Advanced Deep Spatial-Temporal Models

AI602: Recent Advances in Deep Learning

Lecture 4

Slide made by

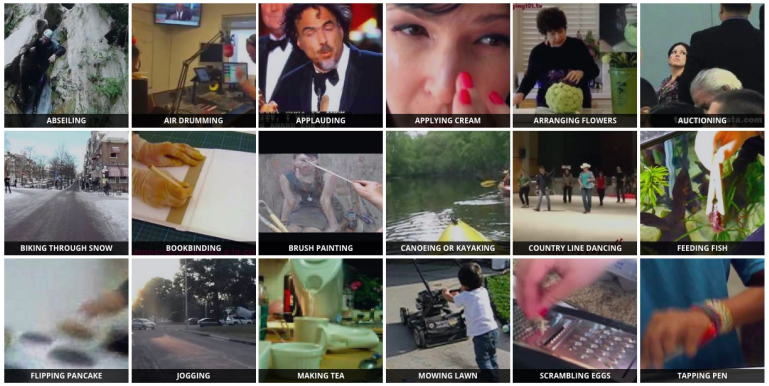
Taegu Han, Jaeyeon Won and Seong Hyeon Park

KAIST EE & AI

Overview: Deep Spatial-Temporal Models

- Recently, deep **spatial-temporal modeling** is rapidly emerging field of research following the advances in spatial models and temporal models

Video Action Recognition [Karpathy et al., 2014]



*source: https://towardsdatascience.com/downloading-the-kinetics-dataset-for-human-action-recognition-in-deep-learning-500c3d50f776

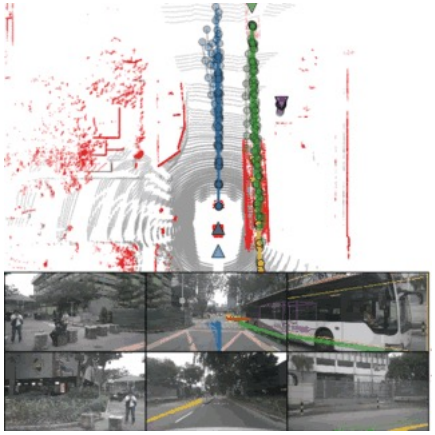
Video Captioning [Pan et al., 2020]

*source: http://www.auto-video-captions.top/2020/

Deep Object Tracking [Wang et al., 2020]

*source: https://github.com/Zhongdao/Towards-Realtime-MOT

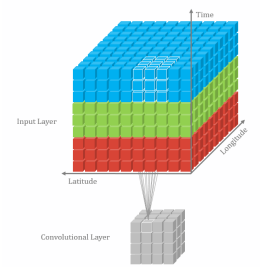
Deep Motion Forecasting [Rhinehart et al., 2



*source: https://sites.google.com/view/precog

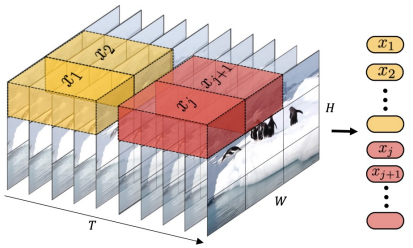
Overview: Deep Spatial-Temporal Models

- Advanced Spatial models and temporal models are leveraged in many ways
  - Directly expanding extra dimensions for spatial models (e.g., CNNs and Vision Transformer [Dosovitskiy et al., 2021])



*source: https://tomms-solution.tistory.com/52

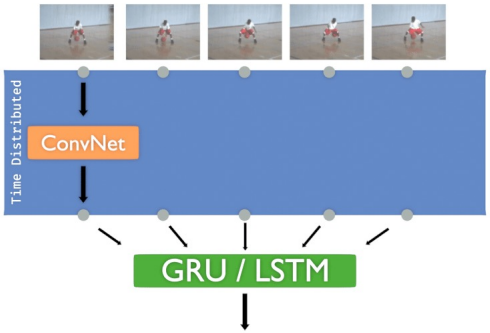
3D Convolutional Neural Networks



*source: [Arnab & Dehghani et al., 2021] A Video Vision Transformer, ICCV 2021

Video Vision Transformers

  - Fusing spatial and temporal architectures (e.g., CNN + LSTM)

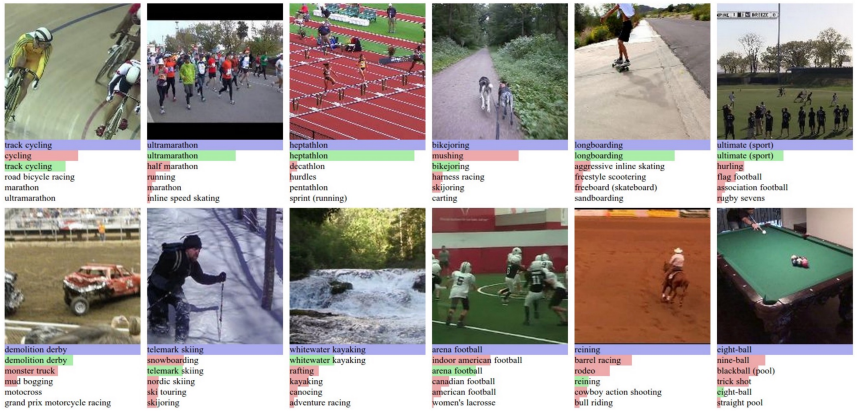


*source: https://medium.com/smileinnovation/training-neural-network-with-image-sequence-an-example-with-video-as-input-c3407f7a0b0f

Overview: Deep Spatial-Temporal Models

- Similarly to the spatial models, **classification** is considered a fundamental task
  - Specifically, **recognizing human actions** in video is the most active research area
  - It is often called **Video Action Recognition** to clearly depict the objective

- Advanced architectures for video action recognition are the backbones for downstream tasks involving spatial-temporal data
  - Recall the roles of **ResNet** [He et al., 2016] and **ViT** [Dosovitskiy et al., 2021] as the backbones for spatial models research



*source: https://cs.stanford.edu/people/karpathy/deepvideo/

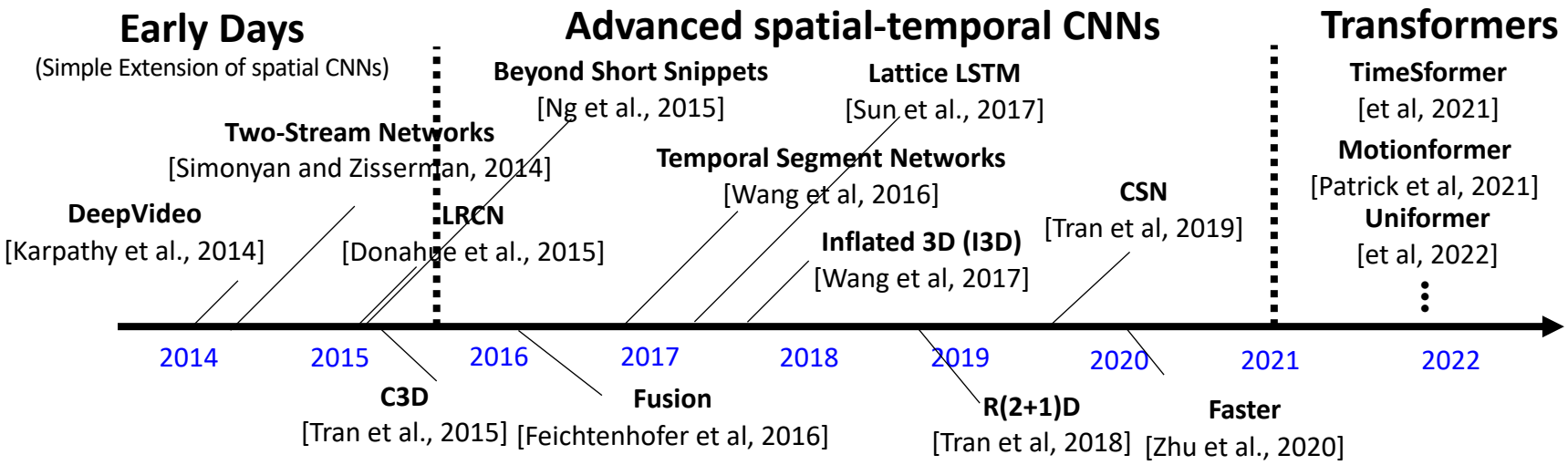
Problem: The curse of dimensionality and spatial-temporal information fusion

1. **Computation Scale**
   - Spatial-temporal data (e.g., video) is inherently high-dimensional
   - Brute-force extension of spatial models is often intractable
   - Data Sub-sampling & approximated network architectures are typically employed
2. **Spatial-temporal Information Fusion**
   - Pipelines for spatial cue (appearance) and temporal cue (motion) are sometimes independent
   - The following question naturally arises:
     - How to fuse information from the two separate pipelines?
     - In which part of the network the fusion should happen?
     - Partially related to **multimodal machine learning** problem
3. **Long-range modeling**
   - Likewise temporal models, the long-range modeling (e.g., recognizing a minutes-long video) is challenging

- Good models should be **computationally scalable** (e.g., linear complexity to temporal dimension) and appropriately dealing with **information fusion** & **long-range modeling** problems

Evolution of CNN Architectures for Video

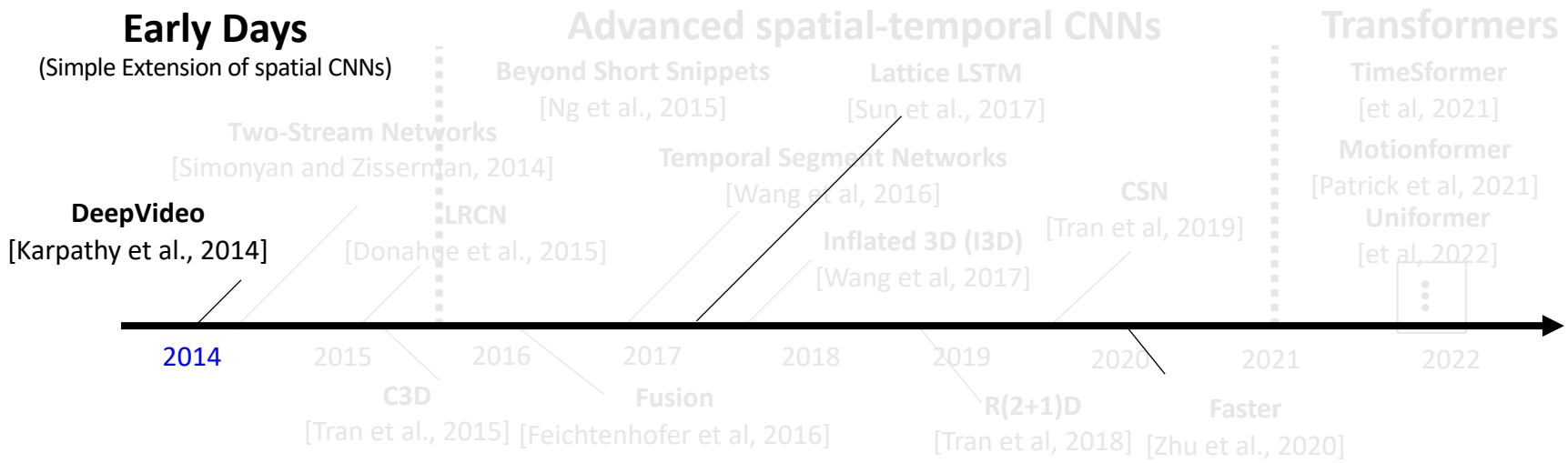
- Advances in **spatial-temporal models** has been much slower than image models
  - **Lack** of public, large-scale and high-quality datasets (e.g., ImageNet)
  - **Heavy compute scale** due to the high-dimensional nature hinders active research
  - **Less attention** as spatial-modeling and temporal-modeling were challenging enough
- There is **no clear model genealogy** for **early** deep spatial-temporal models
  - Early works are often presented without benchmarks in large-scale datasets
  - Though, they are important milestones to recent spatial-temporal models
  - Rough chronology of models that will be covered in this lecture:



*source: [Zhu et al., 2020] A comprehensive study of deep video action recognition

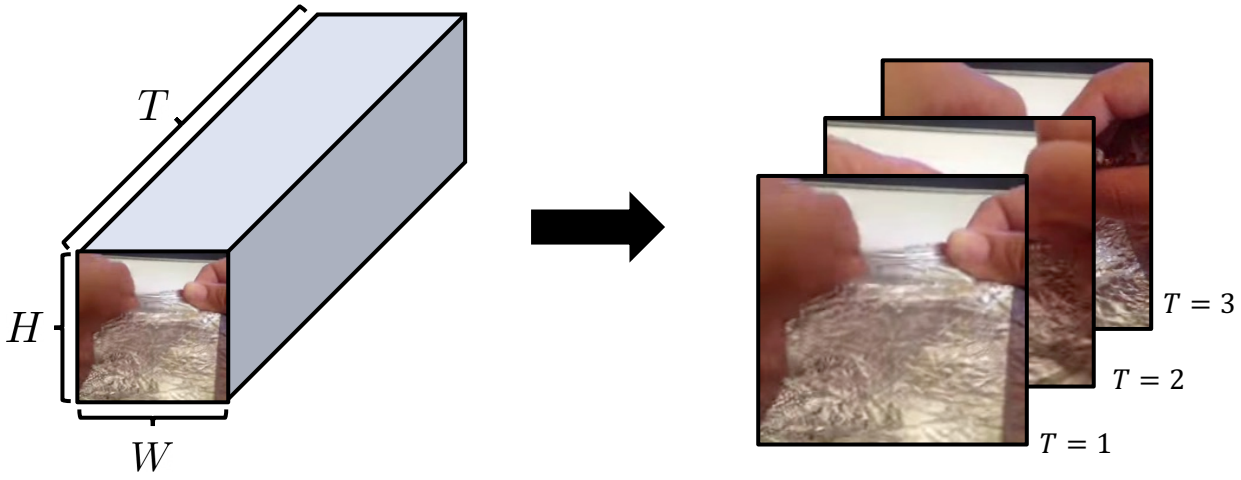
Part 1. Evolution of CNNs for spatial-temporal data

- Early Works: naïve extension of 2D CNNs
- Multi-stream and Temporal Segment Networks
- 3D CNNs
- CNN-RNN fusion models



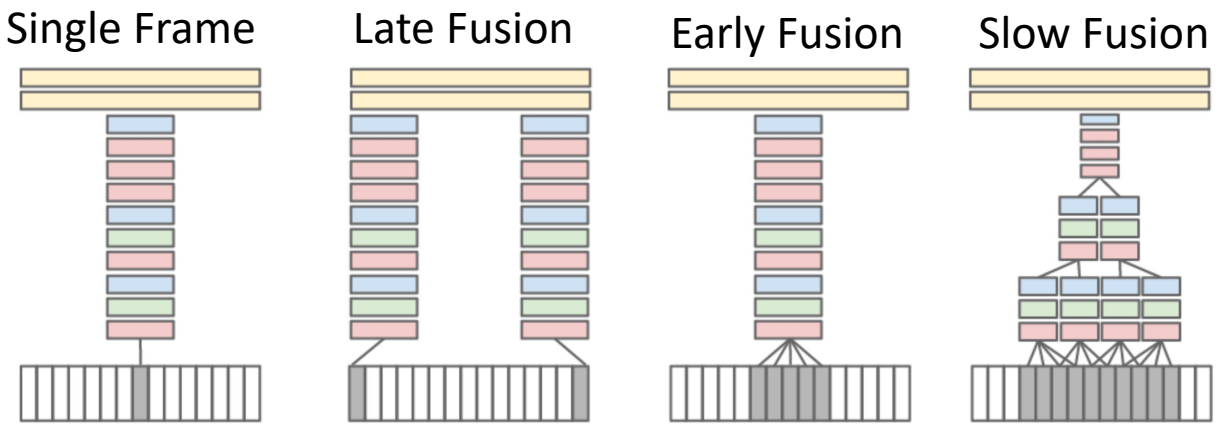
Evolution of CNN Architectures for Video: DeepVideo [Karpathy et al., 2014]

- How is **raw video signal** represented in computers?
  - A video is 3D signal with *height, width and time* dimensions
  - If we fix the temporal index T, we obtain a frame image
    - It is quite natural to consider applying an image classifier to each frame then fusing the outputs to make the final prediction
    - Many design choices depending on fusion strategies



Evolution of CNN Architectures for Video: DeepVideo [Karpathy et al., 2014]

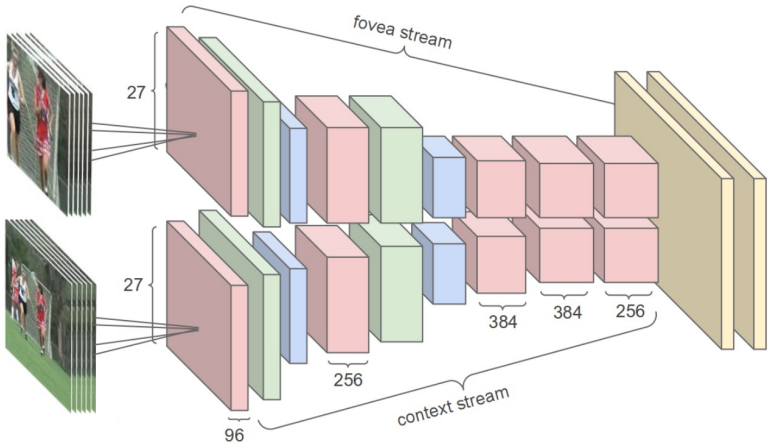
- DeepVideo [Karpathy et al., 2014]
  - Using AlexNet [Krizhevsky et al., 2012] as the image classifier, four different fusion strategies are considered:
    - **Single Frame**: predicting video action based on **one frame**
    - **Late Fusion**: combines information **at the last convolutional layer**
    - **Early Fusion:** combines information **immediately on the pixel level**
    - **Slow Fusion:** combines information **at the pixel level and the feature levels**



*source : Karpathy et al. “Large-scale video classification with convolutional neural networks”. CVPR 2014

Evolution of CNN Architectures for Video: DeepVideo [Karpathy et al., 2014]

- DeepVideo [Karpathy et al., 2014]
  - Multi-resolution CNNs
    - Inspired by the biological vision system, downscale the original input sizes (178×178) to two half-sized inputs (89×89) + (89×89)
    - The **forvea stream** receives the **center-crop at the original scale**
    - The **context stream** receives the **whole frames in downscaled resolution**
    - Boosts the wall-clock training time from orders of weeks to a month (4 weeks)

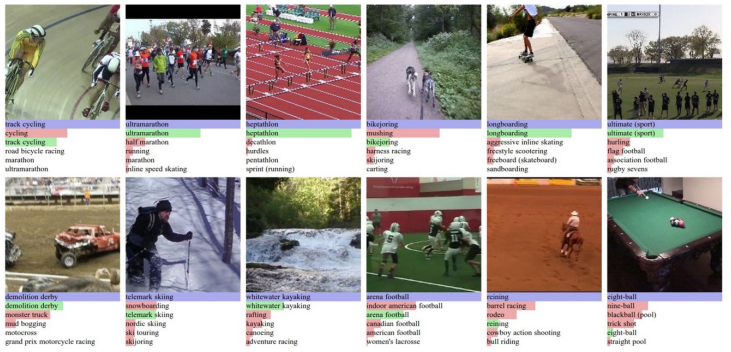


*source : Karpathy et al. “Large-scale video classification with convolutional neural networks”. CVPR 2014

Evolution of CNN Architectures for Video: DeepVideo [Karpathy et al., 2014]

- DeepVideo [Karpathy et al., 2014]
  - The Sports-1M dataset
    - Large in scale (1 million videos), but comes with very noisy auto-generated labels
    - The largest video dataset at the time (not used these days)
  - Experimental results in Sports-1M dataset
    - Single-Frame model (no fusion) can outperform the naively designed early and late fusion models
    - Only a sophisticated Slow Fusion model can outperform the Single-Frame model

Model	Clip Hit@1	Video Hit@1	Video Hit@5
Feature Histograms + Neural Net	-	55.3	-
Single-Frame	41.1	59.3	77.7
Single-Frame + Multires	**42.4**	**60.0**	**78.5**
Single-Frame Fovea Only	30.0	49.9	72.8
Single-Frame Context Only	38.1	56.0	77.2
Early Fusion	38.9	57.7	76.8
Late Fusion	40.7	59.3	78.7
Slow Fusion	**41.9**	**60.9**	**80.2**
CNN Average (Single+Early+Late+Slow)	41.4	63.9	82.4



*source : Karpathy et al. “Large-scale video classification with convolutional neural networks”. CVPR 2014

