IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 6, 2021, accepted July 30, 2021, date of publication August 23, 2021, date of current version August 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3106698

Multi-Perspective Attention Network for Fast Temporal Moment Localization

JUNGKYOO SHIN, (Member, IEEE), AND JINYOUNG MOON
Department of ICT, University of Science and Technology, Daejeon 34113, Republic of Korea
Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea

Corresponding author: Jinyoung Moon (jymoon@etri.re.kr)

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant through the Ministry of Science and ICT (MSIT), Government of Korea (Development of Previsional Intelligence Based on Long-Term Visual Memory Network and Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis) under Grant 2020-0-00004 and Grant 2014-3-00123.

ABSTRACT Temporal moment localization (TML) aims to retrieve the temporal interval for a moment semantically relevant to a sentence query. This is challenging because it requires understanding a video, a sentence, and the relationship between them. Existing TML methods have shown impressive performances by modeling interactions between videos and sentences using fine-grained techniques. However, these fine-grained techniques require a high computational overhead, making them impractical. This work proposes an effective and efficient multi-perspective attention network for temporal moment localization. Inspired by the way humans understand an image from multiple perspectives and different contexts, we devise a novel multi-perspective attention mechanism consisting of perspective attention and multi-perspective modal interactions. Specifically, a perspective attention layer based on multi-head attention takes two memory sequences, one as the base and the other as the reference memory, as inputs. Perspective attention assesses the two different memories, models the relationship, and encourages the base memory to focus on features related to the reference memory, providing an understanding of the base memory from the perspective of the reference memory. Furthermore, multi-perspective modal interactions model the complex relationship between a video and sentence query, and obtain the modal-interacted memory, consisting of a visual feature that selectively learned query-related information. Similar to the heavyweight fine-grained TML methods, the proposed network obtains the accurate complex relationship while being lightweight like coarse-grained TML methods. We also adopt a fast action recognition network to efficiently extract visual features, which reduce the computational overhead. Through experiments on three TML benchmark datasets, we demonstrate the effectiveness and efficiency of the proposed network.

INDEX TERMS Cross-modal interaction, fast temporal moment localization, temporal moment localization, and temporal sentence grounding.

I. INTRODUCTION

As a large number of videos are being created and consumed every day, there is a growing need for an efficient method to search for content. To achieve this, deep learning-based methods that understand untrimmed videos and can localize a specific temporal interval have been proposed. Early works focused on temporal action localization, which aims to retrieve all temporal intervals with the start and end times of action instances belonging to predefined action classes in a video. Recently, temporal moment localization (TML) with a sentence query, whose goal is to retrieve the temporal interval consisting of the start and end times for a moment described by a sentence query, was proposed [1], [2]. In contrast to predefined action labels with a single keyword, queries in a natural language can describe a wide range of semantic information within a video spatially and temporally. This enables TML methods to understand and localize moments that involve spatiotemporally complex activities, including sub-actions related to human–human and human–object interactions.

TML is more challenging than temporal action localization because of the complex relations between the whole and multiple parts of an input video and sentence query for a

The associate editor coordinating the review of this manuscript and approving it for publication was Khoa Luu.

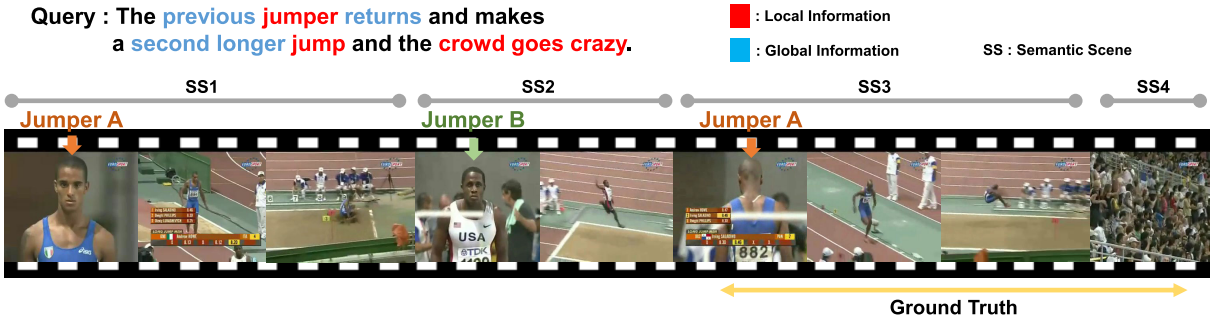

FIGURE 1. TML aims to find the best corresponding moment of the given query in the video. The above figure shows an example of the TML task. Natural language query requires the understanding of the local–global relationship between complex objects within a video. Words colored in cyan signify global information, while those in red signify local information.

TML task. Understanding the complex relations spatially and temporally requires local as well as global information. Fig 1 shows the local and global information necessary to obtain precise TML results for a given input video and query. In Fig. 1, understanding local information from the query Q colored in red, which is visual semantic information from a single scene, requires a TML method to recognize *the jumper* performing a *jumping* action in a scene and to distinguish *the jumper* from other objects within the same scene, such as *the referees* or *the crowd*. To identify *the jumper*, the network must focus on visual semantic information relevant to the jumper and ignore the information irrelevant to the jumper from unrelated objects within a scene simultaneously. In addition, understanding global information from the query Q colored in cyan, such as *previous*, *returns*, and *second longer*, require the TML method to gather the integrated contextual information from multiple scenes and relate them. To understand and localize *the previous* jumper, the TML method should discriminate two different jumpers, Jumper A and Jumper B, from three different semantic scenes, understand the temporal order of the two jumpers in the three scenes (*i.e. A in SS1, B in SS2, and again A in SS3*), and finally align the temporal interval relevant to A occurring second in the third scene and irrelevant to A occurring first in the first scene. Therefore, understanding local–global information simultaneously is an important concern in TML tasks.

The first limitations of the recently proposed TML models [4]–[7] are that they are heavy and slow, which is inadequate for practical scenarios. Early TML methods [1]–[3] integrate visual and query features in a coarse-grained manner, which interact the entire sentence and entire video at once, and have shown relatively low performance. They relate the entire input at once and neglect spatiotemporal information, thus failing to obtain local–global information. To address this issue, recent methods have modeled the relationship in a fine-grained manner, which divides the interaction subprocess into multiple steps for deeper interactions. For instance, some fine-grained methods attend a sentence to every time-step of a video [4], [5] or pool the visual feature into the size of predefined labels and interact each pooled feature with the query, respectively [6]. Mun *et al.* [7] proposed a method that divides query features into several parts and interacts each part with the video, respectively. These methods have shown better performance as they enable a deeper understanding of complex relationships but require a large computational overhead.

