Weakly Supervised Temporal Action Localization via Representative Snippet Knowledge Propagation

Linjiang Huang[1,2] Liang Wang[3] Hongsheng Li[1,2] *
[1]CUHK-SenseTime Joint Laboratory, The Chinese University of Hong Kong
[2]Centre for Perceptual and Interactive Intelligence, Hong Kong
[3]Institute of Automation, Chinese Academy of Sciences
ljhuang524@gmail.com, wangliang@nlpr.ia.ac.cn, hsli@ee.cuhk.edu.hk

Abstract

Weakly supervised temporal action localization aims to localize temporal boundaries of actions and simultaneously identify their categories with only video-level category labels. Many existing methods seek to generate pseudo labels for bridging the discrepancy between classification and localization, but usually only make use of limited contextual information for pseudo label generation. To alleviate this problem, we propose a representative snippet summarization and propagation framework. Our method seeks to mine the representative snippets in each video for propagating information between video snippets to generate better pseudo labels. For each video, its own representative snippets and the representative snippets from a memory bank are propagated to update the input features in an intra- and inter-video manner. The pseudo labels are generated from the temporal class activation maps of the updated features to rectify the predictions of the main branch. Our method obtains superior performance in comparison to the existing methods on two benchmarks, THUMOS14 and ActivityNet1.3, achieving gains as high as 1.2% in terms of average mAP on THUMOS14. Our code is available at https://github.com/LeonHLJ/RSKP.

1. Introduction

Temporal action localization in videos has a wide range of applications in different scenarios. This task aims to localize action instances in untrimmed videos along the temporal dimension. Most existing methods [4, 18, 19, 46, 54] are trained in a fully supervised manner where both video-level labels and frame-wise annotations are provided. In contrast to these strong supervision based methods, weakly-supervised temporal action localization method attempt to localize action instances in videos, leveraging only video-level supervisions. This setting enables the method to bypass the manual annotations of temporal boundaries, which are laborious, expensive and prone to large variations [39].

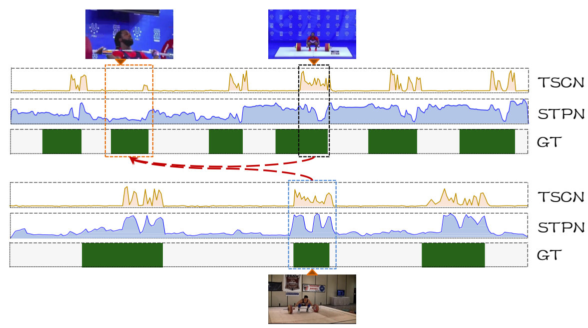

Figure 1. Two examples of the action of "*Clean and Jerk*". The barcode is the ground-truth (GT). The line charts are snippet-wise classification scores. We show the detection results of two methods, TSCN [50] and STPN [33]. The instance in the orange box is hard for both the two methods, because most of the snippets show only a part of the athlete. In contrast, the instances (black box and green box) with frontal view and whole body is much easier to be detected. A direct way to address this issue would be to propagate (red lines) knowledge of representative snippets to other snippets.

Due to the absence of fine-grained annotations, existing works mainly embrace a localization-by-classification pipeline [42, 53], where a classifier is trained with video-level annotations of action categories [31] and is used to obtain a sequence of class logits or predictions, *i.e.*, temporal class activation maps (TCAMs). Usually, detection results are obtained from TCAMs via a post-processing step [33,42] (*e.g.*, thresholding) or a localization branch [23,41]. Therefore, the quality of TCAMs determines the upper bound of the model. However, there is generally be a discrepancy between classification and localization [20,22,24]. Since each video generally contains multiple snippets[1], with only video-level annotations, the model would easily focus on the contextual background or discriminative snippets that contribute most to video-level classification, which hin-

*Corresponding author.

[1]In this paper, we view snippets as the smallest granularity since the high-level features of consecutive frames vary smoothly over time [11,45].

ders the generation of high quality TCAMs.

To address this issue, pseudo label-based methods [27, 35, 49, 50] were proposed to generate snippet-wise pseudo labels for bridging the gap between classification and localization. However, existing methods only leverage limited information, *i.e*., the information within each snippet, to generate pseudo labels, which is insufficient to generate high quality pseudo labels. In Figure 1, we show the detection results of two methods. The first method TSCN [50] is a pseudo label-based method, while the second method STPN [33] is a simple baseline model without using pseudo labels. As we can see, even if TSCN achieves much gain over STPN, neither of the two methods successfully detects the difficult action instance in the orange box, which only shows partial body of the athlete. Obviously, the pseudo labels generated from the inaccurate TCAMs are also inaccurate. In contrast, for the easy instance, *e.g*., the one in the blue box, both of the two methods accurately detect it.

The above observation motivate us to introduce contextual information for pseudo label generation. Specifically, we propose to propagate the knowledge of those representative snippets (*e.g*., the black and blue boxes in Figure 1) in an intra- and inter-video manner to facilitate the pseudo label generation, especially for those difficult snippets (*e.g*., the orange box in Figure 1). To achieve this goal, a major issue is how to determine the representative snippets and how to propagate their useful knowledge to other snippets. Besides, it should be effective to summarize and propagate snippet-level knowledge across videos, so as to take advantage of the large variation of videos in a large scale dataset.

We present a representative snippet knowledge propagation framework. To facilitate knowledge propagation, we propose to mine the representative snippets, which can mitigate the influence of outlier snippets to serve as a bridge to propagate knowledge between snippets. Specifically, we utilize the expectation-maximization (EM) attention [17] to handle the variations caused by different camera views [51], sub-action differences [8, 20], confusing background context [22, 24] and to capture the important semantic of each video, which are severed as the representative snippets in our method. After that, we employ a memory bank to store representative snippets of high confidences for each class. This design enables our method to leverage representative snippets in an inter-video fashion, and also avoids much GPU memory cost during training. Furthermore, to propagate the knowledge of representative snippets, we propose a bipartite random walk (BiRW) module, which integrates the random walk operation to update the features of the input video with intra- and inter-video representative snippets. The TCAMs of the updated features serve as online refined pseudo labels to rectify the predictions of the main branch.

The contribution of this paper is three-fold. (a) We propose a novel representative snippet knowledge propagation framework for weakly supervised temporal action localization, which generates better pseudo labels via representative snippet knowledge propagation to effectively alleviates the discrepancy between classification and detection. (b) The proposed framework can be applied to most existing methods to consistently improve their localization performance. (c) Compared with state-of-the-art methods, the proposed framework yields improvements of **1.2%** and **0.6%** average mAP on THUMOS14 and ActivityNet1.3.

