Contextual Proposal Network for Action Localization

He-Yen Hsieh, Ding-Jie Chen and Tyng-Luh Liu
Institute of Information Science, Academia Sinica, Taiwan

heyen@iis.sinica.edu.tw, djchen.tw@gmail.com, liutyng@iis.sinica.edu.tw

Abstract

This paper investigates the problem of Temporal Action Proposal (TAP) generation, which aims to provide a set of high-quality video segments that potentially contain actions events locating in long untrimmed videos. Based on the goal to distill available contextual information, we introduce a Contextual Proposal Network (CPN) composing of two context-aware mechanisms. The first mechanism, i.e., feature enhancing, *integrates the inception-like module with long-range attention to capture the multi-scale temporal contexts for yielding a robust video segment representation. The second mechanism,* i.e., boundary scoring, *employs the bi-directional recurrent neural networks (RNN) to capture bi-directional temporal contexts that explicitly model actionness, background, and confidence of proposals. While generating and scoring proposals, such bi-directional temporal contexts are helpful to retrieve high-quality proposals of low false positives for covering the video action instances. We conduct experiments on two challenging datasets of ActivityNet-1.3 and THUMOS-14 to demonstrate the effectiveness of the proposed Contextual Proposal Network (CPN). In particular, our method respectively surpasses state-of-the-art TAP methods by 1.54% AUC on ActivityNet-1.3 test split and by 0.61% AR@200 on THUMOS-14 dataset.*

1. Introduction

The research of video content analysis is encouraged by the rapid growth of video sequences derived from the fast development of digital cameras and online video services. The related topics include temporal action detection [16, 51], video summarization [48, 49], video captioning [8, 9], video grounding [7], and visual question answering [1, 17]. Among these topics, the temporal action detection task, which aims to detect the human-action instances within the untrimmed long video sequences, especially plays a pivotal role in several video content analysis methods. Akin to the image object detection task, the temporal action detection can be separated into a TAP generation stage and an action classification stage.

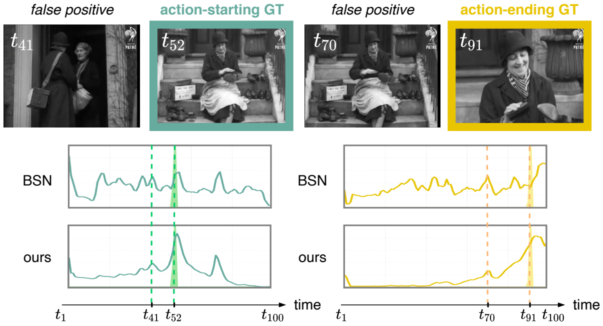

Figure 1: The proposed CPN has low false positive boundary predictions. The top four images are centered frames of 41st, 52nd, 70th, and 91st snippets on the video of *Polishing Shoes* action. The bottom four charts are boundary predictions. Notice that the false positive images are derived from local peaks of the boundary predictions, and our method significantly reduces such phenomenon.

Recent studies [4, 10, 24, 25, 26, 28] demonstrate that pursuing the TAP quality clearly improves the performance of such two-stage temporal action detectors. Since a temporal action proposal generator is demanded to use fewer proposals for capturing the ground-truth action instances in a high recall rate, a high-quality TAP generator consequently reduces the burden of the subsequent action classification stage. In this paper, we introduce an effective temporal action proposal generator, which aims to provide the set of video segments, *i.e.*, action proposals, that precisely and exhaustively cover the human-action instances. Inspired by the temporal boundary prediction mechanism [26], which forms the potential proposals by estimating their boundary probabilities, our method similarly predicts boundary probability over the video sequence to discriminate potential action proposals.

Our boundary-based TAP generator comprises context-aware mechanisms for *feature enhancing* and *boundary scoring* to carry out the high-quality temporal action proposal generation. In video processing, it is common to represent each video sequence into a set of consecutive video segments, where each segment is called a *snippet*, so that the computational cost can be greatly reduced. Then, most of the existing video representations [5, 11, 31, 33, 38, 39, 44] separately encode each snippet to form snippet-level representation, which lacks the correlation between the neighboring snippets. Our feature enhancing mechanism employs the inception-like module [35, 36] to concern the multi-scale temporal contexts for correlating the neighboring snippets. Such multi-scale temporal contexts help to smooth the subsequent snippet-level prediction by considering their neighbors. Before capturing the multi-scale temporal contexts, our feature enhancing mechanism also employs long-range attention [20, 21, 40] to adjust the snippet-level representation.

The boundary prediction is usually estimated on the snippet level. Rather than directly estimating the boundary probability as [26], our boundary scoring mechanism employs the recurrent neural networks for the actionness estimation and the additional background (existing no actions) estimation in a bi-directional temporal manner to co-estimate the action boundaries. This co-estimation derived from the observation that the features for describing the long-time actionness/background are more consistent along the temporal dimension than the short-time. The experiments show that the feature enhancing mechanism enables the CPN model to obtain an effective and robust feature representation, and the boundary scoring mechanism allows the CPN model to estimate the proposal boundaries of much less false positive, as shown in Figure 1. The overview of our temporal action proposal model, *i.e.*, CPN, is shown in Figure 2.

In a nutshell, our contributions are summarized below.

1. We introduce the *feature enhancing* mechanism to generate robust video representation concerning multi-scale temporal contexts and long-range attention.

2. We introduce the *boundary scoring* mechanism to predict the low false positive action boundaries via bi-directional recurrent neural networks. To our best knowledge, this is the first work attempting to formulate boundary-sensitive predictions through bi-directional temporal contexts that explicitly leverage actionness and background. The ablation study supports the benefit of such a formulation.

3. The extensive experiments demonstrate that the proposed CPN model achieves state-of-the-art performance on generating the temporal action proposals.

