EURASIP Journal on Advances
in Signal Processing

RESEARCH **Open Access**

Level-wise aligned dual networks for text–video retrieval

Qiubin Lin[1,2], Wenming Cao[1,2] and Zhiquan He[1,2*]

*Correspondence:
zhiquan@szu.edu.cn

[1] College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China
[2] Guangdong Multimedia Information Service Engineering Technology Research Center, Shenzhen University, Shenzhen, China

Abstract

The vast amount of videos on the Internet makes efficient and accurate text–video retrieval tasks increasingly important. The current methods leverage a high-dimensional space to align video and text for these tasks. However, a high-dimensional space cannot fully use different levels of information in videos and text. In this paper, we put forward a method called level-wise aligned dual networks (LADNs) for text–video retrieval. LADN uses four common latent spaces to improve the performance of text–video retrieval and utilizes the semantic concept space to increase the interpretability of the model. Specifically, LADN first extracts different levels of information, including global, local, temporal, and spatial–temporal information, from videos and text. Then, they are mapped into four different latent spaces and one semantic space. Finally, LADN aligns different levels of information in various spaces. Extensive experiments conducted on three widely used datasets, including MSR-VTT, VATEX, and TRECVID AVS 2016-2018, demonstrate that our proposed approach is superior to several state-of-the-art text–video retrieval approaches.

Keywords: Text–video retrieval, Level-wise aligned mechanism, Semantic space, Latent space

1 Introduction

1.1 Background and significance

Video has become one of the most popular media because it can capture dynamic events and naturally attract human sight and hearing. Furthermore, the explosion of videos on the Internet has made efficiently and accurately searching for videos a significant challenge [1].

This paper focuses on the tasks of video-to-text and text-to-video retrieval. The task of video-to-text retrieval pays attention to finding the text candidate that best describes the video query among a collection of text candidates. Additionally, text-to-video retrieval means that given a query in the form of text modality, the aim is to search for the videos best described by the text query (See Fig. 1). In practice, a ranking list of all the video candidates is returned for each text query, and the video corresponding to the text query is ranked as high as possible.

Traditional methods [2–6] for the retrieval problem mainly focus on semantic concept search, where semantic concepts are pre-defined. However, since semantic concepts

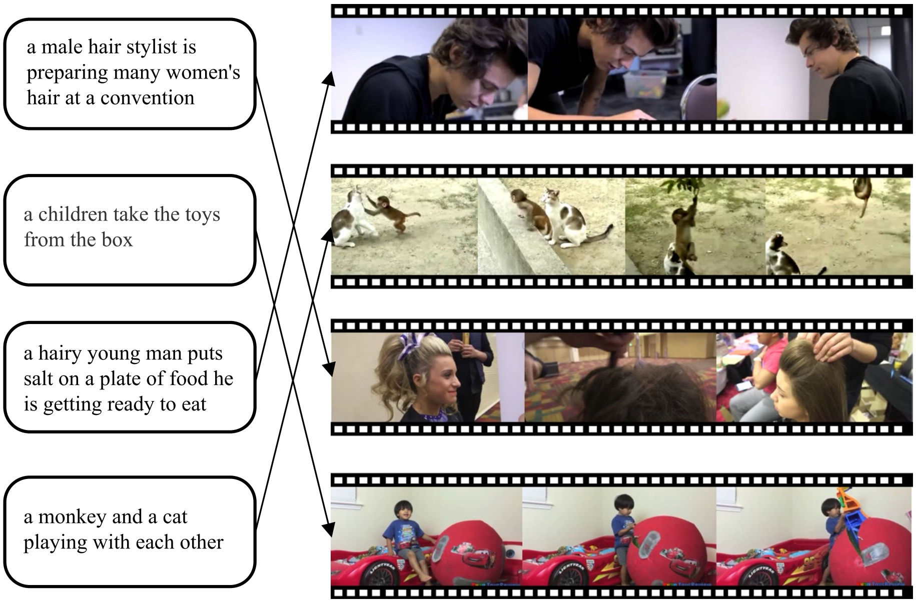
a male hair stylist is preparing many women's hair at a convention
a children take the toys from the box
a hairy young man puts salt on a plate of food he is getting ready to eat
a monkey and a cat playing with each other

Fig. 1 Illustration of text-to-video retrieval: given a text query, retrieve the corresponding video from the database

are limited and unstructured, they cannot accurately search for different fine-grained contents and utilize temporal information. For example, "a dog chases a cat" and "a cat chases a dog" will have the same semantic concepts, while the order of objects in the caption is potentially significant. In addition, a query of "a black dog chases a white cat" is nearly impossible to obtain satisfied retrieval results for a semantic-based video retrieval method. Although the semantic-based method has certain interpretability, how to specify a set of relevant and detectable semantic concepts for video and text features remains unsolved.

To solve the limitations of semantic-based methods, researchers pay more attention to utilizing original sentences that contain rich contextual information than semantic concepts. At present, the main methods for text–video cross-modal retrieval map video and text into a common latent space, where the cross-modal similarity can be measured.

For video representations, a common method is to first extract frame features from the video through a pre-trained convolutional neural network (CNN) model and then combine them by max pooling [7, 8], mean pooling [9, 10], recurrent neural network (RNN) [11, 12], NetVLAD [13], or self-attention mechanisms [14, 15].

For text representations, bag of words remains popular [16, 17], while deep networks are in increasing use. For each word of a sentence, a dense vector is first generated by multiplying its one-hot vector with a pre-trained word embedding matrix. Then, they are combined to generate a sentence-level representation by NetVLAD [8, 13], max pooling [7], Fisher Vector [18], RNN [9, 19], or graph convolutional network [15].

W2VV++ [20] leverages three text representations, including bag of words, word2vec, and gated recurrent unit (GRU), to form a high-dimensional sentence-level representation. Nevertheless, W2VV++ only utilizes the meaning pooling strategy over video frames. Dong et al. [21, 22] utilize a multi-level encoding strategy to extract multiple video representations and combine them as a final video representation.

Although common space learning methods give superior performance to semantic-based methods, each dimension of the common latent space lacks interpretability. Combining the advantages of the common latent space and the semantic concept space can improve the cross-modal retrieval performance and increase the interpretability of the model. Furthermore, unlike Dual Encoding [22], we measure multiple similarities in different latent spaces in addition to taking advantage of common latent space and semantic concept space.

In this paper, we make the following contributions.

- We design a level-wise aligned mechanism to align representations between videos and sentences in different levels. Specifically, we first exploit multi-level encoders to extract global, local, temporal, and spatial–temporal information in videos and text, respectively. Then, they are mapped into four different latent spaces and one semantic space.
- We combine the advantages of common latent space and semantic concept space to improve the cross-modal retrieval performance and increase the interpretability of our model. Specifically, we average four cross-modal similarities of different levels in four different latent spaces. Then, we combine it with the similarity in the semantic concept space.
- Extensive experiments are conducted on three widely used datasets including MSR-VTT [23], VATEX [24], and TRECVID AVS 2016-2018 [25–27]. The experimental results of our approach give superior performance to the state-of-the-art approaches.

The rest of this paper is organized as follows. Some related work is introduced in Sect. 1.2. We present our proposed method and experimental setup in Sects. 2 and 3, respectively. Section 3.4 provides the experimental results. Finally, Sect. 4 concludes our work.

1.2 Related work

This section reviews some previous work on language and video representations learning, and video–text retrieval, including semantic space learning and latent space learning.

1.2.1 Language representations

Bag of words [28] and word2vec [29] are earlier work on text representations, which cannot capture the contextual information in a sentence. Long short-term memory network (LSTM) [30] is one of the first deep models to overcome this shortcoming. Recently, the transformer architecture [31] has given impressive performance in sentence representations by leveraging a self-attention mechanism, where every word in a sentence can focus on all other words. The transformer architecture consists of alternately stacked self-attention layers and fully connected layers, which form the basis of the popular language architecture BERT [32]. Burns et al. [33] leverage different word embeddings and language networks (LSTM, BERT, etc.) to analyze their performance in text–video tasks. They believe that the performance of the pre-trained and frozen BERT architecture is relatively poor than that of an average embedding architecture or the LSTM.

