Prompt Switch: Efficient CLIP Adaptation for Text-Video Retrieval

Chaorui Deng[1,*], Qi Chen[1,*], Pengda Qin[2], Da Chen[3], Qi Wu[1,†]
[1]Australia Institute of Machine Learning, University of Adelaide
[2]Alibaba Group, [3]Department of Computer Science, University of Bath

Abstract

In text-video retrieval, recent works have benefited from the powerful learning capabilities of pre-trained text-image foundation models (e.g., CLIP) by adapting them to the video domain. A critical problem for them is how to effectively capture the rich semantics inside the video using the image encoder of CLIP. To tackle this, state-of-the-art methods adopt complex cross-modal modeling techniques to fuse the text information into video frame representations, which, however, incurs severe efficiency issues in large-scale retrieval systems as the video representations must be recomputed online for every text query. In this paper, we discard this problematic cross-modal fusion process and aim to learn semantically-enhanced representations purely from the video, so that the video representations can be computed offline and reused for different texts. Concretely, we first introduce a spatial-temporal "Prompt Cube" into the CLIP image encoder and iteratively switch it within the encoder layers to efficiently incorporate the global video semantics into frame representations. We then propose to apply an auxiliary video captioning objective to train the frame representations, which facilitates the learning of detailed video semantics by providing fine-grained guidance in the semantic space. With a naive temporal fusion strategy (i.e., mean-pooling) on the enhanced frame representations, we obtain state-of-the-art performances on three benchmark datasets, i.e., MSR-VTT, MSVD, and LSMDC.

1. Introduction

Text-video retrieval [1, 6, 30, 42] is a fundamental task in the area of video-language understanding that seeks to find the most relevant video from a large set of candidates to match a text query. With the rapid growth of video data, text-video retrieval has become increasingly important for various applications, including video recommendation [36, 43], video search [20, 45], and video summarization [7, 29, 31, 33, 37, 47]. Due to the high cost of constructing text-video datasets, one promising approach for this task is to leverage pre-trained text-image foundation models and transfer their powerful representation capabilities to the video domain. Specifically, the CLIP [34] model, which is trained using a text-image alignment objective, is particularly suitable for text-video retrieval and has been frequently studied recently [27, 46, 16]. It has two transformer encoders [38, 9] to process images and texts, respectively. A vector representation is extracted from the input of each modality with the corresponding encoder and is optimized to be close to its paired representation and away from the unpaired ones.

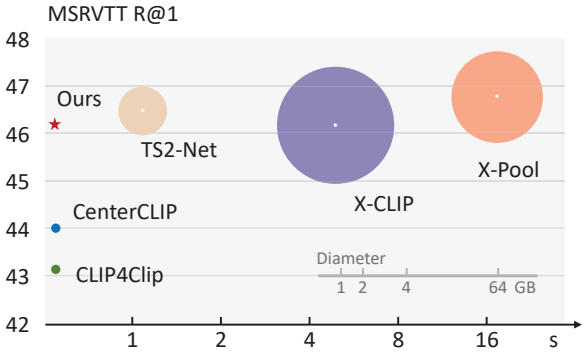


Figure 1: The performance (*i.e.*, R@1), retrieval time, and memory usage during retrieval for baseline models and ours on the MSRVTT dataset. The center of the bubble indicates the value of R@1. The diameter of the bubble or star is proportional to the memory usage (GB) while the horizontal axis indicates the inference time (s).

Adapting CLIP to the video domain is non-trivial and requires careful consideration of both efficiency and effectiveness. In CLIP4Clip [27], the authors directly mean-pool the frame representations extracted by the CLIP image encoder to get the video representation and use it to calculate the cosine similarity with the text representations during retrieval. However, the mean-pooling of frame representations may lose some essential semantic details of the video and hamper the retrieval performance. Thus, more advanced meth-

*Co-first author. †Corresponds to qi.wu01@adelaide.edu.au.
Code: https://github.com/bladewaltz1/PromptSwitch.

ods, such as [14, 16, 18, 24, 28], generate the video representation by applying various cross-modal temporal fusion approaches on the frame representations, using text queries as the condition. While achieving state-of-the-art results, these methods encounter severe efficiency issues in practice, as the text-conditioned fusion of each video has to be performed on-the-fly for every incoming text query. Even with a lightweight fusion module (compared to the CLIP backbone), its computation cost grows geometrically as the number of videos and texts increases.

Formally, given a query set of N_t texts with an average length of N_w words and a candidate set of N_v videos where each video contains N_f frames. Then, the space and time complexities are $\mathcal{O}(N_v N_t N_f)$ for the text-conditioned fusion in X-Pool [16] and TS2-Net [24], and $\mathcal{O}(N_v N_t N_f N_w)$ for that of X-CLIP [28]. While for CLIP4Clip, the complexity is $\mathcal{O}(N_v N_t)$ as it only requires a simple dot-product between the text and mean-pooled frame representations, although its performance is inferior to X-Pool and X-CLIP. To better reveal this gap, we show an example in Figure 1 about the real-world efficiency of several methods while omitting the backbone computation. Here, we set $N_v = 16384$, $N_t = 512$, $N_f = 12$, and $N_w = 10$. From the figure, with large N_t and N_v, the latency and memory consumption for text-conditioned temporal fusion methods [16, 24, 28] are orders of magnitude higher than text-agnostic temporal fusion (*i.e*., mean-pooling) [27, 57], and can rapidly become enormous in large-scale scenarios.

On the other hand, the backbone computation of CLIP is much less of a burden in real-world retrieval systems, as the frame representations of the video can be pre-computed offline and reused for different text queries. Therefore, a more practical CLIP-based text-video retrieval method should focus on improving the backbone representation ability while keeping the cross-modal interaction as simple as possible. Motivated by this, we propose a simple and efficient adaptation method for CLIP to facilitate its ability to capture both the global and detailed semantics of videos.

