ACGNet: Action Complement Graph Network for Weakly-supervised Temporal Action Localization

Zichen Yang[1], **Jie Qin**[2], **Di Huang**[1*]

[1] State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China
[2] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
{yangzichen, dhuang}@buaa.edu.cn, qinjiebuaa@gmail.com

Abstract

Weakly-supervised temporal action localization (WTAL) in untrimmed videos has emerged as a practical but challenging task since only video-level labels are available. Existing approaches typically leverage off-the-shelf segment-level features, which suffer from spatial incompleteness and temporal incoherence, thus limiting their performance. In this paper, we tackle this problem from a new perspective by enhancing segment-level representations with a simple yet effective graph convolutional network, namely action complement graph network (ACGNet). It facilitates the current video segment to perceive spatial-temporal dependencies from others that potentially convey complementary clues, implicitly mitigating the negative effects caused by the two issues above. By this means, the segment-level features are more discriminative and robust to spatial-temporal variations, contributing to higher localization accuracies. More importantly, the proposed ACGNet works as a universal module that can be flexibly plugged into different WTAL frameworks, while maintaining the end-to-end training fashion. Extensive experiments are conducted on the THUMOS'14 and ActivityNet1.2 benchmarks, where the state-of-the-art results clearly demonstrate the superiority of the proposed approach.

Introduction

Understanding human actions in videos is an important research direction and has been actively studied in the computer vision community (Wu et al. 2019; Wang et al. 2020; Zolfaghari, Singh, and Brox 2018; Qin et al. 2017; Li et al. 2020; Qi et al. 2020; Liu et al. 2020; Feichtenhofer et al. 2019; Kong et al. 2020; Yang et al. 2021; Ni, Qin, and Huang 2021). The fundamental step is to build meaningful spatial-temporal representations, which involve not only static features from each frame, but also dynamic dependencies across consecutive frames. Among the main tasks in action understanding, temporal action localization (Wu et al. 2020; Lin et al. 2018, 2019) has received tremendous efforts in the past several years, with a wide range of applications (*e.g.*, intelligent surveillance, video retrieval, and human-computer interaction).

To achieve accurate localization results, conventional (fully-supervised) temporal action localization (FTAL) methods (Shou et al. 2017; Lin et al. 2018, 2019; Zhao et al. 2017; Yang et al. 2019) often make use of deep convolutional neural networks (CNNs) trained on video datasets with frame-level annotations. Unfortunately, as the dataset size grows rapidly and the total video length even reaches several decades (Abu-El-Haija et al. 2016), it is obviously unrealistic to acquire such fine-grained annotations. To this end, weakly-supervised temporal action localization (WTAL) (Wang et al. 2017), where only video-level action categories are annotated, has recently emerged as a more practical task. To tackle WTAL, a common practice is to uniformly sample short segments of equal length, for which classifiers are trained (usually through multiple instance learning (Paul, Roy, and Roy-Chowdhury 2018)) with video-level labels, and localization results are generated based on the classification/activation scores of each segment with regard to action categories.

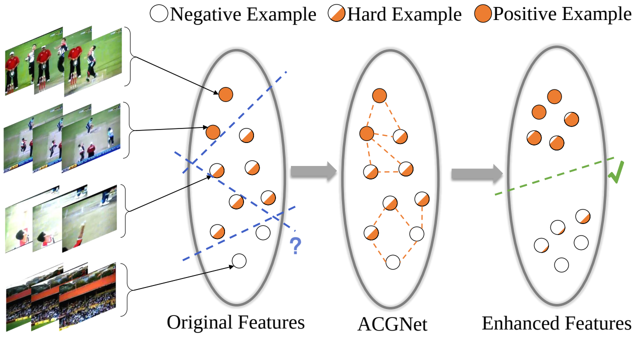

Figure 1: Intuition behind the proposed action complement graph network (ACGNet). By exploiting complementary information across different segments, more discriminative segment-level action representations are learned, leading to more accurate localization results. The blue/green dashed lines indicate the classification hyperplanes.

However, in this paradigm, the evenly sampling strategy incurs two critical issues that greatly limit localization performance. On the one hand, the action segments often suffer from occlusion, blurring, out of field, *etc*., thus lack of certain spatial details. On the other hand, a complete action usually spans a long temporal window and a short action

[*]indicates the corresponding author.

segment is insufficient to observe the full dynamics of that action. We respectively identify the two issues as 'spatial incompleteness' and 'temporal incoherence' of an action segment, both of which make predictions in WTAL unreliable.

In this work, we implicitly address the two issues by a simple yet effective graph convolutional network. The proposed action complement graph network (ACGNet) facilitates an action segment to exploit complementary clues from other segments across the entire untrimmed long video. As shown in Figure 1, after applying our ACGNet, those hard examples can be more easily classified based on the enhanced features. Specifically, we not only consider segment-level similarities but also mitigate negative influences of temporally close segments when constructing the initial action complement graph (ACG). Besides, we make this graph sparse enough to preserve the most informative connections. Through graph convolutions, the complementary information from high-quality segments is propagated to low-quality ones, leading to the enhanced action representation for each segment. In other words, the complementary information provided by other segments is regarded as supervision to learn more discriminative features in the WTAL scenario. Most importantly, owing to the delicately-designed loss function, our ACGNet works as a generic plug-in module and can be flexibly embedded into different WTAL frameworks, further remarkably strengthening the state-of-the-art performance.

In summary, our main contributions are three-fold:

- We propose a novel graph convolutional network for WTAL, namely ACGNet, which greatly enhances the discriminability of segment-level action representations by implicitly exploiting the complementary information and jointly addressing the issues of spatial incompleteness and temporal incoherence.
- We consider multiple vital factors (*i.e.*, segment similarity, temporal diffusion, and graph sparsity) to construct the initial ACG. Moreover, we make the training of our graph network feasible and practical by proposing a novel 'easy positive mining' loss, endowing our ACGNet with the flexibility to be injected into existing frameworks without bells and whistles.
- We equip several recent WTAL methods with the proposed ACGNet. Extensive experiments on two challenging datasets demonstrate its capability to further push the state of the art in WTAL to a large extent.