2. Related work

This section briefly reviews the related literature on video feature representation, attention mechanism, and temporal action proposal generation.

Feature representation. As a de facto trend, instead of using the handcrafted features, the neural-network-based features are widely employed for addressing the action classification task. These popular neural network approaches include the two-stream networks [11, 33, 39], which separately represent the appearance feature and the motion feature, and 3D networks [5, 31, 38, 44], which directly represent a video as the spatio-temporal feature. In this paper, we use the action recognition model [39, 43] to extract snippet-level features for representing each untrimmed video.

Attention mechanism. The attention mechanism is the process of selectively focusing on a few relevant things in comparison with everything. Fields like natural language processing and computer vision, broadly leverage such an attention mechanism. For instances, Bahdanau *et al.* [2] enable their model to focus on searching a group of related words from the input sentences for predicting the target words, Xu *et al.* [45] introduce the soft and hard attention to generate image captions, and LFB [42] introduces long-term feature banks to analyze videos. In our CPN model, our feature enhancing mechanism employs co-attention & co-excitation [20] and SENet [21] to capture long-range dependencies over temporal dimension and channel dimension, respectively, extending the merits of these efforts for constructing a robust video representation.

Temporal action proposal generation. We categorize the TAP generation methods into *Anchor-based* [14, 19, 32] and *boundary-based* [24, 25, 26, 53, 3, 52, 34]. The former focuses on designing several multi-scale anchor boxes to cover action instances, while the latter estimates the temporal location probabilities of the action instances. Besides, some methods [12, 28, 13] also explore the way to integrate the above-mentioned two categories for precisely localizing the temporal boundaries. In anchor-based methods, the S-CNN [32] and Heilbron *et al.* [19] respectively evaluate anchors via C3D network and sparse learning, and TURN [14] suggests regressing the temporal boundaries of action instances. The boundary-based work, TAG [53], generates action proposals via a temporal watershed algorithm to merge contiguous temporal locations of high actionness probabilities. BSN [26] generates proposals as well as their confidence by formulating the probabilities of boundaries and actionness. BMN [25] proposes a boundary-matching mechanism to evaluate the confidence among densely distributed

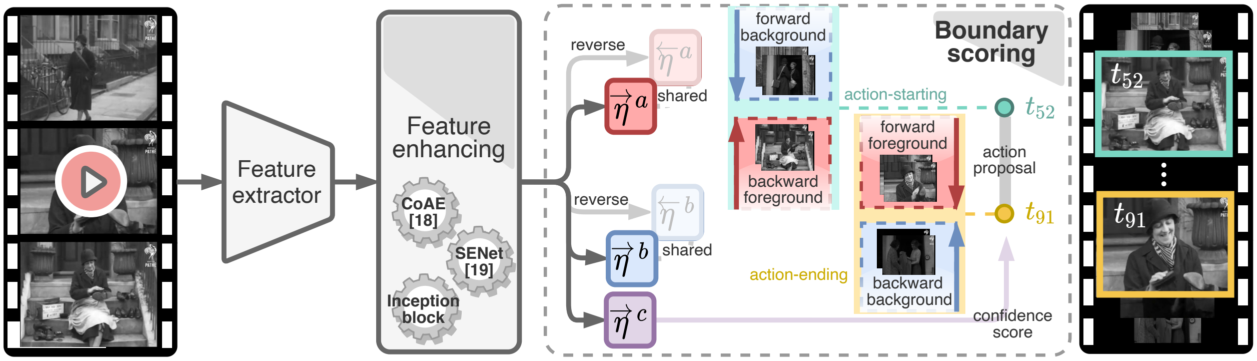

Figure 2: Contextual Proposal Network (CPN) for generating temporal action proposals. Our TAP generator CPN comprises two context-aware mechanisms for feature enhancing and boundary scoring. The feature enhancing mechanism employs CoAE, SENet, and Inception block to capture the multi-scale temporal contexts. The boundary scoring mechanism employs the recurrent neural networks for the actionness estimation and the background estimation in a bi-directional temporal manner to co-estimate the low false-positive action boundaries.

proposals. DBG [24] uses the maps of dense boundary confidence and completeness to further score boundaries for all action proposals. BC-GNN [3] relates proposal boundaries and proposal content as nodes and edges through a Graph Neural Network. Zhao *et al*. [52] introduce two regularization terms, namely intra-phase consistency and inter-phase consistency, to explore the relationship among temporal locations and interrelations of multiple probability sequences, respectively. BSN++ [34] enrich context information with a U-shape structure and refine action boundaries via a bi-directional boundary matching scheme.

In sum, the anchor-based methods focus on the anchor-box design and usually lack the flexible temporal boundaries for aligning to various action instances. The boundary-based methods provide flexible temporal boundaries, yet with noticeable false positive boundary predictions. By contrast, we introduce an RNN-based *boundary scoring* mechanism to correlate the snippet-level features over the temporal dimension for estimating the long-time actionness/background boundaries. Combining with the *feature enhancing* mechanism, we show that our TAP model outperforms the existing leading methods.

3. Contextual proposal network

We first formulate the task of temporal action proposal generation and then detail our method to address it. Figure 2 shows the architecture of our contextual proposal network.

TAP generation. Given a l_v frames video sequence $\mathcal{X} = \{x_n\}_{n=1}^{l_v}$ comprising N_g actions of interest. The TAP task aims at generating a proper set of video segments as action proposals that can be used to detect the underlying human actions in $\mathcal{X}$. We denote an action proposal as $\varphi = (t_s, t_e)$, where t_s and t_e denotes the starting and ending frame of $\mathcal{X}$. Analogous to the object proposals for detection, action proposals are generic and class agnostic. Let the N_g ground-truth actions of $\mathcal{X}$ be $\Psi_g = \{\varphi_n = (\hat{t}_s^n, \hat{t}_e^n)\}_{n=1}^{N_g}$. An action proposal (t_s, t_e) is said to be matched to some ground-truth action $(\hat{t}_s, \hat{t}_e)$ if their time-interval IoU (in terms of frames) is greater than a specified threshold τ. Considering a proposal set Ψ_p of N_p proposals, *i.e*., $\Psi_p = \{\varphi_n = (t_s^n, t_e^n)\}_{n=1}^{N_p}$. The goodness of Ψ_p w.r.t. $\mathcal{X}$ can be explicitly measured by the number of matched action proposals.

