# GMMFormer: Gaussian-Mixture-Model Based Transformer for Efficient Partially Relevant Video Retrieval

**Yuting Wang**[1,3], **Jinpeng Wang**[1,3], **Bin Chen**[2,3*], **Ziyun Zeng**[1,3], **Shu-Tao Xia**[1,3]

[1] Tsinghua Shenzhen International Graduate School, Tsinghua University
[2] Harbin Institute of Technology, Shenzhen
[3] Research Center of Artificial Intelligence, Peng Cheng Laboratory
{wangyt22, wjp20, zengzy21}@mails.tsinghua.edu.cn, chenbin2021@hit.edu.cn, xiast@sz.tsinghua.edu.cn

## Abstract

Given a text query, partially relevant video retrieval (PRVR) seeks to find untrimmed videos containing pertinent moments in a database. For PRVR, clip modeling is essential to capture the partial relationship between texts and videos. Current PRVR methods adopt scanning-based clip construction to achieve explicit clip modeling, which is information-redundant and requires a large storage overhead. To solve the efficiency problem of PRVR methods, this paper proposes GMMFormer, a **G**aussian-**M**ixture-**M**odel based Trans**former** which models clip representations implicitly. During frame interactions, we incorporate Gaussian-Mixture-Model constraints to focus each frame on its adjacent frames instead of the whole video. Then generated representations will contain multi-scale clip information, achieving implicit clip modeling. In addition, PRVR methods ignore semantic differences between text queries relevant to the same video, leading to a sparse embedding space. We propose a query diverse loss to distinguish these text queries, making the embedding space more intensive and contain more semantic information. Extensive experiments on three large-scale video datasets (*i.e.*, TVR, ActivityNet Captions, and Charades-STA) demonstrate the superiority and efficiency of GMMFormer. Code is available at https://github.com/huangmozhi9527/GMMFormer.

## 1 Introduction

With the development of society, video has become the subject of information dissemination. As a result, text-to-video retrieval (T2VR) (Dong, Li, and Snoek 2018; Chen et al. 2020; Miech et al. 2019; Liu et al. 2019a; Li et al. 2019; Faghri et al. 2017; Dong et al. 2019, 2021, 2022b) has received increasing attention from academia and industry. Given a text query, T2VR aims to retrieve semantically relevant videos from a video database. However, videos in T2VR datasets are pre-trimmed to be entirely relevant to corresponding text queries, which exists a gap from the real world. In realistic social media or video platforms (*e.g.*, YouTube), a video is usually long-time and contains several moments, among which only one moment is entirely relevant to the corresponding text query. When handling these untrimmed videos, T2VR models trained on pre-
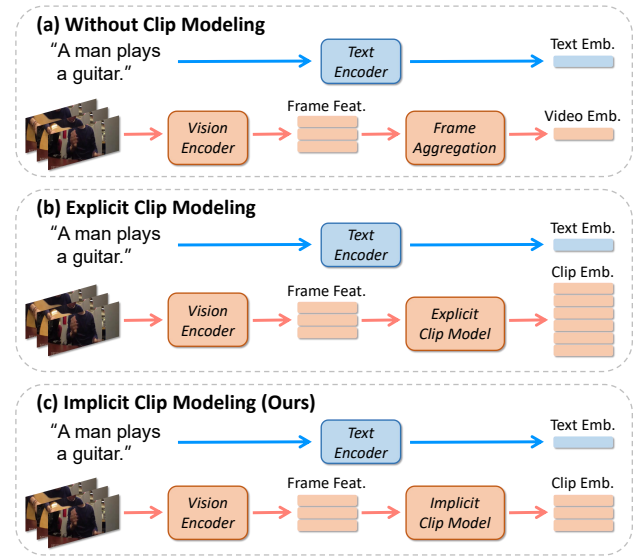


Figure 1: Traditional text-to-video retrieval pipelines (a) generate compact video embeddings and lost clip information. Previous partially relevant video retrieval pipelines (b) adopt explicit clip modeling, which is information-redundant and requires a large storage overhead. We utilize implicit clip modeling (c) to obtain compact clip embeddings, containing multi-scale clip information.

trimmed video datasets may not perform well, resulting in poor user experience. To overcome the above-mentioned problem, (Dong et al. 2022a) proposed the partially relevant video retrieval (PRVR) task, which collects untrimmed videos to form the video database. In particular, a video in PRVR corresponds to several text queries, and a text query is only relevant to one moment within the video. Compared to T2VR, PRVR is more aligned with the natural world and has more research significance.

Given a text-video pair, previous PRVR methods employ pre-trained vision-language models to extract frame and word features. These features will pass through sequential models (*e.g.*, RNN, LSTM, Transformer (Vaswani et al. 2017), etc.) to model global sequential interactions, generating frame and sentence embeddings. After that, they model

---

clip representations to capture the partial relationship between the text and video. Specifically, a multi-scale sliding window strategy is applied to frame embeddings to construct clip embeddings. Finally, the text-video similarity can derive from similarities between sentence embeddings with clip and frame embeddings.

Those PRVR methods have outperformed T2VR methods on untrimmed video datasets. However, their retrieval pipelines still suffer from two problems. **1) Global frame interactions confuse different moments of untrimmed videos.** An untrimmed video contains several moments. These moments correspond to different text queries, which the PRVR model should distinguish. However, we find that global frame interactions will make frame embeddings similar to each other. With these similar embeddings, the model cannot locate the correct time period of the given text query, resulting in poor performance. **2) Explicit clip modeling by scanning-based clip construction is information-redundant and requires a large storage overhead.** The multi-scale sliding window strategy will traverse all possible clips, generating a lot of irrelevant clip embeddings and leading to information redundancy. With frame embeddings of length $M$, the generated clip embeddings will have a length of $M(M+1)/2$. For instance, the past SOTA PRVR method MS-SL (Dong et al. 2022a) downsamples frame features to 32-length and constructs 528-length clip embeddings, within which only five clips are relevant to corresponding text descriptions on the TVR dataset. Although these redundant clip embeddings make the model localize the time period more accurately, they require a large storage overhead and reduce retrieval efficiency.

