Two simple Models for Temporal Action Localization

Jiaqi Su

Overview

- Temporal Action Localization
- Every Moment Counts: Dense Detailed Labeling of Actions in Complex Videos
- Predictive-Corrective Networks for Action Detection

Problem Recap

Temporal Action Localization:

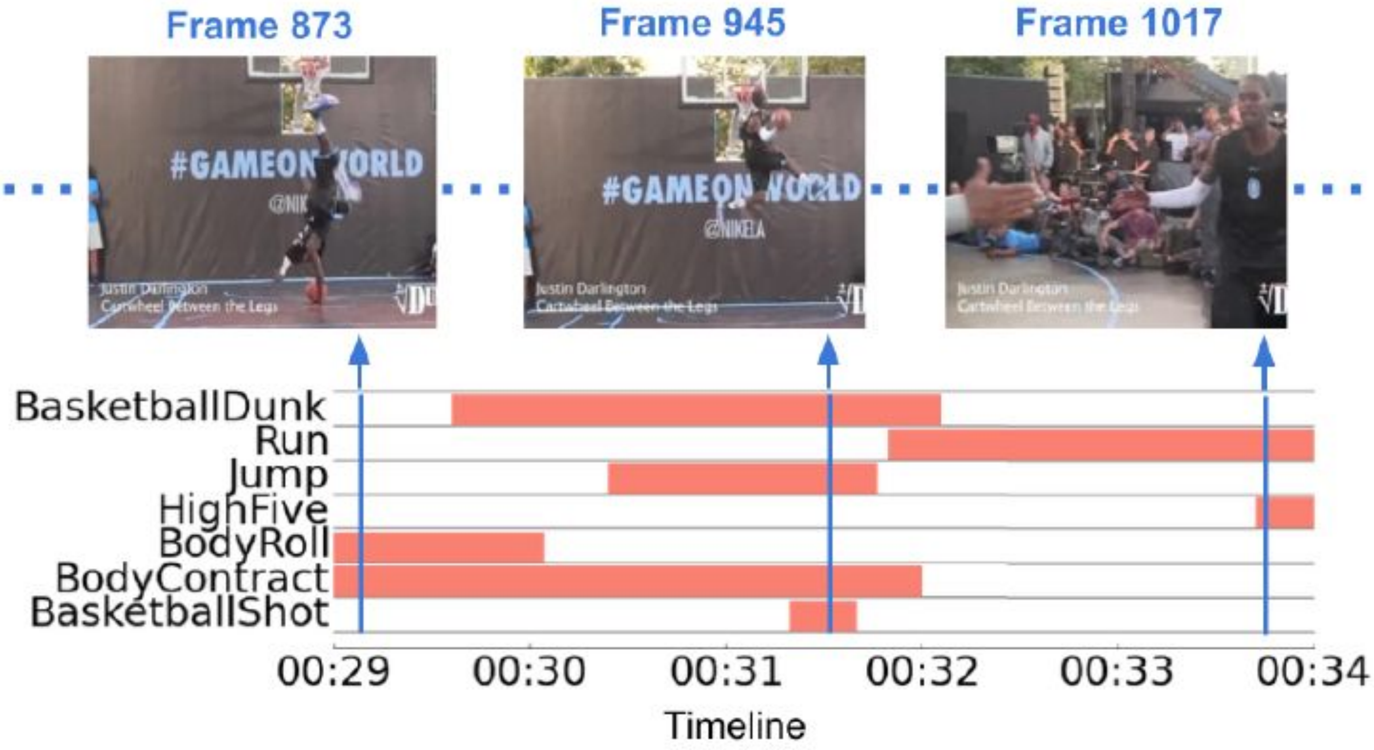
Recognize action, as well as the temporal segment where the action happened in the video.

Every Moment Counts: Dense Detailed Labeling of Actions in Complex Videos

Yeung et al.

- **Motivation**
  - Dense detailed multi-label action understanding
- **Find right dataset**
  - MultiTHUMOS Dataset
- **Develop right model**
  - MultiLSTM Model
- **Experiments**

Motivation



Frame 873
Frame 945
Frame 1017
BasketballDunk
Run
Jump
HighFive
BodyRoll
BodyContract
BasketballShot
00:29
00:30
00:31
00:32
00:33
00:34
Timeline

Road Map

Target problem of **dense detailed multi-label action understanding**

(1) Finding the right dataset

(2) Developing an appropriate model

Road Map

Target problem of **dense detailed multi-label action understanding**

(1) Finding the right dataset

(2) Developing an appropriate model

Exisiting Datasets

	Detection	Untrimmed	Open-world	Multilabel
UCF101 [37]	-	-	yes	-
HMDB51 [14]	-	-	yes	-
Sports1M [10]	-	yes	yes	-
Cooking [29]	yes	yes	-	-
Breakfast [13]	yes	yes	-	-
THUMOS [9]	yes	yes	yes	-
MultiTHUMOS	yes	yes	yes	yes

- **Detection**: temporal localization annotation
- **Untrimmed**: long enough to capture consecutive actions
- **Open-world**: generality of videos, a broad set of actions
- **Multilabel**: label all simultaneous actions in a frame

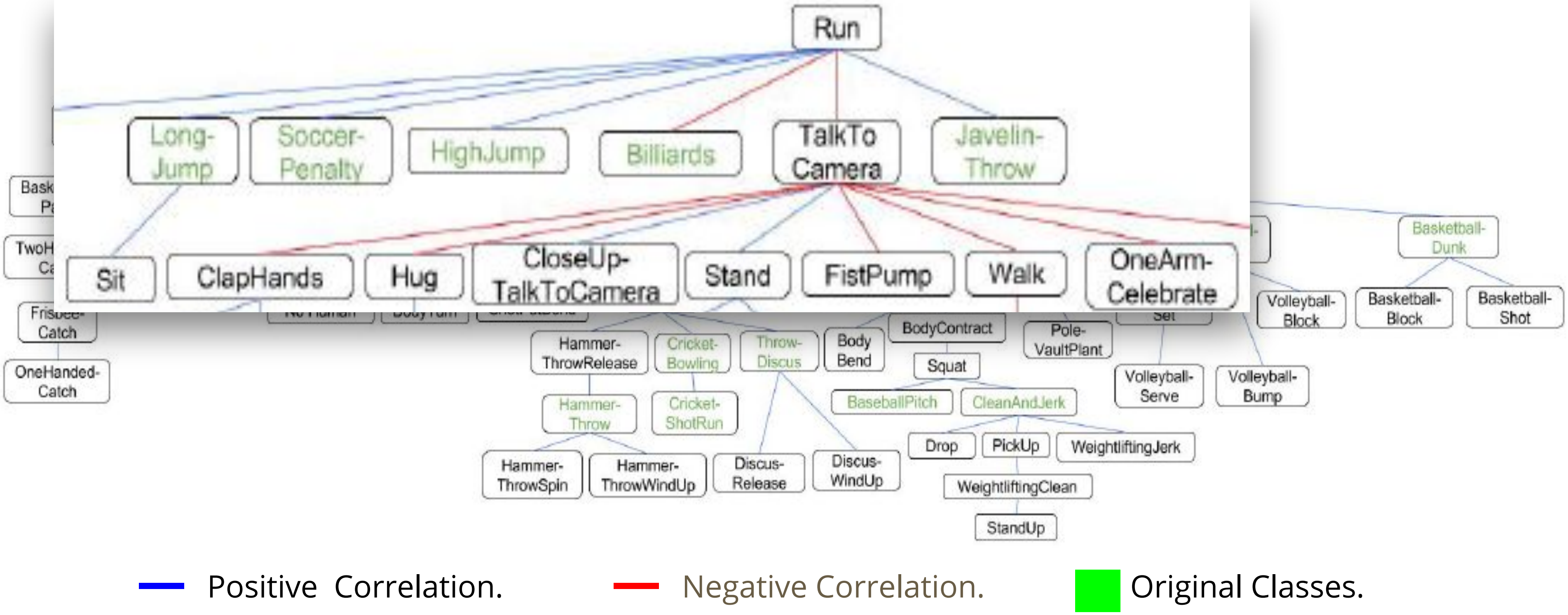
Exisiting Datasets

- **UCF101, HMDB51, Sports1M**
  - Common Challenging action recognition datasets
  - Non-localized labels, temporarily clipped around actions
- **MPII Cooking and Breakfast**
  - Long untrimmed videos with multiple sequential actions
  - Single label per frame, closed-world environment
- **THUMOS**
  - Long untrimmed videos
  - 80% videos only contain a single action class

From THUMOS to MultiTHUMOS

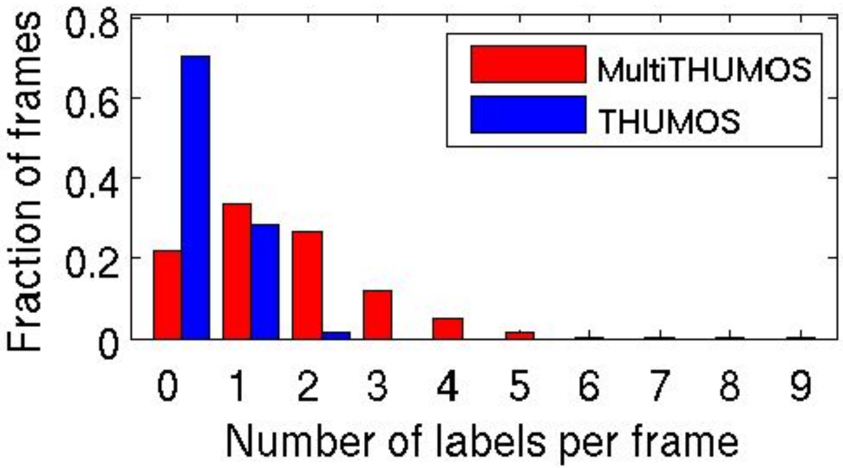
- Action Detection Dataset from THUMOS Challenge 2014
- 30 hours across 413 videos, collected from YouTube
- Classes: 20 → 65
    - Diversity of length
    - Hierarchical, hierarchical within a sport and fine-grained categories
    - Sport-specific and non-sport-specific categories
- Annotations: 6365 → 38690
    - Datatang data annotation service
    - Given the name of an action, a brief description and 2 annotation examples, one worker is asked to annotate the start and end frame of the action if it occurs for each video
    - A second worker verifies each annotation.

