Temporal Context Aggregation Network for Temporal Action Proposal Refinement

Zhiwu Qing[1*] Haisheng Su[2†] Weihao Gan[2] Dongliang Wang[2] Wei Wu[2]
Xiang Wang[1] Yu Qiao[3,4] Junjie Yan[2] Changxin Gao[1] Nong Sang[1†]
[1]Key Laboratory of Image Processing and Intelligent Control,
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2]SenseTime Research
[3]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[4]Shanghai AI Laboratory, Shanghai, China
{qzw,wxiang,cgao,nsang}@hust.edu.cn, yu.qiao@siat.ac.cn
{suhaisheng,ganweihao,wangdongliang,wuwei,yanjunjie}@sensetime.com

Abstract

Temporal action proposal generation aims to estimate temporal intervals of actions in untrimmed videos, which is a challenging yet important task in the video understanding field. The proposals generated by current methods still suffer from inaccurate temporal boundaries and inferior confidence used for retrieval owing to the lack of efficient temporal modeling and effective boundary context utilization. In this paper, we propose Temporal Context Aggregation Network (TCANet) to generate high-quality action proposals through "local and global" temporal context aggregation and complementary as well as progressive boundary refinement. Specifically, we first design a Local-Global Temporal Encoder (LGTE), which adopts the channel grouping strategy to efficiently encode both "local and global" temporal inter-dependencies. Furthermore, both the boundary and internal context of proposals are adopted for frame-level and segment-level boundary regressions, respectively. Temporal Boundary Regressor (TBR) is designed to combine these two regression granularities in an end-to-end fashion, which achieves the precise boundaries and reliable confidence of proposals through progressive refinement. Extensive experiments are conducted on three challenging datasets: HACS, ActivityNet-v1.3, and THUMOS-14, where TCANet can generate proposals with high precision and recall. By combining with the existing action classifier, TCANet can obtain remarkable temporal action detection performance compared with other methods. Not surprisingly, the proposed TCANet won the 1st place in the CVPR 2020 - HACS challenge leaderboard on temporal action localization task.

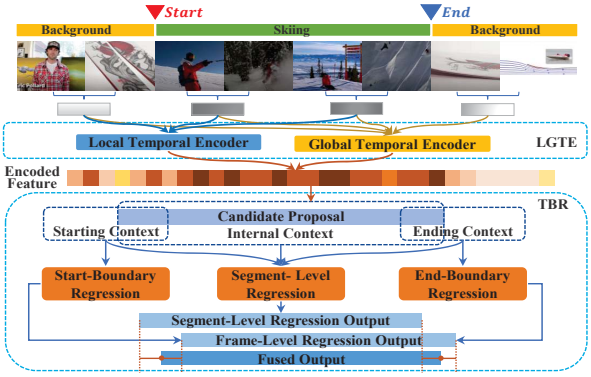

Figure 1. Overview of our proposed method. Given an untrimmed video, TCANet captures the "local and global" temporal relationships in parallel by LGTE. In TBR, the internal and boundary context of proposals are utilized for segment-level boundary regression and frame-level boundary regression, respectively.

1. Introduction

The temporal action detection task requires locating the starting and ending time of action instances from long untrimmed videos and classifying the actions. This task can be applied to many fields, such as video content analysis and video recommendation. Most existing temporal action detection methods follow a two-stage scheme [35, 26], namely temporal action proposal generation and classification. Although action recognition methods [37, 5] have achieved impressive classification accuracy, the temporal action detection performance is still unsatisfactory in several mainstream benchmarks [14, 2, 45]. Hence, many researchers target improving the quality of temporal action proposals.

Current proposal generation methods can be mainly divided into two steps. First, the temporal relationship is captured by stacked temporal convolutions [20, 18, 16, 30, 29]

∗ The work was done during an internship at SenseTime.
† Corresponding author.

or global temporal pooling operations [8]. Then proposals are further generated by the boundary-based regression methods [20, 18] or the anchor-based regression methods [24, 9, 7, 4, 21, 8]. However, existing methods share the following drawbacks: (1) Neither convolution nor global fusion can effectively model the temporal relationship. The $1D$ convolution operations [20, 18, 16] lack flexibility in encoding long-term temporal relationships limited by kernel size. The global fusion methods [8] neglect various global dependencies for each temporal location and the implicit attention to local details, such as local details of boundaries. Besides, simply collecting global features through average pooling may introduce unnecessary noise. (2) Only the internal context or boundary context of proposals used for regression is inferior to generate proposals with precise boundaries. The *internal context* of proposals adopted in anchor-based methods can obtain reliable confidence scores but fails to generate precise boundaries. On the contrary, the *boundary context* of proposals considered in boundary-based methods is sensitive to boundary changes but generates proposals with inferior proposal-level confidence.

To relieve these issues, we propose Temporal Context Aggregation Network (TCANet) for high-quality proposal refinement, as shown in Figure 1. First, the Local-Global Temporal Encoder (LGTE) is proposed to simultaneously capture *local and global* temporal relationships in a channel grouping fashion, which contains two main sub-modules. Specifically, the input features after linear transformation are equally divided into N groups along the channel dimension. Then Local Temporal Encoder (LTE) is designed to handle the first A groups for local temporal modeling. At the same time, the remaining $N - A$ groups are captured by the Global Temporal Encoder (GTE) for global information perception. In this way, LGTE is expected to integrate the long-term context of proposals by global groups while recovering more structure and detailed information by local groups. Second, the Temporal Boundary Regressor (TBR) is proposed to exploit both boundary context and internal context of proposals for frame-level and segment-level boundary regressions, respectively. Concretely, the frame-level boundary regression aims to refine the starting and ending locations of candidate proposals with boundary sensitivity, while the segment-level boundary regression aims to refine the center location and duration of proposals under the overall perception of proposals. Finally, high-quality proposals are obtained through complementary fusion and progressive boundary refinements.