2. Related Work

As stated in [20, 30, 31], there is generally a discrepancy between classification and localization in this task. Recently, many efforts have been made to solve this issue. We divide these methods into four categories.

The first category is the metric learning-based methods. For example, W-TALC [36], 3C-Net [32], RPN [7] and A2CL-PT [29] employed the center loss [44], clustering loss [48], triplet loss [40], *etc*. to learn intra-class compact features. However, these methods starts from the perspective of classification, that is, they learn video-level foreground features [7, 36] or class-wise features [29, 32] to enforce intra-class compactness. Whereas, these video-level features are aggregated from those discriminative snippets, they can hardly influence those less-discriminative features in the snippet-level. In contrast, our method is from the perspective of detection, which generates better snippet-wise pseudo labels by propagating the knowledge of representative snippets, so as to obtain better detection results.

The second category is the erasing-based methods, whose representative methods are Step-by-Step Erasion [55] and A2CL-PT [29]. These methods is based on the adversarial complementary learning [52], which first finds the discriminative regions and then tries to weight more less-discriminative regions from the remaining regions. However, it is difficult to set a proper number of steps for different categories with different complexities.

The methods of the third category are built on a multi-branch [20] or multi-attention architecture [9, 10, 22, 24]. These methods adopted a similar idea to the second category's methods, except for the parallel processing. To avoid trivial solutions, these methods require additional regularization terms to make branches or attention scores different or complementary. Likewise, it is difficult to define a proper number of branches or attentions for all action categories. Our method is also a multi-branch architecture. However, the additional branch generates online pseudo labels for the main branch. Therefore, instead of forcing branches to be different, we force them to be similar.

The fourth category is the pseudo label-based methods. RefineLoc [35] is the first method that generates snippet-level hard pseudo labels for WTAL. However, it simply expanded previous detection results to obtain pseudo labels, which may result in over-complete proposals. EM-MIL [27] put the pseudo-label generation into an expectation-maximization framework. TSCN [50] proposed to generate

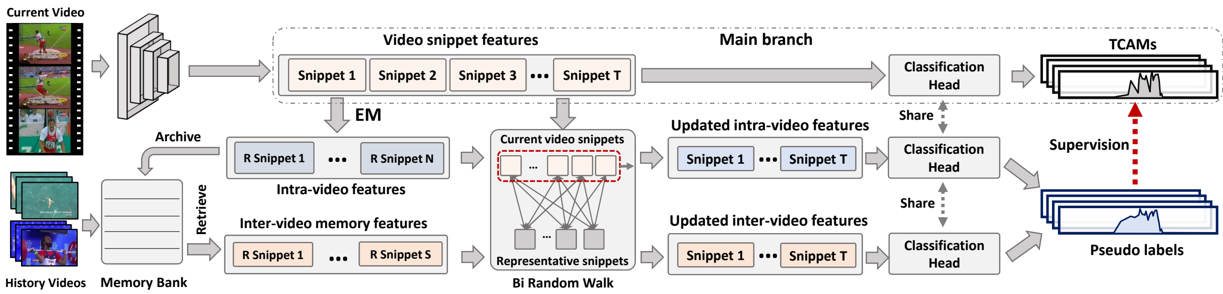

Figure 2. The overview of our method. We first extract snippet-wise features by using a fixed-weighted backbone network appended with a small learnable network. We utilize the expectation-maximization (EM) Attention [17] to learn a Gaussian mixture model (GMM) for each video, whose mean vectors are treated as the representative snippets. We use a memory bank to store representative snippets with high prediction scores for each class. To propagate representative snippets, we propose a bipartite random walk (BiRW) module to gradually update the original features. Given the original features and the updated features, we feed them into three parallel classification heads with sharing parameters. The output TCAMs of the two branches of the updated features are first fused and then serve as the online refined pseudo labels to rectify the predictions of the main branch.

snippet-level pseudo labels from late fusion attention sequence. UGCT [49] presented an uncertainty guided collaborative training strategy to generate pseudo labels with modality collaborative learning and uncertainty estimation. Despite the success of these methods, they leverage little contextual information for pseudo label generation.

3. Proposed Method

In this section, we elaborate on the proposed method. The illustration of the overall method is shown in Figure 2.

Problem definition. Let $V = \{v_t\}_{t=1}^{L}$ be a video of temporal length L. Each video is divided into a series of non-overlapping snippets. Assume that we have a set of N training videos $\{V_i\}_{i=1}^{N}$ being annotated with their action categories $\{\boldsymbol{y}_i\}_{i=1}^{N}$, where $\boldsymbol{y}_i$ is a binary vector indicating the presence/absence of each of k action. During inference, for a video, we predict a set of action instances $\{(c, q, t_s, t_e)\}$, where c denotes the predicted action class, q is the confidence score, t_s and t_e represent the start time and end time.

Overview. We propose a representative snippet knowledge summarization and propagation framework for generating better snippet-level pseudo labels to enhance the final localization performance. Pseudo labels have an important role for bridging the gap between classification and detection. Nevertheless, existing methods merely leverage contextual information for pseudo label generation, leading to inaccurate pseudo labels and compromising performance. Our key idea is to generate pseudo labels by propagating the knowledge of representative snippets, which act as a bridge between discriminative snippets and less-discriminative snippets, thereby indirectly propagate information between all snippets and producing accurate pseudo labels to improve the model's performance. In our method, we first extract video features by a feature extraction module. After that, we summarize the representative snippets from the extracted video features. The representative snippets with high confidences are maintained in a memory bank. For each video, we leverage both the intra- and inter-video representative snippets that are retrieved from the memory bank. A bipartite random walk module is introduced to update the video features with the two kinds of representative snippets. Given the video features and the updated video features, we feed them into three parallel classification heads with sharing parameters. The output TCAMs of the two branches corresponding to the updated features are first fused and then serve as the online refined pseudo labels to rectify the predictions of the main branch.

Feature extraction. Given a video, we first divide it into a series of non-overlapping snippets. Following [33,36], we utilize a fixed-weight backbone network, Inflated 3D (I3D) [3] pre-trained on the Kinetics-400 dataset [13], to encode appearance (RGB) and motion (optical flow) information into a $d = 2048$ dimensional feature for each snippet. The I3D features are encoded into latent embeddings $\mathbf{F} \in \mathbb{R}^{l \times d}$ with a convolutional layer, where l is the number of video snippets of a video. We take $\mathbf{F}$ as the input of our model.

