

Temporal Action Proposal Generation and Detection in Videos

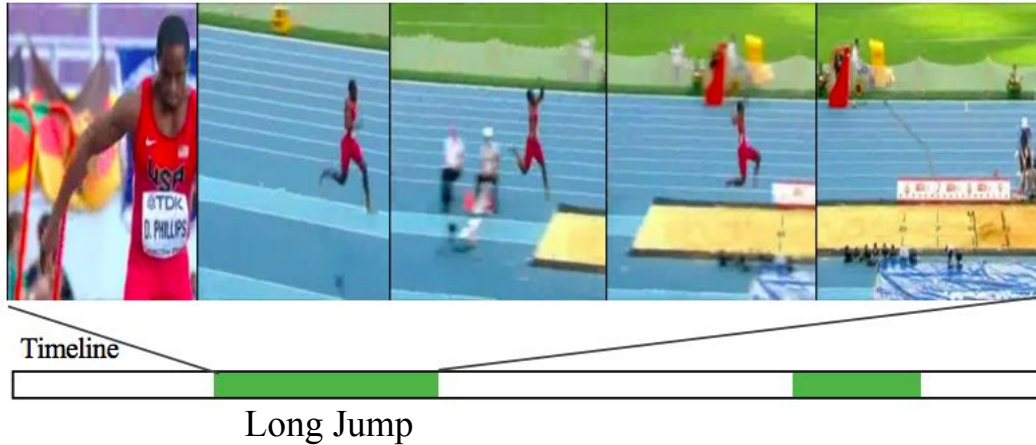
Jiyang Gao

University of Southern California

7/14/2018

Problem Statement

Temporal action detection: Given a long video, the task of temporal action detection is to localize intervals where actions of interest take place and also predict the action categories.



Outline

- Proposal-based methods
- Brief Introduction on frame-based methods
- How to combine?
- Online action detection
- Beyond the fixed activities

TURN TAP: Temporal Unit Regression Network for Temporal Action Proposals

Jiyang Gao^{1*}, Zhenheng Yang^{1*}, Chen Sun², Kan Chen¹, Ram Nevatia¹

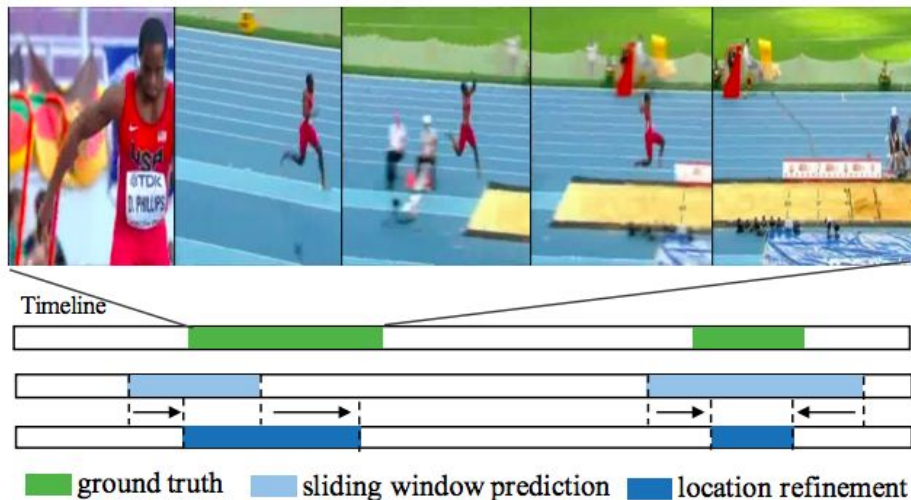
¹University of Southern California

²Google Research

<https://github.com/jiyanggao/TURN-TAP>

Problem Formulation

Generating Temporal Action Proposals (TAP) in long untrimmed videos, akin to generation of object proposals in images for rapid object detection.

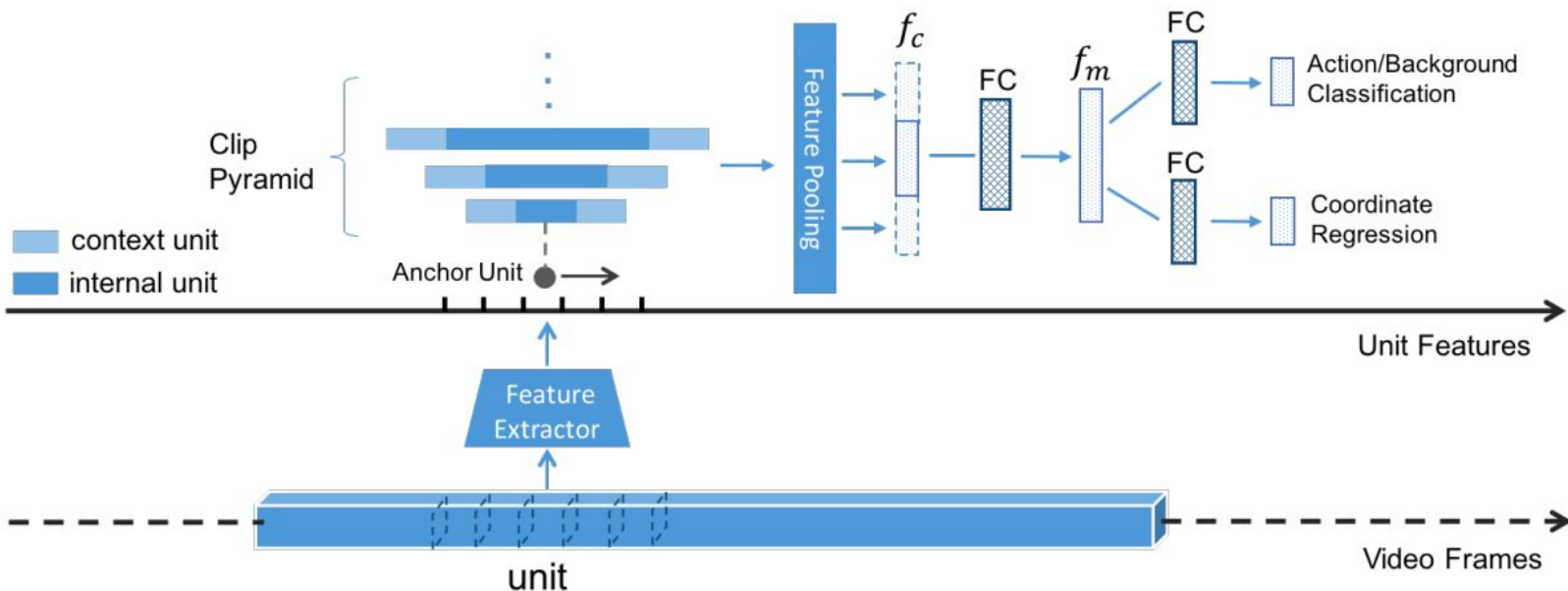


Motivation

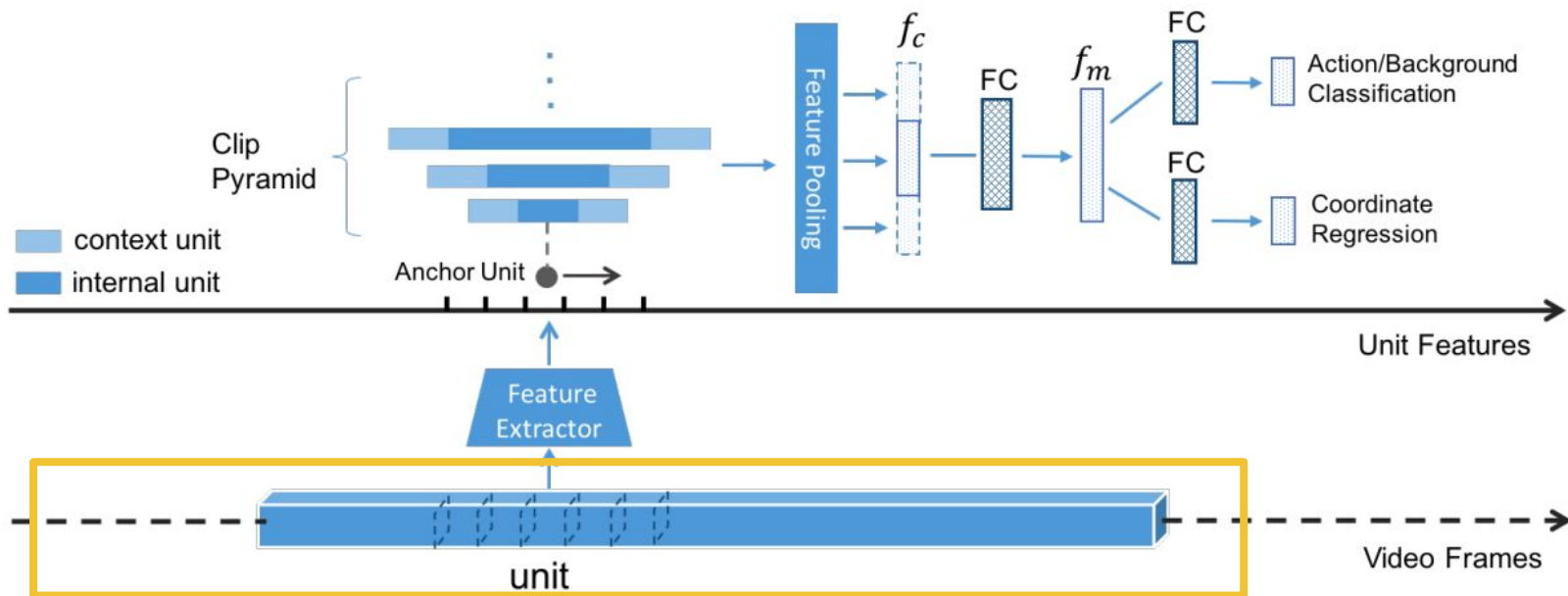
Previous work formulate TAP generation as a binary classification problem (i.e. action vs. background) and apply sliding window approach. Denser sliding windows leads to higher recall at the cost of computation time.

Our Approach: (1) jointly predict action proposals and refines the temporal boundaries by temporal coordinate regression; (2) Enable fast computation by unit feature reuse: a long untrimmed video is decomposed into video units.

