

## Class Semantics-based Attention for Action Detection

Deepak Sridhar<sup>1\*†</sup> Niamul Quader<sup>1\*</sup> Srikanth Muralidharan<sup>1</sup> Yaoxin Li<sup>1,2</sup> Peng Dai<sup>1</sup>  
Juwei Lu<sup>1</sup>

Huawei Noah's Ark Lab, Canada<sup>1</sup> University of Waterloo<sup>2</sup>

{deepak.sridhar1,niamul.quader1,srikanth.muralidharan,yaoxin.li,peng.dai,juwei.lu}@huawei.com

### Abstract

Action localization networks are often structured as a feature encoder sub-network and a localization sub-network, where the feature encoder learns to transform an input video to features that are useful for the localization sub-network to generate reliable action proposals. While some of the encoded features may be more useful for generating action proposals, prior action localization approaches do not include any attention mechanism that enables the localization sub-network to attend more to the more important features. In this paper, we propose a novel attention mechanism, the Class Semantics-based Attention (CSA), that learns from the temporal distribution of semantics of action classes present in an input video to find the importance scores of the encoded features, which are used to provide attention to the more useful encoded features. We demonstrate on two popular action detection datasets that incorporating our novel attention mechanism provides considerable performance gains on competitive action detection models (e.g., around 6.2% improvement over BMN action detection baseline to obtain 47.5% mAP on the THUMOS-14 dataset), and a new state-of-the-art of 36.25% mAP on the ActivityNet v1.3 dataset. Further, the CSA localization model family which includes BMN-CSA, was part of the second-placed submission at the 2021 ActivityNet action localization challenge. Our attention mechanism outperforms prior self-attention modules such as the squeeze-and-excitation in action detection task. We also observe that our attention mechanism is complementary to such self-attention modules in that performance improvements are seen when both are used together.

### 1. Introduction

The creation of digital videos and the need for video understanding has exploded over the last decade due to

widespread availability and presence of digital cameras. Two fundamental components of video understanding are identifying the action components that are present in the video [12, 22, 42] and localizing these actions across the temporal axis [7, 13, 29, 46, 23] and also the spatial axis [18, 37]. For video understanding tasks, it is crucial to learn good video representations, learning relevant encoded features that are useful for the video understanding tasks [12, 23, 25]. These rich video representations or encoded features can then be utilized to perform various video understanding tasks (e.g. a localization sub-network can use these features for detecting action segments, etc.). This paper focuses on the task of temporal action detection in untrimmed videos, which has several applications such as content-based video searching [21], video highlight generation [15] and surveillance [9].

With the availability of large-scale action recognition and detection datasets (e.g. Kinetics-400 [5], ActivityNet v1.3 [4], etc.) and the availability of high performance computing services, deep learning approaches have achieved enormous success in learning video representations or learning encoders that generate features which are useful for video understanding tasks such as temporal action detection [22, 12, 42, 35]. These encoder learning processes can be very comprehensive - e.g. the encoder learning process of BMN [23] (a recently proposed competitive approach) first involves training a TSN-ResNet101 action recognition model [42] on Kinetics-400 action recognition task [52] followed by fine-tuning the action recognition model on ActivityNet v1.3 [4], followed by extracting action class-semantic rich features using the action recognition model, and finally using a localization encoder to encode features that are useful for video understanding tasks such as action localization. These encoded features are processed by the localization sub-network for generating action proposals [51, 25, 23]. It is unlikely that all of these encoded features will be equally useful or important for the localization sub-network for a particular video. In fact, prior works on attention in convolutional neural networks (ConvNets) have shown that attending to more important feature

\*denotes equal contribution

†Corresponding author

channels [19] or locations [43] can accelerate training and usually improve network performances. Despite the possibility that the importance of the encoded feature can vary for different videos and the existing motivation from prior works that have demonstrated improved ConvNet performances via attention mechanisms [43, 19, 40], there are no prior action localization networks that have attention mechanisms for attending to the more important encoded features for accelerating training and improving performance.

Incorporating attention at the encoded feature, while not used before for action localization ConvNets, can easily be implemented with prominent self-attention methods (e.g. squeeze-and-excitation (SE) based attention [19], transformer [40]), which learn the inter-dependencies of the encoded feature to estimate the relative importance of the features. In contrast, we propose a novel attention mechanism that computes the relative importance of the features based on class-specific semantically rich features that are extracted by the action recognition model and that are used at the input of the encoder in the action localization network (Fig. 1). Our rationale on using these class-specific semantically rich features is that the distribution of importance of the encoded feature will likely depend based on which action class (or action classes) the video contains. Evidence for such class-specific dependency of importance distribution was demonstrated in an ablation study in [19] which showed that some specific feature channels were more important than others for one particular class while having low importance for another class within the ImageNet dataset [36]. For action localization tasks, the class-semantics can vary across the temporal axis since different action classes can be present at different time points. Prior works on action localization in the fully and weakly supervised setting (where only class-level supervision is used to learn the temporal boundaries of an action), use class semantic-rich classification features as input and have successfully pushed the state of the art localization performances. This shows that the class-semantics features contain useful information for reliable action localization. To learn the importance of features from the temporally varying class-semantics in videos, our novel attention mechanism jointly learns from both the channel and temporal axes of the encoder input features, and provides attention both along the channel and temporal axes of the encoded features. Our attention mechanism is generic and can be easily applied to prior action localization networks that have an encoder and a localization sub-network (e.g. [23, 8, 46]) (Fig. 1). We demonstrate that our attention mechanism considerably improves on baseline ConvNets [23, 46, 8] on two major action detection datasets (Thumos[20] and ActivityNet v1.3 [4]). Our ablation studies also show that our novel attention mechanism can provide complementary benefits when used together with self-attention mechanisms (such as [19]).

