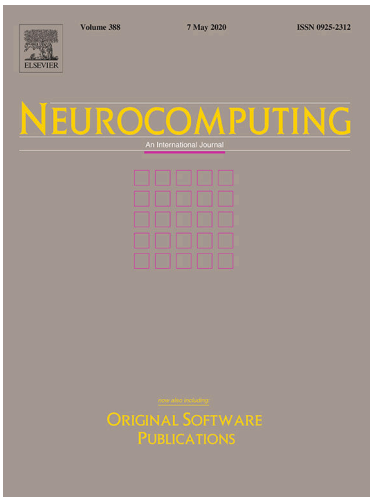
Journal Pre-proofs



Graph-based Temporal Action Co-Localization from an Untrimmed Video

Le Wang, Changbo Zhai, Qilin Zhang, Wei Tang, Nanning Zheng, Gang Hua

PII: S0925-2312(20)32044-0
DOI: https://doi.org/10.1016/j.neucom.2020.12.126
Reference: NEUCOM 23266

To appear in: *Neurocomputing*

Received Date: 18 August 2020
Accepted Date: 28 December 2020

Please cite this article as: L. Wang, C. Zhai, Q. Zhang, W. Tang, N. Zheng, G. Hua, Graph-based Temporal Action Co-Localization from an Untrimmed Video, *Neurocomputing* (2021), doi: https://doi.org/10.1016/j.neucom.2020.12.126

Graph-based Temporal Action Co-Localization from an Untrimmed Video

Le Wang[a,*], Changbo Zhai[a], Qilin Zhang[b], Wei Tang[c], Nanning Zheng[a], Gang Hua[d]

[a]*Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P.R. China*
[b]*ABB Corporate Research Center Raleigh, NC 27606, USA*
[c]*Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607, USA*
[d]*Wormpex AI Research, Bellevue, WA 98004, USA*

Abstract

We present an efficient approach for temporal action co-localization (TACL), which means to simultaneously localize all action instances in an untrimmed video. Compared with the conventional instance-by-instance action localization, TACL can exploit the contextual and temporal relationships among action instances to reduce the localization ambiguities. Motivated by the strong relational modeling capability of graph neural networks, we propose a Graph-based Temporal Action Co-Localization (G-TACL) method. By considering each action proposal as a node, G-TACL effectively aggregates contextual and temporal features from related action proposals to jointly recognize and localize all action instances in a single shot. Moreover, we introduce a novel multi-level consistency evaluator to measure the relatedness between any two action proposals. This is achieved by considering their high-level contextual similarities, low-level temporal coincidences and feature correlations. We exploit the Gated Recurrent Units (GRUs) to iteratively update the features of each node, which are then used to regress the temporal boundaries of action proposals and finally achieve action co-localization. Experimental results on three datasets, *i.e.*, THUMOS14, MEXaction2 and ActivityNet v1.3 datasets demonstrate that our G-TACL is superior or comparable to the state-of-the-arts.

Keywords: Temporal action co-localization, multi-level consistency evaluator

*Corresponding author: lewang@xjtu.edu.cn

1. Introduction

Temporal action co-localization (TACL) aims at simultaneously localizing all action instances in an untrimmed video. This includes simultaneous action recognition (identify the category of each action instance) and temporal action localization (identify the temporal boundaries of each action instance). TACL can be used in a variety of computer vision tasks such as video content understanding, intelligent video surveillance and action analysis. Since different action instances contained in the same video have similar appearance and contextual features, the advantage of TACL over the instance-by-instance temporal action localization (TAL) is that the features of multiple action instances of a common action category can be leveraged to facilitate the localization of each action instance.

Considerable progress has been made to address the TAL problem in untrimmed videos [1–10]. Techniques including hand-crafted features [1, 6], convolution neural networks (CNNs) [4, 7, 8] and 3-dimensional convolution networks (3D ConvNets) [5, 10] have been proposed and empirically demonstrated promising performance. Some methods [1, 7] threshold snippet-level classification predictions to produce the TAL predictions. Moreover, a few works [4, 11] try to exploit contextual features of each independent action instance to improve TAL.

Despite the success of existing methods, there are still potential problems that hinder their practical applications. On one hand, many approaches [12–14] perform action recognition only on trimmed videos, where each video contains only one action instance and there is no interference from other potentially confusing actions or backgrounds. On the other hand, an untrimmed video contains multiple action instances which belong to the same action category in general. But the correlation among those action instances is usually ignored, which could otherwise be beneficial due to their appearance and contextual consistency. The benefit of exploiting the appearance and structural consistency among instances has been demonstrated in image/video object co-segmentation [15–20]. Actually, it is common that a video contains multiple action instances of the same category, such as triple jump videos from the Olympic Games. However, the common action instances in a video usually exhibit dramatic variations in human postures, interacting objects and viewpoints. Therefore, it is desirable to develop a temporal action co-localization model that can not only exploit the correlation among action instances but also account for their variations.

Graph neural networks (GNNs), which inherit the advantages of both CNNs and graphical models, have a strong capability of representing and learning the correlation among entities [21]. They have been widely applied to many tasks and achieved good performance, such as human-object interaction detection [21], relational reasoning [22] and action recognition [23]. Therefore, it is attractive to

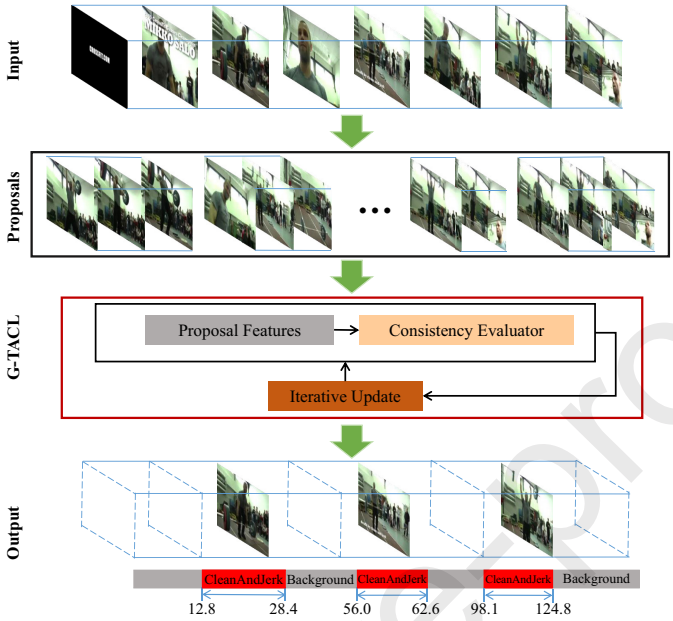


Figure 1: A flowchart of the proposed G-TACL method. The input is an untrimmed video, which contains multiple action instances of a common category (*e.g.*, CleanAndJerk, marked with the red chunks at the bottom), and a large number of background frames (marked with the gray chunks) not containing such action instances. We first generate high-quality action proposals, and then feed them to the G-TACL network. The output comprises the predicted action category and the temporal boundaries of action instances.

use GNNs to capture the correlations among multiple action instances for action co-localization.

Inspired by the success of GNNs, we propose the Graph-based Temporal Action Co-Localization (G-TACL) algorithm. It exploits a GNN to model the correlations among action proposals of the same category to co-localize action instances. In this way, the contextual and appearance features shared by multiple action instances/proposals of the common category can be used to improve the localization accuracy. Figure 1 illustrates an overview of our proposed G-TACL method. The input is an untrimmed video containing multiple action instances of a common category as well as a large portion of background frames not belonging to this category. The output is the predicted action category along with the temporal boundaries of each action instance.

We first divide the input video into several equal-length snippets and employ the two-stream Inflated 3D ConvNet (I3D) [24] to extract snippet-level features. Then, a binary classifier is applied to compute the *actionness* score of each video

snippet, which indicates whether it belongs to the action category or not. To generate high-quality action proposals, a two-step thresholding strategy is utilized to group video snippets according to their actionness scores. Finally, we leverage the G-TACL to model the correlations among multiple action proposals and then iteratively update their features . The nodes of the graph are initialized by the feature representations of action proposals. We propose a multi-level consistency evaluator which exploits the high-level contextual similarity, low-level temporal coincidence and feature correlations between action proposals to compute the adjacency matrix. The node features are updated by a Gated Recurrent Unit (GRU) [25], and the updated features are employed to regress the temporal boundaries of action proposals to obtain the final action co-localization results.

We perform extensive experiments to evaluate our G-TACL method and compare it with other state-of-the-art methods on the THUMOS14 [26], MEXaction2 [27], and ActivityNet v1.3 [28] datasets. Both THUMOS14 and MEXaction2 are consisted of sports actions, and ActivityNet v1.3 is consisted of actions in daily life. Furthermore, we conduct four groups of ablation studies to explore the contribution of each component of our proposed method.