Related Work

Fully-supervised Temporal Action Localization. Action localization has recently attracted numerous research interests (Zhang et al. 2019; Escorcia et al. 2016; Lin, Zhao, and Shou 2017; Lin et al. 2018; Li et al. 2019). A typical pipeline is to first generate temporal action proposals and then classify pre-defined actions based on the proposals. For example, (Shou et al. 2017) proposes a Convolutional-De-Convolutional filter through temporal upsampling and spatial downsampling to precisely detect segment boundaries. (Zhao et al. 2017) presents the Structured Segment Network to model the temporal structure of each action segment via a structured temporal pyramid. (Yang et al. 2019) provides an end-to-end progressive optimization framework (STEP) for more effective spatial-temporal modeling.

Weakly-supervised Temporal Action Localization. Regarding WTAL, only category labels for whole videos are available, without any fine-grained annotation for each action instance. To tackle this challenge, existing methods usually segment the video at equal temporal intervals, and then classify each segment by multiple instance learning. Specifically, the activation score of a segment to each category, *i.e.*, class activation sequence (CAS), is calculated to classify the action segment. (Wang et al. 2017) formally proposes the tasks of 'weakly supervised action recognition and temporal localization' and used attention weights to exclude the video clips that do not contain actions. (Lee, Uh, and Byun 2020) presents BaS-Net by introducing a background class to assist training, inhibiting the activation of background frames to improve the positioning performance. (Shi et al. 2020) deliver a frame-level probability distribution model (*i.e.*, DGAM) based on frame-level attention to distinguish action frames from background frames. BaM (Lee et al. 2021) is an improved variant of BaS-Net, which employs multiple instance learning to estimate the uncertainty of video frame classification and model the background frames.

Currently, most of the existing studies focus more on developing various learning techniques to improve localization performance based on pre-extracted segment-level features. By contrast, this work emphasizes on the enhancement of the segment-level features by exploring and exploiting the spatial-temporal complementarity across segments, universally benefiting different WTAL frameworks.

Graph-based Temporal Action Localization. Recently, some works investigate graph learning to fuse the information among related categories, multiple proposals or multiple sub-actions to infer the possible actions of a certain segment. For example, P-GCN (Zeng et al. 2019) constructs a graph according to the distances and IoUs between proposals, aiming to adjust the category and boundary of each proposal by using context information. G-TAD (Xu et al. 2020) attempts to make use of not only temporal context, but also semantic context captured through graph convolutional networks (GCN), and then temporal action detection is cast as a sub-graph localization problem. GTRM (Huang, Sugano, and Sato 2020) employs GCN to integrate all the action segments within a certain period of time in the action segmentation task. All such efforts are made in the fully-supervised setting.

In WTAL, (Rashid, Kjellstrm, and Yong 2020) establishes a similarity graph to understand how an action appears as well as the sub-actions that comprise the action's full extent. Notably, this is essentially different from our purpose to complement and enhance features by fully mining the complementary information across segments. Moreover, they design a fixed WTAL network, while our ACGNet works as a universal module to improve various WTAL frameworks. In addition, we propose different graph designs and a novel loss function that enables the joint training of ACGNet and WTAL frameworks.

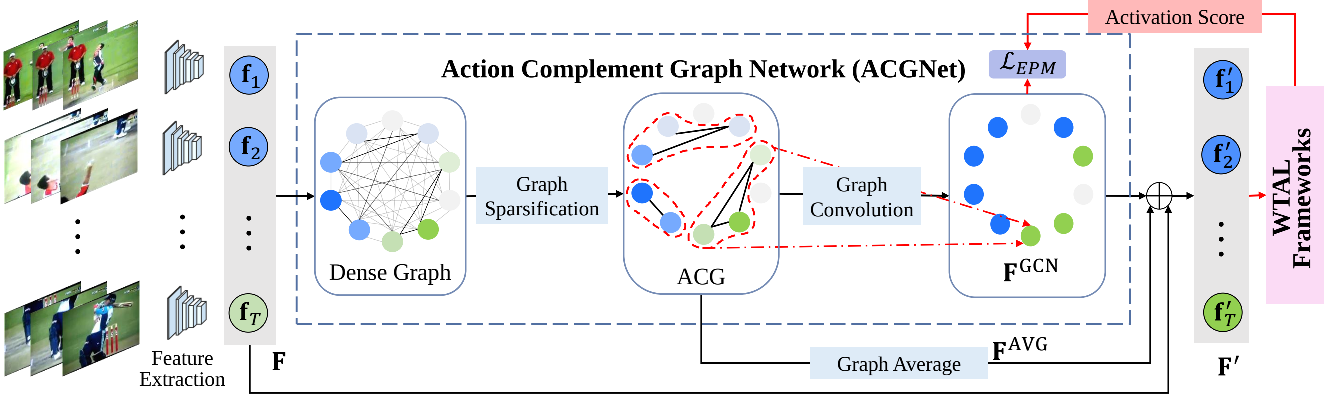

Figure 2: Overall framework of the proposed ACGNet, which takes the segment-level features as input and generates enhanced, more discriminative features by exploiting complementary clues between different segments. More importantly, our ACGNet can be flexibly plugged into various existing WTAL frameworks without bells and whistles.

Action Complement Graph Network

As mentioned above, an input video is uniformly divided into multiple temporal segments, based on which WTAL is performed. The localization accuracy highly depends on the discriminability of segment-level action representations, especially in our weakly-supervised setting. To this end, we aim to enhance segment-level representations, by exploiting the complementary information among different segments. Since our ACGNet is essentially designed for feature enhancement, it can be flexibly plugged into various existing WTAL frameworks, such as (Lee, Uh, and Byun 2020; Lee et al. 2021; Shi et al. 2020) used in our experiments. In the following, we first give a brief introduction of the entire proposed network. Subsequently, we elaborate how to construct the action complement graph (ACG) in a principled way, and how to enhance features based on graph convolution, respectively. Finally, a novel loss is presented to make the training of our graph network feasible. After embedding the ACGNet into existing WTAL frameworks, we follow the standard pipeline provided in (Lee, Uh, and Byun 2020; Lee et al. 2021; Shi et al. 2020) to generate the final localization results.