To this end, we propose a multi-perspective attention network (MPAN) for TML that interacts with two modal memories in two different coarse-grained interaction layers to learn complex relations at a low cost and a high speed. MPAN relates videos and sentences via multi-perspective interactions to understand the deeper relationships through interactions between attended and unattended memories. The original attention layer for modal interactions takes two sequential inputs, one as a base memory and the other as a reference memory. The attention mechanisms used in existing TML methods interact with two input memories and emphasize the part of the base memory that is related to the reference memory. For example, with the visual feature as a base memory and query feature as a reference memory, the attention layer interacts two different memories and emphasize the parts of the video memory that are related to the query memory. This is a simple and lightweight method for modal interactions. However, relying entirely on a single modal interaction using a single attention layer may emphasize the incorrect locations of a memory. This could suppress crucial information, interfere in the understanding of deeper information, and thus hinder the localization subprocess. To avoid losing essential information, we fused attended memories obtained from multiple cases of attention.

Inspired by the way humans understand an image with objects of interest and relations between them from multiple perspectives, we devised a multi-perspective attention mechanism. Humans match visual information with linguistic information from multiple attention stages [11]. For example, finding a ‘‘horse in front of the cart’’ requires recognizing the horse, cart, and the spatial relationship between the two objects. Humans’ cognitive processes do not recognize the three pieces of information at once but proceed to complete this task through multiple steps. Humans localize a scene by understanding the objects of interest in the scene without any attention and then focus on the horse and cart, respectively, in parallel [12]. As such, humans understand an image from multiple perspectives by controlling their attention toward

a specific target from a specific perspective to find deeper information in the image.

Our MPAN attends to video and query memories in various cases and interacts them with each other to understand their relationship from multiple perspectives. Specifically, to obtain memories attended in various cases, we first attended visual features to query features to gain attended query memory. Then, we attended query features to visual features to obtain attended video memory. We fed attended and unattended video memories into the recurrent layer separately to understand semantic information within the video from two different perspectives. The outputs from the two independent recurrent layers were then concatenated as modal interacted video memory. Each timestep within this memory represents visual semantic information obtained from two different perspectives. As each feature is an output of the recurrent layer, each contains different global information from the different perspectives. By interacting with the two different sets of global information, we assumed that the model learns the local information for each timestep focused on the related spatial feature. Finally, we attended the original query feature to modal interacted video memory.

The second limitation of existing TML methods is that they adopt heavyweight visual feature extractors, which makes them difficult to use in practical scenarios. For visual feature extraction, most TML methods use C3D [13] and I3D [14], which require a vast amount of time and excessive computational overhead. C3D and I3D require 38.5 and 53 GFLOPs to convert 16 and 32 frames of video segments into a visual feature, respectively. To address this limitation, we used the fast-action recognition model PAN [15] as a visual feature extractor. PAN requires 35.7 GFLOPs to extract 32 frames, which is 4.2x and 2.1x lighter than I3D and C3D, respectively. To the best of our knowledge, we are the first to consider its practical usage by adopting a fast-action recognition network as a feature extractor and a lightweight core TML architecture. This opens the possibility of processing large volumes of videos for TML in various practical situations.

The contributions of MPAN are primarily three-fold as follows:

- We introduce a coarse-grained multi-perspective attention mechanism as a substitute for existing heavyweight methods that rely on intensive fine-grained interactions.
- Adopting the latest fast AR model to MPAN as a visual feature extractor, our MPAN showed improved results at a speed above real time for practical uses.
- Extensive experiments using three TML benchmark datasets showed that MPAN can achieve equivalent performance compared to state-of-the-art methods with remarkable efficiency and generalizability.

II. RELATED WORK

We reviewed previous studies related to our approach and categorized these works into three research areas.

A. ACTION RECOGNITION

Action recognition (AR) is a basic research area related to video understanding. For a well-trimmed video, AR aims to classify an action instance contained in the given video into a predefined action label. Recent studies have proposed deep learning-based methods to understand the spatiotemporal information within a given video and solve the AR problem. Similar to object recognition and detection, AR methods have significantly improved their performance compared to early works using handcrafted features. A two-stream network for AR [12], which was the first convolutional neural network (CNN) that surpassed traditional models using handcrafted features, extracts appearance and motion information from an RGB frame and stacked optical flow frames, respectively, and then combines them through late fusion. The C3D network [13] feeds 16 consecutive frames to 3D CNNs to extract appearance and motion information directly from raw RGB frames. The I3D network [14] feeds the RGB and optical flows in 64 frames into two-stream 3D CNNs to better learn the appearance and motion features simultaneously. The publicized models, including the two-stream, C3D, and I3D networks pretrained on Sports-1M [21] and Kinetics [14], are being widely employed as backbone networks to extract unit-level video segment features in video-understanding areas, such as temporal action localization and detection, temporal moment localization, video captioning, and video QA.

B. FAST ACTION RECOGNITION

Proposed lightweight 2D CNNs have considered effectiveness as well as efficiency in AR. Existing AR methods based on 3D CNN or the use of optical flows have demonstrated good performance in modeling spatiotemporal information in a video, but they are computationally heavy. Compared to the AR methods based on 3D CNNs, AR methods based on 2D CNNs are comparatively lightweight. Temporal segment networks (TSN) [17] partition a video into multiple segments, randomly sample a snippet within a video segment, predict its action score for each snippet, and then fuse all predicted action scores for the final prediction. However, a TSN cannot consider the temporal relationship between video segments when fusing predicted action scores and cannot infer complicated temporal relationships. Temporal shift modules (TSM) [18] modify the TSN model by shifting the part of the channels along the temporal dimension. This enables the model to facilitate information exchange among neighboring frames; thus, TMSs achieve the performance of 3D CNNs while maintaining the complexity of 2D CNNS. The temporal memory network (TMnet) [19] was proposed as a self-supervised network that explores spatial and temporal information in a video based on a single frame. The persistent appearance network (PAN) [15] is an efficient and effective action recognition network based on a novel motion cue called the persistence of appearance (PA) and various-timescale aggregation pooling (VAP). Compared to

optical flow via the exhaustive search of all possible motions, PA is efficiently obtained by accumulating pixel-wise differences in feature spaces. PAN used to devise the VAP can model long-range temporal relationships across various timescales and to aggregate the short-term dynamics in PA to long-term dynamics. In this work, we applied PAN as a backbone network to extract visual features from video segments effectively.

C. TEMPORAL MOMENT LOCALIZATION

TML, which is a relatively new task proposed by [1] and [2], aims to find the best temporal interval within an untrimmed video that matches a given sentence query. TML requires the understanding of the semantic contexts within a video and query and the successfully modeling of the relationship between those two inputs.

