Memory-Guided Semantic Learning Network for Temporal Sentence Grounding

Daizong Liu[1,2], Xiaoye Qu[2], Xing Di[3], Yu Cheng[4], Zichuan Xu[5], Pan Zhou[1*]

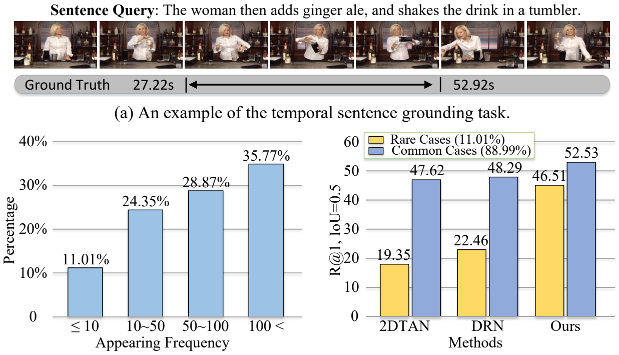
[1]The Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering,
Huazhong University of Science and Technology
[2]School of Electronic Information and Communication, Huazhong University of Science and Technology
[3]ProtagoLabs Inc [4]Microsoft Research [5]Dalian University of Technology
{dzliu, xiaoye, panzhou}@hust.edu.cn, xing.di@protagolabs.com,
yu.cheng@microsoft.com, z.xu@dlut.edu.cn

Abstract

Temporal sentence grounding (TSG) is crucial and fundamental for video understanding. Although the existing methods train well-designed deep networks with a large amount of data, we find that they can easily forget the rarely appeared cases in the training stage due to the off-balance data distribution, which influences the model generalization and leads to undesirable performance. To tackle this issue, we propose a memory-augmented network, called Memory-Guided Semantic Learning Network (MGSL-Net), that learns and memorizes the rarely appeared content in TSG tasks. Specifically, MGSL-Net consists of three main parts: a cross-modal interaction module, a memory augmentation module, and a heterogeneous attention module. We first align the given video-query pair by a cross-modal graph convolutional network, and then utilize a memory module to record the cross-modal shared semantic features in the domain-specific persistent memory. During training, the memory slots are dynamically associated with both common and rare cases, alleviating the forgetting issue. In testing, the rare cases can thus be enhanced by retrieving the stored memories, resulting in better generalization. At last, the heterogeneous attention module is utilized to integrate the enhanced multi-modal features in both video and query domains. Experimental results on three benchmarks show the superiority of our method on both effectiveness and efficiency, which substantially improves the accuracy not only on the entire dataset but also on rare cases.

Introduction

Temporal sentence grounding (TSG) is an important yet challenging task in video understanding, which has drawn increasing attention over the last few years due to its vast potential applications in video summarization (Song et al. 2015; Chu, Song, and Jaimes 2015), video captioning (Jiang et al. 2018; Chen et al. 2020b), and temporal action localization (Shou, Wang, and Chang 2016; Zhao et al. 2017), etc. As shown in Figure 1, this task aims to ground the most relevant video segment according to a given sentence query. It is substantially more challenging as it needs to not only model the complex multi-modal interactions among video and query features, but also capture complicated context information for their semantics alignment.



(a) An example of the temporal sentence grounding task.

(b) Analysis of the rare cases on ActivityNet Caption dataset.

Figure 1: (a) An illustrative example of the TSG task. (b) Data distribution on the ActivityNet Caption dataset, and the performance comparison on the corresponding rare cases.

Most existing works (Anne Hendricks et al. 2017; Ge et al. 2019; Liu et al. 2018a; Zhang et al. 2019a; Chen et al. 2018; Zhang et al. 2019b; Liu et al. 2018b; Yuan et al. 2019; Xu et al. 2019) exploit a proposal-ranking framework that first generates multiple candidate proposals and then ranks them according to their similarities with the sentence query. These methods severely rely on the quality of proposals. Instead of using complex proposals, some recent works (Rodriguez et al. 2020; Yuan, Mei, and Zhu 2019; Chen et al. 2020a; Wang, Huang, and Wang 2019; Nan et al. 2021; Mun, Cho, and Han 2020; Zeng et al. 2020) utilize a proposal-free framework that directly regresses the temporal locations of the target segment. Compared to the proposal-ranking counterparts, these works are much efficient.

Although the above two types of methods have achieved impressive results, we still can observe their performance bottlenecks on the rarely appeared video-query samples, as shown in Figure 1. Here, we select certain pairs of video and sentence as rare samples, which have at least one word (nouns, verbs, or adjectives) whose appearing frequency is less than 10. We can observe that all the existing models can achieve well performance on the common cases, but their performances all drop heavily when evaluated on rare cases. This observation conforms to that deep networks tend to forget the rare cases while learning on a dataset distributed off-balance and diverse (Toneva et al. 2019), especially in practical scenarios where the data distribution could be extremely

*Corresponding author.

imbalanced. To tackle such a challenge, we aim to better match those video-query pair having rarely appeared word-guided semantic for improving the generalization. However, it is still hard to find a balance between the common and rare samples in the dynamic training process.

To this end, in this paper, we propose to learn and memorize the discriminative and representative cross-modal shared semantic covering all samples, which is implemented by a memory-augmented network, called Memory-Guided Semantic Learning Network (MGSL-Net). Given a pair of video and query input, we first encode their contextual features individually and then align their semantic by a cross-modal graph convolutional network. After obtaining the aligned video-query feature pair, we design domain-specific persistent memories in both video and query domains to record cross-modal shared semantic representations which are the most representative. The learned memories are updated and maintained as a compact dictionary shared by all samples. During training, the memory slots in each domain are dynamically associated with both the common and rare samples across mini-batches during the whole training stage, alleviating the forgetting issue. In testing, the rare cases can thus be augmented by retrieving the stored semantic, leading to better generalization. Besides, we also develop a heterogeneous attention module to integrate the augmented multi-modal features in video and query domains by considering their contextual inter-modal interactions and video-based self-calibration.

Our main contributions are summarized as:

- We propose a memory-augmented network MGSL-Net for temporal sentence grounding, by learning and memorizing the discriminative and representative cross-modal shared semantics covering all cases. The memory is dynamically associated with both the common and rare samples seen across mini-batches during the whole training, alleviating the forgetting issue on rare samples.
- To obtain more domain-specific semantic contexts, we design the memory items in both video and query domains to be persistently read and updated. A heterogeneous attention module is further developed to integrate the enhanced multi-modal features in two domains.
- The proposed MGSL-Net achieves state-of-the-art performance on three benchmarks (ActivityNet Caption, TACoS, and Charades-STA), boosting the performance by a large margin not only on the entire dataset but also on the rarely appeared pairwise samples, with limited consumption on computation and memory.