Method Overview

Figure 2 illustrates the overall framework of the proposed ACGNet. Given an input video V, we first evenly divide it into a fixed number of T short temporal segments $\{S_t\}_{t=1}^{T}$, to handle large variations in video lengths. Then, we extract the features of these segments by using a widely-adopted video feature extraction network, *e.g.*, the I3D network (Carreira and Zisserman 2017). The extracted segment-level features are denoted by the D-dimensional feature vectors $\mathbf{f}_t \in \mathbb{R}^D$, which can be concatenated to form the video-level representation $\mathbf{F} = [\mathbf{f}_1, \ \mathbf{f}_2, \ \cdots, \ \mathbf{f}_T] \in \mathbb{R}^{T \times D}$.

The proposed ACGNet receives the original features $\mathbf{F}$ as input and generates the enhanced features $\mathbf{F}'$ based on a graph convolutional network. The action complement graph (ACG) is constructed for each video in a principled way to exchange complementary information between its nodes (*i.e.*, segments). After constructing the ACG, node-level features are propagated and fused by using graph convolution operations. The output graph features can be regarded an enhanced and complementary counterpart of the original features. Finally, the original and the enhanced features are combined as the ultimate discriminative features $\mathbf{F}'$, which can be used as the input to any WTAL methods to improve their localization performance to a large extent. In addition, a novel loss is proposed to facilitate the joint training of our ACGNet and existing WTAL frameworks.

Action Complement Graph

Due to the lack of frame-level annotations, it is difficult to classify individual short segments. However, multiple segments (among which there usually exist easy-to-classify action instances) in a video can complement each other. Thereby, the ACG is to capture the complementary relationships and enhance the representation for each segment.

Formally, the ACG is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ denotes a set of nodes $\{v_t\}_{t=1}^{T}$, corresponding to T segment-level features $\{\mathbf{f}_t\}_{t=1}^{T}$, while $\mathcal{E}$ refers to the edge set where $e_{ij} = (v_i, v_j)$ is the edge between the nodes v_i and v_j. In addition, we define $\mathbf{A} \in \mathbb{R}^{T \times T}$ as the adjacency matrix associated with the graph $\mathcal{G}$. The weight of an edge, *i.e.*A_{ij}, represents the strength of the relationship between two connected nodes, and a larger weight indicates that two segments are more associated to each other.

In the subsequent, we introduce how to construct ACG by taking multiple factors into consideration at the same time.

Segment Similarity Graph. An untrimmed, long video may contain multiple action instances with large variances due to different scenes, illumination conditions, shooting angles, occlusions, *etc*. However, there are always similar motion patterns among multiple instances of the same action category, where some high-quality or easy-to-classify segments recording more complete action instances with less interference provide relatively stable information and low-quality segments can also be complementary to each other. For instance, two temporal segments belonging to the same action class may be occluded in different regions. In this case, one can facilitate the other to perceive the regions that are visible in its own segment. As a result, it is desirable to propagate various kinds of complementary information across all the segments. To this end, we first construct a segment similarity graph by considering the similarities among segment-level features.

Here, we employ the Cosine distance between two original segment-level features to measure their similarity, and construct the similarity graph $\mathcal{G}^{\mathrm{s}}$ by setting the edge weights (*i.e.*, A^{s}_{ij}) between the i-th and j-th nodes as follows:

$$A^{\mathrm{s}}_{ij} = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{||\mathbf{f}_i||\,||\mathbf{f}_j||}, \tag{1}$$

where $(\cdot)$ is the inner product and $||\cdot||$ is the magnitude.

Temporal Diffusion Graph. Since there exist high temporal dependencies across consecutive segments, we also consider the temporal information when constructing the graph. In nature, temporally close segments usually have a high probability of belonging to the same action and tend to have high similarities, *i.e.*, the corresponding edge weights should be relatively large. Moreover, in practice, the temporal convolution in the feature extraction network (*i.e.*, I3D in our experiments) can fuse the temporal information between adjacent segments in a short temporal window. This leads to even higher feature similarities between temporally close segments (*i.e.*, A^{s}_{ij} tends to be large when $i \rightarrow j$). Therefore, if we construct the temporal graph based on the above facts and add it directly to the segment similarity graph, the propagation of the complementary information is likely to be restricted in a short temporal window and cannot be successfully shared between segments that are far apart. For example, the i-th segment S_i containing a high-quality discriminative action instance cannot complement the other inferior instances (belonging to the same action) which are temporally located far away from S_i.

Therefore, we attempt to spread out the complementary information as far as possible so that the discriminability of more segments can be enhanced in the untrimmed, long video, leading to improved localization performance. To this end, we construct a temporal diffusion graph by imposing larger edge weights between farther nodes. Specifically, we construct the temporal diffusion graph $\mathcal{G}^{\mathrm{t}}$ as follows:

$$A^{\mathrm{t}}_{ij} = 1 - \frac{\max(Z - |i-j|, 0)}{Z}, \tag{2}$$

where Z is a hyper-parameter to control the diffusion degree.

Overall Sparse Graph. By simply combining the two sub-graphs $\mathcal{G}^{\mathrm{s}}$ and $\mathcal{G}^{\mathrm{t}}$, we can obtain our final action complement graph $\mathcal{G}$, of which the adjacency matrix is defined as follows:

$$\mathbf{A} = \frac{\mathbf{A}^{\mathrm{s}} + \alpha \mathbf{A}^{\mathrm{t}}}{2}, \tag{3}$$

where the two matrices $\mathbf{A}^{\mathrm{s}}$ and $\mathbf{A}^{\mathrm{t}}$ include A^{s}_{ij} and A^{t}_{ij} as their (i, j)-th entries, respectively, and α is the hyper-parameter for a better trade-off between the two sub-graphs.

