Information Sciences 630 (2023) 206–221

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

APSL: Action-positive separation learning for unsupervised temporal action localization

Yuanyuan Liu [a], Ning Zhou [a], Fayong Zhang [b], Wenbin Wang [a,*], Yu Wang [b], Kejun Liu [a], Ziyuan Liu [c]

[a] School of Computer Science, China University of Geosciences (Wuhan), Wuhan, China
[b] School of Geography and Information Engineering, China University of Geosciences (Wuhan), Wuhan, China
[c] School of Economics and Management, China University of Geosciences (Wuhan), Wuhan, China

ARTICLE INFO

Keywords:
Unsupervised temporal action localization
Clustering
Feature separation learning

ABSTRACT

Unsupervised temporal action localization in untrimmed videos is a challenging and open issue. Existing works focus on the "clustering + localization" framework for unsupervised temporal action localization. However, it heavily relies on features used for clustering and localization, *e.g.*, features implying potential background information would degrade the localization performance. To address this problem, we propose a novel Action-positive Separation Learning (APSL) method. APSL follows a novel "feature separation + clustering + localization" iterative procedure. First, we introduce a novel feature separation learning (FSL) module. FSL employs separation learning to identify action and background features in a video, and then refines and removes potential action-negative and background-negative features (*hard-to-locate*) from the identified features employing contrastive learning, thus obtaining action-positive features (*easy-to-locate*). Next, in "clustering" step, we apply clustering to the separated action-positive features to obtain action pseudo-labels. In "localization" step, with action pseudo-labels and action-positive features, we employ a temporal action localization module to locate action instance regions, in turn, improving the performance of clustering and FSL. The three steps learn iteratively and reinforce each other during training. Comprehensive evaluations conducted on the THUMOS'14 and ActivityNet v1.2 datasets demonstrate that our method outperforms cutting-edge weakly supervised and unsupervised methods, obtaining state-of-the-art performance.

1. Introduction

Video temporal action localization is a challenging and interesting research direction in computer vision. It has generated a great deal of enthusiasm in recent years. The goal of video temporal action localization is to determine the exact start and end time of each action instance from a long, untrimmed video. Video temporal action localization has a variety of potential real-life applications, including video summarization, video highlight detection, and surgical skill assessment among others [1–4].

For video temporal action localization, most of the existing work focuses on fully and weakly supervised learning methods, and remarkable progress has been made [5–7]. However, these methods are heavily dependent on a large amount of training data with

* Corresponding author.
E-mail addresses: liuyy@cug.edu.cn (Y. Liu), zhouning@cug.edu.cn (N. Zhou), zhangfayong@cug.edu.cn (F. Zhang), wangwenbin@cug.edu.cn (W. Wang), vvy190701@cug.edu.cn (Y. Wang), liukejun0927@163.com (K. Liu), liuziyuan@cug.edu.cn (Z. Liu).

https://doi.org/10.1016/j.ins.2023.02.047
Received 9 October 2022; Received in revised form 8 February 2023; Accepted 9 February 2023

Available online 15 February 2023

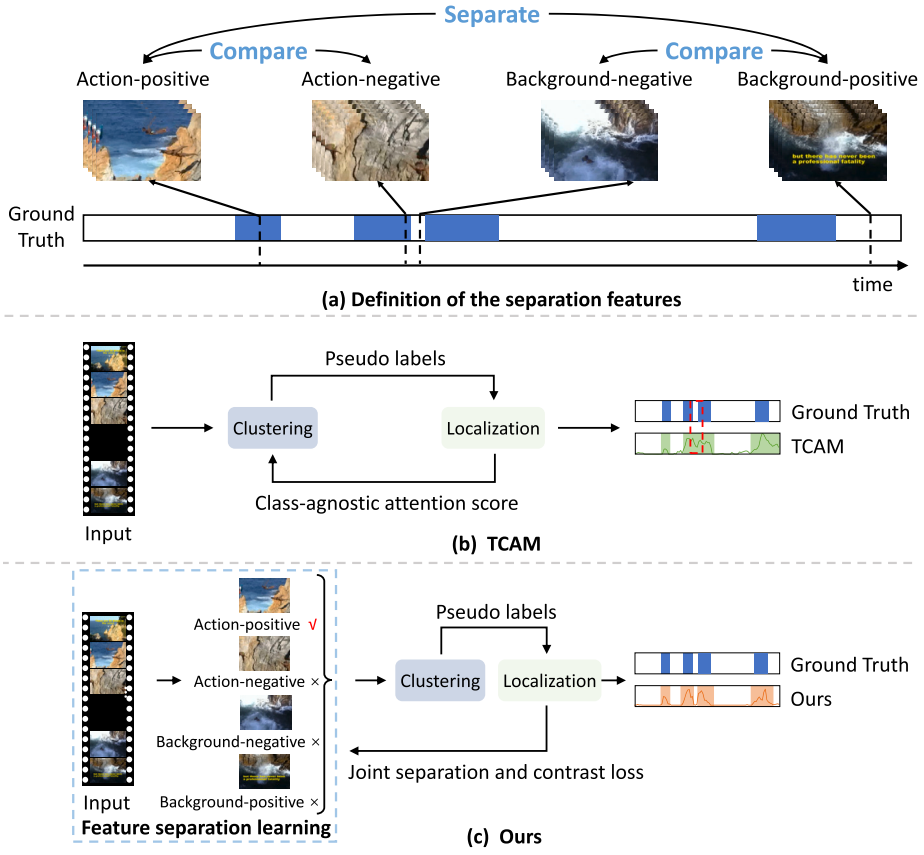

Fig. 1. The motivation of our method. (a) Definition of action-positive, background-positive, action-negative and background-negative features in a video. Our method attempts to extract the action-positive features for robust temporal action localization via joint separation and contrastive learning. (b) and (c) show the frameworks and localized results of TCAM and our method, respectively. The red dashed box in (b) shows the part where TCAM locates the error. Compared to the TCAM, our APSL follows a novel "feature separation + clustering + localization" iterative learning procedure. During the procedure, the more action-positive features are used for clustering and localization, the more accurate action localization can be obtained, meanwhile, the feature separation is improved.

accurate annotation information. For example, the fully supervised methods require very precise frame-level instance annotation information (*i.e.*, start timestamps and end timestamps for each action instance), which is difficult, burdensome, and extremely costly to annotate [8]. Although the weakly supervised methods do not require frame-level action region annotation, they still require video-level action category labels, and action category annotation for large amounts of untrimmed video is also very difficult and expensive [8].

Increasing number of recent research is focusing on the task of unsupervised temporal action localization. Recently, Gong et al. [9] proposed Temporal Co-Attention Models (TCAM) for unsupervised video temporal action localization. TCAM first used a clustering algorithm to group videos into C classes, and then used the video-level features to locate action instances and update the action pseudo-labels of action instances via attention learning. TCAM proposed a good framework for unsupervised temporal action localization. However, during the iterative learning procedure, we observe that the temporal action localization can be heavily affected by the video-level feature extraction when much negative information, such as background information is used for clustering (Fig. 1 (a)), thereby degrading the localization results. Fig. 1 (b) shows an intuitive example of the affected localization results, where the red dashed box indicates an error result located by TCAM.

To address the aforementioned problem in TCAM, we propose the Action-positive Separation Learning (APSL) method for unsupervised video action localization. The main advances and differences between our method and TCAM are shown in Fig. 1. Fig. 1 (a) shows the classification of the four types of features in videos (*i.e.*, action-positive, action-negative, background-positive, and background-negative) by APSL, and Figs. 1 (b) and (c) illustrate the frameworks of the TCAM and APSL for the unsupervised task, respectively. Compared with TCAM, our APSL uses a new unsupervised learning framework and introduces novel learning mechanisms for more salient action representation. As shown in Figs. 1 (a) and (c), the action-positive and background-positive features correspond to features that are *easy-to-locate* away from the boundaries and can be first separated by the APSL, whereas the action-negative and background-negative features are *hard-to-locate* features that are in the boundary region between action and background and are then compared by APSL to keep them away. Instead of using the video-based spatio-temporal features in TCAM, APSL first separates more salient action-positive features via a novel feature separation learning (FSL) module. With the help of effective action-positive feature separation, APSL clusters the *easy-to-locate* features that contribute most to temporal action

localization, while localizing the action regions that contribute most to clustering and feature separation. Therefore, different from the two-stage learning in TCAM, the three steps of APSL can be learned iteratively in an end-to-end and mutually reinforcing manner. In summary, the major contributions of this paper are summarized as follows:

(1) We propose the novel APSL method for unsupervised temporal action localization without any action annotations. The APSL follows the "feature separation + clustering+ localization" iterative procedure and reinforces each other during training.

(2) We propose a plug-and-play FSL module to acquire the more salient action-positive features by introducing the action-background separation loss and negative contrast loss. The former is used to separate the action features from the background features, and the latter further refines and removes the potentially action-negative and background-negative information. Both learning mechanisms pinpoint the precise video-level action categories and temporal-level action regions.

(3) Extensive experiments on the THUMOS'14 and ActivityNet v1.2 datasets demonstrate the effectiveness and robustness of our method in both unsupervised and weakly supervised tasks.

