Collaborative Foreground, Background, and Action Modeling Network for Weakly Supervised Temporal Action Localization

Md Moniruzzaman, Graduate Student Member, IEEE, Zhaozheng Yin, Senior Member, IEEE

Abstract—In this paper, we explore the problem of Weakly-supervised Temporal Action Localization (W-TAL), where the task is to localize the temporal boundaries of all action instances in an untrimmed video with only video-level supervision. The existing W-TAL methods achieve a good action localization performance by separating the discriminative action and background frames. However, there is still a large performance gap between the weakly and fully supervised methods. The main reason comes from that there are plenty of ambiguous action and background frames in addition to the discriminative action and background frames. Due to the lack of temporal annotations in W-TAL, the ambiguous background frames may be localized as foreground and the ambiguous action frames may be suppressed as background, which result in false positives and false negatives, respectively. In this paper, we introduce a novel collaborative Foreground, Background, and Action Modeling Network (FBA-Net) to suppress the background (i.e., both the discriminative and ambiguous background) frames, and localize the actual-action-related (i.e., both the discriminative and ambiguous action) frames as foreground, for the precise temporal action localization. We design our FBA-Net with three branches: the foreground modeling (FM) branch, the background modeling (BM) branch, and the class-specific action and background modeling (CM) branch. The CM branch learns to highlight the video frames related to C action classes, and separate the action-related frames of C action classes from the (C + 1)th background class. The collaboration between FM and CM regularizes the consistency between the FM and the C action classes of CM, which reduces the false negative rate by localizing different actual-action-related (i.e., both the discriminative and ambiguous action) frames in a video as foreground. On the other hand, the collaboration between BM and CM regularizes the consistency between the BM and the (C + 1)th background class of CM, which reduces the false positive rate by suppressing both the discriminative and ambiguous background frames. Furthermore, the collaboration between FM and BM enforces more effective foreground-background separation. To evaluate the effectiveness of our FBA-Net, we perform extensive experiments on two challenging datasets, THUMOS14 and ActivityNet1.3. The experiments show that our FBA-Net attains superior results.

Index Terms—Temporal action localization, foreground modeling, background modeling, action modeling.

I. Introduction

TEMPORAL Action Localization (TAL), localizing the temporal boundaries of all action instances in an untrimmed video, is an important yet challenging task for video understanding, which has potential applications in high-level tasks such as video surveillance [1], video summarization [2], and others. Most existing methods [3], [4], [5], [6], [7], [8], [9], [10] are trained in a fully-supervised manner and achieve a remarkable performance, which expect that the manually annotated temporal boundaries of all action instances are accessible during the training phase. However, collecting such annotations for a fully-supervised setting has several pitfalls. For example, the precise temporal boundary annotation at the frame level is costly, time-consuming, and error-prone, which undermines the potential development of fully-supervised methods in real-world applications. This limitation motivates the research community to deal with the Weakly-supervised Temporal Action Localization (W-TAL), where only video-level labels are provided for the network training (i.e., what actions are included in a video is known during the training, but the exact timestamps of the action in the video are unknown). Compared with precise temporal boundary annotations of action instances of various action classes, collecting only video-level labels for network training is much easier and more practical. In this paper, we explore the temporal action localization task with such weak labels.

Most of the existing W-TAL methods follow a localization-by-classification pipeline [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], which can be divided into two main techniques, attention mechanisms and Multiple Instance Learning (MIL). The attention-based algorithms first learn to highlight the most discriminative video segments based on their relevance to the action class, and then classify the video into the corresponding action classes. On the other hand, the MIL-based algorithms treat the entire untrimmed video as a bag containing both positive instances (action-related frames) and negative instances (non-action background frames), where they first classify individual frames into action classes and then employ top-k aggregation techniques to get the video-level prediction. Both techniques learn a sequence of class-specific scores, named as Class Activation Scores (CAS), which helps to locate the action-related frames (defined as foreground frames in this paper) in the video based on their contribution to the video-level classification. Therefore, the temporal action localization performance depends to a large extent on the quality of the CAS. The quality of CAS is likely to improve in fully-supervised settings, where detailed temporal annotations are available during the training. However, due to the lack of such annotations, usually, a classification loss is used in

M. Moniruzzaman is with the Department of Computer Science, Stony Brook University, Stony Brook, New York, 11794 (e-mail: mmoniruzzama@cs.stonybrook.edu)

Z. Yin is with the Department of Computer Science and Department of Biomedical Informatics, Stony Brook University, Stony Brook, New York, 11794 (e-mail: zyin@cs.stonybrook.edu)

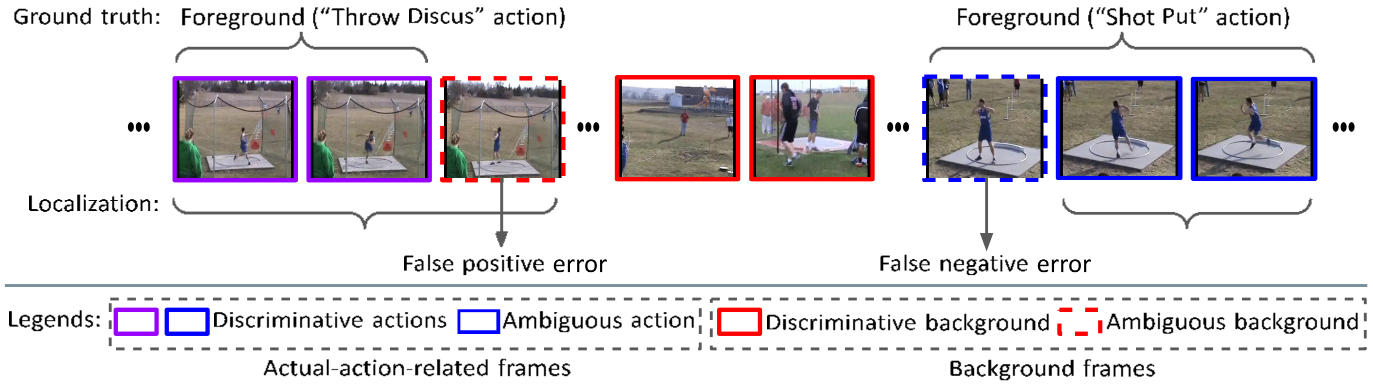


Fig. 1. An untrimmed video contains "Throw Discus" and "Shot Put" actions with many background frames for the Weakly-supervised Temporal Action Localization (W-TAL). Apart from the discriminative action (highly related to the actual actions) and discriminative background (not related to the actual actions) frames, there are a plenty of ambiguous action (less discriminative but related to the actual actions) and ambiguous background (frequently co-occur with the actual-action-related frames, but do not belong to the actual actions) frames. Existing W-TAL methods achieve a good performance by separating the discriminative action and background frames. However, due to the lack of temporal annotations, some ambiguous background frames may be treated as foreground and some ambiguous action frames may be treated as background, yielding false positives and false negatives, respectively. It is desired to suppress both the discriminative and ambiguous background frames, and localize both the discriminative and ambiguous action frames as the foreground, for the precise temporal action localization.

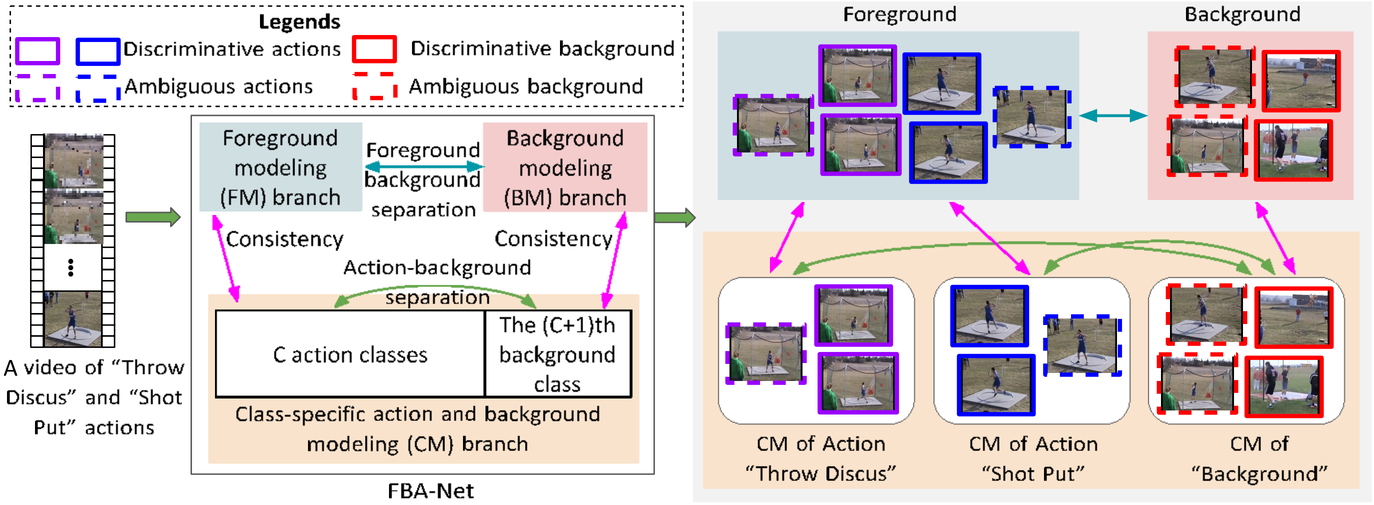

Legends
Discriminative actions
Ambiguous actions
Discriminative background
Ambiguous background
A video of "Throw Discus" and "Shot Put" actions
Foreground modeling (FM) branch
Foreground background separation
Background modeling (BM) branch
Consistency
Consistency
Action-background separation
C action classes
The (C+1)th background class
Class-specific action and background modeling (CM) branch
FBA-Net
Foreground
Background
CM of Action "Throw Discus"
CM of Action "Shot Put"
CM of "Background"

Fig. 2. We introduce a novel collaborative foreground, background, and action Modeling Network (FBA-Net) that consists of three branches, i.e., the foreground modeling (FM) branch, the background modeling (BM) branch, and the class-specific action and background modeling (CM) branch. The FM branch learns to highlight the foreground frames (both the discriminative and ambiguous action frames), without depending on any specific action class. The BM branch learns to highlight the background frames (both discriminative and ambiguous ones). The CM branch learns to highlight the video frames related to C action classes, and separate the action-related frames of C action classes from the $(C+1)$th background class. The collaboration between FM and CM branches regularizes the consistency between the FM and the C action classes of CM, which reduces the false negative rate by localizing the actual-action-related (both the discriminative and ambiguous action) frames of various actions as foreground. On the other hand, the collaboration between BM and CM branches regularizes the consistency between the BM and the $(C+1)$th background class of CM, which reduces the false positive rate by separating both the discriminative and ambiguous background frames from the action-related frames of various actions. Furthermore, the collaboration between FM and BM branches enforces more effective foreground-background separation.

weakly-supervised settings to highlight the foreground frames in CAS. In such weakly-supervised settings, some background frames (i.e., the video frames that contain no related action) may be treated as foreground and some foreground frames may be treated as background, yielding false positives and false negatives, respectively, in the learned CAS.

