Partially Relevant Video Retrieval

Jianfeng Dong
Xianke Chen
Zhejiang Gongshang University

Minsong Zhang
Zhejiang Gongshang University

Xun Yang
University of Science and Technology
of China

Shujie Chen
Zhejiang Gongshang University

Xirong Li*
Key Lab of DEKE, Renmin University
of China

Xun Wang*
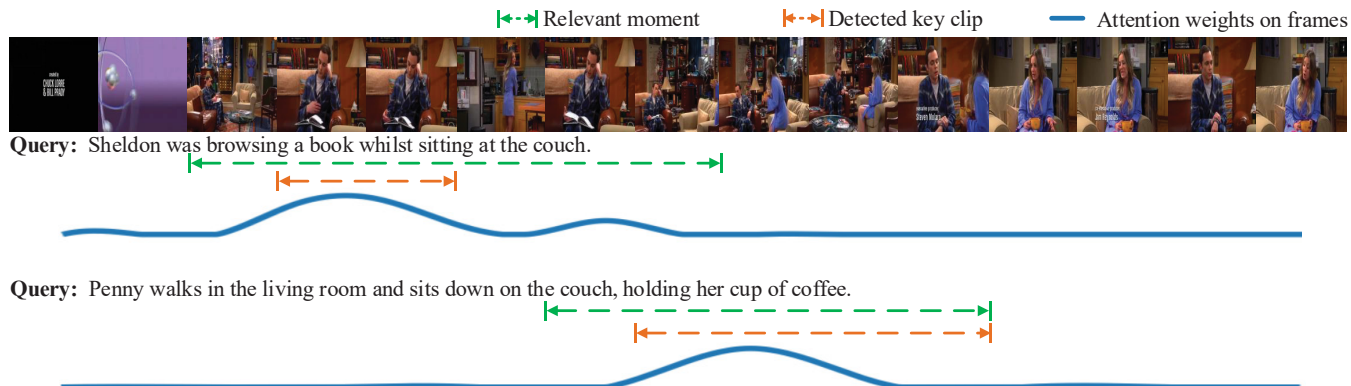Zhejiang Gongshang University

Figure 1: Two textual queries partially relevant to a given video. Only a specific moment in the video is relevant to the corresponding query, while the other frames are irrelevant. We formulate the task of partially relevant video retrieval (PRVR) as a multiple instance learning problem, and propose a MS-SL network. MS-SL first detects a key clip that is most likely to be relevant to the query. Then, the importance of each frame is measured in a fine-grained temporal scale under the guidance of the key clip. The final similarity is computed by jointly considering the query's similarities with the key clip and the frames.

ABSTRACT

Current methods for text-to-video retrieval (T2VR) are trained and tested on video-captioning oriented datasets such as MSVD, MSR-VTT and VATEX. A key property of these datasets is that videos are assumed to be temporally pre-trimmed with short duration, whilst the provided captions well describe the gist of the video content. Consequently, for a given paired video and caption, the video is supposed to be fully relevant to the caption. In reality, however, as queries are not known a priori, pre-trimmed video clips may not contain sufficient content to fully meet the query. This suggests a gap between the literature and the real world. To fill the gap, we propose in this paper a novel T2VR subtask termed *Partially Relevant Video Retrieval* (PRVR). An untrimmed video is considered to be partially relevant w.r.t. a given textual query if it contains a moment relevant to the query. PRVR aims to retrieve such partially relevant videos from a large collection of untrimmed

videos. PRVR differs from single video moment retrieval and video corpus moment retrieval, as the latter two are to retrieve moments rather than untrimmed videos. We formulate PRVR as a multiple instance learning (MIL) problem, where a video is simultaneously viewed as a bag of video clips and a bag of video frames. Clips and frames represent video content at different time scales. We propose a Multi-Scale Similarity Learning (MS-SL) network that jointly learns clip-scale and frame-scale similarities for PRVR. Extensive experiments on three datasets (TVR, ActivityNet Captions, and Charades-STA) demonstrate the viability of the proposed method. We also show that our method can be used for improving video corpus moment retrieval.

CCS CONCEPTS

• **Information systems** → **Video search**.

KEYWORDS

Video-Text Retrieval, Partially Relevant, Multiple Instance Learning, Video Representation Learning

---

*Corresponding authors: Xirong Li and Xun Wang

1 INTRODUCTION

With the advent of the big data era, millions of videos are uploaded to the Internet every day. There is an increasing need of retrieving videos from the big data. As common users prefer to express their information need by natural-language queries, research on text-to-video retrieval (T2VR) is important [8, 32, 60]. Given a query in the form of a natural language sentence, T2VR asks to retrieve videos that are semantically relevant to the given query from a gallery of videos. Current methods [7, 18, 22, 27, 37, 39] for T2VR are trained and tested on video-captioning oriented datasets such as MSVD [4], MSRVTT [59] and VATEX [54]. A key property of these datasets is that videos are assumed to be temporally pre-trimmed with short duration, whilst the provided captions well describe the gist of the video content. Consequently, for a given paired video and caption, the video is supposed to be fully relevant to the caption. In reality, however, as queries are not known a priori, pre-trimmed video clips may not contain sufficient content to fully meet the query. This suggests a gap between the literature and the real world.

To fill the above gap, we propose in this paper a novel T2VR subtask termed *Partially Relevant Video Retrieval (PRVR)*. An untrimmed video is considered to be partially relevant w.r.t. a given textual query as long as the video contains a (short) moment relevant w.r.t. the query, see Fig. 1. PRVR aims to retrieve such partially relevant videos from a large collection of untrimmed videos. Because where the relevant moment is localized and how long it lasts are both unknown (see Fig. 2), PRVR is more challenging than the conventional T2VR task.

Observing the connection between PRVR and Multiple Instance Learning (MIL) [9, 40] at a high level, we tackle the new task by a multi-scale MIL approach. In the current context, a video is simultaneously viewed as a bag of video clips and a bag of video frames. Clips and frames represent video content at different temporal scales, which is helpful for handling moments of varying temporal lengths. Besides, based on the multi-scale video representation, we propose Multi-Scale Similarity Learning (MS-SL) network. In MS-SL, we regard the clip-scale as the coarse temporal granularity as it is typically of longer duration. Besides, the frame-scale is regraded as the fine-grained temporal granularity, as frames usually reflect more detailed content of videos. The multi-scale similarity learning consists of a clip-scale SL branch built on clip-scale video representation and a frame-scale SL branch built on frame-scale video representation. They are jointly learned in a coarse-to-fine manner. Note that the two similarity learning branches are not independent. In clip-scale SL, a key clip that is most likely to be relevant to the query will be detected. Then the clip-scale similarity is computed as the similarity between the key clip and the query. Additionally, the key clip is regarded as a guide for frame-scale SL to measure the importance of each frame in a fine-grained temporal scale. The frame-scale similarity is computed as the similarity between the weighted frames with the query. Finally, the clip-scale similarity and the frame-scale similarity are jointly used to measure the final video-text similarity.

It is worth noting that PRVR differs from *Single Video Moment Retrieval (SVMR)* [58, 61, 65, 69] and *Video Corpus Moment Retrieval (VCMR)* [24, 29, 66, 68], as the latter two are to retrieve moments rather than untrimmed videos. Additionally, although our model
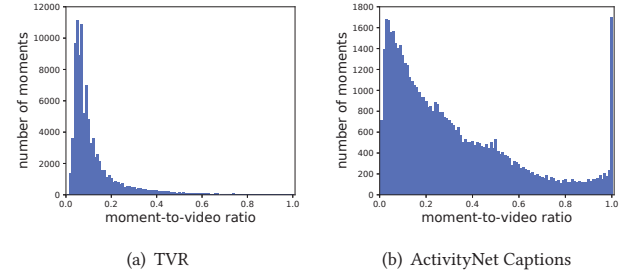

(a) TVR   (b) ActivityNet Captions

Figure 2: Distribution of moment-to-video ratio on (a) TVR and (b) ActivityNet Captions. Moment-to-video ratio indicates the moment's length ratio in the entire video. Moments show a large variance in their temporal lengths.

is proposed for PRVR, it can also be used for improving VCMR. In sum, our main contributions are as follows:

• We propose a new T2VR subtask named PRVR, where an untrimmed video is considered to be partially relevant with respect to a given textual query if it contains a moment relevant to the query. PRVR aims to retrieve such partially relevant videos from a large collection of untrimmed videos.