Temporal Unit Regression Network (TURN) Architecture

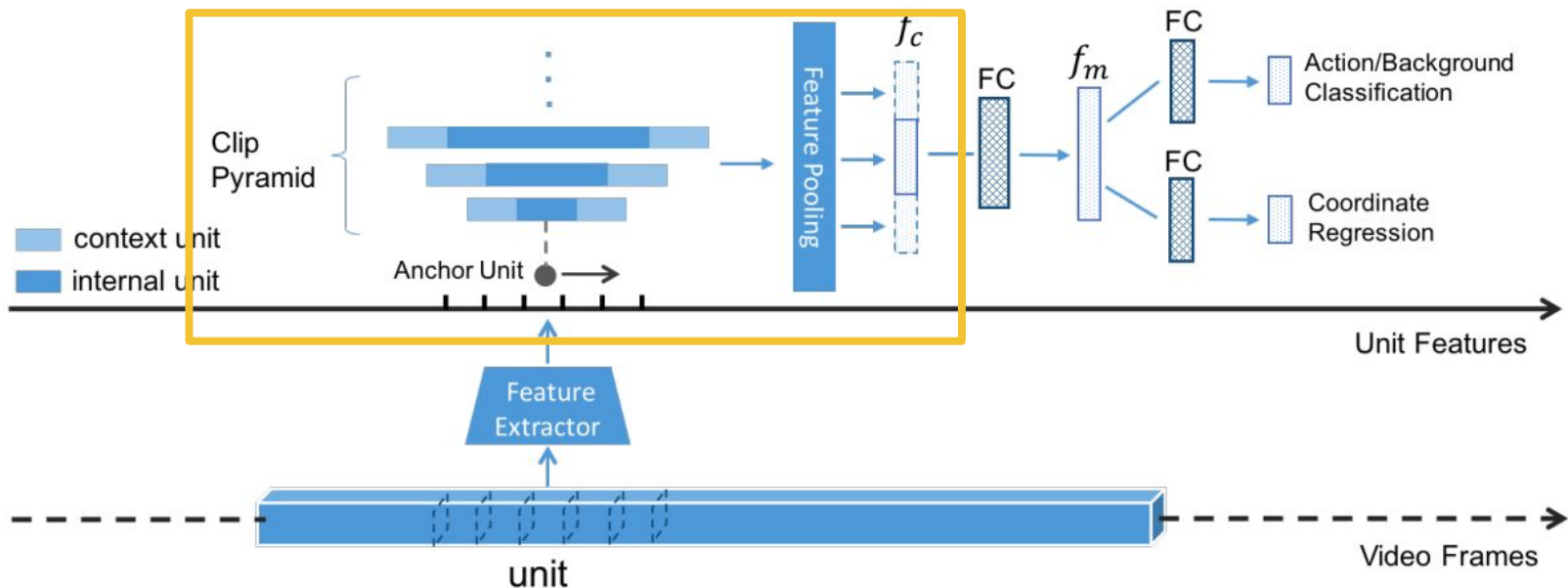


TURN Architecture



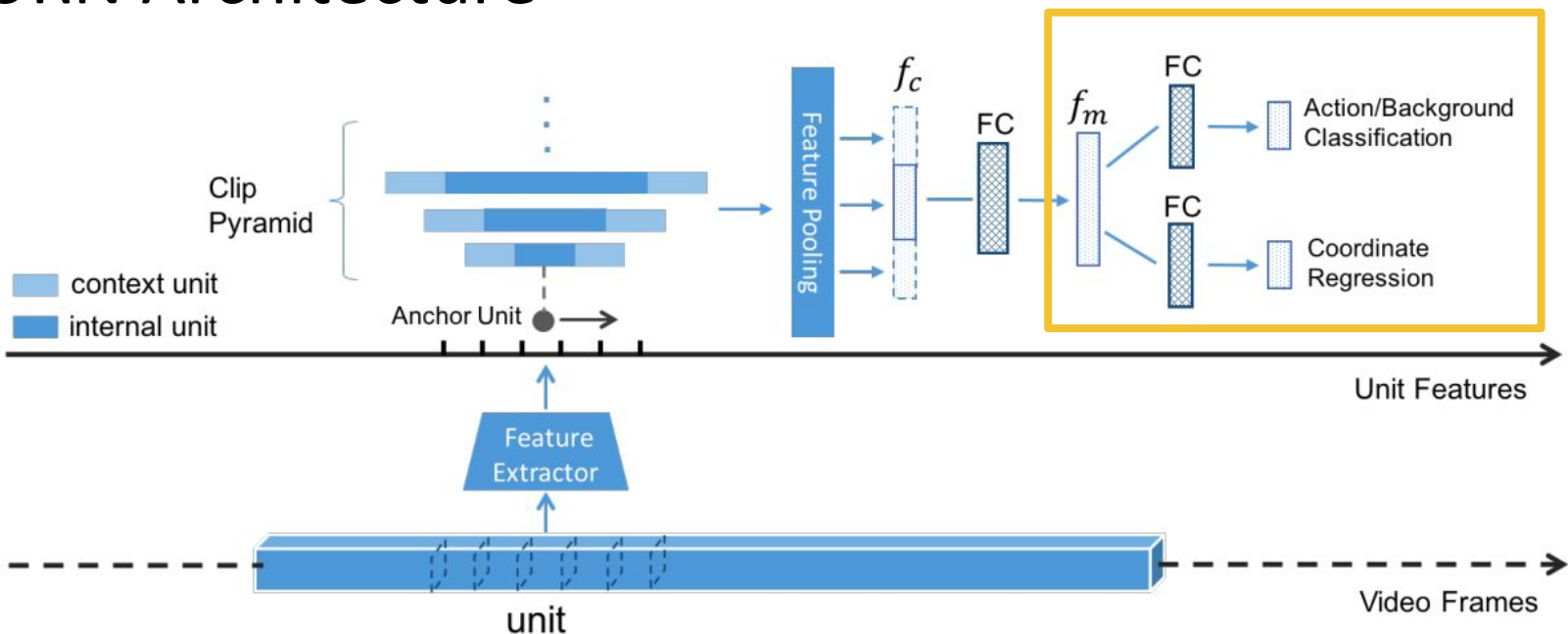
Video Unit Processing: Decompose a long video into short (e.g. 16 or 32 frames) video units. For each unit, we extract unit-level visual features using C3D or flow CNN model.

TURN Architecture



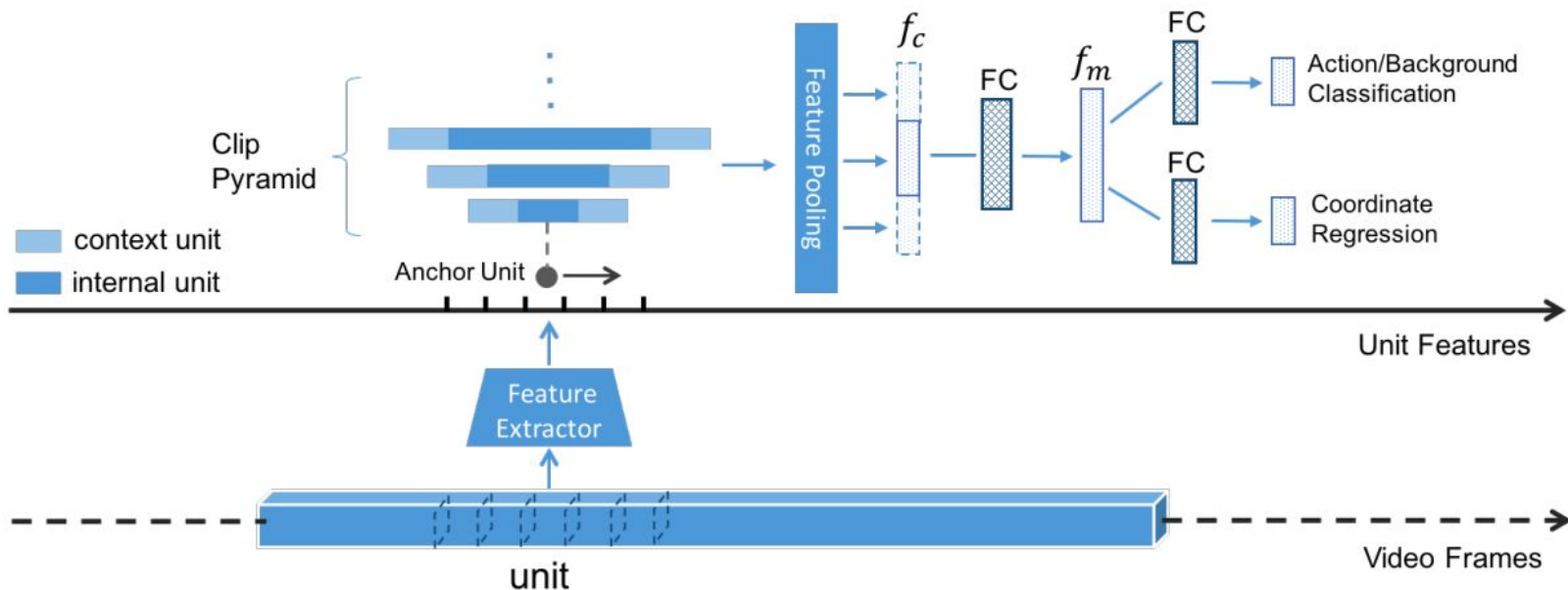
Clip Pyramid Modeling: Pool features from a set of contiguous units into a clip. Multiple temporal scales are used to create a clip pyramid. Pre-context and post-context are considered.

TURN Architecture



Unit-level Temporal Coordinate Regression: Two sibling output layers, one generates a confidence score indicating whether the input clip is an action instance, the other outputs temporal coordinate regression offsets.

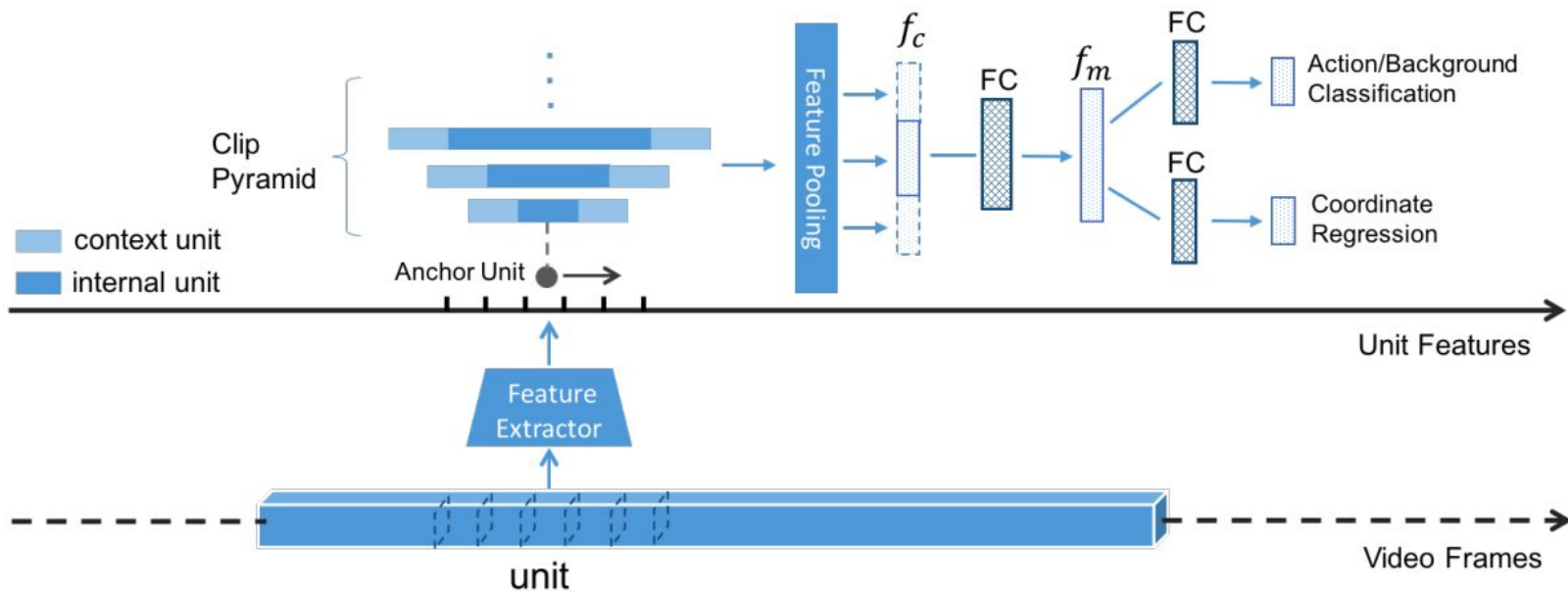
TURN Architecture



Non-parameterization offsets: the offsets of the starting unit coordinates and the ending unit coordinates.

$$o_s = s_{clip} - s_{gt}, \quad o_e = e_{clip} - e_{gt}$$

Training



Loss Functions:

$$L_{reg} = \frac{1}{N_{pos}} \sum_{i=1}^N l_i^* |(o_{s,i} - o_{s,i}^*) + (o_{e,i} - o_{e,i}^*)| \quad L = L_{cls} + \lambda L_{reg}$$

Evaluation

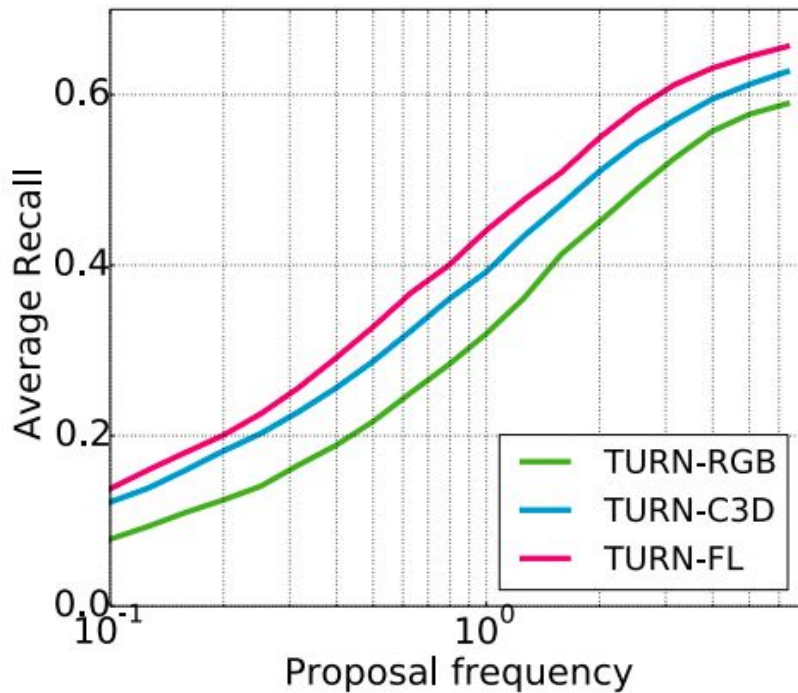
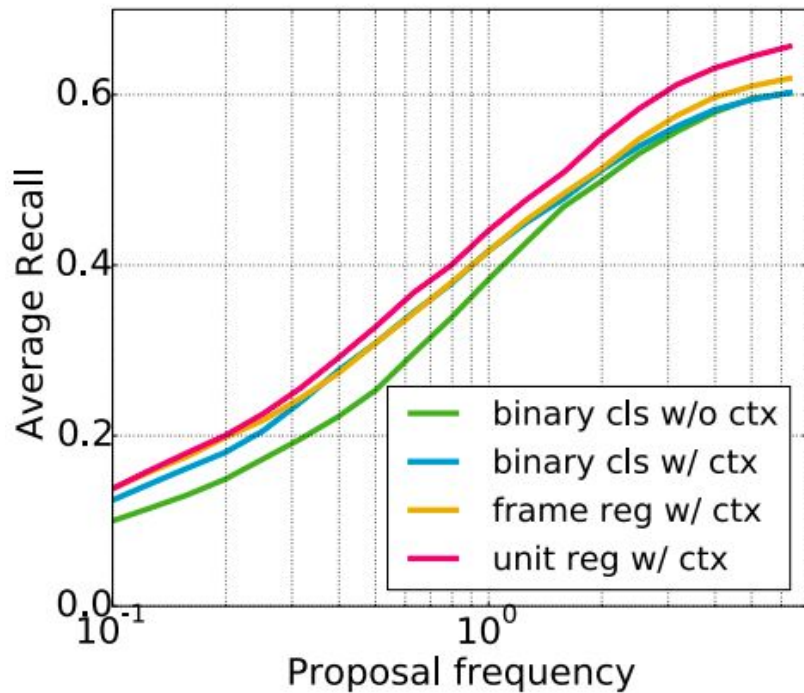
Datasets: [THUMOS-14](#). Over 20 hours of videos from 20 sports classes, 200 videos for training and 213 for test.

Metric: [AR-F curve](#), average recall as a function of proposal frequency (F), which denotes the number of retrieved proposals per second for a video. [mAP@IoU](#): mean average precision at certain intersection over union.