2. Related work

2.1. Fully supervised action localization

Fully supervised temporal action localization locates and classifies the time intervals of action occurrences in large untrimmed videos that use frame-level annotations. Nowadays, most work can be divided into two types, namely, one-stage and two-stage methods. One-stage methods can predict the location as well as the action classes simultaneously. For example, Long *et al.* [10] proposed GTAN with Gaussian kernels to implement one-stage temporal action localization. Recently, Xu *et al.* [5] applied graph convolutional neural networks for one-stage action localization. The two-stage methods generate action proposals first and then classify them, and finally do the regression with time bounds. Earlier methods for generating proposals used the sliding window technique [11], whereas more recent models combined reliable start and end frames of actions [12]. Although the previous models have achieved great performance, their scalability and utility in the actual world are limited by the fully supervised setup [13,14].

2.2. Weakly supervised action localization

Weakly supervised learning-based methods only require video-level annotation to locate the action instances in videos, and are mainly classified into metric learning-based, erasure-based, multi-branching and multi-attention architecture-based approaches. STPN [15] and AutoLoc [16] pioneered the method for localizing action instances by setting class activation sequence thresholds, and most subsequent techniques have been followed. W-TALC [17] used metric learning to push features of the same action to be closer to each other than features of distinct analogs. Hide and Seek [18] attempted to extend the region of distinction by randomly hiding patches or suppressing dominant responses. To discover complete action occurrences, HAM-Net [19] used mixed attention weight to localize complete action instances through multiple parallel and complementary branch learning. Uncertainty Modeling [20] recently investigated frame inconsistency and modeled background frames as out-of-distribution samples, thereby achieving the separation of background and action. CoLA [21] introduced contrastive learning to refine the boundary fragment feature representation, thus reducing the interference caused by boundary fragments. Although CoLA achieved good results, it directly used contrast learning for hard snippet mining, which could lead to suboptimal results due to less significant positive samples for contrast learning. To address this issue, our method first separates action features from background features by separation learning to ensure the number of positive samples, and then refines the hard samples by contrast learning to obtain fine action localization. In summary, weakly supervised temporal action localization still relies on video-level action labels, and action category annotation for large amounts of untrimmed video is also very difficult and expensive [8].

2.3. Unsupervised action localization

An increasing number of recent research efforts have focused on the task of unsupervised temporal action localization because it does not rely on any video annotations. Unsupervised temporal action localization only requires the knowledge of the number of action classes. TCAM [9] proposed the first unsupervised action localization method, which was a two-step "clustering + localization" iterative procedure. TCAM first used a clustering algorithm to obtain action pseudo-labels, and then used the video-level features to locate action instances and update the action pseudo-labels of action instances via attention learning. Despite the progress achieved in unsupervised temporal action localization, TCAM used the overall video features for clustering and localization and obtained suboptimal results because the used features contain several negative information about the background. To address the limitation in TCAM, we propose APSL to select more salient action-positive features for boosting both clustering and localization in unsupervised temporal action localization. First, instead of conventional "clustering + localization" framework in TCAM, APSL follows a novel "feature separation + clustering + localization" iterative procedure. Using the untrimmed video input, we first introduce an easy-to-plug FSL module to extract the more salient action-positive feature and then apply the video-level clustering and clip-level temporal localization to obtain the action pseudo labels and action instance regions, respectively. During the procedure, more action-positive features are used for clustering and localization to obtain accurate action localization. Meanwhile, feature separation is improved. Second, to enable FSL to separate more salient action-positive features from *difficult-to-locate* information, two additional losses (i.e., the action-background separation loss and negative contrast loss) are proposed. Both learning mechanisms help APSL to obtain more precise video-level action categories and temporal-level action regions than TCAM.

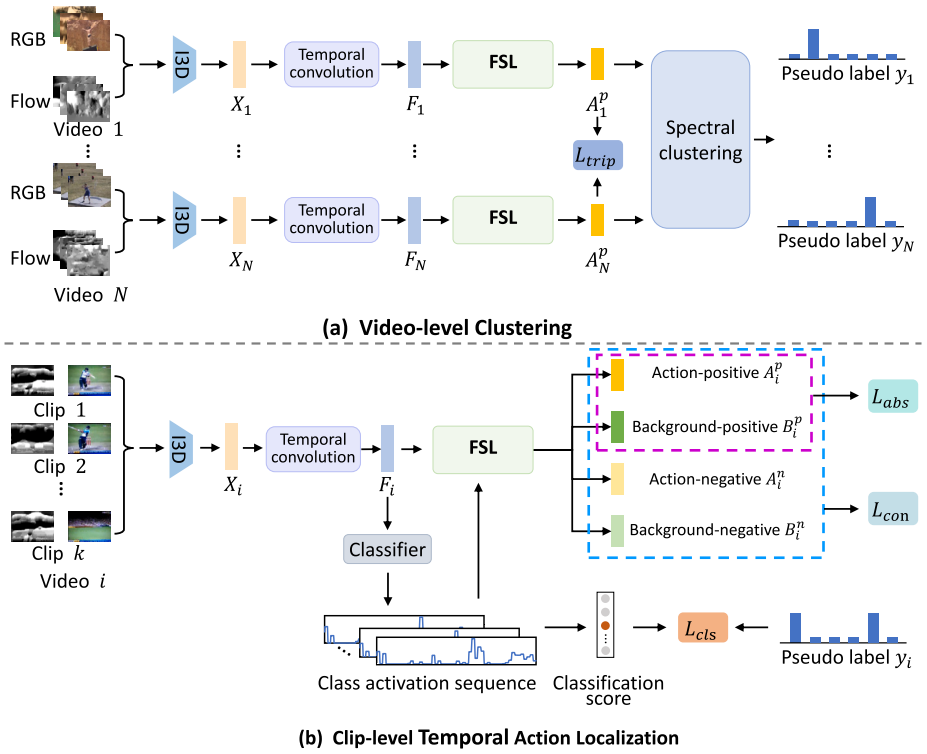

Fig. 2. The training pipeline of the APSL for unsupervised temporal action localization. Using the untrimmed video input, we apply the video-level clustering and clip-level temporal localization for respectively obtaining the action pseudo labels and action instance regions. Moreover, the APSL additionally introduces an easy-to-plug FSL with its losses to extract the more salient action-positive feature, to help obtain more accurate action clustering and localization.

2.4. Contrastive learning

As an important branch of deep metric learning [22], contrastive learning has recently made impressive progress in unsupervised/self-supervised visual representation learning. These approaches learned discriminative visual representations by contrasting positive pairs against negative ones. For example, SimCLR [23] proposed a negative sample selection scheme by using the augmented views of other items in a minibatch during training. MoCo [24] used a momentum updated memory bank of old negative representations to remove the batch size restriction and enable the consistent use of negative samples. In our work, we introduce contrastive learning into the FSL module for action-negative and background-negative feature segmentation, to remove potential *difficult-to-locate* information within the action and background features.

3. Methodology

In this section, we present the proposed APSL for unsupervised temporal action localization in detail. The overall APSL architecture is illustrated in Fig. 2. First, we apply the pre-trained I3D [25] with the input videos to extract and embed per-video task-specific spatio-temporal features. Second, we employ a video-level clustering module and clip-level temporal action localization module to achieve video-level pseudo-labels and temporal action regions. Meanwhile, a novel plug-and-play FSL module is introduced into the clustering and localization modules to identify and classify the spatio-temporal features of the video into four feature sub-spaces. In this way, clustering and localization modules can select the salient action-positive features (*easy-to-locate*) to enhance the clustering and localization performance. In the following sections, we will explain feature extraction and embedding, video-level clustering, clip-level temporal action localization, and FSL.

3.1. Feature extraction and embedding

Given an untrimmed video $\mathcal{V}$, we first decompose it into a series of smaller non-overlapping sub-videos: *i.e.*, $\mathcal{V} = \{v_t\}_{t=1}^{k}$, where v_t represents the t-th sub-video and k is the total number of sub-videos [26]. Each sub-video v_t refers to an action clip that contains several adjacent frames of the video. All the clips have the same length. Then, we use a pre-trained feature extractor I3D [25] to extract the per-clip RGB features as $x_t^{RGB} \in \mathbb{R}^d$, where d is the tensor dimension. Meanwhile, we extract per-clip optical flow feature as $x_t^{flow} \in \mathbb{R}^d$. Finally, we concatenate these two feature tensors to form the per-clip spatio-temporal feature tensor as $x_t = Concat(x_t^{RGB}, x_t^{flow})$. For the $i-th$ input video, we can obtain the video feature tensor as $X_i = [x_1, x_2, ..., x_k] \in \mathbb{R}^{2d \times k}$ [27,28].

Subsequently, we use a temporal convolutional operator and a ReLU [29] activation function, formally expressed as $F_i = g_{embed}(X_i; \phi_{embed})$, to embed the extracted spatio-temporal features X_i into the task-specific feature space, thereby obtaining the task-specific embedded features $F_i \in \mathbb{R}^{2d \times k}$. $g_{embed}(\cdot)$ represents the temporal convolutional operator, and ϕ_{embed} is the operator parameters. The final obtained task-specific embedded features F_i have the same size as the input features X_i.

