

Natural Language Video Localization: A Revisit in Span-Based Question Answering Framework

Hao Zhang¹, Aixin Sun¹, Wei Jing¹, Liangli Zhen¹, Joey Tianyi Zhou¹, and Rick Siow Mong Goh

Abstract—Natural Language Video Localization (NLVL) aims to locate a target moment from an untrimmed video that semantically corresponds to a text query. Existing approaches mainly solve the NLVL problem from the perspective of computer vision by formulating it as ranking, anchor, or regression tasks. These methods suffer from large performance degradation when localizing on long videos. In this work, we address the NLVL from a new perspective, i.e., span-based question answering (QA), by treating the input video as a text passage. We propose a video span localizing network (VSLNet), on top of the standard span-based QA framework (named VSLBase), to address NLVL. VSLNet tackles the differences between NLVL and span-based QA through a simple yet effective query-guided highlighting (QGH) strategy. QGH guides VSLNet to search for the matching video span within a highlighted region. To address the performance degradation on long videos, we further extend VSLNet to VSLNet-L by applying a multi-scale split-and-concatenation strategy. VSLNet-L first splits the untrimmed video into short clip segments; then, it predicts which clip segment contains the target moment and suppresses the importance of other segments. Finally, the clip segments are concatenated, with different confidences, to locate the target moment accurately. Extensive experiments on three benchmark datasets show that the proposed VSLNet and VSLNet-L outperform the state-of-the-art methods; VSLNet-L addresses the issue of performance degradation on long videos. Our study suggests that the span-based QA framework is an effective strategy to solve the NLVL problem.

Index Terms—Natural language video localization, single video moment retrieval, temporal sentence grounding, cross-modal retrieval, multi-modal learning, span-based question answering, multi-paragraph question answering, cross-modal interaction

1 INTRODUCTION

NATURAL language video localization (NLVL) is a prominent yet challenging problem in vision-language understanding. Given an untrimmed video, NLVL is to retrieve a temporal moment that semantically corresponds to a given language query. As illustrated in Fig. 1, NLVL involves both computer vision and natural language processing techniques [1], [2], [3], [4], [5], [6]. Cross-modal reasoning is essential for NLVL to correctly locate the target moment in a video. Prior studies primarily treat NLVL as a ranking task, which apply multimodal matching architecture to find the best matching video segment for a query [7], [8], [9], [10], [11]. Some works [11], [12], [13], [14] assign multi-scale temporal anchors to frames and select the anchor with the highest confidence as the result. Recently, several methods explore to model cross-interactions between video and

query, and to regress temporal locations of target moment directly [15], [16], [17]. There are also studies that formulate NLVL as a sequential decision-making problem and solve it with reinforcement learning [18], [19], [20].

Different from the aforementioned works, we address the NLVL task from a new perspective, i.e., span-based question answering (QA). Specifically, the essence of NLVL is to search for a video moment as the answer to a given language query from an untrimmed video. By treating the video as a text passage, and the target moment as the answer span, NLVL shares significant similarities with the span-based QA conceptually. With this intuition, NLVL could be revisited in the span-based QA framework.

However, the existing span-based QA methods could not be directly applied to solve the NLVL problem, due to the following two technical gaps. First, the data nature is different. To be specific, video is continuous, which results in the continuous causal relation inference between two consecutive video events; natural language, on the other hand, is discrete, and words in a sentence demonstrate syntactic structure. Therefore, changes between consecutive video frames are usually smooth. In contrast, two word tokens may carry different and even totally different meanings, e.g., oxymoron or negation. As a result, events in a video are temporally correlated, and one can be linked with one another along the video sequence. The relationships between words or sentences are usually indirect and can be far apart. Second, small shifts in video frames are less imperceptible to humans than words in a text. *Videlicet*, small drifts between frames usually do not affect the understanding of video content. In contrast, the change of a few words or even one word could change the meaning of a sentence entirely.

- Hao Zhang is with the Institute of High Performance Computing, A*STAR, Singapore 138632, and also with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798. E-mail: hzhang26@outlook.com.
- Aixin Sun is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798. E-mail: axsun@ntu.edu.sg.
- Wei Jing is with the Institute of Infocomm Research, A*STAR, Singapore 138632. E-mail: 21wjing@gmail.com.
- Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh are with the Institute of High Performance Computing, A*STAR, Singapore 138632. E-mail: {zhenll, gohsm}@ihpc.a-star.edu.sg, joey.tianyi.zhou@gmail.com.

Manuscript received 30 Sept. 2020; revised 14 Jan. 2021; accepted 14 Feb. 2021. Date of publication 23 Feb. 2021; date of current version 1 July 2022.

(Corresponding author: Joey Tianyi Zhou.)

Recommended for acceptance by D. Damen.

Digital Object Identifier no. 10.1109/TPAMI.2021.3060449

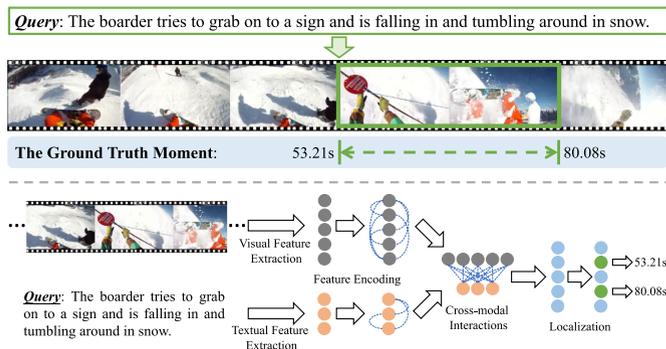


Fig. 1. An illustration of localizing a temporal moment in an untrimmed video by a given language query, and the general procedures of natural language video localization.

By considering the above differences in data nature, we propose a *video span localizing network (VSLNet)*, where a *query-guided highlighting (QGH)* strategy is introduced on the top of the traditional span-based QA framework [21]. In QGH, we consider a region that covers the target moment by extending its starting and ending frames a bit further. In this way, the selected region is regarded as foreground, while the rest is treated as background.

One challenge in NLVL is that the performance of many existing methods degrades significantly along with the increase of video length (see detailed discussion in Section 4.4). Since the NLVL models generally perform well on short videos, one straightforward solution to address this issue is to split a long video into multiple short clip segments. Then each clip segment is regarded as a short video. By treating a long video as a document, a clip segment as a paragraph, NLVL can be viewed as the multi-paragraph question answering (MPQA) task [22]. The target moment in a long video can be considered as the answer span in a document for a given query.

However, how to properly split long video into clip segments is still challenging. Paragraphs in a document are semantically coherent units with boundaries defined by humans. Videos are continuous, and splitting the video into semantically coherent clip segments is difficult, even if it is feasible. In addition, the answer span in MPQA can be found in one of the paragraphs, but we cannot expect the target moment can be found within a single clip segment, regardless of how to split the videos. We propose a *multi-scale split-and-concat strategy* to partition long video into clips of different lengths. Compared with fixed length splitting, the multi-scale splitting strategy increases the chance of locating a target moment in one segment. In this way, even if a target moment is truncated at one or several scales, segments in other scales may still be able to fully contain it. Thus, we can locate the moment in the clips that are more likely to contain it. This network is termed as *VSLNet-L*¹ in the following context.

This paper is a substantial extension of our conference publication [23] with the following improvements. First, we investigate the localization performance degradation issue of existing NLVL models on long videos, and propose VSLNet-L to tackle this problem by introducing the concept

1. “L” represents the multi-scale split-and-concat strategy for Long videos.

of MPQA. Second, we carry out more experimental analyses involving the VSLNet-L and demonstrate its effectiveness on long videos. The contribution and novelty of this work are summarized as follows:

- 1) We provide a new perspective to solve NLVL by formulating it as span-based QA, and analyze the natural differences between them in detail. To the best of our knowledge, this is one of the first works to adopt a span-based QA framework for NLVL.
- 2) We propose VSLNet to explicitly address the differences between NLVL and span-based QA, by introducing a novel query-guided highlighting strategy.
- 3) In addition to VSLNet, VSLNet-L is proposed by incorporating the concept of multi-paragraph QA to address the performance degradation on long videos.

2 RELATED WORK

2.1 Natural Language Video Localization

As introduced by [24], [25], NLVL requires to model the cross-modality interactions between natural language texts and untrimmed videos, to retrieve video segments with language queries. Existing work on NLVL can be roughly divided into five categories: ranking, anchor, reinforcement learning, regression and span based methods.

Ranking-based methods [7], [8], [9], [10], [24], [26], [27], [28] treat NLVL as a ranking task and rely on multimodal matching architectures to find the best matching video moment for a language query. For instance, Gao *et al.* [25] proposed a cross-modal temporal regression localizer to jointly model the queries, as well as predicting alignment scores and action boundary regressions for pre-segmented candidate clips. Chen *et al.* [29] proposed a semantic activity proposal that integrates sentence queries’ semantic information into the proposal generation process to get discriminative activity proposals. Although intuitive, these models are sensitive to negative samples. Specifically, they need densely sampled candidates to achieve good performance, leading to low efficiency and low flexibility.

Various approaches have been developed to overcome the above-mentioned drawbacks. Anchor-based methods [11], [12], [13], [14], [30], [31] sequentially process the video frames and assign each frame with multiscale temporal anchors; then the anchor with the highest confidence is selected as the result. For instance, Yuan *et al.* [32] proposed a semantic conditioned dynamic modulation for better correlating sentence-related video contents over time and establishing a precise matching relationship between sentence and video. Zeng *et al.* [33] designed a dense regression network to regress the distances from each frame to the start/end frame of the video segment described by the query. Liu *et al.* [30] presented a cross- and self-modal graph attention network that recasts NLVL as a process of iterative messages passing over a joint graph to capture high-order interactions between two modalities.

Reinforcement learning-based methods formulate NLVL as a sequential decision-making problem [18], [20]. This process imitates humans’ coarse-to-fine decision-making scheme to observe candidate moments conditioned on queries. For example, Wang *et al.* [18] proposed a recurrent

neural network-based reinforcement learning model to selectively observe a sequence of frames and associate the given sentence with the video content in a matching-based manner. He *et al.* [19] presented a multi-task reinforcement learning framework to learn an agent that regulates the temporal grounding boundaries progressively based on its policy. Wu *et al.* [20] presented a tree-structured policy-based progressive reinforcement learning framework to simulate humans' coarse-to-fine decision-making paradigm and to sequentially regulate the temporal boundary in an iterative refinement manner.

