Weakly-supervised Temporal Action Localization by Uncertainty Modeling

Pilhyeon Lee[1*] **Jinglu Wang**[2] **Yan Lu**[2] **Hyeran Byun**[1,3†]
[1] Department of Computer Science, Yonsei University
[2] Microsoft Research Asia [3] Graduate School of AI, Yonsei University
{lph1114, hrbyun}@yonsei.ac.kr {jinglwa, yanlu}@microsoft.com

Abstract

Weakly-supervised temporal action localization aims to learn detecting temporal intervals of action classes with only video-level labels. To this end, it is crucial to separate frames of action classes from the background frames (*i.e.*, frames not belonging to any action classes). In this paper, we present a new perspective on background frames where they are modeled as out-of-distribution samples regarding their inconsistency. Then, background frames can be detected by estimating the probability of each frame being out-of-distribution, known as *uncertainty*, but it is infeasible to directly learn uncertainty without frame-level labels. To realize the uncertainty learning in the weakly-supervised setting, we leverage the multiple instance learning formulation. Moreover, we further introduce a background entropy loss to better discriminate background frames by encouraging their in-distribution (action) probabilities to be uniformly distributed over all action classes. Experimental results show that our uncertainty modeling is effective at alleviating the interference of background frames and brings a large performance gain without bells and whistles. We demonstrate that our model significantly outperforms state-of-the-art methods on the benchmarks, THUMOS'14 and ActivityNet (1.2 & 1.3). Our code is available at https://github.com/Pilhyeon/WTAL-Uncertainty-Modeling.

1 Introduction

Temporal action localization (TAL) is a very challenging problem of finding and classifying action intervals in untrimmed videos, which plays an important role in video understanding and analysis. To tackle the problem, many works have been done in the fully-supervised manner and achieved impressive progress (Zeng et al. 2019; Lin et al. 2019, 2020; Xu et al. 2020). However, they suffer from the extremely high cost of acquiring precise annotations, *i.e.*, labeling the start and end timestamps of each action instance.

To relieve the high-cost issue and enlarge the scalability, researchers direct their attention to the same task with weak supervision, namely, weakly-supervised temporal action localization (WTAL). Among the various levels of weak supervision, thanks to the cheap cost, video-level action labels are widely employed (Wang et al. 2017). In this setting, each video is labeled as positive for action classes if it contains corresponding action frames and as negative otherwise. Note that a video may have multiple action classes as its label.



(a) **dynamic** background frames in a *SoccerPenalty* video

(b) **inconsistent** background frames in a *GolfSwing* video

Figure 1: Challenges in modeling background: dynamism and inconsistency. (a) The background frames in red boxes showing soccer players celebrating are very dynamic. (b) The background frames are inconsistent. The frames in green boxes show empty scenes with subtitles and the one in a blue box shows a golfer preparing to shoot. These two types do not have consistent appearances and semantics.

Existing approaches commonly cast WTAL as frame-wise classification, and adopt attention mechanism (Nguyen et al. 2018) or multiple instance learning (Paul, Roy, and Roy-Chowdhury 2018) to learn from video-level labels. Nonetheless, they still show highly inferior performances when compared to fully-supervised counterparts. According to the literature (Xu et al. 2019), the performance degradation mainly comes from the false alarms of background frames, since video-level labels do not have any clue for background. To bridge the gap, there appear several studies attempting explicit background modeling in the weakly-supervised manner. Liu *et al*. (Liu, Jiang, and Wang 2019) merges static frames to synthesize pseudo background videos, but they overlook dynamic background frames (Fig. 1a). Meanwhile, some work (Nguyen, Ramanan, and Fowlkes 2019; Lee, Uh, and Byun 2020) tries to classify background frames as a separate class. However, it is undesirable to force all background frames to belong to one specific class, as they do not share any common semantics (Fig. 1b).

[*] Work done during his internship at Microsoft Research Asia.
[†] Corresponding author

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

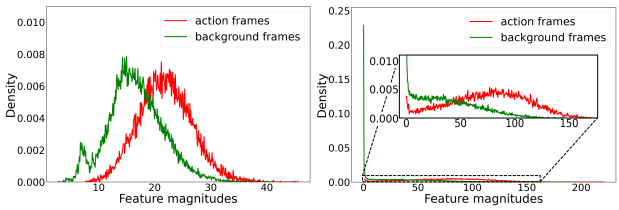
In this paper, we embrace the observation on inconsistency of background frames, and propose to formulate them as out-of-distribution samples (Liang, Li, and Srikant 2018; Dhamija, Günther, and Boult 2018). They can be identified by estimating the probability of each sample coming from out-of-distribution, also known as *uncertainty* (Bendale and Boult 2016; Lakshminarayanan, Pritzel, and Blundell 2017). To model uncertainty, we propose to utilize the magnitude of an embedded feature vector. Generally, features of action frames have larger magnitudes compared to ones from background frames, as shown in Fig. 2a. This is because action frames need to produce high logits for ground-truth action classes. Although the features magnitudes show the correlation with the discrimination between background and action frames, directly using them for discrimination is insufficient since the distributions of action and background are close to each other. Therefore, to further encourage the discrepancy in feature magnitudes, we propose to separate the distributions by enlarging magnitudes of action features and reducing those of background features close to zero (Fig. 2b).

In order to learn uncertainty only with video-level supervision, we leverage the formulation of multiple instance learning (Maron and Lozano-Pérez 1998; Zhou 2004), where a model is trained with a bag (*i.e.*, untrimmed video) instead of instances (*i.e.*, frames). Specifically, from each untrimmed video, we select top-k and bottom-k frames based on the feature magnitude and consider them as pseudo action and background frames, respectively. Thereafter, we design an uncertainty modeling loss to separate their magnitudes, through which our model is able to indirectly model uncertainty without frame-level labels and provides better separation between action and background frames. Moreover, we introduce a background entropy loss to force pseudo background frames to have uniform probability distribution over action classes. This prevents them from leaning toward a certain action class and helps to reject them by maximizing the entropy of their action class distribution. To validate the effectiveness of our method, we perform experiments on two standard benchmarks, THUMOS'14 and ActivityNet. By jointly optimizing the proposed losses along with a general action classification loss, our model successfully distinguishes action frames from background frames. Furthermore, our method achieves the new state-of-the-art performances on the both benchmarks.