Evolution of CNN Architectures for Video: DeepVideo [Karpathy et al., 2014]

- DeepVideo [Karpathy et al., 2014]
  - One of the **earliest** deep video recognition work
  - However, the performance is unsatisfactory
    - Performs **inferior** to the a classical **hand-craft engineered model**

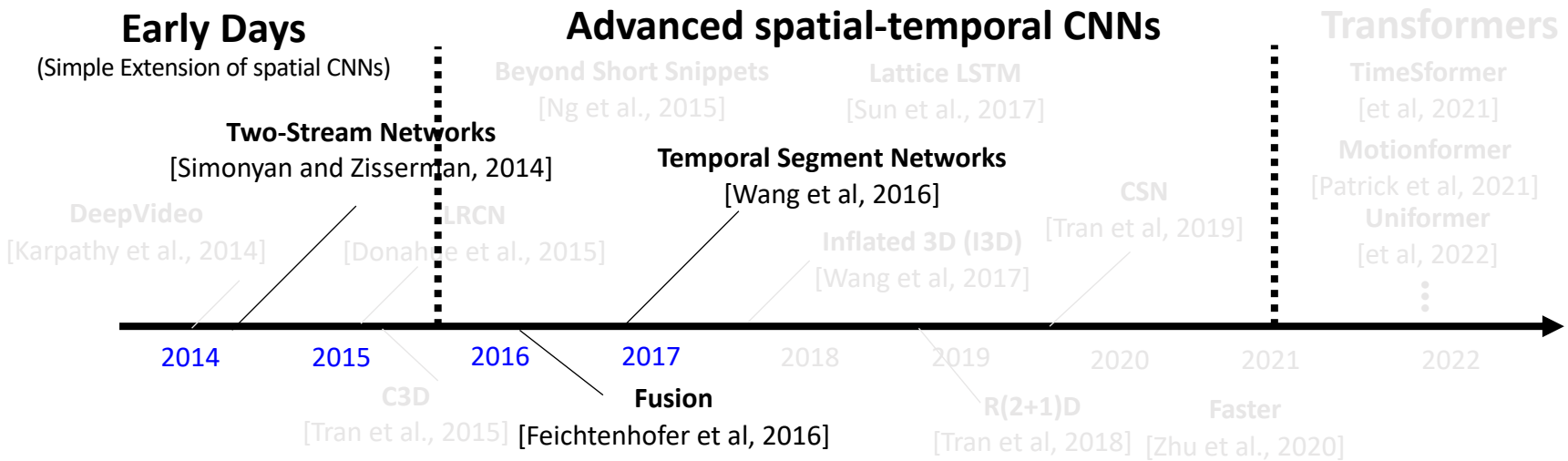
Method	UCF-101	HMDB-51
Improved dense trajectories (IDT) [26, 27]	85.9%	57.2%
IDT with higher-dimensional encodings [20]	**87.9%**	61.1%
IDT with stacked Fisher encoding [21] (based on Deep Fisher Net [23])	-	**66.8%**
Spatio-temporal HMAX network [11, 16]	-	22.8%
"Slow fusion" spatio-temporal ConvNet [14]	65.4%	-

  - Lessons:
    1. CNNs for **Image classification** can be **leveraged** to **classify videos**
    2. Choice of **temporal fusion strategies** largely **affects performances**
    3. It is **non-trivial** to **beat hand-crafted models** with CNNs (unlike image classification)

*source : Karpathy et al. "Large-scale video classification with convolutional neural networks". CVPR 2014

Part 1. Evolution of CNNs for spatial-temporal data

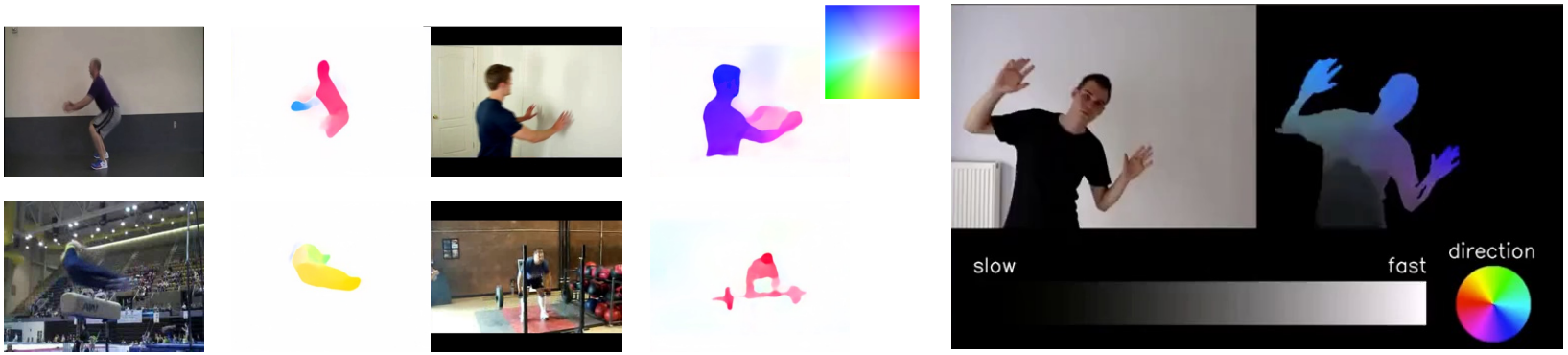
- Early Works: naïve extension of 2D CNNs
- Multi-stream and Temporal Segment Networks
- 3D CNNs
- CNN-RNN fusion models



Evolution of CNN Architectures for Video: Two-stream Networks [Simonyan et al., 2014]

- **Two-stream Networks** [Simonyan and Zisserman, 2014]
- Intuitively, video understanding can be improved with motion information
  - **Optical Flow** is an effective tool to describe the motions of objects in scene

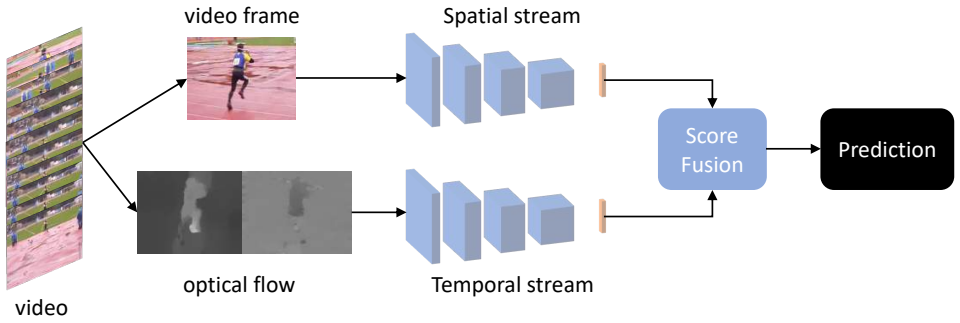
- Optical Flow
  - The representation of distinct motion of objects in scene
  - Visualizations of optical flow by FlowNet2 [Ilg et al., 2017]:
    - Colors indicate the directions of motions



*source : Ilg et al. “Flownet 2.0: Evolution of optical flow estimation with deep networks”. CVPR 2017

Evolution of CNN Architectures for Video: Two-stream Networks [Simonyan et al., 2014]

- Two-stream Networks [Simonyan and Zisserman, 2014]
  - **Spatial Stream**
    - An image CNN processing a single RGB image from video
  - **Temporal Stream**
    - Another image CNN for processing a stack of x- and y-directional optical flows
    - Optical flows for L duration around the selected image is used
    - Stacking L optical flows for x- and y-directions results in $2L$ channels



Two-stream Networks

*source: [Zhu et al., 2020] A comprehensive study of deep video action recognition

Evolution of CNN Architectures for Video: Two-stream Networks [Simonyan et al., 2014]

- Two-stream Networks [Simonyan and Zisserman, 2014]
  - **Spatial Stream**
    - An image CNN processing a single RGB image from video
  - **Temporal Stream**
    - Another image CNN for processing a stack of x- and y-directional optical flows
  - The final prediction is made with a SVM on the average output of the two streams (A naïve late fusion)
  - The first deep learning approach to achieve the comparable performance to its concurrent hand-craft models

Method	UCF-101	HMDB-51
Improved dense trajectories (IDT) [26, 27]	85.9%	57.2%
IDT with higher-dimensional encodings [20]	**87.9%**	61.1%
IDT with stacked Fisher encoding [21] (based on Deep Fisher Net [23])	-	**66.8%**
Spatio-temporal HMAX network [11, 16]	-	22.8%
"Slow fusion" spatio-temporal ConvNet [14]	65.4%	-
Spatial stream ConvNet	73.0%	40.5%
Temporal stream ConvNet	83.7%	54.6%
Two-stream model (fusion by averaging)	86.9%	58.0%
Two-stream model (fusion by SVM)	**88.0%**	**59.4%**

*source: [Simonyan and Zisserman, 2014] Two-stream convolutional networks for action recognition in videos

Evolution of CNN Architectures for Video: Two-stream Networks [Simonyan et al., 2014]

- Two-stream Networks [Simonyan and Zisserman, 2014]
  - **Spatial Stream**
    - An image CNN processing a single RGB image from video
  - **Temporal Stream**
    - Another image CNN for processing a stack of x- and y-directional optical flows
  - The final prediction is made with a SVM on the average output of the two streams (A naïve late fusion)
  - The first deep learning approach to achieve the comparable performance to its concurrent hand-craft models

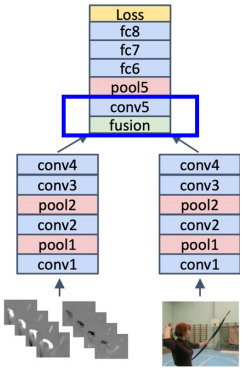
- Lessons:
  1. **It may be difficult for CNNs to directly learn motion from raw RGB frames**
  2. **Providing models explicit motion (e.g., Optical Flow) alleviates this issue**

Evolution of CNN Architectures for Video: Fusion for Two-stream Networks

- **Fusion strategy** affects performance (as shown by DeepVideo [Karpathy et al., 2014])
  - It is quite natural to consider **advanced fusion strategies for Two-stream Networks** [Simonyan and Zisserman, 2014] since it relies on a naïve late fusion

- **Fusion** [Feichtenhofer et al., 2016]
  - The first work to investigate **how to perform fusion in two-stream networks**
  - Considerable amount of experiments are conducted including…
    1. **Which layer** in CNN to perform the fusion
    2. Different **fusion operators** such as convolution, concatenation, sum, etc.
    3. Employing a **deeper architecture** (VGG16)

  - Finds some good practices for fusing multiple streams such as:
    - Fusing information with learned convolution operators
    - Fusing information at the last convolutional layers, before FC-layers



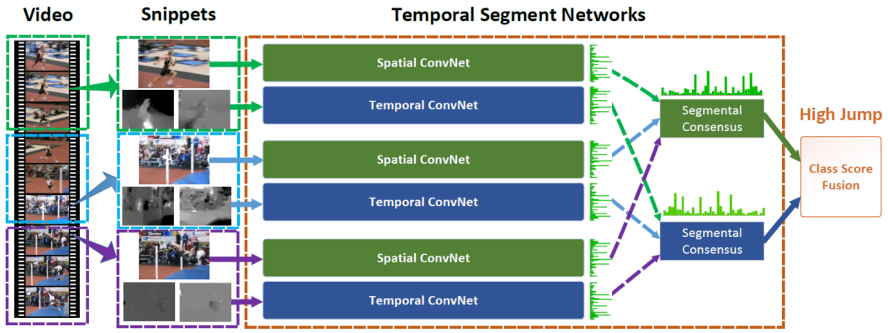
*source: [Feichtenhofer et al., 2016] Convolutional two-stream network fusion for video action recognition

Evolution of CNN Architectures for Video: segment-based methods

- Two-stream Networks model short-term motions with optical flow
  - However, they still reveal **weaknesses** in **long-range temporal modeling**

- Temporal Segment Networks (TSN) [Wang et al., 2016]
  - Divides a video into several **snippets**, then selects a single frame and optical flow within each snippet
  - The selected frames and optical flows are processed through a multi-stream (>2) network
  - Finally, the segmental consensus is fused using a simple average pooling operation

- With this simple architecture, TSN achieves **the state-of-the-art performance**



*source: [Wang et al., 2016] Convolutional two-stream network fusion for video action recognition

Evolution of CNN Architectures for Video: segment-based methods

- TSN [Wang et al., 2016]
  - TSN is the **first to perfectly beat** hand-crafted models by a huge margin (+6.8%)
    - TSN is like AlexNet [Krizhevsky et al., 2012] for deep spatial-temporal models

HMDB51		UCF101	
DT+MVSV [37]	55.9%	DT+MVSV [37]	83.5%
iDT+FV [2]	57.2%	iDT+FV [38]	85.9%
iDT+HSV [25]	61.1%	iDT+HSV [25]	87.9%
MoFAP [39]	61.7%	MoFAP [39]	88.3%
Two Stream [1]	59.4%	Two Stream [1]	88.0%
VideoDarwin [18]	63.7%	C3D (3 nets) [13]	85.2%
MPR [40]	65.5%	Two stream +LSTM [4]	88.6%
$F_{ST}CN$ (SCI fusion) [28]	59.1%	$F_{ST}CN$ (SCI fusion) [28]	88.1%
TDD+FV [5]	63.2%	TDD+FV [5]	90.3%
LTC [19]	64.8%	LTC [19]	91.7%
KVMF [41]	63.3%	KVMF [41]	93.1%
TSN (2 modalities)	68.5%	TSN (2 modalities)	94.0%
TSN (3 modalities)	**69.4%**	TSN (3 modalities)	**94.2%**

- In addition to splitting video to snippets, empirical gains also come from:
  - **ImageNet pretraining**, **Batch Normalization**, and **multi-stream pipelines**.

Training setting	Spatial ConvNets	Temporal ConvNets
Baseline [1]	72.7%	81.0%
From Scratch	48.7%	81.7%
Pre-train Spatial(same as [1])	84.1%	81.7%
+ Cross modality pre-training	84.1%	86.6%
+ Partial BN with dropout	84.5%	87.2%

Modality	Performance
RGB Image	84.5%
RGB Difference	83.8%
RGB Image + RGB Difference	87.3%
Optical Flow	87.2%
Warped Flow	86.9%
Optical Flow + Warped Flow	87.8%
Optical Flow + Warped Flow + RGB	**92.3%**
All Modalities	91.7%

*source: [Wang et al., 2016] Convolutional two-stream network fusion for video action recognition

- In fact, TSN comes with many advances other than the temporal segmenting

1. Leveraging **advanced backbone architectures**
   - An advanced backbone (e.g., BN-Inception) largely improves recognition

2. Carefully designed **initialization and regularizations**
   - Video datasets have orders of smaller dataset scale than image datasets, hence **leveraging pretrained weights** is important.
   - TSN empirically finds that **initializing non-RGB streams with the ImageNet** pretraining is beneficial (i.e., the cross-modality pre-training)
   - **Batch Normalization** [Ioffe et al., 2015] stabilizes training
     (See p.23, Lecture 02)

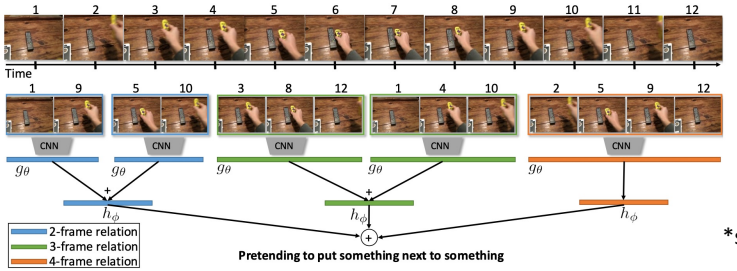
3. Utilizing multiple streams—e.g., RGB, optical flow and warped flow
   - TSN introduces new input streams in addition to RGB and Optical Flow
   - **More data modalities → improved performance**!

Evolution of CNN Architectures for Video: segment-based methods

- TSN [Wang et al., 2016] has established **a de facto standard for splitting a video into snippets** for video action recognition
  - Very recent works (e.g., video transformers [Neimark et al., 2021]) still follow this protocol to preprocess data

- TSN gives **lesson** that introducing **advanced backbones**, **regularization** and **image pretraining** is **important** for video models.

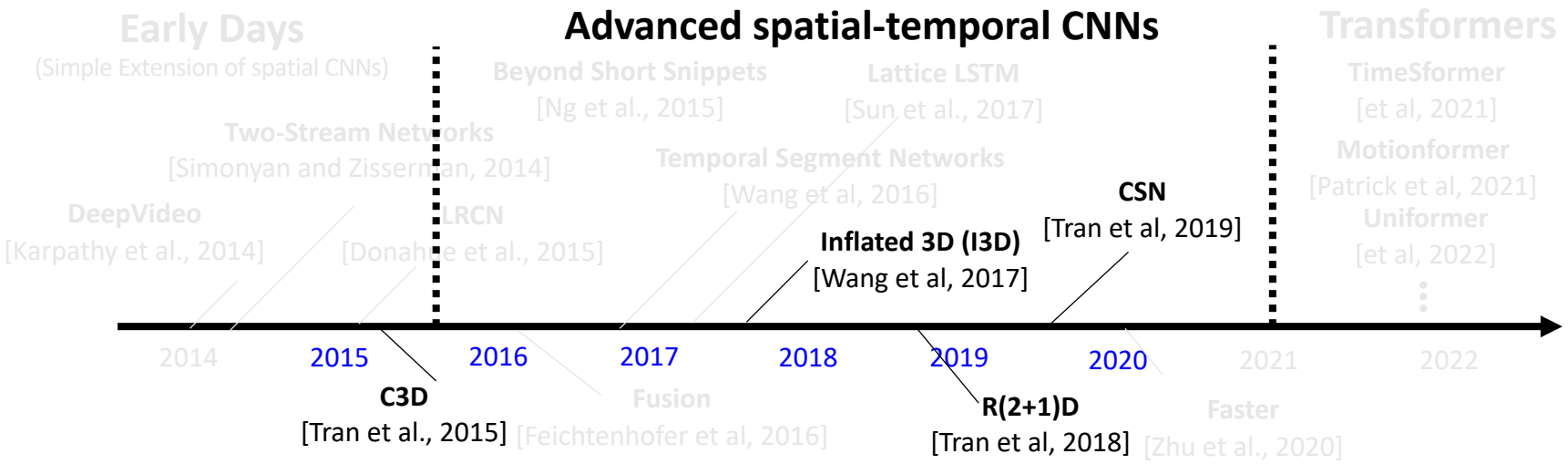
- TSN simply **averages** the classification confidence vectors from each segment
  - Some follow-up works that discover better fusion & segmenting strategies:
    - **Temporal Linear Encoding Network** [Diba et al., 2017]
      - Introduces a learnable bilinear transform for fusing segments
    - **Temporal Relation Network** [Zhou et al., 2018]
      - Introduces multiple time-scales (e.g., 2,3,4) for video snippets



*source: [Zhou et al., 2018] in ECCV 2018

Part 1. Evolution of CNNs for spatial-temporal data

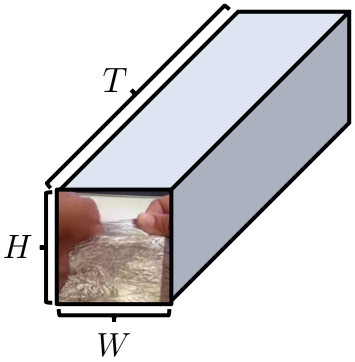
- Early Works: naïve extension of 2D CNNs
- Multi-stream and Temporal Segment Networks
- 3D CNNs
- CNN-RNN fusion models

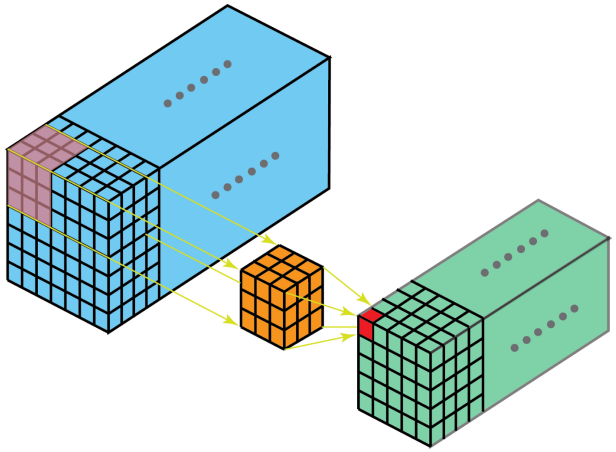


Evolution of CNN Architectures for Video: 3D CNNs

- Pre-computing **optical flow** **is computationally** **intensive**
- Recall the raw video signal's structure:
  - A video is **3D tensor** with two spatial and one time dimension
  - It is quite natural to employ 3D convolutional neural networks for **end-to-end learning of motion** from raw frames

- Some seminal works tried 3D CNNs for video recognition in early days:
  - **3D-Conv** [Ji et al., 2012] and **C3D** [Tran et al., 2015]
  - Their performances were unsatisfactory due to the **optimization difficulty of 3D CNNs** requiring high-quality & large-scale datasets

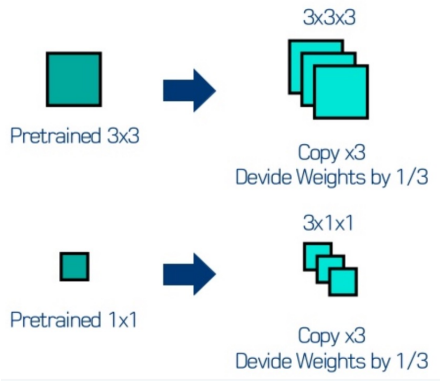




*source : https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215

- The situation changed with **Inflated 3D (I3D)** [Carreira and Zisserman, 2017]
- What has changed with the proposal of I3D?