Concretely, we first feed a tiny ($\sim$0.1M) "**Prompt Cube**" into the image encoder of CLIP, which is a 3D tensor spanning over the spatial, temporal, and channel axis, as shown in the right of Figure 2.[1] It is designed to have the same temporal and spatial sizes and is concatenated with the patch tokens alongside the spatial axis. To propagate temporal semantics among different frames, we *switch* the spatial and temporal dimensions of the prompt cube before each self-attention layer, so that the prompt cube builds up a *peer-to-peer* connection between every two-frame pair. In this way, our modified CLIP model enjoys an improved global semantic modeling ability thanks to the comprehensive spatial-temporal modeling between the prompt cube and the patch tokens of all frames, while only bringing negligible extra parameters and computations. This also allows the prompt cube to serve as a compact summarization of the whole video, and further enables us to design a CLIP-guided Prompt Aggregation approach and obtain the frame representations from the prompt cube. Then, we use naive mean-pooling instead of cross-modal fusion on these frame representations to get the final video representation.

Moreover, since we will not use any fine-grained cross-modal interaction modules in our model, we adopt an Auxiliary Video Captioning objective as an alternative to provide fine-grained guidance in the semantic space when learning video representations. Specifically, we introduce a light captioning head on top of our modified CLIP image encoder during training, which takes the frame representations aggregated from the prompt cube as input and generates the paired text of the input video. This auxiliary objective plays a critical role because CLIP's original contrastive learning objective is relatively easy to fit due to the lack of in-batch negatives during training (which is generally the case in text-video retrieval). During inference, the light captioning branch is removed, thus it incurs no extra computation and memory consumption.

We verify the effectiveness of the proposed method on three text-video retrieval benchmarks, *i.e*. MSR-VTT [50], MSVD [4], and LSMDC [35], where our method consistently achieves state-of-the-art performance while being significantly more efficient than the previous state of the arts. We also provide extensive performance analyses to show the superiority of our proposed method.

2. Related Work

Text-video retrieval is one of the fundamental tasks in video-language modeling and has lots of applications in the industry. Here, we broadly classify the previous works into three categories and briefly review them below.

Off-the-Shelf Single-Modal Representations An early trend in text-video retrieval is to use the off-the-shelf video (*e.g*., I3D [3]) and text representations (*e.g*., GloVe [32]) and design sophisticated feature encoding methods or multi-modal fusion mechanisms on them to improve the retrieval performance [6, 8, 19, 54, 55]. For example, in [6], a hierarchical graph reasoning approach is proposed to decompose video-text matching into global-to-local levels and disentangle texts into a hierarchical semantic graph with three levels of events, actions, and entities. Yu *et al*. [54] propose a joint sequence fusion model for the sequential interaction of videos and texts. Dong *et al*. [8] leverages multi-level single-modal features that represent the rich content of both video and text in a coarse-to-fine fashion. However, their performances are limited due to the mismatched training objectives between the pre-computed representations and the text-video retrieval task.

[1] The channel axis is omitted for simplicity.

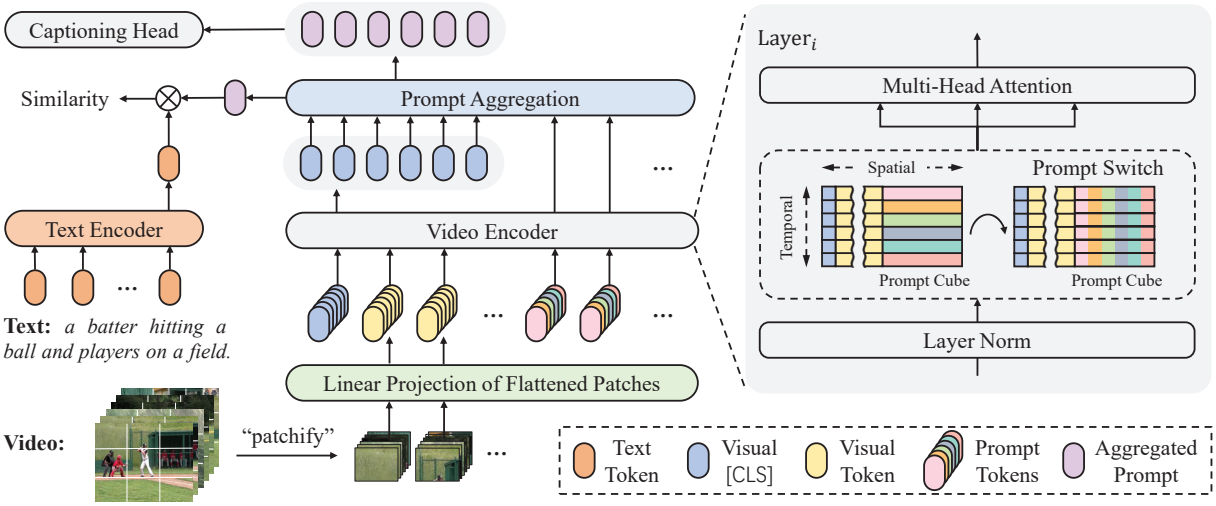


Figure 2: Overall architecture. For a video clip with N_f frames, we accompany it with $N_f \times N_f$ prompt tokens, resulting in a 3D prompt cube. We show the Prompt Switch operation on the right. For simplicity, we omit the feed-forward network and shot-cut connections in the ViT layer. We provide more details for Prompt Aggregation in Figure 4. $\otimes$ is cosine similarity.

Video-Language Pre-training To reduce the gap between pre-training and downstream tasks, large-scale pre-trained video-language representations have been proposed, such as [1, 13, 15, 21, 26, 40, 48, 49, 51]. Most of these models are pre-trained on the large-scale video-text datasets, *e.g.*, HowTo100M [30] and WebVid-2M [1], which boosts promising cross-modal video understanding [22, 23, 26]. One line of works [1, 13, 49] uses two independent encoders for video and text and then projects them into a common latent space. These methods often adopt a contrastive loss to distinguish the paired video-text data. For example, Bain *et al*. [1] design an end-to-end trainable model, aiming to take advantage of both large-scale image and video captioning datasets. Gabeur *et al*. [13] propose a multi-modal transformer that can extract features at different moments and from different modalities (*e.g.*, audio or speech) in a video. The other line of works [21, 26, 48] employs a single cross-modal encoder, which concatenates the video and text sequences as inputs and models them jointly in the transformer, followed by a binary classifier predicting whether these videos and texts are aligned. Despite they can build fine-grained associations between video-text tokens, they need to input each video-text candidate pair into the model for matching score calculation during inference and thus hampers efficiency. Besides, although the idea of video-language pre-training is promising, due to the high cost of collecting wild videos, its scale is generally much smaller than image-language pre-training, leading to an unsatisfactory generalization ability. Thus, like [16, 27], we seek to boost from the image-language pre-training model (*e.g.*, CLIP [34]) for text-video retrieval.

