Uncertainty Guided Collaborative Training for Weakly Supervised Temporal Action Detection

Wenfei Yang[1], Tianzhu Zhang[1,*], Xiaoyuan Yu[2], Tian Qi[2], Yongdong Zhang[1], FengWu[1]
University of Science and Technology of China[1], Huawei Cloud[2]
yangwf@mail.ustc.edu.cn, tzzhang@ustc.edu.cn, {yuxiaoyuan, tian.qi1}@huawei.com
{zhyd73, fengwu}@ustc.edu.cn

Abstract

Weakly supervised temporal action detection aims to localize temporal boundaries of actions and identify their categories simultaneously with only video-level category labels during training. Among existing methods, attention based methods have achieved superior performance by separating action and non-action segments. However, without the segment-level ground-truth supervision, the quality of the attention weight hinders the performance of these methods. To alleviate this problem, we propose a novel **Uncertainty Guided Collaborative Training (UGCT)** *strategy, which mainly includes two key designs: (1) The first design is an online pseudo label generation module, in which the RGB and FLOW streams work collaboratively to learn from each other. (2) The second design is an uncertainty aware learning module, which can mitigate the noise in the generated pseudo labels. These two designs work together to promote the model performance effectively and efficiently by imposing pseudo label supervision on attention weight learning. Experimental results on three state-of-the-art attention based methods demonstrate that the proposed training strategy can significantly improve the performance of these methods, e.g., more than 4% for all three methods in terms of mAP@IoU=0.5 on the THUMOS14 dataset.*

1. Introduction

Temporal action detection aims to localize the temporal boundaries of actions and identify their categories simultaneously in an untrimmed video [14, 8]. Because of its broad application potentials in video summarization [22, 16], video surveillance [17], visual question answering [23], and others, it has attracted more and more attention from both academia and industry in recent years. In the past decades, numerous action detection methods have been proposed based on one-stage detection framework [3, 24, 27] or two-stage detection framework [33, 41, 37]. However, most of the existing methods require a large amount of labeled data, which limits their scalability and practicability in the real-world application scenarios, because it is prohibitive and time-consuming to annotate a large dataset manually.

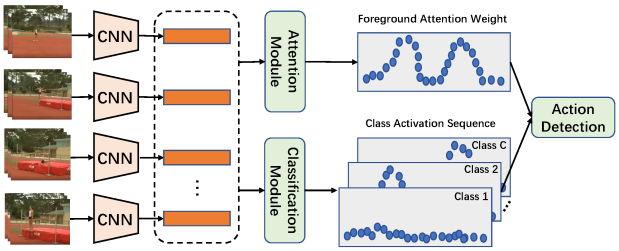


Figure 1. The action detection procedure for attention based methods: (1) The foreground attention weight is class-agnostic and used to separate action and non-action segments by performing threshold truncation on it. (2) Based on the foreground attention weight and class activation sequence, the attention weighted class activation sequence is obtained and used to score each detection region. As a key component, the quality of the attention weight has a significant impact on the detection performance. However, without the ground-truth supervision, current attention based methods cannot separate action and non-action segments well.

To overcome the above limitations, several recent works focus on weakly supervised action detection by using different kinds of information as supervision, e.g., movie scripts [1, 20], web videos [9],text descriptions [48], temporally ordered action lists [2, 11] and video-level category labels [31, 45]. Due to the low labeling cost, video-level category labels based weakly supervised temporal action detection methods have become the mainstream in this area. And existing works can be mainly categorized into three groups, learning background suppression attention weights [31, 32, 36, 21, 53], learning discriminative features [35, 30] and erasing discriminative segments during training [40, 56, 54]. Among these methods, attention based methods have achieved superior performance. Figure 1 shows the action detection procedure for these methods. An attention module is used to generate the foreground attention weight which is class-agnostic and indicates the proba-

*Corresponding Author

bility of a segment belonging to an action, and a classification module is used to generate the class-specific Class Activation Sequence (CAS). Then, action detection is achieved by using the class-agnostic attention weight and the class-specific class activation sequence.

Despite the success of these attention based methods, without the ground-truth supervision on attention weight learning, these methods cannot separate action and non-action segments well, which hinders their detection performance. To learn more robust attention weights for action and non-action segments separation, it naturally comes into mind that is it possible to provide pseudo labels to guide attention weight learning? To achieve this goal, two issues need to be considered. The first issue is how to generate segment-level pseudo labels. It should be both efficient and effective, so as to promote the model performance with little sacrifice of training time. The second issue is how to handle label noise effects in the pseudo labels, since it is inevitable that the generated pseudo labels may be noisy.

Motivated by the above discussions, we propose an Uncertainty Guided Collaboratively Training (UGCT) strategy for weakly supervised temporal action detection, and the framework is shown in Figure 2. To generate reliable pseudo labels, we design an online pseudo label generation module to generate pseudo labels with teacher models, which is inspired by the teacher-student approaches [43]. In specific, we use an exponential moving average strategy to ensemble the network weight in each training iteration to update the teacher model. In this way, the teacher model can take the historical information into consideration and provide more reliable pseudo labels. Instead of training RGB and FLOW streams independently, we propose to train them collaboratively. The teacher model in the RGB stream provides pseudo labels for the FLOW stream and the teacher model in the FLOW stream provides pseudo labels for the RGB stream. In this way, the teacher models serve as a bridge to train RGB and FLOW models collaboratively, which enables them to enhance and promote each other. To mitigate the noise in the generated pseudo labels, we propose an uncertainty aware learning module to reduce the affection of noisy labels. In specific, we add an uncertainty prediction module to predict the uncertainty about the pseudo label. The uncertainty prediction module is only used during training, so it does not bring any extra computations during testing. Based on the predicted uncertainty, a noise-robust pseudo label loss function is developed, in which the predicted uncertainty serves as a loss decay term. Segments with noisy pseudo labels tend to be assigned large uncertainties, so that the negative impact of noisy labels can be reduced.