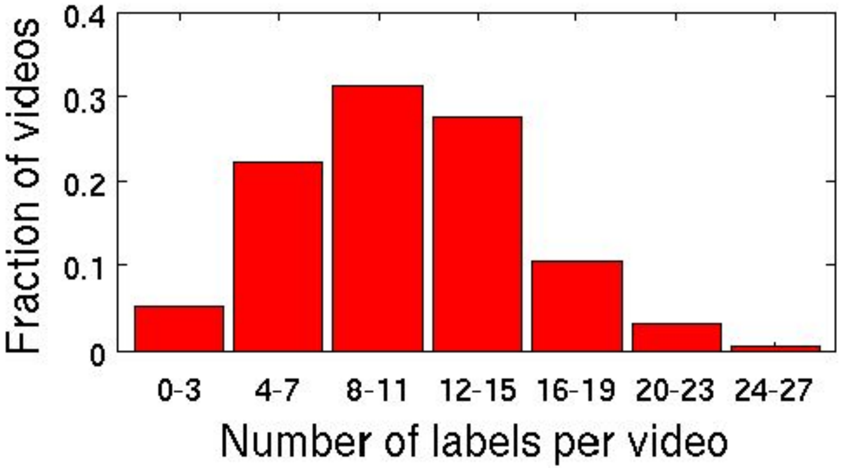
Glance at MultiTHUMOS



Co-occurrence Hierarchy of MultiTHUMOS 65 Action Classes

Comparison with THUMOS



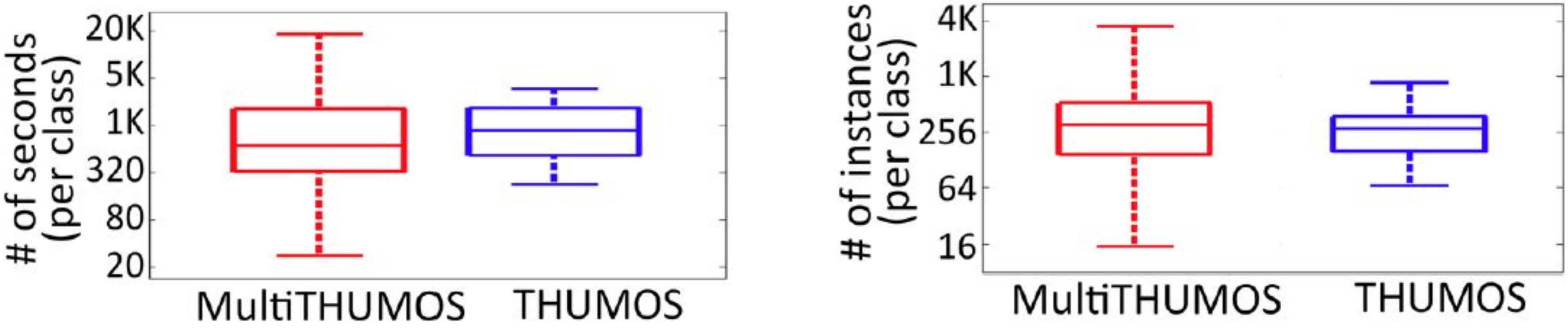


Avergae # of labels per frame: 0.3 → 1.5
Average # of action classes per video: 1.1 → 10.5

Dense interactions between actions!

MultiTHUMOS as Challenging Dataset

- Long tail data distribution
    - Amount of annotated data varies across action classes
    - Requires effective utilization of both small and large amounts of annotated data



MultiTHUMOS as Challenging Dataset

- Shorter length of actions
  - Little visual signal in the positive frames
  - Requires strong contextual modelling and multi-action reasoning

	THUMOS	MultiTHUMOS
Avg. Instance Length	4.8s	3.3s
Avg. Class Length	1.5s ~ 14.7s	--
# of Classes <1s	0	7

MultiTHUMOS as Challenging Dataset

- Fine-grained actions
  - low inter-class variation
  - Requires general action detection approaches that are able to accurately model a diverse set of visual appearances

Hierarchical: throw vs. baseball pitch

Hierarchical within a sport: pole vault vs. plant the pole when pole vaulting

Fine-grained: basketball dunk, shot, dribble, guard, block, and pass

Sport-specific actions: different basketball or volleyball moves

General actions: pump fist, or one-handed catch

MultiTHUMOS as Challenging Dataset

- High intra-class variation
  - Visual difference for the same action across frames
  - Requires insensifitivty to camera viewpoint and accurately focus on semantic information



Road Map

Target problem of **dense detailed multi-label action understanding**.

(1) Finding the right dataset

(2) Developing an appropriate model

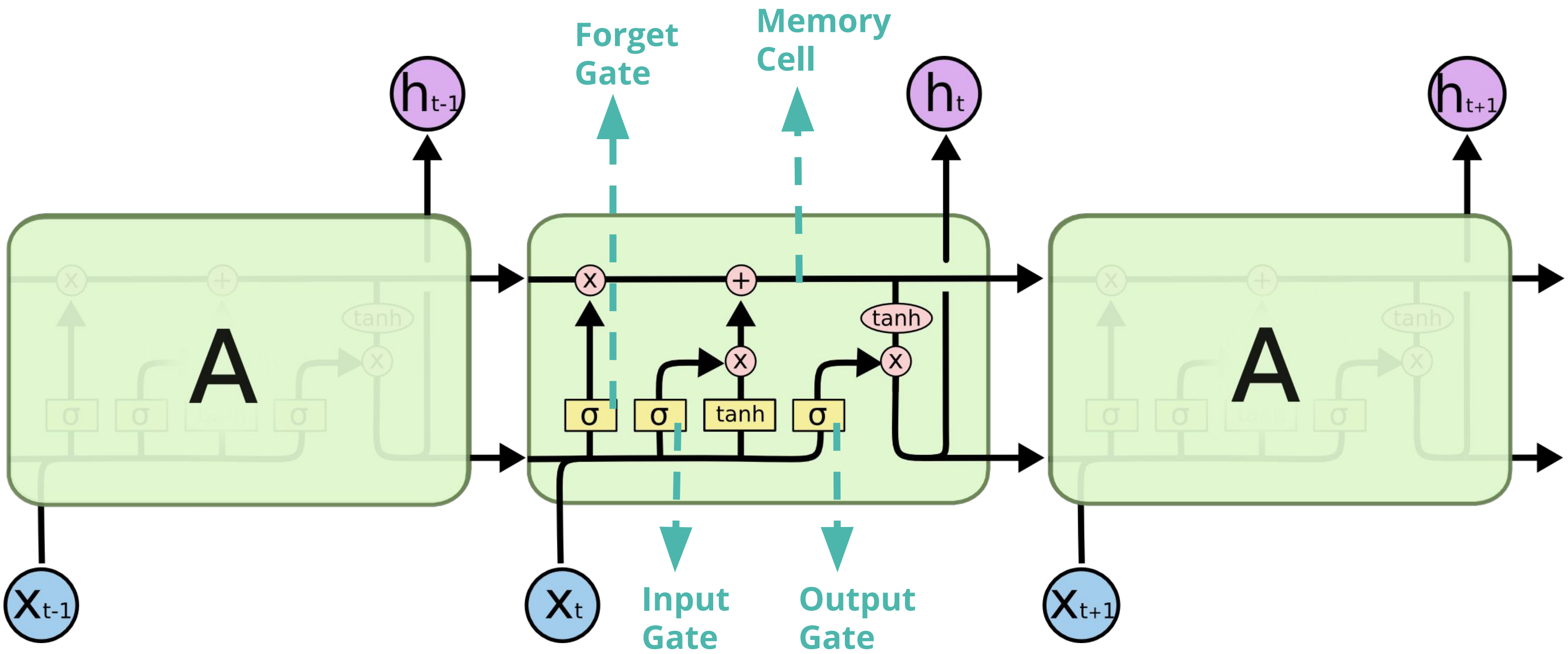
Road Map

Target problem of **dense detailed multi-label action understanding**.

