Hierarchical Visual-Textual Graph for Temporal Activity Localization via Language

Shaoxiang Chen and Yu-Gang Jiang*

Shanghai Key Lab of Intelligent Information Processing,
School of Computer Science, Fudan University
{sxchen13, ygj}@fudan.edu.cn

Abstract. Temporal Activity Localization via Language (TALL) in video is a recently proposed challenging vision task, and tackling it requires fine-grained understanding of the video content, however, this is overlooked by most of the existing works. In this paper, we propose a novel TALL method which builds a Hierarchical Visual-Textual Graph to model interactions between the objects and words as well as among the objects to jointly understand the video contents and the language. We also design a convolutional network with cross-channel communication mechanism to further encourage the information passing between the visual and textual modalities. Finally, we propose a loss function that enforces alignment of the visual representation of the localized activity and the sentence representation, so that the model can predict more accurate temporal boundaries. We evaluated our proposed method on two popular benchmark datasets: Charades-STA and ActivityNet Captions, and achieved state-of-the-art performances on both datasets. Code is available at https://github.com/forwchen/HVTG.

Keywords: Temporal Activity Localization via Language · Hierarchical Visual-Textual Graph · Visual-Textual Alignment

1 Introduction

Localizing temporal region-of-interest in video is a popular research topic in computer vision. Human actions have been the main target for temporal localization in video, and significant progresses are made [27,35,10,4,23,34,55] during the past few years thanks to the development of deep learning. However, actions that are categorized to a limited number of classes are not sufficient for understanding the events in videos. Recently, Gao *et al.* [12] and Hendricks *et al.* [16] proposed to localize complex activities in videos via free-form language queries (*i.e.*, sentence descriptions), and the task is named Temporal Activity Localization via Language (TALL). An example of TALL is shown in Fig. 1. The main difference between TALL and action localization is that an activity can be a composition of multiple actions or sub-activities (*e.g.*, opening refrigerator, taking bottle, and closing refrigerator), and an activity involves frequent interactions among

* Corresponding author.

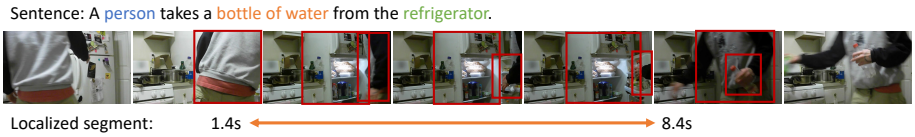

Fig. 1. Demonstration of the Temporal Activity Localization via Language task. The motivation for our work is that the objects and their interactions must be explicitly modeled while jointly considering the sentence content. In this example, the model should understand the interactions among the key objects from the query: bottle, person, and refrigerator, so that it is aware of when the refrigerator is opened/closed and the bottle is taken out of the refrigerator.

persons and objects. Moreover, some actions or sub-activities are not explicitly described by the sentence. Thus, one key to tackling this challenging task is the fine-grained joint understanding of the video's visual content and the sentence's textual content.

TALL has recently attracted attention from both the computer vision and natural language processing communities. Early approaches for solving TALL are inspired by action localization methods and are mainly based on sliding window proposals. In [24,37,25], candidate temporal regions are first generated by sliding windows of fixed scales over the video, then each candidate and the sentence are separately processed by visual and textual encoding modules, and they are finally fed to a cross-modal processing module to generate a ranking score and boundary offsets. These sliding window candidates are query-irrelevant, so some works [48,7] try to generate learned query-guided proposals by leveraging sentence semantics to assign weights to temporal regions. Proposal-based methods generate a large amount of candidate regions in order to achieve a high recall, thus they suffer from high computational cost in both the training and testing phases. Moreover, proposal-based methods are two-stage and cannot be optimized in an end-to-end manner. Similar to the efficient action localization method [4], a group of works [6,42] adopt anchor-based network structure that can generate multi-scale region predictions for all time steps in one pass. Anchor-based methods are end-to-end trainable and can address the efficiency problem, however, they still rely on heuristic rules such as the candidate region scales and strides, which are usually dependent on the dataset statistics. Most recently, some works [53,26] try to discard proposals and directly predict the start and end boundaries of an activity. Although this type of approaches cannot generate multiple proposals, they are computationally more efficient and do not rely on the dataset statistics to design heuristic rules. To summarize, most existing methods focus on generating and refining temporal proposals and improving efficiency, but rarely explore the modeling of fine-grained object interactions.

We argue that while actions can be well captured by their motion patterns [5], understanding activities requires fine-grained modeling of the relations among the objects/persons in videos. As shown in Figure 1, the main motivation for

this work is that object interactions should be modeled jointly with the sentence content for better localization. Existing methods mostly use pretrained action recognition network (such as C3D [39]) to extract global feature representations for video frames, but such representations do not preserve the object-level information inside frames. To address this issue, we extract object-level features in each video frame via an object detection network, and then use a Hierarchical Visual-Textual Graph (HVTG) to encode the features. In the HVTG, objects and words in the sentence are all considered graph nodes. In the first level, the object features first connect and interact with word representations to gather textual information that's relevant to themselves. This achieves a fine-grained understanding of the sentence query based on the video's contents. Next, a fully-connected object graph models the object-object interactions inside each frame, and each node (object) absorbs information from its neighbors. To further aggregate the object graph to obtain a compact representation, a sentence-guided node aggregation is applied to each frame's object features and then bidirectional temporal relation is built via LSTMs. Following the HVTG, we design a convolutional localizer with cross-channel communication [51] between visual and textual modalities (which can be regarded as a more fine-grained visual-textual graph) to predict each frame's relatedness with the sentence. Finally, in order to further close the gap between the visual and textual modalities, an auxiliary loss function is used to align the visual representation of the localized activity and the textual representation. We abbreviate our method as HVTG.

Our contributions in this work are summarized as follows:

- We propose Hierarchical Visual-Textual Graph (HVTG), a novel model for the Temporal Activity Localization via Language (TALL) task. HVTG performs visual-textual interaction in both the object and channel levels, and is among the first methods that consider fine-grained object interactions for the TALL task.
- We propose a novel loss function for the TALL task which aligns the visual representation of the localized activity and the sentence representation, and can effectively improve the localization performance.
- We demonstrate the effectiveness of our HVTG through extensive ablation studies and experiments on two challenging benchmark datasets: Charades-STA and ActivityNet Captions. Our HVTG outperformed recent state-of-the-art methods on both datasets.