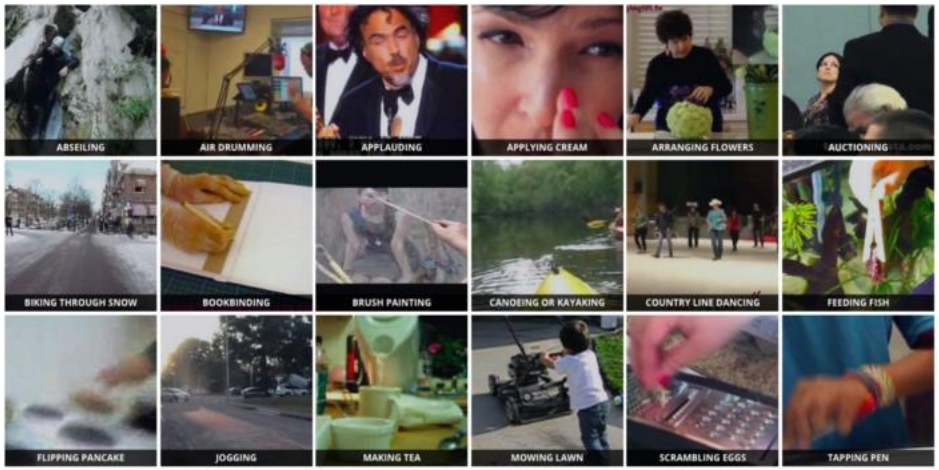
1. I3D directly adapts a very deep 2D CNN architecture to 3D CNN
   - I3D utilizes the Inception architecture
   - Instead of training from scratch, I3D leverages ImageNet-pretraining (How can 3D convolution kernels be pretrained with images?)
     - "**Kernel Inflating**" technique for initializing 3D kernels with 2D kernels



*source : https://chacha95.github.io/2019-07-04-VideoUnderstanding3/

- The situation changed with **Inflated 3D (I3D)** [Carreira and Zisserman, 2017]
- What has changed with the proposal of I3D?

2. Availability of the high-quality & large-scale video datasets
   - Kinetics dataset [Kay et al., 2017]
     - 500k videos with human-annotated labels of 400 action categories
     - One of the popular large-scale video benchmark until these days



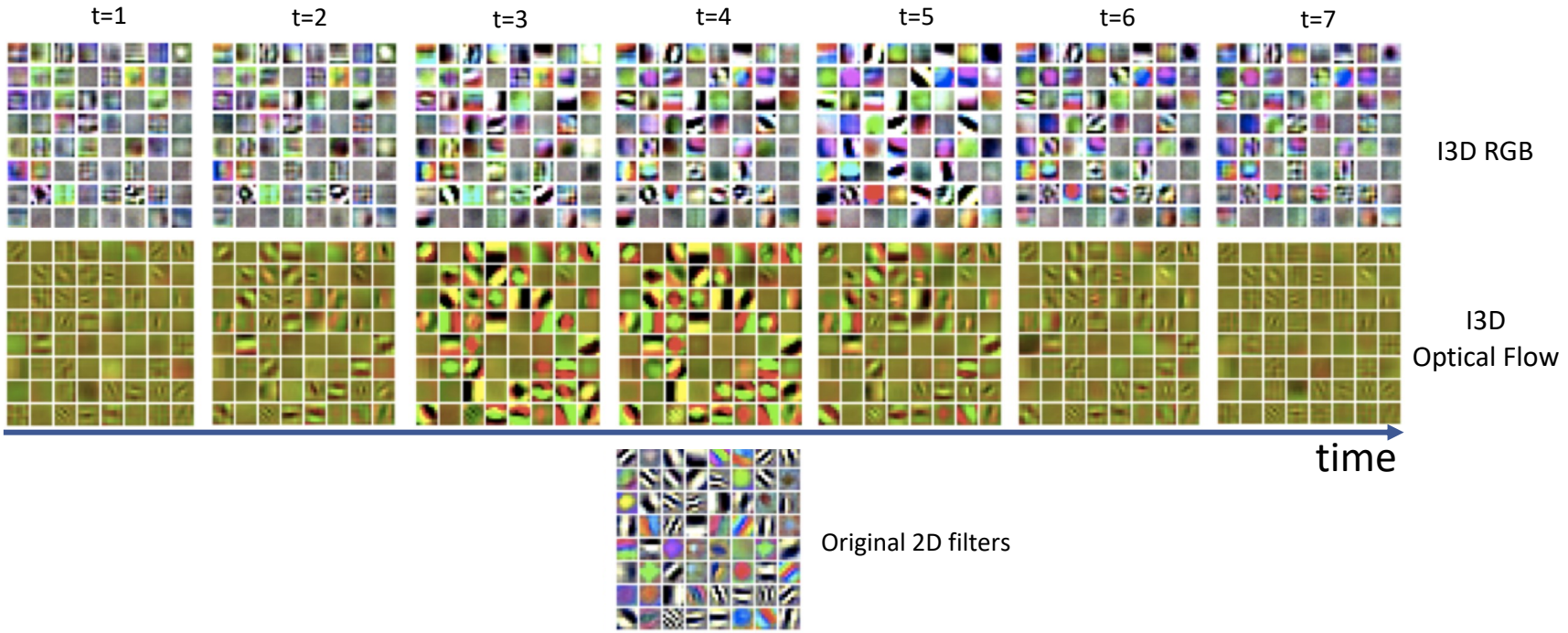
*source : [Kay et al., 2017] The Kinetics Human Action Video Dataset

- **Inflated 3D (I3D)** [Carreira and Zisserman, 2017]
  - The first work to bring 3D CNN to the state-of-the-art video recognition
    - **Kernel Inflating** & **large-scale Kinetics pretraining** are important
  - **3D CNNs** and **multi-stream networks** are not mutually exclusive
    - They are just orthogonal ways to model the temporal relationships
    - I3D performs even better with the multi-stream network design

Model	UCF-101	HMDB-51
Two-Stream [27]	88.0	59.4
IDT [33]	86.4	61.7
Dynamic Image Networks + IDT [2]	89.1	65.2
TDD + IDT [34]	91.5	65.9
Two-Stream Fusion + IDT [8]	93.5	69.2
Temporal Segment Networks [35]	94.2	69.4
ST-ResNet + IDT [7]	94.6	70.3
Deep Networks [15], Sports 1M pre-training	65.2	-
C3D one network [31], Sports 1M pre-training	82.3	-
C3D ensemble [31], Sports 1M pre-training	85.2	-
C3D ensemble + IDT [31], Sports 1M pre-training	90.1	-
RGB-I3D, Imagenet+Kinetics pre-training	95.6	74.8
Flow-I3D, Imagenet+Kinetics pre-training	96.7	77.1
Two-Stream I3D, Imagenet+Kinetics pre-training	**98.0**	80.7
RGB-I3D, Kinetics pre-training	95.1	74.3
Flow-I3D, Kinetics pre-training	96.5	77.3
Two-Stream I3D, Kinetics pre-training	97.8	**80.9**

*source : [Carreira and Zisserman, 2017] Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset

- **Inflated 3D (I3D)** [Carreira and Zisserman, 2017]
  - 3D Convolutional feature map learned by I3D
    - Top row: the 3D filters trained with I3D networks
    - Middle row: the 3D filters for optical flow in a 2-stream I3D
    - Bottom: the original Inception-v1 (an image CNN) filters
  - **I3D-trained RGB filters** are with patterns no more recognizable by humans
  - Interestingly, **optical flow filters** reveal clear patterns close to the **original 2D filters**



*source : [Carreira and Zisserman, 2017] Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset

- Further references for **3D CNNs** for video action recognition
  - **ResNet3D** [Hara et al., 2018]
    - Can Spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?
    - Translates ResNet [He et al., 2016] architecture to 3D CNN
      (**See p.23, Lecture 02**)
  - **ResNeXt** for 3D [Chen et al., 2018]
    - Multi-Fiber Networks for Video Recognition
    - Translates the multiple parallel path to 3D CNN
      (**See p.34, Lecture 02**)
  - **STCNet** [Diba et al., 2018]
    - Spatio-Temporal Channel correlation networks
    - Translates the Sequeeze-and-Excitation mechanism to 3D CNN
      (**See p.65, Lecture 02**)
  - **Advanced 2D CNNs for image recognition are actively translated to 3D CNN**

Evolution of CNN Architectures for Video: Efficient 3D CNNs

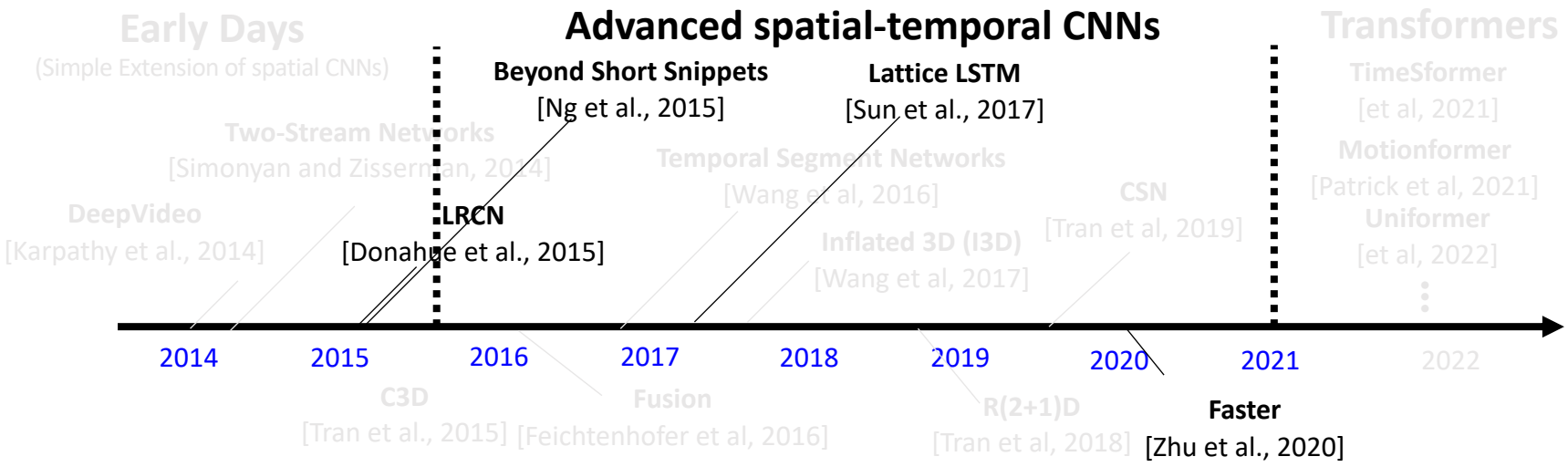
- Training and inferring with **3D CNNs** can be computationally too expensive
  - e.g., I3D [Carreira and Zisserman, 2017] demands computation burden comparable to the state-of-the-art video transformer models (100+ GFLOPs)
  - Hence, there is a line of research pursuing efficient 3D CNN architectures

- **Factorization** of 3D kernel
  - A **3D CNN kernel of size** $(P \times M \times N)$ **can be factorized** to two convolutions;
    - A spatial 2D kernel $(1 \times M \times N)$ and a temporal 1D kernel $(P \times 1 \times 1)$
  - **R2+1D** [Tran et al., 2018] and **P3D** [Qiu et al., 2017] directly adopts this idea to largely save FLOPs

- Application of **channel-wise separated convolutions**
  - **CSN** [Tran et al., 2019] shows the efficacy of separating channel interactions and spatiotemporal interactions
    - State-of-the-art performance is achieved with $\times 3$ less computations than I3D [Carreira and Zisserman, 2017]

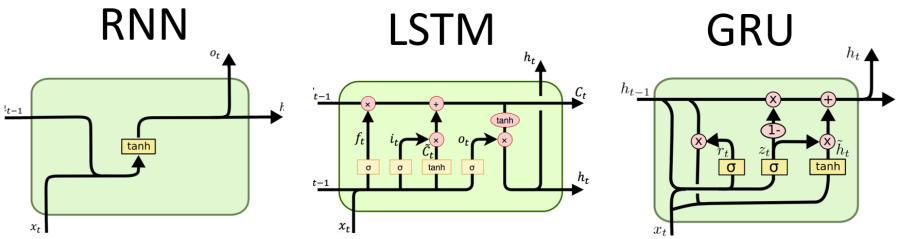
Part 1. Evolution of CNNs for spatial-temporal data

- Early Works: naïve extension of 2D CNNs
- Multi-stream and Temporal Segment Networks
- 3D CNNs
- CNN-RNN fusion models



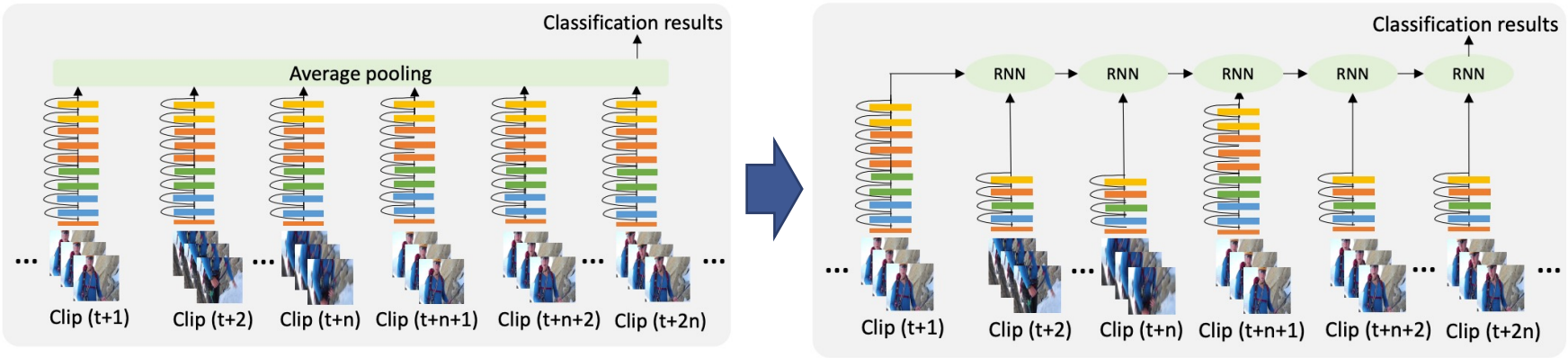
Evolution of CNN Architectures for Video: RNN + CNN models

- A video is essentially a temporal sequence
  - It is a natural direction to combine CNNs with RNNs (e.g., LSTM)
  - RNN recursively accumulates temporal information as hidden states (**See p.02, Lecture 03**)



*source : http://colah.github.io/posts/2015-08-Understanding-LSTMs/

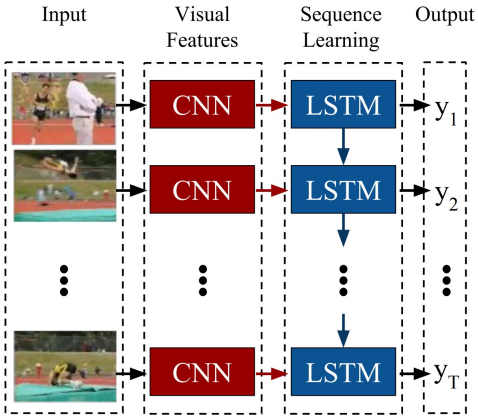
  - This line of research **replaces temporal fusion layers in CNN-based spatial-temporal models with RNN operations**



*source : FASTER [Zhu et al., 2020]

- LRCN [Donahue et al., 2015] & Beyond Short Snippets [Ng et al., 2015]
  - Two earliest concurrent works to fuse CNN and RNN architecture for spatial-temporal model
    - Input CNN features to LSTM [Hochreiter and Schmidhuber, 1997] (See **p.05, Lecture 03** for the details about LSTM)
    - It is shown that Two-streams Networks [Simonyan and Zisserman, 2014] can be improved (a bit) when **LSTM-based temporal fusion is introduced**

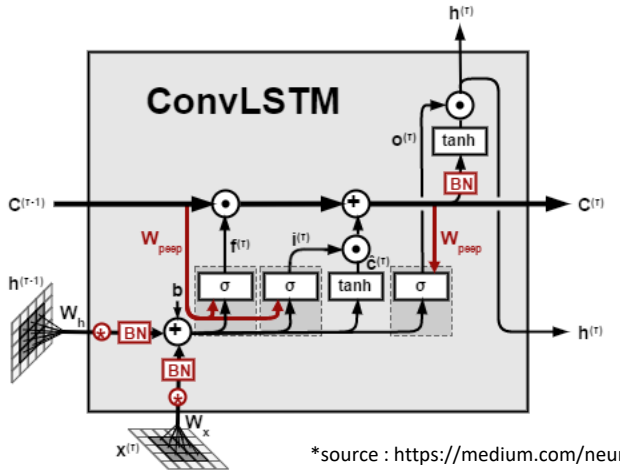
Method	3-fold Accuracy (%)
Improved Dense Trajectories (IDTF)s [23]	87.9
Slow Fusion CNN [14]	65.4
Single Frame CNN Model (Images) [19]	73.0
Single Frame CNN Model (Optical Flow) [19]	73.9
Two-Stream CNN (Optical Flow + Image Frames, Averaging) [19]	86.9
Two-Stream CNN (Optical Flow + Image Frames, SVM Fusion) [19]	88.0
Our Single Frame Model	73.3
Conv Pooling of Image Frames + Optical Flow (30 Frames)	87.6
Conv Pooling of Image Frames + Optical Flow (120 Frames)	**88.2**
LSTM with 30 Frame Unroll (Optical Flow + Image Frames)	**88.6**



$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
g_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

*sources : LRCN [Donahue et al., 2015] and Beyond Short Snippets [Ng et al., 2015]

- ConvLSTM [Shi et al., 2015] & **Lattice-LSTM** [Sun et al., 2017]
  - **ConvLSTM** [Shi et al., 2015] is a tweak of LSTM [Hochreiter and Schmidhuber, 1997] replacing LSTM's affine transformation with 2D convolutions



These affine transforms are replaced with convolutions

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
g_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

*source : https://medium.com/neuronio/an-introduction-to-convlstm-55c9025563a7

- **Lattice LSTM** [Sun et al., 2017] introduces ConvLSTM to video recognition
  - Long-term modeling performance **comparable** to Temporal Segment Networks

Very Deep	C3D (3 nets) [31]	85.2	-
	VideoLSTM[19]	89.2	56.4
	TDD+FV [33]	90.3	63.2
	Fusion [9]	92.5	65.4
Ours	L^2STM	**93.6**	**66.2**
Complex *	ST-ResNet [8]	93.4	66.4
	TSN [34]	94	68.5

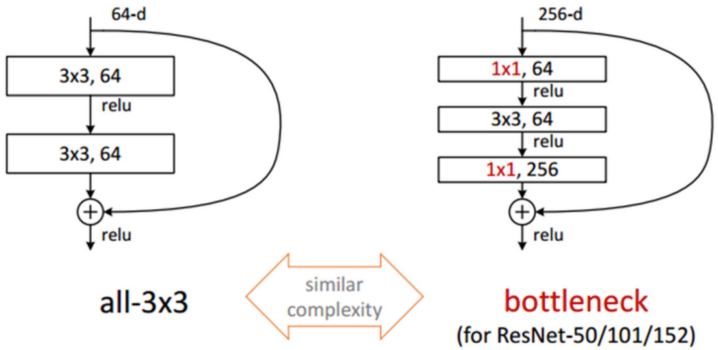
*source : **Lattice-LSTM** [Sun et al., 2017]

