A Simple Yet Effective Method for Video Temporal Grounding with Cross-Modality Attention

Binjie Zhang[1]*, Yu Li[2], Chun Yuan[1], Dejing Xu[2], Pin Jiang[3], Ying Shan[2]
[1]Tsinghua Shenzhen International Graduate School
[2]Applied Research Center (ARC), Tencent PCG
[3]Tianjin University zbj19@mails.tsinghua.edu.cn, yuanc@sz.tsinghua.edu.cn, jpin@tju.edu.cn, {ianyli,djxu,yingshan}@tencent.com

Abstract

The task of language-guided video temporal grounding is to localize the particular video clip corresponding to a query sentence in an untrimmed video. Though progress has been made continuously in this field, some issues still need to be resolved. First, most of the existing methods rely on the combination of multiple complicated modules to solve the task. Second, due to the semantic gaps between the two different modalities, aligning the information at different granularities (local and global) between the video and the language is significant, which is less addressed. Last, previous works do not consider the inevitable annotation bias due to the ambiguities of action boundaries. To address these limitations, we propose a simple two-branch Cross-Modality Attention (CMA) module with intuitive structure design, which alternatively modulates two modalities for better matching the information both locally and globally. In addition, we introduce a new task-specific regression loss function, which improves the temporal grounding accuracy by alleviating the impact of annotation bias. We conduct extensive experiments to validate our method, and the results show that just with this simple model, it can outperform the state of the arts on both Charades-STA and ActivityNet Captions datasets.

1 Introduction

Vision and language are two of the most important representations of information. With the development of computer vision and natural language processing, multi-modality tasks have also drawn increasing attention, such as video understanding (Ma et al. 2019; Zolfaghari, Singh, and Brox 2018), video caption (Zhou et al. 2018a; Iashin and Rahtu 2020), text-to-video retrieval (Mithun et al. 2018; Chen et al. 2020) and video question answer (Cadene et al. 2019; Gao et al. 2019). The core problem among these tasks involves the multi-modality fusion and interaction, which still needs to be resolved due to the semantic gaps.

In this paper, we focus on Video Temporal Grounding (VTG) using a language query, which is first proposed by (Gao et al. 2017). Formally, as illustrated in Figure 1, given an untrimmed video and a language query, we need to localize the particular video clip, which contains the same semantic information corresponding with the query.

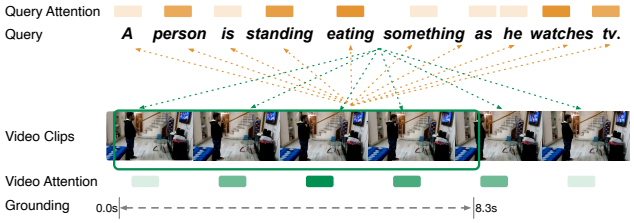

Figure 1: Illustration of Video Temporal Grounding with cross-modality attention. The particular video clip is selected with the guidance of the query, where the orange blocks show the attention value of the middle video clip for each word, and the green ones reflect the relations between the target word "eat" and each video clip.

Generally, localizing the video clip with a query contains three phases: input embeddings for video and query, multi-modality fusion, and localization. The multi-modality fusion methods are quite diverse and with no common baseline. We can briefly group these methods into attention-free (Gao et al. 2017) and attention-based (Yuan, Mei, and Zhu 2019; Yuan et al. 2019; Wang, Ma, and Jiang 2020). Gao *et al*. project video and text features into two global vectors and then use the fully-connected layer to get the fused features (Gao et al. 2017). It suffers from a lack of contextual information due to shallow fusion.

As the attention mechanism is proved as an efficient method for feature extraction, Yuan *et al*. use a multi-modality co-attention mechanism to generate attention attended with a global vector of another modality (Yuan, Mei, and Zhu 2019). Using a global vector as the representation of the query modality suffers the lack of local information when tackling with complex sentences. Mun *et al*. propose a Sequential Query Attention Network to extract distinct phrases from the sentence, which contains more fine-grained semantics, and fuse the video with multiple phrases step by step. However, the fact is ignored that the video can also select and modulate the query sentence.

While promising results have been achieved, there are still some challenges unsolved: 1. Although existing methods are getting higher performance, their modals become more

*This work was done when Binjie's intern at Tencent.

and more complicated. 2. Considering the gaps between the vision and the language, aligning them both locally and globally need to be carefully conducted. 3. We notice the existing datasets contain inevitable annotation bias due to the ambiguity of action boundaries. The occurrence of action is a gradual process, including beginning, middle, and end. However, this continuous transition is not reflected in the action boundary which are just the start and end points.

Usually, videos contain richer information than queries in the VTG task, so it is hard to capture the desired content with just one interaction. Inspired by the effectiveness of the coarse-to-fine strategy in searching for the answer to the question in long documents (Choi et al. 2017), we conduct global and local interactions step by step to dig video information. In detail, we rst take a glimpse of the query, choose diverse semantic phrases, and then browse roughly the video sequence under the guide of phrases. When we ground the relative segments, details need to be compared carefully to nd the precise location. Specifically, we propose a coarse-to-fine encoder-decoder structure for VTG with Cross-Modality Attention (CMA), as depicted in Figure 2. In the encoder layer, video features are adjusted by diverse query guide vectors, which are extracted by our Semantic Phrase Extracting (SPE) Network, using multi-head self-attention. So the coarse-grained features are captured. In the decoder layer, the whole video sequence and all query words are refined alternately using self-attention and bi-attention mechanisms. Query features guide the video to focus on the main content. Meanwhile, video features highlight the decisive words in the query sentence, making the details in the two modalities align well.

