Bottom-Up Temporal Action Localization with Mutual Regularization

Peisen Zhao[1], Lingxi Xie[2], Chen Ju[1], Ya Zhang[1]✉, Yanfeng Wang[1], and Qi Tian[2]

[1] Cooperative Medianet Innovation Center, Shanghai Jiao Tong University
{pszhao, ju_chen, ya_zhang, wangyanfeng}@sjtu.edu.cn
[2] Huawei Inc.
198808xc@gmail.com, tian.qi1@huawei.com

Abstract. Recently, temporal action localization (TAL), *i.e.*, finding specific action segments in untrimmed videos, has attracted increasing attentions of the computer vision community. State-of-the-art solutions for TAL involves evaluating the frame-level probabilities of three action-indicating phases, *i.e.* starting, continuing, and ending; and then post-processing these predictions for the final localization. This paper delves deep into this mechanism, and argues that existing methods, by modeling these phases as individual classification tasks, ignored the potential temporal constraints between them. This can lead to incorrect and/or inconsistent predictions when some frames of the video input lack sufficient discriminative information. To alleviate this problem, we introduce two regularization terms to mutually regularize the learning procedure: the Intra-phase Consistency (IntraC) regularization is proposed to make the predictions verified inside each phase; and the Inter-phase Consistency (InterC) regularization is proposed to keep consistency between these phases. Jointly optimizing these two terms, the entire framework is aware of these potential constraints during an end-to-end optimization process. Experiments are performed on two popular TAL datasets, THUMOS14 and ActivityNet1.3. Our approach clearly outperforms the baseline both quantitatively and qualitatively. The proposed regularization also generalizes to other TAL methods (*e.g.*, TSA-Net and PGCN). code: `https://github.com/PeisenZhao/Bottom-Up-TAL-with-MR`

Keywords: Action localization, action proposals, mutual regularization

1 Introduction

Temporal Action Localization (TAL), aiming to locate action instances from untrimmed videos, is a fundamental task in video content analysis. TAL can be divided into two parts, temporal action proposal and action classification. The latter is relatively well studied with cogent performance achieved by recent action classifiers [6,28,34,37,35]. To improve the performance in standard benchmarks [16,5], how to generate precise action proposals remains a challenge.

Early approaches for generating action proposals mostly adopt a **top-down** approach, i.e., first generate regularly distributed proposals (*e.g.*, multi-scale sliding windows), and then evaluate their confidence. However, the top-down methods [4,9,32,3,13] often suffer from over-generating candidate proposals and rigid proposal boundaries. To solve the above problem, **bottom-up** approaches have been proposed [38,23,22,25,15]. A typical bottom-up method first densely evaluates the frame-level probabilities of three action-indicating phases, i.e. starting, continuing, and ending; then groups action proposals based on the located candidate starting and ending points. This design paradigm enables flexible action proposal generation and achieves a high recall with fewer proposals [38], which has become a more preferred practice in temporal action proposals.

Predicting the frame-level probability of the starting, continuing, and ending phases of actions is crucial for the success of bottom-up approaches. Existing methods model it as three binary classification tasks and use frame-level positive and negative labels converted from action temporal location as supervision, which can suffer the difficulty of learning from limited and/or ambiguous training data. In particular, it is often difficult to determine the accurate time that an action starts or ends, and even when the action continues, there is no guarantee that every frame contains sufficient information of being correctly classified. In other words, one may need to refer to complementary information to judge the status of an action, *e.g.*, if there is no clear sign that an action has ended, the probability that it is continuing is high. Ignoring such temporal relationship may lead to erroneous and inconsistent predictions. Thus, independent classification tasks have the following two drawbacks. **First**, each temporal location is considered as an isolated instance and their probabilities are calculated independently, without considering the temporal relationship among them. In fact, for any of the three phases, the probability is expected to have relatively smooth predictions among contiguous temporal locations. Ignoring the temporal relationship may leads to inconsistent predictions. **Second**, the modeling of the probability for starting, continuing, and ending phases are independent of each other. In fact, for any action, the starting, continuing, and ending phases always come as an ordered triplet. Ignoring the ordering relationship of the three phases could lead to contradictory predictions.

In this paper, we address this issue explicitly by exploring two regularization terms. To enforces the temporal relationship among predictions, **Intra-phase Consistency** (IntraC) regularization is proposed, which targets to minimize the discrepancy inside *positive* or *negative* regions of each phases, and maximize the discrepancy between *positive* and *negative* regions. To meet the ordering constraint of the three phases, we introduce **Inter-phase Consistency** (InterC) regularization, which enforces consistency among the probability of the three phases, by operating between continuing-starting and continuing-ending. When introducing the above two regularization terms to the original loss of bottom-up temporal action localization network, the optimization of IntraC and InterC may be considered as a form of mutual regularization among the three classifiers, since the predictions of the three phases are now coupled via consistency check on classifier outputs. With the above mutual regularization, the entire framework remains end-to-end trainable while enforcing the above constraints.

To validate the effectiveness of the proposed method, we perform experiments on two popular benchmark datasets, THUMOS14 and ActivityNet1.3. Our experimental results have demonstrated that our approach clearly outperforms the state-of-the-arts both quantitatively and qualitatively. Especially on THUMOS14 dataset, we improve absolute 6.8% mAP at a strict IoU of 0.7 settings from the previous best. Moreover, we show that the proposed mutual regularization is independent of the temporal action localization framework. When we introduce IntraC and InterC to other network (TSA-Net [15]) or framework (PGCN [41]), better performance is also achieved.