1.2.2 Video representations

A common method to extract video representations is to extract each keyframe representation by some pre-trained CNN models and then combine them by max pooling or mean pooling. However, these ways cannot attend the temporal information in the video. In order to incorporate spatial and temporal representations from still frames and motion between frames, Simonyan et al. [34] leverage a two-stream model to perform action recognition in videos. Besides, I3D [35] utilizes a two-stream inflated 3D ConvNet to better attend the temporal information in a video. Xie et al. [36] proposed an alternative approach, which replaces 3D convolutions with spatial convolutions in 2D and temporal convolution in 1D.

1.2.3 Semantic space learning

[37, 38] create a concept vector for each test keyframe by concatenating 1000 ImageNet concepts and 345 TRECVID SIN concepts and translate a textual query to relevant predefined concepts by a set of complex linguistic rules. [39] builds a much larger semantic concept bank containing over 50,000 concepts by utilizing a pre-trained CNN architecture and support vector machines (SVMs). [40] recognizes ImageNet hierarchies to gain about 13k concepts and utilizes VideoStory [16] to generate semantic representations. Since semantic concepts are limited and unstructured, it is hard to represent the rich contextual information within both sentence and video. However, encoding video and sentences into concept vectors makes the model somewhat interpretable.

1.2.4 Latent space learning

The methods based on common latent space first extract representations from video and sentence, respectively, and then project them into a latent space, where the cross-modal similarity can be directly calculated. For these methods, what matter are how to extract rich representations from video and sentence separately and measure the video–text similarity. Therefore, we review recent progress from these three aspects.

For video representations, a common method is to first extract frame representations from the video through some pre-trained CNN models and then combine them along the temporal dimension into a video-level representation by mean pooling [9, 10, 13, 41, 42] or max pooling [7, 15, 18].

Yang et al. [14] first leverage GRU to explore the temporal relationship between video keyframes and then use a self-attention mechanism to capture the representation interaction among keyframes. Additionally, [7, 9, 13] leverage motion features extracted from the I3D model [35], and audio features generated by the audio CNN model [43] as part of the visual representations. Nevertheless, these methods still leverage max pooling, mean pooling, or NetVLAD to combine various features into a single feature vector per video.

For text representations, word2vec models are widely used, which are pre-trained on large-scale text corpora. Specifically, for each word of a sentence, a dense vector is first generated by multiplying its one-hot vector with a pre-trained word embedding matrix. Then, they are combined by NetVLAD, max pooling, or Fisher Vector. Although they have achieved good performance, they cannot capture the sequential information in a sentence. Recurrent neural networks (RNNs) are effective in employing sequential information. Moreover, variants of RNN, such as LSTM, bidirectional LSTM, GRU, and bidirectional GRU, are utilized in [9, 19, 44, 45], respectively. For example, in [9], the sentence representations are from the last hidden state of the GRU. [42] and [20] utilize three text representations, including BoW, word2vec, and GRU. However, these methods only leverage mean pooling to obtain the video representation. HGR [15] utilizes a hierarchical decomposition of a sentence to explore the relationship between words, which requires the sentence to be well annotated with certain linguistic rules.

For video–text similarity learning, recent methods map video features and text features into a common latent space where the text–video cross-modal similarity can be computed by cosine similarity and leverage various triplet ranking losses to train their models. In addition to the triplet ranking loss, reconstruction loss and contrastive loss are utilized to learn the latent space in [46]. Recently, an increasing number of methods learn several latent spaces instead of just learning one latent space. Mixture of embedding experts (MEE) [18] computes the final similarity by a weighted combination between sentence and multiple video latent spaces, one for each input including motion, appearance, face, or audio representation. HGR [15] assumes a hierarchical decomposition of the video and text and projects them into three spaces, including events, actions, and entities.

Unlike the existing methods that learn semantic concept space or common latent space, our approach simultaneously learns these two spaces, which takes advantage of the interpretability of semantic concept space and the high performance of common latent space. We separately represent video and text as four complementary representations, including global, temporal, local, and spatial–temporal representation, and learn one common latent space for each representation. Besides, we also map spatial–temporal representation into a semantic space. Thus, our proposed method can align different levels of information in various spaces.

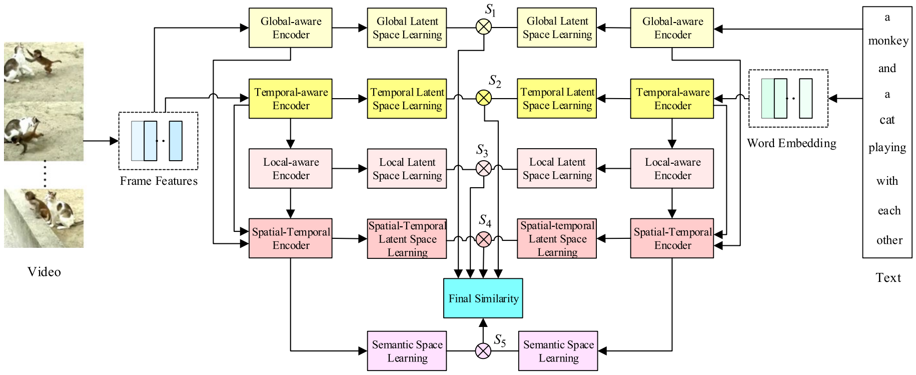

Fig. 2 The framework of our proposed LADN method. LADN first utilizes multi-level encoders to extract global, temporal, local, and spatial–temporal information in videos and text. Then, they are mapped into four different latent spaces and a semantic space. Finally, LADN aligns different levels of information in various spaces

2 Methods

As illustrated in Fig. 2, we put forward an architecture, named level-wise aligned dual networks (LADNs), to improve the performance of text–video cross-modal retrieval.

2.1 Video encoder

Following the setting in [22], we extract n frames at 0.5 second intervals from a video. For each frame, we utilize a pre-trained ImageNet CNN to extract deep representation. Therefore, the video is represented as a sequence of frame representations $\{v_1, v_2, ..., v_n\}$, where v_t denotes the representation of the $t-$th frame.

2.1.1 Global-aware encoder

Given a video, we take the average of all frame features as $f_1^{(v)}$, which denotes a visual pattern repeatedly appearing in the video clip.

2.1.2 Temporal-aware encoder

We utilize a bidirectional gated recurrent units (BiGRU) [47] to extract temporal information from video frame features. The BiGRU consists of two GRU: One encodes frame features in a forward direction, and the other is backward. At a specific time step t, the hidden feature of the forward GRU is expressed as $\overrightarrow{h_t}$ and the one of the backward GRU is expressed as $\overleftarrow{h_t}$. We obtain the BiGRU output by averaging $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ as $h_t^{(v)}$. Then, the temporal-aware feature $f_2^{(v)}$ is obtained by averaging $h_t^{(v)}$ along the time dimension.

2.1.3 Local-aware encoder

The temporal-aware feature cannot extract the subtle difference between each frame. Therefore, we leverage 1D CNN [48] following BiGRU to extract local-aware patterns in the video. The output of BiGRU is represented as $H^{(v)} = \left\{h_1^{(v)}, h_2^{(v)}, \ldots, h_n^{(v)}\right\}$, which is the input of 1D CNN. Conv1$d_{k,r}$ denotes 1D convolutional module including r filters of size k. The activation function of 1D CNN is ReLU. Next, we utilize max pooling to get a fixed length r. The above process can be expressed as follows,

$$c_k^{(v)} = \text{max\_pooling}\left(\text{ReLU}\left(\text{Conv1}d_{k,r}\left(H^{(v)}\right)\right)\right). \tag{1}$$

We set $k = 2, 3, 4, 5$ to generate multi-scale local-aware representations and concatenate them as $f_3^{(v)}$.

2.1.4 Spatial–temporal encoder

$f_1^{(v)}$, $f_2^{(v)}$, and $f_3^{(v)}$ naturally extract global, temporal, and local information in video content, respectively. We assume the three patterns are complementary to each other, with some redundant information. Therefore, we concatenate the three patterns as $f_4^{(v)}$, which captures spatial and temporal information in the video.

