Action Unit Memory Network for Weakly Supervised Temporal Action Localization

Wang Luo[1], Tianzhu Zhang[1,*], Wenfei Yang[1], Jingen Liu[2], Tao Mei[2], Feng Wu[1], Yongdong Zhang[1]
[1] University of Science and Technology of China; [2] JD AI Research
{lw1998,yangwf}@mail.ustc.edu.cn; {tzzhang,fengwu,zhyd73}@ustc.edu.cn;
jingenliu@gmail.com; tmei@live.com

Abstract

Weakly supervised temporal action localization aims to detect and localize actions in untrimmed videos with only video-level labels during training. However, without frame-level annotations, it is challenging to achieve localization completeness and relieve background interference. In this paper, we present an Action Unit Memory Network (AUMN) for weakly supervised temporal action localization, which can mitigate the above two challenges by learning an action unit memory bank. In the proposed AUMN, two attention modules are designed to update the memory bank adaptively and learn action units specific classifiers. Furthermore, three effective mechanisms (diversity, homogeneity and sparsity) are designed to guide the updating of the memory network. To the best of our knowledge, this is the first work to explicitly model the action units with a memory network. Extensive experimental results on two standard benchmarks (THUMOS14 and ActivityNet) demonstrate that our AUMN performs favorably against state-of-the-art methods. Specifically, the average mAP of IoU thresholds from 0.1 to 0.5 on the THUMOS14 dataset is significantly improved from 47.0% to 52.1%.

1. Introduction

Temporal action localization (TAL) is an important yet challenging task for video understanding. Its goal is to localize temporal boundaries of actions with specific categories in untrimmed videos [13, 7]. Because of its broad applications in high-level tasks such as video surveillance [40], video summarization [17], and event detection [15], TAL has recently drawn increasing attentions from the community. Up to now, deep learning based methods have made impressive progresses in this area. However, most of them handle this task in a fully supervised way, requiring massive temporal boundary annotations for actions [24, 51, 5, 42, 36]. Such manual annotations are expensive to obtain, which limits the development potential of fully-supervised methods in real-world scenarios.

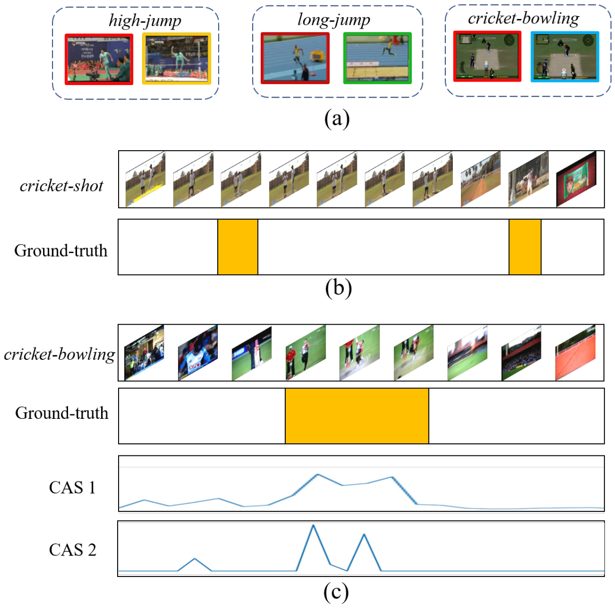

Figure 1. (a) Illustration of “Sharing Units” characteristic. The running (red box) is a shared action unit among *high-jump*, *long-jump* and *cricket-bowling*. (b) Illustration of “Sparsity” characteristic. An action usually occupies a small portion of untrimmed videos. (c) Illustration of “Smoothness” characteristic. CAS1 is more suitable for the action localization task because the CAS2 tends to divide a continuous action into multiple instances.

To relieve this problem, the weakly supervised setting that only requires video-level category labels is proposed [37, 55, 39, 37, 55, 53, 29, 30, 45, 46]. It can be formulated as a multiple instance learning problem, where a video is treated as a bag of multiple segments and fed into a video-level classifier to get a class activation sequence (CAS). There are two primary challenges, named localization completeness and background interference. To solve the first challenge, previous works usually adopt a well-designed erasing strategy [37, 55, 53] or a

[*] Corresponding Author

multi-branch architecture [21]. Both of them aim to force the model to concentrate on different parts of videos and hence discover the whole action without missing any relevant segments. To handle the second challenge, some methods [31, 28, 12] employ an attention-based per-class feature aggregation scheme, where the class-specific attention is obtained by normalizing the CAS along the temporal axis. This scheme helps learn a compact intra-class feature, which enables action segments to be more discriminative than the background. Furthermore, one popular way to handle both challenges is to learn a class-agnostic attention mechanism [29, 30, 16, 34, 10], to highlight action segments and suppress background segments.

By studying all previous TAL methods, we sum up the following three important observations (*i.e.*, TAL Properties): (1) Sharing Units. An action to be detected generally consists of some primary action units, which can be shared with other action classes. For example, as shown in Figure 1 (a), a *high-jump* contains running and jumping upward while a *long-jump* consists of running and jumping forward, so running is a shared action unit. (2) Sparsity. In general, only a sparse set of video segments contains the meaningful target actions. As we can see from Figure 1 (b), an action only occupies a small portion of the video. (3) Smoothness. A smooth CAS is required for localization, because an action is continuous, as shown in Figure 1 (c). These three characteristics are critical for the success of action localization. Unfortunately, they have not been thoroughly addressed by previous studies. To achieve accurate and complete action localization, these three observations should be taken into consideration when designing an action localization model. However, with only video-level labels, it is difficult to model them jointly in a unified model.