2 Related Work

Temporal Action Localization and Proposals. Temporal action localization is closely related to TALL and has been extensively studied in the computer vision literature. The current state-of-the-art methods can be divided into two groups: proposal-based methods and anchor based methods, according to how they generate proposals (candidate temporal segments). Proposal-based methods first generate a set of candidate temporal segments of the target action and then produce classification scores and boundary refinements for the proposals.

Representative methods such as SCNN [35] and TCN [9] generate proposals with simple multi-sale sliding-window strategy, and then exhaustively evaluate the video segments using CNNs. Based on the SCNN proposals, Shou *et al.* [34] further design a more powerful 3D CNN to model the spatio-temporal structure in raw videos and produce fine-grained predictions for temporal boundary refinement. Another direction for improvement is designing more effective proposal generation strategy. Zhao *et al.* [57] propose the Temporal Actionness Grouping (TAG) method for merging high actionness snippets into a proposal using a grouping scheme similar to [32]. Proposal-based methods are not efficient due to their exhaustive evaluation of all the proposals. Anchor-based methods such as DAP [10], SS-TAD [3], and SST [4] overcome this efficiency problem by using RNN (GRU [8] and LSTM [17]) to process videos in one pass and generate a set of fixed-length proposals with confidence scores at every time step. Another line of work adopts CNN architectures to process videos. SSAD [23] applies multiple 2D convolutions to multi-scale video feature maps and predicts action categories and locations at multiple layers. R-C3D [47] extends the ROI-pooling method in object detection, and it applies 3D ROI-pooling to proposals' feature maps from 3D CNN, and the ROI-pooling produces fixed-size features for predicting temporal boundaries and action scores. The proposal generation strategies in most TALL methods are similar to those used in temporal action localization. However, actions are restricted to a predefined categories and can often be captured by their motion patterns. Thus, localizing activities via language is more flexible and challenging.

Video/Image-Text Retrieval. Video/image-text retrieval is also a closely related topic. State-of-the-art methods mainly focus on either learning a visual-semantic common space or encoding techniques for video and text. In [30], video and text (sentence) are respectively encoded with CNN and RNN in parallel and then projected into a common space where embeddings of relevant videos and texts are pulled close. Faghri *et al.* [11] propose VSE++ which improves the visual-semantic common space learning objective by utilizing hard negative image-text pairs. Miech *et al.* [28] design a Mixture of Embedding Experts (MEE) model which computes similarity scores between text and video as a weighted combination of multiple expert embeddings, where the weights are estimated from the aggregated word representations. Wray *et al.* [45] propose to learn separate embeddings for each part-of-speech in a sentence, such as verbs and nouns for retrieving fine-grained actions in videos. The importance of sentence structure is also noted by methods that focus on the encoding techniques. Xu *et al.* [50] and Lin *et al.* [22] both parse a sentence into a parse tree to obtain the syntactic role of each word during sentence encoding. Information fusion is also important when encoding the video or text. Mithun *et al.* [29] utilize multimodal cues in video, such as motion, audio, and object features. JSFusion [52] deeply fuses text and video representations by constructing a pairwise joint representation of word and frame sequences using a soft-attention mechanism. A recent work [54] explores video retrieval with multiple sentences by encoding and aligning video and text hierarchically. Our method draws inspiration from

video-text retrieval to perform feature alignment of the localized activity and sentence in a common space.

Temporal Activity Localization via Language. A large body of TALL methods are proposal-based [12,24,13,25,18,37], which adopt a sliding-window proposal generation strategy and evaluate each proposal using visual-textual cross-modal processing models. Xu *et al.* [49] use the temporal segment proposal network in R-C3D [47], however, R-C3D is designed for action localization and there is a discrepancy between action and activity. Some methods also incorporate semantic information of the sentence into the proposal generation process. The QSPN [48] method uses a query-guided segment proposal network which incorporates sentence embeddings to derive attention weights and reweight the video features for proposal generation. SAP [7] computes frame-wise visual-semantic correlation scores by extracting concepts from video frames and sentence, then uses a score grouping method to form proposal regions. TGN [6], CBP [42], and MAN [56] use the one-shot proposal generation strategy similar to the one in SST [4], the visual and textual features are fused using RNN or CNN to generate a set of proposals at each time step. Recently, ABLR [53] and DEBUG [26] discard proposals and directly predict a (start, end) time pair, which is more efficient. Although there are a few works that use object-level features [18], we emphasize that fine-grained interaction between visual and textual modalities is rarely considered for the TALL task.

3 Proposed Approach

As shown in Fig. 2, our method can be divided into two parts: the HVTG for encoding the video, and the sentence localizer for generating predictions. In Sec. 3.1, we first describe how our HVTG models the interaction between objects and sentence and among the objects. In Sec. 3.2, we then present our convolutional sentence localizer with the cross-channel communication mechanism to further encourage information passing between the visual and textual modalities. In Sec. 3.3, we formulate the set of losses for training our proposed model, among which the visual-textual alignment loss is critical.

3.1 Hierarchical Visual-Textual Graph

Given a video, we first uniformly sample N frames and extract object-level features for each frame: $O_n = \{\boldsymbol{o}_n^1, \boldsymbol{o}_n^2, ..., \boldsymbol{o}_n^M\}$, where $n \in [1, N]$ is the frame index, M is the number of object features, and $\boldsymbol{o}_n \in \mathbb{R}^{d_{obj}}$ is a feature vector. The sentence is represented by a sequence of words $\boldsymbol{S} = \{\boldsymbol{w}^1, \boldsymbol{w}^2, ..., \boldsymbol{w}^Q\}$, where $\boldsymbol{w} \in \mathbb{R}^{d_{word}}$ is a word vector produced by the GloVe [31] word embedding. Note that the sentence is zero-padded or truncated to a fixed length Q. The goal of Hierarchical Visual-Textual Graph is to aggregate object-level visual features to obtain a compact representation for each frame, and simultaneously capture the interaction between objects and sentence and among the objects. As Fig. 2 shows, the HVTG processes the video frame by frame with shared parameters.