## 2. Related Works

### 2.1. Attention in video understanding

Attention mechanisms are used in a variety of video recognition tasks including fully supervised action recognition [17, 30, 3, 38, 34], weakly supervised action detection [41, 31, 26] and spatio-temporal action localization [16]. In action recognition models, attention is used for performing different operations, including weighted spatio-temporal pooling or gating [38, 17, 44, 30], encoding spatio-temporal features by capturing different kinds of relations and dependencies [16, 3, 11]. In weakly supervised action detection methods [41, 31, 32], attention mechanism is used to rank clips for action detection, with the weights learned from video level classification objectives. Quader et al. [34] showed that some feature channels become more important than others during training and introduced an attention mechanism that focused on optimizing those important feature channels to improved action and gesture recognition performances. Unlike these methods, our focus in this work is on fully supervised temporal action localization, where we use our attention mechanism to learn from temporal distribution of semantics of action classes in the multi-crop clip representations and apply it on our encoded features that are then processed by a localization sub-network for generating action proposals.

### 2.2. Attention in temporal action localization

The above mentioned attention mechanisms proposed for action recognition tasks [38, 41, 44, 17, 34] are not directly useful for action localization for the following reasons: (1) action recognition models have spatio-temporal RGB inputs whereas competitive action localization models use encoded temporal inputs, (2) an action recognition model’s learnable parameters (including any attention mechanisms inside) are usually not part of the training process of competitive action localization networks due to computational constraints in end-to-end training [23, 46, 8, 50]. In fact, there exists no prior works on applying attention mechanisms in competitive action localization networks. Recent competitive action localization ConvNets are composed of an encoder sub-network that encodes features for downstream localization task and a localization sub-network that generates action proposals. The most straightforward application of attention in action localization networks is perhaps to apply channel-wise self-attention at the encoder output via processing the inter-dependencies of the encoder output using self-attention mechanisms such as SE [19], CBAM [43] and transformer [40]. In this paper, we explore such self-attention mechanisms, and more importantly, propose a novel attention mechanism that processes the action class-specific semantically rich temporal features at the input of the localization network to identify channel-

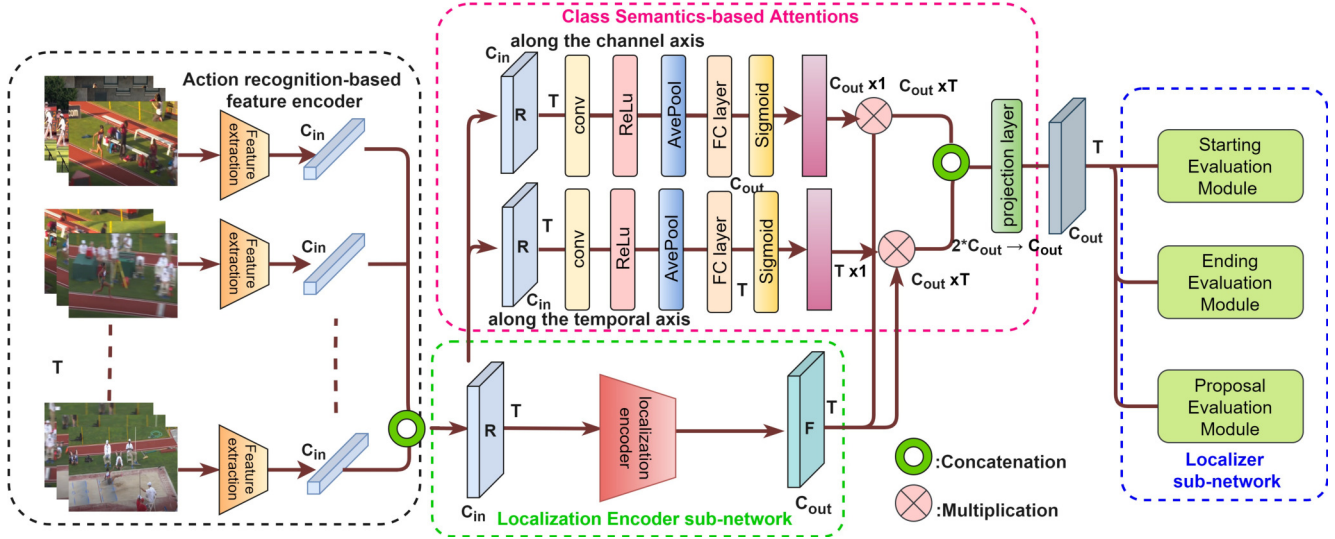


Figure 1: The architecture of a generic action detection network with incorporation of our CSA attention mechanism. The generic action detection architecture consists of three major components: (1) an action recognition-based feature encoder that extracts class semantics rich features  $R$ , (2) a localization encoder sub-network that encodes  $R$  to  $F$ , and (3) a localization sub-network that processes  $F$  for generating action proposals. Our attention mechanism learns attention weights from  $R$  and applies attention on  $F$  both along the channel and the temporal axis, and then fuses the two attention-applied outputs.

wise and temporal-point-wise attention at the encoder output; our rationale - the temporal variation of class-semantics of a video can have important cues on which channels and time-points the localization sub-network should attend to.