In summary, our contributions mainly have three folds:

- We design a Local-Global Temporal Encoder to simultaneously capture *local and global* temporal relationships in a channel grouping fashion. It can be easily embedded into any other proposal generation frameworks for efficient temporal relationships modeling.
- Temporal Boundary Regressor is proposed to perform complementary and progressive boundary refinements, including the local frame-level boundary regression and global segment-level boundary regression.
- Extensive experiments reveal that TCANet can obtain convincing proposals performance on several benchmarks: HACS, ActivityNet-v1.3, and THUMOS-14. By combining with the existing classifier, TCANet can achieve remarkable temporal action detection performance compared with other approaches.

2. Related Work

Action Recognition. Action recognition is an important task in video understanding area, which is in need of spatio-temporal information modeling. Current deep learning-based action recognition methods can be mainly divided into three types. The first type is 2stream networks [25, 36, 6, 41], which adopt RGB frames and optical flow to capture appearance and motion information. The second type [31, 3, 40, 32] directly captures spatio-temporal information from raw videos using 3D convolution. The third type aims to efficiently model spatio-temporal features by decoupled (2 + 1) D convolutions [22, 5, 17, 13, 15, 28]. In this work, 2stream [25] and SlowFast [5] are adopted to encode the input video features.

Temporal Action Proposal Generation and Detection. Current proposal generation methods can be mainly divided into *anchor-based* and *boundary-based* methods. The *anchor-based* methods [35, 24, 9, 7, 21] refer to the temporal boundary refinements of sliding windows or pre-defined anchors. Among them, TURN [9] and CTAP [7] directly concatenate the boundary context and internal context of proposals for boundary refinements (i.e., starting and ending locations), while other methods aim to refine the duration and center location of proposals. However, refinements on boundary locations cannot make full use of contextual information of proposals, while mere refinement of duration and center location of candidate proposals would also neglect the local boundary details. Therefore, it is non-trivial to combine these two regression granularities into a unified framework. *Boundary-based* methods [20, 18, 27] first generate the boundary probability sequence, then apply the Boundary Matching mechanism to generate candidate proposals. MGG [21] simply combines a boundary-based stream and anchor-based stream with a shared backbone to extract features, then each stream is optimized independently, and the results are fused during the inference. In our work, we make full use of boundary context and internal context of proposals to predict the frame-level offsets (i.e., starting and ending) and the segment-level offsets (i.e., center and duration), respectively. Meanwhile, we jointly train

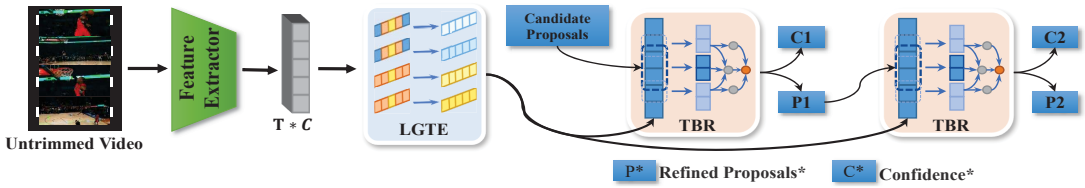

Figure 2. The framework of TCANet. TCANet mainly contains two modules: LGTE and TBR. LGTE is employed to capture *local-global* temporal inter-dependencies simultaneously. TBR is adopted to perform frame-level and segment-level boundary regressions. Finally, high-quality proposals are obtained through complementary fusion and progressive boundary refinements.

these two granularities with supervision performed on the combined results. Finally, complementary and progressive boundary refinements are conducted for better performance.

Self-Attention Mechanism. The self-attention [38] mechanism is widely used in the video understanding area since it can effectively capture long-term dependencies compared with other attention methods such as recurrent models [23] and pooling methods [12]. The Transformer [33] is also based on the self-attention mechanism, which is originally applied in the machine translation task. Girdhar *et al*. [11, 10] adopt Transformer to capture the interactions between human and objects existing in videos. In this paper, we propose Local-Temporal Global Encoder, which can efficiently capture both "local and global" temporal relationships and then integrate rich contexts into extracted video features for temporal proposals generation.

3. TCANet

As shown in Figure 2, we propose **Temporal Context Aggregation Network (TCANet)** to generate high-quality proposals, which mainly consists of two main modules: Local-Global Temporal Encoder and Temporal Boundary Regressor. Firstly, the Local-Global Temporal Encoder (LGTE) is designed to simultaneously encode the input video features' *local and global* temporal relationships. Then the Temporal Boundary Regressor (TBR) is utilized to refine the boundaries of the proposals by exploiting both boundary and internal context for frame-level and segment-level boundary regressions, respectively.

3.1. Problem Formulation

For an untrimmed video X with l frames, we can denote it as $X = \{x_i\}_{i=1}^{l}$, where x_i is the i-th frame of the video. Temporal proposal generation task is to generate a set of proposals $P = \{t_{sj}, t_{ej}\}_{j=1}^{N_p}$ that may contain action instances for video X, where t_{sj} and t_{ej} are the starting time and ending time of the j-th proposal, and N_p is the number of proposals.

