RAP: Efficient Text-Video Retrieval with Sparse-and-Correlated Adapter

Anonymous ACL submission

Abstract

Text-Video Retrieval (TVR) aims to align relevant video content with natural language queries. To date, most of the state-of-the-art TVR methods learn image-to-video transfer learning based on large-scale pre-trained vision-language models (*e.g.*, CLIP). However, fully fine-tuning these pre-trained models for TVR incurs prohibitively expensive computation costs. To this end, we propose to conduct efficient text-video **R**etrieval with a sparse-and-correlated **A**da**P**ter (**RAP**), *i.e.*, fine-tuning the pre-trained model with a few parameterized layers. To accommodate the text-video scenario, we equip our RAP with two indispensable characteristics including temporal *sparsity* and *correlation*. Specifically, we propose a low-rank modulation module to refine the per-image features from the frozen CLIP backbone, which accentuates salient frames within the video features while alleviating temporal redundancy. Besides, we introduce an asynchronous self-attention mechanism that first selects the top responsive visual patches and augments the correlation modeling between them with learnable temporal and patch offsets. Extensive experiments on four TVR datasets demonstrate that our RAP achieves superior or comparable performance compared to the fully fine-tuned counterpart and other parameter-efficient fine-tuning methods.

1 Introduction

Text-Video Retrieval (TVR) (Gabeur et al., 2020; Gorti et al., 2022; He et al., 2021a; Lei et al., 2021; Luo et al., 2022; Ma et al., 2022; Wang et al., 2022) is a pivotal task in the realm of multimodal research, which aims to find the most relevant video content within a repository in response to the text query, and vice versa. With the rapid progress in large-scale image-text pre-training (Jia et al., 2021; Radford et al., 2021; Yu et al., 2022; Yuan et al., 2021), current research focuses on how to transfer pre-trained image-text models (*e.g.*, CLIP (Radford et al., 2021)) to the video-text domain. However, fully fine-tuning the video model is computationally expensive and may have the risk of overfitting.

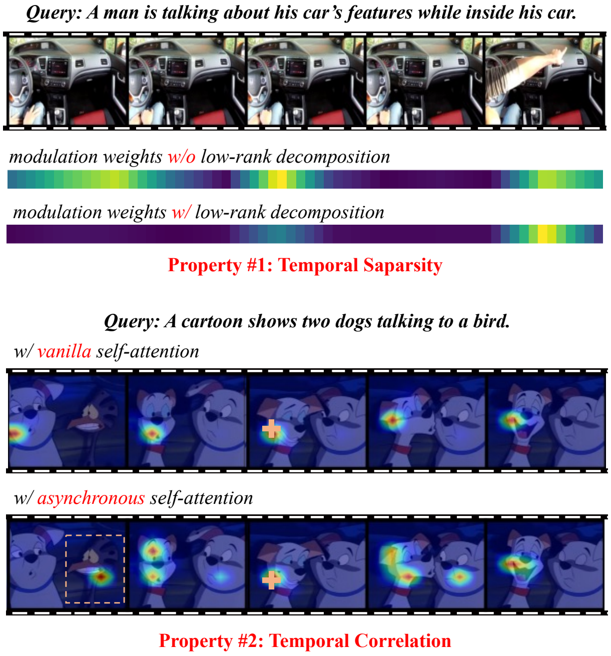
Query: A man is talking about his car's features while inside his car.
modulation weights w/o low-rank decomposition
modulation weights w/ low-rank decomposition
Property #1: Temporal Saparsity
Query: A cartoon shows two dogs talking to a bird.
w/ vanilla self-attention
w/ asynchronous self-attention
Property #2: Temporal Correlation

Figure 1: **Top: Illustrations of temporal sparsity.** We visualize the modulation weight *w/* or *w/o* low-rank decomposition. **Down: Illustrations of temporal correlation.** The query patch is marked by the yellow cross and the similarity map within other frames are plotted.

To alleviate this dilemma, Parameter-Efficient Fine-Tuning (PEFT) stemmed from natural language processing (Houlsby et al., 2019; Lester et al., 2021; Zaken et al., 2022; Hu et al., 2021) has also aroused extensive research interest in the field of computer vision (Chen et al., 2022b,a) and cross-modal learning (Sung et al., 2022). Recently, some exploratory work (Zhang et al., 2023; Jiang et al., 2022) has also attempted to introduce PEFT into TVR. These methods, however, simply introduce existing PEFT algorithms (Houlsby et al., 2019; You et al., 2022; Karimi Mahabadi et al., 2021)

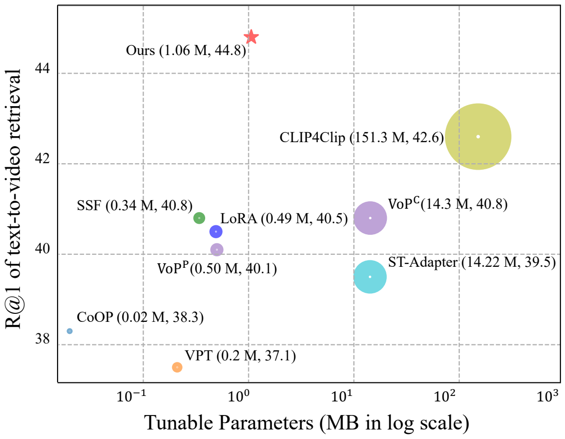

Figure 2: **Text-to-video retrieval performance on MSR-VTT dataset.** Marker sizes are proportional to the number of tunable parameters.

without considering the inherent characteristics of video data.

To this end, we argue that an ideal PEFT method for VTR should be equipped with two characteristics: 1) **Temporal Sparsity**: As shown in Figure 1, the video data inherently contains lots of redundancies or repetitions in the temporal perspective. The visualized frame-by-frame embedded CLIP features are *over-smooth*, resulting in the loss of important details or nuances within the video data. In contrast, the video feature adapted from pre-trained CLIP should capture the most informative frames, allowing for a more sparse representation. 2) **Temporal Correlation**: The desired video adapter is supposed to incorporate the dependencies and relationships between consecutive frames, especially when dealing with actions or events that unfold over several frames, as the features can encapsulate the evolving context over time. For example in Figure 1, the query sentence includes two entities including `dog` and `bird`. Given the query patch (+ in frame #3), we visualize the similarity distribution within the other patches. The vanilla self-attention can only attend to the `dog` instance while the other `bird` instance is overlooked.

In the realm of video processing and analysis, the temporal dimension often contains redundancies due to the inherent correlation between adjacent frames. This redundancy can lead to inefficiencies in computational resources and storage when dealing with large-scale video data. Therefore, there is a need to extract meaningful and informative features while reducing temporal redundancy.

To alleviate these aforementioned issues, we propose an efficient text-video **R**etrieval framework with sparse-and-correlated **A**da**P**ter (dubbed as **RAP**). Our proposed RAP not only streamlines the trainable parameters, enhancing efficiency in computational resources but also tailors the architecture to adeptly capture and model the nuanced temporal characteristics of video data.

To achieve temporal sparsity, we propose a Low-Rank Modulation (LoRM) module to refine the pre-trained CLIP feature (Radford et al., 2021) on the principle of redundancy reduction and essential information extraction. This design stems from a simple hypothesis that the change in temporal weights resides on a low intrinsic rank (Zhang and Tao, 2012). Therefore, we introduce layer-wise low-rank scale parameters and shift parameters, which could be considered as variance and mean to modulate the CLIP feature. Specifically, both scale and shift parameters are instantiated by the multiplication of two low-rank trainable matrices. These parameters are input-independent and therefore more flexible. LoRM allows us to calibrate the video features to highlight salient frames and mitigate temporal redundancy.

