VLG-Net: Video-Language Graph Matching Network for Video Grounding

Sisi Qu, Mattia Soldan, Mengmeng Xu, Jesper Tegner, Bernard Ghanem
King Abdullah University of Science and Technology (KAUST), Saudi Arabia
{sisi.qu, mattia.soldan, mengmeng.xu, jesper.tegner, bernard.ghanem}@kaust.edu.sa

Abstract

Grounding language queries in videos aims at identifying the time interval (or moment) semantically relevant to a language query. The solution to this challenging task demands the understanding of videos’ and queries’ semantic content and the fine-grained reasoning about their multi-modal interactions. Our key idea is to recast this challenge into an algorithmic graph matching problem. Fueled by recent advances in Graph Neural Networks, we propose to leverage Graph Convolutional Networks to model video and textual information as well as their semantic alignment. To enable the mutual exchange of information across the domains, we design a novel Video-Language Graph Matching Network (VLG-Net) to match video and query graphs. Core ingredients include representation graphs, built on top of video snippets and query tokens separately, which are used for modeling the intra-modality relationships. A Graph Matching layer is adopted for cross-modal context modeling and multi-modal fusion. Finally, moment candidates are created using masked moment attention pooling by fusing the moment’s enriched snippet features. We demonstrate superior performance over state-of-the-art grounding methods on three widely used datasets for temporal localization of moments in videos with natural language queries: ActivityNet-Captions, TACoS, and DiDeMo.

1. Introduction

Temporal action understanding is at the forefront of computer vision research. The task of temporally grounding language queries in videos was recently introduced by Hendricks et al. [1] and Gao et al. [14], as a generalization of the temporal action localization task and so as to overcome the constraint of a predefined set of actions. This novel interdisciplinary task has gained momentum within the vision and language community for its relevance and possible applications in video retrieval [11, 52, 77], video question answering [20, 26], human-computer interaction [86], and video storytelling [17]. Recently, two new tasks (video corpus moment retrieval [60] and video-subtitle moment retrieval [27]) were introduced to extend the localization task to a large corpus of videos. However, we maintain that the complexity of the single video setup requires further investigation to devise better methods.

Natural language grounding in videos faces multiple challenges, some of which are inherited from temporal action localization, such as context awareness modelling and candidate moment generation. Semantic context is a fundamental cue necessary to boost the performance of localization methods [10, 13, 30, 62, 72]. To enrich video representation, Hendricks et al. [1] adopted global-context, which is moment independent, leading to sub-optimal performance. Conversely, a moment specific local-context, defined as a moment’s temporal neighbourhood, was adopted in [14, 15, 23, 34, 55, 67]. In our view, non-local context merits deeper analysis, since it has the potential to identify relevant information not restricted to the temporal neighbourhood within one data modality. For example, in Fig. 1(a), although “First throw and fetch” is not in the temporal vicinity of “Second throw and fetch”, it is still semantically related with our target moment, which is a good test-
timony of the importance of non-local context modeling for video grounding. Moreover, as showcased by the example, the free-form nature of the language modality introduces additional challenges, as a model must operate not only with the semantic content of both videos and language queries, but also reason about their multi-modal interactions. Previous studies [14, 16, 55] employ a cross-modal processing unit designed to jointly model text and visual features by calculating element-wise addition, multiplication, and direct concatenation between a moment’s representation and the query embedding, as depicted in Fig 1 (b). Alternatively, Mun et al. [44] only adopt Hadamard product to fuse this information at query-snippet level in Fig. 1 (c).

Motivated by the work in [72], we propose to leverage the representational capability of graphs to encode snippet-snippet, token-snippet, and token-token relations as edge connections. As such, we adopt Graph Convolutional Networks (GCN) [57] to gather intra- and inter-modality context. This sets the stage for resolving finer alignments by re-casting inter-modality interactions as an algorithmic graph matching problem working at the level of snippets and tokens, as shown in Fig. 1(d), different from previous methods that focus on relatively coarse query-moment or query-token interactions. Inspired by [29, 71], we adopt a cross-graph attention-based matching mechanism to enable the mutual exchange of information across the modalities.