Clip-level feature extractor: C3D, Two-stream(Flow)

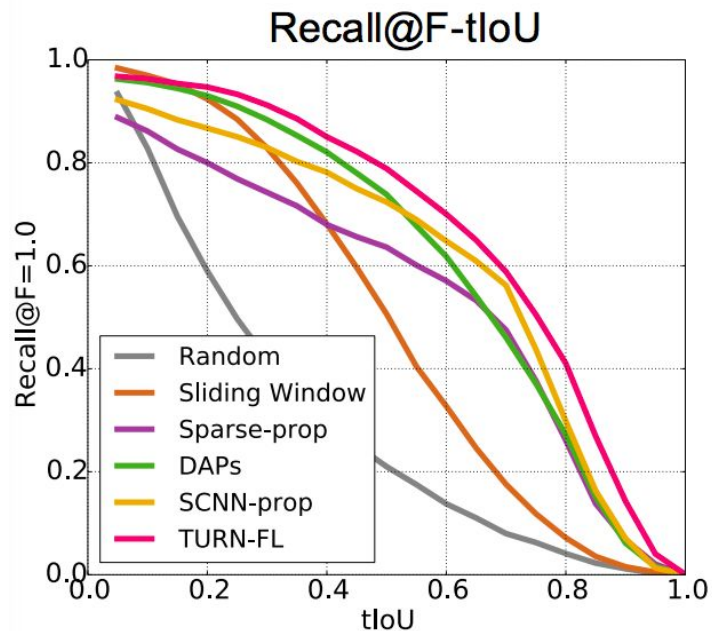
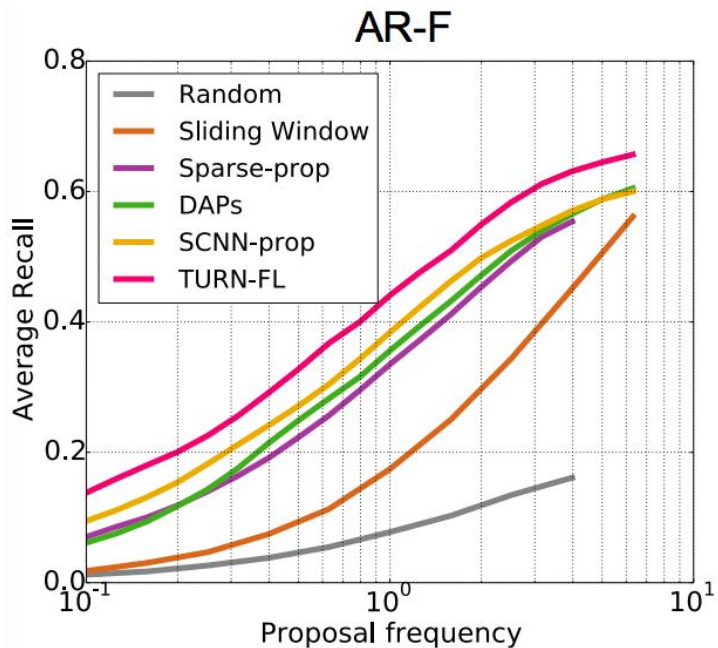
Evaluation

Comparison of different features, and on context and boundary regression.



Evaluation

Comparison with the state of the art methods.



Evaluation

How unit size affects AR and run-time performance?

Table 1. Run-time and AR Comparison on THUMOS-14.

method	AR@F=1.0 (%)	FPS
DAPs [4]	35.7	134.3
SCNN-prop [23]	38.3	60.0
Sparse-prop [2]	33.3	10.2
TURN-FL-16	43.5	129.4
TURN-FL-32	42.4	260.6
TURN-C3D-16	39.3	880.8

Smaller unit give a better AR performance, but also improve the computational cost.

Evaluation

Comparison on temporal action detection performance.

Table 3. Temporal action localization performance (mAP %) comparison at different tIoU thresholds on THUMOS-14.

tIoU	0.1	0.2	0.3	0.4	0.5
Oneata <i>et al.</i> [19]	36.6	33.6	27.0	20.8	14.4
Yeung <i>et al.</i> [33]	48.9	44.0	36.0	26.4	17.1
Yuan <i>et al.</i> [35]	51.4	42.6	33.6	26.1	18.8
S-CNN [23]	47.7	43.5	36.3	28.7	19.0
TURN-C3D-16 + SVM	46.4	41.5	34.3	24.9	16.4
TURN-FL-16 + SVM	48.3	43.2	35.1	26.2	17.8
TURN-C3D-16 + S-CNN	48.8	45.5	40.3	31.5	22.5
TURN-FL-16 + S-CNN	54.0	50.9	44.1	34.9	25.6

Cascaded Boundary Regression for Temporal Action Detection

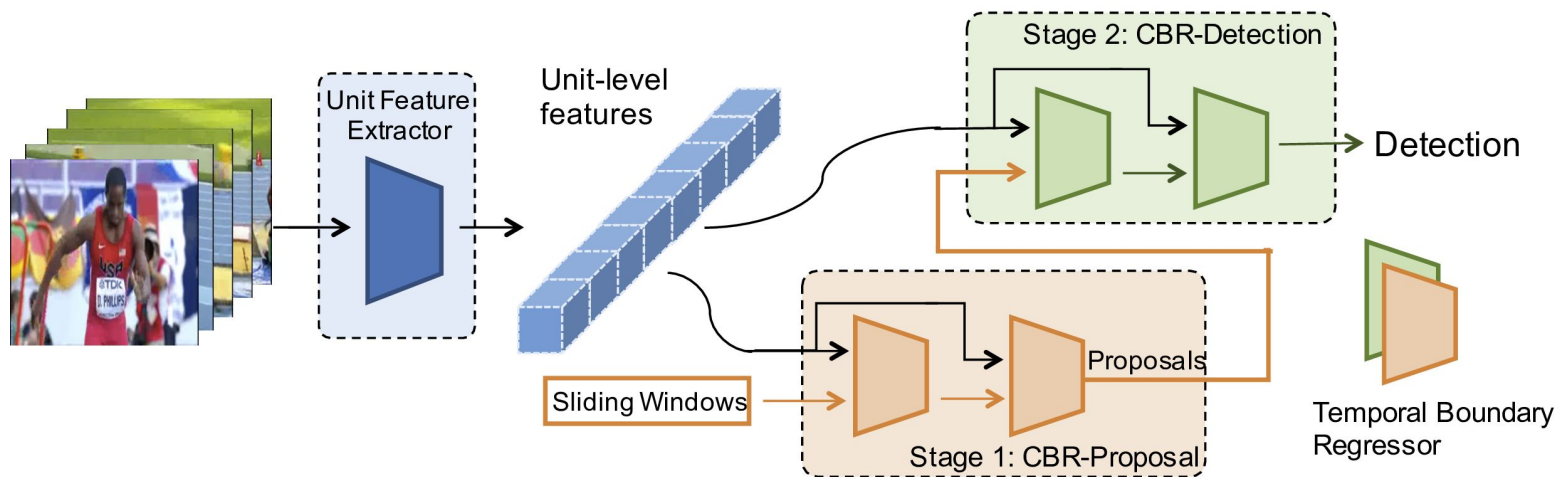
Jiyang Gao¹, Zhenheng Yang¹, Ram Nevatia¹

¹University of Southern California

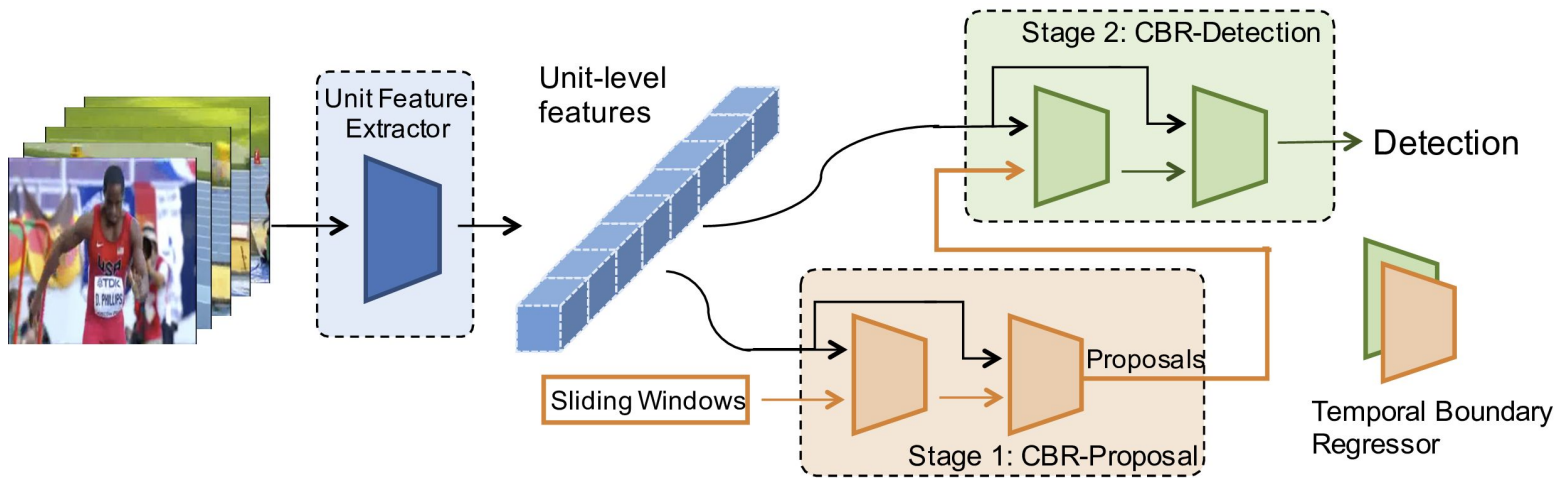
<https://github.com/jiyanggao/CBR>

Cascaded Boundary Regression for Action Detection

- TURN is only for proposal generation
- A unified two stage (proposal+detection) system for action detection
- Boundary regression for both proposal and detection.
- Cascaded Boundary Regression (CBR)

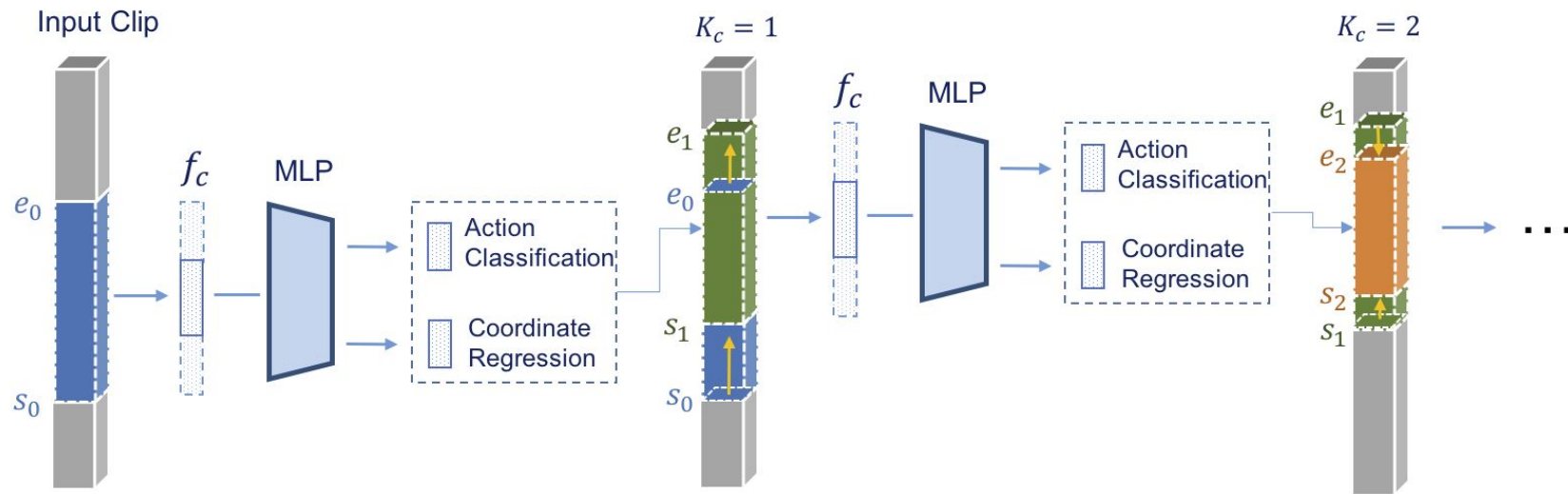


Two-stage Detection Architecture



- **Two-stage detection architecture:**
 - a. Videos are cut into small units, each contain 16 frames;
 - b. Unit-level features are reused for both proposal and detection;
 - c. Boundary regression is conducted in both stages.

CBR Architecture



For proposal generation:

$$c_{K_c^p} = \langle s_{K_c^p}, e_{K_c^p} \rangle, \quad p = \prod_{i=1}^{K_c^p} p_i$$

For action detection:

$$c_{K_c^d} = \langle s_{K_c^d}^z, e_{K_c^d}^z \rangle, \quad p = \prod_{i=1}^{K_c^d} p_i^z$$

Evaluation on THUMOS-14

Table 3: Comparison of cascaded step $K_c^d = 1, 2, 3, 4$ for temporal action detection (% mAP@tIoU=0.5) on THUMOS-14.

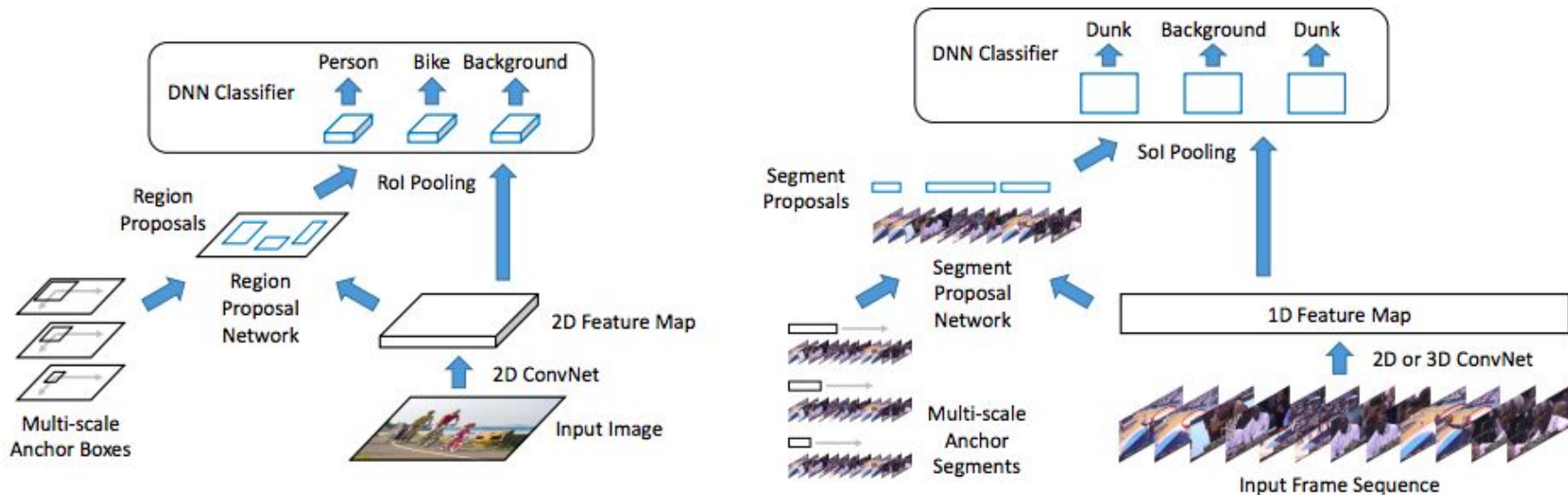
	$K_c^d = 1$	$K_c^d = 2$	$K_c^d = 3$	$K_c^d = 4$
CBR-C3D	21.5	22.7	22.4	22.2
CBR-TS	28.4	31.0	30.5	30.2