Methods to relate the fused modalities with the temporal location can be divided into two categories: anchor-based (Yuan et al. 2019; Wang, Ma, and Jiang 2020) and regression-based (Yuan, Mei, and Zhu 2019; Zeng et al. 2020; Mun, Cho, and Han 2020). Anchor-based methods calculate scores for pre-defined anchors and select the top recall anchors as the final output after using NMS (Neubeck and Van Gool 2006). To avoid the redundant calculation, we adopt the regression-based method to predict the temporal location, and design a task-specific regularizer to minimize the effect of inherent annotation bias. Our model can tackle VTG efficiently through end-to-end training.

The main contributions of our work are four-fold:

- We propose a simple pipeline for VTG using cross-modality attention with fewer parameters.
- We use a coarse-to-fine strategy to dig and align details in both videos and sentences.
- To our best knowledge, We first notice that datasets exist inevitable bias due to the ambiguity of action boundaries. To reduce the impact of annotation bias, we propose a new task-specific regression loss, which is robust for small location errors.
- We conduct extensive experiments to validate the effectiveness of our method and show that it can outperform the state of the arts on both Charades-STA and ActivityNet Captions datasets.

2 Related Work

In this section, we introduce several works in video temporal grounding, mainly focusing on different fusion methods and grounding strategies. Since the encoder-decoder structure adopted in the proposed method is widely used, we also list some related works from other vision-and-language tasks.

2.1 Video Temporal Grounding

Video temporal grounding is to localize the particular video clip guided by a language query. Early work (Anne Hendricks et al. 2017) uses the scan and localize framework, where the specific clip is retrieved by scanning the whole video using sliding windows. Specifically, Hendricks *et al*. build the Moment Context Network (MCN) to directly learn the correlation between the different video segments and the query (Anne Hendricks et al. 2017). This scheme is time-consuming and contains redundant calculations. In contrast, Gao *et al*. extract two global vectors for the video and the query, then fuse them by a Fully-Connected (FC) layer (Gao et al. 2017). Since the FC layer is not enough to obtain the in-depth contextual information, attention mechanism are also incorporated in later methods (Liu et al. 2018; Yuan, Mei, and Zhu 2019; Yuan et al. 2019).

The concrete localization methods mainly belong to two categories, anchor-based (Wang, Ma, and Jiang 2020; Yuan et al. 2019), and anchor-free (Gao et al. 2017; Zeng et al. 2020; Rodriguez et al. 2020). Similar to two-stage object detection[], Yuan *et al*. score for each pre-defined anchor and use NMS to select the top anchor as the prediction (Yuan et al. 2019). Its noticeable that anchor-based methods neglect time dependencies across video clips and exist redundant calculations. To incorporate the temporal information between different clips, Yuan *et al*. and Mun *et al*. directly regresses the coordinates using MLP (Yuan, Mei, and Zhu 2019; Mun, Cho, and Han 2020). Inspired by FCOS (Tian et al. 2019), Zeng *et al*. design a dense regression network (DRN) with three heads to regress the distances from each frame to the start and end time (Zeng et al. 2020).

2.2 Encoder-Decoder Architecture

Videos and queries are both sequential, so the Seq2Seq models (Hochreiter and Schmidhuber 1997; Vaswani et al. 2017) are suitable for dealing with the VTG task. Wang *et al*. use Match-LSTM to aggregate contextual information by modeling the correlation between each frame and its neighbors (Wang, Ma, and Jiang 2020). However, LSTMs can not handle the long sequence dependency due to the information attenuation during forward propagation. Unlike previously described sequence-to-sequence models, Transformer uses the attention mechanism to calculate the whole sequence's weights for each element. Not only the long temporal dependency is solved, but also the processing speeds up because of parallel computation (Dai et al. 2019).

Promising performances are achieved with the Transformer in other multi-modality tasks, such as action recognition (Girdhar et al. 2019), video captioning (Zhou et al. 2018b) and video question answering (Yang et al. 2020). Girdhar *et al*. propose an action transformer to recognize and

