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 involve predicting three values at each time point, corresponding to the probabilities that the action starts, continues and ends, and post-processing these curves for the final localization. This paper delves deep into this mechanism, and argues that existing approaches mostly ignored the potential relationship of these curves, and results in low quality of action proposals. To alleviate this problem, we add extra constraints to these curves, e.g., the probability of “action continues” should be relatively high between probability peaks of “action starts” and “action ends”, so that the entire framework is aware of these latent 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 (in terms of the AR@AN and mAP) and qualitatively (the curves in the testing stage become much smoother). In particular, when we build our constraints beyond TSA-Net and PGCN, we achieve the state-of-the-art performance especially at strict high IoU settings. The code will be available.

1. Introduction

Videos are generated, stored, and transmitted every day. In the face of extensive video data, automatic video content analysis is awaiting to be explored in both academia and industry. Temporal Action Localization (TAL) aiming at locating action instances is a fundamental task in video content analysis. The problem definition of TAL is to find a set of action instances with their start times, end times, and action labels in untrimmed videos.

Analogous to anchor-based object detection, TAL can be divided into temporal action proposal and action classification stages. The latter is relatively well studied since recent action classifiers have achieved cogent performance [6, 22, 28, 31, 29]. But TAL still suffers a low performance in standard benchmarks [12, 5]. Thus, generating precise action proposals is crucial in TAL. Early approaches for generating action proposals are summarized as a top-down fashion, which first generates regularly distributed proposals (e.g., multi-scale sliding windows), then evaluates the confidence of them. This kind of methods [4] often over-generate candidate proposals, and the boundary of the proposals are not flexible. To solve this problem, the other kind of bottom-up approach has been proposed. Beginning in TAG [32] and prospering in BSN [18], typical bottom-up methods [18, 17, 20, 11] first densely evaluate the frame-level probabilities of action starting, continuing, and ending; then collect candidate starting and ending points to group action proposals. This design paradigm can generate flexible action proposals and achieve a high recall.
with fewer proposals [32], which becomes a better practice in temporal action proposals.

However, predicting these probability curves is under-explored in the bottom-up approach. As illustrated in Figure 1, existing methods evaluate these probability curves only through a frame-level classification task with the supervision of foreground and background labels. So we find some unreasonable phenomena, (i) the predicted continuing probabilities (green) are not always staying at a high value during foreground “LongJump” and not always staying at a low value during “Background”, which are highlighted by yellow circles; (ii) the red circle shows that three predicted curves can not support each other since the ending curve runs away. These diverged predictions mainly comes from the separate classification task and will introduce noisy candidate proposals. However, these can be easily constrained by their context in temporal-level and their trends in curve-level, e.g., (i) the probabilities inside foreground and background should be stable; (ii) the probability of “action continues” should be relatively high between probability peaks of “action starts” and “action ends”.

In this paper, we address this problem by adding extra constraints to these curves, so that the entire framework is aware of these latent constraints during an end-to-end optimization process. Like previous studies, we also predict three probability curve to represent an untrimmed video, starting curve, continuing curve, and ending curve. Each curve is divided into the foreground and background regions by ground-truth annotations. We propose two consistency constraints, which are named Internal Constraint (IC) and External Constraint (EC), to regularize the learning process. IC is built inside each probability curve that enforces (i) the probabilities in each foreground or background regions are consistent; (ii) the probabilities between foreground and background regions are separate. EC performs consistency constraint among three probability curves, which adds the constraints between continue-start curve and continue-end curve, (i) if there were an abrupt rise in the continuing curve, the starting curve should give a high probability, and vise versa; (ii) if there were an abrupt drop in the continuing curve, the ending curve should give a high probability, and vise versa.

We perform our experiments on two public datasets THUMOS14 and ActivityNet1.3. Experimental results illustrate that our approach clearly outperforms the baseline 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. In particular, when we build our constraints beyond other network (TSA-Net [11]) or framework (PGCN [33]), we achieve a better performance.

2. Related Work

Action recognition. Same as image recognition in image analysis, action recognition is a fundamental task in video domain. Extensive models [27, 6, 22, 28, 31, 29] on action recognition have been widely studied. The deeper models [6, 22] and more massive datasets [13, 11, 14, 21] promote 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 [14], to extract video features.

