Align and Tell: Boosting Text-video Retrieval with Local Alignment and Fine-grained Supervision

Xiaohan Wang, Linchao Zhu, Zhedong Zheng, Mingliang Xu, Yi Yang, *Senior Member, IEEE,*

***Abstract*—Text-video retrieval is one of the basic tasks for multimodal research and has been widely harnessed in many real-world systems. Most existing approaches directly compare the global representation between videos and text descriptions and utilize the global contrastive loss to train the model. These designs overlook the local alignment and the word-level supervision signal. In this paper, we propose a new framework, called Align and Tell, for text-video retrieval. Compared to the previous work, our framework contains additional modules, *i.e.*, two transformer decoders for local alignment and one captioning head to enhance the representation learning. First, we introduce a set of learnable queries to interact with both textual representations and video representations and project them to a fixed number of local features. After that, local contrastive learning is performed to complement the global comparison. Moreover, we design a video captioning head to provide additional supervision signals during training. This word-level supervision can enhance the visual presentation and alleviate the cross-modal gap. The captioning head can be removed during inference and does not introduce extra computational costs. Extensive empirical results demonstrate that our Align and Tell model can achieve state-of-the-art performance on four text-video retrieval datasets, including MSR-VTT, MSVD, LSMDC, and ActivityNet-Captions.**

***Index Terms*—Text-video retrieval, Multimodal Understanding, Video captioning.**

I. Introduction

TEXT-video retrieval is to return the most relevant videos based on a given text query. Compared with video search systems based on visual information, natural language descriptions are more user-friendly [1] and easy to access for video retrieval. Therefore, with the wide applications of web videos, text-video retrieval has attracted increasing attention, which plays a significant role in online video platforms like YouTube and TikTok. Moreover, It is worth noting that text-video retrieval is also a fundamental research task for multimodal analysis and video understanding. In the past few years, due to the progress of cross-modal pre-training and representation learning [2], [3], remarkable improvement [4]–[7] has been achieved across several text-video retrieval benchmarks [8]–[11].

Xiaohan Wang, Linchao Zhu and Yi Yang are with School of Computer Science, Zhejiang University, Hangzhou, China. E-mail: xiaohan.wang@zju.edu.cn, zhulinchao@zju.edu.cn, yangyics@zju.edu.cn

Zhedong Zheng is with Sea-NExT Joint Lab, School of Computing, National University of Singapore, Singapore 118404. E-mail: zdzheng@nus.edu.sg

Mingliang Xu is with School of Computer Science, Zhengzhou University, Zhengzhou, China. E-mail: iexumingliang@zzu.edu.cn

This work is supported by National Key RD Program of China (No. 2020AAA0108800) and Fundamental Research Funds for the Central Universities (No. 226-2022-00051).

The key to text-video retrieval is mapping the videos and text descriptions to a joint semantic representation space and then measuring the similarities between video and text in this space. Previous works for text-video retrieval can be generally divided into two types based on the training strategies of the video encoder: two-stage optimization and end-to-end optimization. Most of the existing methods [7], [12]–[14] adopt the two-stage optimization strategy and rely on the expert networks. In particular, this line of works pre-extract the visual features and other multi-modal features by the pre-trained experts. These expert models are first trained on task-specific datasets. After that, the extracted video features are utilized to perform cross-modal learning with the text descriptions. As a result, the performance of these two-stage methods highly depends on the capability of the video experts. Besides, the video encoder is fixed after pre-training and can not integrate discriminative information from the text supervision.

In contrast, the other line of works [4], [9], [15] in recent years proposed to optimize the video encoder along with the text encoder in an end-to-end manner. The feature extractors are end-to-end differentiable and thus can be trained in synergy with downstream text-video retrieval tasks. Benefiting from the large-scale datasets [9], [16], these end-to-end models achieve significant performance gain for text-video retrieval. Typically, Frozen [16] utilizes a spatial-temporal video encoder to take both images and videos as input. A curriculum learning schedule is designed to gradually boost performance while increasing the input frames. This model computes the similarity between the text representation and global video features. It overlooks the local alignment for the spatial-temporal video features and word-level text features. ClipBERT [17] builds a cross-modal transformer on top of the video encoder and text encoder for text-video pre-training. This design considers the interactions among the local features. However, the cross-attention operations lead to high computational costs for both training and inference. More recently, Clip4Clip [4] proposes to transfer the knowledge from the text-image pre-trained model CLIP [18] to text-video retrieval. They investigate several post-pretraining strategies and boost the retrieval accuracy by a large margin. According to their findings, the similarity calculators with more parameters and interactions usually perform worse than the simple global comparison. One possible explanation for these results is that the supervision signal provided by contrastive learning is not sufficient to learn complex interaction modules on small-scale text-video datasets.

To address the above-mentioned challenges, in this paper, we exploit the pre-trained CLIP model and propose an Align

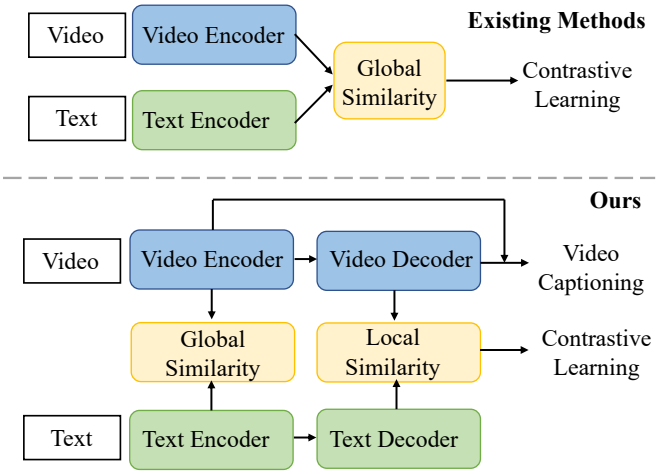

Fig. 1. Comparison between prevailing text-video retrieval learning paradigm (top) and our Align and Tell framework (Bottom). Most existing methods [4], [19] only take the global text-video similarity into consideration and optimize the model with contrastive learning. In contrast, we design the decoder architecture and enable the local comparison for retrieval in an end-to-end manner. Moreover, except for contrastive learning, we leverage an auxiliary video captioning task to provide more supervision for the representation learning.

and Tell framework for text-video retrieval. As shown in Fig. 1, two aspects distinguish our methods from existing works. First, we propose a local alignment module to aggregate the temporal visual features and word-level text features into a fixed number of groups. After that, local contrastive learning is performed to compare the similarity between the local text feature and the corresponding local video feature. We adopt the decoder layers based on transformers to build the local alignment module. This design can provide detailed cross-modal comparison and is complementary to the global similarity calculator. Second, we design a bidirectional captioning head to provide fine-grained supervision signals for the post-training based on the CLIP model. The temporal encoder and the local alignment module introduce extra parameters. The recent literature [4] shows that contrastive learning is not powerful enough to optimize these extra modules well without CLIP initialization. Our captioning head enables word-level supervision for learning better visual representations and interaction modules. This captioning head is only used for training and can be removed during inference. Therefore, it does not introduce any computational cost for deployment. We train the Align and Tell model on various text-video retrieval datasets. Extensive experimental results demonstrate that the model can boost text-video retrieval performance.