To fully leverage the above three characteristics for action localization, we propose a novel end-to-end framework, called Action Unit Memory Network (AUMN), for more effective weakly supervised action localization. Our framework starts with the action unit templates learning. According to the "Sharing Units" characteristic, we design a sub-network as a memory bank of action unit templates, which serve as our learning primary for action localization. To exploit the templates for action classification, we further design a Multi-Layer Perceptron (MLP) network to embed each template into the action class space. Basically, the MLP network helps connect templates to action classes. Intuitively speaking, action unit templates will be projected onto a set of action classifiers. Afterwards, a cross-attention module is proposed to compute the relationships between a video segment and all templates. And according to the "Smoothness" characteristic, we introduce a self-attention module to compute the relationships between different segments in a video for aggregating context information. Leveraging both of the attention mechanisms, we can get refined segment features and be able to dynamically select action classifiers for each video segment, which in turn, simultaneously contribute the adaptive learning of the memory bank.

However, the video-level ground-truth supervision alone is not enough for memory updating. Based on the property of action units and "Sparsity" characteristic, we further design three effective mechanisms to guide the updating of the memory bank: (1) Since action units are different from each other, each template in the memory bank should be unique. To achieve this goal, we design a diversity mechanism to encourage the differences among the templates in the memory. (2) While the diversity mechanism can encourage each template in the memory to be unique, it does not guarantee that no template is useless, which means that a template may have low similarities with all video segments. To avoid this, we design a homogeneity mechanism to encourage a uniform distribution for the occurring probability of templates. (3) In an untrimmed video, action segments only occupy a small portion of the whole video, and most of the video segments are background. Thus we design a sparsity mechanism to encourage that only a sparse set of video segments can have high similarities with the templates in the memory. These three mechanisms together with the supervision of video-level category label can guide the network to learn meaningful action units.

To sum up, the main contributions of our work are three-fold: (1) To the best of our knowledge, we are the first to model the action units with a memory network for the weakly supervised TAL task. (2) We propose two attention modules to ensure our memory to update adaptively and learn action units specific classifiers. Further, three effective mechanisms (diversity, homogeneity and sparsity) are designed to guide the updating. (3) Extensive experimental results on two challenging benchmarks including THUMOS14 [13] and ActivityNet [3] demonstrate that the proposed AUMN performs favorably against state-of-the-art weakly supervised TAL methods.

2. Related Work

In this section, we overview methods that are related to fully and weakly supervised temporal action localization and memory networks.

Fully Supervised Temporal Action Localization. Temporal action localization (TAL) aims to not only recognize actions in untrimmed videos but also give an accurate temporal proposal for each action, which makes it very challenging. To tackle this problem, fully supervised based methods have been extensively studied recently, where the frame-level annotations are required during training [50, 36, 5, 54, 48, 1, 51]. Most of these methods borrow intuitions from the object detection frameworks [9, 33, 22, 8, 32]. In specific, many methods adopt a two-stage pipeline, *i.e.*, action proposals are generated first and then fed into a classification module. For proposal generation, some methods adopt the sliding window [50, 36, 44, 41] and others predict temporal boundaries of action instances directly [5, 54, 2, 20, 18]. In addition to the two-stage methods, one-stage methods are proposed to predict action category and temporal boundaries from raw data directly, which are more flexible and efficient [48, 1, 24, 19].

Weakly Supervised Temporal Action Localization. Weakly supervised methods tackle the same problem but with less supervision, *e.g.*, video-level category labels. This pipeline can alleviate the requirement for expensive action boundary annotations, but raise two challenges named localization completeness and background interference. To handle the two problems, existing methods can be divided into three types. The first type of works attempt to solve the localization completeness by applying a well-designed erasing strategy [37, 55, 53] or a multi-branch architecture [21]. For example, Zhong et al. [55] design a step-by-step erasion approach to train the one-by-one classifiers, via collecting detection results from these classifiers, more action segments are found. And in CMCS [21], a multi-branch network with a diversity loss is proposed to make the model focus on different parts of videos. The second type of works aim to tackle the background interference via a intra-class feature compactness scheme [31, 28, 12, 27]. They first compute the class-specific attention by applying the softmax function to CAS then use this attention to get an aggregated video-level feature. By devising different mechanisms to learn a compact intra-class feature, action and background segments tend to be separated. For example, to decrease the intra-class variance, 3C-Net [28] and A2CL-PT [27] maintain a set of class center and RPN [12] learns class-specific prototypes. The third type of works are based on a class-agnostic attention mechanism [29, 30, 16, 34, 10, 52, 25], which can consider both the challenges simultaneously. Unlike the second type of works, the attention here is generated in a bottom-up way from the raw data and trained for highlighting foreground segments. It is first proposed by STPN [29] and then inspires many following methods. Some of them introduce an auxiliary category to focus on modeling background [30, 16]. And based on the observation that background features differ from action features, DGAM [34] adopts a conditional variation auto-encoder to construct different feature distributions conditioned on the attention. Recently, TSCN [52] and EM-MIL [25] fuse the output of different modalities (RGB and optical flow) to generate pseudo labels for guidance of the attention.

Memory Networks. Memory networks typically involve an internal memory implicitly updated in a recurrent process, e.g., LSTM [11], or an explicit memory bank that can be read or written with an attention based mechanism. Memory networks that can be trained end-to-end are first proposed in the natural language processing research like question answering [26] and sentiment analysis [6]. Recently, in the temporal action localization task, a popular use of memory is exploring the temporal structure based on the LSTM [47, 48, 38]. The ability of LSTM to learn from long sequences with unknown size of background is well-suited for fine-grained action localization from untrimmed videos. Instead of exploiting the temporal relationships in a video, we propose an attention-based memory mechanism to model the action units which are shared among all the videos. This mechanism helps us to deal with the large intra-class variations, so that we can get more complete localization results by discovering various action units.

3. Our Proposed Approach

In this section, we first formulate the task of weakly supervised Temporal Action Localization. Then we describe each composition of the proposed Action Unit Memory Network (AUMN) in details.

