Action Co-Localization in an Untrimmed Video by Graph Neural Networks

Changbo Zhai[1], Le Wang[1*], Qilin Zhang[2], Zhanning Gao[3], Zhenxing Niu[3], Nanning Zheng[1], and Gang Hua[4]

[1] Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P.R.China
[2] HERE Technologies, Chicago, IL 60606, USA
[3] Alibaba Group, Hangzhou, Zhejiang 311121, P.R.China
[4] Wormpex AI Research, Bellevue, WA 98004, USA

Abstract. We present an efficient approach for action co-localization in an untrimmed video by exploiting contextual and temporal feature from multiple action proposals. Most existing action localization methods focus on each individual action instances without accounting for the correlations among them. To exploit such correlations, we propose the Graph-based Temporal Action Co-Localization (G-TACL) method, which aggregates contextual features from multiple action proposals to assist temporal localization. This aggregation procedure is achieved with Graph Neural Networks with nodes initialized by the action proposal representations. In addition, a multi-level consistency evaluator is proposed to measure the similarity, which summarizes low-level temporal coincidences, features vector dot products and high-level contextual features similarities between any two proposals. Subsequently, these nodes are iteratively updated with Gated Recurrent Unit (GRU) and the obtained node features are used to regress the temporal boundaries of the action proposals, and finally to localize the action instances. Experiments on the THUMOS'14 and MEXaction2 datasets have demonstrated the efficacy of our proposed method.

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

1 Introduction

The temporal action co-localization task aims to jointly locate the action instances of the same category within an untrimmed video, which includes simultaneous action recognition (identify the category of each action instance) and temporal localization (identify the temporal boundaries of each action instance).

Considerable progress has been made to address the temporal action localization problem in untrimmed videos [32, 5–7, 26, 38, 39, 4, 8, 20]. Techniques including hand-crafted features (iDT) [32, 37], Convolution Neural Networks (CNNs) [39, 20] and 3-dimensional Convolution Networks (3D ConvNets) [4, 24]

*Corresponding Author, lewang@xjtu.edu.cn

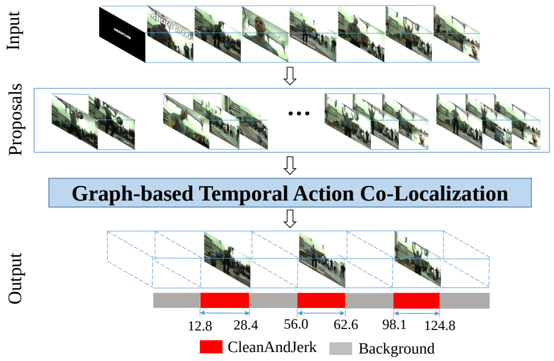


Fig. 1. Flowchart of the proposed G-TACL method. The input is an untrimmed video, which contains multiple action instances of the same category (*e.g.*, CleanAndJerk, marked with the red chunks at the bottom). There are a large number of background frames (marked with the gray chunks) not containing such action instances. The outputs are the predicted categories and the temporal boundaries of action instances.

have been proposed and empirically demonstrated to be beneficial. However, the correlation between multiple action proposals of the video in the same category are usually neglected, which could otherwise be potentially beneficial due to the appearance and structure consistency between proposals. Similar strategy of exploiting the appearance and structural consistency of instances has been demonstrated in image/video object co-segmentation [18, 31, 16]. Actually, it is common that many videos contain multiple action instances of the same category, such as triple jump videos or clean and jerk videos about the Olympic Games. Therefore, a temporal action co-localization algorithm exploiting such correlations could be reasonably desirable.

Graph Neural Networks (GNNs) have been widely adopted in many areas including human-object interaction detection [21] and scene understanding [35]. The GNNs inherit the advantages of both CNNs and graphical models and have strong capabilities of representing and learning the correlation among targets [21]. Inspired by the success of GNNs, we devise the Graph-based Temporal Action Co-Localization (G-TACL) algorithm which represents the correlations using GNNs for co-localizing action instances.

