

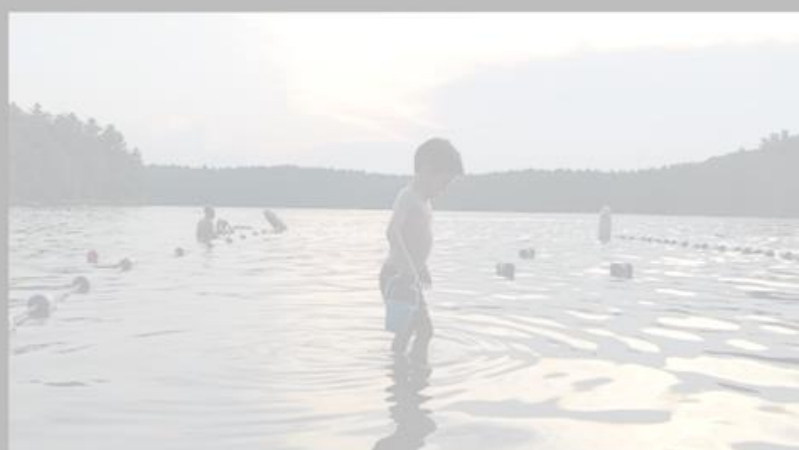


Learning from Videos



Feb 19th, 2024

Tali Dekel



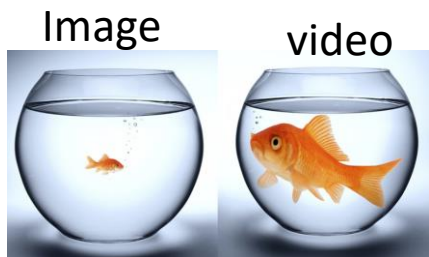
Videos

Videos are all around us

Span an enormous space of spatial and temporal signals



Challenges in Videos: size of video



Size of video \gg size of image

Computational constraints \rightarrow short, low-res clips

$3 \times H \times W$



4D tensor:
 $T \times 3 \times H \times W$
time



~ 30 frames per second (fps)

Uncompressed size (3 bytes per pixel):
SD (640 x 480): ~ 1.5 GB per minute
HD (1920 x 1080): ~ 10 GB per minute

Reduce spatial and temporal resolution



5fps, half the spatial resolution

Challenges in Videos: size of video

Computational constrains → short, low-res clips



Input video



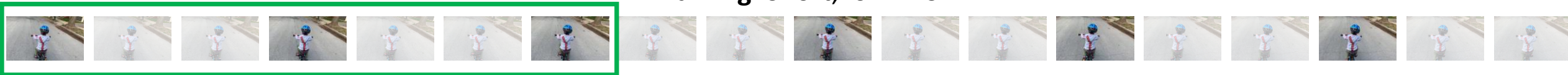
Walking
Running
Cycling
Jumping

·
·

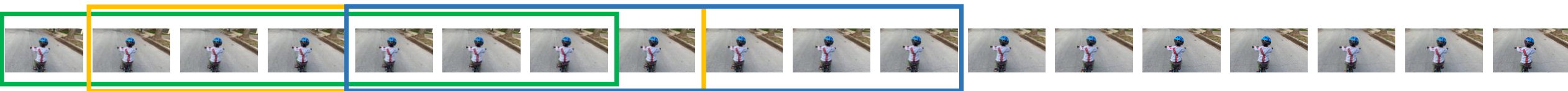
Original video(long, high FPS)



Training: Short, low FPS



Test time: inference on different short clips, average predictions



Challenges in Videos: Videos Datasets

space of video \gg space of image \rightarrow lots of training data

“ImageNet”-equivalent dataset for videos?

Massive human labelling efforts



UCF101

YouTube videos

13320 videos, 101 action categories



Kinetics

YouTube videos

650,000 video clips, 600 human action classes



Sports-1M

YouTube videos

1,133,157 videos, 487 sports labels



YouTube-8M

8M video clips, Machine-generated annotations from 3,862 classes

Today

Deep Learning-based Models for Videos

- How to reduce computation cost without sacrificing accuracy?
- What architecture to best capture temporal patterns?

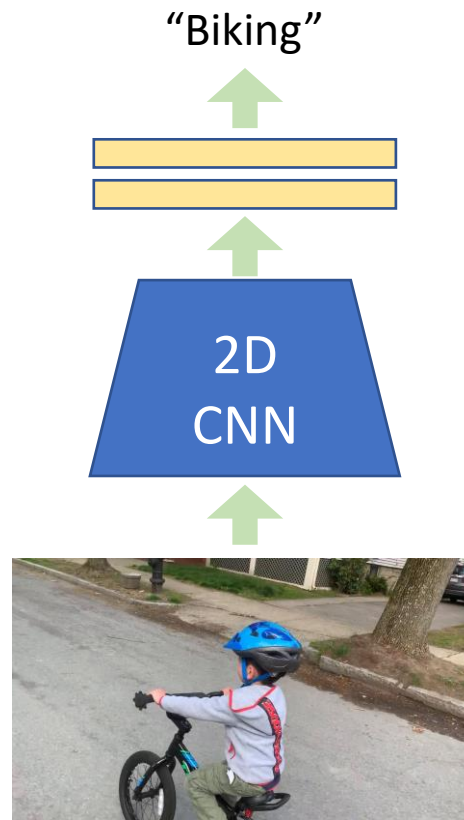
Karpathy et. al., Large-scale Video Classification with Convolutional Neural Networks, CVPR, 2014

Self-Supervision in Videos

- Which types of pretext tasks can we define to capture temporal information?
- Learning from a single video and neural video representation

Models for Videos: Single-Frame Baseline

- Train 2D CNN to classify video frames independently

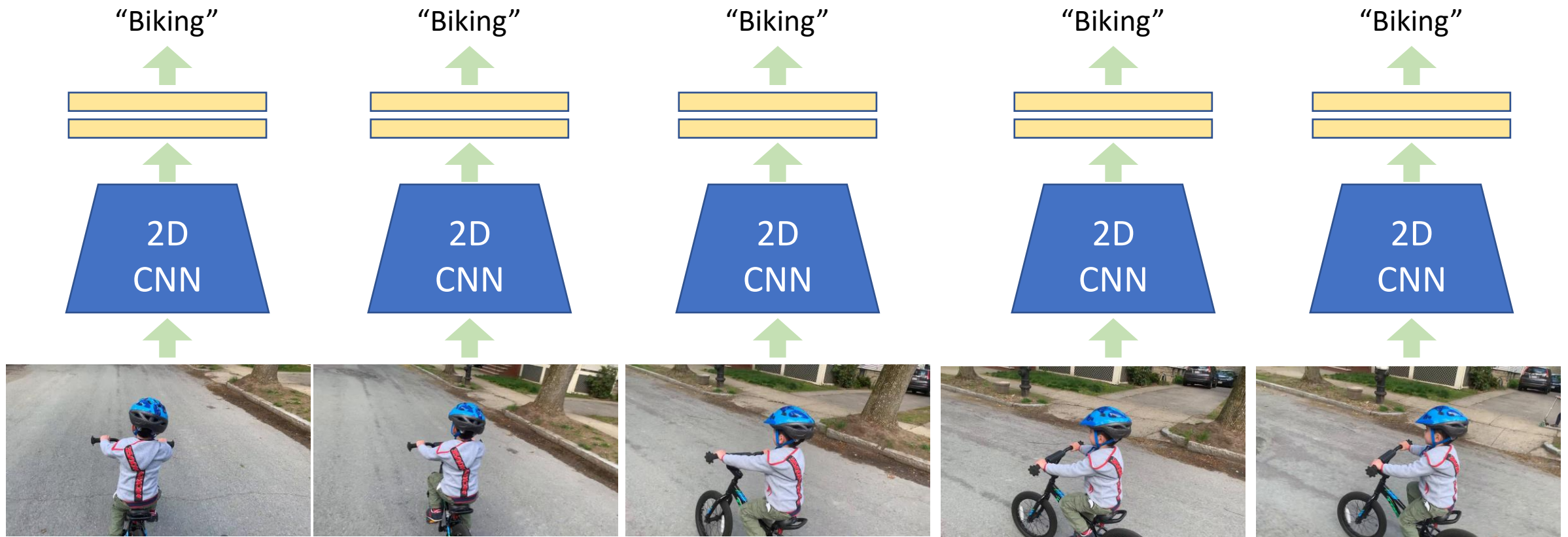


Input video frame

Models for Videos: Single-Frame Baseline

- Train 2D CNN to classify video frames independently
- Average predicted probs at test-time

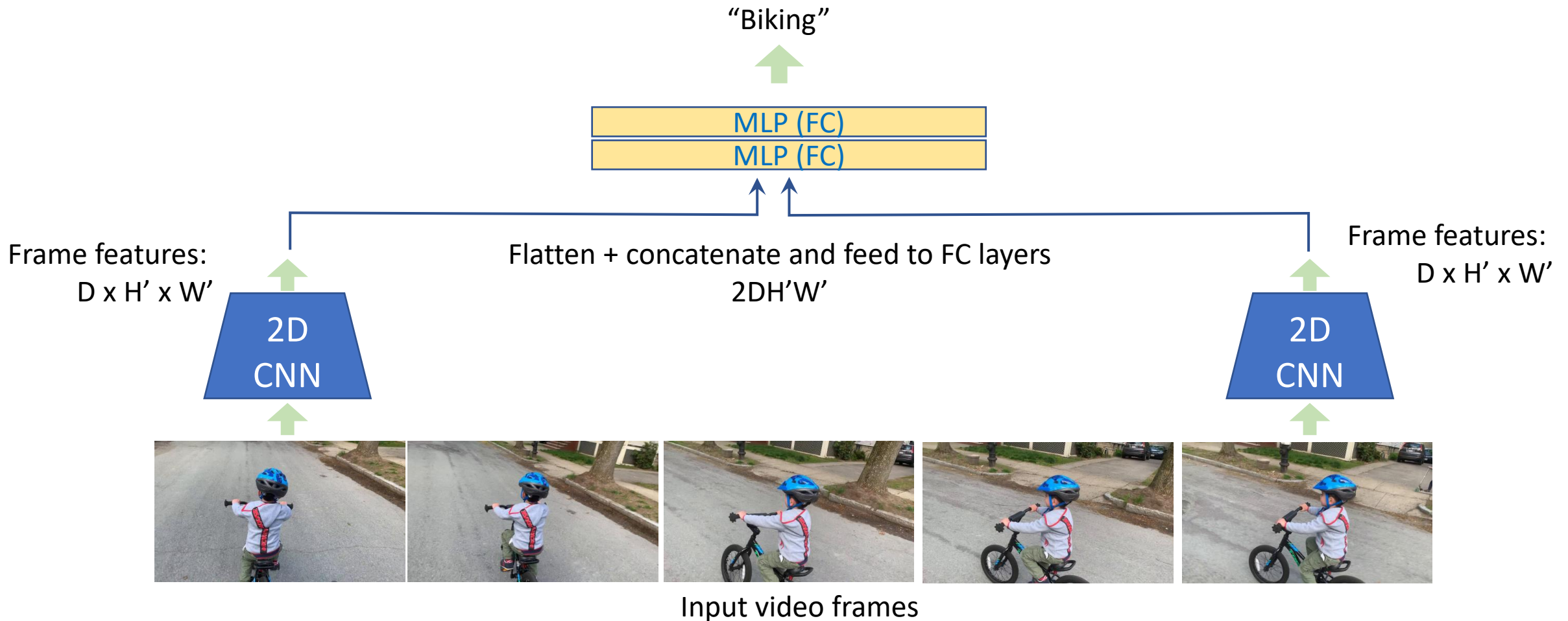
Often a surprisingly strong baseline!



Input video frames

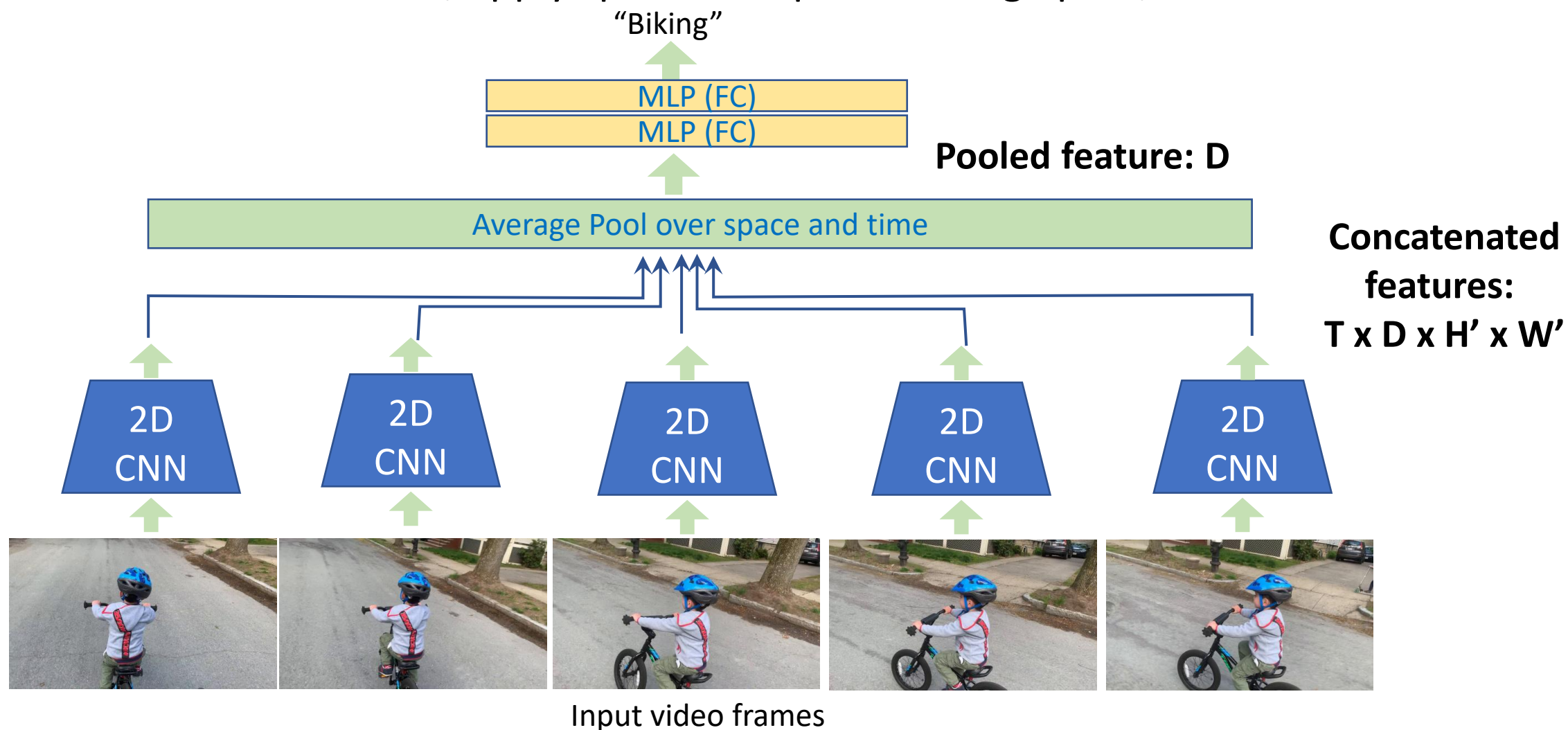
Models for Videos: Late Fusion

- Learn features for each frame using a 2D CNN, concatenate feature, and fuse



Models for Videos: Late Fusion w/ pooling

Learn features for each frame, apply spatial-temporal average pool, and then fuse

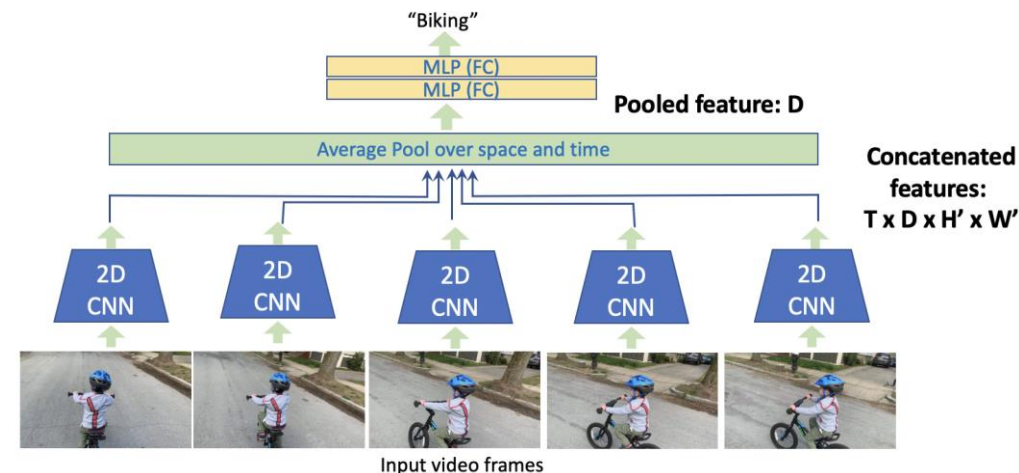


Models for Videos: Late Fusion w/ pooling

Learn features for each frame, apply spatial-temporal average pool, and then fuse

Pros: allow the network to learn global motion characteristics by comparing outputs of both towers

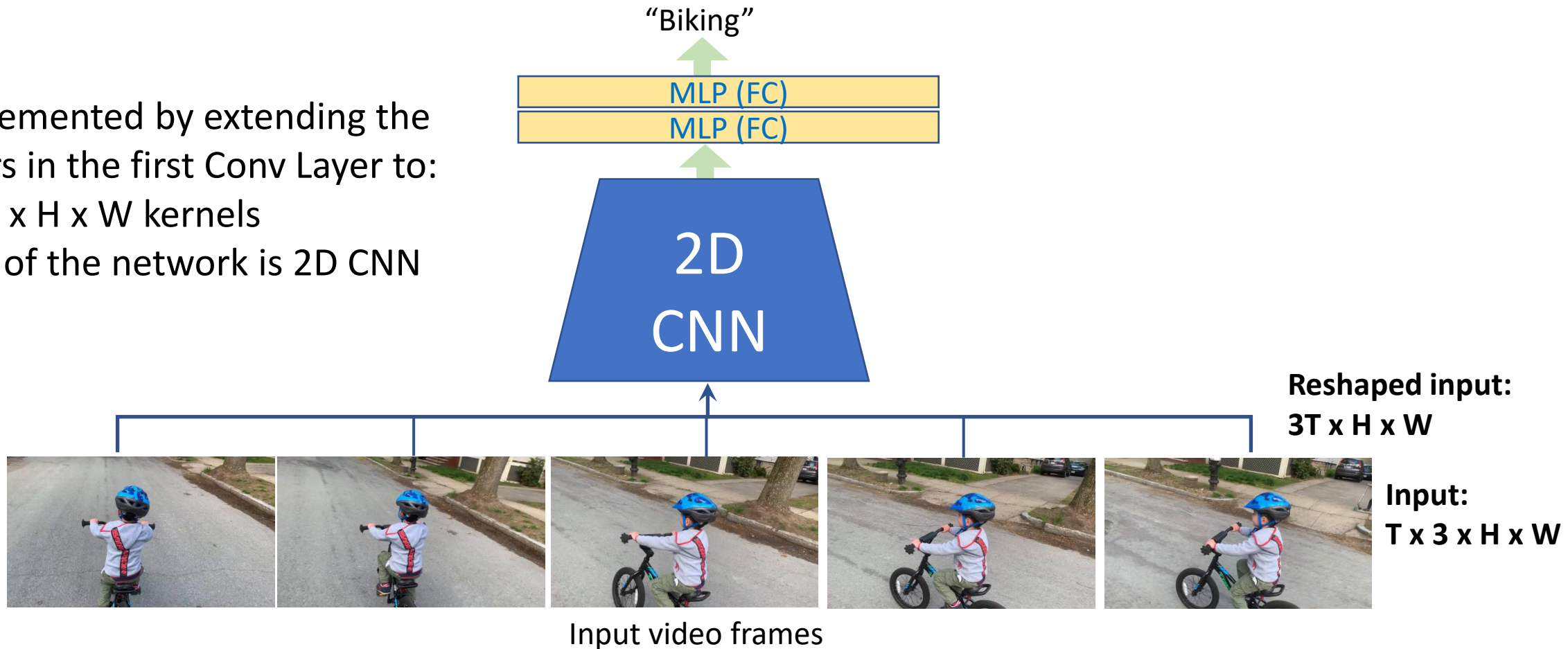
Cons: late fusion is late...
hard to represent low level motion between frames