2 Related Work

Action recognition. Same as image recognition in image analysis, action recognition is a fundamental task in video domain. Extensive models [33,6,34,37,35,27] on action recognition have been widely studied. Deeper models [6,28,21], more massive datasets [17,1,18,26], and smarter supervision [10,11] have promoted the development of this direction. These action recognition approaches are based on trimmed videos, which are not suitable for untrimmed videos due to the considerable duration of the background. However, the pre-trained models on action recognition task can provide effective feature representation for temporal action localization task. In this paper, we use the I3D model [6], pre-trained on Kinetics [18], to extract video features.
Temporal action localization. Temporal action localization is a mirror problem of image object detection[30,29] in the temporal domain. The TAL task can be decomposed into proposal generation and classification stage, same as the two-stage approach of object detection. Recent methods for proposal generation are divided into two branches, top-down and bottom-up fashions. Top-down approaches [3,4,9,13,32,8,39,7] generated proposals with pre-defined regularly distributed segments then evaluated the confidence of each proposal. The boundary of top-down proposals are not flexible, and these generation strategies often cause extensive false positive proposals, which will introduce burdens in the classification stage. However, the other bottom-up approaches alleviated this problem and achieved the new state-of-the-art. TAG [38] was an early study of bottom-up fashion, which used frame-level action probabilities to group action proposals. Lin *et al.* proposed the multi-stage BSN [23] and end-to-end BMN [22] models via locating temporal boundaries to generate action proposals. *Gong al.* [15] also predicted action probabilities to generate action proposals from the perspective of multi scales. Zeng *et al.* proposed the PGCN [41] to model the proposal-proposal relations based on bottom-up proposals. Combined top-down and bottom-up fashions, Liu *et al.* proposed a MGG [25] model, which takes advantage of frame-level action probability as well. [40] is relevant to out study that enforced the temporal structure by maximizing the top-K summation of the confidence scores of the starting, continuing, and ending.

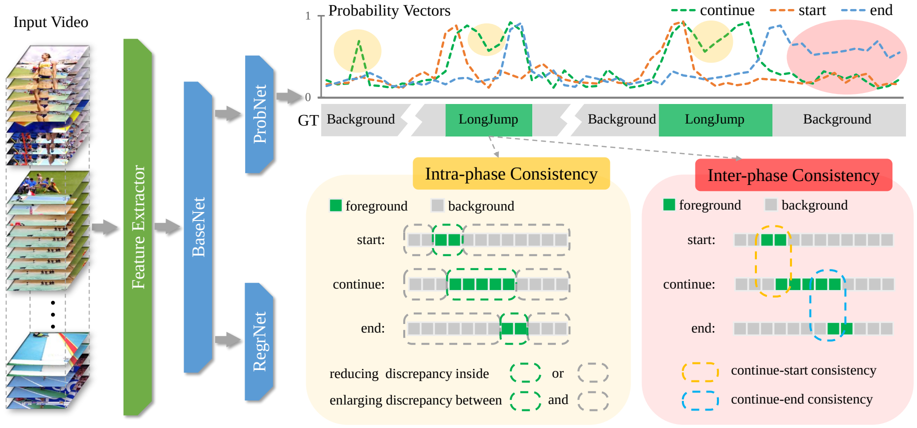

Fig. 1. Schematic of our approach. Three probability phases are predicted by the ProbNet. Intra-phase Consistency loss is built inside each phase by first separating *positive* and *negative* regions, then reduce the discrepancy inside *positive* or *negative*, and enlarge the discrepancy between *positive* and *negative*. Inter-phase Consistency loss is built between the continue-start phase and the continue-end phase.

3 Method

3.1 Problem and Baseline

Notations. Given an Untrimmed video, we denote $\{\mathbf{f}_t\}_{t=1}^T$ as a feature sequence to represent a video, where T is the length of the video and $\mathbf{f}_t$ is the t-th feature vector extracted from continuous RGB frames and optical flows. Annotations are $\varphi = \{(t_{s,n}, t_{e,n}, a_n)\}_{n=1}^N$, where $t_{s,n}$, $t_{e,n}$, and a_n are start time, end time, and class label of the action instance n. N is the number of action annotations. Following previous studies [23,22,25,15], we predict continuing, starting, and ending probability vectors $\mathbf{p}^{\mathrm{C}} \in [0,1]^T$, $\mathbf{p}^{\mathrm{S}} \in [0,1]^T$, and $\mathbf{p}^{\mathrm{E}} \in [0,1]^T$ to generate action proposals. Correspondingly, the ground-truth labels are generated via φ, which are notated by $\mathbf{g}^{\mathrm{C}} \in \{0,1\}^T$, $\mathbf{g}^{\mathrm{S}} \in \{0,1\}^T$, and $\mathbf{g}^{\mathrm{E}} \in \{0,1\}^T$, respectively. Continuing ground-truth $\mathbf{g}^{\mathrm{C}}$ has value "1" inside the action instances $[t_{s,n}, t_{e,n}]$, while starting and ending points are expanded to a region $[t_{s,n} - \delta_n, t_{s,n} + \delta_n]$ and $[t_{e,n} - \delta_n, t_{e,n} + \delta_n]$ to assign the ground-truth label $\mathbf{g}^{\mathrm{S}}$ and $\mathbf{g}^{\mathrm{E}}$. δ_n is set to be 0.1 duration of the action instance n, same as [23,22,15].

Baseline. This paper takes the typical bottom-up TAL framework as our baseline, such as BSN [23]. As illustrated in Figure 1, the baseline network is trained without Intra-phase Consistency and Inter-phase Consistency. We first use 3D convolutional network to extract video features $\{\mathbf{f}_t\}_{t=1}^T$, then feed the feature sequence to several 1D convolutional networks to (i) predict three probability vectors ($\mathbf{p}^{\mathrm{C}}$, $\mathbf{p}^{\mathrm{S}}$, and $\mathbf{p}^{\mathrm{E}}$) by ProbNet, (ii) predict the starting and ending boundary offsets ($\hat{\mathbf{o}}_{\mathrm{S}}$ and $\hat{\mathbf{o}}_{\mathrm{E}}$) by RegrNet. Finally, we generate proposals by combining start-end pairs with high probabilities and classify these candidate proposals.

3.2 Motivation: Avoiding Ambiguity with Temporal Consistency

The first and fundamental procedure in bottom-up TAL is to predict frame-level probabilities of three action-indicating phases, *i.e.* starting, continuing, and ending. Existing approaches use frame-level labels, $\mathbf{g}^{\mathrm{C}} \in \{0,1\}^T$, $\mathbf{g}^{\mathrm{S}} \in \{0,1\}^T$, and $\mathbf{g}^{\mathrm{E}} \in \{0,1\}^T$ to train three binary classification tasks. Since the meaning of "starting", "continuing", and "ending" have certain ambiguity, it is hard to determine the accurate time that an action starts, ends, and continues. Moreover, we find that even in training set, the **False Alarm** of these binary classification tasks reaches 68%, 64%, and 28% for starting, ending, and continuing, respectively. As shown in Figure 1, we can also observe that the continuing phase in green are not stable inside an action instance "LongJump" or background (*yellow* circles); and different action phases are not support each other (*red* circle). Thus, only supervised by classification labels is hard to optimize these problem, because there is no guarantee that every frame contains sufficient information of being correctly classified.