Figure 1 illustrates the flowchart of our proposed action co-localization method. We first employ the Two-Stream network [26] to extract snippet-level features. And then a binary classifier is applied to compute the confidence score of whether each video snippet belongs to the action or not. To generate high-quality action proposals, a two-step thresholding strategy is utilized to group snippets according to confidence scores. Finally, we leverage the G-TACL to model the correlations among multiple action proposals and then iteratively update the action proposal features. The nodes of the graph are initalizated by the representations of action proposals, and we propose a multi-level consistency evaluator which exploits high-level contextual similarity and low-level temporal coincidence between action proposals to construct the adjacency matrix. The node features are updated by GRU [3] and the updated features are employed to regress the temporal boundaries of action proposals and to obtain the final action co-localization result.

The primary contributions of this paper are summarized as follows. (1) We propose G-TACL for video action co-localization, which takes advantage of the correlation among multiple action proposals of the same category in an untrimmed video. (2) We propose a multi-level consistancy evaluator for G-TACL, which accounts for low-level temporal coincidences, features vector dot products and high-level contextual features similarities.

2 Related Work

Action Recognition. Action recognition involves the classification of actions in videos. Methods based on hand-crafted features [13, 30] and deep neural networks [11, 26, 29] have been studied extensively. Karpathy *et al.* [11] propose to use CNNs for video classification. Simonyan *et al.* [26] propose the Two-stream architecture where two structurally identical 2D ConvNets are used respectively to proccess spatial and temporal information in videos. Tran *et al.* [29] propose to extract temporal and spatial features from multiple frames simultaneously by using 3D ConvNets.

Action Localization. Action localization mainly focuses on untrimmed videos that containing at least one action instance and numerous background scenes [32, 37, 39, 25, 34, 17]. Wang *et al.* [32] combine manually crafted features representing motion and CNN features representing appearance for classification. To overcome the drawbacks of hand-crafted features and capture motion characteristics, Shou *et al.* [25] use multi-scale sliding windows and 3D ConvNet to determine the action category and a localization network for the temporal boundaries of action instances. Motivated by the original faster R-CNN [22], Xu *et al.* [34] propose R-C3D where they switch from classical exhaustive sliding windows to the 3D RoI Pooling that proposes temporal regions from a convolutional feature map. Zhao *et al.* [39] propose the Structured Segment Networks (SSN) where they introduce the structured temporal pyramid pooling to describe three major stages of an action proposal, and apply a decomposed discriminative model to jointly determine its category and completeness. Still, in the aforementioned literatures, the correlation among action proposals of the same category are not explicitly addressed as we speculate earlier.

Graph-based Network. Graph is a natural data structure to represent relationships among entities. GNNs extend the powerful learning potential of neural networks to process graph data, and have recently become increasingly popular in various domains [9, 14, 21, 28]. Li *et al.* [14] propose a situation recognition method based on GNNs, which can capture joint dependencies between roles in an image. Qi *et al.* [21] propose the Graph Parsing Neural Network (GPNN) to infer human-object interactions in images and videos. 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.