Our contributions are three-fold: 1) We propose to formulate background frames as out-of-distribution samples, overcoming the difficulty in modeling background due to their inconsistency. 2) We design a new framework for weakly-supervised action localization, where uncertainty is modeled and learned only with video-level labels via multiple instance learning. 3) We further encourage separation between action and background with a loss maximizing the entropy of action probability distribution from background frames.

2 Related Work

Fully-supervised action localization. The goal of temporal action localization is to find temporal intervals of action instances from long untrimmed videos and classify them.



(a) Original features (b) Separated features

Figure 2: Normalized histograms of feature magnitudes. (a) The magnitudes of original features show correlation with action/background, but the centers of two distributions are close. (b) The separated feature magnitudes by our method have better discrimination of action and background.

For the task, many approaches depend on accurate temporal annotations for each training video. Most of them utilize the two-stage approach, *i.e.*, they first generate proposals, and then classify them. To generate proposals, earlier methods adopt the sliding window technique (Shou, Wang, and Chang 2016; Yuan et al. 2016; Shou et al. 2017; Yang et al. 2018; Xiong et al. 2017; Chao et al. 2018), while recent models predict the start and end frames of action (Lin et al. 2018, 2019, 2020). Meanwhile, there appear some attempts to leverage graph structural information (Zeng et al. 2019; Xu et al. 2020). Moreover, there are also a gaussian modeling of each action instance (Long et al. 2019) and an efficient method without proposal generation step (Alwassel, Caba Heilbron, and Ghanem 2018).

Weakly-supervised action localization. Recently, many attempts have been made to solve temporal action localization with weak supervision, mostly video-level labels. (Wang et al. 2017) first tackle the problem by selecting relevant segments on soft and hard ways. (Nguyen et al. 2018) propose a sparsity regularization, while (Singh and Lee 2017) and (Yuan et al. 2019) extend small discriminative parts. (Paul, Roy, and Roy-Chowdhury 2018), (Narayan et al. 2019) and (Min and Corso 2020) employ deep metric learning to force features from the same action to get closer to themselves than those from different classes. (Shi et al. 2020) and (Luo et al. 2020) learn attention weights using a variational auto-encoder and an expectation-maximization strategy, respectively. (Zhai et al. 2020) pursue the consensus between different modalities. Meanwhile, (Shou et al. 2018) and (Liu et al. 2019) attempt to regress the intervals of action instances, instead of performing hard thresholding. Recently, (Ma et al. 2020) propose to exploit the intermediate level of supervision (*i.e.*, single frame supervision).

Apart from the methods above, Some work (Liu, Jiang, and Wang 2019; Nguyen, Ramanan, and Fowlkes 2019; Lee, Uh, and Byun 2020) seeks to explicitly model background. However, as aforementioned, they have innate limitations in that background frames could be dynamic and inconsistent. In contrast, we consider background as out-of-distribution and propose to learn uncertainty as well as action class scores. In Sec. 4, the efficacy of our approach is verified.

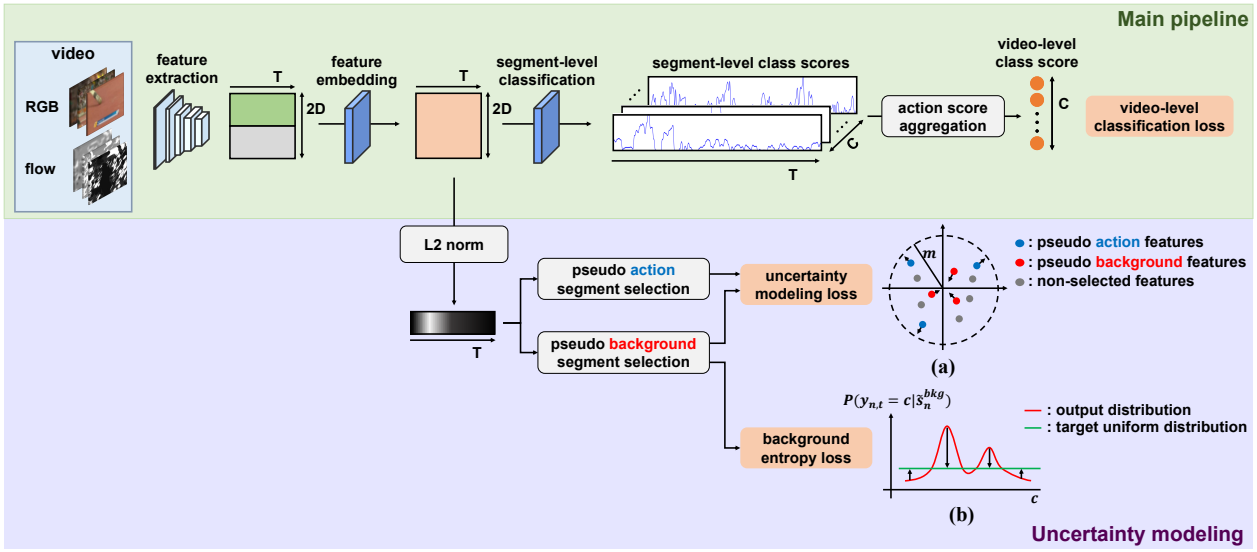


Figure 3: Overview of the proposed method. The main pipeline serves the standard process for weakly-supervised action localization. We discriminate background frames from action frames by modeling *uncertainty*, *i.e.*, probability of being out-of-distribution. In the uncertainty modeling part, pseudo action/background segments are selected based on features magnitudes, which are in turn used to derive two proposed losses for background discrimination: (a) uncertainty modeling loss which enlarges and reduces the feature magnitudes of the pseudo action and background segments respectively; and (b) background entropy loss forcing the pseudo background segments to have uniform action probability distribution.