### 3. Technical Approach

We first document the layout for a generic action localization architecture (Fig. 1), which consists of a pre-trained action recognition ConvNet that extracts class-semantic rich features  $R_{i,t} \in \{r_{1,t}, \dots, r_{C_{in},t}\}$  ( $C_{in}$  represents the number of feature channels at each timepoint) at equidistantly distributed temporal points over the length of a video ( $t \in \{1, \dots, T\}$ ). Action localization ConvNet (composed of sub-networks for encoder and localization) is trained with  $R$  as input to generate action proposals (Fig. 1). As the ConvNet is trained, the encoder sub-network learns to encode the class-semantic rich features  $R$  to another temporally sequential feature  $F_{i,t} \in \{f_{1,t}, \dots, f_{C_{out},t}\}$  more suitable for subsequent action localization tasks (e.g. generating the start and end temporal points of proposals and using confidence maps to group the start and end points [23]), and the localization sub-network learns to use  $F$  to perform these localization tasks.

In this work, we use standard action recognition, encoder, and localization sub-networks, and we only concentrate on the attention mechanism on  $F$  (as shown in Fig. 1 by the pink dashed line). In Sec. 3.1, we propose our novel attention mechanism that learns from the local temporal dis-

tribution of class-specific semantically rich features  $R$  (or class semantics-based attention (CSA)) to estimate a usefulness or attention score for all feature channels and temporal points, which is then used to compute the attention-modified encoder feature  $F_A$  as the new input to the localization sub-network. In Sec. 3.2, we provide technical description of our attention mechanism.

#### 3.1. Class semantics-based attention (CSA)

To help the localization sub-network attend more to feature channels and temporal locations that are more useful for localization tasks, there needs to be an attention mechanism that has the ability to adjust  $F_{i,t}$  based on how useful or important it is to the action localization tasks. To mitigate the lack of input-adaptive channel/temporal attention mechanism in prior action localization networks, we propose an attention module that estimates the per-channel and per-timepoint attention weights independently and apply them independently on the localization encoder sub-network output  $F$ . We now discuss in detail about design of our attention module: its input, its joint learning framework, and the rationale behind this design choice, including the choice of input and the learning framework.

**Input for the attention modules:** Based on prior works on attention, the more straightforward approach for applying attention on  $F$  is to use  $F$  itself, similar to self-attention frameworks such as the SE block [19]. The  $F$  features are likely more potent for identifying background from foreground (having been trained for action localiza-

tion), while the  $R$  features are more potent for identifying which action class/classes the video belongs to (having been trained for action recognition). Assuming that our channel/temporal attention mechanisms are more dependent on the action class information-rich  $R$  features rather than the foreground/background class information-rich  $F$  features,  $R$  is likely a better input choice for the attention modules, and we choose it as input to our attention modules. Comparing with self-attention modules that use  $F$  as their input, we find that our CSA performs significantly better (Sec. 4.3), suggesting that such class semantics-based (or  $R$  input) based attention mechanisms can be more potent.

**Joint learning from class-semantics and temporal context:** In contrast to the task of image classification in the ablation study of [19] that had image as input, action localization tasks have video inputs that not only have class semantic information at different time points, but there is also variation of these class-semantics across the temporal axis. For example, ‘drinking coffee’ action segment in a video can be preceded by ‘cooking’ action and followed by ‘washing dishes’ action. Thus, local temporal understanding of  $R$  can provide indications on which class-semantics are more dominant at a time point. Also, since boundary detection is an important task for action localization network, temporal understanding of this variation of these class semantics could provide indications on which temporal positions are more important for boundary points.

**Computation of the attention weights:** The attention weights are computed by two separate attention modules, each of which have the class semantics rich  $R$  feature as its input. Each of the attention modules (one estimating per-channel attention weight and the other estimating per-timepoint attention weight) learns from the local temporal variation of  $R$  to compute the per-channel and per-timepoint attention weights. Then we apply our channel and temporal attention on  $F$  (Fig. 1) (or enable the localization sub-network to attend to more important feature channels and temporal locations to get  $F_{AC}$  and  $F_{AT}$ , respectively (Fig. 1). Finally, we aggregate  $F_{AC}$  and  $F_{AT}$  to get our final attention modified encoder output  $F_A$  (Fig. 1). Next, we present the technical description of the attention module.

### 3.2. Technical description of the CSA

We use a simple attention block consisting of a 1D convolution (with kernel size  $k$ ) that operates succinctly to learn jointly from both the temporal and the channel axes of  $R$ . On the temporal axis, this 1D convolution has a receptive field of  $k$ , and on the channel axis, it has a receptive field of  $C_{in}$  (or the entire length of feature channels). Thus, it captures information available from the global channel context as well as from the local temporal context simultaneously. Note that our CSA attention has two attention modules: one computes per-channel attention and the other computes per-

time-point attention. To achieve this, we compute a channel attention vector and a temporal attention vector in parallel, assigning weights to specific channel-temporal input locations. Then, we perform late-fusion through a simple concatenation operation. We downsize the resultant output number of channels to that of class features  $F$  using a simple projection layer to be subsequently processed by the localization sub-network.

The input to our temporal attention block is the  $R^{C_{in} \times T}$ , where  $C_{in}$  and  $T$  correspond to number of input channels and temporal points respectively. Similarly, let  $C_{out}$  denote number of output channels, and  $F$  denotes the encoder feature which would be modified by our attention mechanism. Our temporal attention is computed as:

$$\mathbf{y}_T = Avg_{\{t=1,2,\dots,T\}} \left( \mathbf{f}_{Tconv_{C_{in} \times T}}(R) \right) \quad (1)$$

$$\mathbf{A}_T = sigmoid(\mathbf{f}_{c_T}(\mathbf{y}_T)) \quad (2)$$

$$\mathbf{F}_{AT} = (\mathbf{A}_T \mathbb{1}_{1 \times C_{out}})^T \circ F \quad (3)$$

where,  $\mathbf{f}_{Tconv_{C_{in} \times T}}$  is the core component of the temporal attention block, which has a 1D-convolutional layer of kernel size 3, a ReLU activation function. This block learns global channel context and local temporal context, and outputs a  $T$  dimensional vector  $\mathbf{y}_T$ . A fully connected layer (denoted by  $\mathbf{f}_{c_T}$ )-sigmoid map on top of  $\mathbf{y}_T$  yields  $T$  dimensional attention vector  $\mathbf{A}_T$ . Finally, Hadamard operation with the encoder feature  $F$  yields  $F_{AT}$ , temporal attention modified encoder feature.

