Temporal Cycle Consistency Learning

Debidatta Dwibedi, Yusuf Aytar,
Jonathan Tompson, Pierre Sermanet, Andrew Zisserman

Google

DeepMind

Problem Setup

Suppose we have multiple unaligned videos of the same activity:

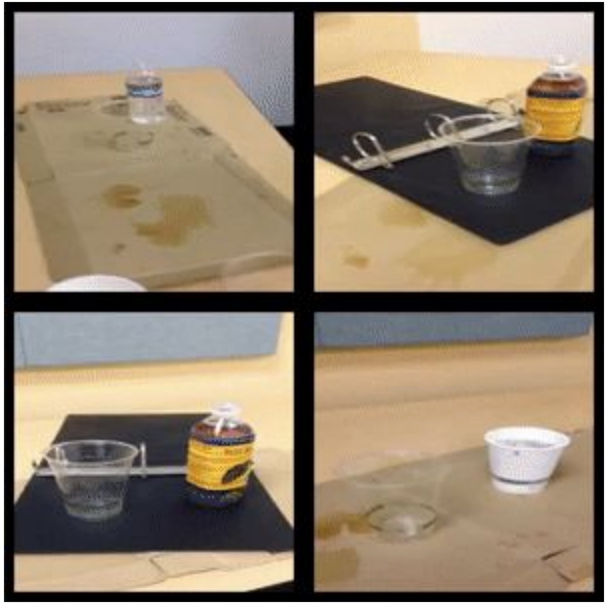
- from different viewpoints
- with different objects
- with camera motion
- with different pace

Goal: time-align the videos

Why would we want to do this?

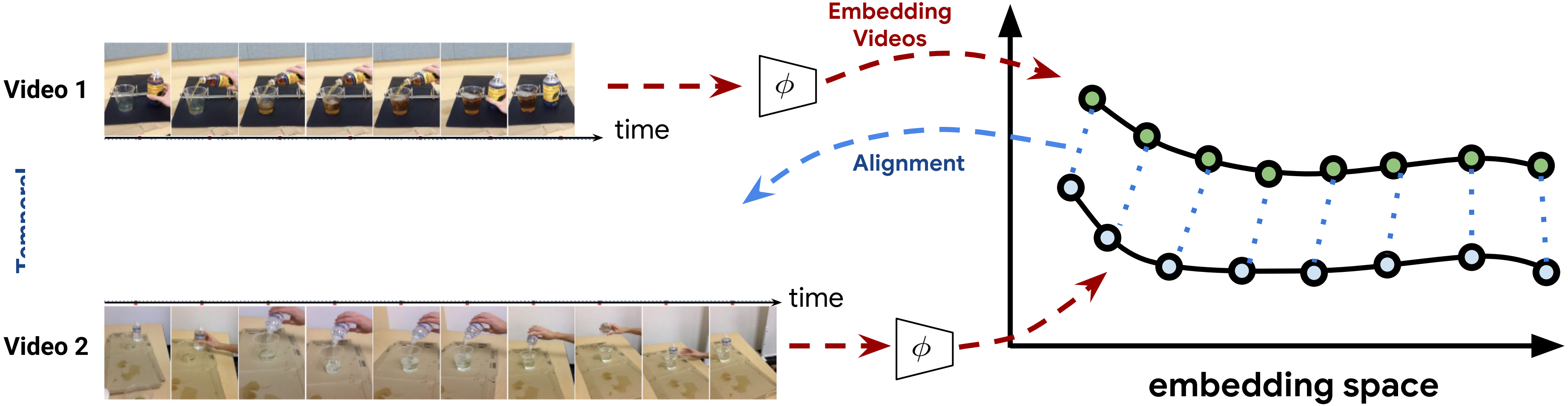
- to be able to compare videos
- to be able to learn from their alignments
- to be able to learn action phases