(1) Finding the right dataset

(2) Developing an appropriate model

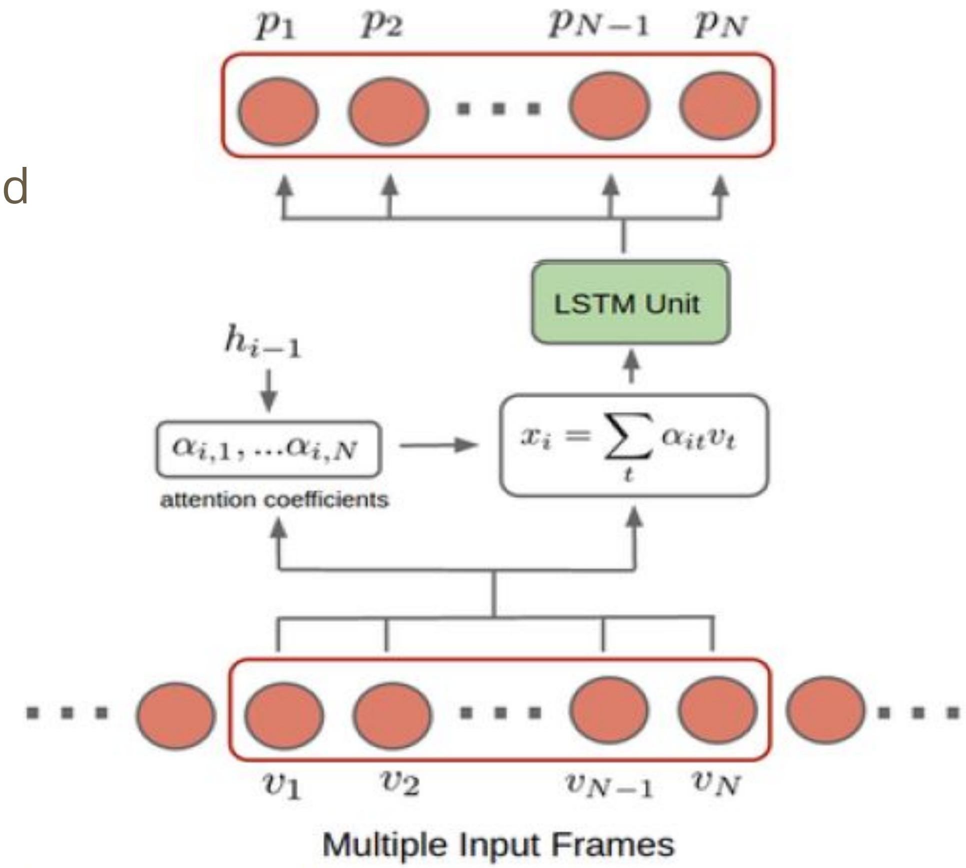
LSTM


Forget Gate
Memory Cell
Input Gate
Output Gate
h_{t-1}
h_t
h_{t+1}
A
A
tanh
tanh
σ
X_{t-1}
X_t
X_{t+1}

MultiLSTM

Idea: expand temporal receptive field of input and output connections of LSTM

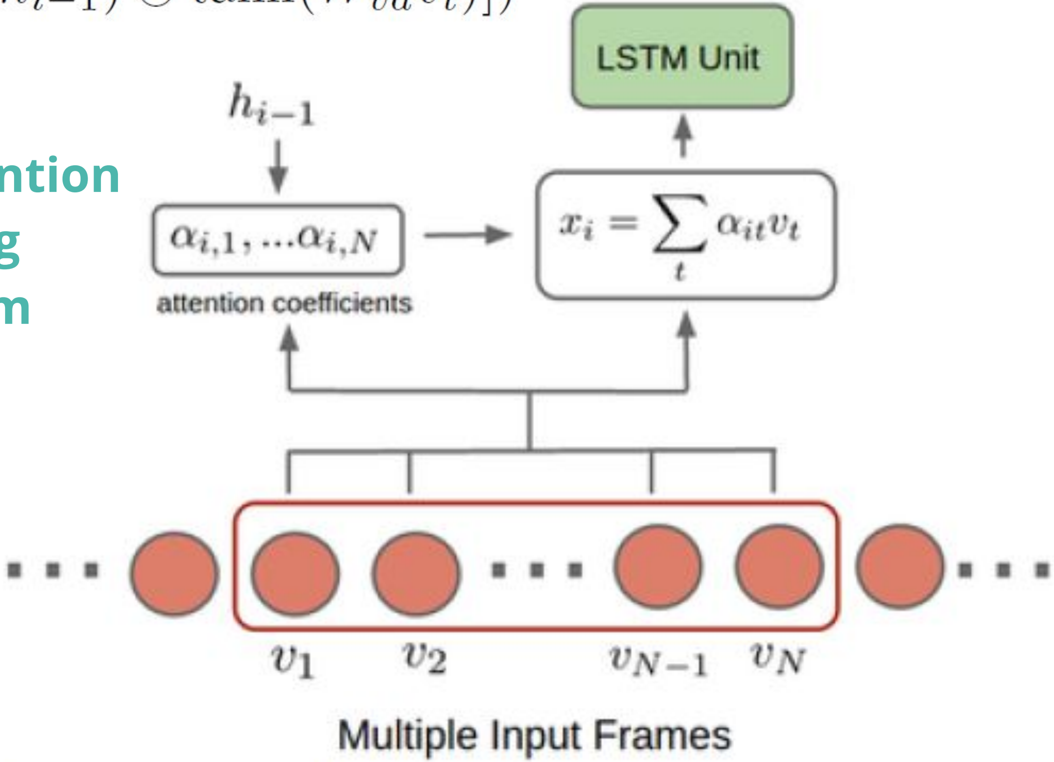
- Direct pathway for referencing previous input frames
- Direct refinement to previous predictions in retrospect after seeing more frames



$$\alpha_{it} \propto \exp(w_{ae}^T [\tanh(W_{ha}h_{i-1}) \odot \tanh(W_{va}v_t)])$$

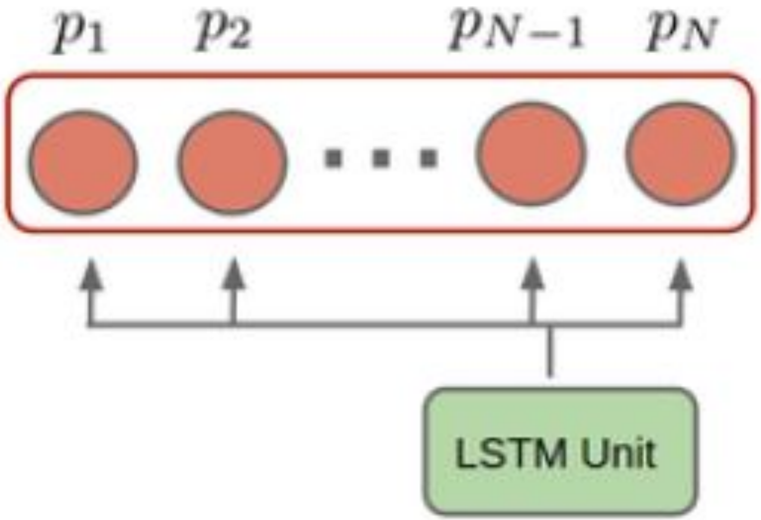
Soft-attention weighting mechnism

Video frames



A window of N frames previous to current time step

Output predictions for a window of N frames previous to current time step



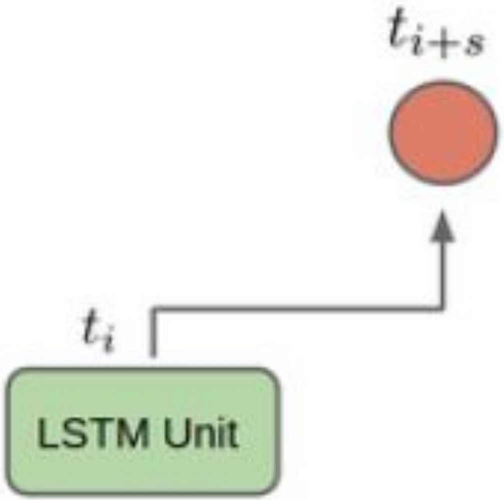
$$y_t = \sum_i \beta_{it} p_{it}$$

y_t Predicted labels for all classes at the t-th frame

p_{it} Predictions at the i-th time step for the t-th frame

β_{it} Weights of contributions

The final predicted label for all classes for a frame is calculated as a weight average of predictions.