CLIP-based Adaptation Recently, due to the great advantage of CLIP [34] for vision-and-language representation learning, many works [11, 14, 17, 18, 28, 41, 52, 56] seek to transfer the knowledge of CLIP to text-video retrieval tasks. Roughly, the existing works transfer CLIP from views of feature aggregation [12, 16, 27, 28, 57], representation alignment [12, 28, 41], and post pre-training [46, 52]. Specifically, X-Pool [16] designs a cross-modal attention model, seeking to enable the model to only focus on the relevant video frames conditioned on a given text. TS2-Net [24] adapts CLIP by introducing a token shift module and a token selection module, which capture the temporal information and remove unimportant tokens, respectively. X-CLIP [28] calculates both the coarse- and fine-grained similarity for higher retrieval accuracy. While these methods can be effective, their retrieval efficiency is low due to the coupling of video and text in the cross-modal fusion process. A more ideal way should focus on improving the representation ability of the backbone while maintaining the cross-modal interaction as efficiently as possible. We follow this principle in our method.

3. Method

The overview of our method is shown in Figure 2. It is based on a pre-trained CLIP with a ViT [10]-based image encoder. Given an input video, we first split each video frame into fix-sized non-overlapping patches and linearly project them into 1D patch embeddings. Following CLIP, a `[CLS]` embedding is concatenated to the embedding sequence of each frame, which is pre-trained to capture the local semantics within the sequence. Then, in the ViT en-

coder, our proposed "**Prompt Cube**" bridges the global semantic information from the patch embeddings of all frames through a Prompt Switch operation. After that, we feed the output `[CLS]` embeddings of the video frames and the final prompt cube into a Prompt Aggregation module, where the prompt cube is aggregated into 1D vectors according to the `[CLS]` embeddings, and is further enhanced with fine-grained semantics through an Auxiliary Captioning Objective. To ensure an efficient measurement of the text-video similarity, we avoid using cross-modal fusion modules and directly average these aggregated vectors into the final video representation. A simple dot product operation is used to compute the similarity. More details are in the following.

3.1. Prompt Cube for Bridging Global Semantics

At the core of our method is the proposed Prompt Cube. It is designed to capture the rich temporal semantics of the whole video while bringing negligible modifications and computations to the CLIP image encoder. Formally, given an input video clip with N_f frames, we first obtain its patch embeddings $\boldsymbol{V} \in \mathbb{R}^{N_f \times L \times D}$, where L indicates the size of the spatial dimension, *i.e.*, the number of patches divided from a video frame plus one `[CLS]` embedding. D is the embedding size. Then, our prompt cube is constructed as a 3D tensor $\boldsymbol{P} \in \mathbb{R}^{N_f \times N_f \times D}$. At the start of the ViT layer, we concatenate $\boldsymbol{V}$ and $\boldsymbol{P}$ alongside the spatial dimension, denoted by $[\boldsymbol{V}; \boldsymbol{P}] \in \mathbb{R}^{N_f \times (L+N_f) \times D}$, and then process them jointly. The first two dimensions of $\boldsymbol{P}$ (corresponding to the temporal and spatial axis, respectively) have the same size, thus they can be transposed flexibly without altering the shape of $\boldsymbol{P}$. This allows us to further propose an efficient Prompt Switch operation to exchange the local spatial semantics of each frame and the global temporal semantics from the whole video through the prompt cube.

Prompt Switch The proposed Prompt Switch operation can be defined in a one-line formula:

$$\mathcal{T} := [\boldsymbol{V}; \boldsymbol{P}] \rightarrow [\boldsymbol{V}; \boldsymbol{P}^{\top}]. \tag{1}$$

See the right of Figure 2, we apply this operation before every self-attention layer of the ViT encoder. Then, self-attention is performed over the spatial dimension of $[\boldsymbol{V}; \boldsymbol{P}]$, where the i-th row of the prompt cube (denoted by $\boldsymbol{p}_i \in \mathbb{R}^{1 \times N_f \times D}$) and the patch embeddings of the i-th frame (denoted by $\boldsymbol{v}_i \in \mathbb{R}^{1 \times L \times D}$) acquire information from each other. In this way, for two consecutive ViT layers in the encoder, each element in the prompt cube (denoted by $\boldsymbol{p}_{i,j} \in \mathbb{R}^{D}$) is first attached to the i-th frame and communicates with $\boldsymbol{v}_i$, and then switch to the j-th frame and communicates with $\boldsymbol{v}_j$. By repeatedly performing this operation, the whole prompt cube builds up a peer-to-peer connection between every two-frame pair in the video clip, enabling comprehensive temporal modeling.

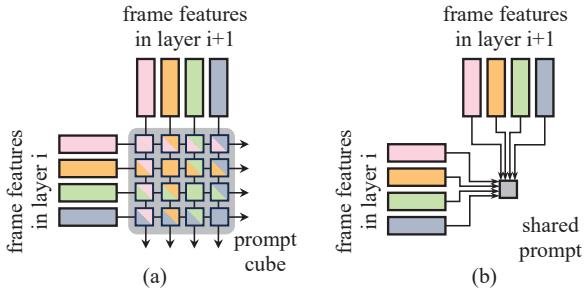


Figure 3: Video Proxy adopts a shared "prompt" to exchange information among all frames (b). Differently, our Prompt Switch method builds up a peer-to-peer connection between every two-frame pair (a) to obtain more comprehensive spatial-temporal modeling. This improves the representation ability of CLIP on video data. Besides, our method may also ease the learning problem as each element in the prompt cube only needs to handle the information of two frames instead of the whole video, thus improving the optimization ability of CLIP on video data.