3 Method

In this section, we first set up the baseline network for weakly-supervised temporal action localization (Sec. 3.1). Thereafter, we cast the problem of identifying background frames as out-of-distribution detection and tackle it by modeling uncertainty (Sec. 3.2). Lastly, the objective functions for training our model (Sec. 3.3) and how the inference is performed (Sec. 3.4) are elaborated. The overview of our method is illustrated in Fig. 3.

3.1 Main Pipeline

Feature extraction. Due to the memory limit, we split each video into multi-frame segments, *i.e.*, $v_n = \{s_{n,l}\}_{l=1}^{L_n}$, where L_n denotes the number of segments in v_n. To handle the large variation in video lengths, a fixed number of T segments $\{\tilde{s}_{n,t}\}_{t=1}^{T}$ are sampled from each original video. From the sampled RGB and flow segments, spatio-temporal features $x_{n,t}^{\text{RGB}} \in \mathbb{R}^D$ and $x_{n,t}^{\text{flow}} \in \mathbb{R}^D$ are extracted by a pre-trained feature extractor, respectively. Next, we concatenate the RGB and flow features into the complete feature vectors $x_{n,t} \in \mathbb{R}^{2D}$, which are then stacked to build a feature map of length T, *i.e.*, $X_n = [x_{n,1}, ..., x_{n,T}] \in \mathbb{R}^{2D \times T}$.

Feature embedding. In order to embed the extracted features into the task-specific space, we employ a single 1-D convolutional layer followed by a ReLU function. Formally, $F_n = g_{\text{embed}}(X_n; \phi_{\text{embed}})$, where g_{embed} denotes the convolution operator with the activation and ϕ_{embed} is the trainable parameters of the layer. Concretely, the dimension of an embedded feature is the same as that of an input feature, *i.e.*, $F_n = [f_{n,1}, ..., f_{n,T}] \in \mathbb{R}^{2D \times T}$.

Segment-level classification. From the embedded features, we predict segment-level class scores, which are later used for action localization. Given the features F_n, the segment-wise class scores are derived by the action classifier, *i.e.*, $\mathcal{A}_n = g_{\text{cls}}(F_n; \phi_{\text{cls}})$, where g_{cls} represents the linear classifier with its parameters ϕ_{cls}. In specific, $\mathcal{A}_n \in \mathbb{R}^{C \times T}$, where C is the number of action classes.

Action score aggregation. Following the previous work (Paul, Roy, and Roy-Chowdhury 2018), we aggregate top k^{act} scores along all segments for each action class and average them to build a video-level raw class score:

$$a_c(v_n) = \frac{1}{k^{\text{act}}} \max_{\hat{\mathcal{A}}_{n;c} \subset \mathcal{A}_n[c,:]} \sum_{\forall a \in \hat{\mathcal{A}}_{n;c}} a, \tag{1}$$

where $\hat{\mathcal{A}}_{n;c}$ is a subset containing k^{act} action scores for the c-th class (*i.e.*, $\left|\hat{\mathcal{A}}_{n;c}\right| = k^{\text{act}}$), and k^{act} is a hyper-parameter controlling the number of the aggregated segments. Thereafter, we obtain the video-level action probability for each action class by applying the softmax function to the aggregated scores:

$$p_c(v_n) = \frac{\exp(a_c(v_n))}{\sum_{c'=1}^{C} \exp(a_{c'}(v_n))}, \tag{2}$$

where $p_c(v_n)$ represents the softmax score for the c-th action of v_n, which is guided by video-level weak labels.

3.2 Uncertainty Modeling

From the main pipeline, we can obtain the action probabilities for each segment, but the essential component for

action localization, *i.e.*, background discrimination, is not carefully considered. Regarding the unconstraint and inconsistency of background frames, we treat background as out-of-distribution (Hendrycks and Gimpel 2017) and model *uncertainty* (probability of being background) for WTAL.

Considering the probability that the segment $\tilde{s}_{n,t}$ belongs to the c-th action, it can be decomposed into two parts with a chain rule, *i.e.*, the in-distribution action probability and the uncertainty. Let $d \in \{0, 1\}$ denote the variable for the background discrimination, *i.e.*, $d = 1$ if the segment belongs to any action class, $d = 0$ otherwise (belongs to background). Then, the posterior probability for class c of $\tilde{s}_{n,t}$ is given by:

$$\begin{aligned} P(y_{n,t} = c|\tilde{s}_{n,t}) &= P(y_{n,t} = c, d = 1|\tilde{s}_{n,t}) \\ &= P(y_{n,t} = c|d = 1, \tilde{s}_{n,t})P(d = 1|\tilde{s}_{n,t}), \end{aligned} \tag{3}$$

where $y_{n,t}$ is the label of the segment, *i.e.*, if $\tilde{s}_{n,t}$ belongs to the c-th action class, then $y_{n,t} = c$, while $y_{n,t} = 0$ for background segments. We describe the single-label case for readability. Without loss of generality, this can be generalized to the multi-label.

Uncertainty formulation. In Eq. 3, the probability for in-distribution action classification, $P(y_{n,t} = c|d = 1, \tilde{s}_{n,t})$, is estimated with the softmax function as in general classification task. Additionally, it is necessary to model the probability that a segment belongs to any action class, *i.e.*, $P(d = 1|\tilde{s}_{n,t})$, to tackle the background discrimination problem. Observing that features of action frames generally have larger magnitudes than those of background frames (Fig. 2), we formulate uncertainty by employing magnitudes of feature vectors. Specifically, background features have small magnitudes close to 0, while action features have large ones. Then the probability that the t-th segment in the n-th video ($\tilde{s}_{n,t}$) is an action segment is defined by:

$$P(d = 1|\tilde{s}_{n,t}) = \frac{\min(m, \|f_{n,t}\|)}{m}, \tag{4}$$

where $f_{n,t}$ is the corresponding feature vector of $\tilde{s}_{n,t}$, $\|\cdot\|$ is a norm function (we use L-2 norm here), and m is the pre-defined maximum feature magnitude. From the formulation, it is ensured that the probability falls between 0 and 1, *i.e.*, $0 \leq P(d = 1|\tilde{s}_{n,t}) \leq 1$.