3.2. Feature Encoding

For a given video, features are extracted by SlowFast [5] and 2stream [25] since their excellent performance on the video classification task. The frame rate of videos is set to r fps, and each snippet contains s frames. Each snippet is encoded into a visual feature $f_i \in R^C$ by a feature extractor. Given an untrimmed video, a video feature sequence of $F = \{f_i\}_{i=1}^{T} \in R^{T \times C}$ is obtained by this method, where $T = l/\delta$, l is the total number of video frames, and δ is the number of frames interval between different snippets.

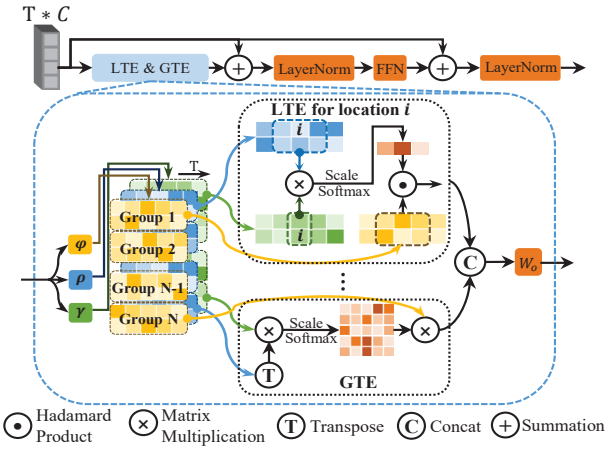

Figure 3. The detailed structure of Local-Global Temporal Encoder (LGTE). The input features are divided into N groups along the channel dimension. Then the first A groups are fed to A Local Temporal Encoders (LTE) separately, where the local dynamic modeling is achieved by calculating the regional attention for each temporal location. The remaining $(N - A)$ groups are processed by $(N - A)$ Global Temporal Encoders (GTE) separately to calculate the similarity between each location and global feature sequence. W_o is a learnable matrix, and Feed Forward Network (FFN) is a nonlinear projection function.

3.3. Local-Global Temporal Encoder

For long videos, long-term temporal dependency modeling is essential, proven by many previous works [8, 39]. Nonlocal [38] is often applied to obtain the relationship between different global locations. However, global modeling only is easy to introduce global noise and insensitive to small boundary changes. We propose a local and global joint modeling strategy to alleviate this problem, as shown in Figure 3.

Local Temporal Encoder (LTE) is responsible for captur-

ing local dependencies based on local details dynamically. Precisely, to measure the relationship between temporal location i and its local areas, the *cosine* similarity between two temporal locations is employed to generate similarity vector S_i^l and weight vector W_i^l:

$$S_i^l = \gamma^l(f_i)\cdot(\rho^l([(f_{i-\lfloor w/2 \rfloor})^T, \cdots (f_{i+\lfloor w/2 \rfloor})^T]^T))^T \in R^{1\times w}, \tag{1}$$

$$W_i^l = \text{Softmax}(\frac{S_i^l}{\sqrt{C}}), \tag{2}$$

where C is the number of channels, w is the size of the modeling area for location i, which is defined as $WindowSize$. For example, the value of w in Figure 3 LTE is 3. γ^l and ρ^l are two different linear projection functions that map the input feature vectors to the similarity measure space.

With equation 1, the relationship between each location and its corresponding modeling area can be calculated. To achieve local information exchange, the following formula will be utilized to collect local context information from the corresponding local area dynamically:

$$f_i^l = W_i^l \cdot (\varphi^l([(f_{i-\lfloor w/2 \rfloor})^T, \cdots (f_{i+\lfloor w/2 \rfloor})^T]^T)), \tag{3}$$

where f_i^l represents the new expression of location i, and φ^l is a linear projection function.

Global Temporal Encoder (GTE) is designed to model the long-term temporal dependencies of videos. Compared with LTE, GTE needs to aggregate global interactions for each location on the temporal dimension. Therefore, the relationship between each location and the global feature is written as follows:

$$S_i^g = \gamma^g(f_i)\cdot(\rho^g(F))^T \in R^{1\times T}, \tag{4}$$

$$W_i^g = \text{Softmax}(\frac{S_i^g}{\sqrt{C}}), \tag{5}$$

where γ^g and ρ^g are two different linear projection functions. The global interaction feature of location i can be updated by weight vector W_i^g:

$$f_i^g = W_i^g \cdot \varphi^g(F), \tag{6}$$

where f_i^g represents the new global feature representation of location i, and φ^g is a linear projection function.

Local-Global Temporal Encoder (LGTE). Each location in the video feature sequence can be modeled locally and globally by LTE and GTE, respectively. However, it is inefficient to combine them in the form of *"LTE-GTE"* simply. To solve this problem, LGTE is implemented in a channel grouping fashion. Specifically, as shown in Figure 3, the input feature is first projected by γ, ρ, and φ. These outputs are then divided into N groups along the channel dimension. Hence the channel number of each group is C/N. The first A groups are handled by LTEs, while the other $N-A$ groups are fed to GTEs. For location i, the combined output of local and global features can be written as:

$$f_i^a = [(f_{1i}^l)^T, \cdots (f_{Ai}^l)^T, (f_{(A+1)i}^g)^T \cdots (f_{Ni}^g)^T]^T \cdot W_o, \tag{7}$$

$$f_i^b = \text{LayerNorm}(f_i^a) + f_i^a, \tag{8}$$

$$f_i^{'} = \text{LayerNorm}(\text{FFN}(f_i^b) + f_i^b), \tag{9}$$

where W_o is a learnable parameter matrix. Inspired by Transformer [33], FFN is adopted to capture the interaction of features among different groups at i-th temporal location: $\text{FFN}(x) = \text{ReLu}(x \cdot W_1 + b_1) \cdot W_2 + b_2$.