3.1. Feature enhancing

To make each video a snippet-level representation, we decompose each video sequence $\mathcal{X}$ into a T-snippet set, denoted as $\mathcal{V} = \{\mathbf{v}_t\}_{t=1}^{T}$. Our approach represents each snippet $\mathbf{v}$ with two-stream features, which are often used to analyze video-related tasks, *e.g*., [25, 39, 53], and comprise one 200-D appearance vector and one 200-D motion vector. To account for videos of various lengths, we follow BSN [26] to sample single-stream features over the temporal dimension to consistently obtain T snippets per video sequence. Precisely, each video sequence $\mathcal{X}$ is represented by an appearance feature tensor $A \in \mathbb{R}^{C\times T}$ and a motion feature tensor $M \in \mathbb{R}^{C\times T}$, where the channel number $C = 200$.

Long-range attention. The step of long-range attention aims at adjusting the snippet-level representation over the temporal dimension and channel dimension. We use co-attention & co-excitation operations in CoAE [20] to carry out the feature adjusting over temporal dimension T. The operations yield snippet-level feature correlations of size $T \times T$ to perform feature re-weighting by conditioning on the other feature. The operations enable our model to emphasize the temporal correlations of human-action descriptions between the appearance and motion cues. We use squeeze-and-excitation operations in SENet [21] to carry out the feature adjusting over channel dimension C. The operations yield channel-level scales of size C to re-weight each feature channel and further highlight the important channel of the appearance and motion cues. We refer the readers to [20, 21] or the supplementary material for further details about our long-range attention step.

Inception. This step aims at collecting the multi-scale temporal contexts to correlate the neighboring snippets. We first define a basic convolutional layer ϕ by

$$\phi(X; f, o) = \mathrm{ReLU}(\mathbf{W}X + \mathbf{b}) \,, \tag{1}$$

where X denotes the input feature maps, f specifies the filter size, o gives the number of the output filters, and ReLU is the activation function. $\mathbf{W}$ and $\mathbf{b}$ respectively denote the weights and bias of ϕ. In our inception-like operation, we employ ϕ with two kinds of filter sizes to represent each of the two-stream features and hence retrieve multiple temporal contexts as $A_1 = \phi(A; 1 \times 3, C)$, $A_2 = \phi(A; 1 \times 5, C)$, $M_1 = \phi(M; 1 \times 3, C)$, and $M_2 = \phi(M; 1 \times 5, C)$. We then concatenate all features followed by another convolution layer to unify them as the video's snippet-level feature

$$\mathbf{F} = \phi(A_1 \parallel A_2 \parallel M_1 \parallel M_2; 1 \times 3, C') \in \mathbb{R}^{C' \times T} \,, \tag{2}$$

where the notation $\parallel$ means concatenation over the channel dimension and the channel number $C' = 400$.

3.2. Boundary scoring

Our *boundary scoring* mechanism employs multiple RNNs to calculate the various snippet-level probabilities and then associates these probabilities for discriminating the potential action-instance proposals. In experiments, we show that the high-quality proposals can be retrieved by scoring the potential proposals based on these probabilities. The main idea of our RNN-based boundary scoring is to employ the recurrent neural networks as the temporal context collector, which accumulates information from various temporal video segments among snippets. In this way, we could estimate the snippet-level boundary probabilities concerning flexible temporal duration. The right part of Figure 2 sketches the boundary scoring mechanism.

Context collection. To begin with, we express an RNN as function $\overrightarrow{\eta}^r$, where the arrow $\rightarrow$ above η denotes the forward ($\leftarrow$ for backward) sequential order while collecting temporal context, and the superscript $r \in \{a, b, c\}$ is to specify that the RNN is used to encode the video segment features for representing a: actionness, b: background, or c: confidence.

The RNNs can be used to generate all sorts of features for each snippet. For example, considering the snippet $\mathbf{v}_i$, its *snippet-level* forward actionness feature can be generated by the ith hidden state $\overrightarrow{h}^a_{1i}$ of $\overrightarrow{\eta}^a$, and similarly, its backward actionness feature as $\overleftarrow{h}^a_{Ti}$ of $\overleftarrow{\eta}^a$. That is,

$$\overrightarrow{h}^a_{1i} = \overrightarrow{\eta}^a(\mathbf{F}[1:i]) \in \mathbb{R}^{C \times 1} \,, \tag{3}$$

$$\overleftarrow{h}^a_{Ti} = \overleftarrow{\eta}^a(\mathbf{F}[T:i]) \in \mathbb{R}^{C \times 1} \,, \tag{4}$$

where the $\mathbf{F}[i:j]$ denotes the sequential representations from the ith snippet to the jth snippet, the pair subscript of a hidden state indicates the encoding order and the snippet segment of the integrated two-stream features within RNN. Analogously, we can extract the forward and backward background features: $\overrightarrow{h}^b_{1i}$ and $\overleftarrow{h}^b_{Ti}$ using the RNNs $\overrightarrow{\eta}^b$ and $\overleftarrow{\eta}^b$, respectively.