Uncertainty learning via multiple instance learning. To learn uncertainty only with video-level labels, we borrow the concept of multiple instance learning (Maron and Lozano-Pérez 1998), where a model is trained with a bag (video) instead of instances (segments). In this setting, considering that each untrimmed video contains both action and background frames, we select pseudo action/background segments representing the video. Specifically, the top k^{act} segments in terms of the feature magnitude are treated as the pseudo action segments $\{\tilde{s}_{n,i}|i \in \mathcal{S}^{\text{act}}\}$, where $\mathcal{S}^{\text{act}}$ indicates the set of pseudo action indices. Meanwhile, the bottom k^{bkg} segments are considered the pseudo background segments $\{\tilde{s}_{n,j}|j \in \mathcal{S}^{\text{bkg}}\}$, where $\mathcal{S}^{\text{bkg}}$ denotes the set of indices for pseudo background. k^{act} and k^{bkg} represent the number of segments sampled for action and background, respectively. Then the pseudo action/background segments serve as the representatives of the input untrimmed video, and they are used for training via multiple instance learning.

3.3 Training Objectives

Our model is jointly optimized with three losses: 1) video-level classification loss $\mathcal{L}_{\text{cls}}$ for action classification of each input video, 2) uncertainty modeling loss $\mathcal{L}_{\text{um}}$ for separating the magnitudes of action and background feature vectors, and 3) background entropy loss $\mathcal{L}_{\text{be}}$ which forces background segments to have uniform probability distribution for action classes. The overall loss function is as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \alpha\mathcal{L}_{\text{um}} + \beta\mathcal{L}_{\text{be}}, \tag{5}$$

where α and β are balancing hyper-parameters.

Video-level classification loss. For multi-label action classification, we use the binary cross entropy loss with normalized video-level labels (Wang et al. 2017) as follows:

$$\mathcal{L}_{\text{cls}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} -y_{n;c}\log p_c(v_n), \tag{6}$$

where $p_c(v_n)$ represents the video-level softmax score for the c-th class of the n-th video (Eq. 2), and $y_{n;c}$ is the normalized video-level label for the c-th class of the n-th video.

Uncertainty modeling loss. To learn uncertainty, we train the model to produce large feature magnitudes for pseudo action segments but ones with small ones for pseudo background segments, as illustrated in Fig. 3 (a). Formally, uncertainty modeling loss takes the form:

$$\mathcal{L}_{\text{um}} = \frac{1}{N}\sum_{n=1}^{N}(\max(0, m - \left\|f_n^{\text{act}}\right\|) + \left\|f_n^{\text{bkg}}\right\|)^2, \tag{7}$$

where $f_n^{\text{act}} = \frac{1}{k^{\text{act}}}\sum_{i\in\mathcal{S}^{\text{act}}} f_{n,i}$ and $f_n^{\text{bkg}} = \frac{1}{k^{\text{bkg}}}\sum_{j\in\mathcal{S}^{\text{bkg}}} f_{n,j}$ are the mean features of the pseudo action and background segments of the n-th video, respectively. $\|\cdot\|$ is the norm function, and m is the pre-defined maximum feature magnitude, the same in Eq. 4.

Background entropy loss. Although uncertainty modeling loss encourages background segments to produce low logits for all actions, softmax scores for some action classes could be high due to the relativeness of softmax function. To prevent background segments from having a high softmax score for any action class, we define a loss function which maximizes the entropy of action probability from background segments, *i.e.*, background segments are forced to have uniform probability distribution for action classes as described in Fig. 3 (b). The loss is calculated as follows:

$$\mathcal{L}_{\text{be}} = \frac{1}{NC}\sum_{n=1}^{N}\sum_{c=1}^{C} -\log(p_c(\tilde{s}_n^{\text{bkg}})), \tag{8}$$

where $p_c(\tilde{s}_n^{\text{bkg}}) = \frac{1}{k^{\text{bkg}}}\sum_{j\in\mathcal{S}^{\text{bkg}}} p_c(\tilde{s}_{n,j})$ is the averaged action probability for the c-th class of pseudo background segments, and $p_c(\tilde{s}_{n,j})$ is the softmax score for class c of $\tilde{s}_{n,j}$.