In summary, the contributions of this paper are as follows: (1) We propose a novel uncertainty guided collaboratively training (UGCT) strategy for weakly supervised temporal action detection, which can significantly improve the performance of attention based methods without introducing any additional computational cost during testing. (2) We conduct comprehensive experiments on two benchmark datasets with three attention based methods to evaluate the effectiveness of the proposed training strategy, and the results demonstrate that the proposed UGCT can consistently improve the performance of these methods. With the proposed training strategy, we set a new state-of-the-art performance on both THUMOS14 and ActivityNet datasets.

2. Related Work

In this section, we briefly overview methods that are related to fully supervised and weakly supervised temporal action detection.

Fully Supervised Action Detection. Fully supervised action detection methods can be divided into two directions: two-stage methods [7, 39, 46, 55] and one-stage methods [3, 24, 27]. For two-stage methods, candidate action proposals are generated first and then each proposal is fed into a classifier. Early work adopts the sliding window or uniform sampling [34, 42, 51] to generate a large number of candidate action proposals, which leads to huge computation cost in later processing. And later work generates candidate proposals with content-dependent algorithms [7, 55, 4], relieving the computation burden to an extent. For example, SST [4] utilizes a recurrent GRU-based model to generate candidate proposals in one pass. And several one-stage methods have been proposed recently [3, 27, 49], where action proposals and classification scores are generated simultaneously. For example, SS-TAD [3] adopts the Recurrent Neural Network to jointly predict action categories and temporal boundaries. Different from two-stage methods, one-stage methods have higher efficiency while lower accuracy. However, fully supervised methods rely on substantial temporal action boundary annotations, which is time-consuming and expensive.

Weakly Supervised Action Detection. To address the limitation of fully supervised action detection, weakly supervised action detection has been drawing increasing research attention. In [44], it is the first method by using video-level category labels as supervision for action detection via a classification module and a selection module. And later work can be mainly grouped into three categories. The first group of works is attention based methods, which aims at highlighting foreground segments and suppressing background segments. In [31], a class-agnostic attention mechanism together with a sparsity loss is used to identify key segments associated with actions. And Nguyen et al. [32] extend this framework by introducing several background modeling losses to encourage the class-agnostic attention weight to be consistent with the learned classifier. Later, several other methods have been proposed by imposing different constrains on the attention weight, such

as DGAM [36], Bas-Net [21], STAR [47] and TSCN [53]. These methods achieve superior performances in this area, which indicates that the foreground-background separation is of vital importance. The second group of works aim at learning more discriminative features, and the basic idea of these methods is to encourage more compact intra-class feature representations by imposing different loss functions. In WTALC [35], a Co-Activity Loss is used to encourage class-specific features from two videos with the same label to be closer. And a Center Loss is proposed in 3C-Net [30] for the same goal. Inspired by above two works, more sophisticated losses are proposed in later work [13, 29, 10]. Although attention based methods and learning discriminative feature based methods have achieved remarkable progress, a common issue for these methods is that they tend to focus on the most discriminative action segments but ignore trivial action segments, which results in incomplete action localization. To mitigate this issue, the third group of works resort to the erasing mechanism to highlight less discriminative segments. For example, Hide-and-Seek [40] proposes to randomly erase input segments during training, which can force the model to discover less discriminative segments. And more sophisticated erasing mechanisms are used in later work such as Zhong et al. [56] and ASSG [54]. In this paper, we focus on the first group of works and develop an uncertainty guided collaborative training strategy, which can help to learn more robust attention weights.

3. Our Proposed Approach

In this section, we introduce some basic notations first in Section 3.1. In Section 3.2, we review the basic framework of attention based methods and introduce three methods based on which we conduct our experiments, including one pioneer work (STPN [31]) and two recent works (WSAL-BM [32], TSCN [53]). We then introduce the proposed training strategy in Section 3.3, and discuss the differences with existing work in Section 3.4.

3.1. Notations and Preliminaries

Given an untrimmed video V, we first divide it into non-overlapping 16-frame segments $V = \{v_i \in R^{16\times H\times W\times 3}\}_{i=1}^{N}$ as in previous methods [35, 38, 25, 31], where N denotes the number of segments. Each segment v_i is then fed into a pre-trained feature extraction network (for example, I3D [6]) to generate a d dimension feature vector x_i, and feature vectors of N segments are stacked together to form a feature sequence $\mathbf{X} = [x_1, x_2, ..., x_N]^\top$ as the video representation. During training, each video is associated with a ground truth label $\mathbf{Y} = [y_1, y_2, ..., y_C]$, where C denotes the number of action categories, $y_i = 1$ indicates that the $i\text{-}th$ action happens in the current video and 0 otherwise. If there may be multiple different action categories in one video, $\mathbf{Y}$ is normalized with L1 normalization [32].

3.2. Review of Attention Based Methods

A typical framework for attention based methods is shown in Figure 2 (a). Since models in the RGB and FLOW streams are trained independently, we introduce them in a unified perspective in this section. Given the extracted feature sequence $\mathbf{X}$, the first step is to feed $\mathbf{X}$ into an attention module to get the attention weight $[\lambda_1, \lambda_2, ..., \lambda_N] \in R^N$. Then the attention weight is used to aggregate segment-level features into a video-level feature as follows,

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \lambda_i * \mathbf{x}_i. \tag{1}$$