Classification head. The classification head is used to generate TCAMs, it can be any existing WSTAL methods. To generate high-quality TCAMs and improve the lower bound of our method, we use the recent FAC-Net [9] as the classification head for its simple pipeline and promising performance. Note that, there are some modifications to FAC-Net in our method. First, we discard the class-wise foreground classification head in our method, since the commonly used class-agnostic attention head and multiple instance learning head already enable our method to achieve a high baseline performance. Second, we use the sigmoid rather than the softmax function to obtain normalized foreground scores. This setting enables our method to use the attention normalization term [50] to obtain highly confident representative snippets. Third, we do not use the hybrid attention strategy that is designed to alleviate the discrepancy between classification and detection.

In the following sections, we first investigate how to obtain summarizations of the representative snippets and to propagate their information to all other snippets.

3.1. Representative Snippet Summarization

A naïve way to obtain representative snippets would be to select the snippets with high prediction scores, *i.e.*, discriminative snippets. However, as shown in Figure 3, even after large-scale pre-training, there are generally low similarities between discriminative snippets and other snippets of the same category. Intuitively, representative snippets should be able to describe most of the snippets of the same class, so as to act as a bridge to associate the snippets of the same class for knowledge propagation. It is therefore ineffective to directly propagate the information of the discriminative snippets to other snippets. Therefore, we propose to summarize the representations of video snippets to obtain representative snippets of each video. In Figure 3, we can see that using cluster centers via clustering video snippet features (*e.g.*, k-means, spectral clustering and agglomerative clustering) as the representative snippets achieves much better performance on building stronger relations with other snippets. According to our experiments, using the clustering methods to summarize representative snippets is important to achieve high detection performance.

In this work, we employ the expectation-maximization (EM) attention [17] to generate the representative snippets of each video. EM attention uses a special EM algorithm based on Gaussian mixture model (GMM) [37]. Specifically, a separated GMM is adopted to capture the feature statistics of each video and models the distribution of $\mathbf{f}_i \in \mathbb{R}^d$ (the i-th snippet feature of $\mathbf{F} \in \mathbb{R}^{l\times d}$) as a linear composition of Gaussians as follow,

$$p\left(\mathbf{f}_i\right) = \sum\nolimits_{k=1}^{n} z_{ik}\mathcal{N}\left(\mathbf{f}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right), \tag{1}$$

where n is the number of Gaussians, $\boldsymbol{\mu}_k \in \mathbb{R}^d$, $\boldsymbol{\Sigma}_k \in \mathbb{R}^{d\times d}$ and z_{ik} denote the mean, covariance and weight for the k-th Gaussian. Following [17], we replace the covariance with identity matrix $\mathbf{I}$ and leave out it in the following equations.

As shown in Figure 4 (top), the EM attention starts from the randomly initialized means $\boldsymbol{\mu}^{(0)} \in \mathbb{R}^{n\times d}$. At the t-th iteration, it first performs the E step to calculate the new weights $\mathbf{Z}^{(t)} \in \mathbb{R}^{t\times n}$ of Gaussians as

$$\mathbf{Z}^{(t)} = \mathrm{softmax}\left(\lambda\,\mathrm{Norm}_2(\mathbf{F})\,\mathrm{Norm}_2(\boldsymbol{\mu}^{(t-1)})^\top\right), \tag{2}$$

where λ denotes a hyper-parameter to control the smoothness of the distribution. The $\mathrm{Norm}_2(\mathbf{F})$ denotes the l_2-norm along each row of $\mathbf{F}$. The softmax operation is performed along each row of $\mathbf{Z}$. Therefore, $z_{ik}^{(t)}$ denotes the probability that the snippet feature $\mathbf{f}_i$ is generated by the k-th Gaussian. After the E step, the M step turns to update the means $\boldsymbol{\mu}$ as

$$\boldsymbol{\mu}^{(t)} = \mathrm{Norm}_1(\mathbf{Z}^{(t)})^\top\mathbf{F}, \tag{3}$$

where $\mathrm{Norm}_1(\mathbf{Z}^{(t)})$ denotes the column-wise l_1 normalization for $\mathbf{Z}^{(t)}$. We can see that Equation (3) updates the means using the weighted summation of the features $\mathbf{F}$. The i-th row and the k-th column of $\mathrm{Norm}_1(\mathbf{Z}^{(t)})$ represents the membership values of the feature $\mathbf{f}_i$ to the k-th Gaussian. Besides, the normalization also ensures the updated $\boldsymbol{\mu}$ lies in the same embedding space as $\mathbf{F}$. Therefore, alternately executing Equations (2) and (3) captures global contexts of a video in a non-local [43] but more efficient way, owing to the much smaller size of the means $\boldsymbol{\mu}^{(t)} \in \mathbb{R}^{n\times d}$ compared with the video features $\mathbf{F} \in \mathbb{R}^{l\times d}$ ($l \gg n$).

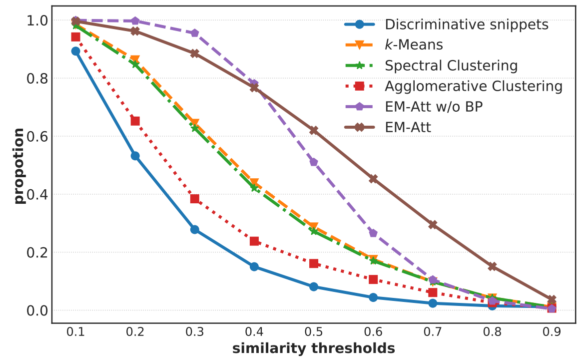

Figure 3. We assign each representative snippet with the label of its nearest snippet and then select the snippets of the same category according to the ground truth. We plot the proportion of the number of snippets, whose similarities to the representative snippet are higher than a certain threshold (0.1:0.9), to the total number of snippets of the same category. The results of EM-Att is calculated upon our full model. Other results are based on the features of a well learned baseline model, which takes the video features $\mathbf{F}$ as input and contains one classification head.

We integrate two EM iterations in our network to obtain the promising representative snippets (*i.e.*, $\boldsymbol{\mu}^{(2)}$). As a result, unlike [17] that uses the moving averaging to update the initialized means $\boldsymbol{\mu}^{(0)}$ to prevent gradient vanishing or explosion caused by too many iterations, we update the initialized means by standard back propagation. Besides, it is interesting that when we initialize $\boldsymbol{\mu}^{(0)}$ with a (semi) orthogonal matrix [38], even if we fix the initialized means (*i.e.*, EM-Att w/o BP in Figure 3), the obtained representative snippets are more representative than the cluster centers of other clustering methods. In our experiments, EM attention without back propagation achieves much better performance than other clustering methods. When we update the initial means $\boldsymbol{\mu}^{(0)}$ by standard back propagation (*i.e.*, EM-Att in Figure 3), they can capture the feature distribution of the dataset and achieves the optimal performance.

