Cascaded MPN: Cascaded Moment Proposal Network for Video Corpus Moment Retrieval

SUNJAE YOON1, (Member, IEEE), DAHYUN KIM1, (Member, IEEE), JUNYEONG KIM2, (Member, IEEE), AND CHANG D. YOO1, (Senior Member, IEEE)

1School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea
2Department of Artificial Intelligence, Chung-Ang University, Seoul 06974, Republic of Korea

Corresponding author: Chang D. Yoo (cd_yoo@kaist.ac.kr)

This work was partly supported by the Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-01381, Development of Causal AI through Video Understanding and Reinforcement Learning, and Its Applications to Real Environments) and partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A2C201270611).

ABSTRACT Video corpus moment retrieval aims to localize temporal moments corresponding to textual query in a large video corpus. Previous moment retrieval systems are largely grouped into two categories: (1) anchor-based method which presets a set of video segment proposals (via sliding window) and predicts proposal that best matches with the query, and (2) anchor-free method which directly predicts frame-level start-end time of the moment related to the query (via regression). Both methods have their own inherent weaknesses: (1) anchor-based method is vulnerable to heuristic rules of generating video proposals, which causes restrictive moment prediction in variant length; and (2) anchor-free method, as is based on frame-level interplay, suffers from insufficient understanding of contextual semantics from long and sequential video. To overcome the aforementioned challenges, our proposed Cascaded Moment Proposal Network incorporates the following two main properties: (1) Hierarchical Semantic Reasoning which provides video understanding from anchor-free level to anchor-based level via building hierarchical video graph, and (2) Cascaded Moment Proposal Generation which precisely performs moment retrieval via devising cascaded multi-modal feature interaction among anchor-free and anchor-based video semantics. Extensive experiments show state-of-the-art performance on three moment retrieval benchmarks (TVR, ActivityNet, DiDeMo), while qualitative analysis shows improved interpretability. The code will be made publicly available.

INDEX TERMS Video corpus moment retrieval, cascaded moment proposal, multi-modal interaction, vision-language system.

I. INTRODUCTION

Comprehending visual context together with natural language has been a desiderata in the vision-language research societies. Numerous respectful works have made great strides in bridging computer vision and natural language processing including video/image captioning [30], [34], video moment retrieval [1], [8], video/image question answering [12], [25]. Especially, recent success of video streaming services (YouTube) has drowned interest in video search technologies at fine-grained level. Accordingly, Video Corpus Moment Retrieval (VCMR) [13] is a task to localize a moment in large video corpus, which includes two sub-tasks: (1) identifying relevant video in multiple videos and (2) searching for a specific moment in the identified video. To be concrete, the training of VCMR is given a single video-query pair and boundary label, so that system is trained to find the moment related to the query in the video. In the inference, system is given video corpus and query, where it is required to find moment in the video corpus-level. In this respect, VCMR perform a general format of single video moment retrieval.

In methodological aspect of moment retrieval, methods are typically grouped into two categories: (1) anchor-based method and (2) anchor-free method [16], [22], [36], [40]. The anchor-based method [16], [22] follows intuitive solution that first presets a set of video candidate proposals (via sliding...
properties: (1) Hierarchical Semantic Reasoning (HSR) which incorporates following two main methods, our proposed Cascaded Moment Proposal Network (Cascaded MPG) which precisely performs moment retrieval via devising cascaded multi-modal feature interaction among anchor-free and anchor-based video semantics. In overall pipeline, the HSR provides anchor-based level and anchor-free level semantics via building bipartite graph among video and subtitles, and this multi-level (anchor-based, anchor-free) semantics generates multi-level moment score maps based on a similarity with given query. Finally, the Cascaded MPG associates the contextual meanings from each level of moment score map in a recursive manner and predicts moment pertinent to the query. Cascaded MPN shows effectiveness on three challenging benchmarks (i.e., TVR, DiDeMo, and ActivityNet) and the code will be made publicly available.

II. RELATED WORK

A. VIDEO MOMENT RETRIEVAL

Video moment retrieval (VMR) is a task of localizing a moment pertinent to given natural language sentence. Gauged from the remarkable advancements of natural language processing [5], [23], VMR system has developed from localizing a temporal activity to a task that understands a natural language query and retrieves the relevant moment [6], [33], [35], [38]. Furthermore, improvements of video representation learning [3], [26] contribute to boosting performance of retrieval system from video retrieval to moment retrieval. The first attempts of retrieval [15], [24] were in temporal activity localization, which aims to predict start-end time of moment corresponding pre-defined actions. Henceforth, a large number of advancements of natural language processing bridges temporal activity localization to a language-based moment retrieval. Gao et al. [8] first proposes video moment retrieval, which localizes moments with a sentence describing actions. Hendricks et al. [1] proposed VMR with a simplified format for clip-level video understanding. In the meantime, Mithun et al. [21] proposed different type of retrieval system that finds a video related to text in multiple videos, which is called video retrieval. Recent efforts advance forward to a general form of video moment retrieval. Lei and Li et al. [13], [14] propose systems that perform moment retrieval in a video corpus level, which incorporates video retrieval and single video moment retrieval. This video corpus moment retrieval contains an insight that moment retrieval systems should be operated on a general situation given multiple videos. Inspired by another step of generality, we strive for enhanced interpretability in VCMR.