The contribution of this paper can be summarized as follows:

To facilitate the alignment of the local text-video feature, we harness the transformer decoder and enable the end-to-end training for mining the discriminative feature in the shared semantic space.

To minimize the semantic gap between the video and text, we introduce an effective strategy to provide word-level supervision during training. Without additional computational costs during inference, we propose a bidirectional captioning head to enhance the visual presentation and alleviate the cross-modal gap.

Third, the empirical results suggest that the Align&Tell model achieves state-of-the-art performance on four standard text-video retrieval benchmarks, *i.e.*, MSRVTT [8], MSVD [11], ActivityNet Captions [20], and LSMDC [10] in both text-to-video and video-to-text retrieval tasks.

II. Related Work

A. Learning Visual Representation with Text Supervision

Traditional visual representation learning depends on the supervision of class labels. Recently, some works [9], [16]–[18], [21] directly adopted the text description to supervise the visual representation learning. This enhances the interactions among multiple knowledge representations from different modalities [22]. It is an emerging research topic due to the benefits of large-scale vision-language pairs from the Internet. Yang *et al.* [5] leveraged two levels of manifold learning to mine the relationships among cross-media data. CLIP (Contrastive Language-Image Pre-training) [18] is a milestone for this direction. With the pretraining on massive-scale image-text pairs, CLIP achieves state-of-the-art performance on image representation learning and the pre-learned knowledge can be transferred to downstream tasks like retrieval. ClipBERT [17] proposed an efficient end-to-end video-language pre-training method with sparsely sampled video frames. More recently, CLIP4Clip [4] proposed to transfer the knowledge from the text-image pre-trained model CLIP [18] to text-video retrieval. They investigated several post-pretraining strategies which boosted the retrieval accuracy by a large margin. In this paper, we also leverage the pre-trained CLIP [18] for text-video retrieval. In contrast to CLIP4Clip [4], we propose a novel local alignment module and introduce an auxiliary video captioning task, which provides more dense supervision and motivates the network to mine fine-grained features. Virtex [23] proposed to predict dense captions for images to learn visual representations. After pre-training, the visual backbone is transferred to downstream visual recognition tasks. Miech *et al.* [24] proposed to train a slow model with the captioning loss and then combine a Fast dual encoder model with the Slow but accurate model via distillation and re-ranking. In contrast to this work, the captioning head is optimized with the dual encoder model during training and is removed for inference.

B. Video Encoder Backbones

Most existing works [25]–[27] utilized 2D or 3D convolutional networks to build the video encoder backbones. Simonyan *et al.* [26] proposed to utilize both RGB frames and optical flow as the 2D CNN input to model appearance and motion, respectively. TSN [25] extended the two-stream CNN by extracting features from multiple temporal segments. Tran *et al.* [27] proposed a 3D CNN to learn the spatial-temporal information. Wang *et al.* [28] combined the 3D CNN with the 2D detection model to enable fine-grained video understanding. [29], [30]. More recently, inspired by the success of ViT [31] for image representation learning, TimeSformer [32] utilized video transformer to modeling

the spatial-temporal information in videos. In this paper, we directly adopt the ViT [31] model in CLIP [18] as our video encoder. Besides, we build a temporal encoder on top of the ViT model to enhance the temporal modeling.

C. Text-Video Retrieval.

There are increasing interests in advancing text-video retrieval performance [7], [13], [15], [33], [34]. A few works [35] proposed to improve visual semantic embedding learning for text and video joint modeling. Mithun *et al.* [35] leveraged a simple text-image embedding method [36] to improve the training strategy with hard negative mining, and incorporated multi-modal features to enrich the video representations. Dong *et al.* [19] proposed a dual-encoding network with multiple levels of features for text-video retrieval, *i.e.*, features obtained by mean pooling, bi-directional Gated Recurrent Unit and Convolution Layers. Some works focus on retrieving instances from multiple cameras [37]–[40]. Liu *et al.* [41] further utilized multiple modalities that can be extracted from videos such as speech contents and scene texts for video encoding. Miech *et al.* [12] introduced a strong joint embedding using mixture-of-expert features, which are later utilized in [13]. QB-Norm [42] is a recent work which re-normalizes query similarities to account for hubs in the embedding space. T2VLAD [7] introduced the local comparison by a shared VLAD. Different from T2VLAD [7], we enable the local alignment with transformer decoders. Besides, our model is optimized in an end-to-end manner while T2VLAD utilizes pre-extracted video features.

III. Method

A. Overview

In this section, we introduce an Align and Tell framework for text-video retrieval, which aligns text and video features by both local and global contrastive learning compounded with captioning supervision. Given the input text descriptions and videos, our goal is to encode them into a joint feature space to measure the similarity. As shown in Fig. 2, we leverage a Visual Transformer (ViT) [31] to extract the visual features for each frame. A temporal transformer encoder is used to enhance the temporal relations among the video frame features (Section III-B). For text encoder, we utilize the CLIP text model to extract contextual word features (Section III-C). After that, we feed the video frame features and word features to the local alignment module. These features are grouped by a text transformer decoder and video transformer decoder, respectively. We compute the local similarity between the corresponding local text-video features. (Section III-D). To provide word-level supervision on the local alignment and the temporal video encoder, we introduce a bidirectional captioning module (Section III-E). Finally, the training objectives and inference strategy are introduced (Section III-F).

B. Video Encoder

Our model takes the raw videos as input and encodes them into the video representations. Given a video V, we first sparsely sample N video frames $\{v_1, v_2, ..., v_N\}$ instead of using all frames. This sampling strategy can save a lot of memory and computational cost while keeping the main information of the input video. It is a precondition to performing end-to-end optimization for text-video pretraining. Specifically, we follow the sampling strategy introduced in TSN [25]. During training, we divide the video into N segments and randomly select one frame from each segment. During inference, N frames are uniformly sampled from the whole video.

For each sampled video frame, we adopt a Visual Transformer to extract the frame features. To take advantage of large-scale text-image pretraining, we utilize the ViT-B/32 from the CLIP model as the frame encoder. Concretely, we first divide each frame into non-overlapping patches with the size of 32×32 pixels and then project them into $1D$ tokens by a linear projection. A set of learnable spatial position embeddings are added to the corresponding tokens. After that, we utilize a transformer encoder E_v^s to model the interactions among the patch tokens and produce the representations. We use the output of the $[CLS]$ token as the frame representation following CLIP [18]. Therefore, for the input video frames $\{v_1, v_2, ..., v_N\}$, the output representations can denote as $Z = \{z_1, z_2, ..., z_N\}$.

Compared to image data, videos contain additional temporal information. To model the temporal relation among the video frames, we build a temporal encoder E_v^t on top of the ViT encoder. Specifically, we utilize a 4-layer transformer with temporal positional embeddings P to enable the interactions among the frame features. The encoding procedure can be formulated as follows,

$$\hat{Z} = E_v^t(Z + P). \tag{1}$$

The output features encode the temporal information of all input frames. These representations also contain local details of the input video. We further utilize them for local alignment and captioning. Meanwhile, to obtain the global representation of the whole video, we perform average pooling on the local features. The global video representation $\hat{z} = MeanPooling(\hat{Z})$.