With an additional convolution layer, each of the above four kinds of RNN hidden states can be designed and learned to predict the probability of actionness or background. Take, for example, the forward actionness probabilities $\overrightarrow{p}^a$ and our formulation yields

$$\overrightarrow{p}^a = \phi(\overrightarrow{\mathbf{h}}^a; 1 \times 3, 1) \in \mathbb{R}^{1 \times T} \,, \tag{5}$$

where $\overrightarrow{\mathbf{h}}^a \in \mathbb{R}^{C \times T}$ includes all the hidden states of running $\overrightarrow{\eta}^a$ over the sequence of T snippets, and ϕ denotes a convolutional layer as in (1) yet uses the sigmoid activation for probability output.

Finally, the confidence RNN $\overrightarrow{\eta}^c$ aims to tackle the *proposal-level* features. Given a proposal starting from the ith snippet to the jth snippet, $\overrightarrow{\eta}^c$ sequentially collects the snippet-level integrated two-stream features ranging from i to j as follows:

$$\overrightarrow{h}^c_{ij} = \overrightarrow{\eta}^c(\mathbf{F}[i:j]) \in \mathbb{R}^{C \times 1} \,. \tag{6}$$

Note that (6) implies that $\overrightarrow{h}^c_{ij}$ adopts the final hidden-state of $\overrightarrow{\eta}^c(\mathbf{F}[i:j])$.

Score calculation. The score calculation employs the context-collected features for scoring each potential proposal. We explore the retrieved hidden-state features for assessing the goodness of each potential proposal via evaluating various probabilities. It is worth mentioning that in predicting the boundaries of an action proposal, most existing techniques use independent snippet-level features; by contrast, our method considers the snippet-level features manipulated by RNN, and thus concerning the other snippet-level features sequentially. As we will demonstrate in the experimental results, the difference would lead to better boundary prediction accuracy. Namely, reducing the predicted false-positive boundaries.

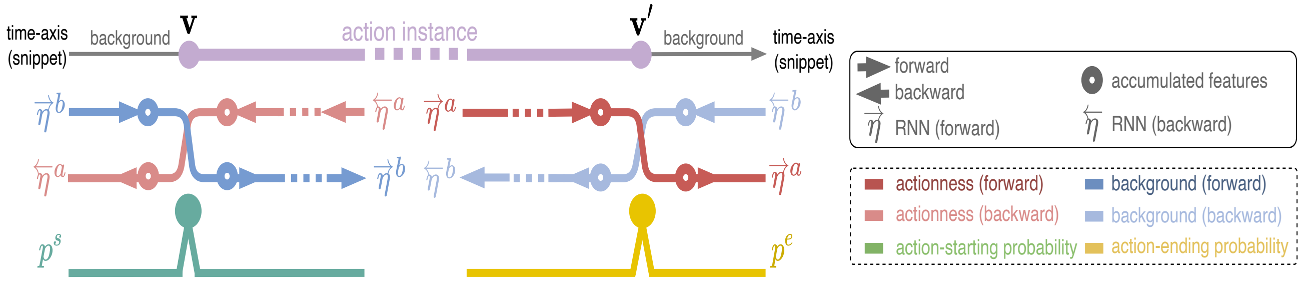

Figure 3: Illustration of our boundary scoring mechanism. In context collection step, $\{\overrightarrow{\eta}^a, \overrightarrow{\eta}^b\}$ and $\{\overleftarrow{\eta}^a, \overleftarrow{\eta}^b\}$ are respectively forward and backward RNNs for actionness and background. The drastic increasing of actionness probability predicted by $\overleftarrow{\eta}^a$ and the decreasing of background probability by $\overrightarrow{\eta}^b$ support a high starting probability p^s at snippet $\mathbf{v}$, and analogously, a high ending probability p^e at snippet $\mathbf{v}'$.

Intuitively, a snippet $\mathbf{v}_i$ could be an action starting boundary if its previous snippet $\mathbf{v}_{i-1}$ is considered as a background, and its subsequent snippet $\mathbf{v}_{i+1}$ is likely as an action instance. On the other hand, a snippet $\mathbf{v}_i$ could be an action ending boundary if $\mathbf{v}_{i-1}$ is also as an action instance, and $\mathbf{v}_{i+1}$ is likely as a background. Bearing these observations in mind, we are now ready to predict the action starting probability p^s and the action ending probability p^e over all snippets with two convolution layers by

$$\begin{aligned}
\mathbf{h}^s &= \phi(\overrightarrow{\mathbf{h}}^b \parallel \overleftarrow{\mathbf{h}}^a; 1\times 3, C) \in \mathbb{R}^{C\times T}, \qquad (7)\\
\mathbf{h}^e &= \phi(\overrightarrow{\mathbf{h}}^a \parallel \overleftarrow{\mathbf{h}}^b; 1\times 3, C) \in \mathbb{R}^{C\times T}, \qquad (8)\\
p^s &= \phi(\mathbf{h}^s; 1\times 1, 1) \in \mathbb{R}^{1\times T}, \qquad (9)\\
p^e &= \phi(\mathbf{h}^e; 1\times 1, 1) \in \mathbb{R}^{1\times T}, \qquad (10)
\end{aligned}$$

where $\overrightarrow{\mathbf{h}}^a, \overrightarrow{\mathbf{h}}^b, \overleftarrow{\mathbf{h}}^a, \overleftarrow{\mathbf{h}}^b \in \mathbb{R}^{C\times T}$ denote all the collected snippet-level hidden states, $\mathbf{h}^s$ and $\mathbf{h}^e$ respectively denote the intermediate results of the action starting hidden states and the action ending hidden states, and the notation $\parallel$ means the concatenation over the channel-dimension. In Figure 3, we illustrate the described RNN-based reasoning about when a snippet instance is likely to be a starting boundary of an action and analogously, the case for being an ending boundary.

