Enhancing Text-Video Retrieval Performance With Low-Salient but Discriminative Objects

Yanwei Zheng, *Member, IEEE*, Bowen Huang, Zekai Chen, and Dongxiao Yu, *Senior Member, IEEE*

***Abstract*— Text-video retrieval aims to establish a matching relationship between a video and its corresponding text. However, previous works have primarily focused on salient video subjects, such as humans or animals, often overlooking Low-Salient but Discriminative Objects (LSDOs) that play a critical role in understanding content. To address this limitation, we propose a novel model that enhances retrieval performance by emphasizing these overlooked elements across video and text modalities. In the video modality, our model first incorporates a feature selection module to gather video-level LSDO features, and applies cross-modal attention to assign frame-specific weights based on relevance, yielding frame-level LSDO features. In the text modality, text-level LSDO features are captured by generating multiple object prototypes in a sparse aggregation manner. Extensive experiments on benchmark datasets, including MSR-VTT, MSVD, LSMDC, and DiDeMo, demonstrate that our model achieves state-of-the-art results across various evaluation metrics.**

***Index Terms*— Text-video retrieval, low-salient but discriminative objects, cross-modal attention.**

I. Introduction

TEXT-VIDEO retrieval is a critical subtask of cross-modal matching [1], [2], [3], [4], which focuses on retrieving videos that most semantically align with a natural language query from a large pool of unlabeled videos. With the exponential growth of online video content, the demand for efficient and accurate retrieval systems has become more pressing. This task is essential for multi-modal visual and language comprehension, where significant gap exist between the text and video modalities.

Existing text-video retrieval studies can be divided into two primary lines: dual-modality methods and multi-modality methods. The former rely solely on frame-based information from videos and word-level information from text. For instance, Stright-CLIP [5] explores the application of CLIP to obtain video representations without the need for annotations made by the users. PromptSwitch [6] precomputes video representations to facilitate learning of enriched semantic features. UCOFIA [7] captures cross-modal similarity information across both coarse and fine levels. In contrast, the latter usually leverage the expert model to extract features from different attributes of the video and text, such as motion, appearance, semantics, and audio. For example, CE [8] introduces a collaborative expert model to effectively aggregate general and specific semantic cues from pre-trained embeddings, improving video retrieval performance. TeachText [9] proposes a novel generalized distillation method that leverages complementary cues from multiple text encoders for enhanced supervision in text-video retrieval. MMT [10] designs a multi-modal transformer that jointly encodes video modality and temporal information, optimizing both visual and language embeddings for video retrieval.

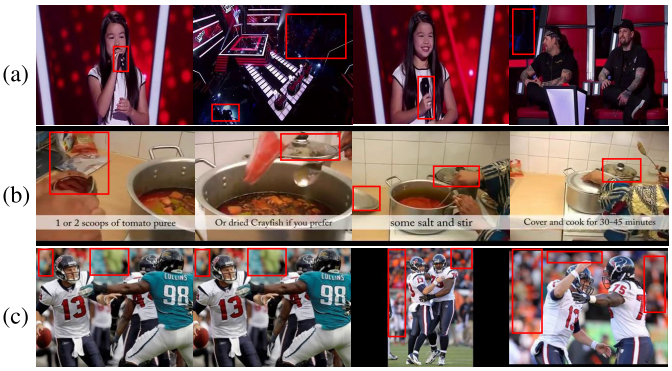


Fig. 1. The examples about LSDOs. In these videos, microphone, pot and audience are important auxiliary information that can help distinguish different scenes.

However, both lines of work often overlook Low-Salient but Discriminative Objects (LSDOs), which can enhance the model's ability to distinguish various scenes. In this paper, LSDOs refer to items in videos that are not particularly salient, but play a crucial role in determining the categories of these videos. As shown in Fig. 1a, the audience and microphone, while not immediately noticeable, serve as crucial contextual cues, significantly aiding in distinguishing different scenes. Similarly, Fig. 1b and Fig. 1c can be effortlessly distinguished based on the presence of the pot and audience, respectively.

Despite the essential contribution of LSDOs to text-video retrieval, fully leveraging them remains a significant challenge. First, current pre-trained models in computer vision typically

Received 30 August 2023; revised 29 November 2024; accepted 3 January 2025. Date of publication 15 January 2025; date of current version 20 January 2025. This work was supported in part by the National Science Fund for Excellent Young Scholars of China under Grant 62122042 and in part by the Key Technology Research and Industrialization Demonstration Projects of Qingdao under Grant 23-1-2-qljh-8-gx. The associate editor coordinating the review of this article and approving it for publication was Prof. Heng Tao Shen. *(Corresponding author: Bowen Huang.)*

Yanwei Zheng, Bowen Huang, and Dongxiao Yu are with the School of Computer Science and Technology, Institute of Intelligent Computing, Shandong University, Qingdao 266237, China (e-mail: zhengyw@sdu.edu.cn; huangbw@mail.sdu.edu.cn; dxyu@sdu.edu.cn).

Zekai Chen is with StandardModel Bio, Inc., Philadelphia, PA 19104 USA (e-mail: zachstarkk@gmail.com).

Code is available on-line at https://github.com/visee-sdu/LSDO.

Digital Object Identifier 10.1109/TIP.2025.3527369

focus on prominent visual features [11], such as regions with vivid colors, high contrast, and clear textures. As a result, LSDOs are generally neglected during fine-tuning for downstream tasks, such as text-video retrieval. Furthermore, even when features related to LSDOs are extracted from videos, how to obtain LSDO features from text modality is still a challenge. This is because the textual descriptions are often brief and lack detailed information about the LSDOs, which creates a substantial gap between the two modalities.

To address the above problems, we fully utilize the LSDO features from videos and texts to enhance the performance our model. More specifically, we explore three approaches for video-level LSDOs, namely capturing from multiple videos of distinct classifications, utilizing a single frame from a specific video, and averaging all frames within a given video, to assess their impact on the retrieval outcomes through extensive experiments. Additionally, we also recognize that disparate frames within a video possess varying degrees of reliance on the video-level LSDOs. For instance, certain frames already encompass abundant contextual information, making additional object features unnecessary. To address this issue, we design an object attention module to allocate distinct weights to different frames after the acquisition of the video-level LSDO features, ultimately obtaining frame-level LSDO representations. More precisely, we designate all frames in a video as the query and the video-level features as both the key and the value, utilizing the cross-modal attention to derive a weighted contextual feature for each frame. After extracting LSDO knowledge from the video, we further address the gap between the video and text modalities by extracting corresponding LSDO features from the text. Specifically, we generate a set of object prototypes based on textual descriptions and assign them varying weights to derive text-level LSDO features. Our primary contributions can be summarized as follows:

- To the best of our knowledge, we are the first to consider those Low-Salient but Discriminative Objects (LSDOs) in text-video retrieval, significantly enhancing the model's ability to distinguish various scenes and improving retrieval performance.
- We fully consider the alignment of LSDO knowledge between the videos and texts, effectively bridging the gap between these two modalities. In the video modality, we sequentially extract video-level and frame-level LSDO features from coarse to fine. In the text modality, we obtain text-level LSDO features based on object prototypes.
- Through extensive experiments, we demonstrate the effectiveness of our method, and achieve state-of-the-art results across multiple public benchmark datasets, including MSR-VTT [12], MSVD [13], LSMDC [14] and DiDeMo [15]. For example, on the LSMDC [14] dataset, our method has improved by 1.2%, 1.1%, and 1.6% respectively in Recall@1, Recall@5, and Recall@10 compared to the current best approach.

Paper organization: In Section II, we present the most related work. Section III details our approaches to acquire video-level, frame-level and text-level LSDO features. Section IV reports our experimental results, and Section V concludes the paper with a future research discussion.

II. Related Work

A. Text-Video Retrieval

Owing to the prevalence of noise within large-scale text-video retrieval datasets, such as HowTo100M [16], researchers often resort to reports on smaller datasets such as MSRVTT [12] and MSVD [13]. Consequently, pre-trained expert models are extensively employed to extract various facets of video and text features [8], [9], [10], [17], [18], including appearance, posture, voice, and semantics, thereby compensating for data scarcity. In alternative approaches [19], [20], [21], [22], [23], videos and texts are introduced into a joint encoder as inputs, followed by a binary classifier trained to predict whether a given video-text pairing constitutes a match. Both ClipBERT [19] and VideoBERT [22] embed text-video pairings through BERT-like architectures, facilitating early cross-modal fusion. HERO [20] employs cross-modal transformer and temporal transformer to capture the local context of a video frame and the global video context, respectively.

Recently, a large-scale language-image model known as CLIP [24] was introduced. Researchers have attempted to build upon this model, extending it to a joint text-video model for text-video retrieval tasks [5], [25], [26]. As a formidable visual-language model, CLIP [24] surpasses the performance of many contemporary models [8], [10] in a zero-shot manner. CLIP4Clip [25] presents three video aggregation schemes - mean-pooling, self-attention, and a multimodal transformer - elevating the text-video retrieval performance to next level. To reduce the number of redundant video tokens, CenterCLIP [27] design a multi-segment token clustering algorithm to find the most representative tokens and drop the non-essential ones. X-CLIP [28] presents a novel multi-grained contrastive model for video-text retrieval, and proposes the Attention Over Similarity Matrix module to make the model focus on the contrast between essential frames and words, thus lowering the impact of unnecessary frames and words on retrieval results. ProST [29] uses a progressive approach to perform spatio-temporal prototype matching, capturing detailed spatial elements along with diverse temporal events. UATVR [30] integrates multi-level semantics flexibly by introducing extra learnable tokens within the encoders. However, these works fail to consider those low-salient but discriminative objects in videos and texts, resulting in the omission of vital features.