Recently, the latest W-TAL methods [16], [17], [20], [21], [22], [23], [24], [25] pay a significant amount of attentions to develop techniques for separating the discriminative action and background frames to reduce the false positive and false negative rates. Although these W-TAL methods achieve a good performance, there is still a big performance gap between the weakly and fully supervised methods. As shown in Fig. 1, the main reason comes from that the untrimmed video contains a significant number of ambiguous action and background frames in addition to the discriminative action and background frames. Due to the lack of temporal annotations in W-TAL, the ambiguous background frames may be localized as foreground since they provide strong clues for the action classification and the ambiguous action frames may be suppressed as background since they are less discriminative compared to the actual actions, which result in false positives and false negatives, respectively. Therefore, the motivated research question is: how to design a W-TAL network that can suppress both the discriminative and ambiguous background frames, and localize both the discriminative and ambiguous action frames as the foreground for the precise temporal action localization?

Our proposal and contribution: Motivated by the above research question, in this paper, we introduce a novel collaborative foreground, background, and action modeling Network (FBA-Net) that consists of three branches, i.e., the foreground modeling (FM) branch, the background modeling (BM) branch, and the class-specific action and background modeling (CM) branch, for the temporal action localization in untrimmed videos, as shown in Fig. 2. Our main contributions are six-fold:

- We introduce the CM branch, where we build the collaboration between C action classes and the $(C+1)$th background class. The collaboration within the CM branch learns to highlight the video frames related to C action classes, and separate the action-related frames of C action classes from the $(C+1)$th background class.
- We introduce the collaborative FM and CM branches, where the collaboration between FM and CM regularizes the consistency between the FM and the C action classes of CM that reduces the false negative rate by localizing the actual-action-related (both the discriminative and ambiguous action) frames of various actions as foreground.
- We introduce the collaborative BM and CM branches, where the collaboration between BM and CM regularizes the consistency between the BM and the $(C+1)$th background class of CM that reduces the false positive rate by separating both the discriminative and ambiguous background frames from the action-related frames of C action classes.
- We introduce the collaboration between FM and BM by inserting a separation loss that enforces more effective foreground-background separation.
- We propose a novel W-TAL network, called FBA-Net, which integrates FM, BM, and CM branches to reduce the false positive and false negative rates by suppressing the background (i.e., both the discriminative and ambiguous background) frames and localizing the actual-action-related (i.e., both the discriminative and ambiguous action) frames, respectively.
- We conduct extensive experiments on the challenging THUMOS14 and ActivityNet1.3 datasets. The results show that our FBA-Net achieves superior performance compared to the latest W-TAL methods.

II. Related works

Fully-supervised temporal action localization: The fully-supervised methods rely on precise frame-level temporal annotations for the temporal action localization task. Motivated by the success of the object detection framework [26], [27], [28], [29], [30], [31], several recent works [5], [6], [32], [33], [34], [35], [36], [37], [38] addressed the temporal action localization problem by adopting a two-stage framework, i.e., action proposals are generated first and then fed into a classification module. More recently, several works [7], [8], [9], [39], [40], [41] developed trainable proposal architectures to localize the start time and end time of the action instances. Even though these methods achieve impressive performance, they heavily rely on precise temporal annotations.

Weakly-supervised temporal action localization: We summarize these W-TAL methods into four categories.

Metric learning-based methods: The metric learning-based methods aim to highlight the most discriminative action-related features by reducing the intra-class and increasing the inter-class variations of feature representations. Paul et al. [12] used a co-activity similarity loss to enforce the feature similarity between the localized instances of the same class within different videos. 3C-Net [42] utilized category, counting, and center losses to learn the class-wise attention and localize the action instances in untrimmed videos. RPN [43] adopted a clustering loss to separate the discriminative action and background frames by learning the intra-compact features. A2CL-PT [44] employed the triplet loss to distinguish the background features from the action-related features for each video. Although these methods achieve remarkable progress, the main challenge for these methods is that they only focus on the most discriminative action frames but ignore the ambiguous action frames, which results in incomplete localization.

Erasing-based methods: The erasing-based methods aim to discover different but complementary action instances for the complete temporal action localization by iteratively erasing the most discriminative features from the feature map. The erasing strategy was first developed to model the completeness of objects in object detection task [45], [46], [47]. Recently, Hide-and-Seek [48] hid random frame sequences to force the network to discover different action parts for the temporal action localization task. More recently, Step-by-Step Erasion [49] and ACM-BANet [17] utilized an iterative multi-pass erasing strategy for discovering different action segments in CAS. During each iteration, these methods first identify the most discriminative features, which are then erased from the feature map and fed into the network of the next iteration. However, it is difficult to define a proper number of iterations to discover different complementary action segments for different action classes.

Pseudo label-based methods: Due to the lack of fine-grained temporal annotations, most of the existing W-TAL methods follow a localization-by-classification pipeline. The pseudo label-based methods seek to generate snippet-wise pseudo labels for bridging the gap between classification and localization. Recently, RefineLoc [50] iteratively generated the snippet-wise pseudo labels from the previous detection results, which were then used to supervise the W-TAL methods to learn snippet-wise foreground and background attention weights. EM-MIL [51] utilized an expectation-maximization framework to generate the pseudo-labels. TSCN [52] introduced an iterative refinement training method, where the snippet-wise pseudo labels were generated from the two-stream late fusion attention sequence. More recently, UGCT [53] introduced an uncertainty guided collaborative training

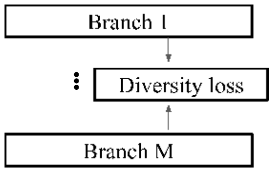
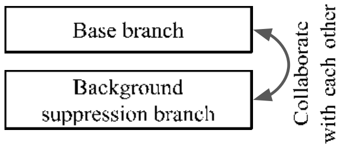
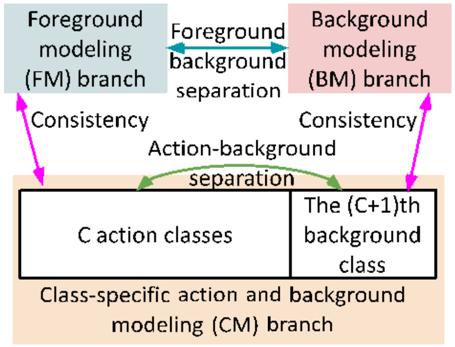
Type	Architecture design	Goal
Vanilla multi-branch architecture without weight-sharing strategy (e.g., [14])		The multiple branches are trained with additional regularization term (such as diversity loss) to discover different action parts.
Vanilla multi-branch architecture with weight-sharing strategy (e.g., [16],[17],[21])	 Base branch Background suppression branch Collaborate with each other	The base branch and the background suppression branch collaborate with each other to localize the discriminative action frames and suppress the discriminative background frames.
FBA-Net (Ours)	 Foreground modeling (FM) branch Foreground background separation Background modeling (BM) branch Consistency Consistency Action-background separation C action classes The (C+1)th background class Class-specific action and background modeling (CM) branch	The FM, BM, and CM branches collaborate with each other to localize both the discriminative and ambiguous action frames, and suppress both the discriminative and ambiguous background frames.

Fig. 3. Vanilla multi-branch architecture vs. our FBA-Net for the W-TAL.

strategy, where the pseudo labels were generated from modality collaborative learning and uncertainty estimation to learn more robust attention weights. However, the pseudo label-based methods usually first utilize an existing W-TAL network as the classification head to generate the initial pseudo labels, and then iteratively refine that W-TAL network with the pseudo labels to improve the localization performance. Therefore, the performances of the pseudo label-based algorithms heavily rely on the performances of the existing methods. At the same time, it is also difficult to set a proper number of iterations to achieve optimal performances.

Multi-branch architecture-based methods: The multi-branch architecture-based methods aim to separate the action and background frames by inserting additional regularization terms between different branches or using the weight-sharing strategy with different training objectives. Recently, CMCS [14] introduced a multi-branch network with a diversity loss to discover different action parts for the action completeness modeling. HAM-Net [54] proposed a hybrid attention mechanism that includes soft, semi-soft and hard attentions to localize action instances. Huang et al. [19] introduced a two-branch relational prototypical network, where the prior knowledge about label dependencies was used to generate relational prototypes. More recently, some methods [16], [17], [21] introduced an asymmetrical multi-branch architecture with a weight-sharing strategy to separate the foreground and background frames. For example, BaS-Net [16] introduced a base branch and a background suppression branch, where both branches share the weights of a class-specific classifier with different training objectives. The base branch classifies an input video regarding the action classes and the background class, while the background suppression branch classifies an input video focusing only on action classes. The weight-sharing strategy with different training objectives ensures that the class-specific classifier learns to separate the foreground and background frames. However, although these methods show a good performance in separating the discriminative action and background frames, the foreground localized through these methods may localize the ambiguous background frames that result in false positives, while the background suppressed through these methods may suppress the ambiguous action frames that result in false negatives. Our method belongs to the category of multi-branch architectures. As summarized in Fig. 3, in contrast to the existing multi-branch architecture-based methods, we introduce a novel foreground, background, and action modeling network (FBA-Net) to suppress both the discriminative and ambiguous background frames, and localize the actual-action-related frames (i.e., both the discriminative and ambiguous action frames) as the foreground, for the precise temporal action localization.