To solve the above-mentioned two problems, in this paper, we propose GMMFormer, a Gaussian-Mixture-Model based Transformer to model clip representations implicitly. Our motivation lies in a natural characteristic: moments in a video are successive and have limited duration, within which each frame should pay more attention to its neighboring frames; the closer it is, the more attention should be paid. Inspired by (Fu et al. 2022; Qu et al. 2020; Zhou, Yu, and Yang 2023; Kim, El-Khamy, and Lee 2020), we design a GMMFormer block to incorporate Gaussian-Mixture-Model constraints during frame interactions to focus each frame on its adjacent frames. In particular, we utilize multi-scale Gaussian windows to model frame interactions of different ranges, generating clip features with several receptive fields. Then we aggregate these features to obtain clip embeddings. These clip embeddings contain multi-scale clip information and can perceive video moments with different lengths. The comparison of different retrieval pipelines is illustrated in Figure 1.

For a video in PRVR, its relevant text queries are semantically diverse. However, the commonly used triplet ranking loss (Dong et al. 2021; Faghri et al. 2017) and infoNCE loss (Miech et al. 2020; Zhang et al. 2021) treat them equally and pull them together in the embedding space. These losses disturb the semantic structure of text representations, thus resulting in a sparse distribution in the embedding space. In this paper, we propose a query diverse loss to distinguish text queries relevant to the same video. Inspired by (Wang and Isola 2020), given an untrimmed video, we push its relevant text queries away from each other, generating discriminative sentence embeddings. Then the embedding space will be more intensive and contain more semantic information.

We conducted extensive experiments on three large-scale video datasets TVR (Lei et al. 2020), ActivityNet Captions (Krishna et al. 2017), and Charades-STA (Gao et al. 2017). The experimental results demonstrate the superiority and efficiency of our GMMFormer. In particular, GMMFormer achieves state-of-the-art results on three datasets. And compared to the past SOTA MS-SL, GMMFormer is about 2.5 times faster and the storage overhead is 20 times smaller.

Overall, our main contributions are as follows:

- We propose GMMFormer, a Gaussian-Mixture-Model based Transformer to model clip representations implicitly. GMMFormer is effective for its multi-scale Gaussian constraints and efficient for its compact clip embeddings with high information density.

- We propose a query diverse loss to distinguish different text queries relevant to the same video, preserving the semantic structure of text representations.

- Extensive experiments and ablation studies on three large-scale datasets (*i.e.*, TVR, ActivityNet Captions, Charades-STA) demonstrate the superiority and efficiency of our GMMFormer.

## 2   Related Work

**Text-to-video Retrieval.** Video analysis (Wang et al. 2023, 2022; Zeng et al. 2022; Liu et al. 2023b,a; Jin et al. 2022) has recently gained much attention due to the increasing video data on the Internet. Among them, the text-to-video retrieval (T2VR) task (Dong, Li, and Snoek 2018; Chen et al. 2020; Li et al. 2019; Faghri et al. 2017; Gao et al. 2023; Lei, Berg, and Bansal 2021; Li et al. 2023) aims to retrieve relevant videos from a set of pre-trimmed video clips given a text description. A standard pipeline for T2VR is to first encode videos and texts to obtain video and sentence representations, and then map them into a common embedding space to measure the cross-modal similarity.

**Partially Relevant Video Retrieval.** The partially relevant video retrieval (PRVR) task (Dong et al. 2022a) aims to retrieve untrimmed videos partially relevant to a given query, which is more in line with the real world than T2VR. For PRVR, clip modeling is crucial in capturing the partial relationship between texts and videos. Previous PRVR methods adopt clip construction to achieve explicit clip modeling. They apply a multi-scale sliding window strategy on frame embeddings to obtain clip embeddings. This practice will traverse all possible clips and generate a lot of irrelevant clip embeddings, requiring a large storage overhead and reducing retrieval efficiency. Besides, PRVR models are easy to overfit, which might be improved by adversarial training (Gao et al. 2023; Bai et al. 2021, 2020; Gudibande et al. 2022). In this paper, we propose GMMFormer, a Gaussian-Mixture-Model based Transformer to model clip representations implicitly. GMMFormer can generate compact clip embeddings with high information density, which is effective and efficient.

**Video Corpus Moment Retrieval.** The video corpus moment retrieval (VCMR) task (Song et al. 2021; Lei et al. 2020) seeks to retrieve moments semantically relevant to a given query from a collection of untrimmed videos. VCMR methods adopt a two-stage pipeline. They retrieve several candidate videos in the first stage, which may contain the target moment, then retrieve moments from the candidate videos in the second stage. VCMR's first stage is similar to PRVR. However, VCMR requires moment-level annotations, which is time-consuming and labor-intensive.

## 3 Methodology

We explain in detail our approach for PRVR. We start with the formulation of PRVR in Section 3.1, then elaborate on the overview of GMMFormer in Section 3.2. Next, we introduce our designed GMMFormer block in Section 3.3 and the learning strategy in Section 3.4.

### 3.1 Problem Formulation

Given a text query, partially relevant video retrieval (PRVR) aims to retrieve videos containing a moment semantically relevant to the given query, from a large corpus of untrimmed videos. Each video in PRVR databases has several moments and is associated with multiple text descriptions, while each text description represents the content of a specific moment in the corresponding video. It is worth mentioning that the start or end time points of moments are unavailable in PRVR.

### 3.2 Overview

In this section, we introduce the overall framework of our GMMFormer, including sentence representation encoding, video representation encoding and similarity measure, as shown in Figure 2.