Besides the snippet-level probabilities p^s and p^e, we further consider the proposal-level probabilities. Given a proposal starting from the ith snippet to the jth snippet, we predict the confidence probability p^c and boundary-relation probability p^{se} as

$$\begin{aligned}
p^c &= \phi(\overrightarrow{\mathbf{h}}^c; 1\times 3\times 3, 1) \in \mathbb{R}^{1\times T\times T}, \qquad (11)\\
p^{se} &= \phi(\mathbf{h}; 1\times 3\times 3, 1) \in \mathbb{R}^{1\times T\times T}, \qquad (12)
\end{aligned}$$

where each ϕ in (11) and (12) again denotes a convolutional layer with the sigmoid activation, $\overrightarrow{\mathbf{h}}^c \in \mathbb{R}^{C\times T\times T}$ includes all the hidden states for all potential proposals, and $\mathbf{h}$ means all the pairwise concatenation of action starting hidden states $\mathbf{h}^s$ and the action ending hidden states $\mathbf{h}^e$. Note that we calculate the probabilities in (11) and (12) by using the filter of size 3×3 to consider the neighboring proposals for smoothing the predictions.

Finally, given an action proposal (i, j) starting from ith snippet the the jth snippet, we empirically define its score $p_{i,j}$ with the probabilities mentioned above as:

$$p_{i,j} = p_i^s \times p_j^e \times p_{i,j}^{se} \times p_{i,j}^c \qquad (13)$$

In our implementation, we follow BSN [26] to collect the potential proposals using the snippet-level probabilities of action-staring p^s and action-ending p^e. We then score each proposal by (13) followed by the soft non-maximum suppression for retrieving the top-scored proposals.

3.3. Optimization

The overall loss function for training is formulated as a multi-task objective that comprises context collection loss ($\mathcal{L}_{con}$) and scoring calculation loss ($\mathcal{L}_{scr}$):

$$\mathcal{L} = \mathcal{L}_{con} + \lambda\, \mathcal{L}_{scr}\,, \qquad (14)$$

where the weighting factor λ is set to 0.5, $\mathcal{L}_{con}$ is used for training the four probabilities $\overrightarrow{p}^a$, $\overrightarrow{p}^b$, $\overleftarrow{p}^a$,$\overleftarrow{p}^b$, and $\mathcal{L}_{scr}$ is designed for learning the remaining probabilities p^s, p^e, p^c, and p^{se}. The context collection loss $\mathcal{L}_{con}$ encourages each RNN to collect all its hidden states for predicting the probabilities of action-instance or background, and the scoring calculation loss $\mathcal{L}_{scr}$ encourages all RNNs to correlate their collected states for ranking the proposals. Both the loss terms in (14) employ the binary logistic regression loss as:

$$loss = \sum\nolimits_i [y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)]\,, \qquad (15)$$

where i denotes the bin index of each probability as mentioned earlier, y_i and p_i respectively denote the ground-truth probability and the predicted probability.

4. Experiments

Datasets and metrics. We conduct experiments on ActivityNet-1.3 [18] dataset and THUMOS-14 [22] dataset. The ActivityNet-1.3 is a large-scale action understanding dataset, which is available for evaluating the tasks of proposal generation, action recognition, temporal detection, and dense captioning. There are 19,994 temporal annotated untrimmed videos comprising 200 action categories. The THUMOS-14 dataset contains 1,010 validation videos and 1,574 testing videos of 20 action categories. There are 200 validation videos, and 212 testing videos contain temporal action annotations. We use the validation set for training and use the testing set for evaluating. There are two kinds of metrics to evaluate the proposal quality. *i*) AR@AN, which evaluates the relation between Average Recall (AR), that calculated with multiple specified IoU thresholds, and Average Number of proposals (AN). *ii*) AUC, which denotes the area under the AR vs. AN curve. Due to the limited space allowed, more detailed experimental results, including carrying out the temporal action detection task, are provided in the supplementary material.

Implementation details. We use the two-stream features via the two-stream network [43] pre-trained on the training set of ActivityNet-1.3 with the same parameter settings as [14, 26]. To form a snippet, we set 16 frames per snippet in ActivityNet-1.3 and 5 frames per snippet in THUMOS-14. Further, we sample the snippets with $T = 100$ via linear interpolation in ActivityNet-1.3 and with $T = 128$ via truncation and overlapped sliding windows in THUMOS-14. All the snippet manipulations are the same as [25, 26]. Our model employs Gated Recurrent Unit (GRU) to carry out the recurrent association phase. For the setting of soft-NMS, we respectively use threshold 0.8 and 0.65 for ActivityNet-1.3 and THUMOS-14, and the same decay parameter 0.85 for both datasets. We train our model using Adam optimizer with batch size 16 and learning rate 10^{-3} for 10 epochs.

4.1. Comparison with state-of-the-arts

Table 1 summarizes the comparison of our approach against state-of-the-art TAP methods. Our model significantly outperforms other TAP methods on all metrics of both datasets. Specially, we improve AR@100 and AUC of ActivityNet-1.3 validation split by 0.93% and 1.21%, respectively. On the ActivityNet-1.3 testing split, we further improve AUC by 1.54%. On the THUMOS-14 testing splits, our improvements range from 0.61% to 1.4% among AR@200, AR@500, and AR@1000, besides the AR@50 and AR@100. In sum, the comparison demonstrates that our temporal action proposal model achieves state-of-the-art performance.

4.2. Ablation study

We carry out a comprehensive ablation study on the ActivityNet-1.3 validation split to assess the importance of each design component of our model for action localization. Table 2 and Table 3 summarize the results of our ablation study. Be reminded that when investigating the effect of a particular component in our ablation evaluation, the mechanisms based on all the other components of the proposed model are included in the respective implementation.

Feature enhancing. Table 2 compares the components within the feature enhancing mechanism. The first row, *i.e.*, baseline-FE, serves as the baseline that directly concatenates the two-stream features over the channel dimensions for the subsequent proposal generation. The comparisons in Table 2 show that all components, *i.e.*, long-range attention and inception block, contribute positively. Precisely, considering long-range attention over both temporal and channel dimensions improves AUC by 0.91%, and using the inception block improves AUC by 0.96%. The complete feature enhancing mechanism can further improve AUC by 1.96%. The results demonstrate that enhancing the two-stream features with the long-range attention over temporal-channel dimensions and with the inception-like multi-scale temporal contexts collection is beneficial.