Temporal action localization. Temporal action localization is a mirror problem of image object detection [24, 23] 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, 7, 9, 26] 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 [32] 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 [18] and end-to-end BMN [17] models via locating temporal boundaries to generate action proposals. Gong et al. also predicted action probabilities to generate action proposals from the perspective of multi scales. Zeng et al. proposed the PGCN [33] to model the proposal-proposal relations based on bottom-up proposals. Combined top-down and bottom-up fashions, Liu et al. proposed a MGG [20] model, which takes advantage of frame-level action probability as well.

3. Method

3.1. Problem and Notations

Given an Untrimmed video, we denote \( \{ f_t \}_{t=1}^{T} \) as a feature sequence to represent a video, where \( T \) is the length of the video and \( f_t \) is the \( t \)-th feature vector extracted from a period of continuous RGB frames and optical flows. Annotation of action instances can be denoted as \( \varphi = \{(t_{s,n}, t_{e,n}, a_n)\}_{n=1}^{N} \), where \( t_{s,n}, t_{e,n}, \) and \( a_n \) are start time, end time, and class label, respectively, of the action instance \( n \). The constant \( N \) is the number of action annotations. Following many previous studies [18, 17, 20, 11],...
Input Video

Figure 2. Schematic of our approach. Three probability curves are predicted by the ProbNet. Internal Constraint is built inside each curve by first separating foreground and background regions, then reduce the discrepancy inside foreground or background, and enlarge the discrepancy between foreground and background. External Constraint is built between the continue-start curve and the continue-end curve.

we predict a continuing probability curve $p^C \in [0, 1]^T$, a starting probability curve $p^S \in [0, 1]^T$, and an end probability curve $p^E \in [0, 1]^T$ to generate action proposals. Correspondingly, the ground-truth supervisions are generated via $\varphi$, which are notated by $g^C \in \{0, 1\}^T$, $g^S \in \{0, 1\}^T$, and $g^E \in \{0, 1\}^T$, respectively. Continuing ground-truth $g^C_t$ 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 $g^S$ and $g^E$. $\delta_n$ is set to be 0.1 duration of the action instance $n$, same as [18][17][11].

3.2. Baseline and Motivation

This paper takes the state-of-the-art bottom-up framework as our baseline for TAL, such as BSN [18], BMN [17], and Scale Matters [11]. As illustrated in Figure 2, we first use 3D convolutional network to extract video features $\{f_i\}_{i=1}^T$, then feed the feature sequence to several 1D convolutional networks to (i) predict three probability curves ($p^C$, $p^S$, and $p^E)$ by ProbNet, (ii) predict the starting and ending boundary offsets ($\delta_S$ and $\delta_E$) by RegrNet. Finally, we generate proposals by combining start-end pairs with high probabilities and classify these candidate proposals.

As shown in Figure 2, the first and fundamental procedure in bottom-up TAL is to predict these probability curves. But there are some common problems of predicting probability curves based on the baseline model. The continuing probability curve in green are not always staying at a high value inside an action instance, and not always staying at a low value during background, which are highlighted by yellow circles. The red circle shows that three probability curves can not support each other since the ending curve runs away. Hence, these diverged predictions will introduce noisy candidate proposals. This is mainly because the supervision of the learning process only comes from a temporal-separate classification task, i.e., three binary cross-entropy losses for action starting, continuing and ending, without utilizing the relations between different time steps and potential relations among them.

By filtering these illogical cases on the baseline model through the testing stage, we can obtain a better result. Therefore, to better explore the learning process of these curves, we propose two consistency constraints during an end-to-end optimization process, that consider the relations between different time steps inside each probability curve, named Internal Constraint (IC) and the relations among different probability curves, named External Constraint (EC).

3.3. Adding Consistency Constraints

As illustrated in Figure 2 we add two constraints, IC and EC, to regularize the learning process. IC is built inside each probability curve that enforces (i) the probabilities in each foreground or background regions are consistent; (ii) the probabilities between foreground and background regions are separate. EC performs consistency constraints among three probability curves, which applies the con-
straints between the continue-start curve and the continue-end curve. (i) if there were an abrupt rise in the continuing curve, the starting curve should give a high probability, and vice versa; (ii) if there were an abrupt drop in the continuing curve, the ending curve should give a high probability, and vice versa.

