Action-aware Masking Network with Group-based Attention for Temporal Action Localization

Tae-Kyung Kang[1], Gun-Hee Lee[2], Kyung-Min Jin[1], and Seong-Whan Lee[1]

[1]Dept. of Artificial Intelligence, Korea University, South Korea
[2]Dept. of Computer Science and Engineering, Korea University, South Korea
{tk_kang, gunhlee, km_jin, sw.lee}@korea.ac.kr

Abstract

Temporal Action Localization (TAL) is a significant and challenging task that searches for subtle human activities in an untrimmed video. To extract snippet-level video features, existing TAL methods commonly use video encoders pre-trained on short-video classification datasets. However, the snippet-level features can incur ambiguity between consecutive frames due to short and poor temporal information, disrupting the precise prediction of action instances. Several methods incorporating temporal relations have been proposed to mitigate this problem; however, they still suffer from poor video features. To address this issue, we propose a novel temporal action localization framework called an Action-aware Masking Network (AMNet). Our method simultaneously refines video features using action-aware attention and considers inherent temporal relations using self-attention and cross-attention mechanisms. First, we present an Action Masking Encoder (AME) that generates an action-aware mask to represent positive characteristics, which is then used to refine snippet-level features to be more salient around actions. Second, we design a Group Attention Module (GAM), which models relations of temporal information and exchanges mutual information by dividing the features into two groups, i.e., long and short-groups. Extensive experiments and ablation studies on two primary benchmark datasets demonstrate the effectiveness of AMNet, and our method achieves state-of-the-art performances on THUMOS-14 and ActivityNet1.3.

1. Introduction

Temporal Action Localization (TAL) is a core task in video understanding. TAL has attracted attention recently, which can be extended to various video-related studies [45], *e.g.*, video retrieval [16, 11], video surveillance [46, 7], and video summarization [43, 10]. Given an untrimmed video, TAL aims to predict start time, end time, and category of actions. It is a challenging task because classification and localization are conducted simultaneously to find complex and vague action instances in the long untrimmed video.

In TAL, various suitable methods have recently been proposed, with most approaches [35, 39, 14, 48, 17] commonly relying on pre-trained video encoders. Specifically, an untrimmed video is split into snippets and features are extracted from every snippet. Then, with the extracted features, the proposed action detection model is used to predict action boundaries and categories.

Existing methods are diverse and have remarkable performances but do not fully utilize inherent semantic information due to limited feature representation. In particular, the video encoders, pre-trained for video-level classification, are not optimized in TAL, and so, the extracted features with snippet-level videos do not provide sufficient contextual information. This is because the snippet-level video contains around 8 to 32 frames. Assuming that the video is 30 fps, it is approximately 0.27 to 1.07 seconds. This limitation causes ambiguity between consecutive frames resulting in the TAL model not being able to distinguish clearly between frames of action and background, which can hinder subsequent detection and classification processes. This ambiguity not only interrupts the precise prediction of action boundaries but also results in inconsistency between classification and localization. Even if the predicted temporal action boundary is exact, inaccurate classification scores can negatively affect the detection performance by non-maximum suppression (NMS) [4].

In this paper, we propose an **Action-aware Masking Network (AMNet)** to address the scene ambiguity through action-aware attention and self-attention. We first characterize original snippet-level features with positive and negative components based on the ground truth in the training stage. Here, positive and negative parts denote the areas of actions and background, respectively. We then train an action-aware attention mask to represent a positive component by maintaining a considerable embedded distance from the negative. Using this mask, we refine the original representation to make it more pronounced in the action areas, considering inherent semantic information. Furthermore, to utilize the refined feature, we divide the feature into multi-scale features and apply a self and cross-attention mechanism.

Our proposed framework consists of three main components: (i) an Action Masking Encoder (AME), (ii) a Group Attention Module (GAM), and (iii) prediction heads such as class, boundary, and matching score heads. The AME generates the action-aware mask from the video feature, masking it as a residual-alike approach. The masked feature benefits temporal action information, maintaining existing feature information. The GAM contains a feature pyramid network that generates multi-scale features to cover action detection of various lengths. Existing methods [25, 42] process each multi-scale feature independently. However, this approach cannot fully utilize a multi-scale structure with different inherent temporal information. As the feature with a long temporal dimension tends to focus on the local context and the feature with a short temporal dimension tends to focus on the global context, we combine the multi-scale features into two groups, *i.e.*, long and short groups. We then conduct cross-attention between the two groups to compensate for the lack of knowledge. Our prediction heads consist of class, boundary, and matching score heads. The matching score head generates matching scores, which are further multiplied by the classification scores.

Our proposed AMNet demonstrates effectiveness by conducting extensive experiments on two benchmark datasets: THUMOS-14 [18] and ActivityNet1.3 [5]. As a result, we achieve state-of-the-art performance, and our contributions can be summarized as follows:

- We propose an AMNet, in which the AME generates an action-aware mask that refines the snippet-level video feature by applying action-aware attention to address the scene ambiguity. It emphasizes the action area of the feature by masking the original video feature.

- We design a GAM, which models the inherent temporal relation by combining multi-scale features into two groups and applying cross-attention.

- We conduct extensive experiments, and our method outperforms other state-of-the-art methods on two primary datasets, *i.e.*, THUMOS-14 and ActivityNet1.3.

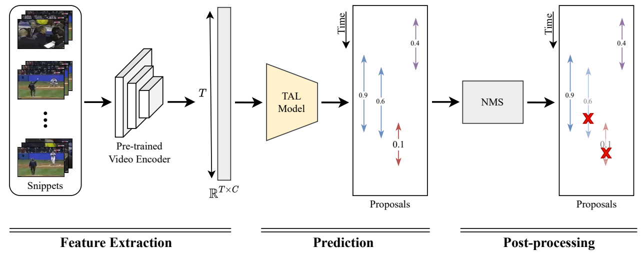

Figure 1. The overall pipeline of the typical TAL. After an video encoder extracts a video feature, the TAL model generates the proposals that consist of the temporal action boundaries and classification scores. In the post-processing stage (right of the figure), NMS suppresses proposals with a lower classification score or overlapping with other proposals over the threshold.