The primary contributions of this paper are summarized as follows:

- To our knowledge, this is the first work to define and solve the temporal action co-localization problem. By taking advantages of the correlation among multiple action instances of the same category, our G-TACL can effectively co-localize action instances in an untrimmed video.
- We propose a multi-level consistency evaluator to compute the correlation between action proposals in G-TACL. It captures the information between any two proposals based on their low-level temporal coincidences, feature correlations and high-level contextual similarities.
- Experimental results on three benchmarks demonstrate the great advantage of the proposed G-TACL over previous state-of-the-arts.

This paper extends the preliminary conference version [29] in four aspects. First, we include and discuss more recent works in Section 2 Related Work. Second, we provide more details about our method and the corresponding implementation. Third, both quantitative and qualitative results on a larger dataset (*i.e.*, ActivityNet v1.3 [28]) are included to verify the effectiveness of our proposed G-TACL over other existing methods. Last but not least, we carry out more extensive ablation studies to explore the contribution of each component to the performance of G-TACL.

The rest of this paper is organized as follows. We review the related works in Section 2. Section 3 describes the technical details of our proposed G-TACL.

Section 4 presents the experiment details and results. Finally, we summarize the paper in Section 5.

2. Related Work

Since our work aims at co-localizing action instances of a common category in an untrimmed video based on GNNs, we review related work on action recognition, temporal action localization and graph-based networks. We also briefly review related work on object detection, because they often adopt a similar framework as temporal action localization.

2.1. Action Recognition

Action recognition means to classify the actions in videos. A considerable amount of previous efforts are limited to classifying actions in manually trimmed short videos [12, 13], where each video contains only one action instance and there is no interference from either other action instances or a complex background. Before the emergence of deep learning, there are many methods relying on hand-crafted features, such as histograms of image gradients (HOG) [30] and improved Dense Trajectory (iDT) [13]. The action category is predicted based on the feature representations of the video, *e.g.*, via Fisher Vector (FV) [31] or Vector of Linearly Aggregated Descriptors (VLAD) [32].

In recent years, CNN-based techniques have revolutionized this area [12, 14, 24, 33], and have significantly pushed forward the state-of-the-art performance. Simonyan *et al*. [12] propose a two-stream architecture where two structurally identical CNNs are used respectively to process spatial and temporal information in videos. Karpathy *et al*. [34] study three fusion strategies (*i.e.*, early fusion, late fusion, and slow fusion) for the two streams, which offers a promising way to speed up the training process. Carreira *et al*. [24] propose to combine two-stream networks with I3D to further boost the action recognition accuracy. Wang *et al*. [14] learn hand-crafted iDT-based video descriptors via CNNs. Tran *et al*. [2] extract temporal and spatial features from multiple frames simultaneously by using 3D ConvNets. Li *et al*. [35] propose a unified Spatio-Temporal Attention Network (STAN) for action recognition in untrimmed videos, which locates the key video segments and spatial areas. Feichtenhofer *et al*. [36] leverage a fast pathway and a slow pathway to capture motion and spatial features, respectively.

In addition, some researchers also try to use the relationship between human and object for action understanding recently. Zhou *et al*. [37] propose a cascade architecture for multi-stage human-object interaction (HOI) understanding on images, which includes a localization network to refine HOI proposals and an interaction recognition network to mine semantic information so as to boost relationship

reasoning. Wang *et al*. [38] propose a hierarchical human parsing method, which uses three different relation networks to decompose, compose and infer the relations of different parts of the human body, and then obtain better parsing results through global message passing. Similar to [38], Wang *et al*. [39] also use the information passing and fusion between different parts of the human body for human parsing, which includes three inference processes, *i.e.*, direct inference using image information, top-down inference using constituent parts, and bottom-up inference using context information. The above three methods all focus on images, since there are obvious interactions between different parts of the human body or HOI in images. However, different action instances in a video can only be determined whether they belong to the same action category or not according to the appearance, action and context information.

2.2. Temporal Action Localization

Temporal action localization (TAL) mainly focuses on untrimmed videos typically containing multiple action instances and numerous background scenes. Most state-of-the-art methods are based on sliding windows [1, 6, 31, 40], frame-wise predictions [3, 41, 42], or action proposals [4, 7, 11, 43, 44].

TAL methods based on sliding windows often divide the video into fixed-length overlapping snippets, and then identify the action instances [1, 6, 31, 40]. Wang *et al*. [1] combine hand-crafted motion features and convolutional appearance features for classification. Yuan *et al*. [6] introduce a Pyramid of Score Distribution Features (PSDF) that circumvents fixed-length windows by encoding features at multiple temporal scales followed by an SVM classifier for TAL. Oneata *et al*. [31] use sliding-window classifiers on FV representations of iDT features of videos. To overcome the drawbacks of hand-crafted features and capture motion characteristics, Shou *et al*. [40] use multi-scale sliding windows and 3D ConvNets to determine the action category, and a localization network to locate the temporal boundaries of action instances.

TAL methods based on frame-wise predictions classify each individual frame to determine whether a specific action is present, and then perform TAL by thresholding [3, 41, 45]. Dai *et al*. [45] use a frame-wise classifier and aggregate frames for TAL. Yuan *et al*. [3] leverage a structural maximal sum of frame-wise classification scores to determine the temporal boundaries. Recently, recurrent neural networks (RNNs), *e.g.*, long short-term memory (LSTM), are exploited to model the dynamics among video frames. Yeung *et al*. [41] leverage LSTMs to produce confidence scores of actions based on CNN features per frame.

TAL methods based on action proposals first generate temporal action proposals and then classify and score them for efficient action localization [4, 7, 11, 43, 44]. Lin *et al*. [7] evaluate the starting, ending, and actionness probabilities of

each temporal location in the video, and generate action proposals based on these probabilities. Motivated by the faster R-CNN [46], Xu *et al*. [44] propose R-C3D and switch from classical exhaustive sliding windows to the 3D RoI Pooling that proposes temporal regions from a deep convolution feature map. Zhao *et al*. [4] propose the Structured Segment Networks (SSN) where they introduce the structured temporal pyramid pooling to describe three major stages of action proposal, and apply a decomposed discriminative model to jointly determine its category and completeness. In order to locate action instances of various durations, Chao *et al*. [11] use a multi-tower network and dilated temporal convolutions with different reception fields to align anchors and action instances. Guo *et al*. [43] identify a series of multi-scale temporal action proposals by temporal convolutions.

In the aforementioned works, the correlation among action instances of the same category are not explicitly addressed as we speculate earlier. Hence, we propose the G-TACL method to exploit the correlations among action instances of the common category to facilitate the action co-localization.

2.3. Graph-based Network

Graph is a natural data structure to represent relationships among entities. GNNs exploit the powerful learning capability of neural networks to process graph data, and have recently become increasingly popular in various domains [21–23]. Qi *et al*. [21] propose the Graph Parsing Neural Network (GPNN) to infer human-object interactions in images and videos. Different from CNNs that only model the local relations, Chen *et al*. [22] try to reason global relations via graph convolutions in the interaction space for image/video understanding. Si *et al*. [23] leverage GNNs to capture the spatial structural correlations in each frame for skeleton-based action recognition. Parsa *et al*. [47] propose to use a feature pyramid structure to capture the association between different parts of human in a video, and use them for action recognition. Fan *et al*. [48] propose a spatio-temporal graph network, which can learn relations among persons and iteratively propagate information over the graph for understanding human gaze communication.

The methods mentioned above mainly focus on using GNNs to capture the dynamic changes of the action, rather than the co-localization of the action. Since action instances of the same category are similar in context and appearance, we try to correlate and update the representations of the action proposals using GNNs, which can readily leverage the similarity among action instances.

2.4. Object Detection

The majority of temporal action localization methods are inspired by the two-stage object detection framework, *i.e.*, object/action proposal generation and object/action classification. R-CNN [46] first generates thousands of category-independent

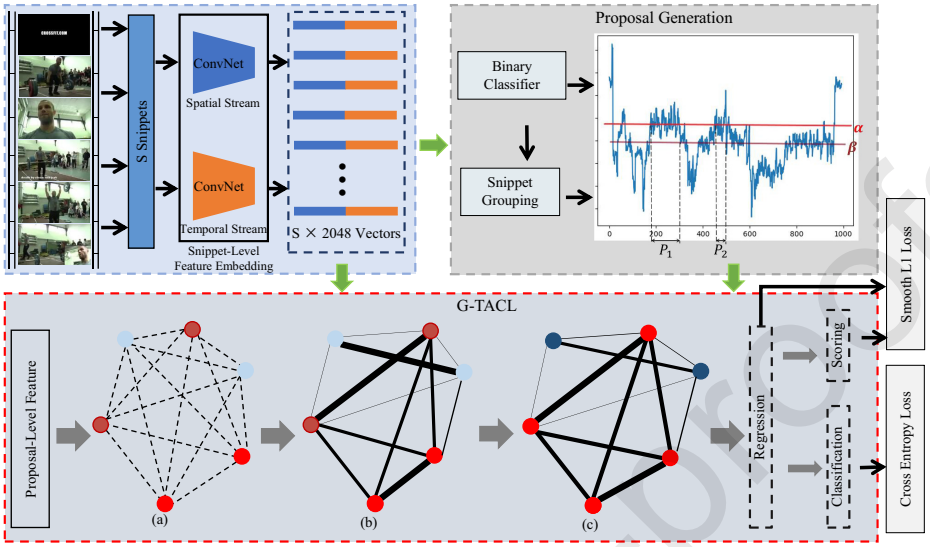


Figure 2: The pipeline of the proposed temporal graph-based temporal action co-localization (G-TACL) method. It consists of three components: feature embedding (upper-left), action proposal generation (upper-right) and G-TACL (bottom).

region proposals using selective search [49], and then extracts a feature vector for each region proposal for classification. Although R-CNN achieves remarkable accuracy, it is computationally expensive because a CNN forward pass is required for each region proposal. Fast R-CNN [50] takes the entire image as input and computes a shared feature map, from which feature vectors are extracted for subsequent classification. Faster R-CNN [51] further speeds up the fast R-CNN by replacing the selective search [49] with the Region Proposal Network (RPN), which computes region proposals from the full-image features with higher accuracy and far less computational cost. Lu *et al.* [52] use the full convolutional network to generate the heatmap of the target object, which is then used to guide the object localization. Wang *et al.* [53] leverage semantic features to guide the generation of proposals. We adopt a similar two-stage framework, *i.e.*, generating action proposals, and detecting action instances and refining their temporal boundaries.