Boundary scoring. Table 3 compares the various probabilities within the boundary scoring mechanism. The first row, *i.e.*, baseline-BS, serves as the baseline that directly predicts the probabilities of p^s and p^e from enhanced feature $\mathbf{F}$ without using RNNs. Here we carry out the boundary prediction as BSN. The other rows are RNN-based predictions by contrast. The second row predicts the probabilities of p^s and p^e via RNNs concerning the temporal context of actionness. The third row predicts the same probabilities of p^s and p^e via RNNs concerning the temporal contexts of actionness and background. The result comparing the second row with the third row shows the 1.2% AUC performance gain, demonstrating that the extra background branch in RNNs helps accurate boundary prediction. In the third row, the snippet-level boundary prediction p^s and p^e improve AUC by 2.58% compared with the baseline-BS. The proposal-level predictions of p^{se} and p^c can further respectively improve AUC by 3.25% and 3.21%, and the complete boundary scoring mechanism improves AUC by 3.84%. In sum, the results in Table 3 show that both the predictions on snippet-level, *i.e.*, p^s and p^e, and proposal-level, *i.e.*, p^{se} and p^c, contribute positively. These ablation results demonstrate that it is beneficial to retrieve action proposals concerning the relationship between the mentioned probabilities, and our RNN-based approach fulfills the goal with noticeable performance gain.

Method	Reference	ActivityNet-1.3			THUMOS-14				
		AR@100 (val)	AUC (val)	AUC (test)	AR@50	AR@100	AR@200	AR@500	AR@1000
CTAP [12]	ECCV'18	73.17	65.72	-	32.49	42.61	51.97	-	-
BSN [26]	ECCV'18	74.16	66.17	66.26	37.46	46.06	53.21	60.64	64.52
*GTAN [29]	CVPR'19	74.80	67.10	67.40	-	-	54.30	-	-
MGG [28]	CVPR'19	74.54	66.43	66.47	39.93	47.75	54.65	61.36	64.06
BMN [25]	ICCV'19	75.01	67.10	67.19	39.36	47.72	54.70	62.07	65.49
RapNet [13]	AAAI'20	76.71	67.63	67.72	40.35	48.23	54.92	61.41	64.47
DBG [24]	AAAI'20	76.65	68.23	68.57	37.32	46.67	54.50	62.21	66.40
Zhao's model [52]	ECCV'20	75.27	66.51	-	44.23	50.67	55.74	-	-
BC-GNN [3]	ECCV'20	76.73	68.05	-	40.50	49.60	56.33	62.80	66.57
Gao's model [15]	PR'20	74.49	66.02	66.40	**45.19**	**51.67**	56.45	-	-
BSN++ [34]	AAAI'21	76.52	68.26	-	42.44	49.84	57.61	65.17	66.83
TCANet [30]	CVPR'21	76.08	68.08	-	42.05	50.48	57.13	63.61	66.88
SSTAP [41]	CVPR'21	75.54	67.53	-	41.01	50.12	56.69	-	68.81
BMN + *BSP [46]	ICCV'21	75.50	67.61	-	-	-	-	-	-
RTD-Net [37]	ICCV'21	73.21	65.78	-	41.52	49.32	56.41	62.91	-
CPN		**77.66**	**69.47**	**70.11**	39.90	49.98	**58.22**	**66.47**	**70.21**

Table 1: Comparison of the state-of-the-art methods on ActivityNet-1.3 validation and testing split and on THUMOS-14 testing split. Notation "*" indicates the model using non-two-stream features.

Component			Feature Enhancing (FE)					
Long-range		Inception	AUC	PG	AR@30	AR@50	AR@80	AR@100
baseline-FE			67.51	-	66.54	71.02	74.47	75.86
TD	-	-	67.91	+0.40	67.11	71.40	74.81	76.24
TD	CD	-	68.42	+0.91	67.52	71.92	75.23	76.55
-	-	✓	68.47	+0.96	67.64	72.03	75.57	76.91
TD	CD	✓	**69.47**	+1.96	**68.74**	**73.26**	**76.27**	**77.66**

Table 2: Ablation study of feature enhancing mechanism on ActivityNet-1.3 validation split. The meanings of abbreviations are TD: temporal dimension; CD: channel dimension; PG: performance gain on AUC; baseline-FE: concatenating the two-stream features directly.

Component			Boundary Scoring (BS)					
p^s, p^e	p^{se}	p^c	AUC	PG	AR@30	AR@50	AR@80	AR@100
baseline-BS			65.63	-	64.74	69.13	72.50	73.85
A	-	-	67.01	+1.38	65.94	70.49	74.12	75.70
A+B	-	-	68.21	+2.58	67.58	71.56	74.93	76.28
A+B	✓	-	68.88	+3.25	68.26	72.37	75.68	77.12
A+B	-	✓	68.84	+3.21	68.34	72.31	75.33	76.51
A+B	✓	✓	**69.47**	+3.84	**68.74**	**73.26**	**76.27**	**77.66**

Table 3: Ablation study of boundary scoring mechanism on ActivityNet-1.3 validation split. The meanings of abbreviations are PG: performance gain on AUC; baseline-BS: boundary prediction without using RNNs; A: boundary prediction with the temporal context of actionness; A+B: boundary prediction with the temporal contexts of actionness and background.