Models for Videos: Early Fusion

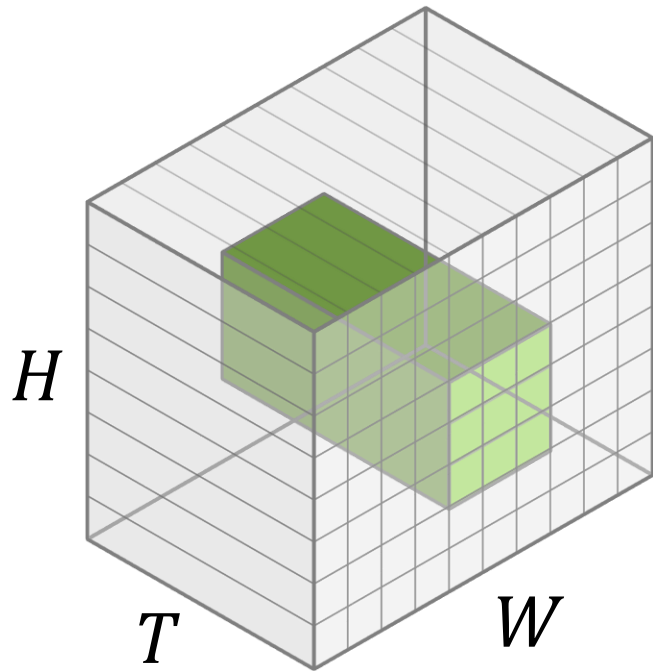
- Combines temporal information immediately on the pixel level
- Treat time as another “channel” dimension

Implemented by extending the filters in the first Conv Layer to:
 $T \times 3 \times H \times W$ kernels
Rest of the network is 2D CNN

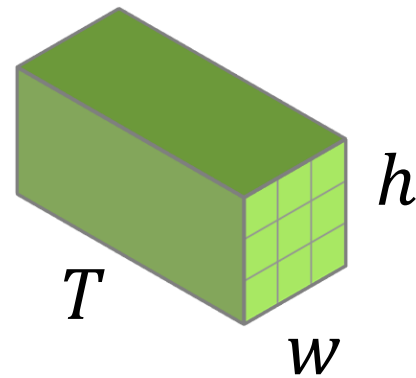


Models for Videos: Early Fusion

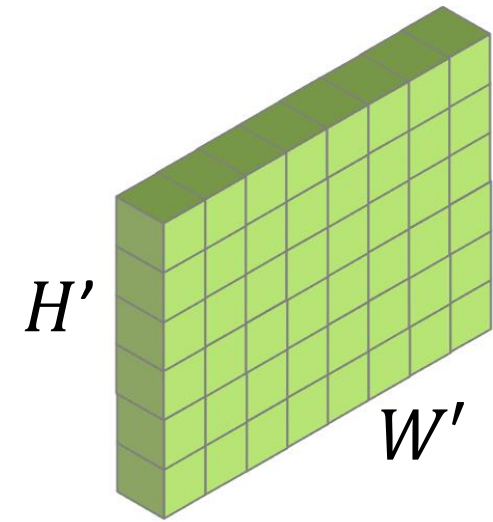
Extending the filters in the first Conv Layer to: $T \times 3 \times H \times W$ kernel



Input: $T \times 3 \times H \times W$



Weights: $C \times T \times 3 \times h \times w$



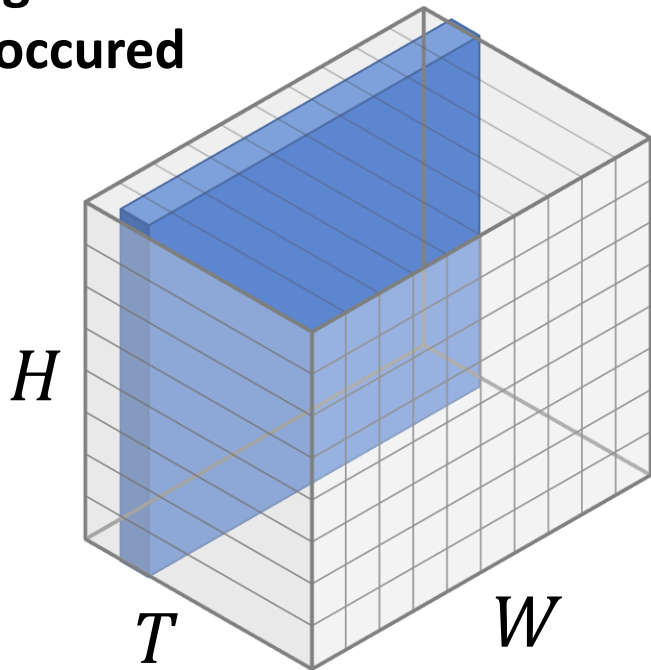
Output: $C \times H' \times W'$

Models for Videos: Early Fusion

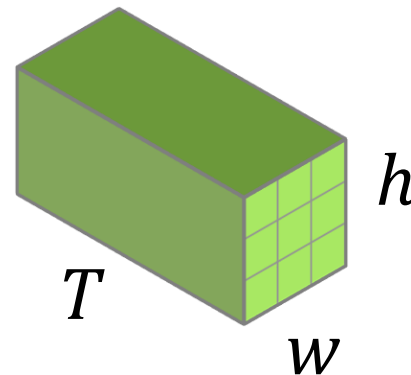
Extending the filters in the first Conv Layer to: $T \times 3 \times H \times W$ kernel

- **Not temporal shift invariance; specific filter is learned to each time step**

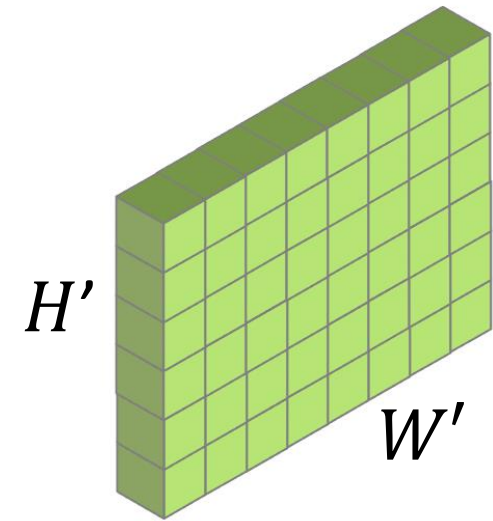
Large motion
occured



Input: $T \times 3 \times H \times W$



Weights: $C \times T \times 3 \times h \times w$



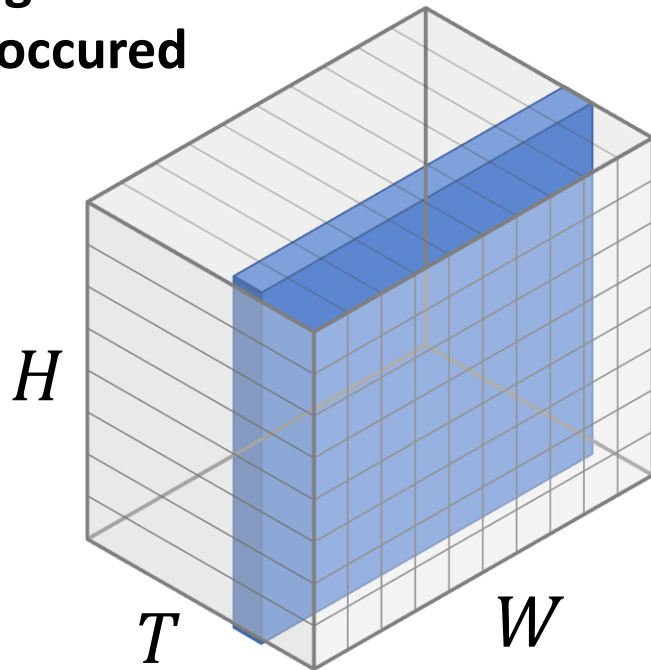
Output: $C \times H' \times W'$

Models for Videos: Early Fusion

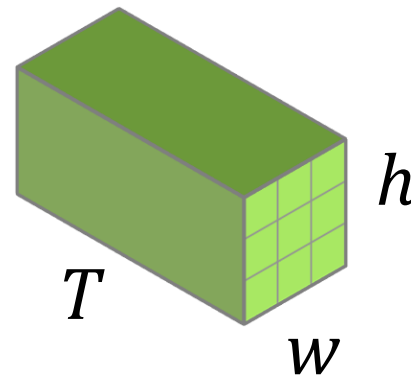
Extending the filters in the first Conv Layer to: $T \times 3 \times H \times W$ kernel

- Not temporal shift invariance; specific filter is learned to each time step

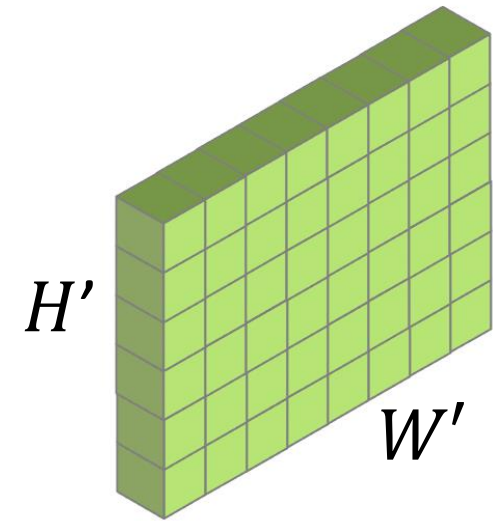
Large motion
occured



Input: $T \times 3 \times H \times W$



Weights: $C \times T \times 3 \times h \times w$



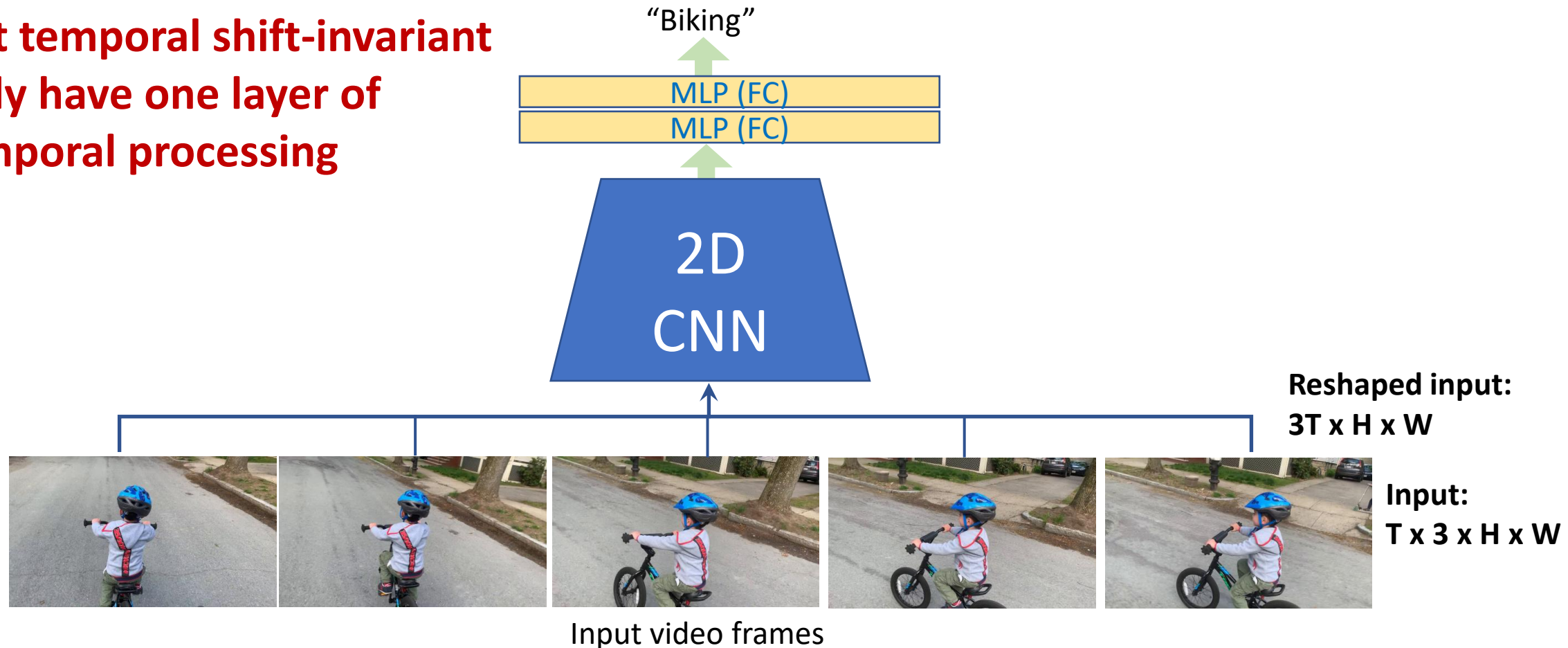
Output: $C \times H' \times W'$

Models for Videos: Early Fusion

Pros: Allow the network to learn local motion characteristics

Cons:

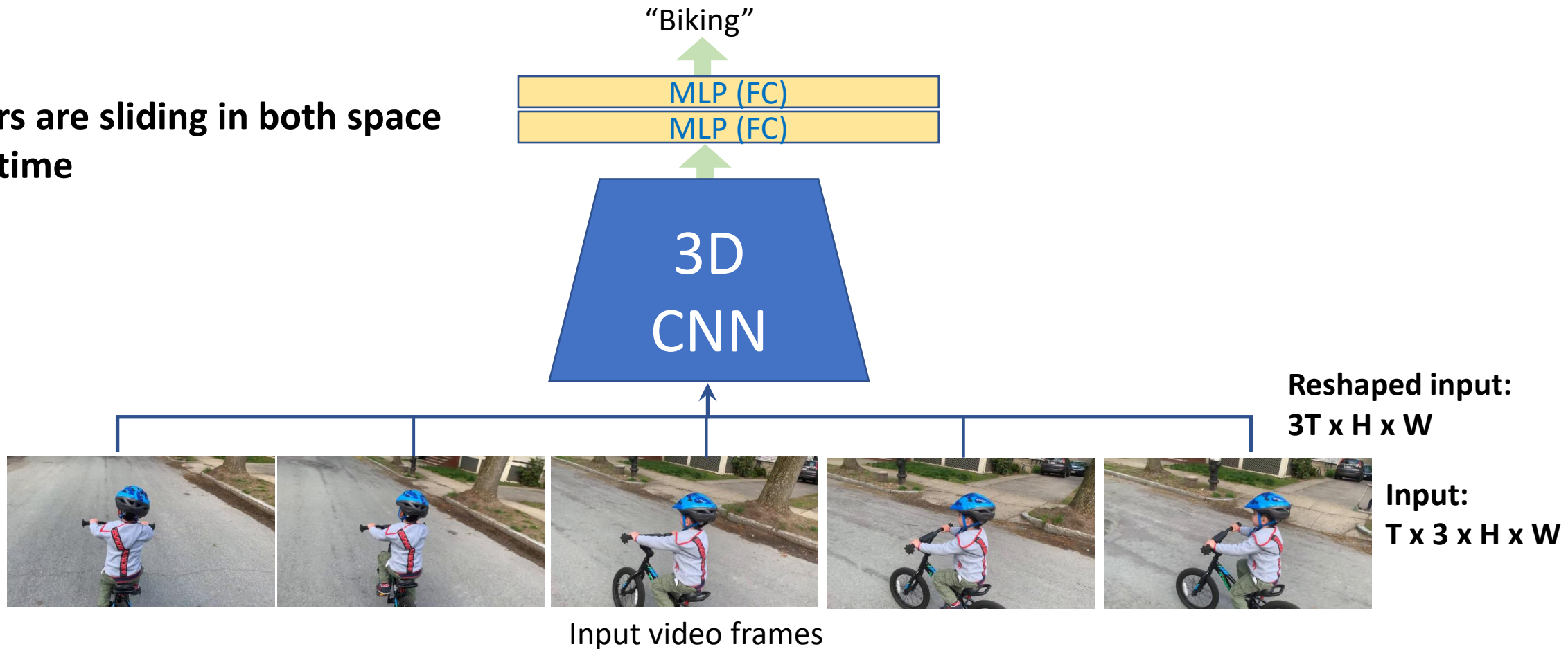
- Not temporal shift-invariant
- Only have one layer of temporal processing



Models for Videos: Slow Fusion a.k.a 3D Convs

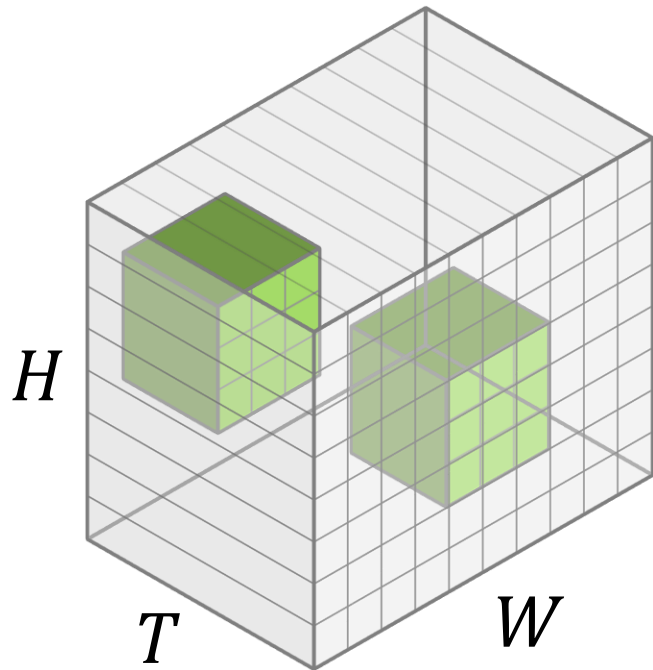
- Extend 2D Convs and pooling to 3D to slowly fuse temporal information throughout the model

Filters are sliding in both space and time



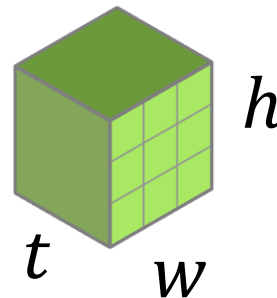
Models for Videos: Slow Fusion a.k.a 3D Convs

- Extend 2D Convs and pooling to 3D to slowly fuse temporal information throughout the model
- **Slide the kernels in both space and time**

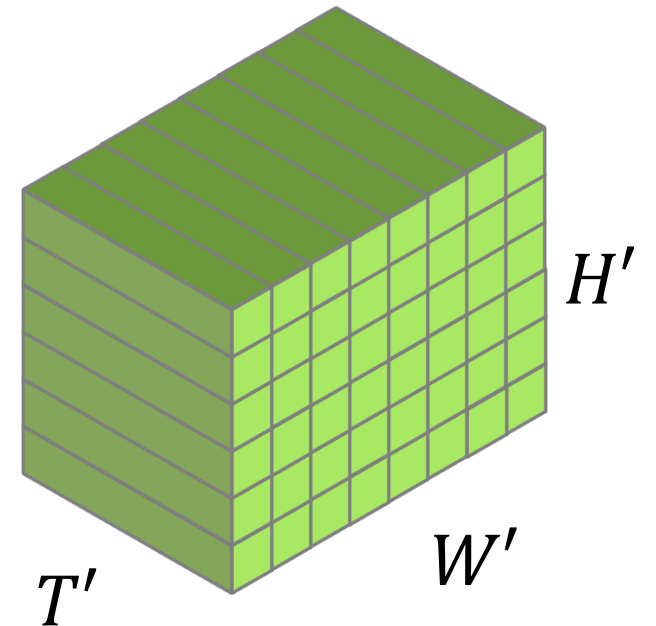


Input: $T \times 3 \times H \times W$

- **Temporal shift-invariant!**



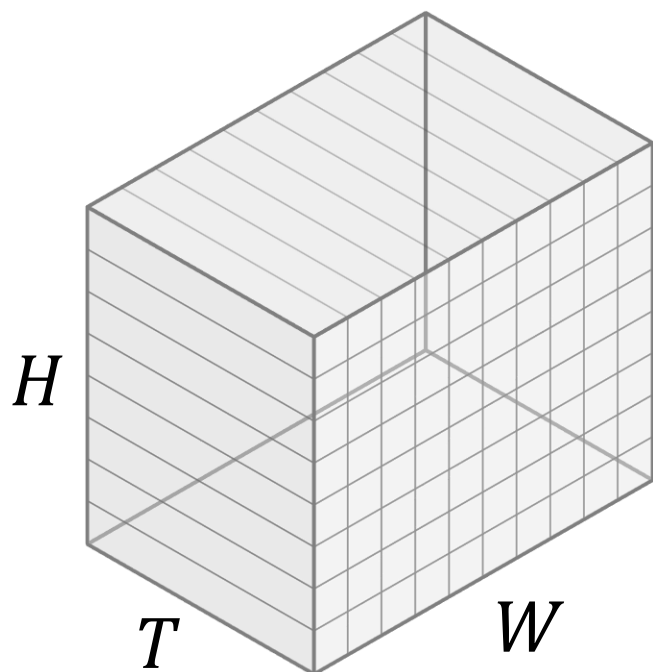
Weights: $C \times t \times 3 \times h \times w$