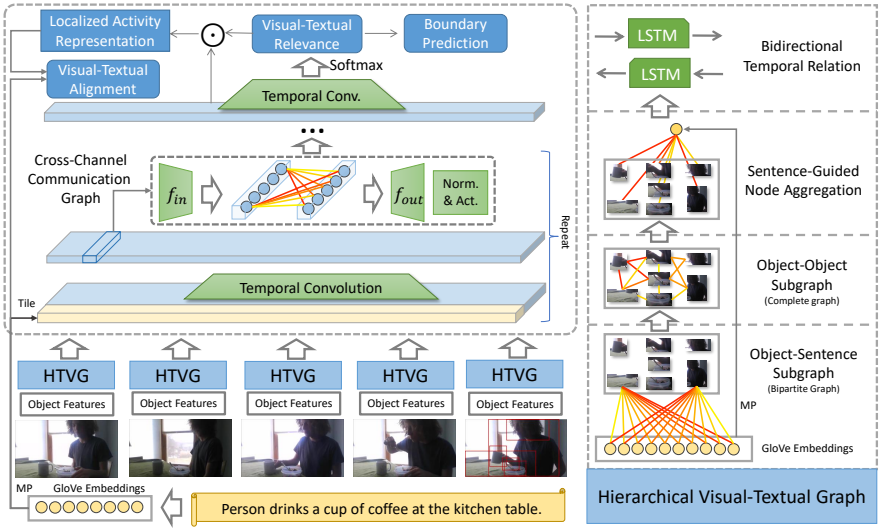

Fig. 2. Overview of our approach. The object features are extracted from each frame, and then processed by the Hierarchical Visual-Textual Graph (HVTG) in four stages: Object-Sentence Subgraph (Eq. (1),(2)), Object-Object Subgraph (Eq. (3),(4)), Sentence-Guided Node Aggregation (Eq. (5),(6)), and Bidirectional Temporal Relation (Eq. (7)). The aggregated visual representation for each frame is processed by multi-layer temporal convolutions with the channel interaction mechanism (Eq. (9),(10)), where f_{in} and f_{out} are linear transformations. Finally, boundary prediction and localized activity representation are obtained based on the visual-textual relevance scores.

Object-Sentence Subgraph (OSS). As Fig. 2 shows, the OSS is a bipartite graph, in which the nodes are divided into two disjoint and independent sets (objects and words) and every edge connects an interacting pair of object and word. We compute the weight of the edge connecting $\boldsymbol{o}_n^i$ and $\boldsymbol{w}^j$ as

$$\boldsymbol{I}_n^{ij} = \mathbf{1}^T\big((\boldsymbol{o}_n^i \boldsymbol{W}_O) \odot (\boldsymbol{w}^j \boldsymbol{W}_S)\big), \tag{1}$$

where $\boldsymbol{W}_O \in \mathbb{R}^{d_{obj} \times d}$ and $\boldsymbol{W}_S \in \mathbb{R}^{d_{word} \times d}$ are projection matrices for projecting the object and word features into a common space, $\odot$ denotes element-wise multiplication, and $\mathbf{1}$ is a vector of ones. We then normalize the edge weights for each object node i, and use them to gather textual information from the whole sentence:

$$\hat{\boldsymbol{I}}_n^{ij} = \mathrm{Softmax}_j(\boldsymbol{I}_n^{ij}), \quad \bar{\boldsymbol{o}}_n^i = \sum_{j=1}^{Q} \hat{\boldsymbol{I}}_n^{ij} (\boldsymbol{w}^j \boldsymbol{U}_S), \tag{2}$$

where $\mathrm{Softmax}_j(\cdot)$ represents column-wise Softmax normalization of a matrix, and $\boldsymbol{U}_S \in \mathbb{R}^{d_{word} \times d_{obj}}$ projects word feature vectors into the object feature space. The resulting object feature $\bar{\boldsymbol{o}}_n^i$ now carries textual information that's relevant

to itself. We also introduce a residual connection from the object feature $\boldsymbol{o}_n^i$ to $\overline{\boldsymbol{o}}_n^i$ to preserve the original object information. Then a Layer Normalization [2] and an object-wise feed-forward network are used to transform the outputs into a lower-dimensional space. The outputs are the sentence-aware object features for the n-th frame, and they are denoted by $\widehat{\boldsymbol{O}}_n = \{\hat{\boldsymbol{o}}_n^1, ..., \hat{\boldsymbol{o}}_n^M\}$.

Object-Object Subgraph (OOS). Since the basic elements of an activity are object-object interactions, we build a complete (fully-connected) subgraph inside each frame. The object features from the OSS are the graph nodes, and the edge weight between two arbitrary objects $\hat{\boldsymbol{o}}_n^i$ and $\hat{\boldsymbol{o}}_n^j$ is computed as

$$\boldsymbol{e}_n^{ij} = g(\hat{\boldsymbol{o}}_n^i || \hat{\boldsymbol{o}}_n^j). \tag{3}$$

Since the object features are already in the same feature space, we use a single-layer feed-forward neural network for $g(\cdot)$ as in [41], and it linearly transforms the concatenation of $\hat{\boldsymbol{o}}_n^i$ and $\hat{\boldsymbol{o}}_n^j$ into a scalar, and it has a LeakyReLU [46] activation. Again, the edge weights for each node i are then normalized and used to gather information from all its neighbors:

$$\hat{\boldsymbol{e}}_n^{ij} = \text{Softmax}_j(\boldsymbol{e}^{ij}), \quad \tilde{\boldsymbol{o}}_n^i = \sum_{j=1}^{M} \hat{\boldsymbol{e}}_n^{ij} \hat{\boldsymbol{o}}_n^j. \tag{4}$$

The resulting object features $\widetilde{\boldsymbol{O}}_n = \{\tilde{\boldsymbol{o}}_n^1, ..., \tilde{\boldsymbol{o}}_n^M\}$ now carry object-object interactions between all pairs of objects inside each frame.

