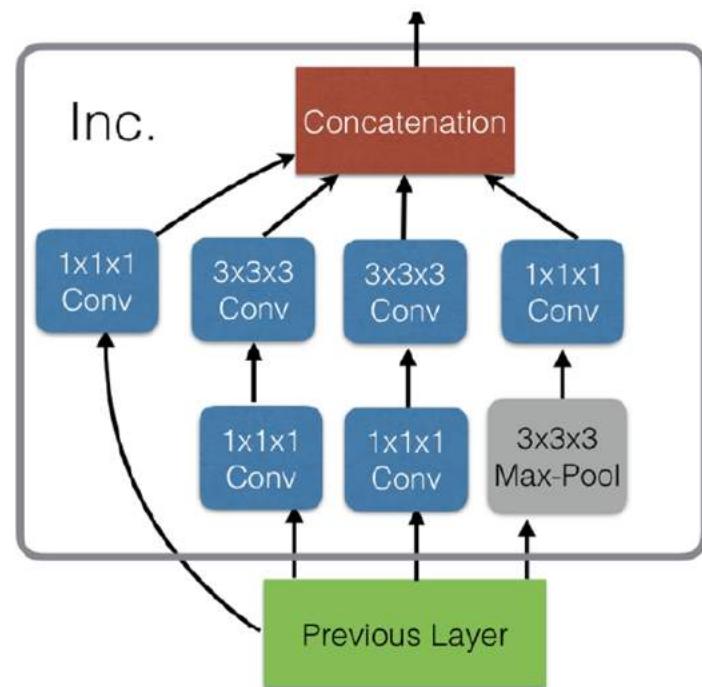
o ₁₁	o ₁₂	o ₁₃
o ₂₁	o ₂₂	o ₂₃
o ₃₁	o ₃₂	o ₃₃