• We formulate the PRVR subtask as a MIL problem, simultaneously viewing a video as a bag of video clips and a bag of video frames. Clips and frames represent video content at different temporal scales. Based on multi-scale video representation, we propose MS-SL to compute the relevance between videos and queries in a coarse-to-fine manner.

• Extensive experiments on three datasets (TVR [29], ActivityNet Captions [28], and Charades-STA [19]) demonstrate the viability of the proposed method for PRVR. We also show that our method can be used for improving video corpus moment retrieval. Source code and datasets are available at http://danieljf24.github.io/prvr

2 RELATED WORK

T2VR. The T2VR task has gained much attention in recent years [7, 10, 21, 22, 25, 36, 49, 53, 60], aiming to retrieve relevant videos by a given query from a set of pre-trimmed video clips. The retrieved clips are supposed to be fully relevant to the given query. A common solution for T2VR is to first encode videos and textual queries and then map them into common embedding spaces where the cross-modal similarity is measured. Therefore, current works mainly focus on video encoding [17, 27, 37, 48], sentence encoding [7, 8, 32], and their cross-modal similarity learning [12, 18, 57, 63]. Different from the above works, we consider a more realistic scenario, where videos are supposed to be partially relevant to a specific query. We thus focus more on how to measure the partial relevance between textual queries and videos.

VMR. The VMR task is to retrieve moments semantically relevant to the given query from a given single untrimmed video or a large collection of untrimmed videos. The former is known as SVMR [1, 34, 35, 47, 56, 62, 70], and the latter is known as VCMR [14, 45, 55, 66]. In SVMR, existing methods mainly concentrate on how to precisely localize temporal boundings of target moments, and could be typically classified as proposal-based methods [5, 20, 52, 64, 67] and proposal-free methods [6, 47, 65].

Proposal-based methods first generate multiple moment proposals, then match them with a query to determine the most relevant one from the proposals. Without generating moment proposals, proposal-free methods predict the start and end time points of the target moment based on the fused video-query feature. As the extension of SVMR, the VCMR task is to retrieve moments (or video segments) that are semantically relevant w.r.t. a given query from a collection of untrimmed videos. The state-of-the-art methods (*e.g.,* ReLoCLNet [68] and XML [29]) for VCMR have a two-stage workflow. The first stage is to retrieve a number of candidate videos which may contain the target moment, while the second stage is to retrieve moments from the candidate videos.

Different from video moment retrieval aiming to retrieve moments, our proposed PRVR task aims to retrieve untrimmed videos. Besides, while PRVR is similar to VCMR's first stage yet requires no moment-level annotations as commonly needed for VCMR. Therefore, a method for PRVR can in principle be used to improve a two-stage method for VCMR, and our proposed model is designed for PRVR, it can also be used for improving VCMR.

MIL. MIL [9, 40] is a classical framework for learning from weakly annotated data, and widely used for classification tasks [30, 33]. In MIL, a sample is defined as a bag of multiple instances, and there is only a label associated with the bag instead of the instance. Besides, a bag is positive if the bag contains at least one positive instance and negative if it contains no such positive instance. Existing MIL methods could be roughly grouped into instance-based methods [16, 44, 46] and embedding-based methods [26, 30, 50]. The former typically predicts a score of each instance in the bag and aggregates them to generate a bag score. The latter usually aggregates embedding of all instances into a bag embedding, then outputs a bag score based on the bag embedding. In this work, we formulate the PRVR task as a MIL problem. Different from the previous MIL works that usually regrade a sample as a specific bag of instances, in this work a video is simultaneously viewed as a bag of video clips and a bag of video frames. Moreover, we employ MIL for the retrieval task instead of the classification task.

3 OUR METHOD

We formulate PRVR as a MIL problem. As moments relevant to queries typically show large variations in their temporal lengths, we devise multi-scale video representation to represent videos at multiple temporal scales, obtaining a bag of video clips of varying lengths and a bag of video frames. Based on the two bags, we further propose multi-scale similarity learning to measure the partial query-video relevance, see Fig. 3.

3.1 Formulation of PRVR

Given a natural language query, the task of *PRVR* aims to retrieve videos containing a moment that is semantically relevant to the given query, from a large corpus of untrimmed videos. As the moment referred to by the query is typically a small part of a video, we argue that the query is partially relevant to the video. It is worth pointing out that PRVR is different from conventional T2V retrieval [7, 11, 22], where videos are pre-trimmed and much shorter, and queries are usually fully relevant to the whole video.
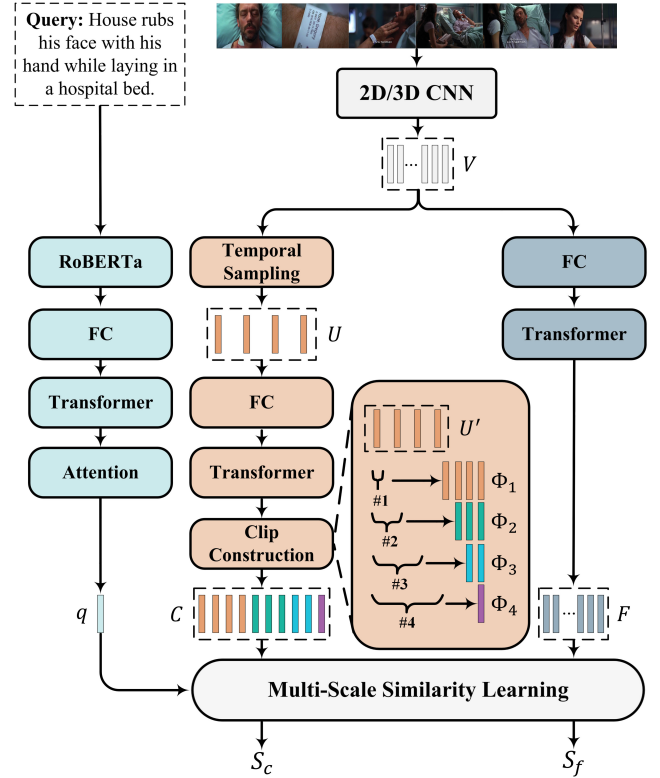

Figure 3: The framework of our proposed model for partially relevant video retrieval. #k denotes a temporal sliding window of size k with a stride of 1.

To build a PRVR model, a set of untrimmed videos are given for training, where each video is associated with multiple natural language sentences. Each sentence describes the content of a specific moment in the corresponding video. Note that we do not have access to the start/end time points of the moments (moment annotations) referred to by the sentences.

3.2 Sentence Representation

For sentence representation, we adopt the method by Lei *et al.* [29], considering its good performance on VCMR. Specifically, given a sentence consisting of n_q words, a pre-trained RoBERTa [38] is firstly employed to extract word features. Then a fully connected (FC) layer with a ReLU activation is utilized to map the word features into a lower-dimensional space. After adding the learned positional embedding to the mapped features, a standard Transformer layer [51] is further employed to obtain a sequence of d-dimensional contextualized word feature vectors $Q = \{q_i\}_{i=1}^{n_q} \in \mathbb{R}^{d \times n_q}$. In the Transformer, the features are fed to a multi-head attention layer followed by a feed-forward layer, and both layers are connected with residual connection [23] and layer normalization [2]. Finally, a sentence-level representation $q \in \mathbb{R}^d$ is obtained by employing a simple attention on Q:

$$q = \sum_{i=1}^{n_q} \alpha_i^q \times q_i, \quad \alpha^q = Softmax(w^\top Q), \tag{1}$$

where $Softmax$ denotes softmax layer, $w \in \mathbb{R}^{d \times 1}$ is trainable vector, and $\alpha^q \in \mathbb{R}^{1 \times n_q}$ indicates the attention vector.

3.3 Multi-Scale Video Representation

Given an untrimmed video, we first represent it by a sequence of d_v-dimensional feature vectors $V \in \mathbb{R}^{d_v \times n_v}$, where n_v denotes the number of the vectors. The feature sequence is obtained by extracting frame-level features using a pre-trained 2D CNN, or extracting segment-level features using a pre-trained 3D CNN. For the ease of description, we regard V as a sequence of frame-level features in the following. Based on V, we construct multi-scale video representation, jointly using a clip-scale feature learning branch and a frame-scale feature learning branch.