Early works [1], [2] were based on sliding window methods in which candidate moments are obtained by scanning the entire video and calculating the matching scores for all the candidate moments. Hendricks *et al.* [2] proposed a moment context network (MCN) to calculate the distance between a candidate moment feature with a given query feature by projecting the two features in the same space. Gao *et al.* [1] developed a cross-modal temporal regression localizer (CTRL) to estimate the alignment scores between candidate moments and a query by element-wise addition, multiplication, and concatenation followed by a fully connected layer. The sliding window-based approaches are not only time-consuming, but they also fail to model the global information within a video. As untrimmed videos may contain complex information, understanding each moment within a video independently may cause the temporal relation between moments in the video to be neglected. Methods to model the relationship between a given query and video in a more effective and efficient manner have been proposed.

Based on interaction granularity, recent TML methods can be summarized into two categories: methods with a coarse-grained manner that interact input features across entire videos and the sentence query and methods with a fine-grained manner that divide a video into video segments and a sentence into words and then model interactions between the divided cross-modal features.

The coarse-grained methods are relatively light, fast, and simple. Temporal moment localization using guided attention (TMLGA) [9] uses a single dynamic filter to transfer language information to the visual domain. Attention-based location regression (ABLR) [3] includes a multi-modal co-attention mechanism that attends the sentence to video, video to sentence, and attended sentence to attended video, sequentially. However, the coarse-grained approach cannot model the complex relationship between video and query, thus showing a poor performance.

Fine-grained methods require a higher computational weight than coarse-grained methods but can obtain deeper information. Yuan *et al.* [5] proposed a semantic conditioned dynamic modulation (SCDM) mechanism that interacts the sentence with each visual feature unit, respectively, for temporal convolutions to better correlate and compose sentence-related video contents. Contextual boundary-aware prediction (CBP) [4] incorporates the match-long short-term memory (LSTM), which is composed of three LSTMs, as each timestep of the video is attended, respectively, by sentences to obtain the next step. The 2D-temporal adjacent network (2D-TAN) [6] pools the visual feature into the two-dimensional map, where one dimension indicates the start time of a moment and the other indicates the end time. After applying the Hadamard product to the 2D temporal feature map and query feature, multiple convolution networks encode and interact diverse moments with different lengths to represent adjacent relations. The local–global interaction network (LGI) [7] divides query features into several segments to interact with the video to reflect multi-modal interactions between the query–segment features and visual features on multiple levels. Chen *et al.* [8] devised a pairwise modality interaction (PMI) mechanism that models modality interactions in the sequence of videos and the sequence of queries in a pairwise fashion.

Our MPAN performs the TML task in a coarse-grained manner but can learn the deeper local–global relationship between two inputs using our multi-perspective attention mechanism, as fine-grained methods do.

III. PROPOSED METHOD

Our proposed network takes a sentence query and an untrimmed video as inputs. The main purpose of our MPAN is to retrieve the temporal interval that consists of the start and end times that match the best moment specified by the sentence query. We first introduce feature extraction for MPAN. Then, we explain the architecture of our MPAN in detail. As illustrated in Fig. 2, our model consists of three main components: 1) feature extraction, 2) multi-perspective modal interaction, and 3) temporal localization modules.

A. FEATURE EXTRACTION

We used pretrained models to extract word and visual features from a sentence query and untrimmed video and fed the extracted features into our proposed MPAN as inputs. For a sentence query Q that consisted of M words, we used the Glove300 word2vec model, which was pretrained on Common Crawl [23] to extract a sequence of word features. Each word feature represents information inherited within each word inferred by a linguistic relation. We denoted the sentence feature as a sequence of word features, as $S = \{w_1 \cdots w_m\} \in \mathbb{R}^{m \times d_s}$, where m is the number of words in the sentence and d_s is the size of extracted word feature.

For an untrimmed video V with L frames, we divided the video into video segments with a fixed size of frames. The fixed-sized frames from each video segment were fed into PAN [15] for fast visual feature extraction. We extracted the visual feature right before the classification layer to represent each video segment. We denoted the extracted visual feature as $V = \{v_1 \cdots v_t\} \in \mathbb{R}^{t \times d_v}$, where t is the number of video

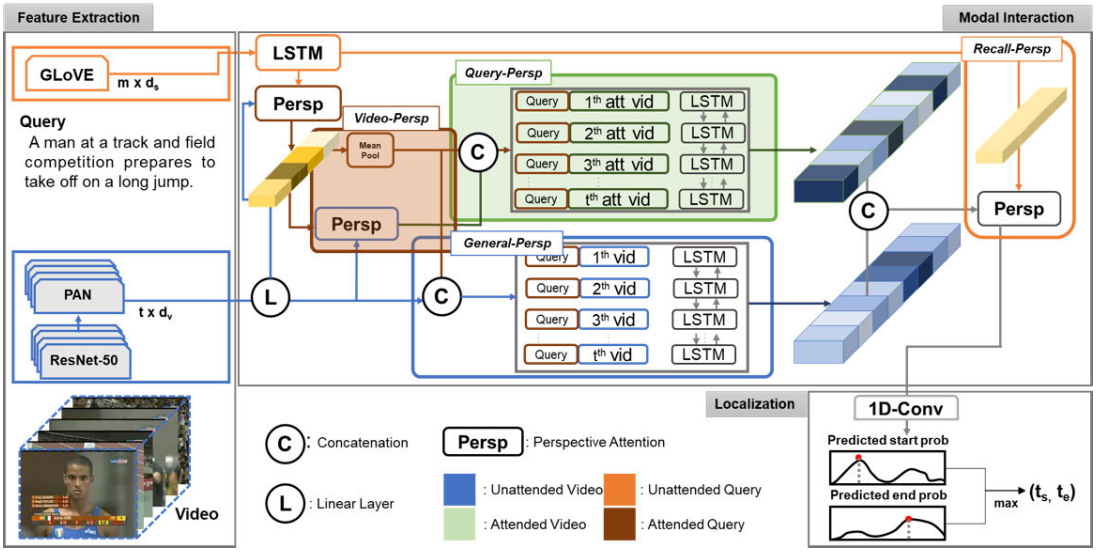

FIGURE 2. MPAN architecture can be organized into three parts: 1) feature extraction, 2) multi-perspective modal interaction, and 3) moment localization modules. Through the multi-perspective attention mechanism based on perspective attention, we added modal interaction in multiple perspectives. This enables MPAN to learn deeper information in a lightweight coarse-grained way instead of in a fine-grained way.

segment within the video, which can be denoted as L/32, and d_v is the size of extracted video feature.

B. PERSPECTIVE ATTENTION

In this section, we present a perspective attention module to understand a memory from a specific perspective of another memory. As mentioned earlier, the attention layer between two different modalities is used to interact with the two memories and learn the modal relationship between them [4], [9]. We applied a multi-head attention layer based on a scaled-dot product, which was originally proposed in the field of machine translation [26]. With two inputs X and Y memories, our multi-head attention is as understanding X from the perspective of Y.