Some previous text-video retrieval methods have also attempted to introduce temporal adaptation to the backbone of the CLIP ViT encoder, such as Token Shift [24] and Video Proxy [52]. Specifically, the Token Shift method shifts token embeddings from adjacent frames to the current frame, which fails to model the temporal semantics from a global perspective. Moreover, it damages the spatial modeling ability of the original CLIP as the information contained in the shifted tokens is no longer accessible in the current frame. In Video Proxy, the information from all video frames is exchanged using several proxy embeddings, which lack peer-to-peer connections within the frames and thus can be inferior in the temporal modeling capacity. We illustrate the importance of building peer-to-peer connections in Figure 3. Besides, a naive full-attention approach has also been investigated where no adaptation is applied to the CLIP model except allowing its self-attention layers to attend to the patch tokens from the whole video. However, this approach is neither effective due to the domain gap between the input data (video) and the pre-training data (image), nor efficient since the computation complexity of the self-attention layers grows quadratically *w.r.t.* the number of attended patch tokens. We show the superiority of our proposed prompt cube to these previous methods in Section 4.1.

Prompt Aggregation As the prompt cube acquires the global semantics through a comprehensive interaction with all patch embeddings, it can serve as a compact summariza-

tion of the video. Therefore, we propose to obtain the final video representation from the prompt cube, instead of the `[CLS]` embedding as the original CLIP. To achieve this, we design a CLIP-guided Prompt Aggregation module, which aggregates the output prompt cube into 1D vectors according to the final `[CLS]` embeddings of the video frames. It is a lightweight multi-head attention (MHA) layer placed before the last layer normalization (LN) layer of the ViT encoder. The `[CLS]` embedding of the i-th frame (denoted by $\boldsymbol{c}_i \in \mathbb{R}^{1\times D}$) is used as the "query" while the "key" and "value" are both the prompt cube flattened over the temporal and spatial dimensions, denoted by $\hat{\boldsymbol{P}} \in \mathbb{R}^{N_f^2\times D}$. Let $\bar{\boldsymbol{p}}_i \in \mathbb{R}^{1\times D}$ be the aggregated prompt vector for the i-th frame, we have

$$\bar{\boldsymbol{p}}_i := \text{LN}(\boldsymbol{c}_i + \text{MHA}(\boldsymbol{c}_i, \hat{\boldsymbol{P}}, \hat{\boldsymbol{P}})), \quad \forall i \in [1, N_f], \tag{2}$$

where LN denotes the last layer normalization layer of the CLIP ViT encoder. Notably, the last linear projection weight in the MHA layer is initialized as a 0 tensor to ensure that $\bar{\boldsymbol{p}}_i$ is optimized from the original output of the CLIP image encoder. The final video representation is the naive mean-pooling of the normalized prompt vectors, *i.e.*,

$$\boldsymbol{x} := \frac{1}{N_f}\sum_{i=1}^{N_f}\frac{\bar{\boldsymbol{p}}_i}{\|\bar{\boldsymbol{p}}_i\|}. \tag{3}$$

We provide a detailed illustration for the Prompt Aggregation module on the left of Figure 4. To enhance the learning of temporal information for the prompt cube, inspired by [58], we adopt a frame sampling strategy where we randomly sample $k < N_f$ prompt vectors for the final mean-pooling operation in Eq. (3) during training.

3.2. Learning Detailed Semantics via Captioning

As we have discussed in Section 1, the key criterion in the design of our model architecture is to ensure the model contains no extra cross-modal interaction procedure during inference except computing the cosine similarity between text and video representations. While this ensures the efficiency of the similarity measurement in large-scale production systems, it also prevents the video representation from utilizing the fine-grained semantic information from the text query. To aid this, inspired by [5, 53], we propose to learn our video representation with an Auxiliary Captioning objective, which alternatively provides fine-grained guidance in the semantic space during training.

Auxiliary Captioning Head As shown on the right of Figure 4, the Auxiliary Captioning Head, denoted by $\mathcal{H}$, is a stack of M transformer decoders. We first shift the text token by one step to the right, and feed it into a Masked MHA layer, where each text token only attends to its preceding

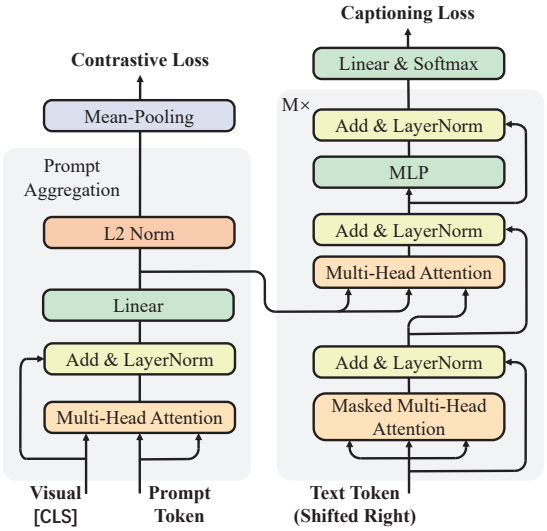


Figure 4: Detailed architecture of Prompt Aggregation (left) and Auxiliary Captioning Head (right).

tokens. Then, another MHA layer is used, where the text tokens attend to the aggregated prompt vectors $\{\bar{\boldsymbol{p}}_i\}_{i=1}^{N_f}$. The model is trained via the autoregressive Teacher Forcing [44] scheme, *i.e.*, the prediction in each step should maximize the likelihood of the token in the next step:

$$\mathcal{L}_{cap} = \sum_{l=1}^{N_w} -\log p(w_l | w_{<l}, \{\bar{\boldsymbol{p}}_i\}_{i=1}^{N_f}), \tag{4}$$

where w_l indicates the l-th token in the text, and $w_{<l}$ denotes the tokens before w_l. N_w is the total number of tokens. Notably, we discard some commonly occurred words (*e.g.*, "a", "an" and "the") by performing a Term Frequency Inverse Document Frequency (TF-IDF) weighting for each word like [39], which ensures that the captioning model focuses only on the informative words. During inference, $\mathcal{H}$ is dropped, thus no extra computation is incurred.