To make the equations clear and avoid confusion, we denote the online calculated representative snippets as $\boldsymbol{\mu}^a$.

3.2. Representative Snippet Memory Bank

After obtaining the representative snippets of each video, we use a memory bank to store representative snippets of all videos with high confidences for each class. The insight is that different videos may contain action instances of the

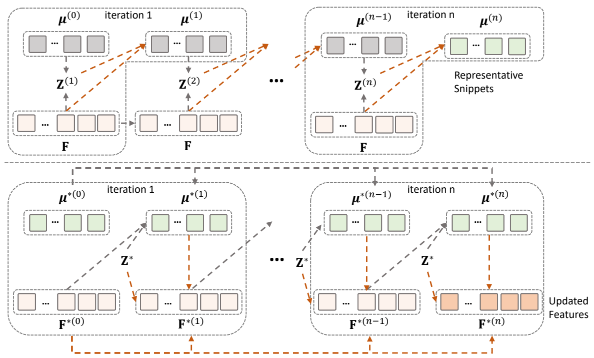

Figure 4. Illustration of the representative snippet summarization (top) and representative snippet propagation (bottom). The gray lines denote the E step, while the red lines denote the M step.

same class but with different appearances. Therefore, propagating representative snippets in an inter-video manner via the memory bank takes advantage of the large variation of many videos of each class to facilitate the network to recognize those difficult action instances.

Specifically, we maintain two memory tables for storing the features of representative snippets and their scores, respectively. We denote the memory table of the representative snippets as $\mathbf{M} \in \mathbb{R}^{c \times s \times d}$, where c is the number of classes and s denotes the number of memory slots (representative snippets) for each class. Given the representative snippets of a video, we utilize the action classifier in the classification head to obtain their class predictions. We then compare their prediction scores of the ground truth class with the representative snippets in the memory table $\mathbf{M}$, the ones with higher prediction scores are archived into the memory table $\mathbf{M}$. Simultaneously, the corresponding scores in the score memory table are also updated. In summary, we only keep the representative snippets of the top-s scores in the memory table $\mathbf{M}$.

Consequently, for each video, we have the online representative snippets $\boldsymbol{\mu}^a$ learned from the current video, and the offline representative snippets $\boldsymbol{\mu}^e$ retrieved from the memory table $\mathbf{M}$ corresponding to the ground truth classes.

3.3. Representative Snippet Propagation

Given the online and offline representative snippets $\boldsymbol{\mu}^a$ and $\boldsymbol{\mu}^e$, a challenge is how to propagate the representative snippets to snippet features $\mathbf{F}$ of the current video. Intuitively, a direct way is to use the affinity $\mathbf{Z}^*$ ($*$ is a or e corresponding to $\boldsymbol{\mu}^a$ or $\boldsymbol{\mu}^e$) calculated by Equation (2) to conduct a random walk operation [1] as $\mathbf{Z}^*\boldsymbol{\mu}^*$. In practice, we would like the updated features do not deviate too far from the video features $\mathbf{F}$. Therefore, the propagation process can be formulated as $\mathbf{F}^* = w \cdot \mathbf{Z}^*\boldsymbol{\mu}^* + (1-w) \cdot \mathbf{F}$, where w denotes a parameter $[0, 1]$ that controls the trade-off between feature propagation and the original features. However, even if the representative snippets have high similarities with most of the snippets of the same class, it is impractical to fully propagate the knowledge of the representative snippets via a single pass of propagation. We find that, the representative snippets $\boldsymbol{\mu}^*$ and the video features $\mathbf{F}$ actually make up a complete bipartite graph, whose affinities are depicted by $\mathbf{Z}^*$. Therefore, we propose a bipartite random walk (BiRW) module to enable multiple passes of propagation to fully fuse representative snippets' knowledge into the video snippet features.

Specifically, there are multiple iterations in the BiRW. At the t-th iteration, the propagation process is formulated as

$$\boldsymbol{\mu}^{*(t)} = w \cdot \mathrm{Norm}_1(\mathbf{Z}^*)^\top \mathbf{F}^{*(t-1)} + (1-w) \cdot \boldsymbol{\mu}^{*(0)}, \quad (4)$$

$$\mathbf{F}^{*(t)} = w \cdot \mathbf{Z}^*\boldsymbol{\mu}^{*(t)} + (1-w) \cdot \mathbf{F}^{*(0)}, \quad (5)$$

where $\mathbf{F}^{*(0)}$ and $\boldsymbol{\mu}^{*(0)}$ are the video snippet features $\mathbf{F}$ and the representative snippets $\boldsymbol{\mu}^a$ or $\boldsymbol{\mu}^e$, respectively. As shown in Figure 4 (bottom), Equations (4) and (5) can be also viewed as a EM process, which fixes the affinity $\mathbf{Z}^*$ to alternately update $\mathbf{F}$ and $\boldsymbol{\mu}^*$. The representative snippets therefore not only are used for propagating representative knowledge (Equation (5)) but also serve as a bridge to propagate the knowledge between the features of $\mathbf{F}$ (Equation (4)). Due to their representativeness, they can better propagate information between features of the same class. This process can be conducted for multiple times to fully fuse representative snippets' knowledge. To avoid gradient vanishing or explosion caused by the unrolled computational graph, we use an approximate inference formulation as (the detail is available in the supplementary materials)

$$\mathbf{F}^* = (1-w)(\mathbf{I} - w^2\mathbf{Z}^*\,\mathrm{Norm}_1(\mathbf{Z}^*)^\top)^{-1}(w\mathbf{Z}^*\boldsymbol{\mu}^* + \mathbf{F}). \quad (6)$$

We can also follow Equation (6) to obtain further refined representative snippets to store in the memory bank. Nevertheless, we find that this way achieves comparable performance with using the original representative snippets. The reason may be that $\boldsymbol{\mu}^a$ has been stable enough after several EM iterations. Besides, there is another feasible way to update the video features $\mathbf{F}$ by fixing the representative snippets $\boldsymbol{\mu}^*$ and alternately updating $\mathbf{F}$ and $\mathbf{Z}^*$. Nevertheless, its performance is comparable with Equations (4) and (5) but inferior to Equation (6). Note that, rather than concatenating $\boldsymbol{\mu}^a$ and $\boldsymbol{\mu}^e$ to propagate representative snippets, we use Equation (6) to propagate the knowledge of $\boldsymbol{\mu}^a$ and $\boldsymbol{\mu}^e$, respectively. This design is to prevent $\boldsymbol{\mu}^a$, which is extracted from the same video of $\mathbf{F}$, being dominated in the propagation. Consequently, after representative snippet propagation, we obtain the updated intra-video features $\mathbf{F}^a$ and the updated inter-video features $\mathbf{F}^e$, respectively.