Regression-based methods [15], [17] tackle NLVL by regressing the temporal time of target moments directly. Yuan *et al.* [15] built a proposal-free model with BiLSTM and a three-step attention module to regress temporal locations of the target moment. Lu *et al.* [16] proposed a dense bottom-up framework, which regresses the distances to start and end boundaries for each frame in the target moment. Then, it selects the one with the highest confidence as the result. Chen *et al.* [17] further extended the bottom-up framework in [16] with graph-based feature pyramid network to boost the performance. Mun *et al.* [34] proposed to extract a collection of semantic phrases in the query and reflect bi-modal interactions between the linguistic and visual features in multiple levels for moment boundaries regression.

Recently, a few attempts have been made to address NLVL that consider the concept of question answering [35], [36]. Specifically, Chen *et al.* [35] introduced a cross-gated attended recurrent network and a self interactor to exploit the fine-grained interactions between the query and video. A segment localizer is adopted to predict the starting and ending boundary of the moment. Ghosh *et al.* [37] presented an extractive approach that predicts the start and end frames by leveraging cross-modal interactions between the text and video. However, they did not state and explain the similarity and differences between NLVL and the traditional span-based QA; and they did not adopt the standard span-based QA framework. In this work, we adopt a standard span-based QA framework to address NLVL; and propose VSLNet to explicitly address the issue caused by the differences between NLVL and the traditional span-based QA tasks.

In addition, several other methods have been applied to NLVL, Escorcia *et al.* [38] extends NLVL to a general case that the model is required to retrieve the target moments from a video corpus instead of from a single video. Shao *et al.* [39] presented a unified framework to learn video-level matching and moment-level localization jointly. Mithun *et al.* [40] explored to build a joint visual-semantic embedding based framework to learn latent alignment between video frames and sentence descriptions under weakly supervised setting. Lin *et al.* [41] proposed a weakly-supervised moment retrieval framework requiring only coarse video-level annotations. Zhang *et al.* [42] modeled the temporal relations between video moments by a two-dimensional map and proposed a temporal adjacent network to encode discriminative features for matching video moments with referring expressions. Chen *et al.* [43] presented a method for learning pairwise modality interactions to exploit complementary information better and improve performance on both temporal sentence localization and event captioning.

2.2 Span-Based Question Answering

Span-based Question Answering (QA) has been widely studied in the past years. Wang *et al.* [44] combined match-LSTM [45] and Pointer-Net [46] to estimate boundaries of the answer span. BiDAF [47] introduced bi-directional attention to obtain query-aware context representation. Xiong *et al.* [48] proposed a coattention network to capture the interactions between context and query. R-Net [49] integrated mutual and self attentions into the RNN encoder for feature refinement. QANet [21] leveraged on a similar attention mechanism in a stacked convolutional encoder to improve performance. FusionNet [50] presented a full-aware multi-level attention to capture complete query information. By treating input video as a text passage, the above frameworks are all applicable to NLVL in principle. However, these frameworks are not designed to consider the differences between video and text passage. Their modeling complexity arises because of the interactions between query and text passage, where the two inputs are in the homogeneous space. In our proposed method, VSLBase adopts a simple and standard span-based QA framework, making it easy to model the differences between video and text by adding additional modules. Our VSLNet addresses the issue caused by the differences with the QGH module.

Moreover, VSLNet-L is conceptually similar to the multi-paragraph question answering (MPQA) framework [22], [51], [52], [53], where both methods explore to locate an answer span from multiple paragraphs/clips. Clark *et al.* [22] adopted standard span-based QA models to entire documents scenario by introducing confidence modules to find an answer span from multiple paragraphs. Wang *et al.* [51] proposed to solve MPQA by jointly training three modules that can predict the result based on answer boundary, answer content, and answer verification factors. Wang *et al.* [52] presented a multi-passage BERT model to globally normalize answer scores across all passages of the same question. Lin *et al.* [53] proposed a learning to rank model with an attention module to select the best-matching paragraph for a question. Pang *et al.* [54] presented a three-level hierarchical answer spans model to extract the answer from multi-paragraphs.

3 METHODOLOGY

In this section, we first describe how to address the NLVL task by adopting a span-based QA framework. Then, we present VSLBase (Sections 3.2, 3.3, and 3.4) and VSLNet (Section 3.5) in detail. Lastly, we detail VSLNet-L (Section 3.6) with multi-scale split-and-concat strategy. The model architectures are shown in Fig. 2.

3.1 Span-Based QA for NLVL

We denote the untrimmed video as $V = [f_1, f_2, \dots, f_T]$ and the language query as $Q = [q_1, q_2, \dots, q_m]$, where T and m are the number of frames and words, respectively. τ^s and τ^e represent the start and end time of the temporal moment i.e., answer span. To address NLVL with the span-based QA framework, we transform the data into a set of SQuAD style triples (*Context, Question, Answer*) [55]. For each video V , we extract its visual feature sequence $V = [v_1, v_2, \dots, v_n]$ by a pre-trained 3D ConvNet [56], where n is

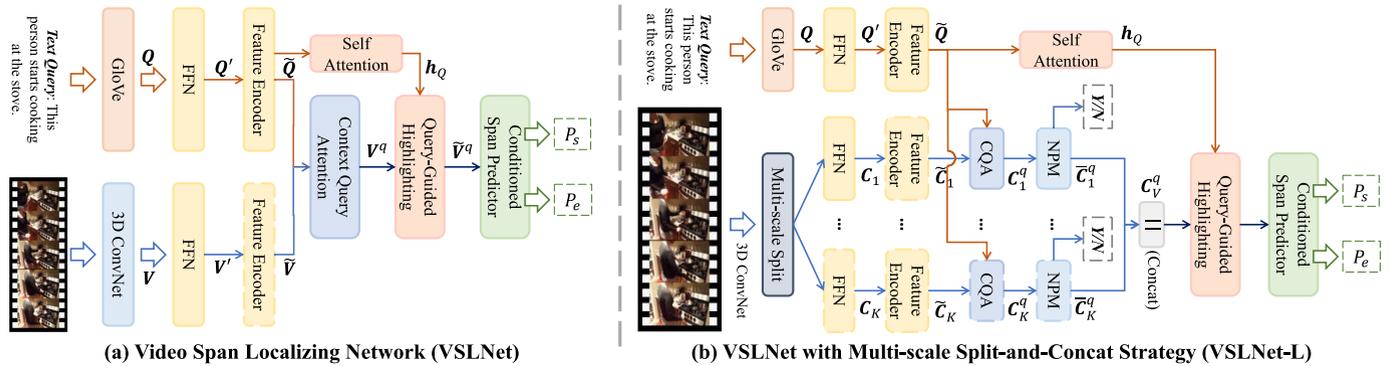


Fig. 2. An overview of the proposed architectures for NLVL. The feature extractors, i.e., GloVe and 3D ConvNet, are fixed during training. (a) depicts the structure of VSLNet. (b) shows the architecture of VSLNet-L. Note the standard span-based QA framework (VSLBase) is similar to VSLNet by removing the Self-Attention and Query-Guided Highlighting modules.

the number of extracted feature vectors. Here, V can be regarded as the sequence of word embeddings for a text passage with n tokens, as in the traditional span-based QA framework. Similar to word embeddings, each feature vector v_i here is a video feature vector.

Since span-based QA aims to predict start and end boundaries of an answer span, the start/end time of a video sequence needs to be mapped to the corresponding boundaries in the visual feature sequence V . Suppose the video duration is T , the start (end) span index is calculated by $a^{s(e)} = \text{Round}(\tau^{s(e)}/T \times n)$, where $\text{Round}(\cdot)$ denotes the rounding operator. During the inference, the predicted span boundary can be easily converted to the corresponding time via $\tau^{s(e)} = a^{s(e)}/n \times T$. After transforming moment annotations in the NLVL dataset, we obtain a set of (V, Q, A) triples. Visual feature sequence $V = [v_1, v_2, \dots, v_n]$ acts as the passage with n tokens; $Q = [q_1, q_2, \dots, q_m]$ is the query with m tokens, and the answer $A = [v_{a^s}, v_{a^s+1}, \dots, v_{a^e}]$ corresponds to a piece in the passage. Then, the NLVL task becomes to find the correct start and end boundaries of the answer span, a^s and a^e .

3.2 Feature Encoder

After obtaining visual features $V = [v_1, v_2, \dots, v_n]^T \in \mathbb{R}^{n \times d_v}$, for a text query Q_r we can compute its word embeddings $Q = [q_1, q_2, \dots, q_m]^T \in \mathbb{R}^{m \times d_q}$ with existing text embedding approaches, e.g., GloVe. We project the video and text feature vectors into the same dimension d , $V' \in \mathbb{R}^{n \times d}$ and $Q' \in \mathbb{R}^{m \times d}$, by two linear layers. Then we build the feature encoder with a simplified version of the embedding encoder layer in QANet [21].

Instead of applying a stack of multiple encoder blocks, we use only one encoder block, as shown in Fig. 3. This encoder block consists of four convolution layers, followed by a multi-head attention layer [57]. A feed-forward layer is used to produce the output. Layer normalization [58] and residual connection [59] are applied to each layer. The encoded visual feature sequence and word embeddings are as follows:

$$\begin{aligned} \tilde{V} &= \text{FeatureEncoder}(V') \\ \tilde{Q} &= \text{FeatureEncoder}(Q'). \end{aligned} \quad (1)$$

The parameters of feature encoder are shared by visual features and word embeddings.

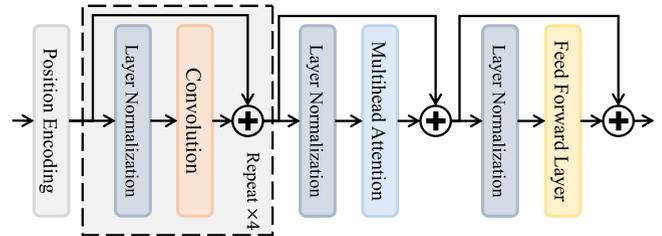


Fig. 3. The structure of feature encoder.

3.3 Context-Query Attention

After feature encoding, we utilize the context-query attention (CQA) [21], [47], [48] to capture the cross-modal interactions between visual and textual features. CQA first calculates the similarity scores, $S \in \mathbb{R}^{n \times m}$, between each visual feature and query feature. Then context-to-query (A) and query-to-context (B) attention weights are computed as:

$$A = S_r \cdot \tilde{Q} \in \mathbb{R}^{n \times d}, B = S_c^T \cdot \tilde{V} \in \mathbb{R}^{n \times d}, \quad (2)$$

where S_r and S_c are the row- and column-wise normalization of S by SoftMax, respectively. Finally, the output of context-query attention is written as:

$$V^q = \text{FFN}([\tilde{V}; A; \tilde{V} \odot A; \tilde{V} \odot B]), \quad (3)$$

where $V^q \in \mathbb{R}^{n \times d}$; FFN is a single feed-forward layer; \odot denotes an element-wise multiplication.