Supervision	Method	mAP@IoU (%)							
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	AVG
Full	S-CNN (Shou, Wang, and Chang 2016)	47.7	43.5	36.3	28.7	19.0	10.3	5.3	27.3
	SSN (Zhao et al. 2017)	66.0	59.4	51.9	41.0	29.8	-	-	-
	TAL-Net (Chao et al. 2018)	59.8	57.1	53.2	48.5	42.8	33.8	20.8	45.1
	BSN (Lin et al. 2018)	-	-	53.5	45.0	36.9	28.4	20.0	-
	P-GCN (Zeng et al. 2019)	69.5	67.8	63.6	57.8	49.1	-	-	-
	G-TAD (Xu et al. 2020)	-	-	66.4	60.4	51.6	37.6	22.9	-
Weak†	STAR (Xu et al. 2019)	68.8	60.0	48.7	34.7	23.0	-	-	-
	3C-Net (Narayan et al. 2019)	59.1	53.5	44.2	34.1	26.6	-	8.1	-
	PreTrimNet (Zhang et al. 2020)	57.5	50.7	41.4	32.1	23.1	14.2	7.7	23.7
Weak	UntrimmedNets (Wang et al. 2017)	44.4	37.7	28.2	21.1	13.7	-	-	-
	Hide-and-seek (Singh and Lee 2017)	36.4	27.8	19.5	12.7	6.8	-	-	-
	STPN (Nguyen et al. 2018)	52.0	44.7	35.5	25.8	16.9	9.9	4.3	27.0
	AutoLoc (Shou et al. 2018)	-	-	35.8	29.0	21.2	13.4	5.8	-
	W-TALC (Paul, Roy, and Roy-Chowdhury 2018)	55.2	49.6	40.1	31.1	22.8	-	7.6	-
	MAAN (Yuan et al. 2019)	59.8	50.8	41.1	30.6	20.3	12.0	6.9	31.6
	Liu *et al.* (Liu, Jiang, and Wang 2019)	57.4	50.8	41.2	32.1	23.1	15.0	7.0	32.4
	CleanNet (Liu et al. 2019)	-	-	37.0	30.9	23.9	13.9	7.1	-
	TSM (Yu et al. 2019)	-	-	39.5	-	24.5	-	7.1	-
	Nguyen *et al.* (Nguyen, Ramanan, and Fowlkes 2019)	60.4	56.0	46.6	37.5	26.8	17.6	9.0	36.3
	BaS-Net (Lee, Uh, and Byun 2020)	58.2	52.3	44.6	36.0	27.0	18.6	10.4	35.3
	RPN (Huang et al. 2020)	62.3	57.0	48.2	37.2	27.9	16.7	8.1	36.8
	DGAM (Shi et al. 2020)	60.0	54.2	46.8	38.2	28.8	19.8	11.4	37.0
	Gong *et al.* (Gong et al. 2020)	-	-	46.9	38.9	30.1	19.8	10.4	-
	ActionBytes (Jain, Ghodrati, and Snoek 2020)	-	-	43.0	35.8	29.0	-	9.5	-
	EM-MIL (Luo et al. 2020)	59.1	52.7	45.5	36.8	30.5	22.7	**16.4**	37.7
	A2CL-PT (Min and Corso 2020)	61.2	56.1	48.1	39.0	30.1	19.2	10.6	37.8
	TSCN (Zhai et al. 2020)	63.4	57.6	47.8	37.7	28.7	19.4	10.2	37.8
	Ours	**67.5**	**61.2**	**52.3**	**43.4**	**33.7**	**22.9**	12.1	**41.9**

Table 1: Comparison on THUMOS'14. AVG is the average mAP under the thresholds 0.1:0.1:0.7, while † indicates the use of additional information, such as action frequency or human pose.

3.4 Inference

At the test time, for an input video, we first obtain the video-level softmax score and threshold on it with θ_{vid} to determine which action classes are to be localized. For the remaining action classes, we calculate the segment-wise posterior probability by multiplying the segment-level softmax score and the probability of being an action segment following Eq. 3. Afterwards, the segments whose posterior probabilities are larger than θ_{seg} are selected as the candidate segments. Finally, consecutive candidate segments are grouped into a single proposal, whose scores are calculated following (Liu, Jiang, and Wang 2019). To enrich the proposal pool, we use multiple thresholds for θ_{seg} and non-maxium suppression (NMS) is performed for overlapped proposals.

4 Experiments

4.1 Experimental Settings

Datasets. THUMOS'14 (Jiang et al. 2014) is a widely used dataset for temporal action localization, containing 200 validation videos and 213 test videos of 20 action classes. It is a very challenging benchmark as the lengths of the videos are diverse and actions frequently occur (on average 15 instances per video). Following the previous work, we use validation videos for training and test videos for test. On the other hand, ActivityNet (Caba Heilbron et al. 2015) is a large-scale benchmark with two versions. ActivityNet 1.3, consisting of 200 action categories, includes 10,024 training videos, 4,926 validation videos and 5,044 test videos. ActivityNet 1.2 is a subset of the version 1.3, and is composed of 4,819 training videos, 2,383 validation videos and 2,480 test videos of 100 action classes. Because the ground-truths for the test videos of ActivityNet are withheld for the challenge, we utilize validation videos for evaluation.

Evaluation metrics. We evaluate our method with mean average precisions (mAPs) under several different intersection of union (IoU) thresholds, which are the standard evaluation metrics for temporal action localization. The official evaluation code of ActivityNet is used for measurement.

Implementation details. As the feature extractor, we employ I3D networks (Carreira and Zisserman 2017) pre-trained on Kinetics (Carreira and Zisserman 2017), which take input segments with 16 frames. It should be noted that we do not finetune the feature extractor for fair comparison. TVL1 algorithm (Wedel et al. 2009) is used to extract optical flow from videos. We fix the number of segments T as 750 and 50 for THUMOS'14 and ActivityNet, respectively. The sampling method is the same as STPN (Nguyen et al. 2018). The number of the pseudo action/background frames

Sup.	Method	mAP@IoU (%) 0.5	0.75	0.95	AVG
Full	SSN (2017)	41.3	27.0	6.1	26.6
Weak†	3C-Net (2019)	37.2	-	-	21.7
Weak	UntrimmedNets (2017)	7.4	3.2	0.7	3.6
	AutoLoc (2018)	27.3	15.1	3.3	16.0
	W-TALC (2018)	37.0	12.7	1.5	18.0
	Liu *et al.* (2019)	36.8	22.9	5.6	22.4
	CleanNet (2019)	37.1	20.3	5.0	21.6
	TSM (2019)	28.3	17.0	3.5	17.1
	RPN (2020)	37.6	23.9	5.4	23.3
	BaS-Net (2020)	38.5	24.2	5.6	24.3
	DGAM (2020)	41.0	23.5	5.3	24.4
	Gong *et al.* (2020)	40.0	25.0	4.6	24.6
	EM-MIL (2020)	37.4	-	-	20.3
	TSCN (2020)	37.6	23.7	5.7	23.6
	Ours	**41.2**	**25.6**	**6.0**	**25.9**

Table 2: Results on ActivityNet 1.2. AVG is the averaged mAP at the thresholds 0.5:0.05:0.95, while † means the use of action counts.