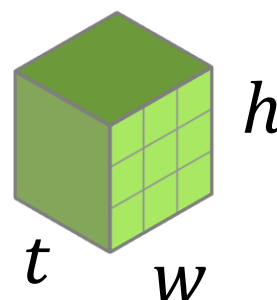
Output: $C \times T' \times H' \times W'$

Models for Videos: Slow Fusion a.k.a 3D Convs

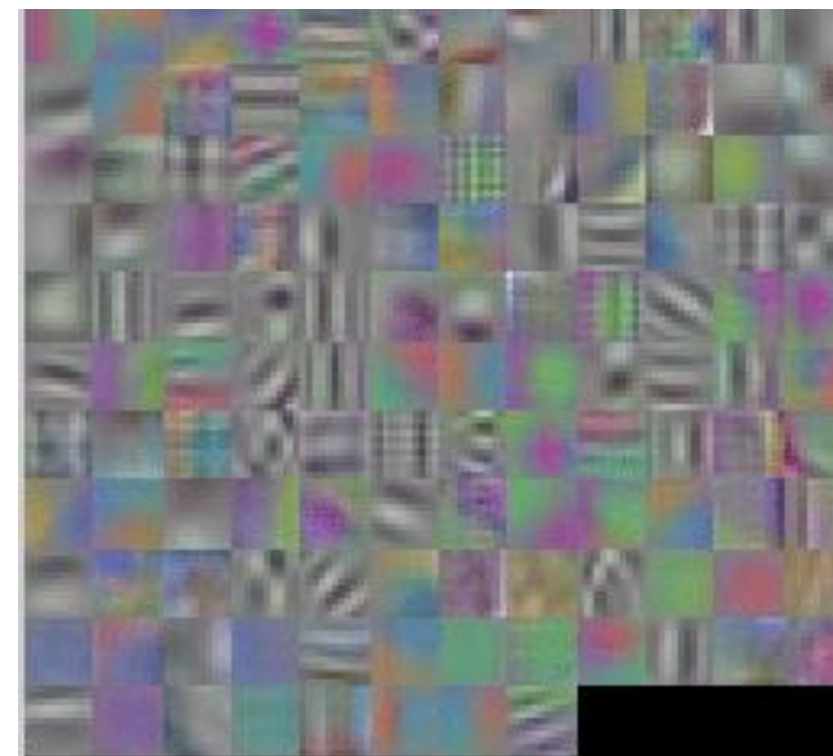
- Extend 2D Convs and pooling to 3D to slowly fuse temporal information throughout the model
- **Slide the kernels in both space and time**



Input: $T \times 3 \times H \times W$



Weights: $C \times t \times 3 \times h \times w$



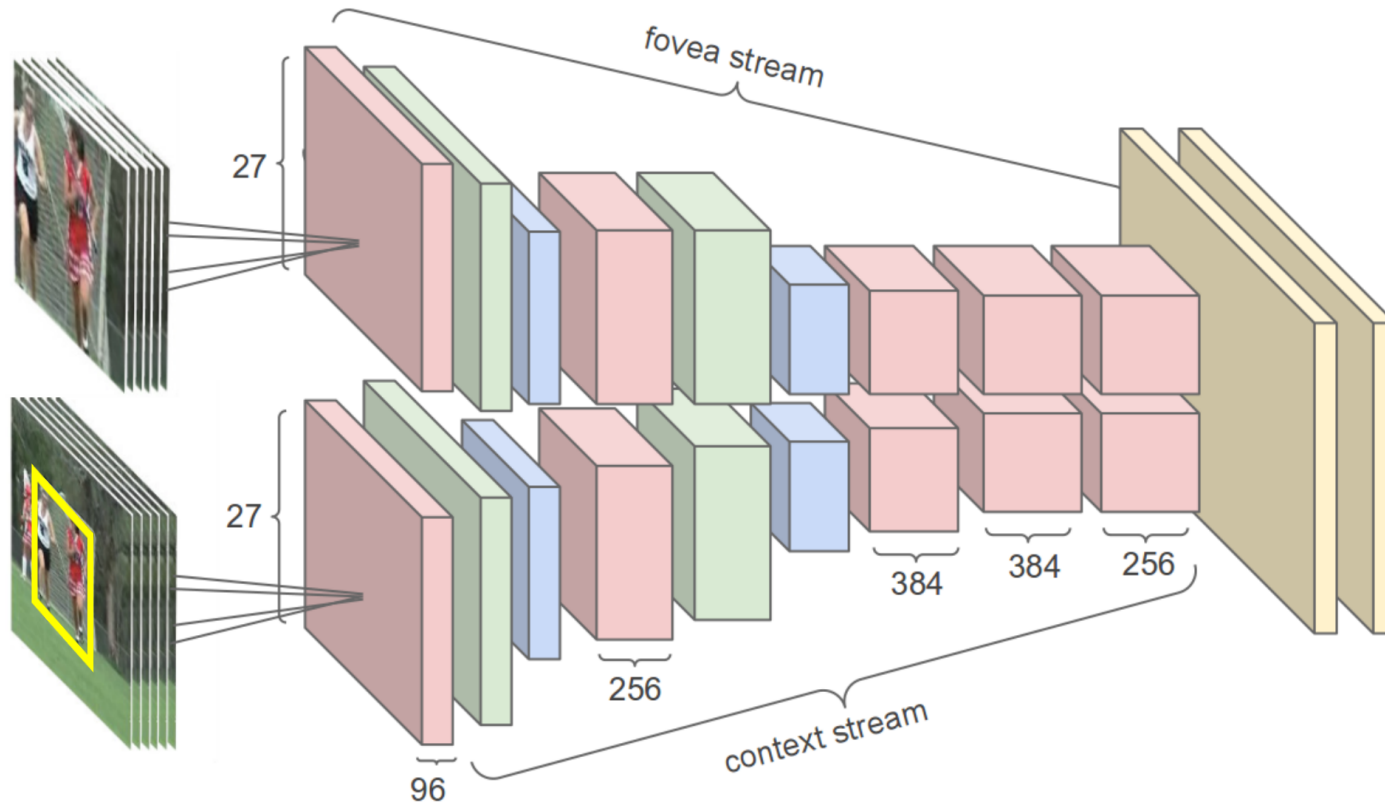
First layer filters
 $3(\text{rgb}) \times 4(t) \times 5(h) \times 5(w)$

Models for Videos: Multi-scale

How can we reduce computational cost while maintaining accuracy?

Reduce video resolution → lower performance

Reduce network's capacity → lower performance



- Context stream (**low res**): process low res video frames ($H/2, W/2$)
- Fovea stream (**high res**): process a $(H/2, W/2)$ crop from the original resolution



Reduce the input dimensionality by half

Action classification -- Sports-1M



track cycling
cycling
track cycling
road bicycle racing
marathon
ultramarathon



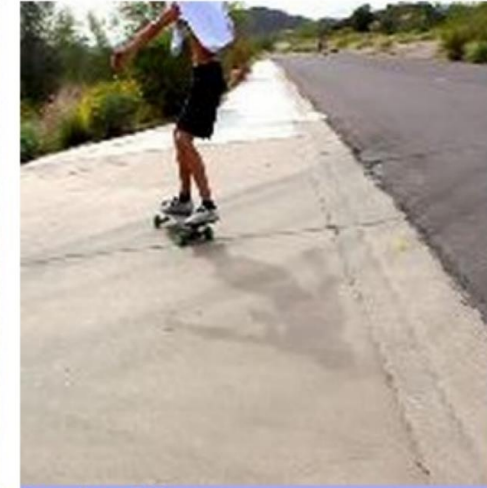
ultramarathon
ultramarathon
half marathon
running
marathon
inline speed skating



heptathlon
heptathlon
decathlon
hurdles
pentathlon
sprint (running)



bikejoring
mushing
bikejoring
harness racing
skijoring
carting

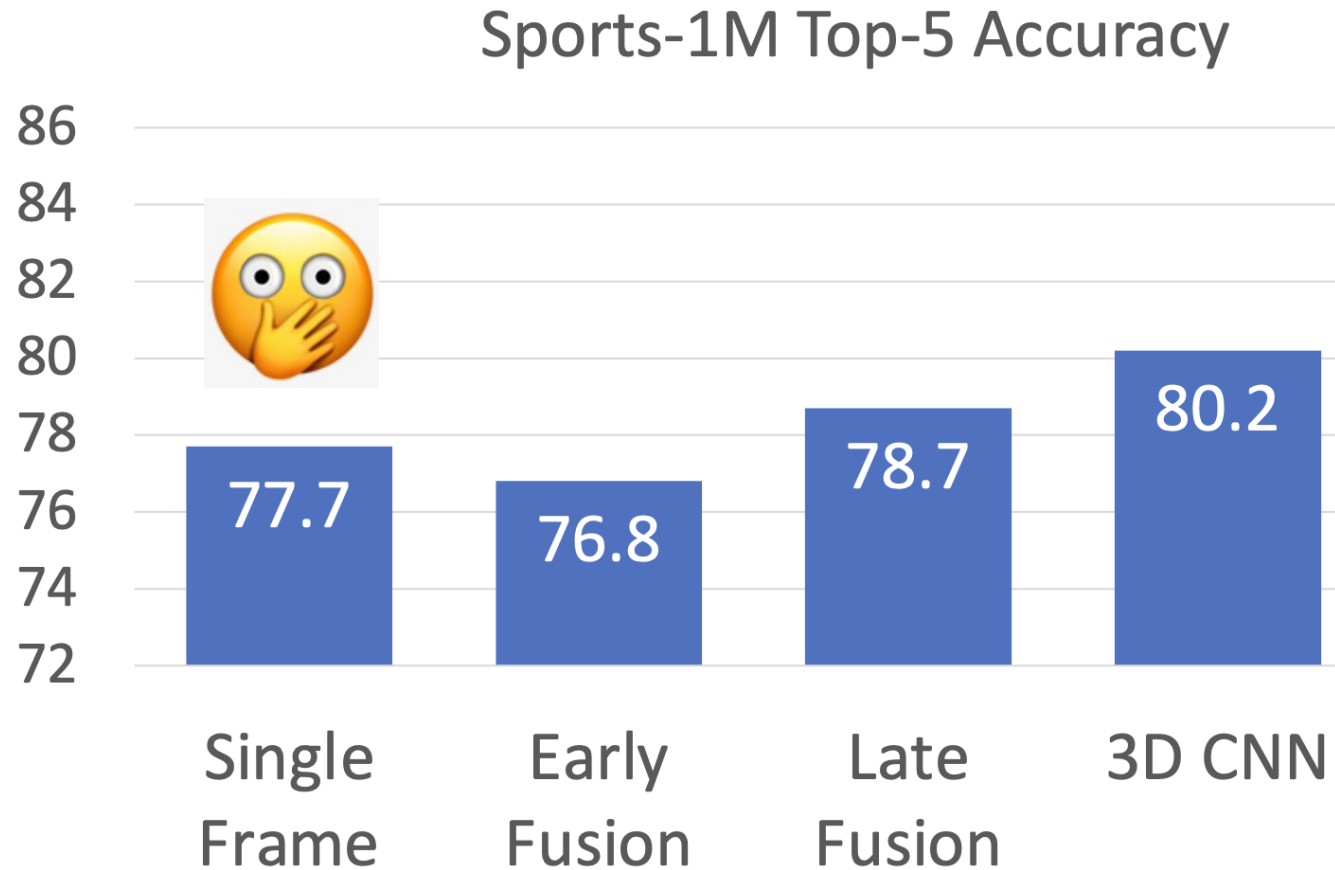


longboarding
longboarding
aggressive inline skating
freestyle scootering
freeboard (skateboard)
sandboarding

- 1 million YouTube videos
- Fine grained labels for 487 different types of sports

- Ground truth
- Correct prediction
- Incorrect prediction

Action classification -- Sports-1M

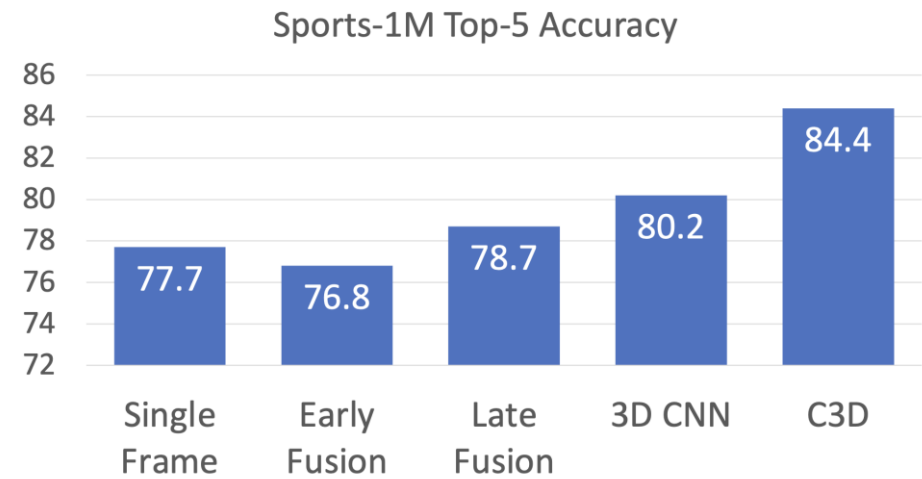
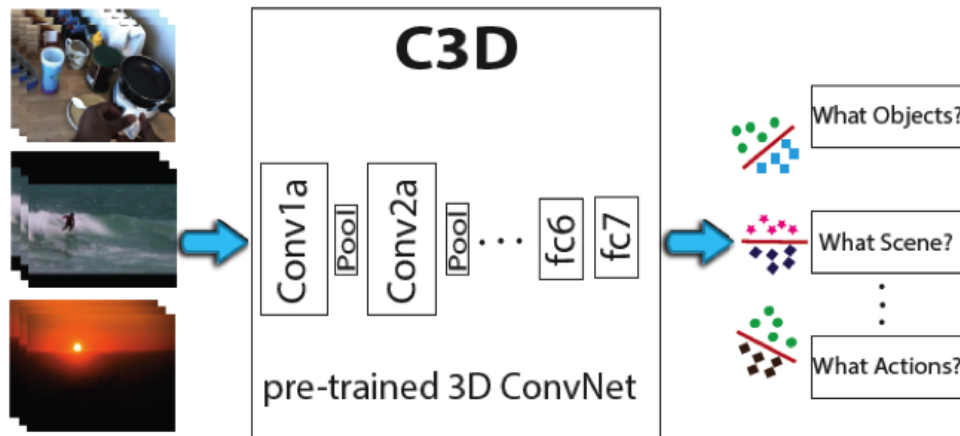


Single frame: a shockingly powerful baseline

This is from 2014...

Models for Videos: C3D (Convolutional 3D)

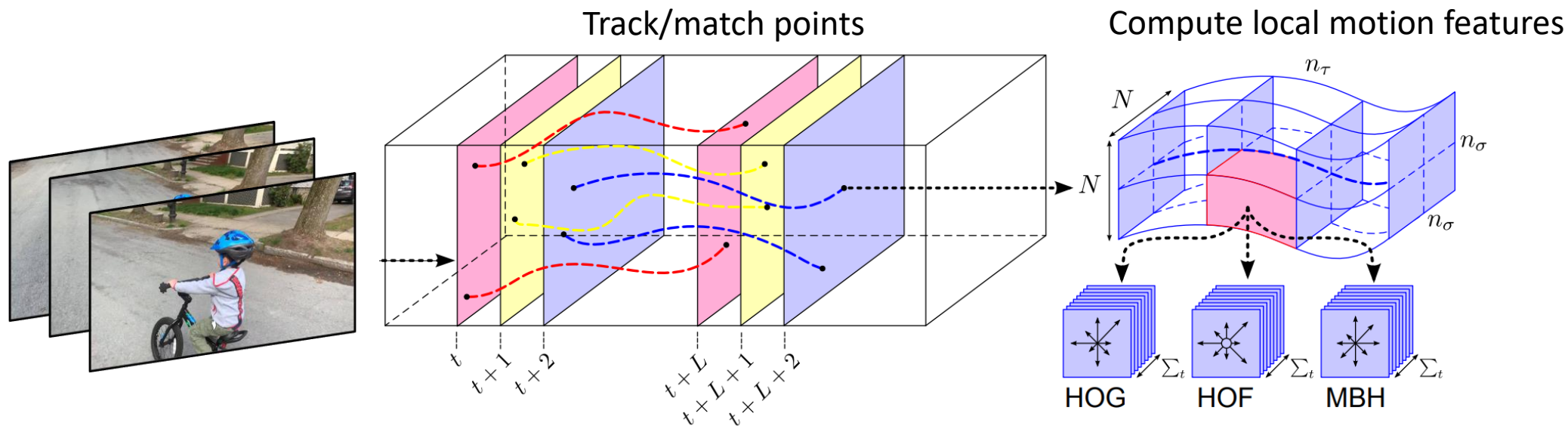
- 3D CNN that uses all 3x3x3 Convs and 2x2x2 poolings
- The “VGG” of 3D CNNs
- Transfer learning: extract learned video features, train a simple linear classifier for various tasks



- Problem: 3D convs are VERY expensive!
C3D on small inputs takes 3x VGG and 56x AlexNet FLOPs

Non-deep learning video classification

Motion is the most informative cue for action recognition → design hand crafted motion features:

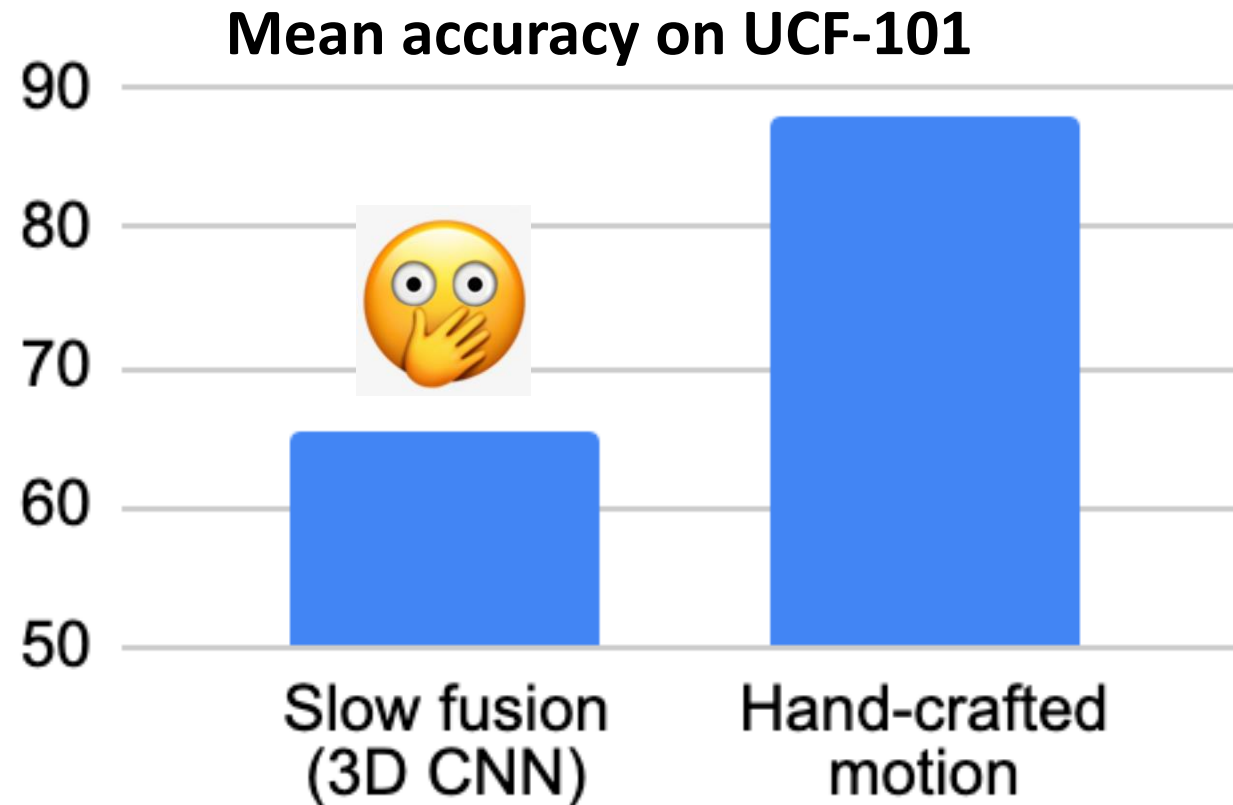


Aggregate local motion features to compute a global representation of the video → linear SVM for action recognition