Variant of output offset

Implementation Details

- 512 unit LSTM, 50 units in the attention component
- WIndow of 15 frames
- Input **x**:
  - 4096-d fc-7 features of VGG16,-pretrained on ImageNet and fine-tuned on MultiTHUMOS on an individual frame level
- Output **y**:
  - Unnormalized log probability of each action class
- Multilabel loss - sum of logsitic regression losses per class:

$$L(\mathbf{y}|\mathbf{x}) = \sum_{t,c} z_{tc} \log(\sigma(y_{tc})) + (1 - z_{tc}) \log(1 - \sigma(y_{tc}))$$

Experiments

- Dataset: MultiTHUMOS
- Action detection
  - Baseline: Single-frame CNN, LSTM
- Action prediction
  - Baseline: A model using ground-truth label distribution

Action Detection Evaluation

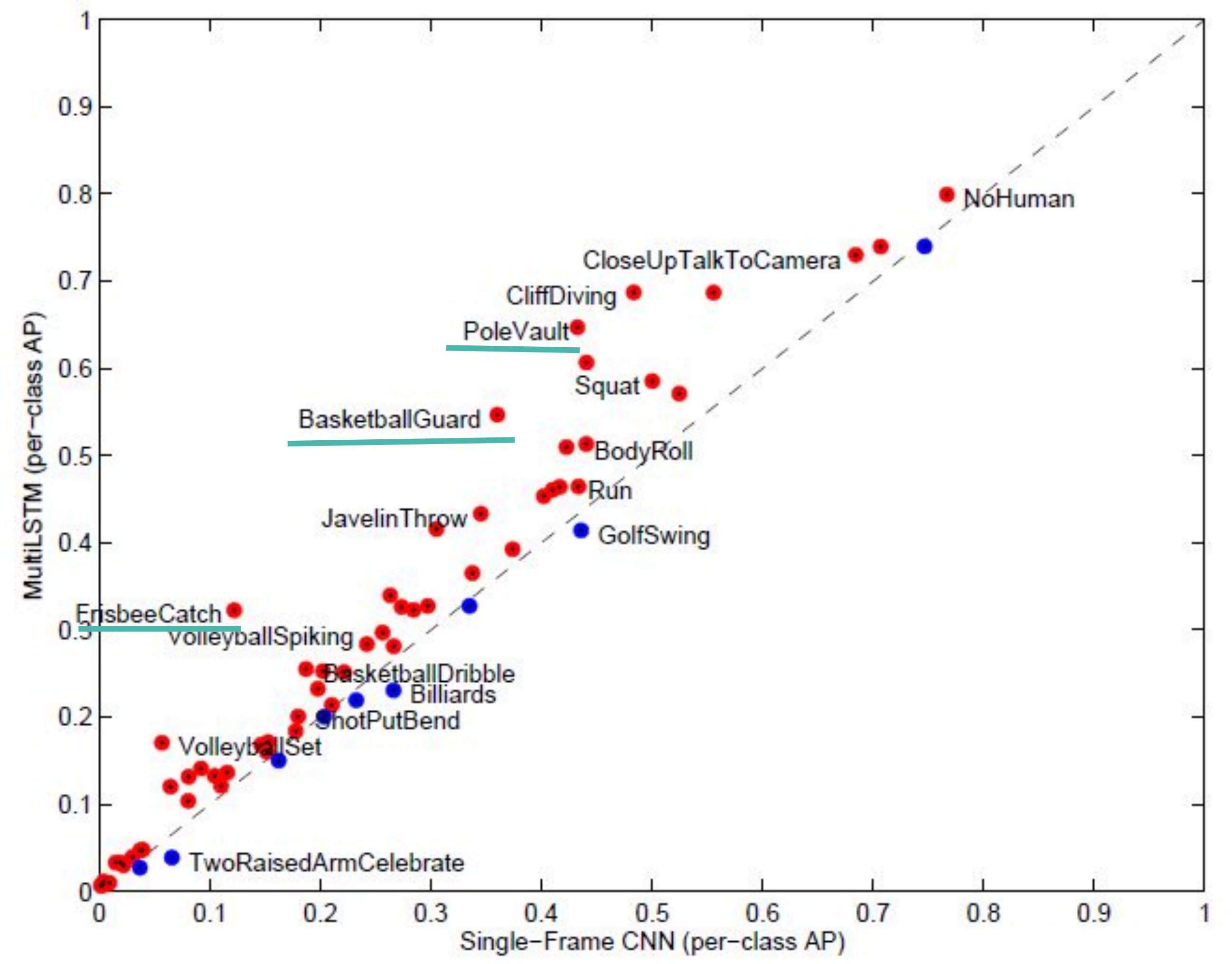
Per-frame mean Average Precision across all action classes

Model	THUMOS mAP	MultiTHUMOS mAP
IDT [46]	13.6	13.3
Single-frame CNN [36]	34.7	25.4
Two-stream CNN [35]	36.2	27.6
LSTM	39.3	28.1
LSTM + i	39.5	28.7
LSTM + i + a	39.7	29.1
MultiLSTM	**41.3**	**29.7**