Score		Loss		mAP@IoU (%)							
softmax	fusion	$\mathcal{L}_{\text{um}}$	$\mathcal{L}_{\text{be}}$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	AVG
✓				42.3	35.1	27.7	21.7	16.4	10.5	4.6	22.6
	✓			58.8	50.0	40.5	31.7	23.6	15.5	7.2	32.5
	✓	✓		65.3	59.2	50.8	42.3	32.4	21.1	10.4	40.2
	✓	✓	✓	67.5	61.2	52.3	43.4	33.7	22.9	12.1	41.9

Table 3: Ablation study on THUMOS'14. AVG represents the average mAP at IoU thresholds 0.1:0.1:0.7. "Score" indicates the way of calculating the final scores. To derive final scores, the softmax method uses the segment-level softmax scores, while the fusion method fuses the segment-level softmax scores and uncertainty as in Eq. 3.

is determined by the ratio parameters, *i.e.*, $k^{\text{act}} = T/r^{\text{act}}$ and $k^{\text{bkg}} = T/r^{\text{bkg}}$. All hyper-parameters are set by grid search; $m = 100$, $r^{\text{act}} = 9$, $r^{\text{bkg}} = 4$, $\alpha = 5 \times 10^{-4}$, $\beta = 1$, and $\theta_{\text{vid}} = 0.2$. Multiple thresholds from 0 to 0.25 with a step size 0.025 are used as θ_{seg}, then we perform non-maxium suppression (NMS) with an IoU threshold of 0.6.

4.2 Comparison With State-of-the-art Methods

We compare our method with the existing fully-supervised and weakly-supervised methods under several IoU thresholds. We separate the entries by horizontal lines regarding the levels of supervision. For readability, all results are reported on the percentage scale.

Table 1 demonstrates the results on THUMOS'14. As shown, our method achieves a new state-of-the-art performance on weakly-supervised temporal action localization. Notably, our method significantly outperforms the existing background modeling approaches, Liu *et al.* (Liu, Jiang, and Wang 2019), Nguyen *et al.* (Nguyen, Ramanan, and Fowlkes 2019) and BaS-Net (Lee, Uh, and Byun 2020), by large margins of 9.5 %, 5.6 % and 6.6 % in terms of average mAP, respectively. This confirms the effectiveness of our uncertainty modeling. Moreover, with a much lower level of supervision, our method beats several fully-supervised methods, following the latest approaches with the least gap.

Sup.	Method	mAP@IoU (%) 0.5	0.75	0.95	AVG
Full	TAL-Net (2018)	38.2	18.3	1.3	20.2
	BSN (2018)	46.5	30.0	8.0	30.0
	BMN (2019)	50.1	34.8	8.3	33.9
	P-GCN (2019)	48.3	33.2	3.3	31.1
	G-TAD (2020)	50.4	34.6	9.0	34.1
Weak†	STAR (2019)	31.1	18.8	4.7	-
	PreTrimNet (2020)	34.8	20.9	5.3	22.5
Weak	STPN (2018)	29.3	16.9	2.6	-
	MAAN (2019)	33.7	21.9	5.5	-
	Liu *et al.* (2019)	34.0	20.9	**5.7**	21.2
	TSM (2019)	30.3	19.0	4.5	-
	Nguyen *et al.* (2019)	36.4	19.2	2.9	-
	BaS-Net (2020)	34.5	22.5	4.9	22.2
	A2CL-PT (2020)	36.8	22.5	5.2	22.5
	TSCN (2020)	35.3	21.4	5.3	21.7
	Ours	**37.0**	**23.9**	**5.7**	**23.7**

Table 4: Comparison on ActivityNet 1.3. † means the use of external data and AVG is the averaged mAP under the thresholds 0.5:0.05:0.95.

m	10	25	50	100	150	200	250
mAP@AVG	39.6	41.1	41.8	41.9	41.7	41.9	41.6

Table 5: Analysis on the maximum feature magnitude m on THUMOS'14. We report the average mAPs under IoU thresholds 0.1:0.1:0.7 with varying m from 10 to 250.

The performances on ActivityNet 1.2 are demonstrated in Table 2. Consistent with the results on THUMOS'14, our method outperforms all weakly-supervised approaches by obvious margins, following SSN (Zhao et al. 2017) with a small gap less than 1 %. We also summarize the results on ActivityNet 1.3 in Table 4. We see that our method surpasses existing weakly-supervised methods including those which use external information. Furthermore, our method performs better than the supervised method TAL-Net (Chao et al. 2018) in terms of average mAP, demonstrating the potential of weakly-supervised action localization.

4.3 Ablation Study

Effects of loss components and score calculation. In Table 3, we investigate the contribution of each component on THUMOS'14. Firstly, the *baseline* is set as the main pipeline only with video-level classification loss ($\mathcal{L}_{\text{cls}}$). We try two types of score calculation methods for the baseline: (1) softmax and (2) fusion of softmax and uncertainty (as in Eq. 3). For the second, as the original feature magnitudes are unconstrained, it is inappropriate to use Eq. 4. Instead, we perform min-max normalization on feature magnitudes to estimate the probability of a segment being in-distribution. Surprisingly, the fusion method largely improves the average mAP by 9.9 %, as the relatively low magnitudes of background segments offset their softmax scores, which reduces false positives. This confirms both the correlation between feature magnitudes and action/background segments