3. Method

Let V denote an untrimmed video with M frames, $V = \{m_t\}_{t=1}^{T}$, where m_t is the t-th frame. V contains a set of action instances $G = \{g_n\}_{n=1}^{N_g}$, where N_g is the number of action instances, $g_n = (t_{s,n}^{gt}, t_{e,n}^{gt}, k_n^{gt})$, and $t_{s,n}^{gt}, t_{e,n}^{gt}, k_n^{gt}$ are the starting frame index, the ending frame index and the action category of the n-

th ground truth action instance g_n, respectively. Our objective is to identify the action instances of the same category and locate the temporal boundaries of them in V. Our G-TACL is a two-stage framework for action co-localization, *i.e.*, action proposal generation followed by classification and temporal boundary regression.

Figure 2 presents the pipeline of our proposed graph-based temporal action co-localization (G-TACL) method. It consists of three components: the feature embedding module (upper-left), the action proposal generation module (upper-right) and the G-TACL (bottom). Our method starts with feature embedding by a pre-trained two-stream I3D network [24]. Then, we generate action proposals by a double-threshold scheme which is robust to noise. Next, these action proposals are fed into the G-TACL for feature enhancement, and finally temporal boundary regression. We describe the details of the three components below.

3.1. Snippet-level Feature Embedding

Snippet-level feature embedding means to obtain a feature representation of the input video. The original video is first split into multiple non-overlapping snippets with fixed-length. A pre-trained two-stream I3D network [24] is applied to embed each snippet into a fixed-length feature vector [1].

For each snippet, it consists of 16 RGB frames or optical flow images and we feed them to the spatial stream or the temporal stream respectively, each resulting in a 1024-dimensional feature vector. The snippet-level features are obtained by concatenating the spatial and the temporal features. Specifically, given the i-th snippet s_i of V, which is divided into S non-overlapping snippets in total, the snippet-level feature embedding can be formulated as

$$\mathbf{F}(i) = [\mathcal{F}_{\text{rgb}}(s_i), \mathcal{F}_{\text{flow}}(s_i)],\ \mathbf{F}(i) \in \mathbb{R}^{1\times 2048}, \tag{1}$$

where $\mathcal{F}_{\text{rgb}}$ and $\mathcal{F}_{\text{flow}}$ denote spatial and temporal stream, respectively. In summary, the output of this step is a feature map of the whole video, *i.e.*, $\mathbf{F} \in \mathbb{R}^{S\times 2048}$.

3.2. Action Proposal Generation

There are two main strategies [3, 40] to generate action proposals. The first strategy classifies each video frame using a pre-trained binary classifier, and obtains action proposals by grouping consecutive frames with classification scores above a certain threshold. However, its computation is very expensive. The second strategy exploits a pre-trained binary classifier to classify video clips generated by multi-scale sliding windows. Then, video clips classified as background are removed

[1] Note that our method is not restricted to any specific feature extractor.

while the ones classified as actions are retained as action proposals. This strategy can only generate fixed-scale action proposals.

Unlike previous methods, we exploit the output scores of an "actionness" [54] binary classifier and design a dual threshold scheme to generate action proposals with flexible lengths and accurate boundaries. Since many background frames exist between action instances, especially in an untrimmed video, it is not appropriate to perform grouping with a fixed threshold. A too large threshold will split a complete action instance into multiple segments while a too small threshold will result in action instances with large portions of background frames.

As illustrated in the upper right part of Figure 2, this category-agnostic binary classifier predicts the actionness scores of input snippets. Empirically, we design a dual threshold scheme, with an action-start threshold α and an action-end threshold β (typically $\beta < \alpha$). A new action proposal starts when the actionness score spikes above α and ends when the actionness score falls below β. With different choices of α and β, a set of L action proposals $\mathbb{P} = \{P_l\}_{l=1}^{L}$ can be obtained, where $P_l = (t_{s,l}, t_{e,l})$, and $t_{s,l}, t_{e,l}$ denote the starting and the ending frame indexes of proposal P_l, respectively. Noting that P_l consists of consecutive snippets so that the starting and the ending boundaries are originally represented by the indexes of snippets, instead of frame indexes. In order to be consistent with the notations of ground truth g_n, we use the index of the first frame of the starting snippet as $t_{s,l}$. Similarly, the index of the last frame of the ending snippet is used as $t_{e,l}$. In our experiments, we explore 8 threshold combinations, *i.e.*, $\alpha \in \{0.5, 0.6, 0.7, 0.8\}$ and $\beta \in \{\alpha - 0.2, \alpha - 0.1\}$.

3.3. G-TACL

After obtaining the feature map $\mathbf{F}$ and the action proposal set $\mathbb{P}$ of the input untrimmed video V, we now describe the proposed Graph-based Temporal Action Co-Localization (G-TACL) method in detail. With action proposals of the same action category, we can intuitively expect higher contextual correlations among them than those across different action categories. In addition, we expect that the quality of these action proposals also affects the contextual correlations. Specifically, we speculate that the correlations among high-quality action proposals of the same category should be higher than those of low quality or of different categories. We formulate such contextual correlations and information transfer/interaction using the GNNs and the iterative GRU [25] updates, respectively.

Defining graph nodes. In the training phase, only action proposals that satisfy one of the following two conditions are used as nodes: (1) Its IoU with a ground truth action instance is greater than 0.5; (2) It has the largest IoU among all action proposals with a ground truth action instance. We denote the set of nodes as $\mathbb{X} = \{X_p\}_{p=1}^{N}$, where X_p is the p-th node. $X_p = (t_{s,p},\ t_{e,p},\ k_p,\ \mathbf{F}_p)$, where

$t_{s,p},\ t_{e,p},\ k_p$, and $\mathbf{F}_p$ denote the starting frame index, the ending frame index, the action category and the feature representation of the corresponding action proposal, respectively. $t_{s,p}$ and $t_{e,p}$ can be directly obtained from the corresponding action proposal. k_p is the category of the ground truth instance which has the largest IoU with X_p. As mentioned in [4], constructing the temporal structure of an action proposal is very helpful for the action localization task. Thus for $\mathbf{F}_p$, we construct the temporal structure by expanding the temporal boundaries of each action proposal on the start and ending boundary. Specifically, for a graph node (*i.e.*, an action proposal) X_p, its feature representation $\mathbf{F}_p$ is obtained by concatenating the features of three parts:

$$\mathbf{F}_p = \text{mean}(\mathbf{F}_p^s,\ \mathbf{F}_p^c,\ \mathbf{F}_p^e),\ \mathbf{F}_p \in \mathbb{R}^{1\times 2048}, \tag{2}$$

where $\mathbf{F}_p^s,\ \mathbf{F}_p^c$, and $\mathbf{F}_p^e$ denote the average features of three snippets before the proposal, the average of all snippet features covered by the action proposal, and the average features of three snippets after the proposal (*i.e.*, the starting, course, and ending stage of an action proposal), respectively.

Computing adjacency matrix. To leverage the correlation between nodes (*i.e.*, action proposals), we need to model them using adjacency matrix. Thus, we use three types of relations to construct the consistency evaluator for adjacency matrix calculation. Specifically, $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ represent low-level temporal coincidence, feature similarity, and high-level contextual similarity between nodes, respectively.

First, if two nodes, X_p and X_q, excessively overlap in the time domain, the proximity between them should be high. Therefore, the corresponding element in the adjacency matrix for low-level temporal coincidences (*i.e.*, $\mathbf{A}_1(p,q)$) is calculated using the temporal overlaps between the two action proposals (noted as $\mathcal{O}(X_p, X_q)$).

Second, the similarity between two feature vectors can be represented by their dot product. The larger their dot product is, the more similar the two feature vectors are. Thus, the corresponding element of the adjacency matrix for features similarity (*i.e.*, $\mathbf{A}_2(p,q)$) is calculated using the dot product between F_p and F_q.