The video-level feature is further fed into the classifier to get the video-level prediction $\widetilde{\mathbf{Y}} = [\widetilde{y}_1, \widetilde{y}_2, ..., \widetilde{y}_C]$, and the classification loss is calculated as

$$\mathcal{L}_{cls} = -\sum_{i=1}^{C} y_i \log \widetilde{y}_i. \tag{2}$$

Besides from the classification loss, attention based methods usually have another attention loss $\mathcal{L}_{att}$ to make the attention weight focus on action-related segments. In this work, we experiment with three attention-based methods and their attention losses are introduced as follows. For STPN [31], a sparsity constrain is imposed on the attention weight to focus on action-related segments as

$$\mathcal{L}_{att} = \frac{1}{N}\sum_{i=1}^{N} \lambda_i. \tag{3}$$

In WSAL-BM [32], a background class is added in the classifier and the background feature is generated as

$$\mathbf{x}_{bg} = \frac{1}{N}\sum_{i=1}^{N} (1-\lambda_i) * \mathbf{x}_i. \tag{4}$$

The background feature is fed into the classifier and the attention loss is computed as the cross-entropy loss between the prediction and a background label [1]. In TSCN [53], an attention normalization loss is proposed to encourage the attention weight to act like a binary selection as

$$\mathcal{L}_{att} = \frac{1}{l}\min_{\substack{\Lambda \in \{\lambda_i\} \\ |\Lambda| = l}} \sum_{\lambda\in\Lambda} \lambda - \frac{1}{l}\max_{\substack{\Lambda \in \{\lambda_i\} \\ |\Lambda| = l}} \sum_{\lambda\in\Lambda} \lambda, \tag{5}$$

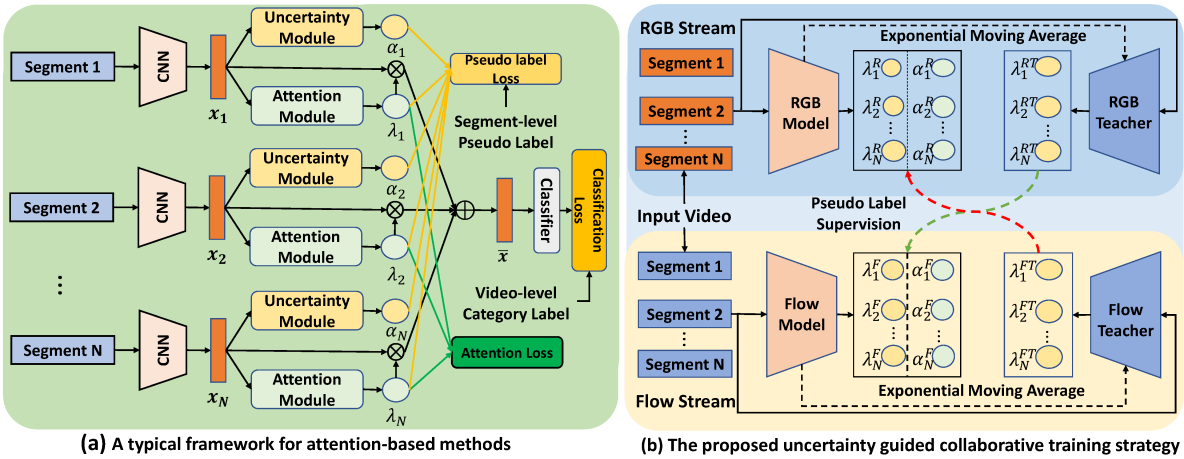
where l is set to be $N/8$.

We use $\{R, F\}$ as the modality indicator for RGB and FLOW streams respectively, and the RGB and FLOW models are trained with the weighted sum of the classification loss and the attention loss as

$$\mathcal{L}^*_{base} = \mathcal{L}^*_{cls} + \beta\mathcal{L}^*_{att}, \forall * \in \{R, F\}. \tag{6}$$

Following the practice of the original papers [31, 53, 32], the weight β for the attention loss is set to be 0.1.

[1] There are another two losses in the original paper, but we find that they cannot further promote the performance in our re-implementation.



(a) A typical framework for attention-based methods

(b) The proposed uncertainty guided collaborative training strategy

Figure 2. **(a)** A typical framework for attention based methods. Note that the uncertainty branch is newly added in our training strategy and is only used during training. **(b)** The proposed uncertainty guided collaborative training strategy. The teacher models of RGB and FLOW streams provide segment-level pseudo labels for each other, which enables the RGB and Flow models to promote each other in a collaborative way. The predicted uncertainty serves as a decay term in the pseudo label loss, which can reduce the negative impact of noisy pseudo labels during training.

3.3. Uncertainty Guided Collaborative Training

As shown in Figure 2, there are four models in our training strategy including the RGB model, the Flow model, the RGB teacher model, and the Flow teacher model. Formally, we denote the RGB and Flow models as $M(\cdot|\theta^R)$ and $M(\cdot|\theta^F)$, and the RGB and Flow teacher models as $M(\cdot|\theta^{RT})$ and $M(\cdot|\theta^{FT})$ respectively, where θ^* denotes the network parameters. There are two key designs in our training strategy including an online pseudo label generation module and an uncertainty-aware learning module.

In the online pseudo label generation module, given the input video, the pre-extracted RGB and FLOW features are fed into the RGB teacher model and the FLOW teacher model respectively. We denote the output attention weights of these two teacher models as $\Lambda^{RT} = [\lambda_1^{RT}, \lambda_2^{RT}, ..., \lambda_N^{RT}] \in R^N$ and $\Lambda^{FT} = [\lambda_1^{FT}, \lambda_2^{FT}, ..., \lambda_N^{FT}] \in R^N$. With Λ^{RT} and Λ^{FT}, the pseudo label $\tilde{\Lambda}^R \in R^N$ for the RGB stream and $\tilde{\Lambda}^F \in R^N$ for the FLOW stream are generated as $\mathcal{G}(\Lambda^{FT})$ and $\mathcal{G}(\Lambda^{RT})$, respectively. Here, $\mathcal{G}$ is a binarization function to get {0, 1}-value pseudo labels as defined in Equation (7).

$$\mathcal{G}(\Lambda)_i = \begin{cases} 1, & \text{if } \lambda_i > mean(\Lambda) \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Although more sophisticated pseudo label generation strategies can be applied, we experimentally find that this simple strategy works well enough. There are three important factors in the online pseudo label generation strategy: (1) The teacher model. During training, when the models are not converged, the teacher model can take historical information into consideration and provide more reliable pseudo labels. (2) The RGB and FLOW models provide pseudo labels for each other. The model in the RGB (FLOW) stream can refer information from the FLOW (RGB) stream, and the improvement in one modality will be delivered to the other modality immediately. (3) The binarization function $\mathcal{G}$. The attention weight is used to separate action and non-action segments by performing threshold truncation on it, thus attention weights near {0, 1}-value are preferred. The binarization function $\mathcal{G}$ can generate {0, 1}-value pseudo labels, which can encourage the model to predict near {0, 1}-value attention weights.