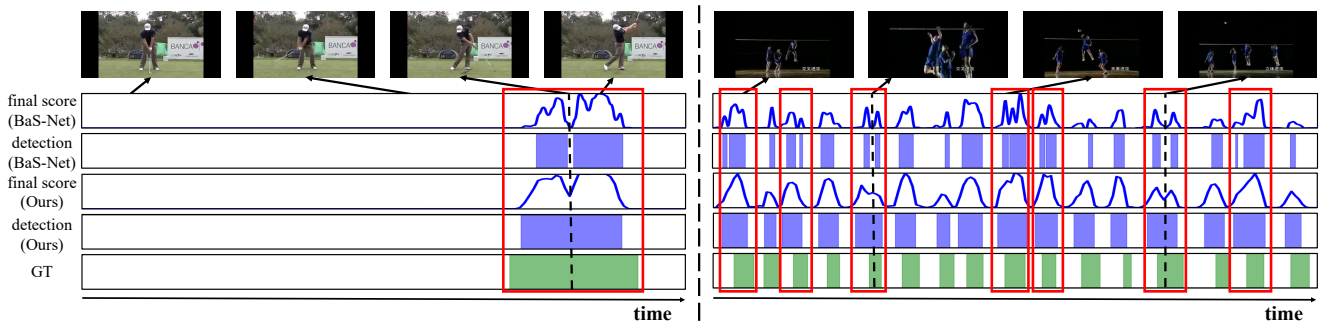


Figure 4: Qualitative comparison with BaS-Net (Lee, Uh, and Byun 2020) on THUMOS'14. We provide two different cases of examples: (1) a sparse action case with *GolfSwing* and (2) a frequent action case with *VolleyballSpking*. There are five plots with sampled frames for each example. The first and second plot show the final scores and the detection results of the corresponding action class from BaS-Net respectively, while the third and fourth plot represent those from our method respectively. The last plot is the ground truth action intervals. The horizontal axis in each plot means the timesteps of the video, while the vertical axes are the score values from 0 to 1. In the red boxes, while BaS-Net fails to cover complete action instances and splits them into multiple detection results, our method precisely localizes them. The black dashed lines indicate the action frames that are misclassified by BaS-Net but detected by our method.

and the importance of background modeling. Thereafter, the proposed losses, *i.e.*, uncertainty modeling loss ($\mathcal{L}_{\text{um}}$) and background entropy loss ($\mathcal{L}_{\text{be}}$), are added subsequently. We note that background entropy loss cannot stand alone, as it is calculated with the pseudo background segments which are selected based on uncertainty. As a result, uncertainty modeling boosts the performance to 40.2 % in terms of the average mAP, which already achieves a new state-of-the-art. In addition, background entropy loss further improves the performance, widening the gap with the existing methods.

Analysis on the maximum feature magnitude m. The maximum feature magnitude m in Eq. 4 determines how different the feature magnitudes from action frames and those from background frames are. We investigate the effect of the maximum feature magnitude m in Table 5, where m is altered from 10 to 250. We notice that the performance differences are insignificant when m is large enough. Meanwhile, the performance decreases with a too small m, because the separation between action and background is insufficient.

4.4 Qualitative Results

To confirm the superiority of our background modeling, we qualitatively compare our method with BaS-Net (Lee, Uh, and Byun 2020) which uses the background class. In Fig. 4, two examples from THUMOS'14 are illustrated: sparse and frequent action cases. In the both cases, we see that our model detects the action instances more precisely. More specifically, in the red boxes, it can be noticed that BaS-Net splits one action instance into multiple incomplete detection results. We conjecture that this problem is because BaS-Net strongly forces inconsistent background frames to belong to one class, which makes the model misclassify confusing parts of action instances as the background class (See the black dashed lines). On the contrary, our model provides better separation between action and background via uncertainty modeling instead of using a separate class. Consequently, our model successfully localizes the complete action instances without splitting them.

5 Conclusion

In this work, we identified the inherent limitations of existing background modeling approaches, with the observation that background frames may be dynamic and inconsistent. Thereafter, based on their inconsistency, we proposed to formulate background frames as out-of-distribution samples and model uncertainty with feature magnitudes. In order to train the model to identify background frames without frame-level annotations, we designed a new architecture, where uncertainty is learned via multiple instance learning. Furthermore, background entropy loss was introduced to prevent background segments from leaning toward any specific action class. The ablation study verified that our uncertainty modeling and background entropy loss both are beneficial for the localization performance. Through the experiments on two most popular benchmarks - THUMOS'14 and ActivityNet, our method achieved a new state-of-the-art with a large margin on weakly-supervised temporal action localization. We believe it would be a promising future direction to adopt the out-of-distribution formulation of background to the fully-supervised setting or other related tasks.

Acknowledgements

This project was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1A2C2003760) and the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (No. 2019-0-01558: Study on audio, video, 3d map and activation map generation system using deep generative model; No. 2020-0-01361: Artificial Intelligence Graduate School Program (YONSEI UNIVERSITY)).

References

Alwassel, H.; Caba Heilbron, F.; and Ghanem, B. 2018. Action search: Spotting actions in videos and its application to temporal action localization. In *ECCV*, 251–266.

Bendale, A.; and Boult, T. E. 2016. Towards open set deep networks. In *CVPR*, 1563–1572.

Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; and Carlos Niebles, J. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 961–970.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 6299–6308.

Chao, Y.-W.; Vijayanarasimhan, S.; Seybold, B.; Ross, D. A.; Deng, J.; and Sukthankar, R. 2018. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, 1130–1139.

Dhamija, A. R.; Günther, M.; and Boult, T. E. 2018. Reducing Network Agnostophobia. In *NeurIPS*.

Gong, G.; Wang, X.; Mu, Y.; and Tian, Q. 2020. Learning Temporal Co-Attention Models for Unsupervised Video Action Localization. In *CVPR*.

Hendrycks, D.; and Gimpel, K. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*.

Huang, L.; Huang, Y.; Ouyang, W.; and Wang, L. 2020. Relational Prototypical Network for Weakly Supervised Temporal Action Localization. In *AAAI*.

Jain, M.; Ghodrati, A.; and Snoek, C. G. M. 2020. ActionBytes: Learning From Trimmed Videos to Localize Actions. In *CVPR*.

Jiang, Y.-G.; Liu, J.; Roshan Zamir, A.; Toderici, G.; Laptev, I.; Shah, M.; and Sukthankar, R. 2014. THUMOS Challenge: Action Recognition with a Large Number of Classes. http://crcv.ucf.edu/THUMOS14/.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*, 6402–6413.

Lee, P.; Uh, Y.; and Byun, H. 2020. Background Suppression Network for Weakly-supervised Temporal Action Localization. In *AAAI*.

Liang, S.; Li, Y.; and Srikant, R. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*.