Third, since $\mathbf{A}_1$ relies on temporal location and $\mathbf{A}_2$ directly relies on the original representation, we introduce a trainable layer for modeling high-level contextual similarities to learn beneficial semantic correlation that cannot be directly obtained from $\mathbf{A}_1$ and $\mathbf{A}_2$. Specifically, we concatenate the features of two nodes and use two 1-dimensional convolution layers to obtain the degree of contextual correlation $\mathbf{A}_3(p,q)$ of these two nodes. The final adjacency matrix $\mathbf{A}$ is a weighted sum of $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}_3$.

In summary, the three adjacency matrixes are calculated as

$$\begin{cases} \mathbf{A}_1(p,q) = \mathcal{O}(X_p, X_q) = \dfrac{(t_{s,p}, t_{e,p}) \cap (t_{s,q}, t_{e,q})}{(t_{s,p}, t_{e,p}) \cup (t_{s,q}, t_{e,q})},\ \mathbf{A}_1 \in \mathbb{R}^{N\times N}, \\ \mathbf{A}_2(p,q) = F_p \cdot F_q,\ \mathbf{A}_2 \in \mathbb{R}^{N\times N}, \\ \mathbf{A}_3(p,q) = \mathcal{F}_s([F_p, F_q]),\ \mathbf{A}_3 \in \mathbb{R}^{N\times N}, \end{cases} \tag{3}$$

where $\mathcal{F}_s$ denotes the two 1-dimensional convolution layers. Afterwards, $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ are normalized following

$$\begin{cases} \bar{\mathbf{A}}_1(p,q) = \dfrac{\mathbf{A}_1(p,q)}{\sum_{q'=1}^{N} \mathbf{A}_1(p,q')}, \\ \bar{\mathbf{A}}_2(p,q) = \dfrac{\mathbf{A}_2(p,q)}{\sum_{q'=1}^{N} \mathbf{A}_2(p,q')}, \\ \bar{\mathbf{A}}_3(p,q) = \dfrac{\mathbf{A}_3(p,q)}{\sum_{q'=1}^{N} \mathbf{A}_3(p,q')}. \end{cases} \tag{4}$$

Finally, the adjacency matrix $\mathbf{A}$ is a weighted sum of $\bar{\mathbf{A}}_1$, $\bar{\mathbf{A}}_2$, and $\bar{\mathbf{A}}_3$ as

$$\mathbf{A} = w_1 \cdot \bar{\mathbf{A}}_1 + w_2 \cdot \bar{\mathbf{A}}_2 + w_3 \cdot \bar{\mathbf{A}}_3, \tag{5}$$

where w_1, w_2, w_3 are constants controlling the trade-off among those three terms (elaborated below in Section 4.3). The values in the adjacency matrix represent the similarity between each pair of graph nodes (*i.e.*, action proposals).

Updating node features. Having obtained the features of nodes $\mathbf{F}_p$ ($p = 1, 2, ..., N$) and the adjacency matrix $\mathbf{A}$, we update the feature of a node by accounting for all other similar nodes based on the adjacency matrix. Therefore, the correlation information among nodes (*i.e.*, action proposals) are leveraged to enhance the original node feature. The updating process consists of two steps, *i.e.*, message propagation and feature updating.

For the message propagation step, the goal is to collect contextual information associated with the node from other nodes. As shown in Figure 3, the message propagation is achieved based on the adjacency matrix, which is defined as

$$\mathbf{m}_p = \sum_{q=1}^{N} \mathbf{A}(p,q) \cdot F_q, \tag{6}$$

where $\mathbf{m_p}$ represents the temporal, appearance and contextual related information of node X_p gathered from all its interacted graph nodes.

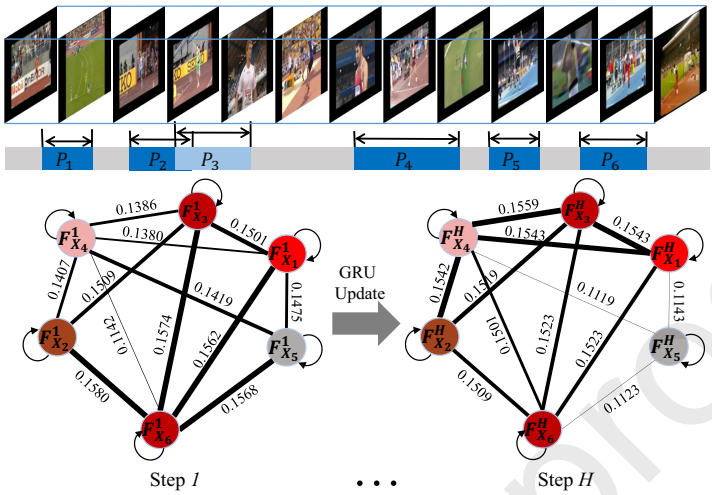


Figure 3: Visualization of the graph updating process. At each iteration, the message prorogation step is to collect the contextual features of a node from similar nodes as defined in Eq. (6). Then a GRU is utilized to update the node features as defined in Eq. (7).

For the feature updating step, we use GRU [25] as the update function to update the node features, since it has fewer parameters and is easy to train. At each iteration step η ($\eta = 1, \ldots, H$), the GRU update is formulated as

$$\begin{cases} \mathbf{h}_p^0 = \mathbf{F}_p, \\ \mathbf{o}_p^\eta,\ \mathbf{h}_p^\eta = \mathrm{GRU}(\mathbf{h}_p^{\eta-1},\ \mathbf{m}_p^\eta), \\ \mathbf{F}_p^\eta = \mathbf{h}_p^\eta, \end{cases} \tag{7}$$

where $\mathbf{m}_\mathbf{p}^\eta$, $\mathbf{o}_\mathbf{p}^\eta$, $\mathbf{h}_\mathbf{p}^\eta$ denotes the aggregated features of node p, the output state and the hidden state of GRU at η-th iteration, respectively. $\mathbf{h}_\mathbf{p}^\eta$ (*i.e.*, $\mathbf{h}_p^0$) is initialized with the initial features of proposals (*i.e.*, $\mathbf{F}_p$) at the first iteration. At each iteration, we update $\mathbf{F}_p^\eta$ with $\mathbf{h}_\mathbf{p}^\eta$.

Regression, classification and scoring. With the updated node features (*i.e.*, proposal features), we classify the actions of these nodes. Meanwhile, we regress the temporal boundaries using the same features, to achieve better alignment with the target ground truth action instances. Since each action instance may generate multiple action proposals, we need to compute the confidence score of each action proposal (node) to retrieve the results.

Specifically, for a node X_p, its temporal boundaries are $t_{s,p}$ and $t_{e,p}$, and the corresponding temporal center location and duration are $l = (t_{s,p} + t_{e,p})/2$ and $d = t_{e,p} - t_{s,p}$, respectively. We obtain the regression results by feeding the updated

features into a stacked 1-dimensional convolution network with a hidden layer. The output consists of two elements Δl and Δd, which represent the predicted center location and length offset, respectively. The regressed center location, duration and new boundaries (localization result) can be calculated as

$$\begin{cases} l' = l + d \cdot \Delta l, \\ d' = d \cdot e^{\Delta d}, \end{cases} \tag{8}$$

$$\begin{cases} t'_{s,p} = l' - d'/2, \\ t'_{e,p} = l' + d'/2, \end{cases} \tag{9}$$

where l' and d' are the temporal center location and duration after regression, respectively. $t'_{s,p}$ and $t'_{e,p}$ are the new temporal boundaries.

3.4. Loss

The regressed action proposals are classified and scored based on the features of the regressed temporal location. We use a fully connected layer for classification and a stack of two 1-dimensional convolution layers for scoring. During the training phase, we freeze the parameters of the feature embedding module and only learn the parameters of G-TACL. We calculate the regression loss L_{reg} and the scoring loss L_{sco} based on the temporal boundaries after the regression, and calculate the classification loss L_{cls} using the classification result. The G-TACL network is trained by penalizing a summation of those three losses as

$$\mathcal{L} = L_{reg} + L_{sco} + L_{cls}, \tag{10}$$

where both L_{reg} and L_{sco} use the smoothing L_1 loss function and L_{cls} uses the Cross Entropy loss function.

4. Experiments

In this section, we evaluate the TAL performance of the proposed G-TACL. Four groups of ablation studies are conducted to explore the performance contribution of each component in G-TACL. Moreover, we compare our G-TACL method with a variety of existing state-of-the-art TAL methods on three standard benchmarks, *i.e.*, THUMOS14 [26], MEXaction2 [27], and ActivityNet v1.3 [28].

4.1. Implementation Details

We implement the model and the evaluation pipeline using PyTorch [55]. We use the I3D [24] network pre-trained on the Kinetics dataset [56] as the backbone to extract video features. The input of the I3D network is a 16-frame RGB/optical flow, and the output is a 1024-dimensional feature vector of the corresponding snippet.

