Stepwise Multi-grained Boundary Detector for Point-supervised Temporal Action Localization

Mengnan Liu[1], Le Wang[1*], Sanping Zhou[1], Kun Xia[1], Qi Wu[1], Qilin Zhang[2], and Gang Hua[3]

[1] National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, National Engineering Research Center for Visual Information and Applications, Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2] DeepNight.ai
[3] Multimodal Experiences Research Lab, Dolby Laboratories

Abstract. Point-supervised temporal action localization pursues high-accuracy action detection under low-cost data annotation. Despite recent advances, a significant challenge remains: sparse labeling of individual frames leads to semantic ambiguity in determining action boundaries due to the lack of continuity in the highly sparse point-supervision scheme. We propose a Stepwise Multi-grained Boundary Detector (SMBD), which is comprised of a Background Anchor Generator (BAG) and a Dual Boundary Detector (DBD) to provide fine-grained supervision. Specifically, for each epoch in the training process, BAG computes the optimal background snippet between each pair of adjacent action labels, which we term *Background Anchor*. Subsequently, DBD leverages the background anchor and the action labels to locate the action boundaries from the perspectives of detecting action changes and scene changes. Then, the corresponding labels can be assigned to each side of the boundaries, with the boundaries continuously updated throughout the training process. Consequently, the proposed SMBD could ensure that more snippets contribute to the training process. Extensive experiments on the THUMOS'14, GTEA and BEOID datasets demonstrate that the proposed method outperforms existing state-of-the-art methods.

Keywords: Temporal Action Localization · Point-Supervised Learning

1 Introduction

Temporal Action Localization (TAL) aims to detect actions of interest within an untrimmed video by identifying both their temporal boundaries and action categories. Fully-supervised TAL methods (FTAL) [1, 16, 17, 46, 48] requires high-quality temporal boundary annotation which is very time-consuming to get and rarely readily available. Hence, weakly-supervised temporal action localization (WTAL) [1, 30, 37, 46–48] where only video-level labels are required

*Corresponding author

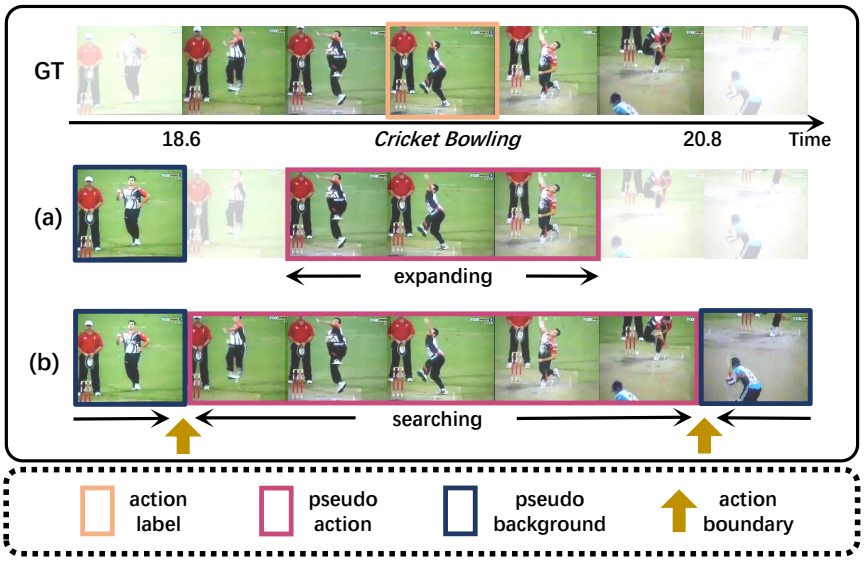

Fig. 1: Comparison. **(a):** Previous methods expand pseudo label from the single-frame action label, without explicit determination of action boundaries. **(b):** SMBD focuses on searching for the boundaries of the actions and assigns pseudo labels to both sides of the boundary.

is proposed and make significant progress. However, WTAL imposes an intractable problem of distinguishing between actions and backgrounds due to missing instance-level annotations. As a middle ground between FTAL and WTAL, Point-supervised TAL (PTAL) [23] is proposed, requiring a few action instance labels per untrimmed video. Since it only requires one randomly annotated snippet per action instance, it alleviates the hard work of pinpoint the accurate temporal action boundaries, which has attracted growing attention in academia and industry. However, sparse labeling of individual frames leads to semantic ambiguity in determining action boundaries due to the lack of continuity in the highly sparse point-supervision scheme. Exiting methods [9, 11, 22] tackle this challenge by either iterative refinement [22] or WTAL-style schemes [19, 26, 28]. Iterative refinement [22] alternates between predicting and applying pseudo labels, which heavily rely on empirical threshold setting. Typical WTAL-style schemes [19, 26, 28] tackle PTAL through taking the point labels as the strong video-level categorical labels, which are prone to produce incomplete results. Despite their advancements, they struggle with complex and dense action instances since they only capture the action snippets that are similar to action labels, as shown in Fig. 1 (a). To fully exploit all snippet in the video, it is necessary to detect action boundaries, thereby effectively learning the semantics of the entire action to improve performance, as shown in Fig. 1 (b).

To this end, this paper proposes a novel framework, Stepwise Multi-grained Boundary Detector (SMBD), to tackle PTAL through searching for action bound-

ary, which is comprised of a Background Anchor Generator (BAG) and a Dual Boundary Detector (DBD). Our main idea is to first detect the reliable background frame in the video and then search for the optimal action boundary between the background frame and the action label.

Specifically, given an untrimmed video, only the action labels of sparse annotated frames are available during training. We first introduce a background anchor generator to locate the background frame with the highest confidence between each pair of action labels by voting of distinct classification heads, where we term such background frame as *background anchor* in this paper. Subsequently, we argue that there must be an action boundary within each adjacent background anchor and action label. On the one hand, action boundary usually refers to the timestamp characterizing the action change between adjacent video frames, *e.g.*, the change of an athlete from standing to running indicates the action starting. On the other hand, action boundary also describe the scene change, *e.g.*, the switch between the foreground frame and the background frame. Building upon this observation, we propose a dual boundary detector to locate action boundary by detecting action changes and scene changes from adjacent background anchors and action labels. The dual boundary detector encompasses a fine-grained boundary detector and a coarse-grained boundary detector, where the former searches for fine-grained boundaries by observing the change of the action classification score while the latter retrieves coarse-grained boundaries through evaluating the difference between the foreground and background scores.