3.2. Video-level clustering

In the absence of any valid video-level supervised information, we introduce the spectral clustering algorithm [30,31] with the cluster-based triplet loss to cluster the input video into C clusters, where C is the number of action categories in the training set. Given that an untrimmed video usually contains a lot of background frames, the direct use of the video embedded features F_i with the action-irrelevant information for clustering could lead to suboptimal results. Therefore, in this study, we employ an easy-to-plug FSL module (which will be described in section 3.4 below) to extract the action-positive features A_i^p. We also use A_i^p for the clustering to alleviate the influence of the action-irrelevant information. The detailed process of the video-level clustering is presented as follows.

Given the action-positive feature set $\{A_i^p\}_{i=1}^N$ of N videos in the training set as input, we build a fully connected affinity graph $G = \{A, E\}$, where A is the set of graph vertexes, and E is the set of its edges. Following the previous work [9], we first represent the action-positive feature of each video as a vertex in A. Then, we calculate the relation weight of any two vertexes, such as A_x^p and A_y^p, as the edge of the graph. The edge is given by:

$$e_{xy} = \exp\left(-\frac{\|A_x^p - A_y^p\|_2^2}{2\sigma^2}\right), \forall e_{xy} \in E \tag{1}$$

where $\sigma = \frac{1}{N^2}\sum_{x=1}^{N}\sum_{y=1}^{N}\|A_x^p - A_y^p\|_2$. $\|\|_2$ represents the Euclidean distance. The more similar the two vertexes are, the smaller the distance of the action-positive features will be, that is, the greater the weight of the relationship in Eq. (1) will be. A spectral clustering algorithm [30,31] is used to divide the G into C clusters with the constructed affinity graph G. Through such a clustering process, two vertices that are more similar are more likely to be grouped into a cluster. Finally, each cluster will be used to map a pseudo-action label and each video can be assigned with a pseudo-action label based on the clustering results. The pseudo-action label mapping can be seen in the section entitled “Pseudo label mapping”.

Cluster-based triplet loss. To produce more accurate pseudo-labels for action localization, we hope that the obtained clusters will have small intra-class spacing and large inter-class spacing. To this end, we introduce the cluster-based triple loss L_{trip} to pull the intra-class features of the same cluster closer and push the inter-class features of different clusters farther apart in the feature space. Formally, in a batch of K training videos, we suppose that videos v_a and v_c are in cluster z, and their distance is the maximum intra-class distance of z; whereas the video v_b is not in cluster z, and its inter-class distance with v_a is the smallest. Thus, we represent their action-positive features as A_a^p, A_b^p, and A_c^p, respectively. Mathematically, the cluster-based triplet loss is given by:

$$L_{trip} = \sum_{z=1}^{C}\sum_{a=1}^{K} max(dis(A_a^p, A_c^p) - dis(A_a^p, A_b^p) + h, 0) \tag{2}$$

where $dis(,)$ is the cosine distance. h represents the positive margin that is used to reduce the sensitivity of noise to clustering. Through the cluster-based triplet loss, the clusters of the same category will be tighter, and clusters of different categories will be further away.

3.3. Clip-level temporal action localization

In this part, we further perform clip-level temporal action localization in a weakly supervised learning manner with the clustered pseudo-action labels. As shown in Fig. 2 (b), given the embedded features F_i of the i-th video, we first use a liner classifier g_{cls} with a temporal convolution and a ReLU [29] activation function to predict the Class Activation Sequence (CAS):

$$S_i = g_{cls}(F_i; \phi_{cls}) \tag{3}$$

where $S_i = \{s_{i:c}\}_{c=1}^{C}$, and $s_{i:c} \in \mathbb{R}^{1 \times k}$ represents the CAS of the action class c. C is the number of clustered action classes, and k is the number of clips in the video. ϕ_{cls} represents the learned parameters of the linear classifier.

Then, we aggregate the top l scores of CAS for each action class and average them to obtain the per-action classification score:

$$a_{i:c} = \frac{1}{l} max \sum_{l} s_{i:c} \tag{4}$$

where $a_{i:c}$ is the classification score for the c-th action class. $l = \lfloor \frac{k}{r} \rfloor$, where k is the clip amount in the video, and r is a hyperparameter that controls the ratio of the aggregated clips.

Multi-label classification loss. For optimization, we introduce multi-label classification loss L_{cls} to predict the multiple action classes for each video. Mathematically, we use the cross-entropy loss as L_{cls}, which can be given by

$$L_{cls} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i:c} \log(p_{i:c}) \tag{5}$$

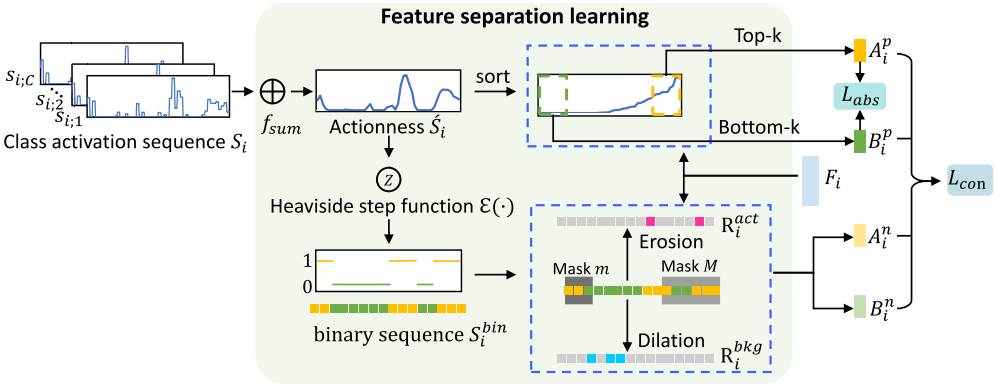

Fig. 3. The implementation pipeline of the FSL for effective feature separation.

where $p_{i;c} = \frac{\exp(a_{i;c})}{\sum_{j=1}^{C}\exp(a_{i;j})}$ represents the predicted classification possibilities for the c-th class of the i-th video, and $y_{i;c}$ is the generated video-level pseudo labels by the clustering.

Inference of action localization.

Given a video, we first calculate its CAS (*i.e.*, S_i) and then predict the video-level action possibilities $p_{i;c}$. Next, we set the threshold θ_{class} on $p_{i;c}$ to discover all the action classes c that satisfy $p_{i;c} > \theta_{class}$. For the retained action classes, we threshold its corresponding CAS with θ_{act} to obtain a set of localization proposals. Each proposal in a video has the form of (b_c, e_c, c), where b_c and e_c denote the start and end times of the c-th action class, respectively; and c is the predicted action class. Finally, we perform non-maximum suppression (NMS) [32] on all these clip proposals to remove duplicated proposals and achieve the final localization output.

3.4. Feature separation learning

Directly applying the aforementioned clustering and localization on raw videos can be suboptimal because of several potential, easily confused noises, and action-unrelated information within clips, thereby making it difficult to obtain robust action-related representation. To obtain more action-related information for action localization, some methods [21,33,34] introduced contrastive learning to locate the potential hard clips and refine their feature representation. However, the video contains many easy-to-locate clips that can greatly improve the efficiency of feature localization if they can be separated first. To this end, a FSL module boosts both clustering and localization to model the clip-based action-positive representation better by introducing the joint separation and contrast learning that can improve the encoding of action spatial-temporal information across clips. The implementation pipeline of the FSL is illustrated in Fig. 3.

The FSL first identifies and separates the action features from the background features (*easy-to-locate*) and then refines and removes action-negative and background-negative features (*hard-to-locate*) to obtain more robust action-positive information. FSL consists of two main steps (*i.e.*, positive feature separation and negative feature removal, each of which is a plug-and-play module that is easy to implement).

Using per-action CAS $s_{i;c}$ of the i-th video from the temporal action localization, which represents the degree of activation of each action in a video, we first utilize the summation of $s_{i;c}$ along the action class amount and apply the Sigmoid function to obtain the actionness $\acute{S}_i \in \mathbb{R}^k$ as:

$$\acute{S}_i = Sigmoid(\sum_c s_{i;c}), \tag{6}$$

where k is the number of clips in a video. Then, the two following steps are used for feature separation and contrast learning.

3.4.1. Positive feature separation

We sort the actionness $\acute{S}_i$ and sample the features with the top-k and bottom-k actionness scores as the action-positive features $A_i^p \in \mathbb{R}^{2d\times k^p}$ and background-positive features $B_i^p \in \mathbb{R}^{2d\times k^p}$, respectively. Mathematically, the sampling process is presented as follows:

$$A_i^p = \{f_{i;t} \mid t \in S_i^{act}, S_i^{act} = S_i^{DESC}[:k^p], f_{i;t} \in F_i\} \tag{7}$$

$$B_i^p = \{f_{i;t} \mid t \in S_i^{bkg}, S_i^{bkg} = S_i^{ASC}[:k^p], f_{i;t} \in F_i\} \tag{8}$$

where S_i^{DESC} and S_i^{ASC} denote the index of $\acute{S}_i$ after sorting by descending and ascending order, respectively. $k^p = max(1, \lfloor \frac{k}{r^p} \rfloor)$, and r^p is a hyperparameter that controls the ratio of the selected positive clips from a video.