For temporal correlation modeling, we replace vanilla self-attention with the proposed Asynchronous Self-Attention (ASA), which introduces temporal dynamics among video frames to capture temporal relationships. Since the attention computing in pre-trained CLIP is constrained within each frame feature, it is challenging to apply to the video domain due to the temporally dynamic nature of video frames. Previous methods employ either temporal Transformer (Jiang et al., 2022; Yang et al., 2022; Zhang et al., 2023) or 3D convolution networks (Yao et al., 2023; Liu et al., 2023) to encode temporal dependencies. Instead of introducing additional modules, we propose an asynchronous self-attention that only warps partial patch tokens in a parameterized way. Firstly, for each frame, we filter semantically significant patches via a parameter-free text-conditioned selection mechanism. Specifically, we compute the similarities between patch features and the corresponding sentence and select the patches with the highest responses. Secondly, each selected patch within the current frame is dynamically warped to attend to the temporally related patches in other frames. The proposed asynchronous self-attention empowers the flexibility in capturing correlations between video frames at the fine-grained patch level.

Overall, the main contributions of this work are:

- We propose RAP to adapt the pre-trained CLIP to efficient TVR, which not only reduces the tunable parameters but also generates temporally sparse and correlated video features.

- To alleviate the temporal redundancy, a low-rank modulation module is introduced to calibrate the frame-wise representation linearly.
- We propose an asynchronous self-attention that captures long-range dependencies with negligible computational overheads.
- Extensive experiments show that our RAP is on par with or even superior to previous PEFT methods and the fully fine-tuned counterpart.

2 Related Work

Text-Video Retrieval. TVR (Yu et al., 2018; Croitoru et al., 2021; Yang et al., 2021; Wang et al., 2021; Chen et al., 2020; Wang and Shi, 2023; Jin et al., 2022, 2023a,b; Liu et al., 2022) is a fundamental research topic in the video-language domain which aims to retrieval the relevant video/text based on the given text/video query. The pioneer works (Yu et al., 2018; Gabeur et al., 2020) rely on pre-extracted features from frozen video and text encoders. To facilitate the end-to-end training, ClipBERT (Lei et al., 2021) proposes a sparse sampling strategy for efficient text-video training. With the great success of large-scale image-text pretraining model CLIP (Radford et al., 2021), the majority of the state-of-the-art TVR methods (Luo et al., 2022; Ma et al., 2022; Wang et al., 2023; Hannan et al., 2023; Jin et al., 2022) focus on transferring the powerful CLIP encoder to the video-text domain by designing various cross-modal alignment strategies. As the first attempt, CLIP4Clip (Luo et al., 2022) employs mean-pooling or Transformer to aggregate video features and conduct coarse-grained (video-sentence level) contrastive alignment. Instead of using the text-agnostic aggregation manner, X-CLIP (Ma et al., 2022) proposes to aggregate video representations conditioned on the text's attention weight and conduct the multi-grained contrastive learning at the frame-word, video-sentence, video-word and sentence-frame levels. For more comprehensive alignment, UCOFIA (Wang et al., 2023) unifies the coarse-grained and fine-grained alignment to capture both the high-level and low-level correspondence between text and video.

Most of the current TVR methods follow the fully fine-tuning paradigm. This scheme, however, is computation-intensive and may have the risk of overfitting. Besides, additional temporal modeling models are required to bridge the image and video gap. In this paper, we propose RAP which conducts parameter-efficient fine-tuning for TVR which provides a more computationally efficient and potentially more robust approach. Besides, the tunable parameters in our RAP also bear the responsibility for temporal modeling, thus eliminating the need for external temporal modules.

Parameter-Efficient Transfer Learning. PEFT (Houlsby et al., 2019; Hu et al., 2021; Lester et al., 2021; He et al., 2021b; Zaken et al., 2022; Sung et al., 2021) is firstly introduced in the NLP domain to reduce the number of trainable parameters while maintaining the comparable performance with the fully fine-tuning setting. Inheriting the merit from NLP, PEFT in computer vision (Jia et al., 2022; Bahng et al., 2022; Jie and Deng, 2022; Sung et al., 2022) also gained extensive research attention. VPT (Jia et al., 2022) follows the prompt tuning strategy by introducing the task-specific learnable prompts on the vision Transformer. To be more compatible with vision tasks, Convpass (Jie and Deng, 2022) introduces the inductive bias of convolutional layers by reconstructing the spatial structure of the token sequence via convolution operations. VL-Adapter (Sung et al., 2022) pioneeringly benchmarks different types of PEFT techniques including Adapter (Houlsby et al., 2019), Hyperformer (Mahabadi et al., 2021), and Compacter (Karimi Mahabadi et al., 2021) in the multi-task setting.

There also exist several works (Yang et al., 2022; Pan et al., 2022; Lin et al., 2022; Li and Wang, 2023; Yao et al., 2023; Jiang et al., 2022; Zhang et al., 2023; Lu et al., 2023) focusing on the image-to-video transfer learning. Based on the pre-trained CLIP model, these methods either introduce temporal convolution (Pan et al., 2022) or Transformer (Lu et al., 2023) in sequential (Zhang et al., 2023; Jiang et al., 2022) or parallel (Yao et al., 2023) ways. However, they overlook the inherent temporal structure of video data while our RAP pinpoints two key issues in video feature modeling and generate more representative video features.

3 Method

Text-video retrieval aims to search for and retrieve relevant videos/texts based on textual/video queries by evaluating the similarity between the video-sentence pairs. Our proposed RAP is devoted to bridging the gap between the frozen CLIP feature

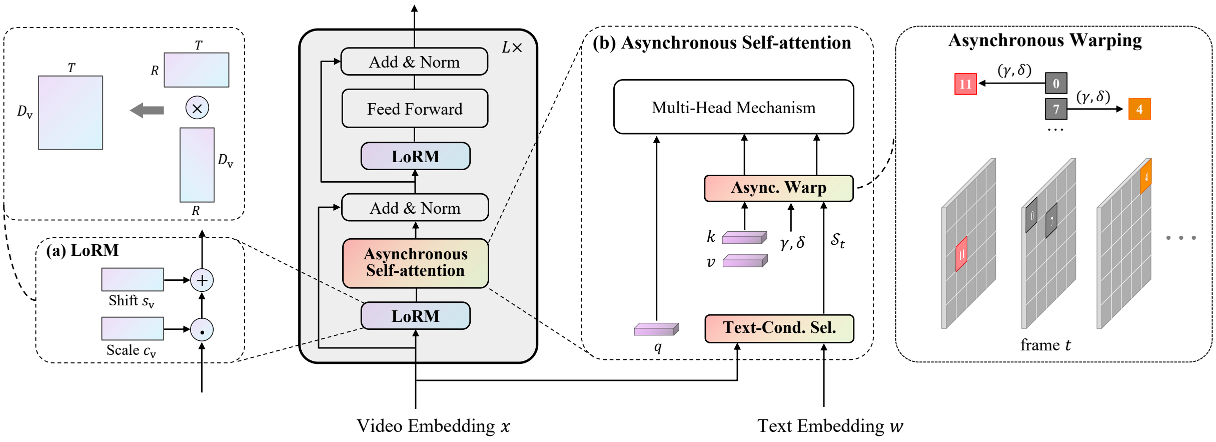

Figure 3: **An overview of RAP.** (a) LoRM sets up learnable shift parameters $\boldsymbol{c}_{\mathrm{v}}$ and scale parameters $\boldsymbol{s}_{\mathrm{v}}$ to calibrate the vanilla CLIP features. For the temporally sparse requirement, $\boldsymbol{c}_{\mathrm{v}}$ and $\boldsymbol{s}_{\mathrm{v}}$ are generated by low-rank decomposition on the temporal dimension. (b) Asynchronous self-attention first filters out patch set $\mathcal{S}_t$ via text-conditioned selection. Then, the filtered patches are warped based on the learnable patch offset γ and temporal offset δ.

and the dynamic video scenario by introducing negligible parameter overheads.