B. Cross-Modal Attention

Cross-modal attention was first proposed in [31] and has since been predominantly applied to text-image tasks [32], [33], [34], [35], [36], [37], [38]. UNITER [32] acquires a universal text-image representation through extensive pre-training, subsequently empowering heterogeneous downstream tasks with joint multimodal capabilities. ALBEF [33] incorporates a contrastive loss to align image and text representations prior to their fusion via cross-modal attention, promoting

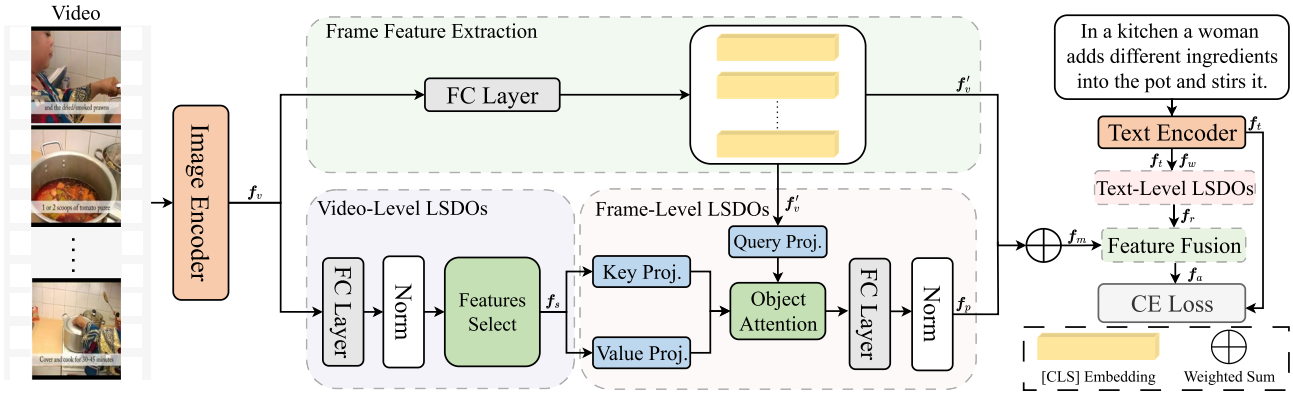


Fig. 2. Framework of our methods.First, we use separate image and text encoders to encode the video frames and the associated text.Next, a fully connected layer in the **Frame Feature Extraction** module further embeds the frame features to obtain the feature representation f'_v.We then select one of three video-level LSDO encoding methods from the **Video-Level LSDOs** block to derive the overall LSDO properties f_s for the video.Using f'_v as the query and f_s as both the key and value, object attention within the **Frame-Level LSDOs** block is employed to obtain the frame-level LSDO information f_p. The **Text-Level LSDOs** block is then utilized to extract the corresponding features f_r from the text.In the **Feature Fusion** module, we apply dot product attention to conditionally aggregate the relevant video frame information based on the text. Finally, the cosine similarity between the video and text representations is computed, followed by the calculation of a cross-entropy loss.

more grounded vision and language representation learning. Oscar [35] introduces a novel cross-modal learning technique that leverages object tags detected in images as anchor points, significantly simplifying the learning of alignments. ViLBERT [36] expands the widely acclaimed BERT architecture into a multimodal two-stream model, processing visual and textual inputs in separate streams that interact through cross-modal attention transformer layers.

Recently, researchers have started to apply cross-modal attention to text-video tasks [26], [39], [40], [41], [42], [43], [44]. ActBERT [39] exploits profound contextual information and fine-grained relations for joint text-video modeling. A multi-layer cross-modal attention network, facilitating the effective optimization of a contrastive loss during training, was proposed in [40]. In [41], a temporal attention mechanism was introduced, which allows to go beyond local temporal modeling and learns to automatically select the most relevant temporal segments. MCQ [43] utilizes the rich semantics of text (nouns and verbs) to formulate questions, enabling the video encoder to capture more regional contents and temporal dynamics. RIVRL [44] employs two branches to learn both the overview and in-depth information of a video, with the latter branch informed by content acquired from the previous branch. X-Pool [26] devises a cross-modal attention mechanism to assign different weights to each video frame based on its textual content. In contrast, our approach employs cross-modal attention to supplement each frame with corresponding LSDO features, thereby compensating for any deficiencies in these information within the frame.

III. Method

Given a query text t and a video index set $\mathcal{V}$, the objective of text-video retrieval is to rank all videos $v \in \mathcal{V}$ according to their similarities with the query text. Fig. 2 illustrates the comprehensive framework of our method for a text-video retrieval task, encompassing five key components: the frame feature extraction block, the video-level LSDOs block, the frame-level LSDOs block, the text-level LSDOs block and the feature fusion block. The first block is used to obtain video frame features. The next three blocks are employed to extract video-level, frame-level and text-level LSDO features respectively, while the last block aggregate the relevant video frame information based on the text. In the following sections, we expound upon the representation of frame and text embeddings in Section III-A. Subsequently, we present the extraction methods for video-level, frame-level and text-level LSDO features in Section III-B, Section III-C, and Section III-D, respectively. Lastly, we introduce the text-conditioned video embedding aggregation technique and the objective function in Section III-E and Section III-F, respectively.

A. Feature Representation

1) Frame-Level Representation: For a video sampled with n frames $\{r_1, r_2, \ldots, r_n\}$, an image encoder processes these frames to obtain frame-level features. Our image encoder is initialized with the public CLIP [24] checkpoints. We first extract the `[CLS]` token from the last layer of the encoder, which represents the aggregate feature for the entire frame. This can be formulated as:

$$\boldsymbol{f}_v^i = \text{ImageEnc}\left(r_i\right), \tag{1}$$

where ImageEnc represents the image encoder, $\boldsymbol{f}_v^i$ is the feature corresponding to the `[CLS]` token of frame r_i. By calculating Eq. (1) for each frame in a video v, we obtain a sequence of `[CLS]` embeddings $\boldsymbol{f}_v = \left[\boldsymbol{f}_v^1, \boldsymbol{f}_v^2, \ldots, \boldsymbol{f}_v^n\right] \in \mathbb{R}^{n \times d}$, where d is the feature size of the `[CLS]` token.

Next, To further represent the features of video frames, we use a fully connected layer in the frame feature extraction block to encode $\boldsymbol{f}_v$, which can be represented as:

$$\boldsymbol{f}'_v = \text{FC}\left(\boldsymbol{f}_v\right), \tag{2}$$

where $\boldsymbol{f}'_v \in \mathbb{R}^{n \times d}$ denotes the features of all frame in a video.

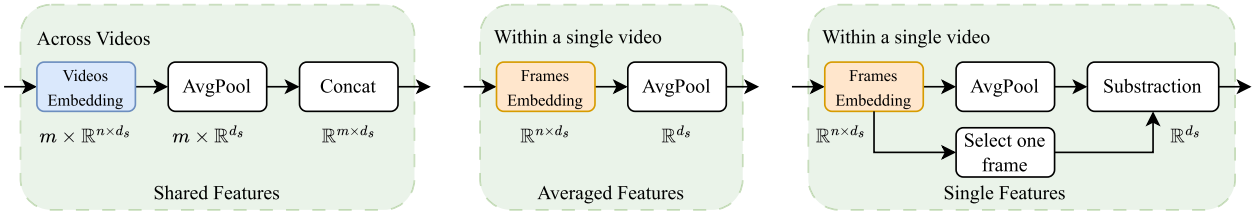


Fig. 3. Features select. The features select module picks a modeling strategy from three video-level LSDO features. Among them, the shared features are derived by encoding m distinct video types, the average features computes the mean of all frames within the video, and the single features represents the difference between a specific video frame and the average features.

2) Textual Representation: Given a text t, we directly employ the text encoder of CLIP to produce the textual representation, initialized with the publicly available CLIP checkpoints [24]. Our model utilizes a lower-cased byte pair encoding tokenizer [45] with a vocabulary size of 49,152. Prior to processing through the text encoder, the textual token sequence is padded with `[BOS]` and `[EOS]` at the beginning and end, respectively. The `[EOS]` token in the last layer is considered the textual feature $\boldsymbol{f}_t \in \mathbb{R}^d$, and the corresponding to the tokens of all words are defined as the word feature vector $\boldsymbol{f}_w \in \mathbb{R}^{L \times d}$, where L denotes the number of words in the sentence. The relationship is described by the following equation:

$$\left[\boldsymbol{f}_t, \boldsymbol{f}_w\right] = \text{TextEnc}\left(t\right), \tag{3}$$

where TextEnc represents the text encoder.

B. Video-Level LSDO Features Extraction

LSDOs are a significant property of videos, which aid in identifying the categories of the videos. To improve retrieval performance, we take these objects in videos fully into account. We propose three methods for obtaining video-level LSDO representations: *shared features*, *average features*, and *single features*, as illustrated in Fig. 3. Shared features are obtained by extracting features from several different types of videos. Then we merge these features from multiple videos to obtain final representations, which can be defined as

$$\boldsymbol{f}_s^{shared} = \text{Concat}\left(\phi\left(v_1\right), \phi\left(v_2\right) \ldots \phi\left(v_m\right)\right), \tag{4}$$

where $v_1, v_2, \ldots v_m$ denote m different type of videos. ϕ is a transformer-structured network, used to extract video features, and Concat concatenates multiple video features. Through this function, we obtain the shared features $\boldsymbol{f}_s^{shared} \in \mathbb{R}^{m \times d_s}$, where d_s is the dimension of the feature generated by ϕ.

However, we observe the effectiveness of shared features heavily depends on the chosen video types. A group of shared features that perform well in some situations may not generalize to other situations. To address this issue, we introduce two alternative schemes that use a single or the average of all frames in the video as the video-level representations. We refer to these schemes as single features and average features. The average features calculates the mean value of all frames to indicate LSDOs, which can be formulated as

$$\boldsymbol{f}_s^{average} = \frac{1}{n}\sum_{i=1}^{n} \text{Norm}\left(\text{FC}\left(\boldsymbol{f}_v^i\right)\right), \tag{5}$$

where $\boldsymbol{f}_v^i$ is the embedding of `[CLS]` token in the i-th frame, Norm is a Normalization layer [46], FC is a fully connected network that projects the dimension of $\boldsymbol{f}_v^i$ from d to d_s and $\boldsymbol{f}_s^{average} \in \mathbb{R}^{d_s}$ is the average features.