3.3.1 Clip-scale video representation. Before constructing video clips, we first downsample the input in the temporal domain to reduce the length of the feature sequence, which helps reduce the computational complexity of the model. Specifically, given a sequence of frame feature vectors V as the input, we downsample them into a fixed number of feature vectors, where each feature vector is obtained by mean pooling over the corresponding multiple consecutive frame features. Then, the video is described by a sequence of new feature vectors $U \in \mathbb{R}^{d_v \times n_u}$, where n_u indicates the number of the corresponding feature vectors. In order to make the features more compact, we employ an FC layer with a ReLU activation. Moreover, we also use standard Transformer [51] with a learned positional embedding to improve the temporal dependency of the features. Formally, through an FC layer and a one-layer Transformer, we obtain $U' \in \mathbb{R}^{d \times n_u}$:

$$U' = \{u_1, u_2, ..., u_{n_u}\} = Transformer(FC(U) + PE) \quad (2)$$

where PE denotes the output of the positional embedding. The reduced feature array U' is further used for clip-scale video representation learning.

For clip construction, we employ a multi-scale sliding window strategy to generate video clips, as illustrated in Fig. 3. Note that different from previous work [66] where video clips are of equal length and non-overlapping, our video clips are of varied lengths and overlapping. Concretely, we apply sliding windows of different sizes over U' along its temporal dimension with a stride of 1. Given a sliding window of size k, a clip feature is obtained by mean pooling over the features within the given window. The resultant feature sequence is denoted as Φ_k. Consequently, by jointly employing sliding windows of varied sizes as $\{1, 2, ..., n_u\}$, we are able to obtain $\{\Phi_1, \Phi_2, ..., \Phi_{n_u}\}$. Putting them together, a video can be represented a sequence of video clips $C \in \mathbb{R}^{d \times n_c}$:

$$C = \{\Phi_1, \Phi_2, ..., \Phi_{n_u}\} = \{c_1, c_2, ..., c_{n_c}\}, \quad (3)$$

where $c_i \in \mathbb{R}^d$ denotes the feature representation of i-th clip, n_c is the number of all generated clips which meets $n_c = n_u(n_u + 1)/2$.

3.3.2 Frame-scale video representation. As the initial frame features are extracted independently, they naturally lack temporal dependency. To bring such dependency back, we again utilize Transformers. Specifically, given the frame feature sequence V, we first
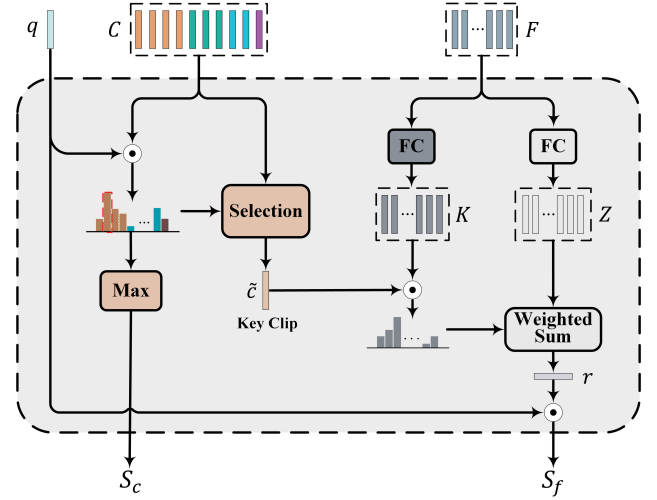

Figure 4: The illustration of multi-scale similarity learning.

utilize an FC layer with ReLU activation to reduce the dimensionality of the input, followed by a standard Transformer with a positional embedding layer. The re-encoded frame features, denoted $F \in \mathbb{R}^{d \times n_v}$, are computed as:

$$F = \{f_1, f_2, ..., f_{n_v}\} = Transformer(FC(V) + PE). \quad (4)$$

Note that the network structures of Transformer, FC and PE are the same as that in the clip-scale branch, but their trainable parameters are not shared. This allows each branch to learn parameters suitable for their own scale.

3.4 Multi-Scale Similarity Learning

As we have no priori about where the relevant content is localized in PRVR, it is challenging to directly compute video-text similarity on a fine-grained scale. Here, we propose multi-scale similarity learning, which computes the similarity in a coarse-to-fine manner. It first detects a key clip that is most likely to be relevant to the query. Then, the importance of each frame is measured in a fine-grained temporal scale under the guidance of the key clip. The final similarity is computed by jointly considering the query's similarities with the key clip and the frames. The hypothesis here is that if one model briefly knows coarse relevant content with respect to the query, it will help the model to find more relevant content on a more fine-grained scale accurately. The framework of the multi-scale similarity learning is illustrated in Fig. 4.

3.4.1 Clip-scale Similarity. Given a video as a sequence of video clips, we first measure the cross-modal similarity of each video clip with the query, and then aggregate the instance similarities to obtain the clip-scale similarity. Specifically, given a sequence of video clips $C = \{c_1, c_2, ..., c_{n_c}\}$, we use cosine similarity between each instance representation and the query representation, followed by a max-pooling operator on the similarities. More formally, the clip-scale similarity is obtained as:

$$S_c(v, q) = \max\{cos(c_1, q), cos(c_2, q), ..., cos(c_{n_c}, q)\}, \quad (5)$$

where $cos(\cdot)$ denotes the cosine similarity function. The max-pooling determines the clip with the highest similarity, and we then utilize

its similarity with the query as the whole video's similarity with the query. Besides, we select this clip as *key clip*, which is used for the latter frame-scale similarity learning.

3.4.2 Frame-scale Similarity. To obtain the frame-scale similarity, we first aggregate a sequence of frame feature vectors to a feature vector under the guidance of the key clip obtained in Section 3.4.1, and then compute its similarity with the query as the frame-scale similarity. Specifically, given a sequence of video frames $F = \{f_1, f_2, ..., f_{n_v}\}$, we devise a Key Clip Guided Attention (KCGA) to aggregate frame features. The implementation of KCGA borrows the idea of multi-head self-attention (MHSA) mechanism in Transformer [51]. MHSA first projects the input into queries, keys, values, and then computes the output as a weighted sum of the values. The weight assigned to each value is computed by a compatibility function of the query with the corresponding key. Different from MHSA utilizes the same input to construct queries, keys, and values, here we take the feature vector of the key clip as the query, and the video frame features as keys and values. Formally, the aggregated frame feature vector is obtained as:

$$r = Softmax(\tilde{c}^{\mathsf{T}}K)Z^{\mathsf{T}}, \quad K = W_k F, \quad Z = W_v F, \quad (6)$$

where $\tilde{c} \in \mathbb{R}^{d \times 1}$ indicates the feature vector of the key clip, $W_k \in \mathbb{R}^{d \times d}$ and $W_k \in \mathbb{R}^{d \times d}$ are two trainable projection matrices. The dot product measures the similarity between frames and the key clip, resulting in larger values for frames that are more similar to the key clip. Therefore, frames that are more similar to the key clip will have greater attention weights.

Finally, the frame-scale similarity is measured as the cosine similarity between the aggregated frame feature vector r and query feature vector q, namely:

$$S_f(v, q) = cos(r, q). \quad (7)$$

3.4.3 Similarity Learning. In this section, we first introduce the definition of positive and negative pairs for similarity learning. Inspired by MIL [9, 40], we define that a query and video pair is positive if the video contains certain content that is relevant to the query, and negative if no relevant content in the video.

Based on the above definition, we jointly use the triplet ranking loss [12, 15] and InfoNCE loss [41, 68] that are widely used in retrieve related tasks, and found them complementary. Given a positive video-query pair (v, q), the triplet ranking loss over the mini-batch $\mathcal{B}$ is defined as:

$$\mathcal{L}^{trip} = \frac{1}{n} \sum_{(q,v) \in \mathcal{B}} [max(0, m + S(q^-, v) - S(q, v)) \\ + max(0, m + S(q, v^-) - S(q, v))], \quad (8)$$

where m is the margin constant, $S(\cdot)$ denotes the similarity function which we can use the clip-scale similarity $S(\cdot)_c$ or the frame-scale similarity $S(\cdot)_f$. Besides, q^- and v^- respectively indicate a negative sentence sample for v and a negative video sample for s. The negative samples are randomly sampled from the mini-batch at the beginning of the training, while being the hardest negative samples after 20 epochs.