Table 5: Temporal action detection performance (mAP %) comparison at different tIoU thresholds on THUMOS-14.

tIoU	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Oneata <i>et al.</i> [15]	36.6	33.6	27.0	20.8	14.4	8.5	3.2
Yeung <i>et al.</i> [27]	48.9	44.0	36.0	26.4	17.1	-	-
Yuan <i>et al.</i> [28]	51.4	42.6	33.6	26.1	18.8	-	-
S-CNN [20]	47.7	43.5	36.3	28.7	19.0	10.3	5.3
CBR-C3D	48.2	44.3	37.7	30.1	22.7	13.8	7.9
CBR-TS	60.1	56.7	50.1	41.3	31.0	19.1	9.9

New Worth-reading

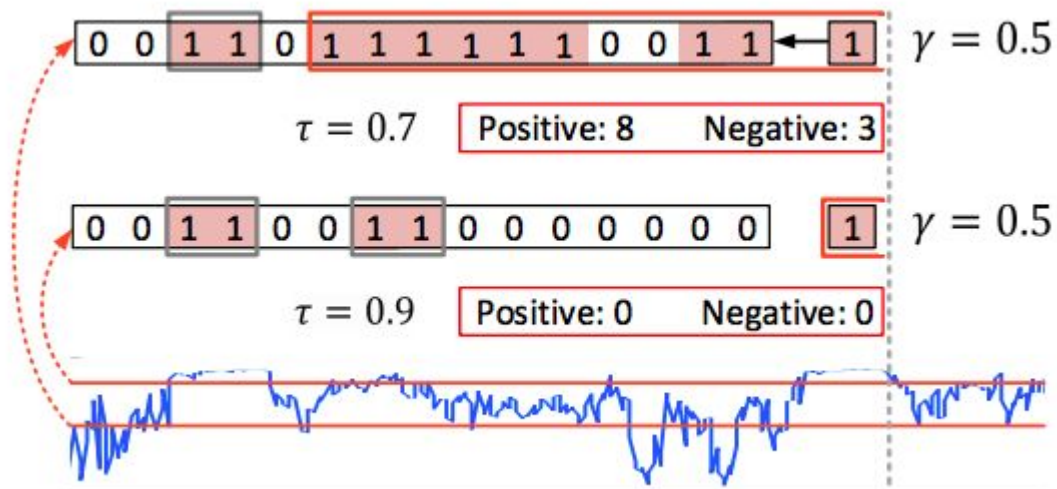
Chao, Yu-Wei, et al. "Rethinking the Faster R-CNN Architecture for Temporal Action Localization." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.



Outline

- Proposal-based methods
- **Brief Introduction on frame-based methods**
- How to combine?
- Online action detection
- Beyond the fixed activities

Temporal Action Grouping (TAG)



- Generating actionness score on every snippet (i.e. several frames).
- Watershed segmentation
- Non-maximum suppression

Outline

- Proposal-based methods
- Brief Introduction on frame-based methods
- **How to combine?**
- Online action detection
- Beyond the fixed activities

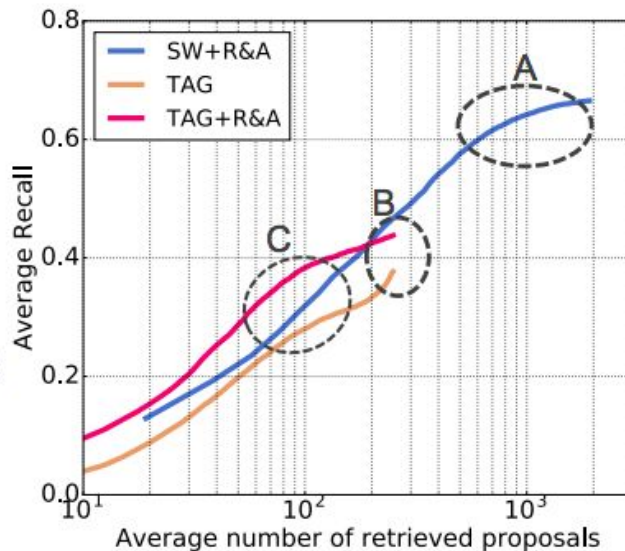
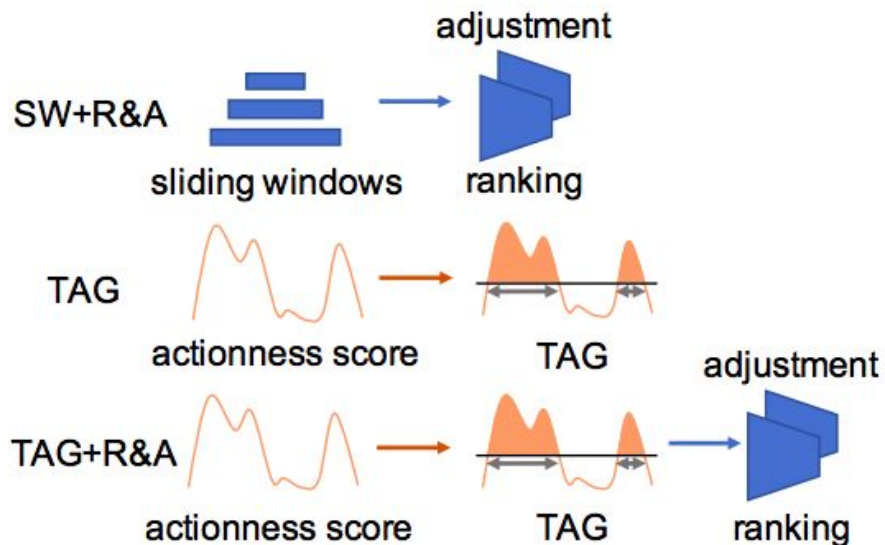
CTAP: Complementary Temporal Action Proposal Generation

Jiyang Gao*, Kan Chen*, Ram Nevatia

University of Southern California

<https://github.com/jiyanggao/CTAP>

Analysis on proposal based and frame based methods

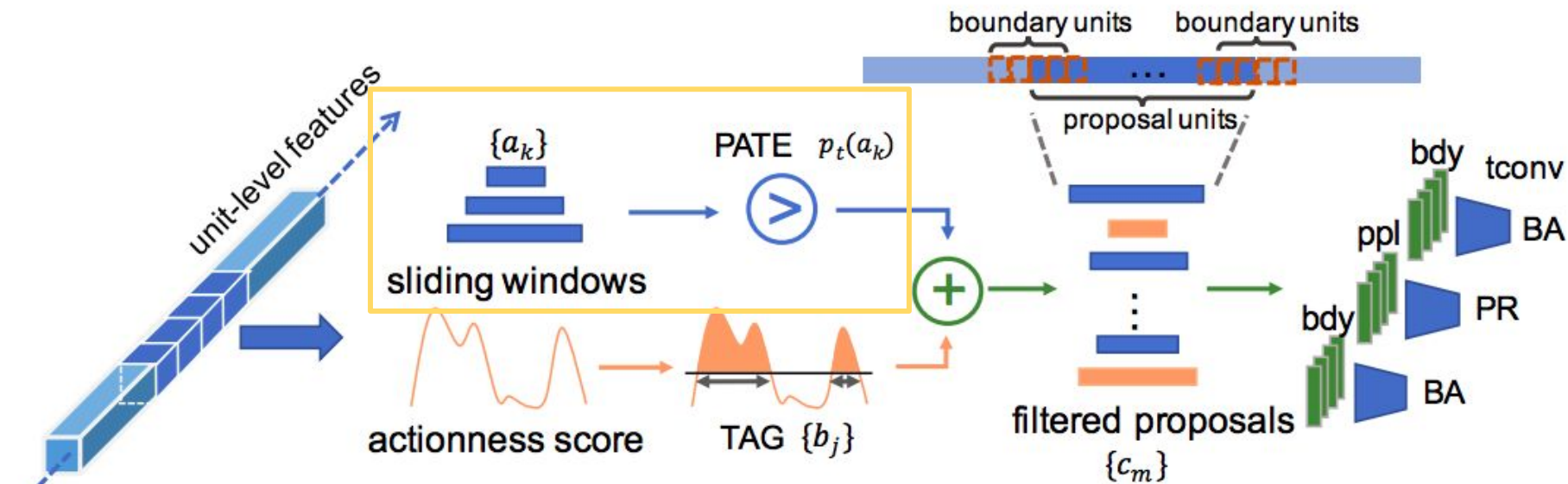


- TAG: precise boundary; poor ranking; missing proposals when actionness fails
- SW+R&A: Uniformly cover all segments; imprecise boundaries
- TAG+R&A: missing proposals when actionness fails

Complementary Properties

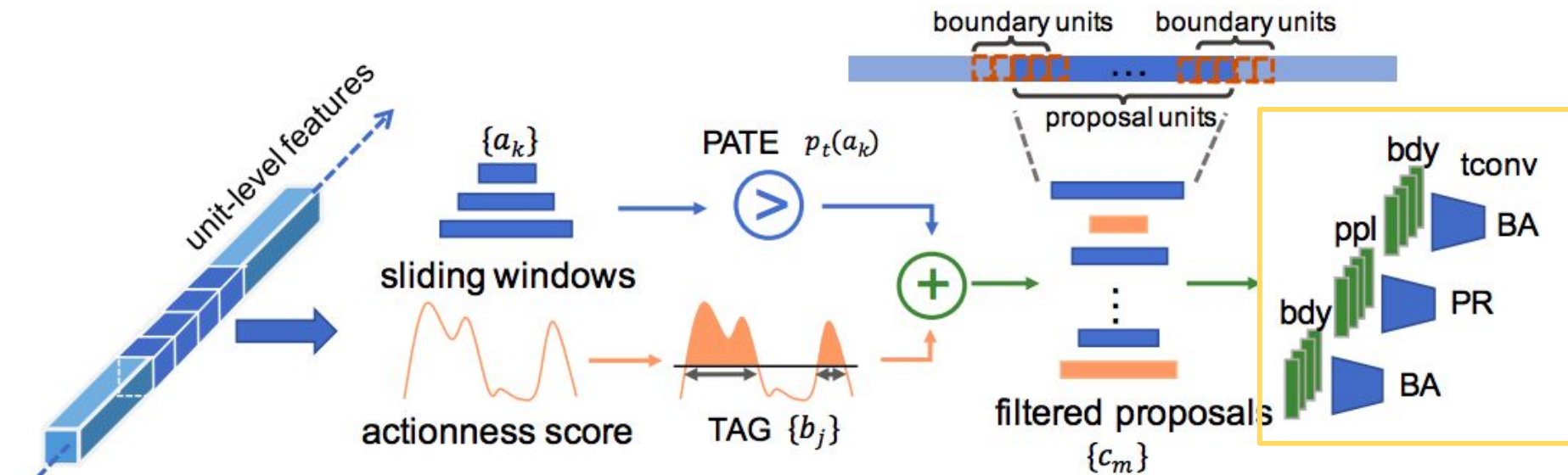
- TAG:
 - precise boundary;
 - poor ranking;
 - missing proposals when actionness fails
- SW+R&A:
 - imprecise boundaries;
 - strong ranking;
 - uniformly cover all segments
- Complementary properties:
 - precise vs imprecise boundary;
 - poor vs strong ranking;
 - missing proposals vs uniformly covering.

Complementary Temporal Proposal Generation



Proposal-level Actionness Trustworthiness Estimator (PATE): taking a sliding window as input, generating the probability that indicates whether this window can be correctly detected by the actionness scores and TAG.

Complementary Temporal Proposal Generation



Temporal convolutional Adjustment and Ranking (TAR): uniformly sample n units inside the proposal; continuously sample m units around the boundary; using temporal convolution to reserve temporal ordering information

Evaluation on THUMOS-14

Table 1. Performance comparison between TAR and TURN [2] on THUMOS-14 test set. Same unit feature (flow-16) and test sliding windows are used on TAR and TURN for fair comparison. Average Recall (AR) at different numbers is reported.

Method	AR@50	AR@100	AR@200
TURN[2]	21.75	31.84	42.96
TAR	22.99	32.21	45.08

Evaluation on THUMOS-14

Table 2. Complementary filtering evaluation on THUMOS-14 test set, compared with “Union” and “tIoU-selection”. Average Recall (AR) at different numbers is reported.

Method	AR@50	AR@100	AR@200
Union	25.80	34.70	46.19
Union+NMS	28.07	39.71	49.60
tIoU-selection	30.35	38.34	42.41
PATE complementary filtering	31.03	40.23	50.13

Evaluation on THUMOS-14

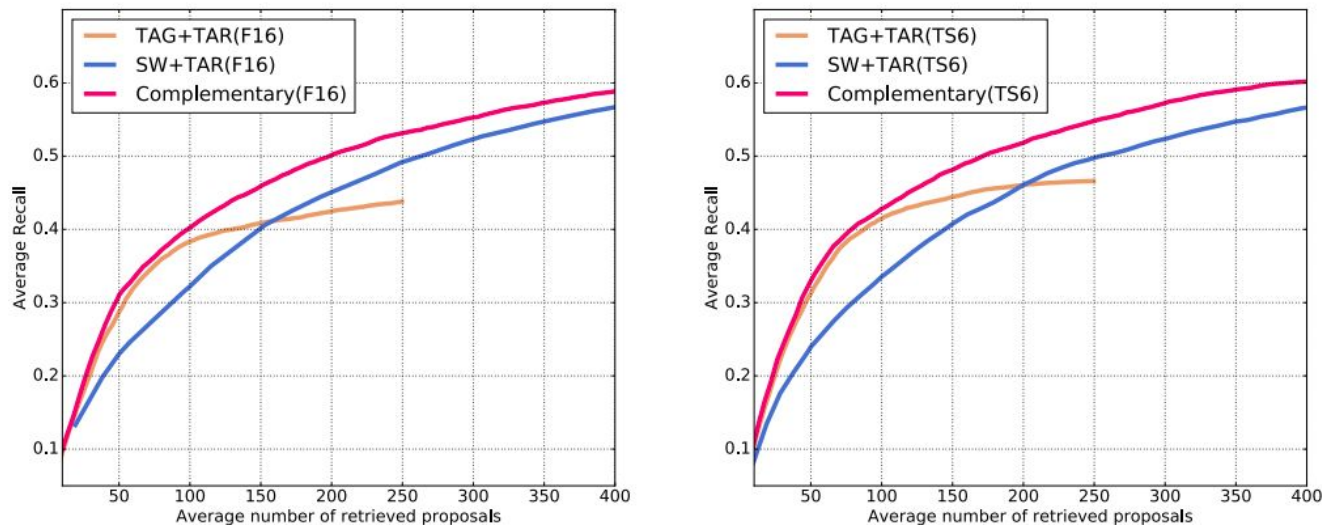


Fig. 3. AR-AN curves of the complementary results with flow-16 feature (F16) and two-stream-6 feature (TS6). Complementary filtering proposals outperform sliding windows (SW+TAR) and actionness proposals (TAG+TAR) consistently.

