SegTAD: Precise Temporal Action Detection via Semantic Segmentation

Chen Zhao, Merey Ramazanova, Mengmeng Xu, and Bernard Ghanem

King Abdullah University of Science and Technology (KAUST), Saudi Arabia
{chen.zhao, merey.ramazanova, mengmeng.xu, bernard.ghanem}@kaust.edu.sa

Abstract. Temporal action detection (TAD) is an important yet challenging task in video analysis. Most existing works draw inspiration from image object detection and tend to reformulate it as a proposal generation - classification problem. However, there are two caveats with this paradigm. *First*, proposals are not equipped with annotated labels, which have to be empirically compiled, thus the information in the annotations is not necessarily precisely employed in the model training process. *Second*, there are large variations in the temporal scale of actions, and neglecting this fact may lead to deficient representation in the video features. To address these issues and *precisely* model TAD, we formulate the task in a novel perspective of semantic segmentation. Owing to the 1-dimensional property of TAD, we are able to convert the coarse-grained detection annotations to fine-grained semantic segmentation annotations for free. We take advantage of them to provide precise supervision so as to mitigate the impact induced by the imprecise proposal labels. We propose a unified framework SegTAD composed of a 1D semantic segmentation network (1D-SSN) and a proposal detection network (PDN). We evaluate SegTAD on two important large-scale datasets for action detection and it shows competitive performance on both datasets.

1 Introduction

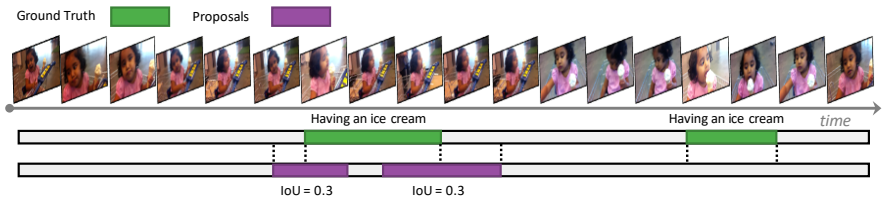

Fig. 1: **Proposal Annotations.** Proposals different in locations, lengths, and content are assigned the same label if they have the same IoU.

Nowadays, millions of videos are produced every day, and high demand arises for automatic video processing and analysis. To this end, various tasks have

emerged, for example, action recognition [19], active speaker detection [2], video-language grounding [41], temporal action localization [26,42]. Among those tasks, temporal action detection in untrimmed videos, in particular, is one of the fundamental yet challenging tasks. It requires not only to recognize what actions take place in a video but also to localize when they start and end.

Most recent works in the literature regard this task as a temporal version of object detection and tackle it by adapting the 2-dimensional solutions on images (e.g., Faster R-CNN [36]) to the 1-dimensional temporal domain for videos [23,5,47,17]. A conventional pipeline is to first identify candidate action segments (i.e., proposals) by analyzing the entire video sequence and then learn to score each segment with an empirically compiled label for each proposal. This object-detection inspired framework has brought significant improvement on the action detection performance [17], especially with the aid of deep neural network in recent years [47,5,48,27]. However, it lays two caveats that might lead to imprecise action detection modeling.

First, proposals are not accompanied by any annotated labels from the dataset since they are generated on the fly. Their training labels have to be manually compiled based on the ground-truth *action* annotations, i.e., the start/end timestamps of *actions* in each video and their corresponding categories. A common practice is to compare each proposal to each ground-truth action in the video in terms of some metric (e.g., temporal Intersection over Union) and use a preset threshold to determine whether a proposal is positive or negative with respect to each category. However, this is obviously not an optimal approach considering that the mapping between action annotations and proposal annotations are not bijective (as shown in Fig. 1). Noise is inevitably introduced to the compiled proposal labels regardless of what metric or threshold is adopted, resulting in imprecise modeling. Note that even in object detection, it is still an open question on how to identify positive and negative proposals, which is crucial to detection performance [52].

Second, object detection is a relatively coarse-grained problem that does not identify every single pixel but predicts a rectangular box surrounding an entire object. However, videos especially in large-scale datasets, e.g., ActivityNet [10], HACS [55], Ego4D [14] contain actions of dramatically varied temporal duration - from less than a second to minutes. Therefore, shifting from the image domain to the video domain without adapting to the video diversity could lead to deficient feature representation (e.g., burying short actions and under-representing long actions by imprecisely modeling temporal correlations), as well as misalignment between proposals and their receptive fields [5].

To address these issues, in this paper, we propose to formulate the task of temporal action detection (TAD) in a novel perspective with semantic segmentation. In the image 2-dimensional (2D) domain, much more effort is demanded to obtain finer-grained annotations for the tasks such as semantic segmentation, considering that not all pixels in a detection bounding box are contained in the object. In contrast, the task of video TAD requires only 1-dimensional (1D) localization of actions — along the temporal domain. Therefore, all frames within the action boundaries naturally belong to the action category. The detec-

tion annotations can be bijectively transformed to segmentation labels without extra effort. We propose a unified TAD framework to take advantage of the fine-grained prediction of semantic segmentation for more precise detection, dubbed as SegTAD. SegTAD contains a 1D semantic segmentation network to learn the category of each single frame using the segmentation labels, which are directly transformed from the detection annotations without introducing any label compilation noise. Regarding the second issue, we design SegTAD modules based on atrous and graph convolutions to precisely represent actions of various temporal duration. **The main contributions are:**
1) We formulate TAD in a novel perspective of semantic segmentation and propose a unified TAD framework SegTAD, which is composed of a 1D semantic segmentation network (1D-SSN) and a proposal detection network (PDN).
2) In 1D-SSN, we design an hourglass architecture with a module of parallel astrous and graph convolutions to effectively aggregate global features and multi-scale local features. In PDN, we incorporate a proposal graph to exploit cross-proposal correlations in our unified framework.
3) The proposed SegTAD achieves competitive performance on two representative large-scale datasets ActivityNet-v1.3 [10] and HACS-v1.1 [55].