2.2 Text encoder

Similar to the encoding strategy for the video modality, the text modality also utilizes four different encoders to extract different information.

Given a sentence s of length m, we utilize classical bag-of-words features to represent it. Let $f_1^{(s)} = [w_1, w_2, ..., w_m]$ denote the global feature of a sentence, where w_m denotes the number of occurrences of the $m-$th word.

We leverage a word embedding matrix to convert each word from a one-hot vector into a dense vector. The matrix was initialized by a word2vec model [42] trained on English tags of 30 million Flickr images. Next, we can obtain a temporal-aware representation $f_2^{(s)}$ like that used by the temporal-aware encoder in the video.

Similar to the video counterpart, we use four 1D CNN modules with $k = 2, 3, 4, 5$ to generate multi-scale representations. Their outputs are concatenated as local-aware feature $f_3^{(s)} = [c_2^{(s)}, c_3^{(s)}, c_4^{(s)}, c_5^{(s)}]$.

We concatenate $f_1^{(s)}$, $f_2^{(s)}$, and $f_3^{(s)}$ as $f_4^{(s)}$, which means the spatial and temporal feature of a sentence s.

2.3 Latent space learning

Given the video features $f_1^{(v)}, f_2^{(v)}, f_3^{(v)}, f_4^{(v)}$ and the text features $f_1^{(s)}, f_2^{(s)}, f_3^{(s)}, f_4^{(s)}$ in different levels, we transform them into four different latent spaces, respectively, as follows,

$$\phi_i^{(x)} = \mathrm{BN}\left(W_i^{(x)} f_i^{(x)} + b_i^{(x)}\right), \tag{2}$$

where $x \in \{v, s\}$, $i = 1, 2, 3, 4$, W_i is the parameter of a fully connected layer, and b_i is its bias item, and BN denotes a batch normalization layer. Then, we utilize cosine similarity $\mathrm{sim\_lat}_i(v,\ s)$ to calculate the video–text similarity between $\phi_i^{(v)}$ and $\phi_i^{(s)}$.

The improved triplet ranking loss is leveraged to make relevant video–text pairs closer than irrelevant pairs during the training phase. We define the bidirectional ranking loss for each level as follows,

$$\begin{aligned} \mathcal{L}\_\mathrm{lat\_rank}_i(v,s) = &\max\left(\mathrm{sim\_lat}_i(v,s^+) - \mathrm{sim\_lat}_i(v,s^-) + m_1, 0\right) \\ &+ \max\left(\mathrm{sim\_lat}_i(s,v^+) - \mathrm{sim\_lat}_i(s,v^-) + m_2, 0\right), \end{aligned} \tag{3}$$

where s^+ and s^- denote a positive sentence sample and a negative one for a video clip v, respectively. v^+ and v^- denote a positive video sample and a negative one for a sentence s, respectively. And m_1, m_2 are the margin. In addition, the negative sample is the most similar yet negative for the anchor v or s. By taking the average of ranking losses in four different levels, the final loss in the latent space can be denoted as $\mathcal{L}\_\mathrm{lat}(v, s)$.

2.4 Semantic space learning

Following the setting in [22], during the training phase, we put all the sentences in the training set together and count the number of occurrences of all semantic concepts. Next, we utilize the top 512 semantic concepts that appear most frequently as semantic categories. In order to transform $f_4^{(v)}$ and $f_4^{(s)}$ into a semantic space, we utilize the following method,

$$\varphi^{(x)} = \sigma\left(\text{BN}\left(W_i^{(x)} f_4^{(x)} + b_i^{(x)}\right)\right), \tag{4}$$

where $x = \{v, s\}$, $i = 5$, and $\sigma(\cdot)$ means a sigmoid activation function which is utilized to output a multi-label classification probability vector. Given a video–sentence pair and their shared ground-truth semantic concept y, the binary cross-entropy (BCE) loss is formulated as

$$\begin{aligned} \mathcal{L}\_\text{sem\_bce}(v, s, y) =& \frac{1}{512}\sum_{i=1}^{512}\left[y_i \log\left(\varphi_i^{(v)}\right) + (1 - y_i)\log\left(1 - \varphi_i^{(v)}\right)\right] \\ &+ \frac{1}{512}\sum_{i=1}^{512}\left[y_i \log\left(\varphi_i^{(s)}\right) + (1 - y_i)\log\left(1 - \varphi_i^{(s)}\right)\right]. \end{aligned} \tag{5}$$

The BCE loss can improve the interpretability of the concept space but cannot improve the performance of video–text retrieval. Therefore, in order to measure the video–sentence similarity in the semantic concept space, we formulate

$$\text{sim\_sem}(v, s) = \frac{\sum_{i=1}^{512} \min(\varphi^{(v)}, \varphi^{(s)})}{\sum_{i=1}^{512} \max(\varphi^{(v)}, \varphi^{(s)})}. \tag{6}$$

We also leverage the improved triplet ranking loss in the semantic space as follows,

$$\begin{aligned} \mathcal{L}\_\text{sem\_rank}(v, s) =& \max(\text{sim\_sem}(v, s^{+}) - \text{sim\_sem}(v, s^{-}) + m_3, 0) \\ &+ \max(\text{sim\_sem}(s, v^{+}) - \text{sim\_sem}(s, v^{-}) + m_4, 0). \end{aligned} \tag{7}$$

The final loss in the semantic space can be formulated as,

$$\mathcal{L}\_\text{sem}(v, s, y) = \mathcal{L}\_\text{sem\_bce}(v, s, y) + \mathcal{L}\_\text{sem\_rank}(v, s). \tag{8}$$

2.5 Joint training of two spaces

By minimizing the sum of the latent-based loss and the semantic-based loss, we can train our LADN model as,

$$\min \mathcal{L}\_\text{lat}(v, s) + \mathcal{L}\_\text{sem}(v, s, y). \tag{9}$$

Therefore, our LADN model can leverage different levels of patterns to improve the ranking performance and is also interpretable.

2.6 Measuring of video–text similarity

In the querying phase, we first obtain four similarities of different levels in four latent spaces and one similarity in the semantic space. By taking the average of four similarities of different levels in four latent spaces, we can obtain the final latent-based similarity between a video v and a sentence s as $\text{sim\_lat}(v, s)$.

Then, min-max normalization is utilized to normalize $\text{sim\_lat}(v, s)$ and $\text{sim\_lat}(v, s)$ as $\text{sim}\tilde{\_}\text{lat}(v, s)$ and $\text{sim}\tilde{\_}\text{sem}(v, s)$, respectively. Finally, we combine them in a weighted method as,

$$sim(v,s) = \gamma \cdot \text{sim}\tilde{\_}\text{lat}(v,s) + (1-\gamma) \cdot \text{sim}\tilde{\_}\text{sem}(v,s), \tag{10}$$

where γ is a weight to stride a balance between the latent space and the semantic space, ranging from 0 to 1.

3 Experiments

3.1 Dataset

The MSR-VTT dataset [23] consists of 10,000 web video clips, each with 20 natural sentences. For this dataset, there are three different ways of the data partition. The original partition leverages 497 videos for validation, 2990 for testing, and 6513 for training. The second partition [18] leverages 1000 and 6656 videos for testing and training, respectively. The third partition [19] leverages 1000 videos for testing and 7010 for training. For the last two partitions, 1000 videos are randomly selected following [22]. We refer to these three partitions as A, B, C, respectively.

The VATEX dataset [24] is a large-scale multilingual dataset for text–video retrieval. Each video contains 10 Chinese sentences and 10 English sentences. In our experiments, only the English sentences are utilized. According to [15], we utilize 25,991 videos for training, 1500 videos for validation, and 1500 videos for testing.

The TRECVID AVS (Ad hoc Video Search) task provides the largest test collection, the IACC.3 dataset, for zero-example video retrieval. The IACC.3 dataset, used in TRECVID AVS 2016–2018 tasks [25–27], contains 335,944 shots. Given an ad hoc query, the task is to return a ranked list of 1000 clips according to their likelihood of about the target query. In addition, TRECVID specifies 30 different queries each year.