B. VIDEO PROPOSAL GENERATION

Video moment retrieval system estimates moments by predicting start-end time of the moments related to given query. Current literature for predicting moments can be largely grouped into two categories depending on the way

![Figure 1. Examples of anchor-based and anchor-free moment retrieval system (best viewed in zoom).](image-url)
of predicting the moments as like anchor-based methods and anchor-free methods. Anchor-based methods facilitate context-level video representation learning, which is suitable for learning sequential semantics in the video. Previous works in anchor-based methods generate several proposals with different sizes by sliding window and retrieve the most pertinent one. Lin et al. [16] applied an algorithm considering exploration and exploitation based on reinforcement learning to select top-K proposals. Ma et al. [19] proposes surrogate proposal selection which reduces the redundant proposals via selecting one surrogate from defined proposal group. In the case of anchor-free methods, they are versatile to predicting expected moments without temporal boundary constraints. Yuan et al. [36] proposed Attention Based Location Regression which regresses start-end points of moment related to query via a series of multi-modal co-attentions. Zhang et al. [40] designed two dimensional map with start time and end time as axes, which covers diverse video moments with different lengths. Xiao et al. [32] proposed two-stage candidate proposal generating method, which prepares the two-dimensional map as [40] and searches candidate moment proposals from the map. Wang et al. [31] also proposed two-stage coarse-to-fine grained multi-modal interaction between video and query. Although many novel thoughts have been proposed as above, there are still room for improvement in terms of converging both anchor-free and anchor-based manners, where we made an effort to perform moment retrieval in a fine-grained level in terms of integrating the beauty of these two methods.

III. METHOD

Cascaded MPN takes video and textual query as inputs and produces a moment score map that includes scores in terms of how similar each moment is to the query. The figure 2 presents overall pipeline of Cascaded MPN, where we define anchor-based and anchor-free semantic features from Hierarchical Semantic Reasoning and explain how they are associated to predict best-matched moment in Cascaded MPG. In training, the system is given a single video-query pair and trained to find the moment in the paired video. In inference, it is required to find moment in a video corpus.

A. HIERARCHICAL SEMANTIC REASONING

1) INPUT REPRESENTATION

VCMR systems are given video, subtitles and single sentence query as $V = \{v_i\}_{i=1}^{N_v}, S = \{s_i\}_{i=1}^{N_s}$ and $Q$, where $N_v$ and $N_s$ is the number of frames and subtitles in a single video. To reason hierarchical semantics in anchor-based and anchor-free level, we first define (1) anchor-free semantic features and construct (2) anchor-based semantic features founded on the anchor-free semantics.

2) ANCHOR-FREE SEMANTIC REASONING

As shown in Anchor-free Semantic Reasoning in figure 2, video frames that share the same subtitle have common contextual meaning from that subtitle. To give this common meaning on video frames, we build a bipartite graph between the shared frames and subtitle. To this, we reorganize the video frames to be aligned with single subtitle $s_i$ as $V^s = \{v_j\}_{j=1}^{M^s_i}$, where $M^s_i$ is the number of frames including a subtitle $s_i$ and also define words in that subtitle $s_i$ as $W^s = \{w_j\}_{j=1}^{M^w_i}$, where $M^w_i$ is the number of words in the subtitle $s_i$. The query $Q$ is defined as $d$-dimensional sentence features $E^q \in \mathbb{R}^d$. The final video and subtitle features are embedded into $d$-dimensional space as follows:

$$E^v_w = \text{LN}(\phi_w(V^s) + \text{PE}(W^s)) \in \mathbb{R}^{M^s_i \times d},$$

$$E^v_t = \text{LN}(\phi_t(W^s) + \text{PE}(W^s)) \in \mathbb{R}^{M^w_i \times d},$$
where \( \phi_0 \) and \( \phi_w \) is \( d \)-dimensional embedder. LN is layer normalization [2] and PE is the positional encoding [27]. The frame embedding \( E_{fw} \) and word embedding \( E_{lw} \) contain common semantic of subtitle \( s_t \) and in order to hold this semantic in each video feature, we formally construct video-subtitle graph \( \mathcal{G}_{sf} = (\mathcal{H}^s, \mathcal{E}^s) \) by regarding \( E_{fw} \) and \( E_{lw} \) as nodes group \( \mathcal{H}^s \) in Equation 3. For the edges \( \mathcal{E}^s \) of video-subtitle graph, we design bipartite graph between the words and frames.