Sentence-Guided Node Aggregation. The object-object subgraph needs to be further aggregated in order to reduce computational cost for subsequent steps. Unlike the previous work [44], which aggregates all the nodes of a graph using mean-pooling, we instead use a sentence-guided attention to assign sentence relevance scores to the objects:

$$\boldsymbol{u} = \tanh(\overline{\boldsymbol{S}}\boldsymbol{W}_S + \widetilde{\boldsymbol{O}}_n\boldsymbol{W}_O + \boldsymbol{b}_O), \quad \hat{\boldsymbol{u}} = \text{Softmax}(\boldsymbol{u}\boldsymbol{W}_r + \boldsymbol{b}_r), \tag{5}$$

where $\overline{\boldsymbol{S}}$ is the mean-pooled sentence feature used to guide the attention, and $\boldsymbol{W}_S \in \mathbb{R}^{d_{word} \times d_{rel}}, \boldsymbol{W}_O \in \mathbb{R}^{d_{obj} \times d_{rel}}$, $\boldsymbol{b}_O \in \mathbb{R}^{d_{rel}}$, $\boldsymbol{W}_r \in \mathbb{R}^{d_{rel} \times 1}$, and $\boldsymbol{b}_r \in \mathbb{R}^1$ are the learnable parameters. The normalized sentence relevance scores $\hat{\boldsymbol{u}} \in \mathbb{R}^M$ are then used to re-weight the object features to obtain an aggregated visual representation for each frame:

$$\boldsymbol{v}_n = \sum_{i=1}^{M} \hat{\boldsymbol{u}}^i \tilde{\boldsymbol{o}}_n^i. \tag{6}$$

Note that the operations in Eq. (1)-(6) are performed for each frame with shared parameters.

Bidirectional Temporal Relation. Finally, we establish temporal relations among the frames with a bidirectional LSTM network:

$$\widetilde{\boldsymbol{V}} = f\big(\overrightarrow{\text{LSTM}}(\{\boldsymbol{v}_1, ..., \boldsymbol{v}_N\}) || \overleftarrow{\text{LSTM}}(\{\boldsymbol{v}_N, ..., \boldsymbol{v}_1\})\big), \tag{7}$$

where $\overrightarrow{\text{LSTM}}$ and $\overleftarrow{\text{LSTM}}$ are the LSTM cells that take the visual representation sequence and produce outputs by aggregating information temporally in forward and backward directions, respectively. We then use a single-layer feed-forward neural network with ReLU activation denoted by $f(\cdot)$ to transform the concatenated LSTM outputs. The final outputs of HVTG, $\widetilde{\boldsymbol{V}} \in \mathbb{R}^{N \times d_{vis}}$, have aggregated the visual information both spatially (in the object level) and temporally, and textual information is also incorporated during the process.

3.2 Sentence Localizer

Given the encoded visual representation $\widetilde{\boldsymbol{V}}$, the sentence localizer's goal is to compute the position-wise visual-textual relevance scores of each frame with the sentence, and then make boundary predictions based on the relevance scores. For this purpose, we first encode the sentence (word sequence) using a bidirectional LSTM like in Eq. (7), and then mean-pool over the LSTM outputs, resulting in an aggregated sentence representation $\widetilde{\boldsymbol{S}} \in \mathbb{R}^{d_{sent}}$. Since computing the frame-wise visual-textual relevance should focus more on the local structure of the visual representation, we use an L-layer temporal convolutional network to process $\widetilde{\boldsymbol{V}}$. Basically, the l-th convolutional layer can be formulated as

$$\boldsymbol{C}^l = \text{Conv}^l(\widetilde{\boldsymbol{V}}^{l-1} || \widetilde{\boldsymbol{S}}; (k^l, c^l)), \tag{8}$$

where $\widetilde{\boldsymbol{V}}^{l-1}$ is the output from the previous layer ($\widetilde{\boldsymbol{V}}^0 = \widetilde{\boldsymbol{V}}$), $\widetilde{\boldsymbol{S}}$ is tiled along the temporal dimension, and $||$ is tensor concatenation along the channel axis. The convolutional operation has two main hyper-parameters: kernel size k^l and number of output channels c^l.

Cross-Channel Communication Graph. We add one key component to the convolutional network, which is a cross-channel communication mechanism [51] that encourages information passing across feature channels in the same layer. By building a cross-channel communication graph (C^3G) on the convolutional outputs of the channel-concatenated visual and textual features, more complementary cross-modal representations can be learned in addition to the HVTG.

Firstly, $\boldsymbol{C}^l \in \mathbb{R}^{N \times c^l}$ is linearly projected to a lower dimensional space to reduce computational cost for the C^3G, resulting in $\boldsymbol{P}^l \in \mathbb{R}^{N \times d^l}$. Then the edge weight for each pair of channels is computed as (the layer index l is omitted)

$$\boldsymbol{T}_n^{ij} = -(\boldsymbol{P}_n^i - \boldsymbol{P}_n^j)^2, \tag{9}$$

where the subscript n stands for index of temporal location (frame), and the superscripts i, j are channel indices. The edge weights for channel i are normalized and averaged across all temporal locations to increase robustness, and then used to aggregate information from all channels:

$$\hat{\boldsymbol{T}}^{ij} = \frac{1}{N} \sum_{n=1}^{N} \text{Softmax}_j(\boldsymbol{T}_n^{ij}), \quad \widehat{\boldsymbol{P}}_n^i = \sum_{j=1}^{d^l} \boldsymbol{P}_n^j \hat{\boldsymbol{T}}^{ij}, \tag{10}$$

where $\widehat{\boldsymbol{P}}_n^i$ is the i-th channel output. In linear transformations (such as convolution), each output channel's computation is independent from the others. By concatenating visual and textual features along the channel axis and then connecting the channels via this C^{3}G, fine-grained cross-modal information will be captured. Finally, $\widehat{\boldsymbol{P}}_n^i$ is projected back to the original c^l-dimensional space, resulting in $\widehat{\boldsymbol{C}}^l \in \mathbb{R}^{N \times c^l}$. Residual connection, Instance Normalization [40], and LeakyReLU [46] activation are sequentially applied to $\widehat{\boldsymbol{C}}^l$, producing the output for the l-th convolutional layer:

$$\widetilde{\boldsymbol{V}}^l = \text{LeakyReLU}(\text{InstanceNorm}(\widehat{\boldsymbol{C}}^l + \boldsymbol{C}^l)). \tag{11}$$