Given a positive video-query pair (v, q), the infoNCE loss over the mini-batch $\mathcal{B}$ is computed as:

$$\mathcal{L}^{nce} = -\frac{1}{n} \sum_{(q,v) \in \mathcal{B}} \left[log\left(\frac{S(q, v)}{S(q, v) + \sum_{q_i^- \in \mathcal{N}_q} S(q_i^-, v)} \right) \right. \\ \left. + log\left(\frac{S(q, v)}{S(q, v) + \sum_{v_i^- \in \mathcal{N}_v} S(q, v_i^-)} \right) \right], \quad (9)$$

where $\mathcal{N}_q$ denotes all negative queries of the video v in the mini-batch, while $\mathcal{N}_v$ denotes all negative videos of the query q in the mini-batch.

As the previous work [32] has concluded that using one loss per similarity function performs better than using one loss on the summation of multiple similarities, we employ the above two losses on both clip-scale similarity and frame-scale similarity, instead of their sum. Finally, our model is trained by minimizing the following overall training loss:

$$\mathcal{L} = \mathcal{L}_c^{trip} + \mathcal{L}_f^{trip} + \lambda_1 \mathcal{L}_c^{nce} + \lambda_2 \mathcal{L}_f^{nce}, \quad (10)$$

where $\mathcal{L}_c^{trip}$ and $\mathcal{L}_f^{trip}$ denote the triplet ranking loss using the clip-scale similarity $S(\cdot)_c$ and frame-scale similarity $S(\cdot)_f$ respectively, and accordingly for $\mathcal{L}_c^{nce}$ and $\mathcal{L}_f^{nce}$. λ_1 and λ_2 are hyper-parameters to balance the contribution of infoNCE loss.

3.5 Model Inference

After the model has been trained, the similarity between a video and a sentence query is computed as the sum of their clip-level similarity and frame-level similarity, namely:

$$S(v, s) = \alpha S_c(v, s) + (1 - \alpha)S_f(v, s) \quad (11)$$

where α is a hyper-parameter to balance the importance of two similarities, ranging within $[0, 1]$. Given a query, we sort all videos from the video gallery in descending order according to their similarity with respect to the given video.

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Datasets. In order to verify the viability of our proposed model for PRVR, queries that are partially relevant to videos are required. As videos in popular T2VR datasets such as MSR-VTT [59], MSVD [4] and VATEX [54] are supposed to be fully relevant to the queries, they are not suited for our experiments. Here, we re-purpose three datasets commonly used for VCMR, *i.e.,* TVR [29], Activitynet Captions [28], and Charades-STA [19], considering their natural language queries partially relevant with the corresponding videos (a query is typically associated with a specific moment in a video). Table 1 summarizes the brief statistics of these datasets, including average lengths of moments and videos, and the average moment length proportion in the whole video (moment-to-video ratio). Note that as we focus on retrieving videos, moment annotations provided by these datasets are not used in our proposed new PRVR task.

TV show Retrieval (TVR) [29] is a multimodal dataset originally for video corpus moment retrieval, where videos are paired with subtitles that are generated by automatic speech recognition.

Table 1: Brief statistics of three public datasets used in our experiments. The length is measured in seconds.

Datasets	Average length		Moment-to-video ratio		
	moments	videos	*min*	*max*	*mean*
TVR	9.1	76.2	0.48%	100%	11.9%
Activitynet Captions	36.2	117.6	0.48%	100%	30.8%
Charades-STA	8.1	30.0	4.3%	100%	26.3%

It contains 21.8K videos collected from 6 TV shows, and each video is associated with 5 natural language sentences that describe a specific moment in the video. As a moment is typically a part of a video, we assume that sentences are partially relevant to videos, and use them to evaluate our model. Following [66, 68], we utilize 17,435 videos with 87,175 moments for training and 2,179 videos with 10,895 moments for testing.

ActivityNet Captions [28] is originally developed for dense video captioning task, and is now a popular dataset for single video moment retrieval. It contains around 20K videos from Youtube, and the average length of videos is the largest among the three datasets we used. On average, each video has around 3.7 moments with corresponding sentence descriptions.We use the popular data partition used in [66, 68].

Charades-STA [19] is a dataset for single video moment retrieval. It contains 6,670 videos with 16,128 sentence descriptions. Each video has around 2.4 moments with corresponding sentence descriptions on average. We utilize the official data partition for model training and evaluation.

4.1.2 Evaluation Metrics. To evaluate PRVR models, we utilize the rank-based metrics, namely $R@K$ ($K = 1, 5, 10, 100$), which are commonly used for the conventional text-to-video retrieval [12, 53]. $R@K$ is the fraction of queries that correctly retrieve desired items in the top K of the ranking list. The performance is reported in percentage (%). Higher $R@K$ means better performance. For overall comparison, we also report the Sum of all Recalls (SumR).

4.1.3 Implementation Details. We use PyTorch as our deep learning environment, and we will release our source code. For video feature, on TVR, we utilize the feature provided by [29], that is, 3,072-D visual feature obtained by the concatenation of frame-level ResNet152 [23] feature and segment-level I3D [3] feature. For ease of reference, we refer to it as ResNet152-I3D. On ActivityNet-Captions and Charades-STA, we only utilize the same I3D feature, which are respectively provided by [66] and [43]. For sentence feature, we use the 768-D RoBERTa feature provided by [29] on TVR, where RoBERTa is finetuned on the queries and subtitle sentences of TVR. On ActivityNet-Captions and Charades-STA, we use the 1,024-D RoBERTa feature extracted by ourselves using the open RoBERTa toolkit[1]. Due to the limited space of the paper, we present more detailed implementation details in the supplementary material.

4.2 Comparison with Baseline Methods

4.2.1 Baseline selection. As models specifically designed for PRVR are non-existing, we compare with models targeted at conventional T2VR and models developed for VCMR. Given the rich literature,

[1]https://pytorch.org/hub/pytorch_fairseq_roberta/

Table 2: Performance of PRVR on the TVR dataset. Models are sorted in ascending order in terms of their overall performance. Visual feature: ResNet152-I3D.

Model	R@1	R@5	R@10	R100	SumR
T2VR models:					
W2VV, TMM18 [10]	2.6	5.6	7.5	20.6	36.3
HGR, CVPR20 [7]	1.7	4.9	8.3	35.2	50.1
HTM, ICCV19 [42]	3.8	12.0	19.1	63.2	98.2
CE, BMVC19 [37]	3.7	12.8	20.1	64.5	101.1
W2VV++, MM19 [31]	5.0	14.7	21.7	61.8	103.2
VSE++, BMVC19 [15]	7.5	19.9	27.7	66.0	121.1
DE, CVPR19 [11]	7.6	20.1	28.1	67.6	123.4
DE++, TPAMI21 [12]	8.8	21.9	30.2	67.4	128.3
RIVRL, TCSVT22 [13]	9.4	23.4	32.2	70.6	135.6
VCMR models w/o moment localization:					
XML, ECCV20 [29]	10.0	26.5	37.3	81.3	155.1
ReLoCLNet, SIGIR21[68]	10.7	28.1	38.1	80.3	157.1
Ours	**13.5**	**32.1**	**43.4**	**83.4**	**172.4**

we have to be selective, choosing *open-source* models for fair and reproducible comparison. In particular, we choose the following nine T2VR models, *i.e.,* VSE++ [15], W2VV [10], CE [37], W2VV++ [31], DE [11], HTM [42], HGR [7], DE++ [12] and RIVRL [13], and the following two VCMR models, *i.e.,* XML [29] and ReLoCLNet[68]. Both XML and ReLoCLNet are two-stage, where a first-stage module is used to retrieve candidate videos followed by a second-stage module to localize specific moments in the candidate videos. As moment annotations are unavailable for PRVR, we have re-trained XML and ReLoCLNet (with their moment localization modules removed) using the same video features as ours.