Similarly, we formulate our channel attention mechanism to obtain the channel attention modified encoder feature  $F_{AC} = A_C \mathbb{1}_{1 \times T} \circ F$ .

In the end, the attention modified encoder feature  $F_A$  is computed using these two outputs. We compute this output by combining our channel attention modified encoder feature and temporal attention modified encoder feature by simply concatenating them, and mapping the number of channels back to  $C_{out}$  using a convolution-ReLU block  $\mathbf{f}_{TC}^{2C_{out} \times C_{out}}$ . An alternative way to perform fusion is to add both the temporal and channel attended features together. However, this results in an averaging effect that diminishes the contribution from each of the attention modules. Mathematically,

$$F_A^{C_{out} \times T} = \mathbf{f}_{TC}^{2C_{out} \times C_{out}}([F_{AT}; F_{AC}]) \quad (4)$$

where “;” denotes the concatenation operator, and  $F_A^{C_{out} \times T}$  denotes the final encoder output, modulated by temporal and channel attention mechanisms. The localization sub-network uses this output and performs subsequent steps required for localizing all the individual actions in the video.

## 4. Experiments and results

### 4.1. Experimental setup

**Datasets:** To validate the utility of our CSA attention mechanism, we choose two popularly used action detection datasets - ActivityNet v1.3 [4] and THUMOS-14 [20]. ActivityNet v1.3 is a larger-scale action localization benchmark (containing around 10K training and around 5K validation videos [4]) compared to the THUMOS-14 challenge dataset (containing 200 training and 213 validation videos [20]). However, THUMOS-14 has more fine-grained action segment annotations overall (around 15 action instances per video) [49, 20] compared to ActivityNet v1.3 (around 1.65 action instances per video) [4].

**Evaluation metric for detection:** Mean Average Precision (mAP) is popularly used to quantitatively evaluate different approaches [23, 25, 46, 29, 27, 51], where the Average Precision (AP) is calculated on each action class respectively. We report the mAP at different Intersection over Union (IoU) thresholds and also report the overall mAP. For fair comparison between different networks, we use pre-trained and publicly available action recognition models (the CUHK classifier [52] for ActivityNet v1.3 [4] and UntrimmedNets [41] for THUMOS-14 [20]), and we use the publicly available  $R$  features (TSN features [42] available in [48] for ActivityNet v1.3 [4] and TSN features [42] available in [46] for THUMOS-14 [20]). Finally, to evaluate proposal quality, we calculate Average Recall (AR) under different Average Number of proposals (AN) as AR@AN (AN varied from 1 to 100), and compute the area under the AR vs. AN curve (AUC).

**Implementation details:** We incorporate our CSA attention mechanism on three competitive and popularly used action localization networks, BMN [23], GTAD [46] and DBG [8]. As is originally used in BMN [23], GTAD [46] and DBG [8], we use a temporal scale of  $T = 100$  for ActivityNet v1.3 and  $T = 256$  for Thumos14. Action localization model performances can also vary depending on the quality of the encoded features that are originally extracted from video sequences (i.e. the quality of the  $R$  feature (Sec. 2.2)). Therefore, for fair comparison with other action detection benchmarks, we use the publicly available  $R$  features (TSN features [42] available in [48] for ActivityNet v1.3 [4] and TSN features [42] available in [46] for THUMOS-14 [20]). We use Adam optimizer with an initial learning rate of 0.001 with weight decay  $1e^{-4}$  and train for 10 epochs on both the datasets using a step size of 7 for ActivityNet v1.3 and 3 for THUMOS-14. To compare utility of CSA against self-attention mechanisms, SE, CBAM and transformer [19, 43, 40], we implement SE, CBAM and transformer [19, 43, 40] at the encoder output (self-attention i.e., attention applied on encoder output learns the attention weights based on the encoder output). To further evalu-

ate whether our CSA benefits only from learning from processing the class-semantics rich encoder input features or whether it also benefits from the design of the CSA module, we experiment with replacing the CSA architecture with the SE architecture (FF-CSA in Table 5).

### 4.2. CSA’s impact on action localization

**Performance improvement:** Incorporating CSA attention within baseline action localization ConvNets provides consistent performance gains on both the THUMOS-14 (3.8% on GTAD baseline and 6.2% on BMN baseline, Table 1) and the ActivityNet v1.3 (0.52% on GTAD baseline and 0.6% on BMN baseline, Table 2), and yields a new state-of-the-art (SotA) performance of 36.25% mAP on the ActivityNet v1.3 dataset when used with BMN baseline [23] and publicly available TSP [1] encoded features. On the more challenging THUMOS-14 dataset that has higher number of action instances per video, the performance gains with CSA attention are significantly better than that on ActivityNet. This significant difference is perhaps due to a stronger necessity of using our CSA attention module (which helps the localization sub-network attend to important feature channels and temporal points) on the more difficult action detection tasks on videos that have higher number of action segments, boundaries and action classes. We also note that while the performance gains on THUMOS-14 across different tIoU are similar (e.g. 5.9% improvement over BMN baseline at tIoU=0.3 and 6.2% improvement over BMN baseline at tIoU=0.7), the performance actually drops on ActivityNet v1.3 for high tIoU=0.95. These action segments where the baseline BMN ConvNet performs better are mostly long duration action segments. Both of the above observations suggest that our CSA attention is particularly effective on challenging videos that either have higher number of action segments or have small duration action segments. In terms of proposal quality, DBG [8] has the SotA AUC among prior approaches. We find that incorporating CSA within DBG improves AR@AN for all AN values (Table 3) and improves overall AUC by 0.8%.