Action-background separation loss. Inspired by the previous work [20], we observe that action clips usually have larger feature magnitudes than background clips in videos. Therefore, we introduce an action-background separation loss L_{abs} to ensure that the action-positive features A_i^p and background-positive features B_i^p are as far away as possible from each other in the feature space. Mathematically, the L_{abs} can be written as:

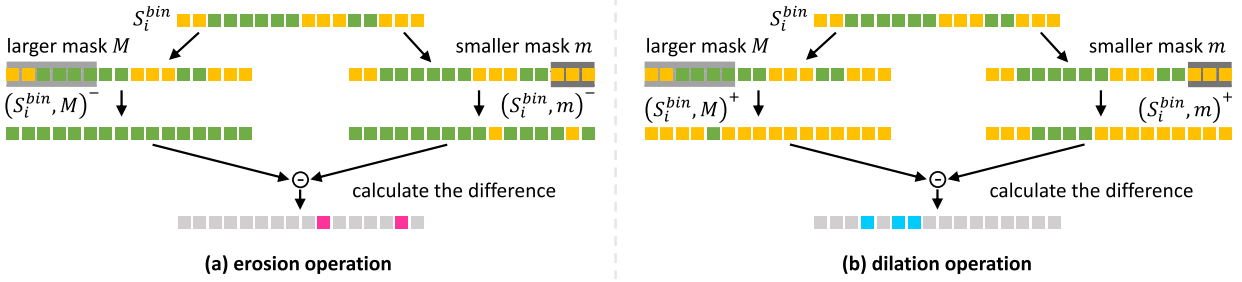

Fig. 4. The process of erosion and dilation for separating action-negative and background-negative clips from a sequence. (a) shows the process of subtracting the eroded sequences with different masks to obtain the action-positive clips. (b) shows the process of subtracting the dilated sequences with different masks to obtain the background-positive clips. **Note:** Yellow squares represent the action clips, green squares represent the background clips, gray squares represent the same part of the two sequences, pink squares represent the action-negative clips and blue squares represent the background-negative clips.

$$L_{abs} = \frac{1}{N}\sum_{i=1}^{N}(max(0, q - ||A_i^p||) + ||B_i^p||)^2 \tag{9}$$

where $||\cdot||$ is a norm function. q is a predefined maximum feature magnitude and is empirically set to 150. Through this learning process, the action and background features can be effectively separated in the feature space.

3.4.2. Negative feature removal

To remove potential *difficult-to-locate* information further within the action and background features, FSL introduces contrastive learning for action-negative and background-negative feature segmentation. We first binarize $\acute{S}_i$ to generate a binary sequence S_i^{bin} as (1 and 0 represent the action and background, respectively):

$$S_i^{bin} = \epsilon(\acute{S}_i - \theta_b) \tag{10}$$

where $\epsilon(\cdot)$ denotes the Heaviside step function, and θ_b represents the threshold value for the binarization. Then, we apply two cascaded erosion and dilation operations to narrow and expand S_i^{bin} with two different temporal intervals, respectively. With the different regions of erosion and dilation, we calculate the action-negative clips R_i^{act} and background-negative clips R_i^{bkg} in a video as:

$$R_i^{act} = (S_i^{bin}; m)^- - (S_i^{bin}; M)^- \tag{11}$$

$$R_i^{bkg} = (S_i^{bin}; M)^+ - (S_i^{bin}; m)^+ \tag{12}$$

where $(-;*)^-$ and $(-;*)^+$ represent the erosion and dilation operations with the mask *, respectively. M and m represent larger and smaller masks, respectively. Fig. 4 illustrates the process of erosion and dilation operations for separating action-negative and background-negative clips from a sequence by narrowing and expanding the S_i^{bin}, respectively. As shown in the figure, the erosion outputs the action clips when all clips in the mask neighborhood belong to the action clips (see the yellow), whereas the dilation outputs the action clips when one clip belongs to the action clips. More specifically, for the erosion operation, we use the larger mask M and the smaller mask m to erode the sequence S_i^{bin}. The erosion of S_i^{bin} with a larger mask M corresponds to a strict selection as $(S_i^{bin}, M)^-$, whereas the erosion of S_i^{bin} with a smaller mask m corresponds to a simple selection as $(S_i^{bin}, m)^-$. The larger mask can obtain action clips with high confidence in erosion. Therefore, the difference between $(S_i^{bin}, m)^-$ and $(S_i^{bin}, M)^-$ is the action-negative clip on the boundary. Similarly, for the dilation operation, we use the larger mask M and the smaller mask m to dilate the sequence S_i^{bin}, the dilation of S_i^{bin} with a larger mask M corresponds to a simple expansion of the action clips as $(S_i^{bin}, M)^+$, whereas the dilation of S_i^{bin} with a smaller mask m corresponds to a strict expansion of the action clips as $(S_i^{bin}, m)^+$. The smaller mask can obtain background clips with high confidence in dilation. Therefore, the difference between $(S_i^{bin}, M)^+$ and $(S_i^{bin}, m)^+$ is the background-negative clip on the boundary. The details of erosion and dilation operations can be seen in this work [35].

Next, similar to positive feature sampling, we select k^n action-negative clips from R_i^{act} to form the action-negative features $A_i^n \in \mathbb{R}^{2d \times k^n}$, where $k^n = max(1, \lfloor \frac{k}{r^n} \rfloor)$, and r^n is a hyperparameter that controls the ratio of the selected negative clips in a video. Similarly, we can segment the background-negative features $B_i^n \in \mathbb{R}^{2d \times k^n}$.

Negative contrast loss. For contrastive learning, we introduce the negative contrast loss L_{con} for optimization. Specifically, taking the action-negative feature of the i-the video as an example, we take the mean operation for A_i^n and A_i^p to obtain the $\overline{A_i^n} \in \mathbb{R}^{2d}$ and $\overline{A_i^p} \in \mathbb{R}^{2d}$, respectively, where d is the dimension of the clip features. For A_i^n, we represent A_i^p as the positive sample and B_i^p as the negative sample. Similarly, for the background-negative features B_i^n, we also average the features B_i^n and B_i^p to obtain the mean feature $\overline{B_i^n} \in \mathbb{R}^{2d}$ and $\overline{B_i^p} \in \mathbb{R}^{2d}$. We represent B_i^p as the positive sample and A_i^p as the negative sample. Following [24], we calculate the distances between the features and their positive and negative samples according to the negative contrast loss L_{con}:

$$L_{con} = \sum_{i=1}^{N} \log[-\frac{\exp((\overline{A_i^n})^T \cdot \overline{A_i^p}/\tau)}{\exp((\overline{A_i^n})^T \cdot \overline{A_i^p}/\tau) + \sum_{t=1}^{k^p} \exp((\overline{A_i^n})^T \cdot b_{i;t}^p/\tau)}]+ \sum_{i=1}^{N} \log[-\frac{\exp((\overline{B_i^n})^T \cdot \overline{B_i^p}/\tau)}{\exp((\overline{B_i^n})^T \cdot \overline{B_i^p}/\tau) + \sum_{t=1}^{k^p} \exp((\overline{B_i^n})^T \cdot a_{i;t}^p/\tau)}] \tag{13}$$

where τ is a hyperparameter in contrastive learning. $(\overline{A_i^n})^T$ and $(\overline{B_i^n})^T$ are the transposes of $\overline{A_i^n}$ and $\overline{B_i^n}$, respectively. $b_{i;t}^p \in B_i^p$ and $a_{i;t}^p \in A_i^p$ are the t-th elements of background-positive and action-positive features, respectively. k^p is the number of elements in B_i^p and A_i^p.

3.5. Overall training objectives

In summary, the APSL has four objectives for optimization, namely, the multi-label classification loss L_{cls}, the cluster-based triplet loss L_{trip}, the action-background separation loss L_{abs}, and the negative contrast loss L_{con}. Mathematically, the total loss corresponds the sum of these four losses and can be given by:

$$L_{total} = L_{cls} + \alpha \cdot L_{trip} + \beta \cdot L_{abs} + \gamma \cdot L_{con}, \tag{14}$$

where α, β, and γ are the hyper-parameters for better balancing the network learning. In this study, we empirically set $\alpha = 0.005$, $\beta = 0.01$, and $\gamma = 0.005$.

4. Experiments and analysis

4.1. Datasets

We evaluated the APSL on two large benchmark datasets: THUMOS'14 [36] and ActivityNet v1.2 [37]. Both datasets contain untrimmed videos, that is, some frames in the videos are not from any target action.

THUMOS'14. THUMOS'14 includes 13,320 untrimmed videos. The video duration is highly variable, and each video may contain multiple action instances. For a fair comparison, following the previous methods [38,39], we used 413 untrimmed videos with 20 action classes in THUMOS'14, where 200 videos are from the validation set for training and 213 videos are from the test set for evaluation.