4.2.2 Results on the TVR dataset. Table 2 summarizes the performance comparison on TVR. Our proposed model consistently outperforms all conventional T2VR models with a clear margin. Even the best performing model RIVRL among the T2VR models, our model outperforms it by 36.8 in terms of SumR. As these models focus on the whole similarity between videos and queries, the results allow us to conclude that such similarity modeling is sub-optimal for PRVR. For the models of the second group, *i.e.,* ReLoCLNet and XML, they perform better than the conventional T2VR models, but they are still worse than ours. ReLoCLNet and XML focus on retrieving moment, which to some extent model the partial relevance, but they compute the similarity only in terms of a specific scale. By contrast, we compute the similarity in terms of both clip scale and frame scale. The results demonstrate the effectiveness of our proposed multi-scale similarity learning for PRVR. Note that when using the extra subtitle feature provided by [29], our model obtains better performance (R@1 of 24.0 and SumR of 220.8).

To gain a further understanding of the individual models, we define *moment-to-video ratio* (M/V) for query, which is measured by its corresponding moment's length ratio in the entire video. The smaller M/V indicates less relevant content while more irrelevant content with respect to the query. Besides, the smaller M/V to some extent means a lower relevance of a query to its corresponding
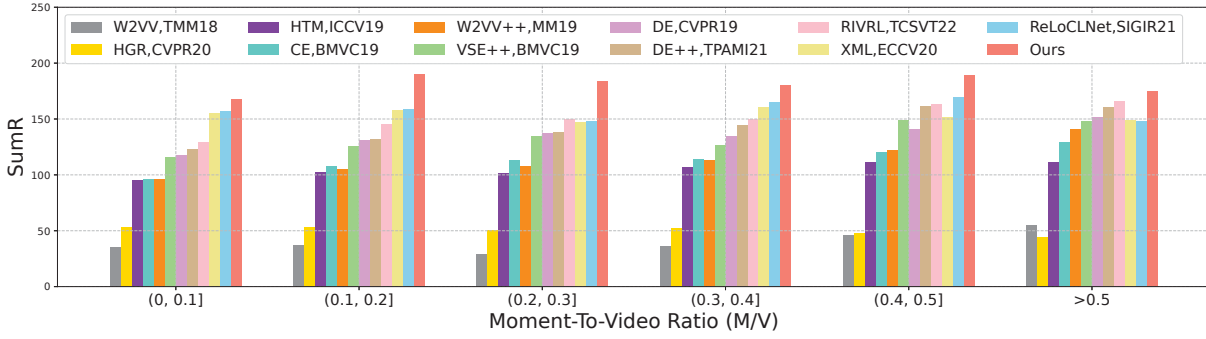
Figure 5: Performance of different models on different types of queries. Queries are grouped according to their M/V.

Table 3: Performance of PRVR on the ActivityNet Captions dataset. Visual feature: I3D.

Model	R@1	R@5	R@10	R100	SumR
T2VR models:					
W2VV [10]	2.2	9.5	16.6	45.5	73.8
HTM [42]	3.7	13.7	22.3	66.2	105.9
HGR [7]	4.0	15.0	24.8	63.2	107.0
RIVRL [13]	5.2	18.0	28.2	66.4	117.8
VSE++ [15]	4.9	17.7	28.2	67.1	117.9
DE++ [12]	5.3	18.4	29.2	68.0	121.0
DE [11]	5.6	18.8	29.4	67.8	121.7
W2VV++ [31]	5.4	18.7	29.7	68.8	122.6
CE [37]	5.5	19.1	29.9	71.1	125.6
VCMR models w/o moment localization:					
ReLoCLNet [68]	5.7	18.9	30.0	72.0	126.6
XML [29]	5.3	19.4	30.6	73.1	128.4
Ours	**7.1**	**22.5**	**34.7**	**75.8**	**140.1**

Table 4: Performance of PRVR on the Charades-STA dataset. Visual feature: I3D.

Model	R@1	R@5	R@10	R100	SumR
T2VR models:					
W2VV [10]	0.5	2.9	4.7	24.5	32.6
VSE++ [15]	0.8	3.9	7.2	31.7	43.6
W2VV++ [31]	0.9	3.5	6.6	34.3	45.3
HGR [7]	1.2	3.8	7.3	33.4	45.7
CE [37]	1.3	4.5	7.3	36.0	49.1
DE [11]	1.5	5.7	9.5	36.9	53.7
DE++ [12]	1.7	5.6	9.6	37.1	54.1
RIVRL[13]	1.6	5.6	9.4	37.7	54.3
HTM [42]	1.2	5.4	9.2	44.2	60.0
VCMR models w/o moment localization:					
ReLoCLNet [68]	1.2	5.4	10.0	45.6	62.3
XML [29]	1.6	6.0	10.1	46.9	64.6
Ours	**1.8**	**7.1**	**11.8**	**47.7**	**68.4**

video, while the larger one indicates a higher relevance. According to M/V, queries can be automatically classified into different groups, which enables a fine-grained analysis of how a specific model responds to the different types of queries. On TVR, the 10,895 test queries are split according to their M/V into six groups, with the performance of each group shown in Fig. 5.

Unsurprisingly, our model consistently performs the best in all groups. Observing the figure from left to right, the average performance of the twelve compared models increases along with the M/V, from 106.8, 114.2 114.3, 118.6, 125.8 and 127.7. The performance in the group with the lowest M/V is the smallest, while the group with the highest M/V is the largest. The result allows us to conclude that the current video retrieval baseline models better address queries of larger relevance to the corresponding video. By contrast, the performance we achieved is more balanced in all groups. This result shows that our proposed model is less sensitive to irrelevant content in videos.

4.2.3 Results on Activitynet Captions and Charades-STA. The performance of different models on Activitynet Captions and Charades-STA are summarized in Table 3 and Table 4, respectively. On both datasets, our model is still at the leading position.The results again verify the effectiveness of our model for measuring partial relevance between videos and queries. Interestingly, we observe that HTM performs badly on TVR and Activitynet Captions, while on Charades-STA it achieves the best SumR score among the T2VR models. We speculate it is due to the fact that Charades-STA has the least training data among the three datasets. Besides, the model structure of HTM is very simple, respectively using an FC layer with gating mechanism to embed videos and sentences into a common space, showing an advantage of training on small-scale data. For our proposed model, it consistently performs the best on the three datasets of the varying number of training samples, which to some extent shows that our model is not sensitive to the scale of training data.

4.3 Comparison on Model Complexity

Table 5 summarizes the model complexity comparison in terms of the time complexity and memory consumption. For a specific method, its time complexity is measured as FLOPs it takes to encode a given video-text pair. In terms of FLOPs, our model is at the mid-level, slightly slower than XML and ReLoCLNet, yet faster than RIVRL, DE and HGR. In terms of memory consumption, our model requires more memory than the majority of compared models, which is mainly due to the usage of transformer and multi-scale video representations. However, we found that our model takes about 0.2 seconds to retrieve videos from 20,000 candidate untrimmed videos, given that the video embeddings are pre-computed. The retrieval speed is adequate for instant response.

Table 5: Model comparison in terms of FLOPs and memory consumption.

	W2VV	HGR	HTM	CE	W2VV++	VSE++	DE	DE++	RIVRL	XML	ReLoCLNet	Ours
FLOPs (G)	0.42	2.96	0.06	0.06	0.4	0.20	5.24	5.30	8.64	0.80	0.96	1.22
Memory (MiB)	1231	8555	1225	1435	1281	1299	5837	3515	4809	2451	2673	5349

(a) IoU=0.5 (b) IoU=0.7
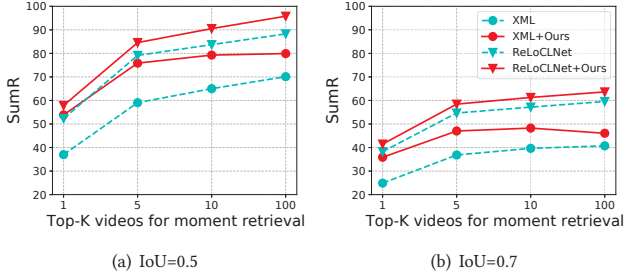
Figure 6: Performance of XML and ReLoCLNet without/with our model as the first stage for VCMR.

