Dual Path Interaction Network for Video Moment Localization

Hao Wang
University of Science and Technology of China
whqaz@mail.ustc.edu.cn

Zheng-Jun Zha*
University of Science and Technology of China
zhazj@ustc.edu.cn

Xuejin Chen
University of Science and Technology of China
xjchen99@ustc.edu.cn

Zhiwei Xiong
University of Science and Technology of China
zwxiong@ustc.edu.cn

Jiebo Luo
University of Rochester
jluo@cs.rochester.edu

ABSTRACT

Video moment localization aims to localize a specific moment in a video by a natural language query. Previous works either use alignment information to find out the best-matching candidate (i.e., top-down approach) or use discrimination information to predict the temporal boundaries of the match (i.e., bottom-up approach). Little research has taken both the candidate-level alignment information and frame-level boundary information together and considers the complementarity between them. In this paper, we propose a unified top-down and bottom-up approach called Dual Path Interaction Network (DPIN), where the alignment and discrimination information are closely connected to jointly make the prediction. Our model includes a boundary prediction pathway encoding the frame-level representation and an alignment pathway extracting the candidate-level representation. The two branches of our network predict two complementary but different representations for moment localization. To enforce the consistency and strengthen the connection between the two representations, we propose a semantically conditioned interaction module. The experimental results on three popular benchmarks (i.e., TACoS, Charades-STA, and Activity-Caption) demonstrate that the proposed approach effectively localizes the relevant moment and outperforms the state-of-the-art approaches.

CCS CONCEPTS

• **Information systems** → **Video search**; **Novelty in information retrieval**.

KEYWORDS

Cross-modal Retrieval; Moment Localization

ACM Reference Format:
Hao Wang, Zheng-Jun Zha, Xuejin Chen, Zhiwei Xiong, and Jiebo Luo. 2020. Dual Path Interaction Network for Video Moment Localization. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413975

*Corresponding author.

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413975

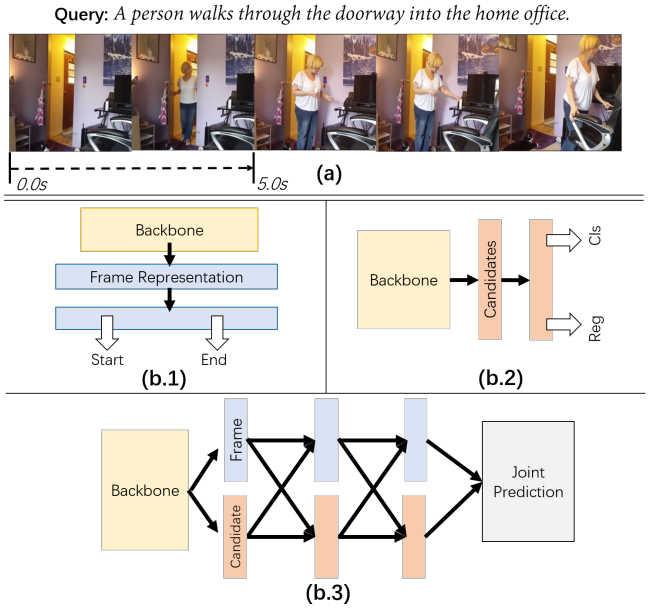

Figure 1: (a): An example of video moment localization with a natural language query. (b.1): A typical bottom-up framework encoding frame-level representation. (b.2): A typical top-down framework encoding candidate-level representation. (b.3): The proposed approach.

1 INTRODUCTION

Automatically finding a specific activity moment described by a natural language query in a video is a challenging task proposed by [1, 9], which requires to understand both the language and vision[22, 23, 26, 42, 45, 46] and has drawn increasing attention from the computer vision and multimedia communities [4, 5, 10–12, 16, 21, 24, 25, 37, 41, 47, 49, 50]. As depicted in Figure 1 (a), given a query "A person walks through the doorway into the home office" and the corresponding video content, this task aims to localize a specific moment best matched with the language description and output the start and end timestamps of the moment.

There are two main types of approaches for temporally localizing a specific activity in a video. As illustrated in Figure 1(b.1), the bottom-up approach encodes the video content and language query into frame-level representation. The high temporal resolution representation helps outputting precise start and end time as the boundary of the moment. However, since these frame-level features are isolated before being combined into a candidate moment, they are unaware of the content of their constructed moment and often lack of consistency. Meanwhile, as shown in Figure 1(b.2), the top-down approach divides a video into several segments as candidate moments and calculates the matching score between each candidate and the query based on candidate-level representation. These candidates possess the alignment information as a whole moment but neglect the boundary discrimination. Although some approaches add another regression layer to predict the location offset of the candidate, the candidate-level alignment representation is not suitable for the additional regression task. In short, both the bottom-up and top-down approaches have advantages and limitations as existing works neglect the complementarity of the frame-level and candidate-level representations.

In this paper, the impact of both frame-level boundary representation and candidate-level alignment representation are taken into consideration simultaneously. We propose a unified bottom-up and top-down approach: Dual Path Interaction Network (DPIN). As shown in Figure 1(b.3), in contrast to previous works that only use frame-level or candidate-level features, DPIN uses two branches to incorporate the two different but complementary representations, and further uses a query conditioned interaction module to make two representations aware of each other.

The key motivation for using a dual path interaction network is that the discriminative information for boundary prediction and candidate matching are different but complementary. We should focus on the moment boundary information in the frame-level path and the semantic alignment information in the candidate-level path. Moreover, dense connections between these two representations ensure the semantic consistency between them. Only the information semantically correlated to the query of one path should be communicated to the other.

In terms of the two representations, the frame-level path contains boundary information focused on boundary regions while the candidate-level path contains alignment information concentrated in the entire candidate moment. We learn the sequence of frame-level features in a 1D temporal fashion and place the candidate-level features in a 2D temporal arrangement fashion based on [48] (i.e., The $(i, j)_{th}$ element of the 2D map indicts a moment start from i and end at j timestamps). We transfer the semantically conditioned information flow of one path to the other because the candidate representation offers matching information for the start and end frames that constitute the moment, and the frame representation offers discriminative boundary information for each candidate. Through the query semantic gate mechanism, we control that only the information highly correlated to the query will be transferred to the other. The gate mechanism ensures that each path only aggregates helpful complementary information from the other path.

Traditional top-down approaches often densely generate candidates and thus incur a high computation burden. With the high-resolution frame-level representation, we can only generate a small number of candidate moments and keep the 2D candidate-level features compact and in low resolution. On the other hand, traditional bottom-up approaches can only output rank-1 results since it is time-consuming to convert high-resolution start and end predictions into moment candidates. To balance the resolution difference between the two paths and obtain a joint output with the expected temporal resolution, we fuse the outputs from the two paths through a resolution adaptive output strategy. Specifically, we temporally down-sample the frame-level start and end boundary prediction scores and up-sample the candidate-level 2D matching score map. Then we fuse them and obtain the final match scoring of each candidate moment.