Discussion. We notice that our LTE is similar to the convolution with fixed kernels. However, the dynamic local interaction modeling for each temporal location is unique for better adaptation to complex temporal changes than conventional convolution. Furthermore, the combination of LTE and GTE enables our LGTE to capture the global dependencies of whole videos and dynamically model local changes with less noise. Besides, the channel grouping fashion ensures high computing efficiency and the diversity of "local and global" relationships.

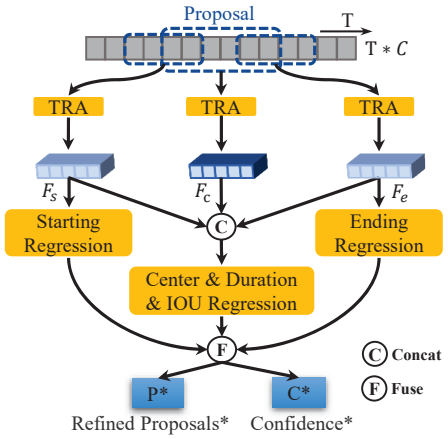

Figure 4. The detailed structure of Temporal Boundary Regressor (TBR). For each input proposal, TBR collects its starting context F_s, internal context F_c and ending context F_e through Temporal Roi Align (TRA). F_s and F_e are adopted to refine the starting and ending locations, respectively. F_s, F_c, and F_e are concatenated to refine the center locations and the duration of proposals. Finally, the accurate proposals are obtained by fusing these two outputs.

3.4. Temporal Boundary Regressor

Anchor-based methods [24, 9, 19, 7, 4, 21, 8] leverage the internal context of proposals to regress center locations and duration, which can obtain reliable scores but with lower recall. *Boundary-based methods* [20, 18] only utilize local boundary context to locate the boundaries, which are sensitive to boundaries but with inferior confidence. Therefore, we propose to combine the boundary context-

based frame-level regression and the internal context-based segment-level regression to refine the boundaries.

Complementary Regression Strategy. As shown in Figure 4, the feature of one proposal is divided into three parts: the starting context F_s, the internal context F_c, and the ending context F_e. To achieve frame-level regression, F_s and F_e are utilized to regress the boundary offsets $\Delta\hat{s}$ and $\Delta\hat{e}$ of the starting time and ending time, respectively:

$$\{\Delta\hat{s}, \Delta\hat{e}\} = \text{Conv1d}(\text{ReLu}(\text{Conv1d}(\{F_s, F_e\}))) \quad (10)$$

The boundary offsets $\Delta\hat{s}$ and $\Delta\hat{e}$ obtained by this method only utilize the starting and ending local features of proposals. It can effectively reduce noise interference and is more sensitive to the boundary position.

However, only using the local features of the boundary will lose the global context of proposals. Therefore, F_s, F_c and F_e are utilized to achieve segment-level regression, which jointly regress the center location offsets $\Delta\hat{x}$ and duration offsets $\Delta\hat{w}$ of the proposals:

$$F_a = [F_s, F_c, F_e], \quad (11)$$

$$\{\Delta\hat{x}, \Delta\hat{w}, p_{conf}\} = \text{Conv1d}(\text{ReLu}(\text{Conv1d}(F_a))), \quad (12)$$

By means of $\Delta\hat{s}$, $\Delta\hat{e}$, $\Delta\hat{x}$ and $\Delta\hat{w}$, two new proposals $(\hat{s_1}, \hat{e_1})$ and $(\hat{s_2}, \hat{e_2})$ can be obtained by:

$$\hat{s_1} = s_p - \Delta\hat{s}w_p, \quad \hat{e_1} = e_p - \Delta\hat{e}w_p, \quad (13)$$

$$\hat{x_2} = x_p - \Delta\hat{x}w_p, \quad \hat{w_2} = w_p e^{\Delta\hat{w}}, \quad (14)$$

$$\hat{s_2} = \hat{x_2} - \hat{w_2}/2, \quad \hat{e_2} = \hat{x_2} + \hat{w_2}/2, \quad (15)$$

where $w_p = e_p - s_p$, denotes the length of the proposals. Finally, the two new proposals will be fused as the final proposals prediction of TBR:

$$\hat{s} = \tau\hat{s_1} + (1-\tau)\hat{s_2}, \quad \hat{e} = \tau\hat{e_1} + (1-\tau)\hat{e_2}, \quad (16)$$

where τ is a fusion parameter, we set it to 0.5 empirically.

Progressive Refinement. To achieve more accurate boundaries of candidate proposals, a progressive refinement strategy is adopted to generate high-quality proposals from coarse to fine. In ablation experiments, we will explore the impact of the number of TBRs on performance.

Discussion. The boundary features based frame-level regression is sensitive to the local changes, which are helpful to detect the action boundaries caused by shot switching. And the internal proposal features based segment-level regression has an overall perception of proposals suitable for detecting actions with indistinct boundaries. Therefore, these two features are complementary and essential for generalized action proposal generation. MGG [21] proposes a multi-granularity generator to integrate boundary-based regression and anchor-based regression into a unified network with a shared backbone used for feature extraction. However, these two regression streams are trained independently, and the results are fused directly during inference. On the contrary, the main idea of TBR is to adopt both boundary and internal context of proposals to predict the frame-level and segment-level offsets, respectively, but jointly train these two granularities with supervision performed on the combined results for gradient backpropagating. Meanwhile, complementary and progressive boundary refinements are conducted for better performance.

4. Training and Inference of TCANet

4.1. Training

Proposals Selection. To better demonstrate the effectiveness of TCANet, we adopt the proposals output from the most competitive method(BMN) [18] as our pool of candidates. In the training phase of TCANet, to make the network learning more efficient, Soft-NMS [1] is adopted to preprocess proposals output by BMN to reduce the redundant samples. Then the top 100 proposals are selected in descending order for training.