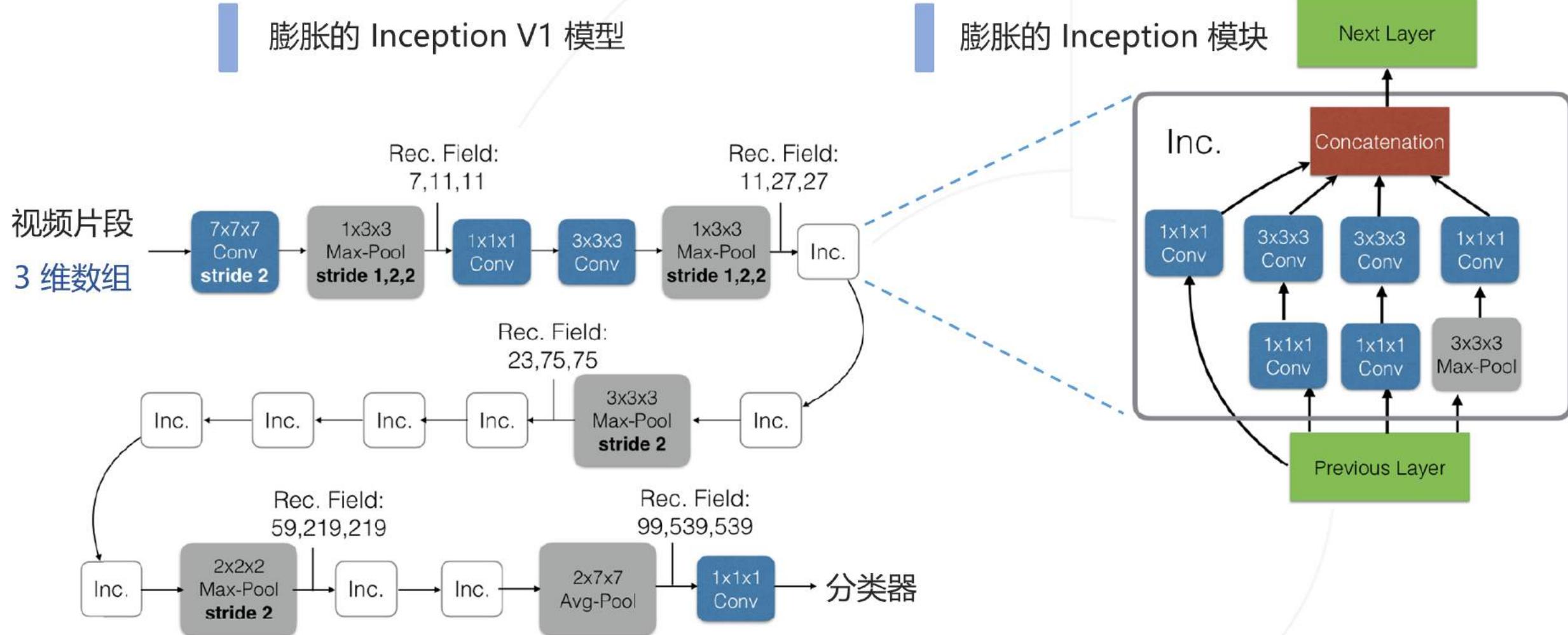
相同的结果

原始的 Inception 模块由 2D
卷积和 2D 池化构成

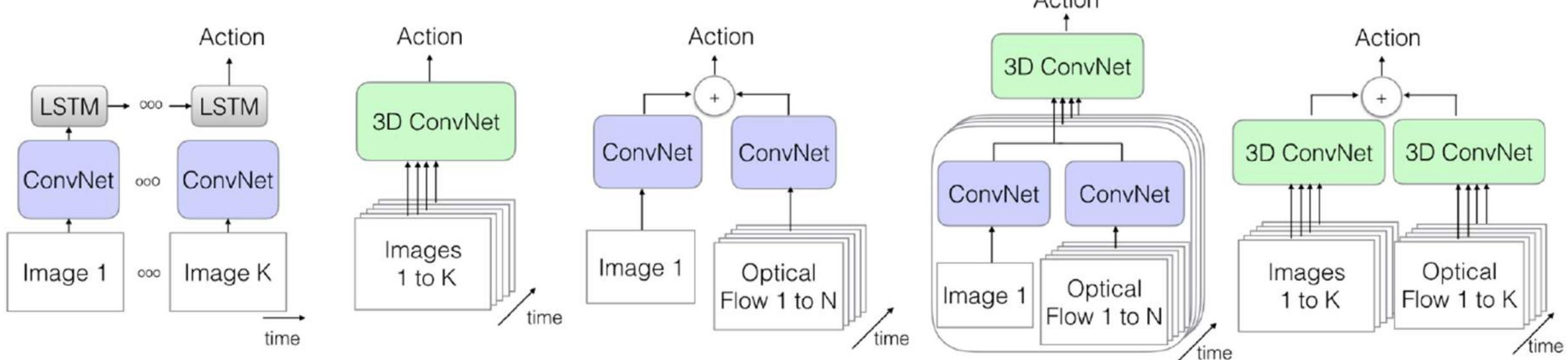


将 2D 卷积和池化膨胀
为 3D 卷积和池化





与其他方法的对比



(a) 类 DeepVideo 模型

(b) C3D 模型

(c) 双流神经网络

(d) 一种 2/3D 混合模型

(e) 双流 I3D

单纯基于空间流
但没有使用膨胀卷积核

只使用空间流就可以达到很好的效果
加入时间流可以进一步提升

在 Kinetics 数据集上对比五种模型

Architecture	Kinetics			ImageNet then Kinetics		
	RGB	Flow	RGB + Flow	RGB	Flow	RGB + Flow
(a) LSTM	53.9	—	—	63.3	—	—
(b) 3D-ConvNet	56.1	—	—	—	—	—
(c) Two-Stream	57.9	49.6	62.8	62.2	52.4	65.6
(d) 3D-Fused	—	—	62.7	—	—	67.2
(e) Two-Stream I3D	68.4 (88.0)	61.5 (83.4)	71.6 (90.0)	71.1 (89.3)	63.4 (84.9)	74.2 (91.3)

观察：

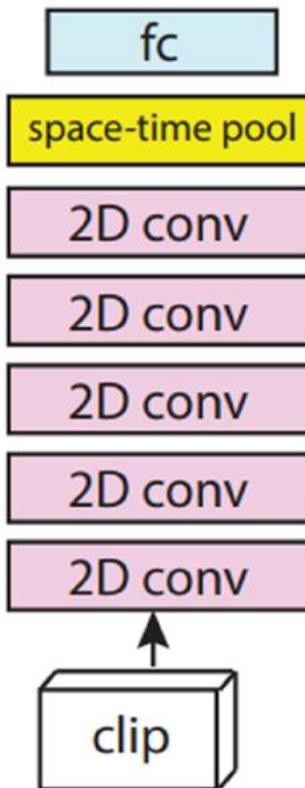
- 单纯基于图像的 I3D 模型已经优于双流神经网络，双流 I3D 性能更高
- 针对所有结构，基于 ImageNet 的预训练有助于提高视频数据集上的训练精度

由于多了时间维度，3D 卷积的参数量和计算量相比于 2D 卷积成倍增加

	2D 卷积	3D 卷积
输入尺寸	$C_{in} \times H \times W$	$C_{in} \times H \times W \times T$
输出尺寸	$C_{out} \times H \times W$	$C_{out} \times H \times W \times T$
卷积核大小	$K_H \times K_W$	$K_H \times K_W \times K_T$
参数量	$C_{in} \times C_{out} \times K_H \times K_W$	$C_{in} \times C_{out} \times K_H \times K_W \times K_T$
计算量	$C_{in} \times C_{out} \times K_H \times K_W \times H \times W$	$C_{in} \times C_{out} \times K_H \times K_W \times K_T \times H \times W \times T$