3.4. Training Objectives

Given the original video snippet features $\mathbf{F}$ and the updated features $\mathbf{F}^a$, $\mathbf{F}^e$, we feed them into three parallel classification heads with sharing parameters to output their TCAMs $\mathbf{T}$, $\mathbf{T}^a$ and $\mathbf{T}^e$, respectively. The TCAMs $\mathbf{T}^a$ and

$\mathbf{T}^e$ are weightedly summed to obtain the TCAMs $\mathbf{T}^f$ that contains both intra- and inter-video representative snippet knowledge. We take $\mathbf{T}^f$ as the online pseudo labels for supervising the TCAMs $\mathbf{T}$ as

$$\mathcal{L}_{kd} = -\frac{1}{l}\sum\nolimits_{i=1}^{l} \mathbf{T}^f(i)\log(\mathbf{T}(i)), \tag{7}$$

where l is the number of snippets. The total loss is the summation of the loss $\mathcal{L}_{kd}$, the video classification loss $\mathcal{L}_{cls}$ of the three classification heads and the attention normalization loss $\mathcal{L}_{att}$ [50] that is only applied to the main branch.

$$\mathcal{L} = \mathcal{L}_{cls} + \alpha\mathcal{L}_{kd} + \beta\mathcal{L}_{att}, \tag{8}$$

where α and β are balancing hyper-parameters.

4. Experiments

4.1. Datasets

THUMOS14. [12] We use the subset from THUMOS14 that offers frame-wise annotations for 20 classes. We train the model on 200 untrimmed videos in its validation set and evaluate it on 212 untrimmed videos from the test set.

ActivityNet1.3. [2] This dataset covers 200 daily activities and provides 10,024 videos for training, 4,926 for validation and 5,044 for testing. We use the training set to train our model and the validation set to evaluate our model.

4.2. Implementation Details

Training details. Our method is trained with a mini-batch size of 10 and 16 with Adam [14] optimizer for THUMOS14 and ActivityNet1.3, respectively. The hyper-parameters w, λ, α and β are 0.5, 5.0, 1.0 and 0.1, respectively. Since the representative snippets are not well learned at the beginning, we only utilize the intra-video representative snippets in the first 100 epochs, and then add the memory bank into the training process. The training procedure stops at 200 epochs with the learning rate 5×10^{-5}.

Testing details. We take the whole sequence of a video as input for testing. When localizing action instances, the class activation sequence is upsampled to the original frame rate. We utilize the main branch and the branch of intra-video representative snippets for testing. Their video prediction scores and TCAMs are fused by weighted sum. During detection, We first reject the category whose class probability is lower than 0.1. Following [15], we use a set of thresholds to obtain the predicted action instances.

4.3. Comparison with State-of-the-art Methods

We compare our method with state-of-the-art weakly supervised methods and several fully supervised methods. The results are shown in Tables 1 and 2. On the THUMOS14 dataset, our method evidently outperforms the previous weak-supervised approaches in terms of almost all metrics. On the important criterion: average mAP (0.1:0.5), we surpass the state-of-the-art method [6] by 1.2%. We also notice that our method achieves better results at IoU 0.1 to 0.5 compared with some recent fully-supervised methods. Even if some methods (*i.e.*, Weak † in Table 2) utilize additional weak supervisions, such as action frequency, our method still outperforms these methods, indicating the effectiveness of our method. On the larger ActivityNet1.3, our method still outperforms all existing weakly supervised methods by 0.6% on average mAP.

Table 1. Results on ActivityNet1.3 validation set. AVG indicates the average mAP at IoU thresholds 0.5:0.05:0.95.

Method	mAP @ IoU			
	0.5	0.75	0.95	AVG
TAL-Net [4]	38.2	18.3	1.3	20.2
BSN [19]	46.5	30.0	8.0	30.0
GTAN [25]	52.6	34.1	8.9	34.3
BaS-Net (I3D) [15]	34.5	22.5	4.9	22.2
A2CL-PT (I3D) [29]	36.8	22.0	5.2	22.5
ACM-BANet (I3D) [30]	37.6	**24.7**	**6.5**	24.4
TSCN (I3D) [50]	35.3	21.4	5.3	21.7
WUM (I3D) [16]	37.0	23.9	5.7	23.7
TS-PCA (I3D) [21]	37.4	23.5	5.9	23.7
UGCT (I3D) [49]	39.1	22.4	5.8	23.8
AUMN (I3D) [26]	38.3	23.5	5.2	23.5
FAC-Net (I3D) [9]	37.6	24.2	6.0	24.0
Ours (I3D)	**40.6**	24.6	5.9	**25.0**

4.4. Ablation Study

We conduct a series of ablation studies on THUMOS14. Unless explicitly stated, we do not utilize the memory bank.

Representative snippet summarization. To validate the effectiveness of the representative snippets, we need to consider: (1) Is generating the representative snippets necessary? (2) If yes, what is an effective way to generate the representative snippets? To answer the two questions, we conduct several experiments in Tables 3 and 4.

We first define a baseline model, which is the same as our main branch that comprises one classification head and also uses the attention normalization loss. As we can see, when we introduce the representative snippets and add the second branch that takes the refined features F^a as input, even if without the supervision between the two branches, the performance is significantly improved over the baseline model. This phenomenon demonstrates the importance of the representative snippets, which makes the model to focus on the representative snippets and thus improves the recognition ability for the whole action class. When we further utilize the TCAMs of the updated features as the online pseudo labels, our method achieves absolute gains of 4.1% and 7.4% in terms of average mAP, over the method with only representative snippets and the baseline model. To further validate the necessity of the representative snippets, we replace representative snippets with the original video features $\mathbf{F}$,

Table 2. Comparisons of detection performance on THUMOS14. UNT and I3D represent UntrimmedNet features and I3D features, respectively. † means that the method utilizes additional weak supervisions, *e.g.*, action frequency.