Due to that the edge weights of the two sub-graphs are mostly above zero, simply combining them to form the ACG results in a very dense graph. If we directly learn the enhanced features based on this dense graph, we may obtain similar global video-level features for each node/segment since each node is expected to perceive the features of all the rest nodes. This implicitly hinders the discriminability of segment-level features, leading to less accurate localization results. Therefore, it is necessary to make the graph sparse enough to only preserve those most informative nodes. In particular, we set our sparsification criterion based on both a threshold λ and a top-K ranking list. The final sparse ACG is constructed as:

$$A'_{ij} = \begin{cases} \mathrm{sgn}(A_{ij} - \lambda) \cdot A_{ij}, & \mathrm{rank}_i(j) \leq K \\ 0, & \mathrm{rank}_i(j) > K \end{cases} \tag{4}$$

where $\mathrm{sgn}(\cdot)$ is an indicator, *i.e.*, $\mathrm{sgn}(x) = 1$ if $x > 0$; otherwise $\mathrm{sgn}(x) = 0$. $\mathrm{rank}_i(j)$ is the ranking number of the j-th node w.r.t. the edge weights among all the adjacent nodes of the i-th node in the dense graph w.r.t. $\mathbf{A}$. Note that we adopt these two criteria regarding λ and K to make the graph sparse, because simply adopting the threshold cannot discard those ambiguous segments in similar scenes but belonging to different action classes. This intuition is also supported by the ablation study in our experiments.

Graph Inference

Graph Average. After constructing the final sparse ACG, a straightforward way to aggregating all the node-level features is to compute the average features by considering the edge weights as follows:

$$\mathbf{f}^{\mathrm{AVG}}_i = \sum_{j=1}^{T} \hat{A}_{ij} \mathbf{f}_j, \tag{5}$$

where $\hat{A}_{ij}$ is the (i, j)-th entry of the matrix $\hat{\mathbf{A}}$, which is the row-wise normalized adjacency matrix w.r.t. $\mathbf{A}'$. In practice, we find the averaged feature $\mathbf{f}^{\mathrm{AVG}}_i$ can exchange complementary information to some extent, achieving satisfactory performance as shown in the subsequent experiments.

Graph Convolution. In addition to the above average features, we incorporate graph convolutions into our ACGNet to better aggregate node-level features. For a graph convolutional network (GCN) with M layers, the graph convolution operation w.r.t. the m-th $(1 \leq m \leq M)$ layer is as follows:

$$\mathbf{F}^{(m)} = \sigma(\hat{\mathbf{A}} \mathbf{F}^{(m-1)} \mathbf{W}^{(m)}), \tag{6}$$

where $\mathbf{F}^{(m)}$ is the feature generated by the m-th graph convolutional layer, $\mathbf{F}^{(0)} = \mathbf{F}$ is the original feature, $\mathbf{F}^{\mathrm{GCN}} = \mathbf{F}^{(M)}$ is the final output of the last graph convolutional layer, $\mathbf{W}^{(m)} \in \mathbb{R}^{D \times D}$ is the trainable parameters of the m-th layer, and $\sigma(\cdot)$ is the ReLU (Nair and Hinton 2010) activation function.

Finally, the original features are combined with the graph averaged features and the output features of the GCN to obtain the enhanced discriminative features:

$$\mathbf{F}' = \mathbf{F} + \mathbf{F}^{\mathrm{AVG}} + \mathbf{F}^{\mathrm{GCN}}. \tag{7}$$

Since $\mathbf{F}'$ is the enhanced counterpart of the original feature, different WTAL methods can replace their original input by $\mathbf{F}'$, further performing the subsequent localization task.

Training Objective

To discover the easy-to-classify segments to enhance the features of other similar ones, making more segments easier to be classified, we propose a novel loss based on an 'easy positive mining' (EPM) strategy for sufficiently training the joint WTAL network with our ACGNet embedded:

$$\mathcal{L}_{\text{EPM}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{i,j=1}^{T}(p_{n,j}||\mathbf{f}'_{n,i} - \mathbf{f}_{n,j}||^2),\ \text{s.t.}\ A'_{n,ij} > 0, \tag{8}$$

where $\mathbf{f}'_{n,i}$ is the output feature of the ACGNet w.r.t. the i-th segment in the n-th video, and $\mathbf{f}_{n,j}$ and $p_{n,j}$ are the original feature and the maximum activation score among all the classes in terms of the j-th segment in the same video, respectively.

Based on Eq. (8), the output features of the ACGNet are encouraged to be consistent with the original features of similar segments, especially those 'easy positive' examples that can be successfully classified with the highest confidence scores. In other words, the 'easy positive' segments can be regarded as the class centroids in the feature space, and we aim to push other similar segments closer to them. Consequently, more action segments become easier to distinguish, finally achieving more accurate location results.

Experiments

Experimental Setup

Datasets. THUMOS'14 (Idrees et al. 2017) contains over 20 hours of videos from 20 sports classes. The dataset is quite challenging due to the diversified video lengths and the large number of action instances ($\sim$15) per video. The training set includes only trimmed videos that are not suitable for action localization, but the validation and test sets provide 200 and 213 untrimmed videos, respectively. Following (Lee, Uh, and Byun 2020; Shi et al. 2020; Lee et al. 2021), we conduct training on the validation set and perform evaluation on the test set. **ActivityNet1.2** (Caba Heilbron et al. 2015) consists of 100 categories of actions. The training, validation, and test sets are composed of 4,819, 2,383, and 2,480 videos, respectively. However, there are no publicly-available action labels in the test set as it is only used for competitions. Therefore, we follow the general practice in (Lee, Uh, and Byun 2020; Shi et al. 2020; Lee et al. 2021) by employing the training set for training and the validation set for testing.