III. Methodology

In this section, we present our collaborative foreground, background, and action modeling network (FBA-Net) in detail.

A. Feature Embedding

Following the recent W-TAL methods [15], [16], [17], [20], [24], for a given untrimmed video V, we first divide it into T

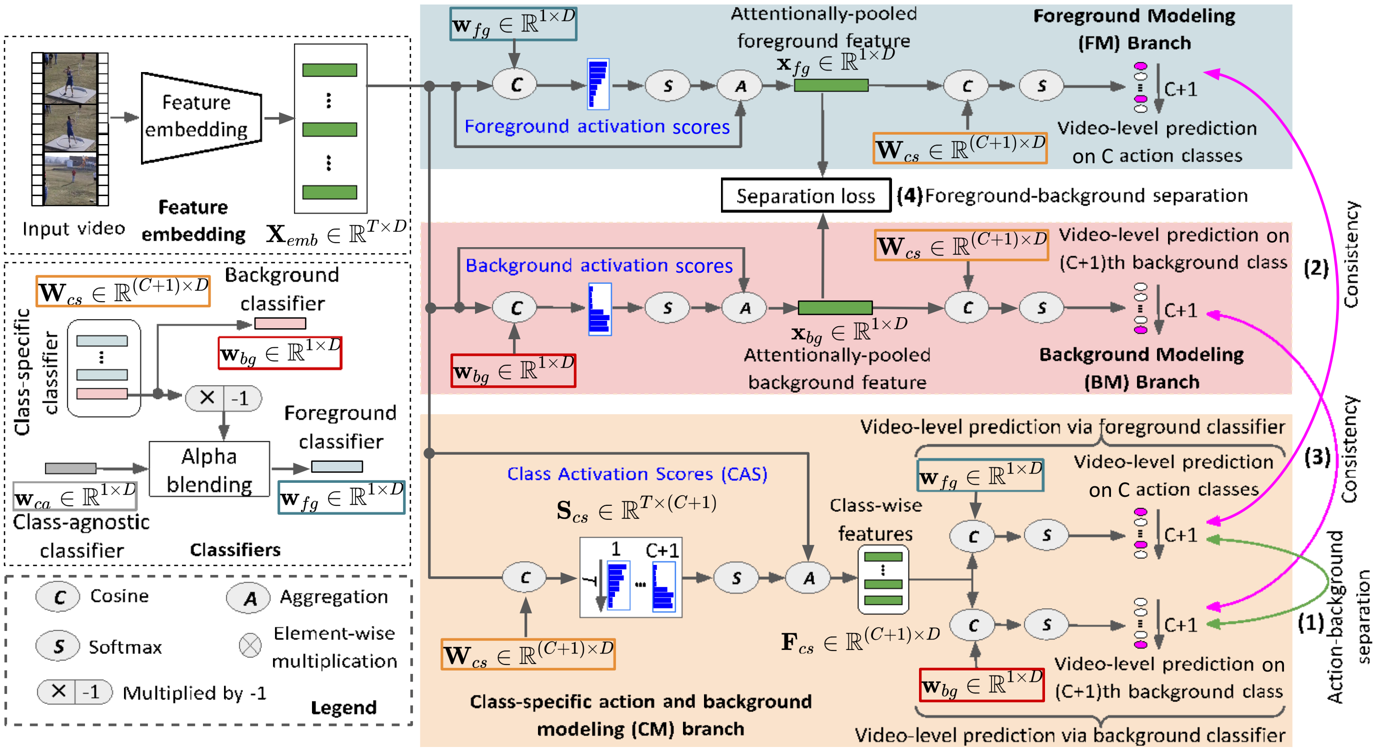

Input video
Feature embedding
Feature embedding
$\mathbf{X}_{emb} \in \mathbb{R}^{T\times D}$
$\mathbf{w}_{fg} \in \mathbb{R}^{1\times D}$
Foreground activation scores
Attentionally-pooled foreground feature
$\mathbf{x}_{fg} \in \mathbb{R}^{1\times D}$
Foreground Modeling (FM) Branch
C+1
$\mathbf{W}_{cs} \in \mathbb{R}^{(C+1)\times D}$
Video-level prediction on C action classes
Separation loss
(4) Foreground-background separation
Background activation scores
$\mathbf{W}_{cs} \in \mathbb{R}^{(C+1)\times D}$
Video-level prediction on (C+1)th background class
C+1
$\mathbf{w}_{bg} \in \mathbb{R}^{1\times D}$
$\mathbf{x}_{bg} \in \mathbb{R}^{1\times D}$
Attentionally-pooled background feature
Background Modeling (BM) Branch
(2) Consistency
Class-specific classifier
$\mathbf{W}_{cs} \in \mathbb{R}^{(C+1)\times D}$
Background classifier
$\mathbf{w}_{bg} \in \mathbb{R}^{1\times D}$
× -1
Foreground classifier
Alpha blending
$\mathbf{w}_{ca} \in \mathbb{R}^{1\times D}$
$\mathbf{w}_{fg} \in \mathbb{R}^{1\times D}$
Class-agnostic classifier
Classifiers
Video-level prediction via foreground classifier
Class Activation Scores (CAS)
$\mathbf{S}_{cs} \in \mathbb{R}^{T\times (C+1)}$
1 C+1
Class-wise features
$\mathbf{w}_{fg} \in \mathbb{R}^{1\times D}$
Video-level prediction on C action classes
(3) Consistency
C+1
C+1
(1) Action-background separation
$\mathbf{F}_{cs} \in \mathbb{R}^{(C+1)\times D}$
$\mathbf{W}_{cs} \in \mathbb{R}^{(C+1)\times D}$
$\mathbf{w}_{bg} \in \mathbb{R}^{1\times D}$
Video-level prediction on (C+1)th background class
Class-specific action and background modeling (CM) branch
Video-level prediction via background classifier
C Cosine
A Aggregation
S Softmax
⊗ Element-wise multiplication
× -1 Multiplied by -1
Legend
C
S
A

Fig. 4. Illustration of our FBA-Net. The FM branch first uses the foreground classifier w_{fg} in its front to compute the foreground activation scores, and then uses the class-specific classifier W_{cs} at the end to classify a video in regard to only the C action classes. The BM branch uses the background classifier w_{bg} in its front to compute the background activation scores first, and then uses the class-specific classifier W_{cs} at the end to classify a video in regard to only the $(C+1)$th background class. The CM branch first uses the class-specific classifier W_{cs} in its front to compute the Class Activation Scores (CAS), and then uses both the foreground classifier w_{fg} to classify a video in regard to only the C action classes and the background classifier w_{bg} to classify a video in regard to only the $(C+1)$th background class. These three branches build four types of collaboration: (1) The collaboration within the CM branch learns to highlight the video frames related to C action classes, and separate the action-related frames of C action classes from the $(C+1)$th background class; (2) The collaboration between FM and CM regularizes the consistency between the FM and the C action classes of CM, which localizes the actual-action-related (i.e., both the discriminative and ambiguous action) frames of various actions in CAS as foreground; (3) The collaboration between BM and CM regularizes the consistency between the BM and the $(C+1)$th background class of CM, which separates both the discriminative and ambiguous background frames from the action-related frames of C action classes; and (4) The collaboration between FM and BM by inserting a separation loss enforces more effective foreground-background separation. The classifiers themselves also collaborate with each other via some shared weights.

non-overlapping video segments (i.e., $V = \{s_t\}_{t=1}^{T}$). Then, we feed each video segment s_t into the pretrained I3D [55] network to extract its D-dimensional feature vector $x_t \in R^{1\times D}$. After that, we stack all segment features to generate a feature map for the entire video, $X = [x_1; \ldots; x_T] \in R^{T\times D}$. Since the I3D network is pretrained on action recognition tasks, it is desired to map the extracted features to W-TAL related features. Therefore, on top of the feature map X, we apply two convolution layers with ReLU activation to generate the embedded feature map, $X_{emb} \in R^{T\times D}$, for the W-TAL task.

B. Collaborative Foreground, Background, and Action Modeling Network (FBA-Net)

We design a collaborative foreground, background, and action modeling network (FBA-Net) with collaborative classifiers to suppress both the discriminative and ambiguous background frames, and localize both the discriminative and ambiguous action frames as foreground, for the precise localization. For this purpose, as shown in Fig. 4, on top of the embedded feature map, we incorporate three branches: (1) Foreground modeling (FM) branch to generate the foreground activation scores that highlight the foreground segments, i.e., the video segments (both discriminative and ambiguous ones) related to various actions, without depending on any specific action class; (2) Background modeling (BM) branch to generate the background activation scores that highlight the background video segments (both discriminative and ambiguous ones); and (3) Class-specific action and background modeling (CM) branch to generate the class activation scores, which highlight the video segments depending on action classes and separate the background segments from the action segments.

1) Classifiers: The FM, BM, and CM branches collaborate with each other via sharing their corresponding classifiers, where the classifiers also collaborate with each other via some shared weights. Usually, the W-TAL methods [11], [12], [13], [15], [24], [38], [42], [56] utilized a class-specific classifier $W \in R^{C\times D}$ and a class-agnostic classifier $w_{ca} \in R^{1\times D}$ for the action localization task, where C represents the number of action classes. With only video-level action class annota-tions, both the class-specific classifier and the class-agnostic classifier can easily highlight the discriminative action frames and suppress the discriminative background frames. But, there are some ambiguous action and background frames, which are hard to localize and suppress, respectively.