- 需要更多的训练数据
 - 需要更多的计算资源
- 需要更高效的 3D 卷积模型

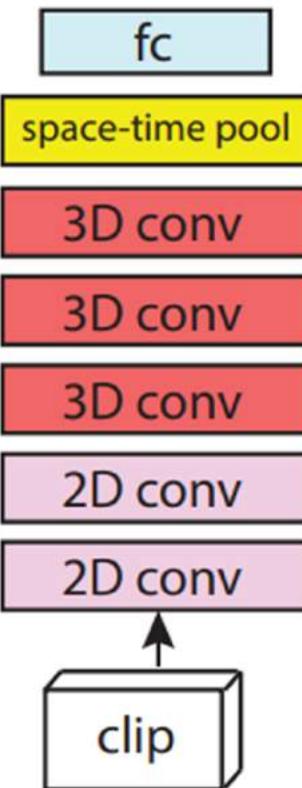
更高效的 3D 卷积网络



全部使用2D卷积
抛弃时间上的关联
参数量降低为 $1/K_T$

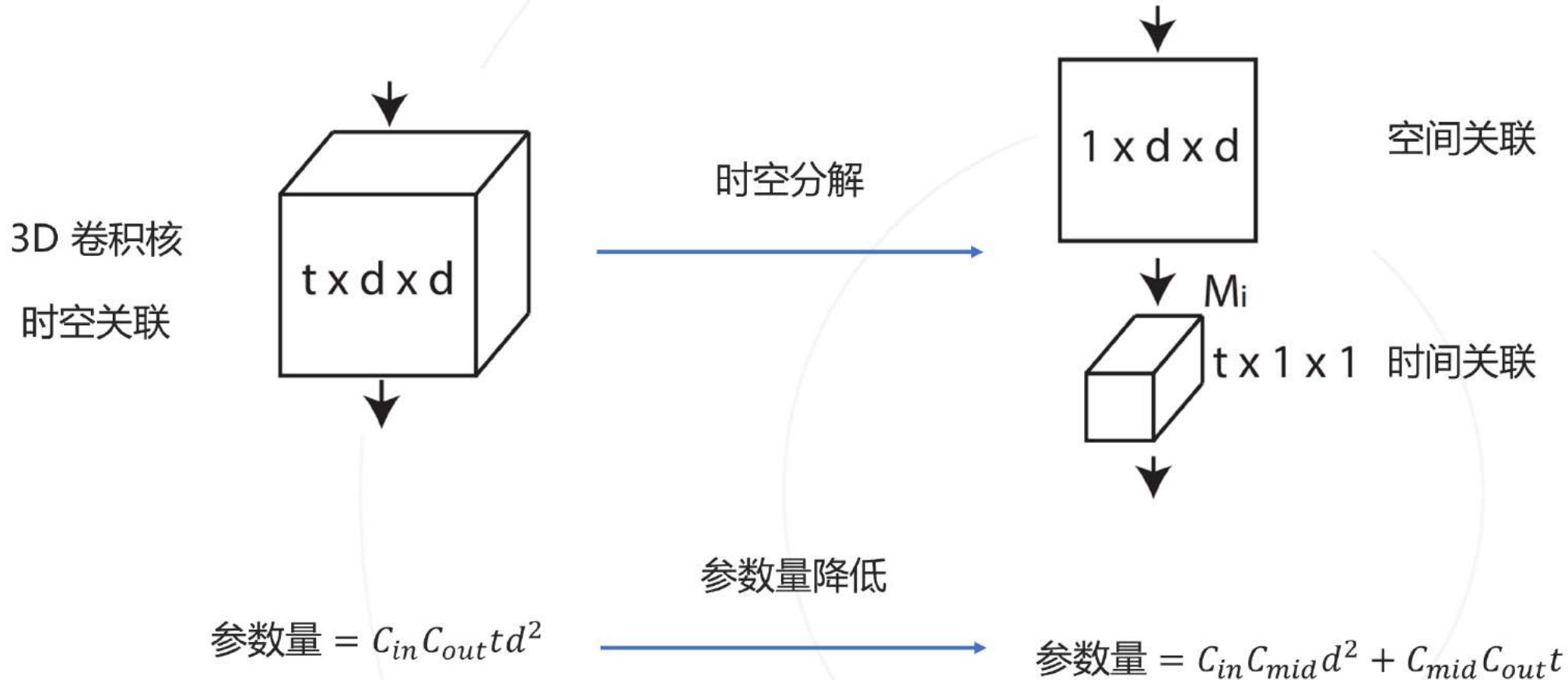


假设时间关联在低层 / 高层更重要
仅在低层 / 高层使用 3D 卷积
参数量居中

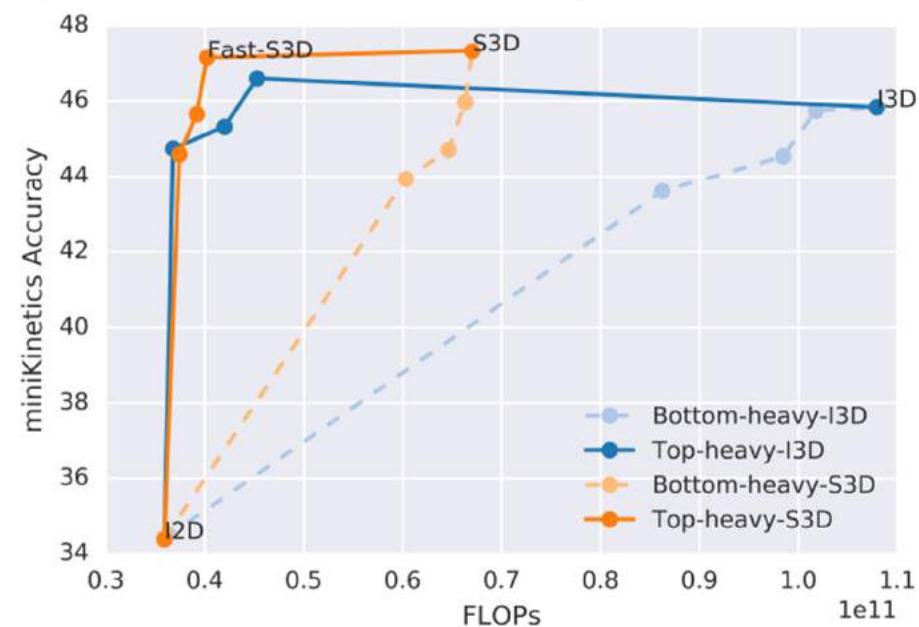
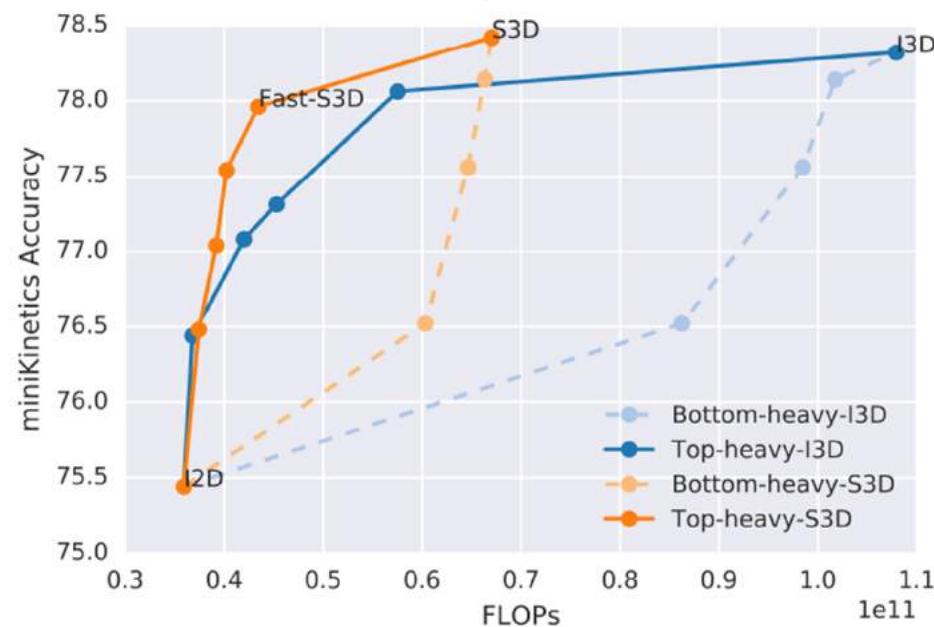