Label Assignment. During the TBR training process, proposals with ground truth temporal Intersection-over-Union(tIoU) greater than a certain threshold k_p are specified as positive samples, and ground truth tIoU less than a certain threshold k_n as negative samples, and those between k_n and k_p Proposals are incomplete samples. The number of positive samples, negative samples and incomplete samples are defined as N_{pos}, N_{incomp} and N_{neg}, respectively. Three kinds of samples are randomly sampled so that $N_{pos} : N_{incomp} : N_{neg} = 1 : 1 : 1$ in training.

Loss Function. The loss functions of the IoU prediction and the position regression of proposals are denoted as L_{iou} and L_{reg}, respectively. We denote L_{iou} and L_{reg} as:

$$L_{iou} = \frac{1}{N_{train}} \left(\sum_{i=1}^{N_{train}} \text{SmoothL1}(p_{conf,i}, g_{iou,i}) \right), \quad (17)$$

$$L_{reg} = \frac{1}{N_{pos}} \left(\sum_{i \in Pos}^{N_{pos}} \sum_{\substack{m \in \{x,w,\\ s,e\}}} \text{SmoothL1}(r_i^m - g_i^m) \right), \quad (18)$$

where

$$N_{train} = N_{pos} + N_{incomp} + N_{neg},$$
$$g_i^x = \Delta x_i, g_i^w = \Delta w_i, r_i^x = \Delta\hat{x}_i, r_i^w = \Delta\hat{w}_i, \quad (19)$$
$$g_i^s = \Delta s_i, g_i^e = \Delta e_i, r_i^s = \Delta\hat{s}_i, r_i^e = \Delta\hat{e}_i$$

The final objective function is written as:

$$Loss = L_{iou} + \lambda L_{reg}, \quad (20)$$

where λ is a balance parameter, we set it to 1.0 empirically.

Table 1. Comparison with state-of-the-art methods on HACS. The results are measured by mAP(%) at different tIoU thresholds and average mAP(%). * indicates our implementation.

Method	0.5	0.75	0.95	Average
2019-Winner [44]	-	-	-	23.49
BMN [18]*	52.49	36.38	10.37	35.76
TCANet[SW]	54.14	37.24	11.32	36.79
TCANet[BMN]	**56.74**	**41.14**	**12.15**	**39.77**

Table 2. Comparison between our TCANet with other state-of-the-arts methods on ActivityNet-v1.3. The results are measured by mAP(%) at different tIoU thresholds and average mAP(%). For fair comparisons, we combined our proposals with video-level classification results from [41]. * indicates the reproduced results.

Method	0.5	0.75	0.95	Average
BSN [20](2stream)	46.45	29.96	8.02	30.03
BMN [18] (2stream)	50.07	34.78	8.29	33.85
G-TAD [42] (2stream)	50.36	34.60	9.02	34.09
BSN++ [27] (2stream)	51.27	35.70	8.33	34.88
BMN [18] (SlowFast)*	52.24	35.89	8.33	35.28
PGCN [43][BSN] (I3D)	48.26	33.16	3.27	31.33
TCANet[BSN] (2stream)	51.91	34.92	7.46	34.43
TCANet[BMN] (2stream)	**52.27**	**36.73**	**6.86**	**35.52**
TCANet[BMN] (SlowFast)	**54.33**	**39.13**	**8.41**	**37.56**

4.2. Inference

In inference, proposals utilized by TCANet should have a high recall rate. Therefore, proposals output by BMN [18] are directly adopted as the input of TCANet. The final confidence of proposals are obtained by fusing the BMN score and TCANet score:

$$S_{proposal} = S_{BMN} * S_{TCANet} \tag{21}$$

Finally, Soft-NMS [1] is employed to remove redundant proposals.

5. Experiments

5.1. Datasets and Setup

HACS [45] is a large-scale dataset for temporal action detection. It contains 37.6k training, 6k validation, and 6k testing videos with 200 action categories.

ActivityNet-v1.3 [2] is a popular benchmark for temporal action detection. It contains 10k training, 5k validation, and 5k testing videos with 200 action categories.

THUMOS14 [14] contains 200 validation videos and 213 testing videos, including 20 action categories. In our experiments, we compare TCANet with the state-of-the-art methods on all three datasets and performed ablation studies on HACS dataset.

Evaluation Metrics. Average Recall (AR) is the average recall rate under specified tIoU thresholds for measuring the quality of proposals. On HACS and ActivityNet-v1.3, these thresholds are set to [0.5:0.05:0.95]. On THUMOS14, they are set to [0.5:0.05:1.0]. By limiting the average number (AN) of proposals for each video , we can calculate the area under the AR vs AN curve to obtain AUC. On ActivityNet-v1.3, AN is set from 1 to 100. The quality of temporal action detection requires to evaluate mean Average Precision(mAP) under multiple tIoU. On HACS and ActivityNet-v1.3, the tIoU thresholds are set to $\{0.5,0.75,0.95\}$, and we also test the average mAP of tIoU thresholds between 0.5 and 0.95 with step of 0.05. On THUMOS14, these tIoU thresholds are set to $\{0.3,0.4,0.5,0.6,0.7\}$.

Table 3. Comparison between our TCANet with other state-of-the-art methods on THUMOS14 dataset. The results are measured by mAP(%) at different tIoU thresholds. We combined our proposals with video-level classifier UntrimmedNet [34].