Table 6: Ablation study on the TVR dataset.

| Model	R@1	R@5	R@10	R100	SumR
Full setup	**13.5**	**32.1**	**43.4**	**83.4**	**172.4**
w/o frame-scale branch	12.3	30.5	41.5	82.3	166.6
w/o clip-scale branch	8.0	21.0	30.0	74.0	133.0
w/o key clip guide	12.2	30.6	41.0	82.4	166.3
w/o InfoNCE	11.3	29.1	40.1	81.3	161.8
w/o Triplet loss	11.2	29.2	40.4	81.9	162.6

4.4 PRVR for VCMR

Our PRVR model can also be used in the first stage of VCMR. To that end, we replace the first stage of two VCMR models, *i.e.,* XML [29] and ReLoCLNet [68], with our model. Both visual and subtitle features are used for video representation.

Fig. 6 shows the performance of the original models and the replaced ones on the TVR dataset. Here, we report *SumR*, the sum of R1/R5/R10/R100. Replacing the first stage with our model improves both XML and ReLoCLNet.

4.5 Ablation Studies

4.5.1 The effectiveness of multi-scale branches. To examine the usefulness of the multi-scale branches, we compare the counterpart without the clip-scale branch or the frame-scale branch. As shown in Table 6, removing any branch results in clear performance degeneration. The result not only demonstrates the effectiveness of the multi-scale solution, but also shows the complementary of the clip-scale and the frame-scale branches.

4.5.2 The effectiveness of key clip guided attention. Additionally, we also compare the model w/o key clip guide, which is implemented by replacing key clip guided attention with a simple attention. The simple attention is implemented as Eq. 1 without any guide. As Table 6 shows, our model with the full setup still performs better, which shows the importance of key clip guided attention for PRVR.
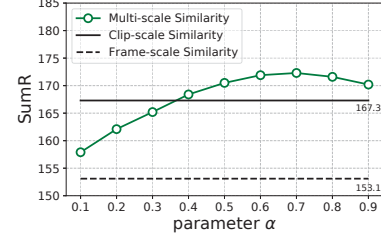
Figure 7: The influence of the hyper-parameter α in Eq. 11.

4.5.3 The effectiveness of the combination of triplet ranking loss and InfoNCE loss. To validate the choice of joint use of the two losses, we compare the results of using either triplet ranking loss or infoNCE loss. As shown in Table 6, triplet ranking loss and InfoNCE give comparable results when used alone, but they are much worse than the model with the full setup of jointly using both. The result demonstrates the benefit of using these two losses jointly.

4.5.4 The effect of α on the retrieval performance. The influence of the hyper-parameter α in Eq. 11 is studied as follows. We try α with its value ranging from 0.1 to 0.9 with an interval of 0.1. As shown in Fig. 7, when the α is larger than 0.3, the performance of using multi-scale similarity are all over 170, which consistently outperform the counterparts using the frame-scale or the clip-scale similarity alone.

5 CONCLUSIONS

In this paper, we have proposed a novel T2VR subtask termed PRVR. Different from the conventional T2VR where a query is usually fully relevant to the corresponding video, it is typically partially relevant in PRVR. Besides, videos in the conventional T2VR are temporally pre-trimmed with short durations, while videos are untrimmed in PRVR and a video is typically partially relevant to multiple sentences of different semantics. Additionally, PRVR differs from SVMR and VCMR, as the latter two are to retrieve moments rather than untrimmed videos. Towards PRVR, we have formulated it as a MIL problem, and propose MS-SL which computes the similarity on both clip scale and frame scale in a coarse-to-fine manner. Extensive experiments on three datasets have verified the effectiveness of MS-SL for PRVR, and have shown that it can also be used for improving VCMR.

REFERENCES

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision*. 5803–5812.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[3] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6299–6308.

[4] David Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 190–200.

[5] Jingyuan Chen, Xinpeng Chen, Lin Ma, Zequn Jie, and Tat-Seng Chua. 2018. Temporally grounding natural sentence in video. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 162–171.

[6] Long Chen, Chujie Lu, Siliang Tang, Jun Xiao, Dong Zhang, Chilie Tan, and Xiaolin Li. 2020. Rethinking the bottom-up framework for query-based video localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 10551–10558.

[7] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. 2020. Fine-grained Video-Text Retrieval with Hierarchical Graph Reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10638–10647.

[8] Ioana Croitoru, Simion-Vlad Bogolin, Yang Liu, Samuel Albanie, Marius Leordeanu, Hailin Jin, and Andrew Zisserman. 2021. TEACHTEXT: CrossModal Generalized Distillation for Text-Video Retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 11583-11593*.

[9] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence* 89, 1-2 (1997), 31–71.

[10] Jianfeng Dong, Xirong Li, and Cees GM Snoek. 2018. Predicting visual features from text for image and video caption retrieval. *IEEE Transactions on Multimedia* 20, 12 (2018), 3377–3388.

[11] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. 2019. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9346–9355.

[12] Jianfeng Dong, Xirong Li, Chaoxi Xu, Xun Yang, Gang Yang, Xun Wang, and Meng Wang. 2021. Dual encoding for video retrieval by text. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[13] Jianfeng Dong, Yabing Wang, Xianke Chen, Xiaoye Qu, Xirong Li, Yuan He, and Xun Wang. 2022. Reading-strategy Inspired Visual Representation Learning for Text-to-Video Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology* (2022).

[14] Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. 2019. Temporal localization of moments in video collections with natural language. *arXiv preprint arXiv:1907.12763* (2019).

[15] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. VSE++: Improving visual-semantic embeddings with hard negatives. In *Proceedings of the British Machine Vision Conference*. 935–943.

[16] Ji Feng and Zhi-Hua Zhou. 2017. Deep MIML network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31. 14747–14350.

[17] Zerun Feng, Zhimin Zeng, Caili Guo, and Zheng Li. 2021. Exploiting visual semantic reasoning for video-text retrieval. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 1005–1011.

[18] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision*. 214–229.

[19] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE International Conference on Computer Vision*. 5267–5275.

[20] Junyu Gao and Changsheng Xu. 2021. Fast video moment retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1523–1532.

[21] Simon Ging, Mohammadreza Zolfaghari, Hamed Pirsiavash, and Thomas Brox. 2020. COOT: Cooperative hierarchical transformer for video-text representation learning. *Advances in Neural Information Processing Systems* 33 (2020), 22605–22618.

[22] Ning Han, Jingjing Chen, Guangyi Xiao, Hao Zhang, Yawen Zeng, and Hao Chen. 2021. Fine-grained Cross-modal Alignment Network for Text-Video Retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3826–3834.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[24] Zhijian Hou, Chong-Wah Ngo, and Wing Kwong Chan. 2021. CONQUER: Contextual query-aware ranking for video corpus moment retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3900–3908.

[25] Fan Hu, Aozhu Chen, Ziyue Wang, Fangming Zhou, Jianfeng Dong, and Xirong Li. 2022. Lightweight Attentional Feature Fusion: A New Baseline for Text-to-Video Retrieval. In *Proceedings of the 17th European Conference on Computer Vision*.

[26] Maximilian Ilse, Jakub Tomczak, and Max Welling. 2018. Attention-based deep multiple instance learning. In *International Conference on Machine Learning*. 2127–2136.

[27] Weike Jin, Zhou Zhao, Pengcheng Zhang, Jieming Zhu, Xiuqiang He, and Yueting Zhuang. 2021. Hierarchical Cross-Modal Graph Consistency Learning for Video-Text Retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1114–1124.

[28] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision*. 706–715.

[29] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2020. TVR: A large-scale dataset for video-subtitle moment retrieval. In *European Conference on Computer Vision*. 447–463.

[30] Bin Li, Yin Li, and Kevin W Eliceiri. 2021. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14318–14328.

[31] Xirong Li, Chaoxi Xu, Gang Yang, Zhineng Chen, and Jianfeng Dong. 2019. W2VV++: Fully deep learning for ad-hoc video search. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1786–1794.

[32] Xirong Li, Fangming Zhou, Chaoxi Xu, Jiaqi Ji, and Gang Yang. 2020. SEA: Sentence encoder assembly for video retrieval by textual queries. *IEEE Transactions on Multimedia* 23 (2020), 4351–4362.

[33] Xirong Li, Yang Zhou, Jie Wang, Hailan Lin, Jianchun Zhao, Dayong Ding, Weihong Yu, and Youxin Chen. 2021. Multi-Modal Multi-Instance Learning for Retinal Disease Recognition. In *Proceedings of the 29th ACM International Conference on Multimedia*. 2474–2482.