In summary, the main contributions of this work are as follows:

- We propose a unified top-down and bottom-up approach called Dual Path Interaction Network (DPIN), which encodes two different but complementary frame-level and candidate-level representations for moment localization;
- We propose a semantically conditioned interaction module, which ensures that each path only aggregate query-correlated information from the other path, to enforce the consistency between two representations;
- We propose a resolution-adaptive joint prediction module to fuse and output consistent prediction with scalable temporal resolution.
- We conduct extensive experiments on three benchmarks, i.e., TACoS, Charades-STA and Activity-Caption, which verify the effectiveness of our proposed approach with superior performance over the state-of-the-art methods.

2 RELATED WORK

2.1 Temporal Action Localization

Temporal action localization aims to predict the start and end time of a specific action and the categorical label for the action instance in an untrimmed video. Authors of [28, 38] learned the temporal boundaries of the action instance by using frame or segment-level classification. Authors of [2, 3, 32, 40, 51] used two-stage temporal detection framework, which generates a series of proposals and makes action classification and boundary refinement on them.

This task is highly related to the task of moment localization by natural language, except that the range of action categories is fixed and pre-defined, which makes the task unable to generalize to the real world with various activities. To overcome the limitation, a new task, i.e., moment localization by language, has been proposed.

2.2 Moment Localization by Language

The task of moment localization by language aims at predicting the start and end times of the activity depicted by a language query within the untrimmed video. This is a challenge task introduced in recent works[1, 9] and has drawn increasing attention recently[4–6, 10–12, 16, 21, 24, 25, 27, 29, 36, 37, 41, 43, 47, 48, 50].

Gao *et al.* [9] proposed cross-modal temporal regression localizer (CTRL) model. It takes independent moments as input through sliding windows with predefined fixed frame length and uses alignment and regression loss for activity location refinement. Though this method is simple and effective, it is unable to aggregate the

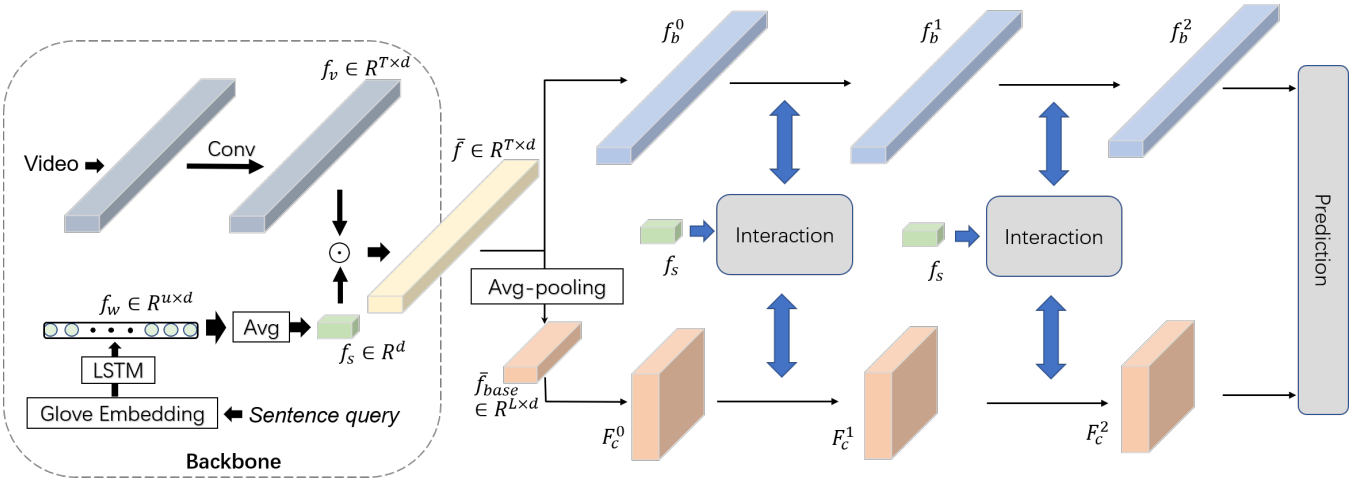

Figure 2: Architecture of the proposed dual path interaction network. The framework takes as input a sequence of video features and a sequence of word embedding features. The generated fused features from the backbone are then fed into two paths to encode frame-level representation and candidate-level representation. With the semantically conditioned interaction module, each path aggregates helpful information and improves the consistency. We make a joint prediction by fusing the two kinds of representation.

cross-modal information and reason the relationship between candidate moments. Following with the CTRL framework of [9], several recent works have been proposed. Liu *et al.* [25] introduced the query attention mechanism which uses the temporal context information to learn the word attention. The method of [24] introduced a temporal memory attention network to memorize the contextual information for each moment. Authors of [10] proposed to mine the activity concepts from both videos and language queries as complementary information. Authors of [16] took advantage of object-level local spatial features to capture the interaction information. Though these methods have utilized the attention mechanism to highlight the useful part, they did not consider the inherent sequence structure of both video and sentence when interacting with the cross-modal information. Authors of [1] proposed the Moment Context Network(MCN), which relies on local and global video features. They compare the similarity between candidate moments and the query sentence in a common embedding space with ranking loss. Authors of [47] integrated candidate moment generating and temporal reasoning by using a single-shot structure.

In addition to the above top-down approaches, authors of [4–6, 27, 29] tackled this task with bottom-up approaches. The method of [4] incorporated the frame-specific sentence representation across video-sentence modalities. Following [4], Authors of [5] utilized cross-gated attended recurrent network with the cross-modal interactor and the self interactor to catch the interactions between the sentence and video. Authors of [27] making full use of positive samples to alleviate the severe imbalance problem. Authors of [6] use a Graph-FPN layer to encoder scene relationships and semantics.

Recently, reinforcement learning methods [12, 37] have been employed. These methods observed the clip and sentence representation of the current location and formulated the selection of start and end time points as a sequential decision-making process.

Different from recent studies, we take into consideration the impact of both candidate-level alignment information and frame-level boundary information in a unified framework. We demonstrate that combining these two different but complementary representations could obtain more discriminative outputs and performs favorably against state-of-the-art methods for video moment localization.