2. Related Work

2.1. Action Recognition

Action recognition [1, 37] has been actively studied for a long time as an area of pattern recognition [36, 23, 22, 24, 15] and a fundamental task for TAL. The traditional action recognition methods can be divided into skeleton-based methods (Shift-GCN [9]), and video-based methods (TSN [41] and I3D [6]). The I3D model, which is a two-stream inflated 3D convolutional network utilizing RGB and optical flow, is most prevalent in TAL. The I3D increases the receptive fields of 2D CNN by inflating the convolution filters and kernel sizes of pooling, thereby considering temporal dimensions. We adopted the I3D model pre-trained on the Kinetics dataset [21] because of its superior ability for action recognition. However, the snippet-level video features extracted by the video encoder can have limited temporal information because of the short-term snippet-level videos. Our proposed method focuses on mitigating this problem.

2.2. Temporal Action Localization

Unlike action recognition, the datasets for TAL are untrimmed long videos. Furthermore, TAL conducts two tasks simultaneously, namely classification and localization of actions. Overall TAL process can be divided into three steps: (i) feature extraction, (ii) prediction using the TAL model, and (iii) post-processing using Soft-NMS [4], as shown in Fig. 1. Most TAL methods [39, 20, 44] utilize the pre-trained action recognition model as the backbone architecture to extract video features. With these extracted features, the TAL methods focus on the prediction stage. However, we argue that offline snippet-level features can be sub-optimal for localization actions because of insufficient temporal knowledge. To address this issue, we refine the snippet-level video features by conducting action-aware attention with an action-aware mask generated by the proposed AME.

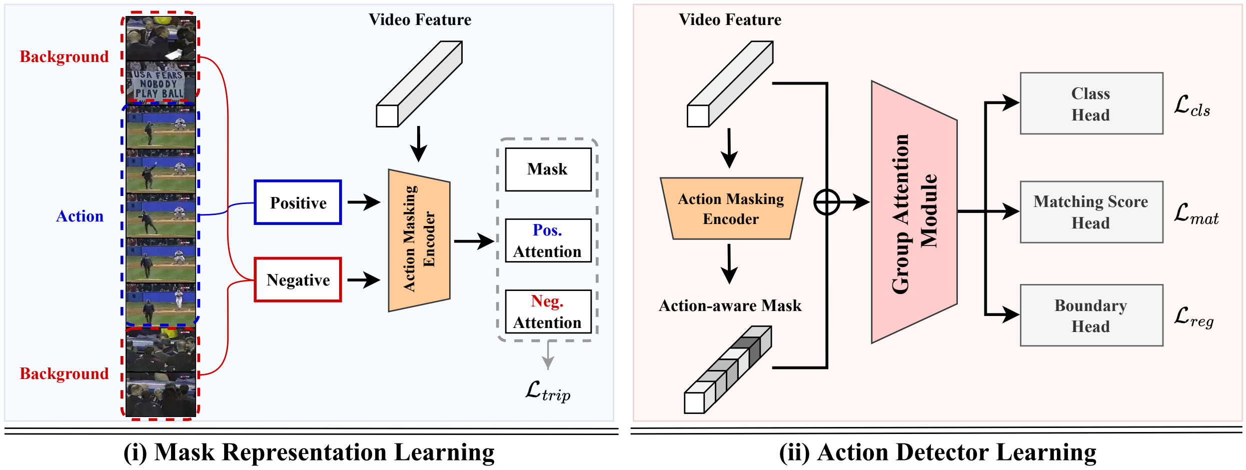

Figure 2. **Action-aware Masking Network (AMNet):** Our training process can be divided into mask representation learning (left) and action detector learning (right). In mask representation learning, we first divide the snippet-level features into positive and negative components, where the positives are inside, and the negatives are outside of the ground truth's temporal boundary. Then, an Action Masking Encoder (AME) is trained to represent the positive parts using a triplet loss. Next, the Group Attention Module (GAM) and predictors generate the final outputs using masked features. Finally, each loss is calculated, *i.e.*, class, boundary, and matching losses.

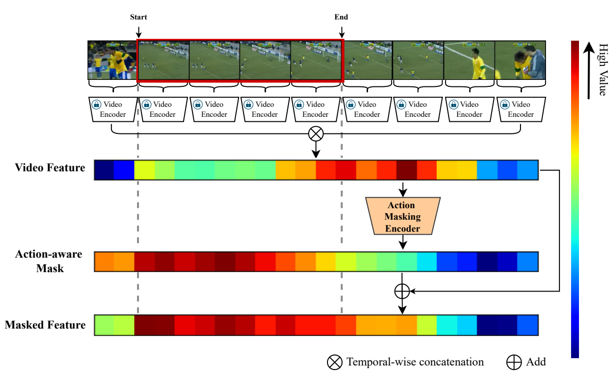

Figure 3. **Action Masking Encoder (AME)** generates an action-aware mask from the video feature. Each feature in the figure denotes a mean of the channels. As shown in this figure, the masked feature is more salient around the action than the video feature.

2.3. Transformer

Since the emergence of the transformer [40] in the area of Natural Language Processing (NLP), transformer-based architectures have been actively studied in computer vision for tasks such as image [12] and video processing [3, 19, 31]. The receptive field of typical convolutional networks is limited due to filter sizes. In contrast, the transformer effectively utilizes global dependencies with multi-head self-attention, thus demonstrating superior performance. Considering this, we also adopt the encoder of the transformer to model the relation between temporal locations. Furthermore, we combine multi-scale features into long and short groups and conduct cross-attention to model the dependency on each group.

