VoP: Text-Video Co-operative Prompt Tuning for Cross-Modal Retrieval

Siteng Huang[1,3]*, Biao Gong[2], Yulin Pan[2], Jianwen Jiang[2], Yiliang Lv[2], Yuyuan Li[3], Donglin Wang[1]†
[1]Machine Intelligence Lab (MiLAB), AI Division, School of Engineering, Westlake University
[2]Alibaba Group [3]Zhejiang University
{huangsiteng, wangdonglin}@westlake.edu.cn, y2li@zju.edu.cn,
a.biao.gong@gmail.com, {yanwen.pyl, jianwen.jjw, yiliang.lyl}@alibaba-inc.com

Abstract

Many recent studies leverage the pre-trained CLIP for text-video cross-modal retrieval by tuning the backbone with additional heavy modules, which not only brings huge computational burdens with much more parameters, but also leads to the knowledge forgetting from upstream models. In this work, we propose the VoP: Text-Video Co-operative Prompt Tuning for efficient tuning on the text-video retrieval task. The proposed VoP is an end-to-end framework with both video & text prompts introducing, which can be regarded as a powerful baseline with only 0.1% trainable parameters. Further, based on the spatio-temporal characteristics of videos, we develop three novel video prompt mechanisms to improve the performance with different scales of trainable parameters. The basic idea of the VoP enhancement is to model the frame position, frame context, and layer function with specific trainable prompts, respectively. Extensive experiments show that compared to full fine-tuning, the enhanced VoP achieves a 1.4% average R@1 gain across five text-video retrieval benchmarks with 6× less parameter overhead. The code will be available at https://github.com/bighuang624/VoP.

1. Introduction

Due to the remarkable progress in large-scale contrastive language-image pre-training [16, 21, 22, 31], a recent popular direction for the crucial text-video cross-modal retrieval [9,25,34,36] task is to transfer pre-trained image-text knowledge to the video domain [10,27,40] with fine-tuning. However, the dominant full fine-tuning strategy inevitably forgets the useful knowledge acquired in the large-scale pre-training phase and poses a risk of overfitting, as the entire model is updated with limited downstream data. Moreover, full fine-tuning requires to maintain an independent model weight for every dataset during deployment, which becomes infeasible due to the increasing model capacity.

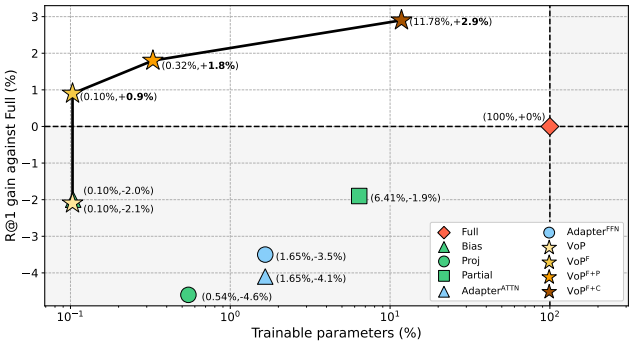

Figure 1. **Fine-tuning comparison of our proposed methods and full fine-tuning (Full)**. For each method, we represent the R@1 (recall at rank 1) gain on the MSR-VTT-9k dataset together with the number of trainable parameters. And we show only a part of our proposed methods for clarity, labeled with ☆. More detailed results are reported in Sec. 4.2.

In this paper, we introduce *prompt tuning* [20, 24] to address the challenges that limit the transferability and generalizability. Keeping the backbone frozen and only tuning a few extra parameters prepended to the input, prompt tuning has been widely applied as a flexible and light-weight fine-tuning protocol. Compared to uni-modal applications [1, 23], text-video cross-modal retrieval requires more parameters to support the dual-branch structure, making it logical to benefit from the parameter-efficient tuning strategy. In addition, different from text descriptions that compose sequential information from words, video-understanding requires summarizing information in both the spatial and temporal dimensions. Therefore, we assume that designing non-trivial video prompts further contributes to prompting both branches for mutual promotion.

According to the above discussion, we propose the **VoP**: Text-**V**ideo C**o**-operative **P**rompt Tuning to simultaneously introduce tunable prompts in both textual and visual encoders. Also, different from existing related efforts [18] that

*Work done during internship at Alibaba DAMO Academy.
†Corresponding author.

only insert prompt vectors into the input textual sequences, we find that preparing prompts for every layer of both encoders can further close the gap to full fine-tuning. As observed in Fig. 1, VoP achieves competitive or superior performance than other efficient tuning protocols with only **0.1%** parameter storage.

To exploit essential video-specific information, we further design three novel video prompts from different perspectives, which can seamlessly replace conventional visual prompts in VoP. Specifically, (1) **position-specific** video prompts model the information shared between frames at the same relative position. (2) Generated **context-specific** video prompts integrate injected contextual message from the frame sequence into the intra-frame modeling. (3) And **function-specific** video prompts adaptively assist to learn intra- or inter-frame affinities by sensing the transformation of layer functions. By exploring video-specific prompts, VoP offers a new way to transfer pre-trained foundation models to the downstream video domain.

We compare our solutions with popular tuning strategies on MSR-VTT [37] (both 9k and 7k splits), DiDeMo [14], ActivityNet [13] and LSMDC [33]. Learning video-specific information while maintaining the pre-trained knowledge, our video prompts deliver an average R@1 improvement of up to 4.2% for VoP, and therefore exceed full fine-tuning by up to 1.4% with much fewer trainable parameters. In summary, the main contributions of our work are three-fold:

- We propose the VoP as a strong baseline that effectively adapts CLIP to text-video retrieval with negligible trainable parameters.
- To exploit video-specific information, we further develop three video prompts respectively conditioned on the frame position, frame context, and layer function.
- Extensive experiments on five text-video retrieval benchmarks demonstrate that various combinations of our video prompts effectively enhance VoP, outperforming full fine-tuning with much less parameter overhead.