Differently, in addition to C action classes, we design our class-specific classifier $W_{cs} \in R^{(C+1)\times D}$ with an additional background class to separate both the discriminative and ambiguous background frames. Specifically, first, we randomly initialize a class-specific classifier $W_{cs} \in R^{(C+1)\times D}$ and a class-agnostic classifier $w_{ca} \in R^{1\times D}$, where the $(C + 1)$-th class in W_{cs} corresponds to the background class. Since the $(C + 1)$-th row of W_{cs} corresponds to the background class, we directly use the weights of the $(C + 1)$-th row of W_{cs} to define the background classifier w_{bg}, as follows:

$$w_{bg} = W_{cs}(C + 1, \cdot), \quad w_{bg} \in R^{1\times D} \tag{1}$$

On the other hand, we define the foreground classifier w_{fg} with the fusion of the class-agnostic classifier w_{ca} and the negative weights of the background classifier $(-w_{bg})$. We use both the w_{ca} and $(-w_{bg})$ classifiers to generate the foreground classifier for the following reasons:

- Since the foreground classifier learns attention weights in a class-agnostic way (i.e., without depending on any specific action class), we consider the class-agnostic classifier w_{ca} for generating the foreground classifier.
- Meanwhile, the activation score generated from the foreground classifier has the opposite meaning of the activation score generated by the background classifier. Therefore, we also consider the negative weights of the background classifier $(-w_{bg})$ for generating the foreground classifier.
- The fusion of the w_{ca} and $(-w_{bg})$ generates better foreground classifier compared to the individual w_{ca} classifier or the $(-w_{bg})$ classifier, which will be validated in the experiment section.

Formally, we define the foreground classifier w_{fg}, as follows:

$$w_{fg} = \alpha w_{ca} + (1 - \alpha)(-w_{bg}), \quad w_{fg} \in R^{1\times D} \tag{2}$$

where α is the combination factor ($\alpha \in [0, 1]$). Ablation studies on different α values are performed in experiments to show the contribution of w_{ca} and $(-w_{bg})$ classifiers to generate the foreground classifier. Usually, the existing W-TAL methods utilized only a class-agnostic classifier $w_{ca} \in R^{1\times D}$ to represent the foreground classifier. But, the class-agnostic classifier may highlight only the discriminative action frames. Differently, we design our foreground classifier with the fusion of the class-agnostic classifier and the negative weights of the background classifier. Since the background classifier generates the background activation scores that highlight both the discriminative and ambiguous background frames, and the activation scores generated from the foreground classifier have the opposite meaning of the activation scores generated by the background classifier, utilizing the negative weights of the background classifier in addition to the class-agnostic classifier encourages the foreground classifier to generate the foreground activation scores that highlight both the discriminative and ambiguous action frames.

Note, since the classifiers w_{bg} and w_{fg} are defined based on W_{cs} and w_{ca} as Eq. 1 and Eq. 2, respectively, only the classifiers W_{cs} and w_{ca} are to be learned from training.

2) Overview of the Collaborative Three Branch Architecture): With the help of a foreground classifier w_{fg}, a background classifier w_{bg} and a class-specific classifier W_{cs}, we design the FM, BM, and CM branches on top of the X_{emb}, as follows:

- The FM branch first uses the foreground classifier w_{fg} in its front to compute the foreground activation scores, and then uses the class-specific classifier W_{cs} at the end to classify a video in regard to only the C action classes.
- The BM branch uses the background classifier w_{bg} in its front to compute the background activation scores first, and then uses the class-specific classifier W_{cs} at the end to classify a video in regard to only the $(C + 1)$th background class.
- The CM branch first uses the class-specific classifier W_{cs} in its front to compute the Class Activation Scores (CAS), and then uses both the foreground classifier w_{fg} to classify a video in regard to only the C action classes and the background classifier w_{bg} to classify a video in regard to only the $(C + 1)$th background class.

As shown in Fig. 2 and Fig. 4, we build four types of collaboration among the three branches:

Collaboration between C action classes and the $(C+1)$th background class within the CM branch: The video-level prediction via foreground classifier within the CM branch aims to classify an input video in regard to only the C action classes, while the video-level prediction via background classifier within the CM branch aims to classify an input video in regard to only the $(C + 1)$th background class. Therefore, the CM branch learns to highlight the video frames related to C action classes, and separate the action-related frames of C action classes from the $(C + 1)$th background class.

Collaboration between FM and CM branches: The FM and CM branches collaborate with each other via sharing a foreground classifier w_{fg} and a class-specific classifier W_{cs}. In addition to sharing classifiers, the FM and CM also collaborate with each other by enforcing consistency in their training objectives. The video-level prediction of the FM branch aims to classify a video in regard to only the C action classes, while the video-level prediction via the foreground classifier within the CM branch also aims to classify an input video in regard to only the C action classes. Therefore, the training objectives of these two branches are aligned, which regularizes the consistency between the FM and the C action classes of CM. Since the FM branch learns to generate the foreground activation scores that highlight both the discriminative and ambiguous action-related frames, regularizing the consistency between the FM and the C action classes of CM localizes the actual-action-related (i.e., both the discriminative and ambiguous action) frames of various actions in CAS as foreground.

Collaboration between BM and CM branches: The BM and CM branches collaborate with each other via sharing a background classifier w_{bg} and a class-specific classifier W_{cs}. In addition to sharing classifiers, the BM and CM also collaborate with each other by enforcing consistency in their training objectives. The video-level prediction of the BM branch aims to classify a video in regard to only the

(C + 1)th background class, while the video-level prediction via the background classifier within the CM branch also aims to classify an input video in regard to only the (C + 1)th background class. Therefore, the training objectives of these two branches are aligned, which regularizes the consistency between the BM and the (C + 1)th background class of CM. Since the BM branch learns to generate the background activation scores that highlight both the discriminative and ambiguous background frames, regularizing the consistency between the BM and the (C + 1)th background class of CM separates both the discriminative and ambiguous background frames from the action-related frames of C action classes.

Collaboration between FM and BM branches: The FM and BM branches collaborate with each other via sharing a class-specific classifier W_{cs}, and by inserting a separation loss between them. The collaboration between FM and BM branches enforces more effective foreground-background separation, leading to improve the action localization performance.

3) Foreground Modeling (FM) Branch: We first compute the cosine similarity between the embedded feature map X_{emb} and the foreground classifier w_{fg} to get the foreground activation scores $s_{fg} \in \mathbb{R}^{T\times 1}$, are then passed through a softamx layer to obtain the foreground attention scores $\tilde{s}_{fg} \in \mathbb{R}^{T\times 1}$, as follows:

$$s_{fg}(t) = \cos(X_{emb}(t,\cdot), w_{fg}), \tag{3}$$

$$\tilde{s}_{fg}(t) = \frac{\exp(s_{fg}(t))}{\sum_i \exp(s_{fg}(i))} \tag{4}$$

Note that the W-TAL methods usually utilize the cosine similarity or the dot product to calculate the similarity. The dot product is magnitude sensitive while the cosine similarity, which measures the angle of two input vectors, has a bounded value range [−1, 1]. Therefore, the cosine similarity does not depend on the magnitude of the two vectors. Since the action localization is achieved by thresholding the segment-wise classification scores, the cosine similarity ensures stable classification scores to provide better localization performances for different videos and action classes. Therefore, we utilize the cosine similarity over the dot product in our FBA-Net.

After that, we aim to perform a pooling operation to get the attentionally-pooled foreground feature vector $x_{fg} \in \mathbb{R}^{1\times D}$ from the embedded feature map X_{emb} and the foreground attention scores $\tilde{s}_{fg}$. Rather than using the conventional pooling mechanism (e.g., average or max pooling), we perform the pooling operation through a feature aggregation process. Formally, we obtain the attentionally-pooled foreground feature vector $x_{fg} \in \mathbb{R}^{1\times D}$ of the video through a feature aggregation process, as follows:

$$x_{fg} = \sum_t \tilde{s}_{fg}(t) X_{emb}(t,\cdot), \quad x_{fg} \in \mathbb{R}^{1\times D} \tag{5}$$

Finally, based on the class-specific classifier W_{cs} and the x_{fg}, we compute the video-level class activation scores $p_{fg}^{FM} \in \mathbb{R}^{(C+1)}$ for the FM branch, as follows:

$$p_{fg}^{FM}(c) = \cos(x_{fg}, W_{cs}(c,\cdot)) \tag{6}$$

The $p_{fg}^{FM} \in \mathbb{R}^{(C+1)}$ is then passed through a softmax layer to get the video-level prediction $\tilde{p}_{fg}^{FM} \in \mathbb{R}^{(C+1)}$. The classification loss of the FM branch is defined by the cross-entropy loss, as follows:

$$L_{fg}^{FM} = \sum_{c=1}^{C+1} -y_{fg}^{FM}(c) \log \tilde{p}_{fg}^{FM}(c) \tag{7}$$

where $y_{fg}^{FM}(c)$ is the video-level label for the c-th class of the video. Since the FM branch aims to classify an input video regarding only action classes, we set $y_{fg}^{FM} = [y_{fg}^{FM}(1), \ldots, y_{fg}^{FM}(c), \ldots, y_{fg}^{FM}(C), 0]$, in which $y_{fg}^{FM}(c)$ is set to 1 if a video contains action class c.