2 Related Work

2.1 Temporal Action Detection

Concurrent TAD methods tend to adopt the two-stage framework: 1) generating candidate action segments (i.e., proposals) from the video sequence; 2) cropping each proposal out of the sequence, and classifying each proposal to obtain its confidence score. A large number of these methods focus on improving the first stage to generate proposals with high recall, applying an off-the-shelf classifier (e.g., SVM) to get the detection results [3,17,9,29,11]. Some other methods focus on the second stage, seeking to build more accurate classifiers on proposals produced by other proposal methods (e.g., sliding windows, the above-mentioned first-stage methods) [39,38,51,57,35]. The third category of methods propose unified approaches, where the features of different frames are aggregated and the actions are predicted from the same network [16,5,47,50,48,30,25,22,54]. Our paper belongs to the third category. Among these methods, our SegTAD is related to but essentially different from them in the following aspects.

Snippet-level classification. Multiple methods have identified the coarse granularity and regular distribution issues of anchor-based proposals, such as BSN [25], TAG [57], MGG [29], and CTAP [11]. They have proposed to incorporate snippet-level proposals as a supplement or replacement to the anchor-based ones. They learn a binary classifier for each snippet, either by a 2D convolutional neural network (CNN) on each snippet [11,57] or applying a temporal CNN on the entire sequence [29,25]. By this means, they obtain the probability of being an action/start/end for each snippet, based on which to generate proposals with flexible duration. In addition, the second-stage method CDC [38] also classifies each snippet, but the purpose is to refine an existing proposal instead. In this

paper, the proposed SegTAD directly formulates a 1D semantic segmentation problem to classify every single frame into different action categories. It enables the use of large temporal resolution and supports multi-scale feature aggregation with the proposed PAG module. Moreover, it doesn't rely on the actionness/startness/endness scores to generate proposals as in TAG or BSN.

Snippet-and-snippet correlations. The method G-TAD [48] exploits temporal correlations between snippets by adopting a graph convolutional network (GCN). It supports limited temporal resolution due to its lack of multi-scale design and the complexity constraint of GCN when more frames are utilized, consequently sacrificing actions of short duration. Comparatively, our SegTAD adopts an hourglass architecture with an encoder and decoder, and only apply graph convolutions in the intermediate layer with the smallest resolution. In this way, it aggregates global information while preserving the temporal resolution.

Proposal-and-proposal correlations. BMN [22] constructs a boundary map with densely-distributed proposals and apply convolutions on the map to utilize the correlations between proposals, whereas 2D-TAN [53] presents a sparse 2D temporal feature map to represent and correlate proposals. The second-stage method P-GCN [51] uses GCNs [28] on proposals obtained by other methods to improve the proposal scores and boundaries. Our SegTAD incorporates graph and edge convolutions to our *unified* detection framework to exploit cross-proposal correlations. Compared to the standalone P-GCN [51], which is essentially a proposal post-processing method and does not consider correlations between frames, SegTAD jointly learns the graph network with the 1D semantic segmentation network and enhances feature representations via cross-frame and cross-proposal aggregation.

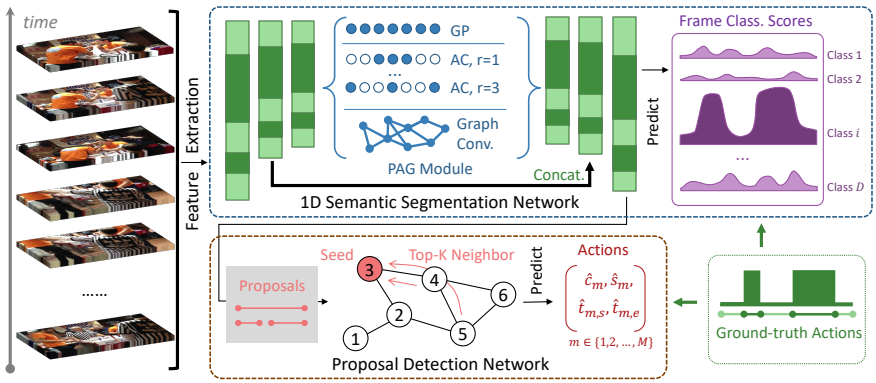

Fig. 2: **Illustration of our proposed SegTAD architecture. Input:** a sequence of video frames; **Output:** scored candidate actions. **Top:** 1D Semantic segmentation network (1D-SSN) that learns to classify each frame in the sequence. We design a module of parallel atrous and graph convolutions (PAG) to effectively aggregate global features and multi-scale local features. **Bottom:** Proposal detection network (PDN) that scores each candidate action. Graph convolutions are utilized to exploit correlations between proposals.

2.2 Object Detection and Semantic Segmentation

In the image domain, object detection [13,12,36,34,6] is a coarse-grained prediction problem, whose output is a rectangular bounding box that surrounds an object in the image. A widely adopted framework for tackling this task is the two-stage method (e.g., R-CNN [13], Fast R-CNN [12], and Faster R-CNN [36]), which first generates candidate proposals from the original image, and then runs a classifier for each proposal. Recent TAD methods tend to draw inspiration from these object detection methods. But by noticing the 1-dimensional property of TAD problem, we see that besides object detection, TAD has another analogy in the image domain, which is semantic segmentation.

Semantic segmentation [37,31,7,20,18] is a fine-grained prediction task that predicts the class label of every pixel in an image. Thus, annotating for segmentation usually requires extraordinarily more efforts than for object detection. Compared to object detection which resorts to proposals, semantic segmentation usually adopts a different framework, which seeks to preserve the dense grid of the input image while learning its high-level semantic features with a convolutional network. Representative works are U-Net [37] and FCN [31], etc. In videos, temporal semantic annotations and TAD annotations are bijectively transferable, so no extra efforts are required to annotate segmentation. In this work, taking advantage of this discovery, we utilize the semantic segmentation methodology to formulate TAD.