The schematic illustration of our RAP is illustrated in Figure 3. In Sec. 3.1, we present the preliminaries of RAP including the video and text feature embedding. Then we describe the proposed low-rank modulation and the asynchronous self-attention in Sec. 3.2 and Sec. 3.3, respectively.

3.1 Feature Embedding

Video Embedding. We utilize the visual backbone (ViT (Dosovitskiy et al., 2020)) of CLIP as the video encoder. Given the video data, we follow ViT (Dosovitskiy et al., 2020) to process each frame independently. Specifically, each frame with shape $H \times W$ is split into non-overlapping patches with shape $P \times P$ and then linearly projected into the embedding space. Such linear projection generates $N = HW/P^2$ patch features for each frame. Besides, a learnable [CLS] token is prepended to each frame patch feature sequence to represent the global frame representations. The positional embedding is also added to incorporate positional information explicitly. Through the above process, we obtain the t^{th} frame feature $\boldsymbol{x}_t^0 \in \mathbb{R}^{(N+1) \times D_{\mathrm{v}}}$, $t \in [1, T]$, where D_{v} is visual feature dimension.

The residual structure with serially connected multi-head self-attention (MHSA) and multilayer perceptron (MLP) is applied to capture sequential dependencies and contextual relationships within each frame patch sequence. Repeating the above steps for each frame, we obtain the video embedding at l^{th} layer $\boldsymbol{x}^l \in \mathbb{R}^{T \times (N+1) \times D_{\mathrm{v}}}$, $l \in [1, L]$. Specifically, we decompose $\boldsymbol{x}^l = [\boldsymbol{f}^l, \boldsymbol{p}^l]$, where $\boldsymbol{f}^l \in \mathbb{R}^{T \times D_{\mathrm{v}}}$ represent the frame-wise features (*i.e.*, [CLS] token feature) while $\boldsymbol{p}^l \in \mathbb{R}^{T \times N \times D_{\mathrm{v}}}$ is patch-wise representation at the l^{th} layer.

Text Embedding. For text embedding, we directly use the text encoder of CLIP to generate the textual representation. The text encoder is a Transformer (Vaswani et al., 2017) with the architecture modifications as described in (Radford et al., 2019). The [EOS] token is also appended to encode the global sentence feature. Concretely, we denote the sentence features at the l^{th} layer as $\boldsymbol{w}^l \in \mathbb{R}^{1 \times D_{\mathrm{t}}}$, where D_{t} is the text feature dimension.

3.2 Low-rank Modulation

In this section, we elaborate on the feature modulation for both video and text features. Since all the layers share the same modulation process, we omit the superscript of layer index l for brevity.

Low-rank Modulation for Video. The frame-by-frame encoded video features $\boldsymbol{x}$ cannot reflect the characteristics of the video data. The redundancy in the temporal dimension is a major feature that distinguishes videos from static images. To this end, we introduce low-rank scale parameters and shift parameters, which serve as the variance and mean values to modulate the pre-trained CLIP feature. These parameters are input-independent, rendering them comparatively lightweight in nature and hopefully more scalable. Specifically, the video scale parameter $\boldsymbol{c}_{\mathrm{v}} \in \mathbb{R}^{T \times D_{\mathrm{v}}}$ and video shift parameter $\boldsymbol{s}_{\mathrm{v}} \in \mathbb{R}^{T \times D_{\mathrm{v}}}$ are decomposed as follows:

$$\boldsymbol{c}_{\mathrm{v}} = \boldsymbol{c}^{\mathrm{a}} \cdot \boldsymbol{c}^{\mathrm{b}}, \quad \boldsymbol{s}_{\mathrm{v}} = \boldsymbol{s}^{\mathrm{a}} \cdot \boldsymbol{s}^{\mathrm{b}}, \tag{1}$$

where $\boldsymbol{c}^{\mathrm{a}}, \boldsymbol{s}^{\mathrm{a}} \in \mathbb{R}^{T \times R}$, $\boldsymbol{c}^{\mathrm{b}}, \boldsymbol{s}^{\mathrm{b}} \in \mathbb{R}^{R \times D_{\mathrm{v}}}$ are learnable parameters and we set rank $R \ll \min(T, D_{\text{v}})$ to implement the low-rank requirement. The low-rank modulation is applied as follows.

$$\boldsymbol{u} = \boldsymbol{c}_{\text{v}} \odot \boldsymbol{x} + \boldsymbol{s}_{\text{v}}, \tag{2}$$

where $\odot$ denotes the element-wise multiplication with broadcast. During training, the vanilla feature $\boldsymbol{x}$ is extracted through frozen CLIP backbone and the learnable $\boldsymbol{c}_{\text{v}}$ and $\boldsymbol{s}_{\text{v}}$ help modify $\boldsymbol{x}$ to be of temporally low-rank. $\boldsymbol{u} \in \mathbb{R}^{T\times(N+1)\times D_{\text{v}}}$ is the modulated video feature.

Modulation for Text. We also modulate the textual embedding $\boldsymbol{w}$ with parameters $\boldsymbol{c}_{\text{t}}$ and $\boldsymbol{s}_{\text{t}}$ as follows.

$$\boldsymbol{z} = \boldsymbol{c}_{\text{t}} \odot \boldsymbol{w} + \boldsymbol{s}_{\text{t}}, \tag{3}$$

where $\boldsymbol{c}_{\text{t}}, \boldsymbol{s}_{\text{t}} \in \mathbb{R}^{1\times D_{\text{v}}}$ are learnable parameters. We do **not** conduct modulation at the word level or use parameter low-rank decomposition since the textual data do not exhibit the sparsity characteristic.

3.3 Asynchronous Self-Attention

Let's review the vanilla self-attention in the video encoder. For clarity, we take the t^{th} frame of the input video for illustration. The corresponding modulated feature is denoted as $\boldsymbol{u}_t \in \mathbb{R}^{N\times D_{\text{v}}}$, $t \in [1, T]$ (*c.f.* Equation (2)). Note that here we define $\boldsymbol{u}_t$ as the patch-wise feature which does not contain the global `[CLS]` token features. We also omit the superscript of layer index l.

The vanilla self-attention first performs three different linear projections on the input feature $\boldsymbol{u}_t$ to obtain the triplet of query, key, and value.

$$\boldsymbol{q}_t = \boldsymbol{u}_t \cdot \mathbf{W}_q,\ \boldsymbol{k}_t = \boldsymbol{u}_t \cdot \mathbf{W}_k,\ \boldsymbol{v}_t = \boldsymbol{u}_t \cdot \mathbf{W}_v, \tag{4}$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{D_{\text{v}}\times D_{\text{v}}}$ are frozen transformation weights. Then the scaled dot-product attention is computed to achieve the contextual information.

The vanilla self-attention only attends to the intra-frame correlation modeling, which leads to the modality gap between video and image. Instead of introducing an additional serial or parallel temporal modeling module (temporal Transformer (Liu et al., 2023; Yang et al., 2022) or 3D Convolution (Pan et al., 2022)), we propose a novel asynchronous self-attention which introduces patch-wise temporal offset to model inter-frame relationship. Besides, to stabilize the training process, we propose a text-conditioned selection mechanism.

Text-conditioned Selection. Here we take the video-to-text retrieval as an example to illustrate this. For the given frame-wise video feature $\boldsymbol{f} \in \mathbb{R}^{T\times D_{\text{v}}}$, we conduct mean pooling on the frame dimension to obtain the video-level features $\overline{\boldsymbol{f}} \in \mathbb{R}^{1\times D_{\text{v}}}$. Then we select the most similar sentence $\boldsymbol{w}^* \in \mathcal{W}$ as follows.