3.4 Conditioned Span Predictor

We construct a conditioned span predictor using two unidirectional LSTMs and two feed-forward layers, inspired by Ghosh *et al.* [37]. The main difference between our method and the method in [37] is that we use unidirectional LSTM instead of bidirectional LSTM. We observe that unidirectional LSTM shows similar performance but with fewer parameters and higher efficiency. We believe that the unidirectional LSTM works as well as bidirectional LSTM in this setting for two possible reasons. First, video is a temporal sequence where events happen earlier lead to later events, but not in the other way round. Second, the end boundary of a target moment always appears after the start boundary in the sequence. Therefore, we stack the two LSTMs so that the LSTM for end boundary is conditioned on the LSTM for

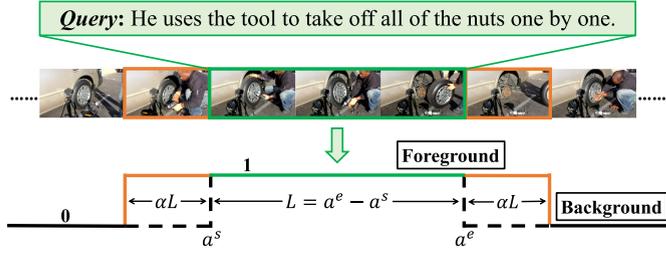


Fig. 4. An illustration of foreground and background of visual features. α is the ratio of foreground extension.

start boundary, to maintain the sequence nature of video. The hidden states of the two LSTMs are then fed into the corresponding feed-forward layers to compute the start and end scores:

$$\begin{aligned} h_t^s &= \text{UniLSTM}_{\text{start}}(v_t^q, h_{t-1}^s) \\ h_t^e &= \text{UniLSTM}_{\text{end}}(h_t^s, h_{t-1}^e) \\ S_t^s &= W_s([h_t^s; v_t^q]) + b_s \\ S_t^e &= W_e([h_t^e; v_t^q]) + b_e, \end{aligned} \quad (4)$$

where $S_t^{s/e} \in \mathbb{R}^n$ denote the scores of start/end boundaries at position t ; v_t^q represents the t th element in V^q . $W_{s/e}$ and $b_{s/e}$ denote the weight matrix and bias of the start/end feed-forward layer, respectively. Then, the probability distributions of start/end boundaries are computed by $P_{s/e} = \text{SoftMax}(S^{s/e}) \in \mathbb{R}^n$, and the training objective is defined as follows:

$$\mathcal{L}_{\text{span}} = \frac{1}{2} [f_{\text{CE}}(P_s, Y_s) + f_{\text{CE}}(P_e, Y_e)], \quad (5)$$

where f_{CE} is the cross-entropy loss function; Y_s and Y_e are the labels for the start (a^s) and end (a^e) boundaries, respectively. During the inference, the predicted answer span (\hat{a}^s, \hat{a}^e) of a query is generated by maximizing the joint probability of start and end boundaries as:

$$\begin{aligned} \text{span}(\hat{a}^s, \hat{a}^e) &= \arg \max_{\hat{a}^s, \hat{a}^e} P_s(\hat{a}^s) P_e(\hat{a}^e) \\ \text{s.t. } &0 \leq \hat{a}^s \leq \hat{a}^e \leq n. \end{aligned} \quad (6)$$

The description of VSLBase architecture has been completed. VSLNet is built on top of VSLBase with QGH, to be detailed in Section 3.5.

3.5 Query-Guided Highlighting

The Query-Guided Highlighting (QGH) strategy is introduced in VSLNet, to address the major differences between text span-based QA and NLVL tasks, as shown in Fig. 2a. With the QGH strategy, we consider the target moment as the foreground, and the rest as background, illustrated in Fig. 4. The target moment, which is aligned with the language query, starts from a^s and ends at a^e with length $L = a^e - a^s$. QGH extends the boundaries of the foreground to cover its antecedent and consequent video contents, where the extension ratio is controlled by a hyperparameter α . As aforementioned in Introduction, the extended boundary could potentially cover additional contexts and also help the network to focus on subtle differences among video frames.

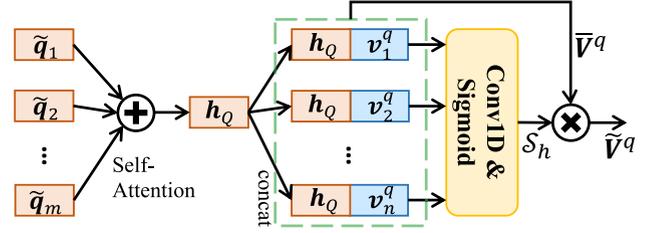


Fig. 5. The structure of query-guided highlighting (QGH).

By assigning 1 to the foreground and 0 to the background, we obtain a binary sequence, denoted as Y_h . QGH is a binary classification module to predict the confidence a visual feature vector belongs to the foreground or background. The structure of QGH is shown in Fig. 5. We first encode word feature representations \tilde{Q} into sentence representation (denoted by h_Q), with self-attention mechanism [60]. Then h_Q is concatenated with each feature element in V^q as $\tilde{V}^q = [\tilde{v}_1^q, \dots, \tilde{v}_n^q]$, where $\tilde{v}_i^q = [v_i^q; h_Q]$. The highlighting score and highlighted feature representations are computed as:

$$\begin{aligned} S_h &= \sigma(\text{Conv1D}(\tilde{V}^q)) \\ \tilde{V}^q &= S_h \cdot \tilde{V}^q, \end{aligned} \quad (7)$$

where $S_h \in \mathbb{R}^n$; σ denotes the Sigmoid activation; \cdot represents multiplication operator, which multiplies each feature of \tilde{V}^q and the corresponding score of S_h . Accordingly, feature V^q in Equation 4 is replaced by \tilde{V}^q in VSLNet to compute $\mathcal{L}_{\text{span}}$. The loss function of query-guided highlighting is formulated as:

$$\mathcal{L}_{\text{QGH}} = f_{\text{CE}}(S_h, Y_h). \quad (8)$$

VSLNet is trained in an end-to-end manner by minimizing the following overall loss:

$$\mathcal{L} = \mathcal{L}_{\text{span}} + \mathcal{L}_{\text{QGH}}. \quad (9)$$

3.6 Multi-Scale Split-and-Concat

To address the localization performance degradation issues on long videos, we introduce VSLNet-L with a multi-scale split-and-concatenation strategy. The architecture of VSLNet-L is shown in Fig. 2b.

3.6.1 Split: Clip Segment Level Detection

As illustrated in Fig. 6a, given a long video, VSLNet-L splits it into K clip segments:

$$V = [C_k]_{k=1}^K \quad \text{and} \quad C_k = [v_i]_{i=(k-1)l}^{kl}, \quad (10)$$

where l is the length of each clip segment C_k , i.e., $K \times l = n$. Note that, in our implementation, we perform video split at visual feature level, instead of the untrimmed video itself for computational efficiency. Specifically, we split the features $V = [v_i]_{i=1}^n$ obtained from the pre-trained 3D ConvNet (see Section 3.1), and use the feature vector V in Eq. (10) accordingly.

Each clip segment C_k is then processed by feature encoder and CQA, to learn query-attended representations

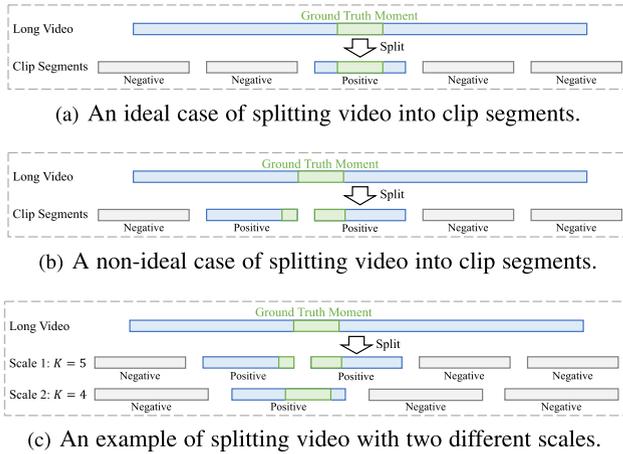


Fig. 6. Illustration of splitting video into clip segments.

$C_k^q = [c_{i=(k-1) \times l}^q]^{k \times l} \in \mathbb{R}^{l \times d}$, as shown in Fig. 2b. Thus, C_k^q encodes the cross-modal features between clip segment k and query. Then, a Nil Prediction Module (NPM) is introduced in VSLNet-L, to predict whether a clip segment contains or partially contains the temporal moment that corresponds to the text query, as shown in Fig. 2b. Next, we detail the structure of the NPM following the illustration in Fig. 7.

For each clip segment, its query-attended features C_k^q are first encoded by a feature encoder as:

$$\hat{C}_k^q = \text{FeatureEncoder}_{\text{NPM}}(C_k^q). \quad (11)$$

The self-attention mechanism [60] is adopted to encode \hat{C}_k^q into clip representation $h_{C_k}^q$, and the nil-score is computed as:

$$\begin{aligned} \alpha_k &= \text{SoftMax}(\text{Conv1D}(\hat{C}_k^q)) \\ h_{C_k}^q &= \sum_{i=1}^l \alpha_{k,i} \cdot \hat{c}_{(k-1) \times l + i}^q \\ S_{\text{nil}}^k &= \sigma(\text{FFN}(h_{C_k}^q)), \end{aligned} \quad (12)$$

where $\alpha_k \in \mathbb{R}^l$ and $h_{C_k}^q \in \mathbb{R}^d$. $S_{\text{nil}}^k \in \mathbb{R}$ is a scalar, which indicates the confidence of clip segment k containing the ground truth moment. The loss of NPM is formulated as:

$$\mathcal{L}_{\text{NPM}} = f_{\text{CE}}(S_{\text{nil}}, Y_{\text{nil}}), \quad (13)$$

Y_{nil} is a 0-1 sequence provided during training. A clip segment is positive (label 1) iff it overlaps with ground truth moment, illustrated in Fig. 6. Clip segments that do not contain ground truth moment are negative (label 0). After computing the nil-scores of all clip segments, i.e., $S_{\text{nil}} = [S_{\text{nil}}^1, \dots, S_{\text{nil}}^K] \in \mathbb{R}^K$, we normalize the scores. The output for clip segment k is:

$$\bar{S}_{\text{nil}} = \frac{S_{\text{nil}}}{\max(S_{\text{nil}})}, \quad \bar{C}_k^q = \bar{S}_{\text{nil}}^k \times \hat{C}_k^q, \quad (14)$$

where \bar{S}_{nil} is the normalized nil-score and \bar{C}_k^q is the re-weighted representations of clip segment k . The NPM highlights the clip segments that contain the target moment, and suppresses other segments. Then the subsequent modules could localize the result by focusing more on the highlighted segments, which is equivalent to narrowing

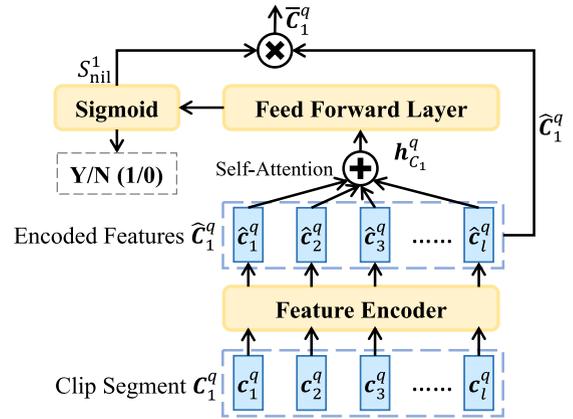


Fig. 7. The structure of nil prediction module (NPM).

down the searching scope from a long video to a short segment of it.