Related Work

Temporal sentence grounding. Various algorithms (Anne Hendricks et al. 2017; Ge et al. 2019; Liu et al. 2018a; Zhang et al. 2019a; Chen et al. 2018; Qu et al. 2020; Liu, Qu, and Zhou 2021; Liu et al. 2018b, 2021a, 2022b, 2020b,a) have been proposed within the scan-and-ranking framework, which first generates multiple segment proposals, and then ranks them according to the similarity between proposals and the query to select the best matching one. Some of them (Gao et al. 2017; Anne Hendricks et al. 2017) propose to apply the sliding windows to generate proposals and subsequently integrate the query with segment representations via a matrix operation. To improve the quality of the proposals, latest works (Wang, Ma, and Jiang 2020; Zhang et al. 2019a; Yuan et al. 2019; Zhang et al. 2019b; Cao et al. 2021) directly integrate sentence information with each fine-grained video clip unit, and predict the scores of candidate segments by gradually merging the fusion feature sequence over time. Instead of generating complex proposals, recent works (Rodriguez et al. 2020; Yuan, Mei, and Zhu 2019; Chen et al. 2020a; Wang, Huang, and Wang 2019; Nan et al. 2021; Mun, Cho, and Han 2020; Zeng et al. 2020; Liu et al. 2022a) directly regress the temporal locations of the target segment. They do not rely on the segment proposals and directly select the starting and ending frames by leveraging cross-modal interactions between video and query. Specifically, they either regress the start/end timestamps based on the entire video representation (Yuan, Mei, and Zhu 2019; Mun, Cho, and Han 2020), or predict at each frame to determine whether this frame is a start or end boundary (Rodriguez et al. 2020; Chen et al. 2020a; Zeng et al. 2020). Although the above methods achieve great performances, they tend to forget the rare cases easily while learning on a dataset distributed off balance and diversely. Different from them, we focus on storing and reading the cross-modal semantic memory to enhance the multi-modal feature representations.

Memory Networks. Memory-based approaches have been discussed for solving various problems. NTM (Graves, Wayne, and Danihelka 2014) is proposed to improve the generalization ability of the network by introducing an attention-based memory module. Memory networks like (Vaswani et al. 2017; Sukhbaatar et al. 2015) have external memory where information can be further written and read. Xiong *et al.* (Xiong, Merity, and Socher 2016) further improve the memory as dynamic memory networks. Different from these unimodal memory models, we propose a cross-modal shared memory which can alternatively interact with multiple data modalities. Although other works (Ma et al. 2018; Huang and Wang 2019) also extend memory networks to multi-modal settings, most of them are episodic memory networks that are wiped during each forward process. Different from them, our memory model persistently memorizes cross-modal semantic representations in multi-modal domains with aggregation during the whole training procedure, to better deal with the unbalanced data learning.

Method

Overview

Given an untrimmed video $\mathcal{V}$ and a sentence query $\mathcal{Q}$, we represent the video as $\mathcal{V} = \{v_t\}_{t=1}^{T}$ frame-by-frame, where v_t is the t-th frame and T is the number of total frames. Similarly, the query with N words is denoted as $\mathcal{Q} = \{q_n\}_{n=1}^{N}$ word-by-word. The TSG task aims to localize the start and end timestamps (τ_s, τ_e) of a specific segment in video $\mathcal{V}$, which refers to the corresponding semantic of query $\mathcal{Q}$.

In this section, we propose a Memory-Guided Semantic Learning Network (MGSL-Net) for TSG task. The overall

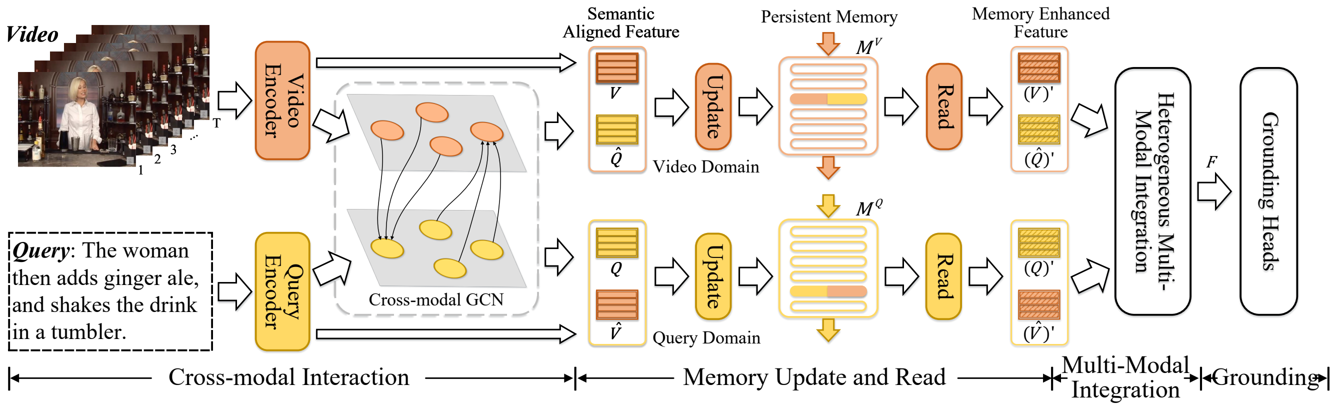


Figure 2: Overall pipeline of the proposed MGSL-Net architecture. Given a pair of video and query input, we first encode their features and exploit a cross-modal graph convolutional network (GCN) to align their semantic. Then, for the aligned features in each domain, we utilize a domain-specific persistent memory item to memorize and enhance the cross-modal shared semantic features. After that, we further develop a heterogeneous attention module to integrate multi-modal features in both domains. At last, we locate the target segment by using the regression based grounding heads.

pipeline of the proposed network, as shown in Figure 2, includes four main steps: we first encode both video and query features with contextual information, and align their features with a cross-modal graph convolutional network; then, we utilize the persistent memory items to learn and memorize the cross-modal shared semantic representations in both video and query domains; after getting the memory enhanced multi-modal features, we develop a heterogeneous attention module to consider inter- and intra-modality interactions for multi-modal feature integration; at last, the grounding heads are utilized to localize the segment.

Cross-Modal Feature Alignment

Video encoder. For video encoding, we first extract the frame-wise features by a pre-trained C3D network (Tran et al. 2015), and then employ a self-attention (Vaswani et al. 2017) module to capture the long-range dependencies among video frames. We further utilize a BiLSTM (Schuster and Paliwal 1997) to learn the sequential characteristic. We denote the extracted video features as $\boldsymbol{V} = \{\boldsymbol{v}_t\}_{t=1}^{T} \in \mathbb{R}^{T\times D}$,where D is the feature dimension.

Query encoder. For query encoding, we first generate the word-level features by using the Glove embedding (Pennington, Socher, and Manning 2014), and also employ a self-attention module and a BiLSTM layer to further encode the query features as $\boldsymbol{Q} = \{\boldsymbol{q}_n\}_{n=1}^{N} \in \mathbb{R}^{N\times D}$.