Prediction. Note that the L-th (last) layer is the prediction layer, which has 1 output channel. The visual-textual relevance scores are obtained by applying Softmax to $\widetilde{\boldsymbol{V}}^L$, and we make boundary predictions based on the scores:

$$\boldsymbol{d} = \text{Softmax}(\widetilde{\boldsymbol{V}}^L), \quad \boldsymbol{r} = \text{ReLU}(\boldsymbol{d}\boldsymbol{W}_d + \boldsymbol{b}_d), \tag{12}$$

where $\boldsymbol{d} \in \mathbb{R}^N$ represents the relevance scores for all the temporal locations (frames) in the video, and $\boldsymbol{r} \in \mathbb{R}^2$ are two scalars representing the predicted start and end time of the localized sentence.

3.3 Losses

The training is done in a mini-batch optimization manner. We denote the batch predictions by $\boldsymbol{D} = \{\boldsymbol{d}_1, ..., \boldsymbol{d}_B\}$ and $\boldsymbol{R} = \{\boldsymbol{r}_1, ..., \boldsymbol{r}_B\}$. The ground-truth labels are denoted by $\widehat{\boldsymbol{R}} = \{\widehat{\boldsymbol{r}}_1, ..., \widehat{\boldsymbol{r}}_B\}$, where $\widehat{\boldsymbol{r}}_i \in \mathbb{R}^2$ is the human-annotated start and end times. We first use two basic objectives for the boundary and relevance score predictions.

Boundary Loss. We simply use the Huber loss for boundary prediction:

$$\text{Loss}_b = \sum_{i=1}^{B} \text{Huber}(\boldsymbol{r}_i - \widehat{\boldsymbol{r}}_i). \tag{13}$$

Relevance Loss. Based on the ground-truth temporal boundaries, we construct the position-wise relevance masks $\widehat{\boldsymbol{D}} = \{\widehat{\boldsymbol{d}}_1, ..., \widehat{\boldsymbol{d}}_B\}$, where $\widehat{\boldsymbol{d}}_i \in \mathbb{R}^N$ is constructed as

$$\widehat{\boldsymbol{d}}_i[n] = \begin{cases} 1 & \widehat{\boldsymbol{r}}_i[0] \leq n \leq \widehat{\boldsymbol{r}}_i[1], \\ 0 & \text{otherwise}, \end{cases} \tag{14}$$

where $[\cdot]$ is the indexing operator along the first axis. Then the relevance loss is computed as

$$\text{Loss}_r = -\sum_{i=1}^{B} \frac{\widehat{\boldsymbol{d}}_i \cdot \log(\boldsymbol{d}_i)}{\sum_n \widehat{\boldsymbol{d}}_i[n]}, \tag{15}$$

where $\cdot$ represents vector dot-product. This loss encourages the visual-textual relevance scores to be high at the ground-truth locations.

Visual-Textual Alignment Loss. Based on the predicted visual-textual relevance scores, we can obtain the representation for the localized activity by a weighted-sum of the visual representation (convolutional outputs prior to the prediction layer) using the relevance scores as the weights:

$$\overline{\boldsymbol{V}}_i = \sum_{n=1}^{N} \boldsymbol{d}_i[n] \widetilde{\boldsymbol{V}}_i^{L-1}[n]. \tag{16}$$

Then the visual-textual alignment is performed across the mini-batch:

$$\text{Loss}_a = \frac{1}{B^2}(\sum_{i=1}^{B} \log(1+\exp(-\text{sim}_{i,i})) + \frac{1}{B}\sum_{i=1}^{B}\sum_{\substack{j=1 \\ j\neq i}}^{B} \log(1+\exp(\text{sim}_{i,j}))). \tag{17}$$

where $\text{sim}_{i,j}$ is the similarity function (*e.g.*, negative ℓ_2 distance) between the visual representation $\overline{\boldsymbol{V}}_i$ and the textual representation $\widetilde{\boldsymbol{S}}_j$ in a mini-batch. The representations of the matching video segment and sentence are pulled close in the feature space by this loss, and otherwise their representations are pushed away. Note that this loss supervises not only the learning of the visual and textual representations, but also the relevance score predictions, since the gradients back-propagate through $\overline{\boldsymbol{V}}$, $\widetilde{\boldsymbol{S}}$, and also $\boldsymbol{d}$.

Feature Normalization Loss. To regularize the feature learning for visual-textual alignment and make the model focus more on predicting better visual-textual relevance scores, we further apply a learnable feature normalization [58] to the $(L-1)$-th convolutional layer's outputs:

$$\text{Loss}_n = \sum_{i=1}^{B}\sum_{n=1}^{N}(||\widetilde{\boldsymbol{V}}_i^{L-1}[n]||_2 - F)^2, \tag{18}$$

where $||\cdot||_2$ is the ℓ_2-norm of a vector, and F is a learnable parameter. Finally, the above losses are combined with constant weights to balance their scales:

$$\text{Loss} = \lambda_b \text{Loss}_b + \lambda_r \text{Loss}_r + \lambda_a \text{Loss}_a + \lambda_n \text{Loss}_n. \tag{19}$$

4 Experiments

In this section, we conduct extensive ablation studies to validate the design choices for the different components of our model, and we also present performance comparisons with state-of-the-art methods on the benchmark datasets.

4.1 Datasets and Experimental Settings

Charades-STA. The Charades-STA dataset [12] is built based on the Charades dataset [36], which contains around 10k videos with temporal action (157 classes)

annotations and video-level descriptions for indoor activities. Charades-STA is built by semi-automatically constructing temporal sentence annotations for the activities, and the annotations are human-checked. There are 12,408 and 3,720 sentence-video pairs for training and testing, respectively. The average video duration is 30 seconds, and the average sentence length is 6.2 words.