全部使用3D卷积
全部层在时间维度有关联
参数量最大





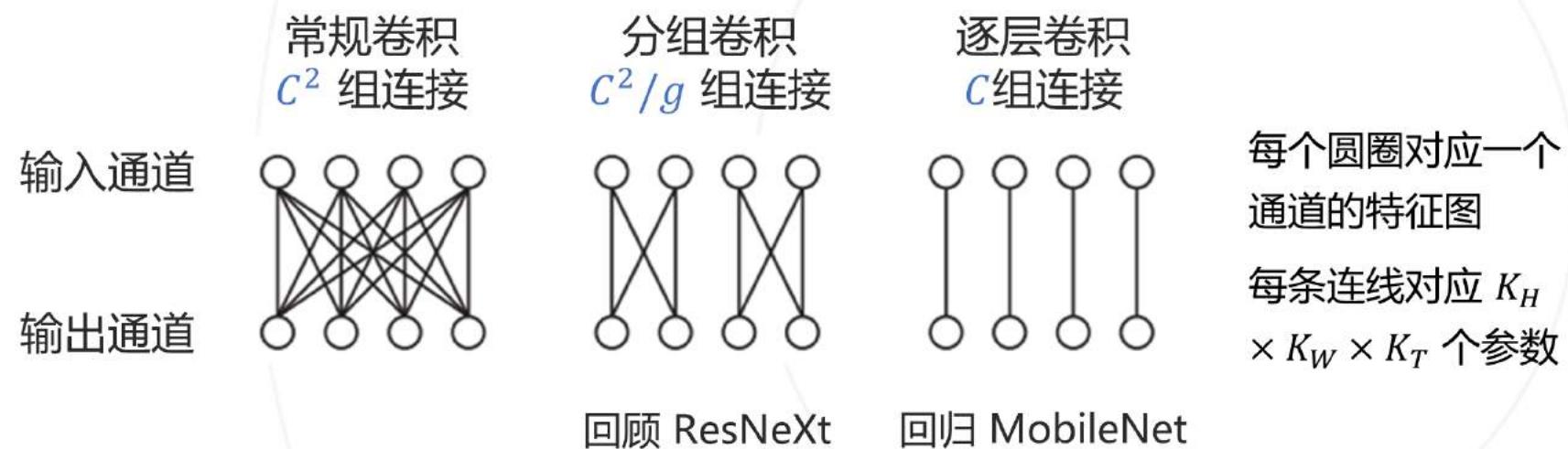
2017年底，DeepMind 提出的 S3D 模型和 FAIR 提出的 R(2+1)D 模型均采用卷积核分解的思路。前者基于 Inception，后者基于 ResNet；
实验表明时空分解的性能和效率优于全 3D 卷积网络和 2/3D 混合网络



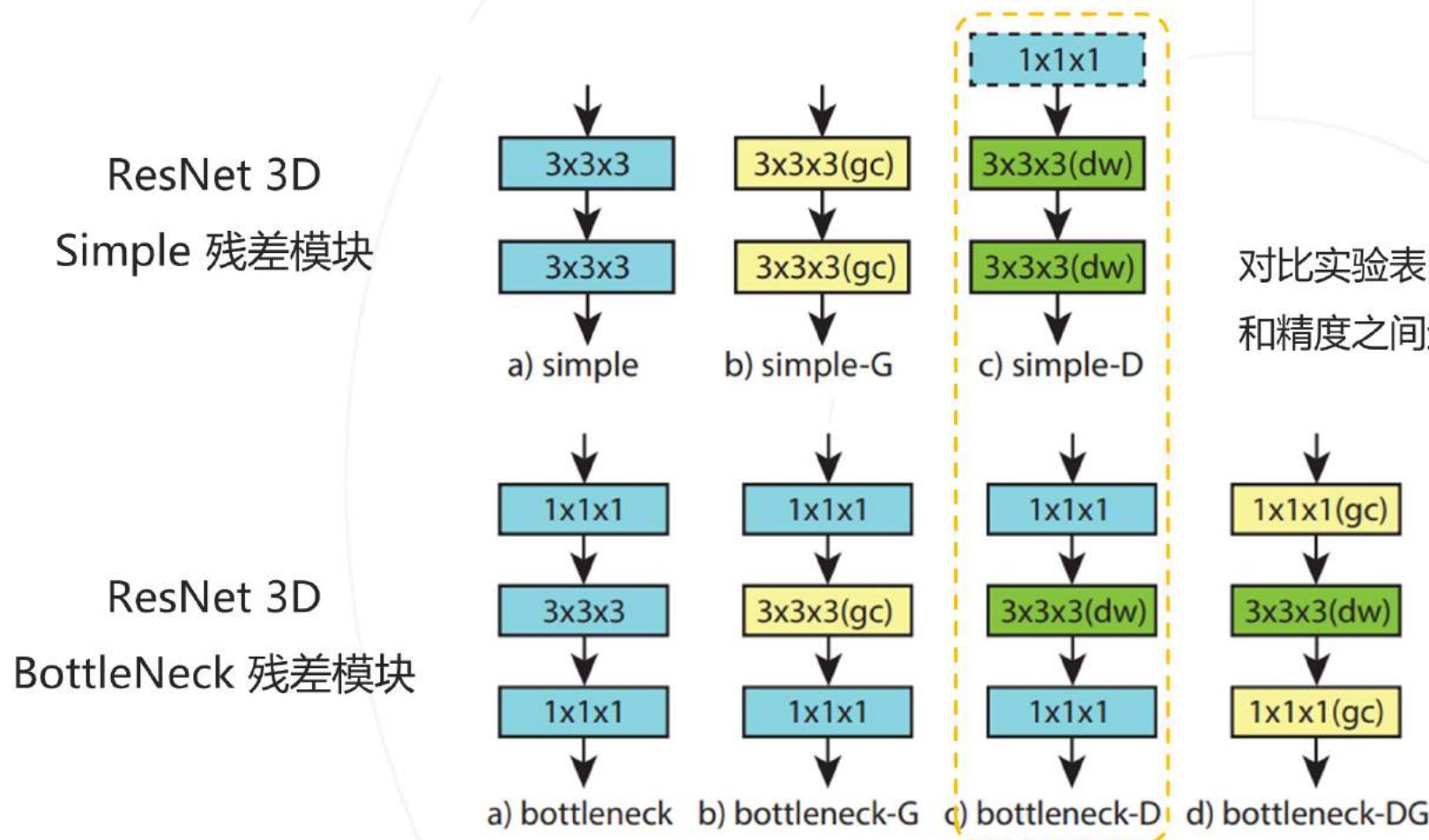
在不同数据集上比较 S3D 模型、I3D 模型与 2/3D 混合的模型的计算量与精度