3 Proposed SegTAD

3.1 Problem Formulation and SegTAD Framework

The task of temporal action detection (TAD) is to predict a set of actions $\Phi = \{\phi_m = (t_{m,s}, t_{m,e}, c_m, s_m)\}_{m=1}^{M}$ given a sequence of T video frames $\{I_t \in \mathbb{R}^{3\times H\times W}\}_{t=1}^{T}$, where $t_{m,s}$ and $t_{m,e}$ are action start and end time respectively, c_m is action label, and s_m is prediction confidence. To achieve this, we first transform each frame I_t to a 1-dimensional feature vector $\mathbf{x}_t \in \mathbb{R}^C$ via a feature extraction network (see Sec. 4.1). Using the 1D features $\mathbf{x}_t$ as input, we apply our proposed 1D semantic segmentation network (1D-SSN), followed by a proposal detection network (PDN). The 1D-SSN temporally aggregates features to learn to segment the video sequence in the frame level according to the action annotations, and generates semantic features $\mathbf{y}_t \in \mathbb{R}^{C'}$ for each frame. The PDN learns to score each candidate action and further refines its boundaries. The two components 1D-SSN and PDN are trained in a unified architecture.

We illustrate the entire architecture of SegTAD in Fig. 2. It shows two main components: 1D semantic segmentation network and proposal detection network, which will be described in the following subsections, respectively.

3.2 1D Semantic Segmentation Network

Different from conventional TAD works, which perform prediction in the coarse segment level and compile segment labels from ground-truth action annotations,

we use 1D semantic segmentation (1D-SSN) to learn to predict for each single frame. Based on our 1D-SSN, we are able to take advantage of the original true action annotations without introducing any label noise. In the following, we first describe the 1D-SSN architecture that aggregates features from a global temporal range as well as multi-scale local range. Then, we present our segmentation loss that uses the original action annotations to train the segmentation network.

1D-SSN Architecture Sufficient semantic information from a long temporal range is essential for TAD. This is usually achieved by enlarging the receptive field via strided 1D convolution or pooling. However, aggressively using these operations will dramatically reduce the temporal resolution and severely impair the feature representation of short actions.

To achieve a large receptive field without severely sacrificing temporal resolution, we consider feature aggregation in two aspects: local feature aggregation and global feature aggregation. The former aggregates features in a surrounding temporal window to learn local patterns. We need to make it scale-invariant to represent actions of different duration. The latter associates features in a global range, not constrained in the neighborhood of each frame. This breaks the constraints of the temporal locations of each frame and makes use the correlations between frames in the global context [48].

For the two aspects, we design an hourglass architecture with atrous and graph convolutions in our 1D-SSN. It has the shape of an hourglass, containing an encoder, a parallel module of atrous and graph convolutions (PAG), and a decoder. The encoder temporally downscales the input features by a small ratio, and the decoder is to restore the temporal resolution. The PAG module enables local feature aggregation in multiple scales using atrous convolutions and global feature aggregation via graph convolutions.

1D-SSN Details **The encoder** is comprised of a stack of L strided 1D convolution layers Conv1D(k=3,s=2), where k is the kernel size, s is the stride, followed by the non-linear activation function ReLU. We only have L=3 such layers in order not to overly downscale the features. It applies along the temporal dimension on the input video feature sequence $\mathbf{X}=[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T] \in \mathbb{R}^{C \times T}$, where C is the input feature dimension, and transforms it into a representation with lower temporal resolution $\mathbf{X}'=[\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_{T/2^L}] \in \mathbb{R}^{C' \times T/2^L}$, where C' is the new feature dimension. The encoder reduces the sequence temporal resolution by a factor of 2^L so as to reduce computation for the subsequent layers, as well as to progressively increase the size of temporal receptive field.

The decoder upscales the temporal resolution of the features $\mathbf{X}'' = \text{PAG}(\mathbf{X}')$, where PAG stands for operations in the module of parallel atrous and graph convolutions (PAG) (detailed in following paragraphs). It contains a layer of linear interpolation to rescale the features along the temporal dimension to the orignal resolution. In order to complement the details information lost from the encoder, we add a highway connection from the low-level features at the second Conv1D layer in the encoder, which consists of Conv1D($k = 1, s = 1$), batch normalization and ReLU. Then we concatenate the output of this connection with the interpolated features, and apply a Conv1D($k = 3, s = 1$) layer to adaptively fuse

them. With this hourglass (encoder-decoder) architecture, we gradually aggregate features from frames further apart while preserving the temporal resolution of the sequences.

The module of parallel atrous and graph convolutions (PAG) (Fig. 2) takes the encoded features $\mathbf{X}'$ as input, and further enlarges the receptive field and empower the features with scale-invariant capability. Considering that a video sequence usually contains actions of various temporal duration, ranging from a couple of seconds to minutes. Excessive pooling or using strided convolutions could impair short actions as mentioned above, whereas long actions require large receptive field to be semantically represented. To adapt to actions of variant temporal scales, we propose this PAG module, which contains atrous convolutions to aggregate features from multi-scale local neighborhood, and graph convolutions to aggregate features from global context. Note that unlike G-TAD [48], which applies graph convolutions in every layer and consequently incurs huge computation cost, we only have them in this intermediate module after the resolution is reduced by the encoder. In this way, it aggregates global information while preserving the temporal resolution.

Atrous convolutions systematically aggregate multi-scale contextual information without losing resolution. They are able to support expansion of the receptive field [49] by filling in empty elements in the convolutional filter. Compared to normal convolutions, they are equipped with a dilation ratio d to specify the number of empty elements in the filter, reflecting the expansion ratio of the receptive field. We use 4 parallel branches of 1D atrous convolutions with different dilation ratios. The choice of dilation ratios will be discussed in Sec. 4.3.

Graph convolutions model the correlations among snippets in a non-local context. We design a graph convolutional network in parallel with the multiple branches of atrous convolutions. Specifically, based on the output features from the encoder $\{\mathbf{x}'_t\}_{t=1}^{T/2^L}$ (we call $\mathbf{x}'_t$ features of a snippet in the following), we build a graph denoted as $\mathcal{G}_s = \{\mathcal{V}_s, \mathcal{E}_s\}$. $\mathcal{V}_s = \{v_t\}_{t=1}^{T/2^L}$ refers to the graph nodes, each corresponding to a snippet, and $\mathcal{E}$ denotes graph edges, which represent the correlations between snippets. To model the correlations of snippets in a global context, we construct the edges dynamically [45] according to the semantic similarity between encoded snippet features rather than their temporal locations, which are computed as minus mean square error (MSE) between two feature vectors. If a snippet is among the top K nearest neighbors of another snippet in terms of their semantic similarity, there is an edge connecting them.