C. Text Encoder

We leverage a pre-trained text encoder E_s from CLIP to extract the contextual word embeddings for each text description. The architecture of the text encoder is a 12-layer Transformer with eight attention heads. The input sentences are tokenized and padded to be a fixed-length sequence. The fixed-length sequence is the input to the text encoder. The text features can be computed as $W = E_s(C)$, where C is the input tokens. $W = \{w_1, w_2, ..., w_M\}$, where M is the sequence length. Following CLIP, the output from the highest layer of the transformer at the $[EOS]$ token is treated as the global representation of the text description. We denote the representation as $\hat{w}$.

D. Local Alignment Module

After extracting the video features and text features, most existing methods only calculate the similarity between the

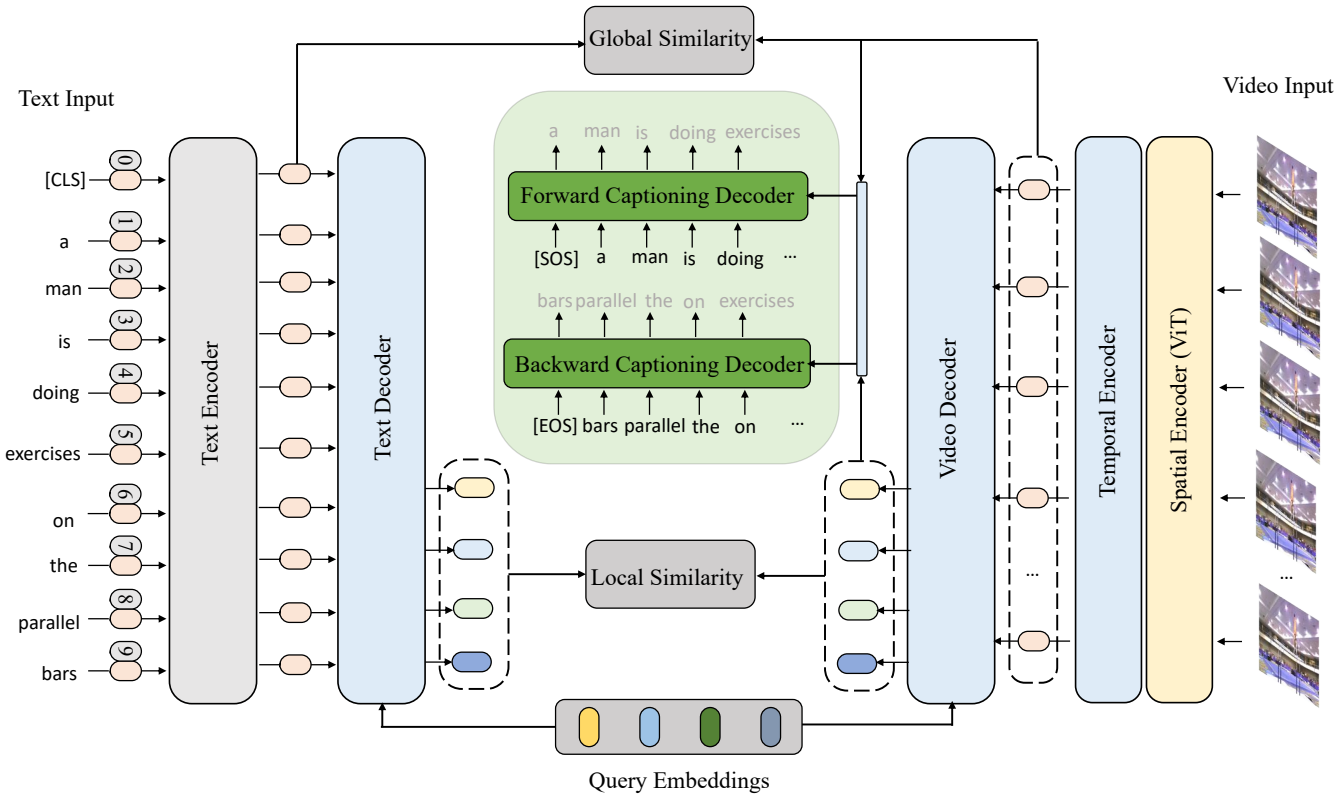

Fig. 2. The proposed Align and Tell framework for text-video retrieval. 1) As shown in the right part, given a video input, we leverage a Visual Transformer (ViT) [31] to extract the visual features for each frame. A temporal transformer encoder is used to enhance the temporal relations among the video frame features. 2) As shown on the left of the image, we utilize the CLIP text model to extract contextual word features. 3) After that, we feed the video frame features and word features to the local alignment module (see the middle of the figure). We initialize a set of query embeddings to interact with these features using a text transformer decoder and video transformer decoder, respectively. The output features contain relevant information to the corresponding query embeddings. We compute the local similarity between the corresponding local text-video features. 4) To provide word-level supervision on the local alignment and the temporal video encoder, we further introduce a bidirectional captioning module. This module is only used during training and is removed for the test phase.

global video representation $\hat{z}$ and the global text representation $\hat{w}$. However, the global comparison overlooks the local details in both videos and text descriptions. The key point is to perform more fine-grained alignment based on the local features. The local video features $\hat{Z} = \{z_1; z_2; \ldots; z_N\}$ and the local text features $W = \{w_1; w_2; \ldots; w_M\}$ have different length. Besides, the number of the local text feature M is not determinate. Therefore, the direct comparison between the two types of local features is not feasible. To address this problem, we propose an automatic decoding method with a fixed set of queries. These query embeddings interact with the local features and extract the most useful information for cross-modal matching. Formally, we denote the query embeddings as $Q = \{q_1; q_2; \ldots; q_K\}$, where K is the number of queries.

We build two separate transformer decoders to perform the local alignment for the video and text, respectively. The video decoder D_v take the queries Q and the local video features $\hat{Z}$ as input. The output video representations are produced as follows,

$$\overline{Z} = D_v(Q; \hat{Z}); \tag{2}$$

where $\overline{Z} = \{\overline{z}_1; \overline{z}_2; \ldots; \overline{z}_K\}$ has the same size as the query embeddings. The local text features W are processed in a similar way:

$$\overline{W} = D_s(Q; W); \tag{3}$$

where D_s is the text decoder, $\overline{W} = \{\overline{w}_1; \overline{w}_2; \ldots; \overline{w}_K\}$ denotes the output text features. In the decoding procedure, each query embedding interacts with the video frame features and the contextual word features. The outputs corresponding to a certain query contain the relevant information from the two modalities. Therefore, the output video feature and a text feature for the same query share similar semantic content. We can measure the text-video similarity for each query to achieve local alignment. The overall local text-video similarity is an average of the K similarities for all queries:

$$s^{local} = \frac{1}{K} \sum_{i}^{K} \frac{\overline{z}_i^{\top} \overline{w}_i}{\|\overline{z}_i\| \|\overline{w}_i\|}, \tag{4}$$

here we use the cosine distance to measure the similarity between the text-video features.