3.3.1 Internal Constraint

We build our Internal Constraint (IC) on three predicted probability curves, continuing curve $p^C$, starting curve $p^S$, and ending curve $p^E$. To constrain these probability curves to have a stable response inside each of them, we first define the foreground and background regions for each probability curves. The foreground regions are defined as the locations where action continues by $g^C_t = 1$, action starts by $g^S_t = 1$, and action ends by $g^E_t = 1$, respectively. The background regions are the rest of the time where $g^C_t = 0$, $g^S_t = 0$, and $g^E_t = 0$. The yellow block in Figure 2 shows an example of the IC on continuing curve $p^C$. Given the predicted continuing probabilities $\{p^C_t\}_{t=1}^T$, we build an adjacency matrix $A \in [0,1]^{T \times T}$ to establish the relationship between predicted probabilities by measuring the distance between them. The elements in $A$ are formulated as $a_{i,j} = f(p^C_i, p^C_j)$, where $f$ is a distance function ($l_1$ distance in our experiments) to measure the difference between $p^C_i$ and $p^C_j$. In terms of the division of the foreground and background region, the predicted continuing probabilities $\{p^C_t\}_{t=1}^T$ are divided into a foreground set $\mathcal{U}^C = \{p^C_t \mid g^C_t = 1\}$ and a background set $\mathcal{V}^C = \{p^C_t \mid g^C_t = 0\}$. To make the prediction pairs inside $\mathcal{U}^C$ set or $\mathcal{V}^C$ set consistent, and the prediction pairs between $\mathcal{U}^C$ set and $\mathcal{V}^C$ set separate, we first use three masks $M_U$, $M_V$, and $M_{UV} \in \{0,1\}^{T \times T}$ to select the corresponding $a_{i,j}$ in adjacency matrix $A$, then shrink the average distance inside foreground set or background set and enlarge the average distance between them. Therefore, the IC on continuing probability curve $p^C$ is formulated in Eq. (1):

$$L_{IC}^C = \frac{1}{N_U} \sum_{i,j} A \odot M_U + \frac{1}{N_V} \sum_{i,j} A \odot M_V + \frac{1}{N_{UV}} \sum_{i,j} A \odot M_{UV},$$

where $i,j$ is the index of matrix $A$. $M_U$, $M_V$, and $M_{UV}$ select the positions with “1” shown in Figure 2 where probability pairs come from only $\mathcal{U}^C$ set, only $\mathcal{V}^C$ set, and between $\mathcal{U}^C$ and $\mathcal{V}^C$ 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. Replicating constraint on continuing curve, we can also obtain the $L_{IC}^S$ and $L_{IC}^E$. Hence, the whole IC is formulated in Eq. (2):

$$L_{IC} = L_{IC}^C + L_{IC}^S + L_{IC}^E,$$

3.3.2 External Constraint

We build our External Constraint (EC) between three probability curves, continuing curve $p^C$, starting curve $p^S$, and ending curve $p^E$. To make the consistency between these probability curves, we propose two hypotheses, (i) if there were an abrupt rise in the continuing curve, the starting curve should give a high probability, and vice versa; (ii) if there were an abrupt drop in the continuing curve, the ending curve should give a high probability, and vice versa. Following these hypotheses, we use the first derivative of $p^C$ to capture the abrupt rise and drop of continuing probability curve. As for the practical discrete case in temporal dimension, we use the difference term instead of the differential term in Eq. (3):

$$\dot{p}^C = \frac{\partial p^C}{\partial t} \approx \Delta p^C = p^C_{t+1} - p^C_t. \quad (3)$$

Illustrated in red block of Figure 2 we build two kinds of constraints for external consistency, the continue-start constraint in yellow circle and the continue-end constraint in blue circle. We use the positive values in $\dot{p}^C$ to represent continuing probability rise rate, notated as $p^+_t = \max\{0, \dot{p}^C_t\}$, and use negative values in $\dot{p}^C$ to represent continuing probability drop rate, notated as $p^-_t = -\min\{0, \dot{p}^C_t\}$. Thus, we can build these consistency between three probability curves, and the EC is formulated in Eq. (4):

$$L_{EC} = \frac{1}{T} \sum_{t=1}^T |p^+_t - p^S_t| + |p^-_t - p^E_t|.$$  \quad (4)

3.4. Proposal Generation and Classification

Following the same rules in BSN [18] and ScaleMatters [11], we select the starting and ending points in terms of $p^S$ and $p^E$; 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 $p^S$ and $p^E$ by two rules [18]: (i) start points $t$ where $p^S_t > 0.5 \times (\max_{t'=1} p^S_{t'} + \min_{t'=1} p^S_{t'})$; (ii) start points $t$ where $p^S_{t-1} < p^S_t < p^S_{t+1}$. The ending points are selected by the same rules. Following these two rules, we prepare a set of candidate starting and ending points 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, $p^S_t \times p^E_t$; (ii) train an additional evaluation network to score candidate
proposals \( \hat{S} \), which is noted as \( \phi(t_s, t_e) \). The detailed information can be found in [11]. Thus, the final confidence score for candidate proposals is \( p^E \times p^S \times \phi(t_s, t_e) \).