Per-class AP

MultiLSTM vs. Single-Frame CNN

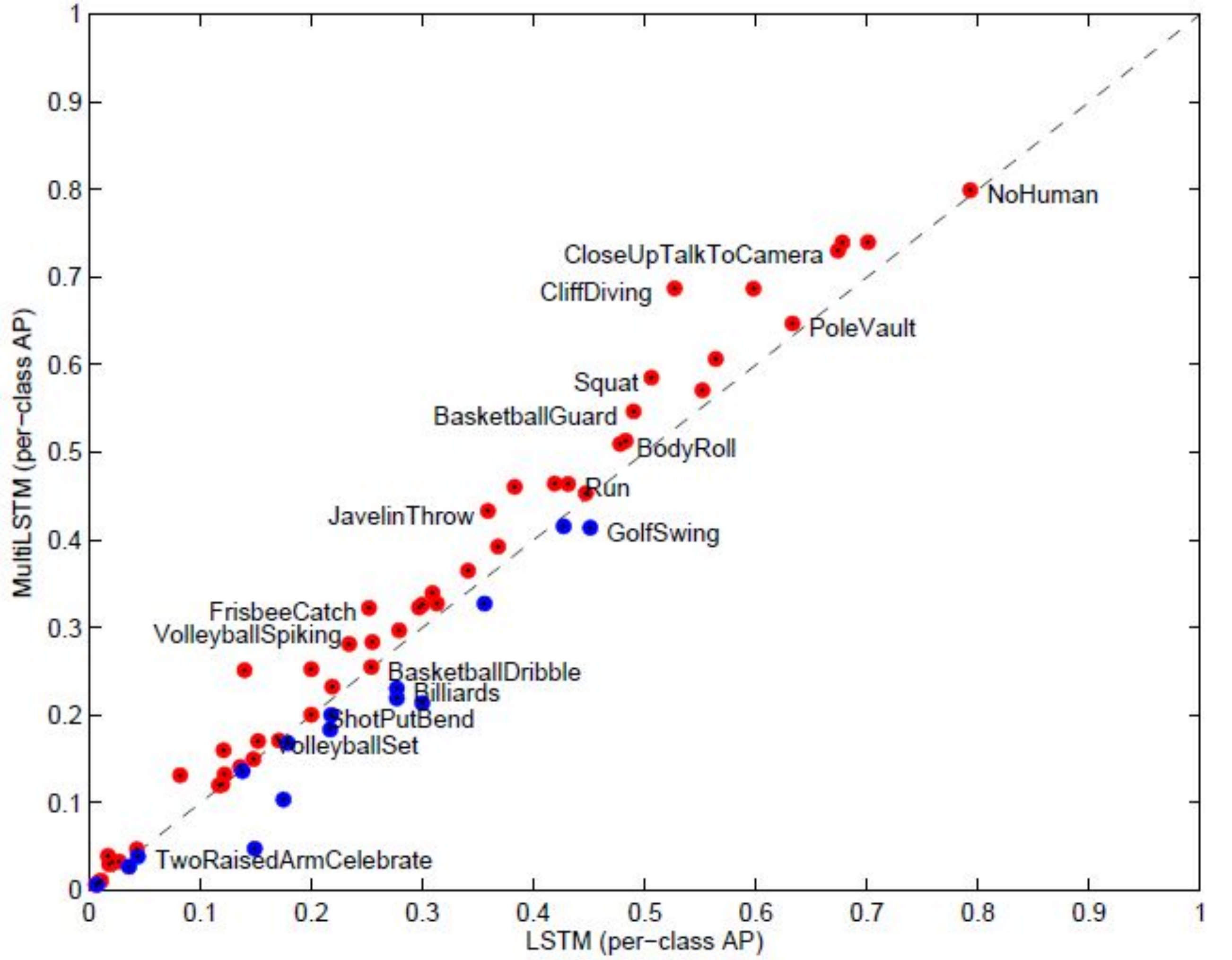
56/65



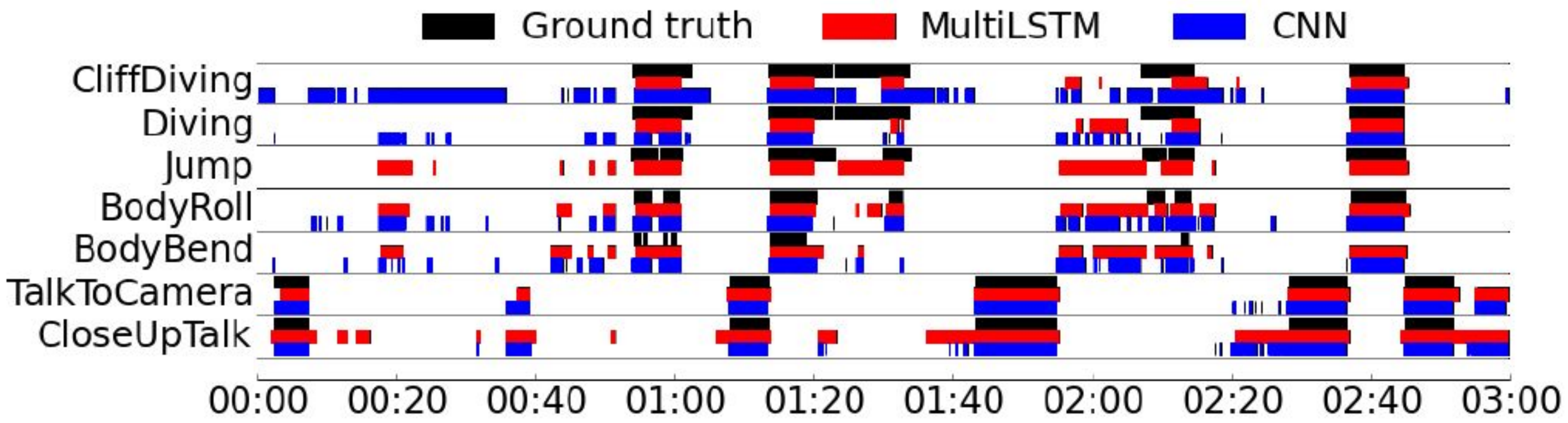
Per-class AP

MultiLSTM vs. LSTM

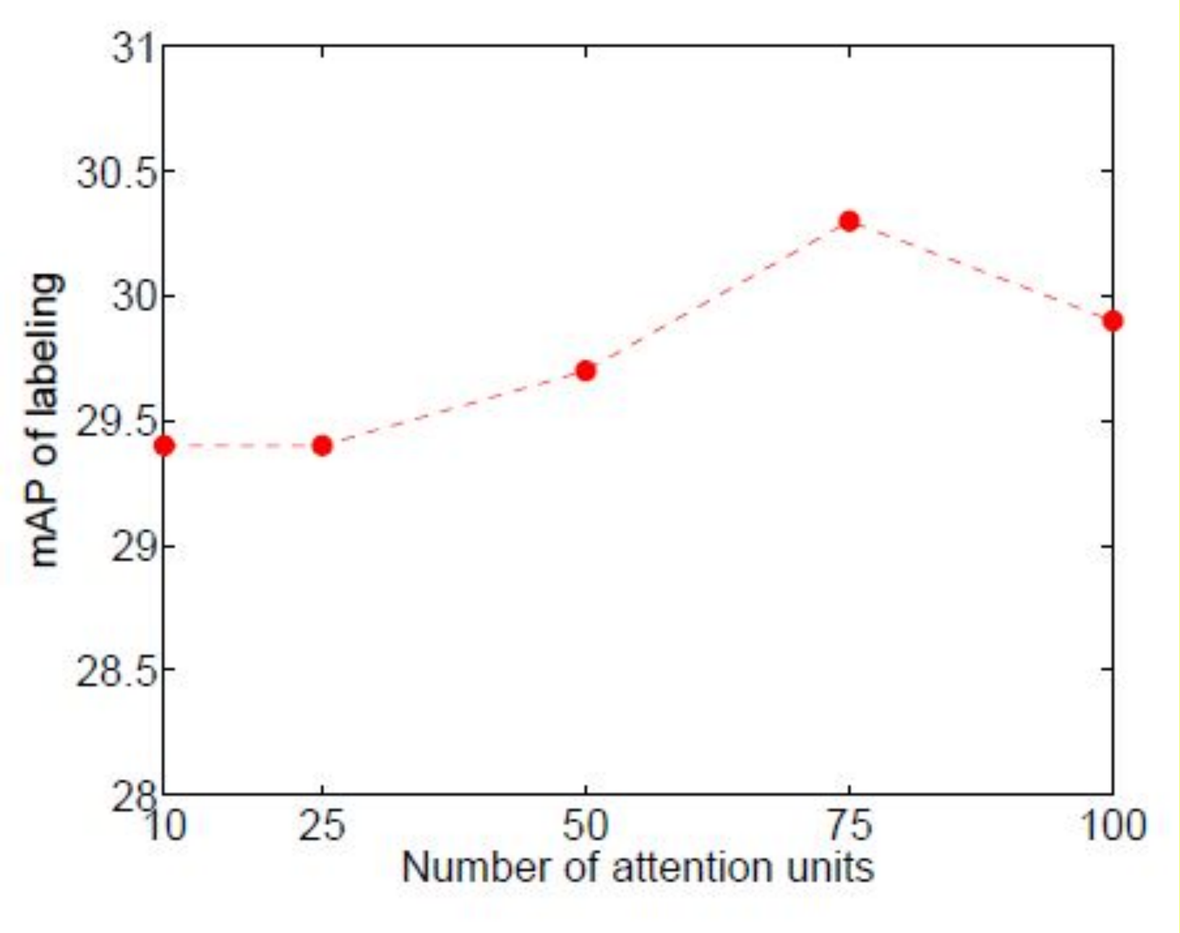
50/65



Example Action Detection Result



Number of Attention Units

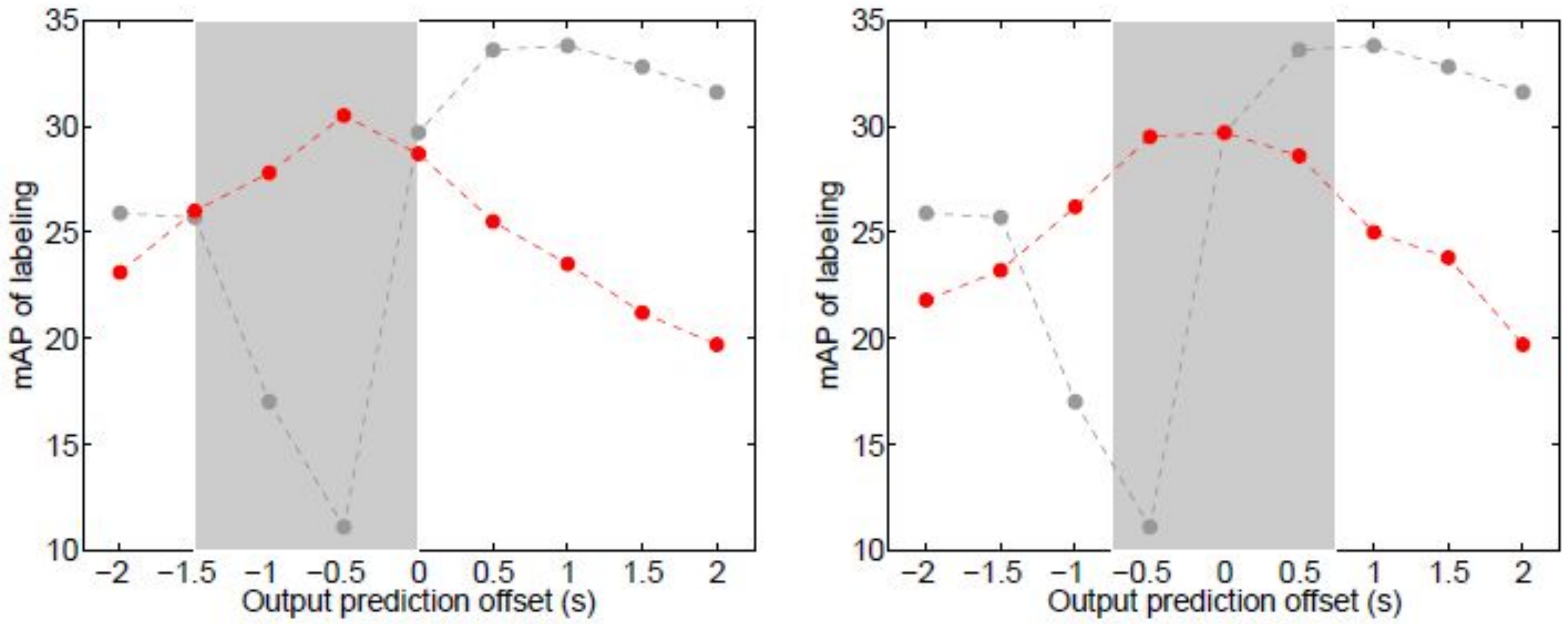


Example Video Retrieval Result

Sequential

Co-occurrence

Action Prediction Evaluation

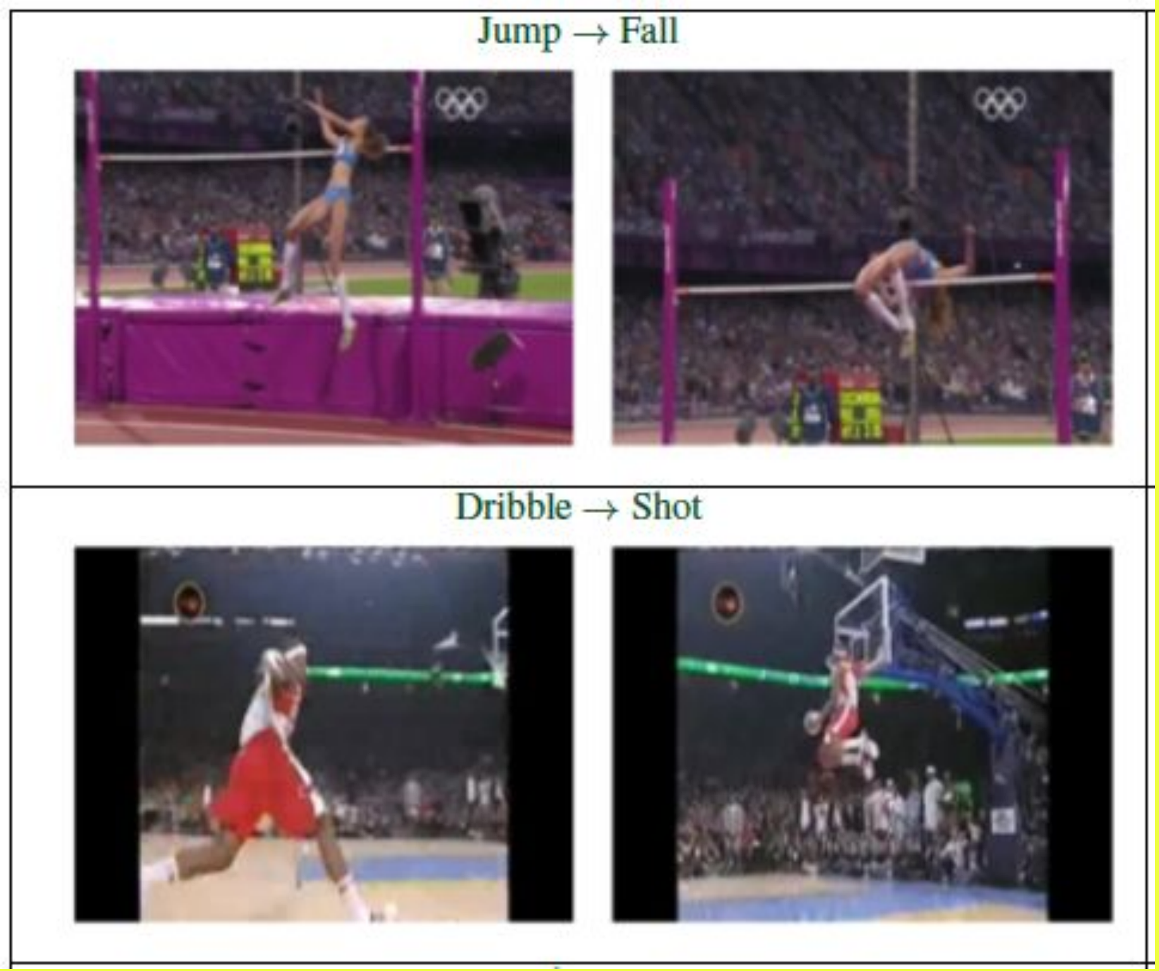


Variant of MultiLSTM

Baseline

Input time window

Example Action Prediction Result



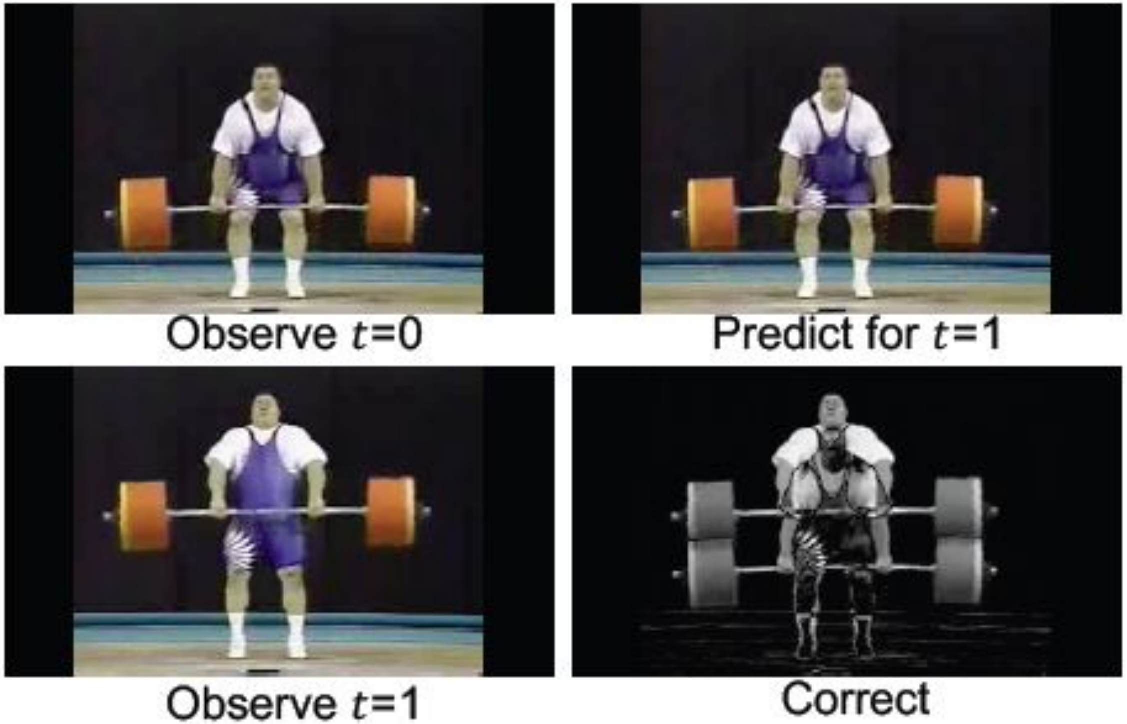
Predictive-Corrective Networks for Action Detection