Unlike the average features, the single features first selects one frame from all frame features of the video, and subtracts it from $\boldsymbol{f}_s^{average}$. Then we set any result greater than the threshold θ to 0 to remove non-LSDO features. The single features $\boldsymbol{f}_s^{single} \in \mathbb{R}^{d_s}$ can be defined as

$$\boldsymbol{f}_{sub} = \text{Norm}\left(\text{FC}\left(\boldsymbol{f}_v^i\right)\right) - \boldsymbol{f}_s^{average}, \tag{6}$$

$$\boldsymbol{f}_s^{single}\left(j\right) = \begin{cases} \boldsymbol{f}_{sub}\left(j\right), & \boldsymbol{f}_{sub}\left(j\right) \le \theta \\ 0, & \boldsymbol{f}_{sub}\left(j\right) > \theta \end{cases}, \tag{7}$$

where θ is our defined threshold, $j \in [0, d_s)$ means the j-th feature in $\boldsymbol{f}_{sub}$ and $\boldsymbol{f}_{sub}$ is the difference between the selected frame and $\boldsymbol{f}_s^{average}$.

C. Frame-Level LSDO Features Extraction

In Section III-B, we have attained three video-level LSDO features. For the convenience of description, we mark them as $\boldsymbol{f}_s$. However, the importance of the LSDOs in different frames is inconsistent. Some frames may need more LSDO information, while others do not. So the direct combination of the same video-level LSDO features and each frame feature cannot adapt to the actual situation. To solve this problem, our idea is to design a frame-level LSDOs block to extract the corresponding information of each frame. The core mechanism is our adaptation of a scaled dot product attention between $\boldsymbol{f}_s$ and all frames in the video. Conditioned on these frames, we gain frame-level LSDO embeddings that learn to capture the most consistent representations according to different frame features. Since frames with higher demand for LSDO information are more dependent on video scenes, our scaled dot product attention mechanism can learn to give greater weight to these frames to get more LSDO features.

To elaborate, in our frame-level LSDOs block, we first project a video embedding $\boldsymbol{f}_v' \in \mathbb{R}^{n \times d}$ into a single query $\boldsymbol{q}_v \in \mathbb{R}^{n \times d_f}$ and $\boldsymbol{f}_s \in \mathbb{R}^{r \times d_s}$ into both key $\boldsymbol{k}_s \in \mathbb{R}^{r \times d_f}$ and value $\boldsymbol{v}_s \in \mathbb{R}^{r \times d_f}$ matrices, where r can be 1 or m depending on the selection of $\boldsymbol{f}_s$, d_f is the size of the projection dimension. The projections are defined as

$$\boldsymbol{q}_v = \text{Norm}\left(\boldsymbol{f}_v'\right) W_q', \tag{8}$$

$$\boldsymbol{k}_s = \text{Norm}\left(\boldsymbol{f}_s\right) W_k', \tag{9}$$

$$\boldsymbol{v}_s = \text{Norm}\left(\boldsymbol{f}_s\right) W_v', \tag{10}$$

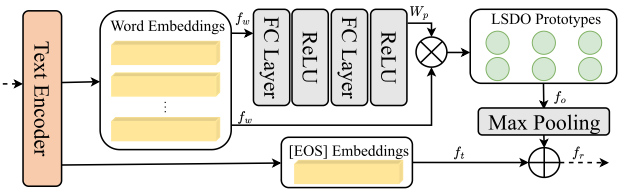


Fig. 4. Based on $\boldsymbol{f}_w$, we employ two fully connected layers to predict a weight matrix that yields LSDO prototypes, denoted as $\boldsymbol{f}_o$. Subsequently, $\boldsymbol{f}_o$ and $\boldsymbol{f}_t$ are fused to obtain the final textual representation.

where W_q', W_k', W_v' are the projection matrices in $\mathbb{R}^{d\times d_f}$, $\mathbb{R}^{d_s\times d_f}$, $\mathbb{R}^{d_s\times d_f}$ respectively. In order to learn flexible conditioning between the given video frames and LSDOs, we then adapt scaled dot product attention from the query-projected frames embeddings to the key-projected video scene embeddings. The dot product attention gives relevancy weights from each frame to $\boldsymbol{f}_s$ which we leverage to acquire frame-level LSDO embeddings:

$$\text{Attention}\left(\boldsymbol{q}_v, \boldsymbol{k}_s, \boldsymbol{v}_s\right) = \text{softmax}\left(\frac{\boldsymbol{q}_v \boldsymbol{k}_s^T}{\sqrt{d_f}}\right)\boldsymbol{v}_s, \tag{11}$$

As such, the $\boldsymbol{q}_v$, $\boldsymbol{k}_s$ and $\boldsymbol{v}_s$ matrices can be interpreted similar to those in the original scaled dot product attention proposed in [47] except with cross-modal interactions. That is, the query-projected frames embeddings is used to seek from the key-projected video-level LSDO embeddings to obtain different weights. The value-projected embeddings represent the overall LSDO features of the video which we want to assign to different frames.

To embed the LSDO features into a joint space with video frames, we project the frame-level LSDO embeddings from the attention module back into $\mathbb{R}^d$ by applying a weight $W_o' \in \mathbb{R}^{d_f\times d}$ to obtain

$$\boldsymbol{f}_p = \text{Norm}\left(\text{Attention}\left(\boldsymbol{q}_v, \boldsymbol{k}_s, \boldsymbol{v}_s\right) W_o'\right), \tag{12}$$

where the resulting output $\boldsymbol{f}_p$ is the frame-level LSDO features conditioned on each frame in a video.

D. Text-Level LSDO Features Extraction

In Section III-B and Section III-C, we derived LSDO features from video data. To bridge the gap between the video and text modalities, we also extract corresponding LSDO embeddings from the text. As shown in Fig.4, based on $\boldsymbol{f}_w$ obtained in Section III-A.2, we apply two fully connected layers to generate a sparse matrix $W_p \in \mathbb{R}^{K\times L}$, where K is the number of prototypes. This matrix is then used to weight $\boldsymbol{f}_w$, resulting in the desired LSDO prototypes, denoted as $\boldsymbol{f}_o$, which can be defined as:

$$\boldsymbol{f}_o = W_p \cdot \boldsymbol{f}_w, \tag{13}$$

where $\boldsymbol{f}_o \in \mathbb{R}^{K\times d}$. Ideally, $\boldsymbol{f}_o$ represents word features in the text that are relevant to LSDOs. Finally, $\boldsymbol{f}_o$ and $\boldsymbol{f}_t$ are combined to represent the final textual features $\boldsymbol{f}_r \in \mathbb{R}^d$.

$$\boldsymbol{f}_r = \gamma \boldsymbol{f}_t + (1-\gamma)\,\text{MaxPooling}\left(\boldsymbol{f}_o\right), \tag{14}$$

where γ is a constant between 0 and 1.

E. Feature Fusion

After obtaining frame-level LSDO features in Section III-C, we need to combine it with $\boldsymbol{f}_v'$ to get representation of frame with LSDO information. We define the representation $\boldsymbol{f}_m \in \mathbb{R}^{n\times d}$ as

$$\boldsymbol{f}_m = \alpha \boldsymbol{f}_v' + (1-\alpha)\,\boldsymbol{f}_p, \tag{15}$$

where α is a digit from 0 to 1.

To compute similarity between given text and video, we need to embed them into a joint space. So the key problem is how to design an aggregation function to fuse multiple frame features in $\boldsymbol{f}_m$ into one frame feature. To address the problem, we employ a text-conditioned video feature aggregation way [26]. Similar to Section III-C, we still use the scaled dot product attention mechanism to aggregate $\boldsymbol{f}_m$.

Specifically, We first use text embedding $\boldsymbol{f}_r \in \mathbb{R}^d$ as query and video frame embedding with LSDO features $\boldsymbol{f}_m \in \mathbb{R}^{n\times d}$ as both key and value, and then project them to dimension d_a. The procedure can be described as

$$\boldsymbol{q}_r = \text{Norm}\left(\boldsymbol{f}_r\right) W_q'', \tag{16}$$

$$\boldsymbol{k}_v = \text{Norm}\left(\boldsymbol{f}_m\right) W_k'', \tag{17}$$

$$\boldsymbol{v}_v = \text{Norm}\left(\boldsymbol{f}_m\right) W_v'', \tag{18}$$

where W_q'', W_k'', W_v'' are projection matrices in $\mathbb{R}^{d\times d_a}$ and $\boldsymbol{q}_r$, $\boldsymbol{k}_v$, and $\boldsymbol{v}_v$ are the corresponding query, key and value, respectively. The dot product attention assigns the weights of text to relevant video frames, which can be defined as

$$\text{Attention}(\boldsymbol{q}_r, \boldsymbol{k}_v, \boldsymbol{v}_v) = \text{softmax}\left(\frac{\boldsymbol{q}_r \boldsymbol{k}_v^T}{\sqrt{d_a}}\right)\boldsymbol{v}_v. \tag{19}$$

Finally, we project the embedding in Eq. (19) into $\mathbb{R}^d$ to maintain the same dimension as the text embedding. The projection is defined as

$$\boldsymbol{f}_a = \text{Norm}\left(\text{Attention}\left(\boldsymbol{q}_r, \boldsymbol{k}_v, \boldsymbol{v}_v\right) W_o''\right), \tag{20}$$

where W_o'' is our projection matrices in $\mathbb{R}^{d_a\times d}$, $\boldsymbol{f}_a$ is our final video representation embedding.