E. Bi-directional Captioning Head

To enhance the video representation and enable the local alignment, we introduce a temporal encoder E_v^t and two decoders D_v, D_s. These parameters can not initialize from the CLIP pre-trained weights. According to the previous work

CLIP4clip [4], these randomly initialized modules are hard to be optimized by the commonly used contrastive objectives. To address this problem, we introduce an auxiliary task to provide additional supervision signals. Inspired by the success in text-image pretraining [23], we design a bi-directional video captioning head to support the text-video pretraining.

The captioning head accepts the video features and predicts the corresponding caption token by token. The captioning head is used to supervise the randomly initialized modules, so we first concatenate the features produced by the temporal encoder and the outputs from the video decoder as the candidate video features:

$$Z^0 = Concat(\hat{Z}, \overline{Z}); \tag{5}$$

where Z^0 contains $N + K$ video feature vectors. The paired text description can denote as $C = \{c_0, c_1, ..., c_{M+1}\}$, where $c_0 = [SOS]$ and $c_{M+1} = [EOS]$ are the special tokens indicating the start and end of sentence. As shown in Fig.2, the bidirectional captioning head consists of a forward model and a backward model. Following the recent advances in the visual-language area, we use two 2-layer transformer decoders to implement the forward captioning model and the backward captioning model, respectively. The transformer decoder propagates the information among the text tokens and also fuses the text features and the video features. Each layer of the decoder consists of a masked text self-attention layer, followed by a cross-attention layer that enables the text to attend to the video features and finally a feed-forward layer. We use independent word embedding for converting the input word to a token feature. The weights of the word embedding are initialized from the word embedding of CLIP. During training, the captioning model predicts the captions token by token, which only depends on the past predictions and the video features. The module is jointly trained with the overall framework to maximize the log-likelihood of the correct caption tokens:

$$\begin{aligned} L_{cap} = & \sum_{t=1}^{M+1} log(p(c_t | c_{t-1}, ..., c_0, Z^0, \theta_{fwd})) \\ & + \sum_{t=0}^{M} log(p(c_t | c_{t+1}, ..., c_{M+1}, Z^0, \theta_{bwd})) \end{aligned} \tag{6}$$

where $p(c_t | c_{t-1}, ..., c_0, Z^0, \theta)$ denotes the output probability of a decoder model parameterized by θ. θ_{fwd} is the parameters of the forward model, and θ_{bwd} is the parameters of the backward model.

F. Training and Inference Strategy

As we mentioned before, we calculate the local similarity between text-video pairs by Eq.4. As complementary to local alignment, we also compute the global similarity between the global video feature $\hat{z}$ and the global text feature $\hat{w}$:

$$s^{global} = \frac{\hat{z}^T \hat{w}}{\|\hat{z}\| \|\hat{w}\|}; \tag{7}$$

The final similarity is the average of the local similarity and the local similarity:

$$s = \frac{1}{2}(s^{global} + s^{local}).$$

Given a batch of B text-video pairs, we calculate a symmetric contrastive loss to enforce the paired text-video samples to be closer than the unpaired samples in the feature space:

$$L_{t2v} = \frac{1}{B} \sum_{i}^{B} log(\frac{exp(s(V_i, C_i))}{\sum_{j}^{B} exp(s(V_j, C_i))}); \tag{8}$$

$$L_{v2t} = \frac{1}{B} \sum_{i}^{B} log(\frac{exp(s(V_i, C_i))}{\sum_{j}^{B} exp(s(V_i, C_j))}); \tag{9}$$

where L_{t2v} is the text-to-video loss and L_{v2t} is the video-to-text loss, $s(V_i, C_i)$ is the final similarity between the video V_i and the text description C_i. The overall training objective is the weighted sum of the contrastive loss and the captioning loss:

$$L = L_{t2v} + L_{v2t} + \lambda L_{cap}; \tag{10}$$

where λ is a hyper-parameter to adjust the weight of the captioning loss.

In the test phase, we remove the captioning head and only use the encoders E_v^t, E_v^s, E_s and decoders D_s, D_v to extract the global text-video features and the local text-video features. The local similarity and the global similarity are calculated between the query features and the gallery features. The final rank is based on the average of the global similarity and the local similarity.

IV. Experiment

A. Dataset

We conduct text-to-video and video-to-text retreival on four standard benchmarks, *i.e.*, MSRVTT [8], MSVD [11], ActivityNet Captions [20] and LSMDC [10].

MSRVTT [8] contains 10,000 videos. These videos are collected from YouTube using 257 queries from a commercial video search engine. We evaluate the performance on three splits. For the "1k-A" split, the train and test are split as introduced in [43]. The "1k-B" split is obtained following [12]. Both splits use 9,000 videos for training, and the remaining 1,000 videos are used for testing.

MSVD [11] contains 1,970 videos, each with a length that ranges from one to 62 seconds. Train, validation and, test splits contain 1200, 100, and 670 videos, respectively. Each video has approximately 40 associated sentences in English.

ActivityNet Captions [20] It consists of 20,000 videos. Each video is densely annotated with multiple sentence descriptions. We follow the setting in [13] to concatenate all the captions to a paragraph and evaluate the video-paragraph retrieval performance on the val1 split.

LSMDC [10] consists of 118,081 short video clips. The videos are extracted from 202 long movies. The validation set contains 7,480 videos, while the test set contains 1000 videos. The clip length ranges from 2 to 30 seconds.

Evaluation Metrics. We report the results with the standard video retrieval metrics, *i.e.*, Rank K (R@K, higher is better), Median Rank (MdR, lower is better). We report R@1, R@5, and R@10 following [41].

TABLE I
THE COMPARISON WITH THE STATE-OF-THE-ART METHODS ON THE MSRVTT [8] DATASET. **1K-A** INDICATES THE TEST SET OF 1000 PAIRS AND THE TRAINING SET OF 9K USED BY [43], **1K-B** IS THE SPLIT THAT THE TRAINING SET CONTAINS 7K PAIRS INTRODUCED IN [12], CLIP-STRAIGHT [18] INDICATES USING THE ORIGINAL CLIP MODEL TO EVALUATE THE ZERO-SHOT RETRIEVAL WITH TEMPORAL MEAN POOLING. CLIP4CLIP-MEANP AND CLIP4CLIP-SEQTRANSF INDICATE THE VERSION WITH MEAN-POOLING AND SEQUENCE TRANSFORMER FOR TEMPORAL AGGREGATION.