2. Related Work

Contrastive Vision-Language Pre-Training. Benefiting from large-scale visual and textual pairs collected from the Internet, learning visual representation under natural language supervision has attracted considerable attention. As a representative example, consuming 400 million pairs of images and texts, CLIP (Contrastive Language-Image Pre-training) [31] matches relevant image-text pairs via training two uni-modal encoders with a contrastive loss. And the success of derivative works demonstrates the potential of adapting pre-trained vision-language models like CLIP [31], ALIGN [16] and ALBEF [21] to various downstream applications [8, 27, 30]. In the video-language understanding area, some existing efforts such as HowTo100M [29] and Frozen in Time [2] have attempted to pre-train on large-scale video datasets, aiming to improve video-text representations for downstream tasks. Despite the progress made, the extremely noisy text supervision of instructional videos requires a much larger scale of video-language pre-training to achieve competitive results. In this work, we follow CLIP4Clip [27] to explore the adaptation of pre-trained CLIP to the text-video retrieval task.

Text-Video Retrieval. Aiming to match semantically similar samples across text and video modalities, text-video retrieval methods commonly apply a dual-branch structure to align the uni-modal features extracted by individual encoders. While most early efforts designed dedicated cross-modal fusion mechanisms after extracting offline features [9, 25, 36, 38], task-specific end-to-end fine-tuning from large-scale pre-trained models has recently achieved noticeable results. For example, ClipBERT [19] suggested that end-to-end fine-tuning with just a few sparsely sampled clips could outperform using densely extracted offline features from full-length videos. CLIP4Clip [27] investigated three similarity calculation mechanisms based on pre-trained CLIP, and further post-pretrained the CLIP on large-scale video-text data to improve both zero-shot and fine-tuned performance. And X-Pool [10] employed cross-modal attention for a text to attend its most semantically similar frames, thus generating text-conditioned video representations for retrieval. Different from the above methods, we seek harmony between efficacy and parameter efficiency. Around prompt tuning, we propose a series of solutions that reduce the overhead of trainable parameters while maintaining promising performance, thus decreasing the difficulties of adaption.

Prompt Learning. Stemming from advances in natural language processing (NLP), prompt learning initially fills the sample into properly handcrafted prompt templates, so that a pre-trained language model can "understand" the task [5]. However, designing handcrafted templates requires extensive expert knowledge and limits the flexibility. Therefore, follow-up works treat prompts as task-specific continuous vectors and directly optimize them during fine-tuning, known as *prompt tuning* [20, 24]. Inspired by CLIP that embeds the textual labels of to-be-recognized objects into descriptive texts for image recognition, CoOp [41] applies trainable text prompts to promote few-shot image classification. As such procedure can be easily transformed into the form of various vision-language problems, text prompts have been adopted for video-understanding tasks including action recognition, action localization and text-video retrieval [18, 35]. Recently, by inserting prompt tokens into the patch token sequence or padding prompt pixels for the input image, pioneer works have successfully applied the prompt tuning method to vision backbones [1,

17]. However, whether visual prompt tuning is effective to the video domain and multi-modal applications remains untouched. In this paper, we not only study prompt tuning for co-operative uni-model encoders, but also explore the video prompts that model a variety of video-specific information, which is the first time to our knowledge.

3. Methodology

In this section, we begin with a review of adapting the pre-trained CLIP to text-video retrieval (Sec. 3.1). Then we introduce our proposed VoP that collaboratively promote the cross-modal alignment with negligible trainable parameters (Sec. 3.2). Finally, we further devise a series of video prompts with the consideration of the inherent nature of video, leading to superior performance than full fine-tuning (Sec. 3.3). The overall framework is illustrated in Fig. 2.

3.1. Preliminary

Problem Formulation. Given the text set $\mathcal{T}$ and video set $\mathcal{V}$, the objective of text-video retrieval is to learn a similarity function s, which produces a high similarity score $s(t, v)$ if a text $t \in \mathcal{T}$ and a video $v \in \mathcal{V}$ are semantically similar, while producing a low score for an irrelevant video-text pair. Then we can rank all videos according to the query text for text-to-video retrieval (denoted as $t2v$), or rank all texts according to the query video for video-to-text retrieval (denoted as $v2t$). In this paper, we define a text t as a sequence of N tokenized words, and a video $v \in \mathbb{R}^{F\times 3\times H\times W}$ as a sequence of F sampled image frames in time.

Revisiting CLIP-based Solution. Following the recent works [10, 27, 40], our work applies CLIP [31] as the pre-trained backbone to benefit from its strong downstream potential. Due to the large-scale contrastive image-text pre-training, the textual and visual encoders of CLIP share a joint latent space, where cross-modal embeddings from a relevant pair can be well aligned. Specifically, the **text** encoder first tokenizes the input text description into the word sequence, and then projects them into word embeddings $\mathbf{W}_0 = \{\mathbf{w}_0^1, \mathbf{w}_0^2, \cdots, \mathbf{w}_0^N\} \in \mathbb{R}^{N\times d^t}$. $\mathbf{W}_0$ is fed into a K-layer Transformer with the architecture modifications described in BERT [32], and for the i-th layer $\mathcal{L}_i^t$,

$$\mathbf{W}_i = \mathcal{L}_i^t(\mathbf{W}_{i-1}) \qquad i = 1, 2, \cdots, K. \tag{1}$$