3 OUR APPROACH

We propose the dual path interaction network for the task of temporal activity localization by natural language. Figure 2 depicts the network architecture of the proposed DPIN. The whole framework is made up of five components: (1) A cross-modal feature encoding backbone. It takes as input a sequence of video features and the query sentence feature extracted from a recurrent neural network and outputs a sequence of fused features. (2) A frame-level representation path. We extract the discriminative boundary information from the input of frame-level fused features in this path. (3) A candidate-level representation path. We arrange the moment candidate features in a 2D temporal map from the down-sampled low resolution fused features and extract the alignment information. (4) Several semantically conditioned interaction modules coupled between two paths. These interaction modules enforce consistency and strengthen the connection between the two different but complementary representations. (5) A resolution adaptive joint prediction module. We obtain a fused output with scalable expected temporal resolution from two paths.

3.1 Problem Formulation

Denote an untrimmed input video as V with T_f frames. The query sentence S is made up of a sequence of words $w = \{w_i\}_{i=1}^{u}$ which depicts a specific moment with the annotation τ_s and τ_e as start and end time in the video. Given the video V and the query sentence S,

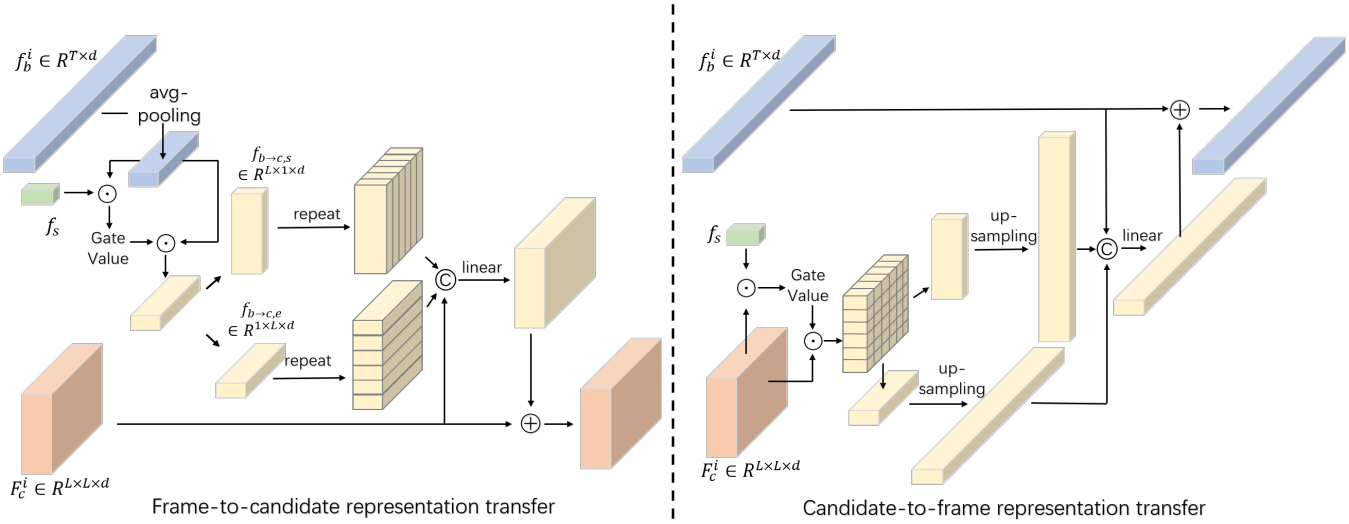

Figure 3: Illustration of the semantically conditioned interaction module. It contains a frame-to-candidate representation transfer sub-module and a candidate-to-frame representation transfer sub-module. Through this module, the candidate path offers matching information for frame representation, and the frame path offers boundary information for candidate representation. We use the query semantic gate to ensure only correlated semantic information will be transferred.

the task is to predict the start and end time of the moment described by the query.

3.2 Cross-modal Backbone

Our goal is to leverage the impact of boundary information from frame level representation and alignment information from candidate level representation when it comes to moment localization by natural language. We first generate fused features for the two paths from a cross-modal backbone.

For the natural language query, we first obtain the embedding vector of each word of the query sentence through the Glove[30] word2vec model. Then an bidirectional LSTM model[15] is utilized to extract the feature of each word, the output of LSTM is denoted as $f_w = \{w_i\}_{i=1}^{u}, w_i \in \mathcal{R}^d$, where u is the number of words in the sentence and d is the dimension of word feature. We obtain the sentence feature $f_s \in \mathcal{R}^d$ by averaging the sequence of word features f_w.

For the given input video, we first obtain a sequence of frame features from the video feature extractor backbone like VGG [34] or C3D [35]. We then uniformly sample T frames of feature to normalize the length of video features to a fixed length. Through a temporal convolution network, we obtain the sequence of video frames representation $f_v \in \mathcal{R}^{T\times d}$ for cross-modal fusion with the sentence query. We set both the dimension of the visual feature and the textual feature to d.

The task of moment localization by query requires correlated understanding between video content and language. To correlate the cross modal information and obtain the vision-language fused representation, we let each frame of visual feature interact with the sentence feature. We obtain the fused feature $\bar{f} \in \mathcal{R}^{T\times d}$ from f_v and f_s:

$$\bar{f} = f_v \odot f_s, \tag{1}$$

where $\odot$ means element-wise product. The yielded representation $\bar{f}$ captures the cross-modal correlation and facilitate the following two paths to produce accurate boundary prediction and discriminative moment candidate.

3.3 Dual Path Modeling

3.3.1 ***Frame-level Representation Path.*** In order to produce accurate boundary prediction, we need discriminative frame-level representation. We encode the frame-level boundary information in this path.

We directly use the fused feature $\bar{f} \in \mathcal{R}^{T\times d}$ as the input for this path because predicting the start and end boundary of a specific moment in a video requires temporally fine-grained feature and $\bar{f}$ with the high temporal resolution is suitable for locating the boundary region. We use the bidirectional GRU network[8] to encode the frame-level contextual information into the feature of each frame. Directly using the fused feature to predict the boundary of a specific activity is hard due to the lack of awareness of surrounding frames. The context-aware frame representation can make boundary prediction by considering surrounding frames. By stacking several GRU layers, each frame representation obtains sufficient contextual information. Specifically, we learn the frame-level representation f_b^i through the GRU model:

$$f_b^0 = \bar{f}, \tag{2}$$

$$f_b^{i+1} = GRU^{(i)}(f_b^i), \tag{3}$$

where f_b^0 is the input frame-level feature for this path, $GRU^{(i)}$ is the i_{th} GRU layer. f_b^i and f_b^{i+1} are the input and output of $GRU^{(i)}$.

Though the frame-level representation gets the surrounding information by using the GRU network, the frame-level contextual information is not enough to make an accurate prediction. Because

these frame-level features are lack of consistency since they are unaware of the content of the moment constituted by them.