We train a binary classifier to score each video snippet and predict whether it contains the target action instances or not, and then group them according to the actionness scores, as described in section 3.2. We prepare the training data as follows. We sample multiple video clips by sliding a temporal window along the untrimmed video. The sliding window spans to 32 frames for THUMOS14 [26], 16 frames for MEXaction2 [27], and 128 frames for ActivityNet v1.3 [28]. For each video clip, we determine its label by measuring its overlapping ratio with the ground truth action instances. If the overlapping ratio is higher than 0.7, we treat it as positive, denoting that it contains an action k ($\forall k \in 1, \ldots, K$); if the overlapping ratio is lower than 0.3, we treat it as negative, denoting that it does not contain any actions. We keep the ratio of positive and negative samples at $1:1$.

We optimize the parameters of G-TACL by Stochastic Gradient Descent (SGD). On THUMOS14 [26] and MEXaction2 [27], the initial learning rate is 10^{-3} which is decayed by 0.1 at epoch 80 and again at epoch 150, along with a momentum fixed at 0.9 throughout the training process. On ActivityNet v1.3 [28], the initial learning rate is fixed at 10^{-3} and decreased to 10^{-4} and 10^{-5} at epoch 20 and 40, respectively. Empirically, the number of node feature updates has little effect on the experimental results, therefore it is fixed at 1 ($H = 1$) for computational efficiency.

4.2. Datasets and Evaluation Metrics

Evaluation datasets. We conduct extensive experiments on three benchmark datasets to evaluate our proposed G-TACL method, including THUMOS14 [26], MEXaction2 [27], and ActivityNet v1.3 [28]. Table 1 presents the statistics of these three datasets.

- THUMOS14 [26] dataset is challenging and widely used in to evaluate TAL, which includes over 200 hours video data and 20 action categories with temporal annotations. It contains 4 subsets, *i.e.,* training, validation, testing, and

[2]Since some video files are corrupted, there are 33 and 25 videos in the training and testing sets, respectively.

[3]There are $9,337$ and $4,575$ videos accessible can be download from YouTube in the training and validation sets, respectively.

Table 1: The statistics of the standard benchmarks we used.

Dataset	THUMOS14 [26]	MEXaction2[2] [27]	ActivityNet v1.3[3] [28]
Train videos	200	33	$9,337$
Test videos	212	25	$4,575$
Action categories	20	2	200
Instances per video	15.2	22.44	1.54

background set. The training set is the UCF101 [57] dataset consisting of $13,320$ trimmed videos, and the validation set and testing set contains $1,010$ and $1,574$ untrimmed videos, respectively. All videos contain multiple action instances, while most of them contain only one action category. The background set consists of $2,500$ untrimmed videos not containing any target action instances. We only use 200 videos in the validation set and 212 videos in the testing set in which temporal annotations are provided. We use the validation set for training and the testing set for evaluation.

- MEXaction2 [27] dataset contains two action categories, *i.e.*, "Bull Charge Cape" and "Horse Riding". It is consisted of YouTube clips, UCF101 Horse Riding clips and untrimmed INA videos. YouTube clips and UCF101 [57] Horse Riding clips are trimmed videos. We just use the INA subset of untrimmed videos in our experiments, which contains 38, 18 and 32 videos for training, validation and testing, respectively. There are $1,336$, 310 and 329 action instances in the training, validation and test set, respectively. The average duration of INA videos is 39 minutes, of which less than 3% are action instances. We train the G-TACL with the training set and test it with the testing set.

- ActivityNet v1.3 [28] dataset is currently the largest dataset for TAL. It includes over 600 hours video data within 200 action categories, which are all from daily life. It is divided into training, validation and test set by 2:1:1, and there are $10,024$, $4,926$ and $5,044$ videos in each of them, respectively. Each video contains an average of 1.54 action instances. Since the ground-truth for test set is not released, we use the training set for training and validation set for testing.

Evaluation metric. The mean average precision (mAP) with respect to different IoUs is used as the evaluation metric, which is conventional in the literature of TAL. A prediction is considered correct if the category label is correct and the temporal IoU with the ground truth exceeds the IoU threshold. Multiple mAP values under

Table 2: Ablation study on node features update. G-TACL outperforms G-TACL without node features update at all IoU thresholds on the THUMOS14 dataset by using both 2D and 3D backbones.

IoU threshold		0.3	0.4	0.5	0.6	0.7	Avg.
2D backbone	Baseline	38.7	32.1	27.5	19.6	11.9	26.0
	G-TACL	**49.4**	**39.5**	**31.1**	**22.0**	**14.7**	**31.3**
3D backbone	Baseline	48.2	40.4	34.3	25.3	17.2	33.1
	G-TACL	**56.8**	**45.8**	**36.5**	**25.6**	**17.9**	**36.5**

Table 3: Ablation study on the consistency evaluator on the THUMOS14 dataset. All three parts in consistency evaluator are compatible and each single part can boost the performance.

$w_1 : w_2 : w_3$	0.3	0.4	0.5	0.6	0.7	Avg.
0 : 0 : 0	48.2	40.4	34.4	25.3	17.2	33.1
1 : 0 : 0	54.9	43.3	35.5	25.4	17.3	35.3
0 : 1 : 0	54.4	44.1	35.1	25.2	17.2	35.2
0 : 0 : 1	55.4	44.9	35.8	25.5	17.6	35.8
1 : 1 : 1	56.3	45.3	36.0	25.4	17.3	36.1
3 : 4 : 3	**56.8**	**45.8**	**36.5**	**25.6**	**17.9**	**36.5**

different IoU thresholds are reported. We use the evaluation code provided by the ActivityNet v1.3 [28] benchmark[4]. A lager mAP at the testing stage indicates better performance.

4.3. Ablation Study

Evaluation of the node features update. To validate the efficacy of the proposed G-TACL, we compare it with a baseline aggregation strategy, *i.e.*, G-TACL without node features update, on the THUMOS14 dataset. The results are summarized in Table 2, where "Baseline" means no feature update and "G-TACL" denotes our proposed method. Note that after removing the "node features update" described in Section 3.3, our network can still perform action localization. The results show that our proposed G-TACL can significantly improve the performance of temporal action co-localization at all the IoU thresholds, regardless of using a 2D backbone (31.1% versus 27.5% with IoU $= 0.5$) or a 3D one (36.5% versus 34.3% with IoU $= 0.5$).

Comparison with different consistency evaluators. We employ three kinds of relations to construct the adjacency matrices. We speculate that the three components of the consistency evaluator might not contribute equally to the node features

[4] https://github.com/activitynet/ActivityNet/tree/master/Evaluation/

Table 4: Exploration of the G-TACL with different number of iterations at multiple IoU thresholds on the THUMOS14 dataset.

IoU threshold		0.3	0.4	0.5	0.6	0.7	Avg.
2D backbone	H=1	49.4	39.5	31.1	22.0	14.7	31.3
	H=2	49.8	39.6	30.6	21.5	13.8	31.0
	H=3	49.4	39.7	30.8	21.7	13.9	31.1
3D backbone	H=1	56.8	45.8	36.5	25.6	17.9	36.5
	H=2	56.7	46.0	36.2	25.6	17.6	36.4
	H=3	56.8	46.1	36.4	25.3	17.3	36.4

update. Thus, we assess each of them by setting the weights of the others to 0. The results at IoU thresholds of $[0.3 : 0.1 : 0.7]$ on the THUMOS14 dataset are presented in Table 3, and they verify our assumptions. It shows every individual component (especially the high-level contextual similarities) can boost the performance. We empirically tune the weights and find a ratio of $w_1 : w_2 : w_3 = 3 : 4 : 3$ yields reasonable performance.

Comparison with different backbones. As mentioned before, our proposed G-TAL is not tied to any specific feature extractor, and can exploit different backbones to extract features. We chose two different backbones, 2D-based Inception-V3 [58] network and 3D-based I3D [24] network in our experiments. As the results on the THUMOS14 dataset shown in Table 2, our G-TAL method is proven to be effective on both 2D and 3D backbones. In general, the representation capability of I3D [24] is stronger than that of Inception-V3 [58]. As a result, it is clear that using I3D [24] as the backbone can further boost the performance (36.5% versus 31.1% with IoU $= 0.5$) compared with the 2D backbone.

Effect of the number of iterations. Our proposed G-TACL can iteratively update node features as elaborated in Section 3.3. Table 4 presents the impact of the number of iterations at IoU thresholds of $[0.3 : 0.1 : 0.7]$ on the THUMOS14 dataset. It can be interpreted that the number of iterations has little effect on the performance. The reason is that one node is connected to all other nodes, indicating by the proposed adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. Therefore, even only one step is used, the proposed method can capture the information from all nodes of the graph. As more iterations result in higher computation cost, we set $H = 1$ in our experiments.

4.4. Comparison with State-of-the-art Methods

We conduct extensive experiments on the THUMOS14 [26], MEXaction2 [27], and ActivityNet v1.3 [28] datasets, and compare the proposed approach with state-of-the-art TAL methods quantitatively and qualitatively. As far as we know, there

Table 5: Comparison with the state-of-the-art TAL methods on the THUMOS14 testing set. The best results are denoted in bold black, and '-' means they do not report the corresponding results.