- FASTER [Zhu et al., 2020]
  - Introduces the 3D convolution operations to GRU [Cho et al., 2014]
    (See **p.12, Lecture 03** for the details about GRUs)
    - Similarly to ConvLSTM [Shi et al., 2015], affine transforms of GRU are replaced with 3D convolutions

These affine transforms are replaced with convolutions

$$\mathbf{r}_t = \sigma(\mathbf{G}_{rx}\mathbf{x}_t + \mathbf{G}_{ro}\mathbf{o}_{t-1}),$$
$$\mathbf{z}_t = \sigma(\mathbf{G}_{zx}\mathbf{x}_t + \mathbf{G}_{zo}\mathbf{o}_{t-1}),$$

  - As discussed, 3D convolutions are with heavy computations
    - FASTER [Zhu et al., 2020] introduces ResNet [He et al., 2015]-inspired **bottleneck layers** to their 3D Convolutional GRUs
    - Performance **comparable** to I3D at 5× cheaper GFLOPs



*source : https://blog.naver.com/laonple/220764986252

Model	Top-1	ImageNet pre-train	GFLOPs×clips
I3D [5]	72.1	✓	108×4
S3D [43]	72.2	✓	66.4×N/A
MF-Net [7]	72.8	✓	11.1×50
A^2-Net [6]	74.6	✓	40.8×30
S3D-G [43]	74.7	✓	71.4×N/A
NL I3D-50 [41]	76.5	✓	282×30
NL I3D-101 [41]	77.7	✓	359×30
I3D [5]	68.4	-	108×4
STC [10]	68.7	-	N/A×N/A
ARTNet [39]	69.2	-	23.5×250
S3D [43]	69.4	-	66.4×N/A
ECO [46]	70.0	-	N/A×N/A
R(2+1)D-34 [36]	72.0	-	152×115
FASTER16	71.7	-	14.4×16
FASTER32	**75.3**	-	67.7×8

Benchmark in Kinetics dataset

Evolution of CNN Architectures for Video: RNN + CNN models

- **RNN + CNN** methods are interesting, yet relatively minor field

  (Pros)

  - Can better suit for **long-range modeling in spatial-temporal recognition** (Since RNNs are originally designed for such purposes!)
  - In some works (e.g., FASTER [Zhu et al., 2020]), it is shown that **RNN+CNN model can achieve comparable performance to state-of-the-art with less computations**

  (Cons)

  - Shows only **comparable or marginally improved performances** compared to CNN-only baselines
  - **Complex designs** with **doubled hyperparameters** due to incorporating two different architectures in one model
    (Recall the importance of hyperparameter search—**See Pg. 66, Lecture 01**)

- Instead, recent line of research are majorly toward **Transformer architectures** for spatial-temporal modeling

Table of Contents

Part 1. Evolution of CNNs for spatial-temporal data

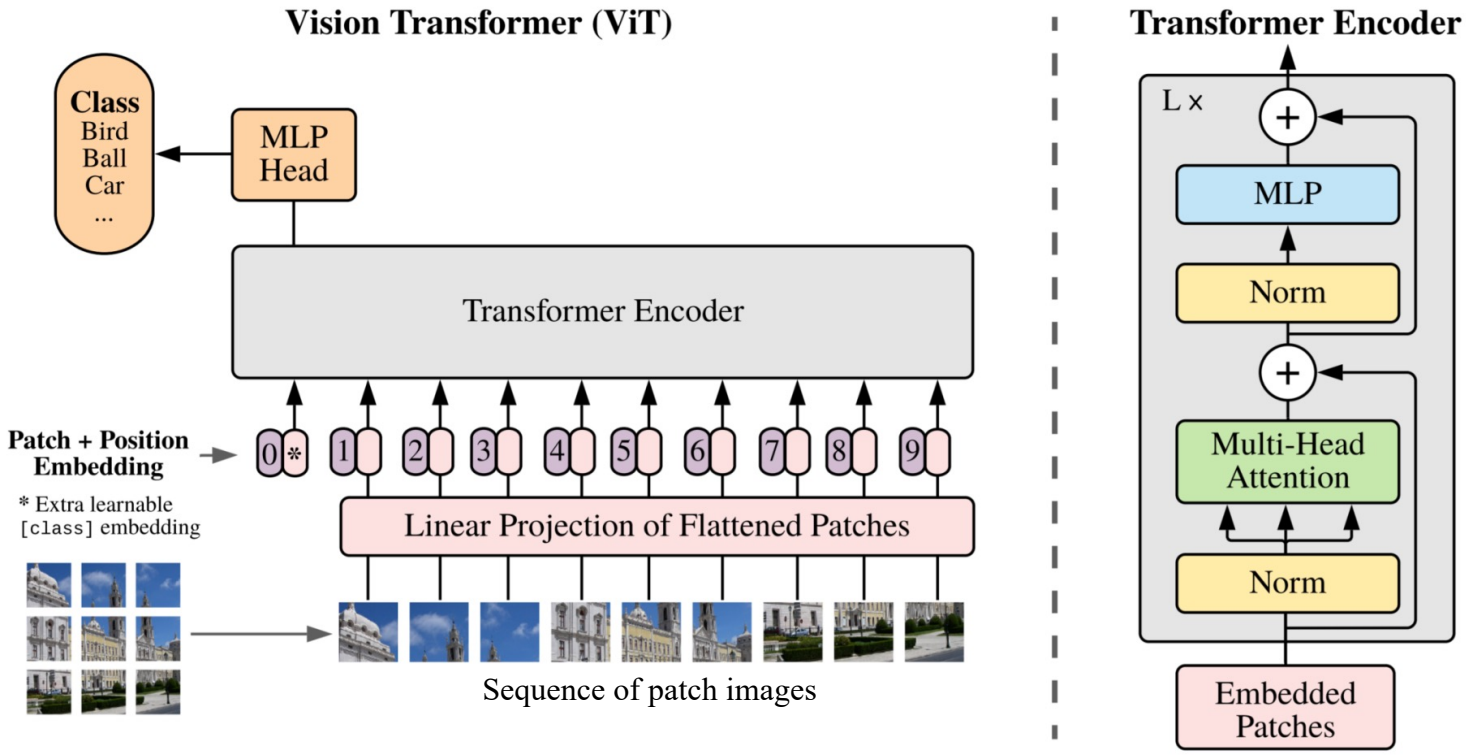
- Early Works: naïve extension of 2D CNNs
- Multi-stream and Temporal Segment Networks
- 3D CNNs
- CNN-RNN fusion models

Part 2. Transformers for spatial-temporal data

- Extension of vision transformer for spatial-temporal data
- Approximated attentions
- Unified transformer-CNN model

Recall: **Vision Transformer (ViT)** [Dosovitskiy et al., 2021]

- Splits an **image** into fixed-size patches (16x16)
  - Linearly embeds each of them
- Adds position embedding & extra learnable [class] token
- Feeds sequence of vectors to standard Transformer encoder



* source : [Dosovitskiy et al. 2021] An image is worth 16x16 words: Transformers for image recognition at scale, ICLR 2021

Transformers for spatial-temporal data : Extension of ViT

Recall: **Vision Transformer (ViT)** [Dosovitskiy et al., 2021]

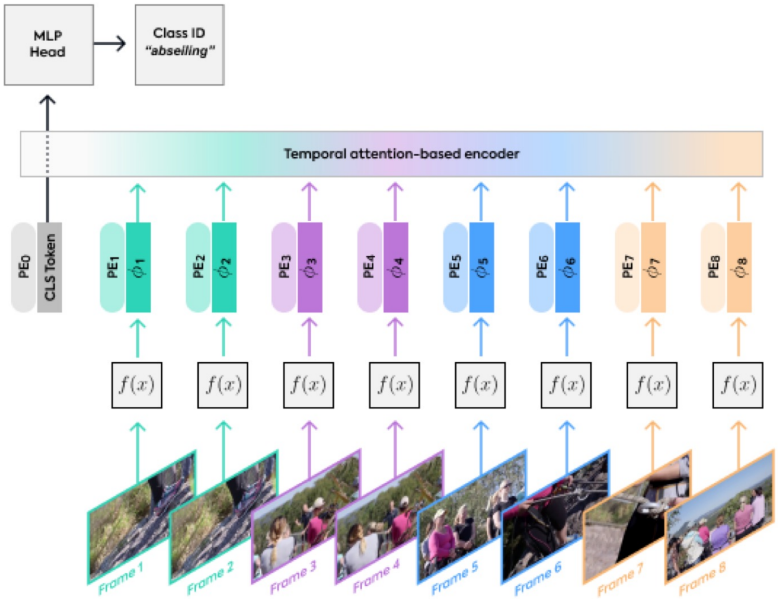
- Splits an **image** into fixed-size patches (16x16)
  - Linearly embeds each of them
- Adds position embedding & extra learnable [class] token
- Feeds sequence of vectors to standard Transformer encoder
- **Dosovitskiy et al.** (2021) pre-trains models on larger datasets (14M-300M images)
  - Vision Transformer achieves **competitive performances** compared to CNNs

- **Vision Transformer (ViT) can be directly extended to videos**
  - We cover the following two seminal works:
    - Video Transformer Network (VTN) [Neimark et al., 2021]
    - Video Vision Transformer (ViViT) [Arnab & Dehghani et al., 2021]

* source : [Dosovitskiy et al. 2021] An image is worth 16x16 words: Transformers for image recognition at scale, ICLR 2021

Video Transformer Network (VTN) [Neimark et al., 2021]

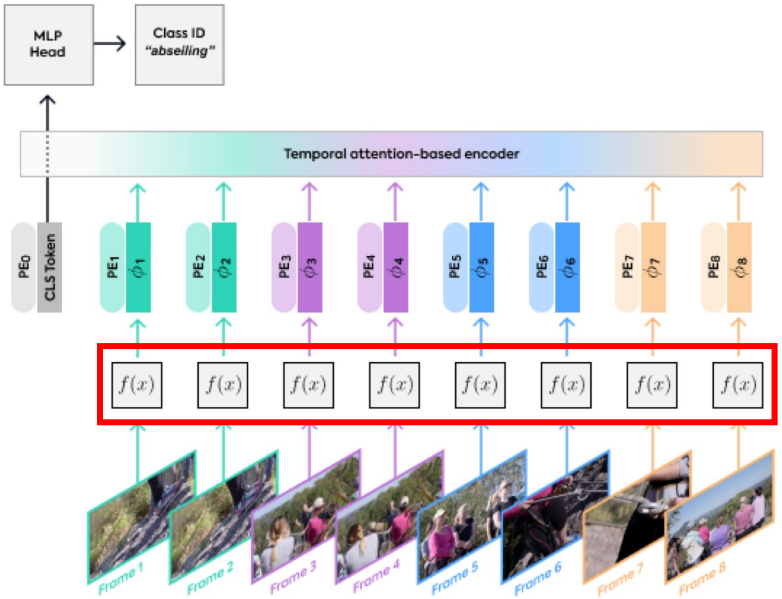
- VTN is a 2-stage **transformer-based** framework for video recognition attending to the entire video sequence information
- Processes entire video via a single end-to-end pass from frame to objective task
- Two key modules
  - 2D spatial backbone / Temporal attention-based encoder



* source : [Neimark et al. 2021] Video Transformer Network, ICCV 2021

Video Transformer Network (VTN) [Neimark et al., 2021]

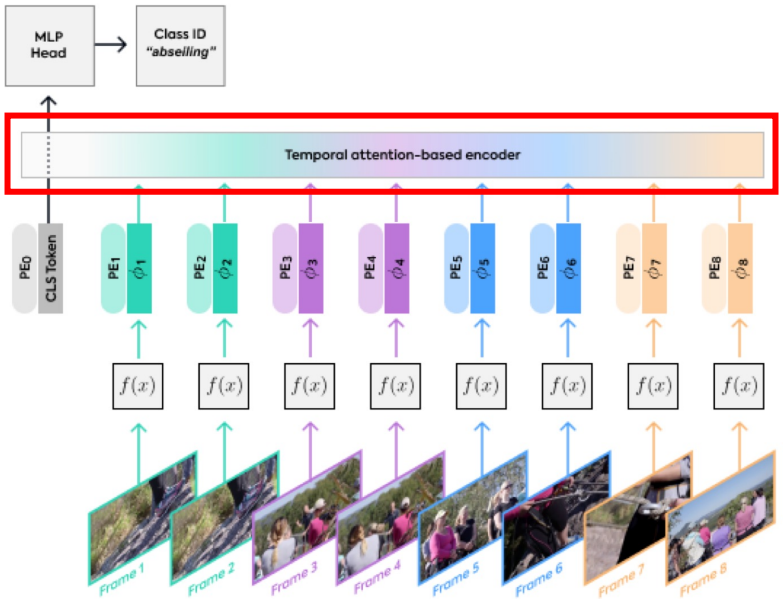
- **2D spatial feature extraction model**
  - Can be any network that works on 2D images
  - **VTN** uses ViT [Dosovitskiy et al., 2021] as the backbone architecture
  - The backbone produces an set of spatial tokens for each frame, which later will be aggregated with temporal encoder

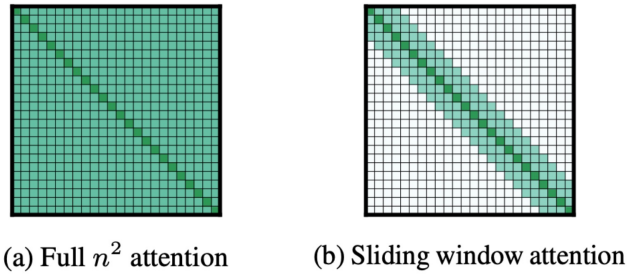


* source : [Neimark et al. 2021] Video Transformer Network, ICCV 2021

Video Transformer Network (VTN) [Neimark et al., 2021]

- **Temporal attention-based encoder**
  - Due to Transformer's **quadratic complexity** with respect to inputs, the number of tokens is limited in long videos
  - To alleviate the complexity issue, VTN chooses **sliding window attention** [Beltagy et al., 2020] over time that result in **linear complexity over time**





(a) Full n^2 attention (b) Sliding window attention

* source : [Beltagy et al. 2020], arXiv 2020

Sliding window attention [Beltagy et al., 2020]

* source : [Neimark et al. 2021] Video Transformer Network, ICCV 2021

Video Transformer Network (VTN) [Neimark et al., 2021]

- **Benchmarks**
  - VTN achieves comparable accuracy to CNN-based baselines
  - However…
    - Due to more parameters, it takes longer to train and test
    - VTN is not pure end-to-end transformer because of the 2-stage designs

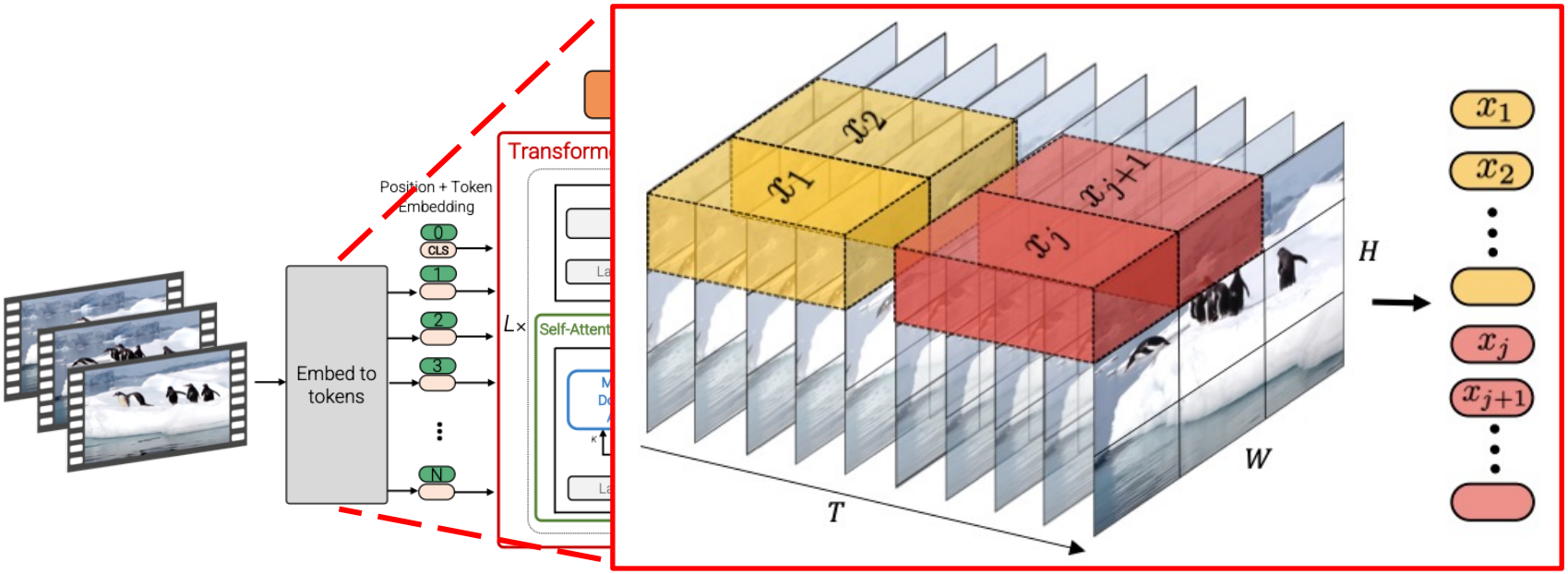
model	training wall runtime (minutes)	# training epochs	validation wall runtime (minutes)	inference approach	params (M)	top-1	top-5
I3D*	30	-	84	multi-view	28	73.5 [11]	90.8 [11]
NL I3D (our impl.)	68	50	150	multi-view	54	74.1	91.7
NL I3D (our impl.)	68	50	31	full-video	54	72.1	90.5
SlowFast-8X8-R50*	70	196 [13]	140	multi-view	35	77.0 [11]	92.6 [11]
SlowFast-8X8-R50*	70	196 [13]	26	full-video	35	68.4	87.1
SlowFast-16X8-R101*	220	196 [13]	244	multi-view	60	78.9 [11]	93.5 [11]
R50-VTN	62	40	32	full-video	168	71.2	90.0
R101-VTN	110	40	32	full-video	187	72.1	90.3
DeiT-Ti-VTN (3 layers)	52	60	30	full-video	10	67.8	87.5
ViT-B-VTN (1 layer)	107	25	48	full-video	96	78.6	93.4
ViT-B-VTN (3 layers)	130	25	52	full-video	114	78.6	93.7
ViT-B-VTN (3 layers)†	130	35	52	full-video	114	**79.8**	**94.2**

Kinetics-400 dataset benchmark

* source : [Neimark et al. 2021] Video Transformer Network, ICCV 2021

Video Vision Transformer (ViViT) [Arnab & Dehghani et al., 2021]

- ViViT is a pure **transformer** framework for video classification
- **Tubelet embedding** (3D extension of ViT)
  - Extract non-overlapping, spatial-temporal tubes from input volume
  - Linearly project them into $\mathbb{R}^d$



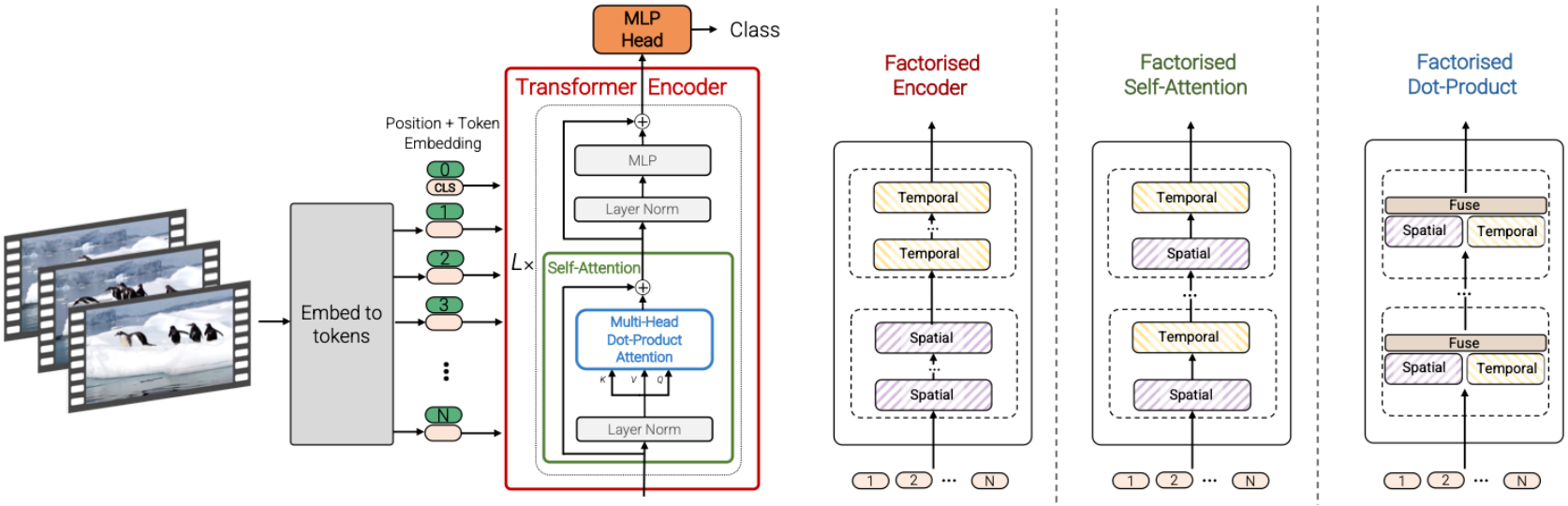
*source: [Arnab & Dehghani et al., 2021] A Video Vision Transformer, ICCV 2021