**Sentence Representation.** Given a sentence containing $N$ words, we first utilize a pre-trained RoBERTa (Liu et al. 2019b) to extract word features. Then we adopt a FC layer with a ReLU activation to embed the word features into a lower-dimensional space. After adding the learnable positional embedding to the mapped features, we employ a vanilla Transformer layer to obtain a sequence of $d$-dimensional contextualized word feature vectors $Q = \{q_i\}_{i=1}^N \in \mathbb{R}^{N \times d}$. It is worth mentioning that we do not use the GMMFormer block here, which is designed for untrimmed videos. Finally, we use a simple attention module on $Q$ to obtain sentence embeddings $q \in \mathbb{R}^d$:

$$q = \sum_{i=1}^N a_i^q \times q_i, a^q = softmax(wQ^T) \qquad (1)$$

where $w \in \mathbb{R}^{1 \times d}$ is a trainable vector and $a^q \in \mathbb{R}^{1 \times N}$ indicates the attention vector.

**Video Representation.** Given an untrimmed video containing $M_f$ frames, we first employ a pre-trained 2D or 3D CNN to extract frame features. Then we pass them through two branches to obtain clip and video embeddings. Clip embeddings help model to locate relevant moments, while video embeddings measure the global text-video similarity.
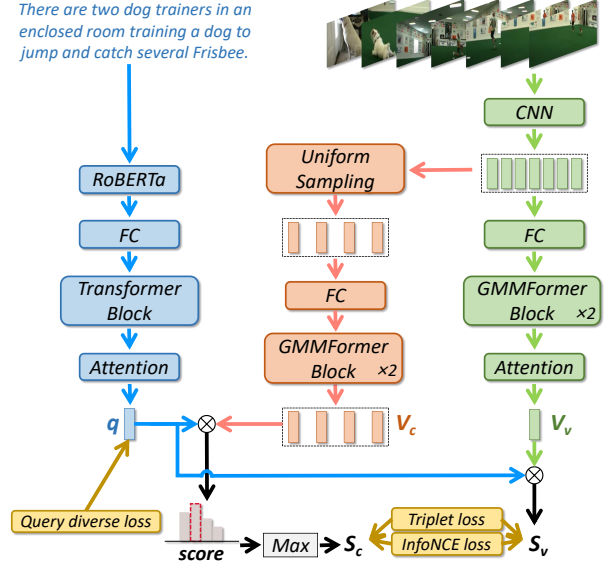


Figure 2: The overall framework of GMMFormer. $\otimes$ denotes the matrix multiplication.

In the clip-level branch, we uniformly sample a fixed number of feature vectors by mean pooling over the corresponding multiple consecutive frame features. Then we use a FC layer with a ReLU activation to reduce dimension, obtaining clip features. Finally, we use two GMMFormer blocks with the learnable positional embedding on clip features to get clip embeddings $V_c = \{c_i\}_{i=1}^{M_c} \in \mathbb{R}^{M_c \times d}$, where $M_c$ is the sampled number and $d$ is the dimension.

In the video-level branch, similarly, we first use a FC layer with a ReLU activation to reduce dimension, then employ two GMMFormer layers with the learnable positional embedding to obtain contextualized features $V_f = \{v_i\}_{i=1}^{M_f} \in \mathbb{R}^{M_f \times d}$. Finally, we employ a simple attention module on $V_f$ to obtain video embeddings $V_v \in \mathbb{R}^d$:

$$V_v = \sum_{i=1}^{M_f} a_i^f \times v_i, a^f = softmax(wV_f^T) \qquad (2)$$

where $w \in \mathbb{R}^{1 \times d}$ is a trainable vector and $a^f \in \mathbb{R}^{1 \times M_f}$ indicates the attention vector.

**Similarity Measure.** Given a text-video pair, we first compute the above-mentioned $q, V_c, V_v$, then the video-level similarity is measured as the cosine similarity between sentence embeddings $q$ and video embeddings $V_v$:

$$S_v(t, v) = cos(q, V_v) \qquad (3)$$

Besides, we use the cosine similarity and max-pooling operation to calculate the clip-level similarity between sentence embeddings $q$ and clip embeddings $V_c$:

$$S_c(t, v) = max\{cos(q, c_1), ..., cos(q, c_{M_c})\} \qquad (4)$$

The similarity of the text-video pair can be computed as the weighted sum of the video-level similarity and clip-level similarity:

$$S(t, v) = \alpha_v S_v(t, v) + \alpha_c S_c(t, v) \qquad (5)$$

where $\alpha_v, \alpha_c \in [0,1]$ are hyper-parameters to balance two similarities, and $\alpha_v + \alpha_c = 1$.

## 3.3 GMMFormer Block

To model the Gaussian-Mixture-Model distribution of video representations, we first propose a Gaussian block to incorporate a Gaussian constraint during frame interactions. Then we employ multi-scale Gaussian blocks in parallel and aggregate their output, making it a Gaussian-Mixture-Model constraint, as shown in Figure 3.

Given $M$ extracted features, we present it in a matrix form $X_i \in \mathbb{R}^{M \times d}$, where $d$ is the feature dimension and $i$ is the video index. In our designed Gaussian block, we project the input matrix $X_i$ to three matrices query, key and value via three learnable parameters $W^q$, $W^k$ and $W^v$. We use the query matrix to perform scaled dot-product attention over the key matrix, obtaining an attention score matrix. Then we design a Gaussian matrix $W^g \in \mathbb{R}^{M \times M}$ composed of $M$ Gaussian windows to perform element-wise product over the attention score matrix. After that, we put the generation through a softmax function to determine attentional distributions over the value matrix. The resulting weight-averaged value matrix forms the output of the Gaussian attention module in the Gaussian block:

$$X_i^{attn} = softmax(W^g \odot \frac{X_i W^q (X_i W^k)^T}{\sqrt{d_k}}) X_i W^v \quad (6)$$

$$W^g(i,j) = \frac{1}{2\pi} e^{-\frac{(j-i)^2}{\sigma^2}} \quad (7)$$

where $d_k$ is the dimension of queries and keys, $\sigma^2$ is the variance of the Gaussian density distribution and $\odot$ indicates the element-wise product function.