F. Loss Function

Our model is trained in a dataset consisting of N text and video pairs $\{(t_i, v_i)\}_{i=1}^N$. In each pair, the text t_i is the corresponding description of the video v_i. The cross entropy loss from [48] is employed to optimize our model, which considers matching text-video pairs as positives and all other pairwise text-video combinations in the batch as negatives. The symmetric text-to-video(t2v) and video-to-text(v2t) losses are defined as

$$\mathcal{L}_{t2v} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp\left(s\left(t_i, v_i\right)\cdot\lambda\right)}{\sum_{j=1}^{B}\exp\left(s\left(t_i, v_j\right)\right)}, \tag{21}$$

$$\mathcal{L}_{v2t} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp\left(s\left(t_i, v_i\right)\cdot\lambda\right)}{\sum_{j=1}^{B}\exp\left(s(t_j, v_i)\right)}, \tag{22}$$

$$\mathcal{L} = \mathcal{L}_{t2v} + \mathcal{L}_{v2t}, \tag{23}$$

where $s\left(t_i, v_i\right)$ represents the cosine similarity between the text and video, λ is a learnable scaling parameter and B is the batch size.

IV. Experiments

A. Datasets

MSR-VTT [12] is a common benchmark text-video retrieval dataset comprising 10,000 videos and 200,000 captions. We observe that videos in the MSR-VTT dataset include 20 distinct scene categories, which bolsters our impetus to derive scene information from the videos. Video durations within this dataset span from 10 to 32 seconds. For comparison with previous works, we use two divisions designated as **MV-7K** and **MV-9k**. The former as designated in [16], encompasses a subset of roughly about 7k videos for the training set while the latter, adhering to the partition in [10], consists of approximately 9,000 videos for the training set. The test set, as defined in [49], comprises 1,000 meticulously chosen text-video pairs.

MSVD [13] contains about 120k captions and 1,970 videos. The duration of each video varies from 1 second to 62 seconds. Within this dataset, the videos embody a variety of scene types, which coincides with our approach. The allocation of training, validation, and test sets consists of 1,200, 100, and 670 videos, respectively. The test partition exhibits an inconsistent number of captions and videos. Consequently, we evaluate by regarding all furnished text-video pairs as discrete instances, in accordance with [25].

LSMDC [14] consists of 118,081 movie clips, each associated with a singular caption description. These clips vary in length, ranging from 2 to 30 seconds. Of these, 101,079 clips are designated for training, 7,408 for validation, and 1,000 for testing.

DiDeMo [15] contains 10,000 videos, accompanied by 40,000 captions, with an average duration of 30 seconds per video. Following the approach described in [50], we merge multiple text descriptions of each video into a single paragraph to conduct paragraph-to-video retrieval tasks. The dataset is divided into 8,395 videos for training, 1,065 for validation, and 1,004 for testing.

We present our results on the test set for all datasets to evaluate their performance.

B. Experimental Settings

1) Implementation Details: We run our experiments on 1 NVIDIA RTX 3090 24GB GPU using the Pytorch[1] library. We adopt CLIP's VIT-B/32 image encoder and a transformer-based text encoder as our Image Encoder and Text Encoder, respectively. Specifically, the text encoder consists of multi-head self-attention and feed-forward networks. The transformer consists of 12 layers and 8 attention heads. The dimension of the query, key and value features is 512. We establish the dimensions of d, d_s, d_f, and d_a as 512, while initializing the scaling parameter λ from a pretrained CLIP model. All projection weight matrices and biases are initialized as identity and zeros, respectively. Our models undergo end-to-end fine-tuning on each dataset. For the MSVD, MSRVTT, and LSMDC datasets, we configured our experiments with a batch size of 32, sampling 12 frames per video. In contrast, for the DiDeMo dataset, we set the batch size to 6, with a higher frame sampling rate of 64 frames per video. During fine-tuning, we designate a learning rate of $1e^{-6}$ for CLIP-initialized weights. For additional parameters, we set learning rates of $3e^{-5}$, $1e^{-5}$, $1e^{-5}$, $3e^{-5}$ and $1e^{-5}$ for MV-9K, MV-7K, MSVD, LSMDC and DiDeMo, respectively. The AdamW optimizer [51] is utilized with a weight decay of 0.2. The initial value of the threshold, θ, is set to 0.9. In the single features, to more accurately capture the relevant LSDO features, we divide the video evenly into 12 segments and randomly select a frame from the 6th segment as $\boldsymbol{f}_v^i$ for the MSRVTT, MSVD, and LSMDC datasets. For the DiDeMo dataset, we divide the video into 64 segments and randomly select a frame from the 32nd segment. In the shared features, The number of video types, m is set to 20. Specifically, considering that the MSRVTT dataset comprises 20 categories of videos, we extract one video from each category. Each video is sampled for 12 frames, which are then encoded using the Image Encoder. Subsequent operations, including average pooling and concatenation, yield the shared features. During text-level LSDO features learning, we also set the number of prototypes, K, to 20. For all videos, each frame is resized to 224×224, in line with prior works [8], [25], [52]. Lastly, The constants α and γ are both set to 0.8.

2) Evaluation Protocols: To evaluate the retrieval performance of our model, we use recall at Rank K (R@K, where higher values are preferable), median rank (MdR, lower is better), and mean rank (MnR, with lower values being more desirable) as the evaluation metrics, which has been widely used in previous works [10], [25], [52], [53], [54], [55].

3) Fast Inference Method: In application, because query texts are not a priori known during inference, the efficiency will be affected with the cross-attention scheme for calculating video-text similarities. To address this problem, we follow the processing method in [26]. We first mean-pool the pre-computer frame embeddings coming from our model and very efficiently obtain a set of most similar candidates from the index set given a retrieval query. Then we run the cross-attention scheme only on these candidates and re-rank them for retrieval. In fact, this processing method does not affect the retrieval performance of our model.

C. Performance Comparison

To evaluate our proposed model, we compare it with the previous works on MSR-VTT, MSVD, LSMDC and DiDeMo. We tabulate the text-to-video (t2v) retrieval performance of our model trained the MV-9K, MV-7K, MSVD, LSMDC and DiDeMo in Table I, Table II, Table III, Table IV and Table V, respectively. *LSDO-shared*, *LSDO-average* and *LSDO-single* denote the results of three distinct videl-level LSDO representations, respectively. In comparison to preceding studies, we attain state-of-the-art (SOTA) results on most metrics all datasets.

Specifically, *LSDO-shared* exhibits superior performance on the MV-9K dataset. Compared with UCOFIA [7] which is the best result in previous works, this method improve Recall@5 and MnR by 3.1% and 0.9, respectively. It also achieves

[1] https://pytorch.org/

TABLE I

t2v RESULTS ON THE MV-9K DATASET

Methods	R@1 ↑	R@5 ↑	R@10 ↑	MdR ↓	MnR ↓
CE [8]	20.9	48.8	62.4	6.0	28.2
MMT [10]	26.6	57.1	69.6	4.0	24.0
Stright-CLIP [5]	31.2	53.7	64.2	4.0	-
Suppot Set [56]	30.1	58.5	69.3	3.0	-
MDMMT [18]	38.9	69.0	79.7	2.0	16.5
Frozen [52]	31.0	59.5	70.5	3.0	-
TeachText-CE+ [9]	29.6	61.6	74.2	3.0	-
MCQ [43]	37.6	64.8	75.1	3.0	-
CLIP4Clip-meanP [25]	43.1	70.4	80.8	2.0	16.2
CLIP4Clip-seqTransf [25]	44.5	71.4	81.6	2.0	15.3
CenterCLIP [27]	44.2	71.6	82.1	2.0	15.1
X-CLIP [28]	46.1	73.0	83.1	2.0	13.2
X-Pool [26]	46.9	72.8	82.2	2.0	14.3
UATVR [30]	47.5	73.9	83.5	2.0	12.3
ProST [29]	48.2	74.6	83.4	2.0	12.4
PromptSwitch [6]	47.8	73.9	82.2	-	14.1
UCOFIA [7]	**49.4**	72.1	-	-	12.9
LSDO-shared	49.1	**75.2**	**84.2**	2.0	**12.0**
LSDO-average	48.8	74.7	84.0	2.0	12.9
LSDO-single	48.1	73.3	82.9	2.0	13.8

TABLE II

t2v RESULTS ON THE MV-7K DATASET. † INDICATES RESULTS REPRODUCED UNDER THE SAME PARAMETER SETTINGS AS THE ORIGINAL METHOD

Methods	R@1 ↑	R@5 ↑	R@10 ↑	MdR ↓	MnR ↓
HowTo100M [16]	14.9	40.2	52.8	9.0	-
ActBERT [39]	8.6	23.4	33.1	36.0	-
NoiseE [57]	17.4	41.6	53.6	8.0	-
ClipBert [19]	22.0	46.8	59.9	6.0	-
CLIP4Clip-meanP [25]	42.1	71.9	81.4	2.0	15.7
CLIP4Clip-seqTransf [25]	42.0	68.6	78.7	2.0	16.2
CenterCLIP [27]	43.7	71.3	80.8	2.0	16.9
X-Pool [26]	43.9	72.5	82.3	2.0	14.6
PromptSwitch† [6]	41.6	70.1	79.6	2.0	16.9
ProST [29]	44.5	72.3	82.4	2.0	13.8
UCOFIA† [7]	44.8	70.4	80.5	2.0	15.3
LSDO-shared	44.7	72.5	81.2	2.0	14.5
LSDO-average	**46.2**	**74.0**	**83.5**	2.0	**13.5**
LSDO-single	45.3	72.9	82.6	2.0	14.1