Video Vision Transformer (ViViT) [Arnab & Dehghani et al., 2021]

- Suggests different designs of **spatial & temporal attention**

1. (Joint) Spatio-temporal attention

- Simply forwards all pairwise interactions between all spatio-temporal tokens through transformer encoder
- Unlike CNN, it can model long-range interactions across the video from the 1st layer
- Requires quadratic complexity, $\mathcal{O}((n_h \cdot n_w \cdot n_t)^2)$, w.r.t number of tokens



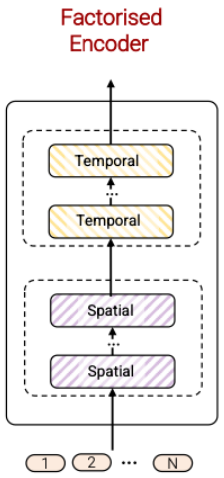
*source: [Arnab & Dehghani et al., 2021] A Video Vision Transformer, ICCV 2021

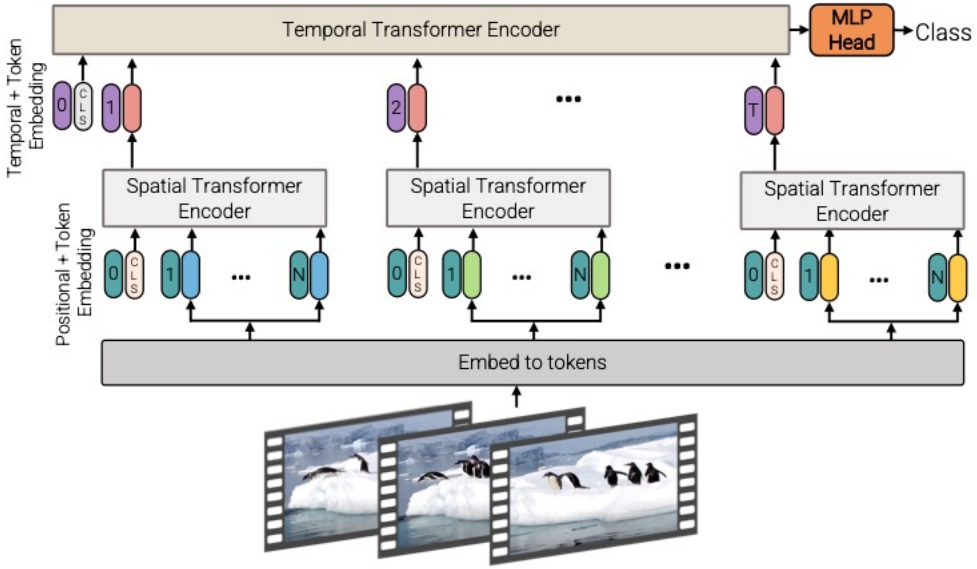
Video Vision Transformer (ViViT) [Arnab & Dehghani et al., 2021]

- Suggests different designs of **spatial & temporal attention**

2. Factorized encoder (Similar to VTN)

- **Spatial encoder** models interactions between tokens from the same temporal index
- **Temporal encoder** models interactions between tokens from different temporal indices
- Requires more transformer layers (i.e., more parameters) than Design 1
- Requires less complexity, $\mathcal{O}((n_h \cdot n_w)^2 + n_t^2)$ than Design 1





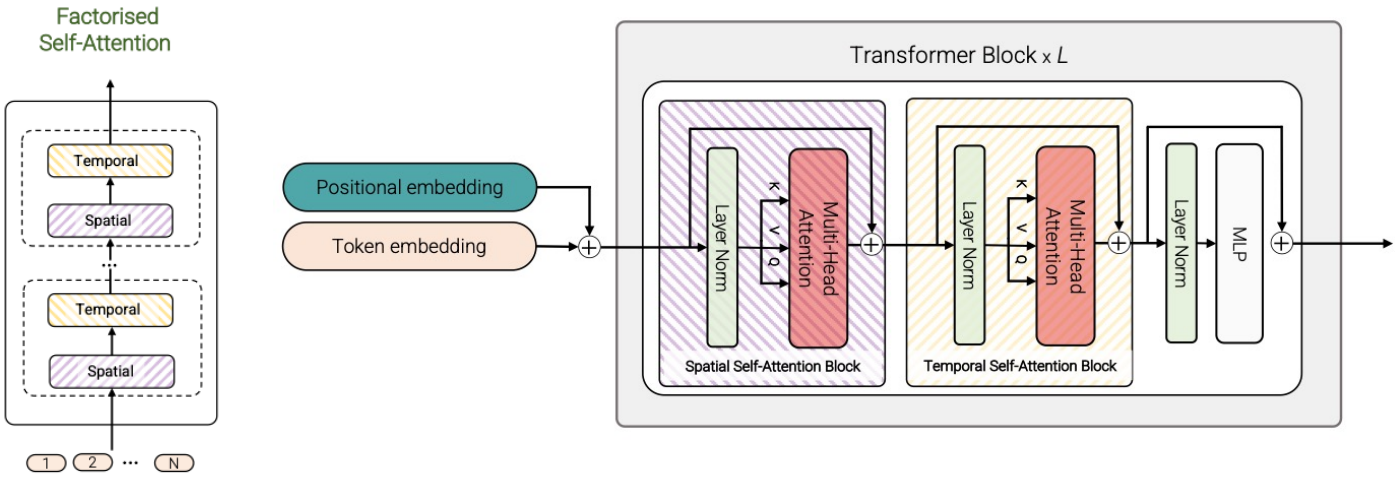
*source: [Arnab & Dehghani et al., 2021] A Video Vision Transformer, ICCV 2021

Video Vision Transformer (ViViT) [Arnab & Dehghani et al., 2021]

- Suggests different designs of **spatial & temporal attention**

3. Factorized self-attention

- First factorize to only compute self-attention spatially (all tokens from same temporal index)
- Then factorize to compute self-attention temporally (all tokens from sample spatial index)
- Requires same number of transformer layers as Design 1
- Requires same less complexity, $\mathcal{O}((n_h \cdot n_w)^2 + n_t^2)$, as Design 2



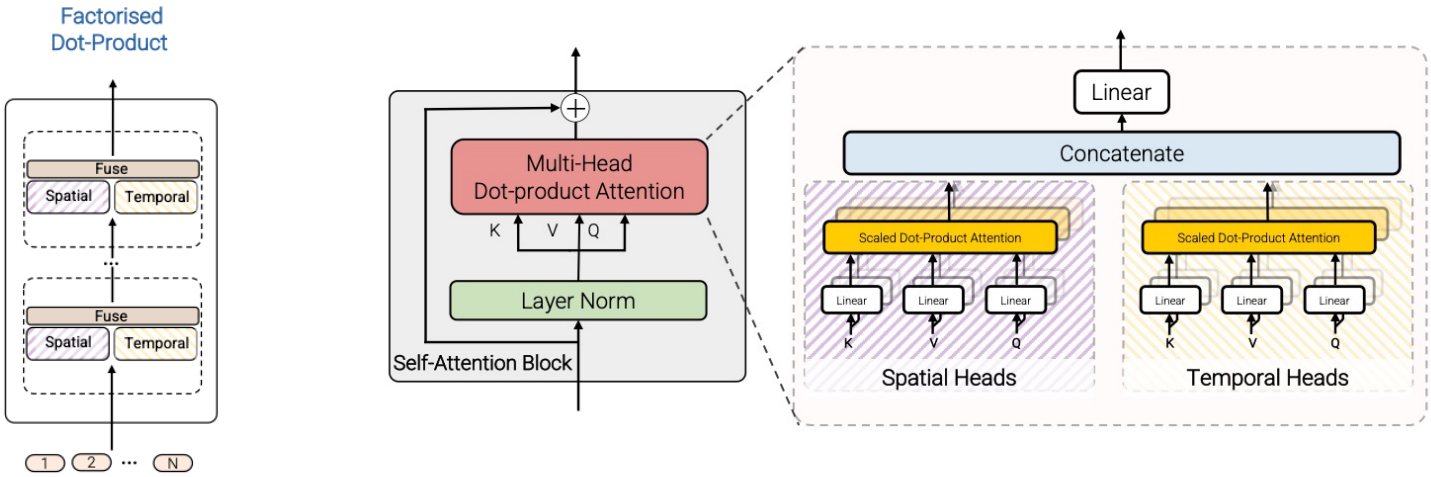
*source: [Arnab & Dehghani et al., 2021] A Video Vision Transformer, ICCV 2021

Video Vision Transformer (ViViT) [Arnab & Dehghani et al., 2021]

- Suggests different designs of **spatial & temporal attention**

4. Factorized dot-product attention

- Modify keys and values for each query to only attend over tokens from the same spatial index and temporal index
- Then factorize multi-head dot-product attention operation
- Requires same number of parameters as unfactorized Design 1
- Requires same less complexity, $\mathcal{O}((n_h \cdot n_w)^2 + n_t^2)$, as Design 2 and 3



*source: [Arnab & Dehghani et al., 2021] A Video Vision Transformer, ICCV 2021

Video Vision Transformer (ViViT) [Arnab & Dehghani et al., 2021]

- The factorized encoder (FE, model #2) shows the best accuracy-to-FLOPs ratio
  - Although ViViT can be a pure-transformer, they found the model #2 (**2-stage design** similar to VTN) is **more efficient**.
  - In fact, pure-transformer video models with good efficiency often come with sophisticatedly designed approximate attention (to be discussed in the **next chapter**)

- Nevertheless, ViViT (model #2) is the first work to surpass the CNN-based models

	K400	EK	FLOPs ($\times 10^9$)	Params ($\times 10^6$)	Runtime (ms)
Model 1: Spatio-temporal	80.0	43.1	455.2	88.9	58.9
Model 2: Fact. encoder	78.8	43.7	284.4	115.1	17.4
Model 3: Fact. self-attention	77.4	39.1	372.3	117.3	31.7
Model 4: Fact. dot product	76.3	39.5	277.1	88.9	22.9
Model 2: Ave. pool baseline	75.8	38.8	283.9	86.7	17.3

Comparison between model variants

Method	Top 1	Top 5	Views	TFLOPs
blVNet [19]	73.5	91.2	–	–
STM [33]	73.7	91.6	–	–
TEA [42]	76.1	92.5	10 × 3	2.10
TSM-ResNeXt-101 [43]	76.3	–	–	–
I3D NL [75]	77.7	93.3	10 × 3	10.77
CorrNet-101 [70]	79.2	–	10 × 3	6.72
ip-CSN-152 [66]	79.2	93.8	10 × 3	3.27
LGD-3D R101 [51]	79.4	94.4	–	–
SlowFast R101-NL [21]	79.8	93.9	10 × 3	7.02
X3D-XXL [20]	80.4	94.6	10 × 3	5.82
TimeSformer-L [4]	80.7	**94.7**	1 × 3	7.14
ViViT-L/16x2 FE	80.6	92.7	1 × 1	3.98
ViViT-L/16x2 FE	**81.7**	93.8	1 × 3	11.94

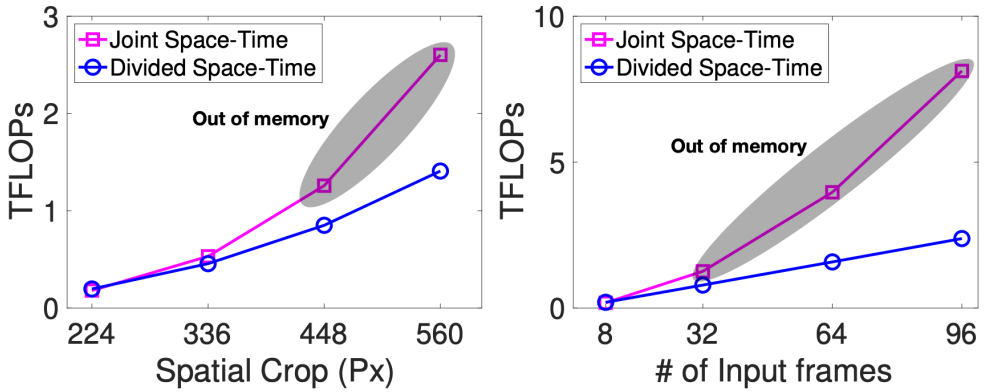
Kinetics-400 dataset benchmark

*source: [Arnab & Dehghani et al., 2021] A Video Vision Transformer, ICCV 2021

Table of Contents

Brute-force joint spatial-temporal attention is intractable for transformers

- Due to the **quadratic complexity** with respect to inputs
- This motivates the development of more efficient attention scheme
  - Time-Space Transformer (TimeSformer) [Bertasius et al., 2021]
  - Video Swin Transformer [Liu et al., 2021]

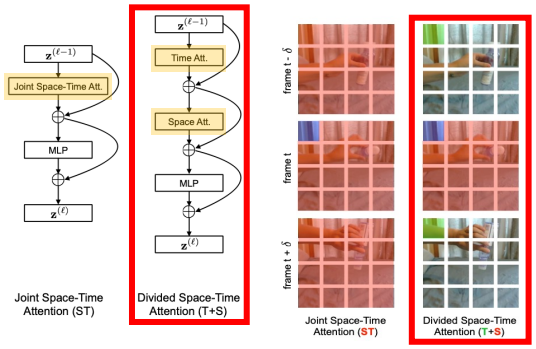


Video classification cost in TFLOPs

Time-Space Transformer (TimeSformer) [Bertasius et al., 2021]

- Proposes **divided space-time attention**
  - Instead of exhaustively comparing all pairs of patches (i.e., joint space-time attention), it separately applies temporal attention and spatial attention one after the other
- **Temporal attention**
  - Each patch (blue) is compared only with the patches at the same spatial location in other frames (green)
  - Initialized to zero (so that function as identity mapping in early training stages)
- **Spatial attention**
  - Each patch (blue) is compared only with the patches within the same frame (red)

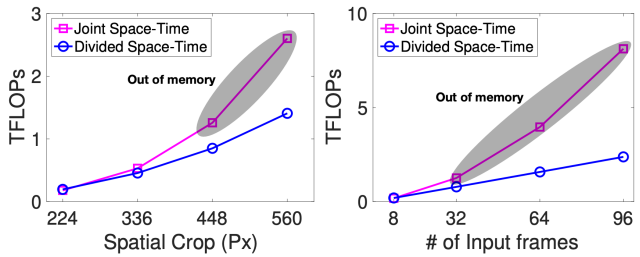
- Designs may look similar to ViViT (model 3) in a big picture, however, implementation details differ including 1) **time– then–space** att., 2) **zero initializations** for temporal layers



*source: [Bertasius et al. 2021] Is Space-Time Attention All You Need for Video Understanding?, ICML 2021

Time-Space Transformer (TimeSformer) [Bertasius et al., 2021]

- Divided space-time attention leads to dramatic computational savings with respect to spatial resolution/video length
- Outperforms SOTA models while requiring less computational complexity
  - $O(S^2T) + O(ST^2)$ instead of $O(S^2T^2)$



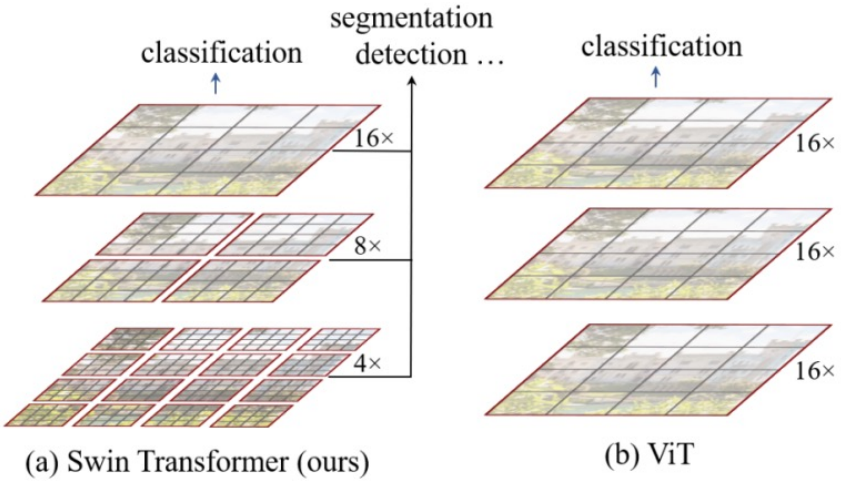
Method	Top-1	Top-5	TFLOPs	
R(2+1)D (Tran et al., 2018)	72.0	90.0	17.5	
bLVNet (Fan et al., 2019)	73.5	91.2	0.84	
TSM (Lin et al., 2019)	74.7	N/A	N/A	
S3D-G (Xie et al., 2018)	74.7	93.4	N/A	
Oct-I3D+NL (Chen et al., 2019)	75.7	N/A	0.84	
D3D (Stroud et al., 2020)	75.9	N/A	N/A	**3D CNNs**
I3D+NL (Wang et al., 2018b)	77.7	93.3	10.8	
ip-CSN-152 (Tran et al., 2019)	77.8	92.8	3.2	
CorrNet (Wang et al., 2020a)	79.2	N/A	6.7	
LGD-3D-101 (Qiu et al., 2019)	79.4	94.4	N/A	
SlowFast (Feichtenhofer et al., 2019b)	79.8	93.9	7.0	
X3D-XXL (Feichtenhofer, 2020)	80.4	94.6	5.8	
TimeSformer	78.0	93.7	**0.59**	**TimeSformer**
TimeSformer-HR	79.7	94.4	5.11	
TimeSformer-L	**80.7**	**94.7**	7.14	

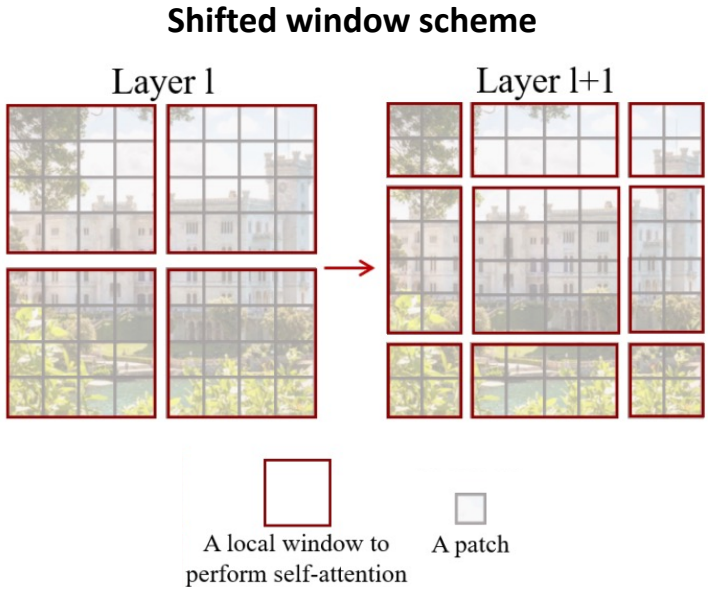
Kinetics-400 dataset benchmark

*source: [Bertasius et al. 2021] Is Space-Time Attention All You Need for Video Understanding?, ICML 2021

Video Swin Transformer [Liu et al., 2021]

- Recall: **Swin Transformer** [Liu et al., 2021]
  - Design of a hierarchical structure
  - Various spatial resolutions (e.g., patch-shape) can be handled via shifted windows
  - Efficient self-attention computation by using shifted windows scheme
  - Concatenating 2 × 2 neighboring patches for downsampling operation
  - Powerful performances in dense prediction tasks
    e.g., object detection and semantic segmentation