Visualization. Figure 4 visualizes the effects of employing our boundary scoring module. The top row images correspond to the centered frames derived from local peaks of the boundary predictions. The bottom four charts show the estimated boundary probabilities of the BSN and our model. Compared with BSN, we estimate the probabilities using RNN-based boundary scoring of the snippet-level probabilities p^s and p^e, and the proposal-level probabilities p^{se} and p^c. The results show that our model contributes to estimating the more accurate proposal boundaries and less false positive estimations, which again demonstrate the effectiveness of our RNN-based boundary scoring model, *i.e.*, CPN.

4.3. Action detection with our proposals

For assessing the quality of our proposals for helping an action classifier, we feed our proposals into the state-of-the-art action classifier, *i.e.*, P-GCN [50] [1]. After obtaining the returned P-GCN classifying result per-proposal, we score that proposal as the multiplication of the P-GCN classification score and our proposal score in (13). Table 4 shows the detection performance compared to the state-of-the-art methods on the THUMOS-14 testing split. Note that the original P-GCN adopts the proposals generated by BSN [26]. When replacing with our proposals, "CPN+P-GCN" achieves 5% performance gains at mAP@0.5 and 2.5% improvements in comparison to "G-TAD+P-GCN," which applies the same action classifier. The experiment shows the

[1] P-GCN source: `https://github.com/Alvin-Zeng/PGCN`

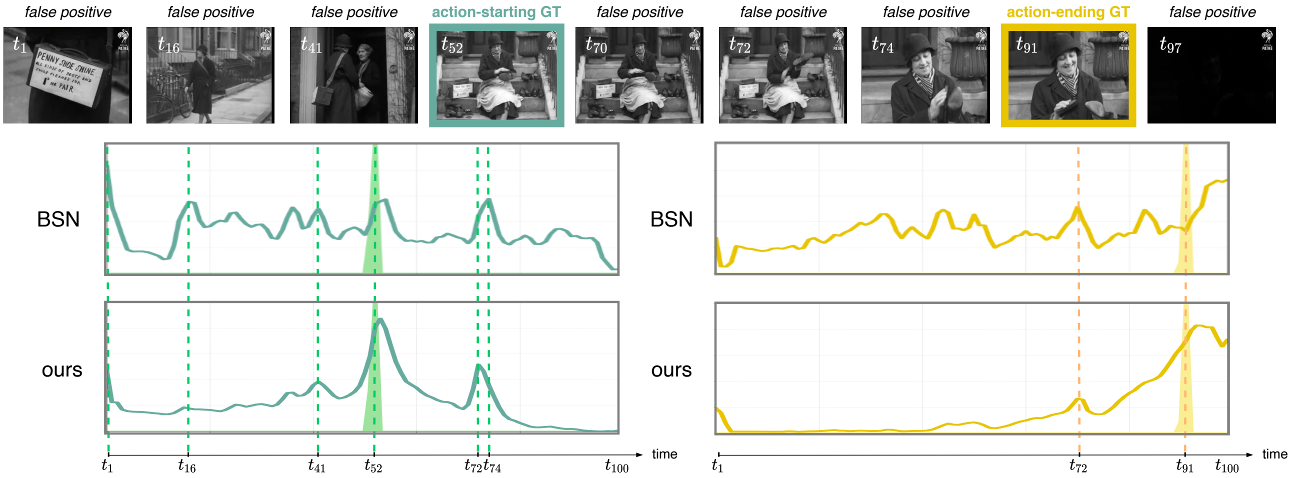

Figure 4: Effect visualization of our model on video id "IDVWoE02zjM." The top images are centered frames of corresponding snippets on the video of *Polishing Shoes* action. The bottom four charts, which plot the predicted boundary probabilities (y-axis) p^s and p^e over the snippet dimension (x-axis), show the false positive reduction of our method.

Method	Reference	mAP@0.1	mAP@0.2	mAP@0.3	mAP@0.4	mAP@0.5	mAP@0.6	mAP@0.7	Average
CTAP [12]	ECCV'18	-	-	-	-	29.9	-	-	-
BSN [26]	ECCV'18	-	-	53.5	45.0	36.9	28.4	20.0	-
TAL-Net [6]	CVPR'18	59.8	57.1	53.2	48.5	42.8	33.8	20.8	45.1
MGG [28]	CVPR'19	-	-	53.9	46.8	37.4	29.5	21.3	-
GTAN [29]	CVPR'19	69.1	63.7	57.8	47.2	38.8	-	-	-
BMN [25]	ICCV'19	-	-	56.0	47.4	38.8	29.7	20.5	-
P-GCN [50]	ICCV'19	69.5	67.8	63.6	57.8	49.1	-	-	-
DBS [16]	AAAI'19	56.7	54.7	50.6	43.1	34.3	24.4	14.7	39.8
DBG [24]	AAAI'20	-	-	57.8	49.4	39.8	30.2	21.7	-
FC-AGCN-P-C3D [23]	AAAI'20	59.3	59.6	57.1	51.6	38.6	28.9	17.0	44.6
PBRNet [27]	AAAI'20	-	-	58.5	54.6	51.3	**41.8**	**29.5**	-
G-TAD [47]	CVPR'20	-	-	54.5	47.6	40.2	30.8	23.4	-
G-TAD [47]+P-GCN	CVPR'20	-	-	66.4	60.4	51.6	37.6	22.9	-
CPN+P-GCN	-	**74.0**	**71.8**	**68.2**	**62.1**	**54.1**	41.5	28.0	**57.1**

Table 4: Temporal action detection results on THUMOS-14 testing split.

advantage of our proposals to action classifier for addressing the action detection task.

5. Conclusions

We have shown that the proposed CPN model, which is composed of the feature enhancing mechanism and the boundary scoring mechanism, better addresses the temporal action proposal generation task and achieves state-of-the-art performance. The extensive experiments show that the performance gain is derived from not only the feature enhancing mechanism, which captures the contextual information over the dimensions of snippets and feature channels for generating a robust snippet-level representation, but also the boundary scoring mechanism, which associates the various scoring probabilities that are obtained by leveraging multiple recurrent neural networks. As a result, the resulting temporal action proposals, therefore, lead to state-of-the-art performances on two challenging datasets.