\[
\mathcal{H}^s = [E_{fw}^s || E_{lw}^s] \in \mathbb{R}^{M^s \times d}, (3)
\]

where \( M^s = M^f + M^w \) is number of nodes in node group \( \mathcal{H}^s \) and \([[:]]\) denotes concatenation along with frame and word axis. To help understanding, in the section of Anchor-free Semantic Reasoning in figure 2, we depict diagram of bipartite graph showing connectivity between nodes in the graph. To associate these frames embedding with the words embedding, we conduct multi-head graph attention [29]. In each head, we use attention coefficient \( a_{mn}^k \) to give association between any linked two nodes \( m \) and \( n \) within node group \( \mathcal{H}^s \), and \( k \) in \( a_{mn}^k \) means \( k \)-th head like below:

\[
a_{mn}^k = \frac{\exp(\text{LeakyReLU}(w_k^T[K^k\mathcal{H}_m^s][K^k\mathcal{H}_n^s]))}{\sum_{1 \leq N_m} \exp(\text{LeakyReLU}(w_k^T[K^k\mathcal{H}_m^s][K^k\mathcal{H}_n^s]))}, (4)
\]

\( w_k \in \mathbb{R}^{2d} \) is weight vector and \( K^k \in \mathbb{R}^{d \times d} \) are shared embedding. \( \mathcal{H}_m^s \in \mathbb{R}^{d} \) is \( m \)-th node feature in \( \mathcal{H}^s \) and \( \mathcal{H}_n^s, \mathcal{H}_m^s \) follow the same meaning. \( N_m \) is the set of all nodes linked to node \( m \) in the bipartite graph. All nodes are updated with this attention coefficients \( a_{mn}^k \) and we define video-subtitle features \( Z^s \) by averaging of this updated nodes features.

Here, we use video-subtitle, because frames and words in one subtitle get the shared semantic by attention. We define final anchor-free level semantic features by adding this \( Z^s \) to original \( \mathcal{H}^s \):

\[
Z^s_m = \frac{1}{K} \sum_{k=1}^{K} \sum_{n \in N_m} a_{mn}^k W^k \mathcal{H}_m^s, (5)
\]

\[
\mathcal{Y}_m^s = (Z^s + \mathcal{H}^s)[:, M^w] \in \mathbb{R}^{M^s \times d}, (6)
\]

where \( K \) is the number of attention heads. Here, we only used video features in \( (\mathcal{H}^s + \mathcal{H}^s) \) as \( \mathcal{Y}^s \), supposing that subtitle semantics are involved in frames by video-subtitle features \( \mathcal{Z}^s \), where \([::] \) is slicing operation along node-axis.

3) ANCHOR-BASED SEMANTIC REASONING

In the anchor-based semantic reasoning, we first collect all the anchor-free level features \( \mathcal{V} = (\mathcal{Y}_m^s)_{N_t}^{N} \in \mathbb{R}^{N \times d} \) and uniformly divide \( \mathcal{V} \) into \( N \) segments. From this segments, we build \( N \) anchor-based semantics \( \mathcal{C}^N \in \mathbb{R}^{N \times d} \). In one segment, we perform multi-head self-attention to \( \mathcal{V} \) using Transformer [28] and treat average of the segment along frame-axis like below:

\[
\mathcal{V}[i \times N : (i + 1)N] = \text{Head}(\mathcal{V}[i \times N : (i + 1)N]), (7)
\]

where \( \text{Head}(\cdot) \) denotes transformation into \( d \)-dimensional embedder. LN is layer normalization [2] and PE is the positional encoding [27]. The frame embedding \( E_{fw} \) and word embedding \( E_{lw} \) contain common semantic of subtitle \( s_t \) and in order to hold this semantic in each video feature, we formally construct video-subtitle graph \( \mathcal{G}_{sf} = (\mathcal{H}^s, \mathcal{E}^s) \) by regarding \( E_{fw} \) and \( E_{lw} \) as nodes group \( \mathcal{H}^s \) in Equation 3. For the edges \( \mathcal{E}^s \) of video-subtitle graph, we design bipartite graph between the words and frames.

\[
\mathcal{H}^s = [E_{fw}^s || E_{lw}^s] \in \mathbb{R}^{M^s \times d}, (3)
\]

where \( M^s = M^f + M^w \) is number of nodes in node group \( \mathcal{H}^s \) and \([[:]]\) denotes concatenation along with frame and word axis. To help understanding, in the section of Anchor-free Semantic Reasoning in figure 2, we depict diagram of bipartite graph showing connectivity between nodes in the graph. To associate these frames embedding with the words embedding, we conduct multi-head graph attention [29]. In each head, we use attention coefficient \( a_{mn}^k \) to give association between any linked two nodes \( m \) and \( n \) within node group \( \mathcal{H}^s \), and \( k \) in \( a_{mn}^k \) means \( k \)-th head like below:

\[
a_{mn}^k = \frac{\exp(\text{LeakyReLU}(w_k^T[K^k\mathcal{H}_m^s][K^k\mathcal{H}_n^s]))}{\sum_{1 \leq N_m} \exp(\text{LeakyReLU}(w_k^T[K^k\mathcal{H}_m^s][K^k\mathcal{H}_n^s]))}, (4)
\]