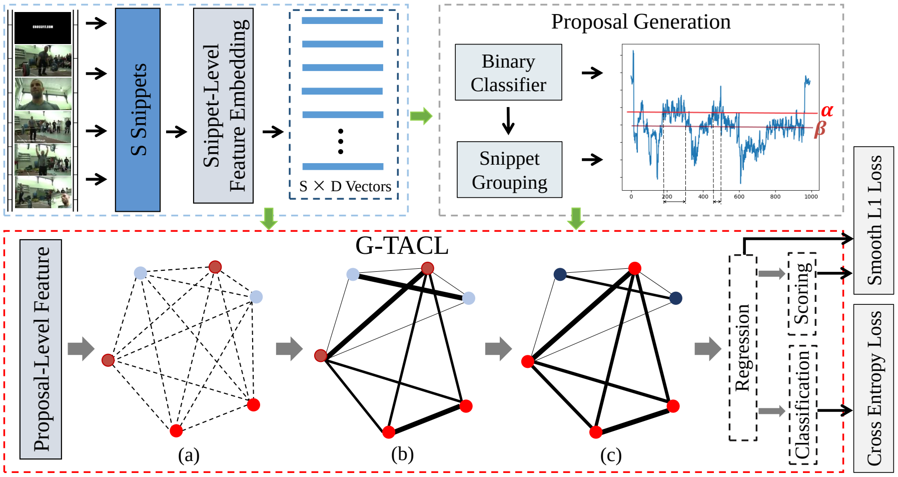


Fig. 2. Pipeline of the proposed temporal action co-localization method. It consists of three parts: feature extraction (upper-left), action proposal generation (upper-right) and G-TACL (bottom).

3 Method

Let V denote an untrimmed video with T frames, $V = \{t_m\}_{m=1}^{M}$, where t_m is the m-th frame. V contains a set of action instances $G = \{g_n = (t_{s,n}, t_{e,n}, k_n)\}_{n=1}^{N_g}$, where N_g is the number of action instances, $t_{s,n}, t_{e,n}$, and k_n are the starting, ending frame and the action category of the n-th instance g_n, respectively. Our goal is to identify the action in video V and to predict the temporal boundaries and the category of each action instance. We adopt a two-stage framework, proposal-and-classification, *i.e.*, we first generate action proposals, and then process them using G-TACL to obtain the final action co-localization result. Figure 2 presents the pipeline of our proposed temporal action co-localization method.

3.1 Snippet-level feature embedding

The goal of this step is to obtain the video representation. The original video is first split into multiple non-overlapping, fixed-length snippets. A pre-trained Two-stream networks [26] is applied to embed each snippet into a fixed-length feature vector[1].

For each snippet, we randomly sample one RGB frame and five consecutive optical flow images and feed them to the spatial stream and the temporal stream respectively, each producing in a 1024-dimensional feature vector. The snippet-level feature is obtained by concating the spatial and the temporal features. Specifically, given a video $V = \{s_i\}_{i=1}^{S}$ with S snippets, where s_i denotes the i-th snippet, the snippet-level feature embedding can be formulated as

$$F_i = [\mathcal{F}_{\text{rgb}}(s_i), \mathcal{F}_{\text{flow}}(s_i)], \tag{1}$$

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

where $\mathcal{F}_{\text{rgb}}$ and $\mathcal{F}_{\text{flow}}$ denote temporal and spatial stream, respectively. To sum up, the output of this step is the feature map $F \in \mathbb{R}^{S\times 2048}$.

3.2 Action proposals generation

Unlike conventional sliding windows-based proposal generation, we exploit the output scores of an "actionness" [33] binary classifier and design a dual threshold scheme. As illustrated in the upper right part of Figure 2, this class-agnostic binary classifier estimates the actionness scores of input snippets. As noted in [39], there are many noisy frames in the untrimmed videos. Empirically, we design a dual threshold scheme, with separate action-starts threshold α and action-ends threshold β (typically $\beta < \alpha$). A new action proposal is obtained once its the actionness score spikes above α and until its actionness score falls below β. With different choices of α and β, a set of L proposals $\mathbb{P} = \{P_l\}_{l=1}^{L}$ can be obtained. In our experimental settings, $\alpha \in \{0.5, 0.6, 0.7, 0.8\}$ and $\beta \in \{\alpha - 0.2, \alpha - 0.1\}$ accordingly, therefore a total of 8 threshold combinations are explored.

3.3 G-TACL

With action proposals of the same action category, we can theoretically expect higher contextual correlations among them than those across different action categories. In addition, we expect that the quality of these action proposals also affect the contextual correlations. Specifically, we speculate that the correlations among high-quality action proposals of the same category should be higher than those among their low-quality counterparts. To leverage such information, we formulate such contextual correlations and information transfer/interaction with the GNNs and the iterative GRU [3] updates, respectively.