Finally, by merging both kind of action boundaries, we are able to assign pseudo labels to all video frames, enabling the learning of complete action semantics during training. Experiments on three benchmarks validate the effectiveness of our method.

In summary, our contributions are as follows:

(1) This paper introduces a Stepwise Multi-grained Boundary Detector for PTAL by emphasizing learning entire action semantics with only sparse point supervision. It could ensure that more video frames contribute to model training by means of searching action boundaries.

(2) We propose a fine-grained boundary detector and a coarse-boundary detector to locate the action boundaries from the complementary perspectives of detecting action changes and scene changes, respectively.

(3) Extensive experiments on the THUMOS'14, GTEA and BEOID datasets validate the superiority of the proposed method over existing point-supervised TAL methods.

2 Related Work

Fully-Supervised Temporal Action Localization requires precise start and end action boundaries, with each snippet assigned an exact label during the training phase. Thanks to action boundaries, the full supervision performance is high in both classification and localization tasks. There are primarily three methods. (1)Proposal-based representation [3, 40, 45, 46, 55], which involves

generating a window that indicates the start and end snippets of the action, followed by boundary refinement via a regression head. (2)Proposal-free representation [15,24,42–44] directly predicts the probability of each snippet, then determines the snippet with the highest probability and considers it as the proposal of the action boundary. (3)Transformer-based [5,20,35,51] methods, which leverage the Transformer architecture [38] for action localization, achieving remarkable performance on TAL benchmarks. Although these methods have demonstrated good results, the high cost of labeling of each snippet remains a challenge.

Weakly-Supervised Temporal Action Localization only provides the label for the video during training. As frame-level labels are not needed, annotation cost is significantly reduced. The common ideas of the algorithm can be divided into three kinds. (1) The attention-based paradigm [21, 26, 27, 49] calculates the probability of frame-level features and utilizes the obtained attention guide classification. (2) The MIL-based paradigm [19,28,39,52] generates frame-level class scores, i.e., the class activation sequence(CAS). Subsequently, the scores are aggregated to produce a video-level label. (3) The graph-convolution-based paradigm [1, 30, 37, 46–48] considers each snippet of the video as a node on the graph, where the edges between the nodes are weighted based on their similarity. In the feature space, the related time snippets become closer to each other, while the unrelated time snippets become more separated.

Point-Supervised Temporal Action Localization bridges the gap between full supervision and weak supervision by requring one label per action instance, leading to effectively a fews labels (typically 15 per video clip on average in THUMOS'14). Moltisanti *et al*. [23] first proposed point-supervised in temporal action localization. SF-Net [22] expands to the snippets adjacent to the single frame label to obtain the pseudo-label for training. Ju *et al*. [9] propose a novel two-stage framework, which divides the entire video into multiple clips and sequentially conquers these clips. Li *et al*. [14] use the model output and annotated timestamps to generate frame-wise labels for action segmentation. Action segmentation task differs from ours in that it does not account for background segments, and all actions are adjacent without gaps in between. LACP [11] generates pseudo labels based on point annotations, then utilizes a greedy algorithm to find the optimal sequence of actions and learns the completeness of the action for the optimal sequence.

However, they do not explicitly detect the action boundary, resulting in the model's inability to recognize the boundaries. Additionally, during the extension of pseudo labels, the feature not assigned pseudo labels are ignored. This results in the model relying excessively on high-confidence action snippets while overlooking features with slightly lower confidence. In contrast, our approach involves finding boundaries to assign pseudo labels to more action snippets to better exploit their semantics.

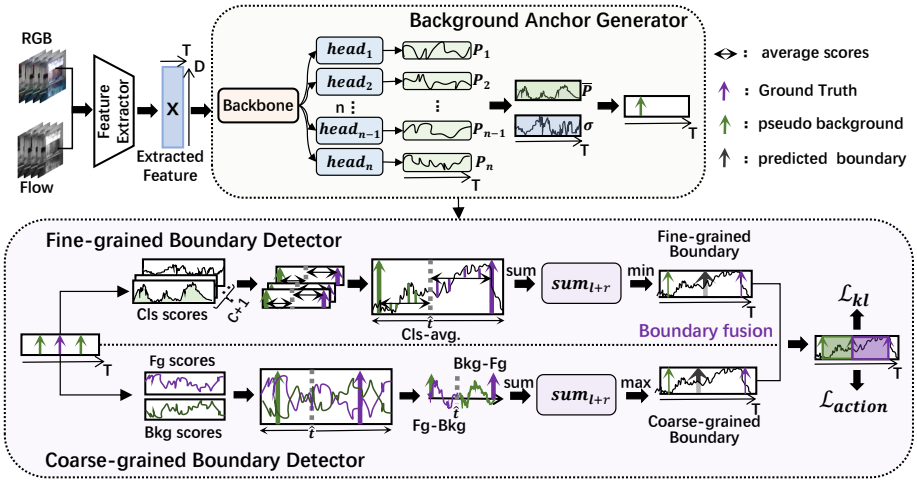

Fig. 2: Overview of the proposed method. Besides the conventional objectives, *e.g.*, video-level and point-level classification losses, we use BAG to determine the optimal background snippet as background anchor. Based on the action label and the background anchor, DBD utilizes the classification scores and foreground-background scores to detect the action boundary between each pair of adjacent action label and the background anchor. Then all snippet are assigned pseudo labels so as to more snippets can be exploited during training.

3 Method

In this section, we present the details of our SMBD with an overall architecture in Fig. 2. The problem setting is described in Sec. 3.1 and the feature embedding is given in Sec. 3.2. We present our BAG, which serves to select the background anchor in Sec. 3.3 and DBD in Sec. 3.4. Subsequently, we detail our loss functions in Sec. 3.5 and elucidate the inference of our SMBD in Sec. 3.6.

3.1 Problem Definition

Point-supervised temporal action localization (PTAL) aims to train a model with only a single-frame annotation of each action instance. Given an untrimmed video, each action instance is annotated with a timestamp t and its action category y. The PTAL model produces the prediction results for each video, where each predicted action instance could be represented as $(s, e, \hat{y}, p)$, where s and e are the starting and ending time of each action instance, $\hat{y}$ is the predicted category, and p is the confidence score.