TABLE III

t2v RESULTS ON THE MSVD DATASET

Methods	R@1 ↑	R@5 ↑	R@10 ↑	MdR ↓	MnR ↓
CE [8]	19.8	49.0	63.8	6.0	23.1
Stright-CLIP [5]	37.0	64.1	73.8	3.0	-
Suppot Set [56]	28.4	60.0	72.9	4.0	-
Frozen [52]	33.7	64.7	76.3	3.0	-
NoiseE [57]	20.3	49.0	63.3	6.0	-
TeachText-CE+ [9]	25.4	56.9	71.3	4.0	-
CLIP4Clip-meanP [25]	46.2	76.1	84.6	2.0	10.0
CLIP4Clip-seqTransf [25]	45.2	75.5	84.3	2.0	10.3
CenterCLIP [27]	47.6	77.5	86.0	2.0	9.8
X-CLIP [28]	47.1	77.8	-	-	9.5
X-Pool [26]	47.2	77.4	86.0	2.0	9.3
UATVR [30]	46.0	76.3	85.1	2.0	10.4
PromptSwitch [6]	47.1	76.9	86.1	-	9.5
UCOFIA [7]	47.4	77.6	-	-	9.6
LSDO-shared	44.6	75.4	84.8	2.0	10.3
LSDO-average	47.6	77.7	86.3	2.0	9.0
LSDO-single	**48.0**	**78.1**	**86.7**	2.0	**8.8**

TABLE IV

t2v RESULTS ON THE LSMDC DATASET. † INDICATES RESULTS REPRODUCED UNDER THE SAME PARAMETER SETTINGS AS THE ORIGINAL METHOD

Methods	R@1 ↑	R@5 ↑	R@10 ↑	MdR ↓	MnR ↓
CE [8]	11.2	26.9	34.8	25.3	-
MMT [10]	12.9	29.9	40.1	19.3	75.0
NoiseE [57]	6.4	19.8	28.4	39.0	-
Stright-CLIP [5]	11.3	22.7	29.2	56.5	-
MDMMT [18]	18.8	38.5	47.9	12.3	58.0
Frozen [52]	15.0	30.8	39.8	20.0	-
TeachText-CE+ [9]	17.2	36.5	46.3	13.7	-
CLIP4Clip-meanP [25]	20.7	38.9	47.2	13.0	65.3
CLIP4Clip-seqTransf [25]	22.6	41.0	49.1	11.0	61.0
CenterCLIP [27]	21.9	41.1	50.7	10.0	55.6
PromptSwitch [6]	23.1	41.7	50.5	-	56.8
X-CLIP [28]	23.3	43.0	-	-	56.0
ProST [30]	24.1	42.5	51.6	9.0	54.6
X-Pool [26]	25.2	43.7	53.5	8.0	53.2
UCOFIA† [7]	22.9	41.3	51.1	10.0	60.3
LSDO-shared	23.8	42.4	52.4	9.0	55.1
LSDO-average	**26.4**	**44.8**	**55.1**	8.0	**52.1**
LSDO-single	25.3	43.8	54.2	8.0	53.5

TABLE V

t2v RESULTS ON THE DIDEMO DATASET. † INDICATES RESULTS REPRODUCED UNDER THE SAME PARAMETER SETTINGS AS THE ORIGINAL METHOD

Methods	R@1 ↑	R@5 ↑	R@10 ↑	MdR ↓	MnR ↓
CE [8]	16.1	41.1	82.7	8.3	-
ClipBERT [19]	20.4	48.0	60.8	6.0	-
Frozen [52]	31.0	59.8	72.4	3.0	-
MCQ [43]	37.0	62.2	73.9	3.0	-
TVMM [50]	36.5	64.9	75.4	3.0	-
CLIP4Clip-meanP [25]	43.4	70.2	80.6	2.0	17.5
CLIP4Clip-seqTransf [25]	42.8	68.5	79.2	2.0	18.9
X-CLIP [28]	45.2	74.0	-	-	14.6
ProST [29]	44.9	72.7	82.7	2.0	13.7
PromptSwitch† [6]	35.6	61.7	72.4	3.0	25.2
UATVR [30]	43.1	71.8	82.3	2.0	15.1
UCOFIA [7]	46.5	**74.8**	-	-	13.4
LSDO-shared	44.7	71.1	81.8	2.0	14.5
LSDO-average	46.3	73.9	84.5	2.0	13.4
LSDO-single	**46.8**	74.4	**83.8**	2.0	**13.1**

comparable results in terms of Recall@1. In addition, *LSDO-shared* achieves comparably better retrieval result compared with the other two video-level LSDO features acquisition methods. However, on several other datasets, the retrieval ability of this method declines noticeably. We observe that the model with a shared features exhibits subpar generalization capacity in our experiments. When transitioning between datasets, the performance of *LSDO-shared* may degrade significantly, as it fails to capture the diverse video-level LSDOs present in different datasets due to the variance in objects.

Conversely, the other two methods exhibit superior generalization capacity, attaining favorable outcomes across all datasets. More precisely, *LSDO-average* yields significant improvement on MV-7K and LSMDC, respectively. For example, on MV-7K, *LSDO-average* outperforms the current best method, UCOFIA [7], with improvements of 1.4%, 3.6%, and 3.0% in Recall@1, Recall@5, and Recall@10, respectively. Additionally, it achieves leading or comparable results on other datasets. The metrics for *LSDO-single* show varying degrees of improvement across all datasets. Our experiments reveal that *LSDO-average* outperforms *LSDO-single* on MV-9K, MV-7K and LSMDC, yet the inverse occurs on MSVD and DiDeMo. This phenomenon arises because the *LSDO-average*, which accounts for all video frames, is more suitable

TABLE VI

ABLATION STUDY ON DIFFERENT MODULES WITH *LSDO-shared*. $\mathcal{I}$, $\mathcal{V}$, $\mathcal{F}$, AND $\mathcal{T}$ REPRESENT THE FRAME-FEATURE EXTRACTION, VIDEO-LEVEL LSDOS, FRAME-LEVEL LSDOS, AND TEXT-LEVEL LSDOS MODULES, RESPECTIVELY

No.	$\mathcal{I}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{T}$	MV-9K			MV-7K			MSVD			LSMDC			DiDeMo		
					R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓
0	✓	×	×	×	46.3	71.5	14.1	42.4	69.6	14.9	46.7	76.8	9.7	24.7	42.4	54.5	45.4	72.0	13.7
1	×	✓	✓	×	19.1	45.9	30.1	11.5	33.1	51.5	7.1	22.7	59.9	3.8	11.7	154.2	5.4	14.5	163.1
2	×	✓	✓	✓	21.3	46.7	28.2	14.2	34.3	50.0	9.5	26.1	57.2	5.0	12.1	150.0	5.9	14.8	161.1
3	✓	✓	×	×	46.9	73.0	14.0	42.7	71.6	14.6	43.9	74.7	11.0	22.1	41.1	56.0	43.1	69.8	16.0
4	✓	✓	✓	×	47.5	73.7	13.8	44.4	72.1	14.8	44.4	75.0	10.6	23.6	42.0	55.6	44.2	70.3	15.4
5	✓	×	×	✓	46.6	71.9	14.1	42.9	71.2	**14.4**	**46.9**	**76.5**	**9.3**	**25.1**	**43.0**	**54.1**	**45.9**	**72.7**	**13.5**
6	✓	✓	✓	✓	**49.1**	**75.2**	**12.0**	**44.7**	**72.5**	14.5	44.6	75.4	10.3	23.8	42.4	55.1	44.7	71.1	14.5

TABLE VII

ABLATION STUDY ON DIFFERENT MODULES WITH *LSDO-average*. $\mathcal{I}$, $\mathcal{V}$, $\mathcal{F}$, AND $\mathcal{T}$ REPRESENT THE FRAME-FEATURE EXTRACTION, VIDEO-LEVEL LSDOS, FRAME-LEVEL LSDOS, AND TEXT-LEVEL LSDOS MODULES, RESPECTIVELY

No.	$\mathcal{I}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{T}$	MV-9K			MV-7K			MSVD			LSMDC			DiDeMo		
					R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓
0	✓	×	×	×	46.3	71.5	14.1	42.4	69.6	14.9	46.7	76.8	9.7	24.7	42.4	54.5	45.4	72.0	13.7
1	×	✓	✓	×	43.6	70.1	16.0	40.6	67.9	16.7	44.4	73.3	18.4	21.4	40.2	55.9	37.3	65.7	22.8
2	×	✓	✓	✓	44.2	70.3	15.3	41.0	68.8	15.5	45.0	73.9	14.1	22.6	41.0	55.2	38.8	66.9	19.2
3	✓	✓	×	×	46.4	72.7	14.1	43.0	72.1	14.8	47.0	77.3	9.5	24.9	43.7	53.8	45.5	73.0	13.4
4	✓	✓	✓	×	47.6	73.0	14.2	45.0	72.6	14.6	47.4	77.1	9.2	25.6	44.0	53.0	45.9	73.4	13.6
5	✓	×	×	✓	46.8	72.3	13.8	44.8	72.1	14.3	46.9	76.8	9.3	25.2	43.2	53.7	45.5	72.6	13.6
6	✓	✓	✓	✓	**48.8**	**74.7**	**12.9**	**46.2**	**74.0**	**13.5**	**47.6**	**77.7**	**9.0**	**26.4**	**44.8**	**52.1**	**46.3**	**73.9**	**13.4**

TABLE VIII

ABLATION STUDY ON DIFFERENT MODULES WITH *LSDO-single*. $\mathcal{I}$, $\mathcal{V}$, $\mathcal{F}$, AND $\mathcal{T}$ REPRESENT THE FRAME-FEATURE EXTRACTION, VIDEO-LEVEL LSDOS, FRAME-LEVEL LSDOS, AND TEXT-LEVEL LSDOS MODULES, RESPECTIVELY

No.	$\mathcal{I}$	$\mathcal{V}$	$\mathcal{F}$	$\mathcal{T}$	MV-9K			MV-7K			MSVD			LSMDC			DiDeMo		
					R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓
0	✓	×	×	×	46.3	71.5	14.1	42.4	69.6	14.9	46.7	76.8	9.7	24.7	42.4	54.5	45.4	72.0	13.7
1	×	✓	✓	×	40.8	69.5	16.0	39.1	67.2	17.3	45.1	75.8	10.2	20.7	39.1	57.4	38.5	66.8	21.1
2	×	✓	✓	✓	41.4	69.7	16.0	39.9	68.0	16.8	45.4	76.0	9.8	21.4	40.2	55.8	39.6	67.6	19.6
3	✓	✓	×	×	46.5	72.4	14.0	42.9	71.7	14.7	47.0	77.2	9.5	24.7	42.8	54.1	45.8	73.2	13.5
4	✓	✓	✓	×	47.4	72.7	14.2	44.8	72.3	14.6	47.5	77.4	9.1	24.9	43.0	53.9	46.2	73.6	13.3
5	✓	×	×	✓	46.5	72.6	**13.8**	43.6	72.0	14.5	47.3	77.0	9.3	24.8	42.4	53.7	46.4	73.5	13.4
6	✓	✓	✓	✓	**48.1**	**73.3**	**13.8**	**45.3**	**72.9**	**14.1**	**48.0**	**78.1**	**8.8**	**25.3**	**43.8**	**53.5**	**46.8**	**74.4**	**13.1**

for large-scale datasets with multiple object types, such as MSR-VTT and LSMDC. For simpler datasets like MSVD and DiDeMo, employing a single frame suffices for obtaining accurate LSDO information, considering additional frames may interfere with the final retrieval results.