Therefore, to better regularize the learning process of avoiding ambiguity, we propose two consistency regularization terms during an end-to-end optimization, that consider the relations between different temporal locations inside each probability phase, named **Intra-phase Consistency** (IntraC) and the relations among different probability phases, named **Inter-phase Consistency** (InterC).

3.3 Adding Mutual Regularization

As illustrated in Figure 1, we add two consistency losses, IntraC and InterC, to regularize the learning process. IntraC is built inside each phase by first separating *positive* and *negative* regions, then reduce the discrepancy inside *positive* or *negative*, and enlarge the discrepancy between *positive* and *negative.* InterC performs consistency among three phases, which operates between continuing-starting and continuing-ending, **(i)** if there were an abrupt rise in the continuing phase, the starting phase should give a high probability, and vise versa; **(ii)** if there were an abrupt drop in the continuing phase, the ending phase should give a high probability, and vise versa.

Intra-phase Consistency. We build our Intra-phase Consistency loss inside each per-frame probability phase of start, end, and continuing. Firstly, we show the detailed operations for continuing phase $\mathbf{p}^{\mathrm{C}}$. The yellow block in Figure 1 shows an example of the IntraC on continuing phase $\mathbf{p}^{\mathrm{C}}$. To make the per-frame predictions supervised by their context predictions, we first define the *positive* and *negative* regions. The *positive* regions are defined as the locations where action continues by $g_t^{\mathrm{C}} = 1$, and the *negative* regions are the rest of the time where $g_t^{\mathrm{C}} = 0$. In terms of the division of the *positive* and *negative* region, the predicted continuing probabilities $\{p_t^{\mathrm{C}}\}_{t=1}^T$ are divided into a positive set $\mathcal{U}^{\mathrm{C}} = \{p_t^{\mathrm{C}} \mid g_t^{\mathrm{C}} = 1\}$ and a negative set $\mathcal{V}^{\mathrm{C}} = \{p_t^{\mathrm{C}} \mid g_t^{\mathrm{C}} = 0\}$. To make each prediction is not only supervised by its own label but other context labels, we optimize this problem by **(i)** $\min f(p_i^{\mathrm{C}}, p_j^{\mathrm{C}}), \forall p_i^{\mathrm{C}} \in \mathcal{U}^{\mathrm{C}}.\forall p_j^{\mathrm{C}} \in \mathcal{U}^{\mathrm{C}}$ **(ii)** $\max f(p_i^{\mathrm{C}}, p_j^{\mathrm{C}}), \forall p_i^{\mathrm{C}} \in \mathcal{U}^{\mathrm{C}}.\forall p_j^{\mathrm{C}} \in \mathcal{V}^{\mathrm{C}}$, where f is a distance function (l_1 distance in our experiments) to measure the difference between p_i^{C} and p_j^{C}. Therefore, the IntraC on continuing probability phase $\mathbf{p}^{\mathrm{C}}$ is formulated in Eq. (1):

$$\mathcal{L}_{\mathrm{Intra}^{\mathrm{C}}} = \frac{1}{N_{\mathrm{U}}}\sum_{i,j}(\mathbf{A}\odot\mathbf{M}_{\mathrm{U}})_{i,j} + \frac{1}{N_{\mathrm{V}}}\sum_{i,j}(\mathbf{A}\odot\mathbf{M}_{\mathrm{V}})_{i,j} + (1 - \frac{1}{N_{\mathrm{UV}}}\sum_{i,j}(\mathbf{A}\odot\mathbf{M}_{\mathrm{UV}})_{i,j}), \tag{1}$$

where $\mathbf{A} \in [0,1]^{T\times T}$ is an adjacency matrix to establish the relationship between predicted probabilities by measuring the distance between them. The elements in $\mathbf{A}$ are formulated as $a_{i,j} = f(p_i^{\mathrm{C}}, p_j^{\mathrm{C}})$. $\mathbf{M}_{\mathrm{U}}$, $\mathbf{M}_{\mathrm{V}}$, and $\mathbf{M}_{\mathrm{UV}} \in \{0,1\}^{T\times T}$ are three masks to select the corresponding pairs $a_{i,j}$ in adjacency matrix $\mathbf{A}$ from $\mathcal{U}^{\mathrm{C}}$ set, $\mathcal{V}^{\mathrm{C}}$ set, and between $\mathcal{U}^{\mathrm{C}}$ and $\mathcal{V}^{\mathrm{C}}$ sets, respectively. The constants N_{U}, N_{V}, and N_{UV} represent the number of "1" in each mask matrix. $\odot$ stand for the element-wise product.

Following this intra consistency between different frame-predictions, we reduce the discrepancy inside *positive* or *negative*, and enlarge the discrepancy between them. Replicating IntraC loss on continuing phase, we can also obtain the $\mathcal{L}_{\mathrm{IC}^{\mathrm{S}}}$ and $\mathcal{L}_{\mathrm{IC}^{\mathrm{E}}}$. Hence, the whole IntraC loss is formulated in Eq. (2):

$$\mathcal{L}_{\mathrm{Intra}} = \mathcal{L}_{\mathrm{Intra}^{\mathrm{C}}} + \mathcal{L}_{\mathrm{Intra}^{\mathrm{S}}} + \mathcal{L}_{\mathrm{Intra}^{\mathrm{E}}}. \tag{2}$$

Inter-phase Consistency. We build our Inter-phase Consistency loss between three probability phases, continuing phase $\mathbf{p}^{\mathrm{C}}$, starting phase $\mathbf{p}^{\mathrm{S}}$, and ending phase $\mathbf{p}^{\mathrm{E}}$. To make the consistency between these probability phases, we propose two hypotheses, (i) if there were an abrupt rise in the continuing phase, the starting phase should give a high probability, and vise versa; (ii) if there were an abrupt drop in the continuing phase, the ending phase should give a high probability, and vise versa. Following these hypotheses, we use the first difference term of $\mathbf{p}^{\mathrm{C}}$ to capture the abrupt rise and drop of the continuing probability phase: $\Delta\mathbf{p}^{\mathrm{C}} = p_{t+1}^{\mathrm{C}} - p_t^{\mathrm{C}}$.