4) Background Modeling (BM) Branch: Similar to FM branch, we first calculate the cosine similarity between the embedded feature map $X_{emb} \in \mathbb{R}^{T\times D}$ and the background classifier $w_{bg} \in \mathbb{R}^{1\times D}$ to get the background activation scores $s_{bg} \in \mathbb{R}^{T\times 1}$. The $s_{bg} \in \mathbb{R}^{T\times 1}$ is then passed through a softmax layer to get the background attention scores $\tilde{s}_{bg} \in \mathbb{R}^{T\times 1}$. After that, we obtain the attentionally-pooled background feature vector $x_{bg} \in \mathbb{R}^{1\times D}$ of the video through a feature aggregation process, as follows:

$$x_{bg} = \sum_t \tilde{s}_{bg}(t) X_{emb}(t,\cdot), \quad x_{bg} \in \mathbb{R}^{1\times D} \tag{8}$$

Finally, we calculate the cosine similarity between $W_{cs} \in \mathbb{R}^{(C+1)\times D}$ and $x_{bg} \in \mathbb{R}^{1\times D}$ to compute the video-level class activation scores $p_{bg}^{BM} \in \mathbb{R}^{(C+1)}$ for the BM branch. The $p_{bg}^{BM} \in \mathbb{R}^{(C+1)}$ is then passed through a softmax layer to get the video-level prediction $\tilde{p}_{bg}^{BM} \in \mathbb{R}^{(C+1)}$. The classification loss of the BM branch is defined by the cross-entropy loss:

$$L_{bg}^{BM} = \sum_{c=1}^{C+1} -y_{bg}^{BM}(c) \log \tilde{p}_{bg}^{BM}(c) \tag{9}$$

where $y_{bg}^{BM}(c)$ is the video-level label for the c-th class of the video. Since the BM branch aims to classify an input video regarding only the background class, we set $y_{bg}^{BM} = [0, \ldots, 0, 1]$, in which only the background class is set to 1.

5) Class-specific Action and Background Modeling (CM) Branch: For the class-specific action and background modeling (CM) branch, we first compute the cosine similarity between the embedded feature map $X_{emb} \in \mathbb{R}^{T\times D}$ and the class-specific classifier $W_{cs} \in \mathbb{R}^{(C+1)\times D}$ to get the segment-level class activation scores $S_{cs} \in \mathbb{R}^{T\times(C+1)}$, and then, we apply softamx along the temporal dimension of S_{cs} to get the normalized class activation scores $\tilde{S}_{cs} \in \mathbb{R}^{T\times(C+1)}$, as follows:

$$S_{cs}(t,c) = \cos(X_{emb}(t,\cdot), W_{cs}(c,\cdot)), \tag{10}$$

$$\tilde{S}_{cs}(t,c) = \frac{\exp(S_{cs}(t,c))}{\sum_k \exp(S_{cs}(k,c))} \tag{11}$$

where t and c represent the t-th segment of the input video and the c-th class, respectively. We compute the class-specific features $F_{cs} \in \mathbb{R}^{(C+1)\times D}$ for the entire video through a feature aggregation process, as follows:

$$F_{cs}(c) = \sum_{t} \tilde{S}_{cs}(t, c) X_{emb}(t, \cdot), \quad F_{cs} \in R^{(C+1)\times D} \tag{12}$$

Now, on top of the class-specific features F_{cs}, we perform the video-level predictions via both foreground and background classifiers.

Video-level prediction via foreground classifier: First, we compute the cosine similarity between the class-specific features $F_{cs} \in R^{(C+1)\times D}$ and the foreground classifier $w_{fg} \in R^{1\times D}$ to get the video-level class activation scores $p^{CM}_{fg} \in R^{(C+1)}$, as follows:

$$p^{CM}_{fg}(c) = \cos(F_{cs}(c, \cdot), w_{fg}) \tag{13}$$

The $p^{CM}_{fg} \in R^{(C+1)}$ is then passed through the softmax layer to get the video-level prediction $\tilde{p}^{CM}_{fg} \in R^{(C+1)}$. The classification loss for the video-level prediction via foreground classifier of CM branch is defined by a cross-entropy loss:

$$L^{CM}_{fg} = \sum_{c=1}^{C+1} -y^{CM}_{fg}(c) \log \tilde{p}^{CM}_{fg}(c) \tag{14}$$

where $y^{CM}_{fg}(c)$ is the video-level label for the c-th class of the video. Since the video-level prediction via foreground classifier of CM branch aims to classify an in-put video regarding only action classes, we set $y^{CM}_{fg} = [y^{CM}_{fg}(1), ..., y^{CM}_{fg}(c), ..., y^{CM}_{fg}(C), 0]$, in which $y^{CM}_{fg}(c)$ is set to 1 if a video contains action class c.

Video-level prediction via background classifier: We compute the cosine similarity between the class-specific features F_{cs} and the background classifier $w_{bg} \in R^{1\times D}$ to get the video-level class activation scores $p^{CM}_{bg} \in R^{(C+1)}$, as follows:

$$p^{CM}_{bg}(c) = \cos(F_{cs}(c, \cdot), w_{bg}) \tag{15}$$

The $p^{CM}_{bg} \in R^{(C+1)}$ is then passed through the softmax layer to get the video-level prediction $\tilde{p}^{CM}_{bg} \in R^{(C+1)}$. The classification loss for the video-level prediction via background classifier of CM branch is defined by a cross-entropy loss:

$$L^{CM}_{bg} = \sum_{c=1}^{C+1} -y^{CM}_{bg}(c) \log \tilde{p}^{CM}_{bg}(c) \tag{16}$$

where $y^{CM}_{bg}(c)$ is the video-level label for the c-th class of the video. Since the video-level prediction via background classifier of CM branch aims to classify an input video in regard to only the background class, we set $y^{CM}_{bg} = [0, ..., 0, 1]$, in which only the background class is set to 1.

C. Training

The feature embedding, the class-specific classifier W_{cs} and the class-agnostic classifier w_{ca}, and the three branches in the proposed FBA-Net, are jointly-trained by minimizing the following loss function:

$$L_{Total} = \lambda_1 L^{FM}_{fg} + \lambda_2 L^{BM}_{bg} + \lambda_3 L^{CM}_{fg} + \lambda_4 L^{CM}_{bg} + \lambda_5 L_{fg-bg} \tag{17}$$

Note, the background classifier w_{bg} and the foreground classifier w_{fg} are defined based on W_{cs} and w_{ca} as Eq. 1 and Eq. 2, respectively, so w_{bg} and w_{fg} are not learned independently in the training process. The λ_1, λ_2, λ_3, λ_4, and λ_5 are the balancing hyper-parameters to control the corresponding weights among the loss terms. In addition to the four classification losses in our FBA-Net, we also introduce a foreground-background separation loss L_{fg-bg} to encourage the generation of more distinguishable foreground and background features, eventually leading to better localization. Thanks to the FM and BM branches, from which we get the attentionally-pooled foreground x_{fg} (Eq. 5) and background x_{bg} (Eq. 8) feature representations, respectively, we insert the foreground-background separation loss between the FM and BM branches (Fig. 4) to learn separable foreground and background feature representations, as follows:

$$L_{fg-bg} = \max(0, \cos(x_{fg}, x_{bg})) \tag{18}$$

D. Temporal Action Localization in Inference

During the inference, given a test video, we first apply the threshold on video level prediction $\tilde{p}^{FM}_{fg} \in R^{(C+1)}$ of the FM branch and select the classes whose confidence scores in $\tilde{p}^{FM}_{fg}$ are above 0.1. Then, following the literature [16], [17], [20], for the selected classes, we apply a set of thresholds on the class activation scores S_{cs} of the CM branch to get the candidate action proposals. Finally, we perform class-wise Non-Maximum-Suppression (NMS) to retain the highly overlapped action proposals as the final localization.

IV. Experiments

A. Datasets and Metrics

THUMOS14 [67]: The THUMOS14 dataset contains temporal annotations for 200 validation and 213 test videos from 20 action classes. As in [11], [12], [13], [15], [16], [17], [20], [24], [68], we use the validation and test sets for training and evaluating, respectively.

ActivityNet1.3 [69]: The ActivityNet1.3 dataset has annotations of 200 categories in 10,024 training and 4926 validation videos. As in [14], [15], [16], [17], [20], we use the training and validation sets to respectively train and evaluate.

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

B. Implementation Details

For the feature extraction, we divide an untrimmed video by sliding a non-overlapping temporal window of 16 frames for both RGB and Optic-flow, which are then fed into the spatial and flow streams of a pre-trained I3D [55] network to extract 1024 dimensional feature vectors for both streams, respectively. We separately train our FBA-Net for both RGB and flow streams, and collect the generated proposals from both streams during testing. By validation, we set hyper-parameters $\lambda_1 = 0.1, \lambda_2 = 0.1, \lambda_3 = 1.0, \lambda_4 = 0.1, \lambda_5 = 0.0001$.

TABLE I

Localization performance comparison with state-of-the-art methods in terms of mAP (%) under different IoU thresholds on the THUMOS14 test set. AVG indicates the average mAP at IoU Thresholds 0.1:0.1:0.7. $^+$ means the method utilizes additional weak supervision, e.g., the number of action instances in videos.