Defining graph nodes. In the training phase, only the proposals that satisfy one of the following two conditions are used as nodes: (1) Its IoU with ground-truth is greater than 0.5; (2) It has the largest IoU of all proposals with ground-truth. We denote the set of nodes as $\mathbb{X} = \{X_j\}_{j=1}^{N}$, where X_j is the j-th node. $X_j = (t_{s,j},\ t_{e,j},\ k_j,\ F_{X_j})$, where $t_{s,j},\ t_{e,j},\ k_j$, and F_{X_j} denote the starting, ending frame, action category and the feature representation of the corresponding proposal, respectively. F_{X_j} is obtained by concating the features of three parts:

$$F_{X_j} = [F^{s}_{X_j}, F^{c}_{X_j}, F^{e}_{X_j}], \tag{2}$$

where $F^{s}_{X_j}$, $F^{c}_{X_j}$, and $F^{e}_{X_j}$ denote the average of three snippets features before the proposal, the average of all snippets features covered by the proposal and the average of three snippets features after the proposal (*i.e.* the starting, course, and ending stage of a proposal), respectively.

Computing adjacency matrix. We use three kinds of relations to construct the consistency evaluator, noted as $\boldsymbol{A_1}$, $\boldsymbol{A_2}$, and $\boldsymbol{A_3}$ below, which represent low-level temporal coincidences, features vector dot products and high-level contextual similarities, respectively. First, if two nodes, X_p and X_q, have a high overlap in the time domain, then the proximity between them should be high. We calculate the overlap of temporal region between proposals (noted as $\mathcal{O}(X_p, X_q)$) to obtain $\boldsymbol{A_1}(p, q)$. Second, the similarity between vectors can be represented by their dot product. We calculate the dot product between F_{X_p} and F_{X_q} to obtain $\boldsymbol{A_2}(p, q)$. Third, we adopt a simple multi-layer perception model with one hidden layer to capture the contextual correlation between nodes. Specifically, we concat the features of two nodes and use two 1-dimensional convolution layer with kernel size of 3 to obtain the degree of contextual correlation $\boldsymbol{A_3}(p, q)$ of these two nodes. The final adjacency matrix is the weighted sum of $\boldsymbol{A_1}$, $\boldsymbol{A_2}$, and $\boldsymbol{A_3}$. It can be formulated as

$$\begin{cases} \boldsymbol{A_1}(p,q) = \mathcal{O}(X_p, X_q) = \dfrac{(t_{s,p}, t_{e,p}) \cap (t_{s,q}, t_{e,q})}{(t_{s,p}, t_{e,p}) \cup (t_{s,q}, t_{e,q})} \\ \boldsymbol{A_2}(p,q) = F_{X_p} \cdot F_{X_q} \\ \boldsymbol{A_3}(p,q) = \mathcal{F}_s([F_{X_p}, F_{X_q}]) \\ \boldsymbol{A}(p,q) = w_1 \cdot \boldsymbol{A_1} + w_2 \cdot \boldsymbol{A_2} + w_3 \cdot \boldsymbol{A_3} \end{cases}, \tag{3}$$

where $\mathcal{F}_s$ denotes the two 1-dimensional convolution layer and w_1, w_2, w_3 are constants to control the trade-off of those three terms (detailed in Section 4.3). The value in the adjacency matrix is the similarity between action proposals.

Updating node features. Our goal is to update the node features based on all other nodes in the graph and its own state during message propagation. We first aggregate features of all similar nodes with the message function, and then use the update function to update the node features. The message function is defined as

$$\boldsymbol{m_p} = \sum_q \boldsymbol{A}(p,q) \cdot F_{X_q}. \tag{4}$$

We use GRU [3] as the update function to update the node features. At each iteration η ($\eta = 1, \ldots, H$), the update function is formulated as

$$\begin{cases} \boldsymbol{o_p^\eta},\ \boldsymbol{h_p^\eta} = \mathrm{GRU}(\boldsymbol{h_p^{\eta-1}},\ \boldsymbol{m_p^\eta}) \\ F_{X_p}^\eta = \boldsymbol{o_p^\eta} \end{cases}, \tag{5}$$

where $\boldsymbol{m_p^\eta}$ denotes the aggregated features of node p at η-th iteration. We update F_{X_p} with the hidden state of the GRU.