\( w_k \in \mathbb{R}^{2d} \) is weight vector and \( K^k \in \mathbb{R}^{d \times d} \) are shared embedding. \( \mathcal{H}_m^s \in \mathbb{R}^{d} \) is \( m \)-th node feature in \( \mathcal{H}^s \) and \( \mathcal{H}_n^s, \mathcal{H}_m^s \) follow the same meaning. \( N_m \) is the set of all nodes linked to node \( m \) in the bipartite graph. All nodes are updated with this attention coefficients \( a_{mn}^k \) and we define video-subtitle features \( Z^s \) by averaging of this updated nodes features. Here, we use video-subtitle, because frames and words in one subtitle get the shared semantic by attention. We define final anchor-free level semantic features by adding this \( Z^s \) to original \( \mathcal{H}^s \):

\[
Z^s_m = \frac{1}{K} \sum_{k=1}^{K} \sum_{n \in N_m} a_{mn}^k W^k \mathcal{H}_m^s, (5)
\]

\[
\mathcal{V}_m^s = (Z^s + \mathcal{H}^s)[:, M^w] \in \mathbb{R}^{M^s \times d}, (6)
\]

where \( K \) is the number of attention heads. Here, we only used video features in \( (\mathcal{H}^s + \mathcal{H}^s) \) as \( \mathcal{V}^s \), supposing that subtitle semantics are involved in frames by video-subtitle features \( \mathcal{Z}^s \), where \([::] \) is slicing operation along node-axis.

B. CASCADED MOMENT PROPOSAL GENERATION

Cascaded Moment Proposal Generation (Cascaded MPG) is introduced to perform moment prediction considering different-level (anchor-free, anchor-base) multi-modal interaction, where it takes inputs as these two semantics, and produces a two-dimensional map containing query-moment similarity score in figure 3(a), where one dimension represents start time of moment and the other dimension represents end time. Cascaded MPG assumes a score map for frame-wise moment retrieval as the same score maps in previous works [13], [14], but also performs contextual reasoning associated with anchor-based semantics. To this, Cascaded MPG includes two main processes: (1) Conditional Moment Score Map generation and (2) Sparsity Pooling, which contribute to multi-modal feature interaction between anchor-based and anchor-free semantics.

1) CONDITIONAL MOMENT SCORE MAP

Conditional Moment Score Map (CMSM) produces two-dimensional moment score map via containing query-moment similarity score in figure 3(a). To build CMSM, we define conditional moment score map generator \( f_{cond} \) by multiplying start time probability of moment \( P(t_s|v, q) \in \mathbb{R}^{L \times 1} \) and conditional end time probability of moment \( P(t_{ed}|t_{st}; v, q) \in \mathbb{R}^{L \times L} \). Given \( d \)-dimensional video feature \( v = [v_1, \ldots, v_L] \in \mathbb{R}^{L \times d} \) with the number of frames \( L \) and sentence feature \( q \in \mathbb{R}^{d} \), the start time probability \( P(t_s|v, q) \) calculates the probability along the frame axis using video-query similarities as:

\[
P(t_s|v, q) = \text{Softm ax ax(Conv}_{1d}^{ed}(vq)) \in \mathbb{R}^{L \times 1}, (9)
\]

where the \( \text{Conv}_{1d}^{ed} \) and \( \text{Conv}_{1d}^{ed} \) below denote 1D convolution layer embedding into start-end probabilities and \( t_s, t_{ed} \) are frame level start-end time. For the conditional end time probability, we first define conditional probability \( P(t_{ed}|t_{st}; v^*, q^*) \),
To this, we first build sparsity mask \( x \) allows the retrieval systems to explore diverse moments in makes the distribution of score map to be sporadic, which performance. To resolve this, our proposed sparsity pooling chance of retrieval in various areas and degrades retrieval per-

Sparsity pooling is introduced to mitigate redundantly over-

that one of the start-frame indices \( i_{st} \in \{0, 1, \ldots, L - 1 \} \) is given as conditional prior. The \( v^*, q^* \) includes start time information from \( \lfloor \lfloor i_{st}/L \rfloor \rfloor \in \mathbb{R}^{L \times (d+1)} \) and \( \lfloor \lfloor i_{st}/L \rfloor \rfloor \in \mathbb{R}^{d+1} \), so that they are utilized for generating conditional end time probability as follows:

\[
v^* = [v]\lfloor i_{st}/L \rfloor_{\text{axis}=1} W_q \in \mathbb{R}^{L \times d}
\]