3.6.2 Concat: Video Level Localization

With the clip segments processed separately in the previous step, we now concatenate the representations of all clip segments, for two reasons. First, a single clip segment may not fully cover the target moment. Second, even if a segment is predicted to be negative (or low confidence), it might provide useful contextual information for localizing the target moment.

$$\bar{C}_V^q = [\bar{C}_1^q \parallel \bar{C}_2^q \parallel \dots \parallel \bar{C}_K^q], \quad (15)$$

where \parallel denotes the concatenation operator and $\bar{C}_V^q \in \mathbb{R}^{n \times d}$. Accordingly, the input feature V^q of QGH is replaced by \bar{C}_V^q in VSLNet-L to compute \mathcal{L}_{QGH} and $\mathcal{L}_{\text{span}}$. The combined training loss for VSLNet-L is:

$$\mathcal{L} = \mathcal{L}_{\text{span}} + \mathcal{L}_{\text{QGH}} + \mathcal{L}_{\text{NPM}}. \quad (16)$$

3.6.3 Multi-Scale Split-and-Concat

Unlike a text document, there are no paragraphs as semantic units in a video. Any split in the video may break important contextual information to different clip segments. Although all clip segments will be concatenated again for video level localization, each clip segment is processed separately, and the contextual information between two segments may not be well captured.

To address this issue, we propose a multi-scale split mechanism, to split the video at different segment lengths, i.e., different K (see Fig. 6c). Suppose we use N_s different scales, and for each scale we have:

$$K_i \times l_i = n, \quad \forall i = 1, 2, \dots, N_s, \quad (17)$$

where K_i and l_i denote the number of clip segments and clip segment length for the i th scale, respectively. Through a multi-scale split, contextual information is well captured. Meanwhile, a multi-scale split also provides variation in training samples for the same video and query pair input. The target moment may be located in multiple different clip segments, and its contexts are also different. Note that the same training process as the single-scale split-and-concat

TABLE 1
Statistics of the Natural Language Video Localization (NLVL) Datasets

Dataset	Domain	# Videos (train/val/test)	# Annotations	N_{vocab}	\bar{L}_{video}	\bar{L}_{query}	\bar{L}_{moment}	Δ_{moment}
Charades-STA	Indoors	5,338/–/1,334	12,408/–/3,720	1,303	30.59s	7.22	8.22s	3.59s
ANetCap	Open	10,009/–/4,917	37,421/–/17,505	12,460	117.61s	14.78	36.18s	40.18s
TACoS _{org}	Cooking	75/27/25	10,146/4,589/4,083	2,033	287.14s	10.05	5.45s	7.56s
TACoS _{tan}	Cooking	75/27/25	9,790/4,436/4,001	1,983	287.14s	9.42	25.26s	50.71s

Note that N_{vocab} is the vocabulary size of lowercase words, \bar{L}_{video} denotes the average length of videos in seconds, \bar{L}_{query} denotes the average number of words in sentence query, \bar{L}_{moment} is the average length of temporal moments in seconds, and Δ_{moment} is the standard deviation of temporal moment length in seconds.

applies, except that the clip segments of each scale are fed separately into VSLNet-L, to optimize the objective.

Consequently, we derive N_s predicted moments for a given video and language query pair, because the clip segments at each scale would lead to a pair of start/end boundaries for a predicted moment. During inference, we adopt two simple candidate selection strategies to derive the final target moment. VSLNet-L- P_m strategy selects the candidate with the highest joint boundary probability, $P_m = \max\{P_{\text{span}}^i\}_{i=1}^{N_s}$, where $P_{\text{span}}^i = P_s^i(\hat{a}^s)P_e^i(\hat{a}^e)$ is the maximal joint boundary probability of the moment generated by i th scale using Eq. (6). VSLNet-L- U strategy selects two moments with the largest overlap from N_s candidates, and computes their union as the final result.

4 EXPERIMENTS

4.1 Datasets

We conduct experiments on three benchmark datasets: Charades-STA [25], ActivityNet Captions [61], and TACoS [62], as summarized in Table 1. *Charades-STA* is prepared by Gao *et al.* [25] based on Charades dataset [63], with the videos of daily indoor activities. There are 12,408 and 3,720 moment annotations for training and test, respectively. *ActivityNet Captions (ANetCap)* contains about 20k videos taken from ActivityNet [64]. We follow the setup in Yuan *et al.* [15], with 37,421 moment annotations for training, and 17,505 annotations for test. *TACoS* is selected from MPII Cooking Composite Activities dataset [65]. There are two versions of TACoS available. TACoS_{org} used in [25] has 10,146, 4,589 and 4,083 annotations for training, validation and test. TACoS_{tan} is from 2D-TAN [42], which contains 9,790, 4,436 and 4,001 annotations in train, validation and test sets.

We conduct experiments on all the benchmark datasets for VSLBase and VSLNet. For VSLNet-L, we evaluate it on ActivityNet Captions and both versions of TACoS. Charades-STA is not used for VSLNet-L due to its short video length.

4.2 Experimental Settings

4.2.1 Evaluation Metrics

We adopt “Rank@ n , IoU = μ ” and “mIoU” as the evaluation metrics, following [8], [15], [25]. The “Rank@ n , IoU = μ ” denotes the percentage of language queries having at least one result whose Intersection over Union (IoU) with ground truth is larger than μ in top- n retrieved moments. “mIoU” is the average IoU over all testing samples. In our experiments, we use $n = 1$ and $\mu \in \{0.3, 0.5, 0.7\}$.

4.2.2 Implementation Details

For a text query Q , we lowercase all its words and initialize them with 300d GloVe [66] embeddings. The word embeddings are fixed during training. For the untrimmed video V , we extract its visual features using 3D ConvNet pre-trained on Kinetics dataset [56]. We set the maximal feature length n as 128 for Charades-STA, i.e., the extracted visual feature sequence of a video will be uniformly downsampled if its length $> n$, or zero-padding otherwise. The n of ANetCap is set to 300, while two maximal feature lengths, $n \in \{300, 600\}$, are used for evaluation on TACoS. When evaluating VSLNet-L, the visual features are split into multiple clip segments using different scales, we use $l = \{60, 75, 100, 150\}$ (i.e., $K = \{5, 4, 3, 2\}$) for $n = 300$, and $l = \{100, 120, 150, 200\}$ (i.e., $K = \{6, 5, 4, 3\}$) for $n = 600$. We set the dimension of all the hidden layers in the model as 128; the kernel size of the convolution layer is 7; the head size of multi-head attention is 8. All the models are trained for 100 epochs with a batch size of 16 and an early stopping strategy for all datasets. Adam [67] is used as the optimizer, with a learning rate of 0.0001, linear decay of learning rate and gradient clipping of 1.0. Dropout [68] of 0.2 is applied to prevent overfitting. All experiments are conducted on dual NVIDIA GeForce RTX 2080Ti GPUs workstation.

4.3 Comparison With State-of-the-Arts

We compare the proposed methods with the following state-of-the-arts: CTRL [25], ACRN [8], TGN [12], ACL [9], DEBUG [16], QSPN [10], SM-RL [18], RWM-RL [19], ABLR [15], SCDM [32], TSP-PRL [20], 2D-TAN [42], GDP [17], CBP [14], LGI [34] and DRN [33]. Scores of the benchmark methods in all result tables are those reported in the corresponding papers. The best results are in **bold** and the second bests are underlined.

Among the proposed methods, VSLBase is a direct implementation of span-based QA framework on the NLVL task; VSLNet is the extension of VSLBase with QGH; VSLNet-L is a further extension of VSLNet with multi-scale split-concat strategy, designed to handle long videos more effectively. In the following, we show that VSLBase is comparable to existing baselines on NLVL tasks, while VSLNet surpasses VSLBase and all existing baselines, and achieves state-of-the-art results. We then show that VSLNet-L well addresses the issue of performance degradation on NLVL for long videos, compared to VSLNet.

The results on Charades-STA are reported in Table 2. It is observed that VSLNet significantly outperforms most of the baselines by a large margin over all metrics. In addition, it is

TABLE 2
Results (%) of “Rank@1, IoU = μ ” and “mIoU” Compared With the State-of-the-Art on Charades-STA

Model	Rank@1, IoU = μ			mIoU
	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	
3D ConvNet without fine-tuning as visual feature extractor				
CTRL	-	23.63	8.89	-
ACL	-	30.48	12.20	-
QSPN	54.70	35.60	15.80	-
SAP	-	27.42	13.36	-
SM-RL	-	24.36	11.17	-
RWM-RL	-	36.70	-	-
MAN	-	<u>46.53</u>	22.72	-
DEBUG	54.95	37.39	17.69	36.34
TSP-PRL	-	37.39	17.69	37.22
2D-TAN	-	39.81	23.31	-
CBP	-	36.80	18.87	-
GDP	54.54	39.47	18.49	-
VSLBase	<u>61.72</u>	40.97	<u>24.14</u>	<u>42.11</u>
VSLNet	64.30	47.31	30.19	45.15
3D ConvNet with fine-tuning on Charades dataset				
ExCL	65.10	44.10	23.30	-
SCDM	-	54.44	<u>33.43</u>	-
DRN	-	53.09	<u>31.75</u>	-
VSLBase	68.06	50.23	30.16	<u>47.15</u>
VSLNet	70.46	<u>54.19</u>	35.22	50.02

worth noting that the performance improvements of VSLNet are more significant under larger values of IoU. For instance, VSLNet achieves 7.47 percent improvement in IoU = 0.7 versus 0.78 percent in IoU = 0.5, compared to MAN. Without query-guided highlighting, VSLBase outperforms all compared baselines over IoU = 0.7, which shows adopting the span-based QA framework is promising for NLVL. Moreover, VSLNet benefits from visual feature fine-tuning, and achieves state-of-the-art results on this dataset.