常规卷积参数量正比于通道数的平方，针对**大卷积核**使用分组或逐点卷积，可以大幅降低参数量和计算量。

	常规 3D 卷积
参数量	$C_{in} \times C_{out} \times K_H \times K_W \times K_T$
计算量	$C_{in} \times C_{out} \times K_H \times K_W \times K_T \times H \times W \times T$



CSN 以 3D 卷积的 ResNet 为基础，针对 simple 模块和 bottleneck 模块给出了不同的改造方式

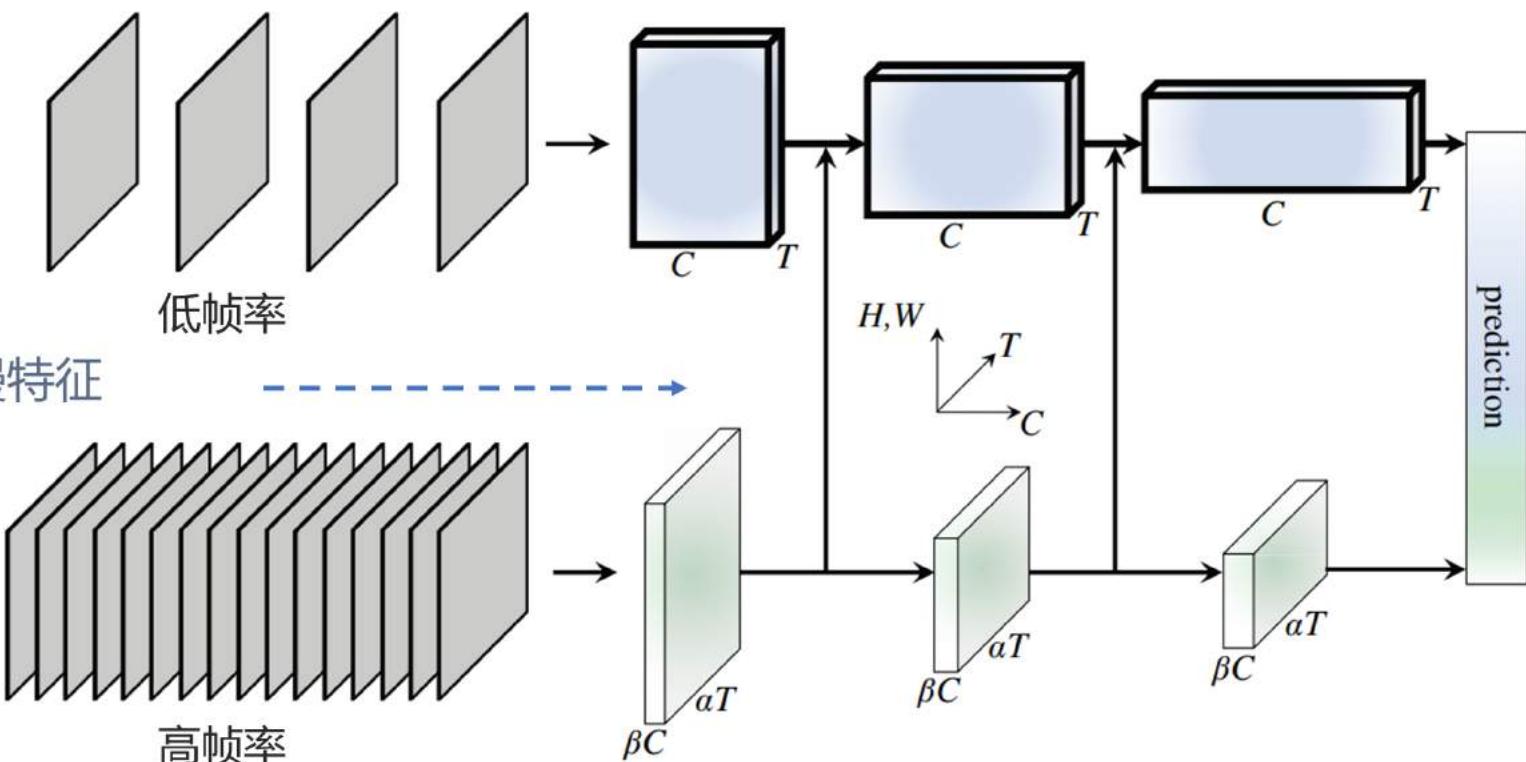


对比实验表明，D 结构可以在参数
和精度之间达到最好的平衡