\[
q^* = [q]\lfloor i_{st}/L \rfloor_{\text{axis}=0} W_q \in \mathbb{R}^d
\]

\[
P(t_{ed}|i_{st}; v^*, q^*) = \text{Softmax} (\text{Conv}_{ed}^{1D}(v^*q^*))^T \in \mathbb{R}^{1 \times L},
\]

where \( W_q, W_q \in \mathbb{R}^{(d+1) \times d} \) are weight matrix and the operation \( [\cdot]\lfloor \cdot \rfloor_{\text{axis}=n} \) is concatenation along the axis \( n \). We stack all the \( P(t_{ed}|i_{st}; v, q) \) along the column axis like Equation 13 and build conditional end time probability \( P(t_{ed}|i_{st}; v, q) \in \mathbb{R}^{L \times L} \). Finally, \( f_{cond}(v, q) \in \mathbb{R}^{L \times L} \) builds CSM by multiplying start time and conditional end time probability, where \( \odot \) is column wise and \( \cdot \) is element wise multiplication:

\[
P(t_{ed}|i_{st}; v, q) = \{P(t_{ed}|0; v^*, q^*); \ldots ; P(t_{ed}|(L-1); v^*, q^*)\}
\]

\[
f_{cond}(v, q) = U_m \cdot (P(t_{st}|i_{st}; v, q) \odot P(t_{ed}|i_{st}; v, q)) \in \mathbb{R}^{L \times L},
\]

where, we give upper triangular mask \( U_m \in \mathbb{R}^{L \times L} \) composed of 1 to remove the score in moments, where end-time comes before start-time. Therefore, the anchor-free score map \( f_{cond}(V, E^0) \) and anchor-level score map \( f_{cond}(C^N, E^0) \) in figure 2 are defined by regarding video feature \( v \) as anchor-free features \( V \) and anchor-based features \( C^N \).

2) SPARSITY POOLING

Sparsity pooling is introduced to mitigate redundantly overlapping moments of frame-level moment score map. The Figure 3(b) shows moment score map in a 3-dimensional view, the high overlapping candidate moments in the map keep similar scores in local region of video, which loses the chance of retrieval in various areas and degrades retrieval performance. To resolve this, our proposed sparsity pooling \( h(x) \) makes the distribution of score map to be sporadic, which allows the retrieval systems to explore diverse moments in the positions and lengths. In detail, the sparsity pooling \( h(x) \) takes input of moment score map \( x \in \mathbb{R}^{L \times L} \) and outputs of the same score map but that holds sparsity in the distribution. To this, we first build sparsity mask \( a \in \mathbb{R}^{L \times L} \) in Figure 4, which includes the following processes: (1) calculating 2D max pooling outputs \( x_N \) from original score map \( x \) with kernel size of \( N \times N \) and stride of \( N \), (2) generating 2D upsampled map \( \hat{x} \) by nearest neighbor upsampling up to the original size of \( x \) and (3) finally, preparing sparsity mask from element-wise dividing \( x \) by \( \hat{x} \). The aforementioned processes can be described as follows:

\[
x_N = \text{MaxPool2D}(x) \in \mathbb{R}^{\frac{L}{N} \times \frac{L}{N}}
\]

\[
\hat{x} = \text{Upsample}(x_N) \in \mathbb{R}^{L \times L}
\]

\[
a = x/\hat{x}
\]

\[
h(x) = a \cdot x \in \mathbb{R}^{L \times L},
\]

where \( ./ \) and \( \cdot \) are element wise dividing and multiplication. In Equation 15 and 16, the \( \hat{x} \) contains a local maximum score of original score map \( x \). In this process, sparsity pooling \( h(x) \) maintains the maximum score in the \( N \times N \) window and builds sparse distribution within the windows.

3) CASSCDED MOMENT PROPOSAL GENERATION

This section introduces cascaded moment proposal generation (Cascaded MPG) algorithm in detail. Based on the anchor-free semantic \( V \in \mathbb{R}^{N_i \times R} \) and anchor-based semantic \( C^N \in \mathbb{R}^{N \times d} \), Cascaded MPG produces 2D moment score map for moment prediction, where the algorithm relies on the conditional moment score map generator \( f_{cond} \) and sparsity pooling \( h(x) \) in a recursive manner. Figure 5 summarizes the pipeline of Cascaded MPG. In the first stage, \( f_{cond}(V, E^0) \) builds anchor-free moment score map \( \bar{M} \). The sparsity pooling \( h(x) \) bridges to the next stage by performing sparsity masking on the score map. At the last stage, \( f_{cond}(C^N, E^0) \) builds an anchor-based moment score map \( \bar{M}^N \), after then the map is up-sampled to the original anchor-free score map and added to the output of the sparsity pooling. The whole pipeline of Cascaded MPG is described in Algorithm 1.