Table 3 reports the results on both versions (if available) of TACoS dataset. In general, VSLNet outperforms previous methods over all evaluation metrics. In addition, with the Split-and-Concat mechanism, VSLNet-L further improves the performance, on top of VSLNet. On TACoS_{org}, the results of VSLNet^(S) is comparable to that of VSLNet^(L), while VSLNet-L^(L) surpasses VSLNet-L^(S) for both candidate selection strategies. Here, L and S denote the maximal video feature length 600 and 300, respectively. Similar observations hold on TACoS_{tan}. These results demonstrate that VSLNet-L is more adept than others at localizing temporal moments in longer videos. Moreover, VSLNet-L- P_m is generally superior to VSLNet-L- U under different n for both versions of the TACoS dataset.

The results on the ActivityNet Captions dataset are summarized in Table 4. We observe that VSLBase shows similar performance to or slightly better than most of the baselines, while VSLNet further boosts the performance of VSLBase significantly. Comparing VSLNet with $n = 128$ and that with $n = 300$, we find that small n leads to better performance on loose metric (e.g., 63.16 versus 61.61 on IoU = 0.3) and large n is beneficial for strict metric (e.g., 26.16 versus 26.54 on IoU = 0.7). Meanwhile, the performance of

TABLE 3
Results (%) of “Rank@1, IoU = μ ” and “mIoU” Compared With the State-of-the-Art on TACoS

Dataset	Model	Rank@1, IoU = μ			mIoU
		$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	
TACoS _{org}	CTRL	18.32	13.30	-	-
	TGN	21.77	18.90	-	-
	ACL	24.17	20.01	-	-
	ACRN	19.52	14.62	-	-
	ABLR	19.50	9.40	-	13.40
	SM-RL	20.25	15.95	-	-
	DEBUG	23.45	11.72	-	16.03
	SCDM	26.11	21.17	-	-
	GDP	24.14	13.90	-	16.18
	CBP	27.31	24.79	19.10	21.59
	DRN	-	23.17	-	-
	VSLNet ^(S)	29.21	24.37	19.37	23.48
	VSLNet ^(L)	29.78	24.71	19.64	23.96
	VSLNet-L- P_m ^(S)	31.94	<u>26.72</u>	22.36	<u>25.71</u>
VSLNet-L- U ^(S)	<u>31.69</u>	26.79	<u>22.02</u>	25.78	
VSLNet-L- P_m ^(L)	32.04	27.92	23.28	26.40	
VSLNet-L- U ^(L)	<u>31.86</u>	<u>27.64</u>	<u>22.72</u>	<u>26.25</u>	
TACoS _{tan}	2D-TAN Pool	37.29	25.32	-	-
	2D-TAN Conv	35.22	25.19	-	-
	VSLNet ^(S)	42.66	32.72	23.12	33.07
	VSLNet ^(L)	41.42	30.67	22.32	31.92
	VSLNet-L- P_m ^(S)	47.66	36.15	26.19	36.24
	VSLNet-L- U ^(S)	47.66	<u>36.12</u>	<u>25.87</u>	<u>35.98</u>
	VSLNet-L- P_m ^(L)	47.11	36.34	26.42	36.61
	VSLNet-L- U ^(L)	<u>46.44</u>	<u>35.74</u>	<u>26.19</u>	<u>36.05</u>

(S) denotes $n = 300$ and (L) represents $n = 600$.

VSLNet-L is comparable to state-of-the-art methods. It is worth noting that 99 percent annotations in ActivityNet Captions belong to videos that are shorter than 4 minutes.

TABLE 4
Results (%) of “Rank@1, IoU = μ ” and “mIoU” Compared With State-of-the-Arts on ActivityNet Captions

Model	Rank@1, IoU = μ			mIoU
	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	
TGN	45.51	28.47	-	-
ACRN	49.70	31.67	11.25	-
ABLR	55.67	36.79	-	36.99
RWM-RL	-	36.90	-	-
QSPN	45.30	27.70	13.60	-
DEBUG	55.91	39.72	-	39.51
SCDM	54.80	36.75	19.86	-
TSP-PRL	56.08	38.76	-	39.21
GDP	56.17	39.27	-	39.80
CBP	54.30	35.76	17.80	-
DRN	-	45.45	24.36	-
LGI	58.52	41.51	23.07	-
2D-TAN Pool	59.45	<u>44.51</u>	26.54	-
2D-TAN Conv	58.75	<u>44.05</u>	<u>27.38</u>	-
VSLBase*	58.18	39.52	23.21	40.56
VSLNet*	63.16	43.22	26.16	43.19
VSLNet	61.61	43.78	26.54	43.22
VSLNet-L- P_m	62.18	43.69	27.22	<u>43.67</u>
VSLNet-L- U	<u>62.35</u>	43.86	27.51	44.06

* denotes that the maximal visual sequence length n of VSLBase and VSLNet is set as 128, which is consistent with [23].

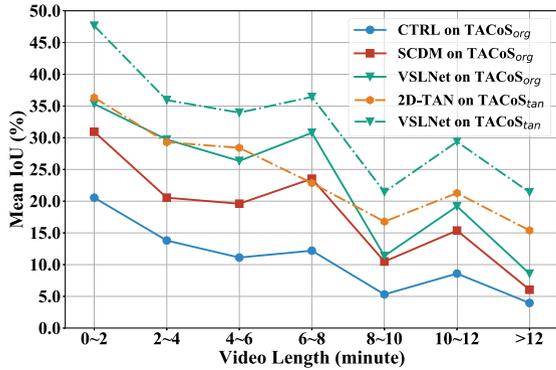


Fig. 8. The Mean IoU (%) performance of CTRL [25], SCDM [32], 2D-TAN Pool [42] and VSLNet on the TACoS dataset.

As VSLNet-L is designed to address performance degradation on long videos, it is reasonable to observe that VSLNet-L achieves less significant performance improvement on ActivityNet Captions compared to TACoS, *w.r.t.* the state-of-the-arts. Moreover, VSLNet-L-U performs better than VSLNet-L- P_m on ActivityNet Captions, different from the observation on TACoS. This could be due to the different ratios of \bar{L}_{moment} and \bar{L}_{video} in the two datasets (see Table 1). The strategy to select longer spans works better on ActivityNet Captions dataset.

4.4 Performance on Videos With Different Length

As discussed in Section 1 and illustrated in Fig. 8, existing methods including VSLNet still underperform on NLVL with long videos. That is, the localization performance decreases dramatically along with the increase of video length. Summarized in Table 5, there are fewer videos/annotations along the increase of video length in the datasets. The relatively small number of training samples may lead to instability of the evaluated models, and performance degradation on long videos, to some extent. However, we believe that the following are the two main reasons for the performance degradation:

- 1) Downsampling of visual features of long videos in most existing methods adversely affects localization accuracy due to information loss. As shown in Fig. 9, sparsely downsampling video feature presentations below certain number (e.g., $n < 200$) would lead to dramatic performance degradation.

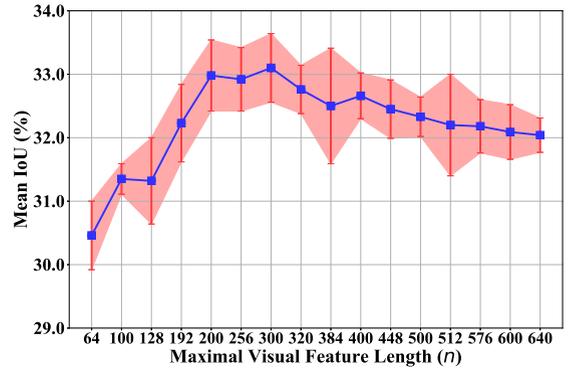


Fig. 9. Mean IoU (%) results of VSLNet on the TACoS_{stan} dataset under different maximal visual representation lengths n .

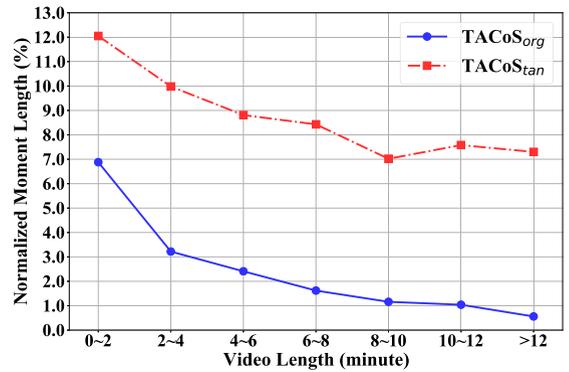


Fig. 10. Statistic of normalized moment lengths in videos for both TACoS_{org} and TACoS_{stan}.

- 2) As plotted in Fig. 10, the average normalized length of ground truth moments gradually decrease along with the increase of video length. The sparsity of moments also contributes to poor performance on long videos.

To address this issue, VSLNet-L splits the video into multiple clip segments with different scales to simulate the multiple paragraphs in the document and maximize the chance of locating the target moment in one segment clip.