Based on the above analysis, the three proposed video-level LSDO feature extraction methods are suited for different situation. *LSDO-shared* excels when there is sufficient time for fine-tuning and training. However, this method exhibits limited generalization capabilities. If the objective is to generalize the trained model to different datasets, *LSDO-average* and *LSDO-single* methods are more advantageous. The former is particularly effective for large-scale datasets with numerous categories and complex objects, whereas the latter is better suited for smaller, simpler datasets.

Furthermore, we discern another intriguing observation. Although our method achieves significant progress on MSR-VTT and LSMDC, its performance on MSVD and DiDeMo does not markedly surpass that of previous works. We attribute this to the small size and simplicity of the MSVD and DiDeMo datasets. These two datasets contain only about 48,000 and 8,400 video-text pairs for training, respectively, which account for merely one-fifth and one-twentieth of the MSRVTT dataset.. The training content is relatively simple and easy to learn. Consequently, prior works like X-Pool and UCOFIA can readily learn LSDOs through basic feature fusion, rendering the addition of new LSDO information insufficient to substantially enhance retrieval outcomes.

D. Ablation Study

In this subsection, we conduct ablation studies under different settings to fully examine the effectiveness of different modules and parameters.

1) Ablation Study on Different Modules of Our Framework: To evaluate the impact of various features on the final retrieval performance, we conduct experimental analyses of different modules across all datasets. Table VI, Table VII, and Table VIII present the effects of each module under different video-level LSDO features. Based on the results from the three tables, we draw the following conclusions.

First, our primary finding is that frame features in videos are more critical than various LSDO features. This is also in line with our intuitive feeling, because the frame features contain the subject information in videos. The results presented in No.1, No.2, No.4, and No.6 across all three tables strongly support this conclusion. For example, in Table VI, the results in No.1 and No.4 show that using frame features, as opposed to not using them, led to a significant improvement in Recall@1 and Recall@5 scores on the MV-9K dataset, with increases of 28.4% and 27.8%, respectively. Moreover, we observe from the results in No.0 and No.2 that even when only frame-level features are used, their performance still outperforms the combination of all three types of LSDO features.

Second, an additional interesting observation arises from the results in No.1 and No.4 of the three tables. It is evident that, after removing frame features, the performance of the *LSDO-shared* approach decreases significantly more than that of the

other two methods. This can be attributed to the fact that, in both *LSDO-average* and *LSDO-single*, the model learns the corresponding LSDO features for each video. When frame-level features are removed, the model tends to compensate by learning more object information. In contrast, the *LSDO-shared* approach does not have this ability to adapt, resulting in a larger performance drop.

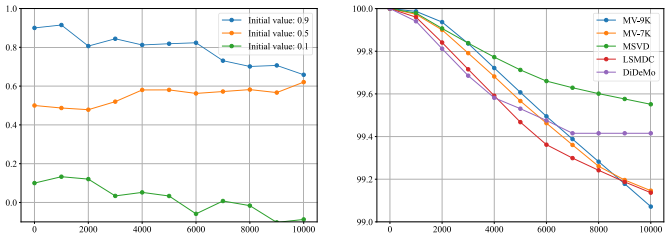
Third, for *LSDO-average* and *LSDO-single*, the inclusion of video-level and frame-level LSDO features contributes positively to the final retrieval performance. This is supported by the results in No.0, No.3, and No.4 of Table VII and Table VIII. We also find that video-level LSDO features tend to improve the Recall@5 score, while frame-level LSDO features enhance the Recall@1 score. For instance, in No.0 and No.3 of Table VII, the addition of video-level LSDO features increased the Recall@5 score by 1.6% on the MV-9K dataset, while the R@1 score showed only a modest increase of 0.1%. When frame-level LSDO features were further incorporated in No.4, the Recall@1 score showed a significant improvement over No.3, with a 1.2% increase on MV-9K. We hypothesize that the video-level LSDO features provide more general information, aiding in distinguishing videos with substantial differences, but they have a limited impact on videos that are more similar. In contrast, frame-level LSDO features are more detailed and help to differentiate videos that are closely related. For *LSDO-shared*, we observe from No.0, No.3, and No.4 in Table Table VI that the combination of video-level and frame-level LSDO features has a positive effect only on the MV-9K and MV-7K datasets. This suggests that the generalization capability of *LSDO-shared* is relatively weak.

Finally, the inclusion of text-level LSDO features yields a consistent improvement in retrieval performance across all datasets. As demonstrated in the results from No.0 and No.5 in Table VII, adding text-level LSDO features resulted in increases of 0.5%, 2.4%, 0.2%, 0.5%, and 0.1% in Recall@1 score on the MV-9K, MV-7K, MSVD, LSMDC, and DiDeMo datasets, respectively.

2) Ablation Study of Threshold θ in LSDO-single: In order to study the influence of the variation of threshold θ on the retrieval performance in *LSDO-single*, we conducted experiments on different settings of θ. The experimental results are shown in Table IX.

According to this table, we have two key findings. First, learnable θ consistently outperforms the non-learnable version. For instance, when the initial value is set to 0.9, the learnable θ improves the Recall@1 score across MV-9K, MV-7K, MSVD, LSMDC, and DiDeMo by 1.6%, 1.2%, 1.1%, 0.2%, and 0.8%, respectively, compared to the non-learnable θ. Second, the initial value of θ plays a significant role in the final experimental outcomes. For example, when θ is non-learnable, setting the initial value to 0.5 results in a 1% improvement in Recall@1 on the MV-9K dataset compared to setting it to 0.1. Even when θ is learnable, the initial value still influences the model's performance, though to a lesser extent.

For the above findings, we have made the following analysis. The higher the threshold θ is set, the more information of the frame will be obtained, so the corresponding LSDO information will be more comprehensive, which can produce better retrieval result. However, if θ is set too high, the obtained LSDO will contain too much irrelevant information, which will disturb the learning of the model. Therefore, selecting an appropriate value for θ is crucial for different datasets. Compared to the non-learnable θ, the learnable θ can automatically adjust its value during training, which often leads to better performance. This also explains why the performance differences between learnable θ values with different initializations are relatively small—the model can adapt and find an superior θ during the learning process.



(a) The progression of θ under different initial values on MV-9k.

(b) The progression of λ with *LSDO-single* across different datasets.

Fig. 5. The evolution of learnable parameters during training. In these two figures, the horizontal axis represents the number of training iterations, while the vertical axis represents the value of θ and λ, respectively.

To further illustrate the variation of the learnable parameter θ during model training, we plot its changes with different initial values over the course of training, as shown in Fig. 5a. As training progresses, the model automatically adjusts and converges to the most suitable value of θ based on the initial setting. When the initial values are 0.5 and 0.9, the final values of θ are similar, which also explains why the results in the last two rows of Table IX are close for the MV-9K dataset. However, when the initial value of θ is set to 0.1, the model converges to a different, yet still suitable, value, indicating that the choice of initial value influences the final retrieval performance.

3) Ablation Study on the Value of λ in the Loss Function: In Eq. (21) and Eq. (22), λ is a learnable parameter whose value evolves throughout the training process. To analyze its dynamics, we present its variation curves across different datasets with *LSDO-single*. As shown in Fig. 5b, in our experiments, the initial value of λ is set to 100, which is inherited from the pre-trained CLIP parameters. We observed that, as training progresses, the value of λ gradually decreases across all datasets. This behavior can be attributed to the role of λ during different training stages. In the early stages, a larger λ helps amplify the differences in similarity scores, enabling the model to more effectively distinguish between positive and negative pairs. As the training proceeds and the model learns more stable and robust features, λ decreases, which reduces the amplification of extreme differences. This prevents excessive gradients and helps maintain the stability of the model during optimization.