Algorithm 1 Cascaded MPG

1: Input: conditional moment score map generator \( f_{cond} \), sparsity pooling \( h \), anchor-based semantics \( C \), anchor-free semantics \( V \), query \( E^0 \).
2: Output: 2D moment score map \( \bar{M} \) Initialize anchor-free score map \( M_0 = f_{cond}(V, E^0) \).
3: for \( i \leftarrow 1 \) to \( T \) do
4:  Perform sparsity pooling: \( M_{i-1} \leftarrow h(M_{i-1}) \)
5:  Update the number of anchors: \( N = 2^i \)
6:  Get anchor-level score map: \( \bar{M}^N = f_{cond}(C^N, E^0) \)
7:  Update score map: \( M_i = \sigma(M_{i-1} + \text{Upsample} (\bar{M}^N)) \) \((\sigma \text{ is sigmoid.})\)
8: end
9: Update output \( \bar{M} = M_T \).

4) TRAINING

The anchor-free semantics \( V \in \mathbb{R}^{N_i \times d} \) and anchor-based semantics \( C \in \mathbb{R}^{N \times d} \) are trained under two types of loss
as follows: (1) video-level loss; and (2) moment-level loss. In video-level loss, we use hinge loss in terms of cosine similarity with query feature $E^q \in \mathbb{R}^d$ like below:

$$L^V_v = \max[0, \Delta_c - p(s(V^+, E^q)) + p(s(V^-, E^q))].$$

(19)

$$L^C_v = \max[0, \Delta_c - p(s(C^+, E^q)) + p(s(C^-, E^q))].$$

(20)

where $+$ is positive from video-query pair and $-$ is negative from other videos in a batch. $p(\cdot)$ is 1D max-pooling and $s(\cdot, \cdot)$ is cosine similarity. $\Delta_c = 0.1$ is a margin and we select surrogate cosine similarity by max-pooling among anchor-based and anchor-free semantics. In moment-level loss, we use cross-entropy loss $CE$ in terms of ground-truth start-end time $(g_{st}, g_{end})$ and predicted start-end time probabilities as:

$$L_m = CE(g_{st}, P(t_{st}|V, E^q)) + CE(g_{end}, P(t_{end}|t_{st} = g_{st}; V, E^q)).$$

(21)

$$L_v = L^V_v + L^C_v$$

(22)

$$L = \alpha L_v + \beta L_m,$$

(23)

Total loss $L$ is defined with $L_v$ and $L_m$ using hyperparameters $\alpha$ and $\beta$.

IV. EXPERIMENTS
A. DATASETS
We validate our proposed Cascaded MPN on three recent benchmarks (TVR, DiDeMo, ActivityNet Captions) as follows: (1) TV show Retrieval (TVR) dataset [13] is constructed under 6 TV shows across 3 genres: medical dramas, sitcoms and crime dramas. TVR contains 109K queries from 21.8K videos with subtitles and each video is about multi-character interactions with 60-90 seconds in length. For the fair comparison [13], [14], [37]. We also split TVR into 80% train, 10% val, 5% test-private, 5% test-public. The test-public is prepared for official leaderboard. (2) The Distinct Describable Moments (DiDeMo) dataset [1] covers over 10,000 unedited, personal videos in diverse scenarios with pairs of trimmed video segments and referring expressions. DiDeMo is split into about 80% train, 10% val, and 10% test, where each video is pruned up to maximum of 30 seconds. To relieve the complexity, all videos are uniformly divided into 5-second segments, so that labels are of 21 possible moments from a single video. (3) ActivityNet Captions [11] contains 20k videos with 100k temporal descriptions. The average length of the video is about 117 seconds, and the queries are about 14.8 words. 10,009 videos are available for training and 4,917 for validation (val_1). We evaluate models on the val_1 split.

B. EXPERIMENTAL DETAILS
1) EVALUATION METRIC
For the evaluation of VCMR, prediction is correct if: (1) a predicted video matches the ground-truth video; and (2) the predicted moment has high overlap with the ground-truth moment. Average recall at K (R@K) over all queries is used as the evaluation metric, where temporal Intersection over Union (tIoU) is used to measure the overlap between the predicted moment and the ground-truth. We first predict top-100 videos from video corpus by measuring $p(s(V, E^q))$ in Equation 19 as and Cascade MPG localizes the best matched moment among the videos.

2) TRAINING DETAILS
We used same video features with [14] using SlowFast [7] pre-trained on Kinetics [10] and ResNet-101 [9] pre-trained on ImageNet [4]. The text features are contextualized token features from pre-trained on RoBERTa [17]. Our model is trained on NVIDIA Quadro RTX 8000 (48GB of memory) GPU. The dimension of hidden layer is $d = 768$ and the number of attention heads in hierarchical semantic reasoning is $K = 8$. We use AdamW optimizer [18] with a learning rate of $3e-5$ weight decay of 0.01 to train the model. The training hyperparameters in Equation 23 are $\alpha = 8$ and $\beta = 0.01$.