Regression, classification and scoring. We use the updated node features to classify the actions of the nodes and regress the temporal boundaries t_s and t_e, so that the regressed temporal region is better aligned with the target action instance. Since each action instance will generate multiple proposals, we need to compute the confidence score of each node to retrieve the results.

Specifically, for a node X_p, its temporal boundaries is $t_{s,p}$ 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 representing the predicted center location and length offset, respectively.The regressed center location, duration and new boundaries (localization result) can be calculated by

$$\begin{aligned} &l' = l + d \cdot \varDelta l, \quad d' = d \cdot e^{\varDelta d}, \\ &t'_{s,p} = l' - d'/2, \quad t'_{e,p} = l' + d'/2. \end{aligned} \tag{6}$$

The regressed proposals are then classified and scored based on the features of the regressed temporal location. We use a fully connected layer for classification and and a stacked two 1-dimensional convolution layer for scoring. During the training phase, we fix the parameters of the feature embedding module and only learn the parameters of G-TACL. We calculate the regression loss and the scoring loss (Smooth L_1) based on the temporal boundaries after the regression, and use the classification result to calculate the classification loss (Cross Entropy).

4 Experiments

4.1 Datasets and evaluation metrics

We conduct extensive experiments on two benchmark datasets to evaluate the proposed method, including THUMOS'14 [10] and MEXaction2 [1].

THUMOS'14 dataset is challenging and widely used in temporal action localization task, which includes 20 action categories with temporal annotations. The validation set and test set contain $1,010$ and $1,574$ untrimmed videos, respectively. Each video contains multiple action instances. We only use 200 videos in validation set and 212 videos in test set in which temporal annotations are provided. We use the validation set for training and the test set for evaluation.

MEXaction2 dataset contains two action categories, *i.e.*, "Bull Charge Cape" and "Horse Riding". It is consisted of YouTube clips, UCF101 Horse Riding clips (these clips are trimmed videos), and untrimmed INA videos. We just use the INA subset of untrimmed videos in our experiments, which contains 38 training and 32 test videos of 2 categories. 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 test set.

The mean average precision (mAP) with respect to different IoUs is used as evaluation metric, which is the conventional metric used in temporal action localization task. A prediction is considered correct if and only if the category label is correct and the temporal IoU with ground-truth exceeds the IoU threshold. Multiple mAP values under different IoU thresholds are reported.

4.2 Implementation details

We implement the model and the evaluation pipeline using PyTorch. We refer the feature embedding module as SSN [39], and use the Inception-V3 [27] network pre-trained on Kinetics dataset[12] as the network backbone, with the last classification layer removed. We optimise the parameters of G-TACL in an end-to-end manner in 35 epochs using stochastic gradient descent (SGD), with an initial learning rate of 0.0001 annealed by 0.1 after epoch 15 and again at epoch 25, and a momentum fixed at 0.9. Empirically, the number of node features updates as little effect on the experimental results, therefore it is fixed at 1 ($H = 1$) for computational efficiency in our experiments.

4.3 Ablation study

Evaluation of the G-TACL. To validate the efficacy of the proposed G-TACL, we compare it with a baseline aggregation strategy, specifically, G-TACL without node feature update, on THUMOS14 dataset. The result is summarised in Table 1, where "Baseline" denotes no feature update and "G-TACL" denotes our proposed method. The results showed that our proposed G-TACL can significantly improve the performance of temporal action co-localization at all the IoU thresholds.