Example Videos: Pouring



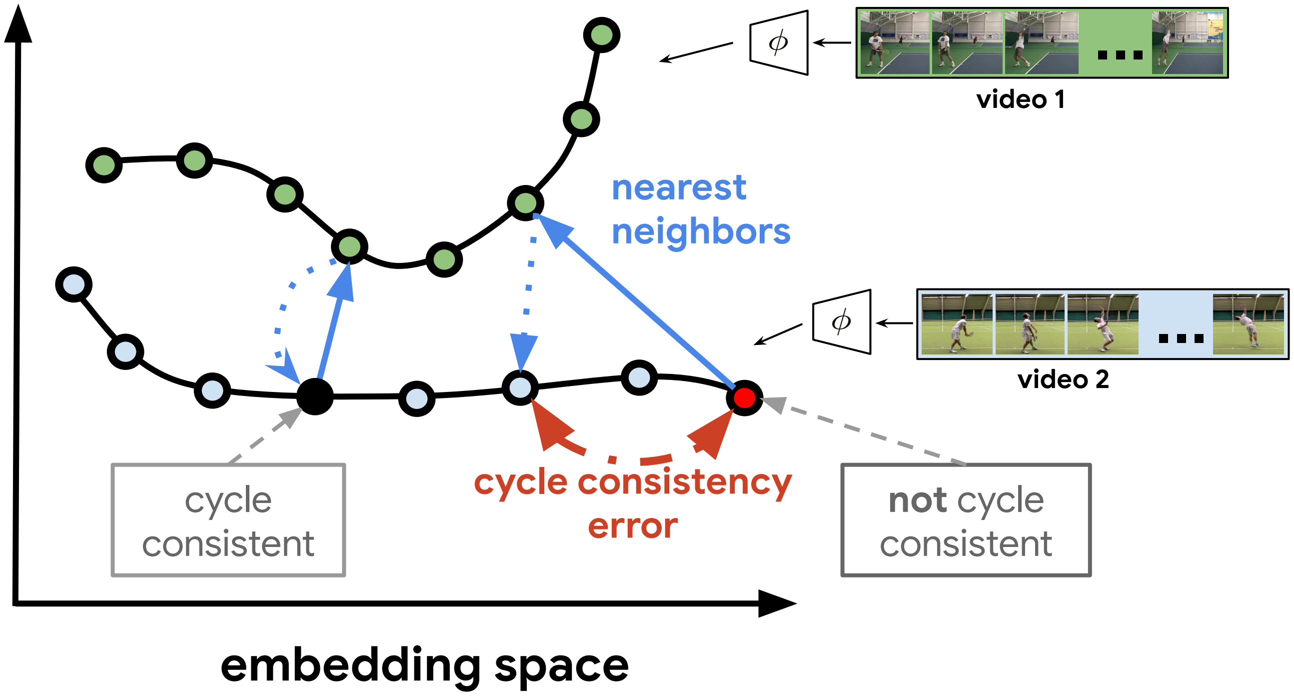
Temporal Cycle Consistency Learning

Self-supervised representation learning through temporal alignment



Finding correspondences across multiple videos despite many factors of variation

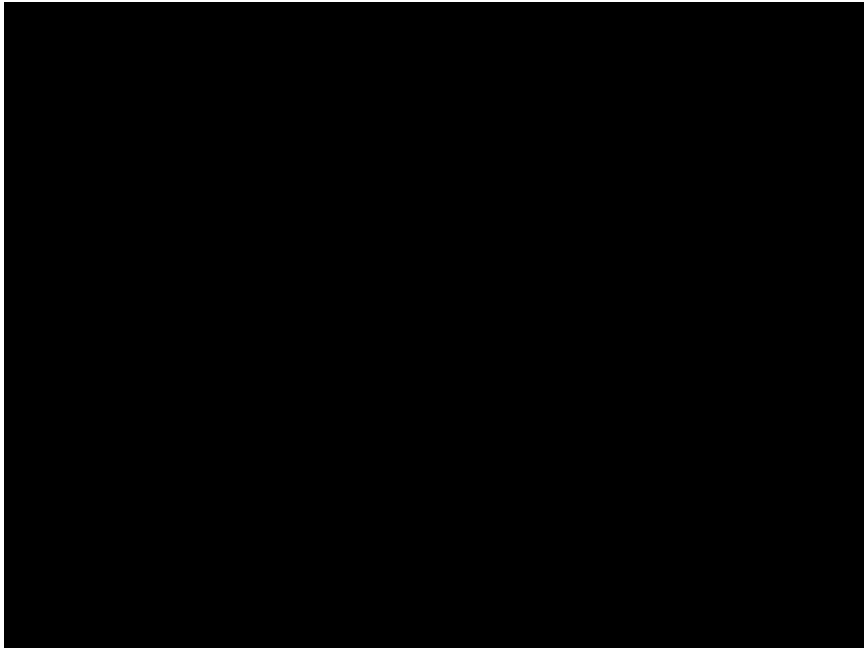
Cycle Consistency



Motivation

Once we have learnt the embedding space ...