ActivityNet v1.2. ActivityNet v1.2 is a popular large benchmark action localization dataset that contains 4,819 training videos, 2,383 validation videos, and 2,480 testing videos with 100 action classes. Following previous work [16], we trained our model on the training set and test on the validation set.

4.2. Implementation details

4.2.1. Feature extractor setting

Following previous methods [38,15,39,40], we used the pre-trained I3D [25] model on the Kinetics-400 [25] dataset to extract RGB and optical flow features. I3D took the non-overlapping clips of 16 stacked RGB or optical flow frames as input and extracted the 1024-dimensional feature for each stream. We adopted the fusion of the RGB and optical flow feature fed into APSL before to generate the final action localization.

4.2.2. Pseudo label mapping

By clustering, we obtained the C clusters, but we only knew the cluster index to which each video belongs to. To make comparisons with other fully or weakly supervised methods, we must further map the cluster indices to action classes to obtain the class label. Considering that some videos may contain multiple action classes, we map each cluster to one or more action classes. We referred to the previous work TCAM [9] for the mapping process. First, suppose in cluster c, we counted the times of action class labels (note that the action class labels were only used when counting the times and were not involved in the training of the model). If y_1 was the most frequently occurring action class and appears t times, then we select the action class labels y_c w.r.t its number of occurrences $\geq \frac{t}{2}$. As a result, the final action pseudo labels y_c of the cluster c can be mapped in a multi-label manner.

4.2.3. Key training parameters

The number of clips in a video k was set to 750 and 50 for THUMOS'14 and ActivityNet v1.2, respectively. We utilized the Adam optimizer with a learning rate of 0.0001. For clarification, other key training parameters are shown in Table 1.

4.2.4. Testing details

For THUMOS'14 and ActivityNet v1.2, we set θ_{class} to 0.2 and 0.1 to determine which action classes are to be localized. We used multiple thresholds for proposal generation. For THUMOS'14, we set θ_{act} to [0.325:0.375:0.025]. For ActivityNet v1.2, we set θ_{act} to [0:0.15:0.015] and then performed non-maximum suppression (NMS) using a threshold of 0.5.

Table 1
The key training parameters involved in this work.

Parameters	Description of the parameters	Values
d	Tensor dimension of each clip	1024
h	Positive margin in L_{trip}	0.8
r	The ratio of the aggregated clips	8
r^p	The ratio of the selected positive clips	5
θ_b	Threshold value for binarization	0.8
r^n	The ratio of the selected negative clips	20
m	The smaller mask	3
M	The larger mask	6
τ	The hyperparameter in contrastive learning	0.07
q	Predefined maximum feature magnitude	150

Table 2
Comparison of action detection on the THUMOS'14 dataset. We denote fully supervised, weakly supervised and unsupervised as FS, WS and US, respectively. The best results are in bold.

Supervision	Method	mAP@t-IoU (%)							
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	Avg
FS	S-CNN(2016) [11]	47.7	43.5	36.3	28.7	19	-	-	-
	SSN(2017) [1]	66	59.4	51.9	41	29.8	-	-	-
	TAL-Net(2018) [42]	59.8	57.1	53.2	48.5	42.8	33.8	20.8	45.1
	GTAN(2019) [10]	69.1	63.7	57.8	47.2	38.8	-	-	-
	TSI(2020) [6]	-	-	61.0	52.1	42.6	33.2	22.4	-
WS	Hide-and-Seek(2017) [18]	36.4	27.8	19.5	12.7	6.8	-	-	-
	AutoLoc(2018) [16]	-	-	35.8	29	21.2	13.4	5.8	-
	STPN(2018) [15]	52	44.7	35.5	25.8	16.9	9.9	4.3	27
	W-TALC(2018) [17]	55.2	49.6	40.1	31.1	22.8	-	7.6	-
	CMCS(2019) [38]	57.4	50.8	41.2	32.1	23.1	15	7	32.4
	DGAM(2020) [39]	60	54.2	46.8	38.2	28.8	19.8	11.4	37
	TCAM(2020) [9]	-	-	46.9	38.9	30.1	19.8	10.4	-
	Bas-Net(2020) [43]	58.2	52.3	44.6	36	27	18.6	10.4	35.3
	RefineLoc(2021) [44]	-	-	40.8	32.7	23.1	13.3	5.3	-
	Liu et al.(2021) [45]	-	-	50.8	41.7	29.6	20.1	10.7	-
	HAM-Net(2021) [19]	65.9	59.6	52.2	43.1	32.6	21.9	12.5	41.1
	Uncertainty Modeling(2021) [20]	67.5	61.2	52.3	43.4	33.7	22.9	12.1	41.9
	CoLA(2021) [21]	66.2	59.5	51.5	41.9	32.2	22	**13.1**	40.9
	D2-Net(2021) [7]	65.7	60.2	52.3	43.4	**36**	-	-	-
	FAC-Net(2021) [46]	67.6	62.1	52.6	44.3	33.4	22.5	12.7	42.2
	ACGNet(2022) [47]	68.1	**62.6**	53.1	**44.6**	34.7	22.6	12	42.5
	Ours	**69.1**	62.4	**53.7**	43.6	33.6	**23.8**	12.8	**42.7**
US	TCAM(2020) [9]	-	-	39.6	32.9	25	16.7	8.9	-
	Ours	**57.7**	**52.4**	**44.1**	**35.9**	**27.9**	**18.5**	**10**	**35.2**

4.2.5. Evaluation metrics

We evaluated our method with mean Average Precision (mAP) under several different intersection over union (IoU) thresholds, which were the standard evaluation metrics for temporal action localization. Both datasets used the benchmark code provided by ActivityNet [37]. In addition, we employed the normalized mutual information (NMI) score and adjusted rand index (ARI) to measure the clustering performance, which have been widely used in clustering tasks [41].

4.3. Comparisons with state-of-the-arts

We compared our method with the existing fully supervised, weakly supervised, as well as unsupervised methods under several IoU thresholds. In the weakly supervised case, APSL only used the temporal action localization and FSL, thereby removing the video-level clustering because action category labels were known.

4.3.1. Evaluation on THUMOS'14 dataset

Table 2 summarizes the results of the THUMOS'14 test set when the IoU threshold varies between 0.1 and 0.7. For mAP@0.5, our method achieved 27.9% in the unsupervised case, which was a 2.9% improvement compared with TCAM. It indicated the effectiveness of FSL in unsupervised temporal action localization. In addition, for mAP@Avg, our unsupervised APSL method achieved 35.2%, which was even better than the results of other state-of-the-art unsupervised methods. In the weakly supervised case, our method also achieved the best result of 42.7% on THUMOS'14, implying that the proposed FSL module is still valid for the weakly supervised framework. In addition, we achieved good results on two widely used metrics in the evaluation of clustering, obtaining

Table 3
Comparison of action detection on the ActivityNet v1.2 dataset. We denote fully supervised, weakly supervised and unsupervised as FS, WS and US, respectively. The best results are in bold.

Supervision	Method	mAP@t-IoU(%)			
		0.5	0.75	0.95	Avg
FS	SSN(2017) [1]	41.3	27	6.1	26.6
WS	AutoLoc(2018) [16]	27.3	15.1	3.3	16.0
	W-TALC(2018) [17]	37	12.7	1.5	18.0
	CMCS(2019) [38]	36.8	22.9	5.6	22.4
	RPN(2020) [48]	37.6	23.9	5.4	23.3
	TSCN(2020) [49]	37.6	23.7	5.7	23.6
	BaS-Net(2020) [43]	38.5	24.2	5.6	24.3
	DGAM(2020) [39]	41	23.5	5.3	24.4
	TCAM(2020) [9]	40	25	4.6	24.6
	Uncertainty Modeling(2021) [19]	41.2	25.6	6	25.9
	CoLA(2021) [21]	42.7	25.7	5.8	26.1
	Ours	**44.3**	**28.5**	**6.2**	**28.2**
US	TCAM(2020) [9]	35.2	21.4	3.1	21.1
	Ours	**43.7**	**28.1**	**5.8**	**27.6**

Table 4
Ablation study of different losses on the THUMOS'14 dataset. The best results are in bold.

Setting	mAP@0.5(%)
Baseline(L_{cls})	16.1
$L_{cls} + L_{abs}$	25.8
$L_{cls} + L_{abs} + L_{con}$	27.1
$L_{cls} + L_{abs} + L_{con} + L_{trip}$	**27.9**

0.821 on normalized mutual information score and 0.639 on adjust rand index, whereas TCAM only obtained 0.811 on normalized mutual information score and 0.612 on adjust rand index.

4.3.2. Evaluation on ActivityNet v1.2 dataset

The results on ActivityNet v1.2 are given in Table 3. Our method was compared with other state-of-the-art unsupervised, weakly supervised, and fully supervised action localization methods. As shown in the table, although without any annotation for videos, our method achieved a good result of 27.6% for mAP@Avg in the unsupervised case, which was even better than some unsupervised methods. In the weakly supervised case, our method improved the state-of-the-art method [21], achieving a mAP increase of 2.1% for mAP@Avg. Moreover, in the evaluation of clustering, we obtained 0.795 on normalized mutual information score and 0.574 on adjust rand index.