Supervision	Method	Feature	mAP @ IoU (%)							AVG	AVG	AVG
			0.1	0.2	0.3	0.4	0.5	0.6	0.7	(0.1:0.5)	(0.3:0.7)	(0.1:0.7)
Full	SSN [54] (ICCV'17)	-	60.3	56.2	50.6	40.8	29.1	-	-	49.6	-	-
	BSN [19] (ECCV'18)	-	-	-	53.5	45.0	36.9	28.4	20.0	-	36.8	-
	GTAN [25] (CVPR'19)	-	69.1	63.7	57.8	47.2	38.8	-	-	55.3	-	-
Weak †	STAR [47] (AAAI'19)	I3D	68.8	60.0	48.7	34.7	23.0	-	-	47.0	-	-
	3C-Net [32] (ICCV'19)	I3D	59.1	53.5	44.2	34.1	26.6	-	8.1	43.5	-	-
Weak	CleanNet [23] (ICCV'19)	UNT	-	-	37.0	30.9	23.9	13.9	7.1	-	22.6	-
	RPN [7] (AAAI'20)	I3D	62.3	57.0	48.2	37.2	27.9	16.7	8.1	46.5	27.6	36.8
	TSCN [50] (ECCV'20)	I3D	63.4	57.6	47.8	37.7	28.7	19.4	10.2	47.0	28.8	37.8
	EM-MIL [27] (ECCV'20)	I3D	59.1	52.7	45.5	36.8	30.5	22.7	**16.4**	45.0	30.4	37.7
	A2CL-PT [29] (ECCV'20)	I3D	61.2	56.1	48.1	39.0	30.1	19.2	10.6	46.9	29.4	37.8
	HAM-Net [10] (AAAI'21)	I3D	65.4	59.0	50.3	41.1	31.0	20.7	11.1	49.4	30.8	39.8
	WUM [16] (AAAI'21)	I3D	67.5	61.2	52.3	43.4	33.7	22.9	12.1	51.6	32.9	41.9
	AUMN [26] (CVPR'21)	I3D	66.2	61.9	54.9	44.4	33.3	20.5	9.0	52.1	32.4	41.5
	CoLA [51] (CVPR'21)	I3D	66.2	59.5	51.5	41.9	32.2	22.0	13.1	50.3	32.1	40.9
	TS-PCA [21] (CVPR'21)	I3D	67.6	61.1	53.4	43.4	34.3	24.7	13.7	52.0	33.9	42.6
	UGCT [49] (CVPR'21)	I3D	69.2	62.9	55.5	46.5	35.9	23.8	11.4	54.0	34.6	43.6
	ASL [28] (CVPR'21)	I3D	67.0	-	51.8	-	31.1	-	11.4	-	-	-
	CO_2-Net [6] (MM'21)	I3D	70.1	63.6	54.5	45.7	**38.3**	**26.4**	13.4	54.4	35.6	44.6
	D2-Net [31] (ICCV'21)	I3D	65.7	60.2	52.3	43.4	36.0	-	-	51.5	-	-
	FAC-Net [9] (ICCV'21)	I3D	67.6	62.1	52.6	44.3	33.4	22.5	12.7	52.0	33.1	42.2
	Ours	I3D	**71.3**	**65.3**	**55.8**	**47.5**	38.2	25.4	12.5	**55.6**	**35.9**	**45.1**

Table 3. Evaluation of the necessity and effectiveness of the representative snippet generation. *Baseline* comprises one classification head and takes the video snippet features $\mathbf{F}$ as input.

Method	mAP @ IoU			
	0.3	0.5	0.7	AVG
Baseline	45.2	29.9	10.2	36.8
+ representative snippets	48.7	33.1	11.5	40.1
+ pseudo label supervision	54.5	37.3	12.5	44.2
+ memory bank	55.8	38.2	12.5	45.1
replace $\boldsymbol{\mu}^a$ with $\mathbf{F}$	52.4	35.5	11.7	42.4

Table 4. Evaluation of different approaches of generating representative snippets. EM-Att denotes the EM attention.

Method	mAP @ IoU			
	0.3	0.5	0.7	AVG
Discriminative snippets	45.0	25.0	6.8	34.8
k-Means	51.7	34.1	11.7	41.5
Agglomerative Clustering	50.0	32.3	11.2	40.1
Spectral Clustering	50.5	31.6	11.1	40.4
EM-Att w/o BP	53.1	35.2	12.1	42.8
EM-Att	54.5	37.3	12.5	44.2

and thus the BiRW module becomes a vanilla random walk module. It is noteworthy that the *replace* $\boldsymbol{\mu}^a$ *with* $\mathbf{F}$ in Table 3 also obtains significant gain over the baseline model, but 1.8% lower on average mAP than our method using the representative snippets under the same setting. Besides, our method contains both the intra- and inter-video representative snippets achieves the best performance of 45.1% on average mAP, indicating the importance of propagating the representative snippets across videos.

To answer the second questions, we also explore several strategies for representative snippet generation. As discussed earlier, we first verify the way of selecting discriminative snippets. It is unsurprised that this way deteriorates the performance of the baseline model, because it makes the model only focus on those discriminative snippets and thus weakens the detection ability. In contrast, some traditional clustering methods, such as k-means, agglomerative clustering and spectral clustering achieve evident gains over the baseline model. We also evaluate the DBSCAN [5]. However, as training progresses, it is hard to update proper hyper-parameters for DBSCAN, leading to very low performance. It is noteworthy that the *EM-Att w/o BP* achieves a high performance of 42.8% in terms of average mAP, which is much better than other clustering methods. It may attribute to its orthogonal initialization, which enables the learned representative snippets to concentrate on different video parts. Besides, the performance is further improved by 1.4% when we allow back propagation. More experimental results are provided in the supplementary materials.

Table 5. Evaluation of the representative snippet propagation.

Method	mAP @ IoU			
	0.3	0.5	0.7	AVG
Random walk $\mathbf{Z}^a\boldsymbol{\mu}^a$	45.0	25.0	6.8	34.8
Equation (6)	54.5	37.3	12.5	44.2
Propagate scores	49.5	28.8	8.2	38.5
Propagate features w/o $\mathcal{L}_{cls}$	48.2	28.4	8.3	37.7

Representative snippet propagation. Several comparative experiments are conducted to evaluate the effectiveness of representative snippet propagation, with results posted in Table 5. We first evaluate some variants of the representative snippet propagation. We notice that the random walk

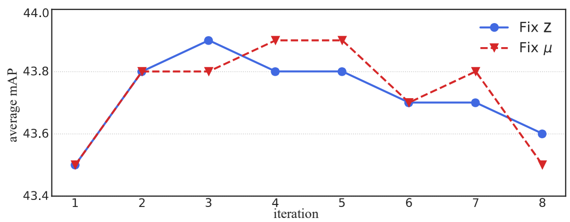

Figure 5. The detection results of two implementations of Equations (4) and (5) with different iteration numbers.