3. Proposed Method

In this section, we present a novel TAL framework called an **Action-aware Masking Network (AMNet)**, which consists of three main components: an **Action Masking Encoder (AME)**, a **Group Attention Module (GAM)**, and prediction heads. Specifically, we refine video features with an action-aware mask generated through AME and model each relation of multi-scale features by grouping through GAM. In training, our method is asynchronously processed in two steps; therefore we first explain (i) mask representation learning. We then introduce (ii) action detector learning. The overall pipeline of our method is shown in Fig. 2.

3.1. Problem Settings and Feature Extraction

Given an untrimmed video, TAL aims to predict the actions' start time, end time, and confidence score. As a first step, we extract the video feature F for each snippet-level video, which contains a few frames (*e.g.*, 16 frames), using the pre-trained video encoder [6]. The extracted video feature can be denoted as $F \in \mathbb{R}^{T \times C}$, where T and C are temporal dimension and channels.

3.2. Mask Representation Learning

In mask representation learning, we train the AME, generating an action-aware mask to refine the video feature F through action-aware attention. Specifically, according to the ground truth, we divide the video feature into positive (Action) and negative (Background) components as shown in Fig. 2. Then, we collect and concatenate the corresponding snippet-level features along the temporal dimension. The positive $F_{pos} \in \mathbb{R}^{T_P \times C}$ and negative features

$F_{neg} \in \mathbb{R}^{T_N \times C}$ have T_P and T_N lengths of which the sum equals T. Next, AME generates the mask, positive and negative attention from the video feature, respectively. We can formulate it as:

$$\mathbf{AME}(x) = \text{Conv1d}(\{\sigma(\varepsilon(\text{Conv1d}(x)))\}_{\times K}),$$
$$\begin{cases} F_{mask} = \mathbf{AME}(F) \\ \hat{F}_{pos} = \mathbf{AME}(F_{pos}) \\ \hat{F}_{neg} = \mathbf{AME}(F_{neg}), \end{cases} \tag{1}$$

where σ, ε, and K denote an activation, normalization function, and the number of layers, respectively. We note that the positive and negative features must have orthogonal properties. Furthermore, the mask must be able to represent each attention. To this end, we adopt a triplet loss [38] widely used for feature representation learning or clustering. To briefly explain it, we set the mask F_{mask} to anchor and find the Euclidean distance of the embedded anchor, positive and negative, as follows:

$$\begin{aligned} d_{pos} &= \left\| F_{mask} - \hat{F}_{pos} \right\|_2^2, \\ d_{neg} &= \left\| F_{mask} - \hat{F}_{neg} \right\|_2^2. \end{aligned} \tag{2}$$

Here, we intend to minimize d_{pos} and maximize d_{neg}. So, the triplet loss $\mathcal{L}_{trip}$ can be formulated as:

$$\mathcal{L}_{trip} = [d_{pos} - d_{neg} + \alpha]_+, \tag{3}$$

where α denotes a margin enforced between positive and negative pairs. With this loss, we can obtain the action-aware mask with AME that encodes the feature salient to the positive one.

3.3. Action Detector Learning

In action detector learning, we start to train our AM-Net in earnest. First, we introduce a detailed refinement of the video feature process using AME. Next, we present the structure of the GAM for modeling inherent temporal relations between long and short groups. Finally, we explain about three prediction heads: (i) class head, (ii) boundary head, and (iii) matching score head. The details are explained below.

Refinement of Video Feature To obtain the optimal feature for TAL that has salient values around the action area, we first generate an action-aware mask using the AME. Next, we obtain a masked feature by action-aware attention, conducting the residual-alike operation as follows:

$$\begin{aligned} F_{mask} &= \mathbf{AME}(F), \\ \hat{F} &= F + F_{mask}, \end{aligned} \tag{4}$$

where $F \in \mathbb{R}^{T \times C}$ and $F_{mask} \in \mathbb{R}^{T \times C}$ denote the video feature and the action-aware mask, respectively. After conducting action-aware attention, we can observe that the masked feature is refined to be salient around the action area, as shown in Fig. 3. Afterward, the masked feature $\hat{F} \in \mathbb{R}^{T \times C}$ is used for input of GAM.

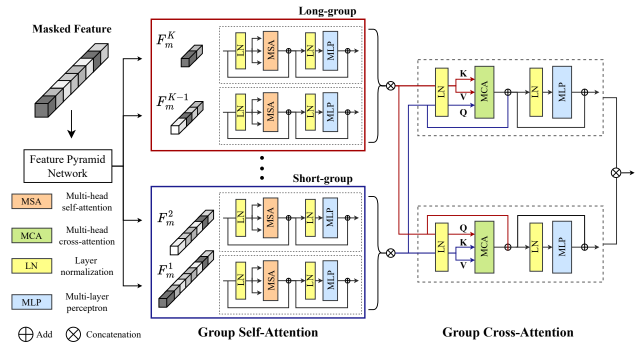

Figure 4. **Group Attention Module (GAM)** consists of three components: (i) feature pyramid network, (ii) self-attention modules, and (iii) cross-attention modules. We combine the multi-scale features generated from the feature pyramid network into two groups, *i.e.*, large and short groups based on the temporal dimension.