Method	Split	Text ! Video				Video ! Text			
		R@1″	R@5″	R@10″	MdR#	R@1″	R@5″	R@10″	MdR#
JSFusion [43]	1k-A	10.2	31.2	43.2	13	-	-	-	-
HT [9]	1k-A	14.9	40.2	52.8	9	-	-	-	-
CE [41]	1k-A	20.9	48.8	62.4	6	20.6	50.3	64.0	5.3
MMT [13]	1k-A	24.6	54.0	67.1	4	24.4	56.0	67.8	4
MMT + HT pretrain [13]	1k-A	26.6	57.1	69.6	4	27.0	57.5	69.7	3.7
SUPPORT-SET [44]	1k-A	27.4	56.3	67.7	3	26.6	55.1	67.5	3
T2VLAD [7]	1k-A	29.5	59.0	70.1	4	31.8	60.0	71.1	3
CLIP-straight [18]	1k-A	31.2	53.7.	64.2	4	27.2	51.7	62.6	5.0
FROZEN [16]	1k-A	31.0	59.5	70.5	3	-	-	-	-
MDMMT [45]	1k-A	38.9	69.0	79.7	2	-	-	-	-
CLIP4Clip-meanP [4]	1k-A	43.1	70.4	80.8	2	43.1	70.5	81.2	2
CLIP4Clip-seqTransf [4]	1k-A	44.5	71.4	81.6	2	42.7	70.9	80.6	2
QB-Norm [42]	1k-A	47.2	73.0	83.0	2	-	-	-	-
Baseline ViT-B/32	1k-A	44.2	71.0	81.2	2	42.3	70.2	80.5	2
Align&Tell ViT-B/32	1k-A	**45.2**	**73.0**	**82.9**	**2**	**43.4**	**70.9**	**81.8**	**2**
Baseline ViT-B/16	1k-A	45.8	71.3	81.4	2	43.2	71.3	82.0	2
Align&Tell ViT-B/16	1k-A	**47.4**	**74.3**	**84.1**	**2**	**45.3**	**73.5**	**83.7**	**2**
MEE [12]	1k-B	13.6	37.9	51.0	10	-	-	-	-
MEE-COCO [12]	1k-B	14.2	39.2	53.8	9	-	-	-	-
CE [41]	1k-B	18.2	46.0	60.7	7	18.0	46.0	60.3	6.5
MMT [13]	1k-B	20.3	49.1	63.9	6	21.1	49.4	63.2	6
T2VLAD [7]	1k-B	26.1	54.7	68.1	4	26.7	56.1	70.4	4
ClipBERT [17]	1k-B	22.0	46.8	59.9	6	-	-	-	-
Clip4Clip-seqTransf [4]	1k-B	42.0	68.6	78.7	2	-	-	-	-
Clip4Clip-meanP [4]	1k-B	42.1	71.9	81.4	2	-	-	-	-
Baseline ViT-B/32	1k-B	42.0	69.1	80.3	2	40.8	68.6	77.4	2
Align&Tell ViT-B/32	1k-B	**43.2**	**72.3**	**81.5**	**2**	**41.6**	**69.3**	**78.6**	**2**
Baseline ViT-B/16	1k-B	43.7	72.6	81.4	2	42.3	69.8	79.2	2
Align&Tell ViT-B/16	1k-B	**45.1**	**73.4**	**82.1**	**2**	**43.5**	**70.6**	**80.3**	**2**

B. Implementation Details.

We implement our models with PyTorch [46] We initial the text encoder and video encoder with CLIP (ViT-B/32 or ViT-B/16) [18] and reuse the similar parameters in CLIP to initialize the new modules such as the word embedding layer of the captioning head and the temporal positional embeddings. The video temporal encoder has 4 layers, and the hidden dimension is 512. The text decoder and the video decoder have the same architecture consisting of 2-layer transformer decoder. Following [4], we train the model with the Adam optimizer [47]. The learning rate is decayed with a cosine schedule. We set the initial learning rate to 1e-7 for the text encoder and the video encoder. For the new modules such as the decoders and the captioning head, we the learning rate to 5e-4. We sparsely sample 12 frames for each video. The caption length is 64. For ActivityNet Captions, we concatenate all the descriptions and conduct paragraph-video retrieval, and the frame length is 64. The batch size is reduced to 64 to save the GPU memory for ActivityNet Captions. We set the weight of the captioning loss to 0.1 for all experiments. The model is optimized for 5 epochs. The batch size is 128. All experiments are conducted on NVIDIA Tesla V100 GPUs.

C. Comparison to the State of the Art

MSRVTT. The results on MSRVTT are shown in Table I. We consistently improve the state-of-the-art on text-to-video retrieval and video-to-text retrieval across all three splits. CE [41], MMT [13] and T2VLAD [7] are proposed to perform text-video retrieval using multi-modal features. With the powerful large-scale text-image pretraining, zero-shot retrieval of CLIP achieves superior performance than most of the methods based on the fixed video features. FROZEN [16] and SUPPORT-SET [44] pretrain the model on large-scale text-video dataset. CLIP4Clip [4] is a recent work which finetunes the CLIP model for the text-video retreival task. QB-Norm [42] is a recent work which re-normalizes query similarities to account for hubs in the embedding space. It achieved the best performance in the compared methods. The baseline model is implemented with the CLIP backbone and a video temporal encoder. Our Align&Telll model outperforms the baselines with ViT-B/32 or ViT-B/16. For text-to-video retrieval, we outperform CLIP4Clip-meanP [4] with 1.1% gain on the R@1 metric on the 1k-B split (43.2% vs. 42.1%). Notably, Clip4Clip-seqTransf achieves lower retrieval accuracy than Clip4Clip-meanP on the split with less training data. The temporal transformer is hard to be optimized when the training data is not sufficient. In contrast, our method

TABLE II
THE COMPARISON WITH THE STATE-OF-THE-ART METHODS ON THE MSVD [11] DATASET. CLIP4CLIP-MEANP [18] AND CLIP4CLIP-SEQTRANSF [18] INDICATE THE MODELS WITH MEAN-POOLING AND SEQUENCE TRANSFORMER FOR TEMPORAL AGGREGATION.

Method	Text → Video				Video → Text			
	R@1 ↑	R@5 ↑	R@10 ↑	MdR ↓	R@1 ↑	R@5 ↑	R@10 ↑	MdR ↓
VSE [36]	12.3	30.1	42.3	14	34.7	59.9	70.0	3
CE [41]	19.8	49.0	63.8	6	-	-	-	-
SSML [48]	20.3	49.0	63.3	6	-	-	-	-
SUPPORT-SET [44]	28.4	60.0	72.9	4	-	-	-	-
FROZEN [16]	33.7	64.7	76.3	3	-	-	-	-
CLIP [18]	37.0	64.1	73.8	3	59.9	85.2	90.7	1
CLIP4Clip-seqTransf [4]	45.2	75.5	84.3	2	62.0	87.3	92.6	1
CLIP4Clip-meanP [4]	46.2	76.1	84.6	2	56.6	79.7	84.3	1
QB-Norm [42]	47.6	77.6	86.1	2	-	-	-	-
Baseline ViT-B/32	45.4	75.2	84.1	2	60.1	86.0	90.8	1
Align&Tell ViT-B/32	**47.1**	**77.0**	**85.6**	**2**	61.8	**87.5**	**92.7**	**1**
Baseline ViT-B/16	47.7	77.5	86.0	2	62.3	84.4	89.6	1
Align&Tell ViT-B/16	**49.3**	**79.1**	**87.9**	**2**	**65.2**	**88.6**	**93.1**	**1**

TABLE III
THE COMPARISONS WITH THE STATE-OF-THE-ART METHODS ON THE ACTIVITYNET CAPTIONS [20] DATASET. CLIP4CLIP-MEANP [18] AND CLIP4CLIP-SEQTRANSF [18] INDICATE THE MODELS WITH MEAN-POOLING AND SEQUENCE TRANSFORMER FOR TEMPORAL AGGREGATION.