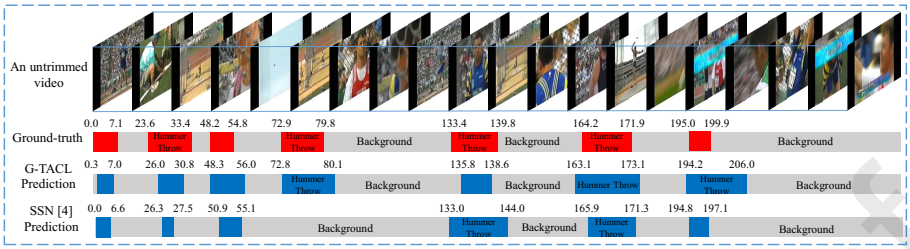
IoU threshold	0.1	0.2	0.3	0.4	0.5	0.6	0.7
LEAR [59]	36.6	33.6	28.8	21.8	15.0	8.5	3.2
S-CNN [40]	47.7	43.5	36.3	28.7	19.0	–	–
SMS [3]	51.0	45.2	36.5	27.8	17.8	–	–
TUBE [5]	–	–	39.8	27.2	20.7	–	–
CDC [60]	–	–	40.1	29.4	23.3	13.1	7.9
R-C3D [44]	54.5	51.5	44.7	35.6	28.9	–	–
SSAD [61]	50.1	47.8	43.0	35.0	24.6	–	–
SS-TAD [62]	–	–	45.7	–	29.2	–	9.6
TCN [45]	–	–	–	33.3	25.6	15.9	9.0
SSN [4]	66.0	59.4	51.9	41.0	29.8	19.6	10.7
TPC [63]	–	–	49.4	39.5	31.1	22.2	14.7
CTAP [8]	–	–	–	–	29.9	–	–
AP-Trees [64]	48.5	44.1	38.2	29.8	20.1	–	–
BSN [7]	–	–	53.5	45.0	36.9	28.4	20.0
TR-C3D [10]	56.9	54.7	51.2	43.0	36.1	–	–
TAL-Net [11]	59.8	57.1	53.2	**48.5**	**42.8**	**33.8**	**20.8**
G-TACL	**67.3**	**62.9**	**56.8**	45.8	36.5	25.6	17.9

is currently no TACL method, so we compare our model with TAL methods.

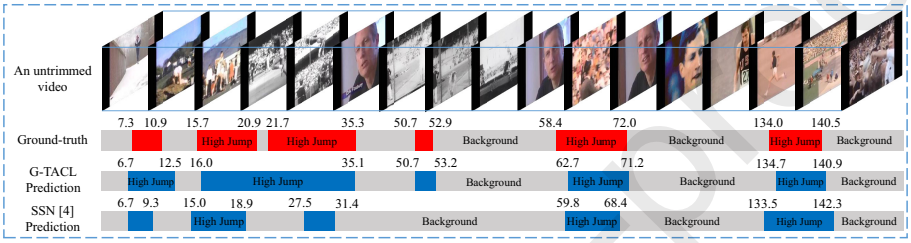
Experiment on THUMOS14. 16 existing TAL methods, *i.e.*, LEAR [59], S-CNN [40], SMS [3], TUBE [5], CDC [60], R-C3D [44], SSAD [61], SS-TAD [62], TCN [45], SSN [4], TPC [63], CTAP [8], AP-Trees [64], BSN [7] TR-C3D [10] and TAL-Net [11], are included as competing algorithms on the THUMOS14 dataset. We present the localization performance of these methods at IoU thresholds of $[0.1 : 0.1 : 0.7]$ in Table 5. Figure 4 visualizes the detection results of two action categories from the THUMOS14 testing set.

As shown in Table 5, our proposed G-TACL outperforms all other methods on all IoU thresholds from 0.1 to 0.7 (by a margin from 0.3% to 21.5% with IoU $= 0.5$), except for TAL-Net [11] and BSN [7]. Compared with TAL-Net [11] and BSN [7], our G-TACL outperforms them when IoU < 0.4 and is comparable to them when IoU threshold ≥ 0.4. This indicates that it is beneficial to utilize the temporal, appearance, and contextual correlation between action proposals to regress the temporal boundaries.

We illustrate the AP (IoU $= 0.5$) on every action category of the THUMOS14 dataset and compare them with three competing methods, *i.e.*, S-CNN [40], R-C3D [44], and SS-TAD [62], in Figure 5. The results show that our G-TACL



(a) An example of *HummerThrow*



(b) An example of *HighJump*

Figure 4: Qualitative examples of the proposed G-TACL on the THUMOS14 testing set. The ground truth temporal locations, predictions and background are illustrated with red, blue and gray bars, respectively.

Table 6: Comparisons with three existing methods on the MEXaction2 testing set (IoU = 0.5). The best results are denoted in bold black.

Category	Bull Charge Cape	Horse Riding	mAP
DTF [27]	0.3	3.1	1.7
S-CNN [40]	11.6	3.1	7.4
SSAD [61]	**16.5**	5.5	11.0
G-TACL	10.9	**15.7**	**13.3**

obviously outperforms the others on most of the categories, but performs poorly on a few categories. A possible reason is that while the proposal features can be enhanced for most videos, they will be weakened if most of action proposals are of low-quality. Thus, the variance of our results on different categories are relatively small. In contrast, the results of the other methods are either particularly good or particularly poor.

Experiment on MEXaction2. We compare our G-TACL with three existing TAL methods, *i.e.*, DTF [27], S-CNN [40] and SSAD [61], on the MEXaction2 dataset. Following conventions [27], we evaluate our method at the IoU threshold of 0.5. The APs of two action categories, *i.e.*, "Bull Charge Cape" and "Horse Riding",

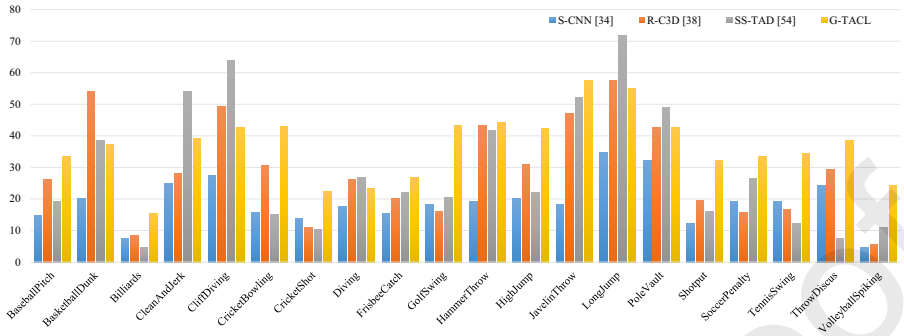


Figure 5: The AP of each action category on the THUMOS14 testing set (IoU = 0.5). Our method obviously outperforms the others in more than half of the categories.

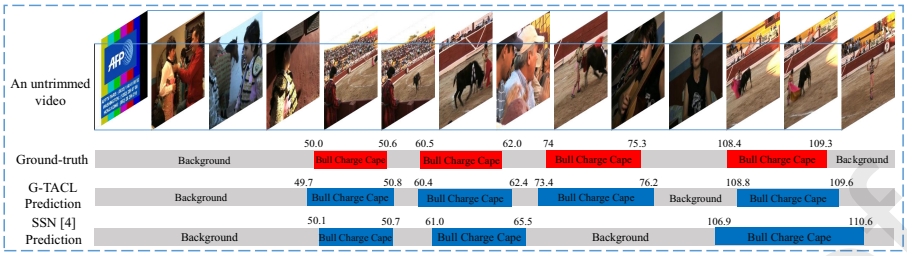
Table 7: Comparisons with 10 state-of-the-art methods on the ActivityNet v1.3 validation set at multiple different IoU thresholds. The best results are denoted in bold black.

IoU threshold	0.5	0.75	0.95	Avg.
SAC[42]	22.7	10.8	0.3	11.3
UTS [65]	43.7	–	–	–
MSB-RNN [66]	26.0	15.2	2.6	14.6
R-C3D [44]	26.8	–	–	12.7
CDC [60]	45.3	26.0	0.2	16.4
TCN [45]	36.4	21.2	3.9	–
SSN [4]	39.1	23.5	5.5	24.0
TR-C3D [10]	27.7	–	–	15.4
TAL-Net [11]	38.2	18.3	1.3	20.2
BSN [7]	46.5	**29.9**	**8.0**	**30.0**
G-TACL	**48.7**	**29.6**	6.8	**29.4**

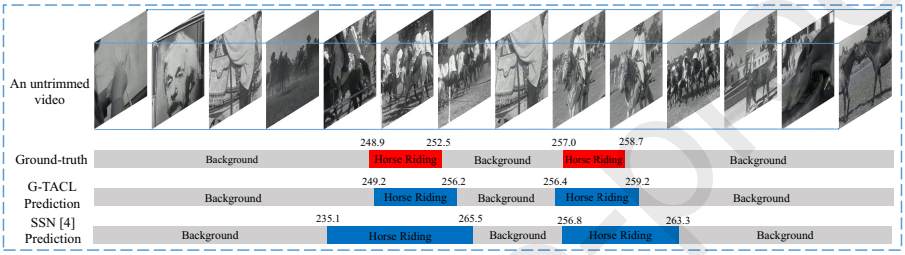
and the mAPs of those methods are summarized in Talel 6. Figure 6 presents the qualitative results of the proposed G-TACL on the MEXaction2 testing set. Note that the video duration in this dataset is very long while the duration of the action instance is very short.