Baselines. The proposed ACGNet works as a universal module that can be incorporated into different WTAL frameworks. The integration into other frameworks is rather straightforward, and we only need to replace the original features by the enhanced ones obtained by the ACGNet. In our experiments, we adopt three recently proposed WTAL methods, including BaS-Net (Lee, Uh, and Byun 2020), DGAM (Shi et al. 2020), and BaM (Lee et al. 2021).

Due to its flexibility, our ACGNet may also be incorporated into other action detection (*e.g.*, FTAL or the more general spatial-temporal action detection) frameworks. However, it should be noted that ACGNet is to enhance segment-level representations, which is especially crucial in the context of WTAL as only video-level annotations are available. In FTAL or other related tasks, segment-level features can be optimized more easily based on the provided fine-grained (frame-level) annotations.

Evaluation Metrics. We adopt the standard metrics for performance evaluation of different methods, *i.e.*, mean Average Precisions (mAPs) under different Intersection of Union (IoU) thresholds. In practice, we adopt the official evaluation code provided by ActivityNet.

Implementation Details. The proposed framework is implemented using the PyTorch library. Our ACGNet and the subsequent action localization network are jointly trained in an end-to-end manner. The action localization networks retain the parameter settings in their original papers, and we apply the stochastic gradient descent (SGD) to simultaneously optimize the joint network on an NVIDIA Tesla V100 GPU. For fair comparison with other WTAL methods, we exploit I3D (Carreira and Zisserman 2017) to extract the initial segment-level features. The hyper-parameters adopted to construct the ACG are empirically set as follows: $Z = 10$, $\alpha = 1$, and $\lambda = 0.85$. When taking BaS-Net as the action localization network, we set K to 50 and $T = 750$ is a fixed value. We set $T = 400$ (consistent with the original paper) and $K = T/10 = 40$ when employing the other two localization frameworks. We utilize a 2-layer graph convolutional network in all the experiments.

Comparison to State-of-the-Art Methods

Results on THUMOS'14. Table 1 shows the localization performance of different methods on THUMOS'14. For fair comparison, we also report the results of the three adopted WTAL frameworks based on our implementations. From the table, we can see that after integrating the proposed ACGNet, the results of the three localization networks are significantly and consistently improved in terms of most IoU thresholds. Notably, when the IoU threshold is set to 0.5, BaS-Net, DGAM, and BaM respectively gain absolute improvements of 3.0%, 1.8%, and 1.3% in mAP. The gain on BaM is not so remarkable, probably due to that BaM greatly improves the discriminability of segment features through background modeling. Such facts indicate the effectiveness of exploiting complementary clues between temporal segments in the weakly-supervised setting. In all, we push the state of the art in WTAL to a large extent, which is even on par with the performance of some fully-supervised approaches.

Results on ActivityNet1.2. Table 2 shows the comparison results on ActivityNet1.2. Similar to the observations on THUMOS'14, our ACGNet greatly strengthens the existing WTAL frameworks with regard to all the IoU thresholds, and the improvement on BaS-Net is particularly encouraging. Specifically, when adopting 0.5 as the IoU threshold, the mAPs of BaS-Net, DGAM, and BaM are improved by 3.9%, 1.1%, and 1.0%, respectively. This again demonstrates the superiority of the proposed feature enhancement network.

Table 1: Comparison results on THUMOS'14. * indicates the results based on our implementations.

Methods	mAP (%) @ IoU							
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	Average
TAL-Net (Chao et al. 2018)	59.8	57.1	53.2	48.5	42.8	33.8	20.8	45.1
BSN (Lin et al. 2018)	-	-	53.5	45.0	36.9	28.4	20.0	-
GTAN (Long et al. 2019)	69.1	63.7	57.8	47.2	38.8	-	-	-
BMN (Lin et al. 2019)	-	-	56.0	47.4	38.8	29.7	20.5	-
P-GCN (Zeng et al. 2019)	69.5	67.8	63.6	57.8	49.1	-	-	-
STPN (Nguyen et al. 2018)	52.0	44.7	35.5	25.8	16.9	9.9	4.3	27.0
W-TALC (Paul, Roy, and Roy-Chowdhury 2018)	55.2	49.6	40.1	31.1	22.8	-	7.6	-
MAAN (Yuan et al. 2019)	59.8	50.8	41.1	30.6	20.3	12.0	6.9	31.6
Liu *et al*. (Liu, Jiang, and Wang 2019)	57.4	50.8	41.2	32.1	23.1	15.0	7.0	32.4
TSM (Yu et al. 2019)	-	-	39.5	-	24.5	-	7.1	-
Nguyen *et al*. (Nguyen, Ramanan, and Fowlkes 2019)	60.4	56.0	46.6	37.5	26.8	17.6	9.0	36.3
RPN (Huang et al. 2020)	62.3	57.0	48.2	37.2	27.9	16.7	8.1	36.8
Gong *et al*. (Gong et al. 2020)	-	-	46.9	38.9	30.1	19.8	10.4	-
ActionBytes (Jain, Ghodrati, and Snoek 2020)	-	-	43.0	35.8	29.0	-	9.5	-
EM-MIL (Luo et al. 2020)	59.1	52.7	45.5	36.8	30.5	22.7	16.4	37.7
A2CL-PT (Min and Corso 2020)	61.2	56.1	48.1	39.0	30.1	19.2	10.6	37.8
TSCN (Zhai et al. 2020)	63.4	57.6	47.8	37.7	28.7	19.4	10.2	37.8
BaS-Net (Lee, Uh, and Byun 2020)	58.2	52.3	44.6	36.0	27.0	18.6	10.4	35.3
BaS-Net* (Lee, Uh, and Byun 2020)	57.5	51.6	44.3	35.5	26.8	18.6	10.2	34.9
+ACGNet	**58.8**$^{+1.3}$	**53.3**$^{+1.7}$	**46.4**$^{+2.1}$	**38.3**$^{+2.8}$	**29.8**$^{+3.0}$	**20.9**$^{+2.3}$	**11.2**$^{+1.0}$	**37.0**$^{+2.1}$
DGAM (Shi et al. 2020)	60.0	54.2	46.8	38.2	28.8	19.8	11.4	37.0
DGAM* (Shi et al. 2020)	59.4	53.4	46.1	37.0	27.8	19.5	11.0	36.3
+ACGNet	**62.5**$^{+3.1}$	**55.9**$^{+2.5}$	**48.2**$^{+2.1}$	**39.5**$^{+2.5}$	**29.6**$^{+1.8}$	**20.4**$^{+0.9}$	**11.2**$^{+0.2}$	**38.2**$^{+1.9}$
BaM (Lee et al. 2021)	67.5	61.2	52.3	43.4	33.7	22.9	12.1	41.9
BaM* (Lee et al. 2021)	66.6	59.8	51.3	43.0	33.4	22.4	12.1	41.2
+ACGNet	**68.1**$^{+1.5}$	**62.6**$^{+2.8}$	**53.1**$^{+1.8}$	**44.6**$^{+1.6}$	**34.7**$^{+1.3}$	**22.6**$^{+0.2}$	**12.0**$^{-0.1}$	**42.5**$^{+1.3}$