ActivityNet Captions. The ActivityNet Captions dataset [20] contains around 20k videos with temporal sentence annotation. The dataset is originally split into 10,024, 4,926, and 5,044 videos for training, validation, and testing, respectively. Since the testing set is not publicly available, we follow previous works to use the validation set for testing. ActivityNet Captions is the largest dataset for TALL, and it also has the most diverse activities. The average video duration is 117 seconds, and the average sentence length is 13.5 words.

Evaluation Metrics. We measure the average recall of our predictions of N sentence-video pairs at different temporal IoU thresholds, and this is the same as previous works [12,53]. Formally, Recall $= \frac{1}{N}\sum_{i=1}^{N} R(s_i, a_i, m)$, where $R(s_i, a_i, m) = 1$ if the temporal IoU between our prediction s_i and annotation a_i is greater than m, otherwise $R(s_i, a_i, m) = 0$. We also choose the same IoU thresholds $\{0.3, 0.5, 0.7\}$ as previous works.

Implementation Details. The video frames are temporally down-sampled with a rate of 1/4 and 1/32 for Charades-STA and ActivityNet Captions, respectively. For each frame, we extract 16 object regions (bounding boxes) with the object detection network [33,1] trained on Visual Genome [21], and then we perform ROIAlign [15] on the InceptionResnet V2 [38] feature maps to obtain object features. The sentences are truncated or padded to a maximum length of 12 words and 30 words for Charades-STA and ActivityNet Captions, respectively. Then the words are initialized with the 300d GloVe [31] embeddings, and all word vectors are fixed during training. The temporal boundaries for each sentence are normalized to be in $[0, 1]$. In the HVTG, we employ 8 attention heads as [41] in the object-sentence subgraph to learn more diverse representations. We use 4 convolutional layers for the sentence localizer, whose output channel numbers are $[512, 256, 256, 1]$ and kernel sizes are set to 5 or 3. The learnable parameter F for the feature normalization loss is initialized to 10. We use the Adam optimizer [19] with a learning rate of 0.0001 and a batch size of 32 to optimize the loss in Eq. (19). The loss weights λ_b, λ_r, λ_a, and λ_n are empirically set to 1, 5, 1, and 0.001, respectively.

4.2 Ablation Study

To demonstrate the effectiveness of our proposed approach, we first examine the effects of each important component by performing ablation studies[1].

Effects of Hierarchical Visual-Textual Graph Components. As shown in Table 1, for the HVTG, we mainly focus on the effects of the object-sentence subgraph (OSS), object-object subgraph (OOS), and sentence-guided node aggregation (SGNA). When all the three components are removed from our model,

[1] Due to the space limit, more experiments are placed in the Supplementary Material.

Table 1. Performances of our model variants with different HVTG configurations. OSS means object-sentence subgraph, OOS means object-object subgraph, and SGNA means sentence-guided node aggregation.

#	OSS	OOS	SGNA	IoU=0.3	IoU=0.5	IoU=0.7
0	✗	✗	✗	55.56	40.38	19.49
1	✓	✗	✗	58.36	44.97	21.77
2	✗	✓	✗	57.42	42.31	19.97
3	✗	✗	✓	58.46	42.93	20.86
4	✓	✓	✓	61.37	47.27	23.30

Table 2. Performances of our model variants with Cross-Channel Communication Graph (C^3G) and alignment loss enabled/disabled. Note that the experiments are done with all the HVTG components of Table 1 enabled.

#	C^3G	Alignment Loss	IoU=0.3	IoU=0.5	IoU=0.7
0	✗	✗	58.18	43.23	20.77
1	✗	✓	60.36	45.51	22.31
2	✓	✗	59.27	44.65	22.42
3	✓	✓	61.37	47.27	23.30

the objects do not connect and interact with the sentence and their features are simply mean-pooled and fed to the bi-LSTM, which is the case of setting 0 in Table 1 and is used as the baseline here. And note that disabling SGNA also means the object features are just mean-pooled, and the sentence localizer used here is the same as in our full model. Comparing settings 1, 2, and 3 with the baseline, we can see that each of the three components can effectively improve the performances. The average improvements brought by OSS, OOS, and SGNA are 9.5%, 3.7%, and 6.3%, respectively. This demonstrates that the visual-textual interaction in the HVTG is crucial for TALL. Combining all the three components consistently yields better performances, which indicates that our HVTG can benefit from the these interactions to better understand the video content and sentence query.

Effects of Cross-Channel Communication Graph. C^3G is the critical component of our convolutional localizer. As shown in Table 2, enabling it leads to a significant performance boost (comparing Settings 0 and 2, or 1 and 3). This means that visual-textual interaction at the object-sentence level is not enough, and channel-level interaction is an effective way to further encourage the messaging passing between the two modalities, and the local associations between video frames and sentence are better captured with C^3G.

Effects of Alignment Loss. Comparing settings 0 and 1 (or 2 and 3) in Table 2, it is clear that the alignment loss is beneficial for the TALL task, which validates our design of encouraging the localized activity and the sentence to be close in the feature space.

4.3 Comparison with State-of-the-Art Methods

Compared Methods. The state-of-the-art TALL methods we compared with are categorized as follows:

Table 3. Performance comparison on the Charades-STA dataset.

Method	IoU=0.3	IoU=0.5	IoU=0.7
MCN [16]	32.59	11.67	2.63
ACRN [24]	38.06	20.26	7.64
ROLE [25]	37.68	21.74	7.82
SLTA [18]	38.96	22.81	8.25
CTRL [12]	-	23.63	8.89
VAL [37]	-	23.12	9.16
ACL [13]	-	30.48	12.20
SAP [7]	-	27.42	13.36
SM-RL [43]	-	24.36	11.17
TripNet [14]	51.33	36.61	14.50
QSPN [48]	54.7	35.6	15.8
CBP [42]	-	36.80	18.87
MAN [56]	-	46.53	22.72
ABLR [53]	51.55	35.43	15.05
DEBUG [26]	54.95	37.39	17.69
HVTG	**61.37**	**47.27**	**23.30**

Table 4. Performance comparison on the ActivityNet Captions dataset.