**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 [18], we use video-level classifier in UntrimmedNet [30] for THUMOS14 dataset. As for ActivityNet1.3 dataset, we use the video-level classification results generated by [35].

### 3.5. Implementation Details

#### Network Design

We build our IC and EC 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 \( \{ f_t \}_{t=1}^{T_w} \) of untrimmed videos. Since untrimmed videos have various video length, we truncate or pad zeros to obtain a fixed length features of window \( T_w \). Through BaseNet, the output features are shared by three 2-layer ProbNets to predict probability curves \( (p^C, p^S, p^E) \) and two RegrNets to predict starting and ending boundary offsets \( (o^S, o^E) \).

**Loss function.** Predicting continuing, starting, and ending probabilities are trained with the cross-entropy loss. We separate the calculation by the foreground and background regions; then mix them with a ratio of 1:1 to balance the proportion of the foreground and the background. The loss of predicting the continuing probability is formulated in Eq. (5):

\[
L_C = \frac{1}{T_C^+} \sum_{t \in T_C^+} \ln(p^C_t) + \frac{1}{T_C^-} \sum_{t \in T_C^-} \ln(1 - p^C_t),
\]

where \( T_C^+ \) and \( T_C^- \) denote the foreground and background set in \( p^C \), while \( T_C^+ \) and \( T_C^- \) are the number of them, respectively. Replacing the script “C” with “S” or “E” in Eq. (5), we can obtain the \( L_S \) and \( L_E \), respectively. Hence, the whole classification loss is formulated in Eq. (6):

\[
L_{cls} = L_C + L_S + L_E.
\]

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 [24, 16], we apply SmoothL1 Loss [10] \( (SL_1) \) to our regression task, which is formulated in Eq. (7):

\[
L_{reg} = \frac{1}{T_s} \sum_{t \in T_s} SL_1(o^S_t, \hat{o}^S_t) + \frac{1}{T_e} \sum_{t \in T_e} SL_1(o^E_t, \hat{o}^E_t)
\]

where \( T_s \) and \( T_e \) are the foreground regions in \( p^S \) and \( p^E \), \( T_s^+ \) and \( T_e^+ \) are the number of them. \( o^S_t \) and \( o^E_t \) are the predicted starting and ending offsets with their ground-truth (\( \hat{o}^S_t \) and \( \hat{o}^E_t \)). Adding our proposed consistency constrains IC and EC, the overall objective loss function is formulated in Eq. (8):

\[
L = L_{cls} + L_{reg} + L_{IC} + L_{EC}
\]

#### Network training

Our BaseNet, ProbNet, and RegrNet are jointly trained from the scratch by multi tasks which are the classification \( (L_{cls}) \) and regression \( (L_{reg}) \) tasks with two constraints IC \( (L_{IC}) \) and EC \( (L_{EC}) \). The ratio of each loss component is equal. As mentioned previously, 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. Since the THUMOS14 dataset is relative small the batch size is set to be 3 for it and 16 for the ActivityNet1.3 dataset. We use a SGD optimization method with 0.9 momentum to train both datasets. In Section 3.4 the additional evaluation network for proposal ranking follows the same settings in [11].

### 4. Experiments

#### 4.1. Datasets and Evaluation Metrics

**Datasets and features.** We validate our proposed IC and EC 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 [19], 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.

**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 IoU 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 performance where IoU is set to be from 0.5 to 0.95 with a stride of 0.05.

### 4.2. Compare with the State-of-the-arts

**Temporal action proposals.** We compare the temporal action proposals generated by our IC\&EC equipped model on THUMOS14 dataset. As illustrated in Table 2 comparing with previous works, we can achieve the best performance especially on AR@50 metric. Our constraints help to generate more precise candidate starting and ending points, so we can achieve a high recall with fewer proposals.