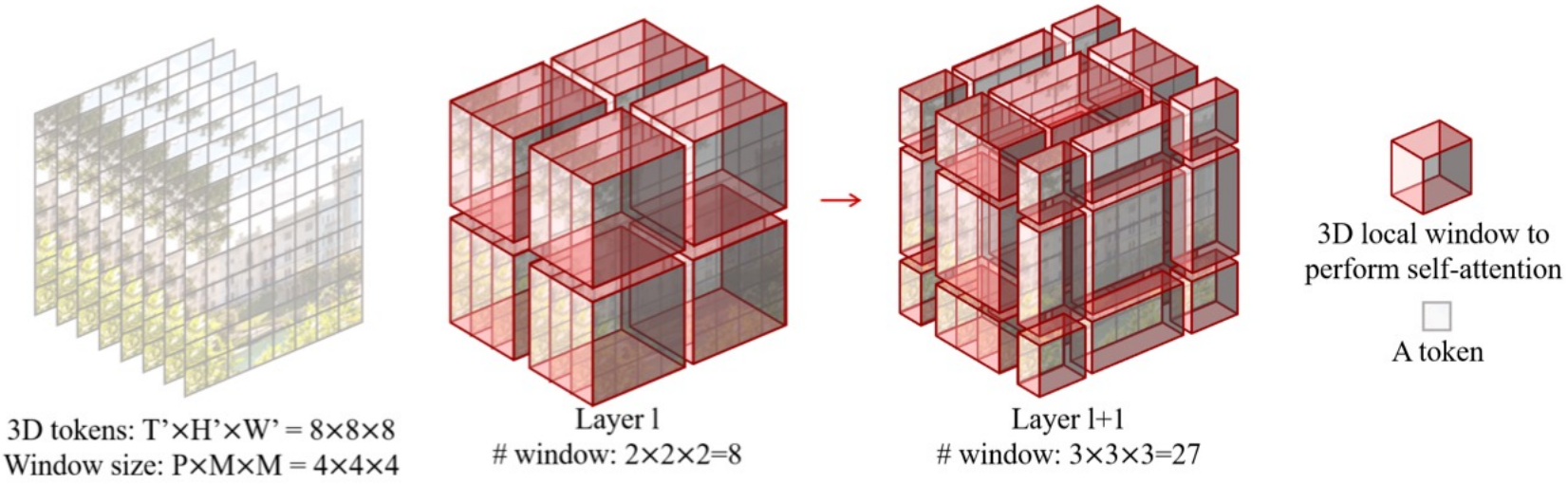


*source: [Liu et al. 2021] Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, ICCV 2021

Video Swin Transformer [Liu et al., 2021]

- In videos, pixels that are closer to each other in spatiotemporal distance are more likely to be correlated (i.e., spatiotemporal locality)
- Thus, **local** attention computation well approximates spatiotemporal self-attention
- Video Swin Transformer is a spatial-**temporal** adaptation of Swin Transformer

  i.e., extension from spatial locality to **spatial-temporal locality**



*source: [Liu et al. 2021] Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, ICCV 2021

Video Swin Transformer [Liu et al., 2021]

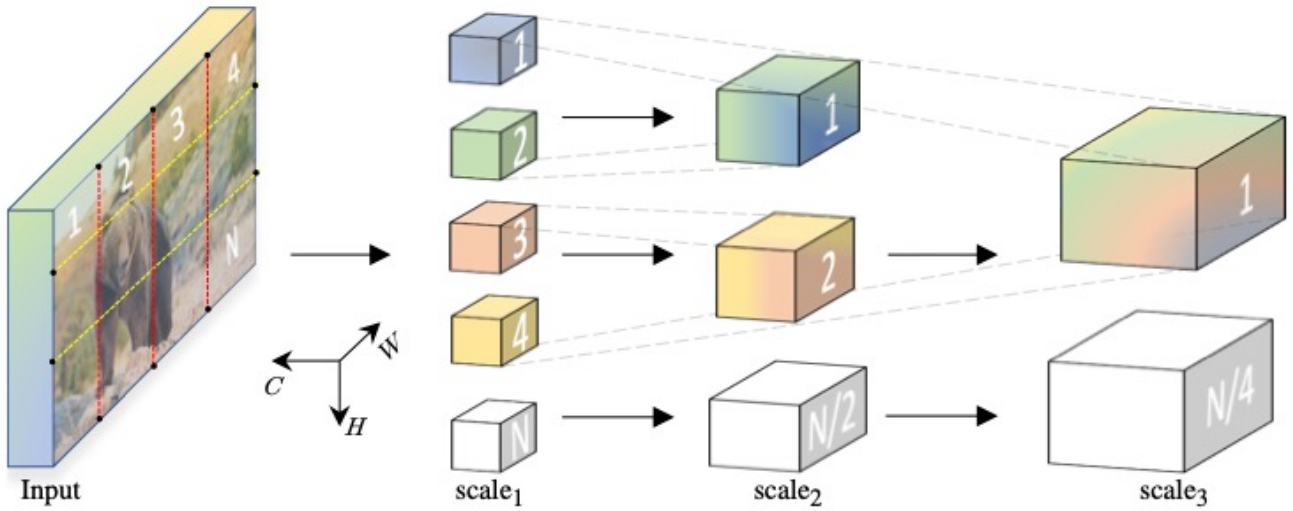
- Outperforms SOTA 3D CNN models while requiring smaller computation costs for inference
- Also outperforms SOTA transformer-based models while requiring half less computational costs

Method	Pretrain	Top-1	Top-5	Views	FLOPs	Param	
R(2+1)D [37]	-	72.0	90.0	10 × 1	75	61.8	3D CNNs
I3D [6]	ImageNet-1K	72.1	90.3	-	108	25.0	
NL I3D-101 [40]	ImageNet-1K	77.7	93.3	10 × 3	359	61.8	
ip-CSN-152 [36]	-	77.8	92.8	10 × 3	109	32.8	
CorrNet-101 [39]	-	79.2	-	10 × 3	224	-	
SlowFast R101+NL [13]	-	79.8	93.9	10 × 3	234	59.9	
X3D-XXL [12]	-	80.4	94.6	10 × 3	144	20.3	
MViT-B, 32×3 [10]	-	80.2	94.4	1 × 5	170	36.6	Transformer-based models
MViT-B, 64×3 [10]	-	81.2	95.1	3 × 3	455	36.6	
TimeSformer-L [3]	ImageNet-21K	80.7	94.7	1 × 3	2380	121.4	
ViT-B-VTN [29]	ImageNet-21K	78.6	93.7	1 × 1	4218	11.04	
ViViT-L/16x2 [1]	ImageNet-21K	80.6	94.7	4 × 3	1446	310.8	
ViViT-L/16x2 320 [1]	ImageNet-21K	81.3	94.7	4 × 3	3992	310.8	
ip-CSN-152 [36]	IG-65M	82.5	95.3	10 × 3	109	32.8	
ViViT-L/16x2 [1]	JFT-300M	82.8	95.5	4 × 3	1446	310.8	
ViViT-L/16x2 320 [1]	JFT-300M	83.5	95.5	4 × 3	3992	310.8	
ViViT-H/16x2 [1]	JFT-300M	84.8	95.8	4 × 3	8316	647.5	
Swin-T	ImageNet-1K	78.8	93.6	4 × 3	88	28.2	Ours
Swin-S	ImageNet-1K	80.6	94.5	4 × 3	166	49.8	
Swin-B	ImageNet-1K	80.6	94.6	4 × 3	282	88.1	
Swin-B	ImageNet-21K	82.7	95.5	4 × 3	282	88.1	
Swin-L	ImageNet-21K	83.1	95.9	4 × 3	604	197.0	
Swin-L (384↑)	ImageNet-21K	84.6	96.5	4 × 3	2107	200.0	
Swin-L (384↑)	ImageNet-21K	**84.9**	**96.7**	10 × 5	2107	200.0	

*source: [Liu et al. 2021] Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, ICCV 2021

Multiscale Vision Transformers (MViT) [Fan et al., 2021]

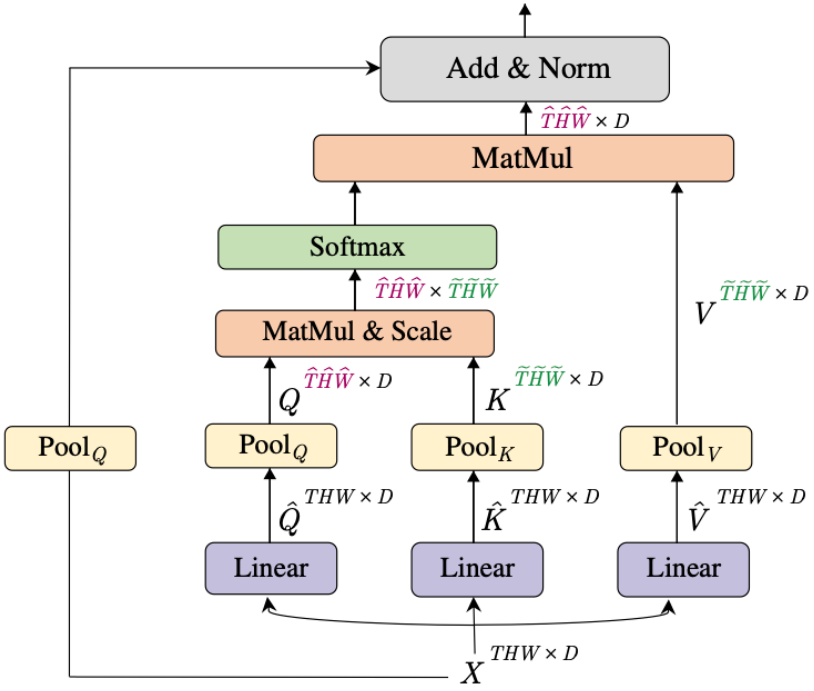
- Utilizes **multiscale channel-resolution stage hierarchy** (pyramidal structure)
- The stages progressively expand channel capacity while reducing spatial resolution
  - Early layers operate at spatially dense resolution & simple low-level features
  - Deeper layers operate at spatially coarse resolution & complex high-dimensional features



*source: [Fan et al. 2021] Multiscale Vision Transformers, ICCV 2021

Multiscale Vision Transformers (MViT) [Fan et al., 2021]

- **Multi Head Pooling Attention**
  - Each stage consists of multiple transformer blocks with specific space-time resolution and channel dimension
  - Pooling **Query** tenors reduces output space-time resolution (down-sampling)
  - Pooling **Key**, **Value** tensors reduces attention computation
  - Channel expansion is done with the MLP block of the previous stage



*source: [Fan et al. 2021] Multiscale Vision Transformers, ICCV 2021

Multiscale Vision Transformers (MViT) [Fan et al., 2021]

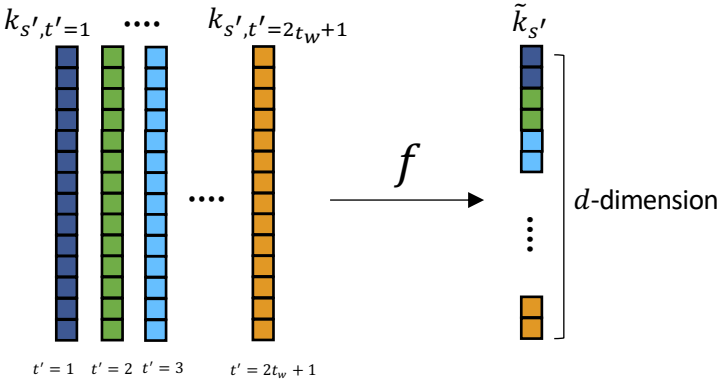
- Without any external pre-training, MViT outperforms both SOTA 3D CNN models & transformer-based models with less parameters and computation

model	pre-train	top-1	top-5	FLOPs×views	Param	
Two-Stream I3D [11]	-	71.6	90.0	216 × NA	25.0	
ip-CSN-152 [96]	-	77.8	92.8	109×3×10	32.8	
SlowFast 8×8 +NL [30]	-	78.7	93.5	116×3×10	59.9	3D CNNs
SlowFast 16×8 +NL [30]	-	79.8	93.9	234×3×10	59.9	
X3D-M [29]	-	76.0	92.3	6.2×3×10	3.8	
X3D-XL [29]	-	79.1	93.9	48.4×3×10	11.0	
ViT-B-VTN [78]	ImageNet-1K	75.6	92.4	4218×1×1	114.0	
ViT-B-VTN [78]	ImageNet-**21K**	78.6	93.7	4218×1×1	114.0	Transformer-based models
ViT-B-TimeSformer [6]	ImageNet-**21K**	80.7	94.7	2380×3×1	121.4	
ViT-L-ViViT [1]	ImageNet-**21K**	81.3	94.7	3992×3×4	310.8	
ViT-B (our baseline)	ImageNet-**21K**	79.3	93.9	180×1×5	87.2	
ViT-B (our baseline)	-	68.5	86.9	180×1×5	87.2	
MViT-S	-	76.0	92.1	32.9×1×5	26.1	
MViT-B, 16×4	-	78.4	93.5	70.5×1×5	36.6	MViT
MViT-B, 32×3	-	80.2	94.4	170×1×5	36.6	
MViT-B, 64×3	-	**81.2**	**95.1**	455×3×3	36.6	

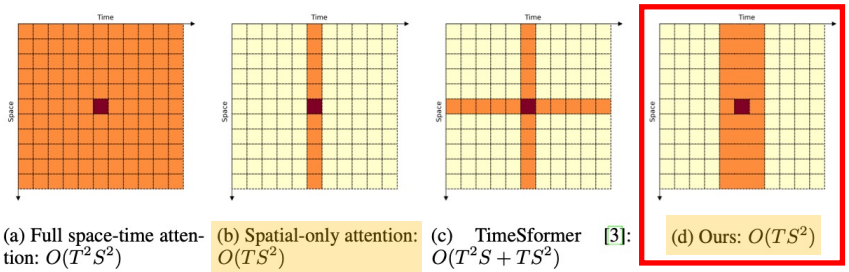
*source: [Fan et al. 2021] Multiscale Vision Transformers, ICCV 2021

X-ViT [Bulat et al., 2021]

- **Space-time mixing attention—$O(TS^2)$ complexity**
  - The following architectural changes in X-ViT **reduce the full quadratic complexity $O(T^2S^2)$** to the proposed $O(TS^2)$
    1. Restricting **attentions within a temporal window** of $[t - t_w, t + t_w]$ for each $q_{s,t}$
       → The complexity becomes $O(T(2t_w + 1)^2S^2)$
    2. Instead of individual space-time keys, the **time compression f** is applied such that a single attention is considered over time with $\tilde{k}_{s'} \triangleq f([k_{s',t-t_w}; \dots; k_{s',t+t_w}])$
    3. Instead of general affine transforms, **"shift trick"** is employed as the implementation of f to further save computations:
       - Given a key $k_{s',t'} \in \mathbb{R}^d$, split its channels into $(2t_w + 1)$ segments, then pick the $t' \in [1, 2t_w + 1]$th index to form the final $\tilde{k}_{s'}$ → The complexity becomes $O(T(2t_w + 1)S^2)$
         Can be disregarded as $2t_w + 1$ is a small constant



The shift trick in X-ViT



X-ViT

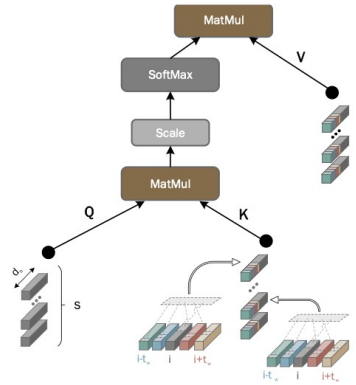
*Red is the query vector
*Orange is the key vector that the query vector attends to

*source: [Bulat et al. 2021] Space-time Mixing Attention for Video Transformer, NeurIPS 2021

X-ViT [Bulat et al., 2021]

- Summary
  - Attentions restricted to within a temporal window
  - Key vector is constructed by mixing tokens from same spatial location within a local temporal window
  - Temporal information is aggregated by indexing subset channels from each token at different temporal locations

- Properties
  - With k transformer blocks, the temporal receptive field becomes $[-kt_w, kt_w]$
    e.g., for a $T = 8$ frames input, $t_w = 1$ and $k = 4$ suffices to achieve the full receptive field
  - Computational complexity scales linearly with number of frames $O(TS^2)$



(b) Proposed space-time mixing attention.

*source: [Bulat et al. 2021] Space-time Mixing Attention for Video Transformer, NeurIPS 2021

X-ViT [Bulat et al., 2021]

- Achieves comparable performance to SOTA models while requiring significantly lower computational complexity
  - **X-ViT (16-frames, 850 GFLOPs)** achieves performance comparable to heavy-weight variants of TimeSformer (96-frames, 7140 GFLOPs) and ViViT (32 frames, 4340 GFLOPs)

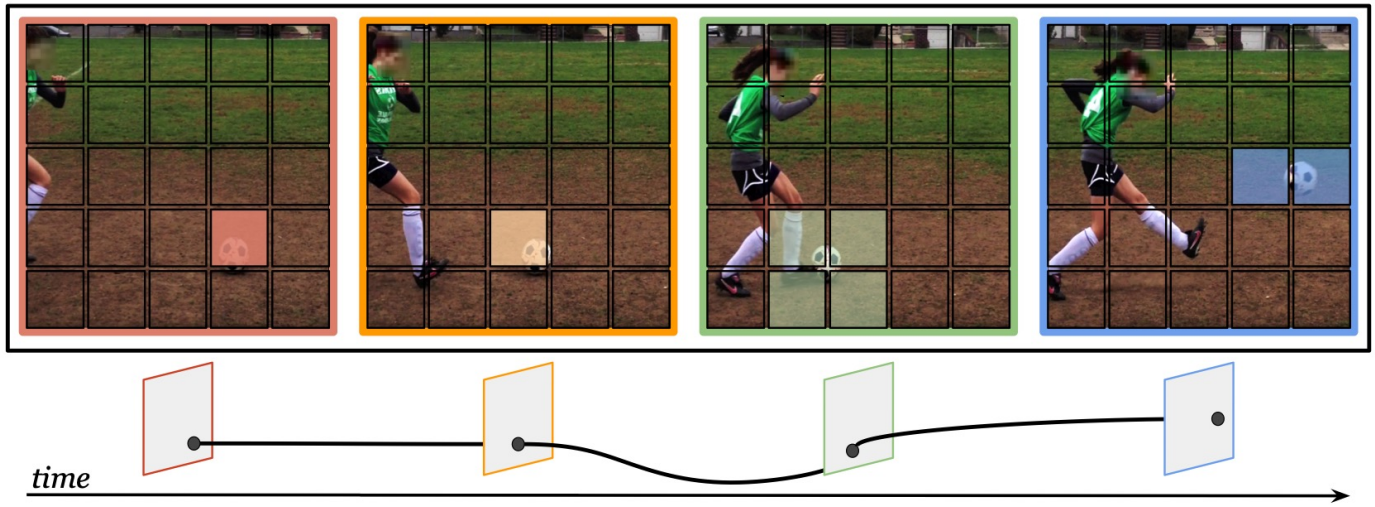
- Allows for an efficient approximation of local space-time attention at no extra cost

Method	Top-1	Top-5	# Frames	Views	Params	FLOPs ($\times 10^9$)
bLVNet [14]	73.5	91.2	24 × 2	3 × 3	25M	840
STM [19]	73.7	91.6	16	-	24M	-
TEA [25]	76.1	92.5	16	10 × 3	25.6M	2,100
TSM R50 [26]	74.7	-	16	10 × 3	25.6M	650
I3D NL [44]	77.7	93.3	128	10 × 3	-	10,800
CorrNet-101 [40]	79.2	-	32	10 × 3	-	6,700
ip-CSN-152 [38]	79.2	93.8	8	10 × 3	-	3,270
LGD-3D R101 [31]	79.4	94.4	16	-	-	-
SlowFast 8×8 R101+NL [16]	78.7	93.5	8	10 × 3	-	3,480
SlowFast 16×8 R101+NL [16]	79.8	93.9	16	10 × 3	-	7,020
X3D-XXL [15]	80.4	94.6	-	10 × 3	20.3M	5,823
TimeSformer-L [3]	**80.7**	94.7	96	1 × 3	121M	7,140
ViViT-L/16x2 [1]	80.6	94.7	32	4 × 3	312M	17,352
X-ViT (Ours)	78.5	93.7	8	1 × 3	92M	425
X-ViT (Ours)	79.4	93.9	8	2 × 3	92M	850
X-ViT (Ours)	80.2	94.7	16	1 × 3	92M	850
X-ViT (Ours)	**80.7**	**94.7**	16	2 × 3	92M	1700

*source: [Bulat et al. 2021] Space-time Mixing Attention for Video Transformer, NeurIPS 2021