**CSA accelerates training:** Incorporating CSA attention on baseline BMN and GTAD speeds up convergence of the networks with faster reduction in both training and validation losses. Fig. 2 shows the comparison of loss curves for baseline and SE BMN models with CSA. We find that CSA has a steeper loss curve for both training and validation as compared to SE attention suggesting that CSA helps the network to learn better. To test whether the mAP performance improvement from incorporating CSA is simply from accelerated training, we train baseline BMN for higher number of epochs (20 instead of original 10 epochs). We find no improvement in baseline BMN performance, suggesting that our CSA attention module not only accelerates training but

Method	tIoU					Avg mAP
	0.3	0.4	0.5	0.6	0.7	
R-C3D [47]	44.8	35.6	28.9	-	-	-
SSN [51]	51.9	41.0	29.8	-	-	-
CBR [14]	50.1	41.3	31.0	19.1	9.9	30.3
ETP [33]	48.2	42.4	34.2	23.4	13.9	32.4
BSN [25]	53.5	45.0	36.9	28.4	20.0	36.8
MGG [28]	53.9	46.8	37.4	29.5	21.3	37.8
BMN [23]	56.0	47.4	38.8	29.7	20.5	38.4
GTAD [46]	54.5	47.6	40.2	30.8	23.4	39.3
TAL-Net [6]	53.2	48.5	42.8	33.8	20.8	39.8
CMS-RC3D[2]	54.7	48.2	40.0	-	-	-
GTAN [29]	57.8	47.2	38.8	-	-	-
*GTAD-PGCN	66.4	60.4	51.6	37.6	22.0	47.8
GTAD (Ours)	54.5	48.2	39.9	30.4	21.2	38.8
BMN (Ours)	58.5	51.6	42.9	32.1	21.2	41.3
GTAD-CSA	<b>58.4</b>	<b>52.8</b>	<b>44.0</b>	<b>33.6</b>	<b>24.2</b>	<b>42.6</b>
BMN-CSA	<b>64.4</b>	<b>58.0</b>	<b>49.2</b>	<b>38.2</b>	<b>27.8</b>	<b>47.5</b>

Table 1: Quantitative comparison of the proposed CSA module with competing methods on THUMOS-14 action detection dataset in terms of mAP @tIoU. BMN+CSA beats all prior approaches by large margins. \* denotes proposal-level post-processing methods that are complementary to other works including CSA.

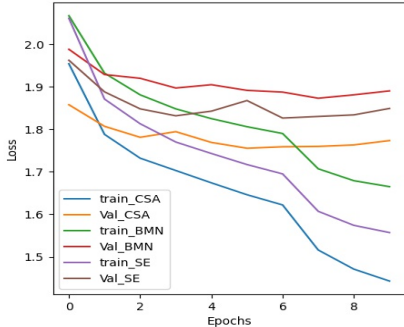


Figure 2: Comparison of training and validation loss curves of CSA with SE attention and baseline BMN on ActivityNet v1.3.

also leads to better optimized networks for action detection.

**Computational considerations:** The added computation cost with CSA attention (around 1ms/video) is similar to that of the efficient SE [19] mechanism. This is because the architecture of CSA is mostly similar to that of SE except that it includes an extra lightweight 1D convolution that learns jointly from both class-semantics and temporal context. Note that CSA provides considerably more mAP gain

Method	tIoU			Avg mAP
	0.5	0.75	0.95	
R-C3D [47]	26.80	-	-	12.70
*CMS-RC3D[2]	32.92	18.36	1.13	18.46
TCN [10]	36.17	21.12	3.89	-
*TAL-Net [6]	38.23	18.30	1.30	20.22
CDC [39]	43.83	25.88	0.21	22.77
Xiong et al.[45]	39.12	23.48	5.49	23.98
SSTAD [24]	44.39	29.65	7.09	29.17
BSN [25]	46.45	29.96	8.02	30.03
BMN [23]	50.07	34.78	8.29	33.85
GTAD [46]	50.36	34.60	<b>9.02</b>	34.09
*GTAN [29]	52.61	34.14	8.91	34.31
DBG [8]	36.69	26.61	7.33	25.67
DBG (Ours)	37.68	27.39	7.76	26.43
BMN (Ours)	51.73	35.74	6.53	34.83
GTAD (Ours)	51.45	35.86	8.86	35.17
DBG-CSA	<b>40.08</b>	<b>29.45</b>	<b>8.78</b>	<b>28.52</b>
BMN-CSA	<b>52.44</b>	<b>36.69</b>	5.18	<b>35.43</b>
GTAD-CSA	<b>51.88</b>	<b>36.88</b>	8.74	<b>35.69</b>
BMN-CSA-SE	<b>52.45</b>	<b>36.75</b>	6.38	<b>35.58</b>
BMN-CSA-Trans	52.03	36.71	<b>9.23</b>	<b>35.69</b>
BMN (TSP [1])	<b>52.56</b>	<b>36.86</b>	6.46	<b>35.89</b>
BMN-CSA (TSP)	<b>52.64</b>	<b>37.75</b>	7.94	<b>36.25</b>

Table 2: Quantitative comparison of the proposed CSA module with competing methods on ActivityNet v1.3 action detection dataset in terms of mAP @tIoU. BMN+CSA beats all prior two-stage approaches by notable margins and achieves SotA results, one-stage approaches are marked with \* which are end-to-end trained from raw RGB frames.