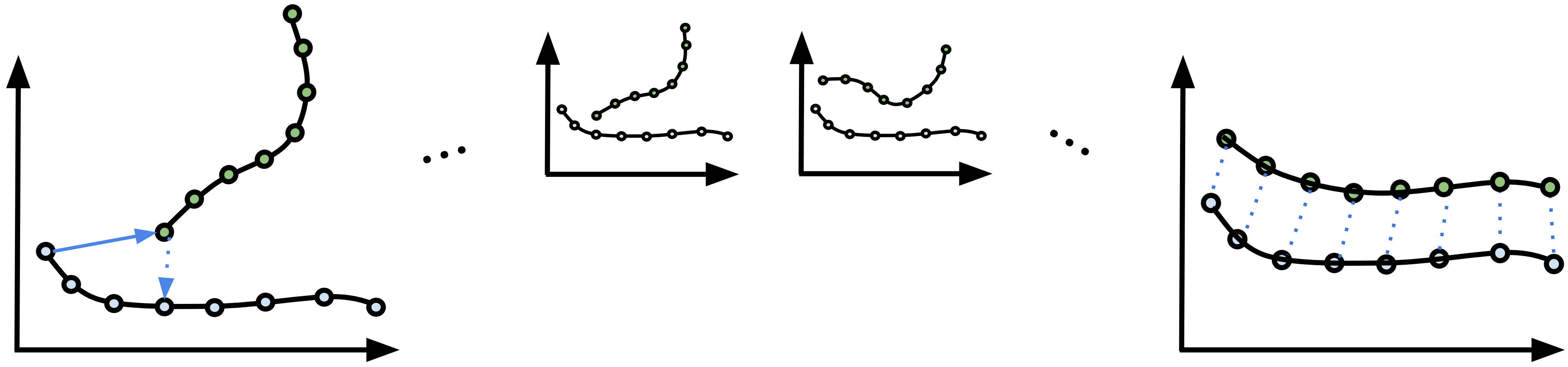
it can be used for aligning videos and encoding phase