Table 6 reports the “mIoU” gains of VSLNet-L on the TACoS dataset for videos with different lengths. Compared to the results of VSLNet (the best performing method without considering video length), larger improvements are observed on longer videos, which demonstrates the

TABLE 5
Statistics of Videos and Annotations *w.r.t.* Different Video Lengths Over NLVL Datasets

Dataset	Split	# Videos	# Annots	# of videos / annotations <i>w.r.t.</i> different video lengths						
				0 ~ 2 min	2 ~ 4 min	4 ~ 6 min	6 ~ 8 min	8 ~ 10 min	10 ~ 12 min	> 12 min
TACoS _{org}	Train	75	10,146	27 / 2,847	29 / 4,015	8 / 1,284	4 / 607	3 / 616	2 / 328	2 / 449
	Val	27	4,589	3 / 312	5 / 771	5 / 887	7 / 1,275	1 / 173	4 / 830	2 / 341
	Test	25	4,083	5 / 578	6 / 937	3 / 564	3 / 447	2 / 373	3 / 617	3 / 567
TACoS _{stan}	Train	75	9,790	27 / 2,769	29 / 3,840	8 / 1,227	4 / 576	3 / 597	2 / 336	2 / 445
	Val	27	4,436	3 / 311	6 / 929	4 / 639	7 / 1,225	1 / 171	4 / 812	2 / 349
	Test	25	4,001	5 / 594	6 / 907	3 / 535	3 / 428	2 / 370	3 / 598	3 / 569
ANetCap	Train	10,009	37,421	5,278 / 17,806	4,715 / 19,551	8 / 28	3 / 10	4 / 22	0 / 0	1 / 4
	Test	4,917	17,505	2,516 / 8,193	2,392 / 9,274	5 / 19	2 / 9	0 / 0	1 / 5	1 / 5

TABLE 6
Comparison of mIoU (%) Between VSLNet and VSLNet-L on TACoS Dataset *w.r.t.* Different Video Lengths

Data	Video Length	0 ~ 2 min	2 ~ 4 min	4 ~ 6 min	6 ~ 8 min	8 ~ 10 min	10 ~ 12 min	> 12 min
TACoS _{org}	# Test Annotations	578	937	564	447	373	617	567
	VSLNet ^(S)	38.93	27.68	21.56	28.95	12.12	20.59	9.01
	VSLNet-L- P_m ^(S)	39.91 +0.98	28.11 +0.43	24.63 +3.07	30.45 +1.50	15.26 +3.14	24.56 +3.97	12.75 +3.74
	VSLNet-L- U ^(S)	39.54 +0.61	28.69 +1.01	24.85 +3.29	30.37 +1.42	15.27 +3.15	24.87 +4.28	12.15 +3.14
	VSLNet ^(L)	35.32	29.72	26.34	30.78	11.37	19.19	8.58
	VSLNet-L- P_m ^(L)	36.11 +0.79	29.84 +0.12	27.85 +1.51	31.88 +1.10	15.68 +4.31	26.50 +7.31	11.90 +3.32
VSLNet-L- U ^(L)	35.75 +0.43	29.98 +0.26	28.21 +1.87	31.78 +1.00	15.31 +3.94	25.37 +6.18	12.26 +3.68	
TACoS _{tan}	# Test Annotations	594	907	535	428	370	598	569
	VSLNet ^(S)	47.64	35.93	33.95	36.43	21.44	29.37	21.40
	VSLNet-L- P_m ^(S)	46.12 -1.52	37.76 +1.83	36.51 +2.56	41.37 +4.94	29.71 +8.27	34.84 +5.47	25.08 +3.68
	VSLNet-L- U ^(S)	45.91 -1.73	37.52 +1.59	36.85 +2.90	41.85 +5.42	28.30 +6.86	33.84 +4.47	25.20 +3.80
	VSLNet ^(L)	46.73	34.19	33.34	33.11	18.66	31.71	19.45
	VSLNet-L- P_m ^(L)	44.32 -2.41	36.86 +2.67	37.70 +4.36	40.15 +7.04	29.99 +11.33	39.33 +7.62	25.88 +6.43
VSLNet-L- U ^(L)	43.56 -3.17	36.74 +2.55	37.90 +4.56	40.06 +6.95	27.97 +9.31	39.01 +7.30	24.48 +5.03	

(*S*) denotes $n = 300$ and (*L*) represents $n = 600$. Performance *gain* and *loss* are indicated in different colors.

superiority of VSLNet-L for localizing temporal moments in long videos. For instance, VSLNet-L^(L) achieves more than 3 percent absolute improvements in mIoU for videos longer than 8 minutes versus less than 2 percent gains for videos shorter than 8 minutes on TACoS_{org}. Despite the slight performance reduction on videos shorter than 2 minutes, along with video length raises, consistent improvements are observed on TACoS_{tan} for both $n = 300$ (*S*) and 600 (*L*), compared to VSLNet. Fig. 11 plots the performance improvements along video lengths for better visualization.

Results on ActivityNet Captions are reported in Table 7. Despite that the videos are relatively short, VSLNet-L manages to improve localization performance, with larger improvements observed on longer videos. These results show consistent superiority of VSLNet-L over VSLNet for both candidate selection strategies, on videos of different lengths.

4.5 Ablation Studies

In this section, we conduct ablative experiments to analyze the importance of different modules, including feature encoder and context-query attention in VSLBase. We also investigate the impact of extension ratio α (see Fig. 4) in

query-guided highlighting (QGH) of VSLNet, and study the impact of the multi-scale split strategy in VSLNet-L. Finally, we visually show the effectiveness of the proposed methods and discuss their limitations. We conduct the ablation studies in this order because VSLNet is built on top of VSLBase, and VSLNet-L is an extension of VSLNet.

4.5.1 Module Analysis

We first study the effectiveness of the feature encoder and context-query attention (CQA) in VSLBase by replacing them with other modules. Specifically, we use bidirectional LSTM (BiLSTM) as an alternative feature encoder. For context-query attention, we replace it by a simple method (named CAT), which concatenates each visual feature with the max-pooled query feature.

Our feature encoder consists of Convolution + Multi-head attention + Feed-forward layers (see Section 3.2), named as CMF here. With these alternatives, we now have four combinations, listed in Table 8. As observed from the results, CMF shows stable superiority over CAT on all metrics regardless of other modules; CQA surpasses CAT whichever feature encoder is used. This study indicates that CMF and CQA are more effective.

Table 9 reports the performance gains of different modules over “Rank@1, IoU = 0.7” metric. The results show that replacing CAT with CQA leads to larger improvements, compared to replacing BiLSTM by CMF. This observation

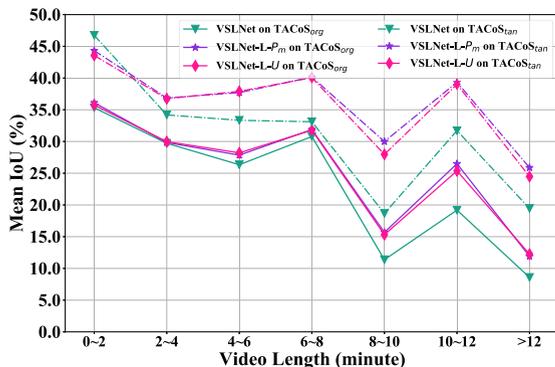


Fig. 11. Visualization of performance improvement of VSLNet-L on different video lengths compared to VSLNet on TACoS.

TABLE 7
Comparison of mIoU (%) Between VSLNet and VSLNet-L on ActivityNet Captions *w.r.t.* Different Video Lengths

Video Length	0 ~ 2 min	2 ~ 4 min	> 4 min
# Test Samples	8,193	9,274	38
VSLNet	46.21	40.60	35.24
VSLNet-L- P_m	46.59 +0.38	41.11 +0.51	38.40 +3.16
VSLNet-L- U	47.03 +0.82	41.46 +0.86	39.63 +4.39

Performance gain and loss are indicated in different colors.

TABLE 8
Comparison Between Models With Alternative Modules in VSLBase on Charades-STA

Module	Rank@1, IoU = μ			mIoU
	$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	
BiLSTM + CAT	61.18	43.04	26.42	42.83
CMF + CAT	63.49	44.87	27.07	44.01
BiLSTM + CQA	65.08	46.94	28.55	45.18
CMF + CQA	68.06	50.23	30.16	47.15

TABLE 9
Performance Gains (%) of Different Modules Over “Rank@1, IoU = 0.7” on Charades-STA

Module	CAT	CQA	Δ
BiLSTM	26.42	28.55	+2.13
CMF	27.07	30.16	+3.09
Δ	+0.65	+1.61	-

suggests that CQA plays a more important role in our model. Specifically, keeping CQA, the absolute gain is 1.61 percent by replacing the encoder module. Keeping CMF, the gain of replacing the attention module is 3.09 percent.

Fig. 12 displays the matrix of similarity score between visual and language features in the context-query attention (CQA) module ($S \in \mathbb{R}^{n \times m}$ in Section 3.3). This figure shows visual features are more relevant to the verbs and their objects in the query sentence. For example, the similarity scores between visual features and “eating” (or “sandwich”) are higher than that of other words. We believe that verbs and their objects are more likely to be used to describe video activities. Our observation is consistent with Ge *et al.* [9], where *verb-object* pairs are extracted as semantic activity concepts. In contrast, these concepts are automatically captured by the CQA module in our method.

4.5.2 The Impact of Extension Ratio in QGH

VSLNet introduces a query-guided highlighting (QGH) module on top of VSLBase to address the technical gaps between video and text. The QGH guides the VSLNet to search for the target moment within a longer highlighted region controlled by the extension ratio α . We now

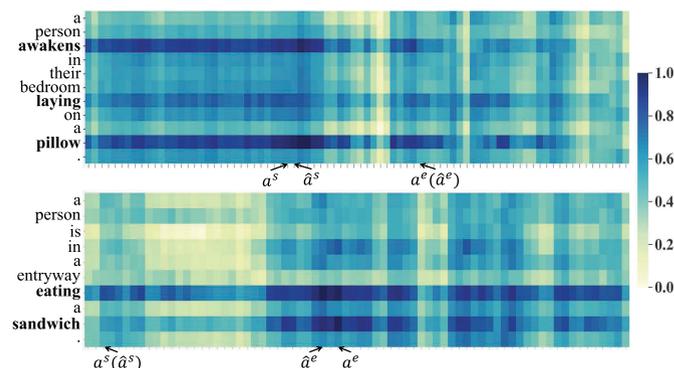


Fig. 12. Similarity scores, S , between visual and language features in the context-query attention. a^s/a^e denote the start/end boundaries of ground truth video moment, \hat{a}^s/\hat{a}^e denote the start/end boundaries of predicted target moment.

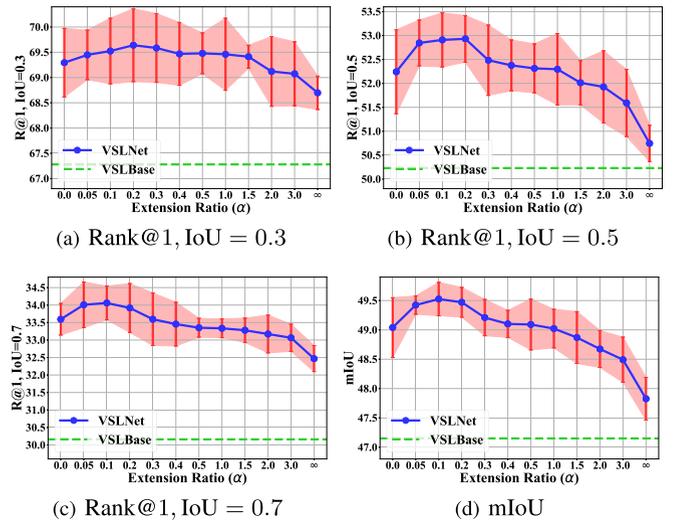


Fig. 13. Analysis of the impact of extension ratio α in Query-Guided Highlighting on Charades-STA.

investigate the impact of the extension ratio α in the QGH module on the Charades-STA dataset. We evaluated 12 different values of α from 0.0 to ∞ in experiments. 0.0 represents no answer span extension, and ∞ means that the entire video is regarded as foreground. The results for various α 's are plotted in Fig. 13. It shows that query-guided highlighting consistently contributes to performance improvements, regardless of α values, i.e., from 0 to ∞ . Along with α raises, the performance of VSLNet first increases and then gradually decreases. The optimal performance appears between $\alpha = 0.05$ and 0.2 over all metrics.