3.1. Notations and Preliminaries

Assume we have N untrimmed training videos $\{v_i\}_{i=1}^{N}$. Each video v_i has its ground-truth label $\mathbf{y}_i \in \mathbb{R}^C$, where C is the number of action categories. $\mathbf{y}_i(j) = 1$ if the action category j is present in the video and $\mathbf{y}_i(j) = 0$ otherwise[1]. During testing, the goal of the temporal action localization is to generate a set of action proposals $\{(c, s, e, q)\}$ for each video, where c and q denote the predicted category and the confidence score, s and e represent the start and the end time respectively. In this paper, we follow previous works [29, 28, 34] to extract features for both RGB and optical flow streams. Given an untrimmed video v_i, we first divide it into non-overlapping 16-frame segments and apply the I3D pretrained on the Kinetics dataset to extract features for each segment. Then we get two segment-wise features $\mathbf{X}_i^{RGB} \in \mathbb{R}^{l_i \times D}$ and $\mathbf{X}_i^{FLOW} \in \mathbb{R}^{l_i \times D}$, where l_i denotes the number of segments in video v_i and D is the dimension of features. Because the RGB and FLOW streams are trained independently, we use $\mathbf{X}_i$ to represent them in the rest of this paper for simplicity. Since the extracted features from I3D are learned for the action recognition task originally, it is desired to add a task-adaption layer to refine the extracted features. In specific, we adopt a temporal convolutional layer with the ReLU activation as

$$\mathbf{X}_i^e = \mathrm{ReLU}(W^{emb} * \mathbf{X}_i + b^{emb}), \tag{1}$$

where the $*$ represents the convolution operation, W^{emb} and b^{emb} are the weights and bias of temporal filters, $\mathbf{X}_i^e \in \mathbb{R}^{l_i \times F}$ is the learned embedding feature, and F is the dimension of learned features.

3.2. Action Unit Memory Network

The overall architecture of our action unit memory network is shown in Figure 2. It consists of three parts including feature extraction, memory bank construction, memory

[1] If there are multiple action categories in one video, $\mathbf{y}_i$ is normalized with the ℓ_1 normalization.

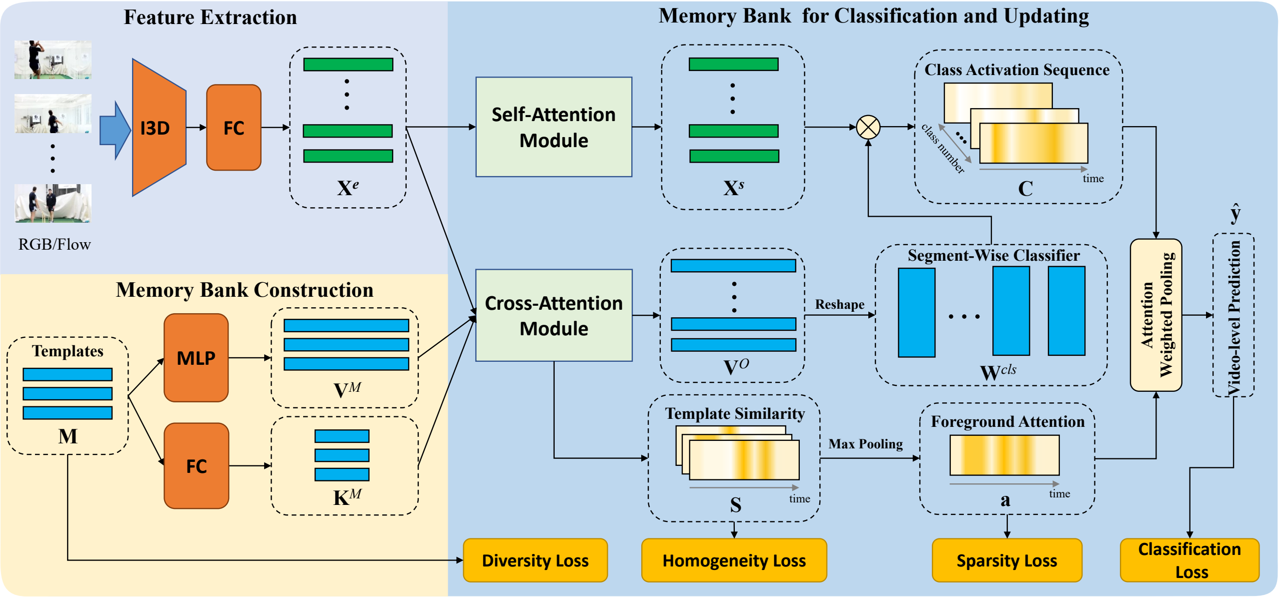

Figure 2. Overall architecture of our proposed Action Unit Memory Network (AUMN), which consists of three parts: feature extraction, memory bank construction, memory bank for classification and updating.

bank for classification and updating. The details are introduced as follows.

Memory Bank Construction. The action unit memory bank stores multiple templates $\mathbf{M} \in \mathbb{R}^{K\times F}$, and each template represents an action unit, where K and F are the number and dimension of templates respectively. We adopt two encoders named $\mathbf{Enc}_K$ and $\mathbf{Enc}_V$ to embed the templates into pairs of keys and values respectively. The $\mathbf{Enc}_K$ is designed to reduce the dimension of the templates for efficient reading from the memory, which is implemented as a fully connected layer (FC). And the $\mathbf{Enc}_V$ is designed to encode each template into a template-specific classifier, which is a MLP network consisted of two FC layers with a bottleneck structure among them to reduce parameters. In this way, the keys store appearance and motion related information for the templates and can be used for template matching during memory reading, and the values store templates specific classifiers and can be used for segment classification. Formally, we formulate the encodings as follows:

$$\mathbf{K}^M = \mathbf{Enc}_K(\mathbf{M}), \tag{2}$$

$$\mathbf{V}^M = \mathbf{Enc}_V(\mathbf{M}), \tag{3}$$

where $\mathbf{K}^M \in \mathbb{R}^{K\times F/m}$ and $\mathbf{V}^M \in \mathbb{R}^{K\times CF}$ are keys and values, M denotes the memory, and m is a hyper-parameter to control memory reading efficiency. Given the memory bank and an input video, we introduce how to perform video classification and memory updating next.