3.2 Feature Embedding

Following [11, 53], we first encode each successive fixed-length frames with a pre-trained feature extractor (*e.g.*, I3D [2]) and obtain snippet-level features.

We denote the video feature sequence as $\boldsymbol{X} \in \mathbb{R}^{T \times D}$, where T and D present the number of video snippets and the feature dimension, respectively. Then, we input the video feature sequence into a snippet-level classification head to obtain class-specific activation sequence (CAS) scores $\boldsymbol{p} \in \mathbb{R}^{T \times (C+1)}$, including C action categories and a background class.

3.3 Background Anchor Generator

In the point-supervision settings, only single-frame action labels are available for model training. We first introduce a background anchor generator (BAG) to locate the background snippet with the highest confidence between each pair of action labels by voting of distinct classification heads, where we term this background snippet as background anchor. Then, we argue that there must be an action boundary within each adjacent background anchor and action label.

Inspired by the Monte Carlo estimate [31], we design N classification heads to evaluate the confidence of each video snippet as a background anchor comprehensively, as shown in Fig. 2.

Based on the class-specific activation score $\boldsymbol{p} \in \mathbb{R}^{T \times (C+1)}$, we could obtain the background probability $\boldsymbol{p}^{\text{bkg}} \in \mathbb{R}^{T \times n}$ and its standard deviation $\boldsymbol{\sigma} \in \mathbb{R}^{T \times 1}$ from the N heads. The N heads can help us reduce the randomness of the model, allowing us to select more stable and reliable background anchors. The probability $\boldsymbol{p}^{\text{bkg}}$ represents the reliability of the predicted background snippet and the standard deviation $\boldsymbol{\sigma}$ represents the deviation from the randomness of the background snippet. Increasing $\boldsymbol{p}^{\text{bkg}}$ correlates with a higher likelihood of the snippet being the background, while decreasing $\boldsymbol{\sigma}$ corresponds to reduced randomness in background classification.

Give $\boldsymbol{p}^{\text{bkg}}$ and $\boldsymbol{\sigma}$, the stability is formulated as $\boldsymbol{s} = \overline{\boldsymbol{p}}^{\text{bkg}}/\boldsymbol{\sigma}$, where $\overline{\boldsymbol{p}}^{\text{bkg}} = 1/n \sum_{i=1}^{n} \boldsymbol{p_i}^{\text{bkg}}$. Then we select the background snippet with the highest stability between each pair of adjacent action label as the background anchor. Taking a pair of adjacent action labels $[t_i^{\text{act}}, t_{i+1}^{\text{act}}]$ as an example, the background anchor is calculated as,

$$t^{\text{bkg}} = \arg\max_t \{\boldsymbol{s}\}, \tag{1}$$

where $t \in [t_i^{\text{act}}, t_{i+1}^{\text{act}}]$ and t_i^{act} denotes the timestamp of i-th action label. $\boldsymbol{s}$ provides a comprehensive metric to measure the quality of pseudo background labels by their average probability $\overline{\boldsymbol{p}}^{\text{bkg}}$ and probability standard deviation $\boldsymbol{\sigma}$. As in Eq. (1), we can obtain all background anchors $\left\{t_j^{\text{bkg}}\right\}_{j=1}^{N^{\text{bkg}}}$ within each pair of adjacent action labels, where N^{bkg} is the number of the boundary anchors in a video. Our BAG aims to provide reliable background snippets for searching action boundary in the next step.

3.4 Dual Boundary Detector

As discussed in Sec. 1, sparse labeling of individual frames fails to capture the semantics of entire action instances due to varying semantic information among

sub-actions. It is crucial to identify action boundaries to learn the complete semantics of actions. The proposed Dual Boundary Detector (DBD) considers each pair of adjacent action snippet and background snippet as a candidate interval for retrieving a boundary within it. DBD adopts a two-branch structure which performs boundary detection from the complementary perspectives of action changes and scene changes, respectively. Afterwards, the two kind of action boundaries, namely fine-grained boundary and coarse-grained boundary, are fused to obtain our final pseudo boundaries.

Fine-Grained Boundary Detector. Action boundary usually refers to the timestamp characterizing the action change between adjacent video snippets, *e.g.*, the start boundary of an action depicts the moment of change from typically static scenes to dynamic ones. Building upon this insight, we propose to locate action boundary by detecting action changes between each pair of adjacent action labels and background anchors. Without loss of generality, taking the temporal interval $[t_i^{\text{bkg}}, t_j^{\text{act}}]$ as an example, we perform the fine-grained boundary detection within it. Given any $\hat{t} \in [t_i^{\text{bkg}}, t_j^{\text{act}}]$, we first calculate the uncertainty of $\hat{t}$ being an action boundary as follows,

$$s_{\hat{t}}^{l} = \frac{1}{\hat{t} - t_i^{\text{bkg}}} \sum_{t=t_i^{\text{bkg}}}^{\hat{t}} \left(\left| p_t - \frac{1}{\hat{t} - t_i^{\text{bkg}}} \sum_{n=t_i^{\text{bkg}}}^{\hat{t}} p_n \right| \right), \tag{2}$$

$$s_{\hat{t}}^{r} = \frac{1}{t_j^{\text{act}} - \hat{t}} \sum_{t=\hat{t}}^{t_j^{\text{act}}} \left(\left| p_t - \frac{1}{t_j^{\text{act}} - \hat{t}} \sum_{n=\hat{t}}^{t_j^{\text{act}}} p_n \right| \right), \tag{3}$$

where t_i^{bkg}, t_j^{act} and p_t represent the i-th background anchor, the j-th action label and the classification score of the t-th snippet, respectively. $s_{\hat{t}}^{l}$ and $s_{\hat{t}}^{r}$ represent the uncertainty of using $\hat{t}$ as a boundary. To be specific, $\hat{t}$ first divides the temporal interval $[t_i^{\text{bkg}}, t_j^{\text{act}}]$ into two subintervals $[t_i^{\text{bkg}}, \hat{t}]$ and $[\hat{t}, t_j^{\text{act}}]$. Then, we calculate the average scores within $[t_i^{\text{bkg}}, \hat{t}]$ and $[\hat{t}, t_j^{\text{act}}]$, which can be considered as the cluster centers of two subintervals. Therefore, the average distances between each p_t and the average scores could depict the uncertainty of $\hat{t}$ being a boundary. We will also obtain $s_{\hat{t}}^{l}$, $s_{\hat{t}}^{r}$ when $\hat{t} \in [t_i^{\text{act}}, t_j^{\text{bkg}}]$ in the same way.