As illustrated in red block of Figure 1, we build two kinds of constraints for InterC, the continue-start constraint in yellow circle and the continue-end constraint in blue circle. We use the positive values in $\Delta\mathbf{p}^{\mathrm{C}}$ to represent continuing probability rise rate, notated as $p_t^+ = \max\{0, \Delta p_t^{\mathrm{C}}\}$, and use negative values in $\Delta\mathbf{p}^{\mathrm{C}}$ to represent continuing probability drop rate, notated as $p_t^- = -\min\{0, \Delta p_t^{\mathrm{C}}\}$. Thus, to make predictions of continuing, starting, and ending support each other, we optimize this problem by **(i)** $\min f(p_t^+, p_t^{\mathrm{S}})$ and **(ii)** $\min f(p_t^-, p_t^{\mathrm{E}})$, where f is a distance function (l_1 distance in our experiments) to measure the distance. Then the InterC is formulated in Eq. (3):

$$\mathcal{L}_{\mathrm{Inter}} = \frac{1}{T}\sum_{t=1}^{T} \mid p_t^+ - p_t^{\mathrm{S}} \mid + \mid p_t^- - p_t^{\mathrm{E}} \mid . \tag{3}$$

Loss function. Predicting continuing, starting, and ending probabilities are trained with the cross-entropy loss. We separate the calculation by the *positive* and *negative* regions; then mix them with a ratio of 1:1 to balance the proportion of the *positive* and the *negative*. The loss of predicting the continuing probability is formulated in Eq. (4):

$$\mathcal{L}_{\mathrm{C}} = \frac{1}{T_{\mathrm{C}}^{+}} \sum_{t \in \mathcal{U}^{\mathrm{C}}} \ln(p_t^{\mathrm{C}}) + \frac{1}{T_{\mathrm{C}}^{-}} \sum_{t \in \mathcal{V}^{\mathrm{C}}} \ln(1 - p_t^{\mathrm{C}}), \tag{4}$$

where $\mathcal{U}^{\mathrm{C}}$ and $\mathcal{V}^{\mathrm{C}}$ denote the *positive* and *negative* set in $\mathbf{p}^{\mathrm{C}}$, while T_{C}^{+} and T_{C}^{-} are the number of them, respectively. Replacing the script "C" with "S" or "E" in Eq. (4), we can obtain the $\mathcal{L}_{\mathrm{S}}$ and $\mathcal{L}_{\mathrm{E}}$, respectively. Hence, the whole classification loss is formulated as: $\mathcal{L}_{\mathrm{cls}} = \mathcal{L}_{\mathrm{C}} + \mathcal{L}_{\mathrm{S}} + \mathcal{L}_{\mathrm{E}}$.

To make the action boundaries more precise, we also introduce a regression task to predict the starting and ending boundary offsets. Inspired by some object detection studies [30,20], we apply SmoothL1 Loss [14] (SL_1) to our regression task, which is formulated in Eq. (5):

$$\mathcal{L}_{\mathrm{reg}} = \frac{1}{T_{\mathrm{S}}^{+}} \sum_{t \in \mathcal{U}^{\mathrm{S}}} \mathrm{SL}_1(o_t^{\mathrm{S}}, \hat{o}_t{}^{\mathrm{S}}) + \frac{1}{T_{\mathrm{E}}^{+}} \sum_{t \in \mathcal{U}^{\mathrm{E}}} \mathrm{SL}_1(o_t^{\mathrm{E}}, \hat{o}_t{}^{\mathrm{E}}), \tag{5}$$

where $\mathcal{U}^{\mathrm{S}}$ and $\mathcal{U}^{\mathrm{E}}$ are the *positive* regions in $\mathbf{p}^{\mathrm{S}}$ and $\mathbf{p}^{\mathrm{E}}$. T_{S}^{+} and T_{E}^{+} are the number of them. $\hat{o}_t{}^{\mathrm{S}}$ and $\hat{o}_t{}^{\mathrm{E}}$ are the predicted starting and ending offsets with their ground-truth (o_t^{S} and o_t^{E}). Adding our proposed consistency constrains IntraC and InterC, the overall objective loss function is formulated in Eq. (6):

$$\mathcal{L} = \mathcal{L}_{\mathrm{cls}} + \mathcal{L}_{\mathrm{reg}} + \mathcal{L}_{\mathrm{Intra}} + \mathcal{L}_{\mathrm{Inter}}. \tag{6}$$

3.4 Inference: Proposal Generation and Classification

Following the same rules in BSN [23] and ScaleMatters [15], we select the starting and ending points in terms of $\mathbf{p}^{\mathrm{S}}$ and $\mathbf{p}^{\mathrm{E}}$; then combine them to generate action proposals; finally rank these proposals and classify them with action labels. Operations are conducted sequentially:

Proposal generation. To generate action proposals, we first select the candidate starting and ending points with predicted $\mathbf{p}^{\mathrm{S}}$ and $\mathbf{p}^{\mathrm{E}}$ by two rules [23]: (i) start points t where $p_t^{\mathrm{S}} > 0.5 \times (\max_{t=1}^{T}\{p_t^{\mathrm{S}}\} + \min_{t=1}^{T}\{p_t^{\mathrm{S}}\})$; (ii) start points t where $p_{t-1}^{\mathrm{S}} < p_t^{\mathrm{S}} < p_{t+1}^{\mathrm{S}}$. The ending points are selected by the same rules. Following these two rules, we obtain starting and ending candidates which have high probability or stay at a peak position. Combining these points under a maximum action duration in training set, we obtain the candidate proposals.

Proposal ranking. To rank action proposals with a confidence score, we provide two methods: (i) directly use the product of the starting and ending probabilities, $p_{t_s}^{\mathrm{S}} \times p_{t_e}^{\mathrm{E}}$. (ii) train an additional evaluation network to score candidate proposals [15], which is noted as $\phi(t_s, t_e)$. The detailed information can be found in [15]. Thus, the final confidence score for candidate proposals is $p_{t_s}^{\mathrm{S}} \times p_{t_e}^{\mathrm{E}} \times \phi(t_s, t_e)$.

Redundant proposal suppression. After generating candidate proposals with the confidence score, we need to remove redundant proposals with high overlaps.