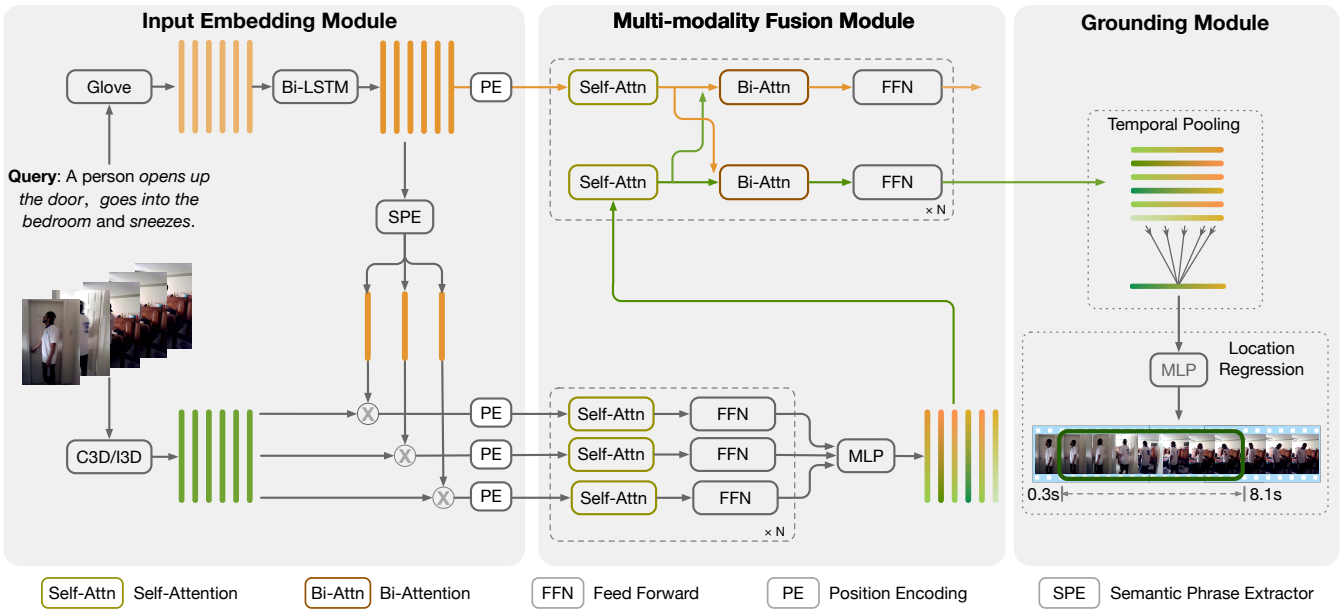

Figure 2: The framework of our proposed method, which contains three main components. (1) Input embedding module encodes the original video and query. (2) Multi-modality fusion module fuses the information from two modalities. (3) Grounding module regresses the temporal location using the aligned features.

localize human actions in video clips, where contextual information from other humans and objects in the neighborhood frames is aggregated (Girdhar et al. 2019). Zhou *et al*. design a dense video captioning model with Transformer (Zhou et al. 2018b). Specifically, the encoder encodes the video and detects the potential events; the decoder generates the descriptions using the output of the encoder. Yang *et al*. jointly model the visual concepts and subtitles with a Transformer (Yang et al. 2020).

Following the above works, we design an encoder-decoder transformer with cross-modality attention to fuse the multi-modal information for the VTG task.

3 Methods

In this section, we introduce our main framework for video temporal grounding, as shown in Figure 2. Our model consists of three primary components: the input embedding, the multi-modality fusion, and the grounding modules.

3.1 Problem Definition

Given an untrimmed video X and a language query Q, Video Temporal Grounding (VTG) is to localize the start time τ^s and end time τ^e of X corresponding to the Q.

$$(\tau^s, \tau^e) = f_\theta(X, Q). \tag{1}$$

3.2 Input Embedding

Video Embedding Generally, static spatial information and dynamic temporal contents are both crucial for video understanding. Following the previous works (Yuan, Mei, and Zhu 2019; Yuan et al. 2019; Mun, Cho, and Han 2020), we use 3D Conv. Network (Tran et al. 2015; Carreira and Zisserman 2017), denoted by $f_v(\cdot)$, to extract the spatio-temporal features from original video sequence X.

Notice that the attention mechanism is permutation invariant (Vaswani et al. 2017), so the Position Encoding (PE), denoted by $f_{PE}(\cdot)$, is crucial to keep the sequential information. Two different PE methods are compared in this paper: constant sine/cosine position encoding and learnable position embedding. More details are showed in Section 4.5. The embedded video features $V^{in} = \{v_1^{in}, v_2^{in}, \ldots, v_N^{in}\}$ can be represented as

$$V^{in} = ReLU(W_v f_v(X)) + f_{PE}([1, 2, \ldots, N]), \tag{2}$$

where $W_v \in \mathbb{R}^{d \times d_v}, V^{in} \in \mathbb{R}^{d \times N}$. d and d_v are the dimensions of embedded video features and C3D features, respectively. N represents the number of video clips.

Query Embedding Each word in the query $Q = \{w_1, w_2, \ldots, w_L\}$ is embedded to a 300D vector using pre-trained Glove (Pennington, Socher, and Manning 2014) word embedding. After getting the static word embedding, a bi-directional LSTM is conducted to encode contextual information h_i. We denote the embedded query sentence $Q^{in} \in \mathbb{R}^{d_q \times L}$ as the concatination of the forward LSTM and backward LSTM. The global sentence embedding s_{global} consists of the concatination of the last h_L^f and the first h_1^b hidden states of forward and backward LSTMs.

$$h_1^f, h_2^f, \ldots, h_L^f = LSTM^f(w_1, w_2, \ldots, w_L), \tag{3}$$

$$h_1^b, h_2^b, \ldots, h_L^b = LSTM^b(w_1, w_2, \ldots, w_L), \tag{4}$$

$$Q^{in} = [h_i^f; h_i^b] + f_{PE}([1, 2, \ldots, L]), \tag{5}$$

$$s_{global} = [h_L^f; h_1^b], \tag{6}$$

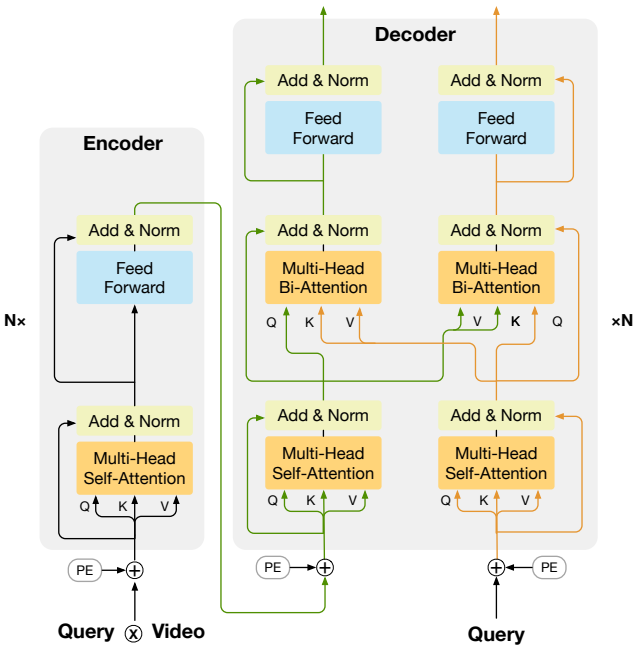

Figure 3: Details of multi-modality fusion module. (a) The encoder coarsely adjust the video features guided by query guide vectors; (b) The decoder consists of two cross-modality branches (left is the video branch and right is query branch), where two modalities are aligned alternatively.

where L is the length of the query sentence, and $[\cdot;\cdot]$ represents the concatenation operation. The dimension of the embedded query feature is d_q.