Comparison with different consistency evaluators. We speculate that the three components of the consistency evaluator might not contribute equally on node feature update, and assess each of them by setting the weights of the others to 0. The results in Table 2 showed that every single component can boost the performance, and thus verified our speculation. We empirically tune the weights and find a ratio of $w_1 : w_2 : w_3 = 2 : 1 : 2$ yields reasonable performance.

The effect of the number of iterations. Our proposed G-TACL can iteratively update node features as detailed in Section 3.3. Table 3 presents the effect of the number of iterations. It can be seen that the number of iterations has little effect on performance. As more iterations will affect the computation efficiency, we set $H = 1$ in our experiments.

Table 1. Ablation study on node feature update. G-TACL outperforms G-TACL without node feature update at multiple IoU thresholds on THUMOS'14 dataset.

IoU	0.3	0.4	0.5	0.6	0.7
Baseline	38.7	32.1	27.5	19.6	11.9
G-TACL	**49.4**	**39.5**	**31.1**	**22.0**	**14.7**

Table 2. Ablation study on consistency evaluator (IoU = 0.5). All the three parts in consistency evaluator are compatible and each single part can boost the performance.

$w_1 : w_2 : w_3$	mAP
0 : 0 : 0	27.5
1 : 0 : 0	29.7
0 : 1 : 0	29.1
0 : 0 : 1	30.1
2 : 1 : 2	**31.1**

Table 3. Exploration of the G-TACL with different number of iterations at multiple IoU thresholds on THUMOS'14 dataset.

IoU threshold	0.3	0.4	0.5	0.6	0.7
H=1	49.4	39.5	**31.1**	**22.0**	**14.7**
H=2	**49.8**	39.6	30.6	21.5	13.8
H=3	49.4	**39.7**	30.8	21.7	13.9

4.4 Comparison with state-of-the-art methods

We compare our method with a variety of recently proposed temporal action localization methods on THUMOS'14 dataset. As shown in Table 4, our method is comparable to other recent methods when IoU threshold < 0.5 and outperforms them when IoU ≥ 0.5. This validated that our proposed G-TACL can generate more accurate temporal boundaries and have better performance. Figure 3 visualizes the localization results of two action categories from the THUMOS'14.

Also, we compare our method with three existing methods on MEXaction2 dataset. As same as the compared methods, we summarise the mAP@IoU=0.5 of each category in Talel 5, our proposed method achieves the best performance.

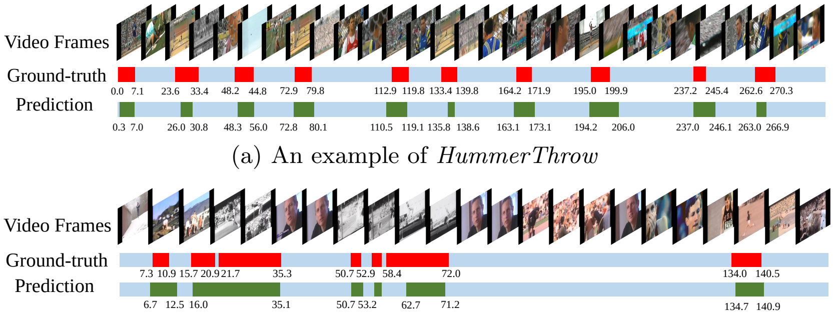
In this paper, we propose the G-TACL to model the contextual correlation between action proposals and update the features of nodes. We compute the AP of each category on the THUMOS'14 dataset and compare them with three existing method in Figure 4. The results showed that our method obviously outperforms the others in more than half of the categories. In the process of information transfer, not only will the feature be enhanced, but also the feature may be weakened, so that our results on all categories are relatively average, unlike other results are particularly good or particularly poor.

Table 4. Comparison with the state-of-the-art temporal action localization methods on the THUMOS'14 test set. G-TACL yields comparable results when IoU threshold < 0.5, and significantly outperforms other methods when IoU threshold ≥ 0.5.