Semantic Alignment. Considering the obtained video and query representations are intrinsically heterogeneous, we propose a cross-modal graph convolutional network (Kipf and Welling 2016) to explicitly perform cross-modal alignment. Specifically, we first construct two adjacent matrices $\boldsymbol{A}_1, \boldsymbol{A}_2$ by measuring the cross-modal similarity between each frame-word pair with different directions as:

$$\boldsymbol{A}_1[t,n] = \frac{exp(a_{tn})}{\sum_{n=1}^{N} exp(a_{tn})}, \boldsymbol{A}_2[n,t] = \frac{exp(a_{nt})}{\sum_{t=1}^{T} exp(a_{nt})}, \tag{1}$$

where value $a_{tn} = a_{nt}^{\top} = \varphi_1(\boldsymbol{v}_t)\varphi_2(\boldsymbol{q}_n)^{\top}$, $\varphi_1(\cdot), \varphi_2(\cdot)$ are two modality-specific linear mappings to project one modality feature into the same latent space as the other one. $\boldsymbol{A}_1 \in \mathbb{R}^{T\times N}$, $\boldsymbol{A}_2 \in \mathbb{R}^{N\times T}$ are the normalized adjacent matrices. Therefore, we can get the aligned video representation $\widehat{\boldsymbol{V}}$ and the aligned query representation $\widehat{\boldsymbol{Q}}$ by:

$$\widehat{\boldsymbol{V}} = \boldsymbol{A}_2\boldsymbol{V}\boldsymbol{W}_V, \widehat{\boldsymbol{Q}} = \boldsymbol{A}_1\boldsymbol{Q}\boldsymbol{W}_Q, \tag{2}$$

where $\boldsymbol{W}_V, \boldsymbol{W}_Q \in \mathbb{R}^{D\times D}$ are the weight matrices. $\widehat{\boldsymbol{V}} = \{\widehat{\boldsymbol{v}}_n\}_{n=1}^{N}$ has the same size $\mathbb{R}^{N\times D}$ as query feature $\boldsymbol{Q}$, and they are semantically aligned. For the n-th row $\widehat{\boldsymbol{v}}_n$ in $\widehat{\boldsymbol{V}}$, it is an aggregated representation weighted by cross-modal similarities between the n-th word and all the frames. Therefore, $\widehat{\boldsymbol{v}}_n$ can be viewed as a visual representation of the n-th word, sharing the same semantic meaning with the word $\boldsymbol{q}_n$. Similarly, $\widehat{\boldsymbol{Q}} = \{\widehat{\boldsymbol{q}}_t\}_{t=1}^{T} \in \mathbb{R}^{T\times D}$ is semantically aligned with $\boldsymbol{V}$, and $\widehat{\boldsymbol{q}}_t$ can be viewed as a textual representation of the t-th frame, sharing the same semantic with the frame $\boldsymbol{v}_t$.

Memory Network

Based on the aligned representation pairs ($\widehat{\boldsymbol{Q}}$, $\boldsymbol{V}$) and ($\widehat{\boldsymbol{V}}$, $\boldsymbol{Q}$), as shown in Figure 2, we propose a memory network to learn and memorize the cross-modal shared semantic features in both video domain and query domain, respectively.

Memory Representation. The domain-specific cross-modal shared semantic memories in video and query domains are designed as matrices $\boldsymbol{M}^V = \{\boldsymbol{m}_{l_v}^V\}_{l_v=1}^{L_V} \in \mathbb{R}^{L_V\times D}, \boldsymbol{M}^Q = \{\boldsymbol{m}_{l_q}^Q\}_{l_q=1}^{L_Q} \in \mathbb{R}^{L_Q\times D}$, respectively. Here, L_V, L_Q are the hyper-parameters that defines the number of memory slots and D is the feature dimension. Each memory item $\boldsymbol{m}_{l_v}^V$ or $\boldsymbol{m}_{l_q}^Q$ can be updated by intra-domain features with similar semantic meanings, as well as read out to enhance previously obtained intra-domain features.

Given the aligned frame-word feature pair $(\widehat{\boldsymbol{q}}_t, \boldsymbol{v}_t)$ from $(\widehat{\boldsymbol{Q}}, \boldsymbol{V})$ in video domain, we aim to interact them with each memory item $\boldsymbol{m}_{l_v}^V$ to read and store their shared cross-modal semantic features. Before the interacting process, we first utilize several linear layers to map $\widehat{\boldsymbol{q}}_t, \boldsymbol{v}_t$ into memory read key, write key, erase value, and write value, respectively. We denote such items of $\widehat{\boldsymbol{q}}_t$ as $\boldsymbol{k}_t^{\widehat{Q},r}, \boldsymbol{k}_t^{\widehat{Q},w}, \boldsymbol{e}_t^{\widehat{Q}}, \boldsymbol{u}_t^{\widehat{Q}}$, and items of $\boldsymbol{v}_t$ as $\boldsymbol{k}_t^{V,r}, \boldsymbol{k}_t^{V,w}, \boldsymbol{e}_t^{V}, \boldsymbol{u}_t^{V}$. We also map the aligned features $\widehat{\boldsymbol{v}}_n, \boldsymbol{q}_n$ into $\boldsymbol{k}_n^{\widehat{V},r}, \boldsymbol{k}_n^{\widehat{V},w}, \boldsymbol{e}_n^{\widehat{V}}, \boldsymbol{u}_n^{\widehat{V}}$ and $\boldsymbol{k}_n^{Q,r}, \boldsymbol{k}_n^{Q,w}, \boldsymbol{e}_n^{Q}, \boldsymbol{u}_n^{Q}$. Details of how to utilize the generated items to update and read memory will be illustrated as follows.

Updating memory. Given the video domain aligned pair $(\widehat{\boldsymbol{q}}_t, \boldsymbol{v}_t)$ and query domain aligned pair $(\widehat{\boldsymbol{v}}_n, \boldsymbol{q}_n)$, we determine to write and delete which memory items in $\boldsymbol{m}_{l_v}^V$ and $\boldsymbol{m}_{l_q}^Q$. Specifically, we first calculate the memory addressing weights w according to the similarity between each input feature and corresponding domain-specific memory as:

$$w_{\boldsymbol{k}_t, \boldsymbol{m}_{l_v}^V} = \frac{exp(s(\boldsymbol{k}_t, \boldsymbol{m}_{l_v}^V))}{\sum_{l_v} exp(s(\boldsymbol{k}_t, \boldsymbol{m}_{l_v}^V))}, s(\boldsymbol{k}_t, \boldsymbol{m}_{l_v}^V) = \frac{\boldsymbol{k}_t (\boldsymbol{m}_{l_v}^V)^\top}{\parallel \boldsymbol{k}_t \parallel_2 \parallel \boldsymbol{m}_{l_v}^V \parallel_2}, \tag{3}$$