Overall Training Objective Our overall training objective consists of a contrastive loss and a captioning loss. Typically, the contrastive loss is defined as

$$\begin{aligned}
\mathcal{L}_{v2t} &= -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(\boldsymbol{x}_i^\top \boldsymbol{y}_i/\tau)}{\sum_{j=1}^{B}\exp(\boldsymbol{x}_i^\top \boldsymbol{y}_j/\tau)}, \\
\mathcal{L}_{t2v} &= -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(\boldsymbol{y}_i^\top \boldsymbol{x}_i/\tau)}{\sum_{j=1}^{B}\exp(\boldsymbol{y}_i^\top \boldsymbol{x}_j/\tau)}, \\
\mathcal{L}_{con} &= \frac{1}{2}(\mathcal{L}_{v2t} + \mathcal{L}_{t2v}),
\end{aligned} \tag{5}$$

where B is the size of the mini-batch, τ is a learnable temperature parameter. $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are the video representa-

Method	R@1↑	R@5↑	R@10↑	MnR↓
baseline CLIP	43.1	70.9	80.4	16.0
+ Prompt Switch	44.8 (+1.7)	71.9	80.9	15.2
+ Temporal Transformer	44.2	71.4	81.1	15.6
+ Prompt Aggregation	45.4 (+0.6)	71.9	81.1	14.9
+ Captioning Loss	**46.1** (+0.7)	**72.8**	**81.8**	**14.4**

Table 1: Performance analysis of our model components for text-video retrieval on MSRVTT 1K-A test set. Both Prompt Aggregation and Temporal Transformer are applied on top of our Prompt Switch. The Captioning Loss is applied only on top of the Prompt Aggregation.

Method	R@1 ↑	R@5 ↑	R@10 ↑	MnR ↓
Temporal Transformer	44.2	71.4	81.1	15.6
Token Shift [24]	43.2	70.7	79.8	15.9
Video Proxy [52]	45.2	71.0	**81.5**	15.3
Full Attention	43.8	71.4	80.9	16.4
Prompt Switch & Aggregation	**45.4**	**71.9**	81.1	**14.9**

Table 2: Comparisons with other temporal modeling methods for text-video retrieval on MSRVTT 1K-A test set.

tion (obtained by Eq. (3) and text representation of the i-th video-text pair in the mini-batch. The final loss function is

$$\mathcal{L} = \mathcal{L}_{con} + \lambda \mathcal{L}_{cap}, \tag{6}$$

where λ is the weighting hyper-parameter.

4. Experiment

Dataset and Metrics We evaluate the effectiveness of our method on three widely used text-video retrieval datasets, including MSR-VTT [50], MSVD [4] and LSMDC [35]. **MSR-VTT** [50] consists of 10,000 videos with 200K descriptions. We follow the setting in previous works [13, 54], training our models with 9,000 videos and evaluating them on the 1K-A test set. **MSVD** [4] contains $1,970$ videos with about 120K captions, where the train, validation, and test splits contain $1,200$, 100 and 670 videos, respectively. **LSMDC** [35] has $118,081$ video-caption pairs, where $109,673$ videos are used for training, $7,408$ videos for validation, and $1,000$ videos for testing. Following [16, 24, 27], we report the results of Recall@K (R@K, $K = 1, 5, 10$), and Mean Rank (MnR) for quantitative evaluation. Besides, we sum the value of R@1, R@5, and R@10 for both text-video and video-text retrieval tasks as an overall evaluation metric, named Meta Sum.

Implementation Details Following previous work [16, 27], both the text and video encoders are initialized with the pre-trained CLIP (ViT-B/32) [34]. The prompt cube is randomly initialized from a Gaussian distribution with a zero mean and 0.02 std. During training, we uniformly sample

Method	R@1 ↑	R@5 ↑	R@10 ↑	Mem. / Time
Text ⇒ Video				
TS2-Net* [24]	46.5	73.6	**83.3**	0.3G / 0.2s
X-Pool [16]	46.9	72.8	82.2	3.9G / 3.9s
ours (mean pool)	46.1	72.8	81.8	**2.0**M / **8.1**ms
ours (attention pool)	46.4	72.9	82.4	0.2G / 0.2s
ours (top-3 pool)	46.7	73.4	82.0	0.3G / 0.3s
ours (X-Pool [16])	**47.8**	**73.9**	82.2	3.9G / 3.9s
Video ⇒ Text				
TS2-Net* [24]	44.5	73.8	83.2	0.3G / 0.2s
X-Pool [16]	44.4	73.3	84.0	3.9G / 3.9s
ours (mean pool)	44.8	73.7	82.4	**2.0**M / **8.1**ms
ours (attention pool)	45.4	73.9	83.2	0.2G / 0.2s
ours (top-3 pool)	45.2	73.6	83.7	0.3G / 0.3s
ours (X-Pool [16])	**46.0**	**74.3**	**84.8**	3.9G / 3.9s

Table 3: Performance and complexity comparisons with different temporal fusion methods on MSRVTT 1K-A test set. * denotes the results reproduced by official code & setting.

6 frames from each video and resize all video frames into 224×224. Therefore, the size of the spatial and temporal dimensions of the prompt cube is set to 6 by default. While for the testing, 12 frames are used as in previous works, so we split them into two 6-frame chunks through interval sampling to make the temporal dimension compatible with the prompt cube. The Prompt Aggregation is applied on the prompt cubes of all chunks. The number of decoder layers in the captioning head is 3. The hyper-parameter k used for frame sampling in Eq. (3) is set to 3. We set the hyper-parameter $\lambda = 0.5$ in Eq. (6). The training epochs are 10 for all the datasets with a batch size of 128. We use AdamW optimizer [25] with a learning rate of 3e-5 and adopt a cosine decay strategy for the learning rate.

4.1. Performance Analysis

We first thoroughly analyze the model designs and critical components of our proposed method and then verify their effectiveness. The experiments are conducted on the MSRVTT dataset and evaluated on the 1K-A test set.