Since we do not have the ground-truth supervision, the generated pseudo labels are prone to be noisy. To deal with this issue, inspired by bayesian deep learning [18], we propose an uncertainty-aware learning module. To be specific, we add an uncertainty prediction module to estimate the uncertainty $[\alpha_1^*, \alpha_2^*, ..., \alpha_N^*](* \in \{R, F\})$ about the pseudo labels. The uncertainty prediction module is removed after training, and it does not bring any extra computations during testing. For a segment with the attention weight λ, pseudo label $\tilde{\lambda}$, and uncertainty α, the noise-robust loss is designed as

$$\mathcal{L}(\lambda, \tilde{\lambda}, \alpha) = e^{-\alpha} D(\lambda, \tilde{\lambda}) + \tau\alpha, \tag{8}$$

where τ is a hyper-parameter to avoid predicting high uncertainties for all pseudo labels, D measures the distance between λ and $\tilde{\lambda}$. To understand why this loss can mitigate the noise in the pseudo labels, we take a theoretical analysis into Equation (8). To minimize Equation (8) under the constrain $\alpha \geq 0$, we can derive a analytical solution to α as

$$\alpha = \begin{cases} \log(\frac{D(\lambda,\tilde{\lambda})}{\tau}), & \text{if } D(\lambda, \tilde{\lambda}) > \tau \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

This means that when $D(\lambda, \tilde{\lambda})$ is larger than τ, the pseudo label $\tilde{\lambda}$ is suspected to be noisy. In this case, $D(\lambda, \tilde{\lambda})$

is multiplied with $e^{-\alpha} < 1$, thus the affection of $\tilde{\lambda}$ on the model training will be decreased. This idea is wildly acknowledged in the area of learning from noisy labels, and the losses of samples with noisy labels are larger than those of clean samples because clean samples quickly get fit at that beginning [12, 15]. However, directly using the analytical solution ignores the input, which treats segments with noisy labels and hard segments with correct labels in the same way. But in our case, pseudo labels are generated by the teacher model in another modality, label noise is not random and usually depends on the content of input. Thus instead of using the analytical solution, we use an uncertainty prediction module to achieve our goal, which takes the input feature into consideration and may discover the common characteristics of segments that may have noisy labels. With the noise-robust loss, the pseudo label losses for RGB and FLOW streams are calculated as

$$\mathcal{L}_{ps}^{R} = \frac{1}{N}\sum_{i=1}^{N} e^{-\alpha_i^R} D(\lambda_i^R, \mathcal{G}(\Lambda^{FT})_i) + \tau\alpha_i^R, \quad (10)$$

$$\mathcal{L}_{ps}^{F} = \frac{1}{N}\sum_{i=1}^{N} e^{-\alpha_i^F} D(\lambda_i^F, \mathcal{G}(\Lambda^{RT})_i) + \tau\alpha_i^F. \quad (11)$$

There are two important factors in the uncertainty aware learning module: (1) The choice of τ. As shown in Equation (9), τ can be regarded as a soft threshold for considering pseudo labels to be noisy, and it should be set according to the choice of distance measure function $\mathcal{D}$. In this paper, we adopt the mean square error and set it to be 0.1. (2) The weight initialization of the uncertainty prediction module. In the beginning, we do not know which segment has noisy label, and all segments should be assigned low uncertainties. To achieve this goal, we initialize the weights of the uncertainty prediction module with a random Gaussian initialization, and the standard deviation and mean are set to 0.0001 and 0 respectively.

By combining the basic loss and the pseudo label loss, the models in RGB and FLOW streams are collaboratively trained with

$$\mathcal{L}^* = \mathcal{L}_{base}^* + w(t)\mathcal{L}_{ps}^*, \forall * \in \{R, F\}. \quad (12)$$

And $w(t)$ is set as a time-varying parameter as in Equation (13), and it is gradually increased to 1 during the network training, since the pseudo label supervision is less reliable in the early training stage.

$$w(t) = \begin{cases} e^{-5*(1-2t/MaxIter)}, & \text{if } t <= MaxIter/2 \\ 1, & \text{otherwise.} \end{cases} \quad (13)$$

For model updating, in the t-th iteration, θ^R and θ^F are updated with the loss backward propagation

$$\theta_t^* = \theta_{t-1}^* - \eta\frac{\partial \mathcal{L}^*}{\partial \theta_{t-1}^*}, \forall * \in \{R, F\}, \quad (14)$$

where η is the learning rate. For the teacher models, θ^{RT} and θ^{FT} are updated with exponential moving average as

$$\theta_t^{*T} = \gamma\theta_{t-1}^{*T} + (1-\gamma)\theta_t^*, \forall * \in \{R, F\}, \quad (15)$$

where γ is a hyper-parameter and is set to be 0.999.