Acknowledgements. This work was supported in part by the MOST, Taiwan under Grants 110-2634-F-001-009 and 110-2221-E-001-017. We are also grateful to the *National Center for High-performance Computing* for providing computational resources and facilities.

References

[1] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. VQA: visual question answering - www.visualqa.org. *Int. J. Comput. Vis.*, 123(1):4–31, 2017.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[3] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *ECCV*, 2020.

[4] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. SST: single-stream temporal action proposals. In *CVPR*, pages 6373–6382, 2017.

[5] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, pages 4724–4733, 2017.

[6] 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 *CVPR*, pages 1130–1139, 2018.

[7] Jingyuan Chen, Lin Ma, Xinpeng Chen, Zequn Jie, and Jiebo Luo. Localizing natural language in videos. In *AAAI*, pages 8175–8182, 2019.

[8] Jingwen Chen, Yingwei Pan, Yehao Li, Ting Yao, Hongyang Chao, and Tao Mei. Temporal deformable convolutional encoder-decoder networks for video captioning. In *AAAI*, pages 8167–8174, 2019.

[9] Shaoxiang Chen and Yu-Gang Jiang. Motion guided spatial attention for video captioning. In *AAAI*, pages 8191–8198, 2019.

[10] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Daps: Deep action proposals for action understanding. In *ECCV*, pages 768–784, 2016.

[11] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, pages 1933–1941, 2016.

[12] Jiyang Gao, Kan Chen, and Ram Nevatia. CTAP: complementary temporal action proposal generation. In *ECCV*, pages 70–85, 2018.

[13] Jialin Gao, Zhixiang Shi, Jiani Li, Guanshuo Wang, Yufeng Yuan, Shiming Ge, and Xi Zhou. Accurate temporal action proposal generation with relation-aware pyramid network. In *AAAI*, 2020.

[14] Jiyang Gao, Zhenheng Yang, Chen Sun, Kan Chen, and Ram Nevatia. TURN TAP: temporal unit regression network for temporal action proposals. In *ICCV*, pages 3648–3656, 2017.

[15] Lianli Gao, Tao Li, Jingkuan Song, Zhou Zhao, and Heng Tao Shen. Play and rewind: Context-aware video temporal action proposals. *Pattern Recognit.*, 107:107477, 2020.

[16] Zhanning Gao, Le Wang, Qilin Zhang, Zhenxing Niu, Nanning Zheng, and Gang Hua. Video imprint segmentation for temporal action detection in untrimmed videos. In *AAAI*, pages 8328–8335, 2019.

[17] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, pages 6325–6334, 2017.

[18] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015.

[19] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *CVPR*, pages 1914–1923, 2016.

[20] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. In *NeurIPS*, pages 2721–2730, 2019.

[21] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018.

[22] Yu-Gang Jiang, Jingen Liu, Amir Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large number of classes. 2014.

[23] Jun Li, Xianglong Liu, Zhuofan Zong, Wanru Zhao, Mingyuan Zhang, and Jingkuan Song. Graph attention based proposal 3d convnets for action detection. In *AAAI*, 2020.

[24] 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.

[25] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: boundary-matching network for temporal action proposal generation. In *ICCV*, 2019.

[26] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: boundary sensitive network for temporal action proposal generation. In *ECCV*, pages 3–21, 2018.

[27] Qinying Liu and Zilei Wang. Progressive boundary refinement network for temporal action detection. In *AAAI*, 2020.

[28] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. In *CVPR*, pages 3604–3613, 2019.

[29] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *CVPR*, pages 344–353, 2019.

[30] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. Temporal context aggregation network for temporal action proposal refinement. 2021.

[31] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, pages 5534–5542, 2017.

[32] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *CVPR*, pages 1049–1058, 2016.

[33] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[34] 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. In *AAAI*, 2021.

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[37] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. 2021.

[38] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[39] 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 *ECCV*, pages 20–36, 2016.

[40] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018.

[41] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Changxin Gao, and Nong Sang. Self-supervised learning for semi-supervised temporal action proposal. 2021.

[42] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krähenbühl, and Ross B. Girshick. Long-term feature banks for detailed video understanding. In *CVPR*, pages 284–293, 2019.

[43] 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. In *CVPR ActivityNet Workshop*, 2016.

[44] Huijuan Xu, Abir Das, and Kate Saenko. R-C3D: region convolutional 3d network for temporal activity detection. In *ICCV*, pages 5794–5803, 2017.

[45] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In Francis R. Bach and David M. Blei, editors, *ICML*, pages 2048–2057, 2015.

[46] Mengmeng Xu, Juan-Manuel Perez-Rua, Victor Escorcia, Brais Martínez, Xiatian Zhu, Li Zhang, Bernard Ghanem, and Tao Xiang. Boundary-sensitive pre-training for temporal localization in videos. 2021.

[47] Mengmeng Xu, Chen Zhao, David S. Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, 2020.

[48] Li Yao, Atousa Torabi, Kyunghyun Cho, Nicolas Ballas, Christopher J. Pal, Hugo Larochelle, and Aaron C. Courville. Describing videos by exploiting temporal structure. In *ICCV*, pages 4507–4515, 2015.

[49] Ting Yao, Tao Mei, and Yong Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *CVPR*, pages 982–990, 2016.

[50] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *ICCV*, 2019.

[51] Tao Zhang, Shan Liu, Thomas H. Li, and Ge Li. Boundary information matters more: Accurate temporal action detection with temporal boundary network. In *ICASSP*, pages 1642–1646, 2019.

[52] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wan, and Qi Tian. Bottom-up temporal action localization with mutual regularization. In *ECCV*, 2020.

[53] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, pages 2933–2942, 2017.