And the final text embedding $\mathbf{z}^t \in \mathbb{R}^d$ is obtained by projecting the last token, which corresponds to the `[EOS]` (the end of sequence) token, from the last layer of the text encoder, *i.e.*, $\mathbf{z}^t = \texttt{TextProj}(\mathbf{w}_K^N)$. For the **visual** encoder, the input image I is first split into M non-overlapping patches, and projected into a sequence of patch tokens $\mathbf{E_0} \in \mathbb{R}^{M\times d^v}$. Then, $\mathbf{E_0}$ is input into a K-layer Transformer-based architecture along with a learnable `[CLS]` token $\mathbf{c}_0$. For the i-th layer $\mathcal{L}_i^v$,

$$[\mathbf{c}_i, \mathbf{E}_i] = \mathcal{L}_i^v([\mathbf{c}_{i-1}, \mathbf{E}_{i-1}]) \qquad i = 1, 2, \cdots, K, \tag{2}$$

where $[\cdot, \cdot]$ indicates concatenation on the sequence length dimension. The final image embedding $\mathbf{z}^I \in \mathbb{R}^d$ is obtained by projecting the `[CLS]` token from the last layer of the visual encoder, *i.e.*, $\mathbf{z}^I = \texttt{VisProj}(\mathbf{c}_K)$. Therefore, the similarity score $s(t, I)$ between the image and the text can be calculated as the cosine similarity of $\mathbf{z}^t$ and $\mathbf{z}^I$.

To adapt CLIP for videos, a common solution is to learn a video embedding $\bar{\mathbf{z}}^v$ based on the frame embeddings $\mathbf{Z}^v = \{\mathbf{z}_1^v, \mathbf{z}_2^v, \cdots, \mathbf{z}_F^v\} \in \mathbb{R}^{F\times d}$ of all the sampled frames of v. And the similarity score between $\mathbf{z}^t$ and $\bar{\mathbf{z}}^v$ can be calculated and used for retrieval. In this paper, we focus on adapting CLIP in a parameter-efficient manner. Therefore, to avoid involving extra parameters, we here apply the non-parametric approach, *i.e.*, taking the average of all frame embeddings as the video embedding, as the starting point. Note that the direction of attaching other heavy architectures on the top of CLIP [10,27] is orthogonal to our exploration, as our modifications have all occurred inside the encoders. And we leave the investigation of their combination for future work.

3.2. Text-Video Co-operative Prompt Tuning (VoP)

By keeping the backbone fixed and only optimizing the introduced trainable continuous embeddings (*i.e.*, *prompts*) during fine-tuning, prompt tuning [1, 23, 24] effectively reduces per-task storage and memory usage when adapting large-scale foundation models to downstream tasks. In the less-studied video domain, a recent literature [18] proposes to insert continuous prompts into the input textual embedding sequence and shows promising results on several public video benchmarks. However, we argue this approach remains two challenges that limit the tuning performance. First, learning prompts only for the text branch overlooks the potential of collaboratively tuning the visual encoder. Second, prompting the mere input layer has only a relatively indirect impact on the output embeddings. To address the above challenges, we propose Text-Video Co-operative Prompt Tuning (VoP) that inserts prompts in each layer of both visual and text encoders (Fig. 2 ①), fully excavating the knowledge embedded in the CLIP model.

Specifically, in the **text** branch, we introduce a set of learnable tokens (*i.e.*, textual prompts) into each layer of the text encoder. The textual prompts for the i-th layer is denoted as $\mathbf{P}_{i-1}^t \in \mathbb{R}^{P^t\times d^t}$, where P^t is the number of the textual prompt tokens. Therefore, Eq. (1) can be transformed as

$$[\_\_, \mathbf{W}_i] = \mathcal{L}_i^t([\mathbf{P}_{i-1}^t, \mathbf{W}_{i-1}]), \tag{3}$$

where "__" indicates the output tokens at the corresponding positions will be discarded. Similarly, in the **vision** branch,

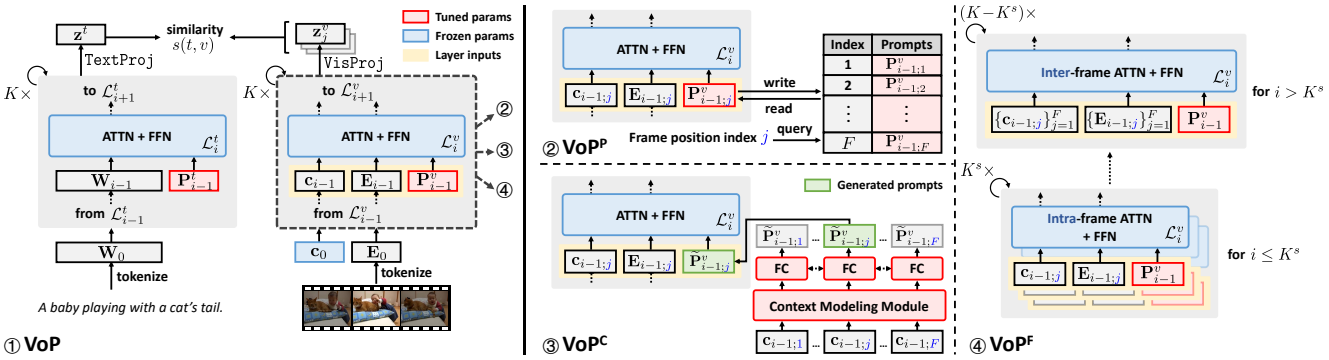

Figure 2. **Overview of our VoP framework before and after being equipped with video prompts**. To efficiently adapt CLIP to text-video retrieval, ① VoP tunes the prompts introduced in all layers of both uni-modal encoders while keeping the rest of the model frozen. In addition, ② position-specific, ③ context-specific, and ④ function-specific video prompts can replace conventional visual prompts to model essential information of the frame position, frame context, and layer function, respectively.