Memory Bank for Classification. For video classification, we use an encoder $\mathbf{Enc}_Q$ which is implemented as a FC layer to encode video feature $\mathbf{X}_i^e$ into a set of queries $\mathbf{Q}_i \in \mathbb{R}^{l_i\times F/m}$, and then feed the segment features and queries into a self-attention module and a cross attention module to generate classification results. In the self-attention module, we first calculate the similarity scores among video segments with queries and then use these scores to refine the segment features by aggregating context information, which can be formulated as

$$\mathbf{X}_i^s = (\mathrm{softmax}(\frac{\mathbf{Q}_i\mathbf{Q}_i^T}{\sqrt{F/m}}) + \mathbf{I})\mathbf{X}_i^e, \tag{4}$$

where $\mathbf{I}$ is the identity matrix used to preserve the original information, and $\mathbf{X}_i^s$ keeps the same dimension with $\mathbf{X}_i^e$. Via this message passing between segments, we can extract global context information and get more discriminative features for both classification and localization.

In the cross-attention module, we read from the memory and get a set of segment-wise classifiers. To achieve this goal, we first calculate the similarity scores $\mathbf{S}_i \in \mathbb{R}^{l_i\times K}$ between video segments and memory templates with the scaled dot-product

$$\mathbf{S}_i = \mathrm{sigmoid}(\frac{\mathbf{Q}_i(\mathbf{K}^M)^T}{\sqrt{F/m}}). \tag{5}$$

Based on the similarity scores, we can obtain a set of segment-wise classifiers by using the similarity scores to aggregate memory values as

$$\mathbf{V}_i^O = \mathbf{S}_i\mathbf{V}^M, \tag{6}$$

where $\mathbf{V}_i^O \in \mathbb{R}^{l_i\times CF}$. Later, to perform classification, we reshape $\mathbf{V}_i^O$ into a set of segment classifiers $\mathbf{W}_i^{cls} = \{\mathbf{W}_i^{cls}(t) \in \mathbb{R}^{F\times C}\}_{t=1}^{l_i}$, which is adaptive to the appearance or motion variations of each segment.

With the refined features of the self-attention module and the segment-wise classifiers of the cross-attention module, we can obtain the segment-level classification results by applying each classifier on the corresponding segment. Since we only have video-level ground-truth supervision, we need to aggregate these segment-level classification results into a video-level prediction. In specific, since the second dimension of the similarity matrix $\mathbf{S}_i$ denotes the similarity between a segment and a template, we can apply the max-pooling operation along the second dimension of the $\mathbf{S}_i$ to get the foreground attention weight $\mathbf{a}_i \in \mathbb{R}^{l_i}$ as

$$\mathbf{a}_i = \mathrm{MaxPool}(\mathbf{S}_i), \tag{7}$$

The video-level classification result $\hat{\mathbf{y}}_i$ is then obtained as an attention weighted pooling

$$\hat{\mathbf{y}}_i = \mathrm{softmax}(\frac{1}{l_i}\sum_{t=1}^{l_i} \mathbf{a}_i(t)(\mathbf{X}_i^s(t)\mathbf{W}_i^{cls}(t))), \tag{8}$$

where $\hat{\mathbf{y}}_i \in \mathbb{R}^C$. Then the classification loss is defined as the cross-entropy loss between the prediction and the video label $\mathbf{y}_i$

$$\mathcal{L}_{cls} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{C} \mathbf{y}_i(j)\log\hat{\mathbf{y}}_i(j). \tag{9}$$

Memory Bank Updating. For the memory updating, we find that the above classification loss alone is not enough to learn a satisfying memory bank. Thus we design three mechanisms (diversity, homogeneity and sparsity) to guide the updating of the memory bank. The diversity mechanism encourages that each template in the memory bank is different from other templates, the homogeneity mechanism encourages that each template in the memory bank is meaningful, and the sparsity mechanism encourages that the templates in the memory bank can suppress background segments. In specific, in the diversity mechanism, we design a diversity loss to ensure the uniqueness of each template in the memory as

$$\mathcal{L}_d = \left\|\mathbf{M}\mathbf{M}^T - \mathbf{I}\right\|_F, \tag{10}$$

where $\mathbf{I}$ is the identity matrix and $\|\cdot\|_F$ is the Frobenius norm of a matrix. While the diversity loss encourages the templates in the memory bank to be unique, it does not guarantee that each template in the memory bank is useful. For example, a template may not represent an action unit and have low similarities with all video segments during training. To deal with this issue, we design a homogeneity loss to encourage a uniform distribution for the occurring probability of templates in the homogeneity mechanism. In specific, we first pool the similarity matrix $\mathbf{S}_i$ over time by a sum operation and then use a softmax function to obtain the occurring probability of each template as

$$\mathbf{p}_i^O = \mathrm{softmax}(\sum_{t=1}^{l_i} \mathbf{S}_i(t)), \tag{11}$$

where $\mathbf{p}_i^O \in \mathbb{R}^K$. Then the homogeneity loss can be formulated as

$$\mathcal{L}_h = \left\|\frac{1}{B}\sum_{i=1}^{B}\mathbf{p}_i^O\right\|_2, \tag{12}$$

where B is the mini-batch size. And based on the observation that an action usually occupies a small portion of a untrimmed video, we design a sparsity loss in the sparsity mechanism to relieve the interference of background segments. And the sparsity loss is designed as

$$\mathcal{L}_s = \frac{1}{B}\sum_{i=1}^{B}\|\mathbf{a}_i\|_1, \tag{13}$$

which encourages background segments to have low similarities with all the templates.

3.3. Network Training and Inference

Training. For training the whole network, we compose the classification loss and three auxiliary losses as

$$\mathcal{L} = \mathcal{L}_{cls} + \alpha\mathcal{L}_d + \beta\mathcal{L}_h + \gamma\mathcal{L}_s. \tag{14}$$

where α, β and γ are hyper-parameters to balance the contribution of each loss function. To summarize, we maintain a set of templates in representation of various action units and update them in the video-level classification task. To better guide the learning of templates, a diversity loss and a homogeneity loss are devised to keep the variety and effectiveness of each template, and a sparsity loss is introduced to relieve background interference.