Model Components As shown in Table 1, we conduct an ablation study on the Prompt Switch, Prompt Aggregation, and Auxiliary Captioning Objective by introducing them gradually into the model for text-video retrieval. The baseline CLIP model directly uses the mean-pooled `[CLS]` embeddings of the video frames as the final video representation, and is trained using only contrastive loss. Specifically, when incorporating the Prompt Switch mechanism, the result of R@1 increases significantly (*i.e.*, from 43.1 to 44.8) compared with the baseline CLIP, which demonstrates the effectiveness of the proposed Prompt Switch. With the help of Prompt Aggregation, it further improves 0.6 and

Methods	Text ⇒ Video				Video ⇒ Text				Meta Sum ↑
	R@1 ↑	R@5 ↑	R@10 ↑	MnR ↓	R@1	R@5↑	R@10↑	MnR ↓	
cross-modal temporal fusion									
CLIP2TV [14]	46.1	72.5	82.9	15.2	43.9	73.0	82.8	11.1	401.2
CLIP2Video [12]	45.6	72.6	81.7	14.6	43.3	72.3	82.1	10.2	397.6
TS2-Net* [24]	46.5	73.6	83.3	13.9	44.5	73.8	83.2	9.2	404.9
EMCL [18]	46.8	73.1	83.1	-	46.5	73.5	83.5	-	406.5
X-CLIP [28]	46.1	73.0	83.1	13.2	46.8	73.3	84.0	9.1	406.3
DRL [41]	47.4	74.6	83.8	-	45.3	73.9	83.3	-	408.3
X-Pool [16]	46.9	72.8	82.2	14.3	44.4	73.3	84.0	9.0	403.6
ours (X-Pool)	**47.8**	**73.9**	82.2	14.1	46.0	**74.3**	**84.8**	**8.5**	**409.0**
text-agnostic temporal pooling									
CLIP4Clip† (seqTransf) [27]	44.5	71.4	81.6	15.3	42.7	70.9	80.6	11.6	391.7
CenterCLIP† (spectral) [57]	44.2	71.6	82.1	15.1	42.8	71.7	82.2	11.1	394.6
X-CLIP (mean pool) [28]	43.0	70.7	81.6	16.3	43.0	70.2	81.2	11.5	389.7
TS2-Net* (mean pool) [24]	44.4	72.1	**82.2**	14.6	43.7	70.8	80.4	11.6	393.6
ours (mean pool)	**46.1**	**72.8**	81.8	**14.4**	**44.8**	**73.7**	**82.4**	**9.9**	**401.6**

Table 4: Comparisons with state-of-the-arts on MSRVTT. † both CLIP4Clip and CenterCLIP have multiple versions in their papers, here we choose the versions with the highest Meta Sum. * denotes the results reproduced by official code & setting.

Methods	Text ⇒ Video				Video ⇒ Text				Meta Sum ↑
	R@1 ↑	R@5 ↑	R@10 ↑	MnR ↓	R@1	R@5↑	R@10↑	MnR ↓	
cross-modal temporal fusion									
CLIP2TV [14]	47.0	76.5	85.1	10.1	-	-	-	-	-
CLIP2Video [12]	47.0	76.8	85.9	9.6	58.7	85.6	91.6	4.3	445.6
X-CLIP [28]	47.1	77.8	-	9.5	60.9	87.8	-	4.7	-
X-Pool [16]	47.2	77.4	86.0	9.3	66.4	90.0	94.2	3.3	461.2
text-agnostic temporal pooling									
CLIP4Clip† (seqTransf) [27]	45.2	75.5	84.3	10.3	62.0	87.3	92.6	4.3	446.9
CenterCLIP† (spectral) [57]	**47.4**	76.5	85.2	9.7	62.7	88.1	92.8	4.1	452.7
ours (mean pool)	47.1	**76.9**	**86.1**	**9.5**	**68.5**	**91.8**	**95.6**	**2.8**	**466.0**

Table 5: Comparisons with state-of-the-arts on MSVD. † both CLIP4Clip and CenterCLIP have multiple versions in their papers, here we choose the versions with the highest Meta Sum values.

achieves 45.4 in R@1. Our final model containing the Auxiliary Captioning Loss further improves the performance in R@1, R@5, R@10, and MnR by a large margin. These results demonstrate that all the proposed components contribute clearly to the final performance. Besides, we also compare the performance of an **alternative** temporal aggregation method to Prompt Aggregation termed Temporal Transformer, which adopts an additional transformer layer on the `[CLS]` embeddings of all video frames and averages the output. From Table 1, Prompt Aggregation is better than Temporal Transformer on all metrics.

Comparison on Temporal Modeling Methods To investigate the effectiveness of our temporal modeling method, *i.e.*, Prompt Switch + Prompt Aggregation, we compare it with other four temporal modeling methods, including the Temporal Transformer, Full Attention (attention over patch tokens from all video frames), Token Shift [24], and Video Proxy [52]. All methods are implemented on the baseline CLIP model without extra components like the Token Selection Transformer in [24], and are trained using our default setting. From Table 2, our Prompt Switch & Aggregation approach outperforms all the baselines on R@1, R@5, and MnR while achieving the second-best result on R@10, which demonstrates its superiority in learning global video semantics across frames.

Cross-Modal Temporal Fusion *vs*. Naive Mean Pooling While our method is able to achieve significantly better performance than the baseline CLIP using the mean-pooling setting, we further investigate whether it is compatible with the advanced cross-modal temporal fusion methods, namely, attention pooling, top-3 pooling, and X-Pool [16]. We simply replace the final mean-pooling in Eq. (3) with these cross-modal approaches and show the performance in Table 3. From the table, when adopting these cross-modal fusion methods, our models obtain clearly boosted performance, especially for X-Pool, which obtains the best per-

Methods	Text ⇒ Video				Video ⇒ Text				Meta Sum ↑
	R@1↑	R@5↑	R@10↑	MnR↓	R@1↑	R@5↑	R@10↑	MnR↓	
cross-modal temporal fusion									
X-Pool [16]	25.2	43.7	53.5	53.2	22.7	42.6	51.2	47.4	238.9
TS2-Net [24]	23.4	42.3	50.9	56.9	-	-	-	-	-
EMCL [18]	23.9	42.4	50.9	-	22.2	40.6	49.2	-	229.2
X-CLIP [28]	23.3	43.0	-	56.0	22.5	42.2	-	50.7	-
text-agnostic temporal pooling									
CLIP4Clip† (seqLSTM) [27]	21.6	41.8	49.8	58.0	20.9	40.7	49.1	53.9	223.9
CenterCLIP† (k-medoids++) [57]	21.9	41.1	**50.7**	**55.6**	21.1	41.2	50.2	**48.7**	226.2
ours (mean pool)	**23.1**	**41.7**	50.5	56.8	**22.0**	40.8	**50.3**	51.0	**228.4**

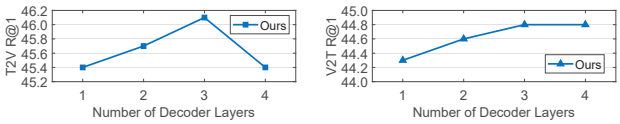
Table 6: Comparisons with state-of-the-arts on LSMDC. † both CLIP4Clip and CenterCLIP have multiple versions in their papers and here we choose the versions with the highest Meta Sum values.

formance for both text-video and video-text retrieval and outperforms the original X-Pool by a large margin. Moreover, even compared with the state-of-the-art cross-modal fusion methods, our model with naive mean-pooling is still competitive for both text-video and video-text retrieval.