3.2 Performance metrics

For the MSR-VTT dataset and Vatex dataset, R@k ($k = 1$, 5, 10, higher is better), Median rank (Med r, lower is better), and mean Average Precision (mAP, higher is better) are utilized to evaluate the performance of text–video cross-modal retrieval. R@k is the proportion of at least one correct item found in the top-k retrieved results. Med r means the median rank of the first correct item in the retrieved results. We also report the sum of all recalls (SumR) to reflect the overall performance.

For the TRECVID AVS tasks on the IACC.3 dataset, we utilize the official performance metric, inferred average precision (infAP, higher is better). For overall performance, we average infAP scores over the queries.

3.3 Experimental details

For VATEX, we utilize a 1,024-d I3D [35] representation to represent a video clip. As for the other datasets, we extract ResNeXt-101 [49] and ResNet-152 [50] representations for each frame. We concatenate these two representations to generate a 4,096-d CNN representation, which we call concatenated ResNeXt-ResNet. In addition, we average these two representations to generate a 2,048-d CNN representation, named average ResNeXt-ResNet.

Our proposed model is implemented using PyTorch. Taking MSR-VTT B [18], for example, we set all margins to 0.2, except for m_2 which is set to 0.3 in Eq. (3). The feature dimension of the BiGRU hidden state is set to 1024. The weight γ is set to 0.6. The dimensions of four different latent spaces are all set to 1536. We utilize stochastic gradient descent with Adam [51] to train our model. The batch size is 128. We set the initial learning rate to 0.0001. The maximum number of epoch is 50. We leverage an early stop mechanism to adjust the training process.

3.4 Experimental results

3.4.1 Experiments on MSR-VTT

We utilize the following twelve state-of-the-art methods for comparison.

- MEE [18] computes the final similarity by a weighted combination between sentence and four video latent spaces including appearance, motion, face, and audio.
- W2VV [42] leverages three text representations, including BoW, word2vec, and GRU, to represent a sentence.
- VSE++ [52] is a state-of-the-art method, which is widely utilized as the baseline for video–text retrieval. We replace its image feature with the feature obtained by mean pooling on frame-level features.
- Mithun et al. [9] learns two latent spaces for videos and text and leverages a weighted triplet ranking loss to train the model.
- W2VV++ [20] is an improved version of W2VV, which takes advantage of better text encoding strategies and an improved triplet ranking loss compared to W2VV.
- CE [13] merges multiple expert features of video by a collaborative gating mechanism to represent a video.
- TCE [14] leverages a tree-based encoder to represent text, and a temporal attentive video encoder to represent videos.
- HGR [15] assumes a hierarchical decomposition of the video and text and projects them into three spaces including events, actions, and entities.
- JPoSE [8] decomposes captions into nouns and verbs and creates two latent spaces for them, respectively.
- JSFusion [19] utilizes a joint sequence fusion to combine text and video representations.
- Miech et al. [7] leverages gated embedding modules to project videos and text into a common latent space.
- Dual Encoding [22] uses multiple encoding strategies to represent text and video, respectively.

For fair comparison, we directly cite results from the original papers where available. However, video representations used in different papers vary. Therefore, we cite results from [22], which are implemented by leveraging the same concatenated ResNeXt-ResNet representation as the video representation. In addition, we retrain the Dual Encoding [22] by utilizing the average ResNeXt-ResNet representation. We train our LADN model by using both the concatenated ResNeXt-ResNet representation and the average ResNeXt-ResNet representation.

Table 1 presents the retrieval performance of three different partitioning approaches in the MSR-VTT database. From Table 1, for all methods, their performance on the A partition is inferior to those on the B and C partition. Because A partition utilizes more

Table 1 Experimental results on MSR-VTT. We utilize three split methods as A [23], B [18], and C [19], respectively. Larger R@k, mAP, and lower Med R denote better performance

Method	**Split**	**Text-to-Video Retrieval**					**Video-to-Text Retrieval**					**SumR**
		R@1	**R@5**	**R@10**	**Med R**	**mAP**	**R@1**	**R@5**	**R@10**	**Med R**	**mAP**	
Mithun et al.* [9]	A [23]	7	20.9	29.7	38	-	12.5	32.1	42.4	16	-	144.6
TCE* [14]		7.7	22.5	32.1	30	-	-	-	-	-	-	-
HGR* [15]		9.2	26.2	36.5	24	-	15	36.7	48.8	11	-	172.4
CE* [13]		10	29	41.2	16	-	15.6	40.9	55.2	8.3	-	191.9
W2VV [42]		1.1	4.7	8.1	236	3.7	17	37.9	49.1	11	7.6	117.9
MEE [18]		6.8	20.7	31.1	28	14.7	13.4	32	44	14	6.6	148
CE [13]		7.9	23.6	34.6	23	16.5	11	31.9	46.1	13	6.8	155.1
VSE++ [52]		8.7	24.3	34.1	28	16.9	15.6	36.6	48.6	11	7.4	167.9
TCE [14]		9.3	27.3	38.6	19	18.7	15.1	36.8	50.2	10	8	177.3
W2VV++ [20]		11.1	29.6	40.5	18	20.6	17.5	40.2	52.5	9	8.5	191.4
HGR [15]		11.1	30.5	42.1	16	20.8	18.7	44.3	57.6	7	9.9	204.4
Dual Encoding [22]		11.6	30.3	41.3	17	21.2	22.5	47.1	58.9	7	10.5	211.7
LADN		12.9	33.6	45.3	14	23.3	22.2	47.4	60.3	6	11.5	221.7
Dual Encoding* [22]		12.1	31.4	42.9	16	22.0	21.3	45.6	58.1	7	10.4	211.3
LADN*		13.1	33.9	45.4	14	23.4	23.0	48.1	60.5	6	11.5	224.0
JPoSE* [8]	B [18]	14.3	38.1	53	9	-	16.4	41.3	54.4	8.7	-	217.5
MEE* [18]		16.8	41	54.4	9	-	-	-	-	-	-	-
TCE* [14]		17.1	39.9	53.7	9	-	-	-	-	-	-	-
CE* [13]		18.2	46	60.7	7	-	18	46	60.3	6.5	-	249.2
W2VV [42]		2.7	12.5	17.3	83	7.9	17.3	42	53.5	9	29.3	145.3
MEE [18]		15.7	39	52.3	9	27.1	15.3	41.9	54.5	8	28.1	218.7
VSE++ [52]		17	40.9	52	10	16.9	18.1	40.4	52.1	9	29.2	220.5
CE [13]		17.8	42.8	56.1	8	30.3	17.4	42.9	56.1	8	29.8	233.1
TCE [14]		17	44.7	58.3	7	30	15.1	43.3	58.2	7	28.3	236.6
W2VV++ [20]		21.7	48.6	60.9	6	34.4	18.6	46.4	59.1	6	31.7	255.3
HGR [15]		22.9	50.2	63.6	5	35.9	20	48.3	60.9	6	33.2	265.9
Dual Encoding [22]		23	50.6	62.5	5	36.1	25.1	52.1	64.6	5	37.7	277.9
LADN		25.5	52.9	66.9	5	38.6	25.3	55.2	66.7	4	39.3	292.5
Dual Encoding* [22]		23.1	51.2	62.6	5	35.9	24.1	52.2	63.6	5	37.18	276.8
LADN*		26.6	55.5	66.9	4	39.9	26.9	55.0	67.4	4	40.1	298.3
JSFusion* [19]		10.2	31.2	43.2	13	-	-	-	-	-	-	-
TCE* [14]		16.1	38	51.5	10	-	-	-	-	-	-	-
Miech et al.* [7]		14.9	40.2	52.8	9	-	-	-	-	-	-	-
CE* [13]		20.9	48.8	62.4	6	-	20.6	50.3	64	5.3	-	267
W2VV [42]		1.9	9.9	15.2	79	6.8	17.3	39.3	50.2	10	27.8	133.8
VSE++ [52]		16	38.5	50.9	10	27.4	16.2	39.3	51.2	10	27.4	212.1
MEE [18]		14.6	38.4	52.4	9	26.1	15.2	40.9	53.8	9	27.9	215.3
W2VV++ [20]		19	45	58.7	7	31.8	16.9	42.7	54.6	8	29	236.9
CE [13]		17.2	46.2	58.5	7	30.3	15.8	44.9	59.2	7	30.4	241.8
TCE [14]		17.8	46	58.3	7	31.1	18.9	43.5	58.8	7	31.4	243.3
HGR [15]	C [19]	21.7	47.4	61.1	6	34	20.4	47.9	60.6	6	33.4	259.1
Dual Encoding [22]		21.1	48.7	60.2	6	33.6	21.7	49.4	61.6	6	34.7	262.7
LADN		24.4	52	63.4	5	37.4	23.6	50.8	62.8	5	36.6	277.0
Dual Encoding* [22]		21.9	48.1	61.5	6	34.5	22.3	48	61.6	6	34.6	263.4
LADN*		24.6	52.5	64.0	5	37.5	22.5	53.0	65.1	5	36.3	281.7