Inference. After modeling action units by the AUMN, we can localize actions by examining whether a segment belongs to a kind of action unit. In specific, we take a two-step approach to perform localization. First, we threshold on video-level prediction scores $\hat{\mathbf{y}}_i$ and discard categories which have confidence scores below a threshold η_{cls}. Thereafter, for each of the remaining action categories, we apply the threshold η_{act} on the foreground attention weight to generate action proposals. To assign a confidence for each proposal, we compute the class activation sequence (CAS) $\mathbf{C}_i \in \mathbb{R}^{l_i \times C}$ first, where

$$\mathbf{C}_i(t,;) = \mathbf{X}_i^s(t)\mathbf{W}_i^{cls}(t), \tag{15}$$

then $\mathbf{C}_i$ is passed through a softmax function along the category dimension to get class scores at each time location, denoted as $\bar{\mathbf{C}}_i$. And the confidence score q in the proposal $\{(c,s,e,q)\}$ is computed as

$$q = \sum_{t=s}^{e} \frac{\theta\mathbf{a}_i^R(t)\bar{\mathbf{C}}_i^R(t,c) + (1-\theta)\mathbf{a}_i^F(t)\bar{\mathbf{C}}_i^F(t,c)}{s-e+1}, \tag{16}$$

where the superscripts R and F denote RGB or $FLOW$ streams respectively, θ is a scalar denoting the relative importance between the two modalities and is set to 0.3 in this work. To remove proposals with a high overlap, the class-wise Non-Maximal Suppression (NMS) is used.

4. Experiment

4.1. Experimental Setup

Datasets. The proposed AUMN is evaluated on two benchmark datasets including THUMOS14 [13] and ActivityNet [3]. **THUMOS14** dataset contains 200 validation videos and 213 testing videos annotated with temporal action boundaries belonging to 20 categories. This dataset is

Table 1. Localization performance comparison with state-of-the-art methods on the THUMOS14 test set. Note that weak$^+$ represents methods that utilize external supervision information besides from video labels, *i.e.*, frequency of action instances.

Supervision	Method	Feature	mAP@IoU							
			0.1	0.2	0.3	0.4	0.5	0.6	0.7	AVG (0.1:0.1:0.5)
Fully	S-CNN [36], CVPR2016	-	47.7	43.5	36.3	28.7	19.0	-	-	35.0
	R-C3D [42], ICCV2017	-	54.5	51.5	44.8	35.6	28.9	-	-	43.1
	SSN [54], ICCV2017	-	66.0	59.4	51.9	41.0	29.8	-	-	49.6
	TAL-Net [5], CVPR2018	-	59.8	57.1	53.2	48.5	42.8	33.8	20.8	52.3
	GTAN [24], CVPR2019	-	69.1	63.7	57.8	47.2	38.8	-	-	55.3
Weakly$^+$	STAR [43], AAAI2019	I3D	68.8	60.0	48.7	34.7	23.0	-	-	47.0
	3C-Net [28], ICCV2019	I3D	59.1	53.5	44.2	34.1	26.6	-	8.1	43.5
Weakly	UntrimmedNet [39], CVPR2017	-	44.4	37.7	28.2	21.1	13.7	-	-	29.0
	Hide-and-Seek [37], ICCV2017	-	36.4	27.8	19.5	12.7	6.8	-	-	20.6
	Zhong et al. [55], MM2018	-	45.8	39.0	31.1	22.5	15.9	-	-	30.9
	AutoLoc [35], ECCV2018	UNT	-	-	35.8	29.0	21.2	13.4	5.8	-
	Clean-Net [23], ICCV2019	UNT	-	-	37.0	30.9	23.9	13.9	7.1	-
	STPN [29], CVPR2018	I3D	52.0	44.7	35.5	25.8	16.9	9.9	4.3	35.0
	WTALC [31], ECCV2018	I3D	55.2	49.6	40.1	31.1	22.8	-	7.6	39.8
	CMCS [21], CVPR2019	I3D	57.4	50.8	41.2	32.1	23.1	15.0	7.0	40.9
	ASSG [53], MM2019	I3D	55.6	49.5	41.1	31.5	20.9	13.7	5.9	39.7
	TSM [49], ICCV2019	I3D	-	-	39.5	31.9	24.5	13.8	7.1	-
	Nguyen et al. [30], ICCV2019	I3D	60.4	56.0	46.6	37.5	26.8	19.6	9.0	45.5
	TCAM [10], CVPR2020	I3D	-	-	46.9	38.9	30.1	19.8	10.4	-
	DGAM [34], CVPR2020	I3D	60.0	54.2	46.8	38.2	28.8	19.8	11.4	45.6
	BaS-Net [16], AAAI2020	I3D	58.2	52.3	44.6	36.0	27.0	18.6	10.4	43.6
	RPN [12], AAAI2020	I3D	62.3	57.0	48.2	37.2	27.9	16.7	8.1	46.5
	EM-MIL [25], ECCV2020	I3D	59.1	52.7	45.5	36.8	30.5	**22.7**	**16.4**	44.9
	A2CL-PT [27], ECCV2020	I3D	61.2	56.1	48.1	39.0	30.1	19.2	10.6	46.9
	TSCN [52], ECCV2020	I3D	63.4	57.6	47.8	37.7	28.7	19.4	10.2	47.0
	AUMN	I3D	**66.2**	**61.9**	**54.9**	**44.4**	**33.3**	20.5	9.0	**52.1**

particularly challenging as it consists of very long videos with multiple action instances of small duration. Following previous works [39, 29, 31, 23, 28, 16, 52], we use the 200 validation videos for training and the 213 testing videos for evaluation. **ActivityNet** dataset includes ActivityNet1.2 and ActivityNet1.3. ActivityNet1.3 consists of 10024 training videos, 4926 validation videos and 5044 testing videos belonging to 200 action categories. And ActivityNet1.2 is a subset of ActivityNet1.3, which covers 100 action categories with 4819 training, 2383 validation and 2480 testing videos. ActivityNet only contains 1.5 instances per video on average and most videos only contain one action category with only 36% background averagely. Following previous works [39, 29, 31, 23, 28, 16, 52], we train our model on the training set and evaluate it on the validation set.