After the Gaussian attention module, we feed $X_i^{attn}$ to a Feed-Forward Network (FFN) to obtain Gaussian block output $X_i^{output}$. Similar to the vanilla Transformer block, we add residual connection (He et al. 2016) and Layer Normalization (Ba, Kiros, and Hinton 2016) in the Gaussian attention module and the FFN module. So the Gaussian block can be formulated as:

$$X_i^{output} = FFN(LayerNorm(X_i^{inter})) + X_i^{inter} \quad (8)$$

$$X_i^{inter} = GauAttn(LayerNorm(X_i)) + X_i \quad (9)$$

where $GauAttn$ indicates the Gaussian attention module, and FFN is composed of two fully connected (FC) layers.

Gaussian block output will contain fixed-length clip information. However, video moments are diverse in length. So we employ multi-scale Gaussian blocks in parallel and aggregate their output. Here, we use average pooling to achieve aggregation:

$$X_i^{GMM} = \frac{1}{K} \sum_{k=1}^{K} GB(X_i, \sigma_k^2) \quad (10)$$

where $GB(X_i, \sigma_k^2)$ is a Gaussian block with the variance $\sigma_k^2$ and $K$ is the number of Gaussian blocks. Specifically, we set $K = 4$ and choose Gaussian blocks respectively with low, medium, high, and infinite variance. $X_i^{GMM}$ denotes the output of the GMMFormer block, which maintains the length of $M$ and contains multi-scale clip information.
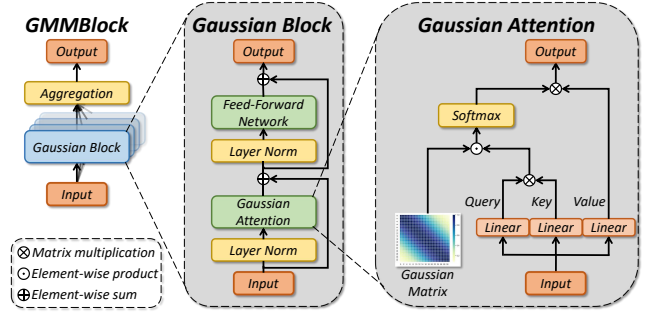


Figure 3: The illustration of the GMMFormer block (GMM-Block).

## 3.4 Learning

We consider a text-video pair positive if the video contains a moment relevant to the text and negative if there is no relevant content. We adopt triplet ranking loss (Dong et al. 2021; Faghri et al. 2017) and infoNCE loss (Miech et al. 2020; Zhang et al. 2021) that are widely used in the retrieval task.

Given a positive text-video pair $(t, v)$, the triplet ranking loss over the mini-batch $\mathcal{B}$ is defined as:

$$\mathcal{L}^{trip} = \frac{1}{n} \sum_{(t,v) \in \mathcal{B}} \{max(0, m + S(t^-, v) - S(t, v)) \\ + max(0, m + S(t, v^-) - S(t, v))\} \quad (11)$$

where $m$ is a margin constant, $t^-$ and $v^-$ indicate a negative text for $v$ and a negative video for $t$. Similar to (Dong et al. 2022a), we randomly sample the negative samples from the mini-batch at the beginning of the training and choose the hardest negative samples after 20 epochs.

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

$$\mathcal{L}^{nce} = -\frac{1}{n} \sum_{(t,v) \in \mathcal{B}} \{log(\frac{S(t,v)}{S(t,v) + \sum_{t_i^- \in \mathcal{N}_t} S(t_i^-, v)}) \\ + log(\frac{S(t,v)}{S(t,v) + \sum_{v_i^- \in \mathcal{N}_v} S(t, v_i^-)})\} \quad (12)$$

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

Besides, given a collection of texts $T$ in a mini-batch, we design a query diverse loss to distinguish different text queries relevant to the same video, defined as:

$$\mathcal{L}^{div} = \frac{1}{n} \sum_{t_i, t_j \in T} \mathbb{1}_{t_i, t_j} log(1 + e^{\alpha(cos(t_i, t_j) + \delta)}) \quad (13)$$

where $\delta > 0$ is a margin, $\alpha > 0$ is a scaling factor and $\mathbb{1}_{t_i, t_j} \in \{0, 1\}$ is an indicator function. $\mathbb{1}_{t_i, t_j} = 1$ when $t_i$ and $t_j$ are relevant to the same video.

$\mathcal{L}^{div}$ will push away semantically diverse texts relevant to the same video, preserving the semantic structure of text representations. Then the embedding space will be more intensive and contain more semantic information.

| Model | R@1 | R@5 | R@10 | R@100 | SumR |
|---|---|---|---|---|---|
| *T2VR models:* | | | | | |
| W2VV | 2.6 | 5.6 | 7.5 | 20.6 | 36.3 |
| HGR | 1.7 | 4.9 | 8.3 | 35.2 | 50.1 |
| HTM | 3.8 | 12.0 | 19.1 | 63.2 | 98.2 |
| CE | 3.7 | 12.8 | 20.1 | 64.5 | 101.1 |
| DE++ | 8.8 | 21.9 | 30.2 | 67.4 | 128.3 |
| RIVRL | 9.4 | 23.4 | 32.2 | 70.6 | 135.6 |
| CLIP4Clip | 9.9 | 24.3 | 34.3 | 72.5 | 141.0 |
| Cap4Video | 10.3 | 26.4 | 36.8 | 74.0 | 147.5 |
| *VCMR models w/o moment localization:* | | | | | |
| XML | 10.0 | 26.5 | 37.3 | 81.3 | 155.1 |
| ReLoCLNet | 10.7 | 28.1 | 38.1 | 80.3 | 157.1 |
| CONQUER | 11.0 | 28.9 | 39.6 | 81.3 | 160.8 |
| *PRVR models:* | | | | | |
| MS-SL | 13.5 | 32.1 | 43.4 | 83.4 | 172.4 |
| GMMFormer | **13.9** | **33.3** | **44.5** | **84.9** | **176.6** |