* denotes results directly cited from the original papers, * denotes numbers obtained by training given the average ResNeXt-ResNet representation, and the others are obtained by training given the concatenated ResNeXt-ResNet representation

Table 2 Model complexity of Dual Encoding and our model LADN

Model	Model complexity	
	Parameters (M)	FLOPs (G)
Dual Encoding	65.7	1.08
LADN	109.4	1.46

candidate videos/sentences during retrieving phase than other partitions. Our model LADN can gain the best performance by utilizing the same concatenated video features. In addition, the performance of Dual Encoding [22] has hardly changed by utilizing different video representations. However, LADN can significantly improve the retrieval performance by utilizing the average ResNeXt-ResNet representation. Moreover, the model utilizing the average ResNeXt-ResNet representation can reduce the trainable parameters compared to the model utilizing the concatenated ResNeXt-ResNet representation. Therefore, we utilize the average ResNeXt-ResNet representation as the input of LADN in the following experiments.

Compared with the SumR of Dual Encoding using the concatenated ResNeXt-ResNet representation, the ones of our method LADN using the average ResNeXt-ResNet representation can improve 5.81%, 7.34%, and 7.23% on MSR-VTT A, B, and C partition, respectively. Dual Encoding only maps spatial–temporal representation into a latent space. However, our proposed method LADN not only performs the same operation, but also projects global, temporal, and local representations into another three different latent spaces. Furthermore, LADN takes the average of four similarities in these four latent spaces to help improve the retrieval performance.

Table 2 presents the model complexity of Dual Encoding and our method LADN. Compared with Dual Encoding, our method LADN needs more computational complexity. This is because LADN utilizes four different latent spaces, while Dual Encoding only leverages one latent space. When LADN projects representations into another three latent spaces, it needs more computational complexity. However, the text–video retrieval performance of LADN is better than the one of Dual Encoding.

Figures 3, 4, and 5 show the text-to-video retrieval results of our method LADN and Dual Encoding on the MSR-VTT B partition [18]. In Fig. 3, LADN can rank the corresponding results in the 1st place, but Dual Encoding fails, which proves the superiority of our method LADN. Figures 4 and 5 are still problematic to LADN and Dual Encoding. For the results in Fig. 4, although these two methods can get the right concept "paper," they cannot find the intrinsic relationship between the sentence and the corresponding video. The possible reason for this is that the dataset contains only a small number of videos about "typewriter." For the results in Fig. 5, although the top 1 retrieved result is incorrect, its semantic is consistent with the semantic of the ground truth for our method LADN and Dual Encoding.

3.4.2 Experiments on VATEX

For the VATEX dataset, we compare our method LADN with W2VV, VSE++, CE, W2VV++, HGR, and Dual Encoding. Table 3 summarizes the performance. W2VV,

Table 3 Experimental results on VATEX

Method	Text-to-Video			Video-to-Text			SumR
	R@1	R@5	R@10	R@1	R@5	R@10	
W2VV [42]	14.6	36.3	46.1	39.6	69.5	79.4	285.5
VSE++ [52]	31.3	65.8	76.4	42.9	73.9	83.6	373.9
CE [13]	31.1	68.7	80.2	41.3	71	82.3	374.6
W2VV++ [20]	32	68.2	78.8	41.8	75.1	84.3	380.2
HGR [15]	35.1	73.5	83.5	-	-	-	-
Dual Encoding [22]	36.8	73.6	83.7	46.8	75.7	85.1	401.7
LADN	37.4	75.5	84.8	51.1	78.1	86.1	413.0

VSE++, and W2VV++ project video and text into a common latent space. However, our method LADN maps four different levels of video and text representations into four latent spaces. By utilizing different levels of information, LADN can perform better than its counterparts.

3.4.3 Experiments on TRECVID AVS 2016-2018

We cite top 3 results on TRECVID AVS tasks for each year, including [53, 55, 57] in 2016, [39, 40, 58] in 2017, [54, 56, 59] in 2018. Additionally, we cite results from [16, 60], and [37]. Other results are cited from [22]. Table 4 shows the experimental results, where the overall performance is the average score over three years. Our proposed method LADN gives the best performance, which demonstrates that LADN can effectively perform large-scale video retrieval by text query.

3.4.4 Ablation study

We design several variants of LADN to verify the effectiveness of each of its components. We construct the LADN(w/o g, t, l alignments) variant by removing global, temporal, and local alignments. The LADN(w/o semantic space) variant is built by removing the semantic space. We construct the LADN(w/ g, t, l semantic space) variant by mapping global, temporal, and local information into three semantic spaces, respectively. We remove the alignments in the global, local, temporal, and spatial–temporal spaces to construct LADN(w/o g alignment), LADN(w/o l alignment), LADN(w/o t alignment), and LADN(w/o s_t alignment), respectively. Table 5 presents the experimental results on MSR-VTT B partition [18]. Compared with LADN, LADN(w/o g, t, l alignments) gains the worst performance. This result proves the effectiveness of the level-wise aligned mechanism. Because LADN can make full use of global, temporal, and local information to further improve the text–video retrieval performance. By comparing LADN and LADN(w/o semantic space), we conclude that the semantic space plays a vital role in improving retrieval performance. Compared with LADN, although LADN(w/ g, l, t semantic spaces) utilizes more semantic spaces, it cannot further improve the retrieval performance. The lack of alignment in any of the four spaces, including global, temporal, local, or spatial–temporal spaces, will result in poor performance. It demonstrates that these four latent spaces are complementary to each other.