Module	AR@1	AR@5	AR@10	AR@100	AUC (%)
BMN	34.27	50.77	57.27	73.53	65.98
DBG	31.02	49.66	57.45	76.51	68.19
BMN-CSA	<b>35.04</b>	<b>51.63</b>	<b>57.90</b>	<b>74.04</b>	<b>66.82</b>
DBG-CSA	<b>32.46</b>	<b>50.48</b>	<b>58.27</b>	<b>77.12</b>	<b>68.89</b>

Table 3: Comparison of AR@AN for baseline methods and methods with our CSA module on ActivityNet v1.3 dataset

Module	0.3	0.4	0.5	0.6	0.7	mAP
Trans	55.1	48.8	41.3	30.5	19.7	39.1
CBAM	56.8	50.3	42.3	31.9	22.3	40.7
SE	62.4	54.8	46.1	34.2	23.6	44.2
Ours	<b>64.4</b>	<b>58.0</b>	<b>49.2</b>	<b>38.2</b>	<b>27.8</b>	<b>47.5</b>

Table 4: Comparison of other attention modules with our CSA module on THUMOS-14



compared to SE (Table 4).

### 4.3. Ablation studies

**Comparison with self-attention mechanisms:** CSA is different from self-attention mechanisms in two ways: (1) CSA estimates the attention weights based on class-semantic rich features at the encoder input and applies it at the encoder output whereas self-attention mechanisms applied at the encoder output estimates the attention weights based on localization-information rich features at the encoder output itself, (2) CSA learns jointly from both class-semantic and temporal context facilitated by a simple 1D convolution. Our CSA attention mechanism outperforms three popular self-attention mechanisms we experimented with, namely the transformer (Trans) [40], CBAM [43] and SE [19] attention mechanisms when applied at the encoder output of the BMN (Table 4). CSA outperforms all of these attention mechanisms. Interestingly, we find the training loss with including Transformer in BMN to be lower than BMN without the attention module, yet the validation mAP with Transformer is lower showing signs of over-fitting. This is not the case for our CSA attention or SE/CBAM, all of which provide mAP gains as well. To find whether self-attention mechanisms and our CSA can be complementary, we incorporate SE after all the 1D convolution layers in the BMN encoder, except at the encoder output where we incorporate the CSA attention. As shown in Table 2 we find that this provides additional gain of **0.15%** mAP. We also conduct the same experiment with Transformer instead of SE and find an additional gain of **0.26%** mAP. This complementary behavior between the two attention mechanisms (i.e. CSA with self-attention) is perhaps due to the difference in the inputs - SE/Transformer learns from the localization-rich features within the encoder sub-network, whereas CSA learns from the class-semantic rich features at the encoder input  $R$ .

Module	0.3	0.4	0.5	0.6	0.7	mAP
FF-CSA	60.9	54.1	45.8	35.5	24.1	44.1
G-CSA	61.7	54.9	46.2	36.2	24.7	44.8
CSA (1-Conv)	<b>64.1</b>	<b>56.3</b>	<b>48.2</b>	<b>38.1</b>	<b>26.6</b>	<b>46.7</b>
LG-CSA	59.8	53.2	44.4	34.5	24.4	43.3

Table 5: Ablation study of using different variations of CSA module on THUMOS-14. Note: CSA 1-Conv is used for fair comparison with other baselines having single convolutional layer.

**Alternate design choices for CSA** We experiment with different design choices for computing the attention weights via processing of the input features  $R$  with different variations to the 1D convolution design choice. These variants

Channel	Temporal	0.3	0.4	0.5	0.6	0.7	mAP
✓		62.1	55.8	46.4	36.5	24.7	45.1
	✓	62.5	56.2	46.5	36.9	26.1	45.7
✓	✓	<b>64.1</b>	<b>56.3</b>	<b>48.2</b>	<b>38.1</b>	<b>26.6</b>	<b>46.7</b>

Table 6: Ablation study of different components of CSA (with one convolution) on THUMOS-14 dataset.

Module	0.3	0.4	0.5	0.6	0.7	mAP
k=1	62.8	55.8	47.2	36.4	24.9	45.4
k=3	<b>64.1</b>	<b>56.3</b>	<b>48.2</b>	<b>38.1</b>	<b>26.6</b>	<b>46.7</b>
k=5	57.9	49.5	40.4	30.8	20.5	39.8
1-Conv	64.1	56.3	48.2	38.1	26.6	46.7
2-Conv	<b>64.4</b>	<b>58.0</b>	<b>49.2</b>	<b>38.2</b>	<b>27.8</b>	<b>47.5</b>
3-Conv	61.8	54.7	45.3	35.5	23.9	44.2

Table 7: Effect of different kernel size and number of convolutional blocks inside CSA module on THUMOS-14

are: (1) Direct (FF) - not using the 1D convolution at all, (2) global context learning (G) - using a transformer encoder [40] that has global receptive field instead of the convolution block. (3) Local context learning (L) - We use 1D convolution to locally process the temporal scales of the input features for computing the attention weights. (4) Local-Global Context learning (LG) - To see if local and global information complement each other, we apply local 1D convolutional operation followed by a global transformer operation. Tables 5, 6 show the results of using different variants and we find that local information (i.e., local temporal features) is key for computing the attention weights as compared to others. While global information can perhaps help in providing better attention weights, we conjecture that since learning such global information requires modules with higher number of learnable parameters (e.g. transformer encoder requires more parameters than a simple 1D convolution), it becomes difficult to train the attention sub-network. On that note, we also experiment with the hyper-parameters within the attention module (Table 7) by increasing the number of sequential 1D convolution - we find supporting evidence in here as well in that increasing the number of sequential 1D convolution from one to two improves the results whereas from two to three decreases the performance significantly. Similarly, increasing the kernel size from 3 to 5, which increases the number of learnable parameters, leads to inferior performance, suggesting that larger CSA modules with higher number of learnable parameters could be difficult to train.