As shown in Table 6, our proposed G-TACL achieves excellent performance compared with DTF [27], S-CNN [40], and SSAD [61]. However, the overall performance on MEXaction2 is still relatively low, since the videos contain a large amount of background frames and the ground truth annotations are not accurate enough, *e.g.*, some complete action instances are over-segmented into multiple ones.

Experiment on ActivityNet v1.3. We further evaluate our proposed G-TACL and compare it with 10 state-of-the-art methods, *i.e.*, SAC [42], UTS [65], MSB-



(a) An example of *Bull Charge Cape*



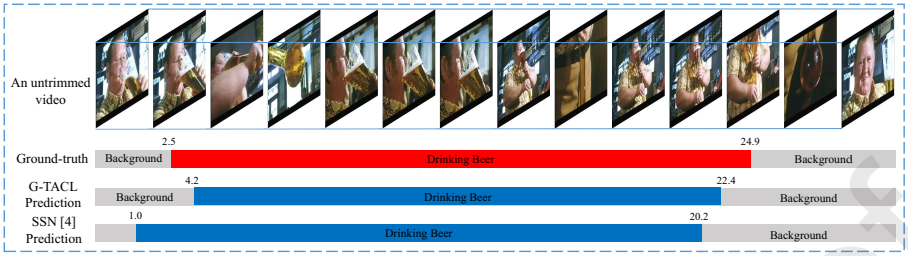
(b) An example of *Horse Riding*

Figure 6: Qualitative examples of the proposed G-TACL on the MEXaction2 testing set. The ground truth temporal locations, predictions and background are illustrated with red, blue and gray bars, respectively.

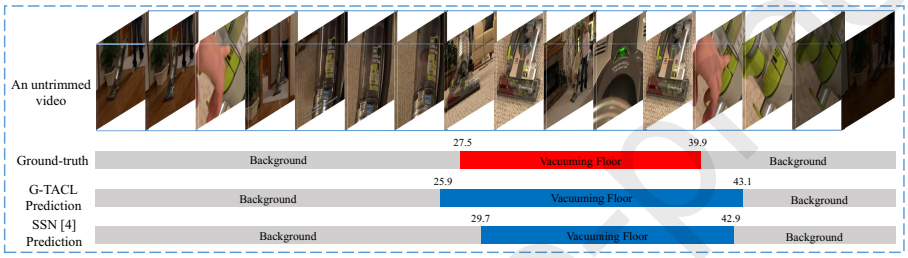
RNN [66], R-C3D [44], CDC [60], TCN [45], SSN [4], TR-C3D [10], TAL-Net [11] and BSN [7], on the ActivityNet v1.3 dataset. The mAPs at different IoU thresholds (the IoU thresholds are chosen from $\{0.5, 0.75, 0.95\}$, the same as previous work) and the average mAPs at IoU thresholds $[0.5 : 0.05 : 0.95]$ are presented in Table 7. We also present two qualitative examples in Figure 7.

The results in Table 7 show that our G-TACL outperforms all other competing methods by a margin from 2.2% to 26% when IoU $= 0.5$. The performance of our G-TACL is significantly better than all other methods except BSN [7], and it performs only slightly worse than BSN [7] at larger IoU thresholds. Although the videos in ActivityNet v1.3 contain fewer action instances, our method can still achieve better performance. This demonstrates that it is beneficial to leverage the temporal coincidence between action proposals for TAL.

It should be noted that TAL-Net [11] is capable of handling videos where the duration of action instances varies greatly, while the duration of action instances is generally long and that of the background is relatively short in ActivityNet v1.3, and thus TAL-Net performs not well. In contrast, our G-TACL is robust to the variation in duration of action instances, regardless of whether the duration is long



(a) An example of *Drinking Beer*



(b) An example of *Vacuuming Floor*

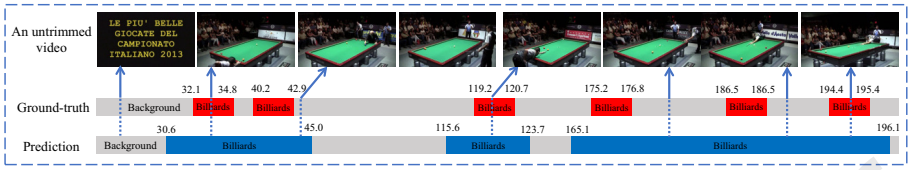
Figure 7: Qualitative examples of the proposed G-TACL on the ActivityNet v1.3 validation set. The ground truth temporal locations, predictions and background are illustrated with red, green and light blue bars, respectively.

(ActivityNet v1.3[28]), short (MEXaction2 [27]), or variable (THUMOS14 [26]), mainly due to the utilization of correlations among multiple action instances of the same category.

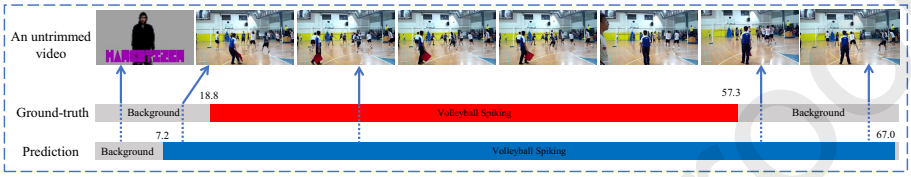
From the results on the above three datasets, it is clear that our G-TACL has achieved excellent results, and is superior or comparable to all state-of-the-art methods. This demonstrates that our G-TACL is capable of capturing the temporal and contextual features across multiple action proposals in an untrimmed video to help the localization of each individual action instance. Moreover, extensive ablation studies verify the effectiveness of each component of G-TACL.

4.5. Failure Cases Discussion

The proposed method leverages the similarity between proposals to facilitate temporal action localization. However, it brings a drawback that if the similarity between the background and action is very high, the information from background proposals may also affect foreground proposals, resulting in inaccurate localization reults. As shown in Figure 8, the action is similar to the background because of the existence of the scene (*i.e.*, the pool table and the volleyball court). In these cases, the scenes dominate the proposal features, making the background proposals



(a) An example of *Billiards*



(b) An example of *VolleyballSpiking*

Figure 8: Qualitative examples of failure cases together with frames from the input untrimmed videos. The ground truth temporal locations and predictions are illustrated with red and light blue bars, respectively. Generally, our method can successfully identify the backgrounds in different scenes. However, if the scene dominates the video, the background proposals will have high similarity with action proposals, which may cause inaccurate localization results.

have high similarity with action proposals. Therefore, the feature updating process will introduce background visual elements to the action nodes, causing the false-positive localization. In the future, we plan to improve the design of the adjacency matrix to eliminate the misleading caused by the dominating scene to achieve more precise temporal action localization.

5. Conclusion

In this paper, we propose the Graph-based Temporal Action Co-Localization (G-TACL) method to simultaneously locate action instances of a common category from an untrimmed video. Different from previous methods, G-TACL exploits appearance and contextual correlations among multiple action instances to facilitate the temporal localization of each individual action instance. We relate action proposals using GNNs whose nodes are initialized by action proposals and iteratively updated by aggregating similar contextual features. This is beneficial for precise temporal boundary regression. Moreover, we propose a multi-level consistency evaluator as an indicator of the similarity between action proposals to calculate the adjacency matrix. Experiments on three benchmark datasets have verified the efficacy of our proposed G-TACL method.

Acknowledgment

This work was supported partly by National Key R&D Program of China Grant 2018AAA0101400, NSFC Grants 61629301, 61773312, and 61976171, China Postdoctoral Science Foundation Grant 2019M653642, and Young Elite Scientists Sponsorship Program by CAST Grant 2018QNRC001.

References

[1] L. Wang, Y. Qiao, X. Tang, Action recognition and detection by combining motion and appearance features, in: Proc. Eur. Conf. Comput. Vis. THUMOS Workshop, 2014.

[2] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 4489–4497.

[3] Z.-H. Yuan, J. C. Stroud, T. Lu, J. Deng, Temporal action localization by structured maximal sums, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 3215–3223.

[4] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, D. Lin, Temporal action detection with structured segment networks, in: Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2914–2923.

[5] X. Duan, L. Wang, C. Zhai, N. Zheng, Q. Zhang, Z. Niu, G. Hua, Joint spatio-temporal action localization in untrimmed videos with per-frame segmentation, in: Proc. IEEE Int. Conf. Image Process., 2018, pp. 918–922.

[6] J. Yuan, B. Ni, X. Yang, A. A. Kassim, Temporal action localization with pyramid of score distribution features, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 3093–3102.

[7] T. Lin, X. Zhao, H. Su, C. Wang, M. Yang, Bsn: Boundary sensitive network for temporal action proposal generation, in: Proc. Eur. Conf. Comput. Vis., 2018, pp. 3–19.