As a result, the optimal the fine-grained boundary $\hat{t}$ could be calculated by minimizing the uncertainty of the two subintervals,

$$t_{\text{FB}} = \arg\min_{\hat{t}} \left(s_{\hat{t}}^{l} + s_{\hat{t}}^{r} \right). \tag{4}$$

We perform the above process between every pair of adjacent action label and background anchor to produce fine-grained boundaries for all action instances.

Coarse-Grained Boundary Detector. Unlike the fine-grained boundary detector that locate action boundary from the perspective of detecting action changes, we argue that action boundary also describe the switch between foreground and background. Therefore, we introduce a coarse-grained boundary

detector to search action boundary through evaluating the difference between foreground and background scores.

Concretely, given $\hat{t} \in [t_i^{\text{act}}, t_j^{\text{bkg}}]$, it divides the temporal interval $[t_i^{\text{act}}, t_j^{\text{bkg}}]$ into two subintervals $[t_i^{\text{act}}, \hat{t}]$ and $[\hat{t}, t_j^{\text{bkg}}]$. Afterwards, we compute the difference between the foreground score and the background score of each snippet within the $[t_i^{\text{act}}, \hat{t}]$ and $[\hat{t}, t_j^{\text{bkg}}]$, as well as the mean of the differences:

$$s_{\hat{t}}^{l} = \frac{1}{\hat{t} - t_i^{\text{act}}} \sum_{t=t_i^{\text{act}}}^{\hat{t}} \left(p_t^{\text{fg}} - p_t^{\text{bkg}} \right), \tag{5}$$

$$s_{\hat{t}}^{r} = \frac{1}{t_j^{\text{bkg}} - \hat{t}} \sum_{t=\hat{t}}^{t_j^{\text{bkg}}} \left(p_t^{\text{bkg}} - p_t^{\text{fg}} \right), \tag{6}$$

where t_i^{act} and t_j^{bkg} are the i-th action label and j-th background anchor. p_t^{bkg} and p_t^{fg} represent the background score and foreground score of t-th snippet, respectively. Therefore, $s_{\hat{t}}^{l}$ and $s_{\hat{t}}^{r}$ could reflect their confidence as foreground and background regions, respectively. As a result, $\hat{t}$ could be the optimal coarse-grained boundary when the confidences of the two intervals is maximum. In the other words, $\hat{t}$ will be highly likely to be the switch between foreground and background, *i.e.*, the action boundary:

$$t_{CB} = \arg\max_{\hat{t}} \left(s_{\hat{t}}^{l} + s_{\hat{t}}^{r} \right). \tag{7}$$

Similarly, we perform the above process between every pair of adjacent action label and background anchor to produce coarse-grained boundaries for all actions.

Boundary fusion. Both of the fine-grained and coarse-grained boundary detectors locate action boundary candidates in a complementary way. Thus, we could obtain reliable action boundaries by a simple boundary fusion,

$$t_B = \lambda t_{FB} + (1 - \lambda) t_{CB}, \tag{8}$$

where λ is weighting parameter for balancing t_B, their value are empirically determined with additional experiments in our supplementary materials.

3.5 Training

Once the action boundary is determined, we could assign the pseudo label for all snippet, where pseudo label is denoted as $\boldsymbol{y}^{\text{p}} \in \mathbb{R}^{N^{\text{f}} \times (C+1)}$. For the optimization of the Fine-grained Boundary Detector, we adopt the focal loss [18] to facilitate the training process,

$$\mathcal{L}_{\text{cls}} = -\frac{1}{N^{\text{f}}} \sum_{i=1}^{N^{\text{f}}} \left(\boldsymbol{y}_i^{\text{p}} (1 - \boldsymbol{p}_i)^{\beta} \log \boldsymbol{p}_i + (1 - \boldsymbol{y}_i^{\text{p}}) \boldsymbol{p}_i^{\beta} \log(1 - \boldsymbol{p}_i) \right), \tag{9}$$

where N^f indicates the number of the snippet, β is the focusing factor, which is set to 2 as in [18]. $\boldsymbol{p}_i$ indicates the predicted probability that the i-th snippet belongs to a certain category. For Coarse-grained Boundary Detector, we assign background and foreground pseudo labels, which is denoted as $\boldsymbol{y}^{\mathrm{b}} \in \mathbb{R}^{N^{\mathrm{f}} \times 2}$. We adopt a binary entropy loss to facilitate the training process,

$$\mathcal{L}_{\mathrm{act}} = -\frac{1}{N^{\mathrm{f}}} \sum_{i=1}^{N^{\mathrm{f}}} \left(\boldsymbol{y}_i^{\mathrm{b}} \log \boldsymbol{p}^{\mathrm{b}}{}_i + (1 - \boldsymbol{y}_i^{\mathrm{b}}) \log(1 - \boldsymbol{p}^{\mathrm{b}}{}_i) \right), \tag{10}$$

where $\boldsymbol{p}^{\mathrm{b}}{}_i$ indicates the predicted probability that the i-th snippet belongs to foreground or background.

To penalize the noise between the detected boundaries and the ground truth, we employ the KL-divergence loss. KL-divergence loss penalizes snippets whose predicted scores and the assigned pseudo label are inconsistent. The loss for pseudo labels is computed by,

$$\mathcal{L}_{\mathrm{KL}} = \frac{1}{N^{\mathrm{f}}} \sum_{i=1}^{N^{\mathrm{f}}} \mathrm{KL}(\boldsymbol{y}_i^{\mathrm{p}} || \boldsymbol{p}_i), \tag{11}$$

$$\mathrm{KL}(\boldsymbol{y}_i^{\mathrm{p}} || \boldsymbol{p}_i) = \sum_{j=1}^{C} \boldsymbol{p}_{ij} \log \left(\frac{\boldsymbol{p}_{ij}}{\boldsymbol{y}_{ij}^{\mathrm{p}}} \right), \tag{12}$$

where $\boldsymbol{y}^{\mathrm{p}}$, $\boldsymbol{p}$ and C represent the pseudo label, predicted probability and the number of action categories, respectively. i indicates i-th snippet and j indcates the j-th action category. The boundaries are continuously updated through back propagation. As the model's capability to identify noise is improved, the searched boundaries become more precise, accordingly.