With this graph, we apply one layer of edge convolutions to aggregate features of connected nodes [45], formulated as

$$\mathbf{X_{GC}} = ([\mathbf{X'}^T,\ \mathbf{A}\mathbf{X'}^T - \mathbf{X'}^T]\mathbf{W})^T, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{T/2^L \times T/2^L}$ is the adjacency matrix defined by edge connections between snippets, its $(i,j)^{\text{th}}$ element $\mathbf{a}_{i,j} = 1$ if there is an edge between the i^{th} and the j^{th} snippets, and $\mathbf{a}_{i,j} = 0$, otherwise. For each snippet, $\mathbf{A}\mathbf{X'}^T$ aggregates features from all its connected snippets in the whole sequence. The operation $[\cdot,\cdot]$ concatenates the two feature vectors. $\mathbf{W} \in \mathbb{R}^{C' \times C'}$ denotes trainable parameters.

In order to aggregate the global context information along the temporal dimension, we add a global fast path that first does global average pooling and linearly upsamples back to the original resolution. This mitigates the weight validity issue when large dilation ratios are used [7]. Then we concatenate the output of all atrous convolutional (AC) branches and the graph convolution (GC) network as well as the global fast path (GP), formulated as

$$\mathrm{PAG}(\mathbf{X}') = [\mathbf{X}_{\mathrm{GC}}, \mathbf{X}^1_{\mathrm{AC}}, \ldots, \mathbf{X}^B_{\mathrm{AC}}, \mathbf{X}_{\mathrm{GP}}]. \tag{2}$$

Finally, a Conv1D layer followed by ReLU is applied to fuse all branches.

Segmentation Loss In 1D-SSN, we formulate TAD as a semantic segmentation problem, and predict the category of each single frame to meet their true categories. In the following, we describe how to generate predictions, and formulate the segmentation loss using action annotations.

The output from the decoder in 1D-SSN is a sequence of aggregated feature vectors $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T] \in \mathbb{R}^{C' \times T}$. We use this to predict per-frame classification labels. Suppose we have D action categories, applying one layer of linear transformation and Softmax operation yields

$$\mathbf{p}_t = \mathrm{Softmax}(\mathbf{W}^T_{seg}\, \mathbf{y}_t), \tag{3}$$

where $\mathbf{p}_t \in \mathbb{R}^D$ refers to the predicted label for t^{th} frame, $\mathbf{W}_{seg} \in \mathbb{R}^{C' \times D}$ contains the parameters in the linear layer.

If we know the ground-truth label $b_t \in \{1, 2, 3, \ldots, D\}$ of each frame, then we can compute the segmentation loss using cross-entropy formulated as follows

$$\mathcal{L}_{seg} = -\frac{1}{T} \sum_{1 \le t \le T} \sum_{1 \le d \le D} \beta_{t,d} \log p_{t,d}, \tag{4}$$

where $\beta_{t,d} = \mathbb{1}\{b_t = d\}$ is the d^{th} one-hot encoding of the label for the t^{th} frame.

Now the question becomes how we obtain the segmentation label b_t. Assume that a video sequence is annotated with N actions $\Psi = \{\psi_n = (t_{n,s}, t_{n,e}, c_n)\}_{n=1}^N$, where $t_{n,s}$ and $t_{n,e}$ denote the start and end time of the n^{th} action instance, respectively, and c_n represents its category. This is a segment-level annotation, without specifying the exact labels of each frame. However, due to the 1D characteristic of TAD, we can easily transform this segment-level annotations to finer-grained frame-level labels. It assigns a frame the label of an action if it falls inside the action boundaries, otherwise, the frame is labeled as background. We see that in the entire process, there is no hyper-parameter and the mapping is bijective, which guarantees precise annotation transformation.

Additionally, considering that action boundaries are important for localize an action, we introduce an auxiliary loss

$$\begin{aligned} \mathcal{L}_{aux} &= \frac{-\sum_{1 \le t \le T} \beta^s_t \log p^s_t + (1 - \beta^s_t) \log(1 - p^s_t)}{T} \\ &= \frac{-\sum_{1 \le t \le T} \beta^e_t \log p^e_t + (1 - \beta^e_t) \log(1 - p^e_t)}{T}, \end{aligned} \tag{5}$$

where $\beta_t^s, \beta_t^e \in \{0, 1\}$ are start and end labels that indicate whether a frame is the first or last frame of an action. p_t^s and p_t^e are predicted confidence scores of a frame being start and end of action, which are generated by

$$\mathbf{p}^s = \text{Sigmoid}(\mathbf{w}_{s2}^T \text{ReLU}(\text{Conv1d}_{k,s=3,1}(\mathbf{Y}; \mathbf{W}_{s1}))) \tag{6}$$

$$\mathbf{p}^e = \text{Sigmoid}(\mathbf{w}_{e2}^T \text{ReLU}(\text{Conv1d}_{k,s=3,1}(\mathbf{Y}; \mathbf{W}_{e1}))) \tag{7}$$

where $\mathbf{W}_{s1}$ and $\mathbf{W}_{e1}$ represent convolutional kernels, and $\mathbf{w}_{s2}, \mathbf{w}_{e2} \in \mathbb{R}^{C' \times 1}$ are parameters of the linear layers.

3.3 Proposal Detection Network

Considering that the actions in the format of $\Phi = \{\phi_m = (t_{m,s}, t_{m,e}, c_m, s_m)\}_{m=1}^M$ are not predicted directly by 1D-SSN, we need an extra detection head, for which we design a proposal detection network (PDN). PDN takes the output features from 1D-SSN along with our designed sparse segment patterns as input, and generate predicted actions. This PDN takes advantage of cross-proposal correlations via a graph network, and further enhances the representation of each frame and each proposal.