Table 2: Comparison results on ActivityNet1.2. * indicates the results based on our implementations.

Methods	mAP (%) @ IoU			
	0.5	0.75	0.95	Average
Wang *et al*. (Wang et al. 2017)	7.4	3.2	0.7	3.6
AutoLoc (Shou et al. 2018)	27.3	15.1	3.3	16.0
W-TALC (Paul, Roy, and Roy-Chowdhury 2018)	37.0	12.7	1.5	18.0
CleanNet (Liu et al. 2019)	37.1	20.3	5.0	21.6
Liu *et al*. (Liu, Jiang, and Wang 2019)	36.8	22.9	5.6	22.4
TSM (Yu et al. 2019)	28.3	17.0	3.5	17.1
RPN (Huang et al. 2020)	37.6	23.9	5.4	23.3
Gong *et al*. (Gong et al. 2020)	40.0	25.0	4.6	24.6
EM-MIL (Luo et al. 2020)	37.4	-	-	20.3
TSCN (Zhai et al. 2020)	37.6	23.7	5.7	23.6
BaS-Net (Lee, Uh, and Byun 2020)	38.5	24.2	5.6	24.3
BaS-Net* (Lee, Uh, and Byun 2020)	36.9	23.3	5.1	22.4
+ACGNet	**40.8**$^{+3.9}$	**25.3**$^{+2.0}$	**5.6**$^{+0.5}$	**25.1**$^{+2.7}$
DGAM (Shi et al. 2020)	41.0	23.5	5.3	24.4
DGAM* (Shi et al. 2020)	40.3	23.2	5.0	24.0
+ACGNet	**41.4**$^{+1.1}$	**24.2**$^{+1.0}$	**5.5**$^{+0.5}$	**24.9**$^{+0.9}$
BaM (Lee et al. 2021)	41.2	25.6	6.0	25.9
BaM* (Lee et al. 2021)	40.8	24.9	5.8	25.6
+ACGNet	**41.8**$^{+1.0}$	**26.0**$^{+1.1}$	**5.9**$^{+0.1}$	**26.1**$^{+0.5}$

Ablation Study

We perform ablation study on BaS-Net as it is the most flexible and efficient among the three baselines. It is worth noting that the number of parameters increases from 26.3 M to 34.6 M when plugging ACGNet into BaS-Net. This complexity is expected as ACGNet includes several processing steps and is not fully optimized. However, considering the flexibility of such a universal module and the consistent performance gains, the increase in complexity is acceptable.

Effects of Graph Design. We first study the (dis)advantages of different graph designs. Table 3 shows the results with various degrees of sparsity (K). $\mathcal{G}^1$ indicates directly using the segment similarity graph $\mathcal{G}^{\text{s}}$; $\mathcal{G}^2$ is a variant of our ACG by subtracting the temporal diffusion graph $\mathcal{G}^{\text{t}}$ from $\mathcal{G}^{\text{s}}$; $\mathcal{G}^3$ is the proposed ACG. Specifically, when K=50, $\mathcal{G}^3$ achieves the highest mAP among the three. However, when the graphs become denser, the results tend to decrease gradually. This aligns well with our previous assumption that dense graphs cannot exploit the most discriminative features across all the segments. Finally, we test the performance by using only $\mathcal{G}^t$. In this case, most edges in $\mathcal{G}^t$ are weighted by one and no meaningful feature enhancement can be guaranteed. Consequently, we only obtain an inferior mAP of 22.1% when K=50.

Table 3: Results of different graph designs on THUMOS'14. * indicates the dense graph without sparsification.

K	mAP (%) @ IoU=0.5		
	$\mathcal{G}^1 = \mathcal{G}^{\text{s}}$	$\mathcal{G}^2 = \mathcal{G}^{\text{s}} - \mathcal{G}^{\text{t}}$	$\mathcal{G}^3 = \mathcal{G}^{\text{s}} + \mathcal{G}^{\text{t}}$
1	19.3	19.7	22.2
5	21.8	21.6	24.3
20	26.8	25.9	28.1
50	**28.0**	**27.2**	**29.8**
200	27.5	26.6	28.6
750*	25.9	25.1	25.3

Effects of Sparsity. As discussed above, Table 3 includes some results when adopting different sparsity levels w.r.t. K. Here, we additionally evaluate how the threshold λ affects the final performance. As shown in Figure 4, the best results are always achieved by considering both factors (*i.e.*, λ and K) for graph sparsification. This indicates that simply using a threshold is not enough to maintain the most discriminative nodes. This may be because the scenes remain unchanged in some videos, *i.e.*, the similarity between different segments is always high even if the segments contain different kinds of action instances. In such cases, simply adopting a threshold preserves those irrelevant nodes belonging to distinct categories. By further imposing the top-K constraint, we can remove the ambiguous nodes and keep the most relevant ones,