Method	Classifier	0.7	0.6	0.5	0.4	0.3
TURN [9]	UNet	6.3	14.1	24.5	35.3	46.3
BSN [20]	UNet	20.0	28.4	36.9	45.0	53.5
MGG [21]	UNet	21.3	29.5	37.4	46.8	53.9
BMN [18]	UNet	20.5	29.7	38.8	47.4	56.0
G-TAD [42]	UNet	23.4	30.8	40.2	47.6	54.5
BSN++ [27]	UNet	22.8	31.9	41.3	49.5	59.9
TCANet	UNet	**26.7**	**36.8**	**44.6**	**53.2**	**60.6**

Table 4. Comparison of our TCANet with other state-of-the-art methods on THUMOS14 dataset in terms of AR@AN.

Feature	Method	@50	@100	@200	@500	@1000
2stream	TAG [46]	18.55	29.00	39.61	-	-
2stream	CTAP [7]	32.49	42.61	51.97	-	-
2stream	BSN [20]	37.46	46.06	53.23	61.35	65.10
2stream	MGG [21]	39.93	47.75	54.65	61.36	64.06
2stream	BMN [18]	39.36	47.72	54.84	62.19	65.49
2stream	BSN++ [27]	**42.44**	49.84	**57.61**	**65.17**	66.83
2stream	TCANet	42.05	**50.48**	57.13	63.61	**66.88**

Implementation Details. On HACS and ActivityNet-v1.3, SlowFast [5] is adopted to extract a 2304-dimensional feature vector for each snippet. Each snippet contains $s = 32$ frames and snippet interval δ is 8. For a fair comparison, 2stream network [25] is adopted for feature encoding following [20, 18] on ActivityNet-v1.3 and THUMOS14.

To reduce information loss, the lengths of the input feature sequence are not down-resized; hence each input sequence is fixed to 1000 and 1500 by zero-padding on HACS and ActivityNet-v1.3 for batch training, respectively. The Number of groups N and A in LGTE are empirically set to 8 and 4. The learning rates on these two datasets are set to 0.0004 and 0.001, and the batch size is both 16 for 10 epochs. For THUMOS14 training, a sliding window with a size of 256 is adopted. We set the learning rate, batch size, and epoch number to 0.0004, 16 and 5, respectively.

5.2. Comparison with State-of-the-art Results

This section will compare with the existing state-of-the-art methods on HACS, ActivityNet-v1.3, and THUMOS14.

Table 5. Comparison between our TCANet with other state-of-the-art methods CTAP [7], BSN [20], MGG [21], BMN [18] on ActivityNet-v1.3 in terms of AR@AN and AUC.

Method	CTAP	BSN	MGG	BMN	TCANet
AR@1(val)	-	32.17	-	-	**34.55**
AR@100(val)	73.17	74.16	74.54	75.01	**76.08**
AUC(val)	65.72	66.17	66.43	67.10	**68.08**

Table 6. Ablation study of TBR, LGTE and progressive refinement strategy on HACS dataset in terms of average mAP(%).

TBR1	TBR2	TBR3	LGTE	Average
				35.76
✓				37.16
✓	✓			37.45
✓	✓	✓		37.78
✓	✓	✓	✓	**38.71**

Table 7. The effect of different *window size* settings in the local dependency matrix of LGTE on HACS in terms of average mAP(%).

$WindowSize$	0.5	0.75	0.95	Average
5	55.27	39.54	11.61	38.41
9	**55.60**	40.01	11.47	**38.71**
15	55.56	39.83	11.89	38.70
25	54.99	**40.06**	**11.94**	38.67
T(GTE only)	54.70	39.64	11.71	38.37

Table 8. The effect of different groups N LGTE on HACS dataset in terms of average mAP(%).

N	0.5	0.75	0.95	Average
2	55.27	39.86	10.91	38.49
4	55.13	39.65	11.28	38.38
8	**55.60**	**40.01**	11.47	**38.71**
16	54.87	40.00	**11.63**	38.56

Table 9. The effect of the number of LGTE on HACS dataset in terms of average mAP(%).

Number of LGTE	0.5	0.75	0.95	Average
0	54.73	39.05	10.72	37.78
1	55.13	39.67	11.42	38.31
2	55.60	40.01	11.47	38.71
4	**55.72**	**40.03**	**11.73**	**38.85**
6	55.65	40.02	11.73	38.80

Table 10. The generalizability of LGTE under different frameworks on HACS dataset in terms of average mAP(%).

Framework	LGTE	0.5	0.75	0.95	Average
BMN	✗	52.49	36.38	10.37	35.76
BMN	✓	54.75	38.72	11.41	37.76
TCANet	✗	54.73	39.05	10.72	37.78
TCANet	✓	**55.60**	**40.01**	**11.47**	**38.71**

HACS. On HACS, TCANet is compared with the existing methods in Table 1 on the validation set. TCANet using only a single model significantly surpass the existing methods. Compared with the benchmark method BMN, TCANet's mAP is improved by 4%.

ActivityNet-v1.3. Table 2 and Table 5 compare TCANet with other methods, where TCANet significantly improve both the temporal action proposal and detection performance. For a fair comparison, TCANet is conducted on the 2stream features for experiments. Under the same settings, TCANet can also obtain 1.67% mAP improvement compared with BMN and significantly outperform other existing methods.

THUMOS14. We compare TCANet with the state-of-the-art methods on THUMOS14 in Table 3 and Table 4. Since that our TCANet improves the Average Recall with the first several proposals, the detection performance are more improved than the recall rate. Especially, in Table 3, when tIou=0.6, TCANet is 4.9% higher than BSN++ [27].

5.3. Ablation Study

In this section, we conduct ablation studies on HACS to verify the effectiveness of each module in TCANet.