Group Attention Module (GAM) To fully utilize the inherent semantic knowledge of the masked feature $\hat{F}$, we build a GAM that models the temporal relations of each time step, as shown in Fig. 4. To obtain temporal action boundaries of various lengths, the masked feature $\hat{F}$ is divided into K multi-scale features $\{F_m^i \in \mathbb{R}^{T_i \times C}\}_{i=1}^K$ by the feature pyramid network, which consists of 1D CNNs. Each multi-scale feature has different temporal dimensions, reduced by half, respectively. Afterward, we conduct self-attention on each multi-scale feature $(F_m^1, F_m^2, \cdots, F_m^K)$ to model the relation between each temporal location. First, the multi-scale features are projected into query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$, respectively as follows:

$$\left\{ \begin{array}{l} \mathbf{Q}_i = W_q^i \cdot F_m^i \\ \mathbf{K}_i = W_k^i \cdot F_m^i \\ \mathbf{V}_i = W_v^i \cdot F_m^i \end{array} \right| \; i \in \{1, 2, \cdots, K\}, \tag{5}$$

where W denotes a learnable weight that projects the feature into query, key, and value. With these projected features, we conduct a self-attention operation, which is formulated as follows:

$$att_i = \text{softmax}(\frac{\mathbf{Q}_i \mathbf{K}_i^{\mathrm{T}}}{\sqrt{D}})\mathbf{V}_i, \tag{6}$$

where D denotes the channel of each attention head. The channel D is calculated as $\frac{C}{N_h}$ where N_h is the number of attention heads.

After conducting the self-attention operation, we combine the multi-scale features into two groups: long and short groups, based on the length of the temporal dimension as follows:

$$
\begin{aligned}
G_{short} &= [g(F_m^1), \cdots, g(F_m^{\frac{K}{2}})], \\
G_{long} &= [g(F_m^{\frac{K}{2}+1}), \cdots, g(F_m^K)],
\end{aligned} \tag{7}
$$

where $[\cdot]$ and $g(\cdot)$ denote temporal-wise concatenation and self-attention, respectively. Note that the features with various temporal dimensions benefit from generating proposals of various lengths. Specifically, the feature with a longer temporal dimension, which focuses on local context, tends to generate relatively short action boundaries. In contrast, the feature with a shorter temporal dimension, which focuses on the global context, tends to generate relatively long action boundaries. It is because the predicted absolute distance values of start and end from specific time steps have a lower percentage in the long temporal dimension than in the short temporal dimension, and vice versa. So, we conduct cross-attention between two groups (long and short) to compensate for the lack of semantic knowledge as follows:

$$
\begin{aligned}
&\begin{array}{l|l}
\mathbf{Q}_S = W_{cq} \cdot G_{short}, & \mathbf{Q}_L = W_{cq} \cdot G_{long}, \\
\mathbf{K}_S = W_{ck} \cdot G_{short}, & \mathbf{K}_L = W_{ck} \cdot G_{long}, \\
\mathbf{V}_S = W_{cv} \cdot G_{short}, & \mathbf{V}_L = W_{cv} \cdot G_{long},
\end{array} \\
&G_{L\to S} = \mathbf{MLP}(\epsilon(\mathbf{MCA}(\mathbf{K}_S, \mathbf{V}_S, \mathbf{Q}_L))), \\
&G_{S\to L} = \mathbf{MLP}(\epsilon(\mathbf{MCA}(\mathbf{K}_L, \mathbf{V}_L, \mathbf{Q}_S))),
\end{aligned} \tag{8}
$$

where ϵ, **MLP** and **MCA** denote layer normalization, multi-layer perceptron, and multi-head cross-attention, respectively. We then reshape each $G_{L\to S}$ and $G_{S\to L}$ as the shape of the original multi-scale features denoted as $\{\hat{F}_i \in \mathbb{R}^{T_i \times C}\}_{i=1}^K$ before predicting the final outputs.

Prediction Heads General TAL methods predict two outputs: temporal boundary and action category. However, these methods unfortunately often neglect the inconsistency between classification and localization derived from scene ambiguity, which is one of the main factors that cause performance degradation. Therefore, we add auxiliary output, the matching score, to make the confidence scores robust against incorrect suppression by Soft-NMS [4] in inference time.

Our prediction heads (*i.e.*, class, boundary, and matching score heads) are composed with 1D convolutional layers. They use the main block of the same structure as follows:

$$
\mathbf{Block}(x) = \{\sigma(\epsilon(\mathrm{Conv1d}(x)))\}_{\times K}, \tag{9}
$$

where σ, ϵ, and K denote an activation function, layer normalization, and the number of layers, respectively. The final outputs, such as the temporal boundaries, confidence scores, and matching score, are generated as follows:

$$
\begin{aligned}
\hat{y}_i &= \mathrm{FC}(\mathbf{Block}(\hat{F}_i)), \\
\hat{\mathbf{B}}_i &= \sigma(\mathrm{FC}(\mathbf{Block}(\hat{F}_i)) \times \omega_B), \\
\hat{m}_i &= \mathrm{FC}(\mathbf{Block}(\hat{F}_i)) \times \omega_M,
\end{aligned} \tag{10}
$$

where $\hat{y}_i \in \mathbb{R}^{T_i \times N_C}$, $\hat{\mathbf{B}} \in \mathbb{R}^{T_i \times 2}$, and $\hat{m} \in \mathbb{R}^{T_i \times 1}$ denote the predicted confidence score with N_C classes, temporal boundary, and matching score, respectively. In addition, we adjust the scales of boundaries and matching scores through the learnable weights ω_B and ω_M, respectively.

3.4. Loss Function

In this section, we introduce the loss functions of our proposed method. As mentioned above, we train our model in two phases: (i) mask representation learning and (ii) action detector learning. The triplet loss $\mathcal{L}_{trip}$ in the mask representation learning first processes back-propagation. And then, we conduct back-propagation of the losses in the action detector learning.