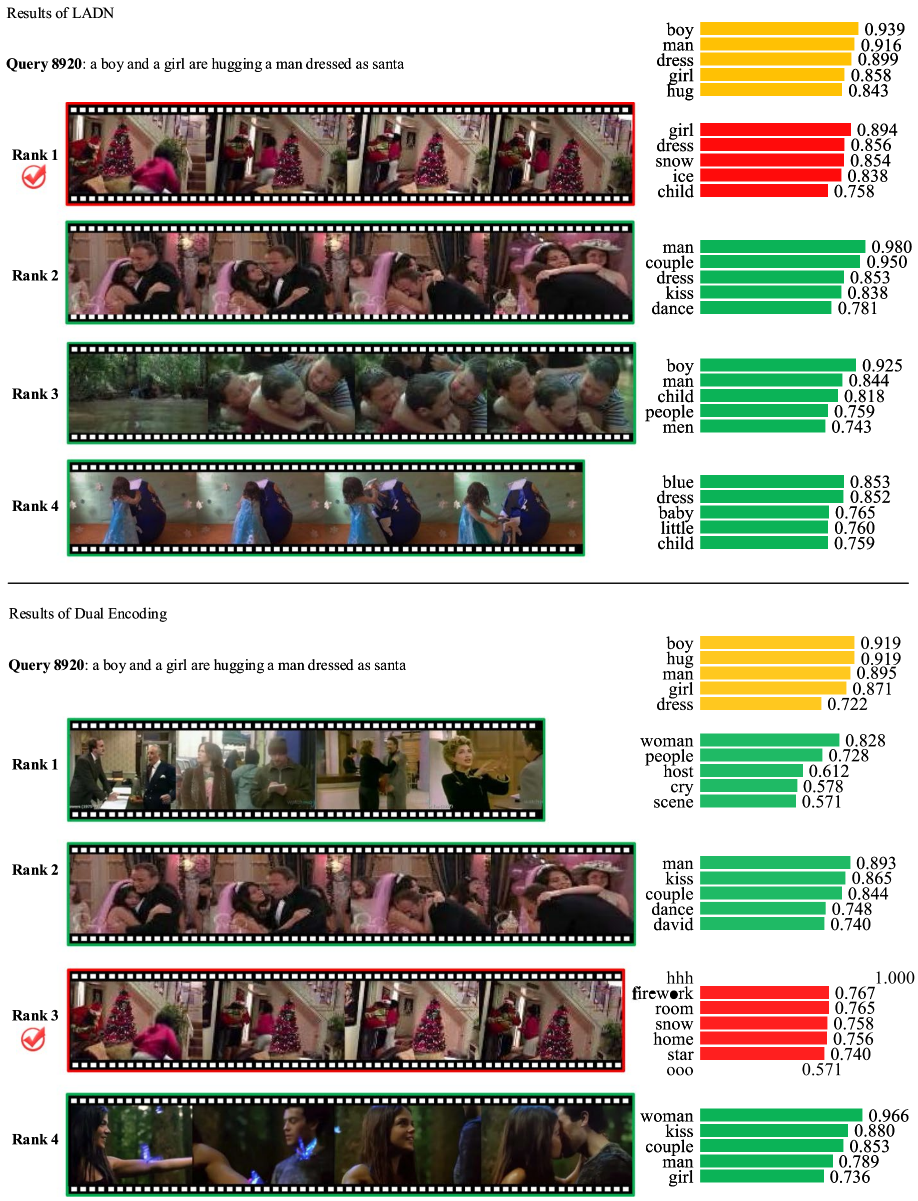

Fig. 3 The text-to-video retrieval results of our LADN method and Dual Encoding on the MSR-VTT B partition [18]. The top 4 ranked videos are shown for each query, where the ground truth is marked with a red box, and the others are marked with a green box. The last column is the predicted concepts corresponding to the second column

4 Conclusion

This paper proposes a method named level-wise aligned dual networks (LADN) for text–video retrieval. LADN first utilizes multi-level encoders to extract global, local, temporal, and spatial–temporal information in videos and sentences. Then, they are mapped into four different latent spaces and one semantic space. Finally, LADN combines the similarities of four latent spaces and one semantic concept space to improve cross-modal retrieval performance and increase interpretability. Extensive experiments conducted on three widely used datasets, including MSR-VTT, VATEX, and TRECVID

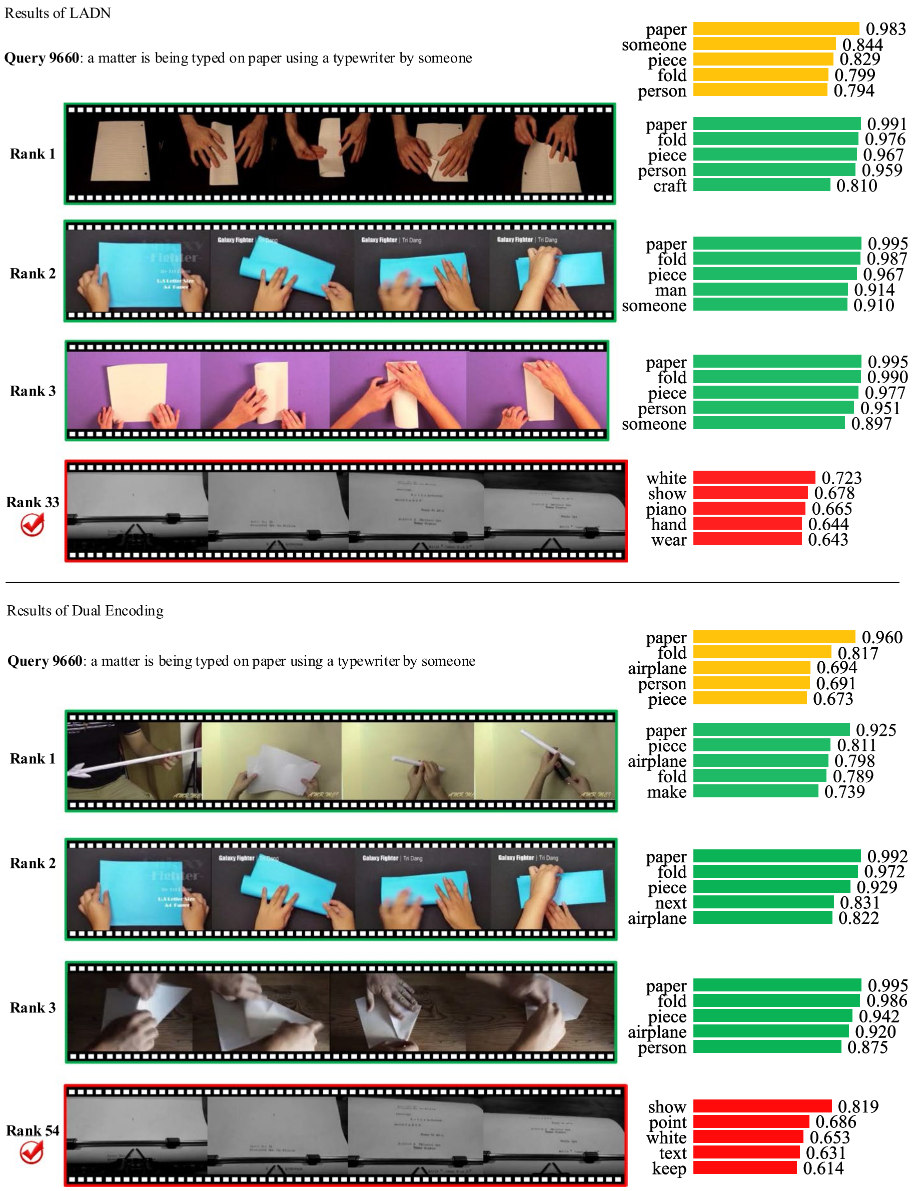

Fig. 4 The text-to-video retrieval results of our LADN and Dual Encoding on the MSR-VTT B partition [18]. The top 3 ranked videos and the ground truth are shown for each query. Additionally, the ground truth is marked with a red box, and the others are marked with a green box. The last column is the predicted concepts corresponding to the second column