Evaluation Metrics. Following the standard evaluation protocol, we evaluate the TAL performance with the mean Average Precision (mAP) values under different intersection over union (IoU) thresholds.

Implementation Details. We use the two-stream I3D networks [4] pre-trained on Kinetics as our feature extractor. Note that for fair comparison, we do not finetune the I3D network. We apply the TV-L1 algorithm to extract optical flow from RGB data. Then we divide both streams into non-overlapping 16 frames segments as the input to the I3D network, the dimension D of the output feature for each segment is 1024. We train separate AUMNs for RGB and FLOW streams and collect the generated proposals from both networks during inference. In AUMN, the embedding layer is composed of a temporal convolutional layer with 1024 input channels and 512 output channels. The number of templates K is 7 if not mentioned specifically. In Eq. (14), the loss function weights α = 0.01, β = 0.02, and γ is set to 0.05 and 0.03 for the RGB stream and the FLOW stream, respectively. During inference, the threshold η_{cls} is 0.1 and η_{act} is the mean value of the corresponding foreground attention $\mathbf{a}_i$ for video v_i. And we use the class-wise NMS with a threshold 0.3 to remove highly overlapped proposals. Our model is trained using Adam optimizer [14] with the learning rate 10^{-4} and batch size 32.

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

Experiments on THUMOS14. Table 1 summarizes the performance comparison between the proposed AUMN and state-of-the-art TAL methods on the THUMOS14 test set. Weakly$^+$ denotes methods that adopt additional supervision during training, *e.g.*, the number of action instances in a video, and AVG indicates the average mAP for IoU thresholds 0.1:0.1:0.5. From the results, we can see that the proposed AUMN outperforms all the previous weakly supervised models and achieves a new state-of-the-art performance (33.3% mAP at IoU0.5). And an absolute gain of 5.1% is achieved in terms of the average mAP when compared to the best previous method (TSCN [52]). It is worth noting that EM-MIL [25] gets a higher mAP at IoU thresholds 0.6 and 0.7 than ours. However, we get 7% improvement than EM-MIL at average mAP. Besides, EM-

Table 2. Localization performance comparison with state-of-the-art methods on the ActivityNet1.2 validation set.

Method	mAP@IoU			
	0.5	0.75	0.95	AVG
UntrimmedNet [39]	7.4	3.9	1.2	3.6
Zhong et al. [55]	27.3	14.7	2.9	15.6
AutoLoc [35]	27.3	15.1	3.3	16.0
WTALC [31]	37.0	14.6	-	18.0
TSM [49]	28.3	17.0	3.5	-
CMCS [21]	36.8	22.0	5.6	22.4
Clean-Net [23]	37.1	20.3	5.0	21.6
3C-Net [28]	37.2	23.7	-	21.7
Bas-Net [16]	38.5	24.2	5.6	24.3
Huang et al. [12]	37.6	23.9	5.4	23.3
TCAM [10]	40.0	25.0	4.6	24.6
DGAM [34]	41.0	23.5	5.3	24.4
EM-MIL [25]	37.4	23.1	2.0	20.3
TSCN [52]	37.6	23.7	**5.7**	23.6
AUMN (Our's)	**42.0**	**25.0**	5.6	**25.5**

MIL adopts a pseudo label scheme to relieve background interference while we adopt a simple sparsity prior. We believe the performance of our approach can be promoted further when equipped with a more effective background suppression techniques. Compared to the weakly$^+$ methods, our method outperforms 3C-Net [28] at all IoU thresholds and achieves 5.1% improvement over STAR [43] in average mAP. When compared with fully supervised methods, we note that the performance of AUMN drops faster than fully supervised methods as the IoU threshold increases. However, we can also get a comparable result at low IoU thresholds, e.g., AUMN outperforms TAL-Net at IoU thresholds 0.1, 0.2 and 0.3.

Experiments on ActivityNet. On the ActivityNet dataset, we follow the standard evaluation protocol [3] by reporting the average mAP scores at different thresholds (0.5:0.05:0.95). The performance comparisons on the ActivityNet1.2 and ActivityNet1.3 are shown in Table 2 and Table 3, respectively. The results are consistent with those on the THUMOS14 dataset, and our AUMN outperforms all previous weakly supervised models in average mAP on both ActivityNet1.2 and ActivityNet1.3, with 25.5% and 23.5% average mAP respectively. It is worth noting THUMOS14 dataset and ActivityNet dataset have different characteristics. For the THUMOS14 dataset, the most important thing is the background suppression. While for the ActivityNet dataset, the most important thing is the localization completeness. At high IoU thresholds, EM-MIL has better performance on the THUMOS14 dataset while worse performance on the ActivityNet dataset. This is because EM-MIL mainly considers background suppression while ignores the localization completeness. Different from existing methods, our AUMN takes both background suppression and localization completeness into consideration, and can achieve favorable performance on both datasets.

4.3. Ablation Studies

In this section, we conduct a series of ablation studies on the THUMOS14 dataset to evaluate the influence of each design.

Table 3. Localization performance comparison with state-of-the-art methods on the ActivityNet1.3 validation set.

Method	mAP@IoU			
	0.5	0.75	0.95	AVG
STPN [29]	29.3	16.9	2.6	16.3
ASSG [53]	32.3	20.1	4.0	18.8
CMCS [21]	34.0	20.9	**5.7**	21.2
STAR [43]	31.1	18.8	4.7	18.2
TSM [49]	30.3	19.0	4.5	-
Nguyen et al. [30]	36.4	19.2	2.9	19.5
Bas-Net [16]	34.5	22.5	4.9	22.2
TSCN [52]	35.3	21.4	5.3	21.7
A2CL-PT [27]	36.8	22.0	5.2	22.5
AUMN	**38.3**	**23.5**	5.2	**23.5**

Table 4. Ablation studies on the THUMOS14 dataset, where $\mathcal{L}_s$, $\mathcal{L}_d$, $\mathcal{L}_h$ denote the sparsity loss, the diversity loss and the homogeneity loss. Here, S denotes the self-attention module.