Evaluation on THUMOS-14

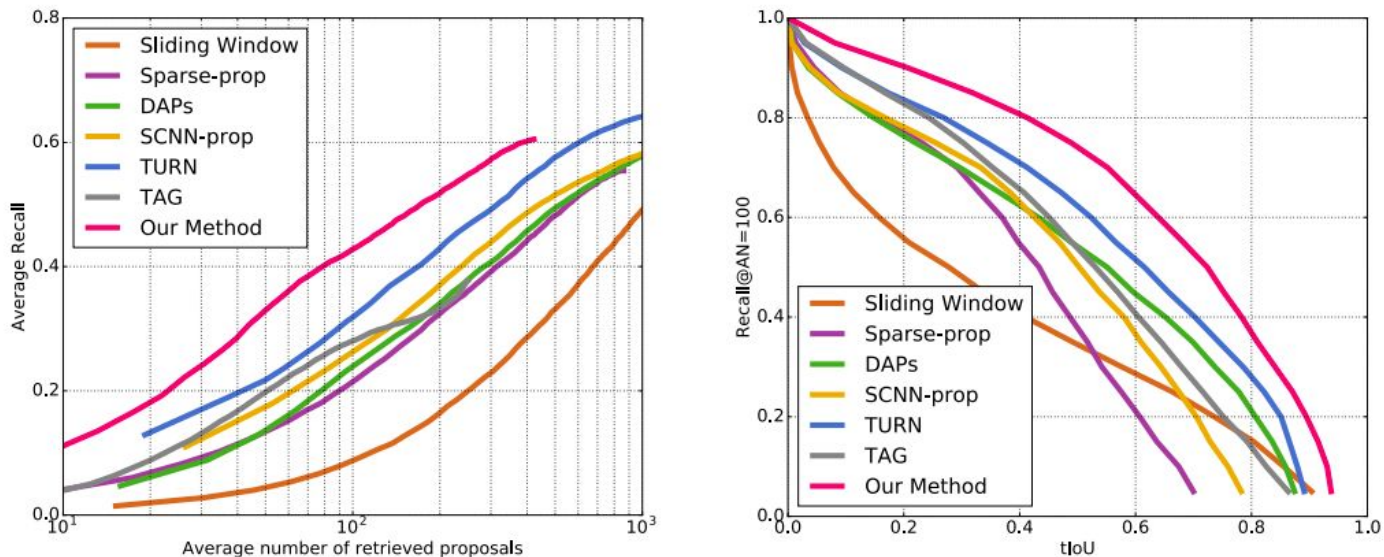


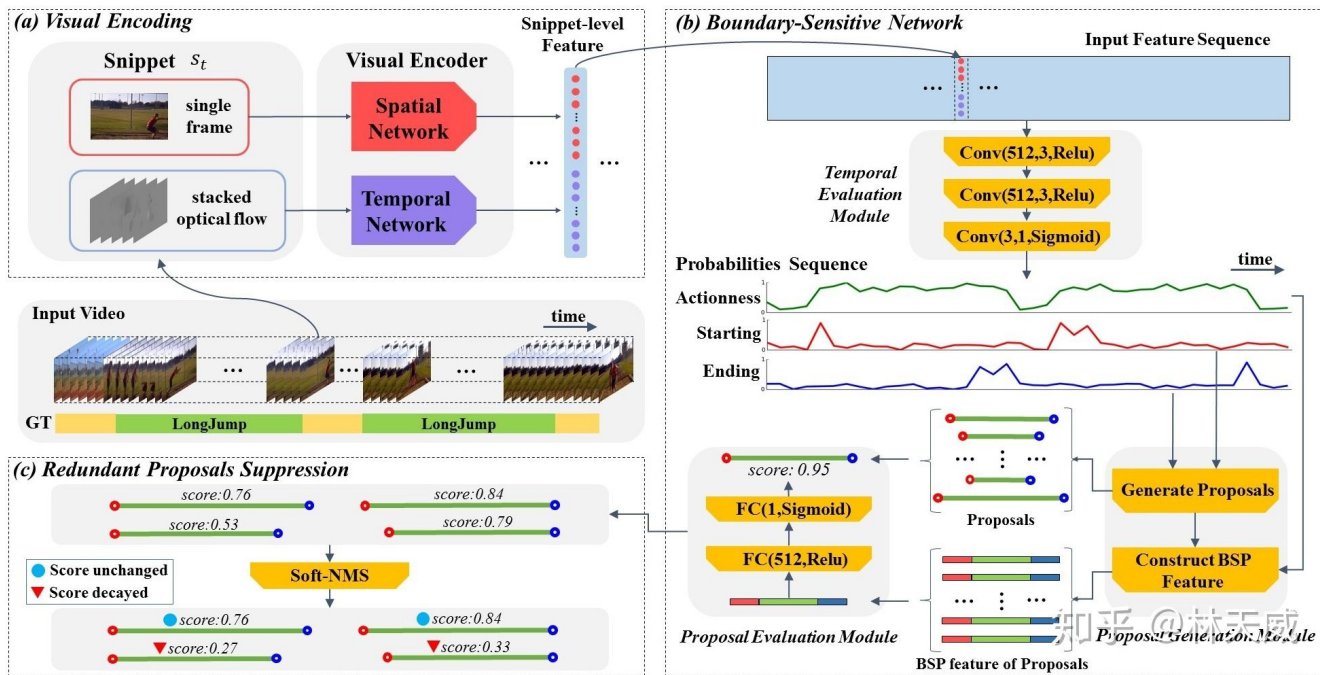
Fig. 4. AN-AR curve and recall@AN=100 curve of CTAP and state-of-the-art methods on THUMOS-14 test set.

Evaluation on THUMOS-14

Table 3. Comparison of CTAP and other proposal generation methods with the same action detector (SCNN) on THUMOS-14 test set, mean Average Precision (mAP % @tIoU=0.5) is reported.

Method	Sparse [8]	DAPs [11]	SCNN-prop[3]	TURN [2]	TAG[4]	CTAP-F16	CTAP-TS6
tIoU=0.5	15.3	16.3	19.0	25.6	25.9	27.9	29.9

New Worth-reading: Boundary Sensitive Network (BSN), ECCV 2018



Outline

- Proposal-based methods
- Brief Introduction on frame-based methods
- How to combine?
- **Online action detection**
- Beyond the fixed activities

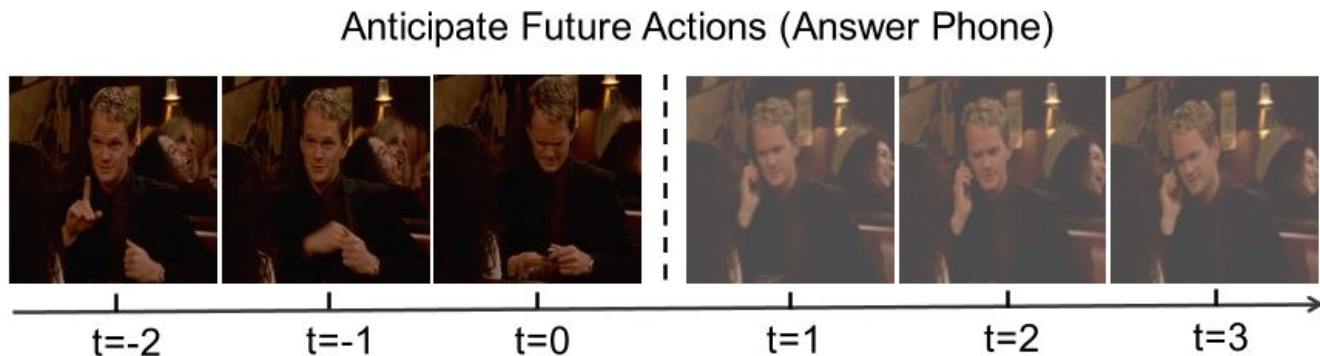
RED: Reinforced Encoder-Decoder Networks for Action Anticipation

Jiyang Gao¹, Zhenheng Yang¹, Ram Nevatia¹

¹University of Southern California

What is Action Anticipation?

Action anticipation refers to detection of an action before it happens.



Note that, online action detection [1] can be viewed as a special case for action anticipation, where the anticipation time is 0.

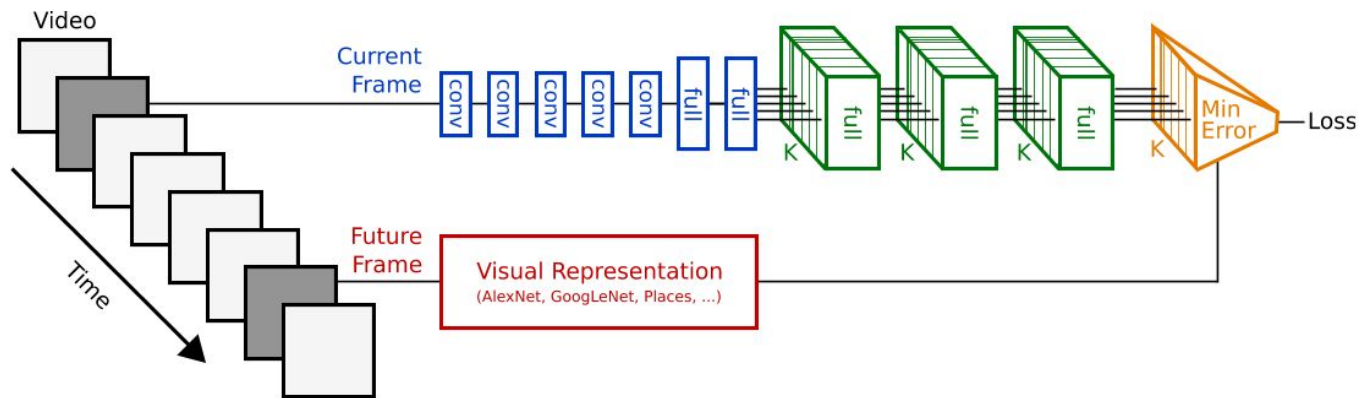
[1] De Geest, Roeland, et al. "Online action detection." *European Conference on Computer Vision*, 2016.

Challenges

- **For normal action detection:** need strong discriminative representations of video clips and ability to separate action instances from background data.
- **For action anticipation:** need the representation to capture sufficient historical and contextual information to make future predictions.

Previous work

Vondrick et al. [2] proposed to use deep neural networks to first anticipate visual representations of future frames and then categorize them to actions.

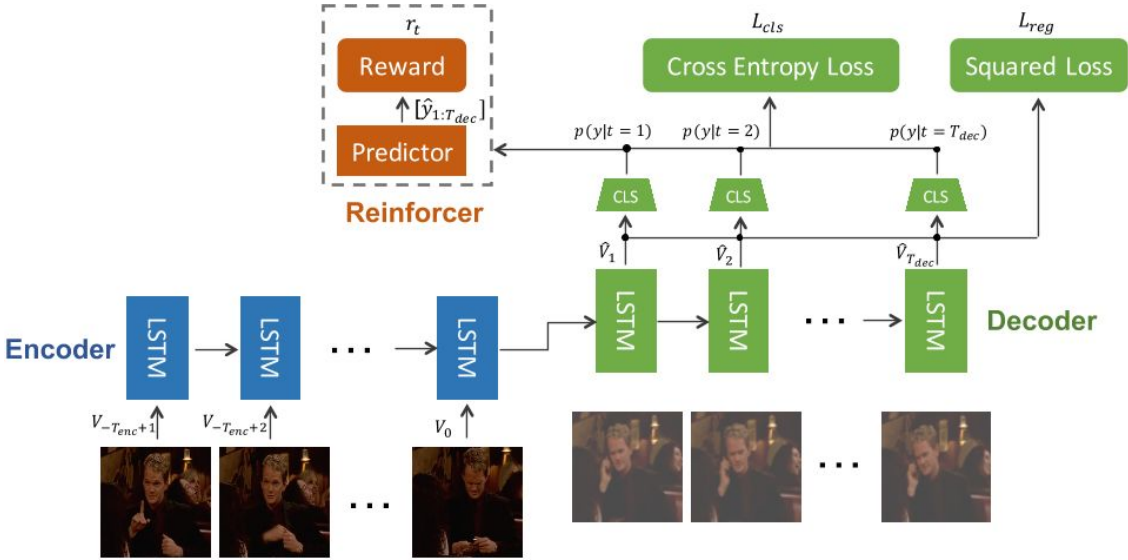


Limitations: 1. Anticipation is based on a single past frame; 2. Only anticipates for a fixed time ahead.