3.4. Discussions

Pseudo labels are also used in TSCN [53] and EM-MIL [28] for attention weight learning, and the differences are discussed as follows: (1) TSCN is an off-line method. The RGB and FLOW models are trained first, and the trained models are then used to generate pseudo labels on attention weights. Based on the pseudo labels, they train the RGB and FLOW models from start again. After retraining, pseudo labels are generated again. This procedure repeats until no performance gain is observed. While in our UGCT, the pseudo labels are used as the bridge to train RGB and FLOW models collaboratively. Besides, the pseudo labels are generated in an online manner by teacher models, which enables the models to be trained efficiently in one pass. (2) In EM-MIL, the class-agnostic attention weight and class activation sequence provide pseudo labels for each other in an iterative way, which is significantly different from our method. Besides, both two methods ignore the noise in the pseudo labels, while we propose an uncertainty aware learning module to deal with this issue.

4. Experiment

4.1. Datasets and Evaluation Metrics

Datasets. We evaluate our method on two benchmark datasets including THUMOS14 [14] and ActivityNet [5]. Here, ActivityNet consists of ActivityNet1.2 and ActivityNet1.3. In **THUMOS14** dataset, there are 200 validation videos and 213 test videos that are annotated with temporal action boundaries belonging to 20 categories. And there are 15 action clips per video in average on this dataset, which makes it very challenging. Follow the protocol in previous work [31, 44, 47, 35, 30, 56, 25, 38], the validation set is used for training and the test set is used for evaluation in this work. **ActivityNet1.2** dataset contains 4819 training videos, 2383 validation videos and 2480 testing videos belonging to 100 action categories. And **ActivityNet1.3** consists of 10024 training videos, 4926 validation videos and 5044 testing videos belonging to 200 action categories. Since the annotations for the test set are not released, we train our model on the training set and evaluate it on the validation set as in [31, 44, 47, 35, 30, 56, 25, 38].

4.2. Implementation Details

In this work, we use the I3D network [6] for feature extraction, and the output feature dimension d is 1024. And TV-L1 optical flow [52] is used to generate optical frames for the FLOW stream. To verify the effectiveness of our training strategy, we re-implement three attention based methods in the same experiment setting. The model is trained with the mini-batch size of 32 using the Adam [19]

Table 1. Detection 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, and (ours) represents our re-implementation.

Supervision	Method	Feature	mAP@IoU							
			0.1	0.2	0.3	0.4	0.5	0.6	0.7	Average(0.1:0.5)
Fully	S-CNN [39], CVPR2016	-	47.7	43.5	36.3	28.7	19.0	-	-	35.0
	CDC [37], CVPR2017	-	-	-	40.1	29.4	23.3	13.1	7.9	-
	R-C3D [46], ICCV2017	-	54.5	51.5	44.8	35.6	28.9	-	-	43.1
	SSN [55], ICCV2017	-	66.0	59.4	51.9	41.0	29.8	-	-	49.6
	TAL-Net [7], CVPR2018	-	59.8	57.1	53.2	48.5	42.8	33.8	20.8	52.3
	GTAN [27], CVPR2019	-	69.1	63.7	57.8	47.2	38.8	-	-	55.3
weak$^+$	STAR [47], AAAI2019	I3D	68.8	60.0	48.7	34.7	23.0	-	-	47.0
	3C-Net [30], ICCV2019	I3D	59.1	53.5	44.2	34.1	26.6	-	8.1	43.5
weak	UntrimmedNet [44], CVPR2017	-	44.4	37.7	28.2	21.1	13.7	-	-	29.0
	Hide-and-Seek [40], ICCV2017	-	36.4	27.8	19.5	12.7	6.8	-	-	20.6
	Zhong et al. [56], MM2018	-	45.8	39.0	31.1	22.5	15.9	-	-	30.9
	AutoLoc [38], ECCV2018	UNT	-	-	35.8	29.0	21.2	13.4	5.8	-
	Clean-Net [26], ICCV2019	UNT	-	-	37.0	30.9	23.9	13.9	7.1	-
	STPN [31], CVPR2018	I3D	52.0	44.7	35.5	25.8	16.9	9.9	4.3	35.0
	WTALC [35], ECCV2018	I3D	55.2	49.6	40.1	31.1	22.8	-	7.6	39.8
	CMCS [25], CVPR2019	I3D	57.4	50.8	41.2	32.1	23.1	15.0	7.0	40.9
	ASSG [54], MM2019	I3D	55.6	49.5	41.1	31.5	20.9	13.7	5.9	39.7
	TSM [50], ICCV2019	I3D	-	-	39.5	31.9	24.5	13.8	7.1	-
	WSAL-BM [32], ICCV2019	I3D	60.4	56.0	46.6	37.5	26.8	19.6	9.0	45.5
	DGAM [36], CVPR2020	I3D	60.0	54.2	46.8	38.2	28.8	19.8	11.4	45.6
	TCAM [10], CVPR2020	I3D	-	-	46.9	38.9	30.1	19.8	10.4	-
	Bas-Net [21], AAAI2020	I3D	58.2	52.3	44.6	36.0	27.0	18.6	10.4	43.6
	RPN [13], AAAI2020	I3D	62.3	57.0	48.2	37.2	27.9	16.7	8.1	46.5
	A2CL-PT [29], ECCV2020	I3D	61.2	56.1	48.1	39.0	30.1	19.2	10.6	46.9
	EM-MIL [28], ECCV2020	I3D	59.1	52.7	45.5	36.8	30.5	22.7	16.4	44.9
	TSCN [53], ECCV2020	I3D	63.4	57.6	47.8	37.7	28.7	19.4	10.2	47.0
	STPN [31] (Ours)	I3D	59.6	54.4	45.6	34.8	21.8	11.7	4.1	43.2
	STPN [31] with UGCT	I3D	**67.0**	**61.7**	**55.0**	**44.1**	**32.4**	**19.6**	**8.9**	**52.2**
	WSAL-BM [32] (Ours)	I3D	65.8	59.4	51.1	40.5	30.3	19.1	8.7	49.4
	WSAL-BM [32] with UGCT	I3D	**69.2**	**62.9**	**55.5**	**46.5**	**35.9**	**23.8**	**11.4**	**54.0**
	TSCN [53] (Ours)	I3D	65.6	60.0	51.0	39.5	29.0	17.3	7.9	49.0
	TSCN [53] with UGCT	I3D	**67.5**	**62.1**	**55.3**	**45.2**	**33.3**	**20.7**	**9.5**	**52.7**

optimizer, the learning rate is set to be 10^{-4}, and the number of iterations is set to be $6K$ for THUMOS14 and $30K$ for ActivityNet. It is worth noting that the performance of our re-implementation is much better than the original papers except for TSCN on ActivityNet dataset [2], and the results are listed in Table 1, Table 2 and Table3.