Note that, when $\alpha = \infty$, which is equivalent to no region is highlighted as a coarse region to locate target moment, VSLNet remains better than VSLBase. Shown in Fig. 5, when $\alpha = \infty$, QGH effectively becomes a straightforward concatenation of sentence representation with each of the visual features. The resultant feature remains helpful for capturing semantic correlations between vision and language. In this sense, this function can be regarded as an approximation or simulation of the traditional multimodal matching strategy [8], [24], [25].

4.5.3 Single Versus Multi-Scale Split-and-Concat

VSLNet-L further introduces a multi-scale split-and-concat strategy to address performance degradation on long videos. Here, we study the impact of the multi-scale split on the TACoS dataset with $n = 600$, against the single-scale split. We evaluate 4 different values of single-scale, i.e., $l \in \{100, 120, 150, 200\}$. The multi-scale mechanism is jointly trained with the four scales. The results are summarized in Table 10. Compared to VSLNet, split-and-concat strategy in VSLNet-L consistently contributes to performance improvements, regardless of the l value. The best single-scale l is 120 for TACoS_{org}, and 150 for TACoS_{tan}. Compared to VSLNet-L with single-scale, VSLNet-L- P_m (-U) further improves all metrics significantly. The multi-scale split-and-concat mechanism not only alleviates the issue of target moment truncation but also captures contextual information in the video; both improve the generalization ability of VSLNet-L.

TABLE 10
Results (%) of VSLNet-L on TACoS Using Different Split Scales With $n = 600$

Dataset	Model	Scales (l)	Rank@1, IoU = μ			mIoU
			$\mu = 0.3$	$\mu = 0.5$	$\mu = 0.7$	
TACoS _{org}	VSLNet	-	29.78	24.71	19.64	23.96
	VSLNet-L	100	30.18	25.81	20.77	24.46
		120	31.13	26.87	21.19	25.12
		150	30.42	26.38	20.89	24.73
		200	30.59	26.07	21.01	24.61
		$-P_m$	32.04	27.92	23.28	26.40
		$-U$	31.86	27.64	22.72	26.25
TACoS _{tan}	VSLNet	-	41.42	30.67	22.32	31.92
	VSLNet-L	100	42.24	32.69	23.67	32.41
		120	43.39	33.37	24.19	33.45
		150	44.61	33.99	24.27	33.71
		200	43.74	33.67	23.74	33.50
		$-P_m$	47.11	36.34	26.42	36.61
		$-U$	46.44	35.74	26.19	36.05

4.5.4 Qualitative Analysis

Fig. 14 shows the histograms of predicted results of VSLBase and VSLNet on test sets of the Charades-STA and ActivityNet Captions datasets. Results indicate that VSLNet beats VSLBase by having more samples in the high IoU ranges, e.g., $\text{IoU} \geq 0.7$ on the Charades-STA dataset. More predicted results of VSLNet are distributed in the high IoU

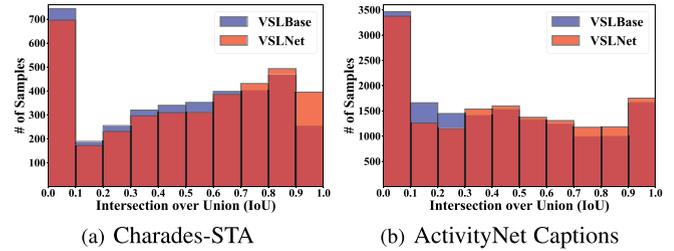


Fig. 14. Histograms of the number of predicted results on test set under different IoUs, on two datasets.

ranges for the ActivityNet Caption dataset. This result demonstrates the effectiveness of the query-guided highlighting (QGH) strategy.

We show two examples of VSLBase and VSLNet in Figs. 15a and 15b from the Charades-STA and ActivityNet Captions datasets, respectively. From the two figures, the localized moments by VSLNet are closer to the ground truth than that by VSLBase. Meanwhile, the start and end boundaries predicted by VSLNet are roughly constrained in the highlighted regions S_{h_t} , computed by QGH.

Fig. 16 depicts two predicted examples from the TACoS dataset as case studies. The localized moments by VSLNet-L are more accurate and closer to the ground truth moment than that of VSLNet. Both figures show the results of VSLNet-L are constrained in the positive clip segments, i.e., the clip segment that contains ground truth. In the



Fig. 15. Visualization of predictions by VSLBase and VSLNet. Figures on the left depict the localized results by the two models. Figures on the right show probability distributions of start/end boundaries and highlighting scores.



Fig. 16. Visualizations of two predicted examples by VSLNet and VSLNet-L on TACoS dataset.

second example, VSLNet does not capture the concept “the cutting board in the sink” and focuses on retrieving “washes” action only, which leads to an error prediction. In contrast, VSLNet-L correctly understands both “washes” and the position of “cutting board”, leading to the correct prediction.

4.5.5 Error Analysis

We further study the error patterns of predicted moment lengths on VSLBase and VSLNet, as shown in Fig. 17. The differences between moment lengths of ground truths and

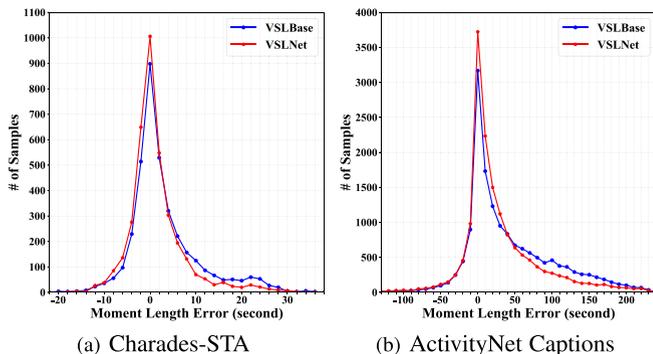
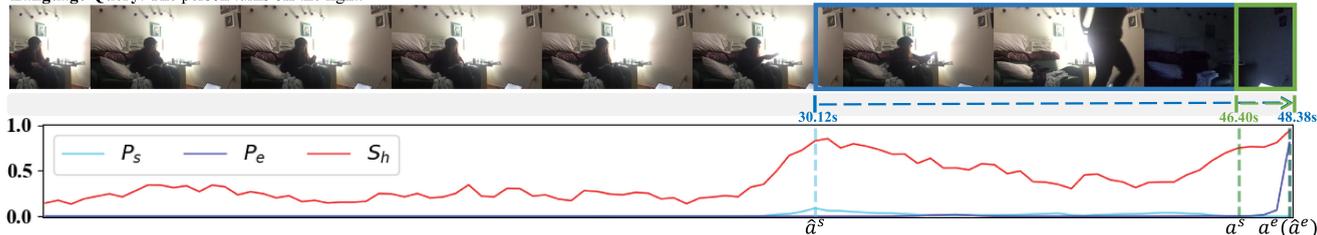


Fig. 17. Plots of moment length errors in seconds between ground truths and results predicted by VSLBase and VSLNet, respectively.

predicted results are measured. A positive length difference means the predicted moment is longer than the corresponding ground truth, while a negative means shorter. Fig. 17 shows that VSLBase tends to predict longer moments, e.g., more samples with length error larger than 4 seconds in Charades-STA or 30 seconds in ActivityNet. On the contrary, constrained by QGH, VSLNet tends to predict shorter moments, e.g., more samples with length error smaller than -4 seconds in Charades-STA or -20 seconds in ActivityNet Caption. This observation is helpful for future research on adopting the span-based QA framework for NLVL.

In addition, we also exam failure cases (with IoU predicted by VSLNet lower than 0.2) shown in Fig. 18. In the first case, as illustrated by Fig. 18a, we observe an action that a person turns towards to the lamp and places an item there. The QGH falsely predicts the action as the beginning of the moment “turns off the light”. The second failure case involves multiple actions in a query, as shown in Fig. 18b. QGH successfully highlights the correct region by capturing the temporal information of two different action descriptions in the given query. However, it assigns “pushes” with a higher confidence score than “grabs”. Thus, VSLNet only

Language Query: The person turns off the light.



Language Query: After, the man grabs the girl’s arm, then the girl pushes the man over the wall.

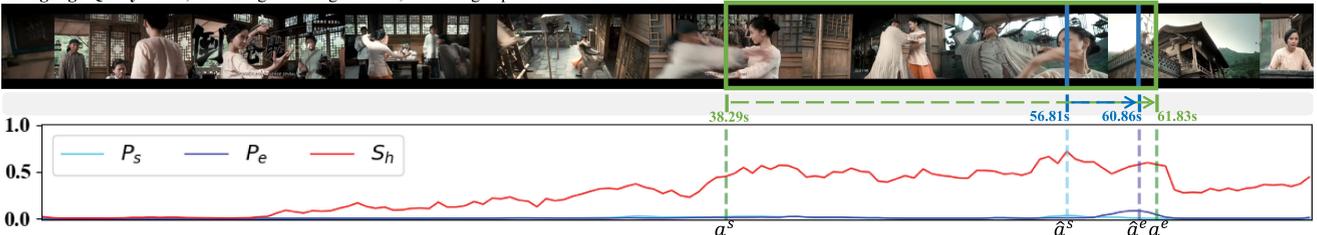


Fig. 18. Two failure examples predicted by VSLNet, a^s/a^e denote the start/end boundaries of ground truth video moment, \hat{a}^s/\hat{a}^e denote the start/end boundaries of predicted target moment.

captures the region corresponding to the “pushes” action, due to its confidence score.