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}\in\mathcal{W}}{\arg\max}\left(\text{Proj}(\overline{\boldsymbol{f}}) \cdot \boldsymbol{w}^{\intercal}\right), \tag{5}$$

where $\boldsymbol{w} \in \mathbb{R}^{1\times D_{\text{t}}}$ is the candidate sentence features. $\text{Proj}(\cdot)$ is a linear projection layer to transform the visual dimension D_{v} to the textual dimension D_{t}.

Then, we compute the sentence-patch similarity and select the top K responded patches.

$$\mathcal{S}_t = \underset{t\in[1,T]}{\arg\text{topk}}\left(\text{Proj}(\boldsymbol{u}_t) \cdot \boldsymbol{w}^{*\intercal}\right), \tag{6}$$

where $\mathcal{S}_t$ is the filtered patch index set.

Asynchronous Self-Attention. Then we only apply the proposed asynchronous self-attention on patches indexed by the set of $\mathcal{S}_t$. Specifically, the query features are adapted as follows.

$$\hat{\boldsymbol{k}}_t^n, \hat{\boldsymbol{v}}_t^n = \begin{cases} \boldsymbol{k}_{t+\boldsymbol{\delta}_t}^{n+\gamma_n}, \boldsymbol{v}_{t+\boldsymbol{\delta}_t}^{n+\gamma_n}, & n \in \mathcal{S}_t \\ \boldsymbol{k}_t^n, \boldsymbol{v}_t^n, & n \notin \mathcal{S}_t \end{cases} \tag{7}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^{N\times 1}$, $\boldsymbol{\delta} \in \mathbb{R}^{T\times 1}$ are layer-shared learnable parameters representing the offset distance in the patch and temporal dimension, respectively. $\boldsymbol{k}_{t+\boldsymbol{\delta}_t}^{n+\gamma_n}$ and $\boldsymbol{v}_{t+\boldsymbol{\delta}_t}^{n+\gamma_n}$ denote the key and value features of the $(n+\gamma_n)^{\text{th}}$ patch in the $(t+\delta_t)^{\text{th}}$ frame, respectively. $\hat{\boldsymbol{k}}_t, \hat{\boldsymbol{v}}_t \in \mathbb{R}^{N\times D_{\text{v}}}$ represents the adapted features. Finally, asynchronous self-attention is computed as follows.

$$\text{Atten}(\boldsymbol{q}_t, \hat{\boldsymbol{k}}_t, \hat{\boldsymbol{v}}_t) = \text{softmax}\left(\frac{\boldsymbol{q}_t \hat{\boldsymbol{k}}_t^{\intercal}}{\sqrt{D_{\text{v}}}}\right)\hat{\boldsymbol{v}}_t, \tag{8}$$

where $\boldsymbol{q}_t$ is illustrated in Equation (4) while $\hat{\boldsymbol{k}}_t$ and $\hat{\boldsymbol{v}}_t$ are defined in Equation (7).

4 Experiments

4.1 Experimental Settings

Datasets. We validate the performance of our proposed RAP on four benchmarked datasets. 1) **MSR-VTT** (Xu et al., 2016) contains 10,000 YouTube videos and each video is associated with 20 textual descriptions. We follow the 1k-A split (Yu et al., 2018) where 9,000 videos are used for training and 1,000 videos for testing. 2) **MSVD** (Chen and Dolan, 2011) is composed of 1,970 videos. Following the official split, we used 1,200 videos for

Table 1: **Comparisons with state-of-the-art methods on MSR-VTT dataset.** 🔒 denotes using the frozen visual encoder. RAP* denotes the RAP model with DSL post-processing (Cheng et al., 2021). 🔓 refers to the text-encoder being trainable. The best performance is in **bold** and the second best is underlined.

		Trainable	Text → Video				Video → Text			
Type	Methods	Params (MB) ↓	R@1↑	R@5↑	R@10↑	MnR↓	R@1↑	R@5↑	R@10↑	MnR↓
with CLIP-ViT-B/32										
Fine-tune	CLIP4Clip (Luo et al., 2022)	151.28	42.6	70.8	79.9	16.1	43.9	70.0	81.4	11.7
	CLIP4Clip (🔒 CLIP)	0	31.1	53.7	63.4	41.6	26.5	50.1	61.7	39.9
Prompt	VPT (Jia et al., 2022)	0.21	37.5	63.0	73.9	21.6	36.5	62.8	74.3	20.0
	VPT 🔓 (Jia et al., 2022)	63.43	40.5	67.3	78.6	17.9	40.9	70.0	79.2	12.5
	CoOp (Zhou et al., 2022)	**0.02**	38.3	62.3	73.4	18.9	41.0	66.6	77.4	13.4
	VoP^P (Huang et al., 2023)	0.50	40.1	65.7	77.7	16.9	42.5	70.0	79.9	12.4
	VoP^C (Huang et al., 2023)	14.30	40.8	68.1	79.0	15.8	42.3	70.1	81.1	11.4
Adapter	ST-Adapter (Pan et al., 2022)	14.22	39.5	65.1	74.2	20.0	37.1	64.5	75.9	19.7
	ST-Adapter 🔓 (Pan et al., 2022)	77.45	42.5	70.0	80.1	17.0	42.1	70.0	81.2	11.4
	LoRA (Hu et al., 2021)	0.49	40.5	67.1	78.9	16.4	42.1	70.0	79.8	13.5
	SSF (Lian et al., 2022)	0.34	40.8	68.2	78.6	17.0	42.0	68.6	80.2	13.2
	RAP (Ours)	1.06	**44.8**	**71.4**	**81.5**	**14.4**	**44.0**	**71.9**	**82.4**	**10.1**
with CLIP-ViT-B/16										
	CLIP4Clip (Luo et al., 2022)	149.62	45.4	72.1	81.1	14.5	44.9	72.2	81.8	10.4
	VoP^P (Huang et al., 2023)	0.50	43.9	70.0	80.9	12.9	-	-	-	-
	VoP^C (Huang et al., 2023)	14.30	44.6	71.8	80.2	14.6	-	-	-	-
	MV-Adapter (Zhang et al., 2023)	3.87	46.0	72.0	82.1	-	45.6	74.0	83.8	-
	RAP (Ours)	1.06	46.5	73.9	82.0	12.1	45.3	76.4	84.8	9.1
	RAP* (Ours)	1.06	**52.1**	**77.3**	**86.7**	**10.0**	**51.6**	**78.7**	**86.9**	**8.0**

training and 670 videos for testing, respectively. 3) **ActivityNet Captions** (Krishna et al., 2017) covers 20,000 untrimmed videos of complex human activities with an average duration of two minutes. We report results on the "val1" split (10,009 training videos and 4,917 testing videos) as in (Gabeur et al., 2020). 4) **DiDemo** (Anne Hendricks et al., 2017) consists of 10,464 unedited, personal videos in diverse visual settings annotated with 40,543 text descriptions. We follow the training and evaluation protocol in (Luo et al., 2022).

Evaluation Metrics. Following the previous work (Luo et al., 2022), we evaluate the performance with standard retrieval metrics: recall at rank K (R@K, higher is better), median rank (MdR, lower is better) and mean rank (MnR, lower is better). R@K is defined as the percentage of samples for which the correct result is found in the top-K retrieved results. We set K to $\{1, 5, 10\}$ in our experiments. MdR calculates the median of the ground-truth results in the ranking while MnR calculates the mean rank of all the correct results.

Implementation Details. We set the input frame length to 12, 64, 12, 64 and the caption token length to 32, 64, 32, 64 for MSR-VTT, DiDeMo, MSVD, and ActivityNet Captions, respectively. The pre-trained CLIP (Radford et al., 2021) was adopted as the video and text encoders. BertAdam was used as the optimizer, with 0.1 proportion warm-up cosine annealing, and a learning rate of 1e-4. All the models were trained for 5 epochs except on DiDeMo which was fine-tuned with 10 epochs. The temporal rank R and the number of selected tokens K were both set to 3. All experiments were carried out on 4 NVIDIA Tesla A100 GPUs.