In this paper, we design a new architecture referred to as Video-Language Graph Matching Network (VLG-Net). First, representation graphs for both video and language are constructed. The video graph models each snippet as a node and takes advantage of two sets of edges to represent both local temporal relations and non-local semantic relations between video snippets. Similarly, we construct a language graph, where each node is a token and each edge reflects token-to-token relations, e.g. syntactic dependencies [39, 40]. These modality-specific graphs are used to model local and non-local context through graph convolutions. Then, we resort to the attention-based graph matching layer to fuse the modalities together and enable inter-modality context aggregation, allowing for fine-grained alignment through the adoption of a specialized set of learnable edges. With this design, we avoid the need for heuristics of context modelling and simultaneously learn the best way for multi-modal fusion.

Contributions. Our contributions are two-fold. (1) We provide a new representation for modeling contextual information and multi-modal fusion for the video grounding task. To the best of our knowledge, we are the first to utilize attention-based graph matching of language and video graphs to learn video-language interactions for grounding. (2) Our VLG-Net improves localization performance by enabling finer alignment between the two modalities. Extensive experiments demonstrate best results with significant improvement over state-of-the-art grounding methods in three commonly used datasets, which verify the effectiveness of our approach.

2. Related work

2.1. Video Grounding

Moment candidates. Previous works can be organized into two major categories: (i) proposal-free and (ii) proposal-based methods. Proposal-free methods, such as [9, 18, 32, 36, 38, 44, 47, 53, 55, 79, 84], aim at directly regressing the temporal boundaries of the queried moment from the multi-modal fused feature. In contrast, proposal-based approaches [1, 4, 6, 7, 15, 22, 34, 61, 67, 83, 85] adopt a propose-and-rank pipeline: by firstly generating moment proposals and then ranking them according to their similarity with the textual query. In 2D-TAN, Songyang et al. [85] show the removal of heavily overlapped moments can significantly reduce the amount of computation while retaining good performance. Accordingly, we design the candidates and score the moments without boundary refinements [14, 15, 33, 34, 61, 67].

Moments’ context. For moment context modelling, some works [6, 18] attempt to use the memory property of LSTM cells [51] to contextualize the video features. However, attention-based mechanisms [59] adopted in [31, 33, 61] improve the aggregation of long-range semantic dependencies. Similar to [61], we suggest that visual context modelling should be dynamic and query-dependent. Songyang et al. [85] claim that neighbouring proposals hold valuable context and apply 2D convolutions with large kernel size on top of the moment representations to gather context information in the latter stages of their pipeline. In contrast to [85], we only use a Multi-Layer Perceptron (MLP) network for moment score computation, which reduces the overall computation, while delegating the context gathering to earlier stages of the pipeline.

Multi-modal fusion. Moving beyond the cross-modal processing unit presented in Section 1, the works of [67] devise a new cross-modality interaction scheme based on circular matrices. In [6, 34, 61], frame features are concatenated with frame-guided attention-pooled features of the language. Lu et al. [38] take advantage of the QANet[76] architecture, which is based on cross-attention and convolution operations, for multi-modal fusion. Dynamic filters generated from language features are adopted in [47, 83] in order to modulate, through convolutions, the visual information given the query. Recently, using the Hadamard product to fuse/gate information [44, 79, 85] has become popular. In contrast to these methods, our graph matching layer specifically models local, non-local, and query-guided context, thereby exploiting the semantic neighborhood of snippets and tokens to fuse the modalities through graph convolutions.
2.2. Graphs and Graph Neural Networks

Graphs in Videos. In various video understanding tasks, such as action recognition [8, 35, 65], and action localization [72, 78], graphs equipped with extensive representational power have been used for representation of data/features. Each video is represented as a space-time region graph in [65] and as a 3D point cloud in the spatial-temporal space in [35]. Alternatively, Zeng et al. [78] define temporal action proposals as nodes to form a graph, while Xu et al. [72] consider video snippets as nodes in a graph. Inspired by [72], video snippets in our work are represented as nodes in a graph and different edges are constructed between them to model various relationships.