The overall loss function for training our model is shown below,

$$\mathcal{L}_{\mathrm{total}} = \gamma_1 \mathcal{L}_{\mathrm{cls}} \ + \gamma_2 \mathcal{L}_{\mathrm{act}} + \gamma_3 \mathcal{L}_{\mathrm{KL}}, \tag{13}$$

where γ_1, γ_2 and γ_3 are the weighting parameters to balance the losses, their value are empirically determined with additional experiments in our supplementary materials.

3.6 Inference

Following previous point-supervised work [22], we predict video-level labels by temporally pooling and thresholding on the scores to determine which actions are present in the video. Then we use score thresholds to select candidate snippets. Candidate snippets are consolidated into a proposal as a result of our localization result. As in [11, 19], we set the confidence of each proposal to its outer-inner contrast score and use multiple thresholds for candidate snippets and perform non-maximum suppression (NMS) to remove overlapping proposals.

Table 1: Comparison with the state-of-the-art methods on THUMOS'14. We also compare the method with fully-supervised and weakly-supervised methods.

Supervision	Method	mAP@IoU(%)					AVG	AVG
		0.3	0.4	0.5	0.6	0.7	[0.1 : 0.5]	[0.3 : 0.7]
Fully supervised	BSN [17]	53.5	45.0	36.9	28.4	20.0	-	36.8
	BMN [16]	56.0	47.4	38.8	29.7	20.5	-	38.5
	G-TAD [46]	54.5	47.6	40.2	30.8	23.4	-	39.3
	BC-GCN [1]	57.1	49.1	40.4	31.2	23.1	-	40.2
	TCANet [29]	60.6	53.2	44.6	36.8	26.7	-	53.2
	AFSD [15]	67.3	62.4	55.5	43.7	31.1	-	52.0
	React [34]	69.2	65.0	57.1	47.8	35.6	-	55.0
	ASL [32]	83.1	79.0	71.7	59.7	45.8	-	67.9
Weakly supervised	CMCS [19]	41.2	32.1	23.1	15.0	7.0	40.9	23.7
	Bas-Net [12]	44.6	36.0	27.0	18.6	10.4	43.6	27.3
	DGAM [33]	46.8	38.2	28.8	19.8	11.4	45.6	29.0
	TSCN [49]	47.8	37.7	28.7	19.4	10.2	47.0	28.8
	CoLA [50]	51.5	41.9	32.2	22.0	13.1	50.3	32.1
	FTCL [7]	55.2	45.2	35.6	23.7	12.2	53.8	34.4
	DELU [4]	56.5	47.7	40.5	27.2	15.3	56.5	37.4
	DDG-Net [36]	58.2	49.0	41.4	27.6	14.8	57.8	38.2
Point supervised	Moltisanti *et al.* [23]	15.9	12.5	9.0	-	-	16.3	-
	SF-Net [22]	53.2	40.7	29.3	18.4	9.6	51.5	30.2
	Ju *et al.* [9]	58.1	46.4	34.5	21.8	11.9	55.3	34.5
	LACP [11]	63.3	55.2	43.9	33.3	20.8	61.6	43.3
	SF-Net+**SMBD**	59.1$^{\uparrow 5.9}$	46.4$^{\uparrow 5.7}$	33.4$^{\uparrow 4.1}$	20.7$^{\uparrow 2.3}$	10.5$^{\uparrow 0.9}$	55.8$^{\uparrow 4.3}$	34.0$^{\uparrow 3.8}$
	LACP+**SMBD**	**66.0**$^{\uparrow 2.7}$	**57.9**$^{\uparrow 2.7}$	**47.0**$^{\uparrow 3.1}$	**36.0**$^{\uparrow 2.7}$	**22.0**$^{\uparrow 1.2}$	**64.2**$^{\uparrow 2.6}$	**45.7**$^{\uparrow 2.4}$

4 Experiment

4.1 Datasets and Evaluation

Our experiments are conducted on three datasets and we adopt the single-frame annotations provided in [22] as the point supervision for each action instance during training.

THUMOS'14. The training data in THUMOS'14 [8] is based on the UCF101 [2] action dataset. There are a total of 200 validation videos and 213 test videos that belong to 20 action classes. On average, each video in the dataset includes 15 action instances, making the task of recognizing all actions quite challenging. Similarly to [14, 16], we use the validation set for training and the test set for evaluation.

GTEA. The GTEA [13] comprises 28 videos and contains seven different types of everyday activities. Each video contains around 20 actions, each lasting about 1 minute. For our experiments, we used 21 videos for training and 7 for testing.

BEOID. The BEOID, as described in [6], comprises 58 videos representing 30 action classes, and each video in the dataset contains an average of 12.5 actions. Following [22], 80% of the videos are used for training, while the remaining 20% are reserved for testing.

Evaluation Metrics. Following the standard procedure for temporal action localization, we calculate the mean average precision (mAP) across different

Table 2: Comparison with the state-of-the-art methods on GTEA and BEOID. AVG represents the mean average precision (mAP) at thresholds [0.1 : 0.1 : 0.7]. LACP* represents the reproduced results by official code of LACP.

Dataset	Method	mAP@IoU(%)				AVG
		0.1	0.3	0.5	0.7	[0.1:0.7]
GTEA	SF-Net [22]	58.0	37.9	19.3	11.9	31.0
	Ju [9] *et al.*	59.7	38.3	21.9	18.1	33.7
	LACP [11]	63.9	55.7	33.9	20.8	43.5
	LACP*	72.6	58.1	39.5	13.5	46.0
	LACP+**SMBD**	**75.0**	**61.3**	**41.1**	14.2	**47.4**
BEOID	SF-Net [22]	62.9	40.6	16.7	3.5	30.9
	Ju [9] *et al.*	63.2	46.8	20.9	5.8	34.9
	LACP [11]	76.9	61.4	42.7	25.1	51.8
	LACP*	74.7	61.8	44.4	21.3	51.9
	LACP+**SMBD**	**78.2**	**71.0**	**52.5**	**25.2**	**57.4**

intersection-over-union (IoU) thresholds to evaluate the action localization performance on the three datasets.