The losses of the action detector consist of class $\mathcal{L}_{cls}$, boundary $\mathcal{L}_{reg}$, and matching score $\mathcal{L}_{mat}$ losses. We adopt a focal loss [28] for classification, which alleviates the class imbalance problem. Also, we use an tIoU loss for the boundary regression, which calculates the percentage of overlapping the predicted boundaries $\hat{\mathbf{B}} = (\hat{t}^s, \hat{t}^e)$ and ground truths, where $\hat{t}^s$ and $\hat{t}^e$ denote the action's start time and end time, respectively. Furthermore, we use the mean squared error between the matching score $\hat{m}$ and the tIoU value of the predicted boundary for the matching loss. Here, we normalize the matching score using a hyperbolic tangent function, which enriches the output range and slightly increases the performance than using a sigmoid function, as shown in Tab. 4. These losses can be formulated as follows:

$$
\begin{aligned}
\mathcal{L}_{cls} &= \sum_k (\mathrm{FL}(\hat{y}_k, y_k)), \\
\mathcal{L}_{reg} &= \sum_k (1 - \mathrm{tIoU}(\hat{\mathbf{B}}_k, \mathbf{B}_k)), \\
\mathcal{L}_{mat} &= \sum_k (\tanh(\hat{m}_k) - \mathrm{tIoU}(\hat{\mathbf{B}}_k, \mathbf{B}_k))^2,
\end{aligned} \tag{11}
$$

where y, $\hat{y}$, and FL denote the ground truth of class, predicted confidence score, and the focal loss, respectively.

The total loss can be formulated as:

$$
\mathcal{L} = \mathcal{L}_{cls} + \lambda_1(\mathcal{L}_{reg} + \mathcal{L}_{mat}) + \lambda_2 \mathcal{L}_{trip}, \tag{12}
$$

where λ_1 and λ_2 denote the weights balancing between the losses.

3.5. Inference

Given an untrimmed video X, our method outputs the distances from each time steps $\{(d_i^s, d_i^e)\}_{i=1}^T$, confidence score $\hat{y}$, and matching score $\hat{m}$, where i denotes the time steps. From the distances, we calculate the boundaries $\hat{\mathbf{B}}_i = (\hat{t}_i^s, \hat{t}_i^e)$ as follows:

$$
\begin{aligned}
\hat{t}_i^s &= i - d_i^s, \\
\hat{t}_i^e &= i + d_i^e.
\end{aligned} \tag{13}
$$

Method	Feature	THUMOS14						ActivityNet1.3			
		0.3	0.4	0.5	0.6	0.7	Avg.	0.5	0.75	0.95	Avg.
BSN (ECCV'18) [27]	TSN [41]	53.5	45.0	36.9	28.4	20.0	36.8	46.5	30.0	8.0	30.0
BMN (ICCV'19) [26]	TSN [41]	56.0	47.4	38.8	29.7	20.5	38.5	50.1	34.8	8.3	33.9
G-TAD (CVPR'20) [44]	TSN [41]	54.5	47.6	40.3	30.8	23.4	39.3	50.4	34.6	9.0	34.1
TCA-Net (CVPR'21) [34]	TSN [41]	60.6	53.2	44.6	36.8	26.7	44.3	52.3	36.7	6.9	35.5
RTD-Net (ICCV'21) [39]	I3D [6]	68.3	62.3	51.9	38.8	23.7	-	47.2	30.6	8.6	30.8
ContextLoc (ICCV'21) [48]	I3D [6]	68.3	63.8	54.3	41.8	26.2	-	56.0	35.2	3.6	34.2
AFSD (CVPR'21) [25]	I3D [6]	67.3	62.4	55.5	43.7	31.1	52.0	52.4	35.3	6.5	34.4
MUSES (CVPR'21) [30]	I3D [6]	68.9	64.0	56.9	46.3	31.0	-	50.0	35.0	6.6	34.0
DCAN (AAAI'22) [8]	TSN [41]	68.2	62.7	54.1	43.9	32.6	52.3	51.2	35.9	9.4	35.3
Zhu *et al.* (AAAI'22) [49]	I3D [6]	72.1	65.9	57.0	44.2	28.5	53.5	**58.1**	36.3	6.2	35.2
Liu *et al.* (CVPR'22) [29]	SlowFast [13]	69.4	64.3	56.0	46.4	34.9	54.2	50.5	36.0	**10.8**	35.1
Ours	I3D [6]	**76.7**	**73.1**	**66.8**	**57.2**	**42.7**	**63.3**	54.3	**37.7**	8.5	**36.4**

Table 1. Comparison of our method with other state-of-the-art methods on THUMOS14 and ActivityNet datasets. The results are measured by mAP (%) at different tIoU thresholds. The second column (Feature) denotes each method's video encoder.

We normalize the confidence and matching scores using the sigmoid and hyperbolic tangent functions in the same manner as training. Then, we obtain a refined confidence $\bar{y}$ score by multiplying each other as follows:

$$\bar{y} = \mathrm{sigmoid}(\hat{y}) \cdot \tanh(\hat{m}). \tag{14}$$

Finally, we can obtain the final outputs after conducting the soft-NMS [4] to suppress redundant proposals based on the refined confidence score.

4. Experiments

In this section, we provide extensive experiments on two primary datasets: THUMOS14 [18] and ActivityNet1.3 [5]. First, we introduce the two datasets, implementation details, and evaluation metrics used for our experiments. Next, we compare our method with previous state-of-the-art methods, and our overall results show high precision in localization and classification. Furthermore, we conduct various ablation studies to verify the effectiveness of our method. Finally, we provide an error profiling [2] that allows us to analyze our result's false positive ratios.

4.1. Datasets

In this section, we introduce two primary datasets used for our experiments:

THUMOS14 [18] contains 413 untrimmed videos with 20 action classes and temporal annotations. According to the public regulation, we split them into 200 videos for training and 213 videos for testing.

ActivityNet1.3 [5] contains 19,994 untrimmed videos with 200 action classes and temporal annotations, which is much larger than THUMOS14. According to the setting of prior works [27, 26, 44], we split the videos into 10,024 videos for training, 4,926 videos for validation, and 5,044 videos for testing by a 2:1:1 ratio.

4.2. Implementation Details

For the THUMOS14 dataset, we train our model for 45 epochs using AdamW [33] optimizer. The batch size is 4 and weight decay is set to 5×10^{-2}. We set a learning rate to 10^{-4} and adopt a cosine annealing [32] manner. We use the I3D [6] model, pre-trained on Kinetics dataset [21], to extract the video features from the video using a sliding window covering 16 frames with 4 strides. The loss weight parameters λ_1 and λ_2 are set to 1, which performed best in the ablation study in Tab. 5.