Semantic Phrase Extracting A complex query sentence usually contains multiple semantic phrases, *e.g.*, the sentence "A person *opens up the door*, *goes into the bedroom* and *sneezes*" consists of three main actions. Inspired by the Sequential Query Attention Network (SQAN) (Mun, Cho, and Han 2020), we propose a Semantic Phrase Extracting Network (SPE), with attention following (Lin et al. 2017). With SPE, we extract multiple query guide vectors, denoted by $G = [g_{(1)}; g_{(2)}; \ldots; g_{(k)}]$ from Q^{in}. These vectors coarsely lead the video to highlight different essential information.

$$A_{spe} = softmax(W_{s2}(tanh(W_{s1}Q^{in}))), \quad (7)$$

$$G = Q^{in}A_{spe}^{\top}, \quad (8)$$

where $W_{s1} \in \mathbb{R}^{d_s \times d}$, $W_{s2} \in \mathbb{R}^{k \times d_s}$, $A_{spe} \in \mathbb{R}^{k \times L}$,$G \in \mathbb{R}^{d \times k}$, k denotes the number of phrases.

Note that the query guide vector is the same as s_{global} when we choose a single phrase as the input of the encoder.

3.3 Multi-modality Fusion

The proposed cross-modality attention mechanism plays an essential role in aligning the local and global information, as depicted in Figure 3. Specifically, we adopt an encoder-decoder structure, where the video features V^{in} are adjusted roughly by query guide vectors $g^{(i)}$ in the encoder, and local contents of video segments and query words are alternatively updated in the decoder.

Encoder To correlate the semantic information of different modalities, we combine video features V^{in} with each query guide vector $g^{(i)}$ to obtain the coarse video features F^{in}. Here we compare three different fusion methods, represented as $f_u(\cdot,\cdot)$, Hadamard Product, Concatenation, and Addition. More details refer to Section 4.5.

$$F^{in}_{(i)} = f_u(V^{in}, g_{(i)}). \quad (9)$$

The yielded representation $F^{in}_{(i)}$ is fed into a multi-head self-attention (SA), which is used to capture the interactions between two modalities in a coarse-grained manner. A MLP is used to aggregrate features adjusted by multiple query guide vectors.

$$F^{enc}_{(i)} = SA(F^{in}_{(i)}, F^{in}_{(i)}, F^{in}_{(i)}), \quad (10)$$

$$F^{enc} = MLP_{enc}([F^{enc}_{(1)}, F^{enc}_{(2)}, \ldots, F^{enc}_{(n)}]). \quad (11)$$

For our single phrase model, Equation (9) is replaced by the following formula:

$$F^{in} = f_u(V^{in}, s_{global}). \quad (12)$$

Decoder Generally, not only the query can help the video to highlight the crucial information, but also the video can choose the keywords from the query. According to the above observation, we design two branches (query branch and video branch) cross-attention mechanism to modulate each modality alternately, and each branch consists of a self-attention and a bi-attention block.

For the query branch, a self-attention takes all word embeddings V^{in} into account, which extracts single modality information F^{Q1} while keeps the original semitics. Then the words are attended by the enhanced video features F^{V1} from another branch with a bi-attention component. The detail in the video branch is as same as the query branch. Two branches are parallel so multi modalities are alternatively aligned. The process of the decoder is given by

$$F^{Q1} = SA(Q^{in}, Q^{in}, Q^{in}), \quad (13)$$

$$F^{V1} = SA(F^{enc}, F^{enc}, F^{enc}), \quad (14)$$

$$F^{Q2} = BA(F^{Q1}, F^{V1}, F^{V1}), \quad (15)$$

$$F^{V2} = BA(F^{V1}, F^{Q1}, F^{Q1}), \quad (16)$$

where $F^{Q1}, F^{Q2} \in \mathbb{R}^{d \times L}$, and $F^{V1}, F^{V2} \in \mathbb{R}^{d \times N}$.

3.4 Grounding

After getting the fused features F^{V2}, there are two feasible ways to localize the clip, anchor-based and anchor-free. As mentioned before, the anchor-based method exists redundant calculation and breaks the temporal consistency, resulting in sub-optimal solutions. We choose the anchor-free grounding method, following ABLR (Yuan, Mei, and Zhu 2019). Specifically, we design a temporal pooling module with attention mechanism to summarize the sequential information,

and conduct a grounding block with a MLP to regress the boundaries (τ^s, τ^e).

$$B = \tanh(W_b F^{V2}), \tag{17}$$

$$a = softmax(u_{ta}^\top B), \tag{18}$$

$$\overline{f} = \sum_{i=1}^{N} F_i^{V2} a_i, \tag{19}$$

$$(\tau^s, \tau^e) = MLP(\overline{f}), \tag{20}$$

where $W_b \in \mathbb{R}^{d\times d}, u_{ta} \in \mathbb{R}^d, \overline{f} \in \mathbb{R}^d$.

3.5 Training

We denote the training set as $\{(X_i, Q_i, T_i, \tau_i^s, \tau_i^e)\}_{i=1}^K$, where T_i represents the duration of the video X_i. And each description Q_i matches one particular video clip in X_i, where the start and end points are τ_i^s and τ_i^e respectively.