Is progressive refinement necessary? The progressive refinement strategy is a part of our TCANet. Here, the necessity of progressive refinement is illustrated by the separation experiment of three TBRs in Table 6. Although each TBR has a positive effect on performance, with more stages, this promotion is gradually weakened. Thus TCANet only contains three stages.

What *WindowSize* and groups in LGTE should be set? In Table 7 and Table 8, we conduct experiments to explore the effect of *WindowSize*. If the *WindowSize* is set extremely small(*WindowSize* = 5 or smaller), the local groups' features fail to collect enough local details. On the contrary(*WindowSize*=T), they will introduce excessive global noise. The number of groups N determines whether various temporal relationships can be modeled. Considering the performance, we finally set the *WindowSize* to 9 and the groups to 8 in our experiments.

What is the effect of the number of LGTE? As an easy-plug-in module, performance can be improved by stacking multiple LGTEs. Table 9 demonstrates the performance of the TCANet improves significantly with the increase of LGTE. However, excessive LGTE will lead to over-fitting. The performance of TCANet can reach the best with four LGTEs. Nevertheless, two LGTEs are employed in other ablation studies to facilitate the experiments.

Is LGTE general? To validate the generalizability of our proposed LGTE, we also add it to the BMN [18] framework. The experimental results are shown in Table 10, which reveals that LGTE can also significantly improve the performance of BMN and demonstrate the importance of temporal relationship modeling for temporal action localization task.

How does the frame-level regression affect the TBR? To verify the effect of frame-level regression in TBR, we

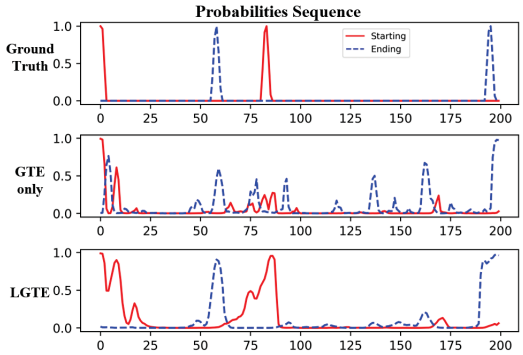

Figure 5. The starting and ending probability sequences generated by LGTE and GTE.

Table 11. The effect of Frame-Level Regression on HACS in terms of mAP(%). SLR and FLR indicate Segment-Level Regression and Frame-Level Regression, respectively.

SLR	FLR	0.5	0.75	0.95	Average
✓	✗	54.58	39.24	**11.72**	38.22
✗	✓	55.02	39.61	9.14	37.92
✓	✓	**55.60**	**40.01**	11.47	**38.71**

conducted experiments using both regression methods separately, as shown in Table 11. If only frame-level regression is applied, the detection performance will drop with only boundary local information. The two methods are combined to boost performance in the final average mAP.

Why mAP not AUC? In our experiments, we find that the detection metric (mAP) mainly depends on Average Recall (AR) with the first several proposals, while the proposal metric (AUC) depends on the first 100 proposals. Hence AR with a small number of proposals has a higher weight in the evaluation metric of detection performance. Extensive experiments have shown that our TCANet can generate fewer proposals with high recall than other methods. Thus the performance improvement of the detection metric is obvious than the proposal metric.

Efficiency Analysis. The input candidate proposals for TCANet need to ensure a high recall rate. Taking the BMN-generated proposals as an example, when 2000 candidate proposals are selected, the recall rate can reach 91% with tIOU=0.5. Our test results are shown in Table 12. For a video, LGTE only needs to encode video features once, and the TBR can process multiple proposals in parallel. Therefore, TCANet only takes 20.9 ms to handle a 9-minute video with 2000 candidate proposals. Compared with BMN, the time consumed by TCANet is only 10%.

5.4. Visualization

To further explore the interpretability of LGTE, GTE only and LGTE are both leveraged to embed the input video features. To facilitate observation, the obtained features are utilized to predict the starting and ending probability sequences. An example is shown in Figure 5. It is observed that the boundary obtained by LGTE is more accurate and smoother than that obtained by GTE. This indicates that LGTE can reduce global noise and enhance the boundary awareness. Figure 6 shows the output of TBR and TCANet. In the top row, both the segment-level and the frame-level output can improve the candidate proposals, but the boundaries are not accurate. The fusion of these two outputs can make the proposal closer to the ground truth. The bottom row shows that our TCANet can generate the proposals from coarse to fine, and provide more reliable confidence scores, especially for short-term action instances.

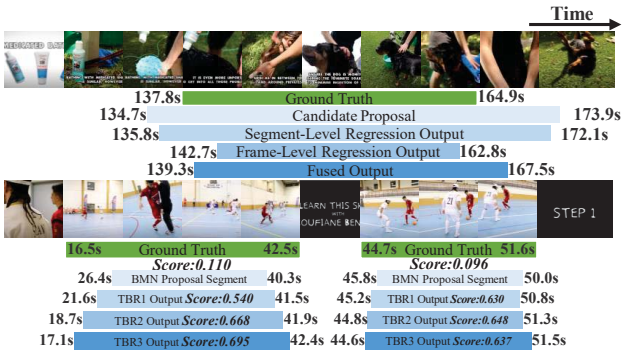

Figure 6. Qualitative examples of proposals generated by TBR(top) and TCANet(bottom) on HACS dataset.

Table 12. The inference time of each module in TCANet on HACS dataset. 2000 candidate proposals were utilized as input to TCANet, and a Nvidia 1080Ti graphic card was employed to process a video for about 9 minutes.