**Temporal action localization.** Classifying the proposed proposals, we obtain the final localization results. As illustrated in Table 3 and Table 4, our method outperforms the previous works. Especially at high IoU settings, we achieve significant improvements since our constraints 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.

### 4.3. Ablation Studies

To explore how these constraints, IC and EC, improve the quality of temporal action proposals, we conduct following detailed ablation studies on THUMOS14 dataset.

**Effectiveness of IC.** As illustrated in Table 5 “Internal Constraints”, we compare the components of IC in terms of the AR@AN. In Section 3.3.1, the IC is introduced to continue probability curve (\(L_{\text{IC}}\)), starting probability curve (\(L_{\text{IC}}\)), and ending probability curve (\(L_{\text{IC}}\)). Compared with the baseline result without any constraint, only introducing \(L_{\text{IC}}\) improves a little. But the IC on starting \(L_{\text{IC}}\) and ending \(L_{\text{IC}}\) can achieve better results than only introducing \(L_{\text{IC}}\). Combined all three IC constraints, the AR@50 is improved from 37.62\% to 39.53\%.

**Effectiveness of EC.** As illustrated in Table 5 “External Constraints”, we compare the components of external constraints in terms of the AR@AN. In Section 3.3.2, the EC is introduced between continue-start (C\&S) and continue-end (C\&E). EC on C\&S (C\&E) makes the consistency between the starting curve (ending curve) and the derivative of continuing curve, which can suppress the false positives only observed from a single probability curve. Only introducing EC to C\&S or C\&E obtains less than 1\% absolute improvement on AR@50. When combined C\&S and C\&E, it can improve 1.49\% on AR@50.

**Combining IC\&EC.** As illustrated in Table 5 “All Constraints”, we compare the different constraints in terms of the AR@AN. Both the IC and EC independently achieve more than 1\% absolute improvement on AR@50. When combined IC and EC, the AR@50 is improved from 37.62\% to 40.98\%. Constraints inside each probability curve and between them are coupled, which leads to a positive feedback. It means when we get the better probability curve that fits the IC, the hypothesis of EC is more appropriate between three probability curves, and vice versa.

**Effectiveness of kernel size and layers.** The scale of

| Method     | Feature | @0.5 | @0.7 | @0.95 |
|------------|---------|------|------|-------|
| SST [3]    | 2-Stream | 41.2 | 31.5 | 20.0  |
| TURN [9]   |         | 46.3 | 35.3 | 24.5  |
| BSN [18]   |         | 53.5 | 45.0 | 36.9  |
| MGG [20]   |         | 53.9 | 46.8 | 37.4  |
| BMN [17]   |         | 56.0 | 47.4 | 38.8  |
| ScaleMatters [11] | 2-Stream | 53.2 | 48.1 | 41.5  |

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

| Method     | Feature | @0.3 | @0.4 | @0.5 | @0.6 | @0.7 |
|------------|---------|------|------|------|------|------|
| SST [3]    |         | 43.8 | 39.1 | 31.5 | 23.4 | 14.1 |
| TURN [9]   |         | 46.3 | 41.2 | 35.3 | 28.8 | 21.3 |
| BSN [18]   |         | 53.5 | 45.0 | 36.9 | 28.4 | 20.0 |
| MGG [20]   |         | 53.9 | 46.8 | 37.4 | 29.5 | 21.3 |
| BMN [17]   |         | 56.0 | 47.4 | 38.8 | 29.7 | 20.5 |
| ScaleMatters [11] | 2-Stream | 53.2 | 48.1 | 41.5 | 31.5 | 21.7 |

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

| Method     | Feature | 0.3  | 0.4  | 0.5  | 0.6  | 0.7  |
|------------|---------|------|------|------|------|------|
| CDC [25]   |         | 43.8 | 25.8 | 0.2  | 22.7 |
| SSN [34]   |         | 39.1 | 23.4 | 5.4  | 23.9 |
| BSN [18]   |         | 46.4 | 29.9 | 8.0  | 29.1 |
| Ours       |         | 43.4 | 33.9 | 9.2  | 30.1 |

**Table 4. Comparisons in terms of mAP (%) on ActivityNet1.3 (val).** The “Average” is calculated at the IoU of \{0.5 : 0.05 : 0.95\}.
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 BaseNet. As illustrated in Table 6, 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. However, increasing the kernel size is more effective than increasing the number of layers. When we fix the kernel size to be 5, increasing the number of layers will improve the result. But when the number of layers exceeds 5, the AR@AN will decrease since more parameters will lead to over-fit. Changing the kernel size has the same phenomenon. When the kernel size exceeds 9, the AR will not increase any more.