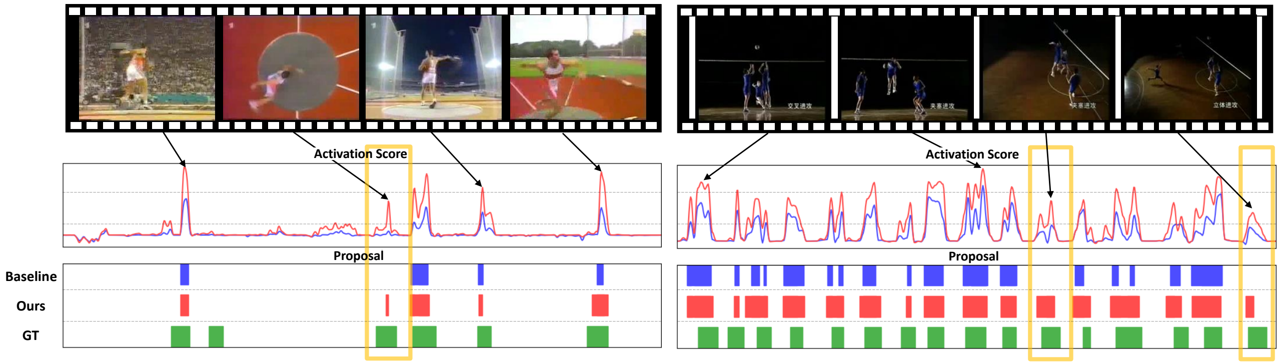

Figure 3: Qualitative visualization of two typical video examples on THUMOS'14. The results of BaS-Net (Baseline), BaS-Net+ACGNet (Ours), and ground truth (GT) are shown in blue, red, and green, respectively. The yellow boxes include some difficult cases that Bas-Net fails to detect but can be successfully localized by our method.

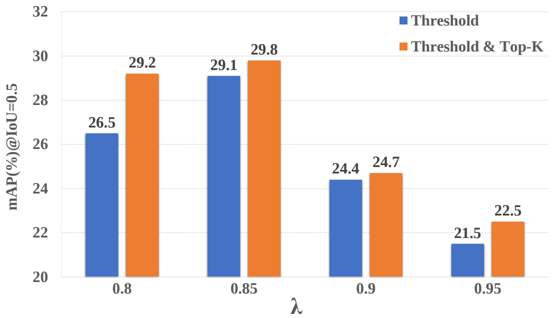

Figure 4: Comparison results of ACGNet (with BaS-Net) by using different levels of sparsity w.r.t. λ on THUMOS'14.

obtaining more discriminative segment-level features.

Component Validation. Table 4 shows the results based on different components in our ACGNet. Concretely, we test the performance when adopting different features and distinct ways of feature combinations.We can see that combining the original features with either the weighted average or the graph convolutional ones can significantly improve the overall accuracy. By fusing all the features, we can achieve the best performance. It is also noteworthy that inferior performance is observed if the EPM loss is discarded during graph training. As mentioned previously, this is because the graph convolutional layers cannot be trained sufficiently. Interestingly, when only using the enhanced graph-based features, the accuracy drops a lot, indicating that taking them as the supplements to the original features shows an effective way to make the potential of the graph features unleashed. In addition, only using $\mathbf{F}^{\text{AVG}}$ performs the worst, as the average features along the whole video cannot represent the distinct temporal dynamics of different action instances.

Qualitative Analysis

Figure 3 visualizes some qualitative results. The curves represent the detection activation scores, while the blocks denote the localization results with the IoU threshold at 0.5. It can be observed that most of our scores are higher than the ones delivered by Bas-Net, indicating that our enhanced features are more discriminative for classification. Meanwhile, the scores of other non-action segments remain relatively low, revealing that our method can successfully distinguish action-related segments from irrelevant background. We also note that our detected proposals are more complete, while Bas-Net tends to split one proposal into several individual shorter proposals, leading to degraded accuracy. The difficult cases in the yellow boxes further demonstrate the superiority of our ACGNet.

Table 4: Results of different components of ACGNet (with BaS-Net) on THUMOS'14.

Feature			$\mathcal{L}_{\text{EPM}}$	Fusion		mAP (%)
I3D	AVG	GCN		sum	concat	IoU=0.5
√						26.8
	√					19.7
√	√			√		28.5
√	√				√	29.1
√		√		√		26.2
√		√			√	27.3
		√	√			22.4
	√	√	√	√		22.2
√		√	√	√		29.0
√		√	√		√	29.1
√	√	√		√		28.2
√	√	√	√	√		**29.8**
√	√	√	√		√	28.7

Conclusion

This paper presents the ACGNet aiming to enhance the discriminability of segment-level representations of videos for WTAL. The complementary clues from other segments in the same video, particularly the easy-to-classify ones, provides certain supervision to learn more discriminative features. Our ACGNet works as a general module that is flexibly embedded into various existing WTAL frameworks, remarkably boosting the state of the art performance on two challenging benchmarks.

Acknowledgment

This work is partly supported by the National Natural Science Foundation of China (No. 62022011), the Research Program of State Key Laboratory of Software Development Environment (SKLSDE-2021ZX-04), and the Fundamental Research Funds for the Central Universities.

References

Abu-El-Haija, S.; Kothari, N.; Lee, J.; Natsev, P.; Toderici, G.; Varadarajan, B.; and Vijayanarasimhan, S. 2016. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*.

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

Carreira, J.; and Zisserman, A. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*.

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

Escorcia, V.; Heilbron, F. C.; Niebles, J. C.; and Ghanem, B. 2016. DAPs: Deep Action Proposals for Action Understanding. In *ECCV*.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. SlowFast Networks for Video Recognition. In *ICCV*.