visual prompts are appended to each layer of the visual encoder. The visual prompts for the i-th layer is denoted as $\mathbf{P}^v_{i-1} \in \mathbb{R}^{P^v \times d^v}$, where P^v is the number of the visual prompt tokens. And Eq. (2) can be transformed as

$$[\mathbf{c}_i, \_, \mathbf{E}_i] = \mathcal{L}^v_i([\mathbf{c}_{i-1}, \mathbf{P}^v_{i-1}, \mathbf{E}_{i-1}]). \tag{4}$$

By jointly minimizing the symmetric text-to-video and video-to-text cross-entropy losses [27], both textual and visual prompts are fine-tuned while the other parameters from the two encoders are frozen. We find that the co-operation of entirely prompting both encoders adequately adapts the latent space of the model to the target domain.

3.3. Equipping with Video Prompts

Being a mature and efficient solution for text-video retrieval, however, the current VoP faces the dilemma that it treats frames as independent images, making it difficult to utilize rich information other than single-frame content. Therefore, we further develop a series of video prompts (Fig. 2 ② ③ ④) specifically for processing videos, which excavates information from different perspectives. We describe how these video prompts are combined with VoP as follows.

VoP with Position-Specific Video Prompts (VoP$^{\text{P}}$). One shortcoming of the current VoP is the learned prompts are shared for all frames, ignoring the order of the frame sequence. To inject information about the relative position of the current input frame, we present position-specific video prompts (Fig. 2 ② VoP$^{\text{P}}$), where visual prompts are only allowed to be shared between all frames at the same relative position in their belonged videos. This can be implemented by maintaining a table, where keys are the position indices and values are the prompts. Thus, the prompts for the current frame can be read by querying the table with the frame position index, and be written back after their optimization. Formally, after introducing the frame position, the flow of each visual encoder layer in Eq. (4) now becomes

$$[\mathbf{c}_{i;j}, \_, \mathbf{E}_{i;j}] = \mathcal{L}^v_i([\mathbf{c}_{i-1;j}, \mathbf{P}^v_{i-1;j}, \mathbf{E}_{i-1;j}]), \tag{5}$$

where j is the position index of the current frame in the video, and $\mathbf{P}^v_{i-1;j} \in \mathbb{R}^{P^v \times d^v}$ is the visual prompts of the i-th layer shared for all the j-th frames in all videos. Allowing to have more tunable position-specific parameters, VoP$^{\text{P}}$ increases the capacity for informative videos. In practice, as a copy of prompts for each layer contains several prompt tokens, we found that changing not all, but only a part of position-agnostic tokens into position-specific ones is sufficient for both effectiveness and efficiency.

VoP with Context-Specific Video Prompts (VoP$^{\text{C}}$). Another piece of information that cannot be exploited by the current scheme is the contextual relationships in videos. When addressing the current frame, a natural intuition is to integrate the contextual information from the rest of the video to emphasize important elements. To convert such information into prompts, we propose the dynamically generated context-specific video prompts (Fig. 2 ③ VoP$^{\text{C}}$) that are input-conditional rather than fixed once learned. Specifically, in each layer, we form the `[CLS]` tokens of frames from the same video into a sequence $\mathbf{C}_{i-1} \in \mathbb{R}^{F \times d^v}$, *i.e.*, $\mathbf{C}_{i-1} = \{\mathbf{c}_{i-1;1}, \mathbf{c}_{i-1;2}, \cdots, \mathbf{c}_{i-1;j} \cdots, \mathbf{c}_{i-1;F}\}$, and feed $\mathbf{C}_{i-1}$ into a Context Modeling Module (CMM) to modulate each frame token with its contextual information:

$$\widetilde{\mathbf{C}}_{i-1} = \texttt{CMM}(\mathbf{C}_{i-1}). \tag{6}$$

Then, a fully-connected (FC) layer generates the prompt tokens conditioned on the modulated frame token, where the input vector is stretched by the projection before splitting into multiple tokens:

$$\widetilde{\mathbf{P}}^v_{i-1;j} = \texttt{FC}(\tilde{\mathbf{c}}_{i-1;j}). \tag{7}$$

Therefore, Eq. (4) now becomes

$$[\mathbf{c}_{i;j}, \_\_, \mathbf{E}_{i;j}] = \mathcal{L}_i^v([\mathbf{c}_{i-1;j}, \widetilde{\mathbf{P}}_{i-1;j}^v, \mathbf{E}_{i-1;j}]). \quad (8)$$

In this way, global contextual information can be encoded into the generated prompts and participate in the intra-frame modeling. Note that $\widetilde{\mathbf{C}}_{i-1}$ does not pass through the next layer since it is used for prompt generation. To avoid introducing excessive parameters to be trained, parameters of CMM and the FC layer are shared between all encoder layers. As there exist several options for the architecture of CMM, experiment results show that the Bi-directional Long Short-Term Memory (BiLSTM) [11] is an overall preferable choice.