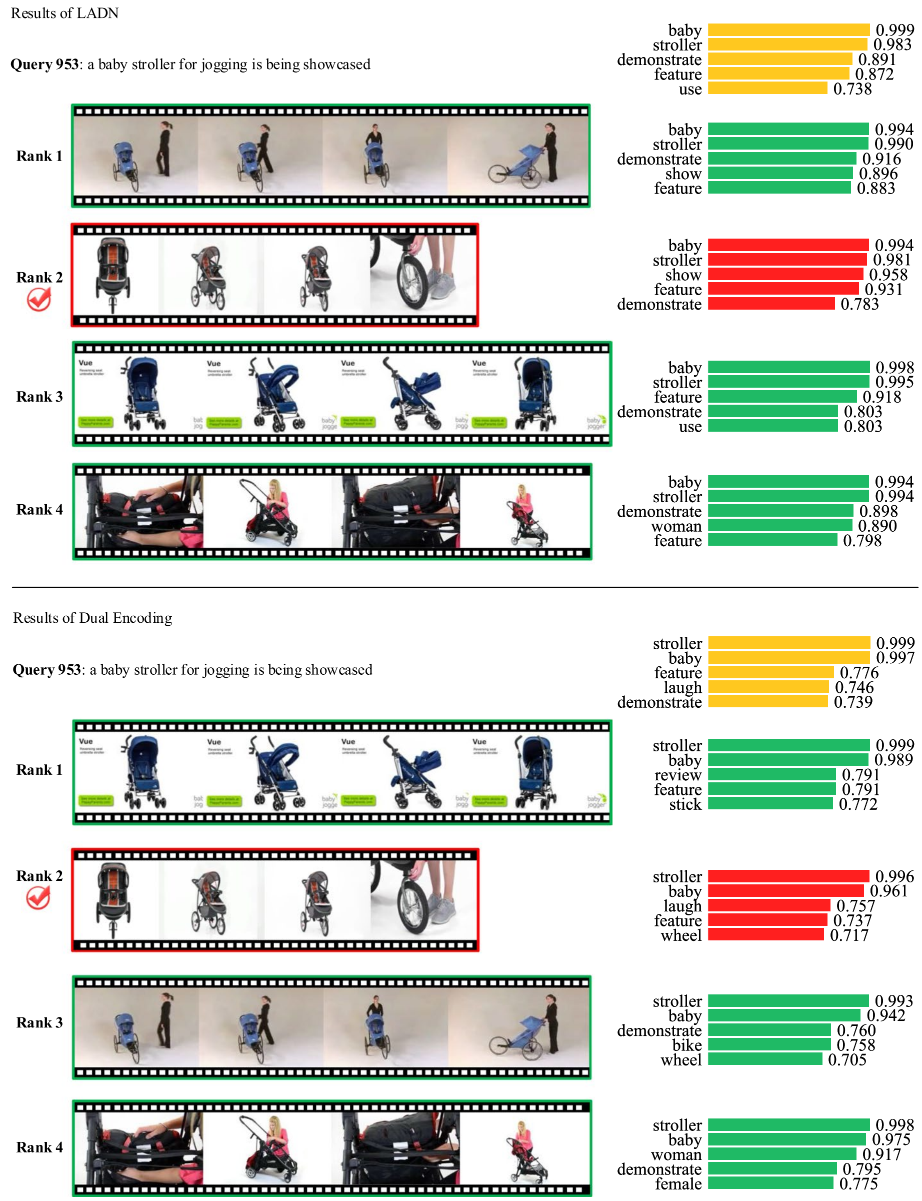

Fig. 5 The text-to-video retrieval results of our LADN method and Dual Encoding on the MSR-VTT B partition [18]. The top 4 ranked videos are shown for each query, where the ground truth is marked with a red box, and the others are marked with a green box. The last column is the predicted concepts corresponding to the second column

Table 4 Experimental results on TRECVID AVS 2016, 2017, and 2018

	TRECVID edition			
	2016	**2017**	**2018**	**OVERALL**
Top 3 TRECVID finalists				
Rank 1	5.4 [53]	20.6 [40]	12.1 [54]	–
Rank 2	5.1 [55]	15.9 [39]	8.7 [56]	–
Rank 3	4 [57]	12 [58]	8.2 [59]	–
Literature methods				
VideoStory [16, 60]	8.7	15	–	–
Markatopoulou et al. [37]	6.4	–	–	–
CE [13]	7.4	14.5	8.6	10.2
VSE++ [52]	13.5	16.3	10.6	13.5
W2VV [42]	14.9	19.8	10.3	15
W2VV++ [20]	15.1	21.3	10.6	15.7
Dual Encoding [22]	15.2	23.1	12.1	16.8
LADN	15.3	24.1	12.6	17.3

Table 5 Ablation Experiments on MSR-VTT B partition [18]. w/ and w/o mean with and without, respectively. g, t, l, s_t denote global, temporal, local, spatial–temporal, respectively

LADN variants	**Text-to-Video retrieval**					**Video-to-Text retrieval**					**SumR**
	R@1	**R@5**	**R@10**	**Med R**	**mAP**	**R@1**	**R@5**	**R@10**	**Med R**	**mAP**	
original LADN	26.6	55.5	66.9	4	39.9	26.9	55.0	67.4	4	40.1	**298.3**
w/o g, t, l alignments	24.2	52.6	61.2	5	37.0	25.3	52.3	62.9	5	38.1	278.5
w/o semantic space	25.3	53.8	64.3	5	38.2	27.0	52.8	64.3	5	39.5	287.5
w/ g, t, l semantic spaces	26.2	54.2	66.6	5	39.0	26.4	54.5	66.4	4	39.8	294.3
w/o g alignment	25.1	55.6	66.3	4	38.9	25.3	54.8	67.0	4	39.4	294.1
w/o l alignment	24.9	53.1	64.6	5	38.1	25.5	54.7	65.8	5	39.1	288.6
w/o t alignment	25.6	56.2	66.4	4	39.2	26.4	54.8	66.6	4	39.9	296.0
w/o s_t alignment	25.9	53.3	66.2	5	38.9	26.0	55.1	66.2	4	39.7	292.7

AVS 2016-2018, demonstrate that our proposed approach is superior to several state-of-the-art text–video retrieval approaches.

Abbreviations

CNN	Convolutional neural network
RNN	Recurrent neural network
GRU	Gated recurrent unit
LSTM	Long short-term memory network
SVMs	Support vector machines
BiGRU	Bidirectional gated recurrent units
BCE	Binary cross-entropy
infAP	Inferred average precision

Acknowledgements

We thank the National Natural Science Foundation of China and the Fundamental Research Foundation of Shenzhen for their support.

Author contributions

QL, WC, and ZH designed the research. QL conducted the experiments and wrote this manuscript. All authors read and approved the final manuscript.

Funding
Funding was supported by the National Natural Science Foundation of China under Grants 61771322, 61871186, and 61971290 and the Fundamental Research Foundation of Shenzhen under Grant JCYJ20190808160815125.

Availability of data and materials
Please contact the authors for data requests.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 25 September 2021 Accepted: 5 June 2022
Published online: 07 July 2022

References

1. L.-Q. Zhang, L.-Y. Huang, X.-l. Duan, Video person reidentification based on neural ordinary differential equations and graph convolution network (2021)
2. J. Dalton, J. Allan, P. Mirajkar, Zero-shot video retrieval using content and concepts, in *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, (2013), pp. 1857–1860
3. L. Jiang, D. Meng, T. Mitamura, A.G. Hauptmann, Easy samples first: Self-paced reranking for zero-example multimedia search, in *Proceedings of the 22nd ACM International Conference on Multimedia*, (2014), pp. 547–556
4. A. Habibian, T. Mensink, C.G. Snoek, Composite concept discovery for zero-shot video event detection, in *Proceedings of International Conference on Multimedia Retrieval*, (2014), pp. 17–24
5. S. Wu, S. Bondugula, F. Luisier, X. Zhuang, P. Natarajan, Zero-shot event detection using multi-modal fusion of weakly supervised concepts, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2014), pp. 2665–2672
6. X. Chang, Y. Yang, A. Hauptmann, E.P. Xing, Y.-L. Yu, Semantic concept discovery for large-scale zero-shot event detection, in *Twenty-fourth International Joint Conference on Artificial Intelligence* (2015)
7. A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, J. Sivic, Howto100m: learning a text-video embedding by watching hundred million narrated video clips, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (2019), pp. 2630–2640
8. M. Wray, D. Larlus, G. Csurka, D. Damen, Fine-grained action retrieval through multiple parts-of-speech embeddings, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (2019), pp. 450–459
9. N.C. Mithun, J. Li, F. Metze, A.K. Roy-Chowdhury, Learning joint embedding with multimodal cues for cross-modal video-text retrieval, in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, (2018), pp. 19–27
10. M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, N. Yokoya, Learning joint representations of videos and sentences with web image search, in *European Conference on Computer Vision*, (Springer, 2016), pp. 651–667
11. A. Torabi, N. Tandon, L. Sigal, Learning language-visual embedding for movie understanding with natural-language. arXiv preprint arXiv:1609.08124 (2016)
12. J. Dong, S. Huang, D. Xu, D. Tao, Dl-61-86 at trecvid 2017: video-to-text description, in *TRECVID* (2017)
13. Y. Liu, S. Albanie, A. Nagrani, A. Zisserman, Use what you have: Video retrieval using representations from collaborative experts. arXiv preprint arXiv:1907.13487 (2019)
14. X. Yang, J. Dong, Y. Cao, X. Wang, M. Wang, T.-S. Chua, Tree-augmented cross-modal encoding for complex-query video retrieval, in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, (2020), pp. 1339–1348
15. S. Chen, Y. Zhao, Q. Jin, Q. Wu, Fine-grained video-text retrieval with hierarchical graph reasoning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2020), pp. 10638–10647
16. A. Habibian, T. Mensink, C.G. Snoek, Video2vec embeddings recognize events when examples are scarce. IEEE Trans. Pattern Anal. Mach. Intell. **39**(10), 2089–2103 (2016)
17. M. Kratochvíl, P. Veselỳ, F. Mejzlík, J. Lokoč, Som-hunter: video browsing with relevance-to-som feedback loop, in *International Conference on Multimedia Modeling*, (Springer, 2020), pp. 790–795
18. A. Miech, I. Laptev, J. Sivic, Learning a text-video embedding from incomplete and heterogeneous data. arXiv preprint arXiv:1804.02516 (2018)
19. Y. Yu, J. Kim, G. Kim, A joint sequence fusion model for video question answering and retrieval, in *Proceedings of the European Conference on Computer Vision (ECCV)*, (2018), pp. 471–487
20. X. Li, C. Xu, G. Yang, Z. Chen, J.Dong, W2vv++ fully deep learning for ad-hoc video search, in *Proceedings of the 27th ACM International Conference on Multimedia*, (2019), pp. 1786–1794
21. J. Dong, X. Li, C. Xu, S. Ji, Y. He, G. Yang, X. Wang, Dual encoding for zero-example video retrieval, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2019), pp. 9346–9355
22. J. Dong, X. Li, C. Xu, X. Yang, G. Yang, X. Wang, M. Wang, Dual encoding for video retrieval by text. IEEE Trans. Pattern Anal. Mach. Intell. (2021)
23. J. Xu, T. Mei, T. Yao, Y. Rui, Msr-vtt: A large video description dataset for bridging video and language, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), pp. 5288–5296
24. X. Wang, J. Wu, J. Chen, L. Li, Y.-F. Wang, W.Y. Wang, Vatex: a large-scale, high-quality multilingual dataset for video-and-language research, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (2019), pp. 4581–4591