**Location of application of CSA attention** We conduct an ablation to study the effect of applying the CSA atten-

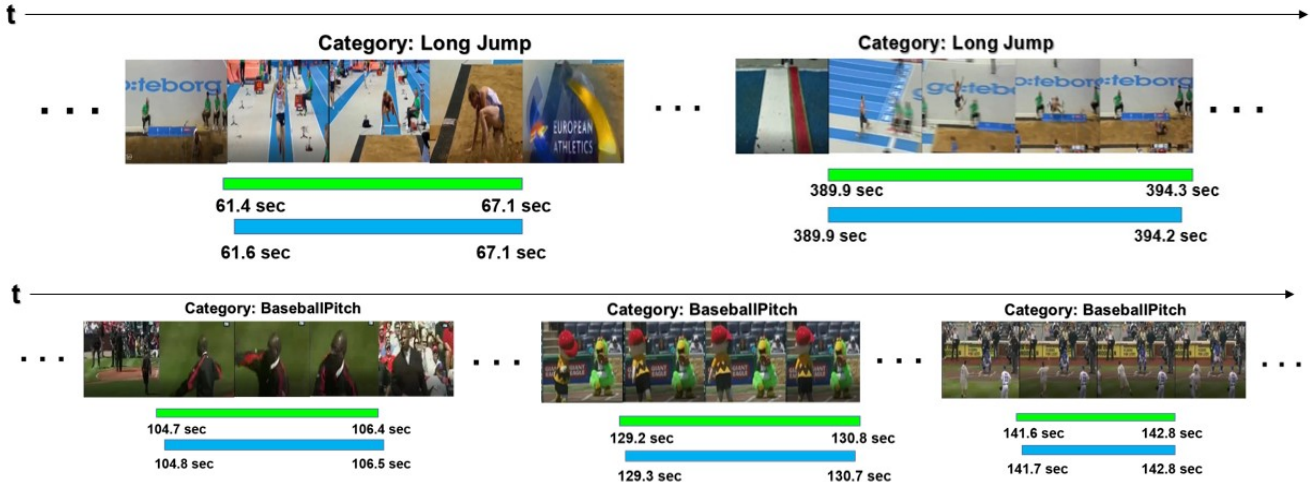


Figure 3: Qualitative results from THUMOS-14 dataset. Green segments show groundtruth and blue segments show predicted segments using BMN-CSA model. Our model is able to accurately detect diverse length sport domain actions.

Module	0.3	0.4	0.5	0.6	0.7	mAP
Start-CSA	60.9	53.6	44.9	34.3	23.3	43.4
Middle-CSA	63.8	57.1	47.7	36.9	25.3	46.2
CSA (2-Conv)	<b>64.4</b>	<b>58.0</b>	<b>49.2</b>	<b>38.2</b>	<b>27.8</b>	<b>47.5</b>

Table 8: Ablation study of applying the CSA module at different locations of the encoder module on THUMOS-14 dataset. Start-CSA applies attention at the encoder input. Middle-CSA applies attention at the middle of the encoder.

tion module at different locations within the encoder module. We identify three cases - (1) At the start (2) At the middle and (3) At the output of the encoder. Table 8 shows the results of applying the CSA attention module at the three locations. We find that the attention applied at the end of the encoder (input to the localization sub-network) outperforms other results. CSA attention applied to the input features gives significantly lower (3.9% drop) results. This shows that applying attention at  $F$  optimally modulates the features such that it helps to localize the start/end times and their corresponding scores more precisely, while not interfering with the encoder sub-network that has a separate objective of learning a better feature encoding process for the input  $R$ .

## 5. Discussion

We find CSA to consistently provide performance gains when incorporated with publicly available and readily reproducible baseline networks [23, 8, 46] on popular action localization datasets (THUMOS-14 and ActivityNet v1.3) with different sets of publicly available encoded features

(e.g. TSN features [42], TSP features [1]). We also find that CSA is better than self-attention mechanisms, and in addition provides complementary gains when used with self-attention mechanisms. Our BMN-CSA with TSP features currently have the SotA in ActivityNet v1.3 and was a core technical contribution behind achieving a top position in the 2021 ActivityNet temporal localization challenge. Further, Fig. 3 shows qualitative results of the SotA BMN-CSA model on THUMOS-14 dataset. From the figure, it can be observed that our model is accurately able to detect action segments of two different categories of different action lengths. Despite these promising results on consistent performance gains and current SotA, our CSA evaluation is currently limited in not being tested with other competitive baseline architectures that have been shown to have better performance than BMN [23], e.g. GTAD-PGCN [46]. We envision that incorporating CSA in such improved network architectures will yield better SotA performances.

## 6. Conclusion

In this paper, we proposed a novel attention mechanism, CSA, for action detection networks that utilizes action-class semantics to compute attention weights. CSA mechanism is effective in applying attention along both the channel and the temporal axes of the encoded feature of an action localization network. We demonstrate that CSA is generic and is easily integrated with prior action detection approaches. Incorporating CSA provides consistent performance gains on action detection tasks, and incorporating it in prior SotA benchmarks yields new SotA results - e.g. 36.25% mAP on the ActivityNet v1.3.