VoP with Function-Specific Video Prompts (VoP$^{\text{F}}$). Although we have designed two special prompts in consideration of the inherent properties of video, they are hardly a complete substitute for spatio-temporal modeling, which leads to the outstanding performance of video Transformers [3]. However, attaching even a "lightweight" Transformer on top of the CLIP visual encoder will increase the number of training parameters by a significant amount. To obtain a "free" spatio-temporal modeling, we propose a transformation of the functionality of existing frozen parameters in the deeper layers (Fig. 2 ④ VoP$^{\text{F}}$). Specifically, we split the current visual encoder into two parts according to the depth of the layer, and each part undertakes different functions. The first part that contains K^s shallow layers still performs spatial self-attention for tokens of each frame, and video prompts discussed above can still be adopted in this part without changes. However, the second part that contains the last $(K - K^s)$ layers now performs inter-frame spatio-temporal self-attention without changing structure. And visual prompts in these layers are prepared for the input sequence that consists of `[CLS]` and patch tokens from all frames of the same video. In other words, following the change of functions at different layers, the visual prompts are adaptively divided into frame-level and video-level ones. Formally, for the i-th layer $\mathcal{L}_i^v$, Eq. (4) remains unchanged when $i \leq K^s$. And for $i > K^s$, Eq. (4) becomes

$$\begin{aligned}&[\mathbf{C}_i, \_\_, \mathbf{E}_{i;1}, \mathbf{E}_{i;2}, \cdots, \mathbf{E}_{i;F}] \\ &= \mathcal{L}_i^v([\mathbf{C}_{i-1}, \mathbf{P}_{i-1}^v, \mathbf{E}_{i-1;1}, \mathbf{E}_{i-1;2}, \cdots, \mathbf{E}_{i-1;F}]). \quad (9)\end{aligned}$$

We note that before feeding into the (K^s+1)-th layer, a trainable frame positional embedding is added to all tokens from the video to retain positional information, which is omitted in the formula for simplification. And we use VoP$^{\text{F+P}}$ and VoP$^{\text{F+C}}$ to indicate the deployment of position-specific and context-specific video prompts at K^s shallow layers while applying VoP$^{\text{F}}$, respectively.

4. Experiments

4.1. Experimental Setup

Datasets. We conduct our experiments on the following benchmarks for text-video retrieval: (1) **MSR-VTT** [37] contains 10,000 videos, each paired with about 20 captions. Following previous works [10, 27, 40], we report results on both two data splits, 'training-9K' [9] and 'training-7K' [28], to compare with baselines. The test data in both splits is 'test 1k-A', which is comprised of 1,000 video-text pairs following JSFusion [38]. We use 'MSR-VTT-9k' and 'MSR-VTT-7k' to refer to the two data splits, respectively. (2) **DiDeMo** [14] contains 10,000 Flickr videos with 40,000 sentences. Following the setting from [2, 19, 25], we concatenate all the sentences of a video to form a paragraph and evaluate the model with paragraph-video retrieval. (3) **ActivityNet** [13] contains 20,000 YouTube videos. Following [9, 27], all descriptions of a video are also concatenated into a single query, and the 'val1' split is used to evaluate the model. (4) **LSMDC** [33] contains 118,081 video clips extracted from 202 movies. There are 109,673 videos in the training set and 7,408 videos in the validation set. And 1,000 videos in the test set are from movies disjoint with the training and validation set.

Evaluation Metrics. We follow the standard retrieval metrics [27] to use R@K (recall at rank K, higher is better), MnR (mean rank, lower is better), and MdR (median rank, lower is better) for evaluation. Specifically, R@1, R@5, and R@10 are reported.

Baselines. We compare our methods with other commonly used fine-tuning protocols: (1) **Full**: fully update all parameters of the pre-trained backbone. (2) **Bias** [6, 39]: fine-tune only the bias terms of the pre-trained backbone. (3) **Proj** [17]: fine-tune only the last linear projection of both encoders. (4) **Partial** [17]: fine-tune only the last layer of both encoders. (5) **Adapter$^{\text{ATTN}}$** [12, 15]: fine-tune only the FC layers inserted in parallel to each multi-head self-attention layer in both encoders. (6) **Adapter$^{\text{FFN}}$** [7, 15]: fine-tune only the FC layers inserted in parallel to each feed-forward network in both encoders.

Implementation Details. 12-layer visual and text encoders are adopted from a pre-trained CLIP (ViT-B/32+Transformer), and all original parameters of the backbone are kept frozen unless otherwise stated. We optimize each model for 5 epochs using the AdamW [26] optimizer with weight decay set to 0.2, and decay the learning rate using a cosine schedule [4]. And the initial learning rate for each method is determined by searching in the range of $[1e^{-6}, 1e^{-2}]$. For all experiments, we uniformly sample 12 frames from each video following previous studies [2, 10, 27]. All video frames are resized to 224×224, and the maximum number of textual tokens is

Methods	Params (M)	$t2v$ R@1	R@5	R@10	MnR↓	MdR↓	$v2t$ R@1	R@5	R@10	MnR↓	MdR↓
Full	119.8 (100%)	41.7	69.2	79.0	16.5	2.0	42.5	70.9	**81.4**	**11.0**	2.0
Bias [6]	0.1 (0.104%)	39.7	66.5	77.3	17.3	2.0	41.1	68.4	79.2	13.6	2.0
Proj [17]	0.7 (0.547%)	37.1	63.0	76.1	20.5	3.0	37.2	64.6	75.9	16.7	3.0
Partial [17]	7.7 (6.410%)	39.8	65.3	75.9	19.3	2.0	37.9	66.1	77.4	15.5	3.0
Adapter$^{\text{ATTN}}$ [12]	2.0 (1.655%)	37.6	63.2	75.8	18.7	3.0	39.6	66.5	76.8	14.7	2.0
Adapter$^{\text{FFN}}$ [7]	2.0 (1.655%)	38.2	63.5	76.4	17.9	3.0	39.9	66.8	77.7	14.2	2.0
VoP	0.1 (0.103%)	39.6	66.7	77.8	17.2	2.0	42.1	68.8	80.7	12.4	2.0
VoP$^{\text{P}}$	0.5 (0.441%)	40.1	65.7	77.7	16.9	2.0	42.5	70.0	79.9	12.4	2.0
VoP$^{\text{C}}$	14.3 (11.898%)	40.8	68.1	79.0	15.8	2.0	42.3	70.1	81.1	11.4	2.0
VoP$^{\text{F}}$	0.1 (0.103%)	42.6	68.4	78.7	15.8	2.0	42.4	70.5	81.0	**11.0**	2.0
VoP$^{\text{F+P}}$	0.4 (0.328%)	43.5	69.3	79.3	**14.8**	2.0	43.6	**71.2**	81.2	**11.0**	2.0
VoP$^{\text{F+C}}$	14.1 (11.785%)	**44.6**	**69.9**	**80.3**	16.3	2.0	**44.5**	70.7	80.6	11.5	2.0

Table 1. **Retrieval results on the MSR-VTT-9k dataset**.