MODEL MOTION EXPLICITLY

Non-deep learning video classification

Motion is the most informative cue for action recognition → hand crafted motion features:



Explicitly modeling motion in deep-based models

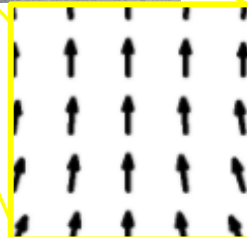
Optical flow: For each pixel in frame t , determines its corresponding pixel in frame $t+1$



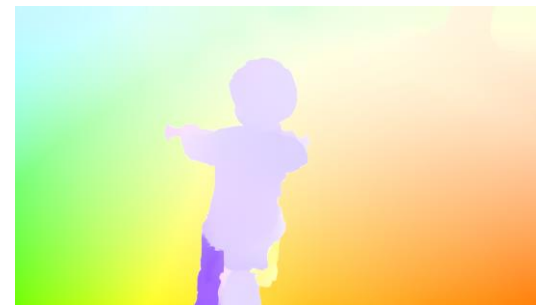
Frame $t+1$



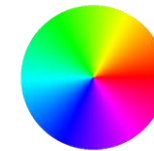
Frame t



Optical flow provides **local motion cues**



Optical flow between two frames



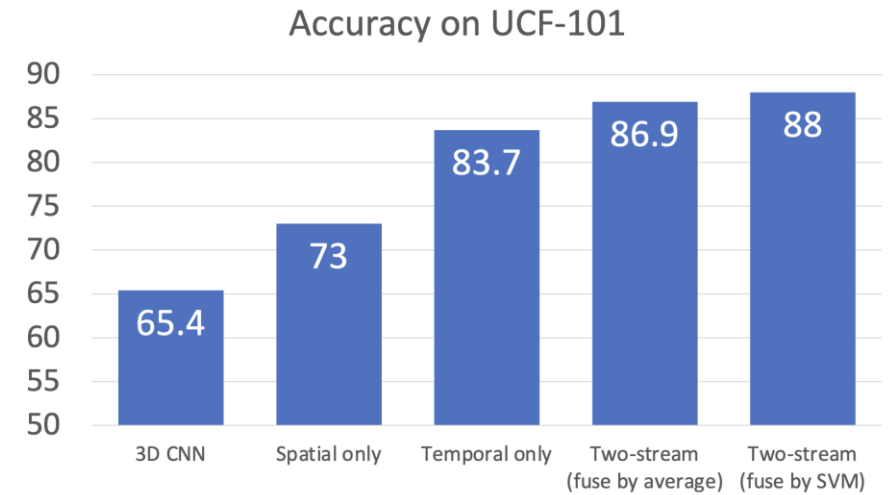
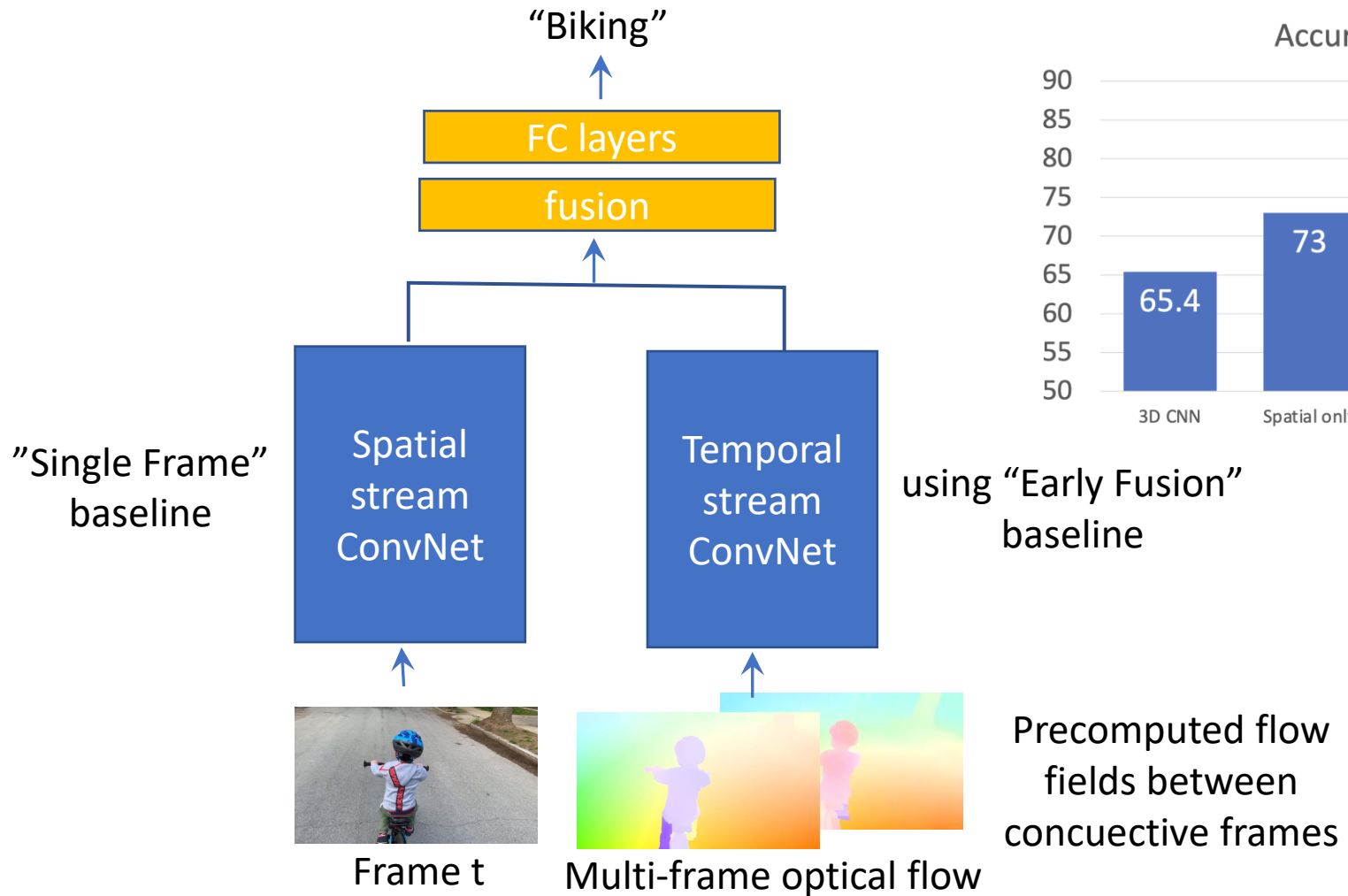
Color wheel

Saturation = mag.

Color = angle

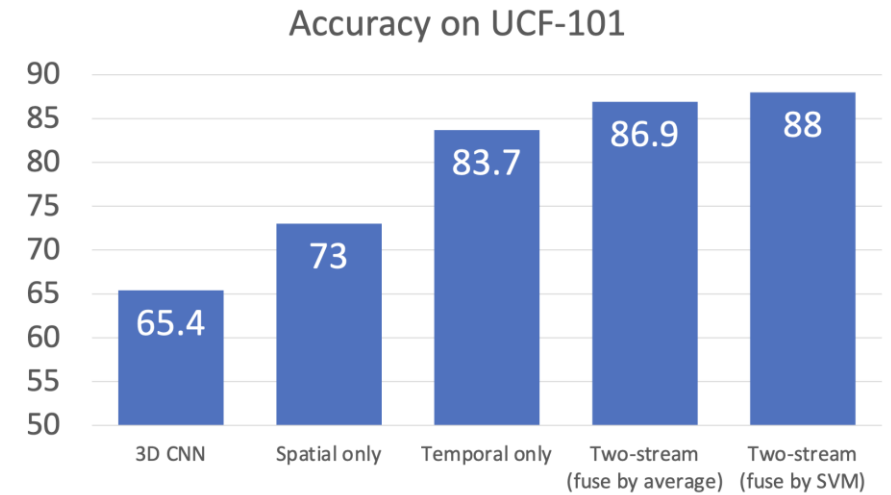
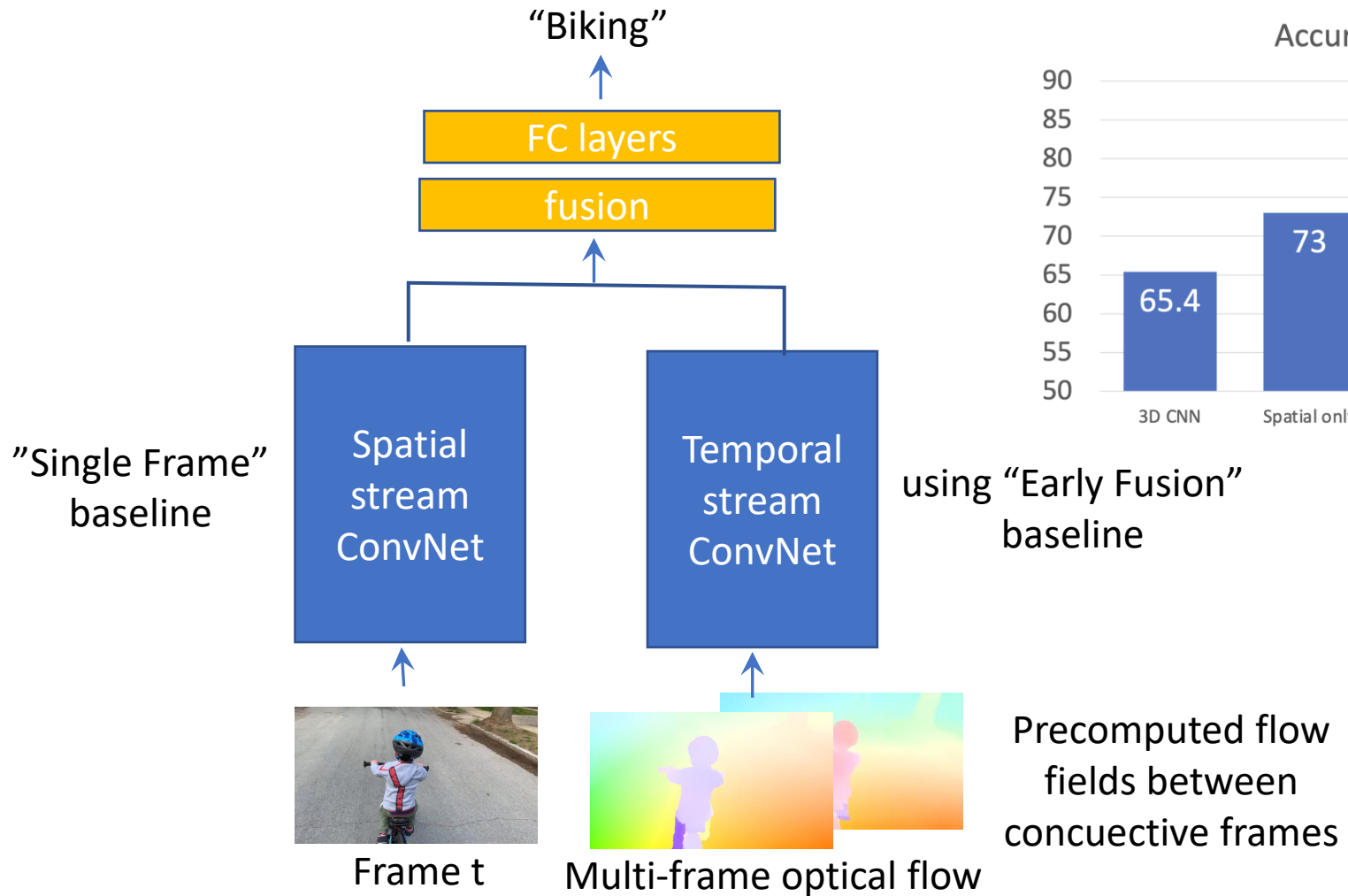
Two Stream Networks: modeling motion explicitly

Idea: separate motion (multi-frame) from static appearance (single frame)



Two Stream Networks: modeling motion explicitly

Idea: separate motion (multi-frame) from static appearance (single frame)



Additional models

Inflating 2D networks to 3D (I3D)

Take an existing 2D CNN model → convert it to a 3D CNN model

Transfer the weights from 2D and 3D

Carreira and Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset", CVPR 2017

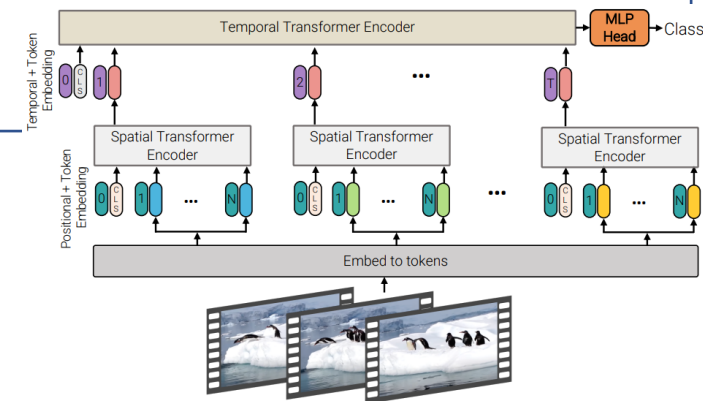
Long range temporal processing

Use LSTMs and RNNs to model long range temporal information

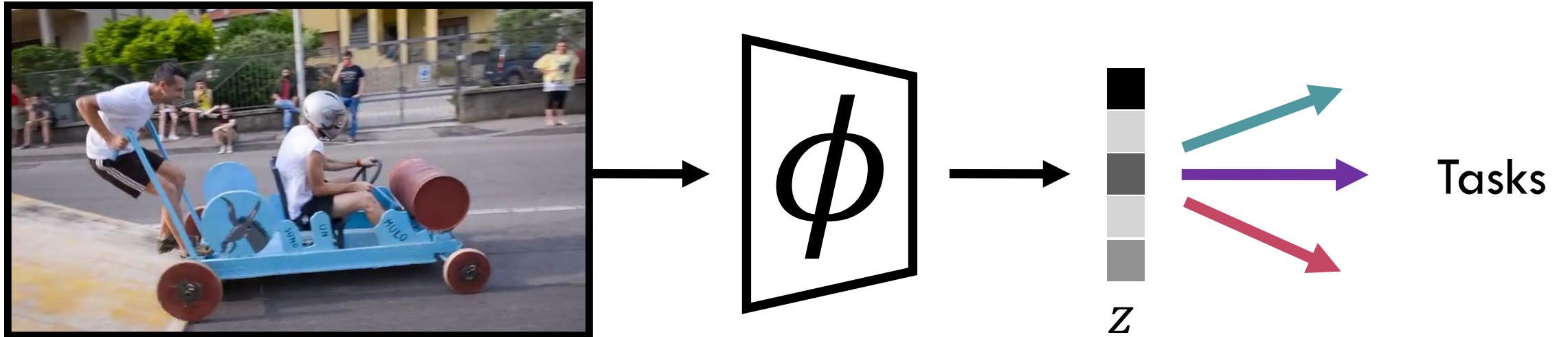
Baccouche et al, "Sequential Deep Learning for Human Action Recognition", 2011 Donahue et al, "Long-term recurrent convolutional networks for visual recognition and description", CVPR 2015

Long range temporal processing

Self attention, non-local networks, Transformers



Self-Supervision in Videos

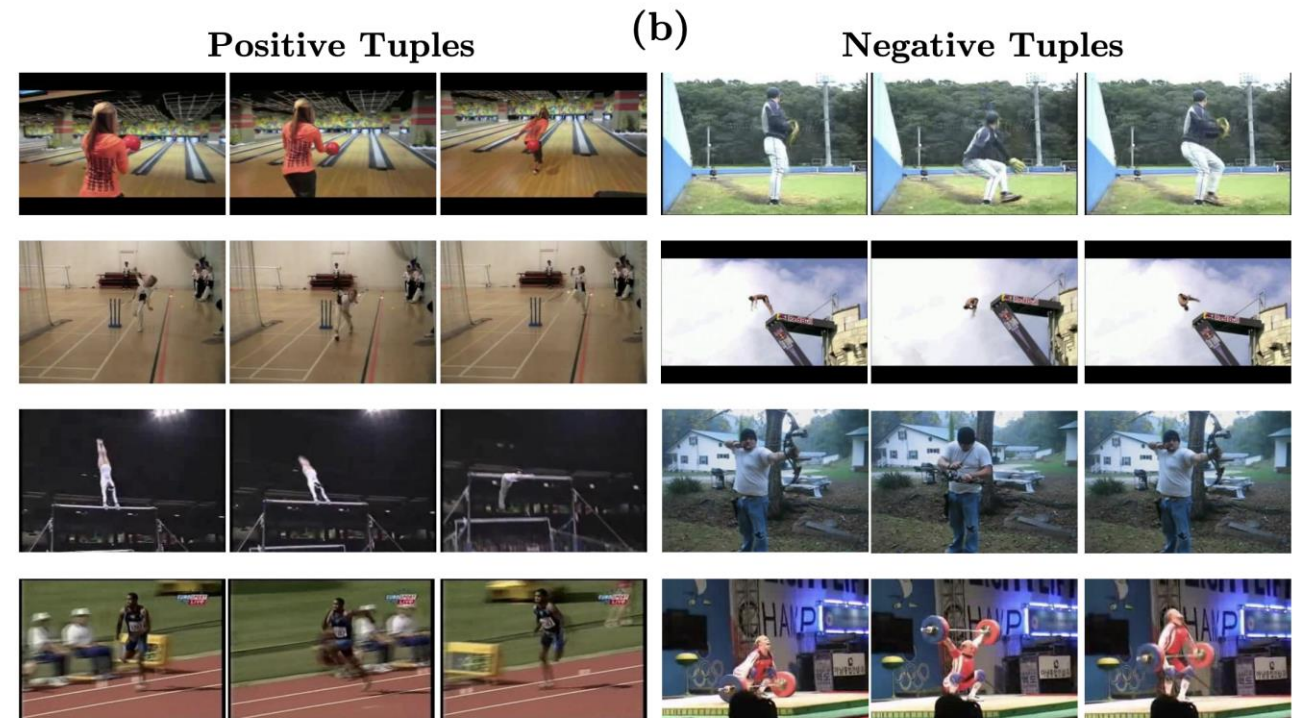
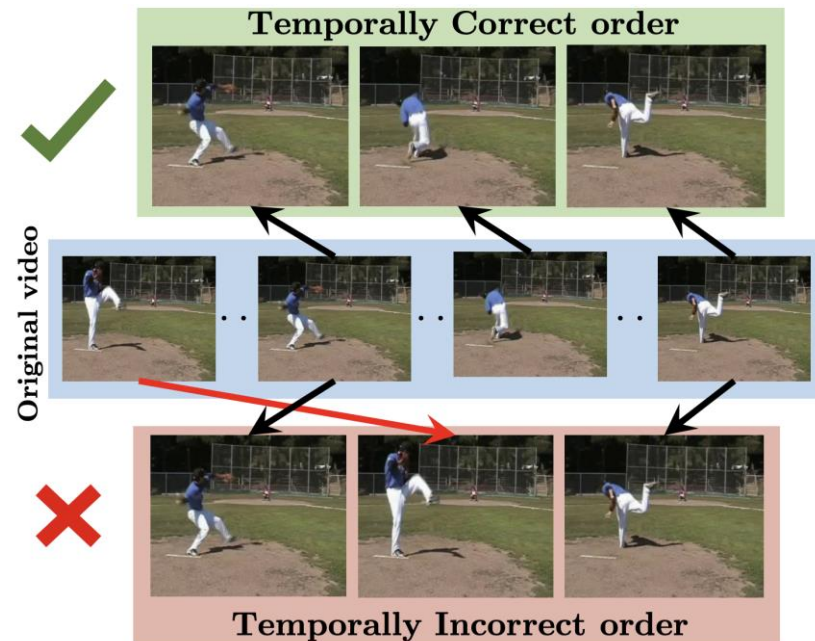