$$w_{\boldsymbol{k}_n, \boldsymbol{m}_{l_q}^Q} = \frac{exp(s(\boldsymbol{k}_n, \boldsymbol{m}_{l_q}^Q))}{\sum_{l_q} exp(s(\boldsymbol{k}_n, \boldsymbol{m}_{l_q}^Q))}, s(\boldsymbol{k}_n, \boldsymbol{m}_{l_q}^Q) = \frac{\boldsymbol{k}_n (\boldsymbol{m}_{l_q}^Q)^\top}{\parallel \boldsymbol{k}_n \parallel_2 \parallel \boldsymbol{m}_{l_q}^Q \parallel_2}, \tag{4}$$

where $\boldsymbol{k}_t \in \{\boldsymbol{k}_t^{\widehat{Q},w}, \boldsymbol{k}_t^{V,w}\}$ and $\boldsymbol{k}_n \in \{\boldsymbol{k}_n^{\widehat{V},w}, \boldsymbol{k}_n^{Q,w}\}$ are the memory write keys, $s(\cdot,\cdot)$ measures the cosine similarity. Then, we can selectively update memory items by adding new semantic features with write value while deleting old memory with erase value as:

$$(\boldsymbol{m}_{l_v}^V)' = w_{\boldsymbol{k}_t^{V,w}, \boldsymbol{m}_{l_v}^V} \boldsymbol{u}_t^V + \boldsymbol{m}_{l_v}^V \odot (1 - w_{\boldsymbol{k}_t^{V,w}, \boldsymbol{m}_{l_v}^V} \boldsymbol{e}_t^V), \tag{5}$$

$$(\boldsymbol{m}_{l_v}^V)'' = w_{\boldsymbol{k}_t^{\widehat{Q},w}, \boldsymbol{m}_{l_v}^V} \boldsymbol{u}_t^{\widehat{Q}} + (\boldsymbol{m}_{l_v}^V)' \odot (1 - w_{\boldsymbol{k}_t^{\widehat{Q},w}, \boldsymbol{m}_{l_v}^V} \boldsymbol{e}_t^{\widehat{Q}}), \tag{6}$$

$$(\boldsymbol{m}_{l_q}^Q)' = w_{\boldsymbol{k}_n^{Q,w}, \boldsymbol{m}_{l_q}^Q} \boldsymbol{u}_t^Q + \boldsymbol{m}_{l_q}^Q \odot (1 - w_{\boldsymbol{k}_n^{Q,w}, \boldsymbol{m}_{l_q}^Q} \boldsymbol{e}_t^Q), \tag{7}$$

$$(\boldsymbol{m}_{l_q}^Q)'' = w_{\boldsymbol{k}_n^{\widehat{V},w}, \boldsymbol{m}_{l_q}^Q} \boldsymbol{u}_t^{\widehat{V}} + (\boldsymbol{m}_{l_q}^Q)' \odot (1 - w_{\boldsymbol{k}_n^{\widehat{V},w}, \boldsymbol{m}_{l_q}^Q} \boldsymbol{e}_t^{\widehat{V}}), \tag{8}$$

where the erase value $\boldsymbol{e}_t \in (0, 1)$ is computed with a sigmoid function, and $\odot$ denotes element-wise multiplication. In video domain, $\boldsymbol{m}_{l_v}^V$ first updates its memory items with the extracted information from the frame and then from the word. In query domain, $\boldsymbol{m}_{l_q}^Q$ first updates its memory items with the extracted information from the word and then from the frame. In fact, the update order can be alternative and does not show a significant impact on the final performance.

Reading memory. During the memory reading, we need to read the most relevant items from domain-specific memory item $(\boldsymbol{m}_{l_v}^V)''$ and $(\boldsymbol{m}_{l_q}^Q)''$ to enhance their representations, respectively. To this end, given the $(\widehat{\boldsymbol{q}}_t, \boldsymbol{v}_t)$ and $(\widehat{\boldsymbol{v}}_n, \boldsymbol{q}_n)$, we first compute the cross-modal read weights $w_{\boldsymbol{k}_t^{\widehat{Q},r}, (\boldsymbol{m}_{l_v}^V)''}, w_{\boldsymbol{k}_t^{V,r}, (\boldsymbol{m}_{l_v}^V)''}$ and $w_{\boldsymbol{k}_n^{\widehat{V},r}, (\boldsymbol{m}_{l_q}^Q)''}, w_{\boldsymbol{k}_n^{Q,r}, (\boldsymbol{m}_{l_q}^Q)''}$ by comparing read keys with memory items like Eq. (3) and (4). Then we can read memory by regarding the obtained read keys of $(\widehat{\boldsymbol{q}}_t, \boldsymbol{v}_t)$ and $(\widehat{\boldsymbol{v}}_n, \boldsymbol{q}_n)$ as queries:

$$(\widehat{\boldsymbol{q}}_t)' = \sum_{l_v}^{L_V} w_{\boldsymbol{k}_t^{\widehat{Q},r}, (\boldsymbol{m}_{l_v}^V)''} (\boldsymbol{m}_{l_v}^V)'', (\boldsymbol{v}_t)' = \sum_{l_v}^{L_V} w_{\boldsymbol{k}_t^{V,r}, (\boldsymbol{m}_{l_v}^V)''} (\boldsymbol{m}_{l_v}^V)'', \tag{9}$$

$$(\widehat{\boldsymbol{v}}_n)' = \sum_{l_q}^{L_Q} w_{\boldsymbol{k}_n^{\widehat{V},r}, (\boldsymbol{m}_{l_q}^Q)''} (\boldsymbol{m}_{l_q}^Q)'', (\boldsymbol{q}_n)' = \sum_{l_q}^{L_Q} w_{\boldsymbol{k}_n^{Q,r}, (\boldsymbol{m}_{l_q}^Q)''} (\boldsymbol{m}_{l_q}^Q)'', \tag{10}$$

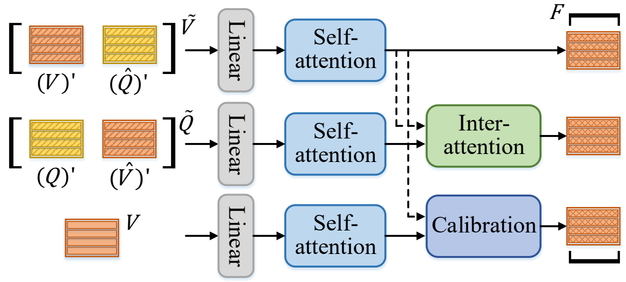


Figure 3: Illustration of Heterogeneous Attention module. The units of self-attention, inter-attention and calibration are implemented by dot product attention.

where $(\widehat{\boldsymbol{q}}_t)', (\boldsymbol{v}_t)'$ and $(\widehat{\boldsymbol{v}}_n)', (\boldsymbol{q}_n)'$ are the read vectors which can be regarded as memory-enhanced representations of video and query domains, respectively.