4) Ablation Study of Initialization in LSDO-shared: We set different initial values for shared features in *LSDO-shared* to study the impact of initialization methods on retrieval results. Table X shows the impact of initial values of shared features on

TABLE IX

ABLATION STUDY ON THE VALUE OF θ IN *LSDO-single*

θ	learnable	MV-9K			MV-7K			MSVD			LSMDC			DiDeMo		
		R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓
0.1	×	46.1	71.5	15.2	44.0	71.9	15.0	46.0	76.5	10.6	24.6	42.8	56.5	45.1	72.7	14.8
0.5	×	47.1	72.8	14.2	44.7	72.0	14.9	46.4	76.9	9.9	25.0	43.0	54.9	45.8	73.1	14.0
0.9	×	46.5	72.0	14.8	44.1	71.7	15.4	46.9	77.3	9.5	25.1	43.4	54.2	46.0	73.5	13.7
0.1	✓	47.6	72.7	13.9	45.1	72.4	14.2	47.4	77.9	9.2	25.4	43.5	54.0	46.1	74.2	13.3
0.5	✓	47.9	73.1	**13.6**	**45.4**	72.8	14.6	47.1	77.4	9.3	**25.7**	**44.1**	**53.2**	46.5	74.2	13.3
0.9	✓	**48.1**	**73.3**	13.8	45.3	**72.9**	**14.1**	**48.0**	**78.1**	**8.8**	25.3	43.8	53.5	**46.8**	**74.4**	**13.1**

TABLE X

ABLATION STUDY OF INITIALIZATION ON THE VIDEO-LEVEL LSDO FEATURES IN *LSDO-shared*

Initialization	MV-9K			MV-7K			MSVD			LSMDC			DiDeMo		
	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓
Zero	47.5	73.5	14.1	44.0	71.5	14.2	43.8	73.7	11.2	22.9	41.6	**54.7**	43.3	69.5	16.0
Unique	47.9	73.4	13.7	44.4	71.3	**13.9**	44.0	74.0	11.2	22.7	**42.4**	56.4	43.6	69.7	16.0
Feature	**49.1**	**75.2**	**12.0**	**44.7**	**72.5**	14.5	**44.6**	**75.4**	**10.3**	**23.8**	**42.4**	55.1	**44.7**	**71.1**	**14.5**

TABLE XI

THE IMPACT OF FEATURES ACROSS DIFFERENT DIMENSIONS

Method	MV-9K			MV-7K			MSVD			LSMDC			DiDeMo		
	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓	R@1 ↑	R@5 ↑	MnR ↓
FC	**48.8**	**74.7**	**12.9**	**46.2**	**74.0**	**13.5**	**47.6**	**77.7**	**9.0**	**26.4**	**44.8**	**52.1**	**46.3**	**73.9**	**13.4**
Bottleneck	41.6	67.2	16.2	41.3	68.8	14.5	44.4	73.3	11.2	17.7	37.8	74.0	40.7	67.9	16.9
Inverted bottleneck	43.6	70.1	16.0	43.8	72.4	14.2	45.4	75.6	10.0	22.5	42.2	55.7	44.6	68.5	16.1

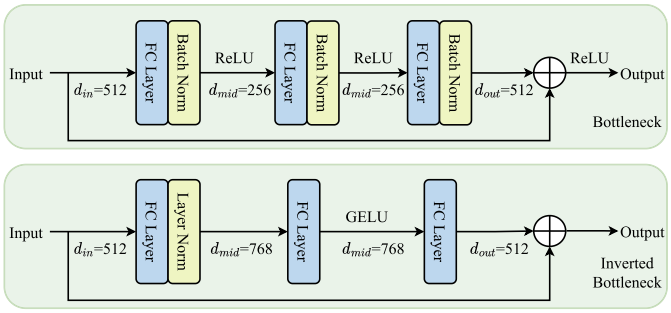


Fig. 6. These two blocks first project the input features to a lower or higher dimensions, and then map them back.

retrieval results. In this experiment, **Zero**, **Unique** and **Feature** respectively represent the use of zeros, one-hot encoding and video feature to initialize shared features.

Based on this table, we find that zeros and one-hot encoding initialization show considerable retrieval efficiency, and the effect of video feature initialization is significantly better than the first two methods. This is because video feature already contain some LSDO information. So when using video feature initialization, our model can learn the corresponding video-level LSDO features more easily, while the other two initialization methods will greatly harm the learning of LSDO information, resulting in worse result.

5) Lower or Higher Feature Dimensions: Inspired by [58], we replace the FC layers in Fig. 2 with bottlenecks or inverted bottlenecks to verify whether the features of lower or higher dimensions can help improve retrieval performance. The structures of these two blocks are shown in Fig. 6. Input features are first mapped to lower or higher dimensions to obtain better feature representations, and then mapped back for subsequent training. We evaluation the effects of the two structures on all datasets employing *LSDO-average*.

Table XI shows the results of replacing the FC layer with bottlenecks or inverted bottlenecks. Based on this table, we make a conclusion that inverted bottlenecks have a slight inhibitory effect on the retrieval performance of our model rather than an improvement effect. We analyze two reasons that led to this result. Firstly, the original feature dimension is already high enough, and further enlargement cannot achieve better results. Secondly, inverted bottlenecks brings more parameters, which may lead to overfitting due to the small amount of video dataset. Additionally, when utilizing bottlenecks to substitute for the FC layers, the retrieval ability of our model significantly drop. This phenomenon indicates that both lower feature dimensions and batch normalization can damage the performance of the retrieval model.

E. Qualitative Results

In this subsection, we present two visualization experiments to demonstrate the model's ability to extract relevant LSDO features. Both experiments are conducted without the text-level LSDOs block to highlight the model's performance in the video modality.

1) Video-Level LSDO Features: To demonstrate that the proposed model can indeed extract video-level LSDO information, we visualized the attention maps of the last transformer layer in Image Encoders with and without the use of LSDO blocks. As shown in Fig. 7, when employing LSDO blocks, the Image Encoder pays significantly more attention to those low-salient but discriminative objects besides the main parts of videos such as human and animals. For instance, in the two samples on the top left, our model with LSDO blocks gives significant attention to both salt and bowl, which is beneficial for identifying cooking videos. Conversely, when not using LSDO blocks, the model tends to focus more on text and overlook important LSDO information. This result indicates

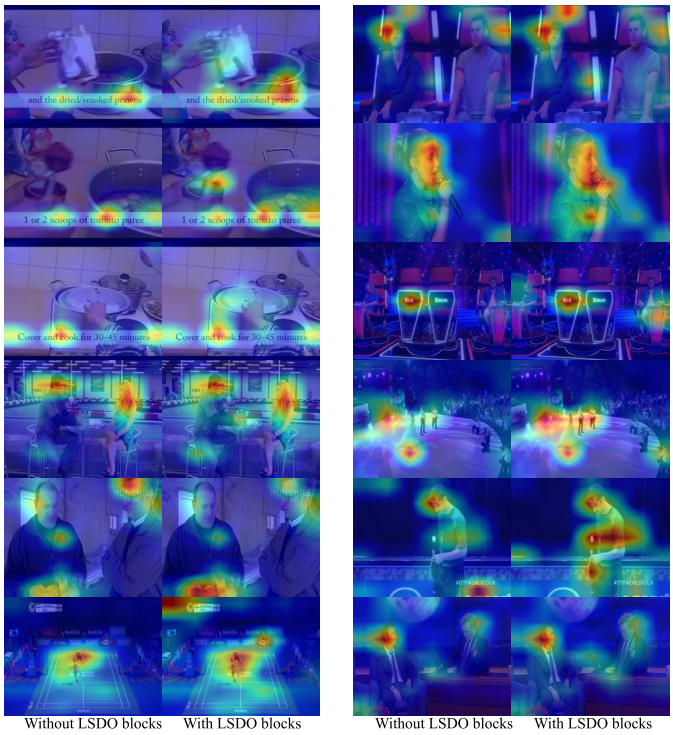


Fig. 7. The figures show the attention maps of the last transformer layer in image encoder.The attention is indicated from low to high with the color blue to red.

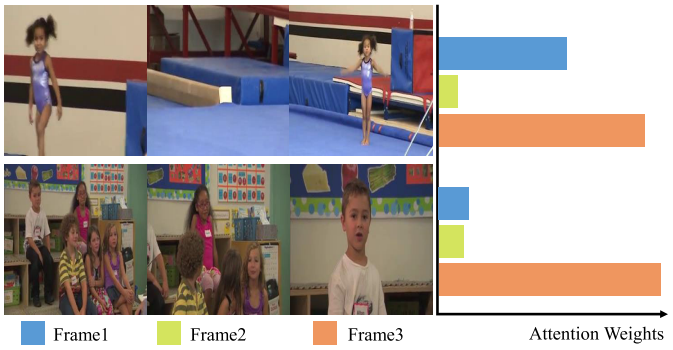


Fig. 8. Qualitative result.For each displayed frame above, the bar plot shows its attention weight of the video-level LSDO features.

the correctness of our conjecture and the effectiveness of the framework.

2) Frame-Level LSDO Features: The frame-level LSDO features are actually derived by weighting the video-level LSDO features for each frame. Therefore, to demonstrate the frame-level LSDO features, we visualize the weights of each frame in the video in Fig. 8. The experiment is completed on the MV-9K dataset and the video-level LSDO features is obtained by the average value of all frames. For each example, we use a histogram to show the attention weights of three sampled frames from a video. In the first example, we can see that the second frame has the lowest weight and the corresponding weights of other frames are higher. Because this frame already contains a lot of LSDO information, no additional LSDO features are needed. Similarly, in the second video, the third frame has the highest weight. This is because the important content of the third frame is a child, and the LSDO information contains less, so it is more necessary to supplement the LSDO information. The above examples fully verify our hypothesis that our model can supplement LSDO information for those frames lacking them.

V. Conclusion and Future Research

In this work, we focus on analyzing the shortcomings of previous text-video retrieval research, which do not fully consider those low-salient but discriminative objects, and propose an alternative framework, which contains video-level, frame-level and text-level LSDOs blocks. We propose three video-level LSDO embeddings acquisition methods and shows how our method to learn different LSDO attention for different frames in a video. When assigning LSDO information to each frame in the video, we introduce a cross-modal attention mechanism. Through this mechanism, we can add relevant features to each frame according to their demand for LSDO information. In the future work, we plan to continue to explore different video-level LSDO acquisition ways and frame-level LSDO attention distribution mechanism. We hope to add more accurate LSDO information to each frame of the video in the subsequent research to improve the effect of text-video retrieval.

References

[1] Z. Wang, X. Xu, J. Wei, N. Xie, Y. Yang, and H. T. Shen, "Semantics disentangling for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 33, pp. 2226–2237, 2024.

[2] Z. Wang, X. Xu, G. Wang, Y. Yang, and H. T. Shen, "Quaternion relation embedding for scene graph generation," *IEEE Trans. Multimedia*, vol. 25, pp. 8646–8656, 2023.