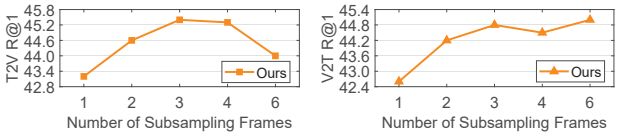
We further measure the memory usage and latency of the compared methods during inference on the test set. To make the experiment setting close to real-world scenarios and for fair comparisons, we use the same pre-computed frame and text representations while only monitoring space and time consumption for the ranking procedure. As shown in the last column of Table 3, our model with mean-pooling is orders of magnitude more efficient than those baseline models with cross-modal temporal fusion. This conclusion is generalizable to other datasets, *i.e.*, the same model will always have the same ranking efficiency if using the same inference and evaluation settings. This reveals the importance of using text-agnostic temporal fusion (*e.g.*, mean-pooling) for real-world text-video retrieval.

4.2. Comparison with the State-of-the-arts

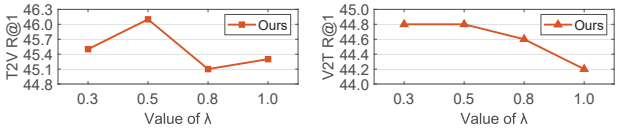
In this section, we compare our proposed method with state-of-the-art methods on MSRVTT, MSVD, and LSMDC dataset. To better reveal the performance gap among the compared methods, we do not consider post-processing techniques like QB-NORM [2]. The results of both text-video and video-text retrieval tasks are presented in Tables 4, 5 and 6. From the tables, when compared under the text-agnostic temporal fusion setting, our model outperforms the baseline models on most of the evaluation metrics, especially for Meta Sum, where it achieves the best performance on all three datasets. Specifically, on MSRVTT, the Meta Sum of our model is 7 points better than the second-best model; on MSVD, our method outperforms CenterCLIP [57] by 13.3; On LSMDC, our model consistently achieves the highest Meta Sum. Moreover, when compared with methods using cross-modal temporal fusion, our model with mean-pooling is still competitive in terms



(a) R@1 performance with different number of decoder layers



(b) R@1 performance with different number of subsampling frames



(c) R@1 performance with different value of hyper-parameter λ

Figure 5: We show the effects of (a) the number of captioning decoder layers; (b) the number of subsampling frames k in training; (c) the values of hyper-parameter λ in Eq. (6).

of Meta Sum, while being much more efficient in practice (refer to Figure 1 and Table 3 for detailed discussions). Specifically, on MSRVTT, it outperforms CLIP2TV [14] and CLIP2Video [12]; on MSVD, it surpasses all compared baselines and improves over the second-best model (X-Pool [16]) by 4.8; on LSMDC, it is still comparable with EMCL [18]. These results show that our proposed method has a better trade-off between performance and efficiency, and is more suitable for large-scale production systems.

4.3. Further Discussions

Number of Decoder Layers Generally, more layers in the captioning decoder would improve its capacity for learning fine-grained knowledge, which, however, also makes it easier to over-fit. To find a better trade-off, we study the

effect of the number of layers. As shown in Figure 5(a), the performance of our model improves and reaches the peak with 3 layers of decoder for both text-video and video-text retrieval tasks. Thus, we set the number of decoder layers to 3 for all the experiments.

Number of Subsampling Frames In practice, we set the number of training frames as 6 and validation frames as 12. Then, we randomly sample k frames before mean-pooling for training while using all 12 frames for evaluation. Notably, if $k = 6$, that means we average all the training frames in the mean-pooling operation. In Figure 5(b), we observe that the performance improves when reducing the value of k and achieves the peak when $k = 3$, which demonstrates its effectiveness. While further decreasing the value of k, the mean-pooling operation may lose a great deal of semantics derived from frames, which causes performance degradation. Therefore, we set $k = 3$ for all the experiments.

Hyper-parameter λ in Eq. (6) From Figure 5(c), in the case of a small λ (*e.g.*, $\lambda = 0.3$), the model can only achieve suboptimal performance due to insufficient exploitation of the detailed semantics. When increasing the value of λ, the performance of our model peaks at the $\lambda = 0.5$ and degrades thereafter. Considering the trade-off between typical contrastive loss and captioning loss, we choose the weighting parameter λ as 0.5 for all the datasets.

5. Conclusion

In this paper, we tackle the task of text-video retrieval, where we aim to learn semantically-enhanced representations purely from the video, allowing for offline computation and reuse for different text queries. Our method introduces a new Prompt Cube into the CLIP image encoder, which is iteratively transposed within the encoder layers to incorporate global video semantics into frame representations. We also adopt an auxiliary video captioning objective to optimize the frame representations, providing detailed guidance in the semantic space. With mean pooling fusion on the enhanced frame representations, the proposed model achieves SoTA performance on three benchmark datasets. Comprehensive experiments verify the effectiveness of all critical components of our proposed method.

References

[1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, pages 1728–1738, 2021.

[2] Simion-Vlad Bogolin, Ioana Croitoru, Hailin Jin, Yang Liu, and Samuel Albanie. Cross modal retrieval with querybank normalisation. In *CVPR*, pages 5194–5205, 2022.

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017.

[4] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, pages 190–200, 2011.

[5] Qi Chen, Chaorui Deng, and Qi Wu. Learning distinct and representative modes for image captioning. *NeurIPS*, 35:9472–9485, 2022.

[6] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *CVPR*, pages 10638–10647, 2020.

[7] Chaorui Deng, Shizhe Chen, Da Chen, Yuan He, and Qi Wu. Sketch, ground, and refine: Top-down dense video captioning. In *CVPR*, pages 234–243, 2021.