Standard method such as soft non-maximum suppression (Soft-NMS) [2] is used in our experiments. Soft-NMS decays the confidence score of proposals which are highly overlapped. Finally, we suppress the redundant proposals to achieve a higher recall.

Proposal classification. The last step of temporal action localization is to classify the candidate proposals. For fair comparison with other temporal localization methods, we use the same classifiers to report our action localization results. Following BSN [23], we use video-level classifier in UntrimmedNet [36] for THUMOS14 dataset. As for ActivityNet1.3 dataset, we use the video-level classification results generated by [42].

3.5 Implementation Details

Network Design. We build our IntraC and InterC on a succinct baseline model with all 1D Convolution layers and the detailed network architecture is shown in Table 1. The input of BaseNet is extracted feature sequence $\{\mathbf{f}_t\}_{t=1}^{T}$ of untrimmed videos. Since untrimmed videos have various video length, we truncate or pad zeros to obtain a fixed length features of window l_w. Through BaseNet, the output features are shared by three 2-layer ProbNets to predict probability phases ($\mathbf{p}^\mathrm{C}$, $\mathbf{p}^\mathrm{S}$, and $\mathbf{p}^\mathrm{E}$) and two RegrNets to predict starting and ending boundary offsets ($\hat{\mathbf{o}}_\mathrm{S}$ and $\hat{\mathbf{o}}_\mathrm{E}$).

Network training. Our BaseNet, ProbNet, and RegrNet are jointly trained from scratch by multiple losses which are the classification loss ($\mathcal{L}_\mathrm{cls}$), regression loss ($\mathcal{L}_\mathrm{reg}$) and consistency losses ($\mathcal{L}_\mathrm{Intra}$ and $\mathcal{L}_\mathrm{Inter}$). We find setting the ratio of each loss component equal get relatively proper numerical values and the loss curve can converge well. As mentioned previous, to contain most action instances in a fixed observed window, the input feature length of window l_w is set to be 750 for THUMOS14 and scaled to be 100 for ActivityNet1.3. The training process lasts for 20 epochs with a learning rate of 10^{-3} in former 10 epochs and 10^{-4} in latter 10 epochs. The batch size is set to be 3 for THUMOS14 and 16 for the ActivityNet1.3. We use a SGD optimization method with a momentum of 0.9 to train both datasets. In Section 3.4, the additional evaluation network for proposal ranking follows the same settings in [15].

Table 1. The detailed network architecture. The output of BaseNet is shared by ProbNet and RegrNet. Three ProbNets (× 3) are used to predict continuing, starting, and ending probability phases. Two RegrNets (× 2) are used to predict starting and ending offsets.

Name	Layer	Kernel	Channels	Activation
BaseNet	Conv1D	9	512	ReLU
	Conv1D	9	512	ReLU
ProbNet	Conv1D	5	256	ReLU
(× 3)	Conv1D	5	1	Sigmoid
RegrNet	Conv1D	5	256	ReLU
(× 2)	Conv1D	5	1	Identity

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets and features. We validate our proposed IntraC and InterC on two standard datasets: **THUMOS14** includes 413 untrimmed videos with 20 action classes. According to the public split, 200 of them are used for training, and 213 are used for testing. There are more than 15 action annotations in each video; **ActivityNet1.3** is a more considerable action localization dataset with 200 classes annotated. The entire $19,994$ untrimmed videos are divided into training, validation, and testing sets by ratio 2:1:1. Each video has around 1.5 action instances. To make a fair comparison with the previous work, we use the same two-stream features of these datasets. The two-stream features, which are provided by [24], are extracted by I3D network [6] pre-trained on Kinetics.
Metric for temporal action proposals. To evaluate the quality of action proposals, we use conventional metrics Average Recall (AR) with different Average Number (AN) of proposals AR@AN for action proposals. On THUMOS14 dataset, the AR is calculated under multiple IoU threshold set from 0.5 to 1.0 with a stride of 0.05. As for ActivityNet1.3 dataset the multiple IoU threshold are from 0.5 to 0.95 with a stride of 0.05. Besides, we also use the area under the AR-AN curve (AUC) to evaluate the performance.
Metric for temporal action localization. To evaluate the performance of action localization, we use mean Average Precision (mAP) metric. On THUMOS14 dataset, we report the mAP with multiple IoUs in set $\{0.3, 0.4, 0.5, 0.6, 0.7\}$. As for ActivityNet1.3 dataset, the IoU set is $\{0.5, 0.7, 0.95\}$. Moreover, we also report the averaged mAP where the IoU is from 0.5 to 0.95 with a stride of 0.05.

4.2 Comparison to the State-of-the-arts

Temporal action proposals. We compare the temporal action proposals generated by our IntraC and InterC equipped model on THUMOS14 and ActivityNet1.3 dataset. As illustrated in Table 2, comparing with previous works, we can achieve the best performance especially on AR@50 metric. Our consistency losses help to generate more precise candidate starting and ending points, so we can achieve a high recall with fewer proposals. In Table 4, we also achieve comparable results on ActivityNet1.3, since it is a well studied dataset.
Temporal action localization. Classifying the proposed proposals, we obtain the final localization results. As illustrated in Table 3 and Table 5, our method outperforms the previous studies. Especially at **high IoU settings**, we achieve significant improvements since our consistency loss can make the boundaries more precise. On THUMOS14 dataset, the mAP at IoU of 0.6 is improved from 31.5% to 38.0% and the mAP at IoU of 0.7 is improved from 21.7% to 28.5%. On ActivityNet1.3 dataset, we can achieve the mAP to 9.21% at IoU of 0.95.
Generalizing IntraC&InterC to Other Algorithms. Our proposed two consistency losses, *i.e.*, IntraC and InterC, are effective in generating the probability phases of continuing, starting, and ending. To prove these consistency

Table 2. Comparisons in terms of AR@AN (%) on THUMOS14.

Method	@50	@100	@200
TAG [38]	18.55	29.00	39.41
CTAP [12]	32.49	42.61	51.97
BSN [23]	37.46	46.06	53.21
BMN [22]	39.36	47.72	54.70
MGG [25]	39.93	47.75	54.65
TSA-Net [15]	42.83	49.61	54.52
Ours	**44.23**	**50.67**	**55.74**

Table 3. Comparisons in terms of mAP (%) on THUMOS14.