Method	IoU=0.3	IoU=0.5	IoU=0.7
QSPN [48]	45.3	27.7	13.6
TGN [6]	43.81	27.93	-
TripNet [14]	48.42	32.19	13.93
CBP [42]	54.30	35.76	17.80
ABLR [53]	55.67	36.79	-
DEBUG [26]	55.91	39.72	-
HVTG	**57.60**	**40.15**	**18.27**

- Sliding-window proposal methods: TALL [12], ACRN [24], MAC [13], ROLE [25], SLTA [18], and VAL [37].
- Learned proposal generation methods: QSPN [48], EF+Cap [49], and SAP [7].
- Anchor-based methods: TGN [6], CBP [42], and MAN [56].
- Direct boundary prediction methods: ABLR [53] and DEBUG [26].
- Reinforcement Learning-based methods: SM-RL [43] and TripNet [14].

Results on Charades-STA. Table 3 shows the performance comparison on the Charades-STA dataset, which is more commonly used. As can be observed, our HVTG outperforms all the compared methods in all three evaluation metrics. Especially when comparing with methods that also adopt the direct boundary prediction strategy, ABLR [53] and DEBUG [26], the advantages of our HVTG is significant, which demonstrates the effectiveness of modeling visual-textual interaction and alignment for the TALL task.

Results on ActivityNet Captions. The video contents of the ActivityNet dataset and the Charades-STA dataset are quite different, because ActivityNet contains not only indoor activities and the duration of its videos are generally longer. Moreover, the language queries in ActivityNet Captions are more complex than Charades-STA. Thus, strong adaptivity is required for TALL methods to work well on both datasets. There are less state-of-the-art methods that have reported results on ActivityNet Captions. As shown in Table 4, our HVTG also outperforms the compared methods, demonstrating the superiority of HVTG. It is worth noting that limited by the GPU memory, the ActivityNet videos are temporally down-sampled with a larger stride, which may have affected the performances of our method.

4.4 Result Visualizations

In Fig. 3, we provide some examples with visualization of the visual-textual relevance scores. As can be observed in the first two examples (top and middle), when the visual-textual alignment is enabled, our model predicts better relevance

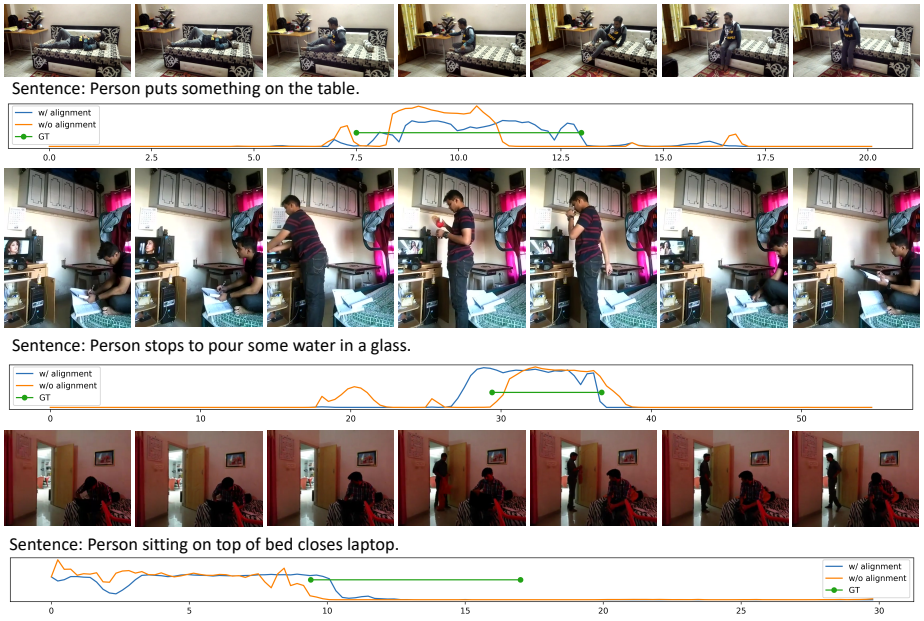

Fig. 3. Example results with the visual-textual relevance scores visualized. The last one is a failed case where the relevance scores do not match the ground-truth well.

scores which are more concentrated in the ground-truth region. We also show a failed example (bottom), where the "closes laptop" activity is not accurately localized. We conjecture the reason is that the poor lighting affects the quality of the captured visual information in the object features.

5 Conclusions

We presented a novel method named Hierarchical Visual-Textual Graph (HVTG) for tackling the Temporal Activity Localization via Language (TALL) task, which is challenging since it requires fine-grained understanding of visual contents while jointly considering the language query. To tackle the challenge, our HVTG method builds a hierarchical graph structure to perform interactions between the object and sentence and among the objects. We also adopt a cross-channel communication graph to encourage more fine-grained information passing between the visual and textual modalities. Finally, visual-textual alignment is enforced to encourage the localized activity to be close to the corresponding language query in the feature space. We achieved state-of-the-art performances on two challenging datasets: Charades-STA and ActivityNet Captions. Future work includes incorporating temporal relation modeling into the object subgraphs and improving the interpretability of visual-textual interactions inside the graph.