Our network is trained with three loss terms: (1) new location regression loss L_{reg}, (2) temporal attention loss L_{ta}, (3) semantic diversity loss, and the total loss is given by

$$L_{all} = L_{reg} + \lambda_{ta} L_{ta} + \lambda_{sd} L_{sd}. \tag{21}$$

Regression Loss We notice that the action boundary can not be distinguished precisely. Different annotators have different standards, which results in the inevitable annotation bias in datasets. Since it is hard to define the exact event boundaries, our goal is to minimize the effect of the bias by introducing a new task-specific loss function.

The quality of prediction depends on two main factors: the whole video duration and the ratio of ground truth interval to the duration. To handle with large-range video duration, Zeng *et al*. construct the feature pyramid (Zeng et al. 2020). In this paper, we normalize the start and end points (τ_i^s, τ_i^e) into $[0, 1]$ by dividing the video duration T_i, which is more efficient and less-calculation.

$$(\overline{t_i^s}, \overline{t_i^e}) = (\tau_i^s / T_i, \tau_i^e / T_i). \tag{22}$$

Previous works (Yuan, Mei, and Zhu 2019; Zeng et al. 2020; Mun, Cho, and Han 2020) directly adopt the Smooth L1 Loss.

$$Smooth_{L1} = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & |x| > 1. \end{cases} \tag{23}$$

However, the normalized time points $(\overline{t_i^s}, \overline{t_i^e})$ are less than 1, which actually becomes L_2 Loss in the region. Though L_2 loss can converge faster than L_1, L_1 loss is robust for outliers than L_2 loss (Girshick 2015). To bare with outliers and avoid the gradient vanishing in the meanwhile, we introduce a dataset-specific threshold $\beta \in (0, 1)$ between L_1 and L_2. In order to choose the best value, we count the ratio of the ground-truth interval to the whole video duration in Charades-STA and ActivityNet Captions datasets, as shown in Figure 4. The results show that the average ratios are 0.27 and 0.33. To control the accuracy of prediction, we introduce the coefficient α to weight the task-specific regression loss, $f(x)$.

Concrete denitions are as follows:

$$f(x) = \begin{cases} \alpha x^2, & |x| < \beta \\ 2\alpha\beta|x| - \alpha\beta^2, & |x| > \beta. \end{cases} \tag{24}$$

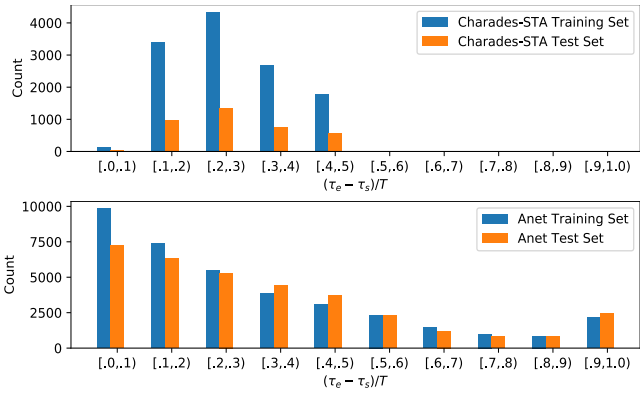

Figure 4: The distribution of ground-truth annotations in Charades-STA and ActivityNet Captions.

Temporal Attention Loss In the grounding module, we first conduct a temporal attention pooling to summarize the fused features. Generally, the components in ground truth play an essential role in localizing, so the attention weights a_{ta} related to the target are expected to be higher than others. Since there is no explicit ground truth of attention, we adopt a temporal attention loss following (Yuan, Mei, and Zhu 2019), which is given by

$$L_{ta} = -\frac{\sum_{i=1}^{N} \mathbb{1}(a_i) \log(a_i)}{\sum_{i=1}^{N} \mathbb{1}(a_i)}, \tag{25}$$

where $\mathbb{1}(a_i) = 1$ if the i-th feature is in the ground truth of boundaries and 0 otherwise.

Semantic Diversity Loss To keep the semantic diversity between different phrases, we conduct a regularization term on the attention matrix A_{spe} obtained by SPE, following (Lin et al. 2017).

$$L_{sd} = ||A_{spe} A_{spe}^\top - I||_F^2, \tag{26}$$

where $A_{spe} \in \mathbb{R}^{k\times L}$, and $||\cdot||_F$ stands for the Frobenius norm of a matrix.

4 Experiments

We demonstrate the feasibility of our CMA model on two public datasets: Charades-STA (Gao et al. 2017), and ActivityNet Captions (Krishna et al. 2017). Then we evaluate the improvement of performance in different modules.

4.1 Datasets

Charades-STA consists of around 10,000 videos, and each video contains activity annotation and video-level description but without clip-level annotation. (Gao et al. 2017) design a semi-automatic method to generate the description annotation with a start and end time. In total, the Charades-STA dataset contains 6,672 videos, 16,128 clip-sentence pairs, and 157 activity categories. There are 12,408 / 3,720 pairs in the training / test set. On average, the duration of videos is around 30s, and the words of each query are about 10.

Table 1: Experimental results on the Charades-STA dataset. $^+$ denotes the full model with multiple guide vectors.

Method	R1@0.3	R1@0.5	R1@0.7	mIoU	
CTRL	-	21.42	7.15	-	
SMRL	-	24.36	9.01	-	
RWM	-	36.70	-	-	-
ExCL	65.10	44.10	22.60	-	
CBP	-	36.80	18.87	-	
SCDM	-	54.44	33.43	-	
TMLGA	67.53	52.02	33.74	-	
DRN	-	53.09	31.75	-	
LGI	71.02	57.34	33.25	49.52	
Ours	**71.38**	**57.56**	**35.18**	**50.01**	
LGI$^+$	**72.96**	59.46	35.48	**51.38**	
Ours$^+$	72.20	**59.81**	**37.72**	51.04	

Table 2: Experimental results on the ActivityNet Captions dataset. $^+$ denotes the full model with multiple guide vectors.