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 7, we give the upper bound “oracle” results by scoring a proposal via the maximum IoU value with ground-truth segments. Then we compare two scoring function, \( p_t^c \times p_t^e \) and \( p_t^s \times p_t^e \times \phi(t_s, t_e) \). Directly using starting and ending probability at boundaries is simple and effective, however, training a new evaluation network [18, 11] to evaluate the confidence of proposals can further improve the performance by a significant margin. It is worth to notice that, “Oracle” result gives a guideline to temporal action proposal task. The absolutely value of the “Oracle” determines the quality of proposals and the gap between the scoring results and the “Oracle” determines the quality of scoring method.

4.4. Generalizing IC&EC to Other Algorithms

Our proposed two consistency constraints, i.e., IC and EC, are effective in generating the probability curves of continuing, starting, and ending. To prove these constraints are valid for other network architecture and framework in TAL, we introduce them to ScaleMatters [11] and PGCN [33], respectively.

ScaleMatters [11] designed a multi-scale architecture TSA-Net with their proposed MDC blocks to predict probability curves of continuing, starting, and ending. To make a fair comparison in the same codebase, we also report the baseline results by our implementation. As illustrated in Table 8, our IC and EC significantly outperforms the baseline models on all three network architectures. PGCN [33] explore the proposal-proposal relations using Graph Convolutional Networks [15] (GCN) to localize action instances. This framework builds upon the proposed proposals from BSN [18] method. We introduce our two constraints to generated candidate proposals for PGCN framework. As illustrated in Table 9, introducing IC and EC to PGCN improves the localization performance.

4.5. Visualization of Qualitative Results

As illustrated in Figure 3, we visualize some examples on both datasets. Comparing the predicted \( p^c \), \( p^s \), and \( p^e \) with or without IC and EC, we find our proposed con-

---

**Table 5. Ablation studies on Internal Constraint and External Constraint in terms of AR@AN (%) on THUMOS14. All numbers are the averaged value in the last 10 epochs.**

| Baseline | AR@50 | AR@100 | AR@200 |
|---------|-------|--------|--------|
|         | 37.62 | 45.72  | 52.50  |

| Continue | Start | End | Internal Constraints |
|----------|-------|-----|----------------------|
| ✓        |       | ✓   | ✓                    |
| ✓        | ✓     | ✓   | ✓                    |
| ✓        | ✓     | ✓   | ✓                    |

| C&S | C&E | External Constraints |
|-----|-----|----------------------|
| ✓   | ✓   | ✓                    |
| ✓   | ✓   | ✓                    |
| ✓   | ✓   | ✓                    |

| IC | EC | All Constraints |
|----|----|----------------|
| ✓  | ✓  | ✓               |
| ✓  | ✓  | ✓               |
| ✓  | ✓  | ✓               |

**Table 6. 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  |
| 2      | 15          | 41.69 | 48.84  | 54.26  |

**Table 7. 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 6. All numbers are the averaged value in the last 10 epochs.**

| Proposal Scoring | AR@50 | AR@100 | AR@200 |
|------------------|-------|--------|--------|
| Oracle           | 50.40 | 56.98  | 58.91  |
| \( p_t^c \times p_t^e \) | 42.63 | 49.85  | 55.32  |
| \( p_t^s \times p_t^e \times \phi(t_s, t_e) \) | 44.23 | 50.67  | 55.74  |
Figure 3. Qualitative results on THUMOS14 (left) and ActivityNet1.3 (right) datasets. “green” lines are ground-truth, “blue” lines are predicted curves by baseline model, and “orange” lines are predicted curves with IC and EC. The top of the figure are action proposals with their confidence scores.

Table 8. Generalizing IC&EC to multi-scale TSA-Net [11] in terms of AR@AN (%) on THUMOS14. * indicates the results that are implemented by ours.