外观和动作的**变化速度**是不同的，外观的变化通常较慢，运动的变化速度较快

→ 针对外观和运动的特点，使用不同结构的模型，优化参数分配

① 低速分支基于低帧率图像
捕捉外观特征



② 高速分支基于高帧率图像
捕捉运动特征

高速分支相对轻量

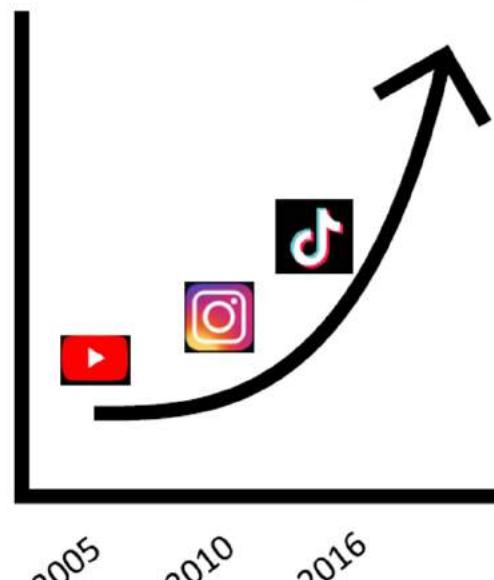
例， $\alpha=8$, $\beta=1/8$

大规模网络数据与弱监督学习

问题：视频数据的标注是昂贵的，但网络上有海量的视频数据。

如何有效利用这些数据，帮助我们更好地训练模型？

互联网视频的规模



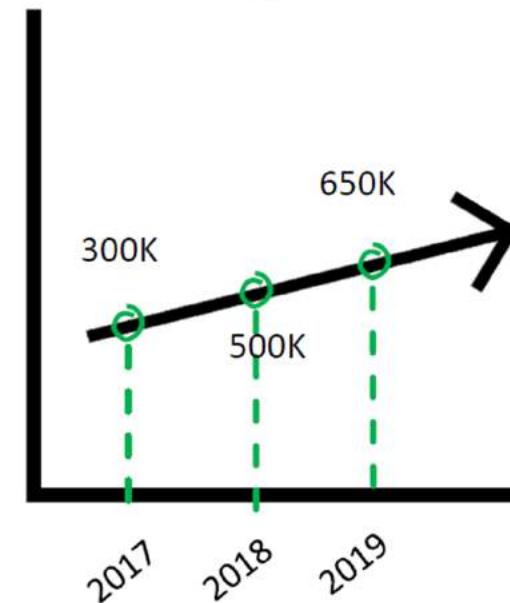
海量数据

但标注粗糙或没有标注

每分钟有数百小时的
视频上传！

标题、标签可以作为
粗糙的标注信息