Dave et al.

- **Motivation**

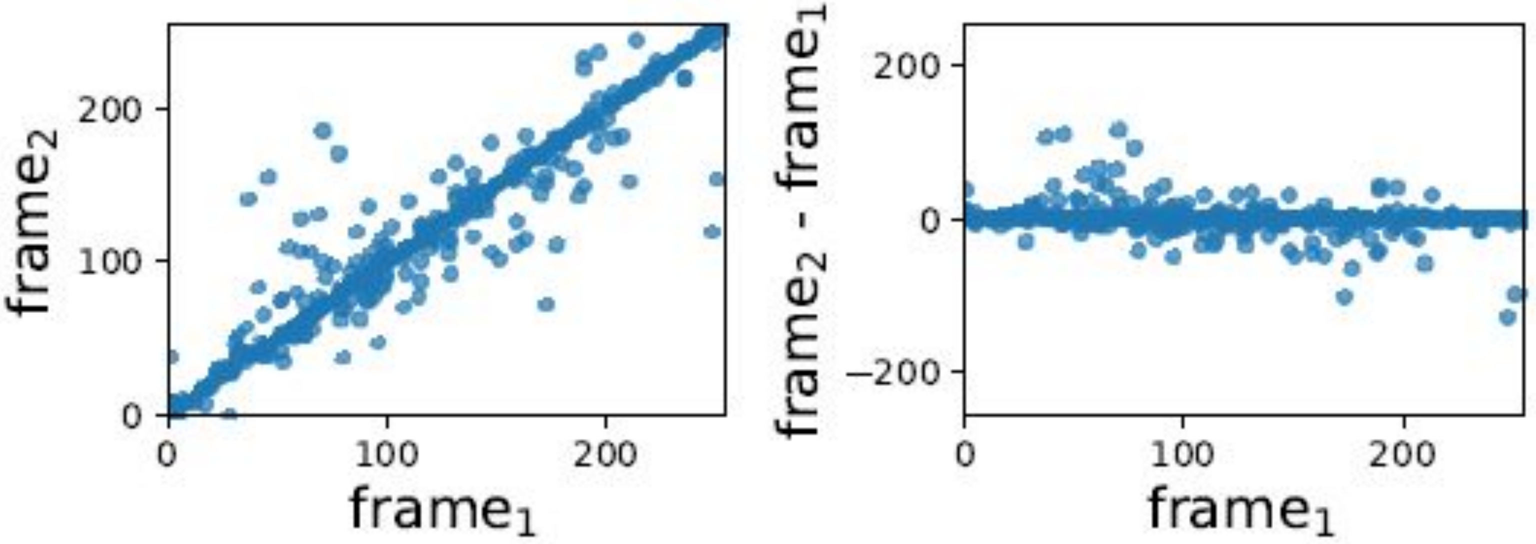
  Predict the future and correct with future observations
- **Model**
  - Predictive-Corrective Model
  - Layered Predictive-Corrective Blocks
  - Dynamic Computation
- **Experiments**

Motivation



The human vision system relies on continuously **predicting** the future and then **correcting** for the unexpected

Motivation



Reasoning frame **differences** de-correlates data

Predictive-Corrective Model

- **Idea**: Inspired by Kalman Filtering
- Suppose our images and action scores evolve smoothly, as with a linear dynamical system:

Latent State (Actions) $\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + noise$

Observation (Frames) $\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + noise$

- Can create improved estimates of action scores by:

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t-1} + \underbrace{g(\mathbf{y}_t - \underbrace{\hat{\mathbf{y}}_t}_{\text{Prediction}})}_{\text{Correction}}$$

Prediction
Correction

Predictive-Corrective Model (Cont')

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + noise$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + noise$$

$\mathbf{x}_t$ Semantic state of frame t

$\mathbf{y}_t$ Appearance of frame t

Posterior Estimate:

$$\hat{\mathbf{x}}_t = \underbrace{\hat{\mathbf{x}}_{t|t-1}}_{\text{prediction}} + \mathbf{K}\underbrace{(\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1})}_{\text{correction}}$$

$\hat{\mathbf{x}}_{t|t-1}$, $\hat{\mathbf{y}}_{t|t-1}$ Prior predictions given observations of previous frames

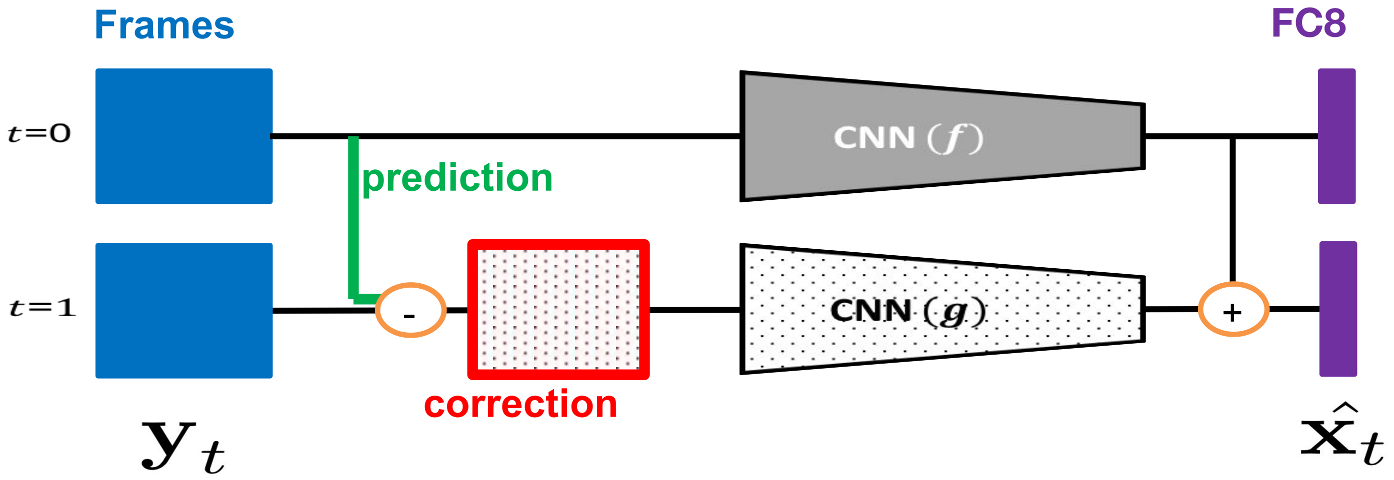
$\mathbf{K}$ Kalman gain matrix

Approximation:

$$\hat{\mathbf{x}}_{t|t-1} \approx \hat{\mathbf{x}}_{t-1} \quad \hat{\mathbf{y}}_{t|t-1} \approx \mathbf{y}_{t-1} \Rightarrow \hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t-1} + g(\mathbf{y}_t - \mathbf{y}_{t-1})$$

Actions and pixel values of a video evolve slowly over time

Predictive-Corrective Model (Cont')



$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t-1} + g(\mathbf{y}_t - \hat{\mathbf{y}}_t)$$

Prediction
Correction

[Dave, Russakovsky, Ramanan. "Predictive-Corrective Networks for Action Detection." CVPR 2017]

Layered Predictive-Corrective Blocks

- **Idea**: Combine **hierarchy** with **predictive-corrective block**
- Model lower layers as observations that are used to infer the hidden states of higher layers

At layer l = 0...L:

$$\hat{\mathbf{z}}_t^l = \hat{\mathbf{z}}_{t-1}^l + g^l(\mathbf{z}_t^{l-1} - \mathbf{z}_{t-1}^{l-1})$$

$\mathbf{z}_t^l$ latent representattion in layer l at frame t

$\hat{\mathbf{z}}_t^l$ prediction for $\mathbf{z}_t^l$

g^l learned non-linear function for layer l

Layered Predictive-Corrective Blocks (Cont')

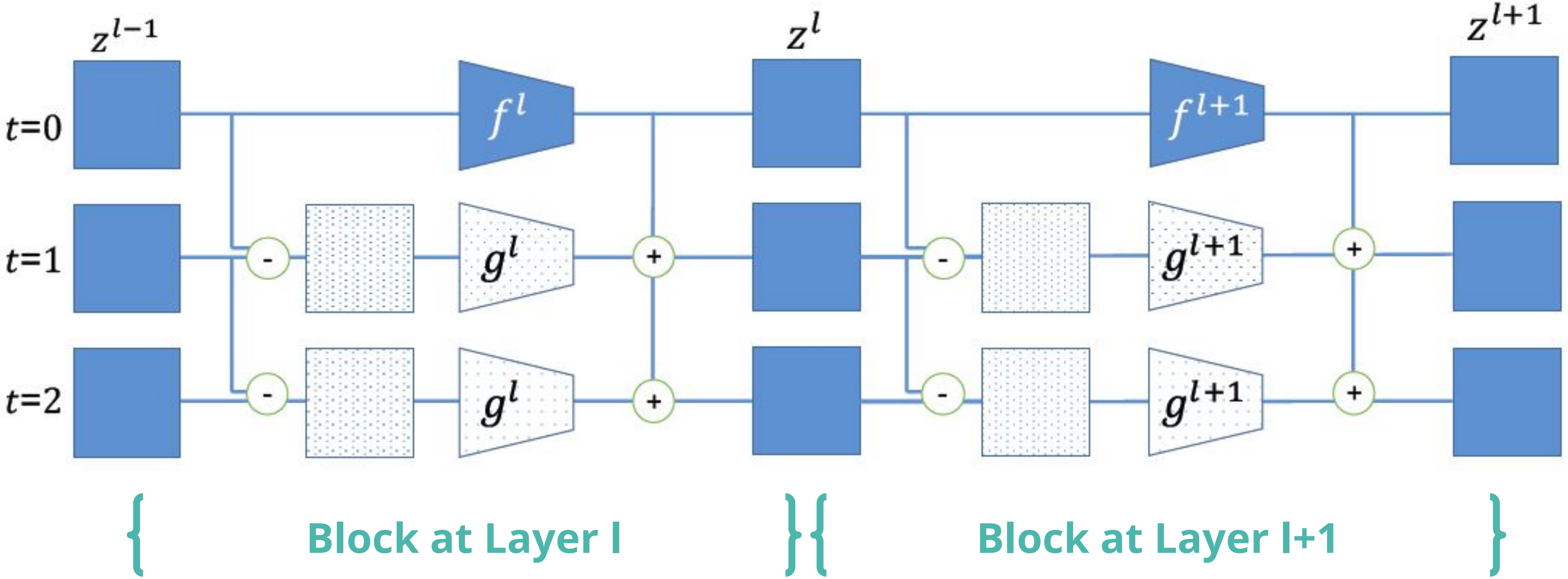
Problem: No ground truth latent state for layers except for l=0

Initialize $\mathbf{z}_t^0$ with the pixel appearance; compute $\hat{\mathbf{z}}_t^1$ and use it as observed $\mathbf{z}_t^1$ to compute $\hat{\mathbf{z}}_t^2$, continuing the layerwise recursion.