25. G. Awad, J. Fiscus, D. Joy, M. Michel, A. Smeaton, W. Kraaij, M. Eskevich, R. Aly, R. Ordelman, M. Ritter, Trecvid 2016: evaluating video search, video event detection, localization, and hyperlinking, in *TREC Video Retrieval Evaluation (TRECVID)* (2016)
26. G. Awad, A. Butt, J. Fiscus, D. Joy, A. Delgado, W. Mcclinton, M. Michel, A. Smeaton, Graham, Y. W. Kraaij, Trecvid 2017: evaluating ad-hoc and instance video search, events detection, video captioning, and hyperlinking, in *TREC Video Retrieval Evaluation (TRECVID)* (2017)
27. G. Awad, A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, D. Joy, A. Delgado, A. Smeaton, Y. Graham, Trecvid 2018: benchmarking video activity detection, video captioning and matching, video storytelling linking and video search, in *Proceedings of TRECVID 2018* (2018)
28. Y. Zhang, R. Jin, Z.-H. Zhou, Understanding bag-of-words model: a statistical framework. Int. J. Mach. Learn. Cybern. **1**(1–4), 43–52 (2010)
29. T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
30. S. Hochreiter, J. Schmidhuber, Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
31. A. Waswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in *NIPS* (2017)
32. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
33. A. Burns, R. Tan, K. Saenko, S. Sclaroff, B.A. Plummer, Language features matter: effective language representations for vision-language tasks, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (2019), pp. 7474–7483
34. K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos. arXiv preprint arXiv:1406.2199 (2014)
35. J. Carreira, A. Zisserman, Quo vadis, action recognition? A new model and the kinetics dataset, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2017), pp. 6299–6308
36. S. Xie, C. Sun, J. Huang, Z. Tu, K. Murphy, Rethinking spatiotemporal feature learning: speed-accuracy trade-offs in video classification, in *Proceedings of the European Conference on Computer Vision (ECCV)*, (2018), pp. 305–321
37. F. Markatopoulou, D. Galanopoulos, I. Patras, V. Mezaris, Iti-certh participation in trecvid 2016. (2016)
38. F. Markatopoulou, D. Galanopoulos, V. Mezaris, I. Patras, Query and keyframe representations for ad-hoc video search, in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, (2017), pp. 407–411
39. K. Ueki, K. Hirakawa, K. Kikuchi, T. Ogawa, T. Kobayashi, Waseda_meisei at trecvid 2017: Ad-hoc video search, in *TRECVID Workshop* (2017)
40. C.G. Snoek, X. Li, C. Xu, D.C. Koelma, University of amsterdam and renmin university at trecvid 2017: searching video, detecting events and describing video, in *TRECVID Workshop* (2017)
41. D. Shao, Y. Xiong, Y. Zhao, Q. Huang, Y. Qiao, D. Lin, Find and focus: retrieve and localize video events with natural language queries, in *Proceedings of the European Conference on Computer Vision (ECCV)*, (2018), pp. 200–216
42. J. Dong, X. Li, C.G. Snoek, Predicting visual features from text for image and video caption retrieval. IEEE Trans. Multimed. **20**(12), 3377–3388 (2018)
43. S.Hershey, S. Chaudhuri, D.P. Ellis, J.F. Gemmeke, A. Jansen, R.C. Moore, M. Plakal, D. Platt, R.A. Saurous, B. Seybold, Cnn architectures for large-scale audio classification, in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (icassp)*,(2017), pp. 131–135. IEEE
44. Y. Yu, H. Ko, J. Choi, G. Kim, End-to-end concept word detection for video captioning, retrieval, and question answering, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2017), pp. 3165–3173
45. Y. Song, M. Soleymani, Polysemous visual-semantic embedding for cross-modal retrieval, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2019), pp. 1979–1988
46. B. Zhang, H. Hu, F. Sha, Cross-modal and hierarchical modeling of video and text, in *Proceedings of the European Conference on Computer Vision (ECCV)*, (2018), pp. 374–390
47. K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
48. Y. Kim, Convolutional neural networks for sentence classification, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics* (2014)
49. S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2017) pp. 1492–1500
50. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), pp. 770–778
51. D.P. Kingma, J. Ba, Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
52. F. Fartash, D. Fleet, J. Kiros, S. Fidler, Vse++: improved visual semantic embeddings, in *British Machine Vision Conference*, (2018), pp. 935–943
53. D.-D. Le, S. Phan, V.-T. Nguyen, B. Renoust, T.A. Nguyen, V.-N. Hoang, T.D. Ngo, M.-T. Tran, Y. Watanabe, M. Klinkigt, Nii-hitachi-uit at trecvid 2016, in *TRECVID* (2016)
54. X. Li, J. Dong, C. Xu, J. Cao, X. Wang, G. Yang, Renmin university of china and zhejiang gongshang university at trecvid 2018: deep cross-modal embeddings for video-text retrieval, in *TRECVID* (2018)
55. M. Foteini, M. Anastasia, G. Damianos, M. Theodoros, K. Vagia, I. Anastasia, S. Symeonidis, Iti-certh participation in trecvid 2016, in *TRECVID 2016 Workshop* (2016)
56. P.-Y. Huang, J. Liang, V. Vaibhav, X. Chang, A. Hauptmann, Informedia@ trecvid 2018: Ad-hoc video search with discrete and continuous representations, in *TRECVID Proceedings*, vol. 70. (2018)
57. J. Liang, J. Chen, P. Huang, X. Li, L. Jiang, Z. Lan, P. Pan, H. Fan, Q. Jin, J. Sun, Informedia@ trecvid 2016, in *TRECVID* (2016)
58. P.A. Nguyen, Q. Li, Z.-Q. Cheng, Y.-J. Lu, H. Zhang, X. Wu, C.-W. Ngo, Vireo@ trecvid 2017: video-to-text, ad-hoc video search, and video hyperlinking, in *TRECVID* (2017)

59. M. Bastan, X. Shi, J. Gu, Z. Heng, C. Zhuo, D. Sng, A.C. Kot, Ntu rose lab at trecvid 2018: Ad-hoc video search and video to text, in *TRECVID* (2018)
60. D. Koelma, C. Snoek, Query understanding is key for zero-example video search, in *TRECVID Workshop* (2017)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.