Method	R1@0.3	R1@0.5	R1@0.7	mIoU
MCN	21.37	9.58	-	15.83
CTRL	-	21.42	7.15	-
ACRN	31.29	16.17	-	24.16
RWM	-	36.90	-	-
ABLR	55.67	36.79	-	36.99
CBP	54.30	35.76	17.80	-
SCDM	54.80	36.75	19.86	-
TMLGA	-	-	-	37.78
LGI	57.79	40.97	23.16	40.74
Ours	**58.67**	**41.43**	**23.59**	**41.23**
LGI$^+$	58.52	41.51	23.07	41.13
Ours$^+$	**59.10**	**41.93**	**24.23**	**41.96**

ActivityNet Captions contains 20,000 videos with 100,000 queries, covers 200 activity classes. It is split into 37421, 17505, 17031 pairs for training, validation, and test. On average, the duration of each video is 2 minutes and the length of each query is 13.48 words. Since the annotations of the test set are non-public, we merge the two validation sets val_1 and val_2 as our test split as previous works do (Mun, Cho, and Han 2020; Wang, Ma, and Jiang 2020).

4.2 Metrics

We choose similar metrics "R@n, IoU@m" and "mIoU" from (Gao et al. 2017) to evaluate the performance of our model. The Intersection over Union (IoU) between the prediction and ground truth is calculated for each query. "R@n, IoU@m" means the percentage of at least one of the top-n retrievals larger than the threshold m. We choose $n = \{1\}, m = \{0.3, 0.5, 0.7\}$ and abbreviate the symbol as "Rn@m". "mIoU" denotes the average IoU for all queries.

4.3 Implementation details

Video Feature We uniformly sample $N = 128$ clips from each video and each clip contains 16 frames. Then we adopt the C3D (Tran et al. 2015) network as the feature extractor of ActivityNet Captions. We also extract the I3D (Carreira and Zisserman 2017) features for Charades-STA.

Language Feature We first tokenize and lowercase the query, and use Glove (Pennington, Socher, and Manning 2014) to get the static word embedding with 300D. Then a two-layer bi-directional LSTM is used to get the contextual word embedding, whose hidden layer dimension is 256. The maximum query words for Charades-STA and ActivityNet Captions are 10 and 25, respectively. Vocabulary sizes are 1,140 and 11,125.

Training Settings Empirically, We set mini-batch size as 100, optimizer as Adam and initial learning rate as 1×10^{-3}. The parameters α, β in the task-specific regression loss are set $10, 0.1$ for Charades-STA; $2, 0.4$ for ActivityNet Captions, respectively. We use two-layer encoder-decoder with 4 heads for two datasets. The trade-off parameters λ_{ta} and λ_{sd} are set to 1.

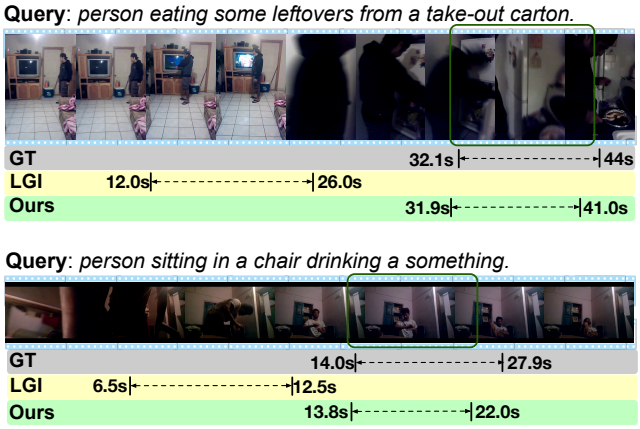

Figure 5: Qualitative results of LGI and our methods on Charades-STA.

4.4 Results

We compare our proposed method with the following: MCN (Anne Hendricks et al. 2017), CTRL (Gao et al. 2017), SMRL (Wang, Huang, and Wang 2019), RWM (He et al. 2019), ExCL (Ghosh et al. 2019), CBP (Wang, Ma, and Jiang 2020), SCDM (Yuan et al. 2019), TMLGA (Rodriguez et al. 2020), DRN (Zeng et al. 2020), and LGI (Mun, Cho, and Han 2020).

Charades-STA Table 1 summarizes performances on Charades-STA. Notice that our simple model (without SPE) makes the relative improvement over LGI (Mun, Cho, and Han 2020) by 0.36% (R1@0.3), 0.22% (R1@0.5), and 1.93% (R1@0.7). Our full model also outperforms LGI$^+$ by 0.35% (R1@0.5) and 2.24% (R1@0.7).

Table 3: Performance comparison for different structures on the Charades-STA dataset.

Method	R1@0.5
Encoder-only	51.16
Decoder-only	55.81
Full model	**57.56**

Table 4: Performance comparison for different position encoding methods on the Charades-STA dataset.

Method	R1@0.5
Ours-w/o PE	53.63
Ours-Embedding matrix	56.29
Ours-Sin/cos PE	**57.56**

ActivityNet Captions The numerical results are shown in Table 2. As can be seen, our simple and full models outperform the counterparts in LGI respectively. Note that our model has only less than half of the number of the parameters, which demonstrates the efficiency of our CMA.

Qualitative Results We provide some qualitative examples to validate the effectiveness of the CMA module. As shown in Figure 5 and Figure 6, our method can localize the approximate boundaries even for complicated situations like long-duration videos and complex sentences, while LGI fails.