Table 1: Performance of various models on the TVR dataset. Models are sorted in ascending order in terms of their SumR.

| Model | R@1 | R@5 | R@10 | R@100 | SumR |
|---|---|---|---|---|---|
| *T2VR models:* | | | | | |
| W2VV | 2.2 | 9.5 | 16.6 | 45.5 | 73.8 |
| HTM | 3.7 | 13.7 | 22.3 | 66.2 | 105.9 |
| HGR | 4.0 | 15.0 | 24.8 | 63.2 | 107.0 |
| RIVRL | 5.2 | 18.0 | 28.2 | 66.4 | 117.8 |
| DE++ | 5.3 | 18.4 | 29.2 | 68.0 | 121.0 |
| CE | 5.5 | 19.1 | 29.9 | 71.1 | 125.6 |
| CLIP4Clip | 5.9 | 19.3 | 30.4 | 71.6 | 127.3 |
| Cap4Video | 6.3 | 20.4 | 30.9 | 72.6 | 130.2 |
| *VCMR models w/o moment localization:* | | | | | |
| ReLoCLNet | 5.7 | 18.9 | 30.0 | 72.0 | 126.6 |
| XML | 5.3 | 19.4 | 30.6 | 73.1 | 128.4 |
| CONQUER | 6.5 | 20.4 | 31.8 | 74.3 | 133.1 |
| *PRVR models:* | | | | | |
| MS-SL | 7.1 | 22.5 | 34.7 | 75.8 | 140.1 |
| GMMFormer | **8.3** | **24.9** | **36.7** | **76.1** | **146.0** |

Table 2: Performance of various models on the ActivityNet Captions dataset.

Finally, our model is trained by minimizing the following overall training loss:

$$\mathcal{L} = \mathcal{L}_c^{trip} + \mathcal{L}_v^{trip} + \lambda_1 \mathcal{L}_c^{nce} + \lambda_2 \mathcal{L}_v^{nce} + \lambda_3 \mathcal{L}^{div} \quad (14)$$

where $\mathcal{L}_c^{trip}$ and $\mathcal{L}_v^{trip}$ denote the triplet ranking losses using the clip-level similarity $S_c$ and video-level similarity $S_v$, and accordingly for $\mathcal{L}_c^{nce}$ and $\mathcal{L}_v^{nce}$. $\lambda_1$, $\lambda_2$ and $\lambda_3$ are hyperparameters to balance corresponding losses.

# 4 Experiments

## 4.1 Experimental Setup

**Datasets.** We evaluate our GMMFormer on three large-scale video datasets (*i.e.*, TV show Retrieval (TVR) (Lei et al. 2020), ActivityNet Captions (Krishna et al. 2017), and Charades-STA (Gao et al. 2017)). Note that moment annotations provided by these datasets are unavailable in the PRVR task. **TVR** contains 21.8K videos collected from 6

| Model | R@1 | R@5 | R@10 | R@100 | SumR |
|---|---|---|---|---|---|
| *T2VR models:* | | | | | |
| W2VV | 0.5 | 2.9 | 4.7 | 24.5 | 32.6 |
| HGR | 1.2 | 3.8 | 7.3 | 33.4 | 45.7 |
| CE | 1.3 | 4.5 | 7.3 | 36.0 | 49.1 |
| DE++ | 1.7 | 5.6 | 9.6 | 37.1 | 54.1 |
| RIVRL | 1.6 | 5.6 | 9.4 | 37.7 | 54.3 |
| HTM | 1.2 | 5.4 | 9.2 | 44.2 | 60.0 |
| CLIP4Clip | 1.8 | 6.5 | 10.9 | 44.2 | 63.4 |
| Cap4Video | 1.9 | 6.7 | 11.3 | 45.0 | 65.0 |
| *VCMR models w/o moment localization:* | | | | | |
| ReLoCLNet | 1.2 | 5.4 | 10.0 | 45.6 | 62.3 |
| XML | 1.6 | 6.0 | 10.1 | 46.9 | 64.6 |
| CONQUER | 1.8 | 6.3 | 10.3 | 47.5 | 66.0 |
| *PRVR models:* | | | | | |
| MS-SL | 1.8 | 7.1 | 11.8 | 47.7 | 68.4 |
| GMMFormer | **2.1** | **7.8** | **12.5** | **50.6** | **72.9** |

Table 3: Performance of various models on the Charades-STA dataset.

TV shows. Five natural language sentences are associated with each video, describing different moments in the video. Following (Dong et al. 2022a), we utilize 17,435 videos with 87,175 moments for training and 2,179 videos with 10,895 moments for testing. **ActivityNet Captions** has around 20K videos from YouTube. On average, each video has about 3.7 moments with corresponding sentence descriptions. We use the popular data partition used in (Zhang et al. 2021, 2020). **Charades-STA** includes 6,670 videos with 16,128 sentence descriptions. Each video holds around 2.4 moments with corresponding text queries on average. We use the official data partition for model training and testing.

**Baselines.** Except the SOTA PRVR model MS-SL (Dong et al. 2022a), we also compare our GMMFormer with models designed for T2VR and VCMR. In particular, we choose the following eight T2VR models, *i.e.*, W2VV (Dong, Li, and Snoek 2018), CE (Liu et al. 2019a), HTM (Miech et al. 2019), HGR (Chen et al. 2020), DE++ (Dong et al. 2021), RIVRL (Dong et al. 2022b), CLIP4Clip (Luo et al. 2022), Cap4Video (Wu et al. 2023), and the following three VCMR models, *i.e.*, XML (Lei et al. 2020), ReLoCLNet (Zhang et al. 2021), CONQUER (Hou, Ngo, and Chan 2021). These VCMR models are two-stage, where a first-stage module retrieves candidate videos, followed by a second-stage module to localize specific moments in the candidate videos. As moment annotations are unavailable in PRVR, we have re-trained VCMR models (removing their moment localization modules) using the same video features as ours. For Cap4Video, we utilize the manual crawling approach to obtain auxiliary captions.