标注数据集的规模

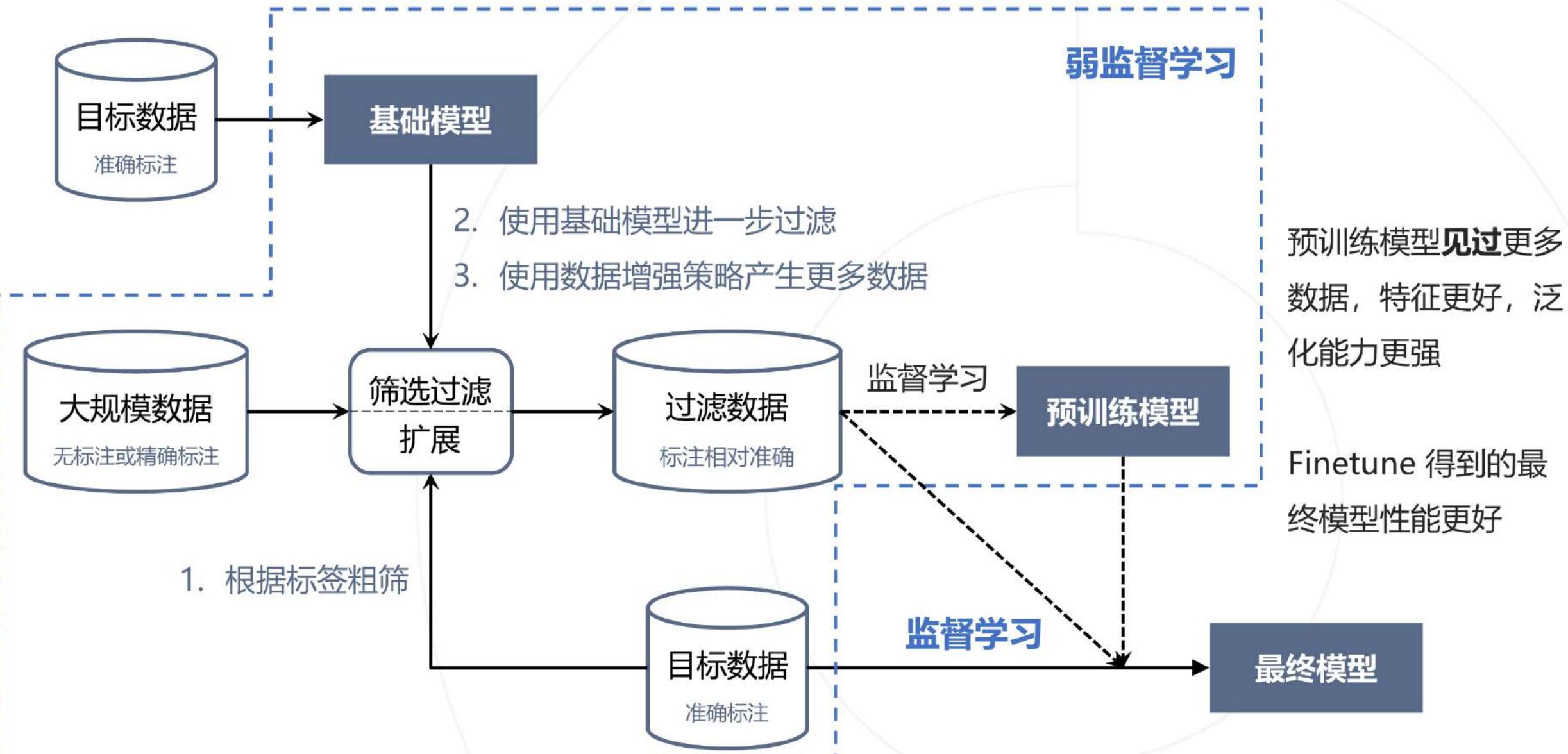


标注准确

但是数量相对较少

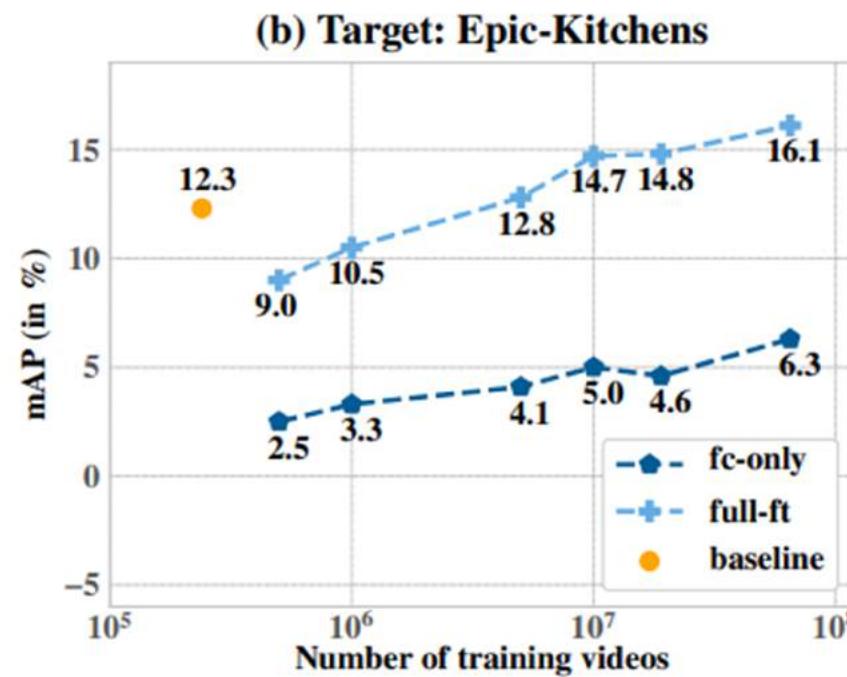
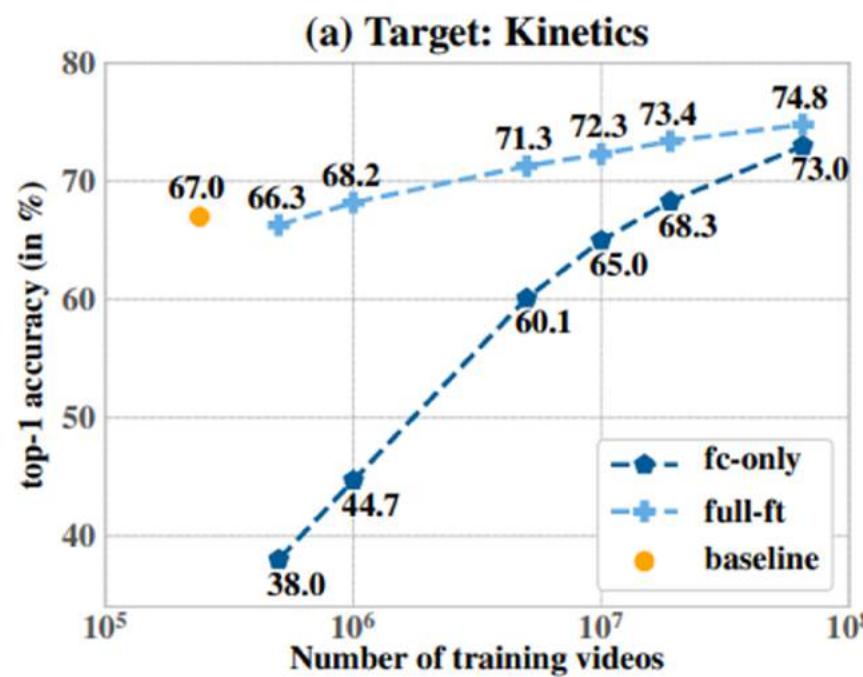


- ➡ 在机器学习中，使用标注不全或不准的数据进行学习的问题称为**弱监督学习** (weakly-supervised learning) 。
- ➡ 近年来，一些工作开始探索这个方向，代表工作包括来自 Facebook 的 IG-65M 以及来自香港中文大学的 OmniSource 等。
- ➡ 实验表明，使用百倍于学术数据集的弱监督数据训练出的模型，可以很好地迁移到学术数据集上，取得领先的监督学习的精度。