4.5 Ablation Studies

Our CMA model consists of multiple components, including bi-modality encoder-decoder transformer and location regression. To evaluate the contribution of each module to the final performance, we conduct three primary ablation studies on Charades-STA dataset. We employ the simple model (without SPE) to demonstrate the efficiency of our cross-modality fusion method.

Encoder-Decoder We rst investigate the contribution of the coarse-to-fine strategy. In this experiment, we train three variants of our model: (1) Encoder-only: we only use encoder layers to coarsely interact with two modalities; (2) Decoder-only: we just align the local information with decoder layers; (3) Encoder-decoder: our full model perform multi-modality fusion based on encoder-decoder structure.

Table 3 summarizes the results where we observe that our full model outperforms the other variants. It is due to two reasons. First, encoder-only use the global sentence information, which is not enough to localize the precise time interval due to the lack of local details. Second, decoder-only is easy to stuck with sub-optical parameters without the guide of global information. This shows both global and regional knowledge are both crucial.

Position Encoding To evaluate the effectiveness of the temporal position embedding, we conduct three experiments: (1)w/o PE: features are fed into the network without adding PE. (2) use sine and cosine PE following (Vaswani et al. 2017), see Equal (28). (3) learn two position embedding matrixes $W^V_{pos} \in \mathbb{R}^{d\times N}, W^Q_{pos} \in \mathbb{R}^{d\times L}$. The result in Table 4 shows that PE is essential for precise localization because the transformer is non-sensitive to the permutation. Another point is that two different PE methods reach similar performance, so we choose the former because its efficient with less parameters.

$$PE_{(pos,2i)} = sin(\frac{pos}{10000^{2i/model}}), \tag{27}$$

$$PE_{(pos,2i+1)} = cos(\frac{pos}{10000^{2i/model}}). \tag{28}$$

Table 5: Performance comparison for different fusion methods on the Charades-STA dataset.

Method	R1@0.5
Ours-Add	53.42
Ours-Concat	56.45
Ours-HadamardProduct	**57.56**

Table 6: Performance comparison for different loss functions on the Charades-STA dataset.

Reg. Loss	α	β	R1@0.5
$Smooth_{L1}$	0.5	1	56.45
Ours	2	0.1	56.59
Ours	2	0.2	57.07
Ours	10	0.1	**57.56**

Fusion Methods Here we compare three combination methods ($f_u(\cdot,\cdot)$): Hadamard Product, Linear Addition, and Concatenation. The results in Table 5 show that Hadamard Product achieves the best performance.

Loss Design According to the distribution in Figure 4, we choose the threshold β from $\{0.1, 0.2, 1\}$. We find that the performance is improved when the threshold is close to the average of the ratio. In addition, properly increasing the value of the coefficient α can also improve performance. We compare different loss settings and find ours achieves the best performance as listed in Table 6.

5 Conclusion

In this paper, we propose a simple but powerful model for video temporal grounding with two-branch cross-modality attention. In this scheme, multi modalities are alternatively modulated, so the local and global contents are aligned well. In the meanwhile, we design a new regression loss that alleviate the effect of the annotation bias. The promising performances achieved on Charades-STA and ActivityNet Captions demonstrate the effectiveness of our method.