4.2 Comparisons with State-of-the-Arts

The comparison results are summarized in Table 1 and Table 2. Specifically, we set three sets of comparison experiments: **1)** Fine-tuning: We take the fully fine-tuned CLIP4clip (Luo et al., 2022) for comparisons. Besides, we also list the zero-shot performance of CLIP4clip, *i.e.*, 🔒 CLIP in Table 1, for comparisons; **2)** Prompt-tuning: We compare our proposed RAP to prompt-tuning methods including CoOp (Zhou et al., 2022), VPT (Jia et al., 2022) and VoP (Huang et al., 2023). Since VPT is tailored for purely visual tasks, we experiment by fine-tuning or freezing the textual branch of CLIP, respectively; **3)** Adapter: We conduct experiments with the state-of-the-art adapters including ST-Adapter (Pan et al., 2022), LoRA (Hu et al., 2021) and SSF (Lian et al., 2022). Notably, ST-Adapter is applied on the visual branch and the textual branch is either fine-tuned or freezed. For the experiments with CoOP, we insert 32 learnable prompt tokens at the input of the textual encoder.

The comparison results demonstrate the superior performance of our proposed RAP. For example,

Table 2: **Comparisons with state-of-the-art methods on DiDeMo, MSVD, and ActivityNet Datasets. 🔒** denotes using the frozen visual encoder. RAP* denotes the RAP model with DSL post-processing (Cheng et al., 2021).

		DiDeMo				MSVD				ActivityNet			
Type	Methods	R@1↑	R@5↑	R@10↑	MnR↓	R@1↑	R@5↑	R@10↑	MnR↓	R@1↑	R@5↑	R@10↑	MnR↓
Fine-tune	CLIP4Clip (Luo et al., 2022)	42.3	69.1	78.2	18.6	45.5	75.4	84.1	10.3	39.4	71.1	83.3	7.9
	CLIP4Clip (🔒 CLIP)	26.8	52.7	62.7	47.0	36.6	64.5	73.9	20.4	21.6	46.5	60.3	37.6
Prompt	VPT (Jia et al., 2022)	32.6	59.7	71.3	30.3	40.8	69.8	79.8	13.7	27.8	56.0	70.0	20.2
	CoOp (Zhou et al., 2022)	29.7	56.9	67.9	34.9	38.9	69.2	78.9	14.0	29.1	57.3	72.2	14.2
	VoP^P (Huang et al., 2023)	38.9	67.7	78.1	17.2	-	-	-	-	32.8	62.3	75.4	12.3
	VoP^C (Huang et al., 2023)	40.0	68.0	78.5	18.3	-	-	-	-	32.6	62.5	76.5	12.0
Adapter	ST-Adapter (Pan et al., 2022)	36.6	63.4	72.0	26.7	42.5	72.0	81.7	12.4	29.8	59.5	73.7	14.5
	LoRA (Hu et al., 2021)	38.4	65.9	75.7	22.6	45.1	75.0	84.0	10.8	27.7	55.8	69.3	18.8
	SSF (Lian et al., 2022)	38.3	65.8	77.7	21.8	43.9	73.3	82.8	11.2	33.2	63.6	77.0	11.3
	RAP (Ours)	42.6	70.4	79.6	18.0	44.9	73.7	83.1	11.1	40.8	71.0	82.2	8.3
	RAP* (Ours)	**47.1**	**74.1**	**82.4**	**13.9**	**49.8**	**78.2**	**86.1**	**9.7**	**48.4**	**76.2**	**86.4**	**7.0**

Table 3: **Comparisons of the memory footprint and GFLOPs.** The input frame number is set to 12 and the ViT-B/32 is employed as the backbone. 🔓 refers to the text-encoder being trainable.

Method	#Params (M)	Memory (G)	GFLOPs	R@1
CLIP4clip	151.3	12.9	**54.4**	42.6
ST-Adapter	14.2	10.3	62.8	39.5
ST-Adapter 🔓	77.5	11.2	62.8	42.5
LoRA	0.5	**9.5**	67.6	40.5
SSF	**0.3**	17.1	54.5	40.8
RAP_light (Ours)	0.4	12.2	55.3	**43.2**

on the MSR-VTT dataset, our RAP surpasses the fully fine-tuned CLIP4clip by 2.2% (42.6 *vs.* 44.8) on R@1 with only 0.7% parameters (1.06 M *vs.* 151.28 M) using CLIP-ViT-B/32 backbone. Besides, we also achieve superior performance compared to current prompt-tuning and adapter-tuning methods. Although the parameters of our RAP are slightly higher than LoRA and SSF, considering the considerable performance improvement, our RAP strikes a better balance between parameters and performance.

Besides, to further probe the memory usage and computational complexity of the proposed model, we summarize the GPU memory usage during the training process and GFLOPs of the model in Table 3. For fair comparisons, we coequally set the number of input frames of each model to 12 frames and experiment with the ViT-B/32 backbone. We set up a lightweight RAP which only applies LoRM and ASA at the last four layers. As shown, compared to the fully fine-tuned Clip4clip, RAP_light remarkably reduces the tunable parameters, slightly lowers the memory footprint and boosts the performance. In brief, our RAP_light achieves the balance between computational overhead and performance, *i.e.*, paying affordable overhead while obtaining considerable performance gains.

Table 4: **Ablations of model components of RAP.**

Mode	LoRM	ASA	R@1	R@5	R@10	#Params (M)
#1	✓	✓	**44.8**	**71.4**	**81.5**	1.06
#2	✓	✗	43.3	70.9	81.8	0.76
#3	✗	✓	42.5	70.1	80.3	0.64
#4	✗	✗	40.8	68.2	78.6	**0.34**

Table 5: **Ablations of decomposition manners.** ∅ denotes RAP without any variants of LoRM. “T", “S" and “L" represent temporal, spatial, and layer, respectively.

	Mode	R@1	R@5	R@10	MdR	MnR
#1	∅	40.8	68.2	78.6	2.0	17.0
#2	T	**43.3**	**70.9**	**81.8**	**2.0**	**14.7**
#3	S-T	43.2	69.4	80.7	2.0	15.1
#4	S-T-L	42.0	67.8	80.3	2.0	14.5

4.3 Ablations Study

We conduct all the ablation studies on the MSR-VTT dataset with the ViT-B/32 backbone. The input frame number is set to 12.

Component Ablations. We ablate the proposed low-rank modulation module and the asynchronous self-attention. The results are summarized in Table 4. We can conclude that both components are crucial to superior performance at the cost of negligible parameter overhead. For example, LoRM yields a 2.3% performance boost on R@1 with the cost of 0.42 M parameter (mode #1 *vs.* mode #3).

Ablations on the low-rank decompose manner of LoRM. In Equation (1), we conduct the low-rank decomposition in the temporal dimension, and the modulation weights are with the dimension of $\mathbb{R}^{T\times D_\mathrm{v}}$, *i.e.*, $\mathbb{R}^{T\times D_\mathrm{v}} \leftarrow \mathbb{R}^{T\times R}\cdot\mathbb{R}^{R\times D_\mathrm{v}}$. Here we ablate more decomposition options: ***i)*** the spatial-temporal decomposition: The modulation is applied at the spatial-temporal dimension with the weight of $\mathbb{R}^{T\times N\times D_\mathrm{v}}$, *i.e.*, $\mathbb{R}^{T\times N\times D_\mathrm{v}} \leftarrow \mathbb{R}^{T\times N\times R}\cdot\mathbb{R}^{R\times D_\mathrm{v}}$, where T and N denote and

Table 6: **Ablations of selection manners.** ∅ indicates that none of the token selection policies is used.