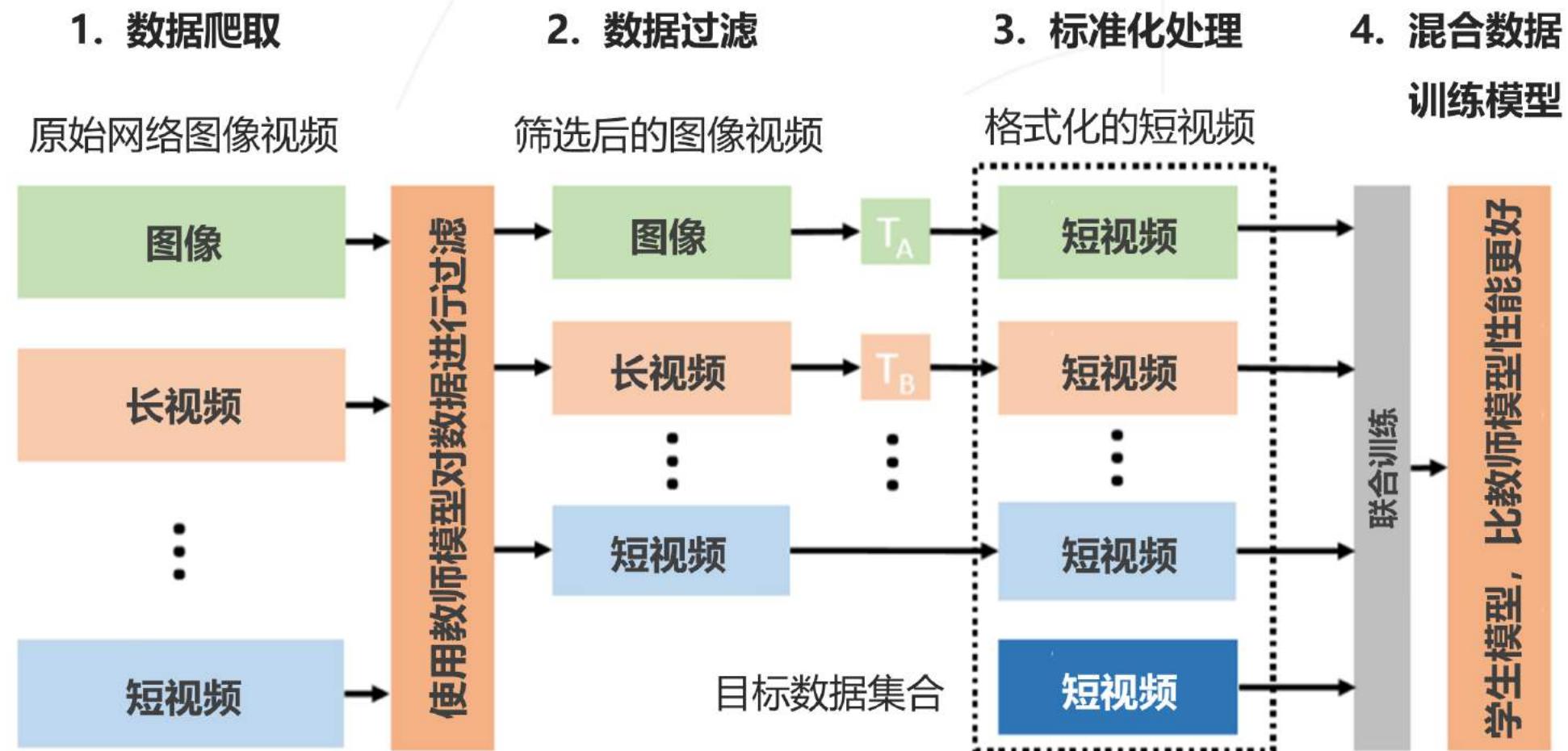


□ 使用 Kinetics 等数据集的关键字，从 Instagram 收集了 6500 万个视频，预训练一个大模型模型，再在 Kinetics 等数据集上微调训练。实验表明：

1. 经过预训练的模型性能**优于**直接在目标数据集上训练的模型
2. 预训练使用的数据越多，性能越好，不准确的标注可以由数据量弥补



2020 年，香港中文大学提出的 OmniSource 使用多种来源的数据联合训练模型；相较于 IG-65M，对数据的利用更为高效



同样是基于 IG-65M 的预训练数据集，使用 OmniSource 框架可以使得 ir-CSN 模型达到更高的精度。

方法	模型	预训练数据集	Top-1 分类精度	Top-6 分类精度
irCSN-32x2	irCSN-152	IG-65M	82.6%	95.3%
irCSN-32x2 (Omni)	irCSN-152	IG-65M	83.6%	96.0%

基本任务

行为识别

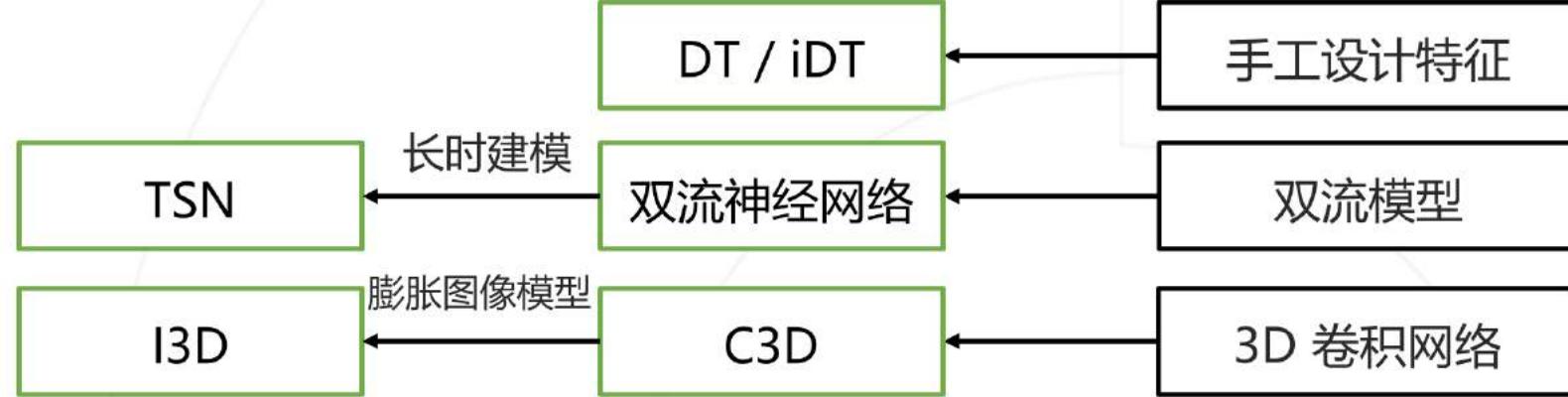
时序动作检测

时空动作检测



问题 1：如何得到更好的动作特征

问题 2：如何提高3D模型的计算效率



改造 3D 卷积核

优化参数分配

时空分解

S3D / R(2+1)D

局部通道连接

CSN

优化参数分配

SlowFast

问题 3：如何利用海量低成本网络视频



弱监督学习

IG-65M

OmniSource

谢谢大家