**Evaluation Protocols.** Following (Dong et al. 2022a), we utilize rank-based metrics, namely $R@K$ ($K = 1, 5, 10, 100$). $R@K$ is the fraction of queries that correctly retrieve desired items in the top $K$ of the ranking list. For overall comparisons, we also report the Sum of all Recalls (SumR).

**Implementation Details.** For video representations on TVR, we utilize features provided by (Lei et al. 2020), 3,072-D visual features obtained by concatenating frame-

|  | CLIP4Clip | Cap4Video | CONQUER | MS-SL | GMMFormer |
|---|---|---|---|---|---|
| **FLOPs (G)** | 5.77 | 7.35 | 5.65 | 1.29 | 1.95 |
| **Params (M)** | 103.65 | 104.84 | 22.55 | 4.85 | 12.85 |

Table 4: Model comparisons in terms of FLOPs and parameters.

| Database Size | 500 | 1,000 | 1,500 | 2,000 | 2,500 |
|---|---|---|---|---|---|
| *runtime (ms):* | | | | | |
| MS-SL | 4.89 | 6.11 | 8.06 | 10.42 | 12.93 |
| GMMFormer | **2.68** | **2.93** | **3.40** | **3.94** | **4.56** |
| *memory usage (M):* | | | | | |
| MS-SL | 50.02 | 100.04 | 150.06 | 200.08 | 250.11 |
| GMMFormer | **2.53** | **5.07** | **7.60** | **10.14** | **12.67** |

Table 5: Comparisons in terms of retrieval efficiency of PRVR models.

| GB | QDL | R@1 | R@5 | R@10 | R@100 | SumR |
|---|---|---|---|---|---|---|
|  |  | 11.6 | 29.6 | 40.4 | 81.8 | 163.5 |
| ✓ |  | 12.9 | 32.2 | 43.9 | 83.9 | 172.9 |
|  | ✓ | 12.3 | 31.4 | 42.5 | 83.6 | 169.9 |
| ✓ | ✓ | **13.9** | **33.3** | **44.5** | **84.9** | **176.6** |

Table 6: Ablation studies of GMMFormer on TVR. GB means GMMFormer block and QDL means query diverse loss.

| CW | R@1 | R@5 | R@10 | R@100 | SumR |
|---|---|---|---|---|---|
| Boxcar | 12.9 | 32.1 | 43.3 | 83.9 | 172.1 |
| Bartlett | 13.1 | 32.6 | 43.8 | 84.4 | 174.0 |
| Gaussian | **13.9** | **33.3** | **44.5** | **84.9** | **176.6** |

Table 7: Ablation studies of the constraint window on TVR. CW means constraint window.

level ResNet152 (He et al. 2016) features and segment-level I3D (Carreira and Zisserman 2017) features. On ActivityNet Captions and Charades-STA, we only utilize I3D features provided by (Zhang et al. 2020) and (Mun, Cho, and Han 2020), respectively. For sentence representations, we use 768-D RoBERTa features provided by (Lei et al. 2020) on TVR. On ActivityNet Captions and Charades-STA, we use 1,024-D RoBERTa features extracted by (Dong et al. 2022a). For four types of Gaussian blocks (*i.e.*, low, medium, high and infinite), we set the Gaussian variance to 0.5, 1.0, 5.0 and $\infty$ respectively.

## 4.2  Main Results

**Retrieval Performance.** Table 1, 2, 3 report the retrieval performance of various models on three large-scale video datasets. As can be seen, T2VR models perform poorly compared to VCMR and PRVR models. They focus on the entire relevance between videos and texts, which makes great sense in the T2VR task but is sub-optimal for PRVR. VCMR models focus on retrieving moments, which to some extent, learn the partial relevance between videos and texts, leading to better performance than T2VR models. PRVR models have excellent performance, which is attributed to clip modeling. Among them, our GMMFormer achieves state-of-the-art performance. Major advantages in GMMFormer lie in 1) multi-scale Gaussian blocks enhance the ability to perceive different video moments, 2) and the query diverse loss preserves the semantic structure of text representations.

**Retrieval Efficiency.** In addition, we compare some competitive models mentioned above in terms of FLOPs and model parameters. As shown in Table 4, PRVR models are more lightweight than T2VR and VCMR models while achieving higher retrieval performance. Our GMMFormer has more parameters and calculations than MS-SL because of parallel Gaussian blocks. However, these Gaussian blocks are located in video branches, which will be offline to compute beforehand. We further compare GMMFormer with MS-SL regarding retrieval efficiency in an actual situation. Specifically, we build a video subset from TVR and measure average runtime and memory usage to complete the retrieval process for a single text query under different database sizes

settings. For a fair comparison, the reported runtime is measured on the same Nvidia RTX3080Ti GPU. As shown in Table 5, GMMFormer is about 2.5 times faster than MS-SL, and the storage overhead of GMMFormer is 20 times smaller than MS-SL. The main superiority of GMMFormer in terms of efficiency lies in compact clip embeddings, which are generated by implicit clip modeling.

## 4.3  Ablation Study

**GMMFormer Block.** For ablations on the proposed GMMFormer block, we first alternate the proposed network into a baseline by replacing GMMFormer blocks with vanilla Transformer blocks and removing the query diverse loss. As illustrated in Table 6, keeping the GMMFormer block for the baseline model will improve retrieval performance, and replacing it will degrade retrieval performance compared to the full setup, demonstrating its effectiveness for PRVR. We owe it that the GMMFormer block can provide multi-scale clip information and perceive video moments with different lengths.