- Temporal order
- Cycle consistency
- Video Speedup
- Video colorization

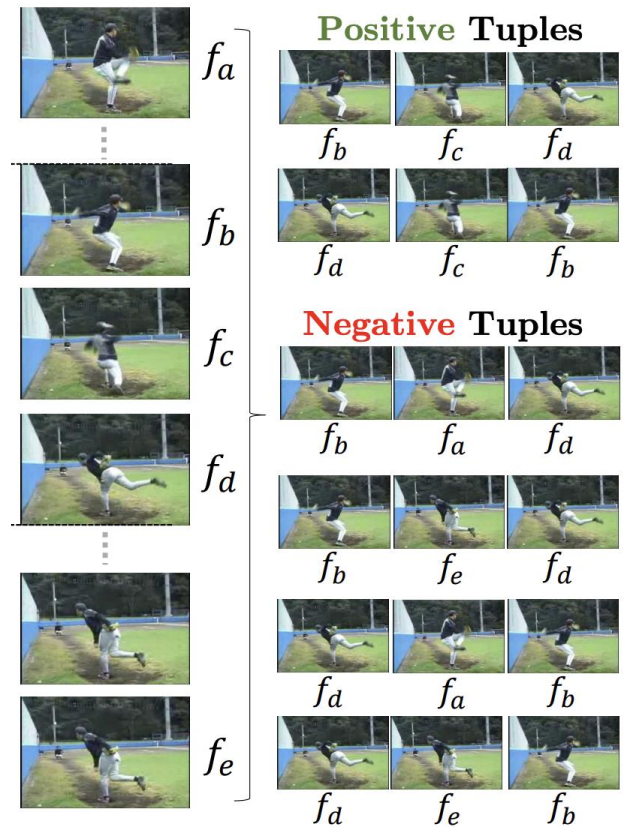
Self-Supervision in Videos: frame ordering

Training data: shuffled video frames, original video frames

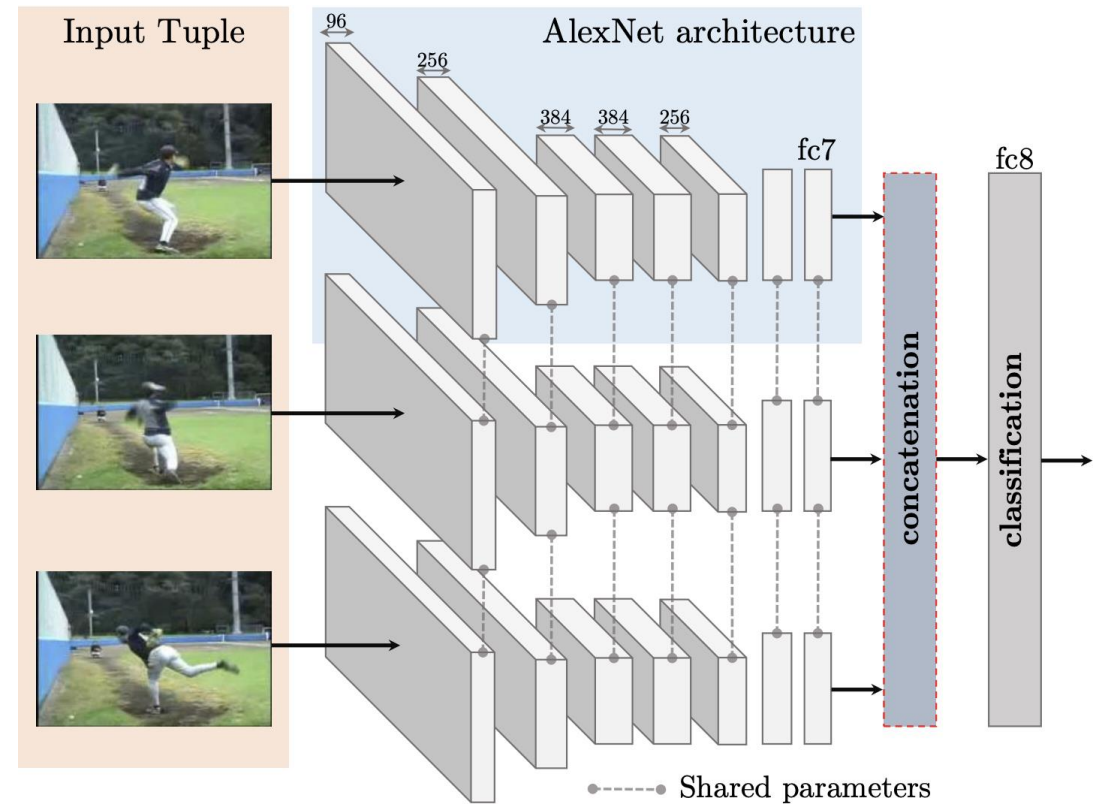
Pretext task: predict if the frames are in the correct **temporal order** (binary classification task)



Self-Supervision in Videos: frame ordering



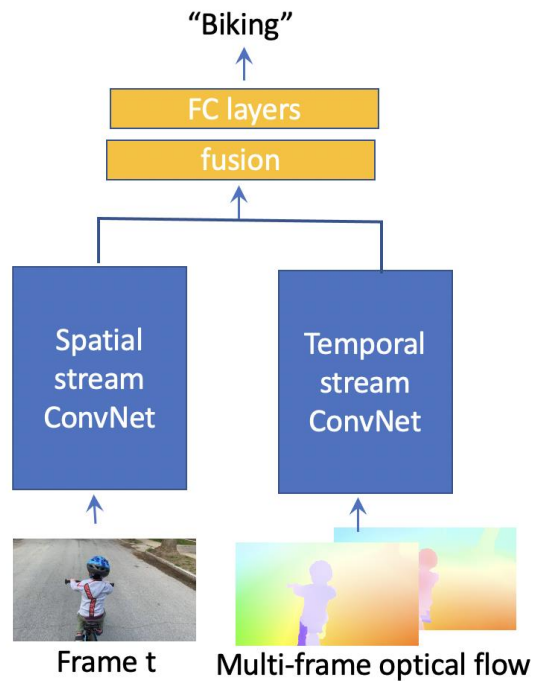
Generating positive and negative examples



Triplet Siamese network for sequence verification

Self-Supervision in Videos: frame ordering

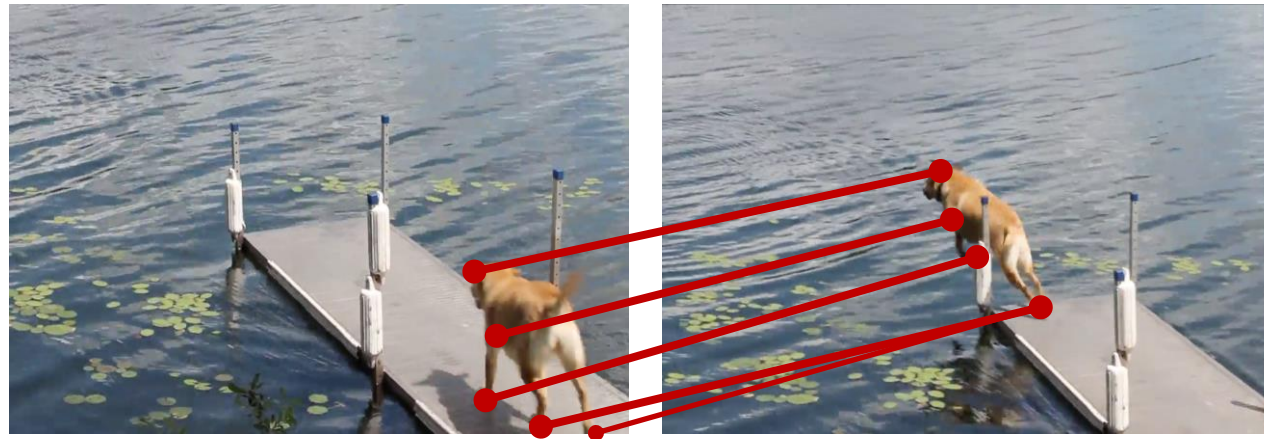
Transfer learning: fine-tune spatial stream for video classification



Dataset	Initialization	Mean Accuracy
UCF101	Random	38.6
	(Ours) Tuple verification	50.2
HMDB51	Random	13.3
	UCF Supervised	15.2
	(Ours) Tuple verification	18.1

Self-Supervision in Videos: Learning correspondence

Ultimate goal: Correspondence

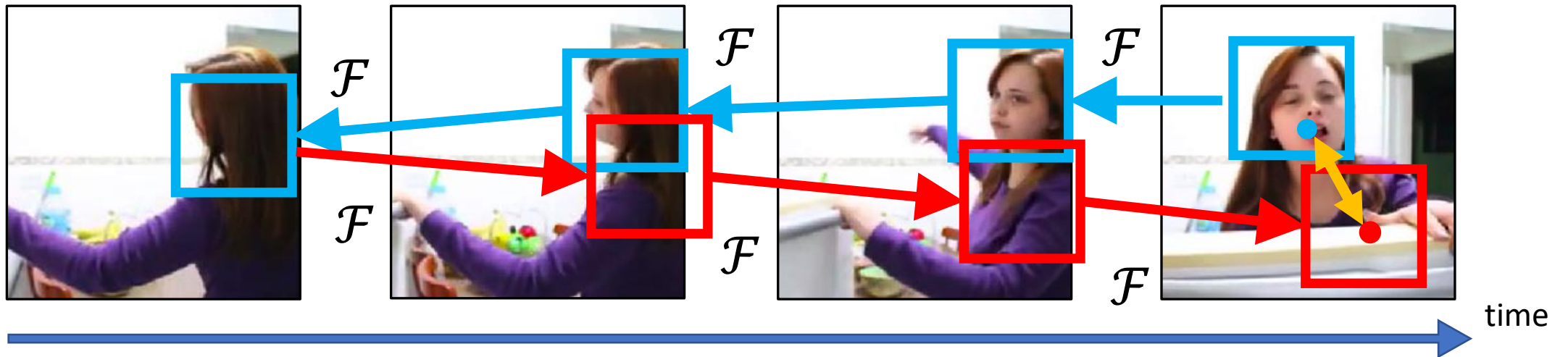


Self-Supervision in Videos: Learning correspondence

Ultimate goal: Correspondence, without using off-the-shelf tracking methods

How to obtain supervision?

Supervision: Cycle-Consistency in Time

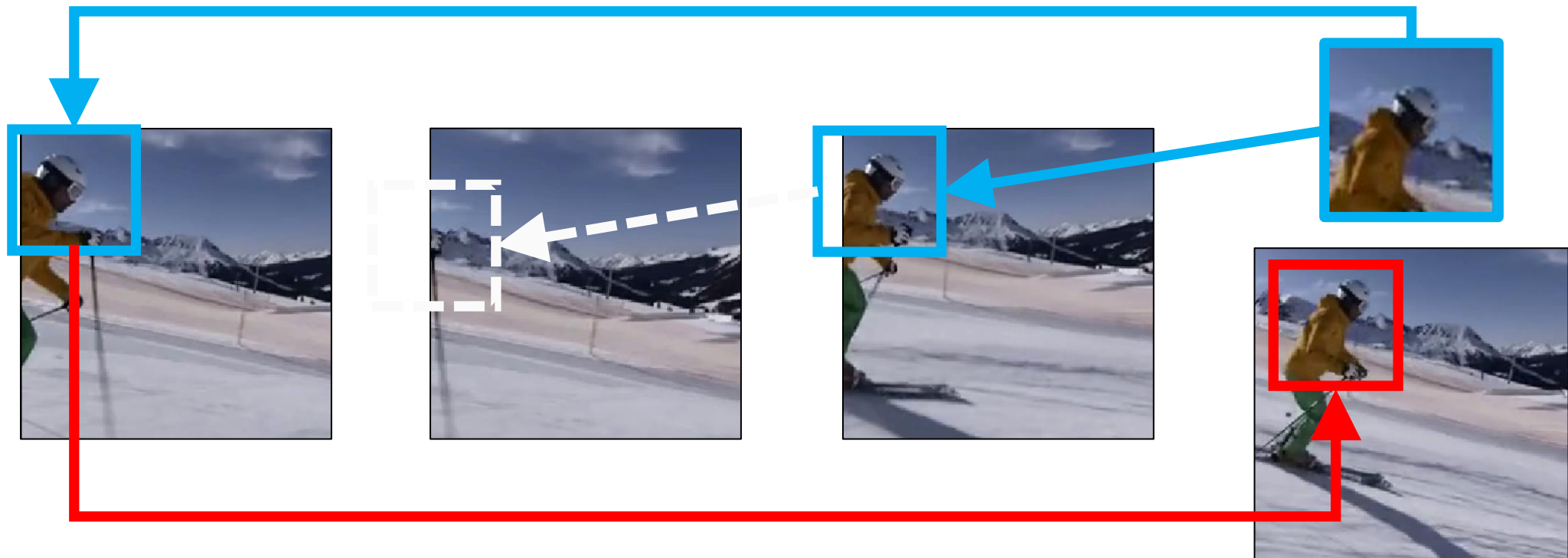


Track backwards in time
Track forwards, back to the future

Self-Supervision in Videos: Learning correspondence

Supervision: Cycle-Consistency in Time

Challenge: Occlusions

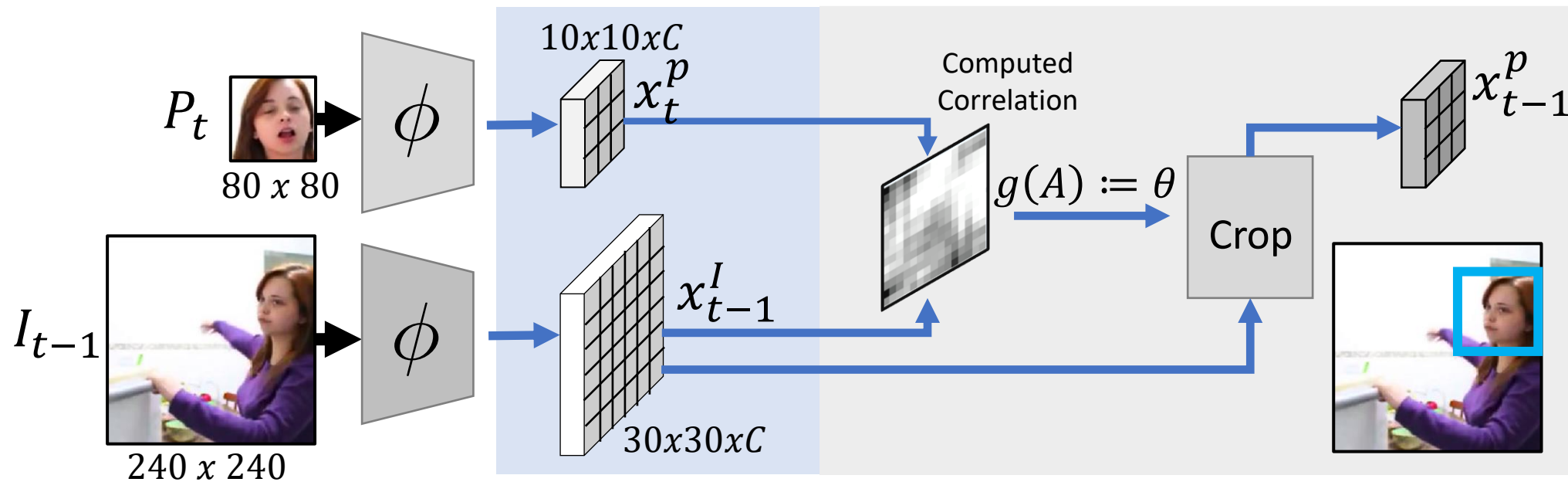


Skip-cycles: skipping occlusions

Self-Supervision in Videos: Learning correspondence

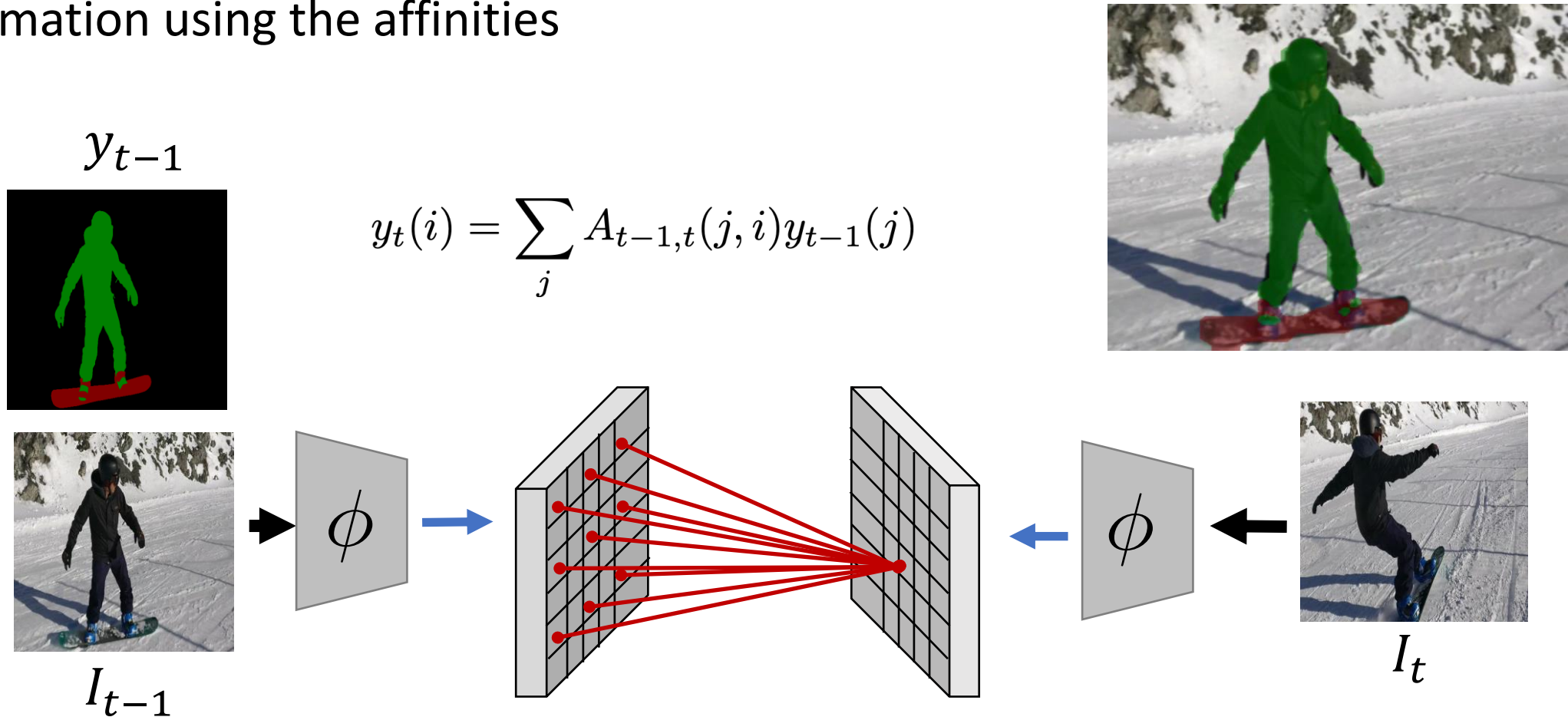
Differentiable tracker: densely match features in learned feature space

$$A(j, i) = \frac{\exp(x^I(j)^\top x^P(i))}{\sum_j \exp(x^I(j)^\top x^P(i))} \quad A \in R^{900 \times 100}$$