3.3.2 ***Candidate-level Representation Path.*** In order to produce discriminative moment candidates, we encode candidate-level alignment information in this path.

To obtain candidate moments as the input for this path, directly using multi-scale sliding windows by scanning the high-resolution fused features $\bar{f}$ is computation-costly. Instead, we employ an average pooling layer to turn the $\bar{f}$ into a series of consecutive basic moments $\bar{f}_{base} \in \mathcal{R}^{L\times d}$, where $L \ll T$ is the numbers of basic moments. With these low-resolution basic moments, we construct the 2D candidate map $F_c^0 \in \mathcal{R}^{L\times L\times d}$ as the candidate-level representation inspired from [20, 48]. Specifically, we denote the $(i,j)_{th}$ element of F_c^0 as $F_{c_{ij}}^0 \in \mathcal{R}^d$. It represents the candidate moment feature with normalized start time of $\frac{i}{L}$ and normalized end time of $\frac{j+1}{L}$, and is constructed from the combination of basic moments range from i to j. We implement the combination operation by mean pooling. Simplicity, F_{cij}^0 is obtained from:

$$\bar{f}_{base} = AvgPooling(\bar{f}), \tag{4}$$

$$F_{c_{ij}}^0 = \frac{1}{j-i+1}\sum_{k=i}^{j} \bar{f}_{base_k}, \tag{5}$$

where $i \le j$ and $i, j \in [0, L-1]$. The left lower corner of the 2D candidate map where $i > j$ is meaningless and set to zero.

To enable each candidate be aware of the adjacent moment, we use 2D convolution network to encode the candidate-level contextual information into the feature of each candidate. We learn the candidate-level representation F_c^i:

$$F_c^{i+1} = \mathbf{ReLU}(Conv^{(i)}(F_c^i)) + F_c^i, \tag{6}$$

where $Conv^{(i)}$ is the i_{th} convolution layer, F_c^i and F_c^{i+1} are the input and output of $Conv^{(i)}$.

The context information makes the candidate related to the adjacent moment, but it is unable to provide boundary information to the candidate-level representation.

3.4 Semantically Conditioned Interaction

To improve the consistency between two kinds of representation and make sure each path only acquires query-correlated information from the other path, we propose the semantically conditioned interaction module, as illustrated in Figure 3.

3.4.1 ***Frame-to-candidate Representation Transfer.*** We transfer the discriminative frame-level representation to the candidate so that the candidate representation is aware of the boundary information. Since the temporal resolution of the frame-level feature is different from candidate-level features, we down-sample the frame feature with average pooling.

For the moment localization task, the semantic information of the query sentence plays an important role. In order to transfer only semantically correlated boundary information and avoid the original representation being drowned by the overflowed transferred information, we design the query semantic gate mechanism. With this gate function, the boundary feature highly correlated with the query will receive high gate value. We obtain the gated boundary feature $f_{b\mapsto c} \in \mathcal{R}^{L\times d}$:

$$f_b' = AvgPooling(f_b), \tag{7}$$

$$g_{b\mapsto c} = 2\sigma(f_b' \odot f_s), \tag{8}$$

$$f_{b\mapsto c} = g_{b\mapsto c} \odot f_b', \tag{9}$$

where f_b' is the down-sampled frame path feature and $g_{b\mapsto c} \in \mathcal{R}^{L\times d}$ represents the vector of gate value. σ denotes the sigmoid function and $\odot$ denotes element-wise product. The coefficient 2 ensures that the randomly initialization make the gate close to identity in the initial training phase and stabilizes the training process. We then obtain the start boundary feature $f_{b\mapsto c,s} \in \mathcal{R}^{L\times d}$and end boundary feature $f_{b\mapsto c,e} \in \mathcal{R}^{L\times d}$ by using linear function:

$$f_{b\mapsto c,s} = Linear(f_{b\mapsto c}), \tag{10}$$

$$f_{b\mapsto c,e} = Linear(f_{b\mapsto c}). \tag{11}$$

The candidate features are arranged in a 2D map with size $L \times L \times d$. The row dimension and col dimension represent the start time and end time of a specific candidate, respectively. In order to fuse the boundary feature with candidate feature, we expand the boundary feature by repeating function to have the same shape as candidate features. Specifically, for the start feature, we first obtain $f'_{b\mapsto c,s} \in \mathcal{R}^{1\times L\times d}$ by inserting 1 dimension from $f_{b\mapsto c,s}$ and then obtain the start map $F_{b\mapsto c,s} \in \mathcal{R}^{L\times L\times d}$ by repeating L times of $f'_{b\mapsto c,s}$. Similar to $F_{b\mapsto c,s}$, we obtain the end map $F_{b\mapsto c,e} \in \mathcal{R}^{L\times L\times d}$ from $f'_{b\mapsto c,e} \in \mathcal{R}^{L\times 1\times d}$, where $f'_{b\mapsto c,e}$ is acquired by inserting dimension to $f_{b\mapsto c,e}$.

After acquiring the start map and end map, we concatenate them with the candidate map. We obtain the updated candidate representation, which aggregates the boundary information through a fully connected layer:

$$F_c = \mathbf{ReLU}(Linear(concat[F_c, F_{b\mapsto c,s}, F_{b\mapsto c,e}])) + F_c. \tag{12}$$

3.4.2 ***Candidate-to-frame Representation Transfer.*** We transfer the candidate alignment information to boundary representation to make it be aware of the moment content and improve the consistency of each pair of boundary features making up a moment. The pipeline of the transferring progress is similar to frame-to-candidate representation transfer. We first obtain the gated candidate feature $F_{c\mapsto b}$ through the query semantic gate function to strengthen the candidate feature highly correlated with the query:

$$G_{c\mapsto b} = 2\sigma(F_c \odot F_s), \tag{13}$$

$$F_{c\mapsto b} = G_{c\mapsto b} \odot F_c, \tag{14}$$

where $G_{b\mapsto c} \in \mathcal{R}^{L\times L\times d}$ represents the gate value. $F_s \in \mathcal{R}^{L\times L\times d}$ is expanded and repeated from $f_s \in \mathcal{R}^d$. Since each row of the 2D candidate feature map have the same start boundary and each col have the same end boundary, we use row-pooling and col-pooling to obtain candidate start representation $f_{c\mapsto b,s} \in \mathcal{R}^{L\times d}$ and candidate end representation $f_{c\mapsto b,e} \in \mathcal{R}^{L\times d}$:

$$f_{c\mapsto b,s} = RowPooling(F_{c\mapsto b}), \tag{15}$$

$$f_{c\mapsto b,e} = ColPooling(F_{c\mapsto b}), \tag{16}$$

where row-pooling or col-pooling is a 2D avg-pooling with kernel size of $(1, L)$ or $(L, 1)$ and stride of $(1, L)$ or $(L, 1)$ with no padding,