[34] Daizong Liu, Xiaoye Qu, Jianfeng Dong, Pan Zhou, Yu Cheng, Wei Wei, Zichuan Xu, and Yulai Xie. 2021. Context-aware biaffine localizing network for temporal sentence grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11235–11244.

[35] Daizong Liu, Xiaoye Qu, Xiao-Yang Liu, Jianfeng Dong, Pan Zhou, and Zichuan Xu. 2020. Jointly cross-and self-modal graph attention network for query-based moment localization. In *Proceedings of the 28th ACM International Conference on Multimedia*. 4070–4078.

[36] Hongying Liu, Ruyi Luo, Fanhua Shang, Mantang Niu, and Yuanyuan Liu. 2021. Progressive Semantic Matching for Video-Text Retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*. 5083–5091.

[37] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. 2019. Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487* (2019).

[38] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[39] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2021. CLIP4clip: An empirical study of clip for end to end video clip retrieval. *arXiv preprint arXiv:2104.08860* (2021).

[40] Oded Maron and Tomás Lozano-Pérez. 1997. A framework for multiple-instance learning. *Advances in Neural Information Processing Systems* 10 (1997).

[41] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2020. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9879–9889.

[42] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2630–2640.

[43] Jonghwan Mun, Minsu Cho, and Bohyung Han. 2020. Local-global video-text interactions for temporal grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10810–10819.

[44] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. 2015. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 685–694.

[45] Sudipta Paul, Niluthpol Chowdhury Mithun, and Amit K Roy-Chowdhury. 2021. Text-based localization of moments in a video corpus. *IEEE Transactions on Image Processing* 30 (2021), 8886–8899.

[46] Pedro O Pinheiro and Ronan Collobert. 2015. From image-level to pixel-level labeling with convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1713–1721.

[47] Xiaoye Qu, Pengwei Tang, Zhikang Zou, Yu Cheng, Jianfeng Dong, Pan Zhou, and Zichuan Xu. 2020. Fine-grained iterative attention network for temporal language localization in videos. In *Proceedings of the 28th ACM International Conference on Multimedia*. 4280–4288.

[48] Xue Song, Jingjing Chen, Zuxuan Wu, and Yu-Gang Jiang. 2021. Spatial-temporal graphs for cross-modal text2video retrieval. *IEEE Transactions on Multimedia* (2021).

[49] Yale Song and Mohammad Soleymani. 2019. Polysemous visual-semantic embedding for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1979–1988.

[50] Ming Tu, Jing Huang, Xiaodong He, and Bowen Zhou. 2019. Multiple instance learning with graph neural networks. *arXiv preprint arXiv:1906.04881* (2019).

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017).

[52] Jingwen Wang, Lin Ma, and Wenhao Jiang. 2020. Temporally grounding language queries in videos by contextual boundary-aware prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 12168–12175.

[53] Wei Wang, Junyu Gao, Xiaoshan Yang, and Changsheng Xu. 2020. Learning Coarse-to-Fine Graph Neural Networks for Video-Text Retrieval. *IEEE Transactions on Multimedia* (2020).

[54] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. VATEX: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4581–4591.

[55] Yunxiao Wang, Meng Liu, Yinwei Wei, Zhiyong Cheng, Yinglong Wang, and Liqiang Nie. 2022. Siamese Alignment Network for Weakly Supervised Video Moment Retrieval. *IEEE Transactions on Multimedia* (2022).

[56] Zheng Wang, Jingjing Chen, and Yu-Gang Jiang. 2021. Visual co-occurrence alignment learning for weakly-supervised video moment retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*. 1459–1468.

[57] Peng Wu, Xiangteng He, Mingqian Tang, Yiliang Lv, and Jing Liu. 2021. HANet: Hierarchical Alignment Networks for Video-Text Retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3518–3527.

[58] Shaoning Xiao, Long Chen, Songyang Zhang, Wei Ji, Jian Shao, Lu Ye, and Jun Xiao. 2021. Boundary proposal network for two-stage natural language video localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 2986–2994.

[59] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5288–5296.

[60] Xun Yang, Jianfeng Dong, Yixin Cao, Xun Wang, Meng Wang, and Tat-Seng Chua. 2020. Tree-Augmented Cross-Modal Encoding for Complex-Query Video Retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research*

and Development in Information Retrieval. 1339–1348.

[61] Xun Yang, Fuli Feng, Wei Ji, Meng Wang, and Tat-Seng Chua. 2021. Deconfounded video moment retrieval with causal intervention. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1–10.

[62] Xun Yang, Shanshan Wang, Jian Dong, Jianfeng Dong, Meng Wang, and Tat-Seng Chua. 2022. Video moment retrieval with cross-modal neural architecture search. *IEEE Transactions on Image Processing* 31 (2022), 1204–1216.

[63] Youngjae Yu, Jongseok Kim, and Gunhee Kim. 2018. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European Conference on Computer Vision*. 471–487.

[64] Yitian Yuan, Lin Ma, Jingwen Wang, Wei Liu, and Wenwu Zhu. 2019. Semantic conditioned dynamic modulation for temporal sentence grounding in videos. *Advances in Neural Information Processing Systems* 32 (2019).

[65] Yitian Yuan, Tao Mei, and Wenwu Zhu. 2019. To find where you talk: Temporal sentence localization in video with attention based location regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9159–9166.

[66] Bowen Zhang, Hexiang Hu, Joonseok Lee, Ming Zhao, Sheide Chammas, Vihan Jain, Eugene Ie, and Fei Sha. 2020. A hierarchical multi-modal encoder for moment localization in video corpus. *arXiv preprint arXiv:2011.09046* (2020).

[67] Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. 2019. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1247–1257.

[68] Hao Zhang, Aixin Sun, Wei Jing, Guoshun Nan, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. 2021. Video corpus moment retrieval with contrastive learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 685–695.

[69] Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. 2020. Learning 2d temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 12870–12877.

[70] Qi Zheng, Jianfeng Dong, Xiaoye Qu, Xun Yang, Yabing Wang, Pan Zhou, Baolong Liu, and Xun Wang. 2022. Progressive localization networks for language-based moment localization. *ACM Transactions on Multimedia Computing, Communications, and Applications* (2022).

A APPENDIX

We report more experimental results and more technical details which are not included in the paper due to space limit:

- More comparisons on the TVR dataset, including extra ablation studies to explore the significance of the transformer module, comparing with models using extra subtitles and comparing with conventional T2VR models using clips (Section A.1).
- Performance comparison on pre-trimmed video datasets (Section A.2).
- Distribution of moment-to-video ratios on Charades-STA (Section A.3.1).
- More technical details of our method (Section A.3.2).

A.1 More Experiments on TVR

A.1.1 Ablations on the usage of the transformer. Among the compared methods on the TVR dataset, the top three ranked methods (ReLoCLNet, XML and RIVRL) use Transformer modules: ReLo-CLNet and XML utilize Transformers for text and video representation, while RIVRL uses a Transformer for video representation. So we conduct extra ablation study to explore whether is the usage of the Transformer module mainly contributes the significant performance improvement of our method. We replace all the three Transformer modules in our model with 1D-CNN, bi-GRU, bi-LSTM, respectively. As shown in the Table 7, our model in the Transformer-free setup obtains SumR of 161.6, 170.2 and 162.3, respectively [29] dataset. While their performance is worse than the counterpart using Transformer, they are still better than the best baseline, *i.e.,* , ReLoCLNet with SumR of 157.1.

A.1.2 Comparison with models using extra subtitles. As the TVR dataset is a multimodal dataset where each video is additionally associated with subtitle (dialogue) texts, in this experiment we compare with models using extra subtitles. Here, we do not compare with conventional T2VR models, as they do not support using extra subtitles. We compare with XML [29] and ReLoCLNet [68], and use the same 768-D subtitles features provided by [29] as extra video

Table 7: Ablations on the usage of the Transformer.

	R@1	R@5	R@10	R@100	SumR
ReLoCLNet (best baseline)	10.7	28.1	38.1	80.3	157.1
Ours (1D-CNN)	10.6	29.3	39.9	81.9	161.6
Ours (bi-LSTM)	11.2	29.0	40.2	81.9	162.3
Ours (bi-GRU)	12.4	31.6	42.9	83.3	170.2
Ours (Transformer)	13.5	32.1	43.4	83.4	172.4

Table 8: Performance of PRVR with using subtitle features. Visual feature:ResNet152-I3D, Subtitle feature:RoBERTa.