PDN Architecture: As shown in Fig. 2, PDN takes the output features $\mathbf{Y}$ from 1D-SSN as well as a sparse pattern of segments. In our framework, in order to precisely detection short actions, we use a high temporal resolution $L = 1000$. Therefore, it is cumbersome to enumerate all possible pairs of frames as proposals as done in [48] and [22]. Instead, we design a sparse pattern of segments, which covers a large variety of action duration and reduces computation compared to dense segments. Let each element $u_{i,j} \in \{0, 1\}$ of the matrix $\mathbf{U} \in \mathbb{R}^{L \times L}$ denote whether the segment starting from i^{th} frame and composed of j frames is selected as a proposal. Its value is determined by the following equation

$$u_{i,j} = \begin{cases} 1, & \text{if } i \,\%\, \eta = 0 \text{ and } j \,\%\, \eta = 0; \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

where η=8 is a step size controlling the sparsity degree. With Φ=$\{\phi_m$=$(t_{m,s}, t_{m,e})\}_{m=1}^M$ being all M proposals specified by $\mathbf{U}$, their features $\mathbf{D}$=$\{\mathbf{d}_m \in \mathbb{R}^{C'}\}_{m=1}^M$ are extracted from video features $\mathbf{Y}$ based on SGAlign [48].

The proposals in the same video are highly correlated and utilizing this property can enhance proposal representations [22]. To model correlations between proposals from any temporal locations, we build a second graph $\mathcal{G}_p = \{\mathcal{V}_p, \mathcal{E}_p\}$ on the proposals and take advantage of graph convolutions in the detection network. Different from the graph $\mathcal{G}_s$ in 1D-SSN, each node in $\mathcal{G}_p$ refers to one proposal, and the edges represent correlations between proposals. Another difference is that the edges $\mathcal{E}_p$ here are constructed based on the temporal intersection over union (tIoU) between proposals, as opposed to the dynamically determined edges in 1D-SSN. We apply the same edge convolutions as shown in Eq. (1), but define each element $a_{i,j}$ in the adjacency matrix $\mathbf{A}$ as an attention value computed as

$a_{i,j} = \mathbf{d}_i^T \mathbf{d}_j / |\mathbf{d}_i| \cdot |\mathbf{d}_j|$ if there is an edge. We stack 3 layers of edge convolutions in PDN.

In order to efficiently train the proposal network as well as to balance the positive and negative samples, we need to sample from our M proposals. Randomly sampling does not guarantee that the sampled proposals have consistent edge connections with each other to form a meaningful graph. So a better strategy is to sample neighborhoods of proposals rather than individual proposals. We adopt the following sampling strategy based on the SAGE method [15]. We first sample a small number of M_0 seed proposals, including $M_0/2$ positive and $M_0/2$ negative samples. Then for each seed proposal, we find its top K neighbors based on its tIoU with other proposals, and put them into the sampling list. For each of the K neighboring proposals, we find its top K neighbors from the remaining proposals, and add these $K \times K$ proposals into the sampling list as well. Hereby, in the sampling list, we totally have $M_0(1 + K + K \times K)$ proposals, all of which have their top K neighbors in the list. $M_0 = 50$ and $K = 4$ by default. In inference, we use all M proposals without sampling.

Detection Loss: The PDN enhances the feature representation of each proposal by aggregating different proposals, formulated as $\mathbf{D}' = \text{PDN}(\mathbf{D})$. We predict proposals' confidence of being actions using the following operation

$$\mathbf{S} = \text{Sigmoid}(\mathbf{W}_{det}^T \mathbf{D}'), \tag{9}$$

where $\mathbf{W}_{det} \in \mathbb{R}^{C' \times 2}$ contains the parameters in the linear layer to predict the confidence scores. Note that $\mathbf{S} = [\mathbf{s}_1; \mathbf{s}_2]$ contains two different scores for each proposal, each corresponding to one loss function we define in the following

$$\mathcal{L}_{det} = \mathcal{L}_{reg}(\mathbf{h}_{reg}, \mathbf{s}_1) + \mathcal{L}_{cls}(\mathbf{h}_{cls}, \mathbf{s}_2), \tag{10}$$

where $\mathbf{h}_{reg}$ is the tIoU between each proposals and their closest ground-truth actions, and $\mathcal{L}_{reg}$ is computed using mean square errors. $\mathbf{h}_{cls} = \mathbb{1}(\mathbf{h}_{reg} > \tau)$, where $\tau = 0.5$ is an tIoU threshold determining a proposal's binary label, and $\mathcal{L}_{cls}$ is a binary cross-entropy loss similarly computed as either term in Eq. (5).

3.4 Training and Inference

Training: We train the proposed SegTAD end to end using the segmentation loss and the detection loss, as well as the auxiliary loss as follows

$$\mathcal{L} = \mathcal{L}_{seg} + \lambda_1 \mathcal{L}_{det} + \lambda_2 \mathcal{L}_{aux} + \lambda_3 \mathcal{L}_r, \tag{11}$$

where $\mathcal{L}_r$ is $\mathcal{L}_2$-norm, λ_1, λ_2, λ_3 denotes the weights for different loss terms, which are all set to 1 by default.

Inference: At inference time, we compute the score of each candidate action using the two scores predicted from the proposal detection network as $s_m = s_{m,1} \times s_{m,2}$. Then we run soft non-maximum suppression (NMS) using the scores and keep the top 100 predicted segments as final output.

Table 1: **Action detection result comparisons on validation set of ActivityNet-v1.3**, measured by mAP (%) at different tIoU thresholds and the average mAP. G-TAD achieves better performance in average mAP than the other methods, even the latest work of BMN and P-GCN shown in the second-to-last block. (* Re-implemented with the same features as ours.)