Self-Supervision in Videos: Learning correspondence

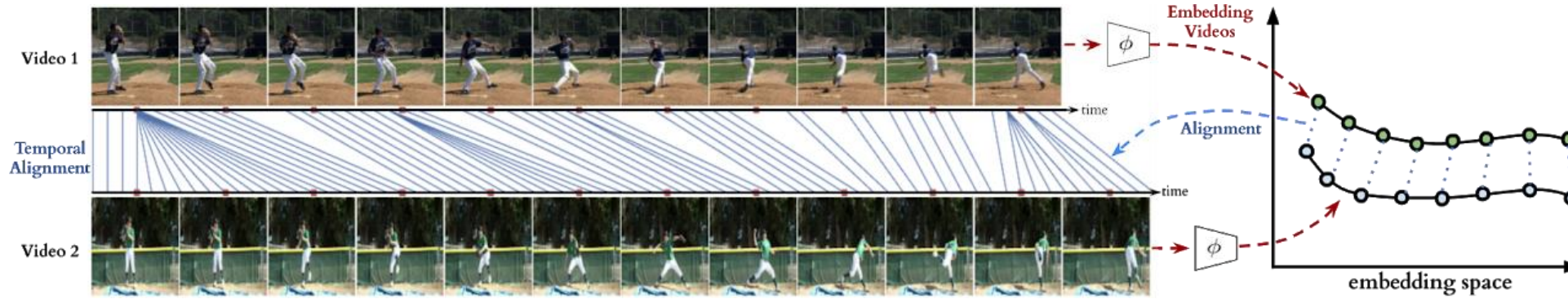
Test time: compute features to each frame, compute features affinity, propagate information using the affinities



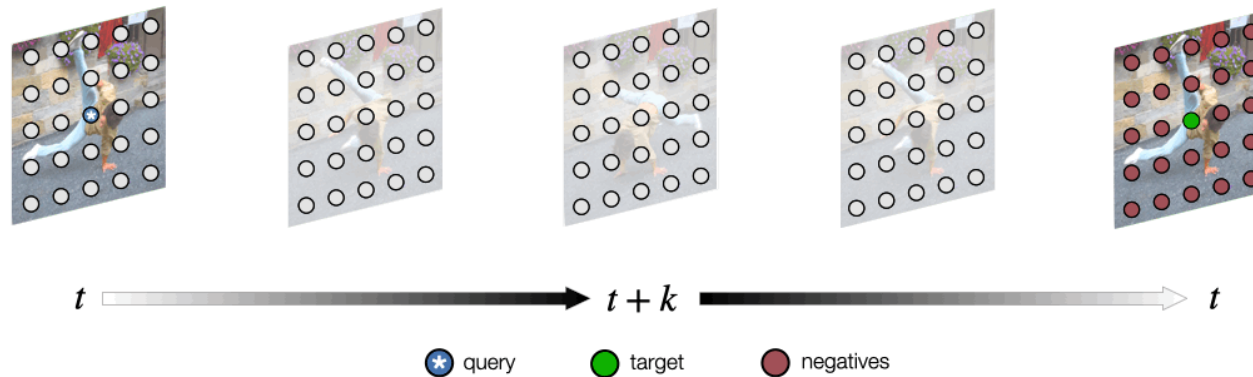
Self-Supervision in Videos: Learning correspondence



Self-Supervision in Videos: Temporal cycle consistency



Dwived et. al. Temporal Cycle-Consistency Learning, CVPR'19

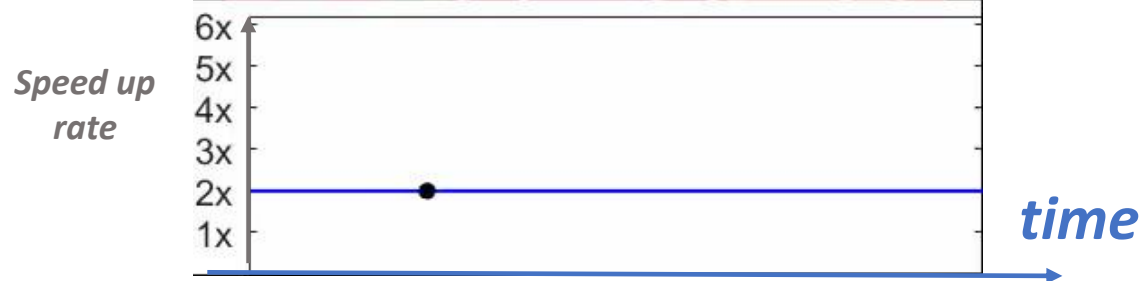


Jabri et. al, Space time correspondence as Contrastive Random Walk, NeurIPS 2020

Self-Supervision in Videos: Learning the Speediness in Videos

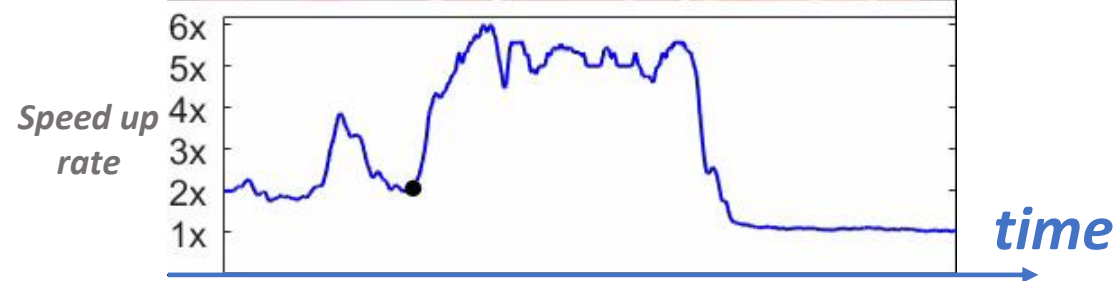
Ultimate goal: Watch video content faster by adaptively speeding up the video

Uniform Speed Up (2x)



Jittery, unnatural motions

Adaptive Speed Up (2x)



Same duration, more natural

“Speediness” in Videos

Slower



Normal speed



Faster



Self-Supervision in Videos: Learning the Speediness in Videos

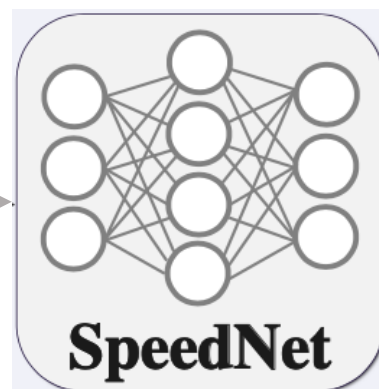
Pretext task: Predict if a given video segment is sped up or not

Training data: sped up video segments + original video segments

Self supervised
training on Kinetics



Input segment
(30 frames)



Normal speed
or
Sped Up

“Learning and Using the Arrow of Time”, Wei et al, CVPR 2018



Self-Supervision in Videos: Learning the Speediness in Videos

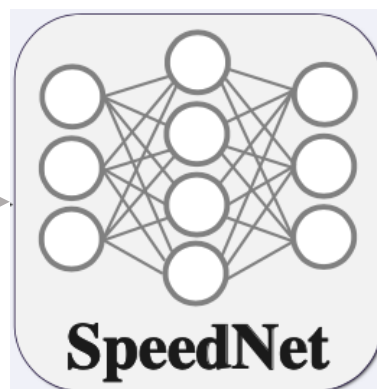
Pretext task: Predict if a given video segment is sped up or not

Training data: sped up video segments + original video segments

Self supervised
training on Kinetics



Input segment
(30 frames)



Normal speed
or
Sped Up

Networks are lazy!



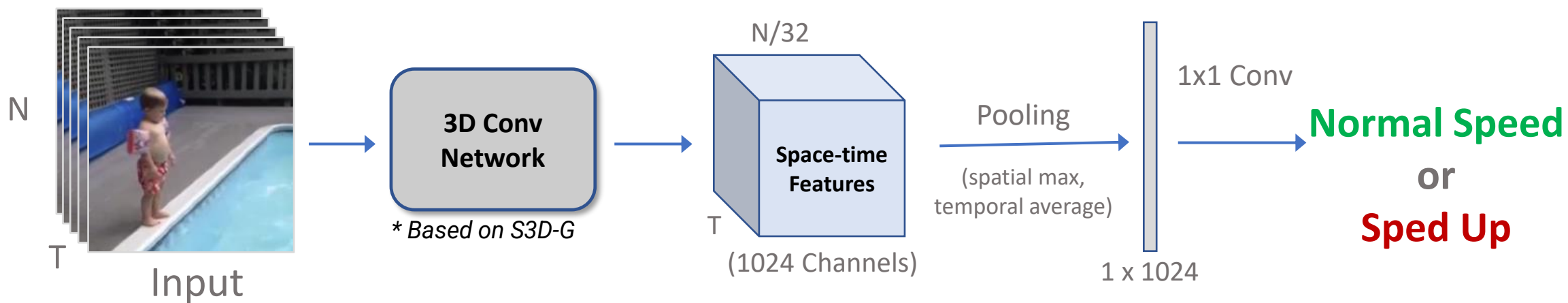
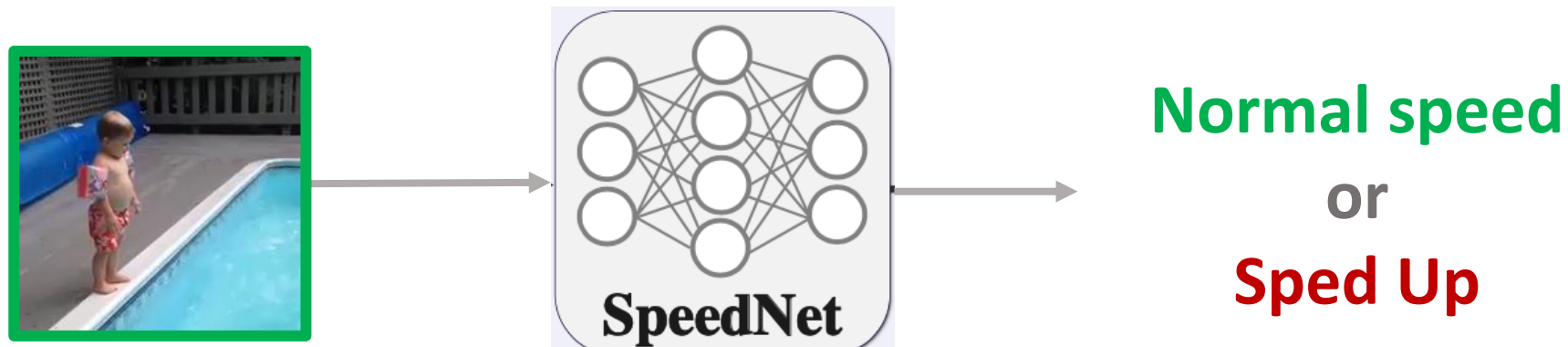
Learning properties of natural motion, avoid “easy cheats” →
very challenging!

Self-Supervision in Videos: Learning the Speediness in Videos

Pretext task: Predict if a given video segment is sped up or not

Training data: sped up video segments + original video segments

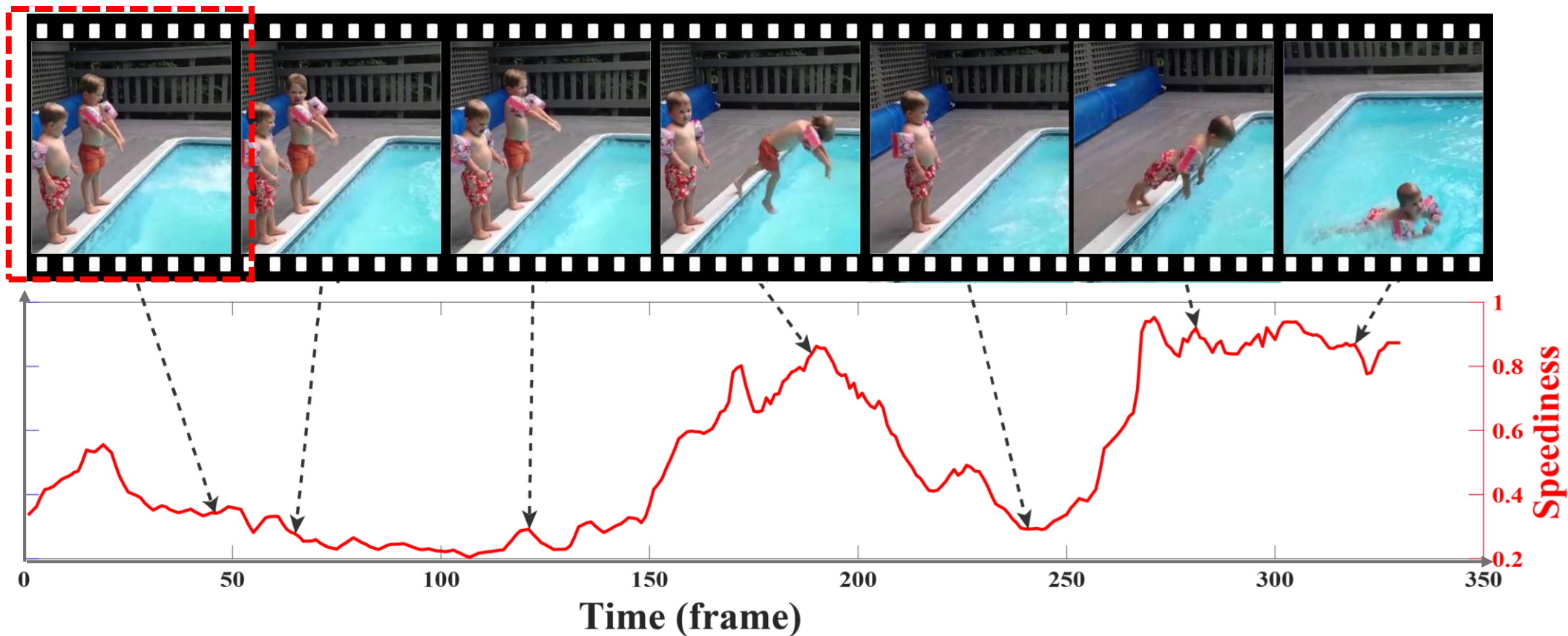
Self supervised
training on Kinetics



* "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification",
Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy, ECCV'18.

Self-Supervision in Videos: Learning the Speediness in Videos

Inference: sliding window \rightarrow prediction for every frame

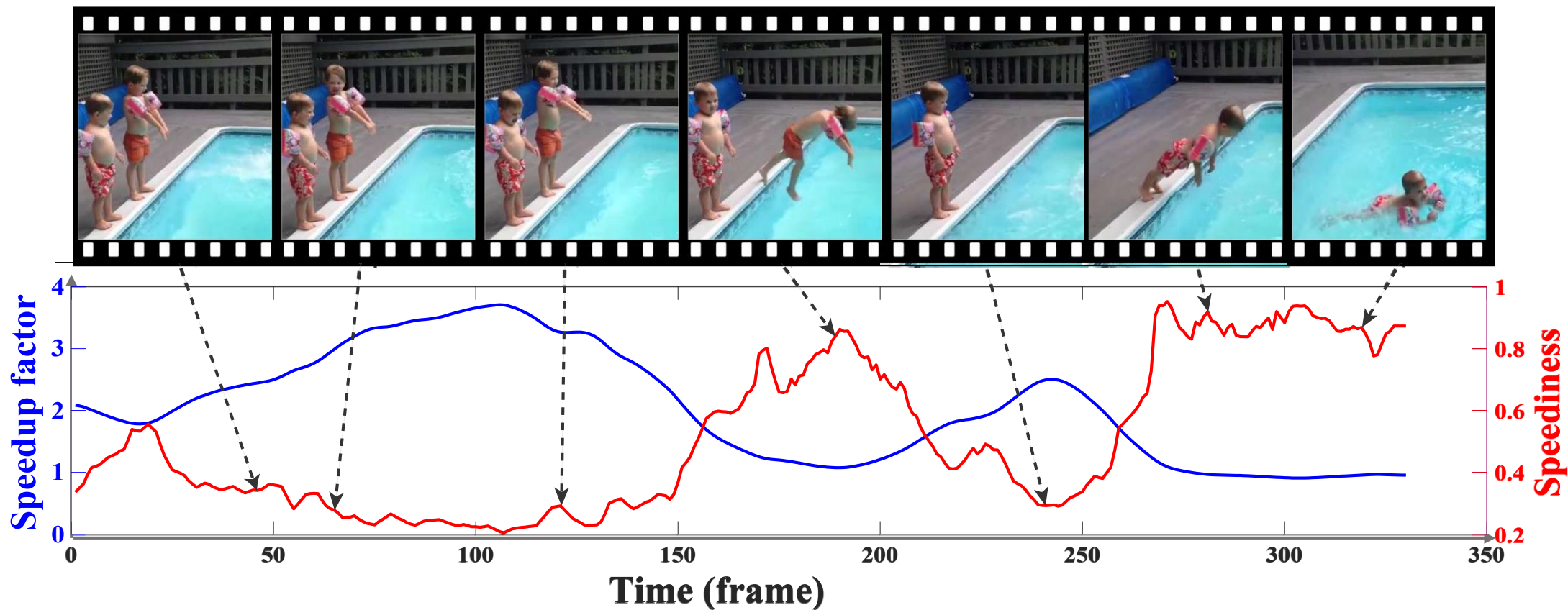


Self-Supervision in Videos: Learning the Speediness in Videos

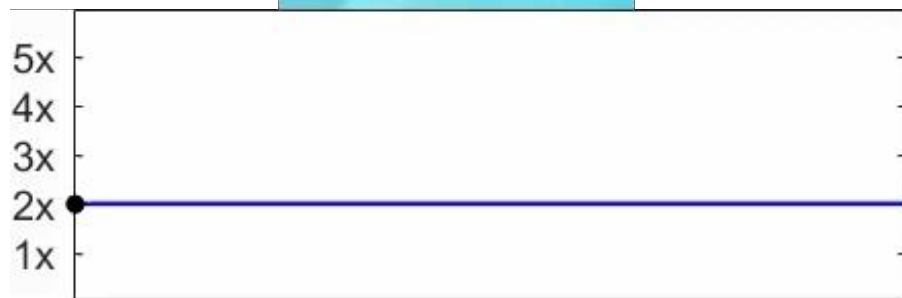
From “Speediness” to Speedup factor:

Low speediness \rightarrow speedup more

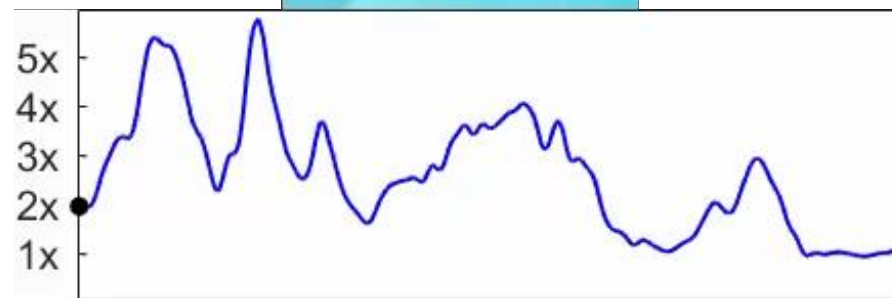
High speediness \rightarrow speedup less



Learning the Speediness in Videos: Adaptive Video Speedup



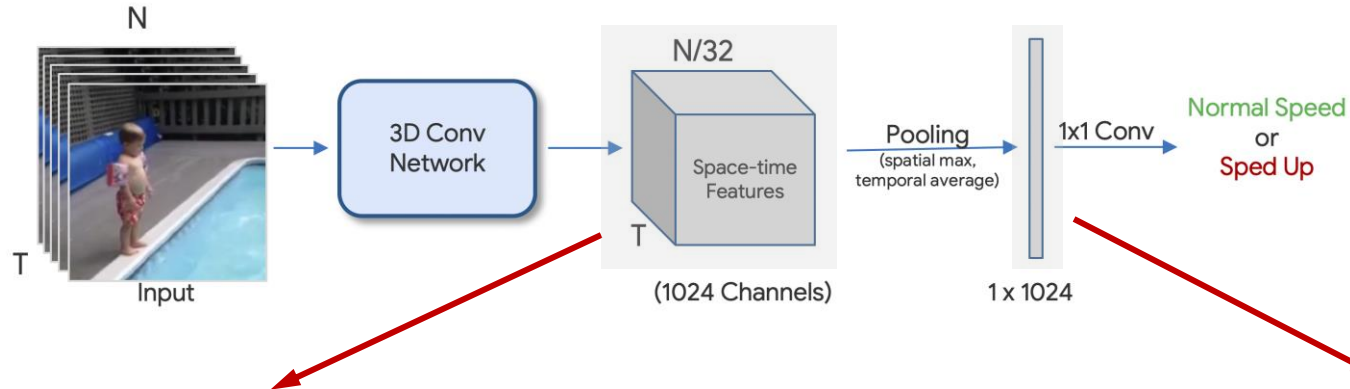
Uniform Speedup 2x



Adaptive Speedup 2x (ours)

Learning the Speediness in Videos: Transfer Learning

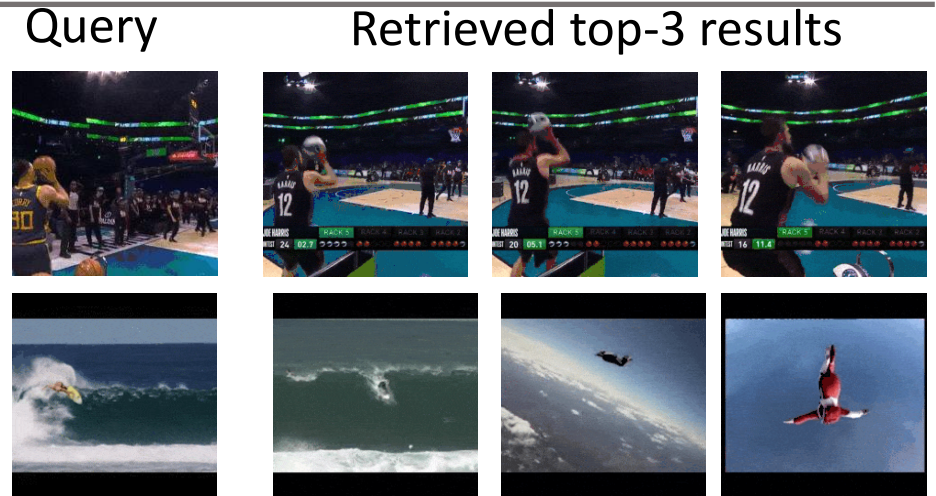
Pre-trained
SpeedNet



Self Supervised Action Recognition

Method	Initialization		Supervised accuracy	
	Architecture	UCF101	HMDB51	
Random init	S3D-G	73.8	46.4	
ImageNet inflated	S3D-G	86.6	57.7	
Kinetics supervised	S3D-G	96.8	74.5	
CubicPuzzle [19]	3D-ResNet18	65.8	33.7	
Order [40]	R(2+1)D	72.4	30.9	
DPC [13]	3D-ResNet34	75.7	35.7	
AoT [38]	T-CAM	79.4	-	
SpeedNet (Ours)	S3D-G	81.1	48.8	

Video Retrieval



Learning the Speediness in Videos: CAM visualizations



“Memory Eleven”
artistic video by Bill Newsing



Our space-time
speediness visualization

blue/green =
normal speed

yellow/orange =
slowed down



https://www.youtube.com/watch?v=djylS0Wi_Io

Speediness \neq Magnitude of Optical Flow

Far from camera Not in frame Close to camera



The Generative AI Revolution



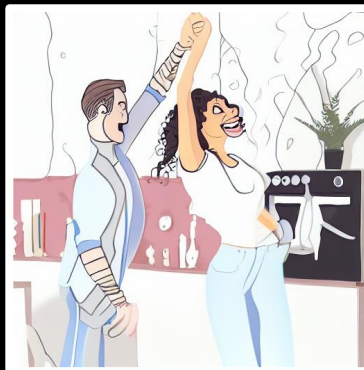
The Generative AI Revolution



The Generative AI Revolution



**Text-to-Image
Model**



What about videos?