Method	Text → Video				Video → Text			
	R@1 ↑	R@5 ↑	R@50 ↑	MdR ↓	R@1 ↑	R@5 ↑	R@50 ↑	MdR ↓
FSE [49]	18.2	44.8	89.1	7	16.7	43.1	88.4	7
CE [41]	18.2	47.7	91.4	6	17.7	46.6	90.9	6
HSE [49]	20.5	49.3	-	-	18.7	48.1	-	-
MMT [13]	22.7	54.2	93.2	5	22.9	54.8	93.1	4.3
ClipBERT [17]	21.3	49.0	-	6	-	-	-	-
TT-CE+ [50]	23.5	57.2	96.1	4	23.0	56.1	95.8	4
T2VLAD [7]	23.7	55.5	93.5	4	24.1	56.6	94.1	4
CLIP4Clip-meanP [4]	40.5	72.4	98.1	2	-	-	-	-
CLIP4Clip-seqTransf [4]	40.5	72.4	98.2	2	-	-	-	-
Baseline ViT-B/32	40.8	72.5	97.9	2	42.7	72.5	98.1	2
Align&Tell ViT-B/32	**42.6**	**73.8**	**98.7**	**2**	**43.5**	**73.6**	**98.3**	**2**
Baseline ViT-B/16	44.0	74.7	98.4	2	43.8	74.9	98.5	2
Align&Tell ViT-B/16	**44.9**	**75.4**	**98.8**	**2**	**45.1**	**76.3**	**98.6**	**2**

consistently outperforms the Clip4Clip-meanP on all splits. This is because the auxiliary captioning head in our model can provide additional supervision, which is especially important for the modules without CLIP initialization.

MSVD. MSVD has fewer training text-video pairs compared with MSRVTT. As shown in Table II, the original CLIP model without finetuning achieves quite good performance. The finetuned model CLIP4Clip further improves the retrieval accuracy. Similar to the results on MSRVTT 1K-B split, the model CLIP4Clip-seqTransf with more complex temporal modeling achieves lower performance. Our model outperforms all state-of-the-art methods on text-to-video retrieval. It indicates that our method is effective and robust even only with small-scale training data. Notably, our Align&Tell model outperforms the recent state-of-the-art method QB-Norm [42] by 1.7% on Rank@1 accuracy for the text-video retrieval.

ActivityNet Captions. ActivityNet Captions consists of long videos and the captions contain several sentences. We concatenate these descriptions and perform paragraph-video retrieval following the standard setting. The results on this dataset are shown in Table III. The compared baselines include CE [41], MMT [13], TT-CE+ [50] and CLIP4Clip [4]. HSE [49] leverages a hierarchical sequence embedding and MMT incorporates multi-layer transformers for strong video feature learning. We consistently improve the state-of-the-art on all benchmark metrics, which demonstrates the effectiveness of our Align&Tell framework on the long-term text and video modeling.

LSMDC. The LSMDC dataset is collected from movies. The results are shown in Table IV. We observe consistent improvements over CLIP4Clip [4]. The results show that our method is capable of dealing with different types of video from various domains and demonstrate the benefits of our local alignment module and the auxiliary captioning head in cross-modal retrieval tasks.

D. Ablation Studies

1) The effectiveness of the captioning head.: We design an auxiliary captioning module during training to enhance the video representation learning. The captioning head is only applied during training and can be removed in the test phase. To investigate the effectiveness of this module, we directly build the captioning head on top of the video encoder. The input for the captioning module is the video features produced by the temporal encoder and the word tokens. Besides, we

TABLE IV
THE COMPARISON WITH THE STATE-OF-THE-ART METHODS ON THE LSMDC DATASET [10]. CLIP4CLIP-MEANP [18] AND CLIP4CLIP-SEQTRANSF [18] INDICATE THE MODELS WITH MEAN-POOLING AND SEQUENCE TRANSFORMER FOR TEMPORAL AGGREGATION.

Method	Text ! Video				Video ! Text			
	R@1 "	R@5 "	R@10 "	MdR #	R@1 "	R@5 "	R@10 "	MdR #
JSFusion [43]	9.1	21.2	34.1	36	-	-	-	-
MEE [12]	9.3	25.1	33.4	27	-	-	-	-
MEE-COCO [12]	10.1	25.6	34.6	27	-	-	-	-
CE [41]	11.2	26.9	34.8	25.3	-	-	-	-
CLIP [18]	11.3	22.7	29.2	56.5	-	-	-	-
MMT [13]	13.2	29.2	38.8	21	12.1	29.3	37.9	22.5
T2VLAD [7]	14.3	32.4	42.2	16	14.2	33.5	41.7	17
TT-CE+ [50]	17.2	36.5	46.3	13.7	17.5	36.0	45.0	14.3
QB-Norm [42]	17.8	37.7	47.6	12.7	-	-	-	-
CLIP4Clip-meanP [4]	20.7	38.9	47.2	13	-	-	-	-
CLIP4Clip-seqTransf [4]	22.6	41.0	49.1	11	-	-	-	-
Baseline ViT-B/32	21.8	40.3	48.7	12	20.7	39.1	49.0	12
Align&Tell ViT-B/32	**23.1**	**41.2**	**49.6**	**11**	**21.3**	**40.2**	**50.1**	**11**
Baseline ViT-B/16	23.5	44.8	54.7	9	22.3	42.5	52.9	9
Align&Tell ViT-B/16	**23.9**	**45.7**	**55.6**	**8**	**24.2**	**45.1**	**53.8**	**8**

TABLE V
THE ABLATION STUDIES ON THE MSRVTT [8] 1K-A DATASET TO INVESTIGATE THE EFFECTIVENESS OF THE LOCAL ALIGNMENT MODULE AND THE CAPTIONING HEAD.

Local Alignment	Captioning Head	Text ! Video			Video ! Text		
		R@1"	R@5"	R@10"	R@1"	R@5"	R@10"
-	-	44.2	71.0	81.2	42.3	70.2	80.5
✓	-	44.3	72.6	82.3	42.5	70.6	80.7
-	✓	45.1	71.6	81.7	43.1	70.3	80.6
✓	✓	**45.2**	**73.0**	**82.9**	**43.4**	**70.9**	**81.8**

TABLE VI
THE RESULTS ON THE MSRVTT [8] 1K-A DATASET WITH DIFFERENT SETTINGS FOR THE CAPTIONING HEAD.