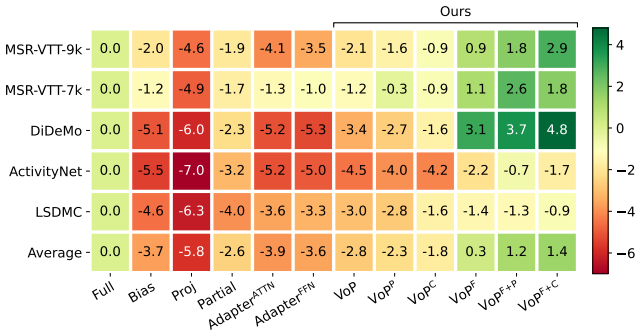

Figure 3. $t2v$ **R@1 gains of all methods in comparison against Full**. The first five rows show the improvement or deterioration on each benchmark, and the last row summarizes the average relative results over all benchmarks.

77, following the original CLIP design. And the batch size is set to 32. For Adapter$^{\text{ATTN}}$ and Adapter$^{\text{FFN}}$, the number of hidden dimensions is set to 64. For methods with VoP, the prompt length for both encoders, *i.e.*, P^t and P^v, are set to 8 as default. And a normal initialization is applied to prompt parameters. For VoP$^{\text{P}}$, VoP$^{\text{C}}$, VoP$^{\text{F+P}}$ and VoP$^{\text{F+C}}$, the length of video prompts is set to 4. And context-specific video prompts applies a 1-layer BiLSTM [11] as CMM to pass contextual information. For VoP$^{\text{F}}$, VoP$^{\text{F+P}}$ and VoP$^{\text{F+C}}$, we set K^s as 8. All experiments are carried out on 4 NVIDIA Tesla V100 GPUs.

4.2. Main Results

We compare our methods with popular tuning protocols on five benchmarks. We here represent $t2v$ R@1 gains relative to full fine-tuning in Fig. 3, and report $t2v$ and $v2t$ results on MSR-VTT-9k in Tab. 1. Detailed results on the other four benchmarks can be found in the supplementary materials. We here highlight some important observations from Fig. 3 as follows:

- **VoP achieves competitive or superior performance than other efficient tuning protocols with only 0.1% parameter storage**. The only exception is that R@1 of Partial is on average 0.2% higher than that of our VoP at the expense of a more than 60× parameter overhead.
- **Appending only one video prompts can significantly improve VoP to even compete with Full**. Position-specific, context-specific and function-specific video prompts respectively bring 0.5%, 1.0% and 3.1% average R@1 gain. While VoP$^{\text{P}}$ and VoP$^{\text{C}}$ obtain a superior performance than all other efficient tuning protocols on average, VoP$^{\text{F}}$ even outperforms Full by 0.3% without introducing more parameters to VoP.
- **Combining the video prompts can lead to higher performance benefits**. Compared to VoP$^{\text{F}}$, VoP$^{\text{F+P}}$ and VoP$^{\text{F+C}}$ further achieve 0.9% and 1.1% R@1 improvements. As a conclusion, we provide sufficient solutions to choose from according to the strictness of the parameter and computation limitations.

Textual	Visual	R@1	R@5	R@10	MnR↓	MdR↓
		31.5	52.8	63.6	42.9	5.0
	✓	36.5	62.7	75.1	18.3	3.0
✓		36.3	63.4	75.0	20.3	3.0
✓	✓	**39.6**	**66.7**	**77.8**	**17.2**	**2.0**

Table 2. **Ablation on co-operative multi-modal prompts in VoP.** The co-operation of prompting multi-modal branches leads to higher performance.

4.3. Ablation Study

We ablate different model design choices on MSR-VTT-9k and report $t2v$ results if no otherwise specified. We note that as many hyper-parameters exist in multiple solutions, we determine the values to be taken based on their performance in the base solution to speed up the search.

Effect of Co-operative Uni-Modal Prompts in VoP. In Tab. 2, we report the performance of using CLIP without fine-tuning, tuning with uni-modal prompts, and using our proposed VoP. While uni-modal prompts outperform directly applying CLIP without tuning, our VoP achieves better results, which demonstrates that prompting both encoders enables better adaptation to the downstream text-video retrieval task.

Choice of CMM	MSR-VTT-9k			MSR-VTT-7k			DiDeMo			ActivityNet			LSMDC		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Transformer	40.1	68.2	78.8	39.5	68.2	78.1	**40.4**	67.3	77.3	32.0	61.5	74.9	20.3	39.5	47.8
LSTM	40.6	**69.5**	**79.7**	39.5	**69.3**	78.0	38.6	66.7	77.0	32.4	62.0	75.4	19.6	38.2	47.7
BiLSTM	**40.8**	68.1	79.0	**40.0**	67.3	**78.2**	40.0	**68.0**	**78.5**	**32.6**	**62.5**	**76.5**	**20.4**	**40.0**	**48.1**

Table 3. **Ablation on the CMM choices in VoP^{C}**. We report the R@1, R@5 and R@10 results on five benchmarks. In general, BiLSTM achieves the best results on more datasets.