C. BENCHMARKING RESULTS
Table 1 and Table 2 summarize the experimental results on TVR, DiDeMo and ActivityNet comparing best performed methods, including CAL, XML, HERO, HAMMER, ReLoCLNet. Table 1 presents experimental results from
TABLE 1. Performance comparisons for VCMR on TVR (test-public), ActivityNet and DiDeMo. †: reconstruction-based results, ‡: without pre-training.

| Method          | TVR †          | ActivityNet Capt. † | DiDeMo       |
|-----------------|----------------|---------------------|--------------|
|                 | tIoU=0.7       | R@1 | R@10 | R@100 | tIoU=0.7 | R@1 | R@10 | R@100 | tIoU=0.7 | R@1 | R@10 | R@100 |
| XML [13]        | 2.76           | 9.08 | 15.97 | -     | -       | -   | -    | 1.59  | 6.17  | 25.44 |
| HERO [14]       | 2.98           | 10.65 | 18.25 | 1.06* | 6.54*  | 15.34* | -       | -     | 2.14  | 11.43 | 36.09 |
| HAMMER [37]     | 5.13           | 11.38 | 16.71 | 1.74  | 8.75   | 19.08 | -       | -     | -     | -     | -     |
| ReLoCLNet [39]  | 4.15           | 14.06 | 32.42 | 1.82  | 6.91   | 18.33 | -       | -     | -     | -     | -     |
| Cascaded MPN    | 5.27           | 16.12 | 35.11 | 2.02  | 9.56   | 23.42 | 3.02   | 15.57 | 41.31 |

TABLE 2. Performance comparisons for VCMR with large data pre-training (HowTo100M) on TVR (test-public) and two sub-tasks: Single video moment retrieval (SVMR†) and video retrieval (VR†) on TVR (validation). †: without pre-training.

| Method          | VCMR†          | SVMR†          | VR†          |
|-----------------|----------------|----------------|--------------|
|                 | tIoU=0.7       | R@1 | R@10 | R@100 | tIoU=0.7 | R@1 | R@10 | R@100 | tIoU=0.7 | R@1 | R@10 | R@100 |
| CAL [5]         | -              | -   | -    | -     | 4.68    | 20.17 | -   | -     | -     | -     | -     | -     |
| XML [13]        | 3.25           | 11.38 | 29.51 | 13.41 | 31.11  | 16.54 | 50.41 |
| HERO [14]       | 6.21           | 14.06 | 36.66 | 3.76  | 9.59   | 19.44 | 52.43 |
| ReLoCLNet [39]  | -              | -   | -    | 15.04 | 45.24  | 22.13 | 57.25 |
| Cascaded MPN    | 7.14           | 23.03 | 45.18 | 16.23 | 46.86  | 28.54 | 61.73 |

TABLE 3. Ablation study on model variants of cascaded MPN on TVR (validation). (SP: sparsity pooling, AFr: anchor-free semantic reasoning, ABr: anchor-based semantic reasoning, CMSM: Conditional moment score map).

| SP | AFr | ABr | CMSM | tIoU=0.7 | R@1 |
|----|-----|-----|------|---------|-----|
| ✓  | ✓   | ✓   | ✓    | 4.34    | ✓   |
| ✓  | ✓   | ✓   | ✓    | 4.96    | ✓   |
| ✓  | ✓   | ✓   | ✓    | 5.02    | ✓   |
| ✓  | ✓   | ✓   | ✓    | 5.21    | ✓   |
| ✓  | ✓   | ✓   | ✓    | 5.32    | ✓   |

TABLE 4. Ablation study on cascaded MPN layer.

| # of Cascaded layers | tIoU=0.7 | R@1 |
|----------------------|---------|-----|
| Cascaded layer n = 1 | 4.03    |     |
| Cascaded layer n = 2 | 4.89    |     |
| Cascaded layer n = 3 | 5.32    |     |
| Cascaded layer n = 4 | 5.30    |     |
| Cascaded layer n = 5 | 5.13    |     |

As the subtitles are unavailable in the ActivityNet and DiDeMo, we utilize video features from video encoder instead of anchor-free semantic features. To the further experiment of DiDeMo, we complement the subtitles with the auxiliary features using Audio Speech Recognition (ASR) from [14], which makes anchor-free semantics available and gives performance gain up to 31.3 in the measure of tIoU=0.7, R@1 on DiDeMo. As reported in [14], the previous results from DiDeMo and TVR are also conducted under pre-training with large-scale dataset HowTo100M [20]. For the fair comparison, we also presents the results from pre-training of HowTo100M on DiDeMo and TVR from Table 1 and 2. Besides, for the two sub-tasks of VCMR: SVMR and VR, Table 2 presents the results on TVR, where the Cascaded MPN also validate the effectiveness.