Method	Text ! Video			Video ! Text		
	R@1"	R@5"	R@10"	R@1"	R@5"	R@10"
Forward captioning	44.9	72.4	82.3	42.9	70.7	81.2
Backward captioning	44.6	72.8	82.7	42.9	70.5	81.0
Bidirectional captioning	**45.2**	**73.0**	**82.9**	**43.4**	**70.9**	**81.8**

TABLE VII
THE RESULTS ON THE MSRVTT [8] 1K-A DATASET WITH DIFFERENT INITIALIZATION FOR THE CAPTIONING WORD EMBEDDING. "INDEPENDENT" INDICATE THE WORD EMBEDDING LAYER FOR THE CAPTIONING HEAD IS INDEPENDENT OF THE CLIP WORD EMBEDDING LAYER.

Method	Text ! Video			Video ! Text		
	R@1"	R@5"	R@10"	R@1"	R@5"	R@10"
Independent (Random Initia.)	40.3	67.9	76.1	39.5	67.3	77.1
Shared with CLIP	42.2	69.7	78.4	40.7	68.4	78.0
Independent (CLIP Initia.)	**45.2**	**73.0**	**82.9**	**43.4**	**70.9**	**81.8**

set up a baseline model that only contains the video encoder and the text encoder. The baseline model is trained with the global contrastive loss. As shown in Table V, the model with captioning head outperforms the baseline model. Without any additional computational cost for inference, the model trained with the captioning loss improves the Rank-1 accuracy by 0.9% over the baseline. These results demonstrate that the captioning task enables the interaction between texts and videos. These interactions provide more fine-grained supervision, which is complementary to standard contrastive learning.

2) The effectiveness of the local alignment.: To make full use of the fine-grained video features and the word-level text features, we propose a local alignment module based on the

TABLE VIII
TRAINING AND INFERENCE TIME ON THE MSRVTT [8] 1K-A DATASET FOR VIT-B/32. TRAINING SPEED INDICATES THE TIME OF ONE FORWARD-BACKWARD ITERATION ON 4 TESLA V100 GPUS FOR BATCH SIZE 128. TESTING TIME INDICATES THE INFERENCE TIME PER VIDEO-TEXT PAIR DURING EVALUATION ON A TESLA V100 GPU.

Method	Training Speed/ms	Testing Speed/ms	T ! V R@5"
Baseline	779	41.6	71.0
Tell&Align	931	45.3	73.0

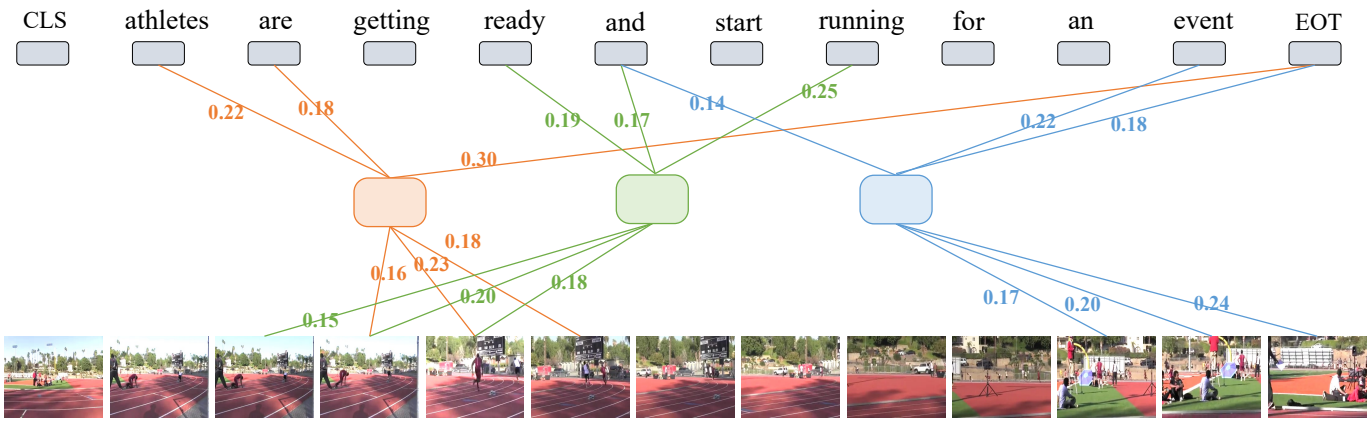

Fig. 3. Visualization of attention weights in the local alignment module on the MSRVTT [8] 1k-A test set. We select three query embeddings and keep the Top-3 weights for better visualization.

transformer decoder architecture. The local video features and text features are grouped by a fixed set of query embeddings. After that, the cross-modal features with similar semantic topics can be compared with each other accordingly. As shown in Table V, the model with the local alignment module achieves better results than the baseline model, which only contains the global alignment. However, the improvement on the Rank-1 accuracy is not clear. This is because the local module introduces additional parameters that can not be initialized from the CLIP weights. The naive contrastive learning can not provide sufficient supervision signals for the local alignment. After introducing the captioning loss, the performance is higher than the baseline by a clear margin. The results are also higher than the model only with the captioning loss. This proves that the local alignment is important for text-video retrieval, and this module should be trained with additional supervision.

3) The weight of the captioning loss.: We investigate the weight of the captioning loss on the MSRVTT [8] dataset. As shown in Figure 4, the Rank@1 accuracy of the model is first improved with the increase of the weight . When is too large, the performance of the model goes down. This is because the value of the captioning loss is an order of magnitude larger than the value of the contrastive loss. Too large weakens the effect of contrastive learning. According to the above analysis, we set = 0:1 in our experiments.

4) The number of queries.: To perform local alignment, we utilize a set of query embeddings to group the local features. We investigate the influence of the number of queries. Fewer queries lead to fewer cross-modal groups. Too few queries can not extract the local details effectively. When the number of queries equals one, the model only performs the global comparison. On the contrary, too many query embeddings increase the difficulty of optimization and the computational cost. As shown in Figure 4, the model with 16 queries slightly outperforms the model with 8 queries and the model 32 queries. The model with 4 queries achieves similar performance compared to the model without the local alignment module. Therefore, we set the number of queries to 16 for all experiments in the other tables.

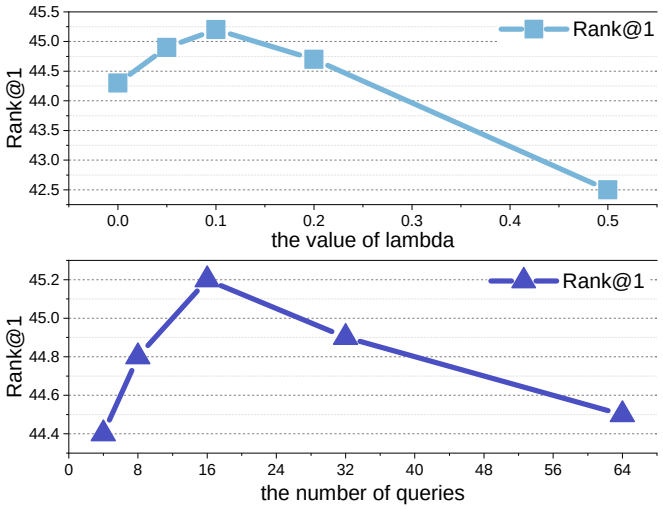

Fig. 4. Results on the MSRVTT [8] 1k-A dataset with the different number of the queries in the local alignment module and different weights of the captioning loss.