Method	0.5	0.75	0.95	**Average**
Wang *et al.* [44]	43.65	-	-	-
Singh *et al.* [40]	34.47	-	-	-
SCC [16]	40.00	17.90	4.70	21.70
Lin *et al.* [24]	44.39	29.65	7.09	29.17
CDC [38]	45.30	26.00	0.20	23.80
TCN [8]	37.49	23.47	4.47	23.58
R-C3D [47]	26.80	-	-	-
SSN [57]	34.47	-	-	-
BSN [25]	46.45	29.96	8.02	30.03
TAL-Net [5]	38.23	18.30	1.30	20.22
P-GCN+BSN [51]	48.26	33.16	3.27	31.11
BMN [22]	<u>50.07</u>	**34.78**	8.29	<u>33.85</u>
BMN* [22]	48.56	33.66	<u>9.06</u>	33.16
I.C & I.C [56]	43.47	33.91	**9.21**	30.12
SegTAD (top-1 cls.)	49.86	34.37	6.50	33.53
SegTAD (top-2 cls.)	**50.52**	<u>34.76</u>	6.85	**33.99**

Table 2: **Action detection results on HACS-v1.1**, measured by mAP (%) at different tIoU thresholds and the average mAP.

Method	Validation				Test
	0.5	0.75	0.95	Average	Average
SSN [55]	28.82	18.80	5.32	18.97	16.10
BMN [1]	-	-	-	-	22.10
S-2D-TAN [53]	-	-	-	-	23.49
SegTAD	**43.33**	**29.65**	**6.23**	**29.24**	**28.90**

4 Experimental Results

4.1 Datasets and Implementation Details

Datasets and Evaluation Metric. We conduct our experiments on two large-scale action understanding dataset, **ActivityNet-v1.3** [10] and **HACS-v1.1** [55] for TAD. **ActivityNet-v1.3** contains around $20,000$ temporally annotated untrimmed videos with 200 action categories. Those videos are randomly divided into training, validation and testing sets by the ratio of 2:1:1. **HACS-v1.1** follows the same annotation scheme as ActivityNet-v1.3. It also includes 200 action categories but collects $50,000$ untrimmed videos for TAD. We evaluate SegTAD performance with the average of mean Average Precision (mAP) over 10 different IoU thresholds [0.5:0.05:0.95] on both datasets.

Implementations. For ActivityNet-v1.3, we sample each video at 5 frames per second and adopt the two-stream network by Xiong et. al. [46] pre-trained on Kinetics-400 [4] to extract frame-level features, and rescale each sequence into 1000 snippets as SegTAD input. For HACS, we use the publicly available features extracted using an I3D-50 [4] model pre-trained on Kinetics-400 [4] and temporally rescale them into 400 snippets. We implement and test our framework using PyTorch 1.1, Python 3.7, and CUDA 10.0. In training, the learning rates are 1e−5 on ActivityNet-1.3 and 2e−3 on HACS-v1.1 for the first 7 epochs, and are reduced by 10 for the following 8 epochs. In inference, we leverage the global video context and take the top-1 or top-2 video classification scores from the action recognition models of [43] and [21], respectively for the two datasets, and multiply them by the confidence score c_j for evaluation.

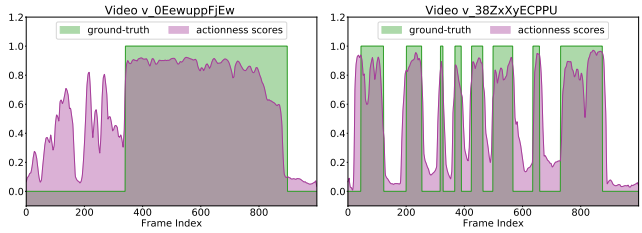

Fig. 3: **Predicted per-frame classification scores compared to ground-truth labels.** We only plot the scores of the ground-truth category. Green curves represent the ground-truth, and score = 1.0 represents the frames are inside action, and score =0.0 otherwise. Purple curves represent the predicted scores.

4.2 Comparison to State-of-the-Art

In Table 1 and Table 2, we compare SegTAD with representative TAD works in the literature. We report mAP at different tIoU thresholds and average mAP.

On ActivityNet-v1.3, SegTAD achieves competitive average mAP of 33.99%, significantly outperforming the recent works I.C & I.C [56] and BMN [22]. Notably, BMN extracts video features from ActivityNet-finetuned model such that the extracted features are more distinguishable on the target dataset. In contrast, we use more general Kinetics-pretrained features. To achieve fair comparison, we also show the re-produced BMN experimental results with the same features as ours, and our performance gain is even more remarkable. On HACS-v1.1, SegTAD reaches 28.90% average mAP on the test set, surpassing both the challenge winner S-2d-TAN [53] and BMN [22] by large margins. Compared with ActivityNet-v1.3, HACS-v1.1 is more challenging because of its substantial data-scale and precise segment annotations. Therefore, our superior performance on HACS-v1.1 makes SegTAD more remarkable.

4.3 Ablation Study

We provide ablation study to demonstrate the importance of the proposed 1D semantic segmentation network to the detection performance. Also we verify the

Table 3: **Effectiveness of our segmentation network.**

Segmentation loss	0.5	0.75	0.95	Avg.
✗	49.15	33.45	3.81	32.45
✓	**49.86**	**34.37**	**6.50**	**33.53**

Table 4: **Different loss functions for segmentation.**

Segment. loss types	0.5	0.75	0.95	Avg.
Binary	49.35	33.77	4.07	32.79
SegTAD	**49.86**	**34.37**	**6.50**	**33.53**

Table 5: **Ablating studies in PAG of 1D-SSN.** AC: Atrous convolutions, GC: graph convolutions, GP: global fast path.

PAG Branches			mAP at different tIoUs			
AC	GC	GP	0.5	0.75	0.95	Avg.
✗	✓	✓	48.55	33.04	5.21	32.37
✓	✗	✓	49.48	33.95	**7.50**	33.30
✓	✓	✗	49.75	34.25	5.97	33.29
✓	✓	✓	**49.86**	**34.37**	6.50	**33.53**

Table 6: **Different sets of dilation ratios of the AC branches.**

Dilation ratios	0.5	0.75	0.95	Avg.
1, 2, 4, 6	49.58	34.14	5.58	33.25
1, 6, 12, 18	49.71	34.01	5.90	33.31
1, 10, 20, 30	**49.86**	**34.37**	**6.50**	**33.53**
1, 16, 32, 64	50.00	34.31	6.35	33.45

effectiveness of our design choice for the 1D semantic segmentation network (1D-SSN) and proposal detection network (PDN). In Table 3, we compare SegTAD to its variants of disabling the segmentation loss in the 1D-SSN component. We can see that using the loss leads to obvious improvement compared to not using it. In Table 4, we show the performance of replacing the segmentation loss using a binary classification loss, which learns whether a frame is inside an action or not. Our multi-class segmentation loss in SegTAD is obviously better.