Supervision	Category	Method IoU →	0.1	0.2	0.3	0.4	0.5	0.6	0.7	AVG
Full		R-C3D [8], ICCV'17	54.5	51.5	44.8	35.6	28.9	-	-	-
		TAL-Net [7], CVPR'18	59.8	57.1	53.2	48.5	42.8	33.8	20.8	45.1
		GTAN [9], CVPR'19	69.1	63.7	57.8	47.2	38.8	-	-	-
		ContextLoc [57], ICCV'21	-	-	68.3	63.8	54.3	41.8	26.2	-
		RefactorNet [58], ICCV'21	-	-	70.7	65.4	58.6	47.0	32.1	-
Weak$^+$		STAR (I3D) [59], AAAI'19	68.8	60.0	48.7	34.7	23.0	-	-	-
		3C-Net (I3D) [42], ICCV'19	59.1	53.5	44.2	34.1	26.6	-	8.1	-
		BM (I3D) [21], ICCV'19	64.2	59.5	49.1	38.4	27.5	17.3	8.6	37.8
Weak	Metric learning-based method	STPN (I3D) [13], CVPR'18	52.0	44.7	35.5	25.8	16.9	9.9	4.3	27.0
		W-TALC (I3D) [12], ECCV'18	55.2	49.6	40.1	31.1	22.8	-	7.6	-
		RPN (I3D) [43], AAAI'20	62.3	57.0	48.2	37.2	27.9	16.7	8.1	36.8
		CoLA (I3D) [60], CVPR'21	66.2	59.5	51.5	41.9	32.2	22.0	13.1	40.9
		D2-Net (I3D) [56], ICCV'21	65.7	60.2	52.3	43.4	36.0	-	-	-
Weak	Erasing-based method	Step-by-step erasion [49], MM'18	45.8	39.0	31.1	22.5	15.9	-	-	-
		A2CL-PT (I3D) [44] ECCV'20	61.2	56.1	48.1	39.0	30.1	19.2	10.6	37.8
		ACM-BANet (I3D) [17], MM'20	64.6	57.7	48.9	40.9	32.3	21.9	13.5	40.0
Weak	Pseudo label-based method	RefineLoc (I3D) [50], ECCV'20	-	-	40.8	32.7	23.1	13.3	5.3	-
		DGAM (I3D) [61], CVPR'20	60.0	54.2	46.8	38.2	28.8	19.8	11.4	37.0
		EM-MIL (I3D) [51], ECCV'20	59.1	52.7	45.5	36.8	30.5	22.7	16.4	37.7
		TSCN (I3D) [52], ECCV'20	63.4	57.6	47.8	37.7	28.7	19.4	10.2	37.8
		TSCN [52] + UGCT (I3D) [53], CVPR'21	67.5	62.1	55.3	45.2	33.3	20.7	9.5	41.9
		BM [21] + UGCT (I3D) [53], CVPR'21	69.2	62.9	55.5	46.5	35.9	23.8	11.4	43.6
		DCC [62], CVPR'22	69.0	63.8	55.9	45.9	35.7	24.3	13.7	44.0
		ASM-Loc [63], CVPR'22	71.2	65.5	57.1	46.8	36.6	25.2	13.4	45.1
		RSKP (I3D) [64], CVPR'22	71.3	65.3	55.8	47.5	38.2	25.4	12.5	45.1
Weak	Multi-branch architecture-based method	CMCS (I3D) [14], CVPR'19	57.4	50.8	41.2	32.1	23.1	15.0	7.0	32.4
		BM (I3D) [21], ICCV'19	60.4	56.0	46.6	37.5	26.8	17.6	9.0	36.3
		BaS-Net (I3D) [16], AAAI'20	58.2	52.3	44.6	36.0	27.0	18.6	10.4	35.3
		HAM-Net (I3D) [54], AAAI'21	65.4	59.0	50.3	41.1	31.0	20.7	11.1	39.8
		ACS-Net (I3D) [65], AAAI'21	-	-	51.4	42.7	32.4	22.0	11.7	-
		UM (I3D) [15], AAAI'21	67.5	61.2	52.3	43.4	33.7	22.9	12.1	41.9
		FAC-Net (I3D) [20], ICCV'21	67.6	62.1	52.6	44.3	33.4	22.5	12.7	42.2
		ACM-Net (I3D) [66], TIP'21	68.9	62.7	55.0	44.6	34.6	21.8	10.8	42.6
		FBA-Net (Ours)	69.2	63.3	54.2	46.3	36.9	23.6	13.1	43.8
Weak	Multi-branch + Pseudo label	FBA-Net (Ours) + RSKP (I3D) [64]	71.9	65.8	56.7	48.6	39.3	26.4	14.2	46.1

C. Comparison with the State-of-the-art

Table I summarizes the performance on the THUMOS14 dataset for action localization methods in the past few years for the different levels of supervision. Since we explore the W-TAL problem, regarding the design choice of the W-TAL methods, we mainly separate them into four categories: (i) Metric learning-based methods: use different loss terms to separate the action-background features; (ii) Erasing-based methods: iteratively erase the most discriminative features to discover different action parts; (iii) Pseudo label-based methods: iteratively generate the snippet-wise pseudo labels from a W-TAL network and then use them to refine that W-TAL network to distinguish the foreground and background snippets; and (iv) Multi-branch network-based methods: parallelly process multiple branches by inserting additional regularization terms or using the weight-sharing strategy with different training objectives to make branches different or complementary. Since our method is a multi-branch architecture, we mainly compare our method with other latest state-of-the-art multi-branch architecture-based methods for a fair comparison. As shown in Table I, our FBA-Net achieves superior performance compared to other multi-branch architecture-based methods. At the same time, our FBA-Net also achieves promising performance compared to the metric learning-based, erasing-based, and many pseudo label-based methods. The performance of our method is slightly inferior compared to the latest pseudo label-based methods such as ASM-Loc [63] and RSKP [64]. The pseudo label-based methods usually first utilize an existing W-TAL network as the classification head to generate the pseudo labels, and then use them to refine that W-TAL network to improve the localization performance, while our FBA-Net achieves the comparable performance even without refining our FBA-Net with pseudo labels. However, motivated by the performance improvement of the pseudo label-based methods, we can apply the pseudo-labeling strategy to our multi-branch architecture to further improve the localization performance. More specifically, as shown in the last row of Table I, we incorporate the latest pseudo label-based RSKP [64] algorithm to refine our FBA-Net with pseudo labels, and we achieve further improvement and establish a new state-of-the-art performance on the THUMOS14 dataset.

Table II presents the performance of our algorithm on the validation set of ActivityNet1.3 dataset, showing the superior performance compared to other state-of-the-art multi-branch architecture and erasing-based methods, and the comparable performance compared to the pseudo label-based methods.

TABLE II

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

Supervision	Category	Method IoU →	0.5	0.75	0.95	AVG
Full		TAL-Net [7], CVPR'18	38.2	18.3	1.3	20.2
		BSN [41], ECCV'18	46.5	30.0	8.0	30.0
		GTAN [9], CVPR'19	52.6	34.1	8.9	34.3
Weak	Erasing-based method	A2CL-PT (I3D) [44], ECCV2020	36.8	22.0	5.2	22.5
		ACM-BANet (I3D) [17], MM2020	37.6	24.7	6.5	24.4
Weak	Pseudo label-based method	TSCN (I3D) [52], ECCV2020	35.3	21.4	5.3	21.7
		TSCN [52] + UGCT (I3D) [53], CVPR2021	38.1	21.2	5.4	22.8
		WSAL-BM [21] + UGCT (I3D) [53], CVPR2021	39.0	21.4	5.1	23.0
		DCC [62], CVPR2022	38.8	24.2	5.7	24.3
		FAC-Net [20] + RSKP (I3D) [64], CVPR2022	40.6	24.6	5.9	25.0
		ASM-Loc [63], CVPR2022	41.0	24.9	6.2	25.1
Weak	Multi-branch architecture-based method	CMCS (I3D) [14], CVPR2019	34.0	20.9	5.7	21.2
		BaS-Net (I3D) [16], AAAI2020	34.5	22.5	4.9	22.2
		UM (I3D) [15], AAAI2021	37.0	23.9	5.7	23.7
		FAC-Net (I3D) [20], ICCV2021	37.6	24.2	6.0	24.0
		FBA-Net (Ours)	38.0	24.8	6.7	24.6
Weak	Multi-branch + Pseudo label	FBA-Net (Ours) + RSKP (I3D) [64]	39.8	25.0	6.8	25.3

TABLE III

Ablation studies of different architectures on THUMOS14 dataset. L^{FM}_{fg} : The FM branch; L^{BM}_{bg} : The BM branch; L^{CM}_{fg} : The CM branch with the video-level prediction via foreground classifier; L^{CM}_{bg} : The CM branch with the video-level prediction via background classifier.

L^{FM}_{fg}	L^{BM}_{bg}	L^{CM}_{fg}	L^{CM}_{bg}	0.1	0.2	0.3	0.4	0.5
✓	✓	✓	✓	69.2	63.3	54.2	46.3	36.9
✗	✓	✓	✓	64.7	59.0	49.4	42.5	33.4
✓	✗	✓	✓	67.2	60.7	51.6	44.4	34.3
✓	✓	✗	✓	60.2	52.3	42.1	34.0	25.3
✓	✓	✓	✗	68.1	62.6	53.0	44.6	34.5

TABLE IV

Effectiveness of our foreground-background separation loss.

Method IoU →	0.1	0.2	0.3	0.4	0.5
FBA-Net without L_{fg-bg}	68.8	63.0	54.0	46.1	36.6
FBA-Net with L_{fg-bg}	69.2	63.3	54.2	46.3	36.9

Likewise, our FBA-Net with RSKP [64] surpasses the pseudo label-based methods and obtains a new state-of-the-art performance on ActivityNet1.3 dataset, on most metrics

D. Ablation Studies

Ablation studies on different branches: As shown in Table III, to systematically evaluate the contribution of each branch, we perform a number of ablation experiments on THUMOS14: (i) We apply our FBA-Net without L^{FM}_{fg} (i.e., without the foreground modeling (FM) branch); (ii) We perform the experiments without L^{BM}_{bg} (i.e., without the background modeling (BM) branch); (iii) We perform the experiment without L^{CM}_{fg} (i.e., without the video-level prediction via foreground classifier in the class-specific action and background modeling (CM) branch); and (iv) We perform the experiment without L^{CM}_{bg} (i.e., without the video-level prediction via background classifier in the CM branch). As shown in Table III, we can clearly see that each branch is contributing to our FBA-Net to improve the localization performance. Our FBA-Net achieves the best performance from the combination of all branches.