4.2 Implementation Details

We use the two-stream I3D network [2] pre-trained on the Kinetic-400 [2] to extract video features. We divide each video into 16-frame non-overlapping snippets and applied the TV-L1 optical flow algorithm [41] to extract optical flow. After obtaining the RGB feature and the optical flow feature, we employ the two-stream fusion operation as described in [25] to integrate the RGB feature and the optical flow feature branches, resulting in a 2048-dimensional vector for each snippet. The number of non-overlapping snippets is denoted as T. For all datasets, we optimize our method using Adam [10] with a learning rate of 10^{-4} and a batch size of 16. To determine the best background snippet, we set the background threshold at 0.85. The weight parameter λ is set to 0.5 and $\gamma_1 = 0.7, \gamma_2 = 1.0, \gamma_3 = 1.0$. The parameter selection experiments are detailed in the supplementary materials.

4.3 Comparison with State-of-the-art Methods

In Table 1, we apply our method to the backbone of SF-Net and LACP, respectively, and compare them under different levels of supervision on THUMOS'14 [8]. We also show our comparison experiment with the state-of-the-art methods HR-Pro [54] in our supplementary materials. The results demonstrate that our method achieves improvements across different backbones, indicating its versatility. It can be observed that due to the dense data provided by fully supervision, the mAP of the fully supervision task at high IoU is significantly better than that of weakly supervision. We utilize single-frame labels to search for the boundary and assign pseudo labels to all snippets as comprehensively as

Table 3: Comparison of different branch of the dual broundary detector on THUMOS'14. AVG represents the average mAP at the IoU thresholds [0.1 : 0.1 : 0.7].

Boundary Detector	mAP@IoU(%)				AVG
	0.1	0.3	0.5	0.7	[0.1:0.7]
Baseline(LACP)	75.4	64.3	45.0	20.4	52.4
Fine-grained Boundary	76.9	66.4	46.4	21.1	53.8
Coarse-grained Boundary	77.1	65.6	46.7	21.3	53.8
Dual Boundary	**77.7**	**66.0**	**47.0**	**22.0**	**54.1**

Table 4: Ablation study on THUMOS'14 with LACP as backbone. AVG represents the average mAP at the IoU thresholds [0.1 : 0.1 : 0.7].

$\mathcal{L}_{\text{cls}}$	BAG	$\mathcal{L}_{\text{KL}}$	mAP@IoU(%)				AVG
			0.1	0.3	0.5	0.7	[0.1:0.7]
			75.4	64.3	45.0	20.4	52.4
✔			77.2	65.4	45.1	19.4	52.7
✔	✔		77.8	66.4	46.4	21.1	53.8
✔		✔	77.5	65.4	46.4	21.3	53.7
✔	✔	✔	**77.7**	**66.0**	**47.0**	**22.0**	**54.1**

possible to compensate for sparse point supervision. Taking LACP [11]+SMBD as an example, we can notice the performance improvements on average mAP, *e.g.*, 2.6% on average mAP@[0.1:0.5] and 2.4% on average mAP@[0.3:0.7]. Additionally, we conduct similar experiments on SF-Net [22] and the results show significant performance improvement compared to the baseline.

We also provide experimental results on the GTEA [6] and BEOID [13] benchmarks in Table 2. In both datasets, our method has shown evident performance boosts under all thresholds compared to the baseline, verifying the efficacy of the proposed boundary detector.

4.4 Ablation Study

Impact of the Dual Boundary Detector. In table 3, we perform ablation experiments on each branch of the dual boundary detector to isolate their contributions. The addition of any individual branch alone improves the average mAP by 1.4% compared to the baseline. We fuse the two boundaries and the performance surpasses that of any single branch added individually, demonstrating the effectiveness of the Dual-branch approach.

Effectiveness of individual component. To further analyze the contribution of model components compared to the baseline setting, we perform a set of ablation studies on THUMOS'14. The results are summarized in table 4. Training

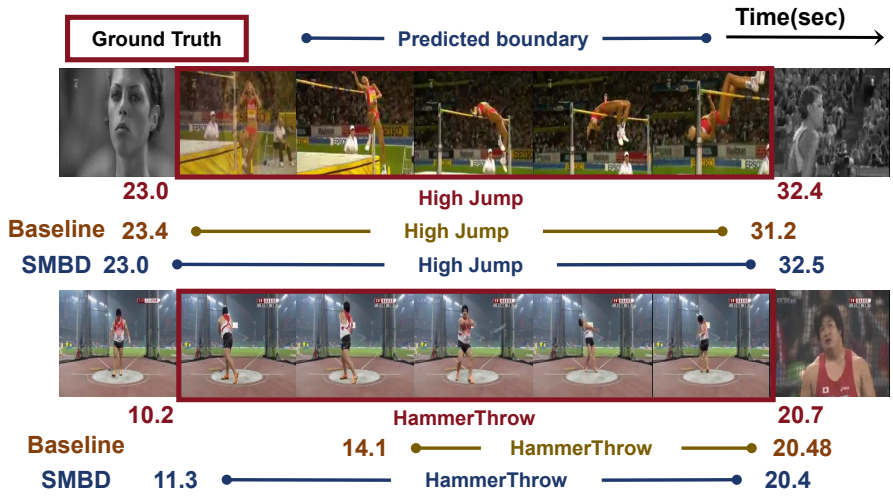

Fig. 3: Comparison with LACP [11] on the accuracy of the boundary snippet detected during training. We provide two examples with action classes of HighJump and HammerThrow. From the results, it can be seen that after incorporating DBD, the detected boundaries become more accurate.

with boundary detector alone improves the average mAP by 0.3% compared to the baseline. After incorporating background anchor for background detection, the average mAP increases by 1.4% due to improved accuracy in determining background snippets. Adding KL-loss results in a further average mAP increase of 1.3%, as the model penalizes noise around the boundaries. The last row experimental results indicate that the three components complement each other, leading to an overall improvement of 1.7%.

Impact of the Background Anchor Generator. We also compared SMBD with LACP in terms of the boundaries obtained during training in Appendix and provide some qualitative results in Fig. 3. We calculate the distance between the predicted boundaries and the ground truth (GT) for each video, and then count the number of videos. The boundaries obtained by SMBD have a smaller discrepancies with GT, and the accuracy of $distance \in [0, 5]$ improves 6.5%, more details are shown in the supplementary materials.