Num×Module	1×BMN	2×LGTE	3×TBR	Total
Num×Time Cost	1×181ms	2×1.6ms	3×5.9ms	201.9ms

6. Conclusion

In this paper, we propose a novel Temporal Context Aggregation Network (TCANet) for temporal action proposal generation. Firstly we introduce the Local-Global Temporal Encoder (LGTE) to capture both *local and global* temporal relationships simultaneously in a channel grouping fashion. Then the complementary boundary regression mechanism is designed to obtain more precise boundaries and confidence scores. Extensive experiments conducted on several famous benchmarks demonstrate that our TCANet can achieve significant improvement on both action proposal and action detection performance.

7. Acknowledgment

This work is supported by the National Natural Science Foundation of China under grant 61871435 and the Fundamental Research Funds for the Central Universities no. 2019kfyXKJC024.

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 5, 6

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. 1, 6

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6299–6308, 2017. 2

[4] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1130–1139, 2018. 2, 4

[5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019. 1, 2, 3, 6

[6] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016. 2

[7] Jiyang Gao, Kan Chen, and Ram Nevatia. Ctap: Complementary temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 68–83, 2018. 2, 4, 6, 7

[8] Jialin Gao, Zhixiang Shi, Guanshuo Wang, Jiani Li, Yufeng Yuan, Shiming Ge, and Xi Zhou. Accurate temporal action proposal generation with relation-aware pyramid network. In *AAAI*, pages 10810–10817, 2020. 2, 3, 4

[9] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Proceedings of the IEEE international conference on computer vision*, pages 3628–3636, 2017. 2, 4, 6

[10] Kirill Gavrilyuk, Ryan Sanford, Mehrsan Javan, and Cees GM Snoek. Actor-transformers for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 839–848, 2020. 3

[11] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019. 3

[12] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *Advances in Neural Information Processing Systems*, pages 34–45, 2017. 3

[13] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2000–2009, 2019. 2

[14] YG Jiang, Jingen Liu, A Roshan Zamir, G Toderici, I Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes. In *Computer Vision-ECCV workshop 2014*, 2014. 1, 6

[15] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2020. 2

[16] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *AAAI*, pages 11499–11506, 2020. 1, 2

[17] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7083–7093, 2019. 2

[18] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3898, 2019. 1, 2, 4, 5, 6, 7

[19] Tianwei Lin, Xu Zhao, and Zheng Shou. Single shot temporal action detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 988–996, 2017. 4

[20] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 1, 2, 4, 6, 7

[21] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3604–3613, 2019. 2, 4, 5, 6, 7

[22] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *Int. Conf. Comput. Vis.*, pages 5533–5541, 2017. 2

[23] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *arXiv preprint arXiv:1511.04119*, 2015. 3

[24] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1049–1058, 2016. 2, 4

[25] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 2, 3, 6

[26] Gurkirt Singh and Fabio Cuzzolin. Untrimmed video classification for activity detection: submission to activitynet challenge. *arXiv preprint arXiv:1607.01979*, 2016. 1

[27] Haisheng Su, Weihao Gan, Wei Wu, Junjie Yan, and Yu Qiao. Bsn++: Complementary boundary regressor with

scale-balanced relation modeling for temporal action proposal generation. *arXiv preprint arXiv:2009.07641*, 2020. 2, 6, 7

[28] Haisheng Su, Jing Su, Dongliang Wang, Weihao Gan, Wei Wu, Mengmeng Wang, Junjie Yan, and Yu Qiao. Collaborative distillation in the parameter and spectrum domains for video action recognition. *arXiv preprint arXiv:2009.06902*, 2020. 2

[29] Haisheng Su, Xu Zhao, and Tianwei Lin. Cascaded pyramid mining network for weakly supervised temporal action localization. In *Asian Conference on Computer Vision*, pages 558–574. Springer, 2018. 1

[30] Haisheng Su, Xu Zhao, Tianwei Lin, Shuming Liu, and Zhilan Hu. Transferable knowledge-based multi-granularity fusion network for weakly supervised temporal action detection. *IEEE Transactions on Multimedia*, 2020. 1

[31] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Int. Conf. Comput. Vis.*, pages 4489–4497, 2015. 2

[32] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6450–6459, 2018. 2

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3, 4

[34] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4325–4334, 2017. 6

[35] Limin Wang, Yuanjun Xiong, Zhe Wang, and Yu Qiao. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*, 2015. 1, 2

[36] Limin Wang, Yuanjun Xiong, Zhe Wang, and Yu Qiao. Towards good practices for very deep two-stream convnets. *arXiv preprint arXiv:1507.02159*, 2015. 2

[37] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 1

[38] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 3

[39] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 284–293, 2019. 3

[40] S Xie, C Sun, J Huang, Z Tu, and K Murphy. Rethinking spatiotemporal feature learning for video understanding (2017). arxiv preprint. *arXiv preprint arXiv:1712.04851*. 2

[41] Yuanjun Xiong, Limin Wang, Zhe Wang, Bowen Zhang, Hang Song, Wei Li, Dahua Lin, Yu Qiao, Luc Van Gool, and Xiaoou Tang. Cuhk & ethz & siat submission to activitynet challenge 2016. *arXiv preprint arXiv:1608.00797*, 2016. 2, 6

[42] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020. 6

[43] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7094–7103, 2019. 6

[44] Songyang Zhang, Houwen Peng, Le Yang, Jianlong Fu, and Jiebo Luo. Learning sparse 2d temporal adjacent networks for temporal action localization. *arXiv preprint arXiv:1912.03612*, 2019. 6

[45] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8668–8678, 2019. 1, 6

[46] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017. 6