X and Y can be both video or query feature, and we assume the shape of the matrix as $x = \mathbb{R}^{n_x \times d}$ and $y = \mathbb{R}^{n_y \times d}$, where n_x and n_y is the length of input feature and d is the dimension size of each feature. Each dimension size of the video and the sentence feature are fixed to d for modal interaction. We denote our scaled-dot product attention as

$$Mr(X, Y) = softmax(\frac{(X \times W_q) \times (Y \times W_k)^T}{\sqrt{d}})$$

$$Mr(X, Y) \in \mathbb{R}^{n_x \times n_y} \quad (1)$$

$$Att(X, Y) = Mr(X, Y) \times (Y \times W_v) \in \mathbb{R}^{n_x \times d} \quad (2)$$

where $W_q, W_k, W_v \in \mathbb{R}^{d \times d_{att}}$ is a learnable matrix and a softmax operation is applied to every row. $Mr(X, Y)$ directly interacts with the two modal memories via dot-product attention. We designed $Att(X, Y)$ as a weight that gives guidance to X to pay attention to the modal of the Y-related location.

Multi-head attention involves a fixed number of independent attention in parallel, and is formulated as

$$att = Att(X, Y) \quad (3)$$

$$multi(X, Y) = \{att_1 || att_2 || \cdots || att_N\} \quad (4)$$

$$persp(X, Y) = (W_m \times multi(X, Y)) + X$$

$$persp(X, Y) \in \mathbb{R}^{n_x \times d} \quad (5)$$

where N stands for the predefined number of heads, and || stands for the concatenation of two matrix. The results of N parallel attentions are stacked, and then multiplied by linear projection matrix $W_m \in \mathbb{R}^{d \times (N \times d)}$ to interact the results of attention with each other. We then added the outputs of multi-head attention with the original feature X, and denote it as $persp(X, Y)$. The obtained $persp(X, Y)$ represents the X memory understood from the Y memory's perspective.

However, modal interaction obtained by a single perspective attention lacks the consideration of local-global information. Therefore, we propose a multi-perspective attention mechanism to understand complex local-global information.

C. MULTI-PERSPECTIVE MODAL INTERACTION

To understand the contextual information implicated in the relation among words within a sentence, we fed the sentence feature, a sequence of word features, into bi-directional LSTM (Bi-LSTM).

$$q^t_{wf} = LSTM_S(w^t, q^{t-1}_{wf}) \quad (6)$$

$$q^t_{wb} = LSTM_S(w^t, q^{t+1}_{wb}) \quad (7)$$

$$Q = \left\{(q^1_{wf} || q^1_{wb}) \cdots (q^m_{wf} || q^m_{wb})\right\} \quad (8)$$

Bi-LSTM consists of two independent LSTM, which takes the query memory sequentially both forward and backward as an input. The hidden states of the two LSTMs are then stacked as a query memory Q to represent the contextual information from words in a query. To match the dimensional size between the sentence feature and the video feature, we fed the video feature into a single linear layer.

Based on the parts of the videos, different parts of a sentence can be considered important. To selectively focus on video-related information within a query, we first applied video-perspective attention to the query memory. Our video-perspective attention to the query is as follows:

$$Q_{att} = persp_Q(Q, V) \tag{9}$$

where $persp_Q$ represents the video-perspective attention. Q_{att} represents the query memory reinterpreted in the video perspective. Q_{att} focuses on the part related to the video memory. For query-related multi-perspectivity, we saved both the original query memory Q and Q_{att}. Using the attended query memory from the video perspective, we then obtained the attended video memory from the query perspective as follows

$$V_{att} = persp_V(V, Q_{att}) \tag{10}$$

where $persp_V$ represents the query-perspective attention. V_{att} is the attended video memory from the video perspective, which focus on the query-related visual feature. For video-related multi-perspectivity, we saved the original video memory V and V_{att}. To understand the temporal information from the video memory V and V_{att}, we employ another Bi-LSTM. The Bi-LSTM for video memory obtains a relationship between visual features and understands the global context. However, during the process, our model must be able to selectively focus only on the information relevant to the query and neglect the irrelevant information as background. To successfully distinguish related information, we average-pooled the query features and concatenate them with each visual feature of every timestep in a given video memory V_{att}. We fed this memory into a Bi-LSTM as follows

$$q_{att} = \theta(Q_{att}) \tag{11}$$
$$h_f^t = LSTM_f^h(v_{att}^t || q_{att}, h_f^{t-1}) \tag{12}$$
$$h_b^t = LSTM_b^h(v_{att}^t || q_{att}, h_b^{t+1}) \tag{13}$$
$$h^t = (h_f^t || h_b^t) \tag{14}$$

where θ stands for average pooling. We assumed that by feeding the average pooled query and visual features of each time-step into a Bi-LSTM simultaneously, the Bi-LSTM effectively handled the query-relevant information and passed them to the hidden state while the Bi-LSTM distinguished and forgot the irrelevant background information.

We applied the same process in parallel for the raw video memory to learn the relationship in multiple perspectives, as follows

$$p_f^t = LSTM_f^p(v^t || q_{att}, p_f^{t-1}) \tag{15}$$
$$p_b^t = LSTM_b^p(v^t || q_{att}, p_b^{t+1}) \tag{16}$$
$$p^t = (p_f^t || p_b^t) \tag{17}$$

The two Bi-LSTM enables our model to understand a video in two-stream from two different perspectives. p^t and h^t represent both pieces of query-related contextual information in each timestep of the video. We assumed that interacting information from different perspectives is similar to a human's cross-checking process. By interacting the obtained visual features in a two-stream manner, our model compares information from each timestep and upholds the overlapping information, distinguishes inconsistent information, and successfully provides local cues for modal relation. We denote this interaction as follows

$$m_t = (p^t || h^t) \tag{18}$$
$$M = \{m_1, m_2, \cdots, m_t\} \tag{19}$$

We then attended the original query memory to obtain M, as a recall layer. In the previous modal interaction, we only used the attended query memory. As mentioned before, attended memories may neglect crucial information. By applying attention from the perspective of the original query memory, we designed our network to recall the context from the original query and name it as the recall perspective. Applying modal interactions with the original query gives additional guidance as to what needs to be searched for. We denote this recall perspective as follows:

$$L = persp_L(M, Q) \tag{20}$$

where $persp_L$ represents the recall layer, and L is the modal interacted memory that represents the query-related visual feature obtained by the multi-perspective attention mechanism. The key purpose of the multi-perspective modal interaction is to selectively understand the video memory that corresponds to the given query.