Method	0.3	0.4	0.5	0.6	0.7
SST [3]	41.2	31.5	20.0	0.9	4.7
TURN [13]	46.3	35.3	24.5	14.1	6.3
BSN [23]	53.5	45.0	36.9	28.4	20.0
MGG [25]	53.9	46.8	37.4	29.5	21.3
BMN [22]	**56.0**	47.4	38.8	29.7	20.5
TSA-Net [15]	53.2	48.1	41.5	31.5	21.7
Ours	53.9	**50.7**	**45.4**	**38.0**	**28.5**

Table 4. Comparisons in terms of AUC and AR@100 (%) on ActivityNet1.3.

Method	AUC	AR@100
TCN [8]	59.58	-
CTAP [12]	65.72	73.17
BSN [23]	66.17	74.16
MGG [25]	66.43	74.54
Ours	**66.51**	**75.27**

Table 5. Comparisons in terms of mAP (%) on ActivityNet1.3 (val). "Average" is caculated at the IoU of $\{0.5 : 0.05 : 0.95\}$.

Method	0.5	0.7	0.95	Average
CDC [31]	43.83	25.88	0.21	22.77
SSN [43]	39.12	23.48	5.49	23.98
BSN [23]	**46.45**	29.96	8.02	29.17
Ours	43.47	**33.91**	**9.21**	**30.12**

losses are valid for other network architecture and framework in TAL, we introduce them to TSA-Net [15] and PGCN [41], respectively. **TSA-Net** [15] designed a multi-scale architecture to predict probability phases of continuing, starting and ending. We introduce our IntraC and InterC to their multi-scale networks, TSA-Net-small, TSA-Net-medium, and TSA-Net-large, respectively. As illustrated in Table 6, our IntraC and InterC significantly outperforms the baseline models on all three network architectures. **PGCN** [41] explore the proposal-proposal relations using Graph Convolutional Networks [19] (GCN) to localize action instances. This framework builds upon the prepared proposals from BSN [23] method. We introduce our two consistency losses to generated candidate proposals for PGCN framework. As illustrated in Table 7, introducing IntraC and InterC to PGCN also improves the localization performance.

4.3 Ablation Studies

As mentioned in dataset description, THUMOS has 10 times action instances per video than ActivityNet (only has 1.5 action instances per video) and THUMOS video also contains a larger portion of background. More instances and more background are challenge for detection task. Thus we conduct following detailed ablation studies on THUMOS14 dataset to explore how these constrains, IntraC and InterC, improve the quality of temporal action proposals.

Effectiveness of IntraC. As illustrated in Table 8 "Intra Consistency", we compare the components of IntraC in terms of the AR@AN. The IntraC is

Table 6. Generalizing IntraC&InterC to multi-scale TSA-Net [15] in terms of AR@AN (%) on THUMOS14. ∗ indicates the results that are implemented by ours.

TSA-Net	AR@50	AR@100	AR@200
Small (Small*)	37.72 (38.32)	45.85 (46.15)	52.03 (52.39)
Small* + IntraC&InterC	**39.73**	**47.69**	**53.48**
Medium (Medium*)	37.77 (39.20)	45.01 (47.17)	50.38 (53.46)
Medium* + IntraC&InterC	**40.05**	**47.53**	**53.88**
Large (Large*)	36.07 (37.91)	44.28 (45.89)	50.80 (52.36)
Large* + IntraC&InterC	**39.68**	**47.47**	**53.50**

Table 7. Generalizing IntraC&InterC to PGCN [41] in terms of mAP (%) on THUMOS14. ∗ indicates the results that are implemented by ours.

Method	0.1	0.2	0.3	0.4	0.5
PGCN	69.50	67.80	63.60	57.80	49.10
PGCN*	69.26	67.76	63.73	58.82	48.88
PGCN* + IntraC&InterC	**71.83**	**70.31**	**66.29**	**60.99**	**50.10**

introduced to continuing probability phase ($\mathcal{L}_{\mathrm{Intra}^{\mathrm{C}}}$), starting probability phase ($\mathcal{L}_{\mathrm{Intra}^{\mathrm{S}}}$), and ending probability phase ($\mathcal{L}_{\mathrm{Intra}^{\mathrm{E}}}$). Compared with the baseline result without any consistency losses, introducing continuing $\mathcal{L}_{\mathrm{Intra}^{\mathrm{C}}}$ or starting $\mathcal{L}_{\mathrm{Intra}^{\mathrm{S}}}$ and ending $\mathcal{L}_{\mathrm{Intra}^{\mathrm{E}}}$ can both achieve better results. Combined all three IntraC losses, the AR@50 is improved from 39.02% to 41.91%.

Effectiveness of InterC. As illustrated in Table 8 “Inter Consistency”, we compare the components of InterC losses in terms of the AR@AN. The InterC is introduced between continue-start (C&S) and continue-end (C&E). InterC on C&S (C&E) makes the consistency between the starting phase (ending phase) and the derivative of continuing phase, which can suppress the false positives only observed from a single probability phase. Only introducing InterC to C&S or C&E obtains around 1% absolute improvement on AR@50. When combined C&S and C&E, it can improve 2.21% on AR@50.

Combining IntraC&InterC. As illustrated in Table 8 “All Consistency”, we compare the IntraC and InterC losses in terms of the AR@AN. Both the IntraC and InterC independently achieve more than 2% absolute improvement on AR@50. When combined IntraC and InterC, the AR@50 is improved from 39.02% to 42.63%. Consistency inside each probability phase and between them are coupled, which leads to a positive feedback. It means when we get the better probability phase that fits the IntraC settings, the potential constraint of InterC is more appropriate between three probability phases, and vise versa.

Effectiveness of kernel size and layers. The scale of the receptive field is crucial in temporal action localization tasks. So we explore different scales of receptive field by adjusting the number of layers and the kernel size of the

Table 8. Ablation studies on Intra-phase Consistency and Inter-phase Consistency in terms of AR@AN (%) on THUMOS14. The baseline model is define in Table 1.

			AR@50	AR@100	AR@200
Baseline			39.02	46.26	53.09
Continue	Start	End	Intra Consistency		
✓			40.46	47.85	53.87
	✓	✓	40.86	48.26	54.16
✓	✓	✓	41.91	49.06	54.82
C&S	C&E		Inter Consistency		
✓			40.21	47.30	53.38
	✓		40.64	47.85	54.01
✓	✓		41.23	48.81	54.47
IntraC	InterC		Intra&Inter Consistency		
✓			41.91	49.06	54.82
	✓		41.23	48.81	54.47
✓	✓		**42.63**	**49.85**	**55.32**

Table 9. Ablation studies on model structures in terms of AR@AN (%) on THUMOS14. All numbers are the averaged value in the last 10 epochs.