Mode	R@1	R@5	R@10	MdR	MnR
text-top-K	**44.8**	**71.4**	**81.5**	**2.0**	**14.4**
text-bottom-K	43.0	70.7	80.3	2.0	14.8
vision-top-K	44.5	71.3	80.7	2.0	14.8
vision-bottom-K	43.5	70.6	80.3	2.0	15.1
random	43.2	70.8	81.2	2.0	14.9
∅	41.4	68.9	79.9	2.0	15.7

frame number and the patch number within each frame, respectively. ***ii)*** the spatial-temporal-layer decomposition: We uniformly decompose all the modulation weights across all the layers. Specifically, the modulation weights are of the shape of $\mathbb{R}^{M\times T\times N\times D_\text{v}}$, *i.e.*, $\mathbb{R}^{M\times T\times N\times D_\text{v}} \leftarrow \mathbb{R}^{M\times R} \cdot \mathbb{R}^{R\times T\times N\times R} \cdot \mathbb{R}^{R\times D_\text{v}}$, where M denotes the inserted module number of all layers.

The comparison results are summarized in Table 5. From the comparison results, we observe that using temporal decomposition alone brings about the optimum performance. Additionally introducing decomposition on the spatial and layer-wise dimension leads to the performance degrade. These results manifest our motivation that video data exhibits a substantial degree of redundancy in the temporal dimension.

Ablations on the text-conditioned selection manners. To stabilize the training process of ASA, we propose a text-conditioned selection strategy to constrain the asynchronous attention computation within the selected top-related patch features (*c.f.* Sec.3.3). For clarity, we denote this filter manner as *text-top-K*. Here we experiment with more visual token selection manners: ***i)*** *random*: randomly select K patch feature within each frame; ***ii)*** *text-bottom-K*: For each patch token feature, we compute the sentence-patch similarity and select lowest K responded patches; ***iii)*** *vision-top-K*: Instead of using sentence features, we compute the similarities between each patch feature and the `[CLS]` token feature of the frame. The filtered set is constituted by selecting the top K responsive patches; ***iv)*** *vision-bottom-K*: Similar to *vision-top-K*, we compute patch-wise similarities with `[CLS]` token and select lowest K responded patches; ***v)*** ∅: none of the selection strategies are used and all the patch features are wrapped.

The comparison results of the above selection strategies are summarized in Table 6. We have the following findings. Firstly, not using the token selection strategy (*i.e.*, ∅ in Table 6) causes substantial performance degradation, *e.g.*, reaching only 41.4% on R@1. This is probably because warping each patch tokens wreaks havoc on the well-trained CLIP weights. Secondly, our proposed *text-top-K* policy outperforms the other ones on all five metrics. This demonstrates that selectively warping partial patch tokens in a parameterized way can better adapt the vanilla CLIP to the video scenario.

Table 7: **Ablations of warping in Asynchronous attention.** T-Warp and S-Warp denote warping only on the temporal and spatial dimensions, respectively.

T-Warp	S-Warp	R@1	R@5	R@10	MdR	MnR
✓	✓	**44.8**	**71.4**	**81.5**	**2.0**	**14.4**
✗	✓	44.0	70.4	81.4	2.0	14.8
✓	✗	44.2	70.9	81.2	2.0	14.8

Table 8: **Ablations on hyper-parameters** including the temporal rank R and the number of selected token K.

R	1	3	5	7	9
R@1	42.6	**44.8**	44.0	43.2	43.0
K	1	3	5	7	9
R@1	44.5	**44.8**	43.8	43.6	42.9

Ablations on the warping manner of ASA. In Sec. 3.3, we predict the patch-wise warping distance in both the temporal and spatial dimensions. Here we ablate either of the two dimensions to see the difference. As shown in Table 7, restricting warping in either temporal or spatial dimension will lead to performance degradation, which demonstrates that free-form patch-wise warping is crucial to the final performance.

Ablations on hyper-parameters. We conduct ablation studies on the temporal rank R and selected token number K in Table 8. We set $R = 3$ and $K = 3$ to achieve the best retrieval performance.

5 Conclusions

In this work, we present RAP to efficiently transfer the pre-trained CLIP model to TVR. To accommodate the inherent video structure and the cross-modality setting, we introduce a low-rank modulation module to achieve the frame-wise sparse representation and an asynchronous self-attention module to enhance the cross-frame correlations. Extensive experiments illustrate that RAP achieves comparable or even better performance than previous arts and the fully fine-tuned counterpart.

Impact Statements

Ethics Statement. Our RAP aims to conduct parameter-efficient text-video retrieval through a temporally sparse and correlated adapter. The ethical issues may exist in the following two perspectives. Firstly, similar to many data-driven methods, there are concerns about the issue of data privacy, anonymization, and compliance with relevant data protection regulations. Secondly, the considerations related to potential bias in the dataset and the retrieval model, especially concerning sensitive topics, should be acknowledged. We are transparent about the ethical considerations in our research to uphold the integrity of the academic process and to ensure that this work aligns with ethical standards and norms in the field.

Limitation. Despite the remarkable progress, our RAP still faces several limitations. Firstly, we use the text-conditioned selection to filter the most representative visual patches. Due to the semantic gap conveyed by the textual and visual signals, the alignment of complex concepts and contexts across different modalities should be conducted in a more fine-grained manner. Secondly, due to the limitations of computing resources, we experiment with the backbone of ViT-B/32 and ViT-B/16. The salable experiments on ViT-L/14 and ViT-E/14 backbones are left for future work.

References

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812.

Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. 2022. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv preprint arXiv:2203.17274*, 3:11–12.

David Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200.

Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. 2020. Fine-grained video-text retrieval with hierarchical graph reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10638–10647.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. 2022a. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. 2022b. Vision transformer adapter for dense predictions. In *The Eleventh International Conference on Learning Representations*.

Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen. 2021. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. *arXiv preprint arXiv:2109.04290*.

Ioana Croitoru, Simion-Vlad Bogolin, Marius Leordeanu, Hailin Jin, Andrew Zisserman, Samuel Albanie, and Yang Liu. 2021. Teachtext: Crossmodal generalized distillation for text-video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11583–11593.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-modal transformer for video retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 214–229. Springer.

Satya Krishna Gorti, Noël Vouitsis, Junwei Ma, Keyvan Golestan, Maksims Volkovs, Animesh Garg, and Guangwei Yu. 2022. 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 5006–5015.

Tanveer Hannan, Md Mohaiminul Islam, Thomas Seidl, and Gedas Bertasius. 2023. Rgnet: A unified retrieval and grounding network for long videos. *arXiv preprint arXiv:2312.06729*.

Feng He, Qi Wang, Zhifan Feng, Wenbin Jiang, Yajuan Lü, Yong Zhu, and Xiao Tan. 2021a. Improving video retrieval by adaptive margin. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1359–1368.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021b. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Siteng Huang, Biao Gong, Yulin Pan, Jianwen Jiang, Yiliang Lv, Yuyuan Li, and Donglin Wang. 2023. Vop: Text-video co-operative prompt tuning for cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6565–6574.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer.

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

Shibo Jie and Zhi-Hong Deng. 2022. Convolutional bypasses are better vision transformer adapters. *arXiv preprint arXiv:2207.07039*.

Peng Jin, Jinfa Huang, Fenglin Liu, Xian Wu, Shen Ge, Guoli Song, David Clifton, and Jie Chen. 2022. Expectation-maximization contrastive learning for compact video-and-language representations. *Advances in Neural Information Processing Systems*, 35:30291–30306.

Peng Jin, Jinfa Huang, Pengfei Xiong, Shangxuan Tian, Chang Liu, Xiangyang Ji, Li Yuan, and Jie Chen. 2023a. Video-text as game players: Hierarchical banzhaf interaction for cross-modal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2472–2482.

Peng Jin, Hao Li, Zesen Cheng, Kehan Li, Xiangyang Ji, Chang Liu, Li Yuan, and Jie Chen. 2023b. Diffusionret: Generative text-video retrieval with diffusion model. *arXiv preprint arXiv:2303.09867*.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. 2021. 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.