$\mathcal{L}_s$	$\mathcal{L}_d$	$\mathcal{L}_h$	S	mAP@IoU							
				0.1	0.2	0.3	0.4	0.5	0.6	0.7	AVG
-	-	-	-	58.5	53.1	45.1	34.9	24.6	14.4	6.8	43.2
✓	-	-	-	65.1	59.8	51.5	40.9	28.8	16.3	7.3	49.2
✓	✓	-	-	65.8	61.0	52.6	42.2	29.5	17.0	7.6	50.2
✓	-	✓	-	65.5	60.9	51.7	41.3	29.4	17.1	7.7	49.8
✓	✓	✓	-	66.1	61.5	54.4	43.3	31.8	19.1	8.9	51.4
✓	✓	✓	✓	**66.2**	**61.9**	**54.9**	**44.4**	**33.3**	**20.5**	**9.0**	**52.1**

Influence of Each Loss Function. As introduced in Section 3.2, we design three auxiliary losses (diversity loss $\mathcal{L}_d$, homogeneity loss $\mathcal{L}_h$ and sparsity loss $\mathcal{L}_s$) to guide the memory updating. To explore the influence of each loss function, we conduct experiments with different loss combinations, and the results are shown in Table 4. From the results, we have the following observations: (1) The sparsity loss $\mathcal{L}_s$ can bring a significant performance improvement at all IoU thresholds. Because there is no frame-level label as the supervision of the foreground attention, $\mathcal{L}_s$ can serve as a prior to guide the action unit templates to focus on the action related segments. (2) The diversity loss $\mathcal{L}_d$ is designed to encourage the action unit templates in the memory bank to be different from each other. Without the diversity loss, we can only rely on the random initialization to achieve our goal. From the results, we can see that the diversity loss can bring a 1.0% performance gain in average mAP, which indicates that the diversity loss is necessary. (3) The homogeneity loss $\mathcal{L}_h$ is designed to guarantee that each learned action unit template is useful. Without the $\mathcal{L}_h$, some templates in the memory bank may be useless, which may decrease the representative ability of the memory bank. When equipped with this loss, we observe a 0.6% performance gain in average mAP. (4) These three losses can promote each other. For example, the $\mathcal{L}_d$ can keep the difference between templates, but cannot ensure each learned template is useful. On the other hand, although $\mathcal{L}_d$ can ensure no template is redundant, it may lead to learning a set of identical memory templates. By combining them together, the mAP is increased by 3% at IoU = 0.5, which is much more significant than applying them independently.

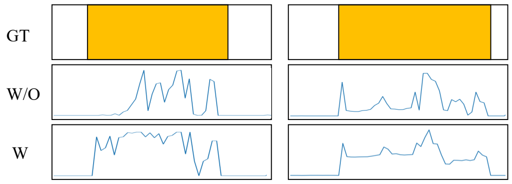

Figure 3. Illustration of the class activation sequence with (W) and without (W/O) the self-attention module.

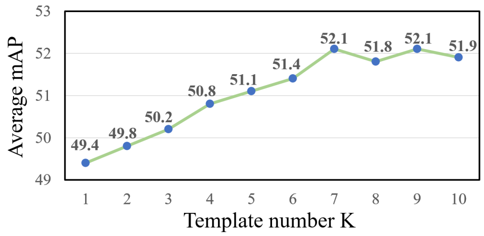

Figure 4. Performance comparison of different template numbers. The average mAP is computed at IoU thresholds 0.1:0.1:0.5. Note that the $\mathcal{L}_d$ and $\mathcal{L}_h$ are removed when K = 1.

Influence of the Self-attention Module. As introduced in Section 3.2, the self-attention module is designed to incorporate context information so as to encourage a smoother temporal classification score, which is important for complete action localization. From the results in Table 4, the self-attention module can consistently improve the performance at all IoU thresholds. And it is worth noting that the performance gain at IoU = 0.4, 0.5, 0.6 is more significant than that at IoU = 0.1, 0.2, 0.3. To further verify the self-attention design, we show several visualization results in Figure 3. With the self-attention module, some less discriminative action segments can be assigned higher confidence scores, and it means that the self-attention module can indeed help to improve localization completeness.

Influence of the Template Number. To explore the influence of the template number, we conduct experiments on the THUMOS14 dataset and report the average mAP at IoU 0.1:0.1:0.5 of AUMN with different template numbers. The results are shown in Figure 4, the average mAP can be consistently improved as the template number K grows from 1 to 7, which means 7 templates are sufficient to model all the action units on the THUMOS14 dataset.

4.4. Qualitative Results

To better understand our method, the qualitative results of our AUMN on three videos from the ActivityNet1.2 validation set are presented in Figure 5. The action instances from left to right are *javelin-throw*, *long-jump* and *high-jump* respectively. We visualize the similarities between video segments with different templates in third to fifth rows. And the 6th row is the foreground attention $\mathbf{a}$. We find that different templates attend to model different visual patterns. For example, the first template has a high similarity to the segments which contain the action unit running while the second is similar to jumping. The third template tends to focus on throwing, which is an important action unit in *javelin-throw*. Interestingly, some segments of throwing are a little similar to the first template, because the man still keeps running while throwing the javelin. It is worth noting that *long-jump* and *high-jump* both contain segments about jumping, to distinguish them from each other, the segment-wise classifiers defined in Eq (6) are desired. In summary, by finding action units in untrimmed videos via the templates from memory and utilizing the segment-wise classifiers, we can correctly recognize an action and obtain robust foreground attentions for complete action localization.

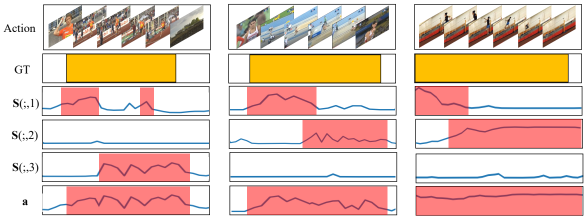

Figure 5. Qualitative results on ActivityNet1.2. The action instances from left to right are *javelin-throw*, *long-jump* and *high-jump* respectively. The six rows in each example are input video, ground truth action instance, three different subsets of similarities scores $\mathbf{S}$ in Eq. (5) and the foreground attention $\mathbf{a}$.