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

Table 9. Generalizing IC&EC to PGCN [33] 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 | 63.83 | 58.82 | 48.88 |
| PGCN* + IC&EC | 71.79 | 69.96 | 65.08 | 59.07 | 48.00 |

Table 10. Introducing oracle information to TAL in terms of mAP (%) on THUMOS14.

| External Constraints | O_rank | O_cls | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|----------------------|--------|-------|-----|-----|-----|-----|-----|
| Others               | 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 |     |

straints indeed make predicted curves become stable inside foreground and background regions. Besides, some false positive activations in $p^S$ and $p^E$ are suppressed, so that these constraints can reduce many candidate proposals via these bad starting and ending points.

4.6. Discussion for Future Direction

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 10, $O_{\text{rank}}$ means that each candidate proposal is ranked by the max IoU score with all ground-truth action instances. $O_{\text{cls}}$ means that the ground-truth action labels are assigned to candidate proposals. When introducing $O_{\text{rank}}$ and $O_{\text{cls}}$ 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 $O_{\text{cls}}$. However, when introducing the oracle ranking information $O_{\text{rank}}$, 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 temporal probability curves are inconsistent, either internally or externally, in the task of action localization. To alleviate this problem, we propose two temporal constraints which can be optimized together with the backbone. Experiments reveal that our approach improves the performance of temporal action localization both quantitatively and qualitatively.

Our research reveals that state-of-the-art video analysis
algorithms, though built upon powerful 3D-based networks, mostly have a limited understanding in the temporal dimension, which can lead to undesired properties, e.g., inconsistency or discontinuity. Our work provides an example of adding temporal priors to deep networks, which we believe, is a promising direction of video analytics.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. arXiv preprint arXiv:1609.08675, 2016.

[2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In Proceedings of the International Conference on Computer Vision (ICCV), pages 5561–5569, 2017.

[3] Shyamal Buch, Victor Escorcia, Chuangqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pages 2911–2920, 2017.

[4] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. 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), pages 1914–1923, 2016.

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 961–970, 2015.

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pages 6299–6308, 2017.

[7] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In Proceedings of the European Conference on Computer Vision (ECCV), pages 768–784. Springer, 2016.

[8] Jiyang Gao, Kan Chen, and Ram Nevatia. Ccup: Complementary temporal action proposal generation. In Proceedings of the European Conference on Computer Vision (ECCV), pages 68–83, 2018.

[9] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In Proceedings of the International Conference on Computer Vision (ICCV), pages 3628–3636, 2017.

[10] Ross Girshick. Fast r-cnn. In Proceedings of the International Conference on Computer Vision (ICCV), pages 1440–1448, 2015.

[11] Guoqiang Gong, Liangfeng Zheng, Kun Bai, and Yadong Mu. Scale matters: Temporal scale aggregation network for precise action localization in untrimmed videos. arXiv preprint arXiv:1908.00707, 2019.

[12] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2014.

[13] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pages 1725–1732, 2014.

[14] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. arXiv preprint arXiv:1705.06950, 2017.

[15] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. pages 1–14, 2017.

[16] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), pages 734–750, 2018.

[17] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. 2019.

[18] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In Proceedings of the European Conference on Computer Vision (ECCV), pages 3–19, 2018.

[19] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pages 1298–1307, 2019.

[20] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pages 3604–3613, 2019.

[21] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Yan Yan, Lisa Brown, Quanfu Fan, Dan Gutfriend, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. IEEE transactions on Pattern Analysis and Machine Intelligence (T-PAMI), 2019.

[22] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 5534–5542. IEEE, 2017.

[23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 779–788, 2016.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NeurIPS), pages 91–99, 2015.

[25] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In Proceedings of the IEEE
[26] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pages 1049–1058, 2016.

[27] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 4489–4497. IEEE, 2015.

[28] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6450–6459, 2018.

[29] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1430–1439, 2018.

[30] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pages 4325–4334, 2017.

[31] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In Proceedings of the European Conference on Computer Vision (ECCV), pages 305–321, 2018.

[32] Yuanjun Xiong, Yue Zhao, Limin Wang, Dahua Lin, and Xiaowu Tang. A pursuit of temporal accuracy in general activity detection. arXiv preprint arXiv:1703.02716, 2017.

[33] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. 2019.

[34] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaowu Tang, and Dahua Lin. Temporal action detection with structured segment networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2914–2923, 2017.

[35] Y Zhao, B Zhang, Z Wu, S Yang, L Zhou, S Yan, L Wang, Y Xiong, D Lin, Y Qiao, et al. Cuhk & ethz & siat submission to activitynet challenge 2017. arXiv preprint arXiv:1710.08011, 2017.