Heterogeneous Multi-Modal Integration

After obtaining memory enhanced representations from both video and query domains, we generate new video representation $\widetilde{\boldsymbol{V}} = \{\widetilde{\boldsymbol{v}}_t\}_{t=1}^T \in \mathbb{R}^{T \times 2D}$ and new query representation $\widetilde{\boldsymbol{Q}} = \{\widetilde{\boldsymbol{q}}_n\}_{n=1}^N \in \mathbb{R}^{N \times 2D}$ by concatenation operation, where $\widetilde{\boldsymbol{v}}_t = [(\boldsymbol{v}_t)'; (\widehat{\boldsymbol{q}}_t)']$ and $\widetilde{\boldsymbol{q}}_n = [(\boldsymbol{q}_n)'; (\widehat{\boldsymbol{v}}_n)']$. To further integrate these two representations, we develop a heterogeneous attention module to consider their inter- and intra-modality interactions. In particular, we additionally take the original video feature $\boldsymbol{V}$ as global context to calibrate the learned memory contents in $\widetilde{\boldsymbol{V}}$. As shown in Figure 3, the proposed heterogeneous attention mechanism first utilizes three linear layers to map the three representations in to the same latent space, and then exploits a self-attention unit to capture the semantic-aware intra-modality relations between the enhanced frame-frame pairs and word-word pairs. After that, the inter-modality relationship is captured by interacting the features of frame-word pair. To further calibrate the memory-wise contents, we take $\boldsymbol{V}$ as the global signals to supervise the enhanced video feature $\widetilde{\boldsymbol{V}}$. These three attentional units are combined in a modular way in defining the heterogeneous attention mechanism, and all the units are based on the dot product attention. Finally, the integrated feature $\boldsymbol{F} = \{\boldsymbol{f}_t\}_{t=1}^T$ is obtained by concatenating all those output features.

Grounding Heads

Taking the fine-grained feature $\boldsymbol{F} = \{\boldsymbol{f}_t\}_{t=1}^T$ as input, we process frame-wise feature $\boldsymbol{f}_t$ by the grounding module, which consists of three components: boundary regression head, confidence scoring head and IoU regression head. Since TSG task aims to localize a specific segment, the boundary regression head is designed to predict the temporal bounding box at each frame. To select the box that matches the query best, we propose the confidence scoring head to predict scores indicating whether the content in each bbox matches the query semantically. A IoU regression head is also utilized to predict score for directly estimating the IoU between each bbox and the ground truth segment.

Boundary regression head. We implement this head as two 1D convolution layers with two output channels, and we only assign regression targets for positive frames. If location t falls inside the ground truth (τ_s, τ_e), the regression targets are $d_t = (d_{t,s}, d_{t,e})$, where $d_{t,s} = t - \tau_s, d_{t,e} = \tau_e - t$. For the predicted $\hat{d}_t$ and ground truth d_t, we define $\mathcal{L}_b$ as:

$$\mathcal{L}_b = \frac{1}{T_p} \sum_{t=1}^{T} \mathbb{1}_t(\mathcal{L}_1(d_t, \hat{d}_t) - ln \frac{min(d_{t,e}, \hat{d}_{t,e}) - max(d_{t,s}, \hat{d}_{t,s})}{max(d_{t,e}, \hat{d}_{t,e}) - min(d_{t,s}, \hat{d}_{t,s})}), \tag{11}$$

where $\mathcal{L}_1$ is a smooth l_1 loss, the second item is a IoU loss. $\mathbb{1}_t$ is the indicator function, being 1 if frame t is positive and 0 otherwise. T_p is the number of positive frames.

Confidence scoring head. This head is implemented as two 1D convolution layers with one output channel. For each frame t, if it falls in the ground truth, we think its generated bbox matches the query semantically and denote its label as $c_t = 1$. If not, we denote it as $c_t = 0$. We utilize a a binary cross entropy loss for confidence evaluation as:

$$\mathcal{L}_c = \frac{1}{T_p} \sum_{t=1}^{T} \mathcal{L}_{bce}(c_t, \hat{c}_t). \tag{12}$$

IoU regression head. We train a three-layer 1D convolution to estimate the IoU between the generated bbox at each frame and the corresponding ground truth. Denoting the ground truth IoU as i_t and predicted one as $\hat{i}_t$, we have:

$$\mathcal{L}_i = \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}_1(i_t, \hat{i}_t). \tag{13}$$

Thus, the final loss is a multi-task loss combing the above three loss functions as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_b + \lambda_2 \mathcal{L}_c + \lambda_3 \mathcal{L}_i, \tag{14}$$

where λ_1, λ_2 and λ_3 are the hyper-paremeters to balance the training weights on different losses.

Experiments

Datasets and Evaluation

ActivityNet Caption. ActivityNet Caption (Krishna et al. 2017) contains 20000 untrimmed videos with 100000 descriptions from YouTube. The videos are 2 minutes on average, and the annotated video clips have much larger variation, ranging from several seconds to over 3 minutes. Following public split, we use 37,417, 17,505, and 17,031 sentence-video pairs for training, validation, and testing respectively.

TACoS. TACoS (Regneri et al. 2013) is widely used on TSG task and contain 127 videos. The videos from TACoS are collected from cooking scenarios, thus lacking the diversity. They are around 7 minutes on average. We use the same split as (Gao et al. 2017), which includes 10146, 4589, 4083 query-segment pairs for training, validation and testing.

Charades-STA. Charades-STA is built on the Charades dataset (Sigurdsson et al. 2016), which focuses on indoor activities. In total, there are 12408 and 3720 moment-query pairs in the training and testing sets respectively.

Evaluation Metric. Following previous works (Gao et al. 2017; Zeng et al. 2020; Zhang et al. 2020b), we adopt "R@n, IoU=m" as our evaluation metrics. It is defined as the percentage of at least one of top-n selected moments having IoU larger than m, which is the higher the better.

Implementation Details

We utilize the 112 × 112 pixels shape of every frame of videos as input, and apply C3D (Tran et al. 2015) to encode the videos on ActivityNet Caption, TACoS, and I3D (Carreira and Zisserman 2017) on Charades-STA. We set the length of video feature sequences to 200 for ActivityNet Caption and TACoS datasets, 64 for Charades-STA dataset. As for sentence encoding, we set the length of word feature sequences to 20, and utilize Glove embedding (Pennington, Socher, and Manning 2014) to embed each word to 300 dimension features. The hidden state dimension of BiLSTM networks is set to 512. The number of memory items (L_V, L_Q) are set to (1024,1024), (512,512), (512,512) for three datasets, respectively. We empirically find that further increasing the memory number results in a convergence of the performance. The balanced weights of $\mathcal{L}$ are $\lambda_1 = \lambda_2 = \lambda_3 = 1.0$. During the training, we use an Adam optimizer with the leaning rate of 0.0001. The model is trained for 50 epochs to guarantee its convergence with a batch size of 128. All the experiments are implemented on a single NVIDIA TITAN XP GPU.