**Gaussian Block.** In Section 3.3, we choose four types of Gaussian blocks with low, medium, high and infinite variance respectively to perceive different-length video moments. In this subsection, we investigate the impact of these Gaussian blocks. We successively remove one kind of these Gaussian blocks and construct four variants (*i.e.*, w/o low, w/o medium, w/o high and w/o infinite). Then, we define the moment-to-video ratio (M/V) of a query measured by its corresponding moment's length ratio in the entire video. Next, we split ActivityNet Captions into four groups according to M/V (*i.e.*, 0.00-0.25, 0.25-0.50, 0.50-1.00, 0.00-1.00). We report the performance (SumR) of different variants on different groups in Figure 4. All variants perform worse than

Figure 4: Ablation studies of the Gaussian block on ActivityNet Captions with different types of queries. Queries are grouped according to their moment-to-video ratios (M/V). Different Gaussian blocks are good at handling different M/V groups. And a GMMFormer variant w/o any Gaussian block will perform poorly on the corresponding group.

the full setup, showing that four types of Gaussian blocks all play their roles in GMMFormer. Interestingly, we find that in the group with low M/V (0.00-0.25), the variant w/o low is the worst performer. The same phenomenon happens to the variant w/o medium in the group with medium M/V (0.25-0.50) and the variants w/o high or infinite in the group with high M/V (0.50-1.00), verifying the rationality of designed multi-scale Gaussian blocks.

**Constraint Window.** We also investigate the design of the constraint window during frame interactions. Specifically, we alternate three types of constraint windows (*i.e.*, Boxcar, Bartlett, Gaussian) and report their performance in Table 7. As can be seen, the variant with the Boxcar window performs poorly, which is consistent with the intuition that video frames should pay more attention to adjacent frames. Besides, the Gaussian window outperforms the Bartlett window. We attribute this to the smooth and natural characteristics of the Gaussian distribution.

**Query Diverse Loss.** We provide ablations on the proposed query diverse loss for PRVR in Table 6. Compared to the full setup, removing query diverse loss will degrade retrieval performance and adding it to the baseline will improve retrieval performance, proving its effectiveness for the PRVR task.

### 4.4 Qualitative Results

**Text-Clip Similarity.** To further reveal the ability of the designed GMMFormer block to explore the partial relevance between videos and texts, we present several text-clip similarity examples on TVR. Specifically, we replace GMMFormer blocks in GMMFormer with vanilla Transformer blocks to build a baseline called w/o GB. As illustrated in Figure 5, the model with GMMFormer blocks can generate more discriminative clip embeddings. For example, in Figure 5 (a), the model w/o GB fails to localize the moment relevant to the text. And in Figure 5 (b) and (c), the model
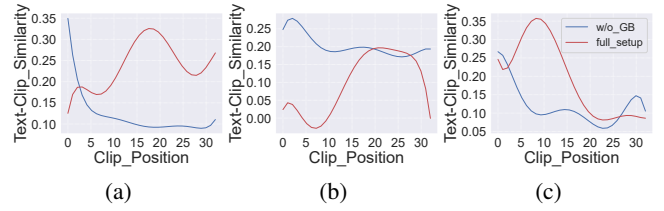


Figure 5: Text-clip similarity visualizations on TVR. w/o GB means a variant of GMMFormer replacing GMMFormer blocks with vanilla Transformer blocks. Note that we smooth out similarity intervals for better observation.
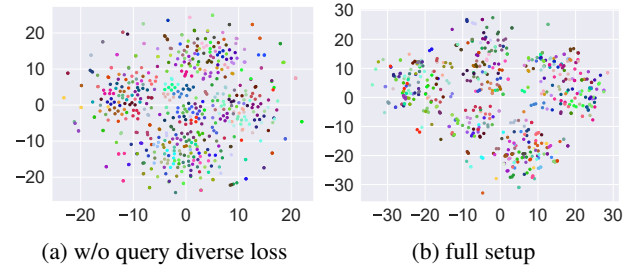


Figure 6: t-SNE visualizations on the TVR subset. Texts relevant to different / same videos are marked with different / same colors.

w/o GB confuses different moments while the model with GMMFormer blocks accurately distinguishes between relevant and irrelevant moments.

**t-SNE Visualization.** To further reveal the ability of the designed query diverse loss to preserve semantic structure of text representations, we show some t-SNE visualizations of GMMFormer without query diverse loss and the full setup. We randomly sample a small subset of videos with their corresponding text queries on TVR for better observation. As shown in Figure 6, the model with the query diverse loss can aggregate relevant text embeddings to a greater extent and make the entire embedding space more discriminative.

## 5 Conclusions

This paper proposes GMMFormer, a Gaussian-Mixture-Model based Transformer for the PRVR task. GMMFormer incorporates a Gaussian-Mixture-Model constraint to model clip representations implicitly and generates compact clip embeddings with high information density. Besides, we propose a query diverse loss to distinguish text queries relevant to the same video, preserving the semantic structure of text representations. Extensive experiments and ablation studies on three large-scale video datasets demonstrate the effectiveness and efficiency of our GMMFormer. In particular, GMMFormer is about 2.5 times faster than the past SOTA MS-SL and the storage overhead of GMMFormer is 20 times smaller than MS-SL.

## Acknowledgments

## References

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bai, Y.; Zeng, Y.; Jiang, Y.; Wang, Y.; Xia, S.-T.; and Guo, W. 2020. Improving query efficiency of black-box adversarial attack. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, 101–116. Springer.

Bai, Y.; Zeng, Y.; Jiang, Y.; Xia, S.-T.; Ma, X.; and Wang, Y. 2021. Improving adversarial robustness via channel-wise activation suppressing. *arXiv preprint arXiv:2103.08307*.

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

Chen, S.; Zhao, Y.; Jin, Q.; and Wu, Q. 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.

Dong, J.; Chen, X.; Zhang, M.; Yang, X.; Chen, S.; Li, X.; and Wang, X. 2022a. Partially Relevant Video Retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, 246–257.

Dong, J.; Li, X.; and Snoek, C. G. 2018. Predicting visual features from text for image and video caption retrieval. *IEEE Transactions on Multimedia*, 20(12): 3377–3388.

Dong, J.; Li, X.; Xu, C.; Ji, S.; He, Y.; Yang, G.; and Wang, X. 2019. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9346–9355.