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

Results on THUMOS14 dataset. In Table 1, we compare our method with state-of-the-art weakly supervised methods and several fully supervised methods on THUMOS14 dataset. By combining three important tricks used in previous methods (random dropout [25], all segments in one video are used during training [32], context information aggregation [10]), our re-implemented WSAL-BM [32] and TSCN [53] have already achieved a comparable performance with state-of-the-art weakly supervised methods. When further applying the proposed UGCT on these two methods, the performances of both methods are significantly improved (4.6% and 3.7% on average mAP). Without bells and whistles, WSAL-BM [32] with UGCT sets a new state-of-the-art performance, surpassing the previous best method by 7% on average mAP. And on STPN, we have the most impressive result. Although our re-implemented result already outperforms the original paper with a large margin, e.g., 43.2% vs 35.0% on average mAP, it is still far behind state-of-the-art performance. However, with the proposed training strategy, the average mAP is boosted to 52.2% with 9.0% performance gain, which surpasses the previous best performance (TSCN [53] 47.0%) by 5.2%. When compared with fully supervised methods, although the performance gap in high IoU thresholds is still large, we have a competitive performance in low IoU thresholds.

Results on ActivityNet dataset. Different from THUMOS14, following the standard protocol in ActivityNet [5], we report the mAP score at IoU=$0.5, 0.75, 0.95$ and the average mAP for IoU=0.5:0.05:0.95. Results on ActivityNet1.2 and ActivityNet1.3 datasets are shown in Table 2 and Table 3. On ActivityNet1.2 dataset, the proposed UGCT strategy can improve STPN [31] by 2.6%, WSAL-BM [32] by 2.1% and TSCN [53] by 7.4% in terms of average mAP. And without bells and whistles, STPN with UGCT outperforms the previous best methods by 1.2% and 1.3% in average mAP on ActivityNet1.2 and ActivityNet1.3 respectively. It is worth noting that among these

[2]The performance of the original paper is achieved by off-line pseudo label supervision, which is not included in our re-implementation.

Table 2. Detection performance comparison with state-of-the-art methods on the ActivityNet1.2 validation set, where (Ours) represents our re-implementation.

Methods	mAP@IoU			
	0.5	0.75	0.95	Average
UntrimmedNet [44]	7.4	3.9	1.2	3.6
WTALC [35]	37.0	14.6	-	18.0
AutoLoc [38]	27.3	17.5	6.8	16.0
Zhong et al. [56]	27.3	14.7	2.9	15.6
CMCS [25]	36.8	22.0	5.6	22.4
TSM [50]	28.3	17.0	3.5	-
Clean-Net [26]	37.1	20.3	5.0	21.6
DGAM [36]	41.0	23.5	5.3	24.4
TCAM [10]	40.0	25.0	4.6	24.6
Bas-Net [21]	38.5	24.2	5.6	24.3
RPN [13]	37.6	23.9	5.4	23.3
TSCN [53]	37.6	23.7	5.7	23.6
EM-MIL [28]	37.4	23.1	2.0	20.3
STPN (Ours)	39.6	22.5	4.3	23.2
STPN with UGCT	**41.8**	**25.3**	**5.9**	**25.8**
WSAL-BM (Ours)	38.5	21.5	4.4	22.6
WSAL-BM with UGCT	**40.9**	**24.3**	**5.4**	**24.7**
TSCN (Ours)	30.0	15.9	3.4	16.9
TSCN with UGCT	**40.0**	**23.6**	**5.6**	**24.3**

three methods, STPN has the worst performance on THUMOS14 dataset but the best performance on ActivityNet dataset. This result is not beyond expectation since THUMOS14 and ActivityNet datasets have different characteristics. Each video in THUMOS14 dataset contains multiple action clips with short duration and small interval among adjacent action clips, and it is of vital importance to recognize the background among clips. While each video in ActivityNet dataset usually contains only one action clip with a long duration, and it is important to detect the long action clip as a whole. Due to different designs in the attention loss, we find that STPN tends to recognize background as foreground, so adjacent actions clips in THUMOS14 tend to be recognized as one action clip. While both TSCN and WSAL-BM tend to recognize foreground as background, a long action clip in ActivityNet dataset is usually cut into several small clips. As a result, STPN is more suitable for ActivityNet dataset.

4.4. Ablation Studies

To look deeper into the proposed training strategy, we perform a series of ablation studies on the THUMOS14 dataset, and detailed results and analysis are as follows.

4.4.1 Effectiveness of Each Design.

As shown in Table 4, the performances of all three methods are significantly improved with the help of collaborative training. The mAP@IoU=0.5 is promoted from 21.8 to 32.4 for a weaker baseline STPN [31], with 10.6% performance gain. And for a stronger baseline (WSAL-BM [32]), the mAP@IoU=0.5 is also promoted from 30.3 to 34.4. These results verify the effectiveness of the collaborative training, this is because the collaborative training can help the model

Table 3. Detection performance comparison with state-of-the-art methods on the ActivityNet1.3 validation set, where (Ours) represents our re-implementation.