Comparisons with the State-of-the-Arts

Compared Methods. We compare the proposed MGSL-Net with state-of-the-art TSG methods on three datasets. These methods are grouped into three categories by the viewpoints of proposal-based and proposal-free approach: 1) proposal-based methods: TGN (Chen et al. 2018), CTRL (Gao et al. 2017), ACRN (Liu et al. 2018a), QSPN (Xu et al. 2019), CBP (Wang, Ma, and Jiang 2020), SCDM (Yuan et al. 2019), CMIN (Zhang et al. 2019b), 2DTAN (Zhang et al. 2020b), and CBLN (Liu et al. 2021b). 2) proposal-free methods: GDP (Chen et al. 2020a), LGI (Mun, Cho, and Han 2020), VSLNet (Zhang et al. 2020a), DRN (Zeng et al. 2020). 3) others: BPNet (Xiao et al. 2021). Note that all the above methods directly utilize deep networks to learn cross-modal retrieval without considering the rarely appeared video-query samples.

Comparison on ActivityNet Caption. Table 1 summarizes the results on ActivityNet Caption. It shows that our MGSL-Net outperforms all the baselines in all metrics. Specifically, we observe that MGSL-Net works well in even stricter metrics, e.g., it achieved a significant 3.82% and 3.30% absolute improvement in R@1, and R@5, IoU=0.7 compared to the previous state-of-the-art method CBLN, which demonstrates the superiority of our model. It is mainly because our memory can store useful cross-modal shared semantic representations, and thus better associate those rarely appeared video and query in the standard test sets.

Comparison on TACoS. From Table 1, we can also find that our MGSL-Net achieves the best performance on TACoS dataset. Note that our model shows much larger improvements on the TACoS dataset than the ActivityNet Caption

Method	ActivityNet Captions				TACoS				Charades-STA			
	R@1, IoU=0.5	R@1, IoU=0.7	R@5, IoU=0.5	R@5, IoU=0.7	R@1, IoU=0.3	R@1, IoU=0.5	R@5, IoU=0.3	R@5, IoU=0.5	R@1, IoU=0.5	R@1, IoU=0.7	R@5, IoU=0.5	R@5, IoU=0.7
TGN	28.47	-	43.33	-	21.77	18.90	39.06	31.02	-	-	-	-
CTRL	29.01	10.34	59.17	37.54	18.32	13.30	36.69	25.42	23.63	8.89	58.92	29.57
ACRN	31.67	11.25	60.34	38.57	19.52	14.62	34.97	24.88	20.26	7.64	71.99	27.79
QSPN	33.26	13.43	62.39	40.78	20.15	15.23	36.72	25.30	35.60	15.80	79.40	45.40
CBP	35.76	17.80	65.89	46.20	27.31	24.79	43.64	37.40	36.80	18.87	70.94	50.19
SCDM	36.75	19.86	64.99	41.53	26.11	21.17	40.16	32.18	54.44	33.43	74.43	58.08
GDP	39.27	-	-	-	24.14	-	-	-	39.47	18.49	-	-
LGI	41.51	23.07	-	-	-	-	-	-	59.46	35.48	-	-
BPNet	42.07	24.69	-	-	25.96	20.96	-	-	50.75	31.64	-	-
VSLNet	43.22	26.16	-	-	29.61	24.27	-	-	54.19	35.22	-	-
CMIN	43.40	23.88	67.95	50.73	24.64	18.05	38.46	27.02	-	-	-	-
2DTAN	44.51	26.54	77.13	61.96	37.29	25.32	57.81	45.04	39.81	23.25	79.33	51.15
DRN	45.45	24.36	77.97	50.30	-	23.17	-	33.36	53.09	31.75	89.06	60.05
CBLN	48.12	27.60	79.32	63.41	38.98	27.65	59.96	46.24	61.13	38.22	90.33	61.69
MGSL-Net	**51.87**	**31.42**	**82.60**	**66.71**	**42.54**	**32.27**	**63.39**	**50.13**	**63.98**	**41.03**	**93.21**	**63.85**

Table 1: Performance compared with the state-of-the-arts on ActivityNet Caption, TACoS, and Charades-STA datasets.

Method	Run-Time	Model Size	R@1, IoU=0.5
ACRN	4.31s	128M	14.62
CTRL	2.23s	**22M**	13.30
TGN	0.92s	166M	18.90
2DTAN	0.57s	232M	25.32
DRN	0.15s	214M	23.17
MGSL-Net	**0.10s**	203M	**32.27**

Table 2: Efficiency comparison run on TACoS dataset.

dataset. It mainly results from that the fewer training data with low diversity of TACoS dataset cannot guarantee the previous models can well capture the relations among small object. But our model can better exploit the auxiliary resources for better learning.

Comparison on Charades-STA. As shown in Table 1, our MGSL-Net achieves new state-of-the-art performance over all metrics on Charades-STA. Since there exists less rarely appeared samples in this dataset, it has less performance improvements than the other two datasets.

Efficiency Comparison

We evaluate the efficiency of our MGSL-Net, by fairly comparing its running time and parameter size with existing methods on a single Nvidia TITAN XP GPU on TACoS dataset. As shown in Table 2, it can be observed that we achieves much faster processing speeds and relatively less learnable parameters. This attributes to: 1) The proposal generation procedure and proposal matching procedure of proposal-based methods (ACRN, CTRL, TGN, 2DTAN) are quite time-consuming. 2) Regression-based method DRN utilizes much convolutional layers to achieve multi-level feature fusion for cross-modal interaction, which is also cost time. 3) Our MGSL is free from complex and time-consuming operations, showing superiority in both effectiveness and efficiency.

Analysis on the Rare Cases

Analyzing with the rarely appeared video-query samples is not easy since such pair-wise data is not easy to define.

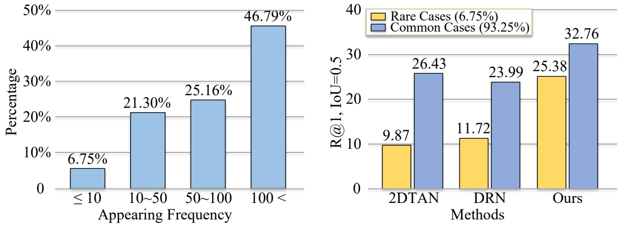


Figure 4: Data distribution on the TACoS dataset, and the performance comparison on its rare cases.

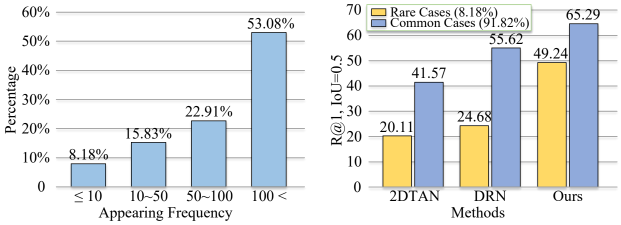


Figure 5: Data distribution on the Charades-STA dataset, and the performance comparison on its rare cases.