Graphs in Language. In natural language processing (NLP), both sequential and non-local relations are crucial. The former is usually captured by recurrent neural networks [49], while the latter can be represented using graph neural networks [2, 3, 28, 41, 43, 55, 80]. Syntactic information has been shown successful for language modelling when combined with GCN in [21, 31, 40, 81]. Driven by these arguments, we stack together LSTM and a Syntactic Graph Convolution Network (SyntacGCN) [21, 31, 40, 81] for modelling and enriching the language features.

Graph Neural Networks in Graph Matching. Graph matching is one of the core problems in graph pattern recognition, aiming to find node correspondences between different graphs [5]. Given the powerful capability of Graph Neural Networks (GNNs) for encoding graph structure information, approaches leveraging GNNs/GCNs have recently surfaced in the graph matching field [63, 71]. For example, Li et al. [29] propose a GNN-based graph matching network to represent graphs as feature vectors to measure their similarity. Following [29], a neighborhood matching network is introduced by [69] to match graph nodes by estimating similarities of their neighborhoods. Due to their superiority in finding consistent correspondences between sets of features, graph matching methods have been widely applied in various tasks, such as cross-lingual knowledge graph alignment [69, 71], object shape matching [73], video re-identification [74], 3D action recognition [19], etc. Motivated by these works, we resort to applying graph matching in the video grounding task by employing a cross-graph attention matching mechanism.

3. Methodology

3.1. Problem Formulation

Given an untrimmed video and a language query, the video grounding task aims to localize a temporal moment in the video that matches the query. Each video-language pair has one associated ground-truth moment, defined as a temporal interval with boundary \((\tau_s, \tau_e)\), which best matches the query. Our method scores \(m\) candidate moments, where the \(k\)-th moment consists of start time \(t_{s,k}\), end time \(t_{e,k}\), and confidence score \(p_k\).

The video stream is represented as a sequence of snippets \(V = \{v_i\}_{i=1}^{n_v}\), where each snippet has \(e\) consecutive frames and \(n_v\) is the number of snippets. Similarly, a language query is represented by \(n_l\) tokens \(L = \{t_{i,j}\}_{j=1}^{n_l}\). The inputs to our VLG architecture are snippet features \(X_v \in \mathbb{R}^{c_v \times n_v}\) and token features \(X_l \in \mathbb{R}^{c_l \times n_l}\) extracted using pre-trained models, where \(c_v\) and \(c_l\) are the snippet and token feature dimensions. We describe the details of feature extraction in Sec. 4.2.

3.2. VLG-Net Architecture

Our video grounding architecture is illustrated in Fig. 2. First, we feed both the video features \(X_v\) and the language embeddings \(X_l\) into a stack of computation blocks, respectively. On the video path, we use 1D convolutions and GCNEx [72] blocks to enrich the visual representation with local and non-local intra-modality context. On the language path, we apply LSTM and SyntacGCN to aggregate temporal and syntactic context, which models the grammatical structure of the language query. The two paths converge in the graph matching layer for cross-modal context modelling and multi-modal fusion. After the graph matching layer, we apply masked moment attention pooling to produce the representations of possible moment candidates. Finally, we use Multi-Layer Perceptron (MLP) to score the query-moment pair based on their representation and post-process the score by non-maximum suppression (NMS). We report top-\(\kappa\) ranked moments as the final predictions.

3.3. Video and Language Representations

Here, we detail the set of operations performed on each modality. The stack of computation blocks of each branch is specifically designed to model intra-modality context to enrich each snippet and token features.

Video Representation. We first apply 1D convolutions to map each input visual feature to a desired dimension. The video is then cast as a graph, where each node represents a snippet and each edge represents a dependency between a snippet pair. We design two types of edges: (i) Ordering Edges and (ii) Semantic Edges. Static Ordering Edges connect consecutive snippets and model the temporal order. Conversely, Semantic Edges are dynamically constructed to connect semantically similar snippets based on their feature representations. Specifically, an ordering or semantic snippet neighborhood is determined, and its aggregated representation is computed through edge convolutions \(F\), similar to [66]. Each edge convolution employs a split-transform-merge strategy [70] to increase the diversity of transformations. These graph operations (called GCNEx) were proposed in [72] to enrich video snippet representations for the purpose of temporal action localization. In our architecture,
The man wiped the frisbee and then threw it again, and the dog caught and brought it back to the owner.