Methods	mAP@IoU			
	0.5	0.75	0.95	Average
STPN [31]	29.3	16.9	2.6	16.3
CMCS [25]	34.0	20.9	5.7	21.2
ASSG [54]	32.3	20.1	4.0	18.8
WSAL-BM [32]	36.4	19.2	2.9	19.5
STAR [47]	31.1	18.8	4.7	18.2
TSM [50]	30.3	19.0	4.5	-
Bas-Net [21]	34.5	22.5	4.9	22.2
TSCN [53]	35.3	21.4	5.3	21.7
A2CL-PT [29]	36.8	22.0	5.2	22.5
STPN (Ours)	38.0	21.5	5.7	22.7
STPN with UGCT	**39.1**	**22.4**	**5.8**	**23.8**
WSAL-BM (Ours)	36.9	20.7	4.2	20.6
WSAL-BM with UGCT	**39.0**	**21.4**	**5.1**	**23.0**
TSCN (Ours)	29.1	14.2	3.3	15.4
TSCN with UGCT	**38.1**	**21.2**	**5.4**	**22.8**

Table 4. Ablation studies about the proposed UGCT on the THUMOS14 test set. Results indicate each design is necessary.

	Collaborative Training	uncertainty Guidance	mAP@IoU=0.5
STPN	✗	✗	21.8
	✓	✗	31.4
	✓	✓	**32.4**
WSAL-BM	✗	✗	30.3
	✓	✗	34.4
	✓	✓	**35.9**
TSCN	✗	✗	29.0
	✓	✗	32.2
	✓	✓	**33.3**

to suppress false positives and false negatives by referring to the information in another modality. To move a step further, when taking the label noise into consideration, we get our Uncertainty Guided Collaborative Training strategy. As shown in Table 4, the performance of three methods can be consistently improved, which indicates the effectiveness of our uncertainty aware learning module.

4.4.2 Ablations on Pseudo Label Generation

Teacher models are necessary. To verify the effectiveness of our teacher models in pseudo label generation, we use the RGB and FLOW models to replace the corresponding teacher models, and the results are shown in Table 5. Without teacher models for pseudo label generation, all three methods suffer from significant performance drop. The results indicate that the teacher model is more reliable for pseudo label generation, and we owe this property to the specific updating strategy of teacher models. When the training process is converged, the teacher model and student models have a similar performance, but the teacher model can make more reliable prediction when the training process is not converged.

Different ways for pseudo label generation. In this section, we compare three different ways to pseudo label generation (Self-supervision, Mean-supervision and Cross-

Table 5. Performance comparison on the THUMOS14 test set with and w/o teacher models for pseudo label generation.

Methods	Teacher Model	mAP@IoU=0.5
STPN	✓	**32.4**
	✗	26.6
WSAL-BM	✓	**35.9**
	✗	31.2
TSCN	✓	**33.3**
	✗	32.0

Table 6. Different ways for pseudo label generation.

	Self	Mean	Cross
$\tilde{\Lambda}^R =$	$\mathcal{G}(\Lambda^{RT})$	$\mathcal{G}((\Lambda^{RT}+\Lambda^{FT})/2)$	$\mathcal{G}(\Lambda^{FT})$
$\tilde{\Lambda}^F =$	$\mathcal{G}(\Lambda^{FT})$	$\mathcal{G}((\Lambda^{RT}+\Lambda^{FT})/2)$	$\mathcal{G}(\Lambda^{RT})$

Table 7. Performance comparison on the THUMOS14 test set with different ways for pseudo label generation.

Methods	Pseudo generation	0.3	0.5	0.7
STPN	✗	45.6	21.8	4.1
	Self	51.8	28.6	7.5
	Mean	52.5	29.5	7.9
	Cross	**54.5**	**31.4**	**8.4**
WSAL-BM	✗	51.1	30.3	8.7
	Self	53.2	**34.6**	9.4
	Mean	53.7	33.0	**10.6**
	Cross	**54.7**	34.4	9.9
TSCN	✗	51.0	29.0	7.9
	Self	53.2	30.8	8.5
	Mean	53.2	31.6	8.9
	Cross	**54.2**	**32.2**	**9.3**

supervision), and the details are shown in Table 6. In the Self-supervision, the RGB and FLOW streams provide pseudo labels for themselves and are trained independently, and the Cross-supervision represents the way we adopt in this paper. From the results in Table 7, all three ways can improve the performance, which indicates that the pseudo label supervision is of great importance. Note that when the RGB/FLOW streams are trained independently, the Self-supervision gets the worst performance. When the RGB/FLOW streams work collaboratively for pseudo label generation, the Mean-supervision and Cross-supervision can achieve much better performance.

Hard pseudo labels vs soft pseudo labels. In our method, we use binarization function $\mathcal{G}$ to generate $\{0, 1\}$-value hard pseudo labels. It is also feasible to directly use soft labels, as shown in Table 8, and the results indicate that hard pseudo labels work better than soft pseudo labels. Note that for STPN and WSAL-BM, hard pseudo labels can yield more significant performance gain, and this is due to that the hard pseudo labels can force the attention weight to act like a binary selection. While in TSCN, the attention loss is already designed for this purpose, thus the advantage of hard pseudo labels is less significant on this method.

4.4.3 Ablations on Uncertainty Estimation

In the proposed training strategy, we add an uncertainty prediction module to mitigate the noise in the pseudo labels, and we denote this method as Uncertainty-P. And the

Table 8. Performance comparison between hard and soft pseudo labels on the THUMOS14 test set.

Methods	Pseudo label	0.3	0.5	0.7
STPN	soft	53.6	25.3	5.3
	hard	**54.5**	**31.4**	**8.4**
WSAL-BM	soft	54.1	32.4	9.4
	hard	**54.7**	**34.4**	**9.9**
TSCN	soft	53.7	31.6	8.3
	hard	**54.2**	**32.2**	**9.3**

Table 9. Results on the THUMOS14 test set with different ways for uncertainty estimation.