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

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

Huang, Y.; Sugano, Y.; and Sato, Y. 2020. Improving Action Segmentation via Graph-Based Temporal Reasoning. In *CVPR*.

Idrees, H.; Zamir, A. R.; Jiang, Y.-G.; Gorban, A.; Laptev, I.; Sukthankar, R.; and Shah, M. 2017. The THUMOS challenge on action recognition for videos “in the wild”. *CVIU*, 155: 1–23.

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

Kong, L.; Huang, D.; Qin, J.; and Wang, Y. 2020. A Joint Framework for Athlete Tracking and Action Recognition in Sports Videos. *IEEE TCSVT*, 30(2): 532–548.

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

Lee, P.; Wang, J.; Lu, Y.; and Byun, H. 2021. Background Modeling via Uncertainty Estimation for Weakly-supervised Action Localization. In *AAAI*.

Li, D.; Yao, T.; Qiu, Z.; Li, H.; and Mei, T. 2019. Long Short-Term Relation Networks for Video Action Detection. In *ACM MM*.

Li, Y.; Ji, B.; Shi, X.; Zhang, J.; Kang, B.; and Wang, L. 2020. TEA: Temporal Excitation and Aggregation for Action Recognition. In *CVPR*.

Lin, T.; Liu, X.; Li, X.; Ding, E.; and Wen, S. 2019. BMN: Boundary-matching network for temporal action proposal generation. In *ICCV*.

Lin, T.; Zhao, X.; and Shou, Z. 2017. Single Shot Temporal Action Detection. In *ACM MM*.

Lin, T.; Zhao, X.; Su, H.; Wang, C.; and Yang, M. 2018. BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. In *ECCV*.

Liu, D.; Jiang, T.; and Wang, Y. 2019. Completeness modeling and context separation for weakly supervised temporal action localization. In *CVPR*.

Liu, Z.; Luo, D.; Wang, Y.; Wang, L.; and Lu, T. 2020. TEINet: Towards an Efficient Architecture for Video Recognition. In *AAAI*.

Liu, Z.; Wang, L.; Zhang, Q.; Gao, Z.; Niu, Z.; Zheng, N.; and Hua, G. 2019. Weakly supervised temporal action localization through contrast based evaluation networks. In *ICCV*.

Long, F.; Yao, T.; Qiu, Z.; Tian, X.; Luo, J.; and Mei, T. 2019. Gaussian temporal awareness networks for action localization. In *CVPR*.

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

Min, K.; and Corso, J. J. 2020. Adversarial Background-Aware Loss for Weakly-Supervised Temporal Activity Localization. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *ECCV*.

Nair, V.; and Hinton, G. E. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair. In *ICML*.

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

Nguyen, P. X.; Ramanan, D.; and Fowlkes, C. C. 2019. Weakly-supervised action localization with background modeling. In *ICCV*.

Ni, J.; Qin, J.; and Huang, D. 2021. Identity-Aware Graph Memory Network for Action Detection. In *ACM MM*.

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

Qi, M.; Wang, Y.; Qin, J.; Li, A.; Luo, J.; and Van Gool, L. 2020. stagNet: An Attentive Semantic RNN for Group Activity and Individual Action Recognition. *IEEE TCSVT*, 30(2): 549–565.

Qin, J.; Liu, L.; Shao, L.; Shen, F.; Ni, B.; Chen, J.; and Wang, Y. 2017. Zero-Shot Action Recognition with Error-Correcting Output Codes. In *CVPR*.

Rashid, M.; Kjellstrm, H.; and Yong, J. L. 2020. Action Graphs: Weakly-supervised Action Localization with Graph Convolution Networks. In *WACV*.

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

Shou, Z.; Chan, J.; Zareian, A.; Miyazawa, K.; and Chang, S.-F. 2017. CDC: Convolutional-De-Convolutional Networks for Precise Temporal Action Localization in Untrimmed Videos. In *CVPR*.

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

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

Wang, Z.; Gao, Z.; Wang, L.; Li, Z.; and Wu, G. 2020. Boundary-Aware Cascade Networks for Temporal Action Segmentation. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *ECCV*.

Wu, C.-Y.; Feichtenhofer, C.; Fan, H.; He, K.; Krahenbuhl, P.; and Girshick, R. 2019. Long-term feature banks for detailed video understanding. In *CVPR*.

Wu, J.; Kuang, Z.; Wang, L.; Zhang, W.; and Wu, G. 2020. Context-Aware RCNN: A Baseline for Action Detection in Videos. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *ECCV*.

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

Yang, X.; Yang, X.; Liu, M.-Y.; Xiao, F.; Davis, L. S.; and Kautz, J. 2019. STEP: Spatio-Temporal Progressive Learning for Video Action Detection. In *CVPR*.

Yang, Z.; Huang, D.; Qin, J.; and Wang, Y. 2021. Human-Aware Coarse-to-Fine Online Action Detection. In *ICASSP*.

Yu, T.; Ren, Z.; Li, Y.; Yan, E.; Xu, N.; and Yuan, J. 2019. Temporal structure mining for weakly supervised action detection. In *ICCV*.

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

Zeng, R.; Huang, W.; Tan, M.; Rong, Y.; Zhao, P.; Huang, J.; and Gan, C. 2019. Graph convolutional networks for temporal action localization. In *ICCV*.

Zhai, Y.; Wang, L.; Tang, W.; Zhang, Q.; Yuan, J.; and Hua, G. 2020. Two-Stream Consensus Network for Weakly-Supervised Temporal Action Localization. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *ECCV*.

Zhang, C.; Xu, Y.; Cheng, Z.; Niu, Y.; Pu, S.; Wu, F.; and Zou, F. 2019. Adversarial Seeded Sequence Growing for Weakly-Supervised Temporal Action Localization. In *ACM MM*.

Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D. 2017. Temporal Action Detection With Structured Segment Networks. In *ICCV*.

Zolfaghari, M.; Singh, K.; and Brox, T. 2018. ECO: Efficient convolutional network for online video understanding. In *ECCV*.