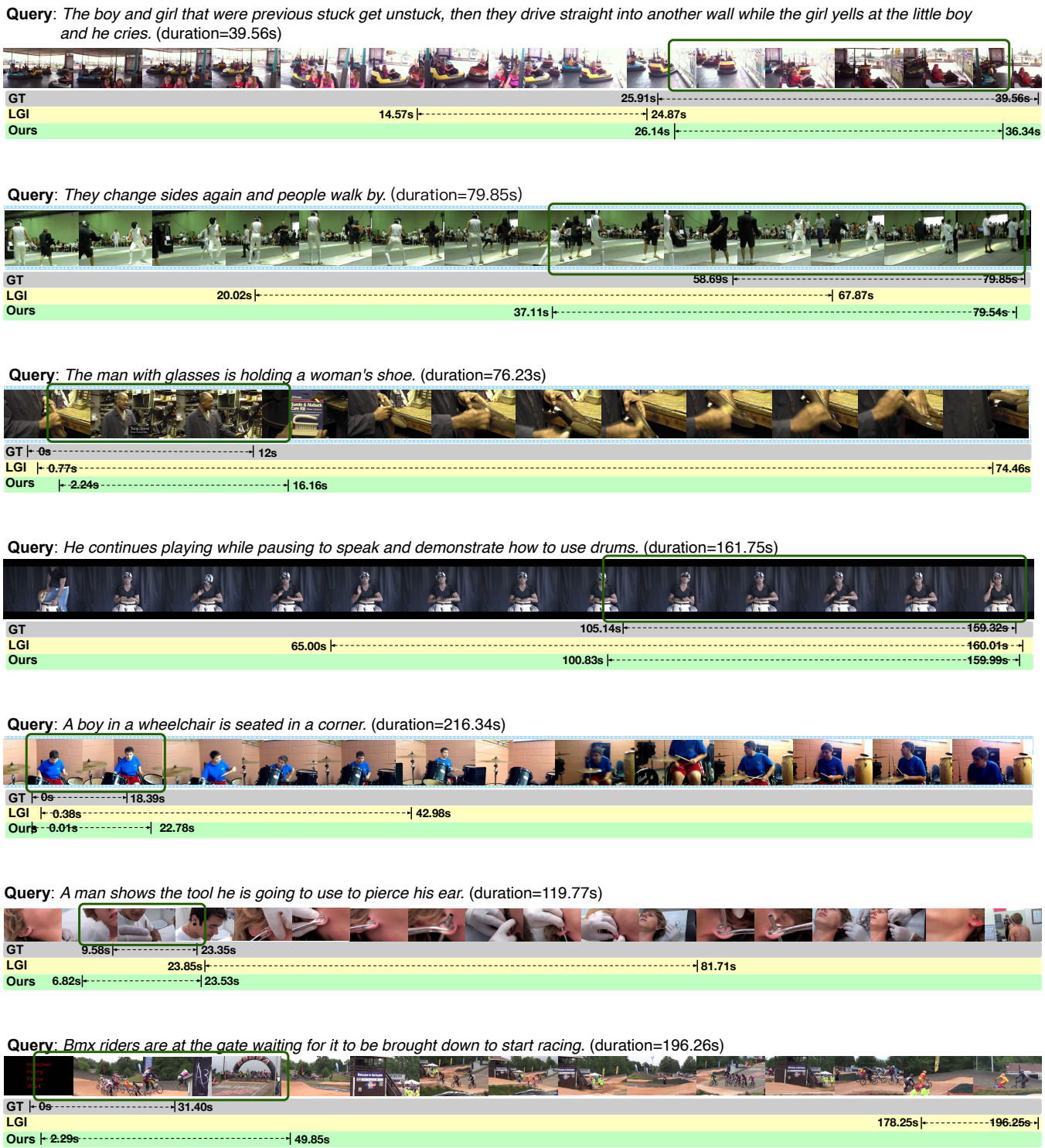

Figure 6: Qualitative results of LGI and our methods on ActivityNet Captions.

References

Anne Hendricks, L.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. 2017. Localizing moments in video with natural language. In *ICCV*.

Cadene, R.; Ben-Younes, H.; Cord, M.; and Thome, N. 2019. Murel: Multimodal relational reasoning for visual question answering. In *CVPR*.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.

Chen, S.; Zhao, Y.; Jin, Q.; and Wu, Q. 2020. Fine-grained Video-Text Retrieval with Hierarchical Graph Reasoning. In *CVPR*.

Choi, E.; Hewlett, D.; Uszkoreit, J.; Polosukhin, I.; Lacoste, A.; and Berant, J. 2017. Coarse-to-fine question answering for long documents. In *ACL*.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q. V.; and Salakhutdinov, R. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860* .

Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R. 2017. Tall: Temporal activity localization via language query. In *ICCV*.

Gao, P.; You, H.; Zhang, Z.; Wang, X.; and Li, H. 2019. Multi-modality latent interaction network for visual question answering. In *ICCV*.

Ghosh, S.; Agarwal, A.; Parekh, Z.; and Hauptmann, A. 2019. Excl: Extractive clip localization using natural language descriptions. *arXiv preprint arXiv:1904.02755* .

Girdhar, R.; Carreira, J.; Doersch, C.; and Zisserman, A. 2019. Video action transformer network. In *CVPR*.

Girshick, R. 2015. Fast r-cnn. In *ICCV*.

He, D.; Zhao, X.; Huang, J.; Li, F.; Liu, X.; and Wen, S. 2019. Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos. In *AAAI*.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Iashin, V.; and Rahtu, E. 2020. A Better Use of Audio-Visual Cues: Dense Video Captioning with Bi-modal Transformer. *arXiv preprint arXiv:2005.08271* .

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Carlos Niebles, J. 2017. Dense-captioning events in videos. In *ICCV*.

Lin, Z.; Feng, M.; Santos, C. N. d.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130* .

Liu, M.; Wang, X.; Nie, L.; He, X.; Chen, B.; and Chua, T.-S. 2018. Attentive moment retrieval in videos. In *ACM SIGIR*.

Ma, C.-Y.; Chen, M.-H.; Kira, Z.; and AlRegib, G. 2019. TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition. *Signal Processing: Image Communication* 71: 76–87.

Mithun, N. C.; Li, J.; Metze, F.; and Roy-Chowdhury, A. K. 2018. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *ACM MM*.

Mun, J.; Cho, M.; and Han, B. 2020. Local-Global Video-Text Interactions for Temporal Grounding. In *CVPR*.

Neubeck, A.; and Van Gool, L. 2006. Efficient non-maximum suppression. In *ICPR*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Rodriguez, C.; Marrese-Taylor, E.; Saleh, F. S.; Li, H.; and Gould, S. 2020. Proposal-free temporal moment localization of a natural-language query in video using guided attention. In *WACV*.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In *ICCV*.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Wang, J.; Ma, L.; and Jiang, W. 2020. Temporally Grounding Language Queries in Videos by Contextual Boundary-Aware Prediction. In *AAAI*.

Wang, W.; Huang, Y.; and Wang, L. 2019. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *CVPR*.

Yang, Z.; Garcia, N.; Chu, C.; Otani, M.; Nakashima, Y.; and Takemura, H. 2020. BERT Representations for Video Question Answering. In *WACV*.

Yuan, Y.; Ma, L.; Wang, J.; Liu, W.; and Zhu, W. 2019. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. In *NeurIPS*.

Yuan, Y.; Mei, T.; and Zhu, W. 2019. To find where you talk: Temporal sentence localization in video with attention based location regression. In *AAAI*.

Zeng, R.; Xu, H.; Huang, W.; Chen, P.; Tan, M.; and Gan, C. 2020. Dense regression network for video grounding. In *CVPR*.

Zhou, L.; Zhou, Y.; Corso, J. J.; Socher, R.; and Xiong, C. 2018a. End-to-end dense video captioning with masked transformer. In *CVPR*.

Zhou, L.; Zhou, Y.; Corso, J. J.; Socher, R.; and Xiong, C. 2018b. End-to-end dense video captioning with masked transformer. In *CVPR*.

Zolfaghari, M.; Singh, K.; and Brox, T. 2018. Eco: Efficient convolutional network for online video understanding. In *ECCV*.