Figure 2. VLG-Net Architecture. The inputs are snippet features and token embeddings. 1D convolutions and $b_t$ GCNeXt operations are applied to the video snippets to enrich their representations. Correspondingly, tokens are fed to a stack of LSTM layers and $b_t$ SyntacGCN layers. A graph matching layer is adopted for cross-modal context modelling and multi-modal fusion. Masked attention pooling lists all possible moment candidates, and a Multi-Layer Perceptron (MLP) computes each moment’s score to rank them as final predictions.

Since we stack $b_t$ GCNeXt blocks together and refer to the input of each block as $X_v^{(i)}$, where $X_v^{(0)}$ is the output of the convolutional layer. The GCNeXt block is formulated as:

$$X_v^{(i+1)} = \text{GCNeXt}(X_v^{(i)}) = \sigma \left( F(X_v^{(i)}, A_v^{(i)}, W_v^{(i)}) + F(X_v^{(i)}, A_v^{(i)}, W_s^{(i)}) + X_v^{(i)} \right),$$

where $A_v^{(i)}$ and $A_s^{(i)}$ are respectively the adjacency matrices of Ordering Edges and Semantic Edges, and $W_v^{(i)}$ are the trainable weights for the $i$-th GCNeXt block. We use Rectified Linear Unit (ReLU) as the activation function $\sigma$. More details about this block can be found in [72]. The output of the last block is referred to as $X_v^{(b_t)}$, which is the input to the graph matching layer.

**Language Representation.** The query token features $X_l$ are fed to a set of $b_l$ LSTM layers to capture semantic information and the temporal context. Moreover, given that language follows a predefined set of grammatical rules, we set out to leverage syntactic information [40, 82] to model grammatical inter-word relations. For this purpose, we adopt SyntacGCN, as shown in Fig. 1b. Syntactic graphs are preferred over fully connected graphs, since the former’s sparsity property makes them more robust against noise in language [21]. Our SyntacGCN represents a query as a sparse directed graph, in which each output of the last LSTM layer, referred to as $X_l^{(0)}$, is viewed as a node, and each syntactic relation as an edge. The adjacency matrix $A_l$ is directly constructed from the query’s syntactic dependencies [39] and the graph convolution can be formulated as in Eq. 2.

$$X_l^{(i+1)} = \sigma \left( X_l^{(i)} + \sum_{k \in \mathcal{N}(j)} A_{l,j} W_{l,j} \cdot X_l^{(i)} \right)$$

Here, $X_l^{(i)}$ is the $j$-th token feature from the $i$-th graph input, $\mathcal{N}(j)$ is the syntactic neighbourhood of node $j$, $W_{l,j}$ is the learnable weight in the $i$-th block, and $\sigma$ is ReLU. Moreover, $\alpha_{l,j}^{(i)}$ is the edge weight learned from the feature of paired nodes $X_l^{(i)}$ and $X_l^{(i)}$, shown in Eq. 3, where $w_{\alpha}$ and $W_{\alpha}$ are learnable parameters and $||$ denotes vector concatenation.

$$\alpha_{l,j}^{(i)} = \text{SoftMax}(w_{\alpha}^{(i)} \cdot \sigma(W_{\alpha}^{(i)} (X_l^{(i)} || X_l^{(i)})))$$

We refer to the last output of the SyntacGCN as $X_l^{(b_l)}$, which will be used to match the video representation $X_v^{(b_t)}$.

### 3.4. Video-Language Graph Matching

The enriched video representation $X_v^{(b_t)}$ and language representation $X_l^{(b_l)}$ meet and interact in the graph matching layer. This layer is adopted for cross-modal context modelling and multi-modal fusion. It achieves this goal by evaluating the correlation between each snippet in the video and each token in the language by exploiting the semantic neighbourhood of snippets/tokens. The process is shown in
1 Introduction

Section 1.1 presents the motivation of the problem. Section 1.2 elaborates on the related work. Section 1.3 introduces the proposed model, followed by Section 1.4 that discusses the experimental setup. Finally, Section 1.5 concludes the introduction with a summary of the key findings.