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

Xinhao Li and Limin Wang. 2023. Zeroi2v: Zero-cost adaptation of pre-trained transformers from image to video. *arXiv preprint arXiv:2310.01324*.

Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. 2022. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123.

Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2022. Frozen clip models are efficient video learners. In *European Conference on Computer Vision*, pages 388–404. Springer.

Ruyang Liu, Jingjia Huang, Ge Li, Jiashi Feng, Xinglong Wu, and Thomas H Li. 2023. Revisiting temporal modeling for clip-based image-to-video knowledge transferring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6555–6564.

Yuqi Liu, Pengfei Xiong, Luhui Xu, Shengming Cao, and Qin Jin. 2022. Ts2-net: Token shift and selection transformer for text-video retrieval. In *European Conference on Computer Vision*, pages 319–335. Springer.

Haoyu Lu, Mingyu Ding, Yuqi Huo, Guoxing Yang, Zhiwu Lu, Masayoshi Tomizuka, and Wei Zhan. 2023. Uniadapter: Unified parameter-efficient transfer learning for cross-modal modeling. *arXiv preprint arXiv:2302.06605*.

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

Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. 2022. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 638–647.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 565–576.

Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. 2022. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems*, 35:26462–26477.

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

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

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5227–5237.

Yi-Lin Sung, Varun Nair, and Colin A Raffel. 2021. Training neural networks with fixed sparse masks. *Advances in Neural Information Processing Systems*, 34:24193–24205.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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

Xiaohan Wang, Linchao Zhu, and Yi Yang. 2021. T2vlad: global-local sequence alignment for text-video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5079–5088.

Yimu Wang and Peng Shi. 2023. Video-text retrieval by supervised multi-space multi-grained alignment. *arXiv preprint arXiv:2302.09473*.

Ziyang Wang, Yi-Lin Sung, Feng Cheng, Gedas Bertasius, and Mohit Bansal. 2023. Unified coarse-to-fine alignment for video-text retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2816–2827.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296.

Jianwei Yang, Yonatan Bisk, and Jianfeng Gao. 2021. Taco: Token-aware cascade contrastive learning for video-text alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11562–11572.

Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. 2022. Aim: Adapting image models for efficient video action recognition. In *The Eleventh International Conference on Learning Representations*.

Huanjin Yao, Wenhao Wu, and Zhiheng Li. 2023. Side4video: Spatial-temporal side network for memory-efficient image-to-video transfer learning. *arXiv preprint arXiv:2311.15769*.

Haoxuan You, Luowei Zhou, Bin Xiao, Noel Codella, Yu Cheng, Ruochen Xu, Shih-Fu Chang, and Lu Yuan. 2022. Learning visual representation from modality-shared contrastive language-image pre-training. In *European Conference on Computer Vision*, pages 69–87. Springer.

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

Youngjae Yu, Jongseok Kim, and Gunhee Kim. 2018. A joint sequence fusion model for video question answering and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 471–487.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. 2021. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9.

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

Zhang Zhang and Dacheng Tao. 2012. Slow feature analysis for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 34(3):436–450.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348.

A Appendix

A. More Experiments

Table 9: **Video-to-text retrieval results on DiDeMo, MSVD, and ActivityNet Datasets.** 🔒 denotes using the frozen visual encoder.

		DiDeMo				MSVD				ActivityNet			
Type	Methods	R@1↑	R@5↑	R@10↑	MnR↓	R@1↑	R@5↑	R@10↑	MnR↓	R@1↑	R@5↑	R@10↑	MnR↓
Fine-tune	CLIP4Clip (Luo et al., 2022)	42.2	70.3	79.3	11.8	56.6	79.7	84.3	7.6	41.9	72.2	84.6	7.3
	CLIP4Clip (🔒 CLIP)	20.2	44.2	55.0	43.1	56.3	82.6	89.8	4.8	17.7	40.7	54.1	42.5
Prompt	VPT (Jia et al., 2022)	33.1	59.8	69.9	22.7	59.5	82.9	88.8	5.9	28.4	56.7	69.4	19.7
	CoOp (Zhou et al., 2022)	32.3	57.0	68.2	23.4	53.8	78.3	82.9	12.4	29.0	57.6	72.4	14.0
	VoP^P (Huang et al., 2023)	40.6	68.3	78.6	11.6	-	-	-	-	34.8	65.0	78.2	10.7
	VoP^C (Huang et al., 2023)	39.1	65.3	76.7	13.8	-	-	-	-	34.2	64.8	78.4	10.7
Adapter	ST-Adapter (Pan et al., 2022)	35.9	61.0	72.0	20.1	53.6	80.5	87.3	5.8	29.7	58.8	73.1	15.5
	LoRA (Hu et al., 2021)	39.7	66.8	77.3	13.9	64.3	87.3	92.5	4.1	30.8	60.0	73.2	15.2
	SSF (Lian et al., 2022)	40.0	67.1	77.4	13.2	61.9	87.0	90.7	4.5	36.2	66.9	79.0	10.4
	RAP (Ours)	44.0	69.2	80.1	10.5	65.2	88.7	93.1	4.2	41.9	73.0	84.0	7.5
	RAP* (Ours)	**47.7**	**74.4**	**83.2**	**9.5**	**69.6**	**91.9**	**95.7**	**2.9**	**48.2**	**76.5**	**86.2**	**6.8**

Video-to-text Performance: We supplement the video-to-text performance on the DiDeMo, MSVD and ActivityNet Captions datasets in Table 9. The experimental results consistently demonstrate the superiority of our RAP. For example on MSVD dataset, RAP surpasses the fully fine-tuned method by 8.6% on R@1.

Low-rank modulation on textual features: In Sec. 3.2, we apply the low-rank decomposition modulation in the visual branch, specifically in the temporal dimension. Here we apply the low-rank modulation on the textual branch to see the differences. Concretely, the modulation weights are with the shape of $\mathbb{R}^{W \times D_\mathrm{t}} \leftarrow \mathbb{R}^{W \times R_\mathrm{t}} \cdot \mathbb{R}^{R_\mathrm{t} \times D_\mathrm{t}}$, where W denotes the total word length, R_t is the low-rank hyper-parameter and D_t is the textual feature dimension. We set $R_\mathrm{t} = 3$.

The ablation results are shown in Table 10. As shown, applying the low-rank modulation on textual features causes performance degradation, which may be because word-level representations do not exhibit the same redundancy as frame-level visual features.

Table 10: **Ablations of the low-rank modulation on the textual branch.**

LoRM on Text	R@1	R@5	R@10	MnR	#Param
✗	**44.8**	**71.4**	**81.5**	**14.4**	**1.06M**
✓	44.3	72.1	81.0	14.4	1.48M

B. Visualization Results

We visualize the per-frame modulation weights with or without the low-rank decomposition. As shown in Figure 4, the modulation weights with decomposition demonstrate more salient distributions, which manifests the temporal sparsity characteristic of video data.

Besides, we visualize the effect of the asynchronous self-attention. We randomly select one patch feature (✚) in the frame and computes its similarity distribution with the patches in other frames. The results in Figure 5 show that the proposed asynchronous self-attention can adaptively attend to patch-of-interest, which leads to effective temporal correlation modeling.