Inspired by TMLGA [9], we applied guidance loss to guide our modal interaction process to the last perspective attention layer of our multi-perspective modal interaction. By minimizing the difference between the ground truth and the $Mr(M, Q)$, guidance loss encourages the model to specify higher attention weights for segments related to queries, We average-pooled the $MR^L(M, Q)$ which is responsible for the modal interaction in the last attention layer, and applied guidance loss as follows.

$$\{att(M, Q)_1 || \cdots || att(M, Q)_j\} = persp_L(M, Q) \tag{21}$$
$$Mr(M, Q) \times (Q \times W_v) = att(M, Q) \tag{22}$$
$$MR^L = (\sum_{u=1}^{j}(Mr^u(M, Q)))/j \tag{23}$$
$$Loss_{gd} = -\sum_{k=1}^{T}(1 - \delta_{T^s \le k \le T^e} log(1 - MR^L)) \tag{24}$$

where δ is Kronecker delta representing the temporal interval of ground truth, T^s and T^e denote the ground truth of the starting and ending points, and j denotes for the number of heads in recall-layer.

D. LOCALIZATION LAYER

With the obtained modal-interacted memory in Section 3.C, we calculated the start and end scores of each timestep within a video and directly obtained the best moment consisting of

TABLE 1. Summary of ActivityNet-Captions, Charades-STA, and TACoS. This table includes the number of videos, average seconds of videos, number of samples in the training, test, and validation sets, the average number of queries per video, average seconds of moments to localization, and average length of a query for the three datasets.

Dataset	Video	Video Time	Train	Test	Val	Query per vid	Moment time	Query Length
ActivityNet-Captions	14,926	117.60	37,417	17,031	17,505	4.82	37.14	13.22
Charades-STA	6,672	30.59	12,408	3,720	-	2.42	8.10	7.23
TACoS	127	287.13	10,146	4,083	4,589	148.17	6.10	8.79

the start and end times with the maximums of the start and end scores, respectively.

$$P^S = conv_s(L) \in \mathbb{R}^{t\times 1}$$
$$P^E = conv_e(L) \in \mathbb{R}^{t\times 1} \quad (25)$$

First, our model obtained the modal-interacted memory by maintaining the temporal dimension of the original visual features and supporting the processing of videos with a variable size. Most of the existing methods applied temporal dimension-related layers of neural networks, such as a fully connected layer and temporal convolution layer, along the temporal dimension to interact with adjacent visual features, summarize the interaction results to regress the relative start and end times, or rank all predefined proposals. The methods of applying the layers require the size of the input video to be fixed and require the relative start and end times of each moment to be obtained. Owing to the fixing of the input size, the methods may lose integrated contextual information via temporal interpolation. Furthermore, they may lack performance generalizability depending on the length of the input video because the trained model may be unsuitable for test videos that are much longer or shorter than trained videos.

Second, we predicted start and end scores at all timesteps within a video, which were used as the indicators of the start or end times of the best-matching temporal moment in MPAN as inspired by [9], [20]. We normalized the start and end scores using a softmax function to obtain their probabilities and pick the temporal interval with the highest scores for start and end times, respectively, as the final output. To rank multiple proposals, calculating a score for all predefined moment proposals requires a high computational overhead.

Inspired by TMLGA [9], we set the final output of our MPAN as the probability distributions of start and end scores and trained our model to minimize the Kullback–Leibler divergence loss between the predicted output probability and the ground truth as follows:

$$K(P^S|GT^S) = \sum_{k=1}^{T} P^S(k) log(P^S(k)/S(K))$$
$$Loss_{loc} = K(P^s|GT^S) + K(P^e|GT^E) \quad (26)$$

where GT^S is a 1–D array that has the value of 1 at the start and end times of the match moment and 0 at the rest. This encourages our MPAN to understand the modal relationship between linguistic and visual information and localize the best matching moment.

Our loss function is expressed as

$$Loss = Loss_{gd} + \alpha Loss_{loc} \quad (27)$$

where α is a hyperparameter used to balance the importance between the two loss scores.

IV. EXPERIMENTS

A. DATASETS

The details of three benchmark TML datasets, ActivityNet-Captions, Charades-STA, and TACoS, are summarized in Table 1.

The ActivityNet-Captions dataset consists of 14,926 diverse and open videos gathered from YouTube. It was originally from the ActivityNet dataset with 19,209 videos developed for the task of dense video captioning. The dataset contains untrimmed videos and multiple sentence descriptions with temporal annotations. As annotations for video captioning and TML are reversible, each multiple natural language description with temporal annotations becomes the query for TML. The average length of a video is about 117.60 seconds, with an average of 4.82 queries per video. Each query sentence consists of 13.22 words on average. The length of a matching temporal moment is 37.14 seconds on average. However, the length varies from a few seconds to a few minutes at most. Following the experimental protocol in [10], we took val_1 and val_2 as the validation and test sets, respectively. ActivityNet-Captions has 37,417 queries for the training set, 17,031 queries for the test set, and 17,505 queries for the validation set.

The Charades-STA dataset consists of 6,672 videos on daily indoor activities. Videos in Charades-STA are from the Charades dataset, which consists of 9,848 videos originally proposed for the video AR task. The original Charades only provided a video-level description, so Gao *et al.* [1] extended the dataset using semi-automatic annotation methods. Gao *et al.* [1] applied sentence-level decomposition to video-level description and a matched keyword for each annotated action moment, creating a new sentence to create a moment–query annotation. Each annotation was then verified by human checking. The average length of a video is 30.59 seconds with an average of 2.42 queries per video. Each query sentence consists of 7.23 words, with a matching temporal moment of 8.10 seconds on average. Charades-STA has 12,408 queries for training and 3,720 queries for evaluation.

The TACoS dataset consists of 127 long videos on cooking scenarios in the kitchen. Videos in the TACoS dataset are from the MPII Cooking dataset. Regneri *et al.* [24] extended this dataset by adding sentence descriptions in moments within the video via crowd sourcing. The average length of a video is 287.13 seconds with an average of 148.17 queries

per video. Each query sentence consists of 8.79 words with a matching temporal moment of 6.10 seconds on average. The TACoS dataset has 10,146 queries for the training set, 4,083 queries for the test set, and 4,589 queries for validation set.

The videos in ActivityNet-Captions consist of the most diverse content because they were originally gathered from YouTube. It also consists of the longest and most complex queries between the three datasets. Charades-STA is composed of relatively short indoor videos compared to the other datasets. Its query annotations also have the lowest complexity, making this dataset relatively easy. TACoS is composed of the longest videos between the three datasets and has the smallest number of videos. The videos contain action sequences related only to cooking in the same background, showing the lowest diversity among all the datasets. Thus, temporal moment matching the given query is relatively short compared to the other datasets, which makes this dataset difficult for TML. Each dataset has different properties, and we show the performance comparisons among the three datasets.