Layers	Kernel Size	AR@50	AR@100	AR@200
2	5	40.98	48.51	54.64
3	5	41.56	49.02	54.88
4	5	41.68	48.93	54.91
5	5	40.98	48.14	54.29
2	3	39.54	47.61	53.84
2	5	40.98	48.51	54.64
2	7	41.49	49.16	55.17
2	9	**42.63**	**49.85**	**55.32**
2	11	42.48	49.32	54.97
2	13	42.17	49.41	55.21

BaseNet. As illustrated in Table 9, we compare results between different kernel sizes and layers in terms of the AR@AN. Deeper layers and larger kernel sizes often lead to a better performance, but using too many layers and/or an over-large kernel size often incurs over-fitting. We also conduct the experiments using different layers with a kernel size of 9 and find that a 2-layer network performs best, so we use this option in the main experiments. This implies that probably increasing the depth is not the best choice here.

Effectiveness of proposal scoring. As mentioned in Section 3.4, we compare two methods for scoring proposals. Once we get proposals of an untrimmed video, a proper ranking method with convincing scores can achieve the high recall with fewer proposals. As illustrated in Table 10, we compare two scoring functions, $p^{\mathrm{S}}_{t_s} \times p^{\mathrm{E}}_{t_e}$ and $p^{\mathrm{S}}_{t_s} \times p^{\mathrm{E}}_{t_e} \times \phi(t_s, t_e)$. Directly using starting and ending probability at boundaries is simple and effective, however, training a new evalu-

Table 10. Ablation studies on proposal scoring in terms of AR@AN (%) on THUMOS14. Experiments are based on 2 “Layers” and 9 “Kernel Size” model in Table 9. All numbers are the averaged value in the last 10 epochs.

Proposal Scoring	AR@50	AR@100	AR@200
$p^{\mathrm{S}}_{t_s} \times p^{\mathrm{E}}_{t_e}$	42.63	49.85	55.32
$p^{\mathrm{S}}_{t_s} \times p^{\mathrm{E}}_{t_e} \times \phi(t_s, t_e)$	**44.23**	**50.67**	**55.74**

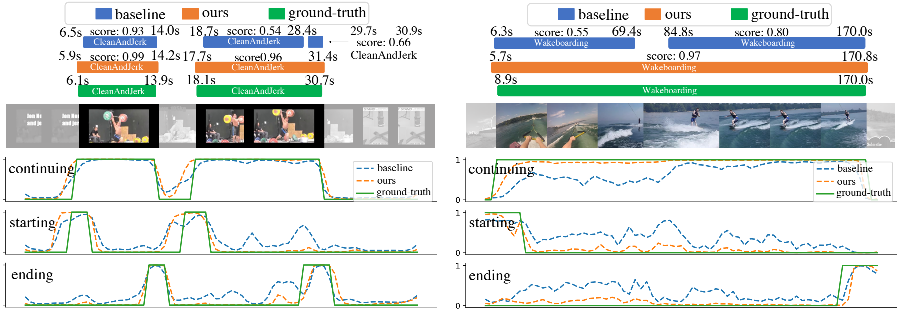

Fig. 2. Qualitative results on THUMOS14 (left) and ActivityNet1.3 (right) datasets. “green” lines are ground-truth, “blue” lines are predicted phases by baseline model and “orange” lines are optimized with IntraC and InterC regularization terms.

ation network [23,15] to evaluate the confidence of proposals can further improve the performance by a significant margin.

4.4 Visualization

As illustrated in Figure 2, we visualize some examples on both datasets. Comparing the predicted $\mathbf{p}^{\mathrm{C}}$, $\mathbf{p}^{\mathrm{S}}$, and $\mathbf{p}^{\mathrm{E}}$ with or without the IntraC and InterC regularization, we find our proposed IntraC and InterC indeed make each predicted phase becomes stable inside *foreground* and *background* regions. Besides, some false positives in $\mathbf{p}^{\mathrm{S}}$ and $\mathbf{p}^{\mathrm{E}}$ are suppressed by their context information, so that we can remove many candidate proposals of poor quality via these wrong starting and ending points. *e.g.*, the second action “CleanAndJerk” and the action “Wakeboarding” are separate by false positive starting point in baseline model. The visualization results show that only introducing binary classification labels is hard to optimize these probability phases, since it discards the potential constraints between the different temporal locations and action phases. We also perform regularization using the smoothness assumption, *i.e.*, using a Gaussian kernel to penalize local inconsistenies within $\mathbf{p}^{\mathrm{C}}$, $\mathbf{p}^{\mathrm{S}}$, and $\mathbf{p}^{\mathrm{E}}$. In experiments, this kinds of regularization does not necessarily push the positive scores to 1 and negative scores to 0, and we believe smoothness might be useful in the unsupervised or weakly-supervised TAL scenarios.

Table 11. Introducing oracle information to TAL in terms of mAP (%) on THUMOS14. O_{rank} is ground-truth rank information and O_{cls} uses ground-truth class label.

O_{rank}	O_{cls}	0.3	0.4	0.5	0.6	0.7
		53.9	50.7	45.4	38.0	28.5
	✓	57.1	53.2	47.3	39.3	29.5
✓		66.4	65.4	63.8	59.9	52.7
✓	✓	**72.1**	**70.9**	**68.8**	**64.1**	**55.6**

4.5 Discussion: the Upper Bounds of TAL

Most temporal action localization method can be divided into the following procedures, (i) generating proposals, (ii) ranking proposals, and (iii) classifying proposals. Which one is most awaiting to improve for the intending researchful keystone? We introduce two types of oracle information to reveal the performance gap between the different upper bounds. As illustrated in Table 11, O_{rank} means that each candidate proposal is ranked by the max IoU score with all ground-truth action instances. O_{cls} means that the ground-truth action labels are assigned to candidate proposals. When introducing O_{rank} or/and O_{cls} to our action localization baseline, it is worth to notice that proposal classification has been well solved since there is a small gap when introducing O_{cls}. However, when introducing the oracle ranking information O_{rank}, the upper bound can improve a lot from 53.9% to 66.4% in terms of mAP at IoU of 0.3. That means there is a significant untapped opportunity in how to rank the action proposals.