A cat next to a dog



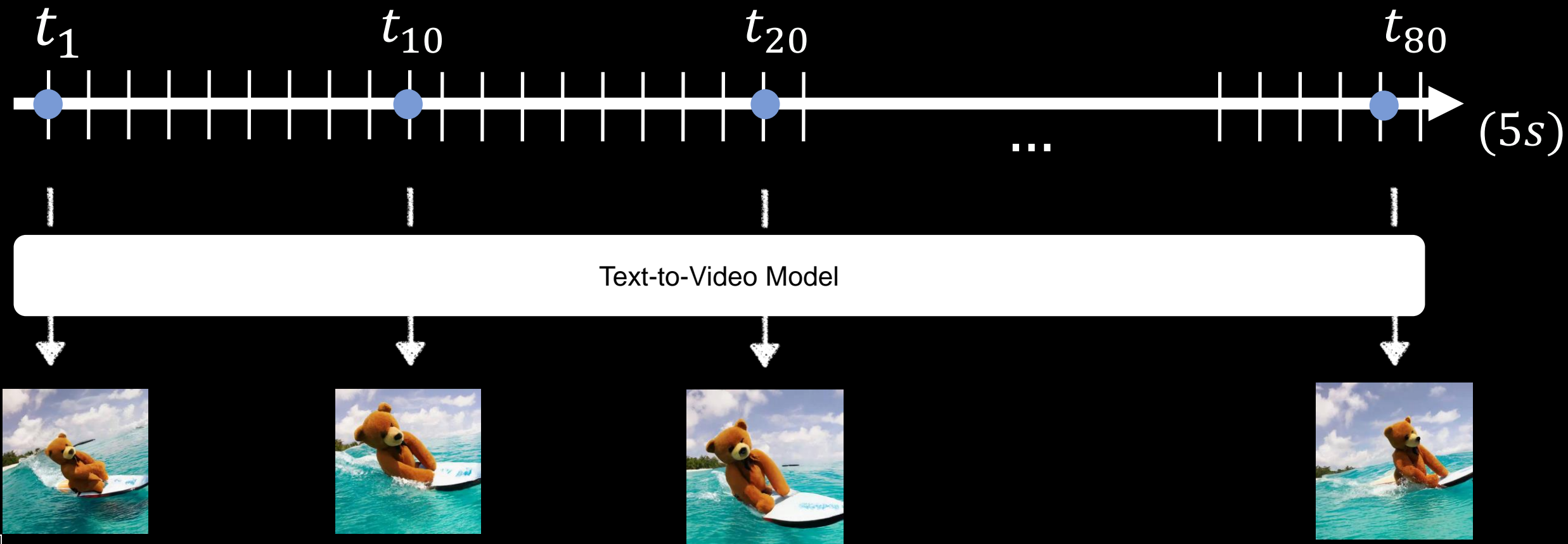
LUMIERE

Processing video is expensive!

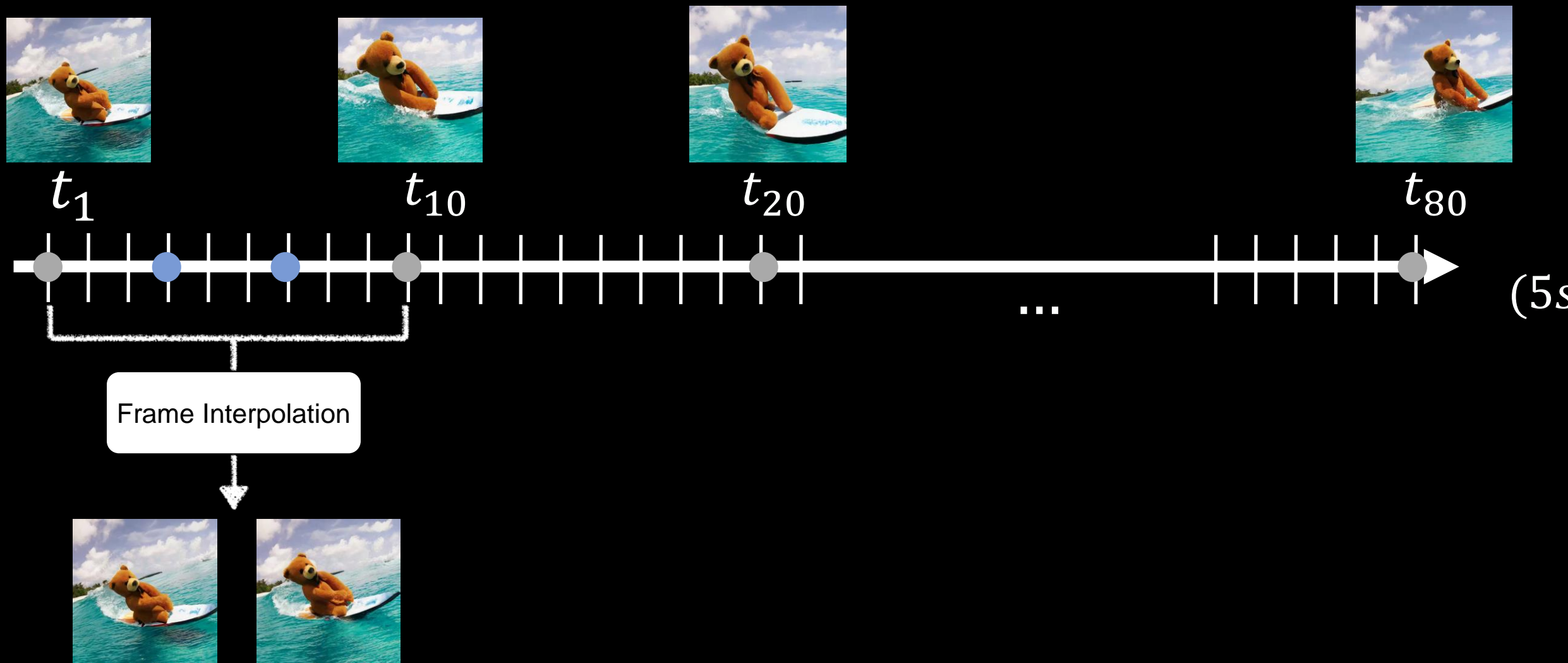
Common approach:

- 1) generate distant keyframes
- 2) fill-in missing content

1) Generate distant keyframes



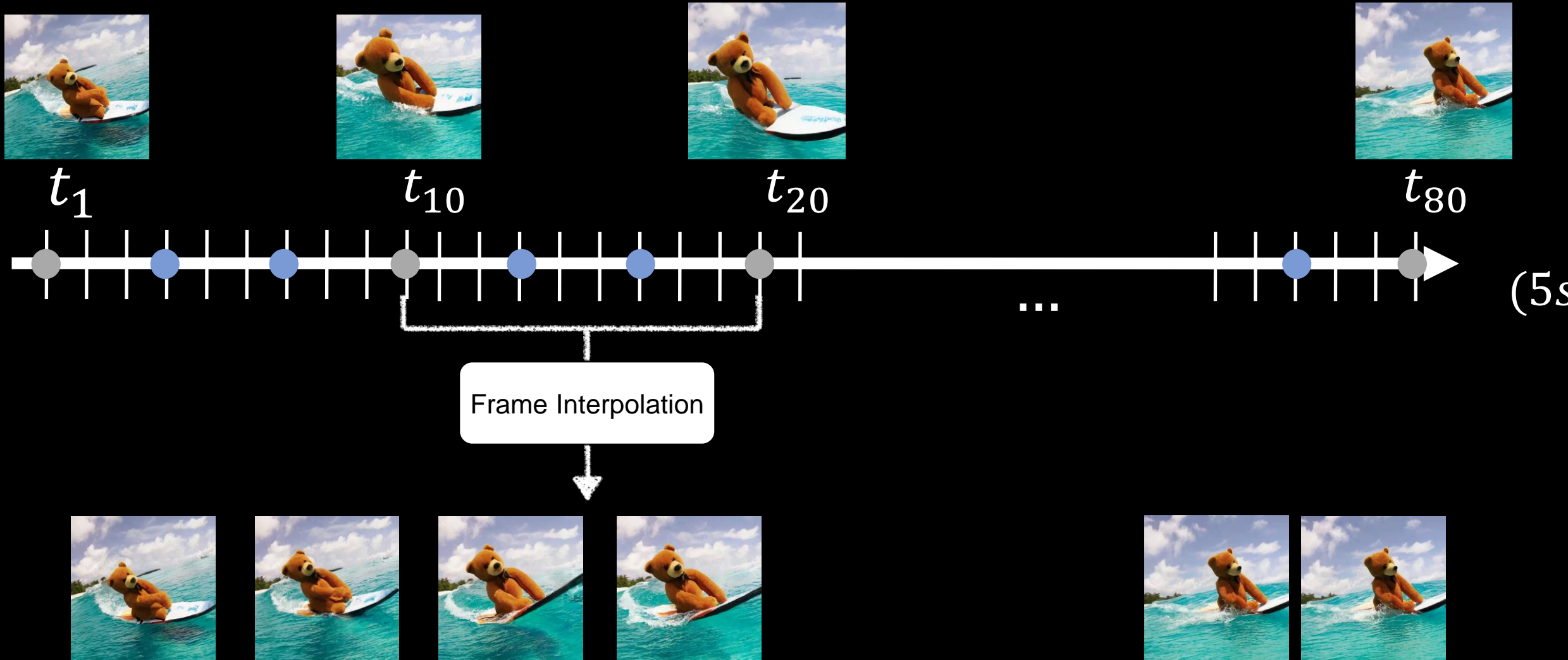
2) Fill-in missing frames



2) Fill-in missing frames

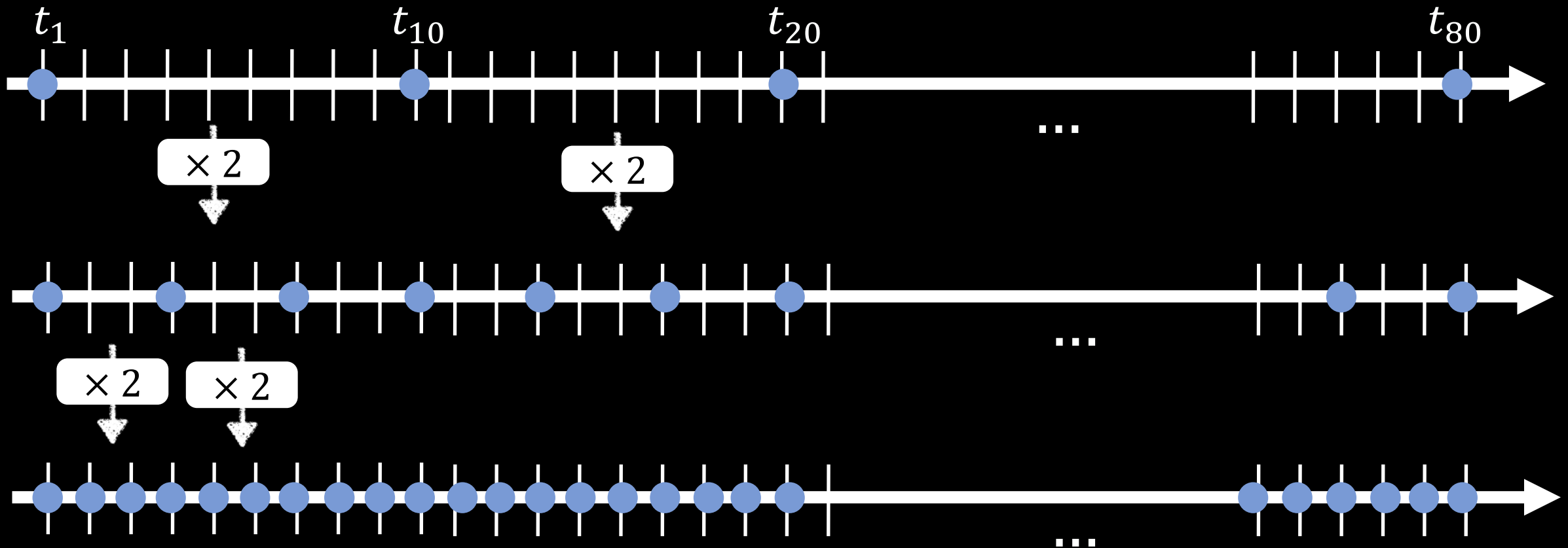


2) Fill-in missing frames



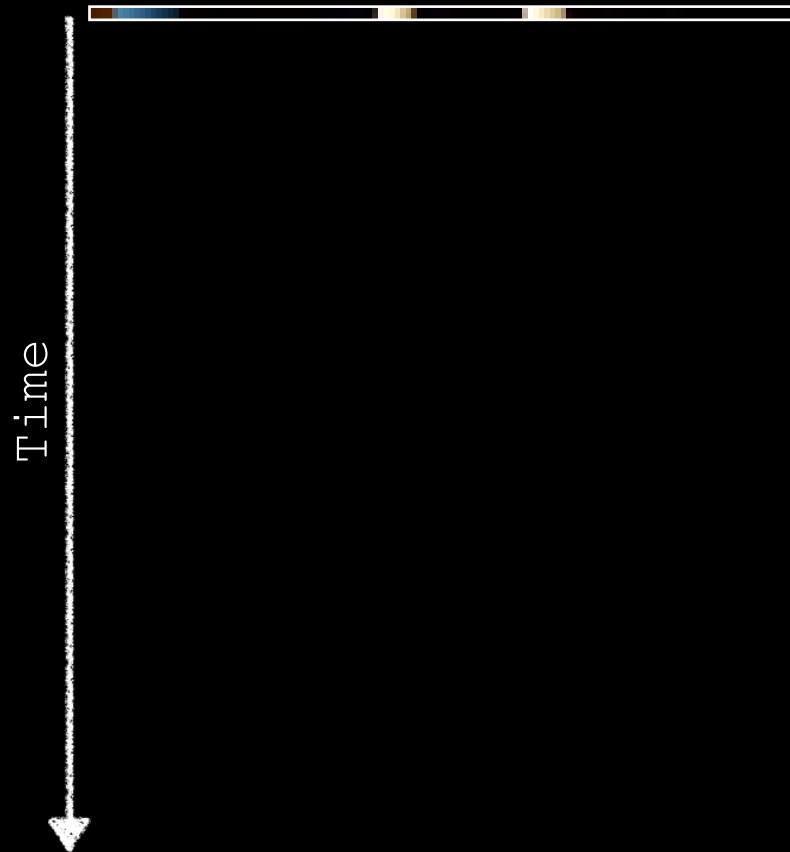
Common approach: distant keyframes → frame interpolation

In practice, requires a system of several frame interpolation models



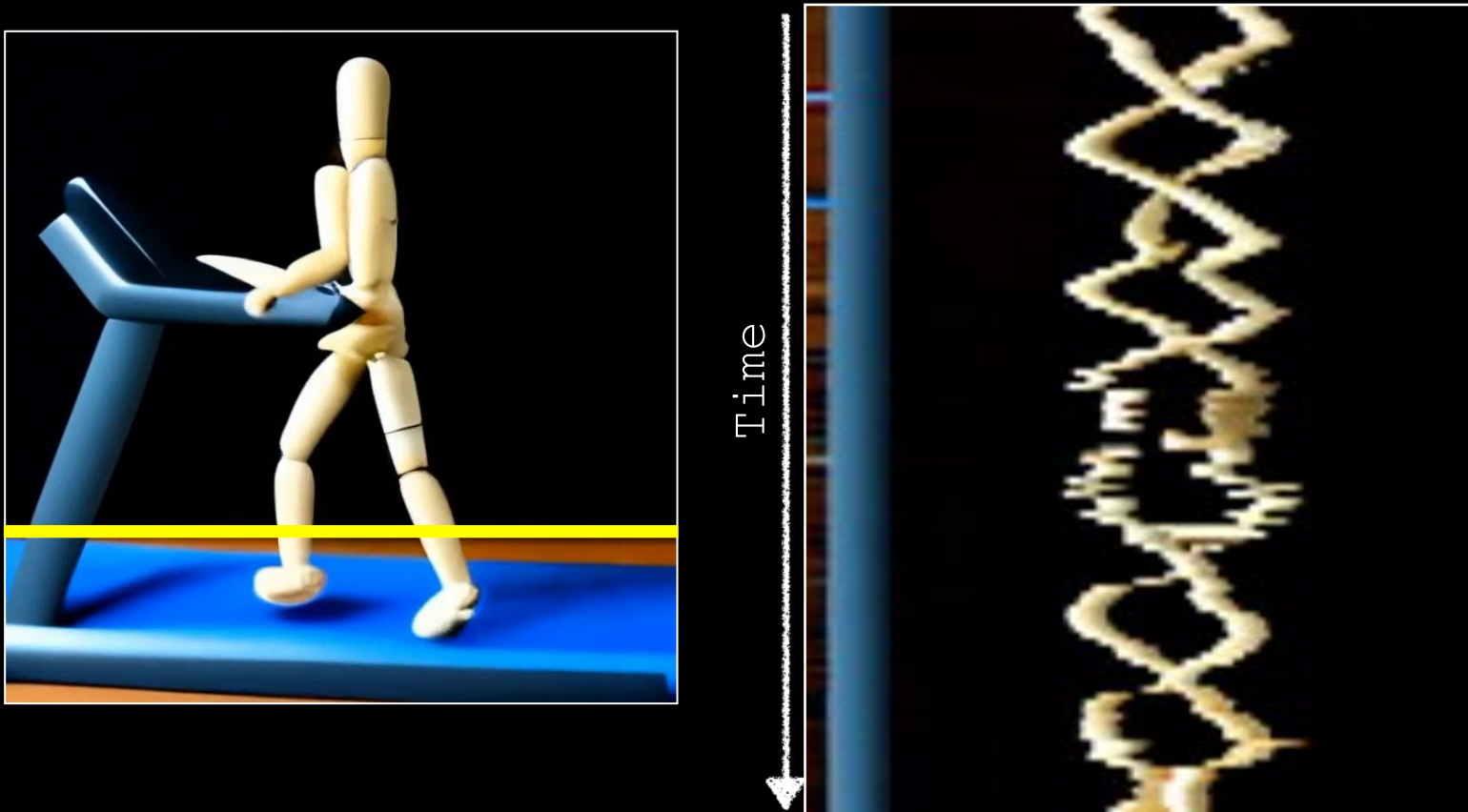
Common approach: distant keyframes → frame interpolation

Aliasing ambiguities cannot be consistently resolved!