https://www.youtube.com/watch?v=iWjjeMQmt8E

Differentiable Cycle Consistency

Maximizing one-to-one mapping capacity

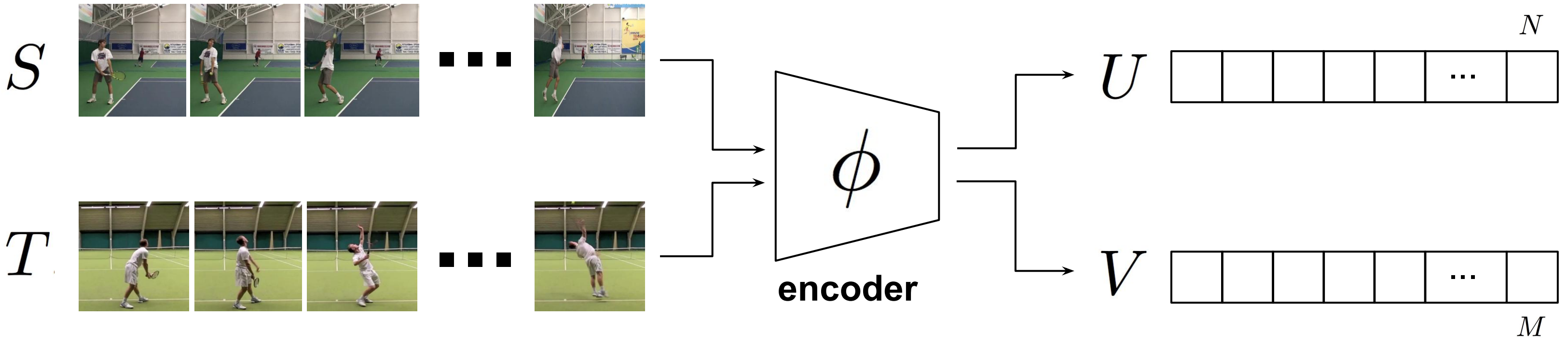


low cycle consistency

high cycle consistency

A differentiable objective

Video Embedding

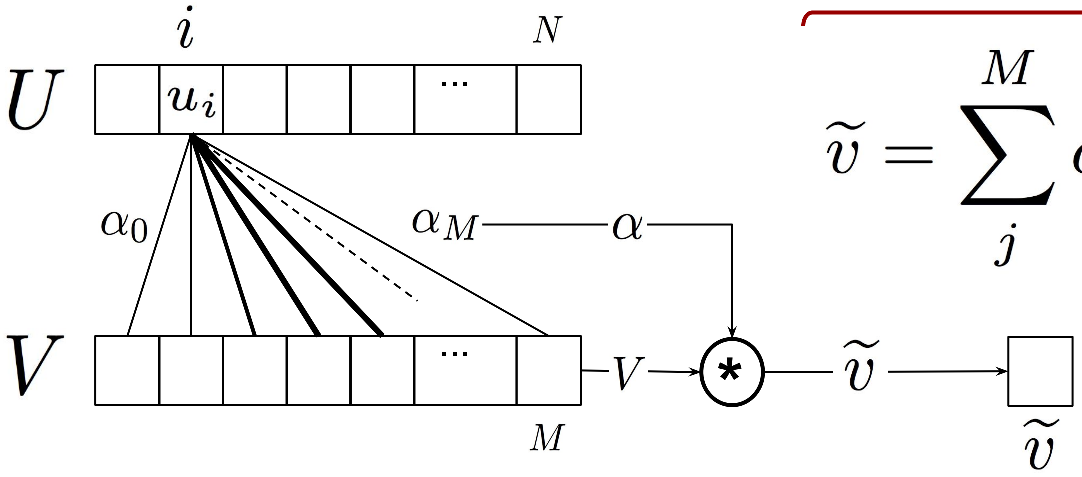


$$U = \{u_1, u_2, ..., u_N\} \qquad V = \{v_1, v_2, ..., v_M\}$$

Differentiable Cycle Consistency

Soft nearest neighbor

$$\underbrace{\alpha_j = \frac{e^{-||u_i - v_j||_2}}{\sum_k^M e^{-||u_i - v_k||_2}}}_{\text{similarity distribution for } u_i}$$

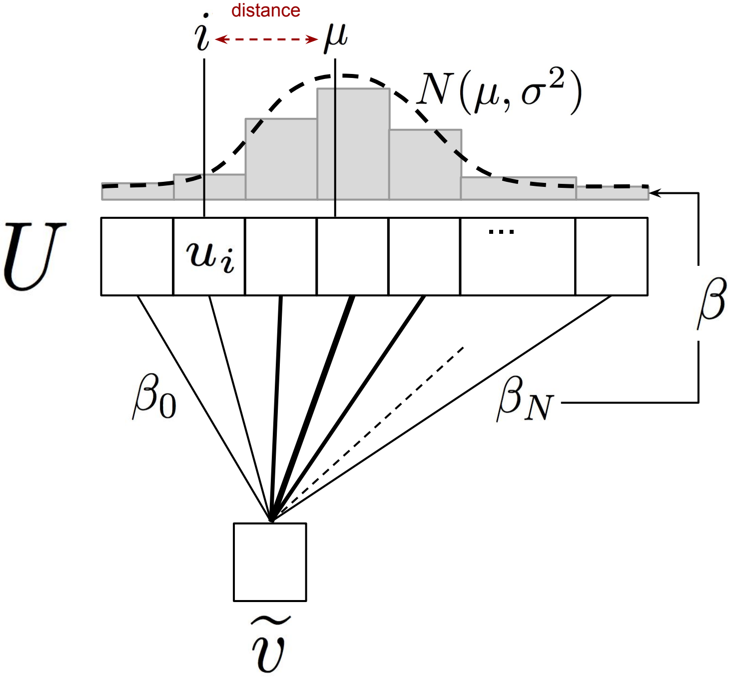


$$\overbrace{\tilde{v} = \sum_j^M \alpha_j v_j}^{\text{soft nearest neighbor of } u_i}$$

Cycle-back regression

Differentiable Cycle Consistency

$$\underbrace{\beta_k = \frac{e^{-||\widetilde{v}-u_k||_2}}{\sum_j^N e^{-||\widetilde{v}-u_j||_2}}}_{\text{similarity distribution for } \widetilde{v}}$$