Problem: Base case of the temporal recursion at time t=0

Use a separate CNN which doesn't consider the evolution of the dynamic system: $\hat{\mathbf{z}}_0^L = f(\mathbf{z}_0^0)$

$$\hat{\mathbf{z}}_t^l = \hat{\mathbf{z}}_{t-1}^l + g^l(\mathbf{z}_t^{l-1} - \mathbf{z}_{t-1}^{l-1}) \qquad \mathbf{z}_0^l = f^l(\mathbf{z}_0^{l-1})$$



Layered Predictive-Corrective Blocks (Cont')

Problem: Efficient end-to-end training to learn the layer-specific functions

$$\Delta_t^l \triangleq \hat{\mathbf{z}}_t^l - \hat{\mathbf{z}}_{t-1}^l$$

$$\hat{\mathbf{z}}_t^l = \hat{\mathbf{z}}_{t-1}^l + g^l(\mathbf{z}_t^{l-1} - \mathbf{z}_{t-1}^{l-1})$$

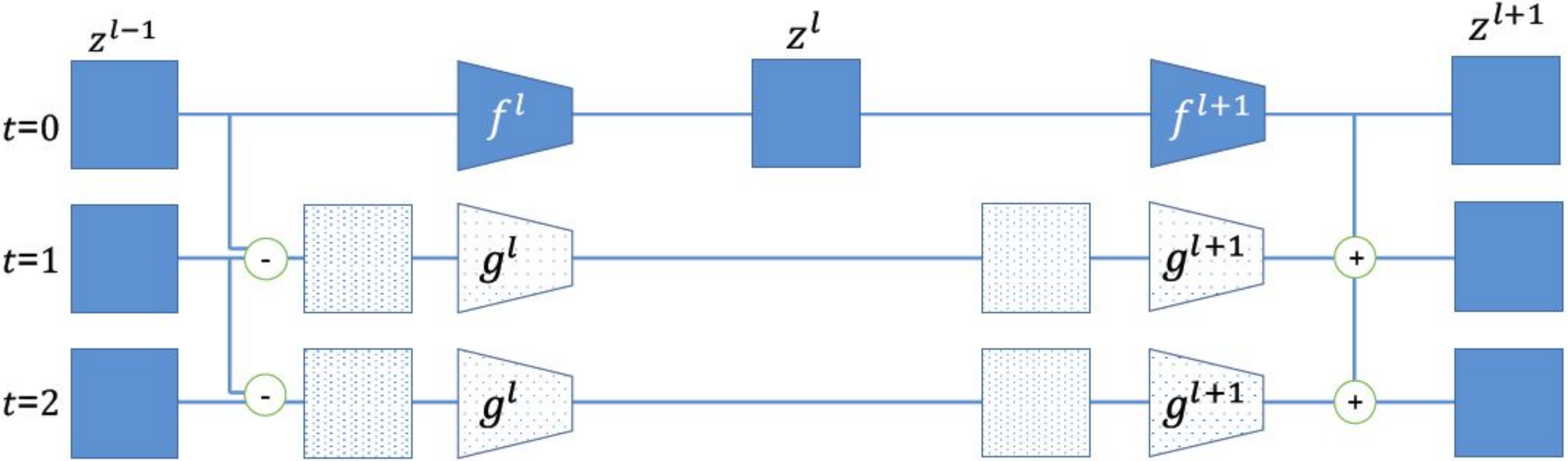
$$\Downarrow$$

$$\Delta_t^L = g^L(\Delta_t^{L-1}) = g^L(g^{L-1}(\cdots g^1(\Delta_t^0))) = g(\mathbf{z}_t^0 - \mathbf{z}_{t-1}^0)$$

Action prediction for frame t as $\hat{\mathbf{z}}_t^L = \hat{\mathbf{z}}_0^L + \sum_{i=1}^t \Delta_i^L$ with ground truth action label $\mathbf{z}_t^L$ as training signal

$$\hat{\mathbf{z}}_0^L = f(\mathbf{z}_0^0)$$

$$\Delta_t^L = g(\mathbf{z}_t^0 - \mathbf{z}_{t-1}^0) \qquad \hat{\mathbf{z}}_t^L = \hat{\mathbf{z}}_0^L + \sum_{i=1}^t \Delta_i^L$$



Collapse of blocks at Layer l & layer l+1

Dynamic Computation

- **Idea**: Adaptively focus computation on “surprising” frames
- Ignore small corrections, re-initialize on large corrections

$$\hat{\mathbf{z}}_t^l = \begin{cases} \hat{\mathbf{z}}_{t-1}^l & \text{if static activations} \\ f^l(\hat{\mathbf{z}}_t^{l-1}) & \text{if re-initializes} \\ \hat{\mathbf{z}}_{t-1}^l + g^l(\mathbf{z}_t^{l-1} - \mathbf{z}_{t-1}^{l-1}) & \text{else} \end{cases}$$

Connections to Prior Art

- Non-linear Kalman Filter
  - Linear dynamics (identity mapping)
  - Nonlinear hierarchical observation model
- RNN
  - Use past output $\mathbf{x}_{t-1}$ in a linear fashion
  - Maintain the previous input $\mathbf{y}_{t-1}$ as part of memory

$$\mathbf{x}_t = \sigma(W\mathbf{y}_t + V\mathbf{x}_{t-1})$$

$$\mathbf{x}_t = \mathbf{x}_{t-1} + \sigma(W(\mathbf{y}_t - \mathbf{y}_{t-1}))$$

Implementation Details

- VGG16 architecture for initial and update models
- Weight initialization:
    - Pre-trained on ILSVRC 2016
    - Fine-tuned on per-frame acion classification task for al actions
- Input
    - Frames extracted from the video at 10 frames per second
    - Resized to 256 x 256, and random cropped to 224 x 224

Experiments

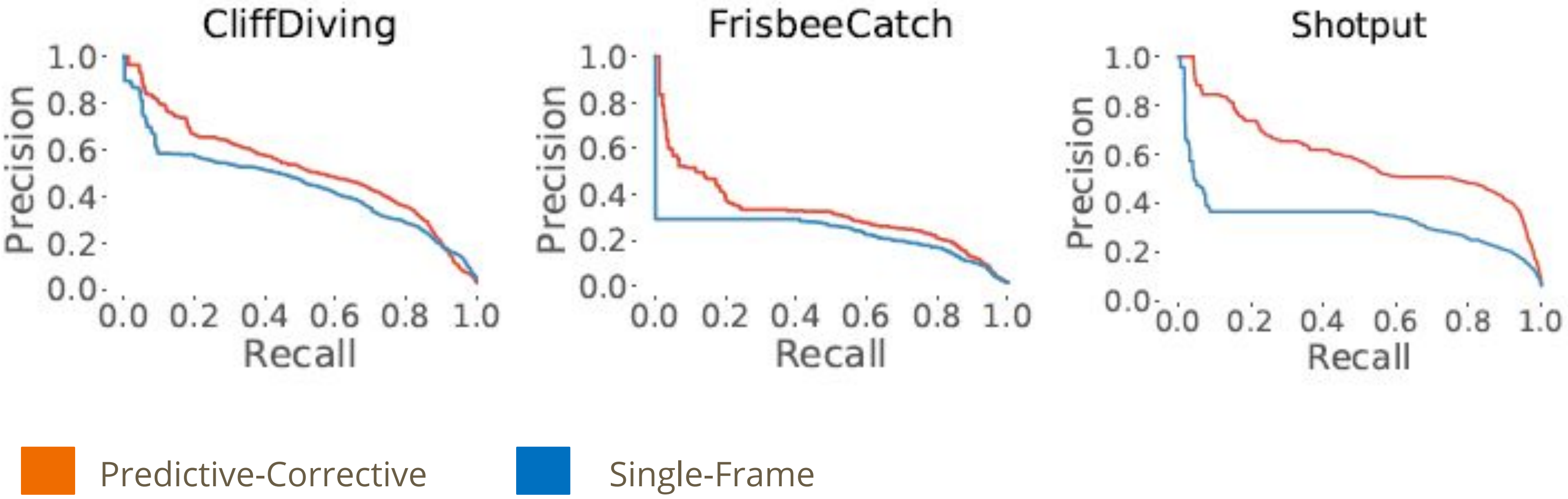
- Model Analysis
  - Comparison with baseline
  - Test-time reinitialization
  - Architectural variations
- Evaluation
  - Benchmarks: THUMOS. MultiTHUMOS and Charades

Model Analysis: Baseline

(Per-frame classification (mAP) on MultiTHUMOS)

Method	MultiTHUMOS mAP
Single-frame RGB	25.1
4-frame late fusion	25.3
Predictive-corrective (our)	**26.9**

Model Analysis: Baseline



Per-frame Precision/Recall on MultiTHUMOS

Model Analysis: Test-time Reinitialization

(Per-frame classification (mAP) on MultiTHUMOS)

- Static reinitialization:

Reinit	Train Reinit 4	Train Reinit 8
Test Reinit 2	26.9	25.9
Test Reinit 4	26.9	26.9
Test Reinit 8	25.4	**27.3**
Test Reinit 16	20.0	25.9

- Dynamic reinitialization by thresholding (at least every 4th frame):
  - 27.2% mAP
- Dynamic discard by thresholding:
  - 26.7% mAP by discarding nearly 50% frames

Model Analysis: Architectural Variations

(Per-frame classification (mAP) on MultiTHUMOS)

Configuration	mAP
conv53 every 4	26.5
fc7 every 4	26.9
fc8 every 4	26.6
conv33 every 1, fc7 every 4	**27.2**
conv43 every 2, conv53 every 4	26.6
conv53 every 2, fc7 every 4	24.8

Evaluation

Per-frame classification (mAP)

Method	MultiTHUMOS	THUMOS
Single-frame [55]	25.4	34.7
Two-Stream[3] [38]	27.6	36.2
Multi-LSTM [55]	**29.7**	**41.3**
Predictive-corrective	**29.7**	38.9

Evaluation

Per-frame classification (mAP)

Method	Charades
Single-frame	7.9
LSTM (on RGB)	7.7
Two-Stream [35]	**8.9**
Predictive-corrective	**8.9**

Thoughts

- Simple extension to exisiting models
- Help recurrent model by direct pathway to neighboring frames
- A dataset right for the task is as important as the approach
- Reduce trouble of correlation of video frames by only focusing on changes
- Difficult task, still a lot of unutiliized spatio-temporal information