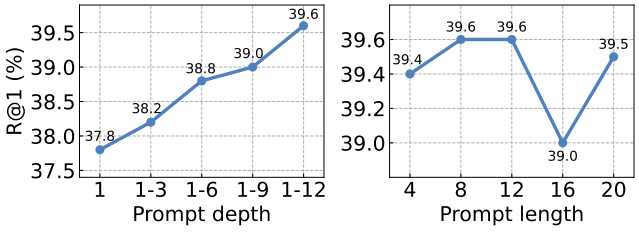

Figure 4. **Ablation on hyperparameters of prompts in VoP**. **Left**: Ablation on prompt depth, *i.e.*, where and how many layers to insert prompts. *i-j* indicates the encoder layer indices that prompts are inserted into, while the 1-st layer refers to the one closest to the input. **Right**: Ablation on prompt length, *i.e.*, the number of prompt tokens in each layer.

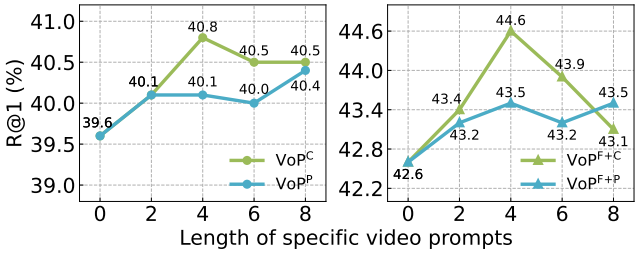

Figure 5. **Ablation on length of video prompts**. The number of visual prompt tokens that are replaced with corresponding video-specific prompts is varied, and the maximum length is set to 8 based on previous experimental results.

Effect of Prompt Depth in VoP. In Fig. 4 (left), we gradually increase the number of layers inserted into prompts. In general, the performance of VoP is positively correlated with the prompt depth. And inserting prompts into every layer of both encoders contributes to the best results.

Effect of Prompt Length in VoP. In Fig. 4 (right), we vary the number of prompt tokens in each layer. Unlike the prompt depth, steadily increasing the prompt length does not lead to continuous growth of performance. And using only 8 tokens remains a competitive performance with parameter efficiency.

Effect of Prompt Length for Video Prompts. Fig. 5 ablates how many conventional visual prompt tokens to change to video-specific ones. We observe that only inserting video-specific prompts does not lead to the best results for all cases. And turning only half of the 8 prompt tokens into specific ones is a more universal choice that achieves a trade-off between effectiveness and efficiency. We explain that the conventional visual prompts learn general knowledge shared between all frames across videos, which complements our video prompts that may focus more on other information around a specific frame.

Role of CMM	R@1	R@5	R@10	MnR↓	MdR↓
Updating [CLS]	38.0	63.6	75.3	18.5	**2.0**
Generating prompts	**40.8**	**68.1**	**79.0**	**15.8**	**2.0**

Table 4. **Ablation on the role of CMM in VoP^{C}**. The current VoP^{C} design represented in the second row achieves better results.

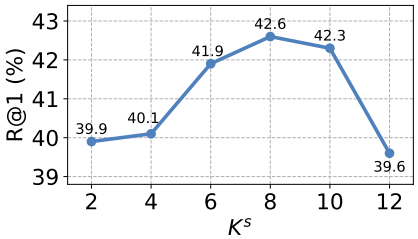

Figure 6. **Ablation on layers with different functions in VoP^{F}**. The first K^s shallow layers perform intra-frame spatial self-attention, and the subsequent deep layers perform inter-frame spatio-temporal self-attention.

Effect of CMM in VoP^{C}. We first ablate different choices on which CMM to model the contextual information. We report the R@1 of three candidates, *i.e.*, Transformer, LSTM and BiLSTM, on the five benchmarks. Note that we use a 4-layer Transformer, which is a common choice when modeling temporal dependencies in previous works [27]. And the number of layers is set to 1 for both LSTM, and BiLSTM. As shown in Tab. 3, BiLSTM is a more effective choice in general.

In VoP^{C}, unlike the common practice in temporal modeling, CMM outputs frame tokens modulated with contextual information for generating prompts instead of directly updating [CLS] tokens. Thus a question naturally arises: Do we indeed need to generate prompts? In other words, will directly updating [CLS] tokens with CMM be a more effectual option? To answer the question, we compare the two solutions in Tab. 4, and our VoP^{C} achieves leadership in nearly all metrics. Our explanation for the results is that generating prompts for the token sequence achieves cross-frame communication with less disruption to the information flow of the original visual encoder.

Figure 7. **Qualitative results of four tuning methods: Full, Partial, VoP and VoP^{F+C}.** Given the query text, we represent the rank-1 retrieval result of each method, which can be incorrect (each first row) or ground truth (each second row).

Effect of Split Layer in VoP^{F}. We vary the value of K_s in VoP^{F} and illustrate the results in Fig. 6. A larger value of K_s means that more shallow layers are used for intra-frame spatial self-attention and fewer deep layers for inter-frame spatio-temporal self-attention. We note that when $K_s = 12$, VoP^{F} degrades to VoP. And the result of $K_s = 0$, *i.e.*, all layers performing inter-frame spatio-temporal self-attention, is not represented as we found it failed to generalize and the performance collapsed. As shown in the figure, R@1 raises with increasing K_s until $K_s = 8$, which shows the necessity of intra-frame message exchanging in shallow layers. Subsequently, R@1 begins to drop as K_s continues to increase, indicating that properly substituting the functions of layers and corresponding prompts brings improvements. And, in conclusion, $K_s = 8$ is the choice that achieves the best trade-off.