B. VIDEO FEATURE EXTRACTION

In this section, we compare the amount of time and computation the visual feature extraction process takes between existing popular methods and PAN [15]. Table 2 summarizes the cost of visual feature extraction for commonly used AR networks and PAN, which we first use for fast TML. By using lite-PAN, we extracted visual features 2.48× faster than C3D [13] and 5.61× faster than I3D [14]. Additionally, the GFLOPs required per frame were 0.46× and 0.12× less than those of C3D and I3D, respectively. The total length of all videos for the ActivityNet-Captions, Charades-STA, and TACoS added up to 487.58 hours, 56.69 hours, and 10.13 hours, respectively. We denote the total time taken to extract features for all the videos contained in each dataset. Based on the total time, we calculate the frames per second (fps) for feature extraction. For I3D feature extraction, it took 189.40 hours, 22.02 hours, and 3.93 hours for ActivityNet-Captions, Charade, and TACoS, respectively. Extracting visual features using C3D took 84.00 hours, 9.76 hours, and 1.75 hours in ActivityNet-Captions, Charade, and TACoS, respectively. For PAN feature extraction, the feature extraction time was 33.78 hours, 3.93 hours, and 0.70 hours, respectively, which is far faster than real time, showing the possibility of the practical use of TML. Our results by PAN were based on a pretrained model trained with the Something-Something V2 [22] dataset, which was originally released by the author. As C3D and I3D models were pretrained based on a much larger dataset with higher quantity and diversity, such as Sports-1M and Kinetics, we assumed that there was a possibility for the pretrained model to show better performance when it was pretrained using a larger dataset.

TABLE 2. Comparisons of used visual features between PAN, C3D, and I3D. The GFLOPs, frames, and fps stand for the computational cost, the number of frames used to extract a feature for a video segment, and the speed of feature extraction, respectively. In addition, ANet and Cha stand for the hours taken to extract features for all videos in ActivityNet-Captions and Charades-STA, respectively.

features	GFLOPs	frames	fps	ANet	Cha	TACoS
PAN	35.7G×1	32	432.96	33.78	3.93	0.70
C3D	38.5G×1	16	174.14	84.00	9.76	1.75
I3D	153G×2	32	77.33	189.40	22.02	3.93

C. IMPLEMENTATION DETAILS

For ActivityNet-Captions, we set the hidden state of all three LSTMs to 256 and the hidden state of three attention layers to 128. We set the number of heads for multi-head attention as 1 for $Multi_S$, 2 for $Multi_V$, and 3 for $Multi_L$. For Charades-STA, we set the hidden state of all three LSTMs to 64 and the hidden state of three attention layers to 128. We set the number of heads for multi-head attention as 1 for $Multi_S$, 2 for $Multi_V$, and 2 for $Multi_L$. For TACoS, we set the hidden state of all three LSTMs to 64 and the hidden state of the three attention layers to 32. We set the number of heads for multi-head attention to 2 for $Multi_S$, 2 for $Multi_V$, and 1 for $Multi_L$. For all three datasets, we used Adam [18] with a fixed learning rate of 1×10^{-4}. We set the batch size to 32 when training ActivityNet-Captions, and set the batch size to 128 when training Charades-STA and TACoS dataset. We adopted batch normalization for normalization and set the dropout rate to 0.1 for every multi-head attention.

D. EVALUATION METRIC

Following [1], we adopted Rank $1@m$ to evaluate and compare our performance. Rank $n@m$ stands for the probability of at least one top-n retrieved moment to exceed the certain threshold m of the temporal intersection over union (tIoU). In this study, we only denoted Rank 1 results because our method localizes moments based on the proposal-free method, returning only a single moment without ranking multiple candidate moments. For runtime measurements, we measured the total runtime for PyTorch code that took extracted visual and word features as inputs and returned a single localized moment as the output. Each runtime was measured using the fixed-size batch that is identical to the size of the batch used to learn our model. We compared the runtime on the same environment, using i7-10700 and a single GTX 1080ti.

E. COMPARISON WITH OTHER TML METHODS

We compared the performance of our MPAN with several TML methods divided into two groups: modal interacted methods in a coarse-grained manner that include CTRL, MCN, ABLR, and TMLGA and modal interaction methods in a fine-grained manner that include SCDM, CBP, CMIN, 2D-TAN, LGI, and PMI. The results on the ActivityNet-Captions, Charades-STA, and TACoS datasets are summarized in Tables 3, 4, and 5, respectively. On the ActivityNet Captions dataset, our MPAN outperformed the existing TML methods, except 2D-TAN and CMIN, including more recent TML methods, such as LGI and PMI, with a

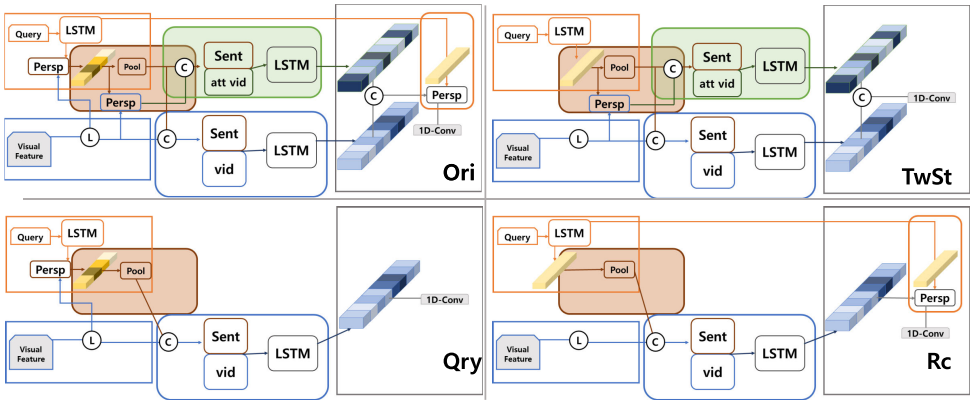

FIGURE 3. Models for the ablation study showed the effectiveness of the component perspective attention layers in a multi-perspective mechanism. Ori means full MPAN. TwSt, Qry, and Rc consist of a single perspective layer for modal interaction, which are a two-stream video perspective layer, query perspective layer, and recall layer, respectively.

TABLE 3. Comparisons of performance, runtime for test sets, and the number of parameters on the ActivityNet-Captions dataset.

Feat	Method	R1@0.5	R1@0.7	Runtime	Param
VGG	MCN [2]	9.58			
C3D	CTRL [1]	14.00			
	ABLR [3]	36.79			
	SCDM [5]	36.75	19.86		
	CBP [4]	35.76	17.80		15.4M
	CMIN [10]	43.40	23.88	45.1	145 M
	2D-TAN (Pool) [6]	**44.51**	26.54	899.5	91.5 M
	2D-TAN (Conv) [6]	44.05	**27.38**		108.6 M
	LGI [7]	41.51	23.07	82.0	53.6 M
	PMI [8]	38.28	17.83		
I3D	TMLGA [9]	33.04	19.26		
PAN	Ours (MPAN)	42.59	23.79	**22.3**	**2.3 M**

TABLE 4. Comparisons of performance, runtime for test sets, and model size, the number of parameters on the Charades-STA dataset.