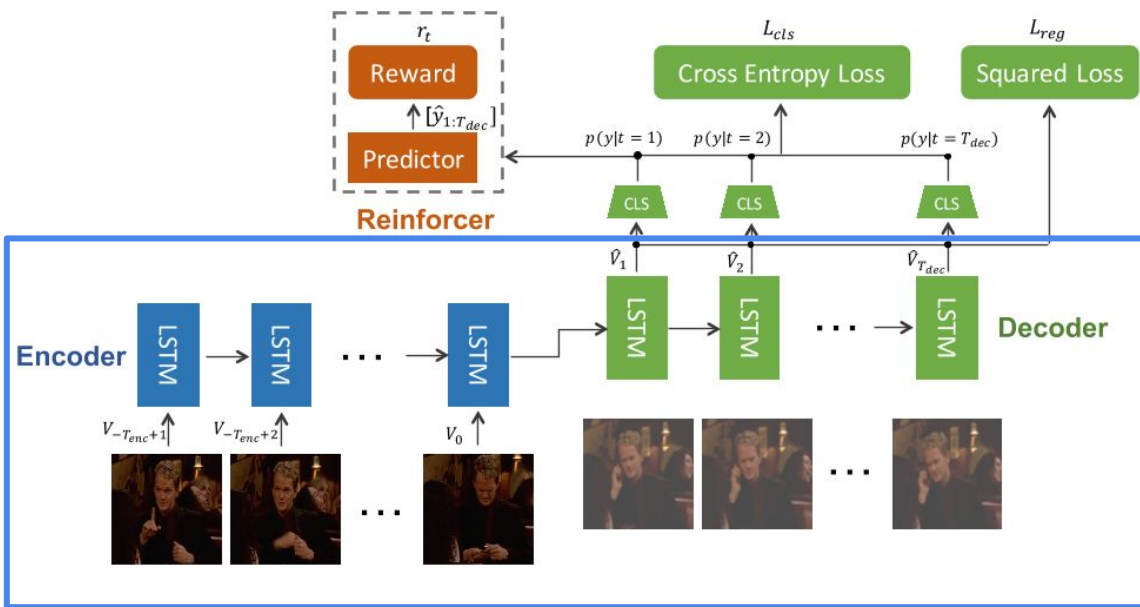
[2] Vondrick, Carl, Hamed Pirsiavash, and Antonio Torralba. "Anticipating visual representations from unlabeled video." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

Reinforced Encoder-Decoder (RED)

Encoder-Decoder Network; Classification Network; Reinforcement Module



RED: Encoder-Decoder Network



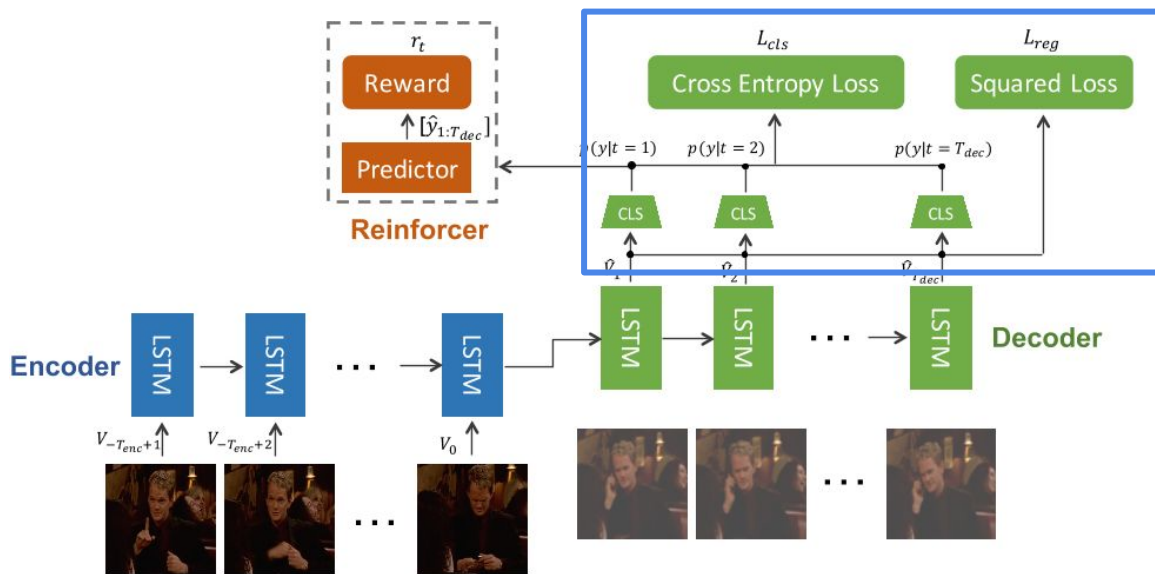
The input to the encoder LSTM is a vector sequence,

$$S_{in} = \{V_i\}, i \in [t - T_{enc}, t)$$

The decoder LSTM takes the last hidden state of encoder LSTM and outputs a prediction for the target sequence

$$S_{out} = \{\hat{V}_j\} j \in [t, t + T_{dec})$$

RED: Classification Network




Two fully-connected layers.


The output vector sequence of the encoder-decoder networks is processed by the classification network


RED: Reinforcement Module

Intuition: make the correct anticipation as early as possible

E.g.

Groundtruth: 011111 

Anticipation 1: 000111 

Anticipation 2: 001110 

"001110" gives the correct anticipation earlier than "000111", "001110" is a better anticipation at sequence level.

Classification cross-entropy loss would not capture such sequence-level distinctions, as it is calculated at each step.

RED: Reinforcement Module

A reward function to encourage the agent to make the correct anticipation as early as possible.

t_f is time that the label changes from background to action in groundtruth.

$$r_t = \frac{\alpha}{t+1-t_f}, \text{ if } t \geq t_f \text{ and } \hat{y}_t = y_t; \quad r_t = 0, \text{ otherwise}$$

$$R = \sum_{t=1}^{T_{dec}} r_t.$$

R is cumulative reward.

RED: Reinforcement Module

$$r_t = \frac{\alpha}{t+1-t_f}, \text{ if } t \geq t_f \text{ and } \hat{y}_t = y_t; \quad r_t = 0, \text{ otherwise}$$

Groundtruth: 011111



Anticipation 1: 000111



0 0 0 $\alpha/3$ $\alpha/4$ $\alpha/5$

Anticipation 2: 001110



0 0 $\alpha/2$ $\alpha/3$ $\alpha/4$ 0



RED: Reinforcement Module

Optimization by REINFORCE[3]

$$J(\theta) = E_{p(y(1:t));\theta} \left[\sum_{t=1}^{T_{dec}} r_t \right] = E_{p(y(1:t));\theta} [R]$$

$$\nabla_{\theta} J \approx \frac{1}{N} \sum_{k=1}^N \sum_{t=1}^{T_{dec}} \nabla_{\theta} \log \pi(a_t^k | h_{1:t}^k, a_{1:t-1}^k) R_t^k$$

$$\nabla_{\theta} J \approx \frac{1}{N} \sum_{k=1}^N \sum_{t=1}^{T_{dec}} \nabla_{\theta} \log p(y_t^k | y_{1:t-1}^k) (R_t^k - b_t^k)$$

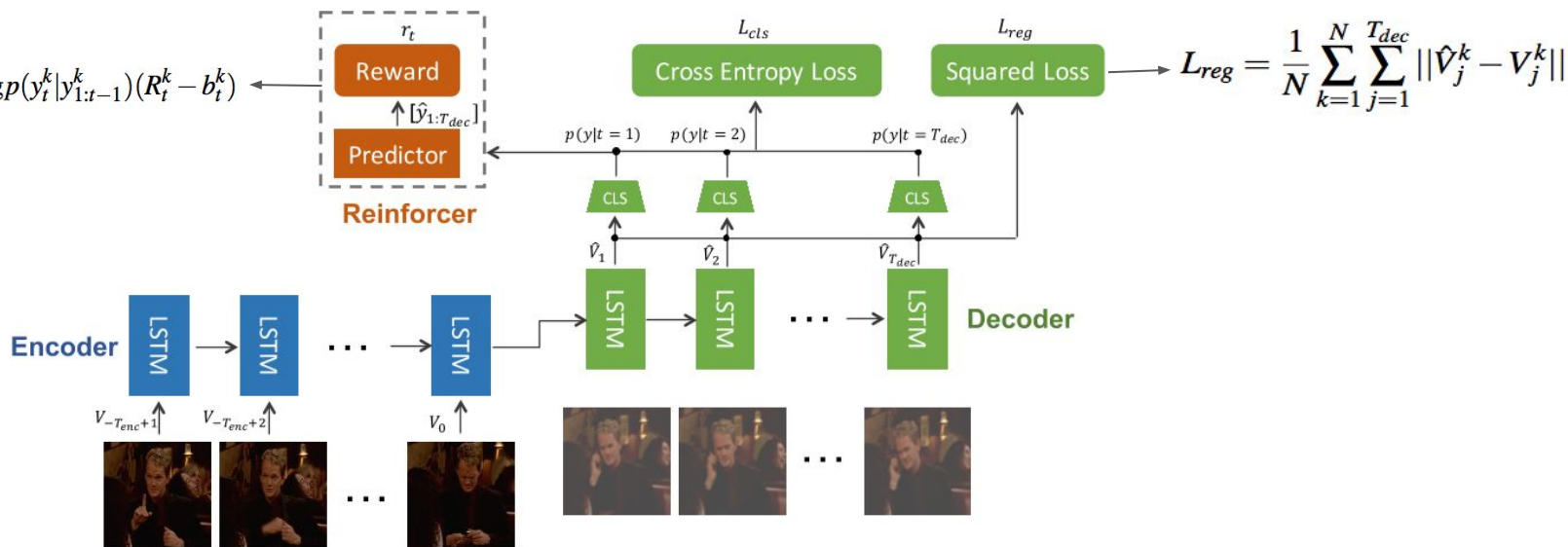
[3] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3-4):229–256, 1992

Joint Optimization

Classification loss, regression loss and reinforcement loss

$$L = L_{reg} + L_{cls} - J$$

$$\nabla_{\theta} J \approx \frac{1}{N} \sum_{k=1}^N \sum_{t=1}^{T_{dec}} \nabla_{\theta} \log p(y_t^k | y_{1:t-1}^k) (R_t^k - b_t^k)$$



Evaluation

Datasets: THUMOS-14, TVSeries, TV-interaction

Metric: average precision (AP), calibrated average precision (cAP).

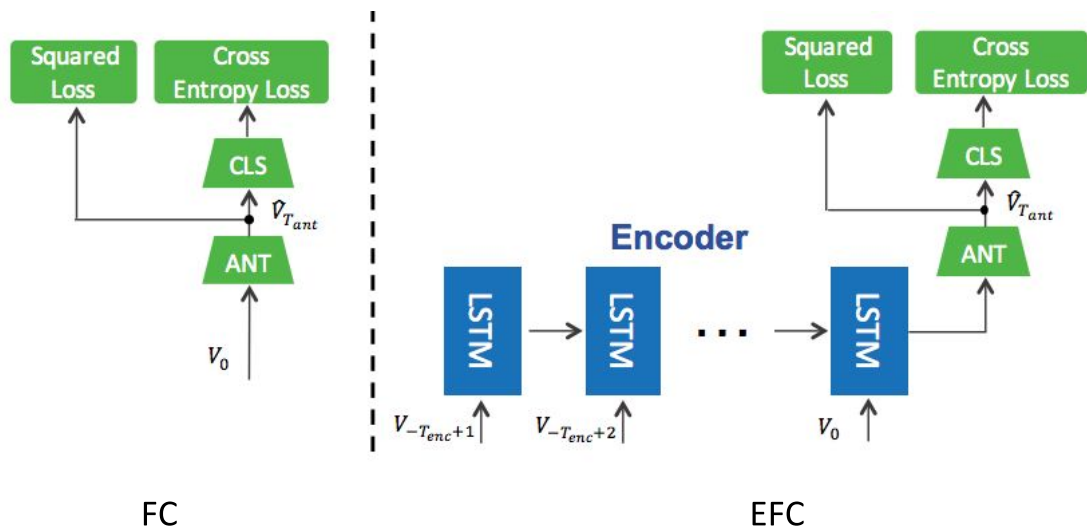
Baselines:

FC (fully connected)

EFC(encoder+fully connected)

ED (encoder-decoder)

RED (reinforced encoder-decoder)



Comparison of anticipation methods

- Compare the performance at anticipation time $T_a = 1s$ on THUMOS-14, TVSeries.

Table 1: Action anticipation comparison on TVSeries (cAP %) test set and THUMOS-14 (per-frame mAP %) test set at 1s (4 chunks) with two-stream features.

	FC	EFC	ED	RED
TVSeries (cAP@ $T_a=1s$ %)	72.4	73.3	74.6	75.5
THUMOS-14 (mAP@ $T_a=1s$ %)	31.7	33.9	36.8	37.5

- TV-interaction: we compare two features, VGG and two-stream. [19] uses Alexnet as base architecture

Table 2: Action anticipation comparison (ACC %) on TV-Human-Interaction at $T_a = 1s$ (4 chunks).

	Vondrick <i>et al.</i> [19] (THUMOS)	RED-VGG (TVSeries)	RED-TS (THUMOS)
ACC@ $T_a=1s$ (%)	43.6	47.5	50.2

Varying anticipation time

- Anticipation times from 0.25s to 2.0s, on TVSeries.

Table 3: Detailed action anticipation (cAP %) comparison for ED and RED on TVSeries test set from $T_a = 0.25s$ to $T_a = 2.0s$ with two-stream representations and VGG features.

time	0.25s	0.5s	0.75s	1.0s	1.25s	1.5s	1.75s	2.0s
ED-VGG	71.0	70.6	69.9	68.8	68.0	67.4	67.0	66.7
RED-VGG	71.2	71.0	70.6	70.2	69.2	68.5	67.5	66.8
ED-TS	78.5	78.0	76.3	74.6	73.7	72.7	71.7	71.0
RED-TS	79.2	78.7	77.1	75.5	74.2	73.0	72.0	71.2

- Anticipation times from 0.25s to 2.0s, on THUMOS-14.

Table 4: Detailed action anticipation (per-frame mAP %) comparison for ED and RED on THUMOS-14 test set from $T_a = 0.25s$ to $T_a = 2s$ with two-stream representations.

time	0.25s	0.5s	0.75s	1.0s	1.25s	1.5s	1.75s	2.0s
ED-TS	43.8	40.9	38.7	36.8	34.6	33.9	32.5	31.6
RED-TS	45.3	42.1	39.6	37.5	35.8	34.4	33.2	32.1

Online action detection

Performance comparison on online action detection (anticipation time=0)

Table 5: Comparison on online action detection in TVSeries test set.