2 Methods

2.1 Preprocessing

2.2 Model Architecture

2.3 Training Details

2.4 Evaluation Metrics

3 Results

3.1 Ablation Study

3.2 Qualitative Analysis

3.3 Comparison with Baseline

4 Conclusion

4.1 Limitations

4.2 Future Work

5 Appendix

5.1 Additional Experiments

5.2 Supplementary Material

Figure 3. The video-language matching graph. The nodes represent video snippets and language tokens. Ordering Edge (black) models the sequential nature of both modalities. Semantic Edge (yellow) connects graph nodes in the same modality according to their feature similarity. Matching Edge (purple) captures the cross-modality relationships. We apply relation graph convolution on the constructed video-language graph for cross-modal context modelling and multi-modal fusion. The neighborhood is related to the node highlighted in red.

Figure 4. Masked attention pooling. Sequence of operations for the learnable cross-attention configuration. Inputs are video nodes $X_v^{(GM)}$ from the graph matching layer and the query embedding $X_l^{(att)}$ computed through self-attention pooling on top of the last SyntacGCN layer’s output. The output $Y$ represents all moment candidates.

Here, $x_i \in \{X_v^{(b_1)}, \ldots, X_v^{(b_s)}, X_l^{(b_1)}, \ldots, X_l^{(b_t)}\}$ is the feature representation of the $i$-th node in the video-language matching graph. $N_i^r$ collects all nodes connected to the $i$-th node by edge type $r \in \{O, S, M\}$. $\beta_{i,j}$ is proportional to $x_i^T x_j$, and $W$ are the learnable weights.

Since $x_i^{(GM)}$ is enriched with intra-modality and inter-modality context, we use it to represent the video-language relation for each snippet and token. As output, we stack all the $x_i^{(GM)}$ to form $X^{(GM)} \in \mathbb{R}^{c \times n_v}$ and pass it to the masked moment pooling layer.

3.5 Masked Attention Pooling

The graph matching layer returns a new video graph and language graph fused with the other modality. Then, a masked attention pooling operation is applied to the new video graph to list the relevant sub-graph representations as candidate moments. The output of this module is denoted as $Y = [y_k]_{k=1}^m$, $y_k \in \mathbb{R}^c$, where $m$ is the number of candidate moments, and $c$ is the feature dimension of each moment. For efficiency purposes, the operation is implemented as a masked attention, allowing us to process each snippet feature only once while computing each moment’s representation.

Specifically, we implement three different schemes, namely: (i) learnable self-attention, (ii) cross-attention, and (iii) learnable cross-attention. In (i), we obtain the unnormalized attention weights by applying a 1D convolutional layer that maps each snippet feature to a single score. In (ii) and (iii), we compute the query representation $X_l^{(att)}$ by applying self-attention pooling on top of the last SyntacGCN layer $X_l^{(b_t)}$. Cross-attention obtains the unnormalized weights by computing the inner product between the snippet and query feature, while learnable cross-attention concatenates each snippet feature with the query feature and uses a 1D convolutional layer to obtain the weights. Configuration (iii) is depicted in Fig. 4. In all cases, the unnormalized weight vector has shape $w \in \mathbb{R}^{n_v \times 1}$ for each video. The vector is repeated $m$ times to obtain the matrix.

$$x_i^{(GM)} = \sum_{j \in N_i^O} (W_O x_j + W_0 x_i)$$

$$+ \sum_{j \in N_i^S} (W_S x_j + W_0 x_i)$$

$$+ \sum_{j \in N_i^M} (\beta_{i,j} W_M x_j + W_0 x_i)$$
$W \in \mathbb{R}^{n_v \times m}$, and a fixed mask $M \in \mathbb{R}^{n_v \times m}$ is applied to it. Similar to Songyang et al. [85], we generate moment candidates and apply a sparse sampling strategy to discard redundant moments. Therefore, the mask is generated according to the sampled moments, highlighting for each of them, which are the snippets that must be taken into account when computing the moment’s pooled feature. The attention scores are then obtained by a softmax operation. Thanks to the masking operation, snippets not related to the $n$-th moment will not be considered. Finally, the moments’ features are obtained simply as a matrix multiplication: $Y = X^{\text{GM}} \text{SoftMax}(W + M)$. Ablation results are reported in Supplementary Material.