Table 6. The detection results of applying our method to existing methods. *Embedding* means we add a learnable network after the backbone network to learn the video features. The results of the original methods are reproduced.

Method	mAP @ IoU			
	0.3	0.5	0.7	AVG
STPN [33] + embedding	38.4	19.1	4.7	28.9
STPN + embedding + Ours	46.5	21.8	5.8	33.7 $_{\uparrow 4.8}$
BM [34]	45.2	26.2	8.7	35.3
BM + Ours	47.4	29.3	9.2	37.7 $_{\uparrow 2.4}$
WUM [16]	51.0	32.8	10.9	40.4
WUM + Ours	53.5	34.0	12.7	42.4 $_{\uparrow 2.0}$
FAC-Net [9]	53.2	34.4	13.7	42.9
FAC-Net + Ours	55.8	38.4	13.3	45.2 $_{\uparrow 2.3}$

$\mathbf{Z}^a\boldsymbol{\mu}^a$ deteriorates the performance. In contrast, as shown in Figure 5, even if we weightedly sum $\mathbf{Z}^a\boldsymbol{\mu}^a$ with the original features (*i.e.*, iteration 1), the performance would be significantly improved, indicating that $\mathbf{Z}^a\boldsymbol{\mu}^a$ deviates too much from the original video features $\mathbf{F}$. Moreover, the performance improves as the number of iterations increases. However, when the number of iterations is too large, the performance slightly drops, which might be caused by gradient vanishing or explosion even if there are residual connections (weighted summation with the video features). Despite their promising performance, our proposed approximation inference obtain the optimal result.

We further validate two more variants of propagation. The first variant, *propagate scores*, follows a knowledge ensembling way to propagate prediction scores of the representative snippets rather than their features. Therefore, only the main branch has the video-level classification losses. The second variant, *propagate features w/o* $\mathcal{L}_{cls}$, also propagates the features of representative snippets but removes the video-level classification losses. Both variants have gains over the baseline model but are much inferior to our final solution. Therefore, the two additional branches not only provide online pseudo labels, but also enforce the model to focus on the representative snippets, thus improving the model's ability to localize action instances.

Integrating our framework to existing methods. In Table 6, we replace our classification heads with some existing methods. In these experiments, we use the default settings of these methods to ensure fair comparisons. Besides, if there are additional losses in these methods, we only impose them on the main branch. As we can see, either for some early methods or state-of-the-art methods, our method can consistently improve their performances.

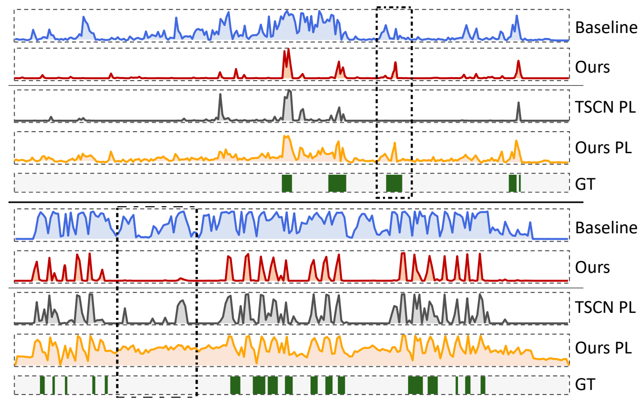

Figure 6. We show the qualitative results of two examples of *shotput* and *diving* on THUMOS14 [12]. PL denotes the pseudo label.

4.5. Qualitative results

We visualize some examples of detected action instances and generated pseudo labels in Figure 6. In the first example of *shotput*, compared with TSCN [50], our method generates more accurate pseudo labels to successfully detect the third action instance (black rectangle). In the second example of *diving*, our method leverages the knowledge of the representative snippets of background to effectively suppress the responses of the background.

5. Conclusion and Limitation

We propose a representative snippet summarization and propagation framework. Our method aims to generate better pseudo labels by propagating the knowledge of representative snippets. We summarize the representative snippets in each video and maintain a memory table to store representative snippets. For each video, the intra- and inter-video representative snippets are propagated to update the video features. The temporal class activation maps of the updated features serve as the online pseudo labels to rectify the predictions of the main branch. Our method achieves the state-of-the-art performance on two popular datasets and can consistently improve the performance of existing methods.

Due to the matrix inversion during propagation, the training time of one epoch in our method is almost twice as long as the baseline model. We will resort to some efficient matrix inversion strategies to solve this issue in the future.

Acknowledgement

This work is supported in part by Centre for Perceptual and Interactive Intelligence Limited, in part by the General Research Fund through the Research Grants Council of Hong Kong under Grants (Nos. 14204021, 14207319), in part by CUHK Strategic Fund.

References

[1] David Aldous and Jim Fill. Reversible markov chains and random walks on graphs, 2002. 5

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970. IEEE, 2015. 6

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308. IEEE, 2017. 3

[4] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, pages 1130–1139. IEEE, 2018. 1, 6

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996. 7

[6] Fa-Ting Hong, Jia-Chang Feng, Dan Xu, Ying Shan, and Wei-Shi Zheng. Cross-modal consensus network for weakly supervised temporal action localization. In *ACM MM*. ACM, 2021. 6, 7

[7] Linjiang Huang, Yan Huang, Wanli Ouyang, and Liang Wang. Relational prototypical network for weakly supervised temporal action localization. In *AAAI*, volume 34, pages 11053–11060, 2020. 2, 7

[8] Linjiang Huang, Yan Huang, Wanli Ouyang, and Liang Wang. Modeling sub-actions for weakly supervised temporal action localization. *IEEE TIP*, 30:5154–5167, 2021. 2

[9] Linjiang Huang, Liang Wang, and Hongsheng Li. Foreground-action consistency network for weakly supervised temporal action localization. In *ICCV*, pages 8002–8011. IEEE, 2021. 2, 3, 6, 7, 8

[10] Ashraful Islam, Chengjiang Long, and Richard Radke. A hybrid attention mechanism for weakly-supervised temporal action localization. In *AAAI*, volume 35, pages 1637–1645, 2021. 2, 7

[11] Dinesh Jayaraman and Kristen Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *CVPR*, pages 3852–3861. IEEE, 2016. 1

[12] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/`, 2014. 6, 8

[13] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 3

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[15] Pilhyeon Lee, Youngjung Uh, and Hyeran Byun. Background suppression network for weakly-supervised temporal action localization. In *AAAI*, volume 34, pages 11320–11327, 2020. 6