[8] Jianfeng Dong, Xirong Li, Chaoxi Xu, Xun Yang, Gang Yang, Xun Wang, and Meng Wang. Dual encoding for video retrieval by text. *IEEE TPAMI*, pages 4065–4080, 2021.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[11] Bo Fang, Chang Liu, Yu Zhou, Min Yang, Yuxin Song, Fu Li, Weiping Wang, Xiangyang Ji, Wanli Ouyang, et al. Uatvr: Uncertainty-adaptive text-video retrieval. *arXiv preprint arXiv:2301.06309*, 2023.

[12] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021.

[13] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, pages 214–229, 2020.

[14] Zijian Gao, Jingyu Liu, Weiqi Sun, Sheng Chen, Dedan Chang, and Lili Zhao. Clip2tv: Align, match and distill for video-text retrieval. *arXiv preprint arXiv:2111.05610*, 2021.

[15] Yuying Ge, Yixiao Ge, Xihui Liu, Dian Li, Ying Shan, Xiaohu Qie, and Ping Luo. Bridging video-text retrieval with multiple choice questions. In *CVPR*, pages 16167–16176, 2022.

[16] Satya Krishna Gorti, Noël Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu. X-pool: Cross-modal language-video attention for text-video retrieval. In *CVPR*, pages 5006–5015, 2022.

[17] Haojun Jiang, Jianke Zhang, Rui Huang, Chunjiang Ge, Zanlin Ni, Jiwen Lu, Jie Zhou, Shiji Song, and Gao Huang. Cross-modal adapter for text-video retrieval. *arXiv preprint arXiv:2211.09623*, 2022.

[18] Peng Jin, Jinfa Huang, Fenglin Liu, Xian Wu, Shen Ge, Guoli Song, David A Clifton, and Jie Chen. Expectation-maximization contrastive learning for compact video-and-language representations. *NeurIPS*, 2022.

[19] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[20] Miroslav Kratochvil, František Mejzlík, Patrik Veselỳ, Tomáš Soućek, and Jakub Lokoć. Somhunter: lightweight video search system with som-guided relevance feedback. pages 4481–4484, 2020.

[21] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, pages 7331–7341, 2021.

[22] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *CVPR*, pages 4953–4963, 2022.

[23] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *EMNLP*, pages 2046–2065, 2020.

[24] Yuqi Liu, Pengfei Xiong, Luhui Xu, Shengming Cao, and Qin Jin. Ts2-net: Token shift and selection transformer for text-video retrieval. In *ECCV*, pages 319–335, 2022.

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019.

[26] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020.

[27] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022.

[28] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *ACM MM*, pages 638–647, 2022.

[29] Safa Messaoud, Ismini Lourentzou, Assma Boughoula, Mona Zehni, Zhizhen Zhao, Chengxiang Zhai, and Alexander G Schwing. Deepqamvs: Query-aware hierarchical pointer networks for multi-video summarization. In *SIGIR*, pages 1389–1399, 2021.

[30] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, pages 2630–2640, 2019.

[31] Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. Clip-it! language-guided video summarization. *NeurIPS*, pages 13988–14000, 2021.

[32] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[33] Bryan A Plummer, Matthew Brown, and Svetlana Lazebnik. Enhancing video summarization via vision-language embedding. In *CVPR*, pages 5781–5789, 2017.

[34] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021.

[35] Anna Rohrbach, Marcus Rohrbach, and Bernt Schiele. The long-short story of movie description. In *Pattern Recognition: 37th German Conference, GCPR 2015, Aachen, Germany, October 7-10, 2015, Proceedings 37*, pages 209–221, 2015.

[36] Lei Sang, Min Xu, Shengsheng Qian, Matt Martin, Peter Li, and Xindong Wu. Context-dependent propagating-based video recommendation in multimodal heterogeneous information networks. *IEEE TMM*, pages 2019–2032, 2020.

[37] Yassir Saquil, Da Chen, Yuan He, Chuan Li, and Yong-Liang Yang. Multiple pairwise ranking networks for personalized video summarization. In *ICCV*, pages 1718–1727, 2021.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[39] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575, 2015.

[40] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. *arXiv preprint arXiv:2203.07303*, 2022.

[41] Qiang Wang, Yanhao Zhang, Yun Zheng, Pan Pan, and Xian-Sheng Hua. Disentangled representation learning for text-video retrieval. *arXiv preprint arXiv:2203.07111*, 2022.

[42] Xiaohan Wang, Linchao Zhu, and Yi Yang. T2vlad: global-local sequence alignment for text-video retrieval. In *CVPR*, pages 5079–5088, 2021.

[43] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video. In *ACM MM*, pages 1437–1445, 2019.

[44] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[45] Jiaxin Wu and Chong-Wah Ngo. Interpretable embedding for ad-hoc video search. In *ACM MM*, pages 3357–3366, 2020.

[46] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. Cap4video: What can auxiliary captions do for text-video retrieval? *arXiv preprint arXiv:2301.00184*, 2022.

[47] Shuwen Xiao, Zhou Zhao, Zijian Zhang, Ziyu Guan, and Deng Cai. Query-biased self-attentive network for query-focused video summarization. *IEEE TIP*, pages 5889–5899, 2020.

[48] Hu Xu, Gargi Ghosh, Po-Yao Huang, Prahal Arora, Masoumeh Aminzadeh, Christoph Feichtenhofer, Florian Metze, and Luke Zettlemoyer. Vlm: Task-agnostic video-language model pre-training for video understanding. *ACLliu2021hit*, 2021.

[49] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *EMNLP*, pages 6787–6800, 2021.

[50] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, pages 5288–5296, 2016.

[51] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *CVPR*, pages 5036–5045, 2022.

[52] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. *ICLR*, 2023.

[53] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[54] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, pages 471–487, 2018.

[55] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, pages 3165–3173, 2017.

[56] Bowen Zhang, Xiaojie Jin, Weibo Gong, Kai Xu, Zhao Zhang, Peng Wang, Xiaohui Shen, and Jiashi Feng. Multimodal video adapter for parameter efficient video text retrieval. *arXiv preprint arXiv:2301.07868*, 2023.

[57] Shuai Zhao, Linchao Zhu, Xiaohan Wang, and Yi Yang. Centerclip: Token clustering for efficient text-video retrieval. In *SIGIR*, pages 970–981, 2022.

[58] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *ICCV*, pages 408–417, 2017.