4.4. Ablation studies

4.4.1. Ablation study on different losses

To analyze the contribution of each loss, we performed ablation studies of losses on the THUMOS'14 dataset in the unsupervised case. The results are shown in Table 4. The baseline was set as the main pipeline only with multi-label classification loss L_{cls}. By introducing L_{abs}, the performance largely gained by 9.7% in mAP@0.5 partially because action-background separation loss L_{abs} can separate action and background very well. As shown in Table 4, the integration of the negative contrast loss L_{con} improved the performance by 1.3%, and further addition of the cluster-based triplet loss L_{trip} resulted in an increase of 0.8%. Finally, we used all losses to train the action localization model and achieved the best result of 27.9% in mAP@0.5.

4.4.2. Ablation study on different features

Table 5 reports the experimental results evaluated using the different features used for clustering and localization in the unsupervised case. As shown in Table 5, the embedded features F_i, which contained $A_i^p + B_i^p + A_i^n + B_i^n$, gained the result of 20.4% in mAP@0.5, thereby showing that the direct use of clip-level embedded features can bring additional noise within the clips for localization. Only using the action-negative A_i^n resulted in a decline in performance and obtained 15.2% in mAP@0.5, which indicated that A_i^n was a *hard-to-locate* features that contained fewer distinguished information for action localization. The integration of the action-positive A_i^p and action-negative A_i^n achieved 21.3% in mAP@0.5, thereby improving localization. This result verified that the action-positive feature A_i^p can improve the clustering and localization well. Finally, the sole use of the

Table 5
Ablation studies of different features on the THUMOS'14 dataset. The best results are in bold.

Setting	mAP@0.5(%)
Embedded features F_i $(A_i^p + B_i^p + A_i^n + B_i^n)$	20.4
Action-negative A_i^n	15.2
Action-positive A_i^p + action-negative A_i^n	21.3
Action-positive A_i^p	**27.9**

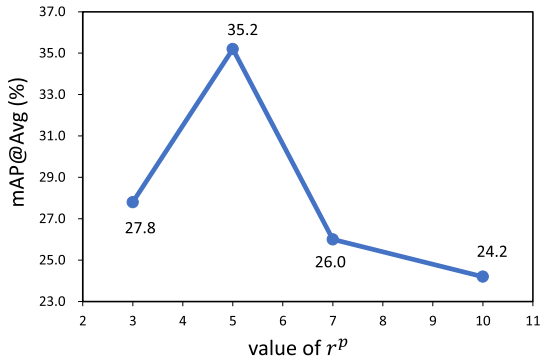

Fig. 5. Analysis of r^p for the effect of positive clips on the THUMOS'14 dataset. We report the mAP@Avg with varying r^p from 3 to 10.

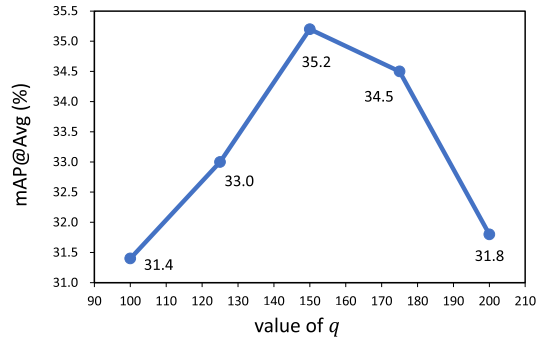

Fig. 6. The influence of q in the action-background separation loss on the THUMOS'14 dataset. We report the mAP@Avg with varying q from 100 to 200.

proposed action-positive features A_i^p achieved the best result of 27.9% in mAP@0.5, verifying that the separated action-positive features effectively removed the influence of negative non-action information via the joint separation and contrastive learning.

4.4.3. Analysis on r^p

$k^p = max(1, \lfloor \frac{k}{r^p} \rfloor)$ determines the number of selected positive clips from a video, and r^p is a hyperparameter that controls k^p. We investigated the effects of r^p in Fig. 5, where r^p was altered from 3 to 10. When r^p was too small, more positive clips were selected, and the clips related to background information were also selected, thereby resulting in lower performance. Meanwhile, when r^p was large, too fewer positive clips were selected, resulting in lower performance as well. In our experiments, the best results were obtained when r^p was 5.

4.4.4. Effects of parameter q

In the action-background separation loss (Eq. (9)), q is a hyperparameter that represents the maximum feature magnitude and is used to control the separation interface of action and background features. Fig. 6 presents the mAP@Avg of action localization with different q in the separation loss. As shown in the figure, the mAP@Avg researched the highest 35.2% when we set the parameter q to 150. We observed that when q was set relatively small, the separation of action-positive features and background-positive features was incomplete, resulting in degraded performance. On the contrary, when q was large, the excessive separation of action and background led to overfitting of the model, thus resulting in performance degradation. In our experiments, the best result was obtained when q was 150.

Table 6
"Only RGB" used only rgb features, "Only flow" used only flow features, "Early fusion" indicates that RGB and optical flow images are fused before inputting I3D network, and "Later fusion" indicates that RGB features and flow features are fused after output from I3D network. The best results are in bold.

feature fusion manner	mAP@Avg(%)
Only RGB	21.5
Only flow	16.1
Early fusion	23.0
Later fusion	**35.2**

Table 7
Comparison of model complexity and efficiency.

Method	Params(M)	MACs(G)	mAP@Avg(%)
Bas-Net	26.26	38.6	35.3
HAM-Net	29.15	21.86	41.1
Uncertainty Modeling	12.63	9.47	41.9
APSL	12.72	9.53	42.7

4.4.5. Effects of different feature fusion manners

To discuss the effects of different feature fusion manners for action localization, we used various feature fusion methods for RGB and optical flow features, and the results were shown in Table 6. "Only RGB" represents only using the extracted RGB features, "Only flow" represents only using the optical flow features, "Early fusion" indicates that RGB and optical flow images are fused before inputting I3D network, and "Later fusion" indicates that RGB features and flow features are fused after output from I3D network. The result of "Later fusion" is the best. The possible reason is that more detailed spatio-temporal information can be fully extracted for optical flow feature and RGB feature separately using the late fusion, and most of the current methods [38,15,39] use the later fusion of RGB and flow features.

4.5. Computational complexity

Table 7 reports the model parameters and computational cost of four temporal action localization methods in the weakly supervised case on the THUMOS'14 dataset. We use Multiply–Accumulate Operations (MACs)[1] to measure the computational cost. APSL has the best performance (mAP@Avg of 42.7%) with less computational cost (9.53G) and parameters (12.72M) among the compared methods, demonstrating that the proposed method exhibits improved accuracy and efficiency.

4.6. Visualization and qualitative results

Fig. 7 shows several visualization results on THUMOS'14 in the weakly supervised case. Compared with Uncertainty Modeling [20], APSL accurately locates the action clips in untrimmed videos. Fig. 7 (a) is an example of the frequent action case with VolleyballSpiking, which makes the localization difficult due to several potential, easy-confused noises and action-unrelated information within clips. Nonetheless, by separating action-positive and background-positive features and by removing the negative features, APSL can accurately locate the action-negative clips. Fig. 7 (b) shows an example of the Billiards action. The Uncertainty Modeling fails to localize the action clips, whereas our APSL successfully identifies and locates the action-negative clips, even at the boundary locations where motion and background information are blurred (see the red dashed boxes in Fig. 7).

In addition, Fig. 8 shows some visualization results on THUMOS'14 in the unsupervised case. The baseline is the TCAM framework [9] with the embedded features F_i. Compared with the TCAM framework, our APSL accurately recognizes the action clip positions in untrimmed videos, without using any video-level annotations. Fig. 8 (a) is a case of the CliffDiving action. As shown in Fig. 8 (a), some background-negative clips are incorrectly identified as action clips by TCAM with F_i, which indicates that F_i includes several potential, easy-confused noises and action-unrelated information within clips, thus enabling it to obtain suboptimal localization results. Fig. 8 (b) shows an example of the GolfSwing action, which is quite challenging, because the characteristics of the player preparing to swing (background-negative) and being swung (action-positive) have similarities, thereby making the model mislocalized. Nevertheless, our APSL can separate the action-positive features from background features well enough to obtain accurate action localization.

Fig. 9 shows the results of clustering visualization. We perform clustering on the ActivityNet v1.2 dataset and selected t-SNE [50] feature maps of nine categories for presentation. Fig. 9 (a) shows the feature maps of the embedded features F_i whereas Fig. 9

[1] https://github.com/sovrasov/flops-counter.pytorch.