TABLE V

Performance of our FBA-Net regarding the combination factor (α in Eq. 2) on THUMOS14 dataset.

Combination factor (α)	0.1	0.2	0.3	0.4	0.5
α = 1	67.1	60.9	52.0	43.5	33.9
α = 0	68.0	62.2	53.2	45.3	35.6
α = 0.5	69.2	63.3	54.2	46.3	36.9

TABLE VI

Performance of our FBA-Net regarding different similarity measures on THUMOS14 dataset.

Similarity measure	0.1	0.2	0.3	0.4	0.5
Dot-product	67.4	61.1	51.8	43.4	33.1
Cosine similarity	69.2	63.3	54.2	46.3	36.9

Effectiveness of foreground-background separation loss: As shown in Table IV, we configure our FBA-Net without and with the foreground-background separation loss L_{fg-bg}, to check its effectiveness on THUMOS14 dataset. We find that the foreground-background separation loss improves the localization performance slightly, indicating that more future works are needed to generate more distinguishable foreground and background features.

Ablation studies on different combination factors for the foreground classifier: In Eq.2, a combination factor (α) is introduced to combine the class-agnostic classifier and the negative weights of the background classifier to define the foreground classifier. The performance of our network for different values of α's on THUMOS14 dataset is summarized in Table V. We find that the average fusion (i.e., α = 0.5) of the class-agnostic classifier and the negative weights of the background classifier to generate the foreground classifier leads to a larger improvement compared to the individual class-agnostic classifier (i.e., α = 1) or the negative weights of the background classifier (i.e., α = 0).

Effectiveness cosine similarity over dot-product: We utilize the cosine rather than the commonly used dot product to calculate the similarity. As shown in Table VI, we find that the

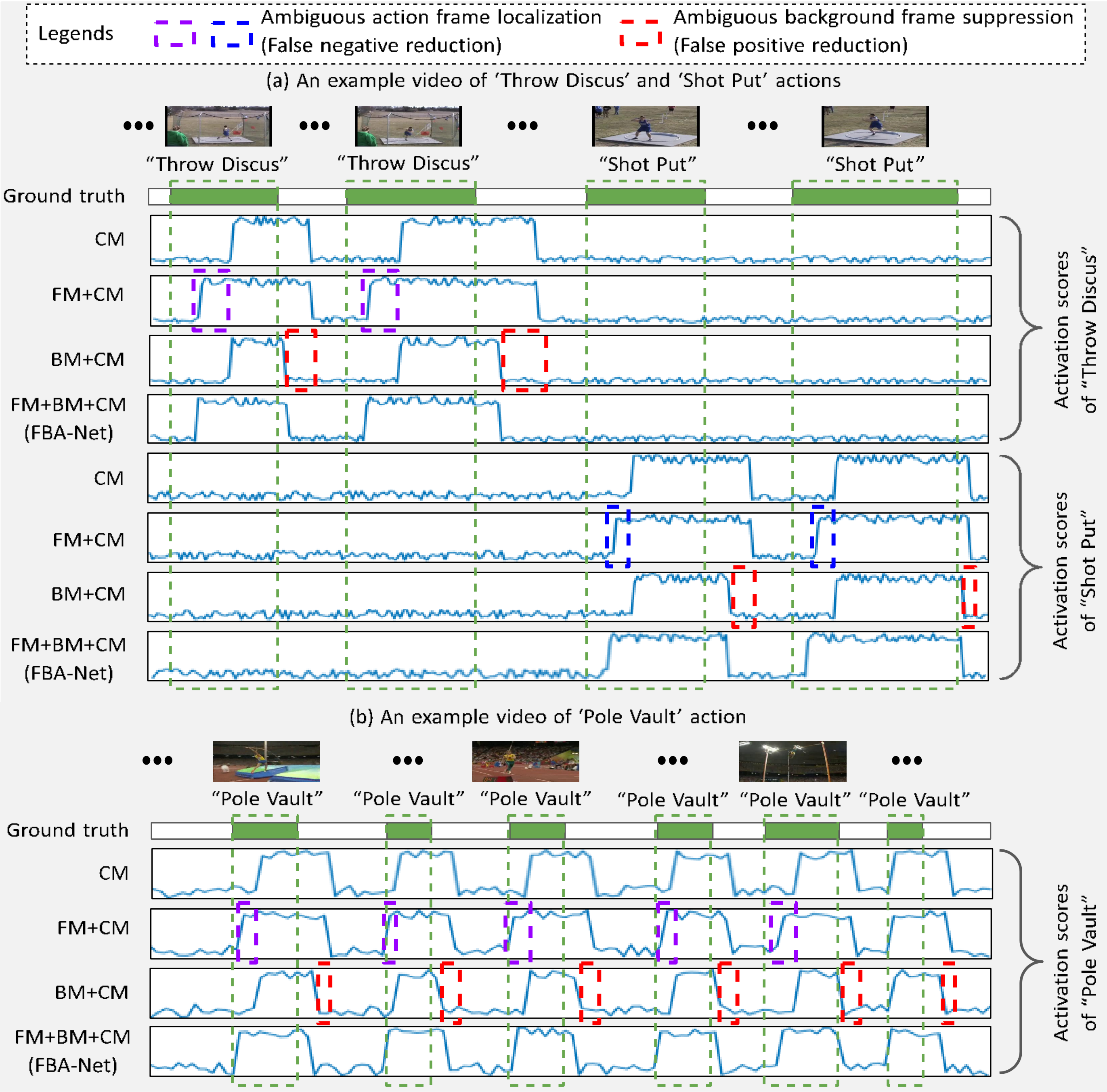


Fig. 5. Qualitative results. With many background frames, the video (a) contains multiple instances of "Throw Discus" and "Shot Put" actions, and the video (b) contains multiple instances of the "Pole Vault" action. The AM branch coarsely localizes the action instances of different classes. The collaboration between FM and CM branches reduces the false negative rate by localizing more actual-action-related frames, while the collaboration between BM and CM branches reduces the false positive rate by suppressing more background frames. The FBA-Net that comprises all three branches reduces both the false positive and false negative rates.

cosine similarity obtains much higher performance compared to the dot product for our action localization task. The main reason on the performance improvement comes from that, the dot product is magnitude sensitive while the cosine similarity ignores the magnitude of the feature and ensures stable classification scores, eventually, leading to better performance.

Model complexity: Although our FBA-Net consists of three branches, our FBA-Net only learns the parameters of a class-specific classifier $W_{cs} \in R^{(C+1)\times D}$ and a class-agnostic classifier $w_{ca} \in R^{1\times D}$ with a commonly used feature embedding module. Hence, the model complexity does not increase. With a single Tesla V100 GPU, it only takes about 15 and 40 minutes to train our FBA-Net on THUMOS14 and ActivityNet1.3, respectively, while it also takes almost a similar time to train each branch independently.

E. Qualitative Analysis

We visualize some qualitative results in Fig. 5, where the Fig. 5(a) contains multiple instances of multiple actions ("Throw Discus" and "Shot Put"), and the Fig. 5(b) contains multiple instances of a single action ("Pole Vault"), with many background frames. We show the activation scores of the predicted classes for different branches in our FBA-Net. We show the activation scores of the predicted classes for different branches in our FBA-Net:

- CM: The CM branch localizes the discriminative action frames and suppresses the discriminative background frames, yielding a coarse localization with many false positives and false negatives.
- FM+CM: The FM branch collaborates with the CM branch to reduce the false negative rate by localizing more actual-action-related frames, i.e., the FM+CM localizes both the discriminative and ambiguous action frames.
- BM+CM: The BM branch collaborates with the CM branch to reduce the false positive rate by suppressing more background frames, i.e., BM+CM suppresses both the discriminative and ambiguous background frames.
- FM+BM+CM (FBA-Net): The FBA-Net that comprises all the FM, BM, and CM branches reduces both the false negative and false positive rates by localizing both the discriminative and ambiguous action frames, and suppressing both the discriminative and ambiguous background frames, respectively.

V. Conclusion

We introduced a W-TAL approach, called FBA-Net, that comprises a foreground modeling (FM) branch, a background modeling (BM) branch, and a class-specific action and background modeling (CM) branch. The collaboration within the CM branch learns to highlight the video frames related to various actions, and separate the background frames from the action frames. The collaboration between FM and CM branches reduces the false negative rate by localizing different actual-action-related (i.e., both the discriminative and ambiguous action) frames in a video as foreground, while the collaboration between BM and CM branches reduces the false positive rate by suppressing both the discriminative and ambiguous background frames. Furthermore, the collaboration between FM and BM enforces more effective foreground-background separation. The experimental results show that our FBA-Net achieves superior localization performances on THUMOS14 and ActivityNet1.3 datasets.

Acknowledgment

This project was supported by National Science Foundation grants CMMI-1954548 and ECCS-2025929.

References

[1] S. Vishwakarma and A. Agrawal, "A survey on activity recognition and behavior understanding in video surveillance," The Visual Computer, 2013.

[2] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization," in CVPR, 2012.

[3] A. Stoian, M. Ferecatu, J. Benois-Pineau, and M. Crucianu, "Fast action localization in large-scale video archives," IEEE-TCSVT, 2015.

[4] C. Yeo, P. Ahammad, K. Ramchandran, and S. S. Sastry, "High-speed action recognition and localization in compressed domain videos," IEEE-TCSVT, 2008.

[5] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in CVPR, 2016.

[6] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in CVPR, 2017.

[7] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the faster r-cnn architecture for temporal action localization," in CVPR, 2018.

[8] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," in ICCV, 2017.

[9] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, and T. Mei, "Gaussian temporal awareness networks for action localization," in CVPR, 2019.

[10] C. Zhao, A. K. Thabet, and B. Ghanem, "Video self-stitching graph network for temporal action localization," in ICCV, 2021.

[11] L. Wang, Y. Xiong, D. Lin, and L. Van Gool, "Untrimmednets for weakly supervised action recognition and detection," in CVPR, 2017.