	CNN[10]	LSTM[10]	FV [10]	RED-VGG	RED-TS
cAP (%)	60.8	64.1	74.3	71.2	79.2

Table 6: Online action detection comparison on THUMOS-14 test set (per-frame mAP %) with two stream features.

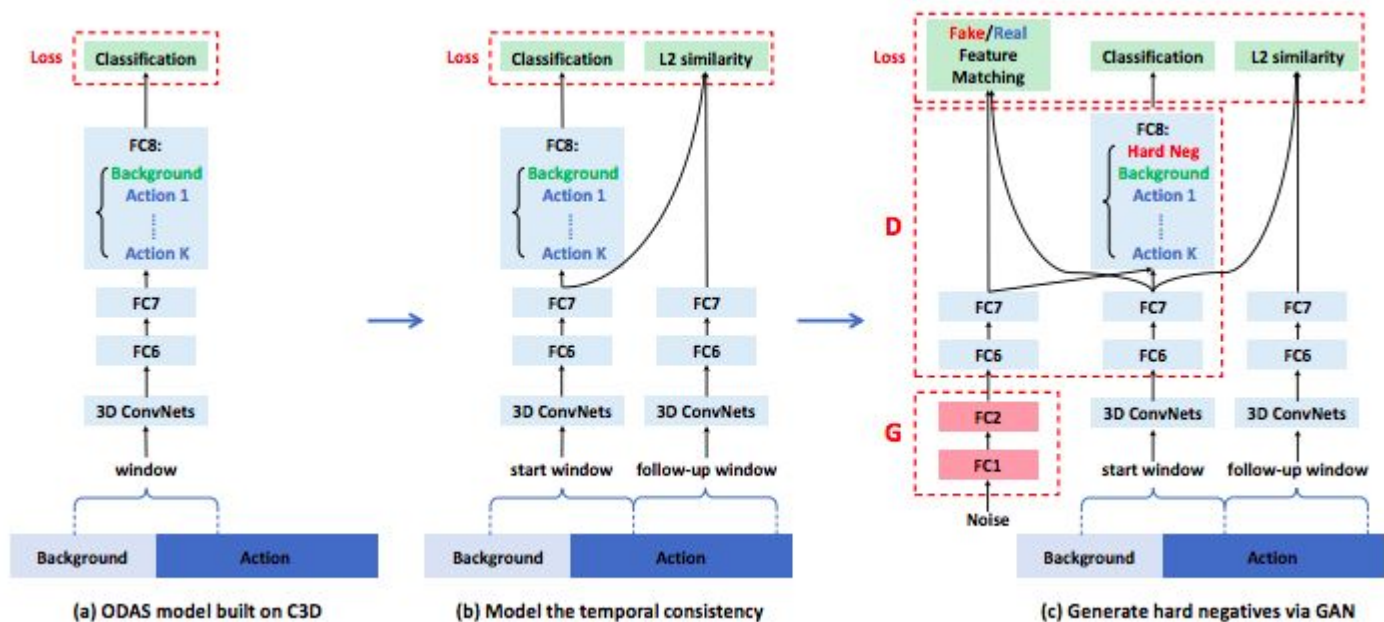
	two-stream[17]	LSTM[23]	MultiLSTM[23]	RED
mAP(%)	36.2	39.3	41.3	45.3

Outline

- Proposal-based methods
- Brief Introduction on frame-based methods
- How to combine?
- Online action detection
- **Beyond the fixed activities**

New Worth-reading

Shou, Zheng, et al. "Online Action Detection in Untrimmed, Streaming Videos-Modeling and Evaluation." *arXiv preprint arXiv:1802.06822* (2018).



Limitations of Temporal Action Detection

- Current action localization methods are restricted by pre-defined list of actions.

Walk/run, look out the window



9.3 s |----->| 14.4 s

- Same action, different names ?
- complex activity queries ?

TALL: Temporal Activity Localization via Language Query

Jiyang Gao¹, Chen Sun², Zhenheng Yang¹, Ram Nevatia¹

¹University of Southern California

²Google Research

<https://github.com/jiyanggao/TALL>

Problem Definition

Language Query:

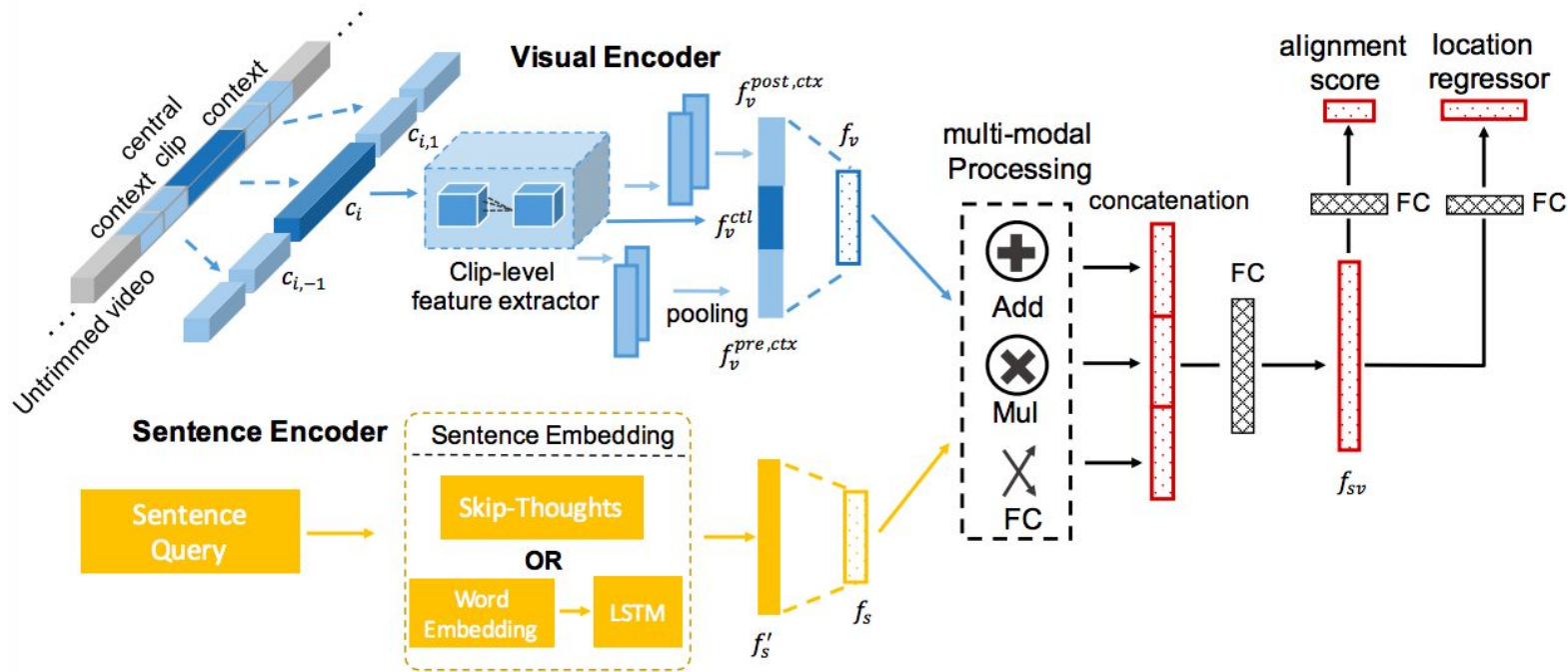
A person runs to the window and then look out



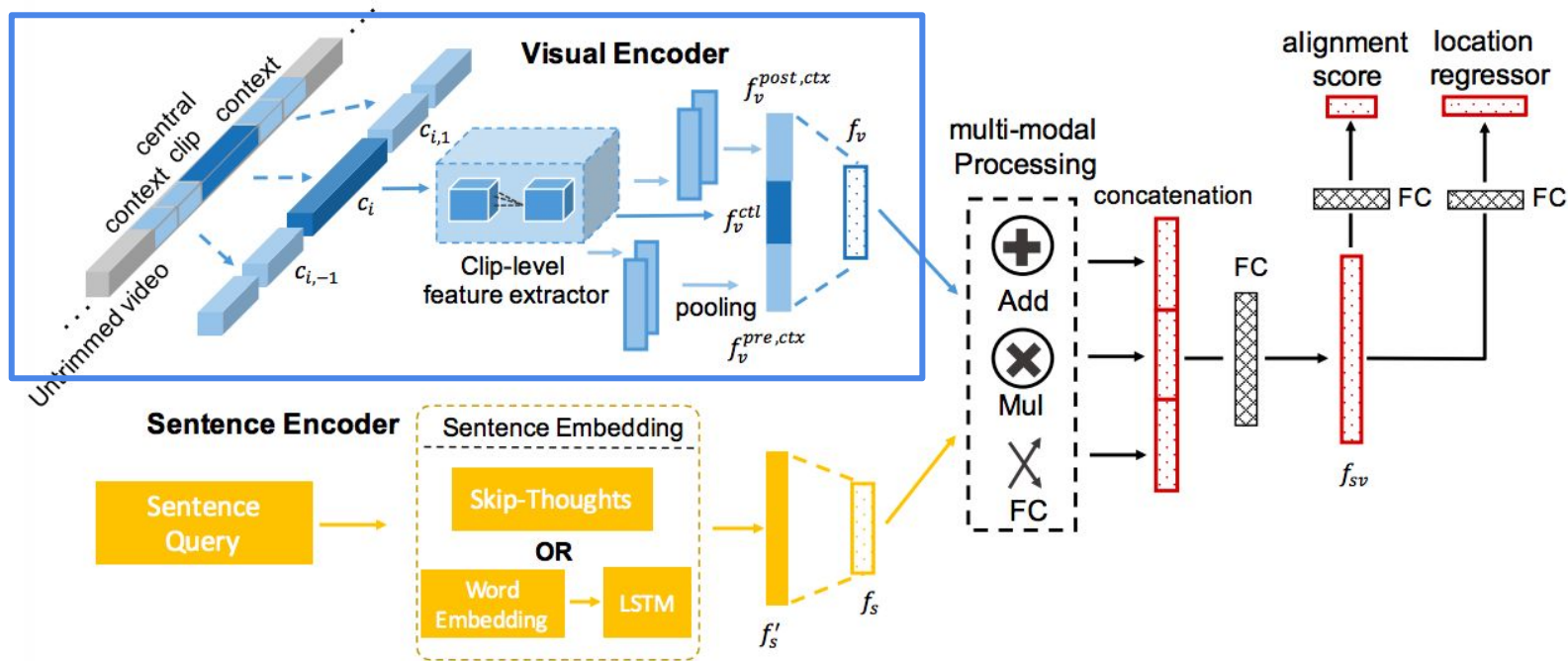
9.3 s |----->| 14.4 s

Given a temporally untrimmed video and a natural language query, the goal is to determine the start and end times for the described activity inside the video.

Cross-modal Temporal Regression Localizer (CTRL)

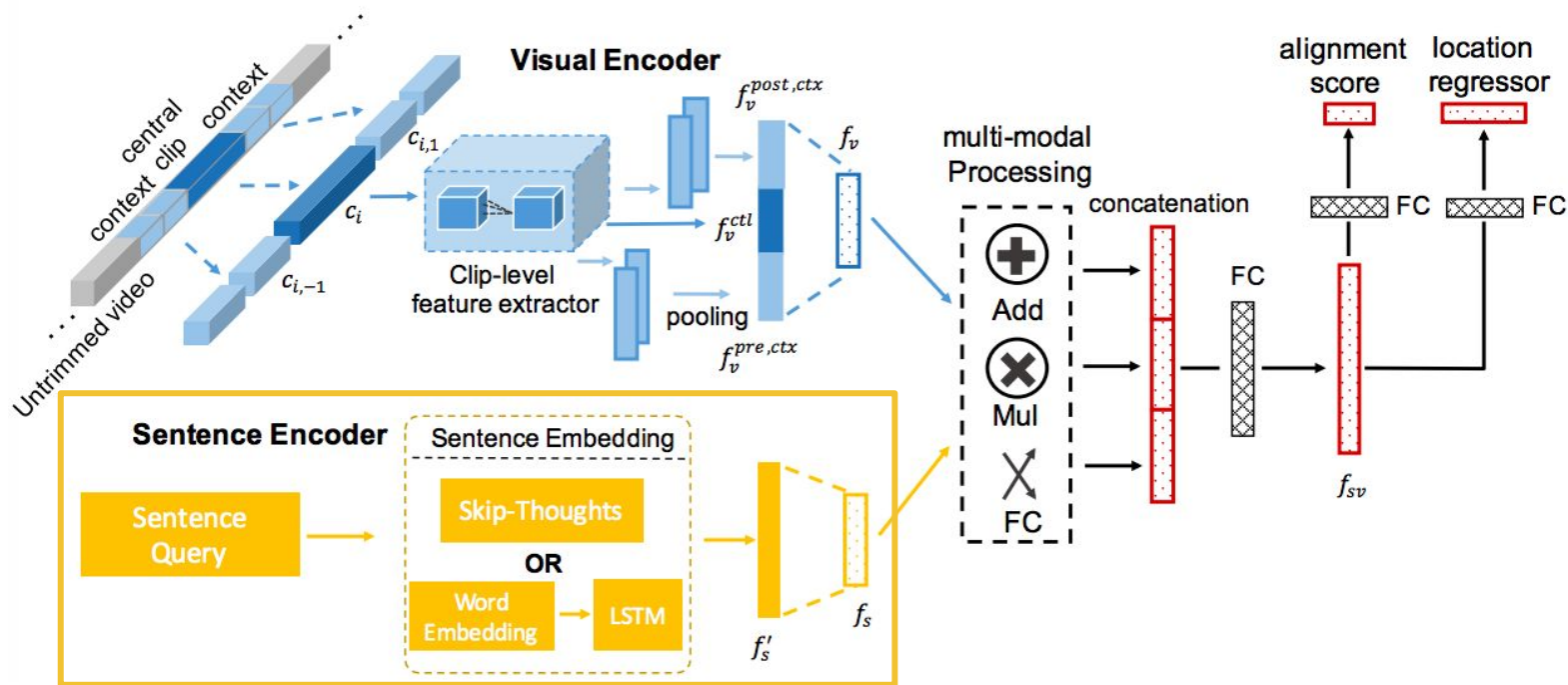


Cross-modal Temporal Regression Localizer (CTRL)