Therefore, we first analyze the data distribution of each dataset as shown in Figure 1, 4 and 5, and then we select certain pairs of video and sentence as rare samples, which have at least one word (nouns, verbs or adjectives) whose appearing frequency is less than 10. The other remained samples are treated as common samples. In these three figures, we show the performance of different methods on the rare cases and common cases. Our proposed method is more effective to handle the rare cases, which brings much more improvements than other methods. We also give the qualitative examples of the grounding results as shown in Figure 6, where our method can learn and memorize the semantics of the rare cases and can ground the segment more accurately.

Domain-specific memory networks	Heterogeneous attention module	R@1, IoU=0.5	R@1, IoU=0.7
		43.65	22.48
✓		49.24	28.01
	✓	46.19	25.73
✓	✓	**51.87**	**31.42**

Table 3: Main ablation study on ActivityNet Caption dataset.

Video domain			Query domain			R@1,	R@1,
$\boldsymbol{V}$	$\widehat{\boldsymbol{Q}}$	shared	$\boldsymbol{Q}$	$\widehat{\boldsymbol{V}}$	shared	IoU=0.5	IoU=0.7
						46.19	25.73
✓						47.64	26.82
✓	✓					48.88	27.90
✓	✓	✓				49.97	28.21
✓	✓	✓	✓			50.56	29.60
✓	✓	✓	✓	✓		51.20	30.85
✓	✓	✓	✓	✓	✓	**51.87**	**31.42**

Table 4: Performance comparisons with the memory network in different settings on ActivityNet Caption dataset.

Self-attention	Inter-attention	Calibration	R@1, IoU=0.5	R@1, IoU=0.7
	✓		49.24	28.01
	✓	✓	50.33	28.98
✓	✓		51.27	30.56
✓	✓	✓	**51.87**	**31.42**

Table 5: Performance comparisons with the heterogeneous attention module in different settings on ActivityNet Caption dataset.

Ablation Study

Main ablation. As shown in Table 3, we first study the influence of each main component in our proposed MGSL-Net. We set the MGSL-Net model without both domain-specific memory networks and heterogeneous multi-modal integration module as the baseline. The table shows that both memory network and heterogeneous attention make great contributions for the final performance, where the memory network brings the largest improvement of 5-6 absolute values.

Investigation on memory network. We investigate the performance comparison with the memory network with different settings as shown in Table 4, where "shared" means utilizing a shared memory slots to learn the semantics of two inputs in each domain. From the table, we can find several points: 1) The two aligned semantics in pairwise data $(\boldsymbol{V}, \widehat{\boldsymbol{Q}})$ or $(\boldsymbol{Q}, \widehat{\boldsymbol{V}})$ are all important for memory learning. 2) In each domain, the shared memory performs better than utilizing two separated memories for reading and updating the pairwise data. 3) The memory-based contexts in both video and query domains are all helpful for grounding.

To further investigate what the shared memory actually learns, we reduce the dimensionality of memory slots with PCA, and show their two-dimensional representations (grey nodes) in Figure 7. We can see that all the nodes distribute in a divergent shape, in which the top nodes are more compact while the bottom ones are more scattered. To figure out the semantic meanings of these memory slots, we take several representative nodes (with arrows) as queries to retrieve video-query pair. We find the rarely appeared content is indeed captured and represented as the scattered nodes while more commonly appeared content is captured and represented as the compact ones.

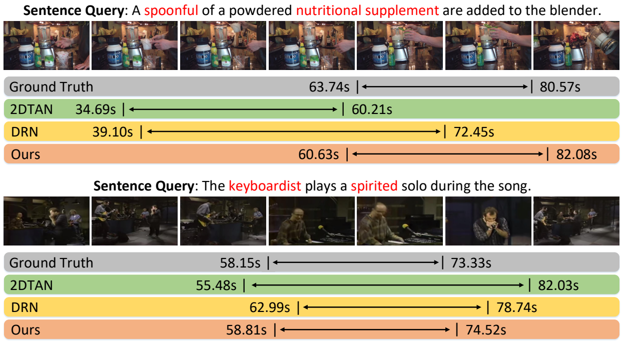


Figure 6: The qualitative results of our proposed method. Rarely appeared words are marked as red.

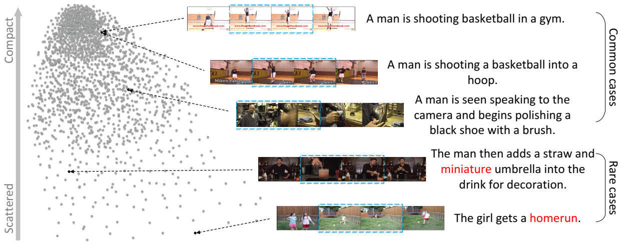


Figure 7: Two-dimensional visualization of learned memory items. The blue rectangle denotes the target segment. Rarely appeared words are marked as red.

Investigation on heterogeneous attention. We also conduct the ablation studies on the heterogeneous attention module in Table 5, where we set the inter-attention branch as the baseline. The self-attention brings largest improvement, since it not only captures the intra-relations among the elements in each modality but also provides the enhanced video features in video domain. The calibration module also makes contribution to the final performance.

Conclusion

In this paper, we have proposed the Memory-Guided Semantic Learning (MGSL) to handle the rarely appeared pairwise samples in temporal sentence grounding task. The main contributions of this work are: 1) we propose a cross-modal graph convolutional network to align the semantic between video and query, 2) we develop two domain-specific persistent memory items to learn and memorize the cross-modal shared semantic representations, and 3) we devise a heterogeneous attention module to integrate the enhanced multi-modal features in both video and query domains. Experimental results shows the superiority of our method on both effectiveness and efficiency.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant (No.61972448, No.62172068, No.61802048).

References

Anne Hendricks, L.; Wang, O.; Shechtman, E.; Sivic, J.; Darrell, T.; and Russell, B. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Cao, M.; Chen, L.; Shou, M. Z.; Zhang, C.; and Zou, Y. 2021. On Pursuit of Designing Multi-modal Transformer for Video Grounding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9810–9823.

Carreira, J.; and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6299–6308.

Chen, J.; Chen, X.; Ma, L.; Jie, Z.; and Chua, T.-S. 2018. Temporally grounding natural sentence in video. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 162–171.

Chen, L.; Lu, C.; Tang, S.; Xiao, J.; Zhang, D.; Tan, C.; and Li, X. 2020a. Rethinking the Bottom-Up Framework for Query-based Video Localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Chen, S.; Jiang, W.; Liu, W.; and Jiang, Y.-G. 2020b. Learning Modality Interaction for Temporal Sentence Localization and Event Captioning in Videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Chu, W.-S.; Song, Y.; and Jaimes, A. 2015. Video co-summarization: Video summarization by visual co-occurrence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 5267–5275.