5 CONCLUSION

In this paper, we revisit the NLVL task and propose to solve it with a multimodal span-based QA framework by considering a video as a text passage. We show that adopting a standard span-based QA framework, VSLBase, can achieve promising results on the NLVL task. However, there are two major differences between video and text in the standard span-based QA framework, limiting the performance of VSLBase. To address the differences, we then propose VSLNet, which introduces a simple and effective strategy named query-guided highlighting (QGH), on top of VSLBase. With QGH, VSLNet is guided to search for answers within a predicted coarse region. The effectiveness of VSLNet (and VSLBase) is demonstrated with experiments on three datasets. The results indicate that it is promising to explore the span-based QA framework to address NLVL problems. Moreover, we have observed that the existing methods including VSLNet suffer from the performance degradation issue on long videos. To address this issue, we further extend VSLNet by regarding a long video as a document with multiple paragraphs. We adopt the concept of MPQA and propose a multi-scale split-and-concat network to capture contextual information in a long video by partitioning the long video multiple times with different clip lengths. Extensive experiments demonstrate that VSLNet-L prevents performance degradation on long videos effectively and advances the state-of-the-art for NLVL on three benchmark datasets.

ACKNOWLEDGMENTS

This work was supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A1b0045 and #A18A2b0046).

REFERENCES

- [1] J. Gao, R. Ge, K. Chen, and R. Nevatia, “Motion-appearance co-memory networks for video question answering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6576–6585.
- [2] Z. Yu et al., “Activitynet-QA: A dataset for understanding complex web videos via question answering,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 9127–9134.
- [3] R. Hu, D. Fried, A. Rohrbach, D. Klein, T. Darrell, and K. Saenko, “Are you looking? Grounding to multiple modalities in vision-and-language navigation,” in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 6551–6557.
- [4] H. Le, D. Sahoo, N. Chen, and S. Hoi, “Multimodal transformer networks for end-to-end video-grounded dialogue systems,” in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5612–5623.
- [5] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. J. Corso, and J. Gao, “Unified vision-language pre-training for image captioning and VQA,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 13 041–13 049.
- [6] J. Lei, L. Yu, T. Berg, and M. Bansal, “TVQA+: Spatio-temporal grounding for video question answering,” in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 8211–8225.
- [7] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, “Localizing moments in video with temporal language,” in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 1380–1390.
- [8] M. Liu, X. Wang, L. Nie, X. He, B. Chen, and T.-S. Chua, “Attentive moment retrieval in videos,” in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 15–24.
- [9] R. Ge, J. Gao, K. Chen, and R. Nevatia, “MAC: Mining activity concepts for language-based temporal localization,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2019, pp. 245–253.
- [10] H. Xu, K. He, B. A. Plummer, L. Sigal, S. Sclaroff, and K. Saenko, “Multilevel language and vision integration for text-to-clip retrieval,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 9062–9069.
- [11] D. Zhang, X. Dai, X. Wang, Y.-F. Wang, and L. S. Davis, “MAN: Moment alignment network for natural language moment retrieval via iterative graph adjustment,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1247–1257.
- [12] J. Chen, X. Chen, L. Ma, Z. Jie, and T.-S. Chua, “Temporally grounding natural sentence in video,” in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 162–171.
- [13] Z. Zhang, Z. Lin, Z. Zhao, and Z. Xiao, “Cross-modal interaction networks for query-based moment retrieval in videos,” in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 655–664.
- [14] J. Wang, L. Ma, and W. Jiang, “Temporally grounding language queries in videos by contextual boundary-aware prediction,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12168–12175.
- [15] Y. Yuan, T. Mei, and W. Zhu, “To find where you talk: Temporal sentence localization in video with attention based location regression,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 9159–9166.
- [16] C. Lu, L. Chen, C. Tan, X. Li, and J. Xiao, “DEBUG: A dense bottom-up grounding approach for natural language video localization,” in *Proc. Conf. Empir. Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 5147–5156.
- [17] L. Chen et al., “Rethinking the bottom-up framework for query-based video localization,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 10551–10558.
- [18] W. Wang, Y. Huang, and L. Wang, “Language-driven temporal activity localization: A semantic matching reinforcement learning model,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 334–343.
- [19] D. He, X. Zhao, J. Huang, F. Li, X. Liu, and S. Wen, “Read, watch, and move: Reinforcement learning for temporally grounding natural language descriptions in videos,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8393–8400.
- [20] J. Wu, G. Li, S. Liu, and L. Lin, “Tree-structured policy based progressive reinforcement learning for temporally language grounding in video,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12386–12393.
- [21] A. W. Yu, D. Dohan, Q. Le, T. Luong, R. Zhao, and K. Chen, “Fast and accurate reading comprehension by combining self-attention and convolution,” in *Proc. Int. Conf. Learn. Representations*, 2018.
- [22] C. Clark and M. Gardner, “Simple and effective multi-paragraph reading comprehension,” in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 845–855.
- [23] H. Zhang, A. Sun, W. Jing, and J. T. Zhou, “Span-based localizing network for natural language video localization,” in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6543–6554.
- [24] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. C. Russell, “Localizing moments in video with natural language,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5804–5813.
- [25] J. Gao, C. Sun, Z. Yang, and R. Nevatia, “Tall: Temporal activity localization via language query,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5277–5285.
- [26] B. Liu, S. Yeung, E. Chou, D.-A. Huang, L. Fei-Fei, and J. C. Niebles, “Temporal modular networks for retrieving complex compositional activities in videos,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 552–568.
- [27] M. Liu, X. Wang, L. Nie, Q. Tian, B. Chen, and T.-S. Chua, “Cross-modal moment localization in videos,” in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 843–851.
- [28] S. Zhang, J. Su, and J. Luo, “Exploiting temporal relationships in video moment localization with natural language,” in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 1230–1238.
- [29] S. Chen and Y.-G. Jiang, “Semantic proposal for activity localization in videos via sentence query,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8199–8206.
- [30] D. Liu, X. Qu, X. Liu, J. Dong, P. Zhou, and Z. Xu, “Jointly cross-and self-modal graph attention network for query-based moment localization,” in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 4070–4078.
- [31] X. Qu, P. Tang, Z. Zhou, Y. Cheng, J. Dong, and P. Zhou, “Fine-grained iterative attention network for temporallanguage localization in videos,” in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 4280–4288.

- [32] Y. Yuan, L. Ma, J. Wang, W. Liu, and W. Zhu, "Semantic conditioned dynamic modulation for temporal sentence grounding in videos," in *Proc. Advances Neural Inf. Process. Syst.*, 2019, pp. 536–546.
- [33] R. Zeng, H. Xu, W. Huang, P. Chen, M. Tan, and C. Gan, "Dense regression network for video grounding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10 287–10 296.
- [34] J. Mun, M. Cho, and B. Han, "Local-global video-text interactions for temporal grounding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10 810–10 819.
- [35] J. Chen, L. Ma, X. Chen, Z. Jie, and J. Luo, "Localizing natural language in videos," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8175–8182.
- [36] C. Rodriguez, A. Marrese-Taylor, F. S. Saleh, H. Li, and S. Gould, "Proposal-free temporal moment localization of a natural-language query in video using guided attention," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2453–2462.
- [37] S. Ghosh, A. Agarwal, Z. Parekh, and A. Hauptmann, "ExCL: Extractive clip localization using natural language descriptions," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2019, pp. 1984–1990.
- [38] V. Escorcia, M. Soldan, J. Sivic, B. Ghanem, and B. Russell, "Temporal localization of moments in video collections with natural language," vol. abs/1907.12763, 2019.
- [39] D. Shao, Y. Xiong, Y. Zhao, Q. Huang, Y. Qiao, and D. Lin, "Find and focus: Retrieve and localize video events with natural language queries," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 202–218.
- [40] N. C. Mithun, S. Paul, and A. K. Roy-Chowdhury, "Weakly supervised video moment retrieval from text queries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11 592–11 601.
- [41] Z. Lin, Z. Zhao, Z. Zhang, Q. Wang, and H. Liu, "Weakly-supervised video moment retrieval via semantic completion network," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11539–11546.
- [42] S. Zhang, H. Peng, J. Fu, and J. Luo, "Learning 2D temporal adjacent networks for moment localization with natural language," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12870–12877.
- [43] S. Chen, W. Jiang, W. Liu, and Y.-G. Jiang, "Learning modality interaction for temporal sentence localization and event captioning in videos," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 333–351.
- [44] S. Wang and J. Jiang, "Machine comprehension using match-LSTM and answer pointer," in *Proc. Int. Conf. Learn. Representations*, 2017.
- [45] S. H. Wang and J. Jiang, "Learning natural language inference with LSTM," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2016, pp. 1442–1451.
- [46] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. Advances Neural Inf. Process. Syst.*, 2015, pp. 2692–2700.
- [47] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," in *Proc. Int. Conf. Learn. Representations*, 2017.
- [48] C. Xiong, V. Zhong, and R. Socher, "Dynamic coattention networks for question answering," in *Proc. Int. Conf. Learn. Representations*, 2017.
- [49] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, "Gated self-matching networks for reading comprehension and question answering," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 189–198.
- [50] H.-Y. Huang, C. Zhu, Y. Shen, and W. Chen, "Fusionnet: Fusing via fully-aware attention with application to machine comprehension," in *Proc. Int. Conf. Learn. Representations*, 2018.
- [51] Y. Wang *et al.*, "Multi-passage machine reading comprehension with cross-passage answer verification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1918–1927.
- [52] Z. Wang, P. Ng, X. Ma, R. Nallapati, and B. Xiang, "Multi-passage BERT: A globally normalized BERT model for open-domain question answering," in *Proc. Conf. Empir. Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 5878–5882.
- [53] D. Lin, J. Tang, K. Pang, S. Li, and T. Wang, "Selecting paragraphs to answer questions for multi-passage machine reading comprehension," in *Information Retrieval*. Cham, Switzerland: Springer, 2019, pp. 121–132.
- [54] L. Pang, Y. Lan, J. Guo, J. Xu, L. Su, and X. Cheng, "Has-QA: Hierarchical answer spans model for open-domain question answering," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 6875–6882.
- [55] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2016, pp. 2383–2392.
- [56] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4724–4733.
- [57] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [58] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," vol. abs/1607.06450, 2016.
- [59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [60] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Representations*, 2015.
- [61] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles, "Dense-captioning events in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 706–715.
- [62] M. Regneri, M. Rohrbach, D. Wetzels, S. Thater, B. Schiele, and M. Pinkal, "Grounding action descriptions in videos," *Trans. Assoc. Comput. Linguistics*, vol. 1, pp. 25–36, 2013.
- [63] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 510–526.
- [64] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 961–970.
- [65] M. Rohrbach, M. Regneri, M. Andriluka, S. Amin, M. Pinkal, and B. Schiele, "Script data for attribute-based recognition of composite activities," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 144–157.
- [66] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1532–1543.
- [67] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
- [68] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.