For the ActivityNet1.3 dataset, we train our model for 10 epochs using AdamW optimizer. The batch size is 16 and weight decay is set to 5×10^{-2}. We set a learning rate to 10^{-3} and adopt a cosine annealing manner. We use the I3D model, pre-trained on Kinetics dataset, to extract the video features from the video using a sliding window covering 16 frames without overlapping, *i.e.*, 16 strides. The loss weight parameters λ_1 and λ_2 are set to 1 as same as THUMOS14 settings. Furthermore, following [27, 44], we utilize the score fusion manner for reliable results. The video classification scores from [47] are multiplied by the confidence score in the inference time.

4.3. Evaluation Metrics

In our experiments, we use mean Average Precision (mAP) to evaluate TAL performance, which is the mean value for the average precision of each action class. Following traditional practice, the temporal Intersection over Union (tIoU) thresholds are set to [0.3:0.1:0.7] for THUMOS14 and [0.5:0.05:0.95] for ActivityNet1.3.

4.4. Main Results

In this section, we demonstrate the effectiveness of our method by comparing it with other state-of-the-art methods

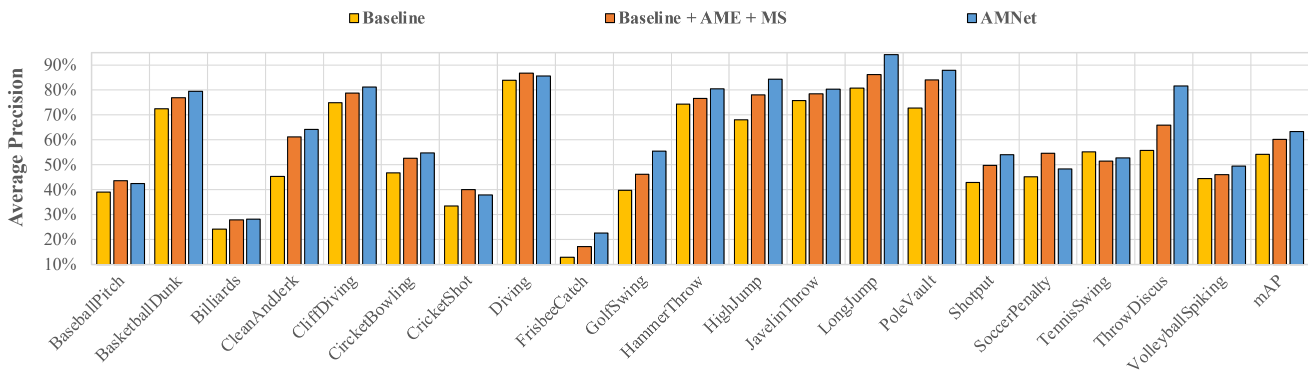

Figure 5. Per-class performance comparison on THUMOS14. The results are measured by AP@Avg on different tIoU thresholds.

AME	GAM	MS	THUMOS-14 0.3	0.4	0.5	0.6	0.7	Avg.
			72.2	66.5	58.5	45.6	29.2	54.4
		✓	73.9	68.1	61.3	48.2	32.1	56.7
✓		✓	75.5	71.2	64.5	52.8	36.8	60.2
	✓	✓	76.4	72.2	65.9	54.7	39.6	61.8
✓	✓	✓	**76.7**	**73.1**	**66.8**	**57.2**	**42.7**	**63.3**

Table 2. Ablation study of the proposed modules such as AME, GAM, and matching score head (MS) on THUMOS14.

Type	THUMOS14 0.5	0.7	Avg.
Baseline	61.3	32.1	56.7
+ Action-aware attention	62.9 (+1.6)	36.1 (+4.0)	59.1 (+2.4)
+ Self-attention	63.8 (+2.5)	38.0 (+5.9)	60.4 (+3.7)
+ Group-based attention	**65.3 (+4.0)**	**39.1 (+7.0)**	**61.5 (+4.8)**

Table 3. Ablation study of different attention on THUMOS14. The baseline model is the same as 2nd row in Tab. 2, which consists of the matching score head.

Confidence Score	THUMOS14 0.3	0.5	0.7	Avg.
sigmoid($\hat{y}$)	75.9	63.8	35.6	59.8
sigmoid($\hat{y}$) · sigmoid($\hat{m}$)	76.5 (+0.6)	65.4 (+1.6)	40.7 (+5.1)	62.1 (+2.3)
sigmoid($\hat{y}$) · tanh($\hat{m}$)	**76.7 (+0.8)**	**66.8 (+3.0)**	**42.7 (+7.1)**	**63.3 (+3.5)**

Table 4. Ablation study of different designs of confidence score on THUMOS14.

λ_1	λ_2	THUMOS14 0.3	0.4	0.5	0.6	0.7	Avg.
0.5	0.5	**76.7**	72.7	66.3	55.4	41.0	62.4
1	0.5	76.0	72.6	64.4	55.1	41.7	62.0
0.5	1	**76.7**	72.5	65.8	55.6	41.4	62.4
1	1	**76.7**	**73.1**	**66.8**	**57.2**	**42.7**	**63.3**
2	1	76.0	72.1	65.1	54.5	41.1	61.8
1	2	75.8	71.9	65.0	55.6	41.6	62.0
2	2	76.5	72.6	64.9	55.1	40.0	61.8

Table 5. Ablation study of the balanced weight between the different losses on THUMOS14.

on THUMOS14 and ActivityNet1.3, as shown in Tab. 1.