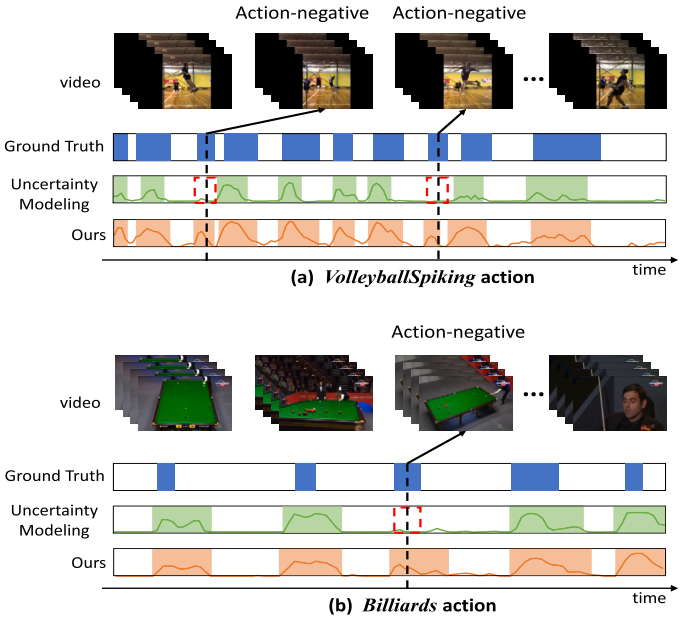

Fig. 7. Visualization results on THUMOS'14 in the weakly supervised case. The red dashed boxes show the unlocated action regions in the Uncertainty Modeling.

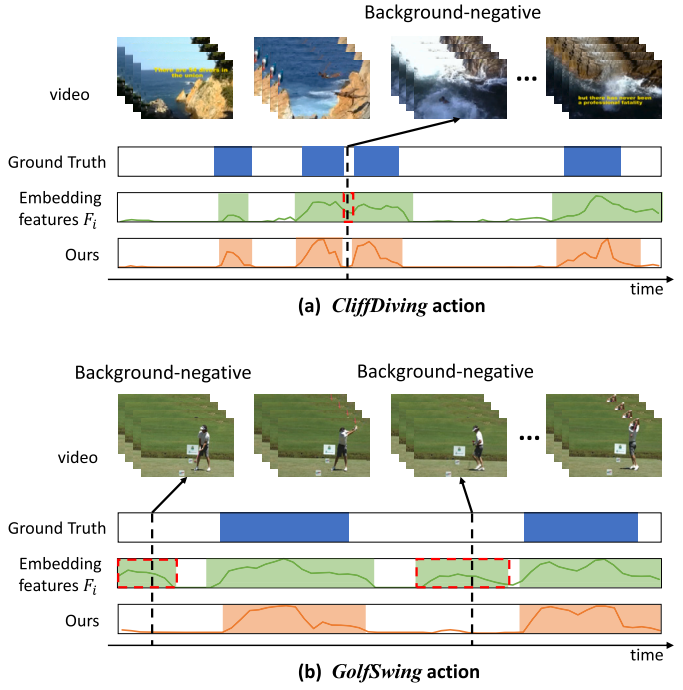

Fig. 8. Visualization results on THUMOS'14 in the unsupervised case. The red dashed boxes show the area where TCAM is mislocated.

(b) shows the feature maps of the proposed action-positive features A_i^p. The clustering that use A_i^p works better with each category being a compact cluster, whereas the clusters obtained using F_i are looser. The red dashed ellipses in Fig. 9 (a) show that the use of F_i clustering causes the Playing kickball category, Mixing drinks category and the Grooming horse category to not cluster effectively, because F_i contains information about the background, thereby leading to loose clusters. Meanwhile, Fig. 9 (b) obtains better clustering results for these three classes, showing that the feature A_i^p only contains action information and effectively enhances the clustering results.

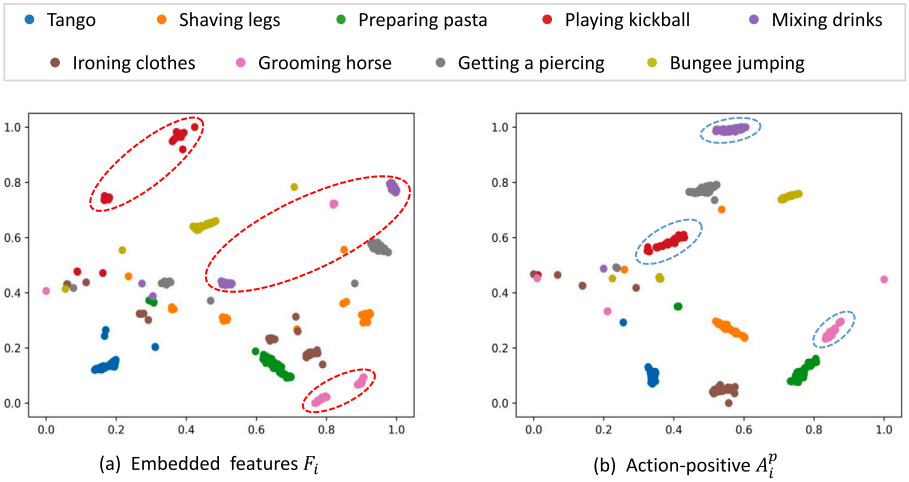

Fig. 9. The visualization results of clustering. The left figure shows the feature map of the video embedded features F_i while the right figure shows the feature map of the action-positive feature A_i^p. The red dashed ellipses in (a) show the loosely clustered categories via F_i while the blue dashed ellipses in (b) indicate the corresponding compact clusters.

5. Conclusion

In this work, we propose a novel APSL for unsupervised temporal action localization without any action annotations. APSL follows a novel "feature separation + clustering+ localization" iterative procedure. The "clustering" clusters the features to obtain video-level action pseudo-labels, and the "localization" uses the pseudo-labels to locate the action temporal regions, while improving the clustering performance. Moreover, we introduce the FSL module to identify and separate salient action-positive features from the video embedded features, to improve clustering and localization. We performed extensive experiments on two widely used temporal action localization datasets, namely THUMOS'14 and ActivityNet v1.2, in weakly supervised and unsupervised settings. The APSL exhibited state-of-the-art performance (*i.e.*, avg@mAP 35.2% and 27.6% in unsupervised setting, on THUMOS'14 and ActivityNet v1.2, respectively).

Despite the effectiveness of our method, some difficult problems that cause our APSL to not perform favorably are still observed. For example, in action localization, APSL only uses clips within the video for contrast learning to refine the action-negative and background-negative features, which results in a less abundant number of contrast learning samples and makes the contrast learning effect not optimal. Furthermore, APSL heavily depends on the performance of the clustering module, and poor clustering will lead to poor subsequent localization. The effective decoupling of the clustering and localization results is still a difficult problem. In the future, we will introduce contrast learning in clustering for coarse action localization and then continue to optimize the fine action positions by contrast learning in the localization phase to form a unified contrast-based clustering-localization paradigm, thus eliminating the effects of separating the clustering and localization phases.

CRediT authorship contribution statement

Yuanyuan Liu: Conceptualization, Methodology, Project administration, Validation, Writing – review & editing. **Ning Zhou:** Data curation, Methodology, Software, Visualization, Writing – original draft. **Fayong Zhang:** Data curation, Writing – review & editing. **Wenbin Wang:** Methodology, Writing – review & editing. **Yu Wang:** Data curation, Visualization, Writing – review & editing. **Kejun Liu:** Writing – review & editing. **Ziyuan Liu:** Writing – review & editing.

Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this manuscript and there has been no significant financial support for this work that could have influenced its outcome.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the National Natural Science Foundation of China grant (62076227), Wuhan Applied Fundamental Frontier Project Grant (2020010601012166).

References

[1] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, D. Lin, Temporal action detection with structured segment networks, in: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22–29, 2017, IEEE Computer Society, 2017, pp. 2933–2942.

[2] R. Yan, L. Xie, J. Tang, X. Shu, Q. Tian, Higcin: hierarchical graph-based cross inference network for group activity recognition, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.

[3] R. Yan, J. Tang, X. Shu, Z. Li, Q. Tian, Participation-contributed temporal dynamic model for group activity recognition, in: S. Boll, K.M. Lee, J. Luo, W. Zhu, H. Byun, C.W. Chen, R. Lienhart, T. Mei (Eds.), ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018, ACM, 2018, 2018, pp. 1292–1300.

[4] L. Wang, X. Yuan, M. Zong, Y. Ma, W. Ji, M. Liu, R. Wang, Multi-cue based four-stream 3d resnets for video-based action recognition, Inf. Sci. 575 (2021) 654–665, https://doi.org/10.1016/j.ins.2021.07.079.

[5] M. Xu, C. Zhao, D.S. Rojas, A.K. Thabet, B. Ghanem, G-TAD: sub-graph localization for temporal action detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, Computer Vision Foundation / IEEE, 2020, pp. 10153–10162.

[6] S. Liu, X. Zhao, H. Su, Z. Hu, TSI: temporal scale invariant network for action proposal generation, in: H. Ishikawa, C. Liu, T. Pajdla, J. Shi (Eds.), Computer Vision - ACCV 2020 - 15th Asian Conference on Computer Vision, Kyoto, Japan, November 30 - December 4, 2020, Revised Selected Papers, Part V, in: Lecture Notes in Computer Science, vol. 12626, Springer, 2020, pp. 530–546.

[7] S. Narayan, H. Cholakkal, M. Hayat, F.S. Khan, M. Yang, L. Shao, D2-net: weakly-supervised action localization via discriminative embeddings and denoised activations, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, IEEE, 2021, pp. 13588–13597.