5. Conclusion

In this paper, we propose an Action Unit Memory Network (AUMN) to model action units for weakly supervised temporal action localization. We design a memory bank to store the appearance and motion information of action units and their corresponding classifiers. We further introduce a cross-attention module to read segment-wise classifiers from the memory and a self-attention module for refining features by aggregating temporal context information. Then we can get segment-level predictions and update the memory in an adaptive way with three auxiliary mechanisms (diversity, homogeneity and sparsity). With a meaningful memory bank, we can achieve more complete localization results by finding action units in untrimmed videos. Extensive experimental results on two benchmarks demonstrate the effectiveness of the proposed AUMN.

6. Acknowledgement

This work was funded by JD AI Research, and partially supported by the National Key Research and Development Program under Grant No. 2017YFC0820600, Strategic Priority Research Program of Chinese Academy of Sciences (No.XDC02050500), National Defense Basic Scientific Research Program (JCKY2020903B002), National Nature Science Foundation of China (Grant 62022078, 62021001, 62071122), Open Project Program of the National Laboratory of Pattern Recognition (NLPR) under Grant 202000019, and Youth Innovation Promotion Association CAS 2018166.

References

[1] Shyamal Buch, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. In *BMVC*, volume 2, page 7, 2017.

[2] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2911–2920, 2017.

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

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[5] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1130–1139, 2018.

[6] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 452–461, 2017.

[7] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Temporal localization of actions with actoms. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2782–2795, 2013.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015.

[10] Guoqiang Gong, Xinghan Wang, Yadong Mu, and Qi Tian. Learning temporal co-attention models for unsupervised video action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9819–9828, 2020.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[12] Linjiang Huang, Yan Huang, Wanli Ouyang, Liang Wang, et al. Relational prototypical network for weakly supervised temporal action localization. 2020.

[13] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos “in the wild”. *Computer Vision and Image Understanding*, 155:1–23, 2017.

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

[15] Suha Kwak, Bohyung Han, and Joon Hee Han. Multi-agent event detection: Localization and role assignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

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

[17] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1346–1353. IEEE, 2012.

[18] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3898, 2019.

[19] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 988–996. ACM, 2017.

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

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

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 3899–3908, 2019.

[24] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 344–353, 2019.

[25] 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. *arXiv preprint arXiv:2004.00163*, 2020.

[26] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*, 2016.

[27] Kyle Min and Jason J Corso. Adversarial background-aware loss for weakly-supervised temporal activity localization. *arXiv preprint arXiv:2007.06643*, 2020.

[28] Sanath Narayan, Hisham Cholakkal, Fahad Shabaz Khan, and Ling Shao. 3c-net: Category count and center loss for weakly-supervised action localization. *arXiv preprint arXiv:1908.08216*, 2019.

[29] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6752–6761, 2018.

[30] Phuc Xuan Nguyen, Deva Ramanan, and Charless C Fowlkes. Weakly-supervised action localization with background modeling. *arXiv preprint arXiv:1908.06552*, 2019.

[31] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 563–579, 2018.

[32] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[34] Baifeng Shi, Qi Dai, Yadong Mu, and Jingdong Wang. Weakly-supervised action localization by generative attention modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1009–1019, 2020.

[35] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 154–171, 2018.

[36] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1049–1058, 2016.

[37] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3544–3553. IEEE, 2017.

[38] Chen Sun, Sanketh Shetty, Rahul Sukthankar, and Ram Nevatia. Temporal localization of fine-grained actions in videos by domain transfer from web images. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 371–380, 2015.

[39] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4325–4334, 2017.

[40] Yunbo Wang, Mingsheng Long, Jianmin Wang, and Philip S Yu. Spatiotemporal pyramid network for video action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1529–1538, 2017.

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

[42] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017.

[43] 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. *arXiv preprint arXiv:1811.07460*, 2018.

[44] Ke Yang, Peng Qiao, Dongsheng Li, Shaohe Lv, and Yong Dou. Exploring temporal preservation networks for precise temporal action localization. *arXiv preprint arXiv:1708.03280*, 2017.

[45] Wenfei Yang, Tianzhu Zhang, Xiaoyuan Yu, Qi Tian, Yongdong Zhang, and Feng Wu. Uncertainty guided collaborative training for weakly supervised temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[46] Wenfei Yang, Tianzhu Zhang, Yongdong Zhang, and Feng Wu. Local correspondence network for weakly supervised temporal sentence grounding. *IEEE Transactions on Image Processing*, 2021.

[47] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, 126(2-4):375–389, 2018.

[48] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2678–2687, 2016.

[49] Tan Yu, Zhou Ren, Yuncheng Li, Enxu Yan, Ning Xu, and Junsong Yuan. Temporal structure mining for weakly supervised action detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5522–5531, 2019.

[50] Jun Yuan, Bingbing Ni, Xiaokang Yang, and Ashraf A Kassim. Temporal action localization with pyramid of score distribution features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3093–3102, 2016.

[51] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. *arXiv preprint arXiv:1909.03252*, 2019.

[52] Yuanhao Zhai, Le Wang, Wei Tang, Qilin Zhang, Junsong Yuan, and Gang Hua. Two-stream consensus networks for weakly-supervised temporal action localization. In *16th European Conference on Computer Vision (ECCV)*, August 2020.

[53] Chengwei Zhang, Yunlu Xu, Zhanzhan Cheng, Yi Niu, Shiliang Pu, Fei Wu, and Futai Zou. Adversarial seeded sequence growing for weakly-supervised temporal action localization. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 738–746. ACM, 2019.

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

[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 *2018 ACM Multimedia Conference on Multimedia Conference*, pages 35–44. ACM, 2018.