THUMOS14 We compare our method with other state-of-the-art methods on THUMOS14 in Tab. 1. Our method noticeably achieves the superior mAP at all thresholds, reaching 63.3%. In particular, our method surpasses the Zhu *et al.* [49] method by +4.6% mAP@0.3 absolute improvement, reaching 76.7%. Furthermore, our method outperforms the previous state-of-the-art method (Liu *et al.* [29]) by +7.8% mAP@0.7 absolute improvement.

ActivityNet1.3 We compare our method with other state-of-the-art methods on ActivityNet1.3 in Tab. 1. At tIoU=0.75, we achieve the highest mAP, which surpasses the TCA-Net [34] method by 1.0% absolute improvement, reaching 37.7%. Furthermore, although our method does not achieve the highest mAP@0.5 and mAP@0.95, we outperform other methods with a 0.9% gap at mAP@Avg. We guess two reasons for weaker performance improvement than on THUMOS14: First, it is more challenging to classify because ActivityNet1.3 has more action categories (200 classes) than THUMOS14 (20 classes). Second, because the temporal locations of ground truths are not diverse, the action detector is overfitted on the biased situations.

4.5. Ablation Study

Effectiveness of Proposed Modules We evaluate the effectiveness of our key modules, such as AME, GAM, and the matching score head (MS), as shown in Tab. 2. We adopt the anchor-free method [25] as the baseline model (1st row), which is improved through various training techniques such as cosine annealing and label smoothing with optimal parameter choices. In 2nd row, the result shows that the matching score head mitigates the inconsistency problem by refining the confidence score, improving +2.3% mAP@Avg compared to the baseline. Furthermore, in 3rd and 4th rows, we can observe that our key modules AME and GAM con-

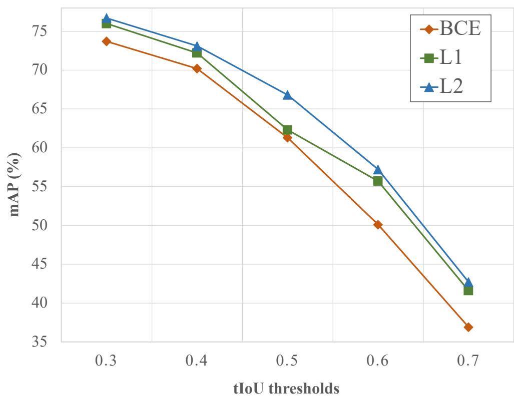

Figure 6. The ablation study of different matching loss designs on THUMOS14, measured by mAP (%) at different tIoU thresholds.

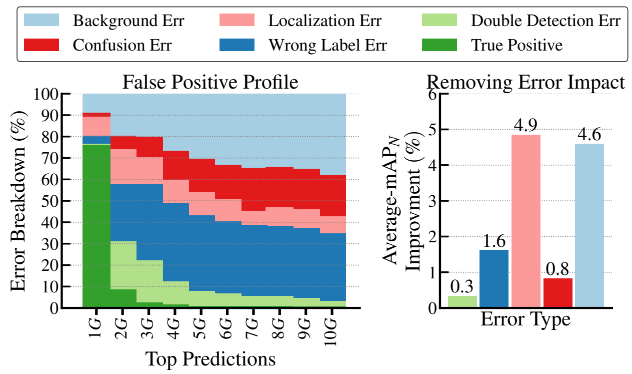

Figure 7. Error chart of our detection result, drawn up using DETAD [2]. There are error rates of 5 types on top-10G predictions, where G denotes the number of ground truths. Detailed instructions about the chart are in DETAD [2].

siderably improve the performance with +3.5% and +5.1% mAP@Avg compared to 2nd row. Finally, our complete model improves the performance by +8.9% mAP@Avg compared to the baseline. We also conduct the ablation study of each attention effect in Tab. 3, where the baseline (1st row) is the same as 2nd row in Tab. 2. Each row (2nd-4th rows) results from the baseline added the corresponding attention. The results show that group-based attention carries the highest gain by +4.8 mAP@Avg compared to the baseline. These ablation studies demonstrate that the AME and GAM contribute significantly to performance improvement. Additionally, we provide the detailed comparison (Fig. 5) of per-class between the baseline (1st row in Tab. 2), the baseline with GAM and MS (3rd row in Tab. 2), and our complete model.

Refinement of Confidence Score To verify the effectiveness of refining the confidence score with matching score, we conduct an ablation study by changing the design of confidence score, as shown in Tab. 4. 1st row denotes the result when our model infers using the vanilla confidence score trained without the matching score head. 2nd and 3rd rows denote the refined confidence scores by different matching scores. The results show that the hyperbolic tangent function slightly improves performance than the sigmoid function. We conjecture it is because the hyperbolic tangent function widens the matching score range.

Matching Loss To choose the suitable loss of the matching score, we experiment with the different designs of the matching losses $\mathcal{L}_{mat}$ on THUMOS14 dataset, as shown in Fig. 6. In the case of binary cross entropy (BCE), we replace the hyperbolic tangent function (eq. 11) with the sigmoid function, as the input of BCE must be positive values. The results show that L2 loss is the most stable and robust for predicting tIoU values of temporal boundaries.

Balancing Weights between Losses To find the optimal balancing weights, we conduct a grid search on THUMOS14 dataset, as shown in Tab. 5. First, we set the two hyper-parameters: λ_1 for regression losses and λ_2 for the triplet loss, considering the weight for classification loss to 1, and we set the weight range to [0.5:0.5:2]. As a result, we can observe that the setting when all weights are equivalent yields the best performance.

Errors of Our Result To analyze the limitations of our model, we provide the false positive error chart [2] of our detection results. The experiment results are reported at the fixed 0.5 tIoU threshold on THUMOS14 dataset. As shown in Fig. 7, we can observe that the impact of localization and background errors is significant. We expect a more precise regression loss design to mitigate them in further works.