Common approach: distant keyframes → frame interpolation

Aliasing ambiguities cannot be consistently resolved!



Common approach: distant keyframes → frame interpolation

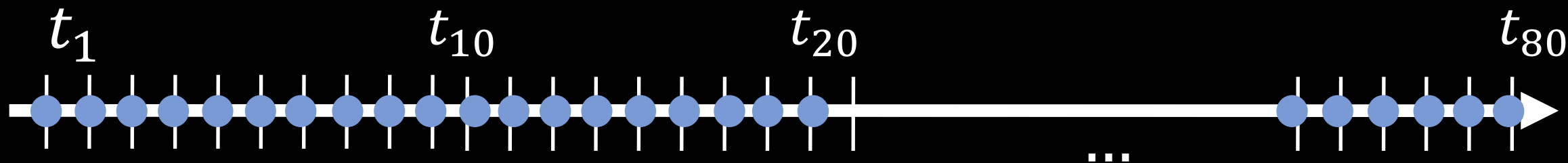
Mistakes in the keyframes are propagated in the interpolation stage



Our approach: generate all frames *at once*

Input text prompt

Text-to-Video



(5s)

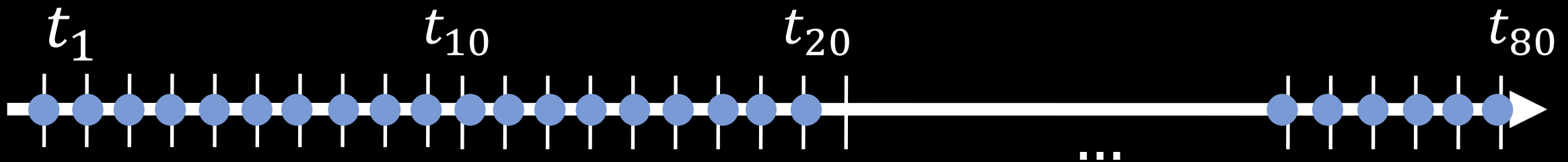
Our approach: generate all frames *at once*

Allows learning globally coherent motion

Simple framework

Input text prompt

Text-to-Video

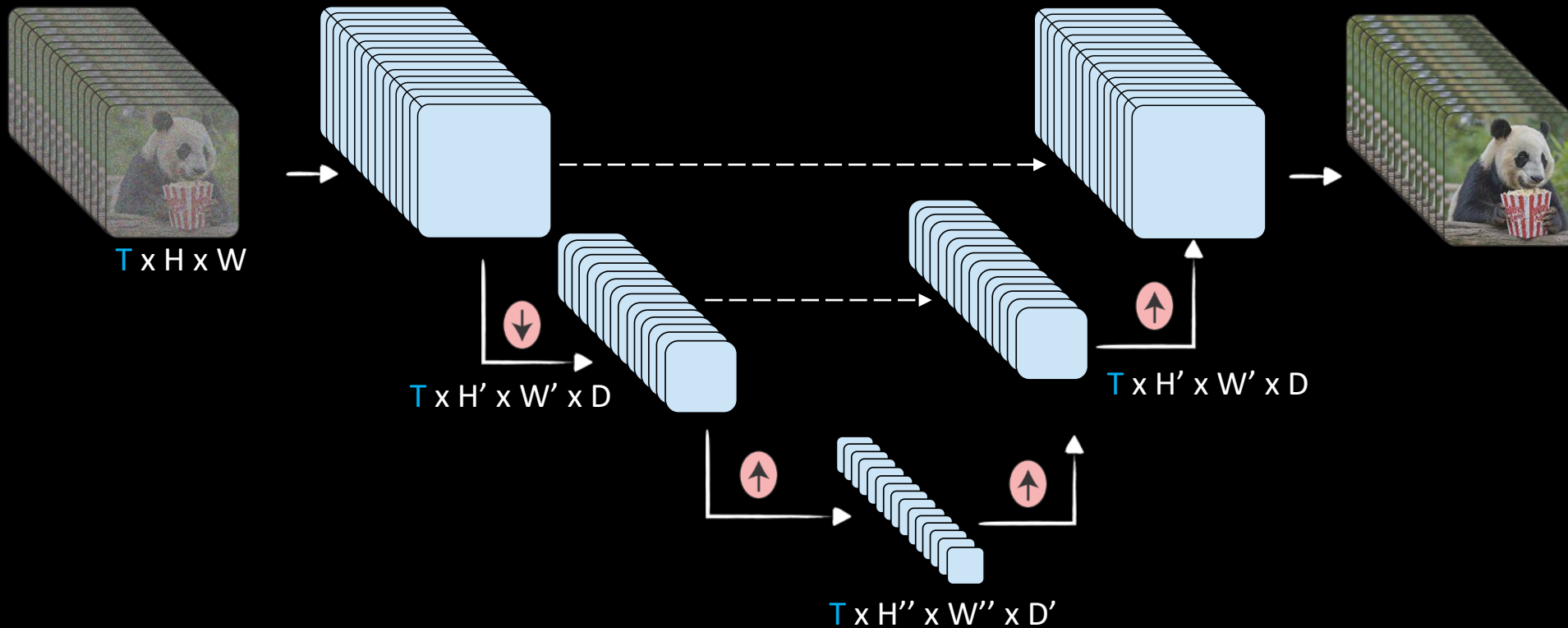


(5s)

Common UNet

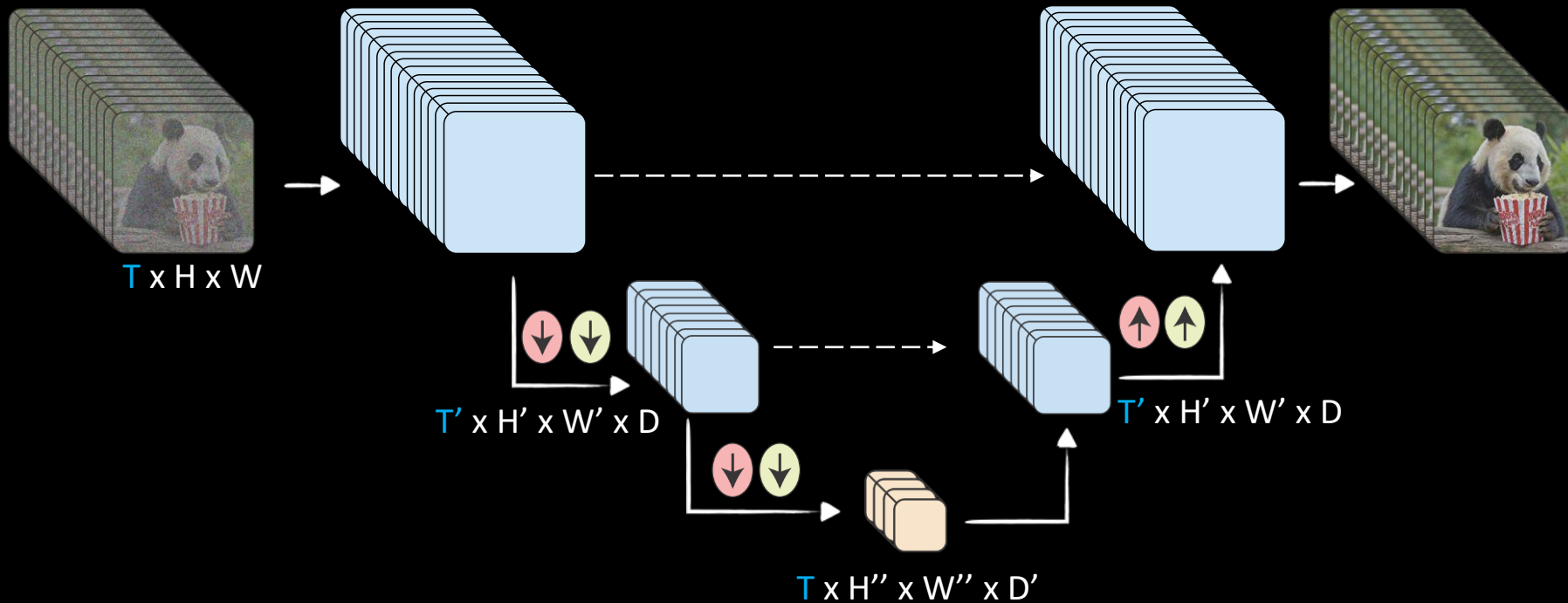
Spatial sub-sampling

No temporal sub-sampling



Space-Time UNet (STUNet)

Spatial sub-sampling
Temporal sub-sampling



Conv inflation block

Pretrained T2I layer

2D Convolution

Norm + activation

1D Convolution

Norm + activation

Linear Projection

+

Attention inflation block

Pretrained T2I layer

1D Attention (xL)

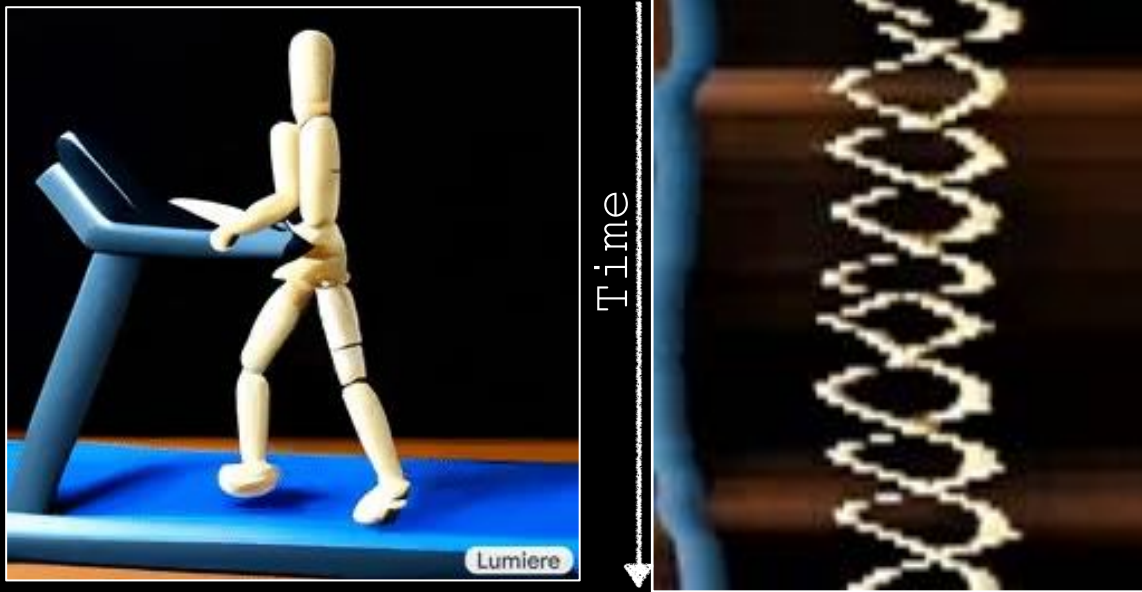
Linear Projection

+

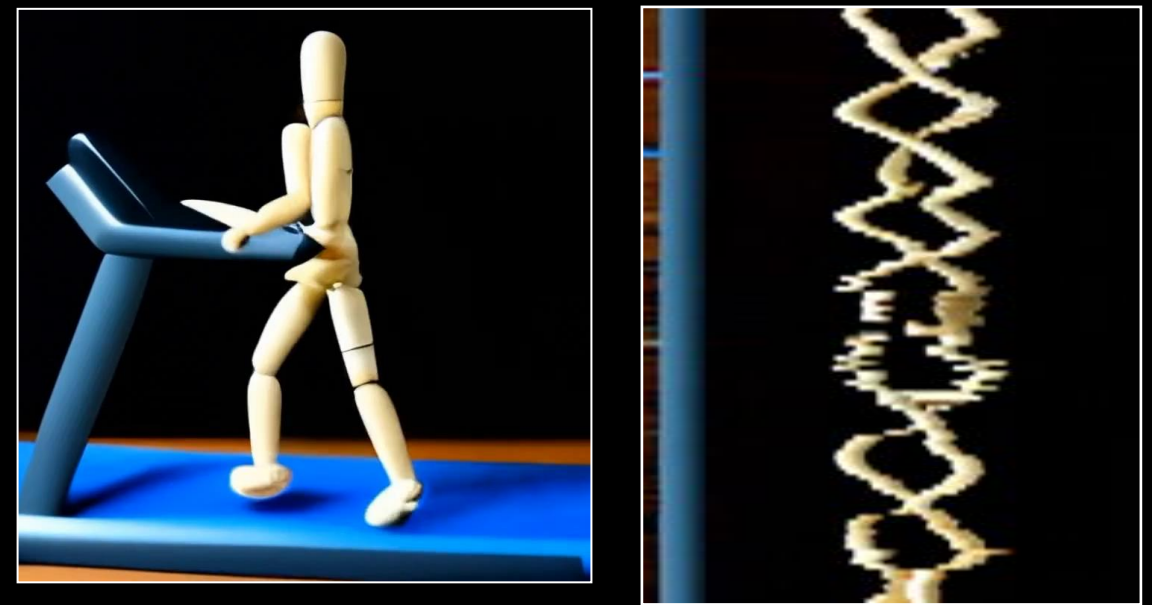
Space-Time UNet (STUNet)



Space-Time UNet (STUNet)



Cascaded Approach



Text-to-Video

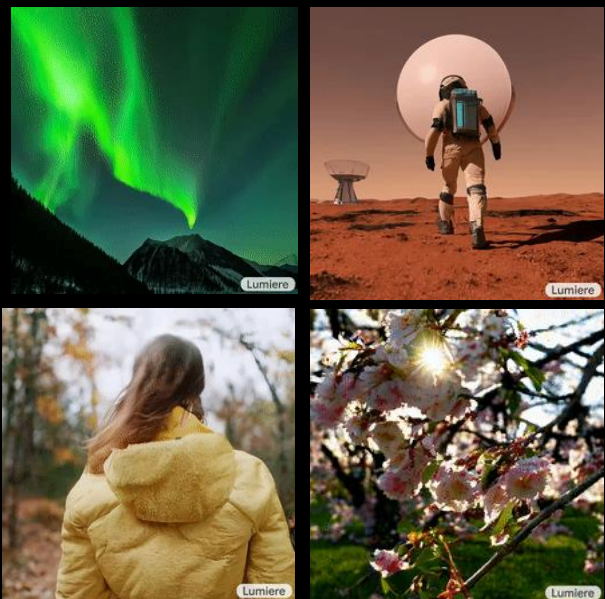
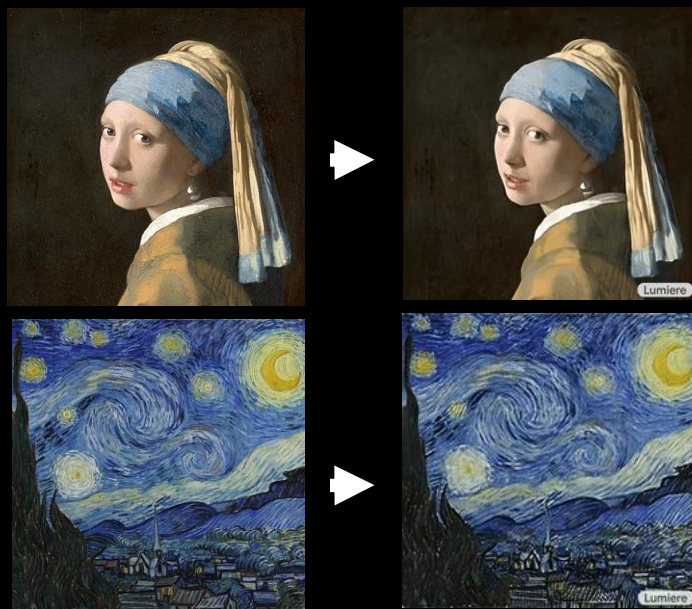
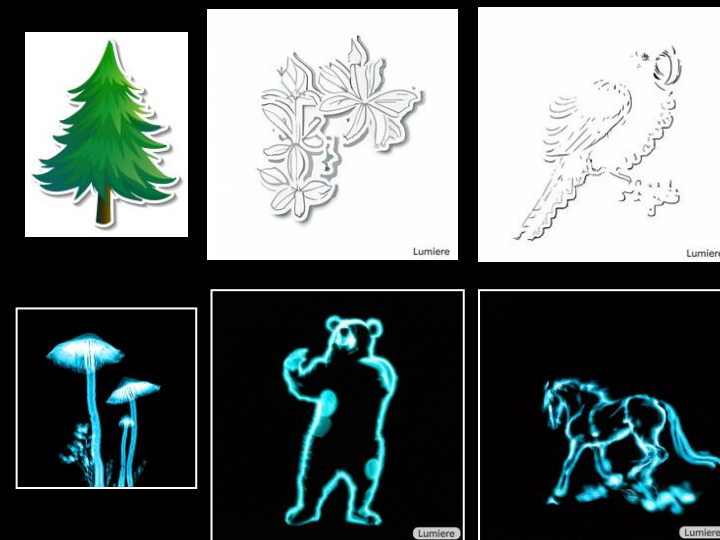


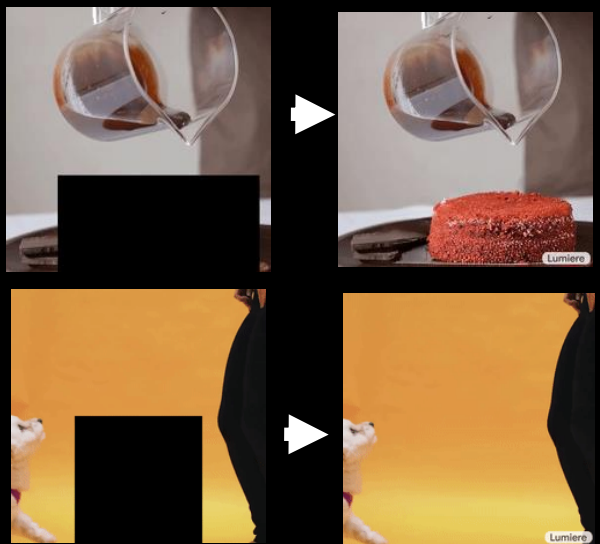
Image-to-Video



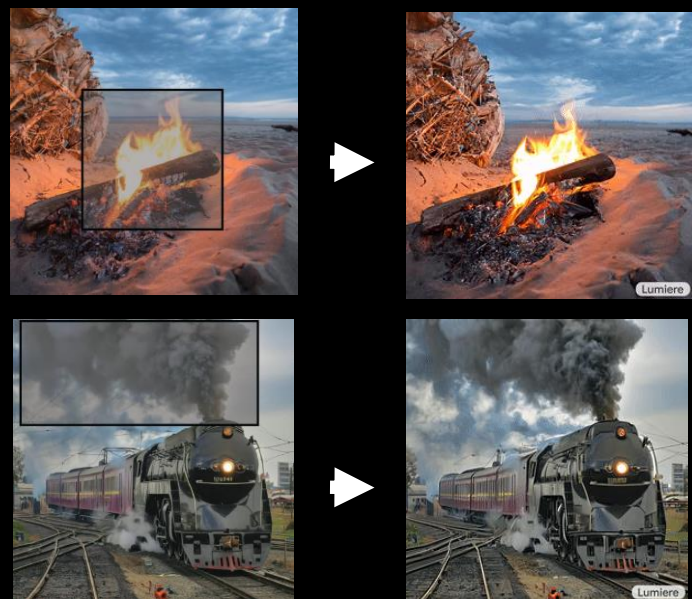
Stylized Generation



Inpainting



Cinemagraphs



Editing