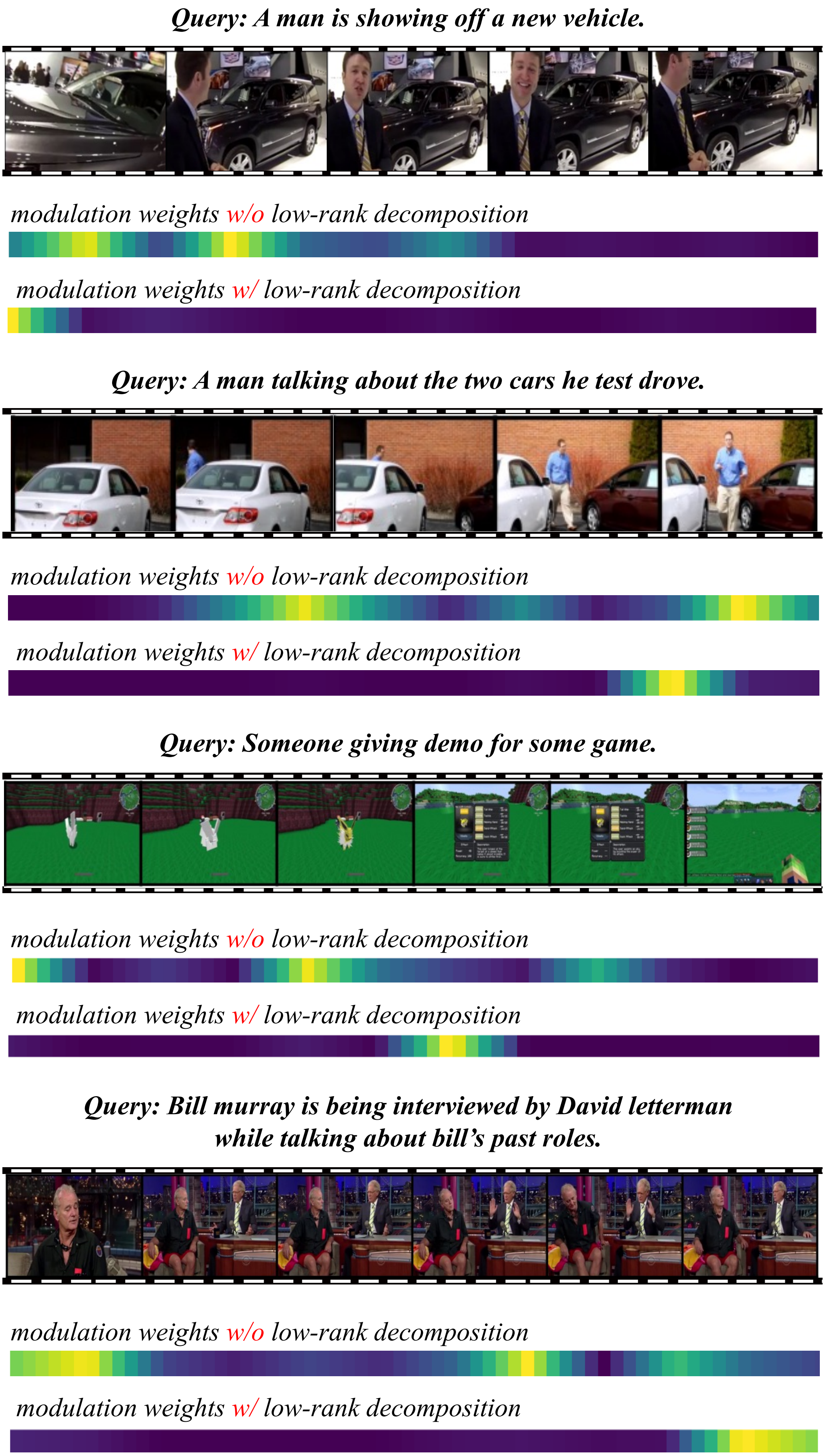

Figure 4: **Illustrations of temporal sparsity.** We visualize the modulation weight *w/* or *w/o* low-rank decomposition.

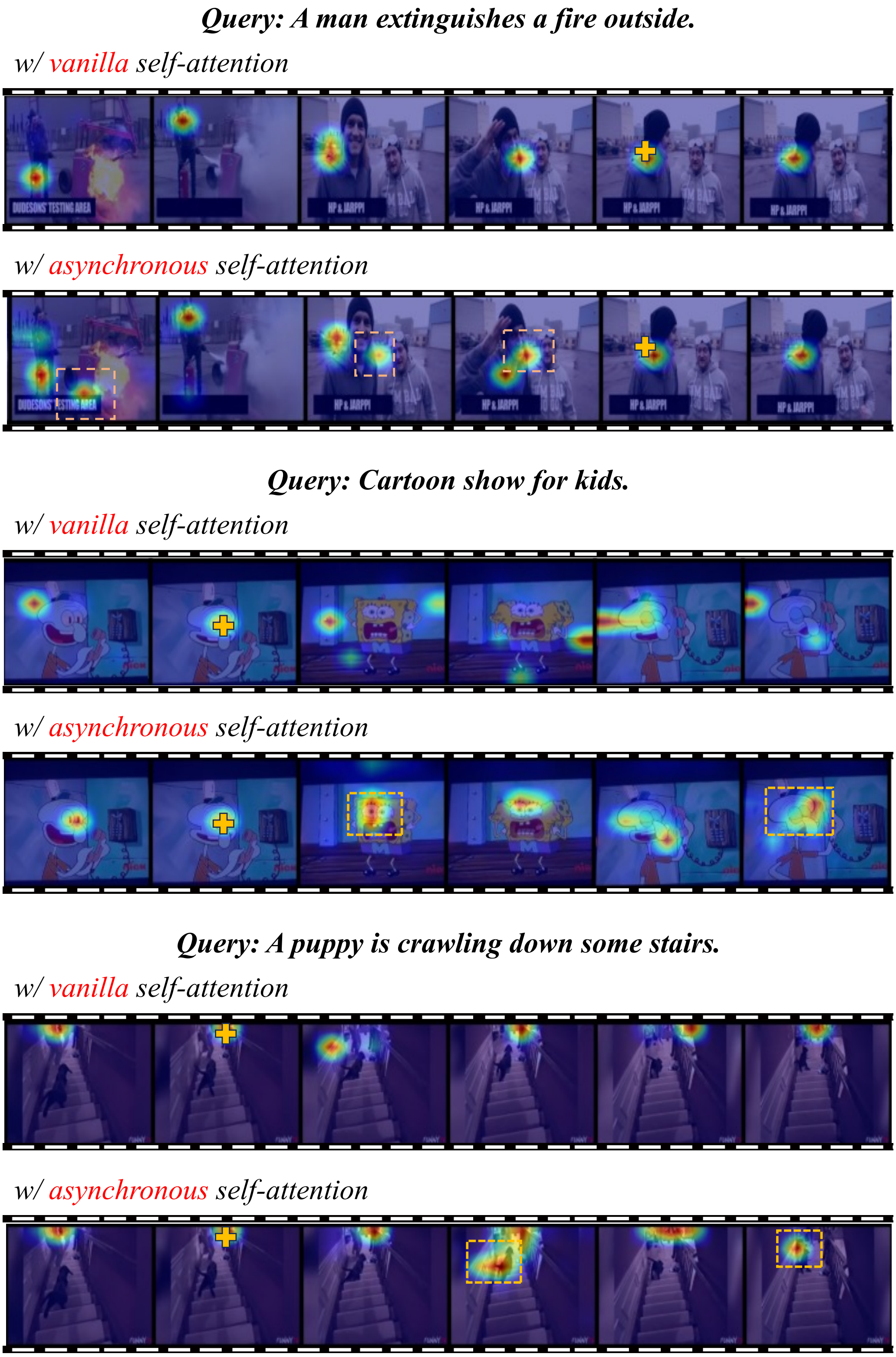
Query: A man extinguishes a fire outside.
w/ vanilla self-attention
w/ asynchronous self-attention
Query: Cartoon show for kids.
w/ vanilla self-attention
w/ asynchronous self-attention
Query: A puppy is crawling down some stairs.
w/ vanilla self-attention
w/ asynchronous self-attention

Figure 5: **Illustrations of temporal correlation.** The query patch is marked by the yellow cross and the similarity map within other frames are plotted.