Feat	Method	R1@0.5	R1@0.7	Runtime	Param
C3D	CTRL [1]	22.63	8.89		
	ABLR [3]	24.36	9.01		
	CBP [4]	36.80	18.87		
	CMIN [10]	39.40	23.31		
	PMI [8]	39.73	19.27		
VGG	2D-TAN [6] (Pool)	39.70	23.31	29.99	60.9 M
	2D-TAN [6] (Conv)	39.81	23.25		69.0 M
I3D	SCDM [5]	54.44	33.43		
	TMLGA [9]	52.02	33.74	4.8	4.0 M
	LGI [7]	**59.46**	35.48	8.2	32.7 M
	Ours(MPAN)	54.70	**36.67**	4.1	**0.4 M**
PAN	Ours (MPAN)	45.16	26.02	**3.4**	**0.4 M**

lightweight architecture. Compared to 2D-TAN and CMIN, which showed the best and second-best performances until now, MPAN achieved more than 86.89% and 98.1% of their performances at 22.3x and 3.7x faster and with 63.0x and 39.8x fewer parameters, respectively.

On the Charades-STA dataset, our MPAN trained by using PAN feature outperformed the TML methods using C3D and VGG except TML methods that used I3D, such as SCMD, TMLGA, and LGI. Given that extracting I3D features requires a high computational overhead, we show the usefulness of PAN features as a practical alternative for TML.

TABLE 5. Comparisons of performance, runtime for test sets, and the number of parameters on the TACoS dataset.

Feat	Method	IoU@0.5	IoU@0.7	Runtime	Param
VGG	MCN [2]	1.25			
C3D	CTRL [1]	13.30			
	ABLR [3]	9.40			
	SCDM [5]	21.17			
	CBP [4]	24.79	**19.10**		15.3 M
	CMIN [10]	18.05		49.0	145.9 M
	2D-TAN (Pool) [6]	**25.32**		470.3	60.9 M
	2D-TAN (Conv) [6]	25.19			82.4 M
I3D	TMLGA [9]	21.65	16.46		
PAN	Ours (MPAN)	25.03	18.12	**10.2**	**0.3 M**

Our MPAN trained by using I3D features outperformed existing state-of-the-art methods at tIoU = 0.7 by 1.19, but showed lower performance on tIoU = 0.5 by 4.76. At the same time, our MPAN achieved 2.4× faster speed with 81.8× fewer parameters.

On the TACoS dataset, MPAN outperformed existing models except 2D-TAN at R1@0.5 and CBP at R1@0.7 Compared to CBP and 2D-TAN, our MPAN achieved comparable but slightly lower performance at tIoU = 0.5 by 0.29 and tIoU = 0.7 by 0.98 with a lightweight architecture. Our MPAN achieved comparable performances in shorter runtime periods with fewer parameters. Compared to 2D-TAN, our MPAN is 46.0× faster at 203.0× fewer parameters.

Thus, by learning deeper information through the coarse-grained multi-perspective attention mechanism, our model showed significantly good performance with an excellent speed-to-accuracy trade-off for the practical use of TML in real-world service scenarios.

F. ABLATION STUDIES

To analyze the effectiveness of a multi-perspective modal interaction mechanism in our MPAN, we investigate the contribution of each attention in the multi-perspective mechanism. For this, We defined the eight variants of our model: (1) None: did not use any attention layer. (2) TwSt: only used the $persp_V$ attention layer and two-stream interaction as mentioned in Fig. 3, (3) Qry: only used the $persp_Q$ attention

layer as mentioned in Fig. 3, (4) Rc: only used the $persp_L$ attention layer as mentioned in Fig. 3, (5) Qry+Rc: excluded the $persp_V$ layer and two-stream interaction, (6) TwSt+Rc: excluded the $persp_Q$, (7) TwSt+Qry: excluded the query recall, $persp_L$ and (8) MPAN: our full model using the three attention layers.

Table 6 summarizes the observed results.

TABLE 6. Ablation study of the effectiveness of the combination of the three attention layers in a multi-perspective mechanism.

Method	ActivityNet		Charades-STA		TACoS	
	R1@0.5	R1@0.7	R1@0.5	R1@0.7	R1@0.5	R1@0.7
None	37.40	20.75	42.69	24.27	22.36	16.71
TwST	38.52	21.87	43.07	24.27	20.22	15.62
Qry	40.31	22.95	41.99	23.32	20.52	13.36
Rc	38.92	22.52	43.10	22.52	22.19	16.58
Qry+Rc	41.55	**23.94**	41.40	23.38	20.32	14.88
TwSt+Rc	39.25	22.53	44.47	25.29	23.81	17.47
TwSt+Qry	41.06	23.24	43.21	24.08	23.20	17.27
MPAN	**42.59**	23.79	**45.16**	**26.02**	**25.32**	**18.34**

First, we show three of the cases, in which applying only a single attention layer that led to worse consequences than applying no attention at all in the three datasets. Especially in the Charades-STA and TACoS datasets, most results from models using only a single attention were lower than the models that did not apply any attention layer. We reason that a single attention layer is insufficient for understanding the modal relationship because the relations between a video and sentence are complex. As a single attention layer focusses on a location that seems to be related, it may suppress meaningful information and hinder the relation-understanding process. However, in all datasets, models that used two or three (full) attention layers showed superior performances compared to models that used only one of the attention layers. According to these results, we demonstrate the effectiveness of a multi-perspective attention mechanism that allows our model to understand deeper contexts and relations by applying various cases of attention layers.

Second, we show a different level of effectiveness in each perspective-attention layer depending on the characteristics of the dataset. The TwST+Rc model on the closed indoor datasets, such as Charades-STA and TACoS, showed good performance while the same model on the ActivityNet-Captions dataset showed relatively worse performance compared to models that used the other two attention layers. Conversely, the (Qry+Rc) model showed high performance on ActivityNet-Caption but relatively low performance on TACoS and Charades-STA. This is because the importance of each perspective layer depends on the characteristics of the three datasets. The importance of some specific perspective attentions can be higher than that of others in a dataset, while they may not be significant in the other datasets. However, with all the perspective attention layers, the model showed the best or comparable performance on all three datasets. This is because by interacting with more perspective layers, the network learns and balances the importance of each perspective and coordinates the optimal weight of each perspective, thus improving performance.