IoU threshold	0.3	0.4	0.5	0.6	0.7
Oneata *et al.* [19]	28.8	21.8	15.0	8.5	3.2
Richard *et al.* [23]	30.0	23.2	15.2	–	–
Yuan *et al.* [38]	36.5	27.8	17.8	–	–
Shou *et al.* [25]	36.3	28.7	19.0	–	–
Duan *et al.* [4]	39.8	27.2	20.7	–	–
Shou *et al.* [24]	40.1	29.4	23.3	13.1	7.9
Xu *et al.*. [34]	44.7	35.6	28.9	–	–
Lin *et al.* [15]	43.0	35.0	24.6	–	–
Buch *et al.* [2]	45.7	–	29.2	–	9.6
Zhao *et al.*[39]	**51.9**	**41.0**	29.8	19.6	10.7
Yang *et al.* [36]	44.1	37.1	28.2	20.6	12.7
G-TACL	49.4	39.5	**31.1**	**22.0**	**14.7**

Table 5. Comparisons with three existing methods on the MEXaction2 test set (IoU = 0.5).

Category	BullCHargeCape	HorseRiding	mAP
MEXaction2 [1]	0.3	3.1	1.7
Shou *et al.* [25]	11.6	3.1	7.4
Lin *et al.* [15]	**16.5**	5.5	11.0
G-TACL	10.0	**13.8**	**11.9**



(a) An example of *HummerThrow*

(b) An example of *HighJump*

Fig. 3. Qualitative examples of the proposed G-TACL on THUMOS'14 test set. The ground-truth temporal locations, predictions and backgrounds are illustrated by red, green and blue bars, respectively.

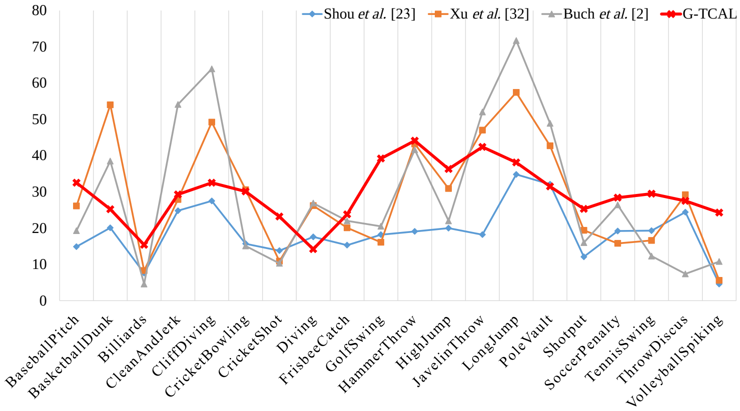


Fig. 4. The AP of each action category on THUMOS'14 test set (IoU = 0.5).

5 Conclusion

In this paper, we propose a graph-based network (G-TACL) for temporal action co-localization in an untrimmed video. In contrast to previous methods, G-TACL can update node by aggregating similar contextual features, which is beneficial for precise temporal boundaries regression. In addition, we propose the multi-level consistency evaluator as an indicator of the similarity between proposals to calculate the adjacency matrix. Experiments on two datasets have verified the efficacy of our proposed method.