3.6. Moment Localization

The output of the previous module is then fed to a Multi-Layer Perceptron (MLP) network to compute the score $p_k$ for each moment candidate. This score predicts the Intersection-over-Union (IoU) of each moment with the ground truth one. For training, we supervise this process using a cross-entropy loss, shown in Eq. 5. We assign the label $t_k = 1$ if the IoU is greater than a threshold $\theta$ and $t_k = 0$ otherwise.

$$L = \frac{1}{m} \sum_{k=1}^{m} t_k \log p_k + (1 - t_k) \log(1 - p_k), \quad (5)$$

At inference time, moment candidates are ranked according to their predicted scores and non-maximum suppression is adopted to discard highly overlapping moments. The remaining top-$\kappa$ moments are involved in the recall computation. The temporal boundaries $(t_n, t_e)$ associated with the top-$\kappa$ moments are used to calculate the Intersection-over-Union (IoU) with the ground-truth video moments $(\tau_n, \tau_e)$ to determine the alignment performance. A formal definition of the metric used and details about the training strategy are presented in Sec. 4.2

4. Experiments

4.1. Datasets

ActivityNet-Captions [25] is a popular benchmark dataset initially collected for the task of dense captioning, and recently adopted for the task of moment localization with natural language [6, 31]. The dataset is subdivided into four splits: train, val_1, val_2, and test. The test set is withheld for competition purposes while leaving the rest publicly available. See Tab. 1 for details about the publicly available splits. Following the setting in [31], we use val_1 as the validation set and val_2 as the testing set.

TACoS [46] consists of videos selected from the MPII Cooking Composite Activities video corpus [48]. It is comprised of 18818 video-language pairs of different cooking activities in the kitchen. On average, every video in TACoS contains 148 queries, some of which are annotations of short video segments.

DiDeMo [1] contains unedited video footage from Flickr with sentences aligned to unique moments in its 10642 videos. It contains more than 40k video-language pairs with coarse temporal annotations. Moment start and end points are aligned to five-second intervals and the maximum annotated moment length is 30 seconds.

Concerns related to Charades-STA [14] dataset led us to the decision of not evaluating our method on it. Limitations include the absence of a validation set, reduced vocabulary size, and a smaller number of video-language pairs compared to the other datasets (Tab. 1). See Supplementary Material for more discussion.

4.2. Implementation

Evaluation Metrics. We follow the commonly used setting in [12], where the Rank@$\kappa$ for IoU=$\theta$ serves as our evaluation metric. For example, given a video-query pair, the result is positive if any of the top-$\kappa$ predictions has IoU with the ground-truth larger or equal to $\theta$; otherwise the result is negative. We average the results across all testing samples. Specifically, we set $\kappa \in \{1, 5\}$ with $\theta \in \{0.3, 0.5, 0.7\}$ for ActivityNet Captions, $\kappa \in \{1, 5\}$ with $\theta \in \{0.1, 0.3, 0.5\}$ for TACoS, and $\kappa \in \{1, 5\}$ with $\theta \in \{0.5, 0.7, 1.0\}$ for DiDeMo.

Language and Video Features. After lower-case conversion and tokenization, we use the pretrained GloVe model [45] to obtain the initial language embedding for every token and extract the syntactic dependencies using the Stanford CoreNLP 4.0.0 parser [39]. The $b_s$ layers of LSTM with 512 hidden units are used as the query encoder. Then, the syntactic GCN encodes syntactic information of the queries and returns a new embedding with 512 dimensions. For visual features, we use pretrained C3D [58] for ActivityNet Captions and TACoS, and VGG16 [54] for DiDeMo, while holding their parameters fixed during training, as they are readily available and commonly used by state-of-the-art methods. We use 1D convolutions to project the input visual features to a fixed dimension (512), and the hyper-parameters of GCNeXt blocks are the same as in [72].