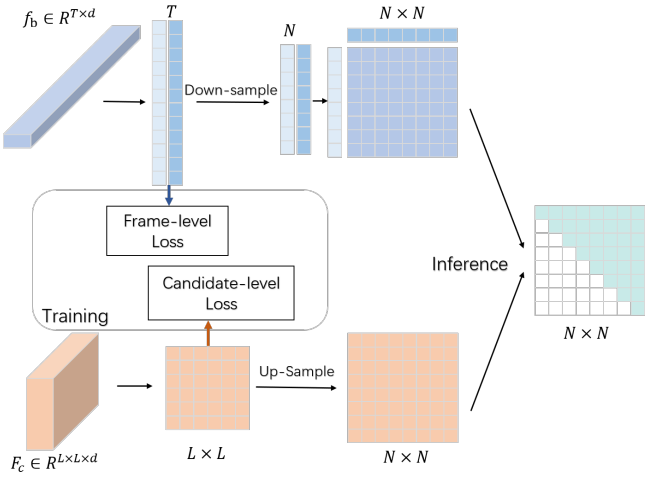

Figure 4: Illustration of training and joint prediction.

respectively. $f_{c\mapsto b,s}$ and $f_{c\mapsto b,e}$ aggregate all the candidates information with the same start or end boundary and facilitate to fuse with the boundary representation from frame path. We then up-sample $f_{c\mapsto b,s}$ and $f_{c\mapsto b,e}$ to have the same temporal resolution with boundary representation for fusion:

$$f'_{c\mapsto b,s} = UpSampling(f_{c\mapsto b,s}), \tag{17}$$

$$f'_{c\mapsto b,e} = UpSampling(f_{c\mapsto b,e}), \tag{18}$$

where $f'_{c\mapsto b,s}$ and $f'_{c\mapsto b,e}$ are the up-sampled features. We finally obtain the updated frame representation aggregating the candidate information:

$$f_b = \mathbf{ReLU}(Linear(concat[f_b, f'_{c\mapsto b,s}, f'_{c\mapsto b,e}])) + f_b. \tag{19}$$

3.5 Resolution Adaptive Joint Prediction

After we obtain the output representation from two paths, we make a joint prediction, as illustrated in Figure 4. For the frame path, we use a pair of fully connected layer to encode the start feature $f_{b,s}$ and end feature $f_{b,e}$ from the frame-level representation, respectively. We then predict the start score $score_s \in \mathcal{R}^{T\times 1}$ and end score $score_e \in \mathcal{R}^{T\times 1}$ based on them:

$$f_{b,s} = \mathbf{ReLU}(Linear(f_b)); f_{b,e} = \mathbf{ReLU}(Linear(f_b)), \tag{20}$$

$$score_s = \sigma(Linear(f_{b,s})); score_e = \sigma(Linear(f_{b,e})). \tag{21}$$

For the candidate path, we obtain the 2D alignment score prediction map $Score_c \in \mathcal{R}^{L\times L}$:

$$Score_c = \sigma(Linear(F_c)). \tag{22}$$

To obtain a joint prediction with expected temporal resolution N ($L \le N \le T$, $N = \mu L, \mu \in \mathcal{N}$), we first down-sample the start and end boundary score prediction and combine them into 2D boundary prediction map $Score'_b \in \mathcal{R}^{N\times N}$:

$$score'_s = DownSampling(score_s); \tag{23}$$

$$score'_e = DownSampling(score_e), \tag{24}$$

$$Score'_b = score'_s \times (score'_e)^\top. \tag{25}$$

We then up-sample the 2D alignment score prediction map to the size of $N \times N$:

$$Score'_c = UpSampling(Score_c). \tag{26}$$

We finally obtain the prediction of each candidate by fusing them:

$$Score = Score'_c \cdot Score'_b. \tag{27}$$

3.6 Training and Loss Function

We use frame loss and candidate loss to train the frame-level and candidate-level features, respectively. For frame path, we use KL divergence (KLD) loss considering the whole distribution of all the frames and binary cross entropy (BCE) loss considering each frame separately. With the normalized ground-truth annotation(τ_s, τ_e) $(\tau_s, \tau_e \in [0,1])$, the KLD label $y_{kld,s} \in \mathcal{R}^T$ or $y_{kld,e} \in \mathcal{R}^T$ is generated by an unnormalized 1D Gaussian $e^{-\frac{x^2}{2\sigma^2}}$ inspired by [19], whose center is at τ_s or τ_e and whose σ is set to $(\tau_e - \tau_s)/5$. It gives fewer penalties to the frames near the temporal boundaries. The BCE label $y_{bce,s} \in \mathcal{R}^T$ or $y_{bce,e} \in \mathcal{R}^T$ is gained from $y_{kld,s}$ or $y_{kld,e}$ through a threshold of 0.5 (i.e., $y_{bce} = \mathbf{1}(y_{kld} \ge 0.5)$). We obtain the frame loss:

$$Loss_{KLD} = KLD(score_s||y_{kld,s}) + KLD(score_e||y_{kld,e}), \tag{28}$$

$$Loss_{BCE} = BCE(score_s, y_{bce,s}) + BCE(score_e, y_{bce,e}), \tag{29}$$

$$Loss_f = \lambda_1 Loss_{KLD} + \lambda_2 Loss_{BCE}. \tag{30}$$

For candidate path, we use binary cross entropy (BCE) loss and iou regression (REG) loss inspired by [20]. REG loss is implemented by mean square error loss. The REG label $y_{reg} \in \mathcal{R}^{T\times T}$ is the iou value of each candidate and BCE label $y_{bce,c} \in \mathcal{R}^{T\times T}$ is gained from y_{reg}through a threshold of 0.5. We obtain the candidate loss:

$$Loss_c = BCE(Score_c, y_{bce,c}) + \lambda_3 REG(Score_c, y_{reg}). \tag{31}$$

The total loss is the sum of frame-level loss and candidate-level loss. $\lambda_1, \lambda_2, \lambda_3$ are set to 10000, 10, 10, respectively.

$$Loss_{total} = Loss_f + Loss_c. \tag{32}$$

4 EXPERIMENTS

In this section, we conduct extensive experiments on three benchmarks to evaluate our method.