D. ABLATION STUDY
We perform ablation studies with several variants of Cascaded MPN. Table 3 summarizes ablative results of sparsity pooling (SP), anchor-free semantic reasoning (AFr), anchor-based semantic reasoning (ABr) and conditional moment score map (CMSM). The absence of anchor-based semantic features gives large performance drop, which implies the contextual understanding is crucial in the video with multi character interactions. For the ablation of AFr, we substitute video-subtitle semantics with original video features embedded into d-dimensional space. For the absence of CMSM, we utilize $P(t_{ed} | v, q)$ instead of $P(t_{ed} | I_{st}; v, q)$ and define score map generator as $f_{ cond}(v, q) = U_m(P(t_{ed} | v, q))P(t_{ed} | v, q)^{-1}$. CMSM is also worth that it saves about half of training time by early saturation. The Table 4 presents experimental results according to the variants of cascaded layer length. The cascaded layer n = 3 shows highest performance with kernel size N = 2, 4, 8 in sparsity pooling and more long layers give a slight deterioration in performance. This is because in the early stage of cascaded proposal generation, it is effective to remove many redundant candidates, but after the layers longer than 5, as there are not many redundant candidates, it may damage the proposal scores in way of dropping performance.

E. QUALITATIVE RESULTS
Figure 6 represents moment prediction, conditional state-end probability, and Figure 7 represents moment score map after sparsity pooling. In the Figure 6, the red curve is the start-probability distribution and the blue curve is the end-probability distribution. From these two distributions, final
based method is vulnerable to heuristic rules of generating moment score probability generation and sparsity pooling. From these, the conditional moment score probability generation and sparsity pooling have a positive effect on the retrieval.

F. LIMITATIONS

We think that Cascaded MPN used many attention weights to represent the two different types of video representations: (1) anchor-free semantic and (2) anchor-based semantic, which took a lot of time to fully learn each features. In this regard, further study would be possible to generate this two hierarchical representations in a more efficient way (weight sharing, model pruning) or another types of representation. We believe that many valuable researches will be built under motivation of overcoming these limitations.

V. CONCLUSION

We propose Cascaded MPN for video corpus moment retrieval to overcome two main challenges: (1) anchor-free method is vulnerable to heuristic rules of generating video proposals, which incurs restrictive moment prediction in length; and (2) anchor-free method systemically suffers from insufficient understanding of long and sequential video semantics. Therefore, our proposed cascaded MPN incorporates following two properties: (1) Hierarchical Semantic Reasoning which gives video understanding from anchor-free level to anchor-based level by building hierarchical video graph, and (2) Cascaded Moment Proposal Generation which precisely performs moment retrieval by devising cascaded multi-modal interaction among anchor-free and anchor-based level video semantics. Experimental results on three benchmarks show effectiveness of our Cascaded MPN.

REFERENCES

[1] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Durrell, and B. Russell, “Localizing moments in video with natural language,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 5803–5812.
[2] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” 2016, arXiv:1607.06450.
[3] J. Carreira and A. Zisserman, “Quo vadis, action recognition? A new model and the kinetics dataset,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6299–6308.
[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.
[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.
[6] V. Escorcia, M. Soldan, J. Sivic, B. Ghanem, and B. Russell, “Temporal localization of moments in video collections with natural language,” 2019, arXiv:1907.12763.
[7] C. Feichtenhofer, H. Fan, J. Malik, and K. He, “SlowFast networks for video recognition,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 6202–6211.
[8] J. Gao, C. Sun, Z. Yang, and R. Nevatia, “TALL: Temporal activity localization via language query,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 5267–5275.
[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[10] W. X. Z. C. H. S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, “The kinetics human action video dataset,” 2017, arXiv:1705.06950.
[11] 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.
[12] J. Lei, L. Yu, M. Bansal, and T. L. Berg, “TVQA: Localized, compositional video question answering,” 2018, arXiv:1809.01896.
[13] J. Lei, L. Yu, T. L. Berg, and M. Bansal, “TVR: A large-scale dataset for video-subtitle moment retrieval,” 2020, arXiv:2001.09099.
[14] L. Li, Y.-C. Chen, Y. Cheng, Z. Gan, L. Yu, and J. Liu, “HERO: Hierarchical encoder for video-i-language omni-representation pre-training,” 2020, arXiv:2005.00200.
[15] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, “BSN: Boundary sensitive network for temporal action proposal generation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 3–19.
[16] Z. Lin, Z. Zhao, Z. Zhang, Q. Wang, and H. Liu, “Weakly-supervised video moment retrieval via semantic completion network,” in Proc. AAAI Conf. Artif. Intell., vol. 34, 2020, pp. 11539–11546.
[17] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RobBERTa: A robustly optimized BERT pretraining approach,” 2019, arXiv:1907.11692.
[18] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” 2017, arXiv:1711.05101.
[19] M. Ma, S. Yoon, J. Kim, Y. Lee, S. Kang, and C. D. Yoo, “VLANet: Video-language alignment network for weakly-supervised video moment retrieval,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2020, pp. 156–171.