[8] J. Gao, K. Chen, R. Nevatia, Ctap: Complementary temporal action proposal generation, in: Proc. Eur. Conf. Comput. Vis., 2018, pp. 68–83.

[9] Z. Gao, L. Wang, Q. Zhang, Z. Niu, N. Zheng, G. Hua, Video imprint segmentation for temporal action detection in untrimmed videos, in: Proc. AAAI Conf. Artificial Intelligence, 2019, pp. 8328–8335.

[10] H. Xu, A. Das, K. Saenko, Two-stream region convolutional 3d network for temporal activity detection, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2019) 2319–2332.

[11] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, R. Sukthankar, Rethinking the faster r-cnn architecture for temporal action localization, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 1130–1139.

[12] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Proc. Adv. Neural Inf. Process. Syst.,, 2014, pp. 568–576.

[13] H. Wang, C. Schmid, Action recognition with improved trajectories, in: Proc. IEEE Int. Conf. Comput. Vis., 2013, pp. 3551–3558.

[14] L. Wang, P. Koniusz, D. Q. Huynh, Hallucinating idt descriptors and i3d optical flow features for action recognition with cnns, in: Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 8698–8708.

[15] X. Lv, L. Wang, Q. Zhang, N. Zheng, G. Hua, Video object co-segmentation from noisy videos by a multi-level hypergraph model, in: Proc. IEEE Int. Conf. Image Process., 2018, pp. 2207–2211.

[16] L. Wang, G. Hua, R. Sukthankar, J. Xue, Z. Niu, N. Zheng, Video object discovery and co-segmentation with extremely weak supervision, IEEE Trans. Pattern Anal. Mach. Intell. 39 (10) (2017) 2074–2088.

[17] C.-C. Tsai, W. Li, K.-J. Hsu, X. Qian, Y.-Y. Lin, Image co-saliency detection and co-segmentation via progressive joint optimization, IEEE Trans. Image Process. 28 (1) (2018) 56–71.

[18] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, F. Porikli, See more, know more: Unsupervised video object segmentation with co-attention siamese networks, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 3623–3632.

[19] W. Wang, X. Lu, J. Shen, D. J. Crandall, L. Shao, Zero-shot video object segmentation via attentive graph neural networks, in: Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 9236–9245.

[20] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, L. Van Gool, Video object segmentation with episodic graph memory networks, in: Proc. Eur. Conf. Comput. Vis., 2020.

[21] S. Qi, W. Wang, B. Jia, J. Shen, S.-C. Zhu, Learning human-object interactions by graph parsing neural networks, in: Proc. Eur. Conf. Comput. Vis., 2018, pp. 407–423.

[22] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, Y. Kalantidis, Graph-based global reasoning networks, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 433–442.

[23] C. Si, Y. Jing, W. Wang, L. Wang, T. Tan, Skeleton-based action recognition with spatial reasoning and temporal stack learning, in: Proc. Eur. Conf. Comput. Vis., 2018, pp. 103–118.

[24] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 6299–6308.

[25] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, in: Proc. Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, 2014, pp. 103–111.

[26] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, R. Sukthankar, Thumos challenge: Action recognition with a large number of classes (2014).

[27] 2015, Mexaction2, `http://mexculture.cnam.fr/xwiki/bin/view/Datasets/Mex+action+dataset`.

[28] F. Caba Heilbron, V. Escorcia, B. Ghanem, J. Carlos Niebles, Activitynet: A large-scale video benchmark for human activity understanding, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 961–970.

[29] C. Zhai, L. Wang, Q. Zhang, Z. Gao, Z. Niu, N. Zheng, G. Hua, Action co-localization in an untrimmed video by graph neural networks, in: Proc. Int. Conf. Multimedia Modeling, 2019, pp. 555–567.

[30] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2005, pp. 886–893.

[31] D. Oneata, J. Verbeek, C. Schmid, Action and event recognition with fisher vectors on a compact feature set, in: Proc. IEEE Int. Conf. Comput. Vis., 2013, pp. 1817–1824.

[32] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2010, pp. 3304–3311.

[33] J.-M. Pérez-Rúa, V. Vielzeuf, S. Pateux, M. Baccouche, F. Jurie, Mfas: Multimodal fusion architecture search, in: Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 6966–6975.

[34] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 1725–1732.

[35] D. Li, T. Yao, L.-Y. Duan, T. Mei, Y. Rui, Unified spatio-temporal attention networks for action recognition in videos, IEEE Trans. Multimedia 21 (2) (2018) 416–428.

[36] C. Feichtenhofer, H. Fan, J. Malik, K. He, Slowfast networks for video recognition, in: Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 6202–6211.

[37] T. Zhou, W. Wang, S. Qi, H. Ling, J. Shen, Cascaded human-object interaction recognition, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 4263–4272.

[38] W. Wang, H. Zhu, J. Dai, Y. Pang, J. Shen, L. Shao, Hierarchical human parsing with typed part-relation reasoning, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 8929–8939.

[39] W. Wang, Z. Zhang, S. Qi, J. Shen, Y. Pang, L. Shao, Learning compositional neural information fusion for human parsing, in: Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 5703–5713.

[40] Z. Shou, D. Wang, S.-F. Chang, Temporal action localization in untrimmed videos via multi-stage cnns, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 1049–1058.

[41] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, L. Fei-Fei, Every moment counts: Dense detailed labeling of actions in complex videos, Int. J. Comput. Vis. 126 (2-4) (2018) 375–389.

[42] G. Singh, F. Cuzzolin, Untrimmed video classification for activity detection: submission to activitynet challenge, arXiv preprint arXiv:1607.01979.

[43] D. Guo, W. Li, X. Fang, Fully convolutional network for multiscale temporal action proposals, IEEE Trans. Multimedia 20 (12) (2018) 3428–3438.

[44] H. Xu, A. Das, K. Saenko, R-c3d: region convolutional 3d network for temporal activity detection, in: Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 5794–5803.

[45] X. Dai, B. Singh, G. Zhang, L. S. Davis, Y. Qiu Chen, Temporal context network for activity localization in videos, in: Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 5793–5802.

[46] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 580–587.

[47] B. Parsa, A. Narayanan, B. Dariush, Spatio-temporal pyramid graph convolutions for human action recognition and postural assessment, in: Proc. IEEE Winter Conf. Applications Comput. Vis., 2020, pp. 1080–1090.

[48] L. Fan, W. Wang, S. Huang, X. Tang, S.-C. Zhu, Understanding human gaze communication by spatio-temporal graph reasoning, in: Proc. IEEE Int. Conf. Comput. Vis., 2019, pp. 5724–5733.

[49] J. R. Uijlings, K. E. Van De Sande, T. Gevers, A. W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.

[50] R. Girshick, Fast r-cnn, in: Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1440–1448.

[51] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Proc. Adv. Neural Inf. Process. Syst.,, 2015, pp. 91–99.

[52] X. Lu, B. Li, Y. Yue, Q. Li, J. Yan, Grid r-cnn, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 7363–7372.

[53] J. Wang, K. Chen, S. Yang, C. C. Loy, D. Lin, Region proposal by guided anchoring, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 2965–2974.

[54] L. Wang, Y. Qiao, X. Tang, L. Van Gool, Actionness estimation using hybrid fully convolutional networks, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2708–2717.

[55] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, in: Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 8024–8035.

[56] W. Kay, J. Carreira, K. Simonyan, B. Zhang, A. Zisserman, The kinetics human action video dataset, arXiv preprint arXiv:1705.06950.

[57] K. Soomro, A. R. Zamir, M. Shah, Ucf101: A dataset of 101 human actions classes from videos in the wild, arXiv preprint arXiv:1212.0402.

[58] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2818–2826.

[59] D. Oneata, J. Verbeek, C. Schmid, The lear submission at thumos 2014, in: Proc. Eur. Conf. Comput. Vis. THUMOS Workshop, 2014.

[60] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, S.-F. Chang, Cdc: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1417–1426.

[61] T. Lin, X. Zhao, Z. Shou, Single shot temporal action detection, in: Proc. ACM Multimedia, 2017, pp. 988–996.

[62] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, J. C. Niebles, End-to-end, single-stream temporal action detection in untrimmed videos, in: Proc. British Mach. Vis. Conf., 2017.

[63] K. Yang, P. Qiao, D. Li, S. Lv, Y. Dou, Exploring temporal preservation networks for precise temporal action localization, in: Proc. AAAI Conf. Artificial Intelligence, 2018, pp. 7477–7484.

[64] H. Song, X. Wu, B. Zhu, Y. Wu, M. Chen, Y. Jia, Temporal action localization in untrimmed videos using action pattern trees, IEEE Trans. Multimedia 21 (3) (2018) 717–730.

[65] R. Wang, D. Tao, Uts at activitynet 2016, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. ActivityNet Challenge Workshop, 2016.

[66] B. Singh, T. K. Marks, M. Jones, O. Tuzel, M. Shao, A multi-stream bi-directional recurrent neural network for fine-grained action detection, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 1961–1970.