4.1 Implementation Details

We use the sequence of visual features extracted offline as input. For a fair comparison, we adopt the same visual features as previous work [48]. Specifically, for TACoS and Activity-Caption dataset, we use C3D feature, and for the Charades-STA dataset, we use VGG16[34] feature provided from the original Charades dataset [33]. We sample the feature sequence to length $T = 200$ as the input for all the three datasets. For the language query, we employ the pre-trained Glove [30] word2vec model to extract the embedding features for each word with the dimension of 300. Each sentence is truncated to have a maximum length of 14,13,20 words for TACoS, Charades-STA, and Activity-Caption, respectively. For the backbone, we use three layers of 1D convolution with a kernel size of 3 and padding size of 1 and with identity connection [13] to extract visual feature. We use 1 layer of bidirectional LSTM with hidden size of 512 to get the word feature. The feature dimension d is set to 512

Table 1: Result of R@$\{1, 5\}$ with IoU=$\{0.1, 0.3, 0.5\}$ on the TACoS dataset.

Method	R@1 0.1	R@1 0.3	R@1 0.5	R@5 0.1	R@5 0.3	R@5 0.5
CTRL [9]	24.32	18.32	13.30	48.73	36.69	25.42
MCF [39]	25.84	18.64	12.53	52.96	37.13	24.73
ACRN [24]	24.22	19.52	14.62	47.42	34.97	24.88
SAP [7]	31.15	-	18.24	53.51	-	28.11
SM-RL [37]	26.51	20.25	15.95	50.01	38.47	27.84
CMIN [50]	32.48	24.64	18.05	62.13	38.46	27.02
ABLR [44]	34.70	19.50	9.40			
ACL [10]	-	24.17	20.01	-	42.15	30.66
GDP [6]	39.68	24.14	-	-	-	-
ExCL [11]	45.50	28.00	13.80	-	-	-
TGN [4]	41.87	21.77	18.90	53.40	39.06	31.02
CBP [36]	-	27.31	24.79	-	43.64	37.40
SCDM [43]	-	26.11	21.17	-	40.16	32.18
2D-TAN [48]	47.59	37.29	25.32	70.31	57.81	45.04
DPIN (ours)	**59.04**	**46.74**	**32.92**	**75.78**	**62.16**	**50.26**

Table 2: Result of R@$\{1, 5\}$ with IoU=$\{0.5, 0.7\}$ on the Charades-STA dataset.

Method	R@1 IoU=0.5	R@1 IoU=0.7	R@5 IoU=0.5	R@5 IoU=0.7
ROLE [25]	12.12	-	40.59	-
CTRL [9]	21.42	7.15	59.11	26.91
SLTA [16]	22.81	8.25	72.39	31.46
SMRL [37]	24.36	11.17	61.25	32.08
ACL [10]	30.48	12.20	64.84	35.13
Xu *et al.* [41]	35.60	15.80	79.40	45.40
GDP [6]	39.47	18.49	-	-
CBP [36]	36.80	18.87	70.94	50.19
2D-TAN [48]	39.70	23.31	80.32	51.26
MAN [47]	41.24	20.54	83.21	51.85
DPIN (ours)	**47.98**	**26.96**	**85.53**	**55.00**

Table 3: Result of R@$\{1, 5\}$ with IoU=$\{0.3, 0.5, 0.7\}$ on the Activity-Caption dataset.

Method	R@1 0.3	R@1 0.5	R@1 0.7	R@5 0.3	R@5 0.5	R@5 0.7
TGN [4]	43.81	27.93	11.86	54.56	44.20	24.84
Xu *et al.* [41]	45.30	27.70	13.60	75.70	59.20	38.30
ABLR [44]	55.67	36.79	-	-	-	-
DEBUG [27]	55.91	39.72	-	-	-	-
GDP [6]	56.17	39.27	-	-	-	-
CBP [36]	54.30	35.76	17.80	77.63	65.89	46.20
ExCL [11]	62.10	41.60	23.90	-	-	-
CMIN [50]	**63.61**	43.40	23.88	80.54	67.95	50.73
2D-TAN [48]	59.45	44.51	26.54	85.53	77.13	**61.96**
DPIN (ours)	62.40	**47.27**	**28.31**	**87.52**	**77.45**	60.03

for all the features in the model. We use 3 GRU layers in the frame-level path and 3 convolution layers in the candidate-level path. We insert one semantically conditioned interaction module(SCIM) between every two layers, which results in the total number of SCIM is 2. We set $L = 10$ for Charades-STA and Activity-Caption, and $L = 50$ for TACoS. The batch-size is set to 64 and the learning rate is set to 0.0005. All the parameters of the DPIN are optimized simultaneously by Adam[17] optimizer.

4.2 Datasets

TACoS. TACoS [31] consists of 127 videos and 17,344 text-to-clip pairs. It contains different activities happened in kitchen room. We use the standard split as [9], i.e., 50% for training, 25% for validation, and 25% for test.

Charades-STA. Charades-STA [9] contains 16,128 pairs of sentence-moment with 12,408 in the training set and 3,720 in the testing set. Charades-STA was built on Charades [33] and the temporal sentence annotations were generated by Gao *et al.* [9].

Activity-Caption. Activity-Caption [18] was built on ActivityNet v1.3 dataset [14] with diverse context. Following [48, 50], we use val_1 as validation set and val_2 as testing set. We have 37, 417, 17, 505, and 17, 031 moment-sentence pairs for training, validation, and testing, respectively.

4.3 Performance Comparison

Following [9], we use the R@n IoU=m metric as the evaluation metric. It is the percentage that at least one of the candidate moments with top-n scores have Intersection over Union (IoU) larger than m. We report the result of $n \in \{1, 5\}$ with $m \in \{0.1, 0.3, 0.5\}$ for TACoS, $n \in \{1, 5\}$ with $m \in \{0.5, 0.7\}$ for Charades-STA, and $n \in \{1, 5\}$ with $m \in \{0.3, 0.5, 0.7\}$ for Activity-Caption, respectively. We evaluate our proposed DPIN approach on three datasets and compare our model with the state-of-the-art methods, including: *Candidate-based (top-down) approaches*: CTRL [9], MCF [39], ACRN [24], SAP [7], CMIN [50], ACL [10], SCDM [43], ROLE [25], SLTA [16],MAN [47], Xu *et al.* [41], SCDM [43], 2D-TAN [48]. *Frame-based (Bottom-up) approaches*: ABLR [44],GDP [6], TGN [4], CBP [36], ExCL [11], DEBUG [27]. *Reinforcement learning approach*: SM-RL [37].