G. QUALITATIVE EVALUATION

For qualitative evaluation, we visualized the localization results on the ActivityNet-Captions dataset in Fig. 4. The results showed the attention weights for the input query obtained by original MPAN model as well as the four pairs of start and end scores predicted by none, TwSt, Rc+TwSt, and MPAN models, which are described in Table 6. The one-dimensional heatmap depicted the average pooled attention weight within $Multi_V$, denoted as *MR*. This heatmap shows which word within the query was more focused on when the perspective layer interacted with the modal relationship between the video and the query. From the given query, "The outside of a building is shown on a snowy day", the attention weight showed the focus was on *building* and *day*. The network understood the context of the query that was focused on *building* and *day*, interacted with the given video to find the scene that matched the understood contexts. However, interaction using a single perspective layer may omit information within words with low attention weights, such as *outside* or *snowy*, which can cause the model to be misinformed. Interaction using this single perspective layer may omit information within words with low attention weights. Figure 4 shows the original MPAN model outperforms the other three models, which are used in the ablation study, for predicting start and end times. By stacking multiple perspective layers, our proposed multi-perspective attention mechanism complements the understanding of information within multiple attended memories, thus successfully localized the moments.

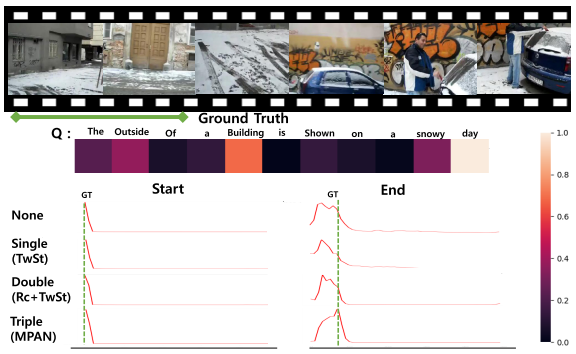

FIGURE 4. Qualitative evaluation of MPAN on the ActivityNet-Captions dataset. Below the sentence query is a attention map visualizing the weight within the sentence-perspective attention. The dotted green lines represent the GT start and end times for this test sample, respectively.

V. CONCLUSION

In this paper, we proposed a novel fast moment localization method using a multi-perspective attention mechanism. To the best of our knowledge, our MPAN is the first attempt that considered the computational overhead for the practical use of TML. Specifically, we devised a multi-perspective

attention mechanism based on multiple perspective attention layers and modal interactions in a coarse-grained manner. In addition, we adopted PAN, a fast AR network, to extract visual features faster than real time. With the extracted PAN features, we employed the multi-perspective mechanism that obtained modal-related information in videos and queries based on three LSTM and attention layers. Then, we employed a one-dimensional convolution layer to localize the best-matching moment in a proposal-free manner by predicting the start and end time scores. On the three benchmark TML datasets, our MPAN achieved comparable performances with a lightweight architecture having far fewer parameters and with even lower prediction and feature extraction times compared to state-of-the-art TML models. As future work, we plan to employ the multi-perspective approach to other tasks requiring modal interactions between video and text, such as VQA and video captioning.

REFERENCES

[1] J. Gao, C. Sun, Z. Yang, and R. Nevatia, "TALL: Temporal activity localization via language query," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Oct. 2017, pp. 5267–5275.
[2] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, "Localizing moments in video with natural language," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5803–5812.
[3] 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*, 2017, pp. 9159–9166.
[4] J. Wang, L. Ma, and W. Jiang, "Temporally grounding language queries in videos by contextual boundary-aware prediction," in *Proc. AAAI*, 2020, pp. 12168–12175.
[5] Y. Yuan, L. Ma, J. Wang, W. Liu, and W. Zhu, "Semantic conditioned dynamic modulation for temporal sentence grounding in videos," in *Proc. NeurIPS*, 2019, pp. 8199–8206.
[6] S. Zhang, H. Peng, J. Fu, and J. Luo, "Learning 2D temporal adjacent networks for moment localization with natural language," in *Proc. AAAI*, 2020, pp. 12870–12877.
[7] J. Mun, M. Cho, and B. Han, "Local-global video-text interactions for temporal grounding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10807–10816.
[8] 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. (ECCV)*, 2020, pp. 333–351.
[9] C. Rodriguez-Opazo, E. 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. (WACV)*, Mar. 2020, pp. 2453–2462.
[10] Z. Zhang, Z. Lin, Z. Zhao, and Z. Xiao, "Cross-modal interaction networks for query-based moment retrieval in videos," in *Proc. SIGIR*, Jul. 2019, pp. 655–664.
[11] G. D. Logan, "Linguistic and conceptual control of visual spatial attention," *Cogn. Psychol.*, vol. 28, no. 2, pp. 103–174, Apr. 1995, doi: 10.1006/cogp.1995.1004.
[12] C. D. Gilbert and W. Li, "Top-down influences on visual processing," *Nature Rev. Neurosci.*, vol. 14, no. 5, pp. 350–363, May 2013, doi: 10.1038/nrn3476.
[13] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.
[14] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Int. Conf. Comput. Vis. (CVPR)*, Jul. 2017, pp. 6299–6308.
[15] C. Zhang, Y. Zou, G. Chen, and L. Gan, "PAN: Towards fast action recognition via learning persistence of appearance," 2020, *arXiv:2008.03462*. [Online]. Available: http://arxiv.org/abs/2008.03462
[16] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. NeurIPS*, 2014, pp. 568–576.
[17] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Amsterdam, The Netherlands: Springer, 2016, pp. 20–36.
[18] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7083–7093.
[19] Z. Liu, J. Li, G. Gao, and A. K. Qin, "Temporal memory network towards real-time video understanding," *IEEE Access*, vol. 8, pp. 223837–223847, 2020, doi: 10.1109/ACCESS.2020.3043386.
[20] J. Lei, L. Yu, T.-L. Berg, and M. Bansal, "TVR: A large-scale dataset for video-subtitle moment retrieval," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 447–463.
[21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1725–1732.
[22] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic, "The 'something something' video database for learning and evaluating visual common sense," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5843–5851.
[23] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. EMNLP*, 2014, pp. 1532–1543.
[24] M. Regneri, M. Rohrbach, D. Wetzel, S. Thater, B. Schiele, and M. Pinkal, "Grounding action descriptions in videos," *Trans. Assoc. Comput. Linguistics*, vol. 1, pp. 25–36, Dec. 2013, doi: 10.1162/tacl_a_00207.
[25] 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. (ICCV)*, Oct. 2017, pp. 706–715.
[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 6000–6010.
[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

JUNGKYOO SHIN (Member, IEEE) received the B.S. degree in information and communication engineering from Dongguk University, Seoul, South Korea, in 2019. He is currently pursuing the M.S. degree under the supervision of Prof. Jinyoung Moon. His current research interests include computer vision, pattern recognition, and machine learning.

JINYOUNG MOON received the B.S. degree in computer engineering from Kyungpook National University (KNU), Daegu, Republic of Korea, in 2000, and the M.S. degree in computer science and the Ph.D. degree in industrial and systems engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2002 and 2018, respectively. Since 2002, she has been working with the Artificial Intelligence Research Laboratory, Visual Intelligence Research Section, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea. Since 2019, she has been with ICT Department, University of Science and Technology (UST), where she is currently an Assistant Professor. Her research interests include action recognition, online and offline action detection, temporal moment localization, and video QA.