Motionformer [Patrick et al., 2021]

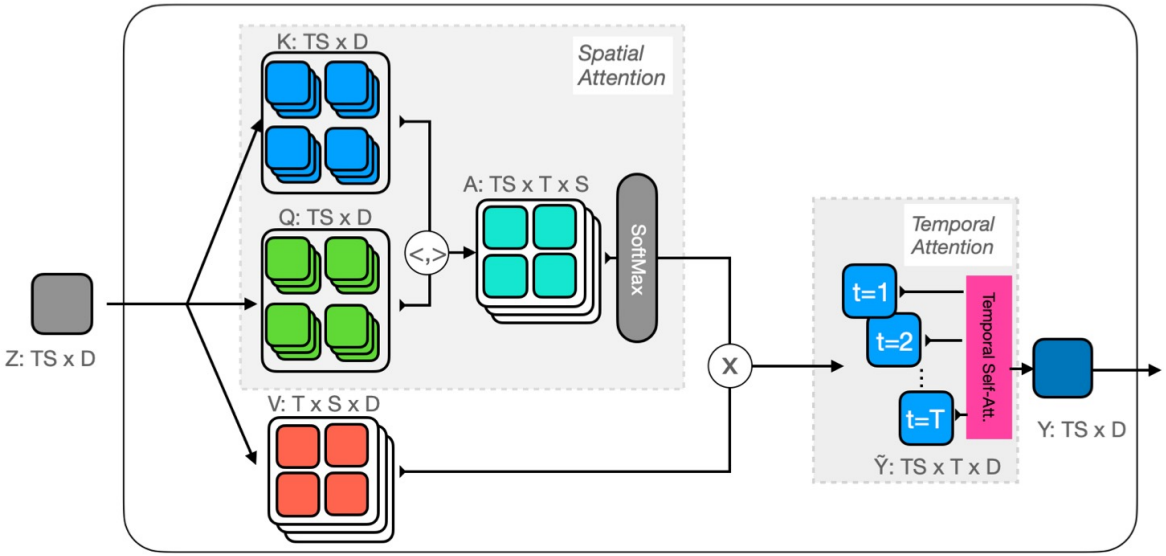
- Depending on object/camera move, physical point at one location may move to diffe rent locations in each frame
- Addressing temporal correspondence, Motionformer proposes **trajectory attention**
  - Aggregates information along implicitly determined motion paths



*source: [Patrick et al. 2021] Keeping Your Eye on the Ball: Trajectory Attention in Video Transformers, NeurIPS 2021

Motionformer [Patrick et al., 2021]

- **Trajectory attention**
  - Aggregates information along implicitly determined motion paths
  - **Spatial attention**
    - Forms a set of ST trajectory tokens for every space-time location
  - **Temporal attention**
    - Pools along those trajectories with a 1D temporal attention operation



*source: [Patrick et al. 2021] Keeping Your Eye on the Ball: Trajectory Attention in Video Transformers, NeurIPS 2021

Motionformer [Patrick et al., 2021]

- Previous works approximate attention structures
  - e.g., divided attention by TimeSformer, locallity-aware attention by Swin Transformer
- **Motionformer** directly attempts to approximate **dot-product attention** itself
  - **Orthoformer algorithm**
    - Approximates attention matrix by selecting most orthogonal subset of queries and keys
    - Allows to significantly improve computational and memory efficiency

1. Randomly subsample R queries and keys to avoid linear dependence on sequence length
2.& 3. Compute two attention matrices Ω_1 and Ω_2 (much smaller than original problem)
4. Multiply them with values

Algorithm 1 Orthoformer (proposed) attention

$\mathbf{P} \leftarrow \text{MostOrthogonalSubset}(\mathbf{Q}, \mathbf{K}, R)$
$\mathbf{\Omega}_1 = \mathcal{S}(\mathbf{Q}^\mathsf{T}\mathbf{P}/\sqrt{D})$
$\mathbf{\Omega}_2 = \mathcal{S}(\mathbf{P}^\mathsf{T}\mathbf{K}/\sqrt{D})$
$\mathbf{Y} = \mathbf{\Omega}_1(\mathbf{\Omega}_2\mathbf{V})$

*source: [Patrick et al. 2021] Keeping Your Eye on the Ball: Trajectory Attention in Video Transformers, NeurIPS 2021

Motionformer [Patrick et al., 2021]

- Motionformer performs favorably against SOTA models
- Achieves strong top-1 accuracy for SSv2 and Epic-Kitchen Nouns datasets, which require greater motion reasoning

(a) **Something–Something V2**

Model	Pretrain	Top-1	Top-5	GFLOPs × views
SlowFast [27]	K-400	61.7	-	65.7×3×1
TSM [51]	K-400	63.4	88.5	62.4×3×2
STM [36]	IN-1K	64.2	89.8	66.5×3×10
MSNet [44]	IN-1K	64.7	89.4	67×1×1
TEA [50]	IN-1K	65.1	-	70×3×10
bLVNet [25]	IN-1K	65.2	90.3	128.6×3×10
VidTr-L [49]	IN-21K+K-400	60.2	-	351×3×10
Tformer-L [8]	IN-21K	62.5	-	1703×3×1
ViViT-L [3]	IN-21K+K-400	65.4	89.8	3992×4×3
MViT-B [24]	K-400	67.1	90.8	170×3×1
Mformer	IN-21K+K-400	66.5	90.1	369.5×3×1
Mformer-L	IN-21K+K-400	**68.1**	**91.2**	1185.1×3×1
Mformer-HR	IN-21K+K-400	67.1	90.6	958.8×3×1

(b) **Kinetics-400**

Method	Pretrain	Top-1	Top-5	GFLOPs × views
I3D [12]	IN-1K	72.1	89.3	108×N/A
R(2+1)D [82]	-	72.0	90.0	152×5×23
S3D-G [94]	IN-1K	74.7	93.4	142.8×N/A
X3D-XL [26]	-	79.1	93.9	48.4×3×10
SlowFast [27]	-	79.8	93.9	234×3×10
VTN [56]	IN-21K	78.6	93.7	4218×1×1
VidTr-L [49]	IN-21K	79.1	93.9	392×3×10
Tformer-L[8]	IN-21K	80.7	94.7	2380×3×1
MViT-B [24]	-	81.2	95.1	455×3×3
ViViT-L [3]	IN-21K	**81.3**	94.7	3992×3×4
Mformer	IN-21K	79.7	94.2	369.5×3×10
Mformer-L	IN-21K	80.2	94.8	1185.1×3×10
Mformer-HR	IN-21K	81.1	**95.2**	958.8×3×10

(c) **Epic-Kitchens**

Method	Pretrain	A	V	N
TSN [85]	IN-1K	33.2	60.2	46.0
TRN [98]	IN-1K	35.3	65.9	45.4
TBN [40]	IN-1K	36.7	66.0	47.2
TSM [51]	IN-1K	38.3	**67.9**	49.0
SlowFast [27]	K-400	38.5	65.6	50.0
ViViT-L [3]	IN-21K+K-400	44.0	66.4	56.8
Mformer	IN-21K+K-400	43.1	66.7	56.5
Mformer-L	IN-21K+K-400	44.1	67.1	57.6
Mformer-HR	IN-21K+K-400	**44.5**	67.0	**58.5**

(d) **Kinetics-600**

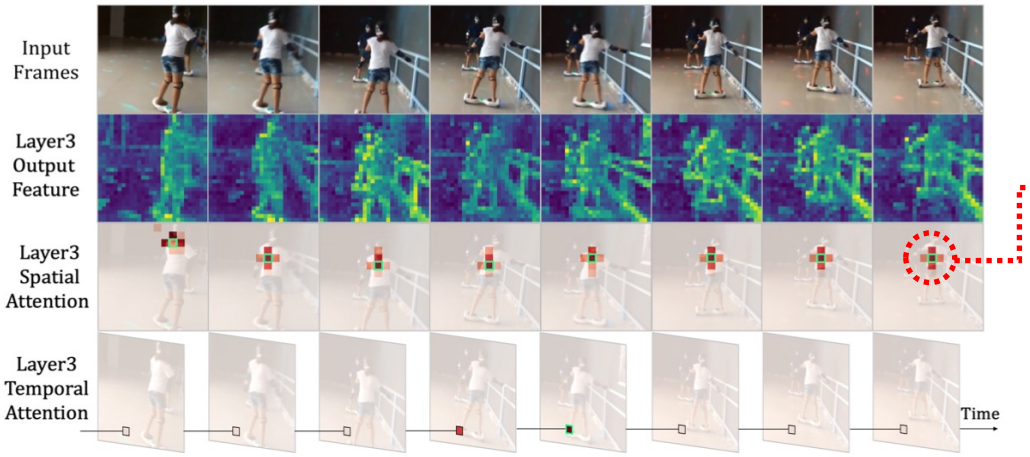
Model	Pretrain	Top-1	Top-5	GFLOPs × views
AttnNAS [89]	-	79.8	94.4	-
LGD-3D [62]	IN-1K	81.5	95.6	-
SlowFast [27]	-	81.8	95.1	234×3×10
X3D-XL [26]	-	81.9	95.5	48.4×3×10
Tformer-HR [8]	IN-21K	82.4	96.0	1703×3×1
ViViT-L [3]	IN-21K	83.0	95.7	3992×3×4
MViT-B-24 [24]	-	**83.8**	**96.3**	236×1×5
Mformer	IN-21K	81.6	95.6	369.5×3×10
Mformer-L	IN-21K	82.2	96.0	1185.1×3×10
Mformer-HR	IN-21K	82.7	96.1	958.8×3×10

*source: [Patrick et al. 2021] Keeping Your Eye on the Ball: Trajectory Attention in Video Transformers, NeurIPS 2021

3D convolutions vs. Vision Transformers

- **3D convolutions**
  - Pro: Can capture detailed local spatiotemporal features to suppress local redundancy
  - Con: Inefficient to capture global (long-range) dependency due to limited receptive field
- **Vision Transformers**
  - Pro: Can capture global (long-range) dependency by self-attention mechanism
  - Con: Inefficient to encode local spatiotemporal feature in shallow layers (local redundancy)

Integrating merits of both, a unified model has been proposed

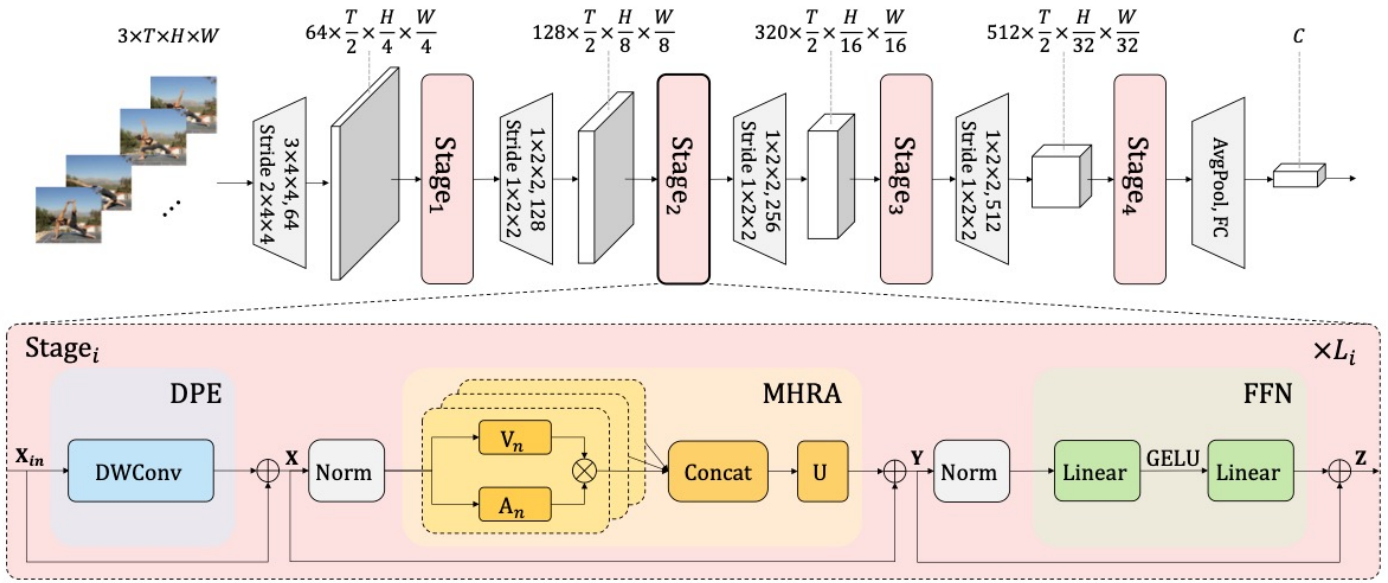


Visualizations of TimeSformer [Bertasius et al., 2021]

- Vision transformer learns local repre sentations **with redundant global** at tention
- This wastes **large computation** to en code only very local spatiotemporal representations

UniFormer [Li et al., 2022]

- Three key modules
  - Dynamic Position Embedding (DPE)
  - Multi-Head Relation Aggregator (MHRA)
  - Feed-Forward Network (FFN)



*source: [Li et al. 2022] Uniformer: Unified Transformer for Efficient Spatiotemporal Representation Learning, ICLR 2022

UniFormer [Li et al., 2022]

- **Dynamic Position Embedding (DPE)**
  - Previous spatiotemporal position embedding methods:
  - **Absolute position embedding** cannot handle different input sizes because it is interpolated to target input size with fine-tuning
  - **Relative position embedding** modifies self-attention and performs worse due to lack of absolute position embedding

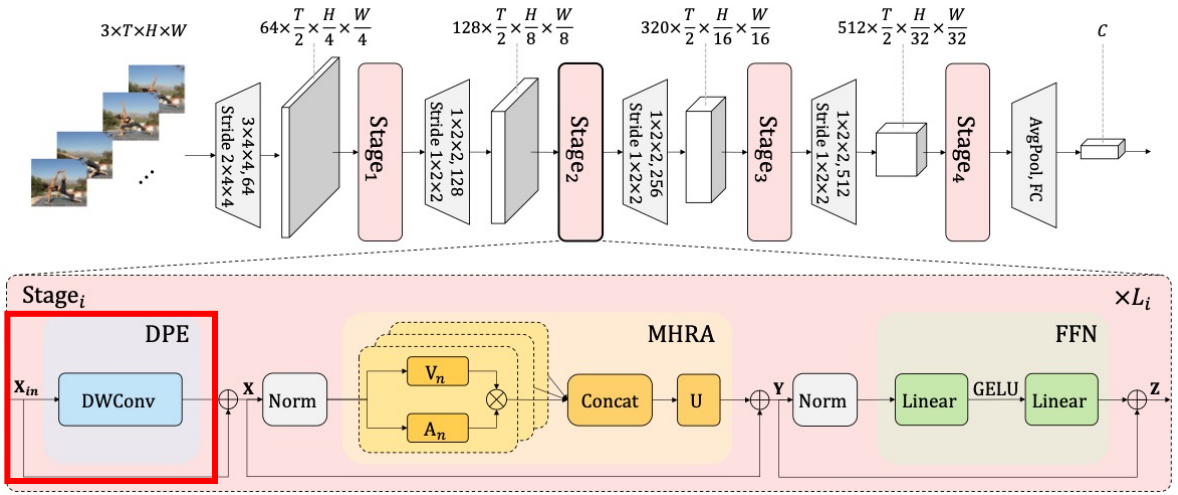
*source: [Li et al. 2022] Uniformer: Unified Transformer for Efficient Spatiotemporal Representation Learning, ICLR 2022

UniFormer [Li et al., 2022]

- **Dynamic Position Embedding (DPE)**
  - To overcome these problems, conditional position encoding (CPE) is extended to dynamic position embedding (DPE)

$$\mathrm{DPE}(\mathbf{X}_{in}) = \mathrm{DWConv}(\mathbf{X}_{in})$$

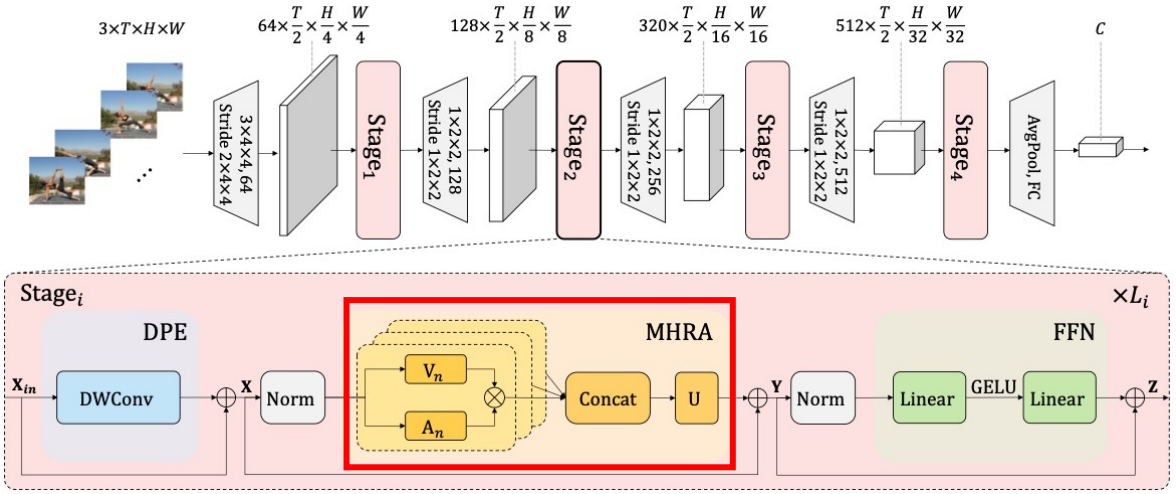
  - DPE dynamically integrates 3D position information into all tokens
  - ***DWConv*** is a simple 3D depth-wise convolution with zero paddings
    - Shared parameters & locality of convolution tackles permutation-invariance
    - In CPE, zero paddings help tokens on the borders be aware of their absolute positions
    - That is, all tokens progressively encode their position information via querying their neighbor



*source: [Li et al. 2022] Uniformer: Unified Transformer for Efficient Spatiotemporal Representation Learning, ICLR 2022

UniFormer [Li et al., 2022]

- **Multi-Head Relation Aggregator (MHRA)**
  - $\boldsymbol{V_n} \in \mathbb{R}^{L \times \frac{C}{N}}$: token context encoding that transforms original token into context via linear transformation ($L = T \times H \times W$)
  - $\boldsymbol{A_n}$: token affinity learning that summarizes context with guidance of token affinity
  - $\boldsymbol{R_n(X)} = \boldsymbol{A_n V_n(X)}$: the relation aggregator (RA) in the n-th head
  - $\boldsymbol{U} \in \mathbb{R}^{C \times C}$: learnable parameter matrix that integrates N heads
  - Tackles **local redundancy** & **global dependency** problems by flexibly designing A_n



*source: [Li et al. 2022] Uniformer: Unified Transformer for Efficient Spatiotemporal Representation Learning, ICLR 2022

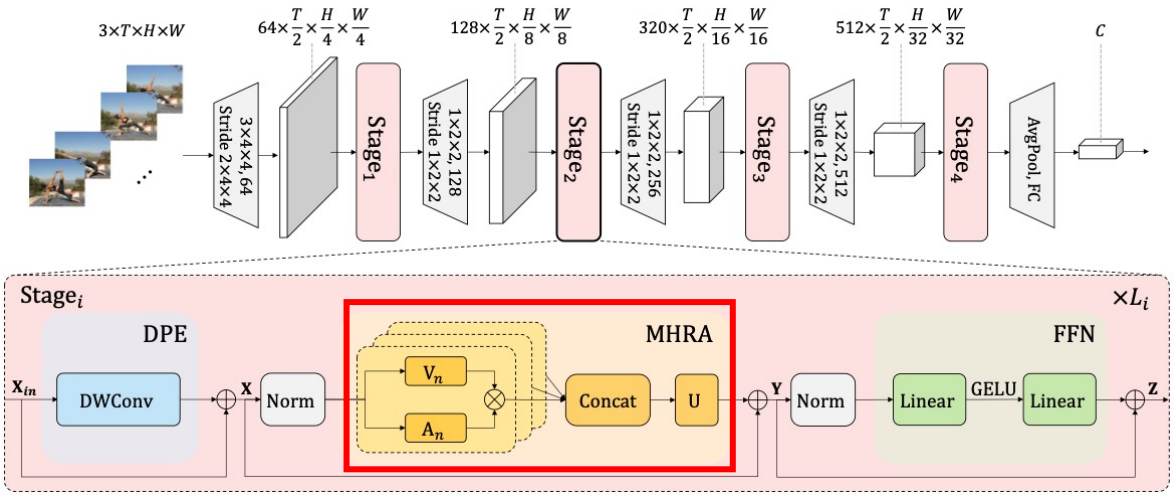
UniFormer [Li et al., 2022]