[8] F. Ma, L. Zhu, Y. Yang, S. Zha, G. Kundu, M. Feiszli, Z. Shou, Sf-net: single-frame supervision for temporal action localization, in: A. Vedaldi, H. Bischof, T. Brox, J. Frahm (Eds.), Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IV, in: Lecture Notes in Computer Science, vol. 12349, Springer, 2020, pp. 420–437.

[9] G. Gong, X. Wang, Y. Mu, Q. Tian, Learning temporal co-attention models for unsupervised video action localization, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020, Computer Vision Foundation / IEEE, 2020, pp. 9816–9825.

[10] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, T. Mei, Gaussian temporal awareness networks for action localization, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE, 2019, pp. 344–353.

[11] Z. Shou, D. Wang, S. Chang, Temporal action localization in untrimmed videos via multi-stage cnns, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, IEEE Computer Society, 2016, pp. 1049–1058.

[12] C. Lin, J. Li, Y. Wang, Y. Tai, D. Luo, Z. Cui, C. Wang, J. Li, F. Huang, R. Ji, Fast learning of temporal action proposal via dense boundary generator, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, the Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, AAAI Press, 2020, pp. 11499–11506, https://ojs.aaai.org/index.php/AAAI/article/view/6815, 2020.

[13] J. Gao, C. Xu, Learning video moment retrieval without a single annotated video, IEEE Trans. Circuits Syst. Video Technol. 32 (3) (2022) 1646–1657, https://doi.org/10.1109/TCSVT.2021.3075470.

[14] J. Gao, T. Zhang, C. Xu, Graph convolutional tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019, Computer Vision Foundation / IEEE, 2019, pp. 4649–4659.

[15] P. Nguyen, T. Liu, G. Prasad, B. Han, Weakly supervised action localization by sparse temporal pooling network, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 6752–6761.

[16] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, S. Chang, Autoloc: weakly-supervised temporal action localization in untrimmed videos, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part XVI, in: Lecture Notes in Computer Science, vol. 11220, Springer, 2018, pp. 162–179.

[17] S. Paul, S. Roy, A.K. Roy-Chowdhury, W-TALC: weakly-supervised temporal activity localization and classification, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part IV, in: Lecture Notes in Computer Science, vol. 11208, Springer, 2018, pp. 588–607.

[18] K.K. Singh, Y.J. Lee, Hide-and-seek: forcing a network to be meticulous for weakly-supervised object and action localization, in: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22–29, 2017, IEEE Computer Society, 2017, pp. 3544–3553.

[19] A. Islam, C. Long, R.J. Radke, A hybrid attention mechanism for weakly-supervised temporal action localization, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, 2021, pp. 1637–1645.

[20] P. Lee, J. Wang, Y. Lu, H. Byun, Weakly-supervised temporal action localization by uncertainty modeling, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, 2021, pp. 1854–1862.

[21] C. Zhang, M. Cao, D. Yang, J. Chen, Y. Zou, Cola: weakly-supervised temporal action localization with snippet contrastive learning, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, June 19–25, 2021, Computer Vision Foundation / IEEE, 2021, pp. 16010–16019.

[22] M. Kaya, H.S. Bilge, Deep metric learning: a survey, Symmetry 11 (9) (2019) 1066, https://doi.org/10.3390/sym11091066.

[23] T. Chen, S. Kornblith, M. Norouzi, G.E. Hinton, A simple framework for contrastive learning of visual representations, in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, Virtual Event, 13-18 July 2020, in: Proceedings of Machine Learning Research, PMLR, vol. 119, 2020, pp. 1597–1607, http://proceedings.mlr.press/v119/chen20j.html.

[24] K. He, H. Fan, Y. Wu, S. Xie, R.B. Girshick, Momentum contrast for unsupervised visual representation learning, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, Computer Vision Foundation / IEEE, 2020, pp. 9726–9735.

[25] J. Carreira, A. Zisserman, Quo vadis, action recognition? A new model and the kinetics dataset, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, IEEE Computer Society, 2017, pp. 4724–4733.

[26] Y. Liu, C. Feng, X. Yuan, L. Zhou, W. Wang, J. Qin, Z. Luo, Clip-aware expressive feature learning for video-based facial expression recognition, Inf. Sci. 598 (2022) 182–195, https://doi.org/10.1016/j.ins.2022.03.062.

[27] Y. Wang, S. Zhou, Y. Liu, K. Wang, F. Fang, H. Qian, Congnn: context-consistent cross-graph neural network for group emotion recognition in the wild, Inf. Sci. 610 (2022) 707–724, https://doi.org/10.1016/j.ins.2022.08.003.

[28] A.J. Wang, Y. Ge, R. Yan, Y. Ge, X. Lin, G. Cai, J. Wu, Y. Shan, X. Qie, M.Z. Shou, All in one: exploring unified video-language pre-training, CoRR, arXiv: 2203.07303, 2022.

[29] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: G.J. Gordon, D.B. Dunson, M. Dudík (Eds.), Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011, in: JMLR Proceedings, vol. 15, 2011, pp. 315–323, JMLR.org, http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf.

[30] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905, https://doi.org/10.1109/34.868688.

[31] S. Ding, H. Jia, M. Du, Y. Xue, A semi-supervised approximate spectral clustering algorithm based on HMRF model, Inf. Sci. 429 (2018) 215–228, https://doi.org/10.1016/j.ins.2017.11.016.

[32] A. Neubeck, L.V. Gool, Efficient non-maximum suppression, in: 18th International Conference on Pattern Recognition (ICPR 2006), Hong Kong, China, 20-24 August 2006, IEEE Computer Society, 2006, pp. 850–855.
[33] X. Shu, B. Xu, L. Zhang, J. Tang, Multi-granularity anchor-contrastive representation learning for semi-supervised skeleton-based action recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2022).
[34] B. Xu, X. Shu, Y. Song, X-invariant contrastive augmentation and representation learning for semi-supervised skeleton-based action recognition, IEEE Trans. Image Process. 31 (2022) 3852–3867, https://doi.org/10.1109/TIP.2022.3175605.
[35] J. Gil, R. Kimmel, Efficient dilation, erosion, opening, and closing algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 24 (12) (2002) 1606–1617, https://doi.org/10.1109/TPAMI.2002.1114852.
[36] H. Idrees, A.R. Zamir, Y. Jiang, A. Gorban, I. Laptev, R. Sukthankar, M. Shah, The THUMOS challenge on action recognition for videos “in the wild”, Comput. Vis. Image Underst. 155 (2017) 1–23, https://doi.org/10.1016/j.cviu.2016.10.018.
[37] F.C. Heilbron, V. Escorcia, B. Ghanem, J.C. Niebles, Activitynet: a large-scale video benchmark for human activity understanding, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, IEEE Computer Society, 2015, pp. 961–970.
[38] D. Liu, T. Jiang, Y. Wang, Completeness modeling and context separation for weakly supervised temporal action localization, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE, 2019, pp. 1298–1307.
[39] B. Shi, Q. Dai, Y. Mu, J. Wang, Weakly-supervised action localization by generative attention modeling, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020, Computer Vision Foundation / IEEE, 2020, pp. 1006–1016.
[40] X. Shu, J. Yang, R. Yan, Y. Song, Expansion-squeeze-excitation fusion network for elderly activity recognition, IEEE Trans. Circuits Syst. Video Technol. 32 (8) (2022) 5281–5292, https://doi.org/10.1109/TCSVT.2022.3142771.
[41] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Am. Stat. Assoc. 66 (336) (1971) 846–850.
[42] Y. Chao, S. Vijayanarasimhan, B. Seybold, D.A. Ross, J. Deng, R. Sukthankar, Rethinking the faster R-CNN architecture for temporal action localization, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 1130–1139.
[43] P. Lee, Y. Uh, H. Byun, Background suppression network for weakly-supervised temporal action localization, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, the Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, AAAI Press, 2020, pp. 11320–11327.
[44] A. Pardo, H. Alwassel, F.C. Heilbron, A.K. Thabet, B. Ghanem, Refineloc: iterative refinement for weakly-supervised action localization, in: IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021, IEEE, 2021, pp. 3318–3327.
[45] Z. Liu, L. Wang, W. Tang, J. Yuan, N. Zheng, G. Hua, Weakly supervised temporal action localization through learning explicit subspaces for action and context, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, 2021, pp. 2242–2250.
[46] L. Huang, L. Wang, H. Li, Foreground-action consistency network for weakly supervised temporal action localization, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, IEEE, 2021, pp. 7982–7991.
[47] Z. Yang, J. Qin, D. Huang, Acgnet: action complement graph network for weakly-supervised temporal action localization, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, the Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, AAAI Press, 2022, pp. 3090–3098.
[48] L. Huang, Y. Huang, W. Ouyang, L. Wang, Relational prototypical network for weakly supervised temporal action localization, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, the Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, AAAI Press, 2020, pp. 11053–11060.
[49] Y. Zhai, L. Wang, W. Tang, Q. Zhang, J. Yuan, G. Hua, Two-stream consensus network for weakly-supervised temporal action localization, in: A. Vedaldi, H. Bischof, T. Brox, J. Frahm (Eds.), Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI, in: Lecture Notes in Computer Science, vol. 12351, Springer, 2020, pp. 37–54.
[50] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.