4.4. Qualitative Results

In Fig. 7, we visualize some $t2v$ retrieval examples from the test set of MSR-VTT-9k. We represent the retrieval results of four tuning methods: Full, Partial, VoP, and VoP^{F+C}. In the **top left** example, Full and our proposed methods can retrieve the correct video while Partial matches an unrelated one, which shows the inferiority of existing efficient tuning protocols. In the **top right** example, Full fails to recognize a "Japanese" book while parameter-efficient tuning methods succeed by capturing visual clues of Japanese characters and related English words like "Tokyo", indicating that updating all parameters might be an unsatisfactory strategy as more knowledge from large-scale text-image pre-training is forgotten. In the **bottom left** example, by fine-tuning all parameters with video datasets or designing specialized prompting solutions for videos, Full and VoP^{F+C} can understand the whole event represented by sequenced frames. Even if some textual elements like "priest" are not visually present, the methods overcome such minor semantic misalignments and select more relevant candidates from a global view. Finally in the **bottom right** example, understanding the concept of "tail" and capturing the interaction of "playing with", VoP^{F+C} can distinguish the correct video from hard negative candidates while all the other three methods fail.

5. Conclusion

In this paper, we continue the vein of prompt tuning to transfer pre-trained CLIP for text-video retrieval with both effectiveness and efficiency. We first devise a simple but competitive baseline VoP, which achieves promising performance with only 0.1% trainable parameters by prompting all layers of both textual and visual encoders. To increase the revenue of VoP, we further explore three video prompts to model different video-specific information. Different combinations of our video prompts can be selected depending on the strictness of the limits on parameter overhead, and achieve at most 1.4% average relative improvement with much fewer trainable parameters compared to full fine-tuning. We hope our work can inspire future research on how to fully exploit the large foundation models in challenging video-understanding tasks.

Acknowledgement

This work was supported by STI 2030—Major Projects (2022ZD0208800), NSFC General Program (Grant No. 62176215). This work was supported by Alibaba Group through Alibaba Research Intern Program.

References

[1] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv preprint arXiv:2203.17274*, 2022. 1, 2, 3

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1708–1718, 2021. 2, 5

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning*, pages 813–824, 2021. 5

[4] Johan Bjorck, Kilian Q. Weinberger, and Carla P. Gomes. Understanding decoupled and early weight decay. *arXiv preprint arXiv:2012.13841*, 2020. 5

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 1877–1901, 2020. 2

[6] Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. TinyTL: Reduce memory, not parameters for efficient on-device learning. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 11285–11297, 2020. 5, 6

[7] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. AdaptFormer: Adapting vision transformers for scalable visual recognition. In *Proceedings of the Advances in Neural Information Processing Systems*, 2022. 5, 6

[8] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14064–14073, 2022. 2

[9] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *Proceedings of the European Conference on Computer Vision*, pages 214–229, 2020. 1, 2, 5

[10] Satya Krishna Gorti, Noel Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guang Wei Yu. X-Pool: Cross-modal language-video attention for text-video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4996–5005, 2022. 1, 2, 3, 5

[11] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, pages 602–610, 2005. 5, 6

[12] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *Proceedings of the International Conference on Learning Representations*, 2022. 5, 6

[13] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 2, 5

[14] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan C. Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5804–5813, 2017. 2, 5

[15] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the International Conference on Machine Learning*, pages 2790–2799, 2019. 5

[16] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the International Conference on Machine Learning*, pages 4904–4916, 2021. 1, 2

[17] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Proceedings of the European Conference on Computer Vision*, pages 709–727, 2022. 2, 5, 6

[18] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *Proceedings of the European Conference on Computer Vision*, pages 105–124, 2022. 1, 2, 3

[19] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7331–7341, 2021. 2, 5

[20] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021. 1, 2

[21] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 9694–9705, 2021. 1, 2

[22] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan.

Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *Proceedings of the International Conference on Learning Representations*, 2022. 1

[23] Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-Tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021. 1, 3

[24] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. GPT understands, too. *arXiv preprint arXiv:2103.10385*, 2021. 1, 2, 3

[25] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *Proceedings of the British Machine Vision Conference*, page 279, 2019. 1, 2, 5

[26] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *Proceedings of the International Conference on Learning Representations*, 2017. 5

[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*, pages 293–304, 2022. 1, 2, 3, 4, 5, 7

[28] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9876–9886, 2020. 5

[29] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019. 2

[30] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-driven manipulation of StyleGAN imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2065–2074, 2021. 2

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763, 2021. 1, 2, 3

[32] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 3

[33] Anna Rohrbach, Marcus Rohrbach, and Bernt Schiele. The long-short story of movie description. In *Proceedings of the German Conference on Pattern Recognition*, pages 209–221, 2015. 2, 5

[34] Alex Jinpeng Wang, Yixiao Ge, Guanyu Cai, Rui Yan, Xudong Lin, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Object-aware video-language pre-training for retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3303–3312, 2022. 1

[35] Mengmeng Wang, Jiazheng Xing, and Yong Liu. ActionCLIP: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 2

[36] Peng Wu, Xiangteng He, Mingqian Tang, Yiliang Lv, and Jing Liu. HANet: Hierarchical alignment networks for video-text retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 3518–3527, 2021. 1, 2

[37] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5288–5296, 2016. 2, 5

[38] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European Conference on Computer Vision*, pages 487–503, 2018. 2, 5

[39] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1–9, 2022. 5

[40] Shuai Zhao, Linchao Zhu, Xiaohan Wang, and Yi Yang. CenterCLIP: Token clustering for efficient text-video retrieval. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 970–981, 2022. 1, 3, 5

[41] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, pages 2337–2348, 2022. 2