- **Multi-Head Relation Aggregator (MHRA)**

1) Local MHRA (for shallow layers)

- Aim for shallow layers is to learn detailed video representation from local spatiotemporal context to reduce redundancy
- Design token affinity to be local learnable parameter matrix, which depends only on relative 3D position between tokens
- RA learns local spatiotemporal affinity between one anchor token X_i and other tokens in the small tube $\Omega_i^{t\times h\times w}$

$$\mathrm{A}_n^{local}(\mathbf{X}_i, \mathbf{X}_j) = a_n^{i-j}, \;\; where \;\; j \in \Omega_i^{t\times h\times w}$$



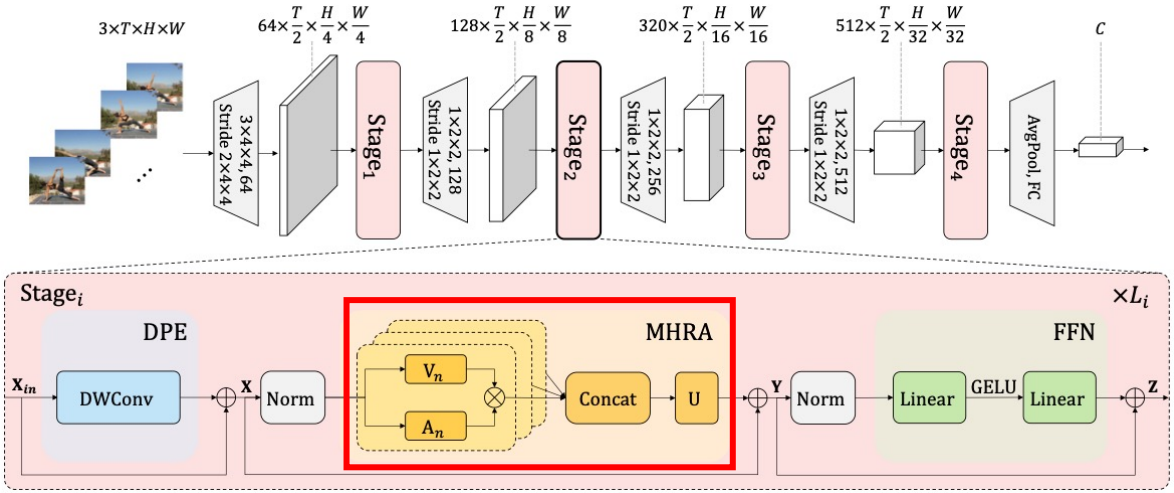
*source: [Li et al. 2022] Uniformer: Unified Transformer for Efficient Spatiotemporal Representation Learning, ICLR 2022

UniFormer [Li et al., 2022]

- **Multi-Head Relation Aggregator (MHRA)**
  - **2) Global MHRA** (for deep layers)
    - Aim for deep layers is to capture long-term token dependency in global video clip
    - Design token affinity via comparing content similarity among all tokens in global view

$$\mathrm{A}_n^{global}(\mathbf{X}_i, \mathbf{X}_j) = \frac{e^{Q_n(\mathbf{X}_i)^T K_n(\mathbf{X}_j)}}{\sum_{j' \in \Omega_{T\times H\times W}} e^{Q_n(\mathbf{X}_i)^T K_n(\mathbf{X}_{j'})}}$$

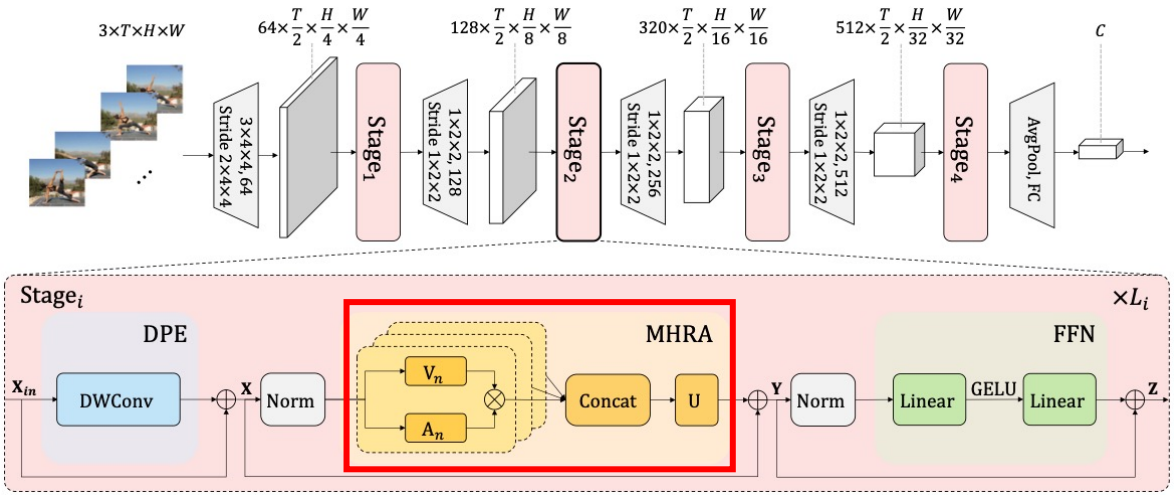
- X_j can be any token in global 3D tube $\Omega_{T\times H\times W}$
- $Q_n(\cdot)$ and $K_n(\cdot)$ are two different linear transformations



*source: [Li et al. 2022] Uniformer: Unified Transformer for Efficient Spatiotemporal Representation Learning, ICLR 2022

UniFormer [Li et al., 2022]

- **Multi-Head Relation Aggregator (MHRA)**
  - Most video transformers requires large amount of calculation because they apply self-attention in all stages
  - While dividing spatial & temporal attention reduces dot-product computation, it deteriorates spatiotemporal relation among tokens
  - MHRA saves computation by performing local relation aggregation in early layers



*source: [Li et al. 2022] Uniformer: Unified Transformer for Efficient Spatiotemporal Representation Learning, ICLR 2022

UniFormer [Li et al., 2022]

- Uniformer outperforms most of the current methods with much fewer computational cost
- Achieves a preferable balance between computation and accuracy

Method	Pretrain	#Frame	GFLOPs	SSV1		SSV2	
				Top-1	Top-5	Top-1	Top-5
TSN(Wang et al., 2016)	IN-1K	16×1×1	66	19.9	47.3	30.0	60.5
TSM(Lin et al., 2019)	IN-1K	16×1×1	66	47.2	77.1	-	-
GST(Luo & Yuille, 2019)	IN-1K	16×1×1	59	48.6	77.9	62.6	87.9
MSNet(Kwon et al., 2020)	IN-1K	16×1×1	101	52.1	82.3	64.7	89.4
CT-Net(Li et al., 2021a)	IN-1K	16×1×1	75	52.5	80.9	64.5	89.3
CT-Net$_{EN}$(Li et al., 2021a)	IN-1K	8+12+16+24	280	56.6	83.9	67.8	91.1
TDN(Wang et al., 2020b)	IN-1K	16×1×1	72	53.9	82.1	65.3	89.5
TDN$_{EN}$(Wang et al., 2020b)	IN-1K	8+16	198	56.8	84.1	68.2	91.6
TimeSformer-HR(Bertasius et al., 2021)	IN-21K	16×3×1	5109	-	-	62.5	-
X-ViT(Bulat et al., 2021)	IN-21K	32×3×1	1270	-	-	65.4	90.7
Mformer-L(Patrick et al., 2021)	K400	32×3×1	3555	-	-	68.1	91.2
ViViT-L(Arnab et al., 2021)	K400	16×3×4	11892	-	-	65.4	89.8
MViT-B,64×3(Fan et al., 2021)	K400	64×1×3	1365	-	-	67.7	90.9
MViT-B-24,32×3(Fan et al., 2021)	K600	32×1×3	708	-	-	68.7	91.5
Swin-B(Liu et al., 2021b)	K400	32×3×1	963	-	-	69.6	92.7
Our UniFormer-S	K400	16×1×1	42	53.8	81.9	63.5	88.5
Our UniFormer-S	K600	16×1×1	42	54.4	81.8	65.0	89.3
Our UniFormer-S	K400	16×3×1	125	57.2	84.9	67.7	91.4
Our UniFormer-S	K600	16×3×1	125	57.6	84.9	69.4	92.1
Our UniFormer-B	K400	16×3×1	290	59.1	86.2	70.4	92.8
Our UniFormer-B	K600	16×3×1	290	58.8	86.5	70.2	**93.0**
Our UniFormer-B	K400	32×3×1	777	60.9	87.3	**71.2**	92.8
Our UniFormer-B	K600	32×3×1	777	**61.0**	**87.6**	**71.2**	92.8

*source: [Li et al. 2022] Uniformer: Unified Transformer for Efficient Spatiotemporal Representation Learning, ICLR 2022

Summary

- For spatial-temporal data, one need a specific vision architecture for processing temporal dependency between frames

- CNN architectures for video have developed in a way that
  - Can better model **motion information** in sequence of frames
  - **Multiteam architectures**, **Temporal segment networks**, and **3D CNNs** are key advances for CNNs for modeling spatial-temporal data

- Recently, **Transformer** is actively applied to video recognition
  - As in other sequential tasks, transformer's ability to model **long-range dependencies** largely benefits video recognition performance
  - For efficiency, **approximated attention** mechanisms enable video transformers to process spatial-temporal data under limited computation resources

- Transformer-based video model is rapidly becoming a de-facto standard

References

[Karpathy et al., 2014] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732).
link : https://ieeexplore.ieee.org/document/6909619

[Pan et al., 2020] Pan, Y., Li, Y., Luo, J., Xu, J., Yao, T., & Mei, T. (2020). Auto-captions on GIF: A Large-scale Video-sentence Dataset for Vision-language Pre-training. arXiv preprint arXiv:2007.02375.
link : https://arxiv.org/abs/2007.02375

[Pan et al., 2020] Wang, Z., Zheng, L., Liu, Y., Li, Y., & Wang, S. (2020, August). Towards real-time multi-object tracking. In European Conference on Computer Vision (pp. 107-122). Springer, Cham.
link : https://link.springer.com/chapter/10.1007/978-3-030-58621-8_7

[Rhinehart et al., 2019] Rhinehart, N., McAllister, R., Kitani, K., & Levine, S. (2019). Precog: Prediction conditioned on goals in visual multi-agent settings. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 2821-2830).
link : https://arxiv.org/abs/1905.01296

[Dosovitskiy et al., 2021] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
link : https://arxiv.org/abs/2010.11929

[He et al., 2016] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
link : https://ieeexplore.ieee.org/document/7780459/

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
link : http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

References

[Simonyan and Zisserman, 2014] Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems, 27.
link : https://proceedings.neurips.cc/paper/2014/hash/00ec53c4682d36f5c4359f4ae7bd7ba1-Abstract.html

[Ilg et al., 2017] Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., & Brox, T. (2017). Flownet 2.0: Evolution of optical flow estimation with deep networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2462-2470).
link : https://openaccess.thecvf.com/content_cvpr_2017/html/Ilg_FlowNet_2.0_Evolution_CVPR_2017_paper.html

[Zhu et al., 2020] Zhu, Y., Li, X., Liu, C., Zolfaghari, M., Xiong, Y., Wu, C., ... & Li, M. (2020). A comprehensive study of deep video action recognition. *arXiv preprint arXiv:2012.06567*.
link : https://arxiv.org/abs/2012.06567

[Wang et al., 2016] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., & Gool, L. V. (2016, October). Temporal segment networks: Towards good practices for deep action recognition. In European conference on computer vision (pp. 20-36).
link : https://arxiv.org/abs/1608.00859

[Feichtenhofer et al., 2016a] Feichtenhofer, C., Pinz, A., & Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1933-1941).
link : https://arxiv.org/abs/1604.06573

[Feichtenhofer et al., 2016b] Christoph, R., & Pinz, F. A. (2016). Spatiotemporal residual networks for video action recognition. Advances in neural information processing systems, 3468-3476.
link : https://arxiv.org/abs/1611.02155

[Wang et al., 2017] Wang, Y., Long, M., Wang, J., & Yu, P. S. (2017). Spatiotemporal pyramid network for video action recognition. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1529-1538).
link : https://arxiv.org/abs/1903.01038

References

[Ioffe et al., 2015] Ioffe, S. & Szegedy, C.. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Proceedings of the 32nd International Conference on Machine Learning, in PMLR 37:448-456
link : http://proceedings.mlr.press/v37/ioffe15.html

[Zhou et al., 2018] Zhou, B., Andonian, A., Oliva, A., & Torralba, A. (2018). Temporal relational reasoning in videos. In Proceedings of the European conference on computer vision (ECCV) (pp. 803-818).
link : https://arxiv.org/abs/1711.08496

[Diba et al., 2017] Diba, A., Sharma, V., & Van Gool, L. (2017). Deep temporal linear encoding networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 2329-2338).
link : https://openaccess.thecvf.com/content_cvpr_2017/html/Diba_Deep_Temporal_Linear_CVPR_2017_paper.html

[Lan et al., 2017] Lan, Z., Zhu, Y., Hauptmann, A. G., & Newsam, S. (2017). Deep local video feature for action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 1-7).
link : https://arxiv.org/abs/1701.07368

[Tran et al., 2015] Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 4489-4497).
link : https://arxiv.org/abs/1412.0767

[Ji et al., 2012] Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence, 35(1), 221-231.
link : https://ieeexplore.ieee.org/abstract/document/6165309/

[Carreira and Zisserman, 2017] Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6299-6308).
link : https://arxiv.org/abs/1705.07750

References

[Kay et al., 2017] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., ... & Zisserman, A. (2017). The kinetics human action video dataset. arXiv preprint arXiv:1705.06950.
link : https://arxiv.org/abs/1705.06950

[Tran et al., 2017] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 6450-6459).
link : https://arxiv.org/abs/1711.11248

[Qiu et al., 2018] Qiu, Z., Yao, T., & Mei, T. (2017). Learning spatio-temporal representation with pseudo-3d residual networks. In proceedings of the IEEE International Conference on Computer Vision (pp. 5533-5541).
link : https://arxiv.org/abs/1711.10305

[Tran et al., 2019] Tran, D., Wang, H., Torresani, L., & Feiszli, M. (2019). Video classification with channel-separated convolutional networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 5552-5561).
link : https://arxiv.org/abs/1904.02811

[Donahue et al., 2015] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2625-2634).
link : https://arxiv.org/abs/1411.4389

[Ng et al., 2015] Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4694-4702).
link : https://arxiv.org/abs/1503.08909

References

[Neimark et al., 2021] Neimark, D., Bar, O., Zohar, M., & Asselmann, D. (2021). Video transformer network. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3163-3172).
link : http://arxiv.org/abs/2102.00719

[Arnab et al., 2021] Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). Vivit: A video vision transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 6836-6846).
link : https://arxiv.org/abs/2103.15691

[Bertasius et al., 2021] Bertasius, G., Wang, H., & Torresani, L. (2021). Is space-time attention all you need for video understanding. arXiv preprint arXiv:2102.05095, 2(3), 4.
link : https://arxiv.org/abs/2102.05095

[Liu et al., 2021] Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., & Hu, H. (2021). Video swin transformer. arXiv preprint arXiv:2106.13230.
link : https://arxiv.org/abs/2106.13230

[Bulat et al., 2021] Bulat, A., Perez Rua, J. M., Sudhakaran, S., Martinez, B., & Tzimiropoulos, G. (2021). Space-time mixing attention for video transformer. Advances in Neural Information Processing Systems, 34.
link : https://proceedings.neurips.cc/paper/2021/hash/a34bacf839b923770b2c360eefa26748-Abstract.html

[Fan et al., 2021] Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., & Feichtenhofer, C. (2021). Multiscale vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 6824-6835).
link : https://arxiv.org/abs/2104.11227

[Li et al., 2022] Li, K., Wang, Y., Gao, P., Song, G., Liu, Y., Li, H., & Qiao, Y. (2022). Uniformer: Unified Transformer for Efficient Spatiotemporal Representation Learning. arXiv preprint arXiv:2201.04676.
link : https://arxiv.org/abs/2201.04676

References

[Shi et al., 2015] Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems, 28.
link: https://arxiv.org/abs/1506.04214

[Sun et al., 2017] Sun, L., Jia, K., Chen, K., Yeung, D. Y., Shi, B. E., & Savarese, S. (2017). Lattice long short-term memory for human action recognition. In Proceedings of the IEEE international conference on computer vision (pp. 2147-2156).
link : https://arxiv.org/abs/1708.03958

[Zhu et al., 2020] Zhu, L., Tran, D., Sevilla-Lara, L., Yang, Y., Feiszli, M., & Wang, H. (2020, April). Faster recurrent networks for efficient video classification. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 07, pp. 13098-13105).
link : https://arxiv.org/abs/1906.04226

[Hu et al., 2018] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7132-7141).
link : https://arxiv.org/abs/1709.01507

[Dosovitskiy et al., 2021] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
link : https://arxiv.org/abs/2010.11929

[Liu et al., 2021] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 10012-10022).
link :
https://openaccess.thecvf.com/content/ICCV2021/html/Liu_Swin_Transformer_Hierarchical_Vision_Transformer_Using_Shifted_Windows_ICCV_2021_paper.html

References

[Heo et al., 2021] Heo, B., Yun, S., Han, D., Chun, S., Choe, J., & Oh, S. J. (2021). Rethinking spatial dimensions of vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 11936-11945).
link : https://arxiv.org/abs/2103.16302

[Li et al., 2021] Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z. H., ... & Yan, S. (2021). Tokens-to-token vit: Training vision transformers from scratch on imagenet. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 558-567).
link : https://arxiv.org/abs/2101.11986

[Kim et al., 2019] Kim, D., Cho, D., & Kweon, I. (2019). Self-Supervised Video Representation Learning with Space-Time Cubic Puzzles. AAAI.
link : https://arxiv.org/abs/1811.09795

[Huang et al., 2021] Huang, L., Liu, Y., Wang, B., Pan, P., Xu, Y., & Jin, R. (2021). Self-supervised Video Representation Learning by Context and Motion Decoupling. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 13881-13890.
link : https://arxiv.org/abs/2104.00862

[Rakhimov et al., 2021] Rakhimov, R., Volkhonskiy, D., Artemov, A., Zorin, D., & Burnaev, E. (2021). Latent Video Transformer. VISIGRAPP.
link : https://arxiv.org/abs/2006.10704

[Weisseborn et al., 2020] Weissenborn, D., Täckström, O., & Uszkoreit, J. (2020). Scaling Autoregressive Video Models. ArXiv, abs/1906.02634.
link : https://arxiv.org/abs/1906.02634

References

[Dorkenwald et al., 2021] Dorkenwald, M., Milbich, T., Blattmann, A., Rombach, R., Derpanis, K.G., & Ommer, B. (2021). Stochastic Image-to-Video Synthesis using cINNs. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3741-3752.
link : https://arxiv.org/abs/2105.04551

[Yan et al., 2021] Yan, W., Zhang, Y., Abbeel, P., & Srinivas, A. (2021). VideoGPT: Video Generation using VQ-VAE and Transformers. *ArXiv, abs/2104.10157*.
link : https://arxiv.org/abs/2104.10157

[Li et al., 2021] Li, Y., Li, S., Sitzmann, V., Agrawal, P., & Torralba, A. (2021). 3D Neural Scene Representations for Visuomotor Control. *ArXiv, abs/2107.04004*.
link : https://arxiv.org/abs/2107.04004

[Skorokhodov et al., 2021] Skorokhodov, I., Tulyakov, S., & Elhoseiny, M. (2021). StyleGAN-V: A Continuous Video Generator with the Price, Image Quality and Perks of StyleGAN2. *ArXiv, abs/2112.14683*.
link : https://arxiv.org/abs/2112.14683

[Yu et al., 2022] Yu, S., Tack, J., Mo, S., Kim, H., Kim, J., Ha, J., & Shin, J. (2022). Generating Videos with Dynamics-aware Implicit Generative Adversarial Networks. In International Conference on Learning Representations.
link : https://arxiv.org/abs/2202.10571

[Li et al., 2021] Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., & Lv, Z. (2021). Neural 3D Video Synthesis. *ArXiv, abs/2103.02597*.
link : https://arxiv.org/abs/2103.02597

[Liu et al., 2019] Liu, X., Qi, C., & Guibas, L.J. (2019). FlowNet3D: Learning Scene Flow in 3D Point Clouds. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 529-537.
link : https://arxiv.org/abs/1806.01411