References

1. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: CVPR (2018)
2. Ba, L.J., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Buch, S., Escorcia, V., Ghanem, B., Fei-Fei, L., Niebles, J.C.: End-to-end, single-stream temporal action detection in untrimmed videos. In: BMVC (2017)
4. Buch, S., Escorcia, V., Shen, C., Ghanem, B., Niebles, J.C.: SST: single-stream temporal action proposals. In: CVPR (2017)
5. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: CVPR (2017)
6. Chen, J., Chen, X., Ma, L., Jie, Z., Chua, T.: Temporally grounding natural sentence in video. In: EMNLP (2018)
7. Chen, S., Jiang, Y.: Semantic proposal for activity localization in videos via sentence query. In: AAAI (2019)
8. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: NIPS Workshop on Deep Learning (2014)
9. Dai, X., Singh, B., Zhang, G., Davis, L.S., Chen, Y.Q.: Temporal context network for activity localization in videos. In: ICCV (2017)
10. Escorcia, V., Heilbron, F.C., Niebles, J.C., Ghanem, B.: Daps: Deep action proposals for action understanding. In: ECCV (2016)
11. Faghri, F., Fleet, D.J., Kiros, J.R., Fidler, S.: VSE++: improving visual-semantic embeddings with hard negatives. In: BMVC (2018)
12. Gao, J., Sun, C., Yang, Z., Nevatia, R.: TALL: temporal activity localization via language query. In: ICCV (2017)
13. Ge, R., Gao, J., Chen, K., Nevatia, R.: MAC: mining activity concepts for language-based temporal localization. In: WACV (2019)
14. Hahn, M., Kadav, A., Rehg, J.M., Graf, H.P.: Tripping through time: Efficient localization of activities in videos. arXiv preprint arXiv:1904.09936 (2019)
15. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: ICCV (2017)
16. Hendricks, L.A., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.C.: Localizing moments in video with natural language. In: ICCV (2017)
17. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8) (1997)
18. Jiang, B., Huang, X., Yang, C., Yuan, J.: Cross-modal video moment retrieval with spatial and language-temporal attention. In: ICMR (2019)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
20. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Niebles, J.C.: Dense-captioning events in videos. In: ICCV (2017)
21. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L., Shamma, D.A., Bernstein, M.S., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision **123**(1) (2017)
22. Lin, D., Fidler, S., Kong, C., Urtasun, R.: Visual semantic search: Retrieving videos via complex textual queries. In: CVPR (2014)

23. Lin, T., Zhao, X., Shou, Z.: Single shot temporal action detection. In: ACM MM (2017)
24. Liu, M., Wang, X., Nie, L., He, X., Chen, B., Chua, T.: Attentive moment retrieval in videos. In: ACM SIGIR (2018)
25. Liu, M., Wang, X., Nie, L., Tian, Q., Chen, B., Chua, T.: Cross-modal moment localization in videos. In: ACM MM (2018)
26. Lu, C., Chen, L., Tan, C., Li, X., Xiao, J.: DEBUG: A dense bottom-up grounding approach for natural language video localization. In: EMNLP-IJCNLP (2019)
27. Mettes, P., van Gemert, J.C., Cappallo, S., Mensink, T., Snoek, C.G.M.: Bag-of-fragments: Selecting and encoding video fragments for event detection and recounting. In: ICMR (2015)
28. Miech, A., Laptev, I., Sivic, J.: Learning a text-video embedding from incomplete and heterogeneous data. arXiv preprint arXiv:1804.02516 (2018)
29. Mithun, N.C., Li, J., Metze, F., Roy-Chowdhury, A.K.: Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In: ICMR (2018)
30. Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J., Yokoya, N.: Learning joint representations of videos and sentences with web image search. In: ECCV Workshops (2016)
31. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP (2014)
32. Pont-Tuset, J., Arbelaez, P., Barron, J.T., Marqués, F., Malik, J.: Multiscale combinatorial grouping for image segmentation and object proposal generation. IEEE TPAMI **39**(1), 128–140 (2017)
33. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)
34. Shou, Z., Chan, J., Zareian, A., Miyazawa, K., Chang, S.: CDC: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: CVPR (2017)
35. Shou, Z., Wang, D., Chang, S.: Temporal action localization in untrimmed videos via multi-stage cnns. In: CVPR (2016)
36. Sigurdsson, G.A., Varol, G., Wang, X., Farhadi, A., Laptev, I., Gupta, A.: Hollywood in homes: Crowdsourcing data collection for activity understanding. In: ECCV (2016)
37. Song, X., Han, Y.: VAL: visual-attention action localizer. In: PCM (2018)
38. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI (2017)
39. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV (2015)
40. Ulyanov, D., Vedaldi, A., Lempitsky, V.S.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
41. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: ICLR (2018)
42. Wang, J., Ma, L., Jiang, W.: Temporally grounding language queries in videos by contextual boundary-aware prediction. In: AAAI (2020)
43. Wang, W., Huang, Y., Wang, L.: Language-driven temporal activity localization: A semantic matching reinforcement learning model. In: CVPR (2019)
44. Wang, X., Gupta, A.: Videos as space-time region graphs. In: ECCV (2018)
45. Wray, M., Larlus, D., Csurka, G., Damen, D.: Fine-grained action retrieval through multiple parts-of-speech embeddings. arXiv preprint arXiv:1908.03477 (2019)
46. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015)

47. Xu, H., Das, A., Saenko, K.: R-C3D: region convolutional 3d network for temporal activity detection. In: ICCV (2017)
48. Xu, H., He, K., Plummer, B.A., Sigal, L., Sclaroff, S., Saenko, K.: Multilevel language and vision integration for text-to-clip retrieval. In: AAAI (2019)
49. Xu, H., He, K., Sigal, L., Sclaroff, S., Saenko, K.: Text-to-clip video retrieval with early fusion and re-captioning. arXiv preprint arXiv:1804.05113 (2018)
50. Xu, R., Xiong, C., Chen, W., Corso, J.J.: Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In: AAAI (2015)
51. Yang, J., Ren, Z., Gan, C., Zhu, H., Parikh, D.: Cross-channel communication networks. In: NeurIPS (2019)
52. Yu, Y., Kim, J., Kim, G.: A joint sequence fusion model for video question answering and retrieval. In: ECCV (2018)
53. Yuan, Y., Mei, T., Zhu, W.: To find where you talk: Temporal sentence localization in video with attention based location regression. In: AAAI (2019)
54. Zhang, B., Hu, H., Sha, F.: Cross-modal and hierarchical modeling of video and text. In: ECCV (2018)
55. Zhang, D., Dai, X., Wang, X., Wang, Y.: S3D: single shot multi-span detector via fully 3d convolutional networks. In: BMVC (2018)
56. Zhang, D., Dai, X., Wang, X., Wang, Y., Davis, L.S.: MAN: moment alignment network for natural language moment retrieval via iterative graph adjustment. In: CVPR (2019)
57. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: ICCV (2017)
58. Zheng, Y., Pal, D.K., Savvides, M.: Ring loss: Convex feature normalization for face recognition. In: CVPR (2018)