[12] S. Paul, S. Roy, and A. K. Roy-Chowdhury, "W-talc: Weakly-supervised temporal activity localization and classification," in ECCV, 2018.

[13] P. Nguyen, T. Liu, G. Prasad, and B. Han, "Weakly supervised action localization by sparse temporal pooling network," in CVPR, 2018.

[14] D. Liu, T. Jiang, and Y. Wang, "Completeness modeling and context separation for weakly supervised temporal action localization," in CVPR, 2019.

[15] P. Lee, J. Wang, Y. Lu, and H. Byun, "Weakly-supervised temporal action localization by uncertainty modeling," in AAAI, 2021.

[16] P. Lee, Y. Uh, and H. Byun, "Background suppression network for weakly-supervised temporal action localization." in AAAI, 2020.

[17] M. Moniruzzaman, Z. Yin, Z. He, R. Qin, and M. C. Leu, "Action completeness modeling with background aware networks for weakly-supervised temporal action localization," in ACMMM, 2020.

[18] T. Zhao, J. Han, L. Yang, B. Wang, and D. Zhang, "Soda: Weakly supervised temporal action localization based on astute background response and self-distillation learning," IJCV, 2021.

[19] L. Huang, Y. Huang, W. Ouyang, and L. Wang, "Two-branch relational prototypical network for weakly supervised temporal action localization," IEEE-TPAMI, 2021.

[20] L. Huang, L. Wang, and H. Li, "Foreground-action consistency network for weakly supervised temporal action localization," in ICCV, 2021.

[21] P. X. Nguyen, D. Ramanan, and C. C. Fowlkes, "Weakly-supervised action localization with background modeling," in ICCV, 2019.

[22] L. Yang, J. Han, T. Zhao, T. Lin, D. Zhang, and J. Chen, "Background-click supervision for temporal action localization," IEEE-TPAMI, 2021.

[23] B. Wang, X. Zhang, and Y. Zhao, "Exploring sub-action granularity for weakly supervised temporal action localization," IEEE-TCSVT, 2021.

[24] W. Luo, T. Zhang, W. Yang, J. Liu, T. Mei, F. Wu, and Y. Zhang, "Action unit memory network for weakly supervised temporal action localization," in CVPR, 2021.

[25] W. Sun, R. Su, Q. Yu, and D. Xu, "Slow motion matters: A slow motion enhanced network for weakly supervised temporal action localization," IEEE-TCSVT, 2022.

[26] R. Girshick, "Fast r-cnn," in ICCV, 2015.

[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in NeurIPS, 2015.

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in CVPR, 2016.

[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in ECCV, 2016.

[30] Z. Jie, W. F. Lu, S. Sakhavi, Y. Wei, E. H. F. Tay, and S. Yan, "Object proposal generation with fully convolutional networks," IEEE-TCSVT, 2016.

[31] K. Kang, H. Li, J. Yan, X. Zeng, B. Yang, T. Xiao, C. Zhang, Z. Wang, R. Wang, X. Wang et al., "T-cnn: Tubelets with convolutional neural networks for object detection from videos," IEEE-TCSVT, 2017.

[32] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in ICCV, 2017.

[33] K. Yang, P. Qiao, D. Li, S. Lv, and Y. Dou, "Exploring temporal preservation networks for precise temporal action localization," in AAAI, 2018.

[34] J. Yuan, B. Ni, X. Yang, and A. A. Kassim, "Temporal action localization with pyramid of score distribution features," in CVPR, 2016.

[35] H. Eun, S. Lee, J. Moon, J. Park, C. Jung, and C. Kim, “Srg: Snippet relatedness-based temporal action proposal generator,” IEEE-TCSVT, 2019.

[36] L. Xu, X. Wang, W. Liu, and B. Feng, “Cascaded boundary network for high-quality temporal action proposal generation,” IEEE-TCSVT, 2019.

[37] T. Lin, X. Zhao, and H. Su, “Joint learning of local and global context for temporal action proposal generation,” IEEE-TCSVT, 2019.

[38] J. Wang, W. Wang, and W. Gao, “Fast and accurate action detection in videos with motion-centric attention model,” IEEE-TCSVT, 2018.

[39] T. Lin, X. Zhao, and Z. Shou, “Single shot temporal action detection,” in ACMMM, 2017.

[40] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Qiu Chen, “Temporal context network for activity localization in videos,” in ICCV, 2017.

[41] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, “Bsn: Boundary sensitive network for temporal action proposal generation,” in ECCV, 2018.

[42] S. Narayan, H. Cholakkal, F. S. Khan, and L. Shao, “3c-net: Category count and center loss for weakly-supervised action localization,” in ICCV, 2019.

[43] L. Huang, Y. Huang, W. Ouyang, and L. Wang, “Relational prototypical network for weakly supervised temporal action localization,” in AAAI, 2020.

[44] K. Min and J. J. Corso, “Adversarial background-aware loss for weakly-supervised temporal activity localization,” in ECCV, 2020.

[45] Q. Hou, P. Jiang, Y. Wei, and M.-M. Cheng, “Self-erasing network for integral object attention,” in NeurIPS, 2018.

[46] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, “Object region mining with adversarial erasing: A simple classification to semantic segmentation approach,” in CVPR, 2017.

[47] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. S. Huang, “Adversarial complementary learning for weakly supervised object localization,” in CVPR, 2018.

[48] K. K. Singh and Y. J. Lee, “Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization,” in ICCV, 2017.

[49] J.-X. Zhong, N. Li, W. Kong, T. Zhang, T. H. Li, and G. Li, “Step-by-step erasion, one-by-one collection: a weakly supervised temporal action detector,” in ACMMM, 2018.

[50] A. Pardo, H. Alwassel, F. Caba, A. Thabet, and B. Ghanem, “Refineloc: Iterative refinement for weakly-supervised action localization,” in WACV, 2021.

[51] Z. Luo, D. Guillory, B. Shi, W. Ke, F. Wan, T. Darrell, and H. Xu, “Weakly-supervised action localization with expectation-maximization multi-instance learning,” in ECCV, 2020.

[52] Y. Zhai, L. Wang, W. Tang, Q. Zhang, J. Yuan, and G. Hua, “Two-stream consensus network for weakly-supervised temporal action localization,” in ECCV, 2020.

[53] W. Yang, T. Zhang, X. Yu, T. Qi, Y. Zhang, and F. Wu, “Uncertainty guided collaborative training for weakly supervised temporal action detection,” in CVPR, 2021.

[54] A. Islam, C. Long, and R. Radke, “A hybrid attention mechanism for weakly-supervised temporal action localization,” in AAAI, 2021.

[55] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in CVPR, 2017.

[56] S. Narayan, H. Cholakkal, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, “D2-net: Weakly-supervised action localization via discriminative embeddings and denoised activations,” in ICCV, 2021.

[57] Z. Zhu, W. Tang, L. Wang, N. Zheng, and G. Hua, “Enriching local and global contexts for temporal action localization,” in ICCV, 2021.

[58] K. Xia, L. Wang, S. Zhou, N. Zheng, and W. Tang, “Learning to refactor action and co-occurrence features for temporal action localization,” in CVPR, 2022.

[59] Y. Xu, C. Zhang, Z. Cheng, J. Xie, Y. Niu, S. Pu, and F. Wu, “Segregated temporal assembly recurrent networks for weakly supervised multiple action detection,” in AAAI, 2019.

[60] C. Zhang, M. Cao, D. Yang, J. Chen, and Y. Zou, “Cola: Weakly-supervised temporal action localization with snippet contrastive learning,” in CVPR, 2021.

[61] B. Shi, Q. Dai, Y. Mu, and J. Wang, “Weakly-supervised action localization by generative attention modeling,” in CVPR, 2020.

[62] J. Li, T. Yang, W. Ji, J. Wang, and L. Cheng, “Exploring denoised cross-video contrast for weakly-supervised temporal action localization,” in CVPR, 2022.

[63] B. He, X. Yang, L. Kang, Z. Cheng, X. Zhou, and A. Shrivastava, “Asm-loc: Action-aware segment modeling for weakly-supervised temporal action localization,” in CVPR, 2022.

[64] L. Huang, L. Wang, and H. Li, “Weakly supervised temporal action localization via representative snippet knowledge propagation,” in CVPR, 2022.

[65] Z. Liu, L. Wang, Q. Zhang, W. Tang, J. Yuan, N. Zheng, and G. Hua, “Acsnet: Action-context separation network for weakly supervised temporal action localization,” in AAAI, 2021.

[66] S. Qu, G. Chen, Z. Li, L. Zhang, F. Lu, and A. Knoll, “Acm-net: Action context modeling network for weakly-supervised temporal action localization,” IEEE-TIP, 2021.

[67] Y.-G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, “Thumos challenge: Action recognition with a large number of classes,” 2014.

[68] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang, “Autoloc: Weakly-supervised temporal action localization in untrimmed videos,” in ECCV, 2018.

[69] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, “Activitynet: A large-scale video benchmark for human activity understanding,” in CVPR, 2015.

Md Moniruzzaman received the B.S. degree from the Department of Electronics and Communication Engineering, Khulna University of Engineering and Technology, Bangladesh. He is currently working toward his Ph.D. degree at the Department of Computer Science, Stony Brook University, NY, USA. His current research interests include human action anticipation, human action recognition, temporal action localization, and human pose estimation.

Zhaozheng Yin is a SUNY Empire Innovation Associate Professor at Stony Brook University. He is affiliated with the AI Institute, Department of Biomedical Informatics, and Department of Computer Science. His group has been working on Biomedical Image Analysis, Computer Vision, and Machine Learning. Zhaozheng is an IEEE senior member and he has served as Area Chair for CVPR, ECCV, MICCAI and WACV, and an Associate Editor for IEEE-TCSVT and JVCI.