Acknowledgements. 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. 2015: Mexaction2. http://mexculture.cnam.fr/xwiki/bin/view/Datasets/Mex+action+dataset
2. Buch, S., Escorcia, V., Ghanem, B., Fei-Fei, L., Niebles, J.C.: End-to-end, single-stream temporal action detection in untrimmed videos. In: BMVC (2017)
3. Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259 (2014)
4. Duan, X., Wang, L., Zhai, C., Zhang, Q., Niu, Z., Zheng, N., Hua, G.: Joint spatio-temporal action localization in untrimmed videos with per-frame segmentation. In: ICIP (2018)
5. Gao, Z., Hua, G., Zhang, D., Jojic, N., Wang, L., Xue, J., Zheng, N.: Er3: A unified framework for event retrieval, recognition and recounting. In: CVPR (2017)
6. Gao, Z., Wang, L., Jojic, N., Niu, Z., Zheng, N., Hua, G.: Video imprint. IEEE Transactions on Pattern Analysis and Machine Intelligence (2018)
7. Gao, Z., Wang, L., Zhang, Q., Niu, Z., Zheng, N., Hua, G.: Video imprint segmentation for temporal action detection in untrimmed videos. In: AAAI (2019)
8. Heilbron, F.C., Niebles, J.C., Ghanem, B.: Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In: CVPR (2016)
9. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-rnn: Deep learning on spatio-temporal graphs. In: CVPR (2016)
10. Jiang, Y., Liu, J., Zamir, A.R., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: Thumos challenge: Action recognition with a large number of classes (2014)
11. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
12. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
13. Laptev, I.: On space-time interest points. International Journal of Computer Vision **64**(2-3), 107–123 (2005)
14. Li, R., Tapaswi, M., Liao, R., Jia, J., Urtasun, R., Fidler, S.: Situation recognition with graph neural networks. In: ICCV (2017)
15. Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. In: ACM MM (2017)
16. Liu, Z., Wang, L., Hua, G., Zhang, Q., Niu, Z., Wu, Y., Zheng, N.: Joint video object discovery and segmentation by coupled dynamic markov networks. IEEE Transactions on Image Processing **27**(12), 5840–5853 (2018)
17. Liu, Z., Wang, L., Zhang, Q., Gao, Z., Niu, Z., Zheng, N., Hua, G.: Weakly supervised temporal action localization through contrast based evaluation networks. In: ICCV (2019)
18. Lv, X., Wang, L., Zhang, Q., Zheng, N., Hua, G.: Video object co-segmentation from noisy videos by a multi-level hypergraph model. In: ICIP (2018)
19. Oneata, D., Verbeek, J., Schmid, C.: The lear submission at thumos 2014. In: ECCV THUMOS Workshop (2014)
20. Paul, S., Roy, S., Roy-Chowdhury, A.K.: W-talc: Weakly-supervised temporal activity localization and classification. In: ECCV (2018)
21. Qi, S., Wang, W., Jia, B., Shen, J., Zhu, S.C.: Learning human-object interactions by graph parsing neural networks. In: ECCV (2018)

22. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS (2015)
23. Richard, A., Gall, J.: Temporal action detection using a statistical language model. In: CVPR (2016)
24. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.F.: Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: CVPR (2017)
25. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage cnns. In: CVPR (2016)
26. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NeurPIS (2014)
27. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016)
28. Tan, H., Wang, L., Zhang, Q., Gao, Z., Zheng, N., Hua, G.: Object affordances graph network for action recognition. In: BMVC (2019)
29. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV (2015)
30. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV (2013)
31. Wang, L., Hua, G., Sukthankar, R., Xue, J., Niu, Z., Zheng, N.: Video object discovery and co-segmentation with extremely weak supervision. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(10), 2074–2088 (2017)
32. Wang, L., Qiao, Y., Tang, X.: Action recognition and detection by combining motion and appearance features. In: ECCV THUMOS Workshop (2014)
33. Wang, L., Qiao, Y., Tang, X., Van Gool, L.: Actionness estimation using hybrid fully convolutional networks. In: CVPR (2016)
34. Xu, H., Das, A., Saenko, K.: R-c3d: region convolutional 3d network for temporal activity detection. In: ICCV (2017)
35. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Graph r-cnn for scene graph generation. In: ECCV (2018)
36. Yang, K., Qiao, P., Li, D., Lv, S., Dou, Y.: Exploring temporal preservation networks for precise temporal action localization. In: AAAI (2018)
37. Yuan, J., Ni, B., Yang, X., Kassim, A.A.: Temporal action localization with pyramid of score distribution features. In: CVPR (2016)
38. Yuan, Z.H., Stroud, J.C., Lu, T., Deng, J.: Temporal action localization by structured maximal sums. In: CVPR (2017)
39. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: ICCV (2017)