Lin, C.; Li, J.; Wang, Y.; Tai, Y.; Luo, D.; Cui, Z.; Wang, C.; Li, J.; Huang, F.; and Ji, R. 2020. Fast Learning of Temporal Action Proposal via Dense Boundary Generator. In *AAAI*, 11499–11506.

Lin, T.; Liu, X.; Li, X.; Ding, E.; and Wen, S. 2019. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In *ICCV*, 3888–3897.

Lin, T.; Zhao, X.; Su, H.; Wang, C.; and Yang, M. 2018. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 3–19.

Liu, D.; Jiang, T.; and Wang, Y. 2019. Completeness Modeling and Context Separation for Weakly Supervised Temporal Action Localization. In *CVPR*, 1298–1307.

Liu, Z.; Wang, L.; Zhang, Q.; Gao, Z.; Niu, Z.; Zheng, N.; and Hua, G. 2019. Weakly Supervised Temporal Action Localization Through Contrast Based Evaluation Networks. In *ICCV*, 3899–3908.

Long, F.; Yao, T.; Qiu, Z.; Tian, X.; Luo, J.; and Mei, T. 2019. Gaussian Temporal Awareness Networks for Action Localization. In *CVPR*, 344–353.

Luo, Z.; Guillory, D.; Shi, B.; Ke, W.; Wan, F.; Darrell, T.; and Xu, H. 2020. Weakly-Supervised Action Localization with Expectation-Maximization Multi-Instance Learning. In *ECCV*.

Ma, F.; Zhu, L.; Yang, Y.; Zha, S.; Kundu, G.; Feiszli, M.; and Shou, Z. 2020. SF-Net: Single-Frame Supervision for Temporal Action Localization. In *ECCV*.

Maron, O.; and Lozano-Pérez, T. 1998. A Framework for Multiple-Instance Learning. In *NIPS*.

Min, K.; and Corso, J. J. 2020. Adversarial Background-Aware Loss for Weakly-supervised Temporal Activity Localization. In *ECCV*.

Narayan, S.; Cholakkal, H.; Khan, F. S.; and Shao, L. 2019. 3C-Net: Category Count and Center Loss for Weakly-Supervised Action Localization. In *ICCV*, 8678–8686.

Nguyen, P.; Liu, T.; Prasad, G.; and Han, B. 2018. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, 6752–6761.

Nguyen, P. X.; Ramanan, D.; and Fowlkes, C. C. 2019. Weakly-Supervised Action Localization With Background Modeling. In *ICCV*, 5501–5510.

Paul, S.; Roy, S.; and Roy-Chowdhury, A. K. 2018. W-talc: Weakly-supervised temporal activity localization and classification. In *ECCV*, 563–579.

Shi, B.; Dai, Q.; Mu, Y.; and Wang, J. 2020. Weakly-Supervised Action Localization by Generative Attention Modeling. In *CVPR*.

Shou, Z.; Chan, J.; Zareian, A.; Miyazawa, K.; and Chang, S.-F. 2017. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, 5734–5743.

Shou, Z.; Gao, H.; Zhang, L.; Miyazawa, K.; and Chang, S.-F. 2018. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *ECCV*, 154–171.

Shou, Z.; Wang, D.; and Chang, S.-F. 2016. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, 1049–1058.

Singh, K. K.; and Lee, Y. J. 2017. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 3544–3553.

Wang, L.; Xiong, Y.; Lin, D.; and Van Gool, L. 2017. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 4325–4334.

Wedel, A.; Pock, T.; Zach, C.; Bischof, H.; and Cremers, D. 2009. An improved algorithm for tv-l 1 optical flow. In *Statistical and geometrical approaches to visual motion analysis*, 23–45. Springer.

Xiong, Y.; Zhao, Y.; Wang, L.; Lin, D.; and Tang, X. 2017. A pursuit of temporal accuracy in general activity detection. *arXiv preprint arXiv:1703.02716* .

Xu, M.; Zhao, C.; Rojas, D. S.; Thabet, A.; and Ghanem, B. 2020. G-TAD: Sub-Graph Localization for Temporal Action Detection. In *CVPR*.

Xu, Y.; Zhang, C.; Cheng, Z.; Xie, J.; Niu, Y.; Pu, S.; and Wu, F. 2019. Segregated Temporal Assembly Recurrent Networks for Weakly Supervised Multiple Action Detection. In *AAAI*.

Yang, K.; Qiao, P.; Li, D.; Lv, S.; and Dou, Y. 2018. Exploring temporal preservation networks for precise temporal action localization. In *AAAI*.

Yu, T.; Ren, Z.; Li, Y.; Yan, E.; Xu, N.; and Yuan, J. 2019. Temporal Structure Mining for Weakly Supervised Action Detection. In *ICCV*, 5521–5530.

Yuan, J.; Ni, B.; Yang, X.; and Kassim, A. A. 2016. Temporal action localization with pyramid of score distribution features. In *CVPR*, 3093–3102.

Yuan, Y.; Lyu, Y.; Shen, X.; Tsang, I. W.-H.; and Yeung, D.-Y. 2019. Marginalized average attentional network for weakly-supervised learning. In *ICLR*.

Zeng, R.; Huang, W.; Tan, M.; Rong, Y.; Zhao, P.; Huang, J.; and Gan, C. 2019. Graph Convolutional Networks for Temporal Action Localization. In *ICCV*, 7093–7102.

Zhai, Y.; Wang, L.; Tang, W.; Zhang, Q.; Yuan, J.; and Hua, G. 2020. Two-Stream Consensus Network for Weakly-Supervised Temporal Action Localization. In *ECCV*.

Zhang, X.; Shi, H.; Li, C.; and Li, P. 2020. Multi-instance Multi-label Action Recognition and Localization Based on Spatio-Temporal Pre-trimming for Untrimmed Videos. In *AAAI*.

Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D. 2017. Temporal action detection with structured segment networks. In *ICCV*, 2914–2923.

Zhou, Z.-H. 2004. Multi-instance learning: A survey. *Department of Computer Science & Technology, Nanjing University, Tech. Rep* .