Training and Inference. We use Adam [24] with a CosineAnnealingLR scheduler [37]. We adopt learning rates ranging from $10^{-3}$ to $10^{-4}$ for different datasets. The number of sampled snippets $n_v$ is set to 64 for ActivityNet

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Num. Videos</th>
<th>Video-Sentence pairs</th>
<th>Vocab. Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActivityNet-Captions [25]</td>
<td>14926</td>
<td>37421</td>
<td>15406</td>
</tr>
<tr>
<td>TACoS [46]</td>
<td>127</td>
<td>10146</td>
<td>2255</td>
</tr>
<tr>
<td>DiDeMo [1]</td>
<td>10642</td>
<td>33005</td>
<td>7523</td>
</tr>
<tr>
<td>Charades-STA [14]</td>
<td>6670</td>
<td>12404</td>
<td>1289</td>
</tr>
</tbody>
</table>

Table 1. Video-language grounding dataset statistics.
4.3. Comparison with State-of-the-Art

**ActivityNet Captions.** In Table 2, we report the performance of our method in comparison with several recently proposed methods. VLG-Net reaches the highest scores for all evaluation metrics except for R@5 IoU=0.5. For R@1 IoU=0.7, VLG-Net achieves the improvement of 2.04 over the previous best method.

<table>
<thead>
<tr>
<th>Model</th>
<th>R@1</th>
<th>R@1</th>
<th>R@1</th>
<th>R@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCN [1]</td>
<td>21.36</td>
<td>6.43</td>
<td>53.23</td>
<td>29.70</td>
</tr>
<tr>
<td>CTRL [14]</td>
<td>20.01</td>
<td>10.34</td>
<td>59.17</td>
<td>37.54</td>
</tr>
<tr>
<td>TGN [6]</td>
<td>27.93</td>
<td>–</td>
<td>44.20</td>
<td>–</td>
</tr>
<tr>
<td>ACRN [32]</td>
<td>31.67</td>
<td>11.25</td>
<td>60.34</td>
<td>38.57</td>
</tr>
<tr>
<td>CMIN [31]</td>
<td>44.62</td>
<td>24.48</td>
<td>69.66</td>
<td>52.96</td>
</tr>
<tr>
<td>QSPN [22]</td>
<td>33.26</td>
<td>13.43</td>
<td>62.39</td>
<td>40.78</td>
</tr>
<tr>
<td>ABLR [75]</td>
<td>30.79</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>TripNet [42]</td>
<td>32.19</td>
<td>13.93</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>PMI [53]</td>
<td>38.28</td>
<td>17.83</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>2D-TAN [85]</td>
<td>44.51</td>
<td>26.54</td>
<td>72.13</td>
<td>61.96</td>
</tr>
<tr>
<td>DRN [79]</td>
<td>45.45</td>
<td>24.39</td>
<td><strong>77.97</strong></td>
<td>50.30</td>
</tr>
</tbody>
</table>

**Table 2. State-of-the-art comparison on ActivityNet Captions.** We report the results at different Recall@k and different IoU thresholds. VLG-Net reaches the highest scores for IoU=0.7 for both R@1 and R@5. The top-2 performance values are highlighted by bold and underline, respectively.

**DiDeMo.** Table 4 shows the video grounding results of our VLG-Net as compared to state-of-the-art methods using the same VGG features. Our proposed technique outperforms the top ranked methods ROLE and ACRN with respect to R@1 and R@5 for IoU=0.5 and 0.7 with evident increases. It also reaches the highest performance in regards to R@1 IoU1.0. For completeness, we report the performances of TGN [32] and TMN [6]; with the caveat that their performance could not be verified as the code is not publicly available.