Ge, R.; Gao, J.; Chen, K.; and Nevatia, R. 2019. Mac: Mining activity concepts for language-based temporal localization. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 245–253.

Graves, A.; Wayne, G.; and Danihelka, I. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401* .

Huang, Y.; and Wang, L. 2019. Acmm: Aligned cross-modal memory for few-shot image and sentence matching. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Jiang, W.; Ma, L.; Jiang, Y.-G.; Liu, W.; and Zhang, T. 2018. Recurrent fusion network for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 499–515.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* .

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Carlos Niebles, J. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 706–715.

Liu, D.; Qu, X.; Dong, J.; and Zhou, P. 2020a. Reasoning Step-by-Step: Temporal Sentence Localization in Videos via Deep Rectification-Modulation Network. In *Proceedings of the 28th International Conference on Computational Linguistics*, 1841–1851.

Liu, D.; Qu, X.; Dong, J.; and Zhou, P. 2021a. Adaptive Proposal Generation Network for Temporal Sentence Localization in Videos. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9292–9301.

Liu, D.; Qu, X.; Dong, J.; Zhou, P.; Cheng, Y.; Wei, W.; Xu, Z.; and Xie, Y. 2021b. Context-aware Biaffine Localizing Network for Temporal Sentence Grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Liu, D.; Qu, X.; Liu, X.-Y.; Dong, J.; Zhou, P.; and Xu, Z. 2020b. Jointly Cross-and Self-Modal Graph Attention Network for Query-Based Moment Localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, 4070–4078.

Liu, D.; Qu, X.; Wang, Y.; Di, X.; Zou, K.; Cheng, Y.; Xu, Z.; and Zhou, P. 2022a. Unsupervised Temporal Video Grounding with Deep Semantic Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Liu, D.; Qu, X.; and Zhou, P. 2021. Progressively Guide to Attend: An Iterative Alignment Framework for Temporal Sentence Grounding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9302–9311.

Liu, D.; Qu, X.; Zhou, P.; and Liu, Y. 2022b. Exploring Motion and Appearance Information for Temporal Sentence Grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Liu, M.; Wang, X.; Nie, L.; He, X.; Chen, B.; and Chua, T.-S. 2018a. Attentive moment retrieval in videos. In *Proceedings of the 41nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 15–24.

Liu, M.; Wang, X.; Nie, L.; Tian, Q.; Chen, B.; and Chua, T.-S. 2018b. Cross-modal moment localization in videos. In *Proceedings of the 26th ACM international conference on Multimedia*, 843–851.

Ma, C.; Shen, C.; Dick, A.; Wu, Q.; Wang, P.; van den Hengel, A.; and Reid, I. 2018. Visual question answering with memory-augmented networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Mun, J.; Cho, M.; and Han, B. 2020. Local-Global Video-Text Interactions for Temporal Grounding. In *Proceedings* *of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 10810–10819.

Nan, G.; Qiao, R.; Xiao, Y.; Liu, J.; Leng, S.; Zhang, H.; and Lu, W. 2021. Interventional Video Grounding with Dual Contrastive Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Qu, X.; Tang, P.; Zou, Z.; Cheng, Y.; Dong, J.; Zhou, P.; and Xu, Z. 2020. Fine-grained iterative attention network for temporal language localization in videos. In *Proceedings of the 28th ACM International Conference on Multimedia*, 4280–4288.

Regneri, M.; Rohrbach, M.; Wetzel, D.; Thater, S.; Schiele, B.; and Pinkal, M. 2013. Grounding action descriptions in videos. *Transactions of the Association for Computational Linguistics* 1: 25–36.

Rodriguez, C.; Marrese-Taylor, E.; Saleh, F. S.; Li, H.; and Gould, S. 2020. Proposal-free temporal moment localization of a natural-language query in video using guided attention. In *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2464–2473.

Schuster, M.; and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45: 2673–2681.

Shou, Z.; Wang, D.; and Chang, S.-F. 2016. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1049–1058.

Sigurdsson, G. A.; Varol, G.; Wang, X.; Farhadi, A.; Laptev, I.; and Gupta, A. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision (ECCV)*, 510–526.

Song, Y.; Vallmitjana, J.; Stent, A.; and Jaimes, A. 2015. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5179–5187.

Sukhbaatar, S.; Weston, J.; Fergus, R.; et al. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems (NIPS)*.

Toneva, M.; Sordoni, A.; Combes, R. T. d.; Trischler, A.; Bengio, Y.; and Gordon, G. J. 2019. An empirical study of example forgetting during deep neural network learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 4489–4497.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, 5998–6008.

Wang, J.; Ma, L.; and Jiang, W. 2020. Temporally Grounding Language Queries in Videos by Contextual Boundary-aware Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Wang, W.; Huang, Y.; and Wang, L. 2019. Language-driven temporal activity localization: A semantic matching reinforcement learning model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 334–343.

Xiao, S.; Chen, L.; Zhang, S.; Ji, W.; Shao, J.; Ye, L.; and Xiao, J. 2021. Boundary Proposal Network for Two-Stage Natural Language Video Localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Xiong, C.; Merity, S.; and Socher, R. 2016. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning (ICML)*.

Xu, H.; He, K.; Plummer, B. A.; Sigal, L.; Sclaroff, S.; and Saenko, K. 2019. Multilevel language and vision integration for text-to-clip retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9062–9069.

Yuan, Y.; Ma, L.; Wang, J.; Liu, W.; and Zhu, W. 2019. Semantic Conditioned Dynamic Modulation for Temporal Sentence Grounding in Videos. In *Advances in Neural Information Processing Systems (NIPS)*, 534–544.

Yuan, Y.; Mei, T.; and Zhu, W. 2019. To find where you talk: Temporal sentence localization in video with attention based location regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9159–9166.

Zeng, R.; Xu, H.; Huang, W.; Chen, P.; Tan, M.; and Gan, C. 2020. Dense regression network for video grounding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 10287–10296.

Zhang, D.; Dai, X.; Wang, X.; Wang, Y.-F.; and Davis, L. S. 2019a. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1247–1257.

Zhang, H.; Sun, A.; Jing, W.; and Zhou, J. T. 2020a. Span-based Localizing Network for Natural Language Video Localization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6543–6554.

Zhang, S.; Peng, H.; Fu, J.; and Luo, J. 2020b. Learning 2D Temporal Adjacent Networks for Moment Localization with Natural Language. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhang, Z.; Lin, Z.; Zhao, Z.; and Xiao, Z. 2019b. Cross-modal interaction networks for query-based moment retrieval in videos. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 655–664.

Zhao, Y.; Xiong, Y.; Wang, L.; Wu, Z.; Tang, X.; and Lin, D. 2017. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2914–2923.