From the results in Table 1, Table 2, and Table 3, we can observe that our proposed DPIN approach outperforms other methods by clear margins, which demonstrates the superiority of our proposed model. Specifically, on the TACoS dataset, as shown in Table 1, our approach significantly outperforms 2D-TAN [48], which achieves the best performance among other approaches, with 9.45% and 7.60% absolute improvements in the R@1,IoU=0.3 and R@1,IoU=0.5 metrics, respectively. On the Charades-STA dataset, as shown in Table 2, our approach significantly surpasses the state-of-the-art method MAN [47] by more than 6% in term of R@1,IoU=0.5 and R@1,IoU=0.7 metrics, respectively. On the Activity-Caption dataset, as shown in Table 3, our approach outperforms the state-of-the-art methods with approximately 3% and 2% performance improvement with respect to R@1,IoU=0.5 and R@1,IoU=0.7, respectively. It shows our approach effectively localizes the activity moment.

Moreover, through the comparison of our method with candidate-based top-down approaches (CTRL [9], ACRN [24], ACL [10], MAN

Table 4: Component ablation results on Charades-STA.

Method	R@1 IoU=0.5	R@1 IoU=0.7	R@5 IoU=0.5	R@5 IoU=0.7
candidate path	43.84	21.91	82.82	51.68
frame path	41.36	20.28	77.42	47.17
c+f	45.53	24.83	83.65	52.52
c+f with c2f	46.88	25.43	84.73	54.70
c+f with f2c	46.65	25.51	85.18	54.11
Full (c+f with c2f+f2c)	**47.98**	**26.96**	**85.53**	**55.00**
Full w/o gate	47.25	26.31	83.95	53.81
Full w/o $Loss_{KLD}$	45.64	24.91	84.53	53.54
Full w/o $Loss_{REG}$	47.58	25.43	84.46	54.67

Table 5: Comparison of output resolution.

Method	TACoS R@1 0.5	TACoS R@1 0.7	TACoS R@5 0.5	TACoS R@5 0.7	Activity-Caption R@1 0.5	Activity-Caption R@1 0.7	Activity-Caption R@5 0.5	Activity-Caption R@5 0.7
N=L	32.92	16.62	**50.26**	**32.24**	47.27	28.31	**77.45**	**60.03**
N=2L	**34.71**	**19.22**	47.99	31.34	**47.44**	**29.55**	74.33	56.44

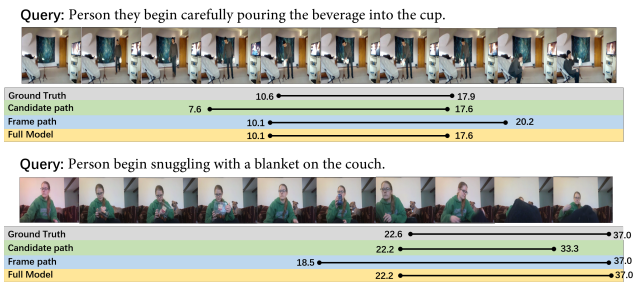

Figure 5: Visualization of the output localization results.

[47], 2D-TAN [48]), we can observe that our method performs better than them. The main reason is that these methods only use candidate-level representation to localize a moment without precise boundary information. Though some of them use an additional regression layer to predict the offsets, their candidate-level feature is not suitable for boundary-level regression and result in inferior performance. On the other hand, by comparing our method with frame-based bottom-up approaches (DEBUG [27], TGN [4], CBP [36], GDP [6]), we can observe that our method works better. Since these approaches only use frame-level representation for moment localization, the boundary features are unaware of the moment content they constitute and lack of consistency, which results in poor performance. Our approach consistently outperforms the candidate-based top-down approaches and frame-based bottom-up approaches because our approach considers the impact of both candidate-level and frame-level representations. We encode them in two paths and allow them to interact with each other to exploit two different and complementary representations. Based on the two representations, our approach can make the joint prediction highly consistent with both the query semantics and moment boundaries.

4.4 Ablation Studies

To evaluate the contribution of each component, we conduct ablation studies on the Charades-STA dataset, and the ablation study results are shown in Table 4. We first set the baseline as the model that only has one path of frame-level representation or candidate-level representation and without the interaction module in Row 1-2. The candidate-level representation works a little better than the frame-level representation since it encodes the alignment information. We then show the result of the joint prediction of two paths, which is denoted as "c+f" in Row 3. We observe combining two kinds of different yet complementary representation surpasses using only one kind of representation since it considers the benefits of both candidate-level and frame-level representation together. Row 4-5 are the results of the joint prediction of two paths with only the candidate-to-frame transfer or frame-to-candidate transfer, respectively. The results show that the interaction module improves the performance and the transferred information in each direction helps the feature in the other path. We show the result of the full model in Row 6 and the full model without the semantic gate mechanism in Row 7. We can conclude that combining candidate-to-frame transfer and frame-to-candidate transfer can further improve the performance since it enables both paths to be aware of the information from the other path. By comparing Row 6 and Row 7, we demonstrate that the gate mechanism is important since it ensures that each path only aggregates the helpful information highly correlated to the query. Row 8-9 are the results without KLD or REG loss, which show that these loss terms help supervise the model. We also show the comparison result of output resolution on TACoS and Activity-Caption in Table 5. The resolution adaptive prediction strategy reduces the computation cost and results in more precise prediction for R@1 on both datasets. The R@5 of $N = 2L$ is lower than $N = L$ because high-resolution prediction results in more candidates. We can adaptively choose the output temporal resolution according to the requirement of the application.

4.5 Visualization Results

We visualize the localization result obtained from the DPIN model, as shown in Figure. 5. We can observe that the proposed model performs well to output the moment matched with the query.

5 CONCLUSIONS

In this paper, we propose a unified top-down and bottom-up approach called Dual Path Interaction Network (DPIN) for the task of temporally localizing the moment depicted by a sentence query. We encode the frame-level and candidate-level representations in two paths to keep them different but complementary. We propose a semantically conditioned interaction module to improve the consistency between these two kinds of representation. We propose a resolution-adaptive joint prediction module to output prediction with expected resolution. As a result, the proposed DPIN outperforms the state-of-the-art approaches on three benchmarks.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China award 2016YFB1001402 and National Natural Science Foundation of China award U19B2038,61620106009.

REFERENCES

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *ICCV*.

[2] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. 2017. SST: Single-Stream Temporal Action Proposals. In *CVPR*.

[3] Yuwei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. 2018. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. In *CVPR*.

[4] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. 2018. Temporally grounding natural sentence in video. In *EMNLP*.

[5] Jingyuan Chen, Lin Ma, Xinpeng Chen, Zequn Jie, and Jiebo Luo. 2019. Localizing natural language in videos. In *AAAI*.

[6] Long Chen, Chujie Lu, Siliang Tang, Jun Xiao, Dong Zhang, Chilie Tan, and Xiaolin Li. 2020. Rethinking the Bottom-Up Framework for Query-based Video Localization. In *AAAI*.