Methods	mAP@IoU=0.5		
	Uncertainty-P	Uncertainty-C	Uncertainty-A
STPN	**32.4**	32.1	31.6
WSAL-BM	**35.9**	35.8	34.3
TSCN	**33.3**	31.5	32.5

analytical solution introduced in Section 3.3 is denoted as Uncertainty-A. Besides, we also design a consistency based method for label uncertainty estimation, which is denoted as Uncertainty-C. To be specific, we feed the input to the teacher network twice with different random noise and get two pseudo labels for each segment, segments with inconsistent two pseudo labels may encounter high uncertainty, thus they are ignored during training. Note that the consistency based method need to feed-forward input twice, it brings more computation cost. Results are shown in Table 9, which indicates that the prediction based method can achieve better performance with less computation cost.

5. Conclusion

In this paper, we propose a simple yet effectively Uncertainty Guided Collaboratively Training strategy for attention based weakly supervised temporal action detection methods. An online pseudo label generation module is designed to provide reliable pseudo labels for attention weight learning, and an uncertainty aware learning module is further designed to deal with the label noise. We conduct experiments on two benchmark datasets with three attention based methods, and experimental results indicate that the proposed method can significantly improve the performance of these methods.

6. Acknowledgement

This work was partially supported by the National Key Research and Development Program under Grant No. 2018YFB0804204, National Defense Basic Scientific Research Program (JCKY2020903B002), Strategic Priority Research Program of Chinese Academy of Sciences (No.XDC02050500), 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] Piotr Bojanowski, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Finding actors and actions in movies. In *Proceedings of the IEEE international conference on computer vision*, pages 2280–2287, 2013. 1

[2] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *European Conference on Computer Vision*, pages 628–643. Springer, 2014. 1

[3] 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. 1, 2

[4] 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. 2

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 5, 6

[6] 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. 3, 5

[7] 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. 2, 6

[8] 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. 1

[9] Chuang Gan, Chen Sun, Lixin Duan, and Boqing Gong. Webly-supervised video recognition by mutually voting for relevant web images and web video frames. In *European Conference on Computer Vision*, pages 849–866. Springer, 2016. 1

[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. 3, 6, 7

[11] De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *European Conference on Computer Vision*, pages 137–153. Springer, 2016. 1

[12] Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3326–3334, 2019. 5

[13] Linjiang Huang, Yan Huang, Wanli Ouyang, Liang Wang, et al. Relational prototypical network for weakly supervised temporal action localization. 2020. 3, 6, 7

[14] 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. 1, 5

[15] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313, 2018. 5

[16] Yifan Jiao, Zhetao Li, Shucheng Huang, Xiaoshan Yang, Bin Liu, and Tianzhu Zhang. Three-dimensional attention-based deep ranking model for video highlight detection. *IEEE Transactions on Multimedia*, 20(10):2693–2705, 2018. 1

[17] Cheng-Bin Jin, Shengzhe Li, and Hakil Kim. Real-time action detection in video surveillance using sub-action descriptor with multi-cnn. *arXiv preprint arXiv:1710.03383*, 2017. 1

[18] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017. 4

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[20] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 1

[21] Pilhyeon Lee, Youngjung Uh, and Hyeran Byun. Background suppression network for weakly-supervised temporal action localization. In *AAAI*, pages 11320–11327, 2020. 1, 3, 6, 7

[22] 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. 1

[23] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*, 2018. 1

[24] 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. 1, 2

[25] 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. 3, 5, 6, 7

[26] 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. 6, 7

[27] 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. 1, 2, 6

[28] 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. 5, 6, 7

[29] Kyle Min and Jason J Corso. Adversarial background-aware loss for weakly-supervised temporal activity localization. *arXiv preprint arXiv:2007.06643*, 2020. 3, 6, 7

[30] Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao. 3c-net: Category count and center loss for weakly-supervised action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8679–8687, 2019. 1, 3, 5, 6

[31] 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. 1, 2, 3, 5, 6, 7

[32] Phuc Xuan Nguyen, Deva Ramanan, and Charless C Fowlkes. Weakly-supervised action localization with background modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5502–5511, 2019. 1, 2, 3, 6, 7

[33] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. Action and event recognition with fisher vectors on a compact feature set. In *Proceedings of the IEEE international conference on computer vision*, pages 1817–1824, 2013. 1

[34] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. Action and event recognition with fisher vectors on a compact feature set. In *Proceedings of the IEEE international conference on computer vision*, pages 1817–1824, 2013. 2

[35] 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. 1, 3, 5, 6, 7

[36] 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. 1, 3, 6, 7

[37] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5734–5743, 2017. 1, 6

[38] 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. 3, 5, 6, 7

[39] 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. 2, 6

[40] 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. 1, 3, 6

[41] Kevin Tang, Bangpeng Yao, Li Fei-Fei, and Daphne Koller. Combining the right features for complex event recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2696–2703, 2013. 1

[42] Kevin Tang, Bangpeng Yao, Li Fei-Fei, and Daphne Koller. Combining the right features for complex event recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2696–2703, 2013. 2

[43] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017. 2

[44] 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. 2, 5, 6, 7

[45] Zhang Wang Luo andTianzhu, Yang Wenfei, Tao Mei, Liu Jingen, Feng Wu, and Zhang Yongdong. Action unit memory network for weakly supervised temporal action localization. *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1

[46] 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. 2, 6

[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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9070–9078, 2019. 3, 5, 6, 7

[48] Wenfei Yang, Tianzhu Zhang, Yongdong Zhanga, and Feng Wu. Local correspondence network for weakly supervised temporal sentence grounding. *IEEE Transactions on Image Processing*, 2021. 1

[49] 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. 2

[50] 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. 6, 7

[51] 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. 2

[52] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007. 5

[53] 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. 1, 3, 5, 6, 7

[54] 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. 1, 3, 6, 7

[55] 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. 2, 6

[56] 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. 1, 3, 5, 6, 7