In 1D-SSN, the module of parallel atrous and graph convolutions (PAG) is important to aggregate features from multiple scales. We ablate different branches in PAG to show the performance change in Table 5. It shows that the network with all three kinds of branches produce the best performance.

We also show the results of different sets of dilation ratios for the atrous convolution branches in Table 6 and choose 1, 10, 20, 30 due to its highest mAP. We evaluate the effectiveness of our proposal detection network (PDN) by replacing its each layer of edge convolutions with a layer of Conv1D($k = 1, s = 1$) and apply it on each single proposal independently. In this way, this variant cannot make use of the cross-proposal correlations. We can see from Table 7 that using graph convolutions brings significant improvement compared to independently learning for each proposal. In Table 8, we compare different metrics to determine the similarity between proposals: tIoU and distance between proposal centers. We adopt tIoU in SegTAD due to its better performance.

4.4 Visualization of Segmentation Output

We visualize the output from the 1D semantic segmentation network and compare to ground-truth labels in Fig. 3. Our output tightly matches the ground-truth even for the video that contains many short action instances such as the bottom example. Such accurate segmentation is important for learning distinctive features for each frame, and consequently benefits the final detection.

Table 7: **Ablating the proposal detection network.**

PDN layers	0.5	0.75	0.95	Avg.
Conv1D($k = 1, s = 1$)	48.31	32.79	5.49	32.12
Graph convolutions	**49.86**	**34.37**	**6.50**	**33.53**

Table 8: **Comparing different similarity metrics: temporal intersection over union and center distance between proposals.**

Node similarity metric	0.5	0.75	0.95	Avg.
Center distance	49.34	33.57	3.62	32.52
Temp. intersection over union	**49.86**	**34.37**	**6.50**	**33.53**

5 Acknowledgement

This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research through the Visual Computing Center (VCC) funding.

6 Conclusion

In summary, we take a novel perspective to formulate TAD based on 1D semantic segmentation to achieve more accurate label assignment and precise localization. We propose SegTAD, which is composed of a 1D semantic segmentation network (1D-SSN) and a proposal detection network (PDN). To suit the large variety of action temporal duration, in 1D-SSN, we design a module of parallel atrous and graph convolutions (PAG) to aggregate multi-scale local features and global features. In PDN, we design a second graph network to model the cross-proposal correlations. SegTAD is a unified framework that is trained jointly using the segmentation and detection losses from both 1D-SSN and PDN, respectively. As a conclusion, we would like to emphasize the need to focus more on the unique characteristics of videos when dealing with detection problems in video.

Relevance to 'AI for understanding and accelerating video editing'. Given the boom of creative video content on various platforms, such as TikTok, Reels, and YouTube, the tedious and time-consuming editing process urgently needs to be transformed [32]. Our SegTAD is able to localize and recognize actions in long untrimmed videos, which is needed by creative tasks such as cutting for the movie editing [32,33]. More specifically, our well-trained model that predicts the location and the type of human actions can be used to find the places where the transition of the scene should happen.

References

1. Report of Temporal Action Proposal. `http://hacs.csail.mit.edu/challenge/challenge19_report_runnerup.pdf` (2020)
2. Alcazar, J.L., Cordes, M., Zhao, C., Ghanem, B.: End-to-end active speaker detection. Proceedings of European Conference on Computer Vision (ECCV) (2022)
3. Buch, S., Escorcia, V., Shen, C., Ghanem, B., Niebles, J.C.: SST: single-stream temporal action proposals. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
4. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
5. Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the faster r-cnn architecture for temporal action localization. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
6. Chen, C., Ling, Q.: Adaptive convolution for object detection. IEEE Transactions on Multimedia (TMM) **21**(12), 3205–3217 (2019). https://doi.org/10.1109/TMM.2019.2916104
7. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. ArXiv **abs/1706.05587** (2017)
8. Dai, X., Singh, B., Zhang, G., Davis, L.S., Qiu Chen, Y.: Temporal context network for activity localization in videos. Proceedings of IEEE International Conference on Computer Vision (ICCV) (2017)
9. Escorcia, V., Heilbron, F.C., Niebles, J.C., Ghanem, B.: DAPs: deep action proposals for action understanding. Proceedings of European Conference on Computer Vision (ECCV) (2016)
10. Fabian Caba Heilbron, Victor Escorcia, B.G., Niebles, J.C.: Activitynet: A large-scale video benchmark for human activity understanding. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
11. Gao, J., Chen, K., Nevatia, R.: Ctap: Complementary temporal action proposal generation. Proceedings of European Conference on Computer Vision (ECCV) (2018)
12. Girshick, R.B.: Fast R-CNN. Proceedings of IEEE International Conference on Computer Vision (ICCV) (2015)
13. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
14. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al.: Ego4d: Around the world in 3,000 hours of egocentric video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18995–19012 (2022)
15. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. Proceedings of Neural Information Processing Systems (NeurIPS) (2017)
16. Heilbron, F.C., Barrios, W., Escorcia, V., Ghanem, B.: SCC: semantic context cascade for efficient action detection. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
17. Heilbron, F.C., Niebles, J.C., Ghanem, B.: Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