4.5 Qualitative Results

To further validate the effectiveness of the proposed method, we provide some qualitative results using LACP [11] and our model on test videos in THUMOS'14. [8] for comparison. We visualize several results in Fig. 4. As is evident, the localization performance of LACP neglects some snippets near the boundaries. On the contrary, our method provides more accurate detection results and a clearer distinction of boundaries, which demonstrate the effectiveness of our method.

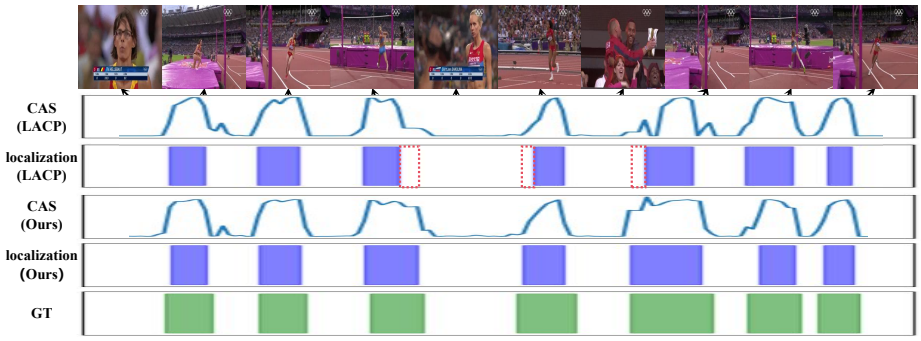

Fig. 4: Qualitative comparison of our proposed method with LACP [11] on THUMOS'14. We provide an example with action classes of HighJump. The first row is the input video, the lower two rows are class-specific activation sequence (CAS) and the localization results of LACP. The third and fourth rows are CAS and localization results from Ours. The bottom row is the ground-truth intervals. It can be seen that our detection results show higher IoUs with the ground truths.

5 Conclusion

In this paper, we presented a new strategy for point-supervised temporal action localization, where more action snippets are assigned pseudo labels during training. Specifically, we introduce the new concept of *Background Anchor* and conduct a boundary detection via our dual boundary detector. Subsequently, we assign pseudo labels to each snippet of the video based on the detected action boundaries. To mitigate noise around the action boundaries, we employ the KL divergence loss, which penalizes inconsistencies between pseudo labels and predicted scores. Experimental results validate that our background anchor generator can enhance the accuracy of pseudo backgrounds, while the dual boundary detector effectively improves the localization precision of action instances. Moreover, our method achieves significant improvements across different backbone architectures on three benchmarks.

Acknowledgements

This work was supported in part by National Science and Technology Major Project under Grant 2023ZD0121300, National Natural Science Foundation of China under Grants 62088102, 12326608 and 62106192, Natural Science Foundation of Shaanxi Province under Grant 2022JC-41, and Fundamental Research Funds for the Central Universities under Grant XTR042021005.

References

1. Bai, Y., Wang, Y., Tong, Y., Yang, Y., Liu, Q., Liu, J.: Boundary content graph neural network for temporal action proposal generation. In: ECCV. pp. 121–137 (2020)

2. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR. pp. 6299–6308 (2017)
3. 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: CVPR. pp. 1130–1139 (2018)
4. Chen, M., Gao, J., Yang, S., Xu, C.: Dual-evidential learning for weakly-supervised temporal action localization. In: ECCV. pp. 192–208 (2022)
5. Cheng, F., Bertasius, G.: Tallformer: Temporal action localization with a long-memory transformer. In: ECCV. pp. 503–521 (2022)
6. Damen, D., Leelasawassuk, T., Haines, O., Calway, A., Mayol-Cuevas, W.W.: You-do, i-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In: BMVC. vol. 2, p. 3 (2014)
7. Gao, J., Chen, M., Xu, C.: Fine-grained temporal contrastive learning for weakly-supervised temporal action localization. In: CVPR. pp. 19999–20009 (2022)
8. 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)
9. Ju, C., Zhao, P., Chen, S., Zhang, Y., Wang, Y., Tian, Q.: Divide and conquer for single-frame temporal action localization. In: ICCV. pp. 13455–13464 (2021)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Lee, P., Byun, H.: Learning action completeness from points for weakly-supervised temporal action localization. In: ICCV. pp. 13648–13657 (2021)
12. Lee, P., Uh, Y., Byun, H.: Background suppression network for weakly-supervised temporal action localization. In: AAAI. pp. 11320–11327 (2020)
13. Lei, P., Todorovic, S.: Temporal deformable residual networks for action segmentation in videos. In: CVPR. pp. 6742–6751 (2018)
14. Li, Z., Abu Farha, Y., Gall, J.: Temporal action segmentation from timestamp supervision. In: CVPR. pp. 8365–8374 (2021)
15. Lin, C., Xu, C., Luo, D., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Fu, Y.: Learning salient boundary feature for anchor-free temporal action localization. In: CVPR. pp. 3320–3329 (2021)
16. Lin, T., Liu, X., Li, X., Ding, E., Wen, S.: Bmn: Boundary-matching network for temporal action proposal generation. In: ICCV. pp. 3889–3898 (2019)
17. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: Bsn: Boundary sensitive network for temporal action proposal generation. In: ECCV. pp. 3–19 (2018)
18. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2980–2988 (2017)
19. Liu, D., Jiang, T., Wang, Y.: Completeness modeling and context separation for weakly supervised temporal action localization. In: CVPR. pp. 1298–1307 (2019)
20. Liu, X., Wang, Q., Hu, Y., Tang, X., Zhang, S., Bai, S., Bai, X.: End-to-end temporal action detection with transformer. IEEE TIP **31**, 5427–5441 (2022)
21. Liu, Z., Wang, L., Zhang, Q., Gao, Z., Niu, Z., Zheng, N., Hua, G.: Weakly supervised temporal action localization through contrast based evaluation networks. In: ICCV. pp. 3899–3908 (2019)
22. Ma, F., Zhu, L., Yang, Y., Zha, S., Kundu, G., Feiszli, M., Shou, Z.: Sf-net: Single-frame supervision for temporal action localization. In: ECCV. pp. 420–437 (2020)
23. Moltisanti, D., Fidler, S., Damen, D.: Action recognition from single timestamp supervision in untrimmed videos. In: CVPR. pp. 9915–9924 (2019)
24. Nag, S., Zhu, X., Song, Y.Z., Xiang, T.: Proposal-free temporal action detection via global segmentation mask learning. In: ECCV. pp. 645–662 (2022)