[7] Shaoxiang Chen and Yu-Gang Jiang. 2019. Semantic Proposal for Activity Localization in Videos via Sentence Query. In *AAAI*.

[8] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *EMNLP*.

[9] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *ICCV*.

[10] Runzhou Ge, Jiyang Gao, Kan Chen, and Ram Nevatia. 2019. MAC: Mining Activity Concepts for Language-based Temporal Localization. In *WACV*.

[11] Soham Ghosh, Anuva Agarwal, Zarana Parekh, and Alexander Hauptmann. 2019. ExCL: Extractive Clip Localization Using Natural Language Descriptions. *arXiv preprint arXiv:1904.02755* (2019).

[12] Dongliang He, Xiang Zhao, Jizhou Huang, Fu Li, Xiao Liu, and Shilei Wen. 2019. Read, Watch, and Move: Reinforcement Learning for Temporally Grounding Natural Language Descriptions in Videos. *arXiv preprint arXiv:1901.06829* (2019).

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

[14] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*.

[15] S Hochreiter and J Schmidhuber. [n.d.]. Long Short-Term Memory. *Neural Computation* 9, 8 ([n. d.]), 1735–1780.

[16] Bin Jiang, Xin Huang, Chao Yang, and Junsong Yuan. 2019. Cross-Modal Video Moment Retrieval with Spatial and Language-Temporal Attention. In *ICMR*.

[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *ICCV*.

[19] Hei Law and Jia Deng. 2018. CornerNet: Detecting Objects as Paired Keypoints. In *ECCV*.

[20] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. 2019. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In *ICCV*.

[21] Bingbin Liu, Serena Yeung, Edward Chou, De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. 2018. Temporal modular networks for retrieving complex compositional activities in videos. In *ECCV*.

[22] Daqing Liu, Zheng-Jun Zha, Hanwang Zhang, Yongdong Zhang, and Feng Wu. 2018. Context-Aware Visual Policy Network for Sequence-Level Image Captioning. In *ACM MM*.

[23] Daqing Liu, Hanwang Zhang, Zheng-Jun Zha, and Feng Wu. 2019. Learning to Assemble Neural Module Tree Networks for Visual Grounding. In *ICCV*.

[24] Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua. 2018. Attentive moment retrieval in videos. In *SIGIR*.

[25] Meng Liu, Xiang Wang, Liqiang Nie, Qi Tian, Baoquan Chen, and Tat-Seng Chua. 2018. Cross-modal moment localization in videos. In *ACM MM*.

[26] Xuejing Liu, Liang Li, Shuhui Wang, Zheng-Jun Zha, Dechao Meng, and Qingming Huang. 2019. Adaptive Reconstruction Network for Weakly Supervised Referring Expression Grounding. In *ICCV*.

[27] Chujie Lu, Long Chen, Chilie Tan, Xiaolin Li, and Jun Xiao. 2019. DEBUG: A Dense Bottom-Up Grounding Approach for Natural Language Video Localization. In *EMNLP*.

[28] Alberto Montes, Amaia Salvador Aguilera, Santiago Pascual, and Xavier Giro I Nieto. 2016. Temporal activity detection in untrimmed videos with recurrent neural networks. In *NIPS*.

[29] Cristian Rodriguez Opazo, Edison Marresetaylor, Fatemeh Sadat Saleh, Hongdong Li, and Stephen Gould. 2020. Proposal-free Temporal Moment Localization of a Natural-Language Query in Video using Guided Attention.. In *WACV*.

[30] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

[31] Michaela Regneri, Marcus Rohrbach, Dominikus Wetzel, Stefan Thater, Bernt Schiele, and Manfred Pinkal. 2013. Grounding Action Descriptions in Videos. *Transactions of the Association for Computational Linguistics* 1 (2013), 25–36.

[32] Zheng Shou, Dongang Wang, and Shihfu Chang. 2016. Temporal Action Localization in Untrimmed Videos via Multi-stage CNNs. In *CVPR*.

[33] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *ECCV*.

[34] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[35] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

[36] Jingwen Wang, Lin Ma, and Wenhao Jiang. 2020. Temporally Grounding Language Queries in Videos by Contextual Boundary-aware Prediction. In *AAAI*.

[37] Weining Wang, Yan Huang, and Liang Wang. 2019. Language-Driven Temporal Activity Localization: A Semantic Matching Reinforcement Learning Model. In *CVPR*.

[38] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)* 38, 5 (2019), 146.

[39] Aming Wu and Yahong Han. 2018. Multi-modal Circulant Fusion for Video-to-Language and Backward. In *IJCAI*.

[40] Huijuan Xu, Abir Das, and Kate Saenko. 2017. R-C3D: Region Convolutional 3D Network for Temporal Activity Detection. In *ICCV*.

[41] Huijuan Xu, Kun He, Bryan A Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. 2019. Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*.

[42] Tianhao Yang, Zheng-Jun Zha, and Hanwang Zhang. 2019. Making History Matter: History-Advantage Sequence Training for Visual Dialog. In *ICCV*.

[43] Yitian Yuan, Lin Ma, Jingwen Wang, Wei Liu, and Wenwu Zhu. 2019. Semantic Conditioned Dynamic Modulation for Temporal Sentence Grounding in Videos. In *NIPS*.

[44] Yitian Yuan, Tao Mei, and Wenwu Zhu. 2019. To find where you talk: Temporal sentence localization in video with attention based location regression. In *AAAI*.

[45] Zheng-Jun Zha, Daqing Liu, Hanwang Zhang, Yongdong Zhang, and Feng Wu. 2020. Context-Aware Visual Policy Network for Fine-Grained Image Captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), 1–1.

[46] Zheng-Jun Zha, Jiawei Liu, Tianhao Yang, and Yongdong Zhang. 2019. Spatiotemporal-Textual Co-Attention Network for Video Question Answering. *ACM Transactions on Multimedia Computing, Communications, and Applications* 15, 2 (2019), 53.

[47] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. 2019. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *CVPR*.

[48] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. 2020. Learning 2D Temporal Adjacent Networks for Moment Localization with Natural Language. In *AAAI*.

[49] Songyang Zhang, Jinsong Su, and Jiebo Luo. 2019. Exploiting Temporal Relationships in Video Moment Localization with Natural Language. In *ACM MM*.

[50] Zhu Zhang, Zhijie Lin, Zhou Zhao, and Zhenxin Xiao. 2019. Cross-Modal Interaction Networks for Query-Based Moment Retrieval in Videos. In *SIGIR*.

[51] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. 2017. Temporal Action Detection with Structured Segment Networks. In *ICCV*.