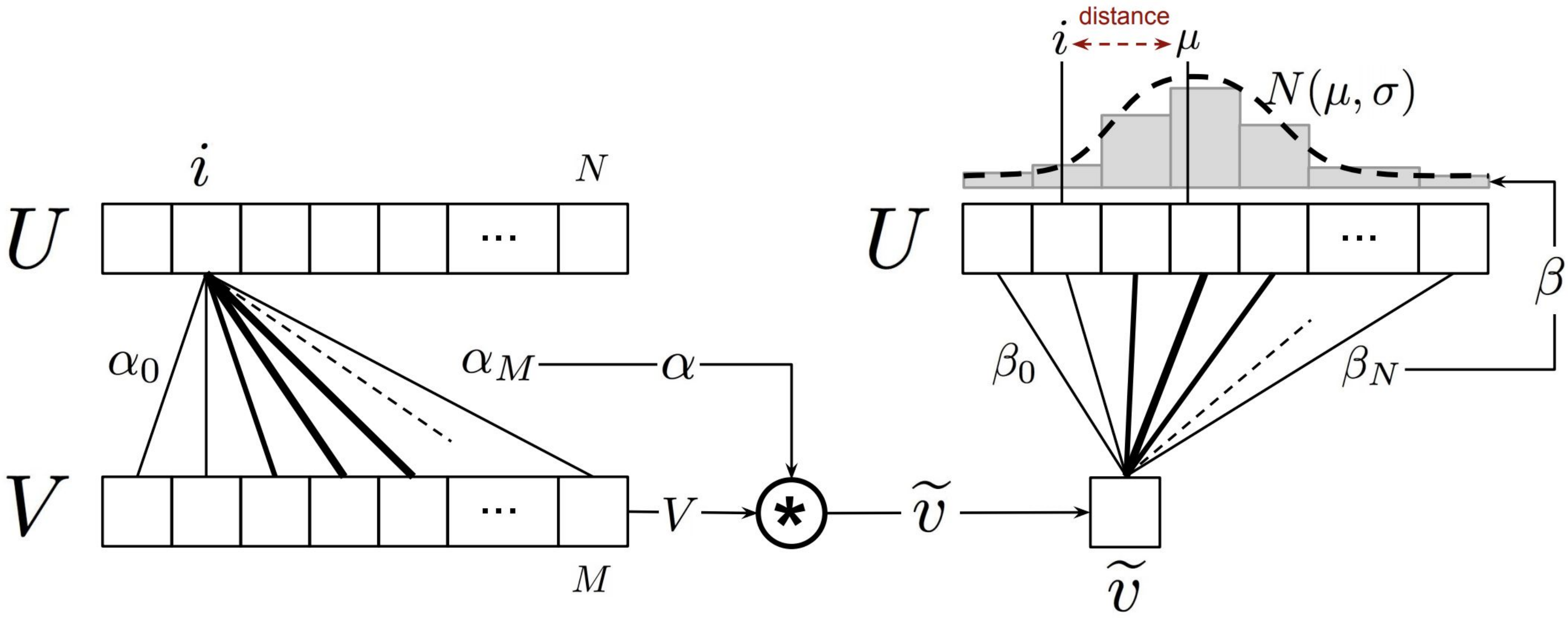
Objective Function:

$$L_{cbr} = \frac{|i-\mu|^2}{\sigma^2} + \lambda \log(\sigma)$$

$$\mu = \sum_k^N \beta_k * k$$

$$\sigma^2 = \sum_k^N \beta_k * (k-\mu)^2$$

TCC Learning



Datasets

Pouring & Penn Action

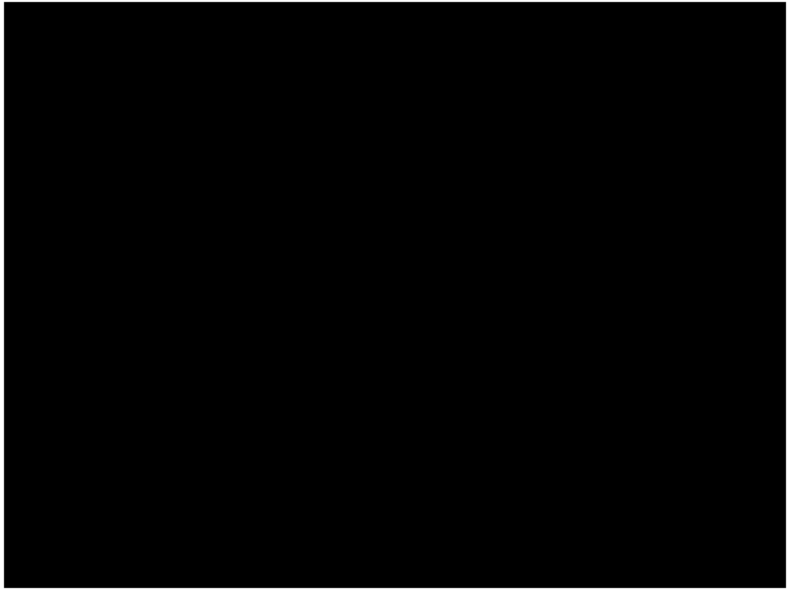
Pouring Dataset

Penn Actions Dataset

Applications

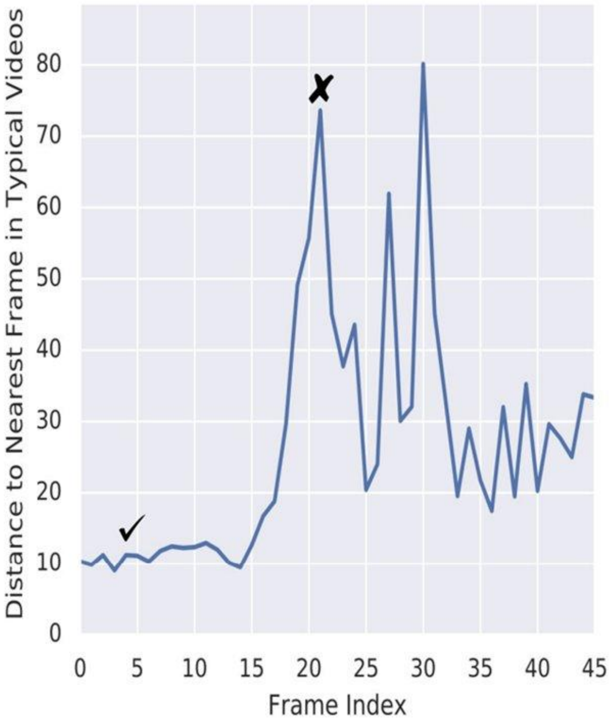
Pace Transfer

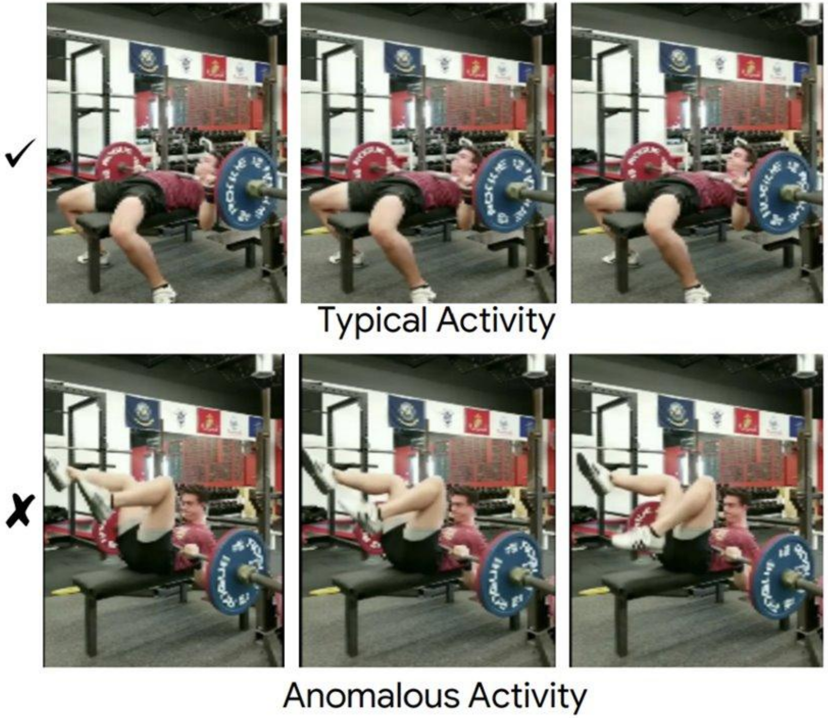
Synchronizing multiple videos



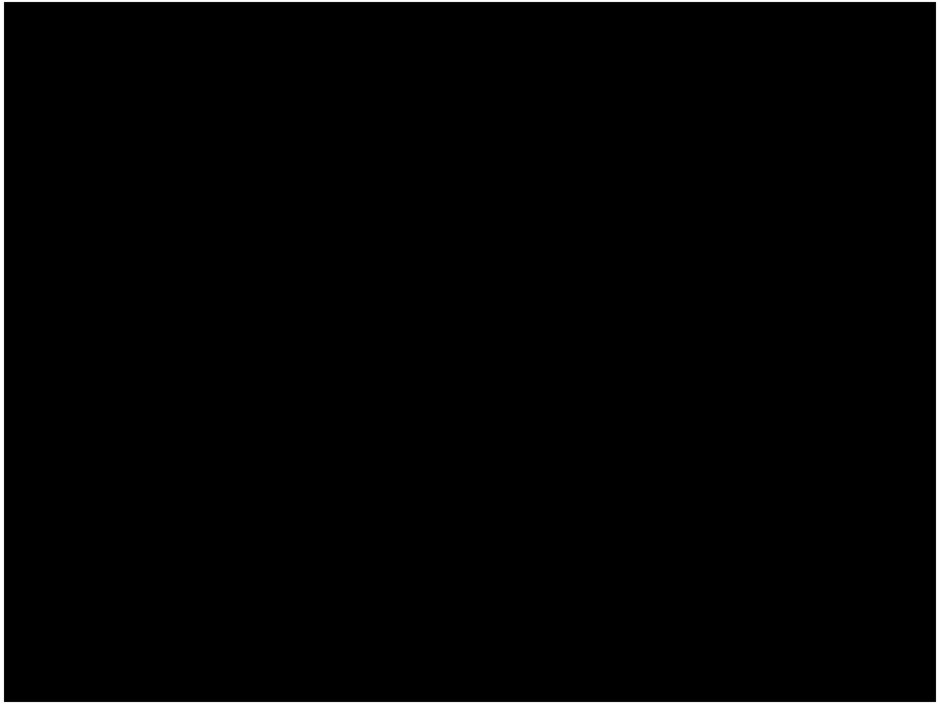
https://www.youtube.com/watch?v=iWjjeMQmt8E