[3] Z. Wang, Z. Gao, K. Guo, Y. Yang, X. Wang, and H. T. Shen, "Multilateral semantic relations modeling for image text retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 2830–2839.

[4] Z. Wang, Z. Gao, Y. Yang, G. Wang, C. Jiao, and H. T. Shen, "Geometric matching for cross-modal retrieval," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 23, 2024, doi: 10.1109/TNNLS.2024.3381347.

[5] J. A. Portillo-Quintero, J. C. Ortiz-Bayliss, and H. Terashima-Marín, "A straightforward framework for video retrieval using clip," in *Pattern Recognition*, E. Roman-Rangel, Á. F. Kuri-Morales, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, and J. A. Olvera-López, Eds., Cham, Switzerland: Springer, 2021, pp. 3–12.

[6] C. Deng, Q. Chen, P. Qin, D. Chen, and Q. Wu, "Prompt switch: Efficient CLIP adaptation for text-video retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 15602–15612.

[7] Z. Wang, Y.-L. Sung, F. Cheng, G. Bertasius, and M. Bansal, "Unified coarse-to-fine alignment for video-text retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 2804–2815.

[8] Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman, "Use what you have: Video retrieval using representations from collaborative experts," in *Proc. 30th Brit. Mach. Vis. Conf.*, Cardiff, U.K., Jan. 2019, p. 279.

[9] I. Croitoru et al., "TeachText: CrossModal generalized distillation for text-video retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 11563–11573.

[10] V. Gabeur, C. Sun, K. Alahari, and C. Schmid, "Multi-modal transformer for video retrieval," in *Computer Vision—ECCV*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham, Switzerland: Springer, 2020, pp. 214–229.

[11] J. Dai et al., "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[12] J. Xu, T. Mei, T. Yao, and Y. Rui, "MSR-VTT: A large video description dataset for bridging video and language," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 5288–5296.

[13] D. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Portland, OR, USA, 2011, pp. 190–200.

[14] A. Rohrbach et al., "Movie description," *Int. J. Comput. Vis.*, vol. 123, no. 1, pp. 94–120, Jan. 2017.

[15] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, "Localizing moments in video with natural language," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5804–5813.

[16] A. Miech, D. Zhukov, J. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, Korea (South), Oct. 2019, pp. 2630–2640.

[17] A. Miech, I. Laptev, and J. Sivic, "Learning a text-video embedding from incomplete and heterogeneous data," 2018, *arXiv:1804.02516*.

[18] M. Dzabraev, M. Kalashnikov, S. Komkov, and A. Petiushko, "MDMMT: Multidomain multimodal transformer for video retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 3354–3363.

[19] J. Lei et al., "Less is more: CLIPBERT for video-and-language learning via sparse sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7331–7341.

[20] L. Li, Y.-C. Chen, Y. Cheng, Z. Gan, L. Yu, and J. Liu, "HERO: Hierarchical encoder for video+language omni-representation pre-training," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 2046–2065.

[21] H. Luo et al., "UniVL: A unified video and language pre-training model for multimodal understanding and generation," 2020, *arXiv:2002.06353*.

[22] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "VideoBERT: A joint model for video and language representation learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7463–7472.

[23] H. Xu et al., "VLM: Task-agnostic video-language model pre-training for video understanding," in *Proc. Findings Assoc. Comput. Linguistics*, 2021, pp. 4227–4239.

[24] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, M. Meila and T. Zhang, Eds., 2021, pp. 8748–8763.

[25] H. Luo et al., "CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning," *Neurocomputing*, vol. 508, pp. 293–304, Oct. 2022.

[26] S. K. Gorti et al., "X-pool: Cross-modal language-video attention for text-video retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4996–5005.

[27] S. Zhao, L. Zhu, X. Wang, and Y. Yang, "CenterCLIP: Token clustering for efficient text-video retrieval," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.* New York, NY, USA: Association for Computing Machinery, Jul. 2022, pp. 970–981.

[28] Y. Ma, G. Xu, X. Sun, M. Yan, J. Zhang, and R. Ji, "X-CLIP: End-to-end multi-grained contrastive learning for video-text retrieval," in *Proc. 30th ACM Int. Conf. Multimedia*. New York, NY, USA: Association for Computing Machinery, Oct. 2022, pp. 638–647.

[29] P. Li et al., "Progressive spatio-temporal prototype matching for text-video retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 4077–4087.

[30] B. Fang et al., "UATVR: Uncertainty-adaptive text-video retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 13677–13687.

[31] R. You, Z. Guo, L. Cui, X. Long, Y. Bao, and S. Wen, "Cross-modality attention with semantic graph embedding for multi-label classification," in *Proc. 34th Conf. Artif. Intell. (AAAI), 32nd Innov. Appl. Artif. Intell. Conf. (IAAI), 10th Symp. Educ. Adv. Artif. Intell. (EAAI)*, New York, NY, USA, Feb. 2020, pp. 12709–12716.

[32] Y. Chen et al., "UNITER: Universal image-text representation learning," in *Computer Vision—ECCV*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham, Switzerland: Springer, 2020, pp. 104–120.

[33] J. Li, R. R. Selvaraju, A. Gotmare, S. Joty, C. Xiong, and S. C. H. Hoi, "Align before fuse: Vision and language representation learning with momentum distillation," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., Dec. 2021, pp. 9694–9705.

[34] L. Harold Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "VisualBERT: A simple and performant baseline for vision and language," 2019, *arXiv:1908.03557*.

[35] X. Li et al., "OSCAR: Object-semantics aligned pre-training for vision-language tasks," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, 2020, pp. 121–137.

[36] J. Lu, D. Batra, D. Parikh, and S. Lee, "ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., Vancouver, BC, Canada, Dec. 2019, pp. 13–23.

[37] A. Miech, J.-B. Alayrac, I. Laptev, J. Sivic, and A. Zisserman, "Thinking fast and slow: Efficient text-to-visual retrieval with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9826–9836.

[38] H. Tan and M. Bansal, "LXMERT: Learning cross-modality encoder representations from transformers," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. China: Association for Computational Linguistics, 2019, pp. 5100–5111.

[39] L. Zhu and Y. Yang, "ActBERT: Learning global-local video-text representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8743–8752.

[40] R. Tan, B. A. Plummer, K. Saenko, H. Jin, and B. Russell, "Look at what i'm doing: Self-supervised spatial grounding of narrations in instructional videos," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 14476–14487.

[41] L. Yao et al., "Describing videos by exploiting temporal structure," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 4507–4515.

[42] C. Zhang, A. Gupta, and A. Zisserman, "Temporal query networks for fine-grained video understanding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4486–4496.

[43] Y. Ge et al., "Bridging video-text retrieval with multiple choice questions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 16146–16155.

[44] J. Dong et al., "Reading-strategy inspired visual representation learning for text-to-video retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5680–5694, Aug. 2022.

[45] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 1715–1725.

[46] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[47] A. Vaswani et al., "Attention is all you need," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., Long Beach, CA, USA, Jun. 2017, pp. 5998–6008.

[48] A. Zhai and H. Wu, "Classification is a strong baseline for deep metric learning," in *Proc. Brit. Mach. Vis. Conf.*, Cardiff, U.K., 2019, p. 91.

[49] Y. Yu, J. Kim, and G. Kim, "A joint sequence fusion model for video question answering and retrieval," in *Computer Vision—ECCV*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Cham, Switzerland: Springer, 2018, pp. 487–503.

[50] C. Lin et al., "Text-adaptive multiple visual prototype matching for video-text retrieval," in *Proc. 36th Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, Jan. 2022, pp. 38655–38666.

[51] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.

[52] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, "Frozen in time: A joint video and image encoder for end-to-end retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 1708–1718.

[53] S. Antol et al., "VQA: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 2425–2433.

[54] C. Jia et al., "Scaling up visual and vision-language representation learning with noisy text supervision," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, M. Meila and T. Zhang, Eds., 2021, pp. 4904–4916.

[55] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021.

[56] M. Patrick et al., "Support-set bottlenecks for video-text representation learning," in *Proc. 9th Int. Conf. Learn. Represent.*, Jan. 2020, pp. 1–18.

[57] E. Amrani, R. Ben-Ari, D. Rotman, and A. M. Bronstein, "Noise estimation using density estimation for self-supervised multimodal learning," in *Proc. 34th Conf. Artif. Intell. (AAAI), 33rd Conf. Innov. Appl. Artif. Intell. (IAAI), 11th Symp. Educ. Adv. Artif. Intell. (EAAI)*, 2021, pp. 6644–6652.

[58] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 11976–11986.

Yanwei Zheng (Member, IEEE) received the B.S. degree from Shandong Jianzhu University in 1999, the M.S. degree from Shandong University in 2004, and the Ph.D. degree from Beihang University in January 2019, supervised by Prof. Zhang Xiong. He is currently an Associate Professor with the Institute of Intelligent Computing (IIC), School of Computer Science and Technology, Shandong University. His research interests include computer vision, visual navigation, and digital twins.

Bowen Huang received the B.S. degree in computer science and technology from Shandong University, Qingdao, China, in 2021, where he is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology. His research interests include computer vision and visual-language navigation.

Zekai Chen received the Ph.D. degree in computer science from The George Washington University in 2021. He is currently Chief Scientist Officer at StandardModel Bio, Inc., focusing on multimodal foundational model and building artificial intelligence for healthcare. He has published several influential papers in top-tier conferences and journals, including but not limited to AAAI, ACL, WACV, IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON COMPUTERS, and among others. His research interests include foundational large model, multi-modal deep learning, multi-task learning, and medical imaging.

Dongxiao Yu (Senior Member, IEEE) received the B.S. degree from the School of Mathematics, Shandong University, in 2006, and the Ph.D. degree from the Department of Computer Science, The University of Hong Kong, in 2014. He became an Associate Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, in 2016. He is currently a Professor with the School of Computer Science and Technology, Shandong University. His research interests include wireless networks, distributed computing, and graph algorithms.