5. Conclusion

In this paper, we propose a novel temporal action localization framework called AMNet, to address the ambiguity between consecutive frames caused by poor temporal information of video features. In particular, we present an AME to represent semantic action features and explicitly apply action-aware attention to video features extracted from a pre-trained video encoder. Furthermore, we propose a GAM to model temporal semantic knowledge by grouping multi-scale features. The extensive experimental results on THUMOS14 and ActivityNet1.3 demonstrated that our AMNet has high fidelity of localization and classification and can therefore achieve state-of-the-art performance.

Acknowledgement This work was partially supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2019-0-00079, Artificial Intelligence Graduate School Program(Korea University), No. 2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation).

References

[1] Mohiuddin Ahmad and Seong-Whan Lee. Human action recognition using multi-view image sequences. In *FG*, 2006.

[2] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *ECCV*, 2018.

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021.

[4] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, 2017.

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[7] Yunpeng Chang, Zhigang Tu, Wei Xie, and Junsong Yuan. Clustering driven deep autoencoder for video anomaly detection. In *ECCV*, 2020.

[8] Guo Chen, Yin-Dong Zheng, Limin Wang, and Tong Lu. Dcan: improving temporal action detection via dual context aggregation. In *AAAI*, 2022.

[9] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *CVPR*, 2020.

[10] Wen-Sheng Chu, Yale Song, and Alejandro Jaimes. Video co-summarization: Video summarization by visual co-occurrence. In *CVPR*, 2015.

[11] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *CVPR*, 2019.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.

[14] Basura Fernando, Cheston Tan, and Hakan Bilen. Weakly supervised gaussian networks for action detection. In *WACV*, 2020.

[15] Hiromichi Fujisawa, Hiroshi Sako, Yoshihiro Okada, and Seong-Whan Lee. Information capturing camera and developmental issues. In *ICDAR*, 1999.

[16] Valentin Gabeur, Arsha Nagrani, Chen Sun, Karteek Alahari, and Cordelia Schmid. Masking modalities for cross-modal video retrieval. In *WACV*, 2022.

[17] He-Yen Hsieh, Ding-Jie Chen, and Tyng-Luh Liu. Contextual proposal network for action localization. In *WACV*, 2022.

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

[19] Kyung-Min Jin, Gun-Hee Lee, and Seong-Whan Lee. OTPose: Occlusion-Aware Transformer for Pose Estimation in Sparsely-Labeled Videos. In *SMC*, 2022.

[20] Tae-Kyung Kang, Gun-Hee Lee, and Seong-Whan Lee. HTNet: Anchor-free Temporal Action Localization with Hierarchical Transformers. In *SMC*, 2022.

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

[22] Seong-Whan Lee, Jin H Kim, and Frans CA Groen. Translation-, rotation-and scale-invariant recognition of hand-drawn symbols in schematic diagrams. *IJPRAI*, 1990.

[23] Seong-Whan Lee and Sang-Yup Kim. Integrated segmentation and recognition of handwritten numerals with cascade neural network. *SMC, Part C*, 1999.

[24] Seong-Whan Lee and Alessandro Verri. *Pattern Recognition with Support Vector Machines: Proc. of First International Workshop, Niagara Falls, Canada*. Springer, 2003.

[25] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *CVPR*, 2021.

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

[27] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018.

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[29] Xiaolong Liu, Song Bai, and Xiang Bai. An empirical study of end-to-end temporal action detection. In *CVPR*, 2022.

[30] Xiaolong Liu, Yao Hu, Song Bai, Fei Ding, Xiang Bai, and Philip HS Torr. Multi-shot temporal event localization: a benchmark. In *CVPR*, 2021.

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[32] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[34] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. Temporal context aggregation network for temporal action proposal refinement. In *CVPR*, 2021.

[35] Maheen Rashid, Hedvig Kjellstrom, and Yong Jae Lee. Action graphs: Weakly-supervised action localization with graph convolution networks. In *WACV*, 2020.

[36] Myung-Cheol Roh, Tae-Yong Kim, Jihun Park, and Seong-Whan Lee. Accurate object contour tracking based on boundary edge selection. *Pattern Recognition*, 2007.

[37] Myung-Cheol Roh, Ho-Keun Shin, and Seong-Whan Lee. View-independent human action recognition with volume motion template on single stereo camera. *Pattern Recognition Letters*, 2010.

[38] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[39] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *ICCV*, 2021.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[41] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *TPAMI*, 2018.

[42] Yuetian Weng, Zizheng Pan, Mingfei Han, Xiaojun Chang, and Bohan Zhuang. An efficient spatio-temporal pyramid transformer for action detection. *arXiv preprint arXiv:2207.10448*, 2022.

[43] Guande Wu, Jianzhe Lin, and Claudio T Silva. Intentvizor: Towards generic query guided interactive video summarization. In *CVPR*, 2022.

[44] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, 2020.

[45] Hee-Deok Yang and Seong-Whan Lee. Reconstruction of 3d human body pose from stereo image sequences based on top-down learning. *Pattern Recognition*, 2007.

[46] M Zaigham Zaheer, Arif Mahmood, M Haris Khan, Mattia Segu, Fisher Yu, and Seung-Ik Lee. Generative cooperative learning for unsupervised video anomaly detection. In *CVPR*, 2022.

[47] Yue Zhao, Bowen Zhang, Zhirong Wu, Shuo Yang, Lei Zhou, Sijie Yan, Limin Wang, Yuanjun Xiong, D Lin, Y Qiao, et al. Cuhk & ethz & siat submission to activitynet challenge 2017. *arXiv preprint arXiv:1710.08011*, 2017.

[48] Zixin Zhu, Wei Tang, Le Wang, Nanning Zheng, and Gang Hua. Enriching local and global contexts for temporal action localization. In *ICCV*, 2021.

[49] Zixin Zhu, Le Wang, Wei Tang, Ziyi Liu, Nanning Zheng, and Gang Hua. Learning disentangled classification and localization representations for temporal action localization. In *AAAI*, 2022.