Method	R@1	R@5	R@10	R100	SumR
XML[29]	17.4	39.3	51.5	89.1	197.3
ReLoCLNet[68]	19.1	40.3	51.5	87.0	197.9
Ours	**24.0**	**47.8**	**58.8**	**90.2**	**220.8**

features. The results are shown in Table 8, and our proposed model again performs the best.

A.1.3 Comparison with conventional T2VR models using clips. As the conventional T2VR models are typically designed for retrieving video clips, in this experiment we employ conventional T2VR models using clips to explore their potential for PRVR. Specifically, during the inference, we first split videos into multiple clips, and then compute the similarity of each clip with the query. The maximum similarity is regarded as the final similarity between the video and the query.
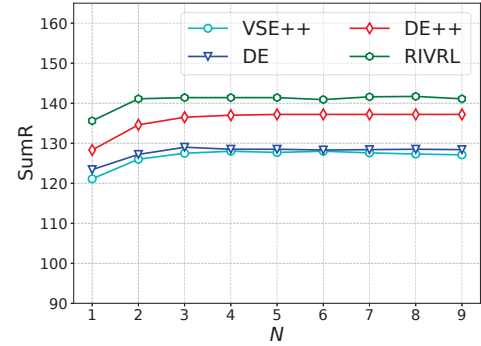
(a) Content-agnostic Strategy
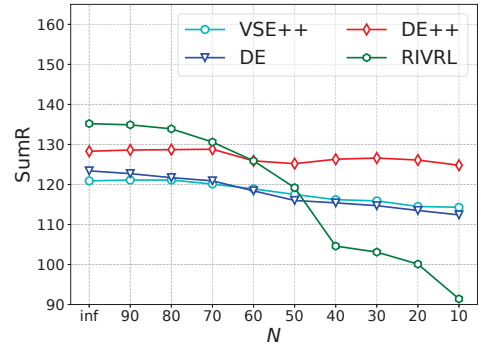
(b) Content-aware Strategy

Figure 8: Performance of four conventional T2VR models using clips generated by (a) content-agnostic strategy and (b) content-aware strategy on the TVR dataset. Their performance is still much worse than our proposed model which achieves a SumR score of 172.4.

Besides, we adopt two strategies to generate clips from video, *i.e.,* content-agnostic strategy and content-aware strategy. The content-agnostic strategy first splits the video into N video units evenly, then constructs video clips by using a specific video unit or concatenating adjacent video units. The larger N means generating more video clips, and $N = 1$ indicates using the whole video for inference. The content-aware strategy generates video clips by a scene detector toolkit[2] with an own provided threshold θ,

---
[2]https://github.com/Breakthrough/PySceneDetect/

which automatically splits the video into individual clips according to their content change. The smaller threshold means generating more video clips.

We conduct experiments with top-4 performing T2VR models on TVR, *i.e.,* VSE++, DE, DE++, RIVRL, and the results are shown in Fig. 8. As shown in 8 (a), all the models achieve performance gains when N is large than 1, and obtain the best performance when $N = 3$. Recall that $N = 1$ indicates using the whole video for inference. The results allow us to conclude that the T2VR models could be improved by splitting the video into multiple clips with the content-agnostic strategy. However, their performance is still much worse than our proposed model which achieves a SumR score of 172.4 on TVR. Fig. 8 (b) shows the results when the content-aware strategy is used. Note that $\theta = inf$ indicates using the whole video without splitting. We found that splitting the video into multiple clips by the content-aware strategy result in relative performance degeneration, which shows the scene detector is not suitable for PRVR. Besides, we speculate it is due to the content in a moment may has scene changes, and the scene detector is likely to split a moment into multiple parts.

Table 9: Performance comparison on the MSR-VTT and MSVD dataset. Visual feature:ResNeXt101+ResNet-152.

	R@1	R@5	R@10	SumR
On MSR-VTT:				
CE, BMVC19[37]	7.9	23.6	34.6	66.1
VSE++, BMVC19[15]	8.7	24.3	34.1	67.1
DE, CVPR19[11]	11.1	29.4	40.3	80.8
W2VV++, MM19[31]	11.1	29.6	40.5	81.2
DE++, TPAMI21[12]	11.6	30.3	41.3	83.2
HGR, CVPR20[7]	11.1	30.5	42.1	83.7
SEA, TMM21[32]	12.4	32.1	43.3	87.8
RIVRL, TCSVT22[13]	13.0	33.4	44.8	91.2
Ours	11.3	30.4	42.2	83.9
On MSVD:				
DE, CVPR19[11]	20.3	46.8	59.7	126.8
CF-GNN, TMM21[53]	22.8	50.9	63.6	137.3
W2VV++, MM19[31]	22.4	51.6	64.8	138.8
SEA, TMM21[32]	24.6	55.0	67.9	147.5
Ours	22.0	52.6	67.2	141.8

A.2 Results on Pre-trimmed Datasets

Although our proposed model is designed for untrimmed video, it can also be utilized for retrieving pre-trimmed by text. Therefore, we conduct experiments on MSR-VTT [59] and MSVD [4], two commonly used pre-trimmed datasets for T2VR. For MSR-VTT, we follow the official partition, where 6513 video clips for training, 497 video clips for validation and the remaining 2,990 video clips for testing. For MSVD, we also follow the official partition, 1200 video clips are used for training, 100 video clips for validation and 670 video clips for testing Following the previous works [12, 13, 31], we use the concatenation of 2048-dim ResNeXt-101 and 2048-dim

ResNet-152 features as the video feature. For text representation, we use the open RoBERTa toolkit to extract 1,024-D sentence feature.

The results are shown in the Table 9, where all the methods use the same video feature. Note that not all the compared methods report their performance on both datasets. As expected, our model is not on par with the state-of-the-art models on the two pre-trimmed datasets. Recall that the rationale for our proposed model is to first detect a key clip that is most likely to be relevant to the query and then measure the importance of other frames in a fine-grained temporal scale under the guidance of the key clip. As for pre-trimmed videos in MSR-VTT and MSVD, the majority of their frames are actually relevant w.r.t. the associated descriptions, making key clip detection unnecessary. Our method is thus suboptimal for text-to-video retrieval on MSR-VTT and MSVD.

A.3 Others

A.3.1 Distribution of Moment-to-Video Ratios. Fig. 9 shows the distribution of moment-to-video ratio on Charades-STA. Moment-to-video ratio indicates the moment's length ratio in the entire video. Moments on Charades-STA show a large variance in their temporal lengths.
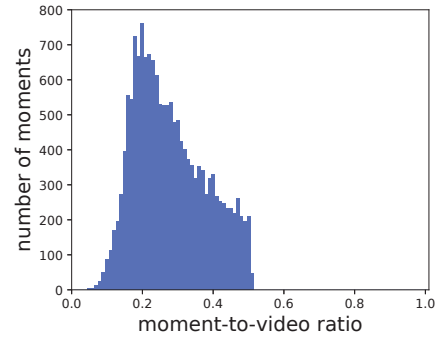

Figure 9: Distribution of moment-to-video ratio on Charades-STA. Moment-to-video ratio indicates the moment's length ratio in the entire video.

A.3.2 More Implementation Details. For the video representation module, we set the fixed number n_u to 32 in the downsampling strategy. Besides, we set the maxmium frame number n_v to 128. Once the number of frame is over n_v, it will be downsampled to n_v. For sentences, we set the maximum length of query n_q to 30 on TVR and Charades-STA, 64 on ActivityNet Captions, and the words outside the maximum length are simply discarded. For the Transformer module used in our model, we set its hidden size $d = 384$, and 4 attention heads are employed. For hyper-parameters in the loss functions, we empirically set λ_1=0.02 and λ_2=0.04 to make all loss elements have a similar loss value at the beginning of the model training. For model training, we utilize an Adam optimizer with a mini-batch size of 128. The initial learning rate is set to 0.00025, and we take a learning rate adjustment schedule similar to [29]. Early stop occurs if the validation performance does not improve in ten consecutive epochs. The maximal number of epochs is set to 100. Note that we will release our source code and data.