25. Narayan, S., Cholakkal, H., Khan, F.S., Shao, L.: 3c-net: Category count and center loss for weakly-supervised action localization. In: ICCV. pp. 8679–8687 (2019)
26. Nguyen, P., Liu, T., Prasad, G., Han, B.: Weakly supervised action localization by sparse temporal pooling network. In: CVPR. pp. 6752–6761 (2018)
27. Nguyen, P.X., Ramanan, D., Fowlkes, C.C.: Weakly-supervised action localization with background modeling. In: ICCV. pp. 5502–5511 (2019)
28. Paul, S., Roy, S., Roy-Chowdhury, A.K.: W-talc: Weakly-supervised temporal activity localization and classification. In: ECCV. pp. 563–579 (2018)
29. Qing, Z., Su, H., Gan, W., Wang, D., Wu, W., Wang, X., Qiao, Y., Yan, J., Gao, C., Sang, N.: Temporal context aggregation network for temporal action proposal refinement. In: CVPR. pp. 485–494 (2021)
30. Rashid, M., Kjellstrom, H., Lee, Y.J.: Action graphs: Weakly-supervised action localization with graph convolution networks. In: ICCV. pp. 615–624 (2020)
31. Rubinstein, R.Y., Kroese, D.P.: Simulation and the Monte Carlo method. John Wiley & Sons (2016)
32. Shao, J., Wang, X., Quan, R., Zheng, J., Yang, J., Yang, Y.: Action sensitivity learning for temporal action localization. arXiv preprint arXiv:2305.15701 (2023)
33. Shi, B., Dai, Q., Mu, Y., Wang, J.: Weakly-supervised action localization by generative attention modeling. In: CVPR. pp. 1009–1019 (2020)
34. Shi, D., Zhong, Y., Cao, Q., Zhang, J., Ma, L., Li, J., Tao, D.: React: Temporal action detection with relational queries. In: ECCV. pp. 105–121 (2022)
35. Tan, J., Tang, J., Wang, L., Wu, G.: Relaxed transformer decoders for direct action proposal generation. In: ICCV. pp. 13526–13535 (2021)
36. Tang, X., Fan, J., Luo, C., Zhang, Z., Zhang, M., Yang, Z.: Ddg-net: Discriminability-driven graph network for weakly-supervised temporal action localization. In: ICCV. pp. 6622–6632 (2023)
37. Ullah, W., Hussain, T., Min Ullah, F.U., Muhammad, K., Hassaballah, M., Rodrigues, J.J., Baik, S.W., Albuquerque, V.H.C.d.: Ad-graph: Weakly supervised anomaly detection graph neural network. IJIS **2023**, 1–12 (2023)
38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS **30** (2017)
39. Wang, L., Xiong, Y., Lin, D., Van Gool, L.: Untrimmednets for weakly supervised action recognition and detection. In: CVPR. pp. 4325–4334 (2017)
40. Wang, Q., Zhang, Y., Zheng, Y., Pan, P.: Rcl: Recurrent continuous localization for temporal action detection. In: ICCV. pp. 13566–13575 (2022)
41. Wedel, A., Pock, T., Zach, C., Bischof, H., Cremers, D.: An improved algorithm for tv-l 1 optical flow. In: SGAVMA (2009)
42. Xia, K., Wang, L., Shen, Y., Zhou, S., Hua, G., Tang, W.: Exploring action centers for temporal action localization. IEEE TMM **25**, 9425–9436 (2023)
43. Xia, K., Wang, L., Zhou, S., Hua, G., Tang, W.: Dual relation network for temporal action localization. PR **129**, 108725 (2022)
44. Xia, K., Wang, L., Zhou, S., Zheng, N., Tang, W.: Learning to refactor action and co-occurrence features for temporal action localization. In: CVPR. pp. 13884–13893 (2022)
45. Xu, H., Das, A., Saenko, K.: R-c3d: Region convolutional 3d network for temporal activity detection. In: ICCV. pp. 5783–5792 (2017)
46. Xu, M., Zhao, C., Rojas, D.S., Thabet, A., Ghanem, B.: G-tad: Sub-graph localization for temporal action detection. In: CVPR. pp. 10156–10165 (2020)
47. Yang, Z., Qin, J., Huang, D.: Acgnet: Action complement graph network for weakly-supervised temporal action localization. In: AAAI. vol. 36, pp. 3090–3098 (2022)

48. Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C.: Graph convolutional networks for temporal action localization. In: ICCV. pp. 7094–7103 (2019)
49. Zhai, Y., Wang, L., Tang, W., Zhang, Q., Yuan, J., Hua, G.: Two-stream consensus network for weakly-supervised temporal action localization. In: ECCV. pp. 37–54 (2020)
50. Zhang, C., Cao, M., Yang, D., Chen, J., Zou, Y.: Cola: Weakly-supervised temporal action localization with snippet contrastive learning. In: CVPR. pp. 16010–16019 (2021)
51. Zhang, C.L., Wu, J., Li, Y.: Actionformer: Localizing moments of actions with transformers. In: ECCV. pp. 492–510 (2022)
52. Zhang, C., Xu, Y., Cheng, Z., Niu, Y., Pu, S., Wu, F., Zou, F.: Adversarial seeded sequence growing for weakly-supervised temporal action localization. In: ACM MM. pp. 738–746 (2019)
53. Zhang, H., Wang, X., Xu, X., Qing, Z., Gao, C., Sang, N.: Hr-pro: Point-supervised temporal action localization via hierarchical reliability propagation. arXiv preprint arXiv:2308.12608 (2023)
54. Zhang, H., Wang, X., Xu, X., Qing, Z., Gao, C., Sang, N.: Hr-pro: Point-supervised temporal action localization via hierarchical reliability propagation. In: AAAI. vol. 38, pp. 7115–7123 (2024)
55. Zhao, C., Thabet, A.K., Ghanem, B.: Video self-stitching graph network for temporal action localization. In: ICCV. pp. 13658–13667 (2021)