18. Kang, B., Lee, Y., Nguyen, T.Q.: Depth-adaptive deep neural network for semantic segmentation. IEEE Transactions on Multimedia (TMM) **20**(9), 2478–2490 (2018). https://doi.org/10.1109/TMM.2018.2798282
19. Li, J., Liu, X., Zhang, W., Zhang, M., Song, J., Sebe, N.: Spatio-temporal attention networks for action recognition and detection. IEEE Transactions on Multimedia (TMM) **22**(11), 2990–3001 (2020). https://doi.org/10.1109/TMM.2020.2965434
20. Li, Y., Guo, Y., Guo, J., Ma, Z., Kong, X., Liu, Q.: Joint crf and locality-consistent dictionary learning for semantic segmentation. IEEE Transactions on Multimedia (TMM) **21**(4), 875–886 (2019). https://doi.org/10.1109/TMM.2018.2867720
21. Lin, J., Gan, C., Han, S.: TSM: temporal shift module for efficient video understanding. Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2019)
22. Lin, T., Liu, X., Li, X., Ding, E., Wen, S.: BMN: boundary-matching network for temporal action proposal generation. Proceedings of IEEE International Conference on Computer Vision (ICCV) (2019)
23. Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. Proceedings of ACM International Conference on Multimedia (ACM MM) (2017)
24. Lin, T., Zhao, X., Shou, Z.: Temporal convolution based action proposal: Submission to ActivityNet 2017. ActivityNet Large Scale Activity Recognition Challenge workshop at CVPR (2017)
25. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: BSN: boundary sensitive network for temporal action proposal generation. Proceedings of European Conference on Computer Vision (ECCV) (2018)
26. Liu, H., Wang, S., Wang, W., Cheng, J.: Multi-scale based context-aware net for action detection. IEEE Transactions on Multimedia (TMM) **22**(2), 337–348 (2020). https://doi.org/10.1109/TMM.2019.2929923
27. Liu, H., Wang, S., Wang, W., Cheng, J.: Multi-scale based context-aware net for action detection. IEEE Transactions on Multimedia (TMM) **22**(2), 337–348 (2020)
28. Liu, K., Gao, L., Khan, N.M., Qi, L., Guan, L.: A multi-stream graph convolutional networks-hidden conditional random field model for skeleton-based action recognition. IEEE Transactions on Multimedia (TMM) **23**, 64–76 (2021). https://doi.org/10.1109/TMM.2020.2974323
29. Liu, Y., Ma, L., Zhang, Y., Liu, W., Chang, S.F.: Multi-granularity generator for temporal action proposal. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
30. Long, F., Yao, T., Qiu, Z., Tian, X., Luo, J., Mei, T.: Gaussian temporal awareness networks for action localization. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
31. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
32. Pardo, A., Caba, F., Alcázar, J.L., Thabet, A.K., Ghanem, B.: Learning to cut by watching movies. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6858–6868 (2021)
33. Pardo, A., Heilbron, F.C., Alcázar, J.L., Thabet, A., Ghanem, B.: Moviecuts: A new dataset and benchmark for cut type recognition. arXiv preprint arXiv:2109.05569 (2021)
34. Qiu, H., Li, H., Wu, Q., Meng, F., Xu, L., Ngan, K.N., Shi, H.: Hierarchical context features embedding for object detection. IEEE Transactions on Multimedia (TMM) **22**(12), 3039–3050 (2020). https://doi.org/10.1109/TMM.2020.2971175

35. Ramazanova, M., Escorcia, V., Heilbron, F.C., Zhao, C., Ghanem, B.: Owl (observe, watch, listen): Localizing actions in egocentric video via audiovisual temporal context. ArXiv **abs/2202.04947** (2022)
36. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6), 1137–1149 (2016)
37. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2015)
38. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.F.: CDC: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
39. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage cnns. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
40. Singh, G., Cuzzolin, F.: Untrimmed video classification for activity detection: submission to ActivityNet Challenge. ActivityNet Large Scale Activity Recognition Challenge workshop at CVPR (2016)
41. Soldan, M., Pardo, A., Alcázar, J.L., Caba, F., Zhao, C., Giancola, S., Ghanem, B.: Mad: A scalable dataset for language grounding in videos from movie audio descriptions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5026–5035 (2022)
42. Su, H., Zhao, X., Lin, T., Liu, S., Hu, Z.: Transferable knowledge-based multi-granularity fusion network for weakly supervised temporal action detection. IEEE Transactions on Multimedia (TMM) **23**, 1503–1515 (2021). https://doi.org/10.1109/TMM.2020.2999184
43. Wang, L., Xiong, Y., Lin, D., Van Gool, L.: Untrimmednets for weakly supervised action recognition and detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
44. Wang, R., Tao, D.: UTS at ActivityNet 2016. ActivityNet Large Scale Activity Recognition Challenge (2016)
45. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. ACM Transactions on Graphics (TOG) **38**(5), 1–12 (2019)
46. Xiong, Y., Wang, L., Wang, Z., Zhang, B., Song, H., Li, W., Lin, D., Qiao, Y., Van Gool, L., Tang, X.: CUHK & ETHZ & SIAT submission to ActivityNet Challenge 2016. arXiv:1608.00797 (2016)
47. Xu, H., Das, A., Saenko, K.: R-C3D: region convolutional 3d network for temporal activity detection. Proceedings of IEEE International Conference on Computer Vision (ICCV) (2017)
48. Xu, M., Zhao, C., Rojas, D.S., Thabet, A., Ghanem, B.: G-TAD: sub-graph localization for temporal action detection. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
49. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. Proceedings of International Conference on Learning Representations (ICLR) (2016)
50. Yuan, Z.H., Stroud, J.C., Lu, T., Deng, J.: Temporal action localization by structured maximal sums. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)

51. Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C.: Graph convolutional networks for temporal action localization. Proceedings of IEEE International Conference on Computer Vision (ICCV) (2019)
52. Zhang, S., Chi, C., Yao, Y., Lei, Z., Li, S.Z.: Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
53. Zhang, S., Peng, H., Yang, L., Fu, J., Luo, J.: Learning sparse 2d temporal adjacent networks for temporal action localization. HACS Temporal Action Localization Challenge at IEEE International Conference on Computer Vision (ICCV) (2019)
54. Zhao, C., Thabet, A.K., Ghanem, B.: Video self-stitching graph network for temporal action localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 13658–13667 (2021)
55. Zhao, H., Yan, Z., Torresani, L., Torralba, A.: HACS: human action clips and segments dataset for recognition and temporal localization. Proceedings of IEEE International Conference on Computer Vision (ICCV) (2019)
56. Zhao, P., Xie, L., Ju, C., Zhang, Y., Wang, Y., Tian, Q.: Bottom-up temporal action localization with mutual regularization. Proceedings of European Conference on Computer Vision (ECCV) (2020)
57. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. Proceedings of IEEE International Conference on Computer Vision (ICCV) (2017)