Dong, J.; Li, X.; Xu, C.; Yang, X.; Yang, G.; Wang, X.; and Wang, M. 2021. Dual encoding for video retrieval by text. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8): 4065–4080.

Dong, J.; Wang, Y.; Chen, X.; Qu, X.; Li, X.; He, Y.; and Wang, X. 2022b. Reading-strategy inspired visual representation learning for text-to-video retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(8): 5680–5694.

Faghri, F.; Fleet, D. J.; Kiros, J. R.; and Fidler, S. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*.

Fu, Q.; Xu, Q.; Ong, Y. S.; and Tao, W. 2022. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 35: 3403–3416.

Gao, J.; Sun, C.; Yang, Z.; and Nevatia, R. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, 5267–5275.

Gao, K.; Bai, J.; Chen, B.; Wu, D.; and Xia, S.-T. 2023. Backdoor Attack on Hash-based Image Retrieval via Clean-label Data Poisoning. In *BMVC*.

Gudibande, A.; Chen, X.; Bai, Y.; Xiong, J.; and Song, D. 2022. Test-time Adaptation of Residual Blocks against Poisoning and Backdoor Attacks. *Preprint*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hou, Z.; Ngo, C.-W.; and Chan, W. K. 2021. CONQUER: Contextual query-aware ranking for video corpus moment retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*, 3900–3908.

Jin, Y.; Liu, J.; Wang, F.; and Cui, S. 2022. Where Are You Looking? A Large-Scale Dataset of Head and Gaze Behavior for 360-Degree Videos and a Pilot Study. In *Proceedings of the 30th ACM International Conference on Multimedia*, 1025–1034.

Kim, J.; El-Khamy, M.; and Lee, J. 2020. T-gsa: Transformer with gaussian-weighted self-attention for speech enhancement. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6649–6653. IEEE.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Carlos Niebles, J. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, 706–715.

Lei, J.; Berg, T. L.; and Bansal, M. 2021. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34: 11846–11858.

Lei, J.; Yu, L.; Berg, T. L.; and Bansal, M. 2020. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, 447–463. Springer.

Li, P.; Xie, C.-W.; Xie, H.; Zhao, L.; Zhang, L.; Zheng, Y.; Zhao, D.; and Zhang, Y. 2023. Momentdiff: Generative video moment retrieval from random to real. *arXiv preprint arXiv:2307.02869*.

Li, X.; Xu, C.; Yang, G.; Chen, Z.; and Dong, J. 2019. W2vv++ fully deep learning for ad-hoc video search. In *Proceedings of the 27th ACM international conference on multimedia*, 1786–1794.

Liu, J.; Wang, Y.; Wang, Y.; Wang, Y.; Cui, S.; and Wang, F. 2023a. Mobile Volumetric Video Streaming System through Implicit Neural Representation. In *Proceedings of the 2023 Workshop on Emerging Multimedia Systems*, 1–7.

Liu, J.; Zhu, B.; Wang, F.; Jin, Y.; Zhang, W.; Xu, Z.; and Cui, S. 2023b. CaV3: Cache-assisted Viewport Adaptive Volumetric Video Streaming. In *2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, 173–183. IEEE.

Liu, Y.; Albanie, S.; Nagrani, A.; and Zisserman, A. 2019a. Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luo, H.; Ji, L.; Zhong, M.; Chen, Y.; Lei, W.; Duan, N.; and Li, T. 2022. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508: 293–304.

Miech, A.; Alayrac, J.-B.; Smaira, L.; Laptev, I.; Sivic, J.; and Zisserman, A. 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.

Miech, A.; Zhukov, D.; Alayrac, J.-B.; Tapaswi, M.; Laptev, I.; and Sivic, J. 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.

Mun, J.; Cho, M.; and Han, B. 2020. Local-global video-text interactions for temporal grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10810–10819.

Qu, L.; Liu, M.; Cao, D.; Nie, L.; and Tian, Q. 2020. Context-aware multi-view summarization network for image-text matching. In *Proceedings of the 28th ACM International Conference on Multimedia*, 1047–1055.

Song, X.; Chen, J.; Wu, Z.; and Jiang, Y.-G. 2021. Spatial-temporal graphs for cross-modal text2video retrieval. *IEEE Transactions on Multimedia*, 24: 2914–2923.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, J.; Zeng, Z.; Chen, B.; Wang, Y.; Liao, D.; Li, G.; Wang, Y.; Xia, S.-T.; and Intelligence, P. C. 2022. Hugs Are Better Than Handshakes: Unsupervised Cross-Modal Transformer Hashing with Multi-granularity Alignment. In *33nd British Machine Vision Conference*.

Wang, T.; and Isola, P. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, 9929–9939. PMLR.

Wang, Y.; Wang, J.; Chen, B.; Zeng, Z.; and Xia, S.-T. 2023. Contrastive masked autoencoders for self-supervised video hashing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 2733–2741.

Wu, W.; Luo, H.; Fang, B.; Wang, J.; and Ouyang, W. 2023. Cap4Video: What Can Auxiliary Captions Do for Text-Video Retrieval? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10704–10713.

Zeng, Z.; Wang, J.; Chen, B.; Wang, Y.; Xia, S.-T.; and Intelligence, P. C. 2022. Motion-Aware Graph Reasoning Hashing for Self-supervised Video Retrieval. In *33nd British Machine Vision Conference*.

Zhang, B.; Hu, H.; Lee, J.; Zhao, M.; Chammas, S.; Jain, V.; Ie, E.; and Sha, F. 2020. A hierarchical multi-modal encoder for moment localization in video corpus. *arXiv preprint arXiv:2011.09046*.

Zhang, H.; Sun, A.; Jing, W.; Nan, G.; Zhen, L.; Zhou, J. T.; and Goh, R. S. M. 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.

Zhou, H.; Yu, J.; and Yang, W. 2023. Dual Memory Units with Uncertainty Regulation for Weakly Supervised Video Anomaly Detection. *arXiv preprint arXiv:2302.05160*.