## References

- [1] Humam Alwassel, Silvio Giancola, and Bernard Ghanem. Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. *arXiv preprint arXiv:2011.11479*, 2020.
- [2] Yancheng Bai, Huijuan Xu, Kate Saenko, and Bernard Ghanem. Contextual multi-scale region convolutional 3d network for activity detection. *arXiv preprint arXiv:1801.09184*, 2018.
- [3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.
- [4] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–970, 2015.
- [5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.
- [6] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1130–1139, 2018.
- [7] Peihao Chen, Chuang Gan, Guangyao Shen, Wenbing Huang, Runhao Zeng, and Mingkui Tan. Relation attention for temporal action localization. *IEEE Transactions on Multimedia*, 2019.
- [8] Jian Li Chuming Lin, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *AAAI Conference on Artificial Intelligence*, 2020.
- [9] Marco Cristani, Manuele Bicego, and Vittorio Murino. Audio-visual event recognition in surveillance video sequences. *IEEE Transactions on Multimedia*, 9(2):257–267, 2007.
- [10] Xiyang Dai, Bharat Singh, Guyue Zhang, Larry S Davis, and Yan Qiu Chen. Temporal context network for activity localization in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5793–5802, 2017.
- [11] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021.
- [12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019.
- [13] Jialin Gao, Zhixiang Shi, Guanshuo Wang, Jiani Li, Yufeng Yuan, Shiming Ge, and Xi Zhou. Accurate temporal action proposal generation with relation-aware pyramid network. In *AAAI*, pages 10810–10817, 2020.
- [14] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Cascaded boundary regression for temporal action detection. *arXiv preprint arXiv:1705.01180*, 2017.
- [15] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Socccernet: A scalable dataset for action spotting in soccer videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1711–1721, 2018.
- [16] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019.
- [17] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. *arXiv preprint arXiv:1711.01467*, 2017.
- [18] Tao He, David Richter, Zhiguo Wang, and Floris P de Lange. Spatial and temporal context jointly modulate the sensory response within the ventral visual stream. *bioRxiv*, 2020.
- [19] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
- [20] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos “in the wild”. *Computer Vision and Image Understanding*, 155:1–23, 2017.
- [21] Meng Jiang, Peng Cui, Rui Liu, Qiang Yang, Fei Wang, Wenwu Zhu, and Shiqiang Yang. Social contextual recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 45–54, 2012.
- [22] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.
- [23] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3898, 2019.
- [24] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. *ACM*, 2017.
- [25] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.
- [26] Daochang Liu, Tingting Jiang, and Yizhou Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1298–1307, 2019.
- [27] Qinying Liu and Zilei Wang. Progressive boundary refinement network for temporal action detection. In *AAAI*, pages 11612–11619, 2020.
- [28] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. *CVPR*, 2018.
- [29] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks

- for action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 344–353, 2019.
- [30] Xiang Long, Chuang Gan, Gerard De Melo, Jiajun Wu, Xiao Liu, and Shilei Wen. Attention clusters: Purely attention based local feature integration for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7834–7843, 2018.
- [31] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6752–6761, 2018.
- [32] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 563–579, 2018.
- [33] Haonan Qiu, Yingbin Zheng, Hao Ye, Yao Lu, Feng Wang, and Liang He. Precise temporal action localization by evolving temporal proposals. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 388–396, 2018.
- [34] Niamul Quader, Md Mafijul Islam Bhuiyan, Juwei Lu, Peng Dai, and Wei Li. Weight excitation: Built-in attention mechanisms in convolutional neural networks. In *European Conference on Computer Vision*, pages 87–103. Springer, 2020.
- [35] Niamul Quader, Peng Lu, Juwei cnd Dai, and Wei Li. Towards efficient coarse-to-fine networks for action and gesture recognition. In *European Conference on Computer Vision*, pages 35–51. Springer, 2020.
- [36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- [37] Nataliya Shapovalova, Michalis Raptis, Leonid Sigal, and Greg Mori. Action is in the eye of the beholder: Eye-gaze driven model for spatio-temporal action localization. In *Advances in Neural Information Processing Systems*, pages 2409–2417. Citeseer, 2013.
- [38] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*, 2015.
- [39] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5734–5743, 2017.
- [40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.
- [41] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. *CVPR*, 2017.
- [42] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.
- [43] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.
- [44] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv preprint arXiv:1712.04851*, 1(2):5, 2017.
- [45] Yuanjun Xiong, Yue Zhao, Limin Wang, Dahua Lin, and Xiaoou Tang. A pursuit of temporal accuracy in general activity detection. *arXiv preprint arXiv:1703.02716*, 2017.
- [46] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020.
- [47] Yan Yan, Chenliang Xu, Dawen Cai, and Jason J Corso. Weakly supervised actor-action segmentation via robust multi-task ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1298–1307, 2017.
- [48] Dahua Lin Yue Zhao, Yuanjun Xiong. Mmaction, 2019.
- [49] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7094–7103, 2019.
- [50] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. Bottom-up temporal action localization with mutual regularization. In *European Conference on Computer Vision*, pages 539–555. Springer, 2020.
- [51] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017.
- [52] Y Zhao, B Zhang, Z Wu, S Yang, L Zhou, S Yan, L Wang, Y Xiong, D Lin, Y Qiao, et al. Cuhk & ethz & siat submission to activitynet challenge 2017. *arXiv preprint arXiv:1710.08011*, 8, 2017.