5 Conclusions

In this paper, we investigate the problem that frame-level probability phases of starting, continuing, and ending are not self-consistent in the bottom-up TAL approach. Our research reveals that state-of-the-art video analysis algorithms, though supervised with classification labels, mostly have a limited understanding in the temporal dimension, which can lead to undesired properties, *e.g.*, inconsistency or discontinuity. To alleviate this problem, we propose two consistency losses (IntraC and InterC) which can mutually regularize the learning process. Experiments reveal that our approach improves the performance of temporal action localization both quantitatively and qualitatively.

Our work reveals that introducing priors for self-regularization is important for learning from high-dimensional data (*e.g.*, videos). We will continue along this direction in the future, and explore the possibility of learning such priors from self-supervised data, *e.g.*, unlabeled videos.

Acknowledgements This work is supported by the National Key Research and Development Program of China (No. 2019YFB1804304), SHEITC (No. 2018-RGZN-02046), 111 plan (No. BP0719010), and STCSM (No. 18DZ2270700), and State Key Laboratory of UHD Video and Audio Production and Presentation.

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark. In: arXiv preprint arXiv:1609.08675 (2016)
2. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: Proceedings of the International Conference on Computer Vision (ICCV). pp. 5561–5569 (2017)
3. Buch, S., Escorcia, V., Shen, C., Ghanem, B., Carlos Niebles, J.: Sst: Single-stream temporal action proposals. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2911–2920 (2017)
4. Caba Heilbron, F., Carlos Niebles, J., Ghanem, B.: Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1914–1923 (2016)
5. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 961–970 (2015)
6. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6299–6308 (2017)
7. Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the faster r-cnn architecture for temporal action localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1130–1139 (2018)
8. Dai, X., Singh, B., Zhang, G., Davis, L.S., Qiu Chen, Y.: Temporal context network for activity localization in videos. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 5793–5802 (2017)
9. Escorcia, V., Heilbron, F.C., Niebles, J.C., Ghanem, B.: Daps: Deep action proposals for action understanding. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 768–784. Springer (2016)
10. Gan, C., Sun, C., Duan, L., Gong, B.: Webly-supervised video recognition by mutually voting for relevant web images and web video frames. In: European Conference on Computer Vision (2016)
11. Gan, C., Yao, T., Yang, K., Yang, Y., Mei, T.: You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
12. Gao, J., Chen, K., Nevatia, R.: Ctap: Complementary temporal action proposal generation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 68–83 (2018)
13. Gao, J., Yang, Z., Chen, K., Sun, C., Nevatia, R.: Turn tap: Temporal unit regression network for temporal action proposals. In: Proceedings of the International Conference on Computer Vision (ICCV). pp. 3628–3636 (2017)
14. Girshick, R.: Fast r-cnn. In: Proceedings of the International Conference on Computer Vision (ICCV). pp. 1440–1448 (2015)
15. Gong, G., Zheng, L., Bai, K., Mu, Y.: Scale matters: Temporal scale aggregation network for precise action localization in untrimmed videos. In: International Conference on Multimedia and Expo (ICME). pp. 1–6 (2020)

16. Jiang, Y.G., Liu, J., Zamir, A.R., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: Thumos challenge: Action recognition with a large number of classes (2014)
17. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1725–1732 (2014)
18. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. In: arXiv preprint arXiv:1705.06950 (2017)
19. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR). pp. 1–14 (2017)
20. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 734–750 (2018)
21. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: ICCV (2019)
22. Lin, T., Liu, X., Li, X., Ding, E., Wen, S.: Bmn: Boundary-matching network for temporal action proposal generation. In: Proceedings of the International Conference on Computer Vision (ICCV) (2019)
23. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: Bsn: Boundary sensitive network for temporal action proposal generation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–19 (2018)
24. Liu, D., Jiang, T., Wang, Y.: Completeness modeling and context separation for weakly supervised temporal action localization. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1298–1307 (2019)
25. Liu, Y., Ma, L., Zhang, Y., Liu, W., Chang, S.F.: Multi-granularity generator for temporal action proposal. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3604–3613 (2019)
26. Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S.A., Yan, Y., Brown, L., Fan, Q., Gutfreund, D., Vondrick, C., et al.: Moments in time dataset: one million videos for event understanding. In: IEEE transactions on Pattern Analysis and Machine Intelligence (T-PAMI). IEEE (2019)
27. Peisen, Z., Lingxi, X., Ya, Z., Qi, T.: Universal-to-specific framework for complex action recognition. In: arXiv preprint arXiv:2007.06149 (2020)
28. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 5534–5542. IEEE (2017)
29. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 779–788 (2016)
30. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 91–99 (2015)
31. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.F.: Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5734–5743 (2017)
32. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage cnns. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1049–1058 (2016)

33. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 4489–4497. IEEE (2015)
34. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6450–6459 (2018)
35. Wang, L., Li, W., Li, W., Van Gool, L.: Appearance-and-relation networks for video classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1430–1439 (2018)
36. Wang, L., Xiong, Y., Lin, D., Van Gool, L.: Untrimmednets for weakly supervised action recognition and detection. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4325–4334 (2017)
37. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 305–321 (2018)
38. Xiong, Y., Zhao, Y., Wang, L., Lin, D., Tang, X.: A pursuit of temporal accuracy in general activity detection. In: arXiv preprint arXiv:1703.02716 (2017)
39. Xu, H., Das, A., Saenko, K.: R-c3d: Region convolutional 3d network for temporal activity detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 5783–5792 (2017)
40. Yuan, Z., Stroud, J.C., Lu, T., Deng, J.: Temporal action localization by structured maximal sums. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3684–3692 (2017)
41. Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C.: Graph convolutional networks for temporal action localization. In: Proceedings of the International Conference on Computer Vision (ICCV) (2019)
42. Zhao, Y., Zhang, B., Wu, Z., Yang, S., Zhou, L., Yan, S., Wang, L., Xiong, Y., Lin, D., Qiao, Y., et al.: Cuhk & ethz & siat submission to activitynet challenge 2017. In: arXiv preprint arXiv:1710.08011 (2017)
43. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 2914–2923 (2017)