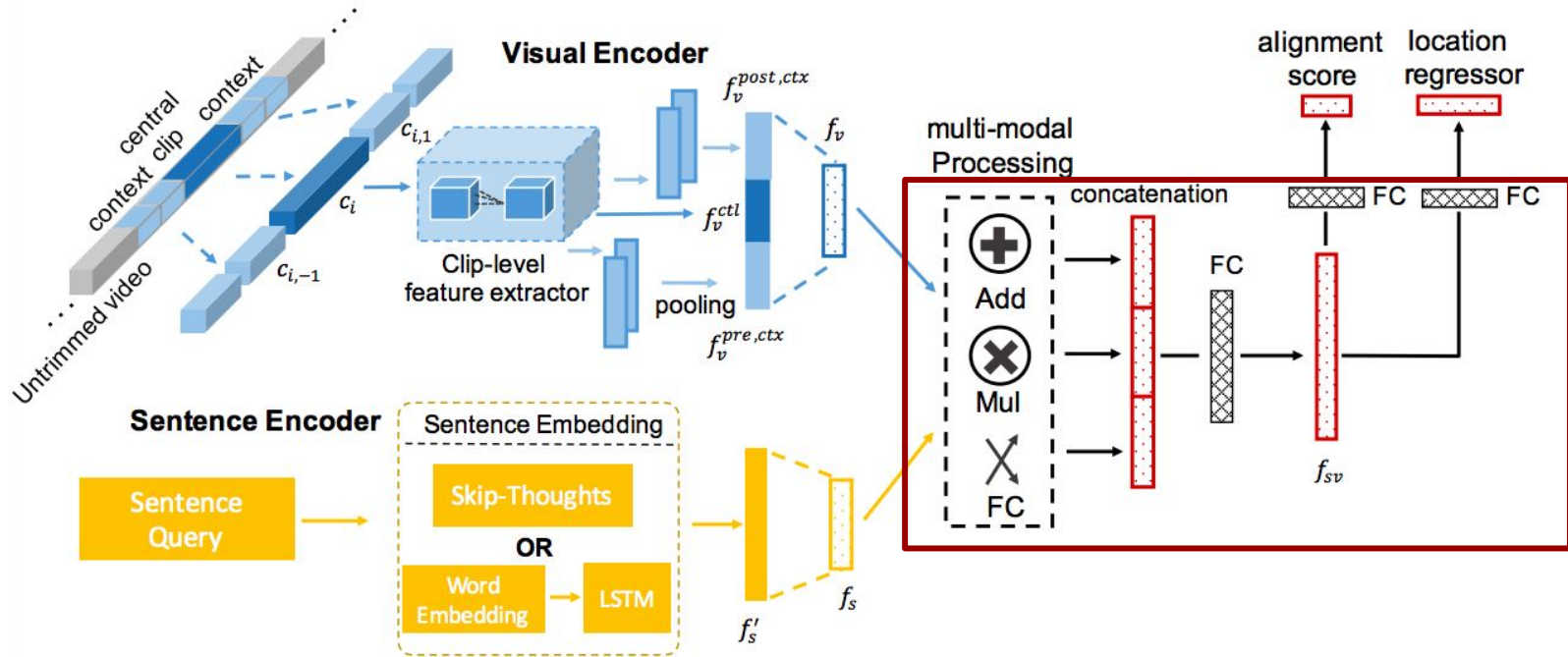
Visual Encoder: Extract clip level features, which contains context feature and central feature.

Cross-modal Temporal Regression Localizer (CTRL)



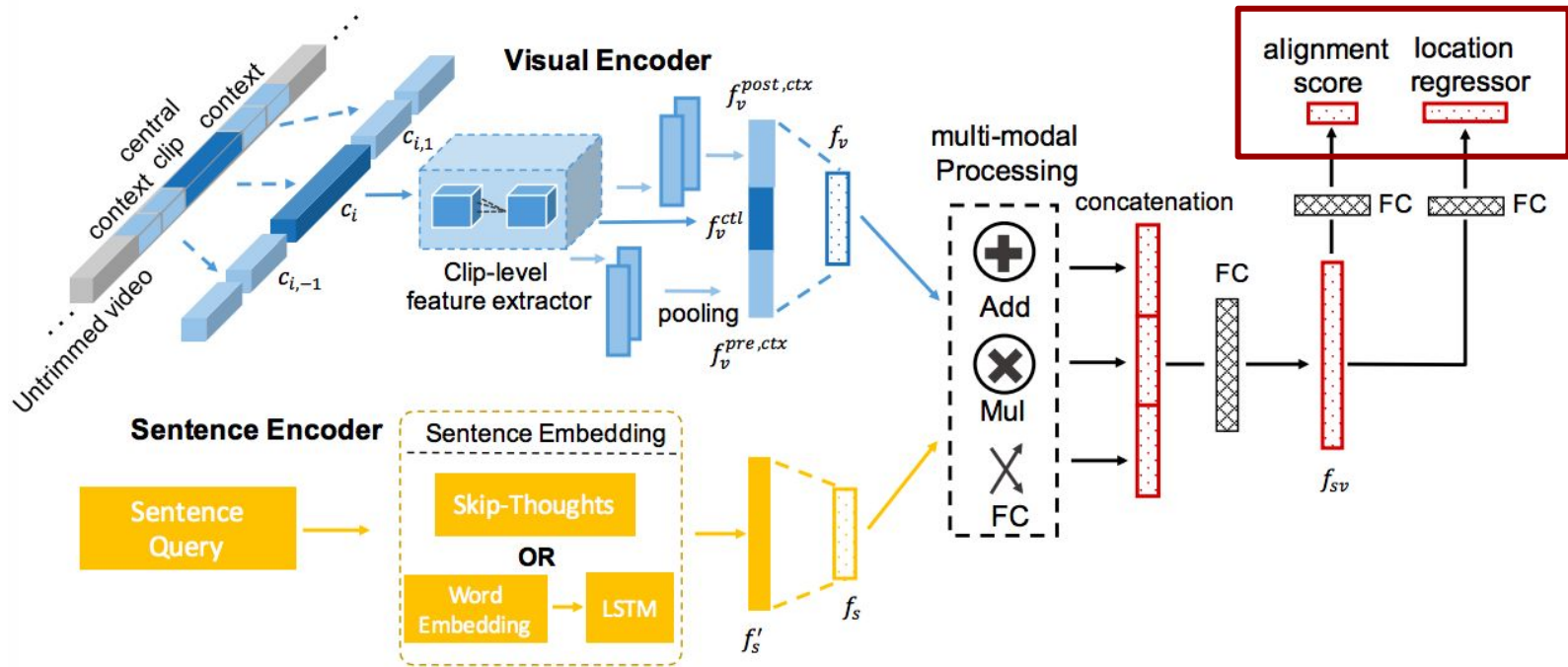
Sentence Encoder: Extract sentence embeddings, off-the-shelf Skip-thought or LSTM encoder.

Cross-modal Temporal Regression Localizer (CTRL)



Multimodal Processing: Combine the information from vision and text by Add, Mul and FC.

Cross-modal Temporal Regression Localizer (CTRL)



Alignment & Regression: visual-semantic alignment and boundary regression.

Visual-semantic alignment

High positive scores for aligned clip-sentence pairs

Low negative scores for misaligned clip-sentence pairs.

$$L_{aln} = \frac{1}{N} \sum_{i=0}^N [\alpha_c \log(1 + \exp(-cs_{i,i})) + \sum_{j=0, j \neq i}^N \alpha_w \log(1 + \exp(cs_{i,j}))]$$

Temporal Boundary Regression

- Parameterized offsets: central point and length of the clip

$$t_p = (p - p_c)/l_c, t_l = \log(l/l_c)$$

- Non-parameterized offsets: start and end time of the clip

$$t_s = s - s_c, t_e = e - e_c$$

- Regression Loss:

$$L_{reg} = \frac{1}{N} \sum_{i=0}^N [R(t_{x,i}^* - t_{x,i}) + R(t_{y,i}^* - t_{y,i})]$$

Evaluation

Datasets: [TACoS](#), contains 127 videos that have natural language descriptions and temporal locations; 17344 pairs of sentence and video clips. [Charades-STA](#) contains 13898 clip-sentence pairs.

Metric: $R@n$, $IoU=m$, the percentage of at least one of the top-n results having Intersection over Union (IoU) with the groundtruth larger than m.

Baselines: [Verb and object classifiers](#), train classifiers based on annotations of pre-defined actions and objects; [VSA-RNN](#), RNN + C3D + visual-semantic alignment network; [VSA-STV](#), Skipthought + C3D + visual-semantic alignment network

Evaluation

Comparison of different methods on TACoS.

Table 1. Comparison of different methods on TACoS

Method	R@1	R@1	R@1	R@5	R@5	R@5
	IoU=0.5	IoU=0.3	IoU=0.1	IoU=0.5	IoU=0.3	IoU=0.1
Random	0.83	1.81	3.28	3.57	7.03	15.09
Verb	1.62	2.62	6.71	3.72	6.36	11.87
Verb+Obj	8.25	11.24	14.69	16.46	21.50	26.60
VSA-RNN	4.78	6.91	8.84	9.10	13.90	19.05
VSA-STV	7.56	10.77	15.01	15.50	23.92	32.82
CTRL (aln)	10.67	16.53	22.29	19.44	29.09	41.05
CTRL (loc)	10.70	16.12	22.77	18.83	31.20	45.11
CTRL (reg-p)	11.85	17.59	23.71	23.05	33.19	47.51
CTRL (reg-np)	13.30	18.32	24.32	25.42	36.69	48.73

Evaluation

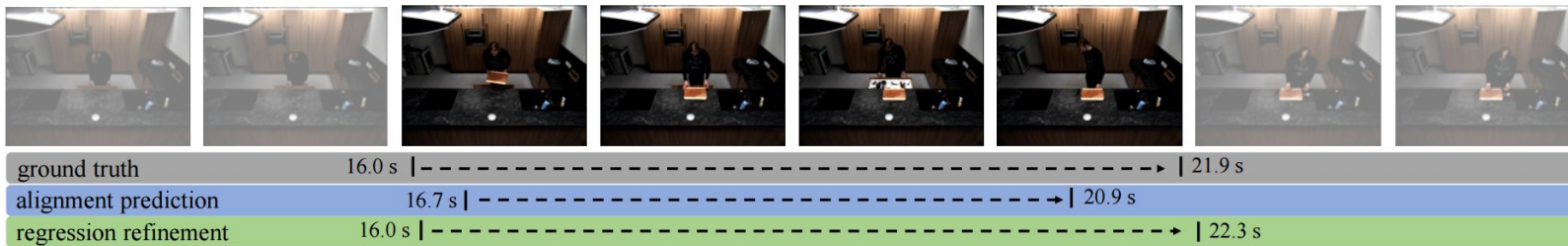
Comparison of different methods on Charades-STA.

Table 2. Comparison of different methods on Charades-STA

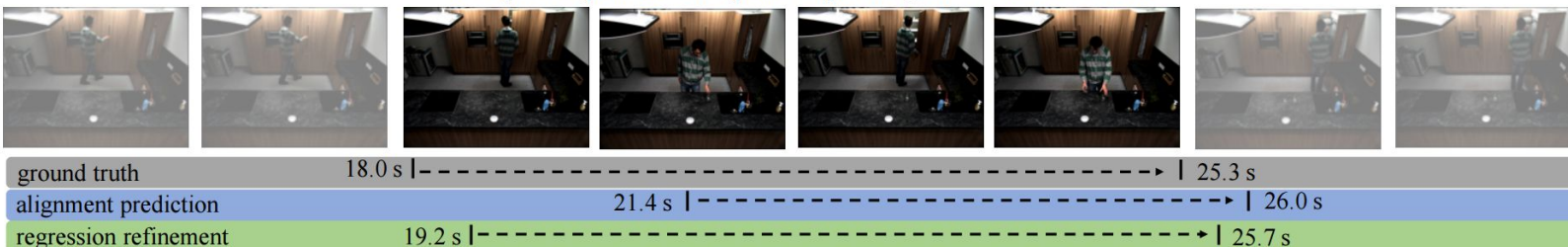
Method	R@1	R@1	R@5	R@5
	IoU=0.5	IoU=0.7	IoU=0.5	IoU=0.7
Random	8.51	3.03	37.12	14.06
VSA-RNN	10.50	4.32	48.43	20.21
VSA-STV	16.91	5.81	53.89	23.58
CTRL (aln)	18.77	6.53	54.29	23.74
CTRL (loc)	20.19	6.92	55.72	24.41
CTRL (reg-p)	22.27	8.46	57.83	26.61
CTRL (reg-np)	23.63	8.89	58.92	29.52

Visualization Results

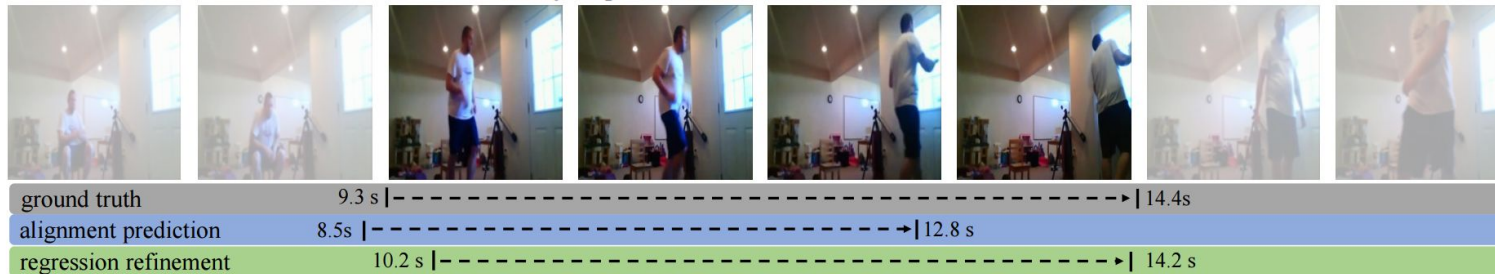
Query: He gets a cutting board and knife.



Query: The person sets up two separate glasses on the counter.



Query: A person runs to the window and then look out



Thanks