5) Different settings for the captioning head: We investigate the variants of the captioning head. The results are shown in Table VI and Table VII. The model with a bidirectional captioning head achieves higher performance than the model with single direction captioning. These results indicate that bidirectional captioning can provide more supervision for text-video pretraining. We generate word tokens for the captioning head by a word embedding layer. In our model, the word embedding layer is independent of the CLIP word embedding, but we use the CLIP weights to initialize the layer. If we use the shared word embedding with CLIP or randomly initialize the layer, the performance drops clearly. These results demonstrate the effectiveness of our initialization strategy.

6) Training and inference speed: We evaluate the training and testing speed of the baseline model and our Tell&Align model. The results are shown in Table VIII. Our Tell&Align model has two additional modules compared to the baseline.

The captioning head introduces more computational cost and provides word-level supervision. The local alignment module is lightweight and provides fine-grained comparisons. The two modules are both used during training so the speed of our model is slightly slower than the baseline. The captioning head is removed during testing and The local alignment module only increases limited inference time.

E. Visualization

We visualize the attention weights in the local alignment module to investigate the effectiveness of this design. As shown in Figure 3, the aggregated text features and video features on the same query embedding share similar semantic meanings. For example, the text feature with the largest attention weights on the second query embedding is "running", and the video frames with the largest attention weights on the second query embedding also contain the semantic meaning of "running". These results demonstrate our assumptions about the local alignment. However, the words with a higher frequency of occurrence tend to have high attention weights, such as "are" and "and", which does not benefit the text-video alignment intuitively. We will try to solve this problem in future work.

V. Conclusion

In this paper, we investigate the text-video retrieval task. We propose an Align and Tell framework and optimize the model in an end-to-end manner. Two aspects distinguish our method from existing works. First, we introduce a local alignment module to aggregate the temporal video features and the word-level text features into a fixed number of groups, and local contrastive learning is performed to align the local features. This local alignment is complementary to the global comparison. Second, we design a captioning head to provide more supervision signals during training. This word-level supervision can enhance the visual presentation and alleviate the cross-modal gap. The captioning head can be removed during inference to save the computational cost. Extensive experimental results demonstrate that our Tell and Align model can achieve state-of-the-art bi-directional retrieval performance on four standard text-video retrieval datasets.

References

[1] B. Manaris, "Natural language processing: A human-computer interaction perspective," in *Advances in Computers*. Elsevier, 1998, vol. 47, pp. 1–66.

[2] Y. H. Ling SHEN, Richang HONG, "Advance on large scale near-duplicate video retrieval," *Frontiers of Computer Science*, vol. 14, no. 5, p. 145702, 2020.

[3] Z. Zheng, L. Zheng, M. Garrett, Y. Yang, M. Xu, and Y.-D. Shen, "Dual-path convolutional image-text embeddings with instance loss," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 2, pp. 1–23, 2020, doi:10.1145/3383184.

[4] H. Luo, L. Ji, M. Zhong, Y. Chen, W. Lei, N. Duan, and T. Li, "Clip4clip: An empirical study of clip for end to end video clip retrieval," *arXiv preprint arXiv:2104.08860*, 2021.

[5] Y. Yang, Y.-T. Zhuang, F. Wu, and Y.-H. Pan, "Harmonizing hierarchical manifolds for multimedia document semantics understanding and cross-media retrieval," *IEEE Transactions on Multimedia (TMM)*, vol. 10, no. 3, pp. 437–446, 2008.

[6] X. Yang, T. Zhang, and C. Xu, "Text2video: An end-to-end learning framework for expressing text with videos," *IEEE Transactions on Multimedia (TMM)*, vol. 20, no. 9, pp. 2360–2370, 2018.

[7] X. Wang, L. Zhu, and Y. Yang, "T2vlad: global-local sequence alignment for text-video retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5079–5088.

[8] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *CVPR*, 2016.

[9] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips," in *ICCV*, 2019.

[10] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele, "A dataset for movie description," in *CVPR*, 2015.

[11] D. Chen and W. B. Dolan, "Collecting highly parallel data for paraphrase evaluation," in *ACL-HLT*, 2011.

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

[13] V. Gabeur, C. Sun, K. Alahari, and C. Schmid, "Multi-modal transformer for video retrieval," in *ECCV*, 2020.

[14] J. Dong, X. Li, and C. G. Snoek, "Predicting visual features from text for image and video caption retrieval," *IEEE Transactions on Multimedia (TMM)*, 2018.

[15] H. Fan and Y. Yang, "Person tube retrieval via language description," in *AAAI*, 2020.

[16] M. Bain, A. Nagrani, G. Varol, and A. Zisserman, "Frozen in time: A joint video and image encoder for end-to-end retrieval," in *ICCV*, 2021.

[17] J. Lei, L. Li, L. Zhou, Z. Gan, T. L. Berg, M. Bansal, and J. Liu, "Less is more: Clipbert for video-and-language learning via sparse sampling," in *CVPR*, 2021.

[18] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021.

[19] J. Dong, X. Li, C. Xu, S. Ji, Y. He, G. Yang, and X. Wang, "Dual encoding for zero-example video retrieval," in *CVPR*, 2019.

[20] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. Carlos Niebles, "Dense-captioning events in videos," in *ICCV*, 2017.

[21] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman, "End-to-end learning of visual representations from uncurated instructional videos," in *CVPR*, 2020.

[22] Y. Yang, Y. Zhuang, and Y. Pan, "Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies," *Frontiers of Information Technology & Electronic Engineering*, vol. 22, no. 12, pp. 1551–1558, 2021.

[23] K. Desai and J. Johnson, "Virtex: Learning visual representations from textual annotations," in *CVPR*, 2021.

[24] A. Miech, J.-B. Alayrac, I. Laptev, J. Sivic, and A. Zisserman, "Thinking fast and slow: Efficient text-to-visual retrieval with transformers," in *CVPR*, 2021.

[25] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV*, 2016.

[26] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NeurIPS*, 2014.

[27] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[28] X. Wang, L. Zhu, Y. Wu, and Y. Yang, "Symbiotic attention for egocentric action recognition with object-centric alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[29] L. Zhu, H. Fan, Y. Luo, M. Xu, and Y. Yang, "Temporal cross-layer correlation mining for action recognition," *IEEE Transactions on Multimedia (TMM)*, pp. 1–1, 2021.

[30] Y. Ding, H. Fan, M. Xu, and Y. Yang, "Adaptive exploration for unsupervised person re-identification," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 1, pp. 1–19, 2020.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[32] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *ICML*, 2021.

[33] L. Zhu and Y. Yang, "Actbert: Learning global-local video-text representations," in *CVPR*, 2020.

[34] S. Zhao, L. Zhu, X. Wang, and Y. Yang, "Centerclip: Token clustering for efficient text-video retrieval," in *SIGIR*, 2022.