<table>
<thead>
<tr>
<th>Model</th>
<th>R@1</th>
<th>R@1</th>
<th>R@1</th>
<th>R@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TGN [32]</td>
<td>–</td>
<td>–</td>
<td>18.71</td>
<td>–</td>
</tr>
<tr>
<td>ROLE [34]</td>
<td>27.44</td>
<td>16.65</td>
<td>–</td>
<td>48.42</td>
</tr>
<tr>
<td>ACRN [33]</td>
<td>29.40</td>
<td>15.08</td>
<td>–</td>
<td>70.72</td>
</tr>
<tr>
<td>VLG-Net</td>
<td><strong>32.28</strong></td>
<td><strong>24.82</strong></td>
<td><strong>84.80</strong></td>
<td><strong>68.09</strong></td>
</tr>
</tbody>
</table>

**Table 4. State-of-the-art comparison on DiDeMo.** Our proposed model outperforms the top ranked method ROLE and ACRN with respect to IoU=0.5 and 0.7 for R@1 and R@5 with clear margins. It also reaches the highest performance in regards to R@1 IoU1.0.

4.4. Ablation Study

To evaluate the effectiveness of every module/component in VLG-Net and to justify our design choices, we perform several ablation experiments.

First, we investigate the impact of the architecture’s main modules, including GCNExt for video modeling, SyntacGCN for language modeling, and graph matching for cross-modality interactions (Tab. 5). In this experiment, we train four models on two datasets, by ablating modules one-by-one and focus on R@1 IoU=0.7 as being the most challenging metric. By removing the GCNExt module, the video intra-modality context is not modelled as a graph, leading to the drop of performance, thus motivating our choice for intra-modality context gathering for videos via graphs. Similarly, ablating SyntacGCN results in inferior performance. Notably, when the graph matching module is eliminated, the performance is severely impaired, indicating its crucial importance. Combining all these modules together registers the best performance, which attests to the superiority of our proposed VLG-Net architecture in modeling intra- and inter- interactions between video snippets and language queries.

We also analyze the contributions of different types of put features is illustrated in Table 3. Our model outperforms all the competing methods and achieves the highest scores for all IoU thresholds with significant improvement. Specifically, VLG-Net exceeds the previously best method 2D-TAN [85] by a margin ranging from 2.98 to 7.67 across all evaluation settings.
edgess considered in the graph matching module for two datasets, as shown in Tab. 6. For each dataset, removing Ordering Edges or Semantic Edges leads to inferior results. This supports the notion that temporal local and non-local relationships modelling is beneficial for snippets/tokens features enrichment. Importantly, excluding Matching Edges impairs the performance significantly. This is because this setup prevents the exchange of information between video and language in the graph matching layer, while the two modalities only interact in the moment attention pooling module. Again, the best performance is obtained when all three types of edges are considered, thus justifying our design for the multi-modal fusion operation.

4.5. Visualization

We show several qualitative grounding results from ActivityNet-Captions in Fig. 5. Our VLG-Net can generate precise moment boundaries that match the query well in different scenarios.

Worth mentioning, our method can sometimes give predictions that are more meaningful than the ground truth annotation. As shown in Fig. 5(d), although the ground truth aligns to the very beginning of the video only, the query “Three people are riding a very colourful camel on the beach” can semantically match the whole video. Obviously, our VLG-Net, in this case, gives a more reasonable grounding result.

5. Conclusion

In this paper, we address the problem of text-to-video temporal grounding and we recast it into an algorithmic graph matching problem. We formulate the untrimmed video and language query as video and language graphs respectively, where video snippets and language tokens are graph nodes, and snippet/token correlations are edges. We propose Video-Language Graph Matching Network (VLG-Net) to match the video graph and language graph through GCN, and explore four types of edges, Syntactic Edge, Ordering Edge, Semantic Edge, and Matching Edge, that encode local, non-local, and cross-modality relationships for finer alignment of the video-query pair. Extensive experiments show that our VLG-Net can efficiently model inter- and intra-modality context, learn the best way for multi-modal fusion, and surpass the current state-of-the-art performance on three widely adopted datasets: ActivityNet-Captions, TACoS, and DiDeMo.
References


[27] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. TVR: A Large-Scale Dataset for Video-Subtitle Moment Re-


[57] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, 2017. 2