Anomaly Detection



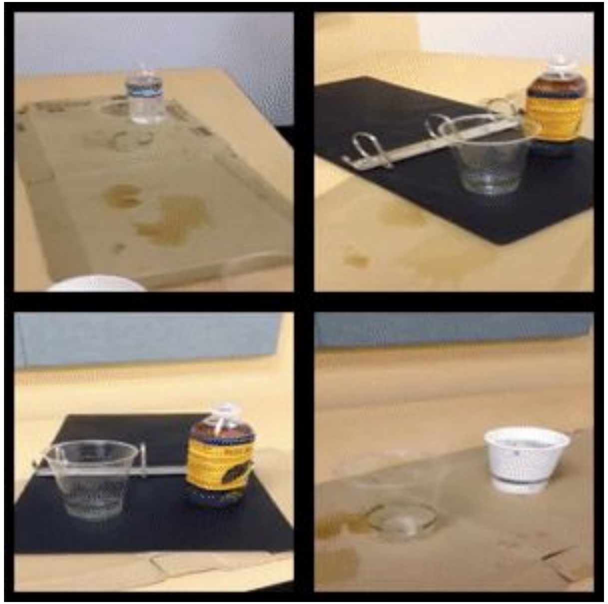


Sound Transfer



https://www.youtube.com/watch?v=ATDGVqX3INo

Understanding Multiple Stages of a Process



Note the **variation** in real world videos:

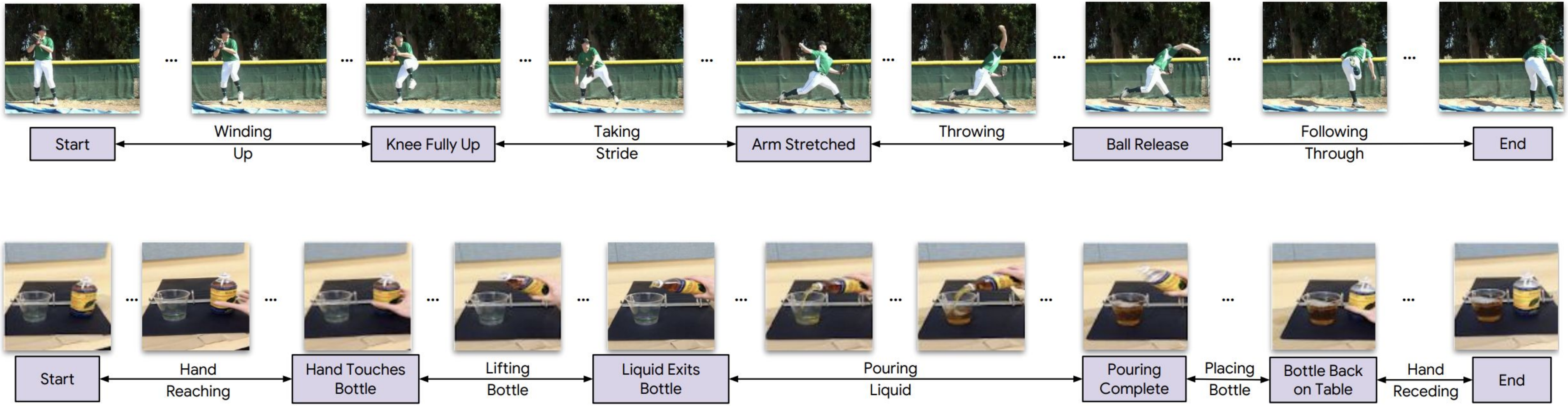
1. Viewpoint changes
2. Different objects
3. Camera Motion
4. Pace of the action

Key Events:

1. Hand touches the bottle
2. Liquid exits the bottle
3. Pouring complete
4. Bottle back on the table

Action Phase Classification

Pouring & Penn Action



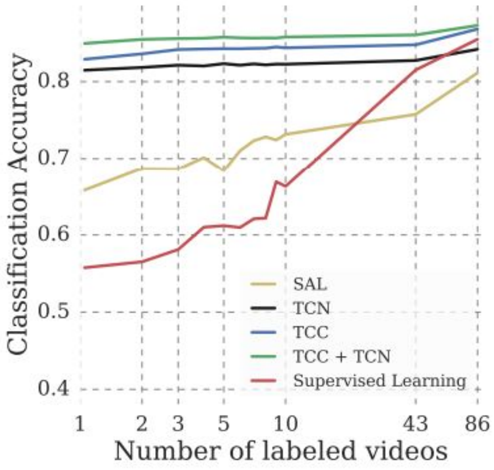
Example labels for the actions ‘Baseball Pitch’ (top row) and ‘Pouring’ (bottom row). The key events are shown in boxes below the frame (e.g. ‘Hand touches bottle’), and each frame in between two key events has a phase label (e.g. ‘Lifting bottle’).

Action Phase Classification

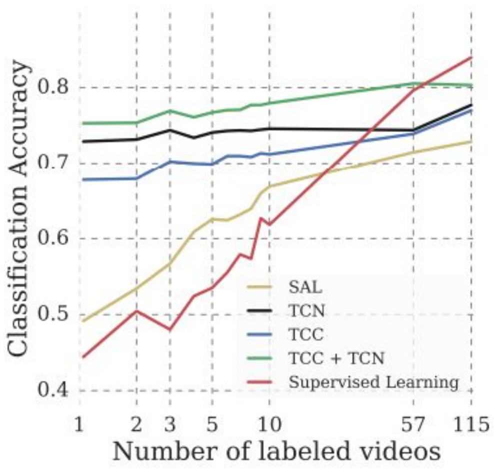
Results

Datasets	% of Labels →	0.1	0.5	1.0
Penn Action	Supervised Learning	50.71	72.86	79.98
	SaL [27]	66.15	71.10	72.53
	TCN [35]	69.65	71.41	72.15
	TCC (ours)	**74.68**	**76.39**	**77.30**
Pouring	Supervised Learning	62.01	77.67	88.41
	SaL [27]	74.50	80.96	83.19
	TCN [35]	76.03	83.27	84.57
	TCC (ours)	**86.82**	**89.43**	**90.21**

Few Shot Action Phase Classification



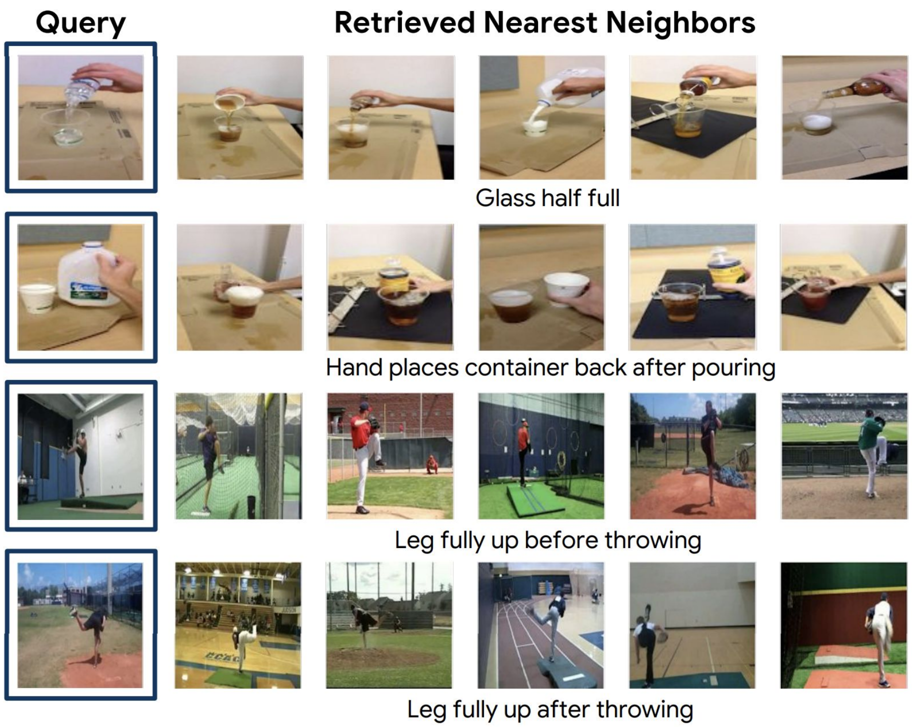
(a) Golf Swing



(b) Tennis Serve

- Few shot classification benefits from self-supervised pre-training
- Similar conclusion in "Data-Efficient Image Recognition with Contrastive Predictive Coding", Henaff, et al

Fine grained retrieval



Conclusion

- Self-supervised representation learning method.
- Based on temporal alignment of videos.
- Useful per-frame features for fine-grained tasks.