[16] Pilhyeon Lee, Jinglu Wang, Yan Lu, and Hyeran Byun. Weakly-supervised temporal action localization by uncertainty modeling. In *AAAI*, volume 35, pages 1854–1862, 2021. 6, 7, 8

[17] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *ICCV*, pages 9167–9176. IEEE, 2019. 2, 3, 4

[18] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *ICCV*, pages 3889–3898, 2019. 1

[19] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, pages 3–19. Springer, 2018. 1, 6, 7

[20] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *CVPR*, pages 1298–1307. IEEE, 2019. 1, 2

[21] Yuan Liu, Jingyuan Chen, Zhenfang Chen, Bing Deng, Jianqiang Huang, and Hanwang Zhang. The blessings of unlabeled background in untrimmed videos. In *CVPR*, pages 6176–6185. IEEE, 2021. 6, 7

[22] Ziyi Liu, Le Wang, Wei Tang, Junsong Yuan, Nanning Zheng, and Gang Hua. Weakly supervised temporal action localization through learning explicit subspaces for action and context. In *AAAI*, volume 35, pages 2242–2250, 2021. 1, 2

[23] Ziyi Liu, Le Wang, Qilin Zhang, Zhanning Gao, Zhenxing Niu, Nanning Zheng, and Gang Hua. Weakly supervised temporal action localization through contrast based evaluation networks. In *ICCV*, pages 3899–3908. IEEE, 2019. 1, 7

[24] Ziyi Liu, Le Wang, Qilin Zhang, Wei Tang, Junsong Yuan, Nanning Zheng, and Gang Hua. Acsnet: Action-context separation network for weakly supervised temporal action localization. In *AAAI*, volume 35, pages 2233–2241, 2021. 1, 2

[25] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *CVPR*, pages 344–353. IEEE, 2019. 6, 7

[26] Wang Luo, Tianzhu Zhang, Wenfei Yang, Jingen Liu, Tao Mei, Feng Wu, and Yongdong Zhang. Action unit memory network for weakly supervised temporal action localization. In *CVPR*, pages 9969–9979. IEEE, 2021. 6, 7

[27] Zhekun Luo, Devin Guillory, Baifeng Shi, Wei Ke, Fang Wan, Trevor Darrell, and Huijuan Xu. Weakly-supervised action localization with expectation-maximization multi-instance learning. In *ECCV*, pages 729–745. Springer, 2020. 2, 7

[28] Junwei Ma, Satya Krishna Gorti, Maksims Volkovs, and Guangwei Yu. Weakly supervised action selection learning in video. In *CVPR*, pages 7587–7596. IEEE, 2021. 7

[29] Kyle Min and Jason J Corso. Adversarial background-aware loss for weakly-supervised temporal activity localization. In *ECCV*, pages 283–299. Springer, 2020. 2, 6, 7

[30] Md Moniruzzaman, Zhaozheng Yin, Zhihai He, Ruwen Qin, and Ming C Leu. Action completeness modeling with background aware networks for weakly-supervised temporal ac-

tion localization. In *ACM MM*, pages 2166–2174. ACM, 2020. 2, 6

[31] Sanath Narayan, Hisham Cholakkal, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. D2-net: Weakly-supervised action localization via discriminative embeddings and denoised activations. In *ICCV*, pages 13608–13617. IEEE, 2021. 1, 2, 7

[32] Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao. 3c-net: Category count and center loss for weakly-supervised action localization. In *ICCV*, pages 8679–8687. IEEE, 2019. 2, 7

[33] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, pages 6752–6761. IEEE, 2018. 1, 2, 3, 8

[34] Phuc Xuan Nguyen, Deva Ramanan, and Charless C Fowlkes. Weakly-supervised action localization with background modeling. In *ICCV*, pages 5502–5511. IEEE, 2019. 8

[35] Alejandro Pardo, Humam Alwassel, Fabian Caba, Ali Thabet, and Bernard Ghanem. Refineloc: Iterative refinement for weakly-supervised action localization. In *Winter Conference on Applications of Computer Vision*, pages 3319–3328. IEEE, 2021. 2

[36] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *ECCV*, pages 563–579. Springer, 2018. 2, 3

[37] Sylvia Richardson and Peter J Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: series B (statistical methodology)*, 59(4):731–792, 1997. 4

[38] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013. 4

[39] Konrad Schindler and Luc Van Gool. Action snippets: How many frames does human action recognition require? In *CVPR*, pages 1–8. IEEE, 2008. 1

[40] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823. IEEE, 2015. 2

[41] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *ECCV*, pages 154–171. Springer, 2018. 1

[42] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, pages 4325–4334. IEEE, 2017. 1

[43] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803. IEEE, 2018. 4

[44] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, pages 499–515. Springer, 2016. 2

[45] Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002. 1

[46] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, pages 5783–5792. IEEE, 2017. 1

[47] Yunlu Xu, Chengwei Zhang, Zhanzhan Cheng, Jianwen Xie, Yi Niu, Shiliang Pu, and Fei Wu. Segregated temporal assembly recurrent networks for weakly supervised multiple action detection. In *AAAI*, volume 33, pages 9070–9078, 2019. 7

[48] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *CVPR*, pages 3474–3482. IEEE, 2018. 2

[49] Wenfei Yang, Tianzhu Zhang, Xiaoyuan Yu, Tian Qi, Yongdong Zhang, and Feng Wu. Uncertainty guided collaborative training for weakly supervised temporal action detection. In *CVPR*, pages 53–63. IEEE, 2021. 2, 3, 6, 7

[50] Yuanhao Zhai, Le Wang, Wei Tang, Qilin Zhang, Junsong Yuan, and Gang Hua. Two-stream consensus network for weakly-supervised temporal action localization. In *ECCV*, pages 37–54. Springer, 2020. 1, 2, 3, 6, 7, 8

[51] Can Zhang, Meng Cao, Dongming Yang, Jie Chen, and Yuexian Zou. Cola: Weakly-supervised temporal action localization with snippet contrastive learning. In *CVPR*, pages 16010–16019. IEEE, 2021. 2, 7

[52] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, pages 1325–1334. IEEE, 2018. 2

[53] Tao Zhao, Junwei Han, Le Yang, and Dingwen Zhang. Equivalent classification mapping for weakly supervised temporal action localization. *arXiv preprint arXiv:2008.07728*, 2020. 1

[54] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, pages 2914–2923. IEEE, 2017. 1, 7

[55] Jia-Xing Zhong, Nannan Li, Weijie Kong, Tao Zhang, Thomas H Li, and Ge Li. Step-by-step erasion, one-by-one collection: A weakly supervised temporal action detector. In *ACM MM*, pages 35–44. ACM, 2018. 2